



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

José Guedes dos Santos Júnior

**Método Robusto à Oclusão para Rastreamento 6-DOF de Objetos em Imagens
RGB-D com Otimização por Enxame de Partículas**

Recife

2022

José Guedes dos Santos Júnior

**Método Robusto à Oclusão para Rastreamento 6-DOF de Objetos em Imagens
RGB-D com Otimização por Enxame de Partículas**

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Doutor em Ciência da Computação.

Área de Concentração: Mídia e interação

Orientador (a): Veronica Teichrieb

Coorientador (a): João Paulo Silva do Monte Lima

Recife

2022

Catálogo na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

S237m Santos Júnior, José dos Guedes
Método robusto à oclusão para rastreamento 6-DOF de objetos em
imagens RGB-D com otimização por enxame de partículas / José dos Guedes
Santos Júnior. – 2022.
129 f.: il., fig.

Orientadora: Veronica Teichrieb.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da
Informação, Recife, 2022.

Inclui referências e apêndice.

1. Mídia e interação. 2. Otimização. I. Teichrieb, Veronica (orientadora). II.
Título.

006.7 CDD (23. ed.) UFPE - CCEN 2022-100

José Guedes dos Santos Júnior

**“Método Robusto à Oclusão para Rastreamento 6-DOF de Objetos em
Imagens RGB-D com Otimização por Enxame de Partículas”**

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação. Área de Concentração: Mídia e Interação

Aprovado em: 17/02/2022.

Orientadora: Profa. Dra. Veronica Teichrieb

BANCA EXAMINADORA

Prof. Dr. Silvio de Barros Melo
Centro de Informática / UFPE

Prof. Dr. Cleber Zanchettin
Centro de Informática / UFPB

Prof. Dr. João Paulo Papa
Departamento de Computação / UNESP

Prof. Dr. Cláudio Rosito Jung
Instituto de Informática / UFRGS

Prof. Dr. Péricles Barbosa Cunha de Miranda
Departamento de Computação / UFRPE

Dedico este trabalho a Maria Bernardete.

AGRADECIMENTOS

Agradeço à minha esposa Kaline pela compreensão e companheirismo ao longo desse curso, e aos meus filhos Heitor e Gabriel pelo carinho. Vocês com certeza são um dos principais motivos de eu ter chegado até aqui.

Agradeço a toda minha família, em especial à minha mãe, Maria Bernardete (Dona Beta), e às minhas quatro irmãs, Rosy, Diu, Gê e Jeane, por toda motivação durante esse caminho. “Nossos dias serão para sempre”.

Agradeço a todos do Voxar que contribuíram de forma direta ou indireta na minha formação e na realização deste trabalho, em especial à professora Veronica Teichrieb pela oportunidade de cursar o doutorado. Vocês são realmente incríveis, gostaria de ter convivido e trabalhado com vocês mais de perto.

Agradeço ao professor João Paulo pela dedicação e compromisso ao longo de toda a nossa pesquisa, admiro demais o seu profissionalismo em todos os momentos (fáceis ou difíceis) durante esse período. Você nunca foi formalmente meu professor, mas foi uma das pessoas que mais me ensinou no decorrer dessa jornada acadêmica, obrigado!

Agradeço aos meus colegas de trabalho da escola Arnaldo Alves Cavalcanti pelo o apoio dado quando foi necessário. Em especial à diretora Solange Morato pelo estímulo e pela disposição em me atender sempre que precisei.

Agradeço também ao amigo Cícero Renan por toda a ajuda disponibilizada quando estive longe de casa, e ao amigo Sérgio Fideles, uma das pessoas mais incríveis que já conheci, por sempre me receber e por toda ajuda que me deu quando mais precisei. Obrigado, sem vocês esta tarefa teria sido muito mais difícil.

RESUMO

Rastreadores visuais podem ser usados para determinar a trajetória de objetos 3D rígidos com 6 graus de liberdade em tempo real. Essa funcionalidade é necessária para vários tipos de aplicações, como em realidade aumentada ou robótica, e sua eficiência muitas vezes está relacionada com a acurácia, a robustez a falha e o desempenho de tempo desses rastreadores. Nos últimos anos, vários trabalhos contribuíram para a melhoria dessas técnicas, apresentando métodos acurados que são executados em tempo real, dentre os quais se destacam aqueles baseados em otimização, baseados em aprendizagem e híbridos, ou seja, que utilizam otimização e aprendizagem de forma cooperativa. Neste contexto, o presente trabalho propõe melhorias no rastreamento com 6 graus de liberdade de objetos 3D arbitrários que usa otimização por enxame de partículas. Dentre elas, destacam-se: o aperfeiçoamento da função de aptidão usando média harmônica das coordenadas 3D, cor e normais dos pontos; a seleção dinâmica da região de interesse na nuvem de pontos da cena; a filtragem dos pontos visíveis do modelo considerando a auto-occlusão e a oclusão por outros objetos da cena; o cálculo das fronteiras do subespaço de soluções em tempo de execução, observando a inércia do objeto alvo; a filtragem de poses em tempo de execução a partir da trajetória encontrada; e a implementação da otimização por enxame de partículas completamente em GPU. Experimentos mostraram que tais mudanças possibilitaram melhorias na robustez a falhas, na acurácia e no desempenho de tempo da técnica. Quando comparado com métodos presentes no estado da arte, o rastreador proposto nessa tese foi, em média, 19,3% e 16,3% mais acurado em relação aos erros de translação e rotação, respectivamente, e apresentou um número de falhas 78,4% menor. O rastreador proposto ainda foi de 5 a 7 vezes mais rápido que uma técnica existente baseada em otimização por enxame de partículas.

Palavras-chaves: rastreamento de objetos 3D; otimização por exame de partículas; imagens RGB-D.

ABSTRACT

Visual trackers can be used to determine the trajectories of rigid 3D objects with 6 degrees of freedom in real-time. This functionality is required for many types of applications, such as augmented reality or robotics, and its efficiency is commonly related to the accuracy, robustness to failure and time performance of these trackers. In the last years, several works contributed to the improvement of these techniques, presenting accurate methods that are executed in real-time, among which stand out optimization-based, learning-based, and hybrid methods, that is, that use optimization and learning cooperatively. In this context, the present work proposes several improvements in the tracking with 6 degrees of freedom of arbitrary 3D objects that uses particle swarm optimization. Among them, stand out: the improvement of the fitness function using the harmonic mean of 3d coordinate, color and normal vector of the points; the region of interest dynamic selection in the scene point cloud; filtering of the visible model points considering self-occlusion and occlusion by other objects in the scene; calculating solution subspace boundaries at runtime observing the target object inertia; filtering of the poses from the found trajectory in real-time and the implementation of the particle swarm optimization completely in GPU. Experiments have shown that such changes led to improvements in accuracy, robustness and execution time of the technique. When compared to a state-of-the-art learning-based technique, the tracker proposed in this thesis was, on average, 19.3% and 16.3% more accurate with respect to translation and rotation errors, respectively, and presented 78.4% less tracking fails. The proposed tracker was also 5 – 17 times faster than an existing particle swarm optimization-based technique.

Keywords: 3D object tracking; particle swarm optimization; RGB-D images.

LISTA DE FIGURAS

Figura 1 – Exemplos práticos do uso de rastreamento em aplicações com Realidade Aumentada (RA). Cenário de jornal televisivo (esquerda), aplicação educacional (centro) e transmissão de jogos esportivos (direita).	20
Figura 2 – Princípio físico da projeção de uma imagem em uma câmera de orifício. . .	25
Figura 3 – Sistemas de coordenadas do mundo e da câmera.	25
Figura 4 – Transformação do sistema de coordenadas do mundo para obter o sistema de coordenadas da câmera.	26
Figura 5 – Representação dos ângulos de Euler.	28
Figura 6 – Modelo matemático de uma câmera de orifício e algumas de suas principais características geométricas internas.	29
Figura 7 – Imagem RGB (esquerda) e o seu mapa de profundidade correspondente (centro). Dispositivo Microsoft Kinect v1 (direita), em destaque o projetor Infrared (IR), a câmera RGB e o sensor IR.	31
Figura 8 – Topologias do Particles Swarm Optimization (PSO): global ou <i>gbest</i> (à esquerda) e local ou <i>lbest</i> (à direita).	37
Figura 9 – Exemplos de marcadores pontuais e situações em que são utilizados. . . .	40
Figura 10 – Exemplos do uso de marcadores pontuais, para aquisição de expressões faciais (esquerda), do movimento do corpo humano (centro) e de valores de ground truth da trajetória de objetos.	41
Figura 11 – Marcador planar (esquerda), aplicação móvel de RA usando um marcador planar (centro) e Quick Response Code (QR code) (direita).	42
Figura 12 – Modelos articulados do corpo e de mãos humanas.	43
Figura 13 – Exemplo de um dos modelos usados e resultados do rastreamento de diferentes objetos obtidos pela técnica apresentada em Azad et al. (2011). . .	45
Figura 14 – Exemplos de modelos e resultados do rastreamento de dois objetos usando a técnica proposta em Choi e Christensen (2013) nos casos de teste com cenas reais. Os modelos coloridos transformados pela pose encontrada são projetados sobre a cena em tons de cinza.	45

Figura 15 – Exemplos de modelos e resultados obtidos pelo rastreador proposto em Krull et al. (2014). A silhueta em azul representa a pose do <i>ground truth</i> e a silhueta em verde a pose estimada.	46
Figura 16 – Exemplos de modelos usados (imagens à esquerda), duas imagens capturadas em diferentes instantes (ao centro), uma delas distorcida devido à velocidade da câmera, e os respectivos resultados do rastreamento simultâneo nesses dois quadros, realizado pela técnica proposta em Akkaladevi et al. (2016) (imagens à direita).	47
Figura 17 – Exemplos de modelos e resultados obtidos pela técnica proposta em Garon e Lalonde (2017) em dois casos de teste.	48
Figura 18 – Exemplo de modelo e de resultados obtidos pela técnica proposta em Júnior e Lima (2018) usando casos de teste com 40% oclusão na horizontal da base de dados apresentada em Garon e Lalonde (2017). Imagens RGB de entrada (acima) e resultados (abaixo).	49
Figura 19 – Exemplo de modelo e de resultados obtidos pela técnica proposta em Maroukhas et al. (2020) usando um dos casos de teste do cenário <i>interaction-hard</i> da base de dados de Garon, Laurendeau e Lalonde (2018).	50
Figura 20 – Representação das etapas da técnica de rastreamento abordada por este trabalho.	53
Figura 21 – Exemplos de modelos de objetos 3D da base de dados de Garon, Laurendeau e Lalonde (2018).	54
Figura 22 – Seleção de Region of Interest (ROI) em uma imagem RGB da cena a partir da transformação e projeção dos vértices da caixa delimitadora de eixo alinhado.	56
Figura 23 – Teste de visibilidade para filtragem de pontos oclusos por oclusão do tipo 1 e 2 (esquerda) ou oclusão do tipo 3 (direita).	59
Figura 24 – Exemplo do cálculo dos limites do subespaço de busca para um quadro q a partir da pose encontrada pelo rastreador no quadro $q - 1$ (valores inteiros usados apenas para propósitos didáticos).	61
Figura 25 – Correspondência entre um ponto $\mathbf{m}$ do modelo projetado e o seu par $\mathbf{m}'$ na projeção da cena.	63
Figura 26 – Renderização de poses encontradas pelo PSO sobre imagens RGB da cena para análise qualitativa.	67

Figura 27 – Modelos de objetos 3D disponíveis na base de dados de Garon, Laurendeau e Lalonde (2018).	72
Figura 28 – Modelos de objetos 3D disponíveis na base de dados de Choi e Christensen (2013).	73
Figura 29 – Comparação dos resultados usando os casos de teste do <i>Dragon</i> com 40% de oclusão na horizontal de Garon e Lalonde (2017). Imagens de entrada (1ª coluna), resultados da técnica proposta (2ª coluna), resultados de Garon e Lalonde (2017) (3ª coluna) e resultados de Tan et al. (2015) (4ª coluna).	78
Figura 30 – Comparação dos resultados usando os casos de teste do <i>Dragon</i> com 40% de oclusão na vertical de Garon e Lalonde (2017). Imagens de entrada (1ª coluna), resultados da técnica proposta (2ª coluna), resultados de Garon e Lalonde (2017) (3ª coluna) e resultados de Tan et al. (2015) (4ª coluna).	79
Figura 31 – Médias dos erros de translação e rotação do <i>rastreador 4</i> de todos os níveis de oclusão por caso de teste do cenário <i>occlusion</i> da base de dados Garon, Laurendeau e Lalonde (2018), sem os casos de teste com 0% de oclusão.	89
Figura 32 – Comparação dos resultados usando os casos de teste do <i>Shoe</i> com 60% de oclusão na horizontal de Garon, Laurendeau e Lalonde (2018). Imagens de entrada (1ª coluna), resultados da técnica proposta usando o <i>rastreador 4</i> (2ª coluna), resultados de Garon, Laurendeau e Lalonde (2018) usando <i>object-specific</i> (3ª coluna), resultados de Garon, Laurendeau e Lalonde (2018) usando <i>generic</i> (4ª coluna), resultados de Garon e Lalonde (2017) (5ª coluna) e resultados de Tan et al. (2015) (6ª coluna).	91
Figura 33 – Comparação dos resultados usando os casos de teste do <i>Dragon</i> com 75% de oclusão na horizontal de Garon, Laurendeau e Lalonde (2018). Imagens de entrada (1ª coluna), resultados da técnica proposta usando o <i>rastreador 4</i> (2ª coluna) e resultados de Júnior e Lima (2018) (3ª coluna)	92
Figura 34 – Aptidão média do enxame em experimentos com valores de 100, 200, 300 e 400 iterações para o critério de parada.	96
Figura 35 – Variância média da população em experimentos com valores de 100, 200, 300 e 400 iterações para o critério de parada.	98
Figura 36 – Aptidão da melhor partícula em experimentos com valores de 100, 200, 300 e 400 iterações para o critério de parada.	99

Figura 37 – Avaliação qualitativa das poses encontradas pelo PSO com 32 partículas e 100 iterações em relação às aquelas apresentadas pelo <i>ground truth</i> da base de dados, nos casos de teste <i>Dragon</i> 0% e <i>Dragon</i> 45% de oclusão.	103
Figura 38 – Avaliação qualitativa das poses encontradas pelo PSO com 32 partículas e 100 iterações em relação às aquelas apresentadas pelo <i>ground truth</i> da base de dados, no caso de teste <i>Dragon</i> 60% de oclusão.	104
Figura 39 – Avaliação qualitativa das poses encontradas pelo PSO com 32 partículas e 100 iterações em relação às aquelas apresentadas pelo <i>ground truth</i> da base de dados, no caso de teste <i>Watering can</i> 0% de oclusão.	104
Figura 40 – Transformação dos pontos do modelo <i>Dragon</i> sobre a nuvem de pontos da cena no caso de teste <i>Dragon</i> 0% de oclusão, segundo a pose encontrada pelo PSO (acima) e segundo a pose definida no <i>ground truth</i> da base de dados (abaixo).	105
Figura 41 – Transformação dos pontos do modelo <i>Watering can</i> sobre a nuvem de pontos da cena no caso de teste <i>Watering can</i> 0% de oclusão, segundo a pose encontrada pelo PSO (acima) e segundo a pose definida no <i>ground truth</i> da base de dados (abaixo).	106
Figura 42 – Comparação dos valores das aptidões das poses encontradas pelo PSO em relação aos valores das aptidões das poses definidas pelo <i>ground truth</i> no cenário <i>Stability</i> no caso de teste <i>Kinect stability-far-2</i>	108
Figura 43 – Comparação dos valores das aptidões, distâncias entre as coordenadas 3D, cores e normais das poses encontradas pelo PSO em relação aos respectivos valores das poses definidas pelo <i>ground truth</i> nos 400 primeiros quadros do casos de teste <i>Tide</i>	109
Figura 44 – Pontos dos modelos <i>Tide</i> e <i>Kinect box</i> transformados sobre os pontos das nuvens de pontos de suas respectivas cenas, segundo as poses do PSO e do <i>ground truth</i> . Nos destaques, e em verde, aparecem os pontos penalizados.	110
Figura 45 – Pontos dos modelos <i>Tide</i> e <i>Kinect box</i> transformados sobre os pontos das nuvens de pontos de suas respectivas cenas, segundo as poses do PSO e do <i>ground truth</i> , e usando o limiar $\theta > 110^\circ$ para seleção de pontos na borda. Nos destaques, e em verde, aparecem os pontos penalizados.	111

Figura 46 – Comparação dos valores das aptidões, distâncias entre as coordenadas 3D, cores e normais das poses encontradas pelo PSO em relação aos respectivos valores das poses definidas pelo *ground truth* nos 400 primeiros quadros do casos de teste *Tide*, usando o limiar $\theta > 110^\circ$ para seleção de pontos na borda. 112

LISTA DE TABELAS

Tabela 1 – Coeficientes fixos para filtragem linear com diferentes tamanhos de entradas.	66
Tabela 2 – Médias dos erros de rotação e translação nos casos de teste do <i>Dragon</i> com 40% de oclusão de Garon e Lalonde (2017) obtidas pelo método proposto (sem reinicialização do rastreador) em relação às técnicas propostas em Tan et al. (2015) e em Garon e Lalonde (2017) (com reinicialização a cada 15 quadros), melhores resultados em negrito.	80
Tabela 3 – Média e desvio padrão dos erros de rotação e translação nos casos de teste do <i>Dragon</i> com 40% de oclusão de Garon e Lalonde (2017) obtidas pela técnica proposta e por Júnior e Lima (2018).	80
Tabela 4 – Tempo médio para o rastreamento em um quadro e fps da técnica proposta em relação ao tamanho do enxame e o número de pontos do modelo. . . .	81
Tabela 5 – Médias dos erros de translação e rotação com diferentes configurações do rastreador proposto, agrupadas por níveis de oclusão e pela direção do aparato de oclusão (horizontal ou vertical), no cenário <i>occlusion</i> da base de dados de Garon, Laurendeau e Lalonde (2018).	85
Tabela 6 – Médias dos erros de translação e rotação com diferentes configurações do rastreador da técnica proposta, agrupadas por diferentes níveis de oclusão, no cenário <i>occlusion</i> da base de dados de Garon, Laurendeau e Lalonde (2018).	85
Tabela 7 – Médias dos erros de rotação e translação do <i>rastreador 4</i> por modelo nos casos de teste apenas com oclusão na horizontal (mais os casos de teste sem oclusão) do cenário <i>occlusion</i> da base de dados de Garon, Laurendeau e Lalonde (2018).	88
Tabela 8 – Comparação das médias dos erros de rotação e translação de diversas técnicas de rastreamento, avaliadas nos casos de teste do cenário <i>occlusion</i> da base de dados de Garon, Laurendeau e Lalonde (2018), considerando todos os 11 objetos alvo e em relação ao nível de oclusão (os melhores resultados aparecem em negrito).	89

Tabela 9 – Comparação entre o número de falhas e as médias de erros de translação e rotação de diferentes técnicas de rastreamento (os melhores resultados estão em negrito).	93
Tabela 10 – Comparação entre o somatório do número de falhas de cada rastreador considerando todas as 11 sequências do caso de teste <i>interaction-hard</i> do cenário <i>interaction</i> da base de dados de Garon, Laurendeau e Lalonde (2018) (melhor resultado em negrito).	93
Tabela 11 – Comparação entre o grau de <i>jitter</i> de cada rastreador no cenário <i>stability</i> de Garon, Laurendeau e Lalonde (2018) (melhores resultados em negrito).	94
Tabela 12 – Comparação do tempo de processamento entre técnicas de rastreamento 3D baseadas em PSO (melhores resultados em negrito) usando os modelos Dragon e Lego e o caso de teste <i>interaction-hard</i> do cenário <i>interaction</i> da base de dados de Garon, Laurendeau e Lalonde (2018).	94
Tabela 13 – Erro médio de rotação e translação nos experimentos com 100, 200, 300 e 400 iterações.	100
Tabela 14 – Comparação entre as poses encontradas pelo PSO com 32 partículas e 100 iterações em relação àquelas apresentadas pelo <i>ground truth</i> da base de dados nos casos de teste <i>Dragon 0%</i> , <i>Dragon 45%</i> e <i>Dragon 60%</i> de oclusão na vertical (melhores resultados em negrito).	101
Tabela 15 – Comparação entre as poses encontradas pelo PSO com 128 partículas e 250 iterações em relação àquelas apresentadas pelo <i>ground truth</i> da base de dados, no caso de teste <i>Dragon 0%</i> de oclusão (melhores resultados em negrito).	102
Tabela 16 – Comparação entre as poses encontradas pelo PSO com 32 partículas e 100 iterações em relação àquelas apresentadas pelo <i>ground truth</i> da base de dados, no caso de teste <i>Watering can 0%</i> de oclusão (melhores resultados em negrito).	102
Tabela 17 – Comparação entre as poses encontradas pelo PSO com 32 partículas e 100 e àquelas definidas pelo <i>ground truth</i> em relação às poses do PSO modificadas (PSO/GT), no caso no caso de teste <i>Watering can 0%</i> de oclusão (melhores resultados em negrito).	107

Tabela 18 – Comparação do Root Mean Square (RMS) dos erros obtidas pelas diferentes versões do PSO nos 4 casos de testes sintéticos da base de dados de Choi e Christensen (2013) (melhores resultados em negrito).	114
Tabela 19 – Comparação do RMS dos erros obtidas por diferentes técnicas de rastreamento, avaliadas nos 4 casos de testes sintéticos da base de dados de Choi e Christensen (2013) (melhores resultados em negrito).	116

LISTA DE ABREVIATURAS E SIGLAS

CAD	Computer Aided Design
CCD	Charge-coupled Device
DOF	Degrees of Freedom
GA	Genetic Algorithm
GSL	GNU Scientific Library
IR	Infrared
PCL	Point Cloud Library
PF	Particle Filter
PSO	Particles Swarm Optimization
QR code	Quick Response Code
RA	Realidade Aumentada
RMS	Root Mean Square
ROI	Region of Interest

SUMÁRIO

1	INTRODUÇÃO	19
1.1	DEFINIÇÃO DO PROBLEMA DE PESQUISA	21
1.2	OBJETIVOS DA PESQUISA	22
1.3	ESTRUTURA DA TESE	23
2	CONCEITOS MATEMÁTICOS	24
2.1	REPRESENTAÇÃO DA CÂMERA	24
2.2	SENSORES E IMAGENS RGB-D	30
2.3	RASTREAMENTO TOP-DOWN	31
2.4	DEFINIÇÃO DO PSO	33
3	RASTREAMENTO DE OBJETOS 3D	39
3.1	RASTREAMENTO A PARTIR DE MARCADORES ARTIFICIAIS	39
3.2	RASTREAMENTO A PARTIR DE CARACTERÍSTICAS NATURAIS	42
3.3	RASTREAMENTO DE OBJETOS 3D ARBITRÁRIOS COM 6-DOF	44
4	PSO PARA RASTREAMENTO DE OBJETOS 3D	52
4.1	VISÃO GERAL DO MÉTODO PROPOSTO	52
4.2	DADOS DE ENTRADA	54
4.3	PRÉ-PROCESSAMENTO	54
4.3.1	Construção da Nuvem de Pontos da Cena	55
4.3.2	Vetores Normais e Sistemas de Cores	56
4.3.3	Seleção de Pontos Visíveis do Modelo	57
4.4	INSTANCIACÃO DO PSO	59
4.5	AVALIAÇÃO DAS PARTÍCULAS	62
4.5.1	Correspondência entre Pontos	62
4.5.2	Cálculo da Distância	63
4.6	CRITÉRIO DE PARADA	65
4.6.1	Filtragem de Poses	65
4.6.2	Renderização dos Resultados	66
4.7	PROCESSAMENTO EM GPU	67
5	EXPERIMENTOS E RESULTADOS	70
5.1	DESCRIÇÃO DOS EXPERIMENTOS	70

5.1.1	Base de Dados	71
5.1.2	Etapas dos Experimentos	73
5.2	METODOLOGIA E RESULTADOS DA PRIMEIRA ETAPA	74
5.2.1	Avaliação de Acurácia	76
5.2.2	Avaliação de Desempenho de Tempo	80
5.3	METODOLOGIA E RESULTADOS DA SEGUNDA ETAPA	81
5.3.1	Estudo de Ablação	83
5.3.2	Avaliação da Acurácia e Robustez a Falhas	86
5.3.3	Avaliação do Desempenho de Tempo	93
5.4	METODOLOGIA E RESULTADOS DA TERCEIRA ETAPA	95
5.4.1	Diversidade Populacional e Teste de Convergência do Enxame	95
5.4.2	Estudo da Função de Aptidão	100
5.4.3	Alterações Propostas e Avaliação de Acurácia	112
6	CONSIDERAÇÕES FINAIS	117
6.1	CONCLUSÕES	117
6.2	CONTRIBUIÇÕES	120
6.3	TRABALHOS FUTUROS	121
	REFERÊNCIAS	123
	APÊNDICE A – VÍDEOS E RESULTADOS	128

1 INTRODUÇÃO

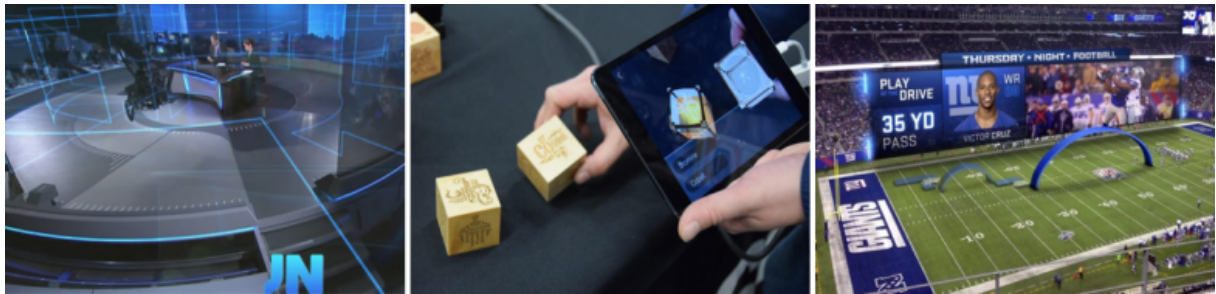
Rastrear um objeto 3D a partir de uma filmagem digital implica em determinar a trajetória desse objeto ao longo do tempo em relação a um referencial, geralmente a própria câmera. Uma vez que a câmera digital faz apenas uma amostragem do movimento na cena filmada, essa trajetória é discreta e construída recursivamente a partir do cálculo da pose do objeto alvo a cada novo quadro capturado, partindo de uma pose inicial previamente conhecida. Ao fixarmos a origem do sistema de coordenadas adotado no objeto ou cena rastreada, é possível também determinar a trajetória da própria câmera em relação a esse sistema.

O rastreamento de objetos 3D pode ser útil em diversas aplicações, como robótica ou automação de veículos (PRICE et al., 2018), onde existem situações em que é necessário atualizar em tempo real as poses dos objetos em uma cena em relação a quem filma. Em RA, por exemplo, o rastreamento da câmera pode proporcionar a adição correta de informações ou objetos virtuais em uma cena real (AZUMA, 1997; LIMA et al., 2017). Na interação humano-computador, o rastreamento pode ser usado para substituir periféricos de entrada tradicionais em aplicações diversas, como jogos digitais de lazer (JORDÀ et al., 2007) ou em aplicações de reabilitação (GAMA et al., 2016; BARIONI et al., 2017) nas quais os comandos de entrada podem ser lidos através do rastreamento do corpo humano.

Já é possível citar diversos exemplos de aplicações em RA comuns e com finalidade fora do escopo acadêmico que usam o rastreamento para adicionar informações virtuais a cenas reais. O programa de televisão Jornal Nacional, da Rede Globo, por exemplo, utiliza diariamente RA na composição do design do seu cenário (Figura 1 à esquerda). As linhas virtuais azuis, apesar do movimento da câmera, parecem fazer parte do cenário real porque são corretamente projetadas ao redor da bancada dos apresentadores a cada quadro da filmagem. A imagem ao centro da Figura 1 mostra ainda uma aplicação educacional que usa RA no processo de ensino-aprendizagem de elementos químicos. As características desses elementos são projetadas dinamicamente sobre cubos rastreados que interagem entre si à medida que são agrupados. Ainda na Figura 1, à direita, a técnica é usada para acrescentar informações às imagens do campo durante a transmissão de partidas de futebol americano.

Existem diferentes abordagens que possibilitam o rastreamento de objetos. As primeiras técnicas ficaram conhecidas como *baseadas em marcadores* porque usavam marcadores fiduciais para determinar a pose do objeto. Esse tipo de marcador corresponde a marcas ou figuras

Figura 1 – Exemplos práticos do uso de rastreamento em aplicações com RA. Cenário de jornal televisivo (esquerda), aplicação educacional (centro) e transmissão de jogos esportivos (direita).



Fonte: Figura adaptada pelo autor partir de Newscast Studio (2020), Professor Josh (2020) e Matellio (2020).

previamente conhecidas inseridas artificialmente na cena para facilitar o rastreamento (KATO; BILLINGHURST, 1999). Apesar de facilitar o processo de rastreamento, uma desvantagem direta desse tipo de técnica é a obrigatoriedade de inserção manual desses marcadores. Para situações com contextos em que não é conveniente a manipulação da cena, os marcadores fiduciais não são desejados. Nesses casos podem ser usados métodos de rastreamento baseados em características presentes naturalmente na cena ou nos objetos rastreados, tais como arestas, cores, normais, coordenadas 3D, dentre outros. Essas técnicas, por não usarem marcadores artificiais, são chamadas usualmente de *sem marcadores*.

As técnicas de rastreamento sem marcadores podem ser classificadas em *bottom-up*, quando calculam a pose diretamente a partir das informações contidas nas imagens capturadas pela câmera, e *top-down*, quando um conjunto de hipóteses de pose é avaliado de acordo com as informações presentes na imagem de entrada com o objetivo de selecionar e encontrar a hipótese de pose que melhor representa a pose real do objeto no quadro em questão. Nas técnicas *top-down*, o conjunto de hipóteses de pose pode ser interpretado matematicamente como subconjunto de um espaço de soluções em que cada um dos seus pontos corresponde a uma possível pose do objeto rastreado em dado instante. Nesse espaço, espera-se ainda que exista um ponto globalmente ótimo que representa a melhor pose dentre as demais.

Dessa forma, as técnicas de rastreamento *top-down* podem ser implementadas por meio de algoritmos de otimização tradicionais, tais como filtros de partículas (Particle Filter (PF)) (CHOI; CHRISTENSEN, 2013; R. Ueda, 2012) ou otimização por enxame de partículas (PSO) (OIKONOMIDIS; KYRIAZIS; ARGYROS, 2012; QIAN et al., 2014; JÚNIOR; LIMA, 2018), e são normalmente chamadas de *baseadas em otimização*. Uma alternativa ao uso de técnicas baseadas em otimização são as técnicas *baseadas em aprendizagem*. Essas técnicas usam algoritmos de

aprendizagem de máquina, tais como florestas aleatórias (AKKALADEVI et al., 2016; TAN et al., 2015) ou redes neurais convolucionais (GARON; LALONDE, 2017) para prever a pose do objeto a partir de informações extraídas desse objeto e de características presentes na cena. Técnicas baseadas em otimização ou baseadas em aprendizagem não são mutualmente excludentes. É possível criar algoritmos de rastreamento usando técnicas híbridas, isto é, que englobam os dois conceitos em diferentes etapas do rastreamento, como em Krull et al. (2014), Tan, Navab e Tombari (2017).

Tais abordagens, baseadas em otimização, em aprendizagem ou híbridas, podem ser usadas para rastrear objetos 3D. Nesses casos, muitas vezes é necessário calcular não só a posição do objeto 3D como também a sua orientação. Assim, determina-se que o rastreamento possui seis graus de liberdade (Degrees of Freedom (DOF)), sendo três correspondentes à rotação e três à translação. Contudo, calcular a trajetória de objetos 3D com 6-DOF a partir de câmeras monoculares comuns pode ser uma tarefa difícil, uma vez que essas câmeras, ao produzirem apenas projeções 2D do ambiente filmado, perdem diversas características tridimensionais dos objetos reais presentes na cena.

Uma alternativa ao uso apenas de imagens 2D obtidas por câmeras monoculares comuns são as imagens RGB-D obtidas por sensores que, além de capturarem imagens RGB da cena, são capazes de construir mapas de profundidade que descrevem em tempo real a profundidade de cada pixel da imagem em relação à câmera (canal D da imagem). A partir de imagens RGB-D é possível realizar a construção de nuvens de pontos 3D mais fiéis à cena real, tornando possível o uso de características 3D tais como coordenadas 3D e normais, antes acessíveis em imagens RGB comuns apenas com o uso de técnicas adicionais, como múltiplas vistas ou estereoscopia.

1.1 DEFINIÇÃO DO PROBLEMA DE PESQUISA

Ao usar métodos de rastreamento de objetos 3D baseados em PSO, o problema de determinar com acurácia os pontos da trajetória do objeto alvo corresponde a encontrar os ótimos globais em um espaço de busca dinâmico, isto é, que se modifica a cada iteração do rastreador. Alguns dos principais desafios relacionados à implementação de uma solução a esse problema são: criar uma forma de otimizar o conjunto de hipótese de poses observando um desempenho de tempo mínimo aceitável para os padrões de aplicações em áreas tais como RA (em torno de 20 fps); determinar um método que lide com o problema de redefinir o subespaço de busca

a cada nova iteração de forma eficiente, isto é, minimizando o tamanho desse subconjunto mas sem correr o risco de excluir os pontos ótimos pertencentes a ele; e, aperfeiçoar a função de avaliação do conjunto de hipóteses de pose, tornando-a o mais independente possível do tipo de objeto rastreado e das características do ambiente e evitando problemas como o aprisionamento das partículas em soluções subótimas ou a convergência prematura do enxame.

Considerando esse contexto, o problema da pesquisa abordado neste trabalho pode ser expresso pela seguinte questão: “Como desenvolver um rastreador de objetos 3D, sem marcadores, com 6-DOF, baseado em otimização, com acurácia e robustez a falhas superiores às dos rastreadores presentes no estado da arte”.

As seguintes hipóteses serão observadas no decorrer da tese:

- H1: O rastreamento de objetos 3D com 6-DOF baseado em PSO pode apresentar melhor acurácia em relação a métodos de estado da arte;
- H2: O rastreamento de objetos 3D com 6-DOF baseado em PSO pode apresentar melhor robustez a falhas em relação a métodos de estado da arte.

1.2 OBJETIVOS DA PESQUISA

O objetivo geral deste trabalho é propor, desenvolver e avaliar uma nova abordagem para o problema do rastreamento de objetos 3D, sem marcadores, com 6-DOF, visando obter um rastreador baseado em PSO com acurácia e robustez a falhas superiores às do estado da arte.

Os objetivos específicos deste trabalho estão listados abaixo:

- Verificar os métodos de rastreamento sem marcadores baseados em otimização e em aprendizagem presentes no estado da arte;
- Identificar os principais pontos fortes e fracos das técnicas de rastreamento de objetos 3D relacionadas, a fim de nortear a definição e escolha de hipóteses de solução que podem colaborar no desenvolvimento do método proposto;
- Criar protótipos que implementem cada hipótese de solução levantada com o intuito de avaliar a sua viabilidade, bem como medir as possíveis melhorias proporcionadas;
- Implementar o método proposto de acordo com as principais características definidas por meio de protótipos;

- Definir e realizar um conjunto de experimentos que possibilitem uma ampla avaliação qualitativa e quantitativa da técnica proposta, comparando os resultados obtidos com métodos de rastreamento relacionados presentes no estado da arte.

1.3 ESTRUTURA DA TESE

Esta tese foi estruturada em seis capítulos. O Capítulo 2 apresenta os principais conceitos matemáticos utilizados no desenvolvimento da técnica de rastreamento. O Capítulo 3 faz uma breve exposição da evolução das técnicas de rastreamento de objetos relacionadas, classificando-as segundo as principais abordagens e tecnologias que caracterizam cada método. O Capítulo 4 explica detalhadamente como o PSO pode ser usado para possibilitar o rastreamento de objetos 3D e as principais características da técnica proposta neste trabalho. O Capítulo 5 apresenta a metodologia dos experimentos realizados e discute os resultados obtidos em relação a outras técnicas de rastreamento presentes no estado da arte. O Capítulo 6 trata das considerações finais a partir da análise dos resultados obtidos.

2 CONCEITOS MATEMÁTICOS

Neste capítulo são apresentados os conceitos matemáticos relacionados ao método de rastreamento de objetos 3D proposto. Na Seção 2.1 é apresentada a geometria de projeção de imagens em câmeras de orifício e a relação entre os sistemas de coordenadas da câmera e do mundo; na Seção 2.2 a criação de nuvens de pontos 3D a partir de imagens RGB-D e suas transformações; na Seção 2.3 a definição de rastreamento 3D com 6-DOF, suas abordagens e técnicas de otimização envolvidas no processo; e na Seção 2.4 é apresentada a definição do PSO e suas principais características.

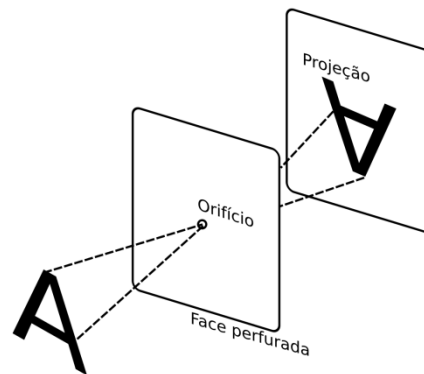
2.1 REPRESENTAÇÃO DA CÂMERA

O objetivo básico de toda câmera fotográfica consiste em capturar a imagem de uma cena 3D do mundo através do seu registro em um plano de imagem 2D. Esse processo é chamado de *projeção* e consiste fundamentalmente em fazer uma amostragem dos pontos do mundo mapeando-os corretamente nas coordenadas desse plano (HARTLEY; ZISSERMAN, 2004). Existem vários modelos de câmeras usadas para projetar imagens de cenas 3D. O modelo que será usado no escopo deste trabalho é o da *câmera de orifício* (ou câmera *pinhole*).

O princípio de uma câmera de orifício é bastante simples, pois não utiliza lentes de distorção em sua composição e consiste apenas de uma câmera escura com um pequeno orifício por onde a luz pode passar e ser projetada na face oposta à face perfurada, também conhecida como *plano de projeção*. Uma descrição desse princípio pode ser vista na Figura 2. Nessa figura, a região em que ocorre a projeção é onde deve se encontrar o filme sensível à luz, no caso das câmeras analógicas, ou o sensor eletrônico de matriz de carga acoplado (Charge-coupled Device (CCD)) caso seja uma câmera digital (GONZALEZ; WOODS, 2000).

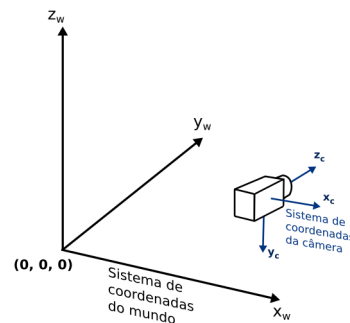
Considerando apenas as características ópticas geométricas do funcionamento de uma câmera de orifício, para calcular a projeção da imagem de objetos reais de um mundo 3D em um plano 2D é necessário definir a posição dos pontos 3D dos objetos presentes na cena em relação a um sistema de coordenadas comum. 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 3.

Figura 2 – Princípio físico da projeção de uma imagem em uma câmera de orifício.



Fonte: Elaborada pelo autor (2017).

Figura 3 – Sistemas de coordenadas do mundo e da câmera.



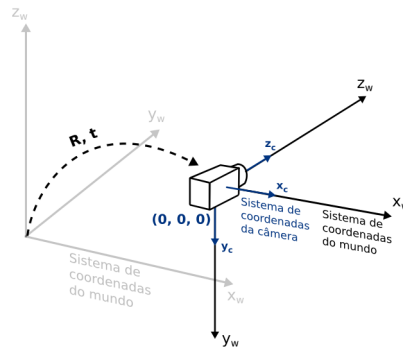
Fonte: Elaborada pelo autor (2017).

Uma vez que a projeção da imagem de um objeto pertencente à cena é realizada no plano de projeção contido na câmera, o resultado dessa projeção obviamente não depende apenas da localização do objeto no mundo, mas também da posição dele em relação à câmera que está possibilitando a tarefa. Assim, é comum estabelecer um sistema de coordenadas alternativo, considerando como origem as coordenadas do centro da câmera usada em relação ao mundo. Esse sistema é conhecido como *sistema de coordenadas da câmera* e a posição de cada um de seus pontos pode ser expressa pelas coordenadas (x_c, y_c, z_c) . Note que, dado um conjunto de pontos quaisquer no sistema de coordenadas do mundo, se a posição da câmera nesse sistema for conhecida, é possível criar uma relação que determine as coordenadas desses pontos no sistema de coordenadas da câmera, pois há uma relação biunívoca entre o tripla de coordenadas atribuído a cada ponto.

Uma forma de obter essa relação, fixando assim o sistema de coordenadas da câmera é realizando uma *transformação*, isto é, uma rotação e uma translação, no sistema de coordenadas

do mundo de modo que os respectivos eixos desse sistema coincidam com os eixos daquele, como mostrado na Figura 4. Essa rotação e translação, usadas para alinhar os dois sistemas de coordenadas, é representada por uma matriz de rotação $\mathbf{R}_{3 \times 3}$ e um vetor de translação $\mathbf{t}_{3 \times 1}$.

Figura 4 – Transformação do sistema de coordenadas do mundo para obter o sistema de coordenadas da câmera.



Fonte: Elaborada pelo autor (2017).

Para simplificar a notação desse tipo de transformação é comum fazer a notação da matriz $\mathbf{R}$ e do vetor $\mathbf{t}$ em uma única matriz $[\mathbf{R}|\mathbf{t}]_{3 \times 4}$. Uma vez que a partir dessa matriz é possível conhecer os parâmetros externos da câmera, tais como sua localização e inclinação no espaço, essa matriz é conhecida como *matriz de parâmetros extrínsecos* (*matriz de pose* ou simplesmente *pose*) da câmera. De posse da matriz de pose da câmera, é possível realizar a transformação necessária para que os pontos de um objeto, ou de uma cena, no sistema de coordenadas do mundo sejam representados no sistema de coordenadas da câmera em questão. Isso pode ser realizado simplesmente calculando o produto da matriz $[\mathbf{R}|\mathbf{t}]_{3 \times 4}$ por cada uma das matrizes $[x_w, y_w, z_w, 1]^T$ formada pelas coordenadas homogêneas de cada ponto desse objeto:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}. \quad (2.1)$$

Dessa forma, na Equação 2.1, a matriz resultante $[x_c, y_c, z_c]^T$ representa as coordenadas transformadas de um dos pontos do objeto, ou seja, que passou a ter como referencial o sistema de coordenadas da câmera.

Uma das formas de se obter a matriz de rotação $\mathbf{R}_{3 \times 3}$, que representa uma rotação em torno de um eixo arbitrário, é através do produto matricial das rotações em torno dos 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 x_w , y_w e z_w a matriz $\mathbf{R}$ pode ser calculada da seguinte forma:

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

A rotação também pode ser representada em *vetor de Euler*, que é uma forma mais compacta de expressar a rotação de um objeto 3D. O vetor de Euler consiste em três componentes $\omega = (\omega_x, \omega_y, \omega_z)^T$ que especificam o eixo a partir do qual o objeto será girado. Nesse tipo de notação a norma desse vetor $\theta = \|\omega\|$ expressa ainda o ângulo que esse giro terá (LIMA, 2014). A relação entre a matriz de rotação $\mathbf{R}$ e o vetor de Euler ω pode ser expressa matematicamente através da fórmula de Rodrigues (BROCKETT, 1984), em que é possível obter a matriz de rotação através do vetor de Euler e vice versa:

$$\mathbf{R} = \cos \theta \mathbf{I} + (1 - \cos \theta) \omega \omega^T + \sin \theta \Omega, \quad (2.3)$$

onde $\mathbf{I}$ é uma matriz identidade e

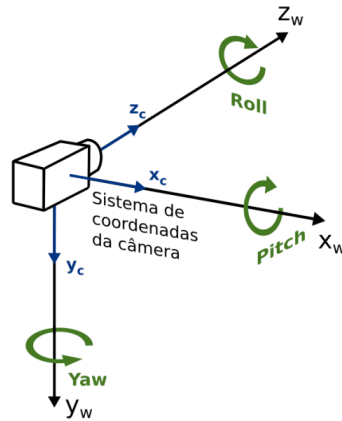
$$\Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \quad (2.4)$$

Para determinar o vetor de Euler a partir da matriz de rotação basta utilizar

$$\sin \theta \Omega = \frac{\mathbf{R} - \mathbf{R}^T}{2}. \quad (2.5)$$

É comum também representar a matriz de rotação usando uma notação conhecida como *ângulos de Euler*. Essa notação representa um vetor diretor do eixo e o ângulo de giro em torno dele por meio três parâmetros α , β e γ , que podem ser chamados respectivamente de *pitch*, *yaw* e *roll* (ver Figura 5). A notação da rotação em ângulos de Euler tem a vantagem de ser concisa, porém um problema que pode ocorrer com o uso dessa notação é a eventual perda de um grau de liberdade durante o movimento de rotação do objeto, situação que ocorre quando

Figura 5 – Representação dos ângulos de Euler.



Fonte: Elaborada pelo autor (2017).

quaisquer dois eixos de rotação ficam alinhados paralelamente durante o giro. Esse fenômeno também é conhecido como *gimbal lock*.

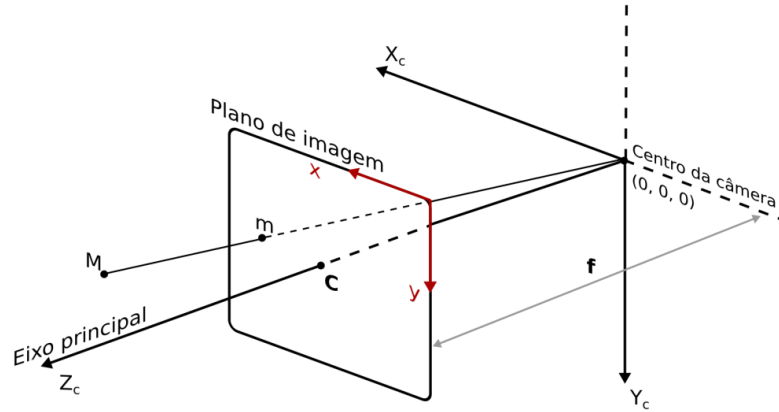
Além da matriz de rotação, vetor e ângulos de Euler, é possível representar a rotação por meio de *quatérnios*. Os quatérnios são vetores de quatro dimensões que podem ser representados como $h = a + bi + cj + dk$ em que a, b, c e $d \in \mathbb{R}$, e as componentes i, j e k são unidades imaginárias em que $i^2 = j^2 = k^2 = ijk = -1$. Uma propriedade importante dos quatérnios é que eles podem representar rotações no $\mathbb{R}^3$ e ao mesmo tempo garantir a unicidade dessa representação, o que possibilita uma maior segurança, por exemplo, no cálculo de médias aritméticas e ponderadas de diferentes rotações. É possível obter uma relação entre uma matriz de rotação $\mathbf{R}$ e o quatérnio $\mathbf{h}$ correspondente a partir da seguinte igualdade de matrizes:

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} = \begin{bmatrix} 2a^2 - 1 + 2b^2 & 2bc - 2ad & 2bd + 2ac \\ 2bc + 2ad & 2a^2 - 1 + 2c^2 & 2cd - 2ab \\ 2bd - 2ac & 2cd + 2ab & 2a^2 - 1 + 2d^2 \end{bmatrix} \quad (2.6)$$

Além dos parâmetros extrínsecos que a matriz de pose representa, a câmera ainda possui características físicas internas que influenciam diretamente no resultado da projeção de cada ponto da imagem. A Figura 6 ilustra um modelo matemático com algumas das principais características geométricas internas de uma câmera de orifício. Nela, os eixos X_c , Y_c e Z_c representam o sistema de coordenadas da câmera com origem em $(0, 0, 0)$, também 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, $\mathbf{M}$ é um

ponto 3D qualquer no sistema de coordenadas da câmera e $\mathbf{m}$ um ponto que corresponde à projeção de $\mathbf{M}$ nas coordenadas 2D do plano de imagem (em vermelho na Figura 6). O eixo Z_c é chamado de *eixo principal* ou *eixo óptico* e corta o plano de imagem no ponto $\mathbf{C}$ chamado de *ponto principal* onde $c_z = f$, em que f é a *distância focal* da câmera.

Figura 6 – Modelo matemático de uma câmera de orifício e algumas de suas principais características geométricas internas.



Fonte: Elaborada pelo autor (2017).

Com o intuito de descrever o modelo algebricamente, define-se como *matriz de parâmetros intrínsecos*, ou simplesmente $\mathbf{K}$, a matriz que contém as informações internas sobre a câmera necessárias para possibilitar a realização de projeções corretas, dessa forma:

$$\mathbf{R} = \begin{bmatrix} n_x f & 0 & c_x \\ 0 & n_y f & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.7)$$

em que 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.

Obtidas as matrizes de parâmetros intrínsecos $\mathbf{K}$ e de parâmetros extrínsecos $[\mathbf{R}|\mathbf{t}]$ da câmera, é possível calcular as projeções de qualquer conjunto de pontos da cena segundo a equação:

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} n_x f & 0 & c_x \\ 0 & n_y f & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}, \quad (2.8)$$

em que s é um fator de escala e $[su, sv, s]^T$ corresponde às coordenadas homogêneas da projeção do ponto $[x_w, y_w, z_w]^T$ no sistema de coordenadas do mundo. O produto entre as matrizes que representam os parâmetros intrínsecos e extrínsecos dá origem à matriz conhecida como *matriz de projeção* $\mathbf{P}$, matematicamente definida como:

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]. \quad (2.9)$$

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

$$s\tilde{\mathbf{m}} = \mathbf{P}\tilde{\mathbf{M}}, \quad (2.10)$$

que é equivalente à Equação 2.8, porém de forma simplificada.

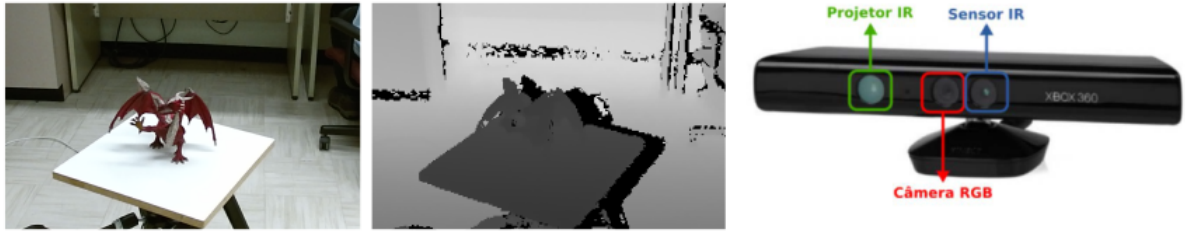
2.2 SENSORES E IMAGENS RGB-D

Dispositivos comerciais, tais como o Microsoft Kinect v1, ficaram conhecidos por possibilitarem a filmagem de cenas usando sensores de profundidade que possibilitam a captura de imagens RGB-D em tempo real. Uma imagem RGB-D, além dos canais RGB comuns, possui um quarto canal (*depth* – D) que traz a informação da profundidade dos objetos na cena. A adição da informação de profundidade se dá por meio da aquisição de uma imagem conhecida como *mapa de profundidade*, que mapeia a profundidade de cada pixel da imagem RGB, levando em consideração a distância do ponto correspondente na cena em relação à câmera a cada quadro da filmagem. Na Figura 7 é possível verificar (à esquerda) uma imagem RGB de uma cena e seu respectivo mapa de profundidade (ao centro) representado em escala de cinza. Quanto mais claro o tom de cinza, mais distante da câmera se encontra o ponto que o pixel representa.

Para gerar esse tipo de imagem, o Kinect v1 utiliza um projetor e um sensor de luz infravermelha (IR) presentes na parte frontal do dispositivo, em detalhes na Figura 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 pontos projetados.

A vantagem do uso de imagens RGB-D está na possibilidade de se reter informações geométricas da cena que, sem o uso de processos adicionais, tais como, estereoscopia ou múltiplas vistas, são inacessíveis em imagens RGB comuns. Uma câmera RGB, ao realizar

Figura 7 – Imagem RGB (esquerda) e o seu mapa de profundidade correspondente (centro). Dispositivo Microsoft Kinect v1 (direita), em destaque o projetor IR, a câmera RGB e o sensor IR.



Fonte: Figura adaptada pelo autor partir de Garon, Laurendeau e Lalonde (2018) e Wikimedia Commons (2020).

a projeção da imagem da cena no plano de imagem 2D naturalmente não consegue reter informações geométricas 3D dos objetos presentes na cena, mas apenas informações de textura. Uma das formas de se contornar esse problema é através do uso de sensores de profundidade como o do Kinect, pois a partir de uma imagem RGB-D e dos parâmetros intrínsecos $\mathbf{K}$ da câmera usada na captura, é possível gerar uma *nuvem de pontos 3D* da cena, viabilizando assim a recuperação de diversas informações 3D perdidas durante a projeção. Ter uma nuvem de pontos da cena em um determinado instante significa conhecer a posição 3D de cada um dos pontos amostrados e visíveis dessa cena nesse instante e em relação à direção de vista da câmera no momento da captura.

2.3 RASTREAMENTO TOP-DOWN

Rastrear um objeto 3D com 6-DOF implica em determinar a trajetória desse objeto com o tempo em relação a um referencial a partir de informações obtidas por uma filmagem digital. Uma vez que a câmera ao filmar uma cena captura uma quantidade finita de quadros em um intervalo de tempo qualquer, essa trajetória deve ser discreta e cada ponto pertencente a ela corresponde a uma matriz de pose $[\mathbf{R}|\mathbf{t}]$ do objeto rastreado obtida a partir das informações contidas no quadro correspondente. Assim, se $[\mathbf{R}|\mathbf{t}]_i$ corresponde à pose do objeto rastreado no i -ésimo quadro capturado pela câmera e N é o número total de quadros que compõe o vídeo, dessa forma a trajetória de um objeto pode ser definida pelo conjunto $T = \{[\mathbf{R}|\mathbf{t}]_1, [\mathbf{R}|\mathbf{t}]_2, \dots, [\mathbf{R}|\mathbf{t}]_N\}$ de poses e o rastreamento como método usado para calcular os elementos de T iterativamente a partir das informações contidas nas imagens capturadas.

Existem vários métodos de rastreamento usados para prever a trajetória de um objeto a partir de informações extraídas dos quadros de uma filmagem. Por conveniência didática,

esses métodos podem ser classificados como *bottom-up* ou *top-down*, e isso ocorre segundo a natureza das abordagens usadas durante a predição. Nos métodos que usam uma abordagem *bottom-up*, primeiro são extraídas características da imagem capturada pela câmera para então, a partir delas, predizer a pose do objeto alvo naquele quadro (VACCHETTI; LEPETIT; FUA, 2004). Já nas abordagens *top-down*, primeiro é criado um conjunto de poses candidatas para então avaliá-las a partir das informações contidas na imagem capturada e, através de um método iterativo de busca, determinar aquela que mais se aproxima da pose real do objeto rastreado (JÚNIOR; LIMA, 2018). O método investigado nesta tese trata-se de uma abordagem *top-down*.

Uma possível forma de tratar o problema de rastreamento usando uma abordagem *top-down* é interpretá-lo como um *problema de otimização*. Nessa abordagem, o conjunto de poses candidatas corresponde a um subconjunto de um espaço de soluções possíveis. Esse conjunto pode ser otimizado para determinar a pose candidata mais próxima da pose real do objeto no momento da captura do quadro em questão. Desse modo, cada pose candidata $[\mathbf{R}|\mathbf{t}]_i$ é chamada de hipótese de pose e sugere uma transformação sofrida pelo objeto alvo ou pela câmera naquele instante, passível de ser avaliada a partir das informações contidas no modelo desse objeto e no quadro capturado pela câmera. Dessa forma, para cada quadro q capturado pela câmera deve-se:

1. Gerar um conjunto de hipóteses de pose em torno de uma pose inicial, conhecida *a priori* ou calculada pelo próprio algoritmo no quadro anterior $q - 1$;
2. Criar uma heurística que defina uma função de avaliação, permitindo assim calcular a qualidade de cada elemento desse conjunto de hipóteses de pose por meio de informações contidas na imagem em q ;
3. Otimizar esse conjunto de soluções por meio da função de avaliação, com o objetivo de tentar encontrar uma *hipótese de pose ótima*, isto é, aquela que mais se aproxima da pose real do objeto no instante de captura do quadro q ;
4. Utilizar a pose encontrada em q como pose inicial para cálculo da pose em $q + 1$.

Para otimizar o conjunto de hipóteses de pose, é possível usar técnicas de otimização conhecidas na literatura tais como PF, algoritmo genético (Genetic Algorithm (GA)) ou PSO. Um dos problemas da representação da hipótese de pose por uma matriz de pose $[\mathbf{R}|\mathbf{t}]$ é a sua

dimensionalidade. Uma vez que os vetores $\mathbf{R}$ e $\mathbf{t}$ possuem juntos 12 componentes, um problema de otimização com essa representação teria um espaço de busca de 12 dimensões. Uma forma de diminuir a dimensionalidade do espaço de busca, é fazer uma conversão da matriz de rotação $\mathbf{R}$ para vetor de Euler, deixando a representação da rotação mais compacta. Dessa forma, uma hipótese de pose pode ser representada por um vetor $\mathbf{p}$ de apenas seis dimensões, as três primeiras representando a rotação do objeto e as três últimas a translação, isto é, $\mathbf{p} = (\omega_1, \omega_2, \omega_3, t_1, t_2, t_3)$. Além de ajudar a diminuir a dimensionalidade do espaço de busca, o uso da rotação em vetor de Euler não sofre do problema de *gimbal lock*.

2.4 DEFINIÇÃO DO PSO

O PSO é uma meta-heurística usada com o objetivo de encontrar boas soluções em problemas de otimização não-lineares, isto é, aqueles que possuem comportamentos complexos e imprevisíveis. Existem problemas não-lineares em que, mesmo imprevisíveis e complexos, é relativamente fácil avaliar quaisquer uma das soluções pertencentes ao conjunto de soluções possíveis, geralmente chamado de *espaço de busca* ou *espaço de soluções*. Porém, uma vez que em problemas reais esses espaços costumam ser extremamente grandes, é difícil determinar ou garantir que a solução avaliada é ótima globalmente. É nesse tipo de problema que meta-heurísticas, tais como o PSO, podem ser utilizadas para encontrar boas soluções.

Diz-se que o PSO é uma técnica *bioinspirada*, pois quando James Kennedy e Russel Eberhart a propuseram em 1995 (KENNEDY; EBERHART, 1995; EBERHART; KENNEDY, 1995) fizeram isso inspirados no comportamento de bandos de pássaros que cooperam entre si para atingir objetivos comuns. Nessa analogia, ao tentar resolver um problema usando PSO, o *enxame de partículas* corresponde ao conjunto de soluções válidas pertencentes ao espaço de soluções do problema em questão. Essas partículas se movimentam por esse espaço segundo regras bem definidas que determinam a posição e velocidade de cada uma delas a cada iteração. Essas regras, além de atualizarem a posição e velocidade das partículas, têm como objetivo proporcionar uma convergência do enxame em direção a regiões que contém boas soluções.

O quão influente vai ser uma partícula do enxame sobre as demais vai depender da qualidade da solução que essa partícula representa naquele instante. A heurística que determina essa qualidade é implementada na *função de avaliação* (ou *função de aptidão*), que corresponde à *função objetivo* em problemas de otimização. Essa função é usada no cálculo da aptidão

de cada partícula da população sempre que houver movimentação dessas partículas ao longo do espaço de busca, isto é, a cada iteração do algoritmo. Ao usar o PSO no contexto do rastreamento de objetos 3D, a avaliação das partículas é definida por uma função que calcula a qualidade da pose encontrada usando informação presentes na cena. A função de aptidão do PSO será apresentada com mais detalhes na Seção 4.5 desta tese.

Para que a convergência do enxame ocorra em direção às regiões promissoras do espaço de soluções, existe uma comunicação e persistência de informações entre suas partículas que proporciona um aprendizado coletivo ao longo das iterações. Esse aprendizado é influenciado pelo *caráter cognitivo* e *caráter social* de suas partículas. O primeiro ocorre porque cada partícula memoriza e atualiza sua melhor posição no espaço ao longo de toda a sua trajetória. Já o segundo ocorre porque cada uma das partículas conhece a melhor posição já visitada por sua vizinhança. Tanto o conhecimento cognitivo quanto o social influenciam no movimento das partículas ao longo da evolução do enxame e, portanto, ambos têm um papel importante na solução final da otimização.

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. No PSO inercial, 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

$$\mathbf{V}_i(t+1) = \mu \mathbf{V}_i(t) + c_1 r_1 [\mathbf{X}_i^{best}(t) - \mathbf{X}_i(t)] + c_2 r_2 [\mathbf{X}_i^{gbest}(t) - \mathbf{X}_i(t)], \quad (2.11)$$

$$\mathbf{X}_i(t+1) = \mathbf{X}_i(t) + \mathbf{V}_i(t+1), \quad (2.12)$$

em que:

- t indica em qual iteração se encontra o algoritmo;
- $\mathbf{X}_i$ é o vetor que representa a posição da partícula i dentro do espaço de busca;
- $\mathbf{X}_i^{best}$ é o vetor que guarda as componentes da melhor solução obtida pela partícula i ao longo de sua trajetória pelo espaço de busca;

- $\mathbf{X}_i^{gbest}$ é o vetor que guarda a melhor posição obtida por uma vizinhança da partícula i até então;
- $\mathbf{V}_i$ é o vetor que representa a velocidade da partícula i ;
- μ é chamado de *peso de inércia*, esse parâmetro regula a influência da velocidade das partículas no processo de otimização. Ele impede que as partículas adquiram altas velocidades, o que dificultaria a convergência do PSO;
- 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 $[0, 1]$. São usados para ponderar aleatoriamente os parâmetros sociais e cognitivos durante o cálculo da velocidade.

O vetor velocidade $\mathbf{V}_i$ de uma partícula i qualquer na iteração $t + 1$ é calculado segundo a soma vetorial expressa na Equação 2.11. O primeiro termo representa o produto entre o peso de inércia μ e a velocidade da partícula na iteração t anterior. O peso de inércia μ é um parâmetro que define o quanto a velocidade anterior de uma partícula vai interferir no cálculo da sua velocidade atual. Valores altos para μ tendem a manter as partículas com velocidades altas, colaborando assim para uma exploração global no espaço de busca, enquanto valores baixos favorecem uma exploração mais local, ou seja, em regiões próximas à posição em que se encontra a partícula em questão.

Na literatura existem várias estratégias para calcular μ e, segundo Nickabadi, Ebadzadeh e Safabakhsh (2011), elas podem ser categorizadas em três classes. Na primeira delas, chamada de *pesos de inércia constantes e aleatórios*, como o próprio nome sugere, μ pode assumir um valor constante ao longo das iterações ou é calculado de forma aleatória em cada uma delas. Nesse tipo de peso de inércia não é usada nenhuma informação do enxame como entrada. Na segunda classe, chamada aqui de *pesos de inércia temporais*, o valor de μ é definido dinamicamente em função do tempo ou do número de iterações do PSO. Na terceira classe, conhecida como *pesos de inércia adaptativos*, o valor de μ é calculado de acordo com funções que monitoram características do enxame a cada iteração.

Nesta tese foram usados dois tipos de peso de inércia, o peso de inércia temporal e o peso de inércia adaptativo. Para o cálculo do peso de inércia temporal, foi usado um valor dinâmico para o μ dentro de um intervalo $[\mu_{min}, \mu_{max}]$, que começa em um valor μ_{max} e decresce em

função do número máximo de iterações t_{max} do PSO (BRATTON; KENNEDY, 2007), como descrito nas equações:

$$h = \frac{3t_{max}}{4}, \quad (2.13)$$

$$\mu(t) = \begin{cases} \mu_{min} + (\mu_{max} - \mu_{min})\frac{h-t}{h} & \text{se } t \leq h \\ \mu_{min} & \text{se } t > h \end{cases}. \quad (2.14)$$

Para o cálculo do peso de inércia adaptativo foi usado o método proposto em Nickabadi, Ebadzadeh e Safabakhsh (2011), nele μ é calculado usando a função:

$$\mu(t) = (\mu_{max} - \mu_{min})P_s + \mu_{min}, \quad (2.15)$$

em que $P_s(t) \in [0, 1]$ e representa a porcentagem de partículas do enxame que conseguiram melhorar a aptidão na última iteração, e pode ser representado pela equação:

$$P_s(t) = \frac{\sum_{i=1}^n S(i, t)}{n}, \quad (2.16)$$

em que $S(i, t)$ representa a função de sucesso da partícula, que define se as partículas ainda estão convergindo ou não. Assim, para cada partícula i na iteração t , $S(i, t)$ pode ser calculado usando:

$$S(i, t) = \begin{cases} 1 & \text{se } \text{aptidão}(\mathbf{X}_i^{best}(t)) < \text{aptidão}(\mathbf{X}_i^{best}(t-1)) \\ 0 & \text{se } \text{aptidão}(\mathbf{X}_i^{best}(t)) = \text{aptidão}(\mathbf{X}_i^{best}(t-1)) \end{cases}. \quad (2.17)$$

O segundo termo da Equação 2.11 calcula a influência do conhecimento da partícula sobre sua nova velocidade. O resultado é obtido a partir da diferença entre a melhor posição da partícula ao longo da sua trajetória e sua posição no instante t . O vetor resultante nesse termo ainda é ponderado por c_1 e r_1 . Ainda na Equação 2.11, o terceiro termo calcula a influência do conhecimento social do enxame sobre a partícula. Nele acontece a subtração entre a posição atual da partícula e posição da melhor partícula de uma vizinhança de i , o resultado é ponderado pelos parâmetros c_2 e r_2 .

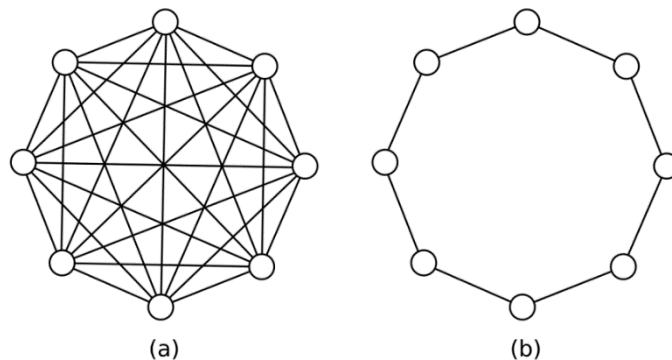
Os parâmetros c_1 e c_2 podem ser escolhidos para modular a intensidade com que o caráter cognitivo ou social vai influenciar o movimento das partículas no enxame, ou seja, se $c_1 > c_2$ o enxame terá um movimento com caráter mais individual, enquanto que $c_1 < c_2$ proporcionará um caráter mais cooperativo entre as partículas. Porém, apesar dessas possibilidades, é comum o uso de valores iguais para c_1 e c_2 (MARINI; WALCZAK, 2015). No entanto, devido ao uso dos números aleatórios r_1 e r_2 , não existem garantias de que as influências cognitivas e

sociais tenham o mesmo peso a cada atualização da velocidade da partícula. Eles são usados justamente para assegurar que as partículas apresentem trajetórias semi-aleatórias durante o movimento.

A Equação 2.12 representa a atualização da posição da partícula i no instante $t + 1$, realizada através da soma vetorial da sua posição no instante anterior t com a velocidade atual dessa partícula, calculada anteriormente como exposto na Equação 2.11. Essas equações são usadas a cada iteração do PSO, atualizando as posições das partículas até que um critério de parada pré-definido seja alcançado. Geralmente, os critérios de parada usados são o número de iterações computadas ou o valor da aptidão da melhor partícula do enxame.

Durante o movimento do enxame a comunicação entre as partículas ocorre por meio do vetor $\mathbf{X}_i^{gbest}$. Esse vetor guarda a posição da melhor partícula dentro de uma vizinhança para ser usada na Equação 2.11. O que define quantas e quais partículas fazem parte da vizinhança de uma partícula qualquer do enxame é a *topologia* do PSO, que deve ser escolhida previamente e com cautela, pois ela influencia diretamente na maneira como as partículas se comunicam e, portanto, na qualidade das soluções encontradas. Existem muitas topologias na literatura, sendo as duas mais conhecidas a *topologia local* (ou apenas *lbest*) e a *topologia global* (ou simplesmente *gbest*). Na primeira, cada partícula conhece e se comunica com apenas dois vizinhos. Já na segunda, cada partícula conhece e troca informações com todas as outras partículas do enxame. Na Figura 8 podemos observar dois grafos representando ambas as topologias descritas. Os vértices representam as partículas e as arestas a comunicação entre partículas de uma mesma vizinhança.

Figura 8 – Topologias do PSO: global ou *gbest* (à esquerda) e local ou *lbest* (à direita).



Fonte: Elaborada pelo autor (2017).

Cada uma dessas topologias possui características que podem favorecer ou não a resolução

do problema em questão e deve ser usada observando o contexto do problema e a natureza das soluções que se deseja encontrar. A topologia *gbest*, por exemplo, pode ser usada quando se quer uma convergência mais rápida do enxame em detrimento da qualidade das soluções encontradas. Essa topologia favorece uma exploração mais ágil do espaço de busca, porém, devido à troca rápida de informações entre todas as partículas, ela pode causar uma convergência prematura do enxame, correndo o risco de diminuir a qualidade das soluções encontradas, pois aumenta as chances do enxame não explorar o espaço de busca de forma adequada, convergindo para um mínimo local. Já a topologia *lbest* possui uma exploração mais lenta do espaço de busca, porém com a tendência de evitar convergências ruins ou prematuras, apresentando uma chance maior de encontrar melhores resultados.

3 RASTREAMENTO DE OBJETOS 3D

Este capítulo é dividido em três seções que apresentam aspectos da evolução das técnicas de rastreamento de objetos 3D, desde as primeiras abordagens que usavam marcadores até aquelas baseadas em modelo que rastreiam os objetos a partir de características naturais. Na Seção 3.1 são apresentadas algumas propriedades importantes das primeiras técnicas de rastreamento que usavam marcadores inseridos artificialmente na cena. A Seção 3.2 trata do uso de características naturais como forma de superar os problemas relacionados ao uso de marcadores. A Seção 3.3 são apresentadas técnicas de rastreamento baseadas em aprendizagem, em otimização ou híbridas, que usam modelos para rastrear objetos 3D arbitrários, ou seja, que não pertencem necessariamente a uma classe específica de objetos. Algumas das técnicas apresentadas nesta última seção fazem parte do estado da arte e são trabalhos relacionados ao método proposto nessa tese.

3.1 RASTREAMENTO A PARTIR DE MARCADORES ARTIFICIAIS

Na literatura existem diversas técnicas utilizadas para realizar o rastreamento de objetos 3D a partir de uma filmagem, e o que há em comum entre elas é que todas precisam extrair dos quadros da filmagem informações suficientes para possibilitar o cálculo da pose do objeto alvo. O tipo de informação e a técnica de processamento usados para realizar essa tarefa podem servir para distinguir uma técnica de outra. As primeiras técnicas de rastreamento adicionavam artificialmente informações à cena. Isso era feito através do uso de *marcadores artificiais* inseridos nos objetos do ambiente para facilitar o processamento das informações contidas nas imagens.

Marcadores artificiais, ou simplesmente *marcadores*, são figuras que originalmente não fazem parte da cena e possuem características previamente conhecidas, escolhidas intencionalmente para se destacarem das demais e assim serem facilmente reconhecidas pelo rastreador. Servem como pontos fiduciais estrategicamente arranjados sobre a cena com o intuito de tornar o cálculo da pose do objeto uma tarefa menos complexa e, portanto, facilitar o rastreamento. Os primeiros marcadores serviam para especificar as coordenadas de pontos da imagem, por isso eram chamados de *marcadores pontuais*, porém, quando agrupados em conjunto e dispostos em padrões geométricos específicos possibilitavam determinar a pose 3D de um objeto

e não apenas sua localização (HOFF; NGUYEN; LYON, 1996). Na Figura 9 é possível observar dois tipos de marcadores pontuais e exemplos de aplicações.

Figura 9 – Exemplos de marcadores pontuais e situações em que são utilizados.



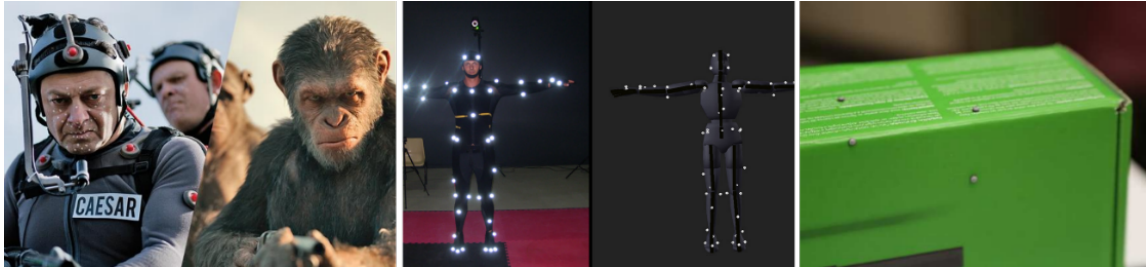
Fonte: Figura adaptada pelo autor partir de Insurance Institute for Highway Safety (2020) e Hoff, Nguyen e Lyon (1996).

O agrupamento desses marcadores em padrões pré-estabelecidos, como ocorre na quarta imagem da Figura 9, possibilita o cálculo da pose 3D da câmera a partir da criação de uma relação $M_i \leftrightarrow m_i$ entre a posição 3D de cada marcador no mundo e seus respectivos correspondentes no plano 2D. Em virtude do seu formato, os marcadores circulares sofrem poucas variações em diferentes perspectivas e, devido ao cálculo do centroide, proporcionam uma ótima estimativa de posição 2D na projeção da imagem. Essa facilidade de reconhecimento e de detecção da localização dos marcadores pontuais faz com que o rastreamento que usa esse tipo de estratégia tenha uma boa acurácia em relação àqueles realizados a partir de características naturais do objeto (LEPETIT, 2005).

Apesar de terem sido criados há muitos anos, os marcadores pontuais ainda são bastante utilizados, principalmente naquelas situações em que é possível realizar manipulação na cena e/ou quando é necessário um rastreamento com uma melhor acurácia. Na pós-produção de filmes, por exemplo, os marcadores pontuais são usados em técnicas como a captura de movimento durante a aquisição de expressões faciais de personagens, como pode ser visto na Figura 10 à esquerda, ou da dinâmica do corpo humano, na Figura 10 ao centro. Em contextos científicos, marcadores pontuais são bastante aplicados quando se deseja adquirir valores de *ground truth* com boa acurácia em sequências de imagens. Essa foi a técnica usada por exemplo em Garon, Laurendeau e Lalonde (2018) durante a criação de sua base de dados. Quatro marcadores pontuais utilizados na ocasião podem ser vistos no detalhe da imagem à direita na Figura 10.

Ao ser inserido em uma cena, cada marcador pontual expressa apenas uma localização. Caso o método necessite fazer a leitura de uma pose 3D, é necessário ter mais de um marcador

Figura 10 – Exemplos do uso de marcadores pontuais, para aquisição de expressões faciais (esquerda), do movimento do corpo humano (centro) e de valores de ground truth da trajetória de objetos.



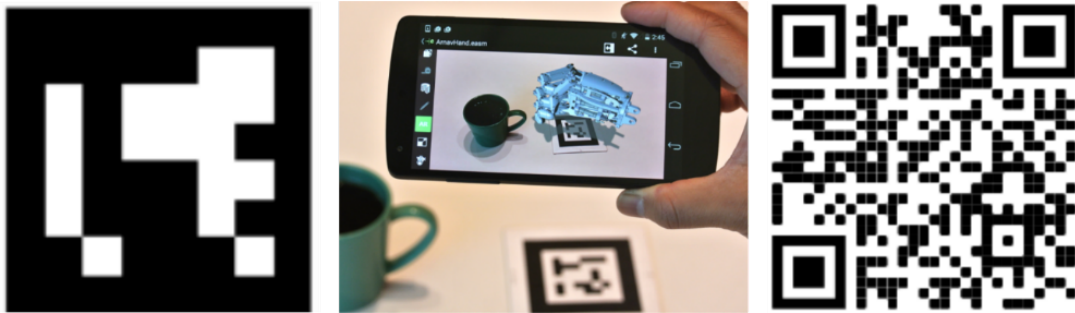
Fonte: Figura adaptada pelo autor partir de Backstage (2020), Every Little Achievement Counts (2020) e Garon, Laurendeau e Lalonde (2018).

como referência. Para contornar esse tipo de limitação, foram criados os *marcadores planares* (Figura 11) (KOLLER et al., 1997). Os quatro cantos desse tipo de marcador são usados para determinar a pose de objetos através da correspondência $M_i \leftrightarrow m_i$ de suas coordenadas. Dessa forma, é necessário apenas um marcador planar para estimar a pose 3D de um plano ou de um objeto em relação à câmera.

Um marcador planar pode ainda conter informações em sua área interna, codificadas em sequências de pequenos quadrados dispostos em padrões pré-estabelecidos. Assim, um único marcador planar, além possibilitar o cálculo da pose da câmera, pode ainda conter um código de identificação única. Na Figura 11 ao centro, por exemplo, um marcador planar está sendo utilizado em uma aplicação de RA. A informação codificada no seu interior pode ser utilizada para especificar qual objeto virtual 3D ele representa, enquanto as coordenadas 3D de seus cantos fornecem a pose na qual esse objeto será renderizado na tela para o usuário. Uma das versões mais populares de marcadores planares é o QR code, que pode ser visto na Figura 11 à direita. Esse tipo de marcador é muito útil, pois contém informações que podem ser convertidas em um texto, geralmente um endereço de página na Internet, um número de telefone, uma geolocalização, dentre outros.

Apesar das vantagens do uso de rastreamento baseado em marcadores, ainda existem situações que são incompatíveis com esse tipo de abordagem devido à necessidade de manipulação da cena, ou seja, são casos em que a inserção manual de figuras é intrusiva além do tolerável ou sequer possível. Com o intuito de contornar esse tipo de problema, trabalhos como Wagner, Langlotz e Schmalstieg (2008) passaram a propor o uso de marcadores menos invasivos. Outras pesquisas foram além, e propuseram técnicas que não necessitam desse tipo de interferência na cena (NEUMANN; YOU, 1999), a solução encontrada foi usar pontos fiduciais que já estivessem presentes naturalmente nos objetos que compõem o ambiente.

Figura 11 – Marcador planar (esquerda), aplicação móvel de RA usando um marcador planar (centro) e QR code (direita).



Fonte: Figura adaptada pelo autor partir de Trenta Minuti (2020) e QR Code Generator (2020).

3.2 RASTREAMENTO A PARTIR DE CARACTERÍSTICAS NATURAIS

Quando uma técnica é usada para rastrear objetos 3D reais em uma cena por meio de marcadores colocados sobre eles, implica em dizer que a técnica está apta a rastrear esses marcadores e não os objetos em si. O posicionamento dos marcadores sobre a cena apenas transfere características geométricas dos objetos para os marcadores. Isso muitas vezes ocorre de forma bastante eficiente, mas ao custo de manter um processo artificial e invasivo, que nem sempre se adequa a todas as situações que demandam o rastreamento de objetos 3D. Com o intuito de contornar esse tipo de limitação, foram criadas técnicas de rastreamento baseadas em características naturais, ou *sem marcadores*. Esses métodos apresentam soluções em que os marcadores são substituídos por características presentes naturalmente nos próprios objetos que compõem a cena, tais como cantos, linhas, arestas, círculos, texturas, dentre outras, identificadas através de técnicas de visão computacional.

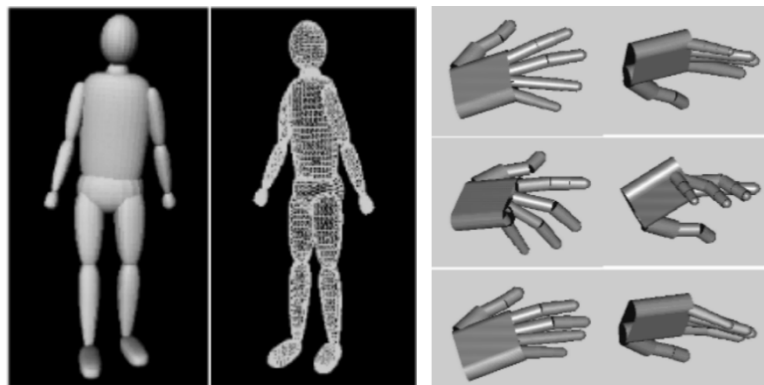
Em geral, as técnicas de rastreamento sem marcadores são classificadas como *baseadas em modelo* ou *baseadas em reconstrução* (TEICHRIEB et al., 2007). Aquelas baseadas em modelo usam o conhecimento prévio do objeto rastreado por meio de um desenho assistido por computador (Computer Aided Design (CAD)) desse objeto. O modelo de um objeto alvo trata-se de sua versão digital contendo características geométricas ou de textura necessárias à técnica de rastreamento que irá usá-lo. Ele possibilita o acesso e manipulação dessas características durante a execução do método. As técnicas baseadas em reconstrução não precisam de informações prévias da cena para o rastreamento ser realizado. Apesar de serem mais complexas do que aquelas baseadas em modelo e mais suscetíveis a acúmulo de erros (LEPETIT, 2005), abordagens desse tipo podem ser usadas para reconstruir ambientes desconhecidos completos

através do seu rastreamento (TEICHRIEB et al., 2007).

Assim como nas técnicas baseadas em marcadores, um método baseado em características naturais precisa estabelecer uma relação entre as características do objeto 3D real, escolhidas para possibilitar o rastreamento, e suas correspondências nas projeções desse objeto no plano de imagem 2D. Desse modo, para conseguir prever as poses a partir de imagens da cena, esses métodos precisam implementar uma forma de medir e minimizar os erros de reprojeção considerando as correspondências estabelecidas. O uso de modelos 3D facilita bastante a realização dessa tarefa, uma vez que eles podem ser facilmente transformados, projetados e analisados a partir de diversas poses.

Como exemplo, métodos de rastreamento do corpo humano como em Sminchisescu e Triggs (2001), Deutscher, Blake e Reid (2000) usam modelos conhecidos como esqueletos cinemáticos (Figura 12 à esquerda). Esse tipo de modelo possui um conjunto de partes articuladas com diversas características como quantidade de juntas, proporções internas e tamanho dos membros, usadas durante o rastreamento para determinar a pose de humanos com vários graus de liberdade. Modelos cinemáticos de mãos, como na Figura 12 (à direita), também são muito usados no rastreamento de mãos humanas (OIKONOMIDIS; KYRIAZIS; ARGYROS, 2012; REHG; KANADE, 1994). Devido à flexibilidade na criação de modelos CAD, existem métodos baseados em modelo para uma grande variedade de objetos e contextos, tais como reconhecimento facial (GOKTURK et al., 2002) e detecção e rastreamento de veículos (KOLLER et al., 1992).

Figura 12 – Modelos articulados do corpo e de mãos humanas.



Fonte: Figura adaptada pelo autor partir de Sminchisescu e Triggs (2001) e Rehg e Kanade (1994).

O conjunto de técnicas de rastreamento baseadas em modelo é amplo e diversificado em contexto. Porém, apesar de resolverem problemas complexos de rastreamento, cada um dos exemplos citados anteriormente funciona apenas com objetos pertencentes a uma classe

pré-definida, dentro de um domínio específico para qual cada rastreador em questão foi desenvolvido. Ou seja, esse tipo de abordagem, apesar de usar características naturais dos objetos e serem aptas a rastrear diferentes objetos de uma mesma classe, não conseguem rastrear *objetos arbitrários*, ou seja, que não necessariamente pertençam a uma só classe de objetos. Um caminho adotado para resolver esse tipo de problema foi o desenvolvimento de técnicas de rastreamento a partir de características naturais comuns a todos os objetos, oferecendo assim soluções mais gerais.

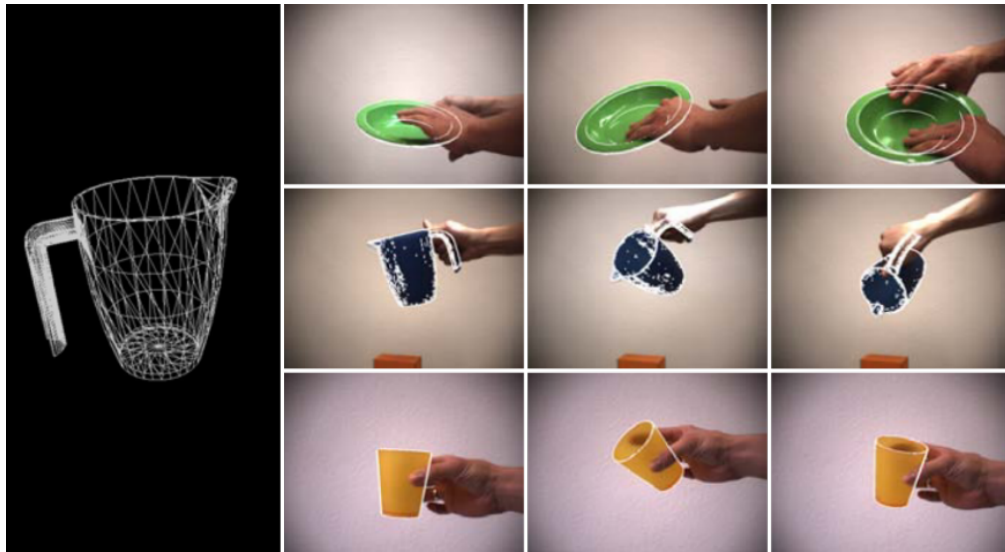
3.3 RASTREAMENTO DE OBJETOS 3D ARBITRÁRIOS COM 6-DOF

Existem técnicas que conseguem rastrear objetos complexos, isto é, com articulações que proporcionam movimentos com dezenas de graus de liberdade, que podem apresentar diferentes tamanhos ou ainda interagir com outros objetos na cena, como no rastreamento de mãos ou de corpos humanos. Apesar de conseguir bons resultados, essas técnicas são extremamente especializadas dentro do contexto para o qual foram planejadas. Uma desvantagem direta disso é a falta de flexibilidade em relação à escolha do objeto alvo para o rastreamento. Ou seja, uma técnica que rastreia mãos humanas bem, por exemplo, pode encontrar sérias dificuldades para rastrear outros tipos de objetos, não pertencentes à classe “mão”.

Para evitar esse tipo de problema, diversas pesquisas propuseram métodos de rastreamento a partir de características que são comuns em vários tipos de objetos, tais como coordenadas de pontos 3D, cor, normais e curvatura, tornando assim o rastreamento menos dependente de um contexto específico. Em Azad et al. (2011) foi proposta uma técnica de rastreamento de objetos 3D arbitrários a partir de imagens RGB baseada em otimização. Esse trabalho usou um PF para estimar as melhores poses a partir do conhecimento a priori do modelo do objeto rastreado. Experimentos mostraram que a técnica possibilita rastrear objetos contidos em ambientes reais, inclusive com oclusões parciais. O tipo de modelo usado e alguns resultados podem ser vistos na Figura 13. Porém, apesar de usar GPU, os resultados ainda apresentaram um alto custo computacional, não possibilitando um rastreamento em tempo real.

Posteriormente, Choi e Christensen (2013) também apresentou um método de rastreamento de objetos 3D *top-down* baseado em PF. Porém, nesse trabalho os pesquisadores optaram por avaliar as hipóteses de pose a partir de características extraídas de imagens RGB-D, possibilitando o uso de características geométricas 3D na avaliação das partículas. A técnica proposta é similar ao rastreador da Point Cloud Library (PCL) (R. Ueda, 2012), mas trouxe

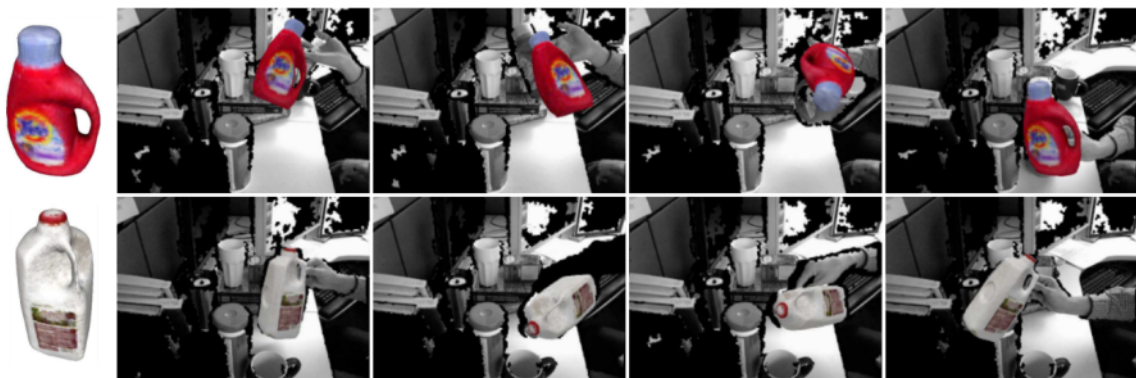
Figura 13 – Exemplo de um dos modelos usados e resultados do rastreamento de diferentes objetos obtidos pela técnica apresentada em Azad et al. (2011).



Fonte: Figura adaptada pelo autor partir de Azad et al. (2011).

melhorias que a tornou superior, tanto em acurácia quanto em desempenho de tempo. Para possibilitar um bom desempenho de tempo, Choi e Christensen (2013) ainda usou o processamento em GPU nos trechos críticos da técnica, o que dessa vez, diferente de Azad et al. (2011), possibilitou um rastreamento a taxas próximas de 20 fps na maioria dos casos de testes. A Figura 14 mostra exemplos de modelos usados na base de dados apresentada em Choi e Christensen (2013) e alguns quadros com resultados do rastreamento realizado pela técnica proposta.

Figura 14 – Exemplos de modelos e resultados do rastreamento de dois objetos usando a técnica proposta em Choi e Christensen (2013) nos casos de teste com cenas reais. Os modelos coloridos transformados pela pose encontrada são projetados sobre a cena em tons de cinza.

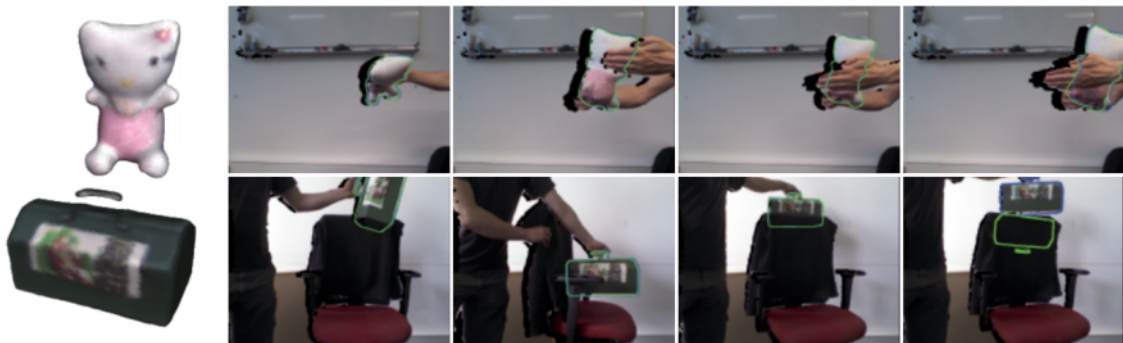


Fonte: Figura adaptada pelo autor partir de Choi e Christensen (2013).

O método apresentado em Choi e Christensen (2013), no entanto, foi superado por Krull et al. (2014), que propôs uma técnica de rastreamento híbrida utilizando PF e floresta aleatória

para rastrear objetos 3D com 6-DOF em imagens RGB-D. Nela ocorre um treinamento *offline* de uma floresta aleatória usando um conjunto de objetos potencialmente rastreáveis. Durante os testes, a floresta aleatória treinada é usada para regredir as coordenadas do objeto 3D rastreado, possibilitando um rastreamento acurado quando esse resultado é combinado com um PF. Experimentos mostraram que a abordagem híbrida de Krull et al. (2014) obteve melhores resultados que Choi e Christensen (2013), R. Ueda (2012), mesmo quando foi usado um PF com um conjunto de partículas muitas vezes menor que aquele usado por Choi e Christensen (2013). Alguns resultados da técnica proposta em Krull et al. (2014) e o exemplo de dois modelos utilizados podem ser vistos na Figura 15.

Figura 15 – Exemplos de modelos e resultados obtidos pelo rastreador proposto em Krull et al. (2014). A silhueta em azul representa a pose do *ground truth* e a silhueta em verde a pose estimada.



Fonte: Figura adaptada pelo autor partir de Krull et al. (2014).

Os trabalhos Akkaladevi et al. (2016), Tan et al. (2015) mostraram que o uso de métodos baseados em aprendizagem poderia sozinho proporcionar um rastreamento com boa acurácia e em tempo real, mesmo quando processados apenas em CPU. Em Tan et al. (2015) é proposto um rastreador que utiliza uma floresta aleatória para realizar o rastreamento de dezenas de objetos 3D simultaneamente e em tempo real a partir de imagens RGB-D. O método se mostrou eficiente tanto no rastreamento baseado no conhecimento prévio do modelo e aprendizagem *offline*, quanto quando não se conhece o modelo do objeto rastreado. Nesse caso o algoritmo inicia apenas com a pose inicial do objeto e a partir das imagens de profundidade o objeto alvo vai sendo aprendido de forma adaptativa durante o rastreamento, classificando a aprendizagem como *online*. As duas versões da técnica foram testadas em base de dados reais e sintéticas. A versão com aprendizagem *offline* obteve melhor acurácia que Choi e Christensen (2013), R. Ueda (2012), Krull et al. (2014) em experimentos com a base de dados de Choi e Christensen (2013), enquanto que a versão com aprendizagem *online* da técnica, apesar de obter bons resultados, superou apenas R. Ueda (2012) nestes testes.

O método proposto em Akkaladevi et al. (2016) também possibilitou um rastreamento simultâneo. A técnica se mostrou eficiente mesmo quando testada em sequências de imagens obtidas a partir do movimento rápido da câmera, o que produz quadros borrados que dificultam o processo de rastreamento. Em Akkaladevi et al. (2016) ainda é proposta uma base de dados com sequências de imagens RGB-D, em que é possível verificar a acurácia do rastreamento usando quatro diferentes objetos 3D. Experimentos realizados na base de dados proposta pelo próprio trabalho mostraram que o rastreador de Akkaladevi et al. (2016) foi mais acurado que os métodos em Choi e Christensen (2013), R. Ueda (2012), Tan et al. (2015), porém, ficando atrás da técnica de Tan et al. (2015) no desempenho de tempo nessa série de testes. A Figura 16 mostra exemplos de modelos usados nos experimentos em Akkaladevi et al. (2016) e o resultado do rastreamento simultâneo de quatro objetos em dois instantes da sequência.

Figura 16 – Exemplos de modelos usados (imagens à esquerda), duas imagens capturadas em diferentes instantes (ao centro), uma delas distorcida devido à velocidade da câmera, e os respectivos resultados do rastreamento simultâneo nesses dois quadros, realizado pela técnica proposta em Akkaladevi et al. (2016) (imagens à direita).



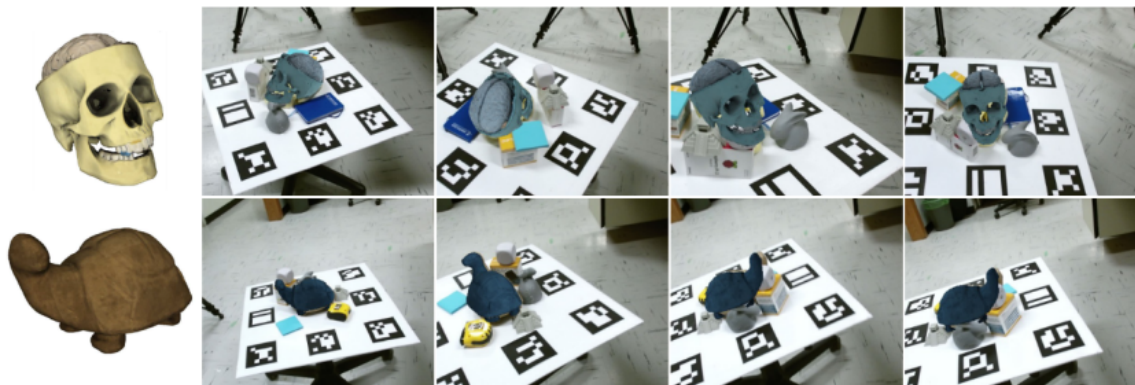
Fonte: Figura adaptada pelo autor partir de Akkaladevi et al. (2016).

Em Kehl et al. (2017) foram utilizados métodos de otimização para encontrar a pose de objetos 3D a partir de informações do contorno e do interior do modelo obtidas a partir de pré-renderizações *offline*. A técnica criada, além de ter uma acurácia compatível com aquela obtida por técnicas do estado da arte, ainda possibilitou um ótimo desempenho com o rastreamento em tempo real em imagens RGB-D utilizando uma única CPU. Outro método que possibilitou o rastreamento de objetos 3D com 6-DOF em tempo real usando apenas uma CPU foi aquele apresentado em Tan, Navab e Tombari (2017). Esse trabalho propôs uma nova técnica híbrida

que propiciou rastrear objetos de forma simultânea, com uma acurácia e desempenho de tempo muito bons. Testes realizados na base de dados sintética (CHOI; CHRISTENSEN, 2013) apresentaram uma taxa de erro próxima de zero em relação ao *ground truth*, superando os resultados obtidos anteriormente pelo próprio em Tan et al. (2015) e por terceiros em Choi e Christensen (2013), R. Ueda (2012), Krull et al. (2014).

Em Garon e Lalonde (2017) foi apresentado um rastreador baseado em aprendizagem que utiliza redes neurais convolucionais. A técnica, após treinamento *offline*, consegue estimar as mudanças na pose do objeto 3D com 6-DOF a partir das informações RGB-D presentes a cada novo quadro. Uma série de experimentos em uma base proposta pelos autores e na PROFACTOR 3D Tracking Dataset (AKKALADEVI et al., 2016), mostraram que o rastreador proposto obteve melhores resultados quando comparados com Akkaladevi et al. (2016) e com Tan et al. (2015). Na Figura 17 podem ser visualizados alguns dos resultados obtidos pelo rastreador proposto em Garon e Lalonde (2017) durante esses experimentos e dois dos modelos utilizados.

Figura 17 – Exemplos de modelos e resultados obtidos pela técnica proposta em Garon e Lalonde (2017) em dois casos de teste.

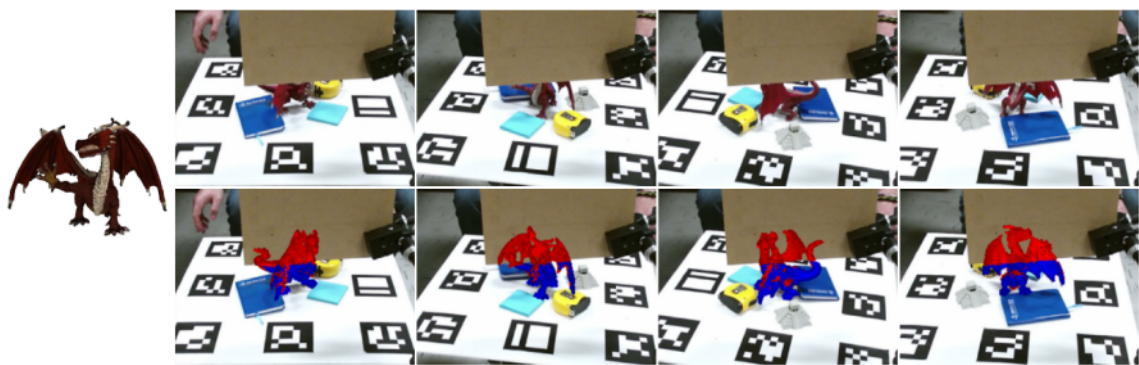


Fonte: Figura adaptada pelo autor partir de Garon e Lalonde (2017).

O conjunto de resultados dos últimos experimentos em Tan, Navab e Tombari (2017) e Garon e Lalonde (2017) parecem apontar uma tendência de que rastreadores de objetos com 6-DOF, quando baseados em aprendizagem ou híbridos, se destacam de forma positiva em comparação àqueles baseados apenas em otimização. No entanto, em Júnior e Lima (2018) propusemos uma técnica de rastreamento baseada em otimização capaz de obter resultados competitivos em relação àquelas baseadas em aprendizagem (TAN et al., 2015; GARON; LALONDE, 2017), com a vantagem de não necessitar de treinamentos demorados. Uma desvantagem do método apresentado em Júnior e Lima (2018) foi o desempenho de tempo, uma

vez que não foi possível executar o rastreamento com uma alta taxa de quadros, como é esperado em aplicações de RA. Na Figura 18 é possível observar resultados do rastreador proposto em Júnior e Lima (2018) em casos de teste com oclusão parcial do objeto alvo. A técnica descrita em Júnior e Lima (2018) é parte integrante do arcabouço proposto neste trabalho e foi combinada com melhorias adicionais que levaram a ganhos de acurácia, robustez e tempo de execução.

Figura 18 – Exemplo de modelo e de resultados obtidos pela técnica proposta em Júnior e Lima (2018) usando casos de teste com 40% oclusão na horizontal da base de dados apresentada em Garon e Lalonde (2017). Imagens RGB de entrada (acima) e resultados (abaixo).



Fonte: Figura adaptada pelo autor partir de Júnior e Lima (2018).

Em Garon, Laurendeau e Lalonde (2018) foi proposta uma nova versão do rastreador 6-DOF apresentado em Garon e Lalonde (2017). Essa nova versão trouxe aprimoramentos no conjunto de parâmetros da arquitetura de aprendizagem profunda e na metodologia de treinamento do rastreador. Essas modificações possibilitaram resultados experimentais superiores àqueles obtidos com o rastreador apresentado anteriormente. Além das melhorias na qualidade do rastreador, Garon, Laurendeau e Lalonde (2018) propuseram ainda uma extensa base de dados com 297 diferentes casos de testes distribuídos em 3 tipos de cenários reais: o *occlusion*, que proporciona a realização de experimentos em cenas com diversos níveis de oclusão do objeto alvo; o *interaction*, que traz casos de teste com a movimentação manual do objeto alvo em diferentes velocidades de rotação e translação, o que impõe diferentes níveis de dificuldade ao rastreador; e o *stability*, no qual é possível medir o nível de *jitter* do rastreador a partir de cenários estáticos. A base de dados ainda foi criada usando um sistema de captura de movimentos que permitiu um *ground truth* mais acurado que aquele da base de dados proposta anteriormente pelos autores em Garon e Lalonde (2017).

Um dos trabalhos que usou a base de dados de Garon, Laurendeau e Lalonde (2018) foi aquele apresentado em Maroungkas et al. (2020). Esse artigo propõe um rastreador de objetos

3D em imagens RGB-D baseado em modelos, no qual se usa um sistema que incorpora métodos que lidam com *background clutter* e oclusões parciais nas cenas a partir da integração de módulos em uma rede neural convolucional multitarefa. Os resultados experimentais foram comparados com aqueles apresentados por Garon, Laurendeau e Lalonde (2018). Nesses experimentos, a técnica proposta por Maroungkas et al. (2020) obteve os melhores resultados na maioria dos testes. Duas imagens ilustrativas desses experimentos podem ser vistas na Figura 19.

Figura 19 – Exemplo de modelo e de resultados obtidos pela técnica proposta em Maroungkas et al. (2020) usando um dos casos de teste do cenário *interaction-hard* da base de dados de Garon, Laurendeau e Lalonde (2018).



Fonte: Figura adaptada pelo autor partir de Maroungkas et al. (2020).

Além dos trabalhos já citados, que propõem o rastreamento 6-DOF a partir de imagens RGB-D, existem trabalhos recentes que não usam imagem de profundidade como informação de entrada, isto é, que podem realizar o rastreamento 6-DOF apenas usando imagens RGB comuns. O trabalho apresentado em Majcher e Kwolek (2020), por exemplo, propõe uma técnica de rastreamento de objetos 3D com 6-DOF a partir de sequências de imagens RGB. Para possibilitar a realização dessa tarefa, os autores propuseram uma abordagem híbrida que se inicia com a segmentação do objeto de interesse usando uma rede neural treinada antecipadamente. Após a segmentação, um filtro de partículas combinado com um PSO são usados para estimar a pose a partir da silhueta do modelo objeto alvo projetada sobre a imagem RGB de entrada correspondente. Uma desvantagem desse método é a necessidade de treinar uma rede neural para cada objeto alvo a partir de dezenas de imagens segmentadas manualmente.

Um outro exemplo que pode ser citado é o trabalho apresentado em Deng et al. (2021), no qual foi proposto um rastreador 6-DOF que pode usar como informação de entrada tanto imagens RGB-D quando apenas imagens RGB. Para realizar o rastreamento, o método combinou um filtro de partículas *Rao-Blackwellized* com uma rede *autoencoder*. A técnica foi avaliada

usando duas bases de dados, T-LESS (HODAN et al., 2017) e a YCB Video dataset (XIANG et al., 2017) e os resultados foram compatíveis com métodos de rastreamento presentes no estado da arte.

4 PSO PARA RASTREAMENTO DE OBJETOS 3D

Este capítulo apresenta os detalhes de como o PSO pode ser usado como algoritmo de otimização de múltiplas hipóteses de pose durante o rastreamento de objetos 3D arbitrários. Ele é dividido em oito seções que detalham aspectos técnicos de cada uma das etapas do método proposto. A Seção 4.1 mostra uma visão geral do método dividindo-o didaticamente em cinco etapas. A Seção 4.2 explica características dos dados de entrada necessários ao rastreamento. A Seção 4.3 apresenta como é realizado o pré-processamento dos dados de entrada, com a construção das nuvens de pontos da cena, o cálculo das normais e seleção dos pontos visíveis do modelo. A Seção 4.4 relata como as partículas são criadas e o espaço de busca é definido ao instanciar o PSO. A Seção 4.5 explica os detalhes da função de aptidão do PSO, como a correspondência entre pontos do modelo e da cena e o cálculo das distâncias. A Seção 4.6 trata do critério de parada adotado e da suavização de poses. E por fim, a Seção 4.7 apresenta informações sobre o processamento da técnica em GPU como forma de melhorar o desempenho de tempo da técnica.

4.1 VISÃO GERAL DO MÉTODO PROPOSTO

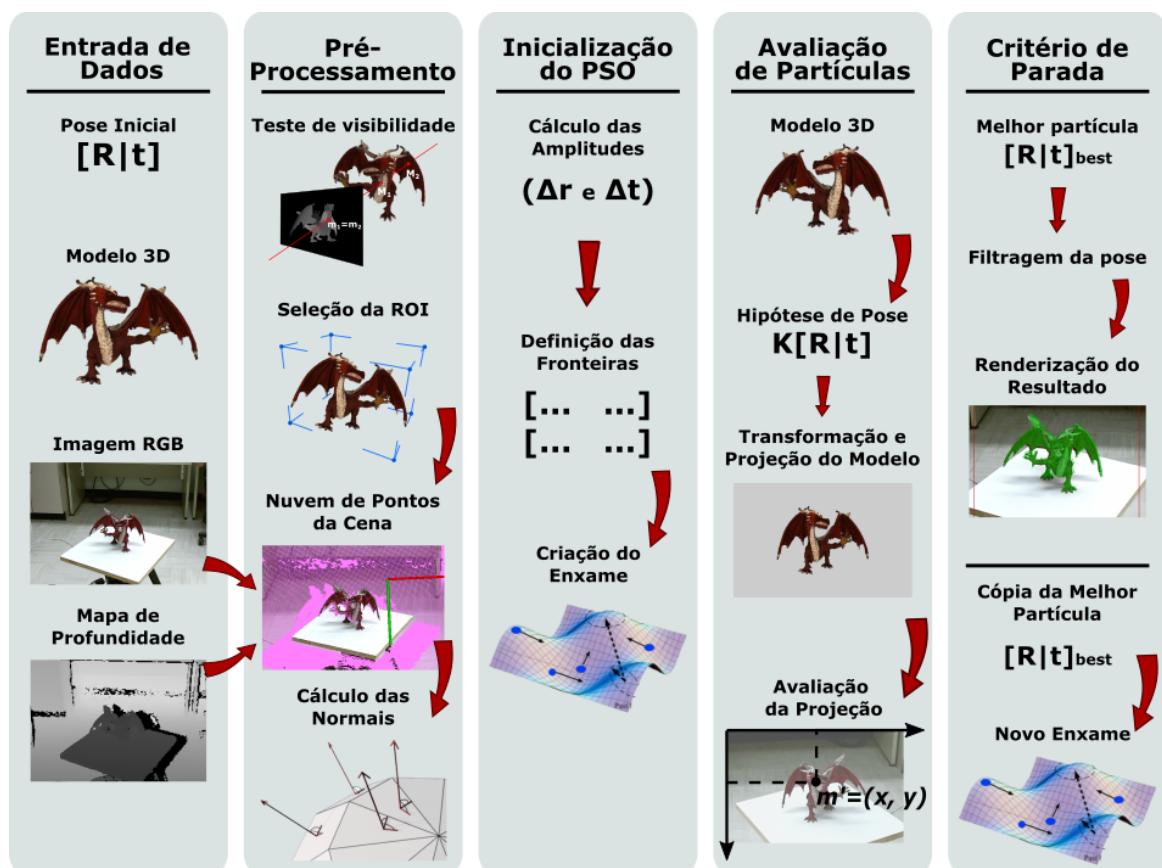
O método de rastreamento abordado neste trabalho é *top-down* e baseado em otimização, pois considera a criação e avaliação de múltiplas hipóteses de pose e usa o PSO como algoritmo para otimizá-las. É ainda baseado em modelo e realiza a avaliação das hipóteses de pose sem marcadores, visto que utiliza apenas características naturais da cena e do objeto rastreado nesse processo. A técnica considera ainda a aquisição das imagens da cena por sensores RGB-D, o que possibilita o cálculo e uso de características 3D da cena durante a predição das poses, permitindo o rastreamento de objetos 3D arbitrários, isto é, que não pertencem necessariamente a uma classe específica de objetos. A abordagem proposta baseia-se no rastreador de objetos descrito pelos autores desta tese em Júnior e Lima (2018) e apresenta uma série de melhorias em relação ao mesmo.

No processo de rastreamento da pose do objeto em um quadro q qualquer, cada hipótese de pose corresponde a uma partícula do PSO, usado para otimizar o conjunto de hipóteses candidatas. Durante o processamento do PSO, suas partículas são avaliadas usando as informações 3D das nuvens de pontos da cena. Essa avaliação é possível devido ao conhecimento

prévio do modelo do objeto rastreado. Assim, uma vez que cada partícula do PSO representa uma transformação dos pontos do modelo, a correspondência entre os pontos transformados e os pontos da nuvem da cena pode quantificar o quão próximo o modelo transformado pela hipótese de pose está da transformação real do objeto rastreado, medindo assim a qualidade dessa partícula. Esse processo aplicado a todas as partículas do enxame possibilita o PSO prever boas hipóteses de pose para o quadro analisado. Ao final o processo é repetido de forma recursiva para todos os quadros da filmagem.

Para melhor compreensão, a técnica pode ser dividida didaticamente em cinco etapas, sendo elas: entrada de dados, pré-processamento, instanciação do PSO, avaliação das partículas e critério de parada. Uma representação dessa divisão pode ser observada na Figura 20. A explicação detalhada de cada uma delas pode ser encontrada nas próximas seções desse capítulo.

Figura 20 – Representação das etapas da técnica de rastreamento abordada por este trabalho.

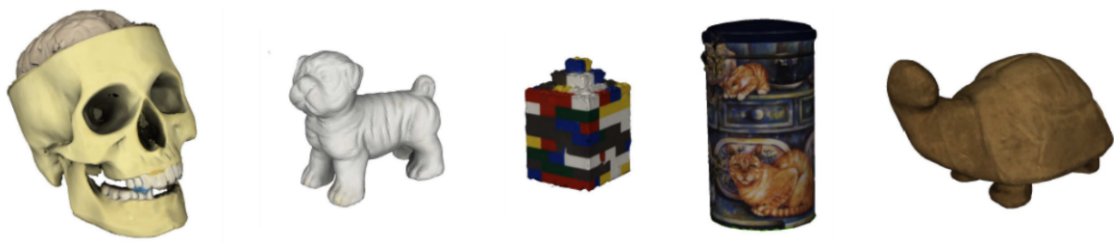


Fonte: Elaborada pelo autor (2021).

4.2 DADOS DE ENTRADA

A primeira etapa do rastreamento é onde ocorre a inserção dos dados conhecidos *a priori* pelo rastreador. São informações necessárias para a instanciação do PSO a cada quadro da filmagem e para dar início ao próprio processo de rastreamento no primeiro quadro da sequência. São basicamente três tipos de dados de entrada: a pose inicial do objeto rastreado, a imagem RGB-D da cena capturada pela câmera e o modelo 3D do objeto alvo. A representação de alguns modelos retirados da base de dados apresentada em Garon, Laurendeau e Lalonde (2018) pode ser observada na Figura 21.

Figura 21 – Exemplos de modelos de objetos 3D da base de dados de Garon, Laurendeau e Lalonde (2018).



Fonte: Figura adaptada pelo autor partir de Garon, Laurendeau e Lalonde (2018).

Uma vez que o rastreamento é um processo recursivo, ao iniciar a busca da pose do objeto em um quadro q qualquer é necessário ter conhecimento prévio da pose desse objeto no quadro anterior $q - 1$. Essa informação prévia é chamada de pose inicial, e serve para definir as amplitudes de rotação e translação e as fronteiras do subespaço de busca, onde as partículas do PSO serão criadas naquele instante. A imagem RGB-D de cada quadro contém as informações da cena capturadas pela câmera naquele instante. Essas informações são necessárias para o PSO avaliar cada hipótese de pose. Isso é feito usando o modelo que contém informações como coordenadas 3D, cores e normais dos pontos do objeto.

4.3 PRÉ-PROCESSAMENTO

Na etapa de pré-processamento ocorre a preparação dos dados de entrada antes da utilização pelo PSO. Nela, ocorre a seleção da região de interesse (ROI) da imagem RGB-D, a construção da nuvem de pontos 3D da cena, o cálculo das normais dessa nuvem e a seleção dos pontos visíveis do modelo de acordo com a pose inicial.

4.3.1 Construção da Nuvem de Pontos da Cena

Para construir a nuvem de pontos de uma cena, considere $\mathbf{m}$ 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 $C = (c_x, c_y, f)$, o ponto $\mathbf{M}$ na nuvem de pontos 3D, com coordenadas no sistema da câmera, cuja projeção corresponde ao ponto $\mathbf{m}$, pode ser obtido da seguinte forma:

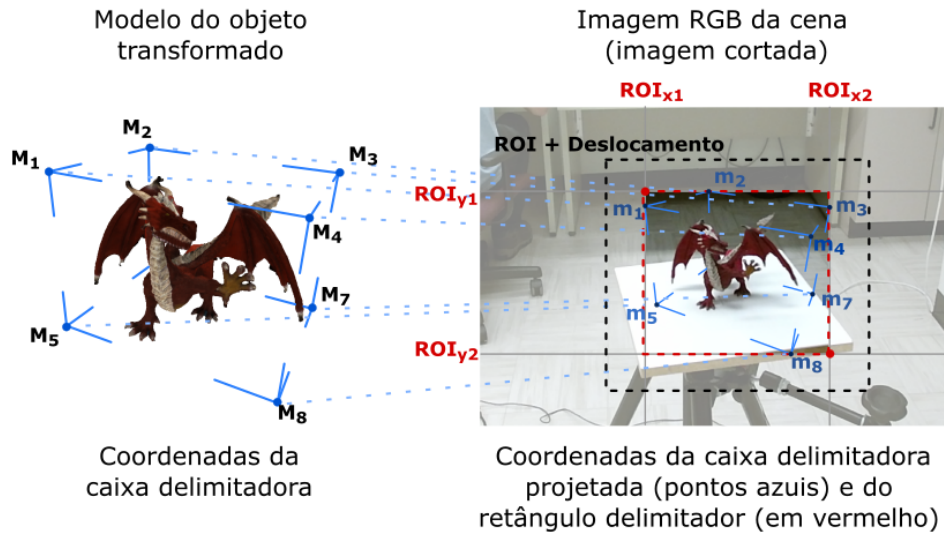
$$\mathbf{M} = \begin{bmatrix} (m_x - c_x) \frac{d}{f} \\ (m_y - c_y) \frac{d}{f} \\ d \end{bmatrix}. \quad (4.1)$$

Fazendo isso para todos os pontos da imagem RGB-D, é possível obter todos os pontos da nuvem 3D equivalente. A partir dessa nuvem de pontos é possível extrair informações 3D da cena.

Durante a construção da nuvem de pontos da cena, a caixa delimitadora de eixo alinhado do modelo pode ser usada para definir uma ROI da imagem RGB-D da cena. A posição da ROI em relação à imagem RGB-D de entrada em cada novo quadro é definida a partir da pose do objeto obtida no quadro processado anteriormente (pose inicial). Os vértices $\mathbf{M}_i$ da caixa delimitadora alinhada ao eixo do modelo são transformados e projetados de acordo com a pose inicial. Então as coordenadas 2D $\mathbf{m}_i$ dessas projeções são usadas para encontrar um retângulo delimitador de eixo alinhado na imagem RGB-D, definindo assim a ROI que será usada. Durante a realização dos experimentos (Capítulo 5), para garantir que o objeto ficasse completamente contido na ROI, o tamanho do retângulo delimitador ainda foi aumentado usando um deslocamento de 50 pixels para objetos regulares e 80 pixels para objetos alongados da base de Garon, Laurendeau e Lalonde (2018), esses valores foram definidos empiricamente. A Figura 22 ilustra esse processo.

A seleção de uma ROI para cada quadro evita o uso de regiões da imagem RGB-D da cena que não possuem informações úteis para o rastreamento. Isso é diferente do que fizemos em Júnior e Lima (2018), em que o uso de quadros completos causa o processamento e alocação de memória de regiões desnecessárias da imagem. Isso acontece em passos do rastreamento que possuem um impacto considerável no tempo de execução da técnica, tais como construção da nuvem de pontos da cena, cálculo dos vetores normais e operações de alocação/*upload* de memória na GPU.

Figura 22 – Seleção de ROI em uma imagem RGB da cena a partir da transformação e projeção dos vértices da caixa delimitadora de eixo alinhado.



Fonte: Elaborada pelo autor (2021).

4.3.2 Vetores Normais e Sistemas de Cores

As coordenadas 3D da nuvem de pontos da cena são usadas para calcular as normais de cada um de seus pontos. 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 $\mathbf{n}_i$ para cada ponto $\mathbf{M}_i$ encontrando o vetor médio $\overline{\mathbf{M}}$ dos n pontos vizinhos de $\mathbf{M}_i$ e com isso calcular a matriz de covariância $\mathbf{C}$ desses pontos usando

$$\mathbf{C} = \frac{1}{n} \sum_{i=1}^n (\mathbf{M}_i - \overline{\mathbf{M}})(\mathbf{M}_i - \overline{\mathbf{M}})^T. \quad (4.2)$$

A vizinhança de $\mathbf{M}_i$ pode ser definida por um raio ou pelo número n de vizinhos mais próximos de $\mathbf{M}_i$. De posse da matriz $\mathbf{C}$, é possível encontrar seus autovalores λ_j e os respectivos autovetores $\mathbf{v}_j$ definidos como $\mathbf{C} \cdot \mathbf{v}_j = \lambda_j \cdot \mathbf{v}_j$, com $j \in \{0, 1, 2\}$. Assim, uma vez que a direção do eixo que minimiza a variância entre os pontos é o autovetor $\mathbf{v}_k$ cujo autovalor λ_k correspondente é o menor dentre os três autovalores calculados, o vetor normal $\mathbf{n}_i$ corresponde a esse autovetor $\mathbf{v}_k$ (BERKMANN; CAELLI, 1994).

Então, assim como o modelo, a nuvem de pontos da cena passa a ter além das coordenadas 3D e cores, as normais de seus pontos. O sistema de cores dos pontos da nuvem recém criada e do modelo 3D do objeto ainda 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 $[0, 1]$. Isso é

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

4.3.3 Seleção de Pontos Visíveis do Modelo

Ao usar um modelo 3D para avaliar um conjunto de hipóteses de pose, é importante determinar quais são seus pontos visíveis em relação à câmera no momento da avaliação. Pontos não visíveis, ou simplesmente *pontos oclusos*, do objeto real no momento da captura da imagem RGB-D naturalmente não aparecem no plano de imagem da câmera e, portanto, não estão contidos na nuvem de pontos 3D da cena usada no rastreamento. Dessa forma, ao usar regiões oclusas do modelo transformado durante a avaliação de uma hipótese de pose, os pontos dessas regiões não terão correspondentes corretos na nuvem da cena, por mais que a pose esteja correta. É desejado que esse tipo de correspondência seja evitada, pois diminui a acurácia do rastreamento, uma vez que adiciona ruídos indesejáveis no cálculo da qualidade dessa pose (ver mais sobre o cálculo da qualidade de uma pose na Seção 4.5).

Neste trabalho são definidos três tipos de oclusão: a oclusão do tipo 1, que é a mais comum, pois ocorre com aqueles pontos que não estão visíveis simplesmente por pertencerem à superfície oposta ao lado do objeto que a câmera está filmando; a oclusão do tipo 2, que trata-se da auto-occlusão, ocorrendo quando o ponto ocluso pertence a uma região que está sendo encoberta por outras partes do próprio objeto, ou seja, o ponto não fica visível apesar de pertencer a uma superfície que no momento da captura está virada na direção da câmera; e a oclusão do tipo 3, que ocorre quando a oclusão é causada por outros objetos pertencentes à cena.

Em Júnior e Lima (2018), nós realizamos o teste de visibilidade de um ponto do modelo medindo o ângulo entre sua normal e a direção de vista. Entretanto, essa abordagem não lida corretamente com oclusões dos tipos 2 e 3. Neste trabalho nós adotamos um método diferente que é capaz de lidar com tais casos. Primeiramente é utilizado o algoritmo de *buffer* de profundidade (ou *Z-buffer*) do pipeline de renderização do OpenGL. Esse algoritmo possibilita a construção de uma imagem I_d contendo o mapa de profundidade do modelo transformado pela pose inicial $[\mathbf{R}|\mathbf{t}]_{inic}$. O OpenGL utiliza nativamente o *Z-buffer* para realizar testes de profundidade e detectar oclusões durante a renderização de objetos de uma cena. O mapa de profundidade I_d , no entanto, possui valores normalizados pertencentes ao intervalo real $[0, 1]$. Antes de usá-lo é necessário converter esses valores para uma escala de profundidade

no sistema de coordenadas da câmara. Isso pode ser realizado usando a função

$$D(\mathbf{x}, I_d) = \frac{2Z_n Z_f}{Z_f + Z_n - (2I_d(\mathbf{x}) - 1)(Z_n - Z_f)}, \quad (4.3)$$

em que Z_n e Z_f são respectivamente as distâncias dos planos de corte próximo e distante da câmara e $\mathbf{x} = [x, y]^T$ a localização 2D do pixel em I_d (TJADEN et al., 2018).

De posse do mapa de profundidade I_d com a escala devidamente ajustada, são calculadas então as coordenadas de projeção de todos os pontos do modelo transformado usando a matriz de pose inicial: $\tilde{\mathbf{m}} = \mathbf{P}_{inic} \tilde{\mathbf{M}}$. Assim, as coordenadas de projeção $\mathbf{m}_i$ de um ponto $\mathbf{M}_i = [x, y, z]^T$ qualquer do modelo são utilizadas para determinar o valor $I_d(\mathbf{m}_i)$ de seu correspondente no mapa de profundidade. O teste de visibilidade do ponto então é realizado comparando esse valor de profundidade $I_d(\mathbf{m}_i)$ com a coordenada z do ponto $\mathbf{M}_i$ analisado da seguinte forma:

$$V(\mathbf{M}_i, I_d(\mathbf{m}_i)) = \begin{cases} visivel & se \mathbf{M}_i(z) < I_d(\mathbf{m}_i) + S \\ não visivel & se \mathbf{M}_i(z) \geq I_d(\mathbf{m}_i) + S \end{cases}, \quad (4.4)$$

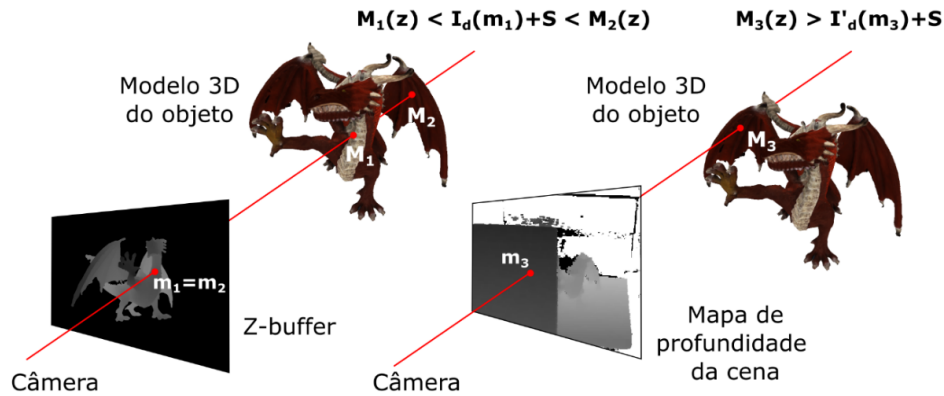
em que S é um pequeno deslocamento para evitar *Z-fighting*.

O procedimento descrito pela função na Equação 4.4 tem o objetivo de evitar as oclusões do tipo 1 e do tipo 2, porém não evita a oclusão do tipo 3. Uma vez que esse tipo de oclusão é causado por outros objetos do ambiente, informações apenas do modelo do objeto rastreado não são suficientes para detectá-la. Para isso é necessário obter informações da cena através das imagens RGB-D capturadas. Assim, para identificar os pontos do modelo que possuem oclusão do tipo 3, a coordenada z do ponto $\mathbf{M}_i$ é comparada com o valor correspondente $I'_d(\mathbf{m}_i)$ no mapa de profundidade da cena no quadro q :

$$V(\mathbf{M}_i, I'_d(\mathbf{m}_i)) = \begin{cases} visivel & se \mathbf{M}_i(z) < I'_d(\mathbf{m}_i) + S \\ não visivel & se \mathbf{M}_i(z) \geq I'_d(\mathbf{m}_i) + S \end{cases}. \quad (4.5)$$

Para melhor compreensão desse processo, uma ilustração da relação entre um ponto do modelo e seu correspondente no *Z-buffer* e no mapa de profundidade da cena é mostrada na Figura 23.

Figura 23 – Teste de visibilidade para filtragem de pontos oclusos por oclusão do tipo 1 e 2 (esquerda) ou oclusão do tipo 3 (direita).



Fonte: Elaborada pelo autor (2021).

4.4 INSTANCIACÃO DO PSO

Com a disponibilidade dos dados de entrada e todos os passos do pré-processamento realizados, um PSO então pode ser instanciado. Nele, cada partícula do enxame representa uma hipótese de pose $\mathbf{p}_i$ do objeto rastreado no quadro q . Essas partículas são criadas aleatoriamente dentro de um subespaço de busca composto por possíveis soluções do problema e limitado por fronteiras definidas a partir da pose inicial $\mathbf{p}_{(q-1)} = (\omega_1, \omega_2, \omega_3, t_1, t_2, t_3)$ e de amplitudes de rotação Δ_ω e translação Δ_t preestabelecidas. Assim, a cada novo quadro o subespaço de busca do PSO é determinado pelos intervalos $[\omega_k + \Delta_\omega, \omega_k - \Delta_\omega]$ para a rotação e $[t_k + \Delta_t, t_k - \Delta_t]$ para a translação, com $k = \{1, 2, 3\}$.

No método proposto em Júnior e Lima (2018), adotamos fronteiras que definem um subespaço de busca de tamanho fixo. Isso ocorre quando os valores de amplitude Δ_ω e Δ_t são constantes durante todo o processo de rastreamento. Porém, isso não é intuitivo, uma vez que o rastreamento de objetos lentos deve demandar subespaços de busca menores que o rastreamento de objetos rápidos. Uma alternativa é o cálculo das amplitudes separadamente para cada componente de rotação e translação a cada quadro em tempo de execução, observando a direção, o sentido e a magnitude do vetor que representa o movimento do objeto para definir os limites inferiores e superiores de cada intervalo. Assim, a técnica passa a lidar com um subespaço de busca de tamanho variável, com *amplitudes dinâmicas*. Nesse caso, o valor da amplitude da componente que aponta para a mesma direção e sentido do movimento do objeto pode aumentar no decorrer do movimento, ou diminuir quando apontar para o sentido

oposto. Isso faz com que os limites do subespaço de busca sejam escolhidos de forma mais eficiente, adequando seu tamanho ao movimento do objeto ao longo do rastreamento.

As amplitudes dinâmicas são calculadas para cada componente da pose de forma independente, portanto são representadas por um vetor $\mathbf{r} = (\Delta_{\omega_1}, \Delta_{\omega_2}, \Delta_{\omega_3}, \Delta_{t_1}, \Delta_{t_2}, \Delta_{t_3})$ de seis dimensões, em que as três primeiras correspondem às amplitudes das componentes de rotação e as demais às amplitudes das componentes de translação. Para calcular o vetor $\mathbf{r}$ em um quadro q qualquer (com $q > 3$), primeiro são criados os vetores $\mathbf{v}_1$ e $\mathbf{v}_2$, que armazenam as variações das componentes das poses obtidas entre os três últimos quadros, da seguinte forma:

$$\mathbf{v}_1(x) = \mathbf{p}_{q-2} - \mathbf{p}_{q-3} \quad (4.6)$$

$$\mathbf{v}_2(x) = \mathbf{p}_{q-1} - \mathbf{p}_{q-2} \quad (4.7)$$

Observe que a pose $\mathbf{p}_{(q-1)}$ corresponde à pose inicial usada no quadro q , e que os vetores $\mathbf{v}_1$ e $\mathbf{v}_2$ representam cada um deles a variação das componentes das poses por quadro. Para medir a variação entre os vetores $\mathbf{v}_1$ e $\mathbf{v}_2$, e saber se o objeto está acelerando positivamente ou negativamente na direção de cada componente, o vetor $\mathbf{a}$ é definido da seguinte forma:

$$\mathbf{a}(x) = \mathbf{v}_2(x) - \mathbf{v}_1(x) \quad (4.8)$$

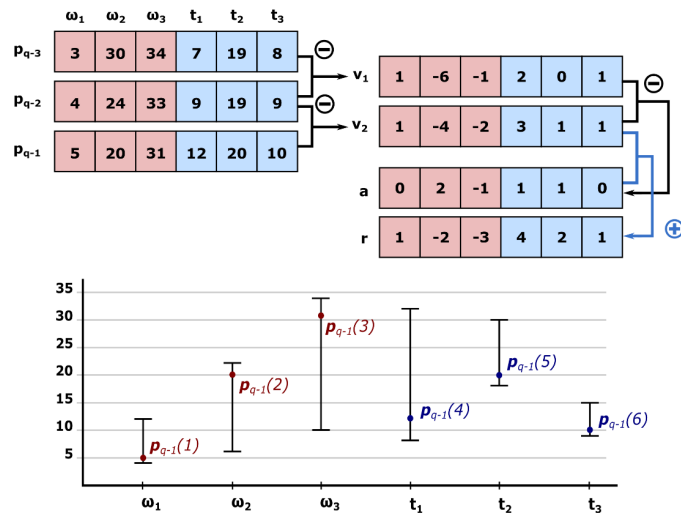
Assim, as amplitudes de rotação e translação são calculadas segundo a última variação das componentes da pose do objeto e a variação $\mathbf{a}$:

$$\mathbf{r}(x) = \mathbf{v}_2(x) + \mathbf{a}(x) \quad (4.9)$$

Desse modo, os intervalos que determinam o subespaço de busca do PSO com as amplitudes dinâmicas são $[\omega_k + \alpha \mathbf{r}(k), \omega_k - \mathbf{r}(k)]$ para a rotação e $[t_k + \beta \mathbf{r}(k+3), t_k - \mathbf{r}(k+3)]$ para a translação, com $k = \{1, 2, 3\}$. Temos que α e β são constantes que, se configuradas com valores $\alpha, \beta > 1$, garantem amplitudes maiores na direção e sentido do movimento que o objeto está fazendo no momento do cálculo. Note ainda que a subtração dos valores $-\mathbf{r}(k)$ ou $-\mathbf{r}(k+3)$ em cada componente de rotação e translação nos intervalos define uma amplitude menor no sentido oposto ao movimento do objeto. A Figura 24 mostra uma representação desse processo.

Quando usadas as amplitudes dinâmicas, ainda é necessário definir limites mínimos e máximos de amplitudes de rotação e translação. Isso impede que o subespaço de busca se torne

Figura 24 – Exemplo do cálculo dos limites do subespaço de busca para um quadro q a partir da pose encontrada pelo rastreador no quadro $q - 1$ (valores inteiros usados apenas para propósitos didáticos).



Fonte: Elaborada pelo autor (2021).

muito pequeno quando as componentes do vetor r adquire valores muito baixos, o que pode acarretar na criação de subespaços que não contenham suas respectivas poses ótimas, ou se torne muito grande, o que ocorre quando r adquire valores elevados, fazendo com que o PSO tenha dificuldade de encontrar poses boas devido ao tamanho exagerado do subespaço de busca. Na fase experimental (Capítulo 5), para realização dos testes propostos, esses valores foram definidos de forma empírica.

Com o subespaço de busca definido, o enxame então é criado dentro dele. Cada partícula do PSO corresponde a um ponto escolhido aleatoriamente dentro desse subespaço de soluções usando uma distribuição uniforme em cada uma de suas componentes, com exceção da partícula que representa a pose inicial, essa é inserida como melhor partícula do enxame já na primeira iteração. Ao término da criação do enxame, as partículas então passam a se mover nesse espaço, em busca de boas soluções. Apoiado em regras bem definidas do PSO (ver a Seção 2.4), esse movimento ocorre de acordo com a qualidade da solução representada por cada partícula em relação às demais, métrica que é atualizada a cada iteração pela função de aptidão que avalia cada uma das partículas observando sua posição no espaço de busca. Além de definir as fronteiras e criar o enxame, é durante a instanciação do PSO que são definidos todos os seus parâmetros de configuração, tais como topologia, critério de parada, número de máximo de iterações, tamanho do enxame, parâmetros cognitivos e sociais, peso de inércia, dentre outros.

4.5 AVALIAÇÃO DAS PARTÍCULAS

A etapa de avaliação das partículas é realizada pela função de aptidão do PSO. Essa função avalia todo o enxame a cada nova iteração do algoritmo, dando uma espécie de “nota” às suas partículas. Essa “nota” é conhecida como *aptidão da partícula* e tem como objetivo medir a qualidade da solução que ela representa no momento da avaliação em relação às demais. A acurácia das soluções que o PSO consegue encontrar ao final da otimização depende muito da heurística usada na função de aptidão, ou seja, de quão bem ela representa matematicamente o problema real. No contexto do rastreamento de objetos 3D, uma partícula é considerada boa na medida em que expressa uma hipótese de pose mais próxima o possível da pose real do objeto alvo no instante que o quadro usado na avaliação foi capturado.

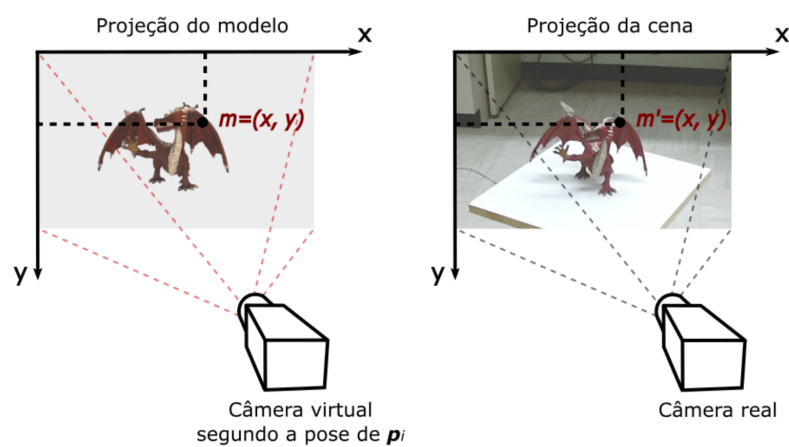
Durante a avaliação de uma partícula $\mathbf{p}_i$ qualquer do enxame, a função de aptidão precisa receber como entrada a ROI da nuvem de pontos 3D da cena, os pontos visíveis do modelo 3D do objeto segundo a pose da partícula inicial e o vetor que representa a hipótese de pose da partícula avaliada. A aptidão da partícula então é calculada através da comparação entre os pontos do modelo transformado pela hipótese de pose de $\mathbf{p}_i$ e a representação 3D do objeto real contida na nuvem de pontos da cena. Para melhor compreensão de como isso é realizado, essa tarefa pode ser dividida em duas etapas: a primeira trata da criação da correspondência entre os pontos visíveis do modelo transformado e os pontos da nuvem de pontos 3D da cena; a segunda etapa, uma vez obtidas essas correspondências, refere-se ao cálculo das distâncias, definidas como a diferença entre os pontos do modelo transformado e seus correspondentes na nuvem de pontos da cena. Essas duas etapas são explicadas com mais detalhes nas subseções a seguir.

4.5.1 Correspondência entre Pontos

Dado um ponto $\mathbf{M}_i$ do modelo do objeto rastreado, para encontrar o seu ponto correspondente $\mathbf{M}'_i$ na nuvem de pontos da cena durante a avaliação de uma partícula $\mathbf{p}_j$ qualquer, primeiramente suas coordenadas 3D são representadas como coordenadas homogêneas $\tilde{\mathbf{M}}_i = [X, Y, Z, 1]^T$, para então ser transformado e projetado, obtendo as coordenadas 2D homogêneas $\tilde{\mathbf{m}}_i = [x, y, 1]^T$ correspondentes, esse cálculo é realizado usando a Equação 2.10, na qual a matriz de $\mathbf{P}_j$ é obtida através dos parâmetros extrínsecos contidos na pose da partícula $\mathbf{p}_j$ avaliada e dos parâmetros intrínsecos da câmera $\mathbf{K}$ usada.

Feito isso para todos os pontos do modelo, a localização de cada um deles no plano de projeção é dada por um par ordenado $\mathbf{m} = (x, y)$. Processo semelhante ocorre com os pontos da nuvem de pontos 3D da cena no momento da captura, em que cada um dos seus pontos $\mathbf{M}'$ possui uma localização 2D no plano de imagem da câmera, representada aqui pelo par $\mathbf{m}' = (x, y)$. Dessa forma, usando as coordenadas x e y de cada ponto projetado $\mathbf{m}$ do modelo, é possível descobrir seu correspondente $\mathbf{m}'$ na projeção da cena, uma ilustração dessa correspondência pode ser vista na Figura 25.

Figura 25 – Correspondência entre um ponto $\mathbf{m}$ do modelo projetado e o seu par $\mathbf{m}'$ na projeção da cena.



Fonte: Elaborada pelo autor (2021).

É importante notar que a correspondência $\mathbf{m}_i \leftrightarrow \mathbf{m}'_i$ entre os pontos dos planos de projeção também é válida para os respectivos pontos 3D $\mathbf{M}$ e $\mathbf{M}'$ do modelo e da nuvem de pontos da cena. Observe ainda que a semelhança entre um ponto $\mathbf{M}_i$ qualquer e seu correspondente $\mathbf{M}'_i$ vai depender da qualidade da projeção do modelo feita pela matriz de projeção $\mathbf{P}_i$ usada, ou seja, quanto mais próximo a hipótese de pose for da pose real do objeto naquele instante, mais semelhantes serão esses pontos correspondentes e menor será a distância.

4.5.2 Cálculo da Distância

Dado um par de pontos 3D correspondentes, a distância entre eles é calculada a partir de três características naturais, sendo elas: as coordenadas 3D, representadas aqui pelos próprios vetores $\mathbf{M}_k$ e $\mathbf{M}'_k$, os vetores normais $\mathbf{n}_k$ e $\mathbf{n}'_k$ e os vetores $\mathbf{c}_k$ e $\mathbf{c}'_k$ com as componentes das cores em HSV normalizado. Para esse cálculo são usadas três funções d_p , d_c e d_n , em que as duas primeiras verificam, respectivamente, a distância euclidiana entre as coordenadas 3D e entre as cores dos pares de pontos correspondentes, e a terceira, o ângulo entre as normais.

Essas funções são descritas da seguinte forma:

$$d_p(\mathbf{M}_k, \mathbf{M}'_k) = \begin{cases} \|\mathbf{M}_k - \mathbf{M}'_k\| & \text{se } \|\mathbf{M}_k - \mathbf{M}'_k\| \leq l \\ 1 & \text{se } \|\mathbf{M}_k - \mathbf{M}'_k\| > l \end{cases}, \quad (4.10)$$

$$d_c(\mathbf{c}_k, \mathbf{c}'_k) = \|\mathbf{c}_k - \mathbf{c}'_k\|, \quad (4.11)$$

$$d_n(\mathbf{n}_k, \mathbf{n}'_k) = \frac{\cos^{-1}(\mathbf{n}_k^T \mathbf{n}'_k)}{\pi}. \quad (4.12)$$

Na Equação 4.10 um limiar l ainda é usado para tornar a d_p constante quando a distância entre os pontos retornada for maior que esse parâmetro. Isso impede que d_p retorne valores altos demais em relação às demais funções.

Em Júnior e Lima (2018), nós obtivemos o valor de aptidão computando a média aritmética da soma das distâncias de pares de pontos com pesos. Neste trabalho, descobrimos que usar a média harmônica da soma das distâncias de pares de pontos com pesos provê resultados mais robustos. Ela permite lidar melhor com casos em que há uma pequena quantidade de pontos visíveis, o que frequentemente acontece quando o objeto sofre oclusões severas. Desta forma, a partir do cálculo das distâncias para cada um dos N pares de pontos correspondentes, a aptidão da partícula é definida como

$$soma = \sum_{k=1}^N \frac{1}{\lambda_p \cdot d_p(\mathbf{M}_k, \mathbf{M}'_k) + \lambda_c \cdot d_c(\mathbf{c}_k, \mathbf{c}'_k) + \lambda_n \cdot d_n(\mathbf{n}_k, \mathbf{n}'_k)}, \quad (4.13)$$

$$aptidão = \frac{N}{soma}. \quad (4.14)$$

A média harmônica foi usada neste trabalho devido a sua propriedade de tender para os menores elementos em um conjunto de números. Ela é a menor média dentre as médias pitagóricas. Uma vez que as menores distâncias retornadas pelas funções d_p , d_c e d_n ocorrem quando avalia-se uma hipótese de pose que está próxima à pose real, o uso da média harmônica tende a beneficiar boas hipóteses e penalizar as ruins. Durante o cálculo do somatório na Equação 4.13, as funções d_p , d_c e d_n ainda têm seus valores de retorno ponderados pelas constantes λ_p , λ_c e λ_n . Esses valores são calculados no primeiro quadro da sequência, tal que

$$\lambda_p = 1, \quad (4.15)$$

$$\lambda_c = \frac{\sum_{k=1}^N d_p(\mathbf{M}_k, \mathbf{M}'_k)}{\sum_{k=1}^N d_c(\mathbf{c}_k, \mathbf{c}'_k)}, \quad (4.16)$$

$$\lambda_n = \frac{\sum_{k=1}^N d_p(\mathbf{M}_k, \mathbf{M}'_k)}{\sum_{k=1}^N d_n(\mathbf{n}_k, \mathbf{n}'_k)}. \quad (4.17)$$

e são usados para equilibrar a influência das características no valor final da aptidão durante todo o rastreamento, fazendo com que cada uma delas tenham pesos aproximadamente iguais na composição final da média harmônica.

4.6 CRITÉRIO DE PARADA

A última etapa do processo de otimização é o critério de parada do PSO e a filtragem da pose encontrada. O critério de parada estabelece quando deve acontecer o encerramento da busca realizada pelo enxame. É desejado que todo PSO possua um critério de parada pré-definido, seja pela aptidão mínima (ou máxima) da melhor partícula do enxame, o grau de diversidade no enxame, a quantidade de iterações, ou alguma outra heurística pensada de acordo com o domínio do problema que o PSO está otimizando. Na metodologia usada por esse trabalho, o critério de parada é definido por um número máximo de iterações do PSO, que pode mudar de acordo com os objetivos de cada experimento.

A escolha de um valor máximo de iterações ainda está associada a critérios de decisão entre acurácia e desempenho de tempo do rastreador, isto é, um rastreamento com maior número de iterações pode proporcionar resultados mais acurados, no entanto irá levar mais tempo para calcular as poses. Foram realizados experimentos (Seção 5.4) em que parâmetros tais como, número de iterações, diversidade populacional e aptidão da melhor partículas são usados para analisar o critério de parada do PSO no contexto desta tese.

4.6.1 Filtragem de Poses

No protocolo adotado, quando o critério de parada ocorre, a hipótese de pose da melhor partícula do enxame no momento da parada é considerada a pose p_q do objeto para o quadro q em questão. Essa pose, no entanto, pode passar por um processo de filtragem com o objetivo de diminuir o *jitter* da trajetória desenhada, podendo assim aumentar a acurácia final do rastreamento sem distorcer os resultados. Quando usada, a filtragem de poses é uma das últimas etapas de cada iteração do rastreamento. Nela a pose final em um determinado quadro corresponde à média móvel das últimas poses obtidas pelo rastreador, considerando uma amostra

de dados de tamanho fixo. Ou seja, ao passar por uma filtragem, cada uma das componentes da pose recente é recalculada usando a média ponderada das componentes correspondentes das últimas poses, usando para isso coeficientes derivados do método Savitzky-Golay (PRESS; TEUKOLSKY, 1990) de acordo com o tamanho do filtro, como pode ser visto na Tabela 1.

Tabela 1 – Coeficientes fixos para filtragem linear com diferentes tamanhos de entradas.

Nº de coeficientes	Tamanho do Filtro					
	2	3	4	5	6	7
7	0	0	0	0	0	-0,17857
6	0	0	0	0	-0,19048	-0,07143
5	0	0	0	-0,20000	-0,04762	0,03571
4	0	0	-0,20000	0	0,09524	0,14286
3	0	-0,16667	0,10000	0,20000	0,23810	0,25000
2	0	0,33333	0,40000	0,40000	0,38095	0,35714
1	1,00000	0,83333	0,70000	0,60000	0,52381	0,46429

Fonte: Elaborada pelo autor (2021)

Dessa forma, para obter a pose filtrada p em um determinado quadro q , com um filtro de tamanho 3, por exemplo, os coeficientes usados seriam os seguintes:

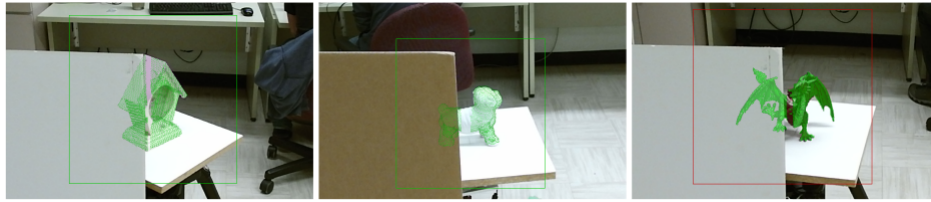
$$p = 0,833p_q + 0,333p_{q-1} - 0,166p_{q-2}. \quad (4.18)$$

Assim, a pose p em questão corresponderia à pose filtrada a partir da pose atual p_q e de suas antecessoras p_{q-1} e p_{q-2} .

4.6.2 Renderização dos Resultados

Ao obter a pose final para um determinado quadro, uma representação dessa pose é então renderizada sobre a imagem RGB do quadro para análise qualitativa do resultado do rastreamento. Na Figura 26 é possível ver a representação, em pontos verdes, da pose encontrada pelo PSO durante o rastreamento de três modelos (*clock*, *dog* e *dragon*) do cenário *occlusion* da base de dados de Geron, Laurendeau e Lalonde (2018). Uma vez que o rastreamento é um processo recursivo, a melhor partícula é usada como pose inicial no próximo quadro da sequência, sendo usada para compor o novo enxame e definir as novas fronteiras do espaço de busca no rastreamento do quadro do próximo quadro $q + 1$.

Figura 26 – Renderização de poses encontradas pelo PSO sobre imagens RGB da cena para análise qualitativa.



Fonte: Figura adaptada pelo autor partir de Garon, Laurendeau e Lalonde (2018).

4.7 PROCESSAMENTO EM GPU

Um dos problemas enfrentados por técnicas de rastreamento baseados em PSO é o desempenho de tempo (JÚNIOR; LIMA, 2018). Resolver esse tipo de problema não é algo trivial, uma vez que o rastreador baseado em PSO usa uma instância do algoritmo a cada quadro, sendo necessário a avaliação de todas as partículas do enxame a cada iteração do PSO instanciado, tornando a quantidade de chamadas da função de aptidão muito grande.

Pretendendo melhorar o desempenho de tempo do rastreador, a proposta apresentada em Júnior e Lima (2018) sugeriu a implementação da função de aptidão em GPU, fazendo assim a paralelização do algoritmo no nível dos pontos do modelo. Todas as etapas computadas na função de aptidão, tais como transformação do modelo, seleção de pontos visíveis, projeção, definição de pontos correspondentes e cálculo das distâncias, descritas na seção anterior, são processadas de forma paralela em GPU, uma vez que a realização dessa sequência de passos em um ponto do modelo não depende das propriedades dos demais pontos.

Apesar do paralelismo usado em Júnior e Lima (2018) ter melhorado o desempenho de tempo do algoritmo, o resultado ainda não possibilitou um rastreamento com altas taxas de quadros, o que em muitas situações pode ser visto como um problema, uma vez que aplicações que usam o rastreamento de objetos 3D comumente demandam desempenho em tempo real. Um possível caminho para melhorar o desempenho de tempo de técnicas de rastreamento baseadas em PSO, apresentadas nesta tese, é a paralelização não só da função de aptidão, mas também das demais partes do algoritmo do PSO passíveis de serem paralelizáveis, tais como a inicialização do enxame, a atualização das posições e velocidades das partículas, o cálculo do peso de inércia e a instanciação e atualização da matriz de comunicação.

Uma vez que a avaliação de uma partícula qualquer não depende dos estados das partículas restantes nessa mesma iteração, é possível ainda realizar o paralelismo do PSO no nível de suas partículas e não apenas no nível dos pontos do modelo como realizado em Júnior e Lima

(2018). Com essas mudanças, é possível disponibilizar um PSO com algoritmo totalmente processado em GPU. Dessa forma, algumas vantagens em relação ao desempenho de tempo da técnica podem ser observadas:

- A diminuição da frequência de comunicação entre as memórias do *host* e do *device*. Quando parte do PSO é implementada em CPU, as trocas de dados entre o *host* e o *device* sobre o enxame ocorrem com uma frequência elevada, ou seja, antes e depois da avaliação de cada partícula do enxame em cada iteração. Com a implementação do PSO completamente em GPU, a comunicação é necessária apenas em dois momentos, na instanciação do PSO e após seu critério de parada ser atingido;
- O cálculo em paralelo nas Equações 4.13 e 4.14 para a composição da aptidão. Quando realizado em CPU, o somatório e a contagem da quantidade de pontos correspondentes N são realizados de forma sequencial;
- A geração dos números aleatórios necessários para a atualização das velocidades em GPU realizada de forma paralela;
- Paralelização de outras partes do algoritmo do PSO, como a instanciação do PSO, a criação das partículas com posição e velocidade no espaço de soluções e a atualização dos estados de suas partículas.

Para realizar o processamento do PSO em GPU, o modelo do objeto rastreado e a nuvem de pontos da cena são alocados e copiados no *device*. Durante a instanciação do PSO, todas as matrizes e vetores utilizados no processamento do enxame também são alocados no *device*. Uma vez que o algoritmo do PSO possui etapas não paralelizáveis, as mesmas são implementadas em vários *kernels* que são chamados conforme a ordem de execução do algoritmo. Nessa ordem, os *kernels* realizam as seguintes tarefas:

- Iniciação do PSO, em que a matriz de comunicação é criada conforme o tamanho do enxame pré-definido e as partículas são distribuídas aleatoriamente ao longo do subespaço de busca;
- Para cada iteração:
 - As partículas são avaliadas usando o modelo e a nuvem de pontos da cena (conforme explicado na Seção 4.5);

-
- A média harmônica associada à aptidão de cada partícula é calculada;
 - A posição do melhor vizinho de cada partícula é salvo;
 - O peso de inércia, a posição e velocidades das partículas são atualizadas dentro do subespaço de soluções;
- As partículas são avaliadas novamente e a melhor partícula do enxame é atualizada.

Para que o algoritmo em GPU paralelize o rastreamento no nível das partículas do PSO, o número de *threads* usadas durante as chamadas do *kernel* que avalia as partículas é definido como o produto entre o tamanho do enxame e o número de pontos do modelo. Portanto, cada par ordenado desse produto, em que sua primeira componente é uma partícula do enxame e a segunda um ponto do modelo, precisa estar associado a uma *thread* de maneira biunívoca. Assim, cada *thread* calcula a transformação, seleção, projeção e distância de apenas um ponto do modelo, segundo a hipótese de pose da partícula contida no par ordenando em questão.

5 EXPERIMENTOS E RESULTADOS

Este capítulo apresenta detalhadamente a metodologia usada durante a fase experimental deste trabalho, assim como mostra os resultados obtidos em relação a técnicas de rastreamento presentes no estado da arte. Na Seção 5.1 são apresentadas as características das bases de dados usadas e descritas as etapas seguidas durante os experimentos. A Seção 5.2 mostra a metodologia e os resultados obtidos na primeira etapa dos experimentos, em que a primeira versão do rastreador com PSO em GPU é testada e os resultados são comparados com técnicas presentes na literatura. Na Seção 5.3 é explicada a metodologia experimental da segunda etapa dos experimentos. Nessa etapa são apresentados resultados que mostram o impacto das melhorias propostas na qualidade e desempenho de tempo do rastreamento da técnica avaliada. Na Seção 5.4 é apresentado a metodologia e os resultados experimentais da terceira etapa dos experimentos, que trata de um estudo a respeito das características do enxame de partículas durante cada iteração, bem como e da forma como a função de aptidão avalia as hipóteses de pose.

5.1 DESCRIÇÃO DOS EXPERIMENTOS

Os experimentos apresentados neste capítulo têm o objetivo de verificar a qualidade da técnica de rastreamento proposta por meio da medição da acurácia e robustez dos resultados obtidos e do desempenho de tempo durante a sua execução. A análise da acurácia e robustez foi realizada tanto de forma qualitativa, por meio de imagens e vídeos contendo a renderização desses resultados, quanto de forma quantitativa, através de métricas que expressam objetivamente a diferença das poses encontradas em relação aos seus respectivos valores de *ground truth*. Os resultados obtidos são ainda usados para avaliar a qualidade da técnica em relação a outros métodos de rastreamento presentes no estado da arte, tais como Júnior e Lima (2018), Tan et al. (2015), Garon e Lalonde (2017), Garon, Laurendeau e Lalonde (2018), Kehl et al. (2017), Tan, Navab e Tombari (2017), Marougkas et al. (2020). Para possibilitar a comparação entre diferentes técnicas de forma justa, os experimentos foram realizados a partir de bases de dados públicas que também foram usadas pelas técnicas tomadas como referência.

5.1.1 Base de Dados

Todos os experimentos foram realizados usando sequências de imagens obtidas a partir de três bases de dados, apresentadas em Garon e Lalonde (2017), Garon, Laurendeau e Lalonde (2018) e Choi e Christensen (2013). A escolha dessas três bases de dados se deu por meio de características que proporcionam vantagens objetivas aos experimentos, tais como: possuir casos de testes variados, obtidos a partir de ambientes reais e sintéticos, configurados em diferentes cenários e níveis de dificuldade, proporcionando conhecer melhor tanto os pontos fortes como os limites da técnica avaliada; conter diferentes objetos 3D com características geométricas e de texturas diversas; serem bases de dados disponibilizadas publicamente, o que facilita a comparação com outras técnicas do estado da arte, assim como possibilita a replicação dos experimentos por outros pesquisadores; e ainda possuir valores de *ground truth* do movimento da câmera para todas as sequências, permitindo uma avaliação quantitativa dos resultados.

Na primeira base de dados, apresentada em Garon e Lalonde (2017), são disponibilizados 4 modelos de objetos 3D (os quatro primeiros objetos na Figura 27) usados para o rastreamento em 12 diferentes casos de teste. Desses, 8 sequências foram planejadas para possibilitar a avaliação da robustez à oclusão do rastreador. Nelas o nível de oclusão do objeto alvo é controlado através de um aparato colocado entre ele e a câmera, proporcionando oclusões na horizontal e na vertical com 10%, 20%, 30% ou 40% do objeto alvo ocluso. Essa base de dados ainda oferece mais 4 casos de teste sem oclusão controlada em cenários onde o objeto gira em uma plataforma ao mesmo tempo em que a câmera é movimentada durante a captura das imagens.

Na segunda base de dados, apresentada em Garon, Laurendeau e Lalonde (2018), são disponibilizados 11 modelos de objetos 3D com diferentes características geométricas e de textura (todos os modelos na Figura 27). Essa base possui 297 sequências de imagens divididas em três diferentes cenários. O primeiro, chamado de *stability*, pode ser usado para medir o grau de “tremido” (*jitter*) das poses encontradas durante o rastreamento de objetos que permanecem em repouso em relação à câmera. Esse cenário ainda é subdividido em três categorias: *stability-near*, com objetos a uma distância de 0,8m da câmera; *stability-far*, com objetos 1,5m distantes da câmera; e *stability-occluded*, com objetos 0,8m distantes da câmera e em que há oclusão de parte do objeto alvo. Cada uma dessas categorias foi obtida com a captura das imagens a partir de quatro pontos de vista para cada um dos objetos da base, totalizando 44 sequên-

cias em cada categoria. O segundo cenário, chamado de *occlusion*, foi criado principalmente para avaliar a robustez à oclusão do rastreador. Nele as sequências apresentam 6 níveis de oclusão tanto na horizontal quanto na vertical, variando de 0% a 75% de oclusão do objeto (sendo incrementados de 15% em 15%), totalizando 11 sequências diferentes para cada objeto. Por fim, temos o cenário *interaction*, em que o objeto é manipulado manualmente. Esse cenário é dividido em 4 categorias de acordo com a natureza do movimento realizado: a primeira, chamada de *interaction-rotation*, prioriza o movimento de rotação do objeto; a segunda, *interaction-translation*, favorece o movimento de translação; e as duas últimas, *interaction-full* e *interaction-hard*, incluem os dois tipos de movimento ao mesmo tempo, com a última contendo ainda trechos de oclusão. Essa base de dados possibilita a realização de um conjunto de experimentos mais completo, uma vez que ela possui uma quantidade maior de casos de teste e modelos, melhor acurácia dos valores de *ground truth* e maior variação nos níveis de dificuldade de rastreamento, principalmente nos casos de teste com oclusão controlada.

Figura 27 – Modelos de objetos 3D disponíveis na base de dados de Garon, Laurendeau e Lalonde (2018).



Fonte: Figura adaptada pelo autor partir de Garon, Laurendeau e Lalonde (2018).

A terceira base de dados, proposta em Choi e Christensen (2013), é composta por 4 modelos (ver Figura 28) que integra seis diferentes casos de teste, sendo dois deles obtidos a partir de cenas reais e outros quatro obtidos a partir de uma cena sintética. O uso de uma base

de dados sintética proporcionou o planejamento e execução de experimentos em ambientes mais controlados, sem ruídos nas nuvens de pontos 3D, sem imprecisões nos valores de *ground truth* das trajetórias e com modelos sintéticos que representam perfeitamente o objeto alvo do rastreamento. Nos experimentos realizados nesta tese usando essa base de dados apenas os casos de testes que possuem *ground truth* foram utilizados, ou seja, apenas os casos de testes sintéticos.

Figura 28 – Modelos de objetos 3D disponíveis na base de dados de Choi e Christensen (2013).



Fonte: Figura adaptada pelo autor partir de Choi e Christensen (2013).

5.1.2 Etapas dos Experimentos

Os resultados preliminares apresentados neste capítulo foram obtidos através de experimentos realizados em diferentes momentos da pesquisa, portanto são apresentados separadamente em três etapas. A primeira delas teve o objetivo de avaliar a qualidade do rastreador com o PSO implementado completamente em GPU, ou seja, em que o paralelismo é realizado no nível das partículas do enxame (ver Seção 4.7), usando uma função de aptidão com média harmônica das distâncias e uma ROI de tamanho fixo na determinação da parte útil da imagem RGB-D da cena. As demais partes do algoritmo foram realizadas da mesma forma que em Júnior e Lima (2018). Nessa etapa foram usadas as sequências de imagens da base de dados em Garon e Lalonde (2017) e os resultados encontrados foram comparados com aqueles obtidos em Júnior e Lima (2018), Tan et al. (2015), Garon e Lalonde (2017), Garon, Laurendeau e Lalonde (2018).

A segunda etapa foi realizada com o objetivo de avaliar o impacto na acurácia, robustez e desempenho de tempo do rastreamento referente a seis mudanças realizadas na técnica, publicadas em Júnior, Lima e Teichrieb (2021), em relação a versão do rastreador publicada em Júnior e Lima (2018), sendo elas: o cálculo da função de aptidão usando média harmônica das distâncias (ver Subseção 4.5.2); seleção de pontos visíveis do modelo que contemplam

as oclusões do tipo 1, 2 e 3 (ver Subseção 4.3.3) realizada pela comparação dos pontos do modelo transformado em relação ao seu mapa de profundidade gerado pelo *Z-buffer* do OpenGL e também em relação ao mapa de profundidade da cena; o uso de amplitudes de rotação e translação dinâmicas para o cálculo dos limites do subespaço de busca em tempo de execução (ver Seção 4.4); ROI de tamanho dinâmico na determinação da parte útil da imagem RGB-D da cena (ver Subseção 4.3.1); filtragem da trajetória em tempo de execução do rastreador (ver Subseção 4.6.1); e processamento do PSO completamente em GPU (ver Seção 4.7). Nessa etapa, os experimentos foram realizados a partir da base de dados sugerida em Garon, Laurendeau e Lalonde (2018) e os resultados foram comparados com aqueles obtidos em Júnior e Lima (2018), Tan et al. (2015), Garon e Lalonde (2017), Garon, Laurendeau e Lalonde (2018), Marougkas et al. (2020).

A terceira etapa, descrita na Seção 5.4, foi realizada com o intuito de realizar um estudo para identificar possíveis problemas e pontos de melhoria na função de aptidão e no comportamento do enxame de partículas durante a execução do PSO. Essa seção é dividida em três subseções, sendo que na primeira delas (a Subseção 5.4.1) são analisadas a diversidade populacional e a convergência do enxame, por meio da observação das mudanças nas aptidões e nas posições das partículas do enxame ao longo das iterações. Na subseção 5.4.2 é realizado um estudo a respeito da função de aptidão por meio de avaliações quantitativas dos dados gerados durante a execução do PSO e avaliações qualitativas das transformações dos pontos do modelo sobre a nuvem de pontos da cena, usando para isso as poses encontradas pelo rastreador e as poses definidas pelo *ground truth*. Na subseção 5.4.3 são apresentados experimentos de acurácia do rastreador com o objetivo de avaliar possíveis alterações no PSO, propostas com o intuito de contornar os problemas identificados nas Subseções 5.4.1 e 5.4.2. Os resultados desses testes ainda são comparados com outras técnicas de rastreamento presentes no estado da arte.

5.2 METODOLOGIA E RESULTADOS DA PRIMEIRA ETAPA

Na primeira etapa todos os experimentos foram realizados em um computador de mesa, com processador Intel Core i7 @ 3,07 GHz, 12 GB de RAM e placa de vídeo NVIDIA Tesla K20. O rastreador foi implementado em C e C++, com o suporte de diversas bibliotecas, tais

como: OpenCV¹, PCL², CUDA SDK³, GNU Scientific Library (GSL)⁴ e um PSO implementado em C⁵.

Em todos os experimentos foi usado um PSO com topologia *lbest* e peso de inércia dinâmico com valores máximo e mínimo $\mu_{max} = 0,7298$ e $\mu_{min} = 0,3$, respectivamente, baseados em Shi e Eberhart (1998). Os parâmetros social e cognitivo foram $c_1 = c_2 = 1,496$ (CLERC; KENNEDY, 2002). As amplitudes de rotação são fixas e possuem valores respectivamente de $\Delta_\omega = 0,75$ e $\Delta_t = 0,04m$. O limiar da distância euclidiana entre os pares de pontos foi $l = 0,01m$, como proposto em Choi e Christensen (2013). Foi utilizado um tamanho fixo para a ROI da nuvem de pontos da cena de 200×140 pixels. Uma vez que o tempo de processamento e a qualidade das respostas do PSO estão diretamente relacionados com o número de partículas, a quantidade de iterações e o número de pontos do modelo do objeto rastreado, os valores desses três parâmetros foram definidos empiricamente e escolhidos de acordo com o objetivo de cada experimento.

Como o número de *threads* usadas pela GPU durante o cálculo de predição da pose é igual ao produto entre o número de pontos do modelo e o de partículas do enxame, é esperado que modelos mais densos e enxames maiores proporcionem um rastreamento mais acurado, porém com um menor desempenho de tempo, e vice-versa. Portanto, ao ajustar parâmetros como o tamanho do modelo, o critério de parada e a quantidade de partículas do PSO, é possível variar a acurácia e o desempenho de tempo do rastreador em função desses valores.

Durante as comparações dos resultados nessa etapa foi escolhido um rastreador com configurações que proporcionem um rastreamento em tempo real. Para isso foi definido um limite mínimo de 20 fps. Testes realizados com frequências abaixo de 20 fps são utilizados apenas para expor os limites da técnica em relação à capacidade de rastreamento sem perdas definitivas de poses e ao desempenho de tempo. É importante ressaltar que a acurácia do rastreamento nos experimentos relatados nesta seção não é a melhor que a técnica pode proporcionar, mas sim a melhor acurácia considerando as configurações necessárias para um rastreamento em tempo real. Em outras palavras, a melhor precisão com pelo menos 20 fps.

Os experimentos foram realizados usando os casos de teste com os maiores níveis de oclusão controlada na horizontal e na vertical da base de dados de Garon e Lalonde (2017), isto é,

¹ <http://opencv.org>

² <http://pointclouds.org>

³ <https://developer.nvidia.com/cuda-toolkit>

⁴ <http://www.gnu.org/software/gsl>

⁵ <https://github.com/kkentzo/ps0>

aqueles com 40% de oclusão do objeto alvo. Foram escolhidos empiricamente dois tamanhos de enxames, com 8 e 16 partículas, e dois tamanhos de modelo obtidos a partir de amostragens dos pontos do modelo original fornecido pela base de dados. Esses modelos amostrados ficaram 10 e 20 vezes menores que o modelo original, isto é, com 4.740 e 2.370 pontos, respectivamente. O critério de parada do PSO a cada quadro foi definido empiricamente em 100 iterações.

5.2.1 Avaliação de Acurácia

Os primeiros resultados dos experimentos nesta etapa, usando um PSO com 8 partículas e o modelo *Dragon* com 4.740 pontos, foram comparados com resultados obtidos pelas técnicas de rastreamento baseadas em aprendizagem em Tan et al. (2015), Garon e Lalonde (2017) e com o método baseado em otimização de Júnior e Lima (2018). A comparação quantitativa com as técnicas em Tan et al. (2015), Garon e Lalonde (2017) é direta, pois, como reportado em Garon e Lalonde (2017) durante os experimentos, devido às repetidas perdas de pose do objeto rastreado em ambas as técnicas, esses rastreadores precisaram ser reinicializados usando informações de *ground truth* a cada 15 quadros, evitando assim penalidades exageradas na média de erros final. Uma vez que os casos de teste com oclusão de 40% na horizontal e 40% na vertical possuem respectivamente 763 e 665 quadros, isso significa que os rastreadores foram reiniciados aproximadamente 50 vezes no primeiro e 44 no segundo. Diferente desses resultados, tanto a técnica proposta nesse trabalho quanto aquela apresentada em Júnior e Lima (2018) não necessitaram ser reiniciadas em momento algum, pois não houve perdas de pose de forma definitiva ao longo de toda as sequências com oclusões, tanto na horizontal quanto na vertical, mesmo naquelas que usaram 40% do objeto ocluso.

Uma avaliação qualitativa pode ser realizada através das imagens nas Figuras 29 e 30, em que alguns quadros com os resultados dos experimentos com 40% de oclusão na horizontal e na vertical obtidos pela técnica proposta são comparados com aqueles obtidos pelos métodos em Tan et al. (2015), Garon e Lalonde (2017). Os quadros em cada linha correspondem aos resultados em diferentes instantes da sequência. Nos quadros correspondentes ao resultado da técnica proposta, os pontos do modelo projetados na cor verde são aqueles com uma distância euclidiana maior que o limiar l , já os azuis são aqueles com uma distância euclidiana menor ou igual a esse limiar. O retângulo vermelho demarca a ROI da imagem usada para construir a nuvem de pontos 3D da cena a cada novo quadro. A diferença de qualidade entre cada uma das técnicas pode ser averiguada com mais detalhes no vídeo resultados etapa 1 do material

suplementar no Apêndice A.

A partir das imagens nas Figuras 29 e 30, e do vídeo *resultados etapa 1* adicionado ao material suplementar deste trabalho (Apêndice A), podemos afirmar que a técnica proposta possibilitou um resultado qualitativamente melhor em relação às demais. Das três técnicas avaliadas, aquela proposta por Tan et al. (2015) foi a que apresentou o resultado mais impreciso, perdendo a pose com bastante facilidade logo após ser reiniciada, principalmente no caso de teste com oclusão na horizontal.

É possível também realizar uma avaliação quantitativa a partir dos erros médios de rotação e de translação obtidos pelo rastreador de cada uma das técnicas e apresentados em Garon e Lalonde (2017). As métricas usadas para calcular a acurácia de cada pose encontrada foram aquelas adotadas por Garon e Lalonde (2017), em que a diferença entre uma pose $\mathbf{p}_i$ e seu respectivo *ground truth* $\mathbf{g}_i$ é obtida usando

$$\mathbf{d}_i = |\mathbf{p}_i(k) - \mathbf{g}_i(k)|, \quad (5.1)$$

com as rotações representadas em ângulos de Euler. A partir da diferença encontrada em $\mathbf{d}_i$, é calculada o erro de rotação δ_R da pose avaliada usando a média das três primeiras componentes de $\mathbf{d}_i$

$$\delta_R = \frac{\mathbf{d}_i(1) + \mathbf{d}_i(2) + \mathbf{d}_i(3)}{3}, \quad (5.2)$$

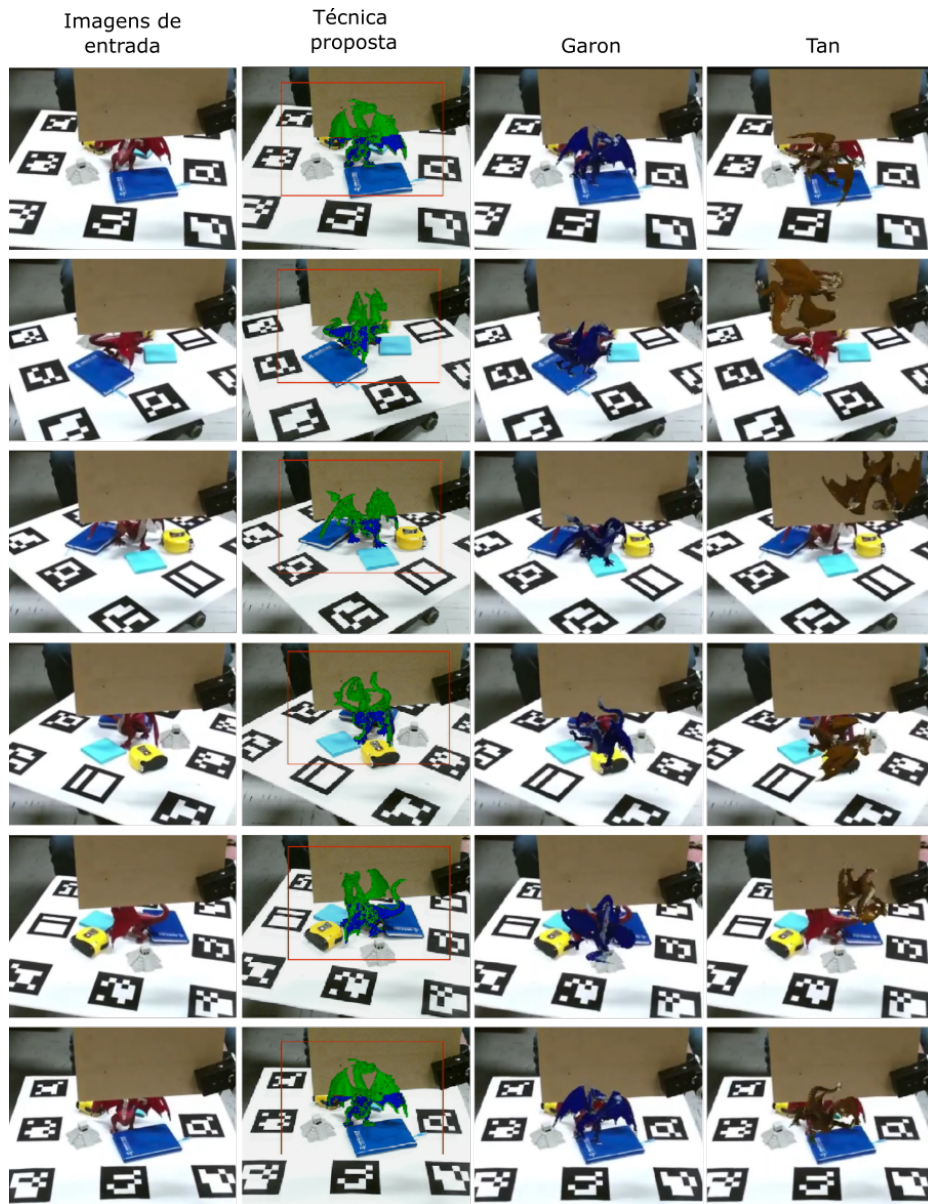
e o erro de translação δ_t a partir das três últimas componentes

$$\delta_t = \frac{\mathbf{d}_i(4) + \mathbf{d}_i(5) + \mathbf{d}_i(6)}{3}. \quad (5.3)$$

Uma vez que em Garon e Lalonde (2017) os resultados experimentais obtidos usando os casos de teste com oclusão foram apresentados apenas através de gráficos de caixa, não é possível conhecer o valor exato das médias de erro de cada técnica. Dessa forma, como pode ser visto na Tabela 2, a comparação foi realizada usando intervalos. Para não correr o risco de fazer comparações injustas, o limite inferior de cada intervalo foi arredondado sempre para um valor inteiro menor mais próximo.

Então, a partir dos resultados apresentados na Tabela 2, é possível observar que a técnica proposta obteve melhores resultados em todos os experimentos realizados usando duas configurações do rastreador, a primeira com um PSO de 8 partículas e o modelo *Dragon* amostrado com 4.740 pontos e a segunda com um PSO contendo 16 partículas e o modelo *Dragon* com 2.370 pontos, ambas possibilitando um rastreamento em tempo real. Mesmo reiniciando os rastreadores a cada 15 quadros, as técnicas usadas como referência obtiveram médias de erros

Figura 29 – Comparação dos resultados usando os casos de teste do *Dragon* com 40% de oclusão na horizontal de Garon e Lalonde (2017). Imagens de entrada (1ª coluna), resultados da técnica proposta (2ª coluna), resultados de Garon e Lalonde (2017) (3ª coluna) e resultados de Tan et al. (2015) (4ª coluna).

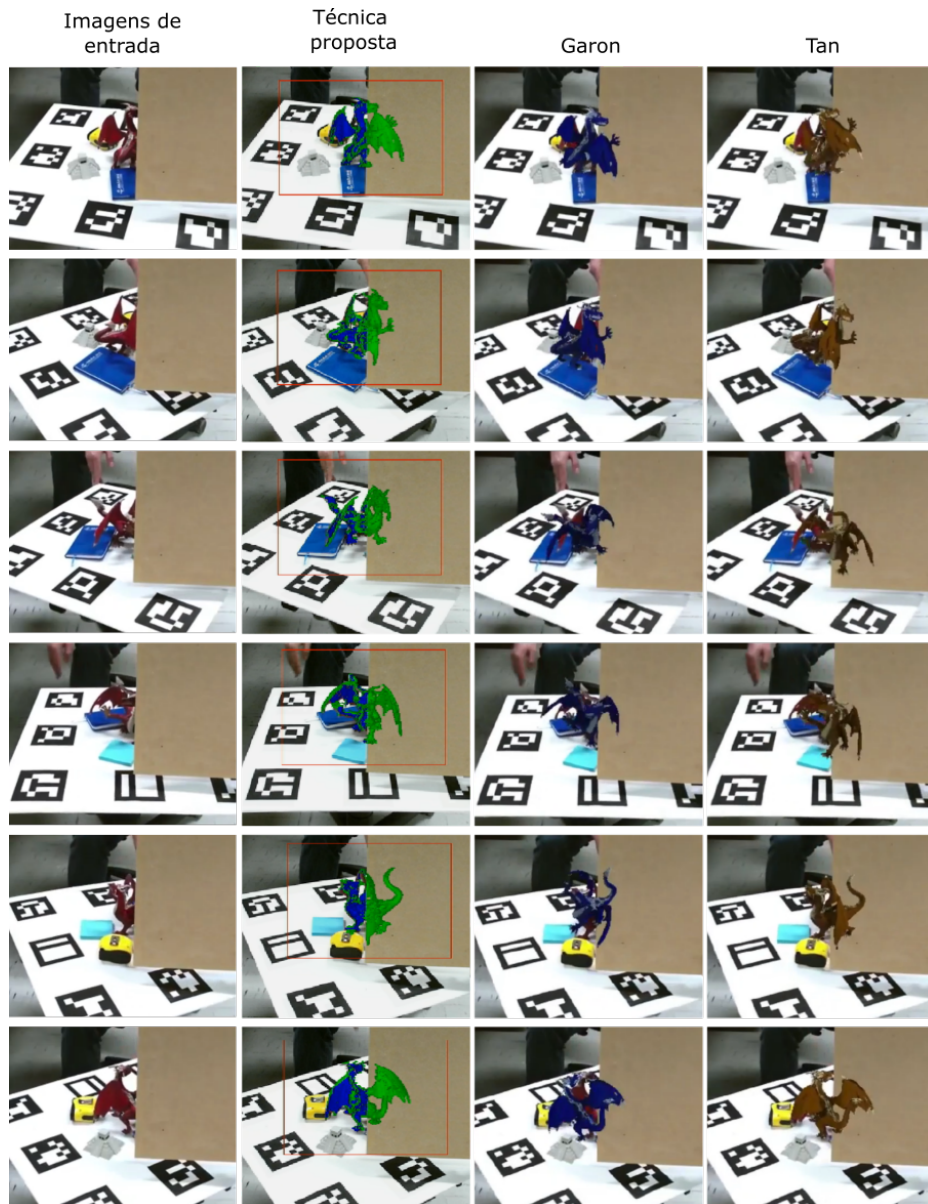


Fonte: Elaborada pelo autor (2021).

de rotação e translação consideravelmente maiores que aquelas obtidas pela técnica proposta sem reinicialização alguma do rastreador. Em todos os testes, a técnica proposta apresentou melhores resultados quando usou a configuração com um enxame de 8 partículas e o modelo com 4.740 pontos, tanto em relação à rotação quanto em relação à translação e em ambas as direções do aparato de oclusão.

A Tabela 3, por sua vez, compara as médias e desvios padrão dos erros obtidos pela técnica proposta em relação aos resultados obtidos pelo rastreador apresentado em Júnior e

Figura 30 – Comparação dos resultados usando os casos de teste do *Dragon* com 40% de oclusão na vertical de Garon e Lalonde (2017). Imagens de entrada (1ª coluna), resultados da técnica proposta (2ª coluna), resultados de Garon e Lalonde (2017) (3ª coluna) e resultados de Tan et al. (2015) (4ª coluna).



Fonte: Elaborada pelo autor (2021).

Lima (2018). É possível notar que em todos os casos de teste a técnica de Júnior e Lima (2018) obteve erros médios ligeiramente inferiores, mas com a maioria dos desvios padrão mais altos, especialmente nos casos de teste com oclusão na vertical e em relação à translação. Um ponto importante a ser citado nessa comparação é que, apesar da acurácia semelhante em ambas as técnicas, a versão proposta neste trabalho se destaca por oferecer um processamento em tempo real, o que não foi possível com o rastreador em Júnior e Lima (2018). Os detalhes do desempenho de tempo da técnica proposta são apresentados a seguir, na próxima subseção.

Tabela 2 – Médias do erros de rotação e translação nos casos de teste do *Dragon* com 40% de oclusão de Garon e Lalonde (2017) obtidas pelo método proposto (sem reinicialização do rastreador) em relação às técnicas propostas em Tan et al. (2015) e em Garon e Lalonde (2017) (com reinicialização a cada 15 quadros), melhores resultados em negrito.

Caso de teste	Vetor	Nº de Partículas	Nº de pontos do modelo	Média de erros		
				Técnica proposta	Tan et al.	Garon e Lalonde
40% horizontal	Rot (graus)	8	4.740	2,9	>45,0	>10,0
		16	2.370	3,0		
	Trans (mm)	8	4.740	6,9	>25,0	>16,0
		16	2.370	7,1		
40% vertical	Rot (graus)	8	4.740	2,3	>24,0	>11,0
		16	2.370	2,5		
	Trans (mm)	8	4.740	7,8	>16,0	>11,0
		16	2.370	8,0		

Fonte: Elaborada pelo autor (2021)

Tabela 3 – Média e desvio padrão dos erros de rotação e translação nos casos de teste do *Dragon* com 40% de oclusão de Garon e Lalonde (2017) obtidas pela técnica proposta e por Júnior e Lima (2018).

Caso de teste	Nº de Partículas	Nº de pontos do modelo	Vetor	Técnica proposta		Santos Jr e Lima	
				Média	Desvio padrão	Média	Desvio padrão
40% horizontal	8	4.740	Rot (graus)	2,9	1,5	2,6	1,1
		2.370	Trans (mm)	6,9	1,0	6,2	1,2
	16	4.740	Rot (graus)	3,0	1,5	2,5	1,1
		2.370	Trans (mm)	7,1	1,1	6,2	1,3
40% vertical	8	4.740	Rot (graus)	2,3	1,2	2,0	1,3
		2.370	Trans (mm)	7,8	1,9	7,4	2,1
	16	4.740	Rot (graus)	2,5	1,2	2,1	1,3
		2.370	Trans (mm)	8,0	1,9	7,6	2,1

Fonte: Elaborada pelo autor (2021)

5.2.2 Avaliação de Desempenho de Tempo

Uma das mudanças propostas neste trabalho para melhoria do desempenho de tempo da técnica de rastreamento baseada em otimização foi fazer com que o PSO usado seja processado completamente em GPU. Experimentos realizados com diferentes quantidades de partículas e diferentes amostragens do modelo evidenciaram que a técnica pode rodar em tempo real e obter resultados compatíveis ou melhores que aqueles presentes no estado da arte, especialmente em casos de teste com oclusões severas.

O desempenho de tempo dessa versão do rastreador usando o caso de teste com oclusão

de 40% na horizontal pode ser observado na Tabela 4. Nela, são apresentados os tempos de processamento e suas respectivas taxas de fps quando usados enxames com 8 e 16 partículas, combinados com quatro diferentes amostragens do modelo, sendo a primeira o modelo completo e as outras três com uma quantidade de pontos 10, 20 e 30 vezes menor. Nos testes usando enxames com 8 partículas, apenas o rastreador com o modelo completo não obteve um fps acima de 20. Já com 16 partículas, apenas aqueles usando modelos 20 e 30 vezes menor obtiveram um rastreamento em tempo real.

Tabela 4 – Tempo médio para o rastreamento em um quadro e fps da técnica proposta em relação ao tamanho do enxame e o número de pontos do modelo.

Nº de partículas do PSO	Tamanho do modelo	Tempo (s)	Taxa de quadros (fps)
8	47.402	0,210	4,76
	4.740	0,035	28,57
	2.370	0,026	38,46
	1.580	0,019	52,63
16	47.402	0,409	2,44
	4.740	0,058	17,24
	2.370	0,037	27,02
	1.580	0,029	34,48

Fonte: Elaborada pelo autor (2021)

5.3 METODOLOGIA E RESULTADOS DA SEGUNDA ETAPA

Na segunda etapa os experimentos foram realizados em um notebook com processador Intel Core i7-855U @ 1,80 GHz (8 CPUs) 2,0GHz, 8 GB de RAM e placa de vídeo NVIDIA GeForce 930MX. O rastreador foi implementado em C e C++, com o suporte de diversas bibliotecas, tais como: OpenCV⁶, OpenGL⁷, PCL⁸, CUDA SDK⁹, GSL¹⁰ e um PSO implementado em C¹¹.

Assim como na primeira etapa dos experimentos, os testes na segunda etapa também usaram um PSO com topologia *lbest* e peso de inércia dinâmico com valores máximos e mínimos $\mu_{max} = 0,7298$ e $\mu_{min} = 0,3$, respectivamente, parâmetros sociais e cognitivos $c_1 = c_2 = 1,496$ e limiar da distância euclidiana entre par de pontos $l = 0,01m$. As amplitudes

⁶ <http://opencv.org>

⁷ <https://www.opengl.org/>

⁸ <http://pointclouds.org>

⁹ <https://developer.nvidia.com/cuda-toolkit>

¹⁰ <http://www.gnu.org/software/gsl>

¹¹ <https://github.com/kkentzo/ps0>

de rotação e translação, porém, desta vez não foram usadas como parâmetros fixos em todos os experimentos, seus valores foram definidos em cada experimento conforme o objetivo do mesmo, porém com limites máximos nas amplitudes de rotação e translação respectivamente iguais a $\Delta_{\omega_{max}} = 0,3$ e $\Delta_{t_{max}} = 50mm$ e limites mínimos nas amplitudes de rotação e translação respectivamente iguais a $\Delta_{\omega_{min}} = 0,1$ e $\Delta_{t_{min}} = 10mm$.

Os experimentos foram realizados em todos os cenários da base de dados de Garon, Laurendeau e Lalonde (2018). Foram usados exames com 32 partículas e os modelos com seus tamanhos originais, exceto pelo modelo do *skull*, que precisou ser reamostrado devido à falta de homogeneidade na densidade de seus pontos. O critério de parada do PSO foi de 100 iterações. Nessa etapa, foram avaliados os impactos no rastreamento ao usar a seleção da ROI dinâmica (Subseção 4.3.1) e dos pontos visíveis do modelo considerando diferentes tipos de oclusão (Subseção 4.3.3), amplitudes dinâmicas de rotação e translação para definição do subespaço de busca (Seção 4.4) e a média harmônica das distâncias na função de aptidão (Subseção 4.5.2) em um estudo de ablação (Subseção 5.3.1). Os resultados ainda são comparados em relação à acurácia, robustez de desempenho de tempo de técnicas de rastreamento presentes no estado da arte.

Para determinar a acurácia da técnica foram usadas as métricas propostas em Garon, Laurendeau e Lalonde (2018) para calcular a diferença entre poses. Sejam duas poses $P_1 = [\mathbf{R}_1 | \mathbf{t}_1]$ e $P_2 = [\mathbf{R}_2 | \mathbf{t}_2]$ quaisquer, a diferença entre elas é dada pelas funções

$$\delta_t(\mathbf{t}_1, \mathbf{t}_2) = \|\mathbf{t}_1 - \mathbf{t}_2\|_2, \quad (5.4)$$

que calcula a distância euclidiana entre os respectivos vetores de translação, e

$$\delta_R(\mathbf{R}_1, \mathbf{R}_2) = \cos^{-1} \left(\frac{\text{Tr}(\mathbf{R}_1^T \mathbf{R}_2) - 1}{2} \right), \quad (5.5)$$

que expressa a diferença entre as matrizes de rotação em graus. Na Equação 5.5 a função $\text{Tr}(\mathbf{R}_i)$ retorna o traço da matriz $\mathbf{R}_i$.

Quando essa diferença é calculada usando a pose encontrada pelo rastreador em relação ao seu respectivo *ground truth* em um determinado quadro, os valores retornados pelas funções δ_t e δ_R correspondem respectivamente aos erros de translação e rotação da predição feita pela técnica naquele quadro. Nos experimentos descritos nesta seção, os termos *erro de rotação* e *erro de translação* se referem sempre às métricas definidas a partir dessas diferenças.

Para possibilitar a comparação dos resultados obtidos com os resultados de outros métodos de rastreamento já publicados na literatura, a metodologia experimental adotada nesta etapa

segue aquela utilizada em Garon, Laurendeau e Lalonde (2018). Nela, são definidas regras para a avaliação das técnicas em cada cenário (*stability*, *occlusion* e *interaction*) da base de dados proposta. No cenário *stability*, a acurácia de uma pose $[\mathbf{R}_q | \mathbf{t}_q]$ encontrada pelo rastreador no quadro q é expressa pelas distâncias de rotação e translação em relação à pose anterior $[\mathbf{R}_{q-1} | \mathbf{t}_{q-1}]$. Esse cálculo é realizado para todos os quadros da sequência usada e a média dos valores obtidos é usada para mostrar o nível de *jitter* da técnica. No cenário *occlusion*, a acurácia da pose encontrada pelo rastreador é definida pelos erros de rotação e translação em relação ao *ground truth* da pose no quadro avaliado. Nesse cenário, o rastreador usa os valores de *ground truth* para reiniciar o rastreador a cada 15 quadros. Segundo Garon, Laurendeau e Lalonde (2018), isso foi usado para contornar o problema das perdas de pose, ou falhas do rastreador, em trechos com quadros difíceis de serem rastreados. No cenário *interaction*, os erros de rotação e translação também são medidos em relação ao *ground truth*, porém a forma como o rastreador é reiniciado durante os experimentos foi definida de acordo com cada caso de teste. Nos casos de teste *interaction-translation*, *interaction-rotation* e *interaction-full*, os rastreadores são reiniciados a cada 15 quadros ou quando o rastreador falha. Nessa metodologia, é considerado que o rastreador falha quando $\delta_t(\mathbf{t}_q^*, \mathbf{t}_q) > 3cm$ ou $\delta_R(\mathbf{R}_q^*, \mathbf{R}_q) > 20^\circ$ por mais de 7 quadros consecutivos, em que, $\mathbf{t}_q^*$ e $\mathbf{R}_q^*$ correspondem ao vetor de translação e à matriz de rotação do *ground truth* da pose no q-ésimo quadro da sequência, e $\mathbf{t}_q$ e $\mathbf{R}_q$ ao vetor de translação e à matriz de rotação da pose encontrada pelo rastreador nesse mesmo quadro. No caso de teste *interaction-hard*, o rastreador é reiniciado apenas quando ocorre uma falha. O número de vezes que uma técnica falha durante uma sequência é uma métrica que pode ser usada para avaliar a robustez do rastreador.

5.3.1 Estudo de Ablação

Nesta segunda etapa, os primeiros experimentos foram realizados usando todos os casos de teste do cenário *occlusion* da base de dados em Garon, Laurendeau e Lalonde (2018) com o objetivo de avaliar os impactos de cada melhoria no rastreador em relação à acurácia dos resultados. Dessa forma, cinco versões do rastreador foram consideradas, como descritas abaixo:

- *Rastreador 1*: configurado com amplitudes de rotação e translação fixas, com valores $\Delta_\omega = 0,3$ e $\Delta_t = 50mm$, média aritmética na função de aptidão e seleção de pontos

visíveis como em Júnior e Lima (2018);

- *Rastreador 2*: configurado como o *rastreador 1*, porém usando média harmônica na função de aptidão como descrito na Subseção 4.5.2;
- *Rastreador 3*: configurado como o *rastreador 2*, porém usando a seleção de pontos visíveis como proposto na Subseção 4.3.3;
- *Rastreador 4*: configurado como o *rastreador 3*, porém com amplitudes de rotação e translação calculadas em tempo de execução (como descrito na Seção 4.4) usando as constantes $\alpha = 7$ e $\beta = 5$;
- *Rastreador 5*: configurado como o *rastreador 4*, porém usando a filtragem de poses, com filtro de tamanho 3.

Os primeiros resultados obtidos podem ser vistos na Tabela 5. Nela, são apresentadas a acurácia de cada uma das versões do rastreador proposto usando os casos de teste com diferentes níveis de oclusão na vertical e na horizontal do cenário *occlusion*. A acurácia é medida por meio das médias dos erros de rotação (em graus) e translação (em milímetros) calculadas pelas Equações 5.4 e 5.5 em relação aos seus respectivos valores de *ground truth*, considerando todos os resultados do rastreamento dos 11 objetos disponibilizados na base de dados, com os melhores valores aparecendo em negrito. Note que os experimentos com 0% de oclusão obviamente não possuem aparato de oclusão algum, portanto, a classificação dos resultados em duas direções diferentes (horizontal e vertical) não se aplica a esses casos de teste. Por isso, no erro médio por nível de oclusão em diferentes direções não foram considerados os resultados com 0% de oclusão.

A partir desses resultados é possível verificar que o *rastreador 4* e o *rastreador 5* obtiveram resultados muito parecidos. No entanto, considerado as médias gerais, ou seja, por nível de oclusão nas duas diferentes direções, é possível notar que o rastreador 4 obteve um melhor desempenho que o *rastreador 5*, e do que todos os outros rastreadores. Fazendo uma comparação entre os *rastreadores 4* e *5* em relação aos demais (*rastreadores 1, 2* e *3*) o *rastreador 4* obteve melhores resultados em todos experimentos, já o *rastreador 5* conseguiu melhores resultados na maioria deles, com apenas uma exceção: no experimento com caso de teste com 45% de oclusão na vertical, em que obteve resultado inferior ao *rastreador 3*.

É possível observar ainda que o *rastreador 3* obteve melhores resultados em relação ao *rastreador 2* e ao *rastreador 1* em quase todos os casos de teste e também na média dos

Tabela 5 – Médias dos erros de translação e rotação com diferentes configurações do rastreador proposto, agrupadas por níveis de oclusão e pela direção do aparato de oclusão (horizontal ou vertical), no cenário *occlusion* da base de dados de Garon, Laurendeau e Lalonde (2018).

Direção	Rastr.	Translação (mm)						Rotação (graus)					
		15%	30%	45%	60%	75%	Média	15%	30%	45%	60%	75%	Média
Horizontal	1	8,22	8,97	14,06	36,59	82,08	29,98	4,54	5,51	12,46	25,36	36,75	16,93
	2	8,34	9,41	11,12	18,54	44,83	18,45	4,06	6,24	8,35	14,07	26,97	11,94
	3	7,81	8,39	9,43	11,75	16,87	10,85	4,03	4,97	6,64	8,89	15,10	7,93
	4	7,77	8,23	9,31	11,05	12,60	9,79	3,97	4,43	5,76	7,03	11,29	6,50
	5	7,76	8,23	9,28	11,05	12,67	9,80	3,96	4,51	5,72	6,90	11,43	6,50
Vertical	1	7,12	8,98	14,23	52,56	98,31	36,24	5,48	8,82	15,21	29,90	41,35	20,15
	2	7,05	8,81	13,06	31,95	82,68	28,71	5,78	7,65	12,28	20,95	36,09	16,55
	3	7,03	8,17	10,24	22,80	61,27	21,90	4,87	7,41	10,99	22,41	35,80	16,30
	4	6,85	7,94	9,66	17,34	24,99	13,35	4,38	6,53	9,72	18,06	24,44	12,63
	5	6,93	7,91	10,14	16,61	23,02	12,92	4,66	6,14	12,17	19,21	23,47	13,13

Fonte: Elaborada pelo autor (2021)

erros por nível de oclusão. O mesmo pode ser dito do *rastreador 2* em relação ao *rastreador 1*, já que este obteve piores resultados na maioria dos casos de teste e na média por nível de oclusão em comparação com todos os demais. Dessa forma, a partir desses resultados experimentais foi possível observar que as contribuições realizadas a cada nova versão do rastreador possibilitaram melhorias na técnica, com uma ênfase menor em relação à filtragem de poses, uma vez que a contribuição do *rastreador 5* em relação ao *rastreador 4* não ficou tão clara nesses experimentos, como ocorreu em relação às demais versões.

Essas observações se confirmam ao verificar a média desses erros na Tabela 6, que agrupa os resultados apenas pelo nível de oclusão e inclui os casos de testes sem oclusão, uma vez que as médias de erros de rotação e translação mostram uma tendência de queda a cada contribuição realizada na técnica. Ainda nessa tabela, verificamos mais uma vez que os *rastreadores 4* e *5* obtiveram os melhores resultados em todos os experimentos, exceto nos casos de teste sem oclusão (coluna com 0%) em relação à média dos erros de translação, em que a versão do rastreador que obteve melhor acurácia foi a do *rastreador 2*.

Tabela 6 – Médias dos erros de translação e rotação com diferentes configurações do rastreador da técnica proposta, agrupadas por diferentes níveis de oclusão, no cenário *occlusion* da base de dados de Garon, Laurendeau e Lalonde (2018).

Rastr.	Translação (mm)							Rotação (graus)						
	0%	15%	30%	45%	60%	75%	Média	0%	15%	30%	45%	60%	75%	Média
1	7,18	7,67	8,97	14,14	44,57	90,19	28,79	5,16	5,01	7,17	13,84	27,63	39,05	16,31
2	7,16	7,70	9,11	12,09	25,24	63,75	20,84	4,20	4,92	6,94	10,31	17,51	31,53	12,57
3	7,26	7,42	8,28	9,83	17,28	39,07	14,86	4,51	4,45	6,19	8,81	15,65	25,45	10,85
4	7,25	7,31	8,08	9,49	14,20	18,79	10,85	4,10	4,17	5,48	7,74	12,55	17,86	8,65
5	7,26	7,34	8,07	9,71	13,83	17,85	10,68	4,11	4,31	5,33	8,95	13,05	17,45	8,87

Fonte: Elaborada pelo autor (2021)

Como esperado, observou-se ainda que, à medida que o nível de oclusão aumenta, os erros médios de rotação e de translação também ficam maiores, porém em termos gerais isso ocorreu de forma mais acentuada nos experimentos com oclusão na vertical. Acredita-se que essa diferença ocorreu em função de falhas nos mapas de profundidade das imagens RGB-D da cena. Essas falhas, que correspondem a regiões do mapa de profundidade da cena que contém conjuntos de pixels com valores nulos, são geradas no momento da captura, e costumam ocorrer devido à posição do projetor e do sensor de IR da câmera RGB-D em relação aos objetos da cena. No caso da base de dados usada, são falhas próximas à borda do aparato usado para causar a oclusão controlada do objeto rastreado, provocando buracos nas nuvens de pontos da cena justamente na região onde deveria se encontrar parte dos pontos desse objeto. Para a técnica avaliada neste trabalho, na prática, esse fenômeno aumenta consideravelmente o nível de oclusão do objeto, uma vez que essas regiões da nuvem não possuem pontos válidos, usados como informação da cena durante a avaliação de uma pose. Naturalmente, esse problema também ocorre na obtenção das imagens das sequências com o aparato de oclusão configurado na horizontal. Porém, nesse caso, devido ao fato de o projetor e o sensor IR do Kinect usado na captura se alinharem horizontalmente, este problema ocorre com uma intensidade muito menor.

Outra observação que pode ser feita a partir desses resultados é que, apesar da semelhança entre as médias dos *rastreadores 4* e *5*, é possível notar que o *rastreador 5* conseguiu bons resultados principalmente em relação ao erro de translação. Isso pode ser justificado pela própria natureza do movimento realizado pelo objeto nos casos de teste do cenário *occlusion*, uma vez que ocorre com ênfase na rotação do objeto e com pouco movimento de translação. Essa observação pode apontar para uma tendência da filtragem da pose favorecer melhores resultados principalmente em cenas com objetos estáticos, como discutimos melhor nos experimentos com o cenário *stability* na Subseção 5.3.2, no entanto não pode ser considerada em definitivo, já que os resultados apresentados no cenário *interaction-hard* (Subseção 5.3.2) mostram que a filtragem de poses pode também possibilitar melhorias em cenários com uma grandes quantidades de movimento no objeto alvo.

5.3.2 Avaliação da Acurácia e Robustez a Falhas

Seguindo a análise dos resultados experimentais nessa etapa, a Tabela 7 apresenta os erros de rotação e translação agrupadas por modelo, nos casos de teste do cenário *occlusion*

com oclusão na horizontal e vertical, obtidos dessa vez usando apenas o *rastreador 4*, que foi a versão que apresentou os melhores resultados. As porcentagens apresentam os níveis de oclusão, sendo 0% aqueles sem oclusão. Uma vez que os modelos apresentam características geométricas e de textura variadas, os resultados apresentados podem apontar para aquelas características que possibilitam melhores resultados de rastreamento em relação às demais. Verificando-se a acurácia na translação com aparato de oclusão na horizontal, por exemplo, os três objetos que proporcionaram o rastreamento com melhor acurácia foram *Lego*, *Cookie jar*, e *Dog*. Já em relação à rotação, os objetos que possibilitaram o rastreamento mais acurado foram *Clock*, *Cookie jar* e *Dragon*. Nos experimentos com o aparato de oclusão na vertical, os modelos *Turtle*, *Dragon* e *Clock* obtiveram as melhores médias em relação aos erros de translação, e aqueles do *Dragon*, do *Clock* e do *Kinect* em relação aos erros de rotação. O modelo *Shoe* possibilitou uma boa média nos experimentos com oclusão na vertical, mas não possui caso de teste para todos os níveis de oclusão nessa direção. Na base de dados também não há casos de teste com 75% de oclusão na horizontal para o modelo *Kinect*.

Alguns desses experimentos apresentam resultados interessantes, como aqueles casos de testes com o *Cookie jar*. Por ter um formato cilíndrico, esse objeto possui pouca informação geométrica para ajudar no rastreamento em relação ao tipo rotação realizado pela mesa usada como base giratória para os objetos no cenário *occlusion*. Mesmo assim possibilitou uma boa acurácia em relação aos demais objetos nesse aspecto. Outro resultado que chama atenção é o rastreamento do *Dog*, que apesar de ter bastante informação geométrica devido ao seu formato e suas dobras na superfície, possui pouca informação de textura, pois é todo branco. Esse tipo de resultado pode ser útil durante o estudo de versões da técnica de rastreamento que não usam informações RGB. Apesar de ter bastante informação de textura ao longo de sua superfície e ser no formato de um paralelepípedo, o que deveria facilitar na detecção do movimento de rotação da mesa, o *Walkman* obteve o pior resultado na média dos erros de translação e um dos piores na média dos erros de rotação.

É importante ressaltar que em alguns casos de teste com oclusão na vertical acima de 45% existem trechos com sequências de quadros em que o objeto alvo fica completamente ocluído. Isso ocorre de forma mais recorrente nos casos de teste com 75% de oclusão, principalmente devido aos buracos formados nas nuvens de pontos na região da cena próxima à borda do aparato de oclusão. Como já foi mencionado anteriormente, essas regiões não possuem pontos com informações de coordenadas 3D e normais válidas. Esse fato aumenta consideravelmente a dificuldade de rastrear o objeto alvo nesse tipo de sequência. Isso pode ser verificado nos

Tabela 7 – Médias dos erros de rotação e translação do *rastreador 4* por modelo nos casos de teste apenas com oclusão na horizontal (mais os casos de teste sem oclusão) do cenário *occlusion* da base de dados de Garon, Laurendeau e Lalonde (2018).

Direção	Modelo	Translação (mm)							Rotação (graus)						
		0%	15%	30%	45%	60%	75%	Média	0%	15%	30%	45%	60%	75%	Média
Hor.	Clock	5,05	9,35	8,74	7,46	6,30	8,49	7,57	2,36	1,76	1,92	2,17	2,78	5,55	2,76
	Cookie jar	10,74	4,08	2,53	3,78	4,14	6,53	5,30	2,84	2,92	3,28	4,43	4,38	5,35	3,87
	Dog	6,12	2,95	2,93	3,06	5,21	14,32	5,77	3,38	2,95	3,13	3,68	4,41	17,94	5,92
	Dragon	3,34	7,77	10,96	10,15	15,73	14,22	10,36	2,57	1,97	1,85	2,59	4,87	11,37	4,20
	Kinect	6,90	10,93	11,89	16,19	18,16	–	12,81	3,98	4,22	4,11	5,11	4,29	–	4,34
	Lego	6,23	3,52	3,76	3,14	3,37	5,72	4,29	5,26	4,53	4,54	4,99	5,75	8,43	5,58
	Shoe	8,08	9,25	10,83	14,51	18,28	13,51	12,41	3,01	3,30	4,21	5,66	6,48	9,13	5,30
	Skull	15,01	8,44	8,43	7,78	6,33	9,58	9,26	9,25	7,65	8,10	8,33	13,83	16,47	10,61
	Turtle	2,59	9,43	11,51	11,23	12,70	15,01	10,41	2,09	3,27	4,93	9,55	16,72	18,62	9,20
	Walkman	5,64	14,82	16,76	19,31	20,82	24,20	16,93	5,43	7,58	8,90	9,86	10,54	11,00	8,89
Ver.	Watering can	10,07	4,93	2,16	5,82	10,56	14,42	7,99	4,93	3,54	3,79	6,97	3,27	9,03	5,26
	Clock	5,05	5,38	5,28	5,13	7,19	22,78	8,47	2,36	2,43	2,40	2,52	3,66	21,32	5,78
	Cookie jar	10,74	3,50	6,54	6,44	22,93	24,05	12,37	2,84	3,49	11,04	10,26	33,67	32,09	15,57
	Dog	6,12	2,94	5,67	5,83	14,59	40,10	12,54	3,38	2,01	2,85	3,12	16,26	26,93	9,09
	Dragon	3,34	5,79	4,29	5,72	8,49	18,80	7,74	2,57	2,41	2,07	2,51	5,49	16,66	5,29
	Kinect	6,90	6,09	7,59	10,17	23,71	37,13	15,27	3,98	3,77	4,44	5,05	13,37	23,54	9,03
	Lego	6,23	6,73	7,81	12,98	22,49	21,64	12,98	5,26	5,34	5,19	7,43	23,93	25,41	12,09
	Shoe	8,08	7,15	8,33	13,22	20,63	–	11,48	3,01	3,69	4,54	6,23	10,64	–	5,62
	Skull	15,01	15,59	17,29	18,71	26,32	26,83	19,96	9,25	14,24	19,91	22,73	33,45	35,87	22,58
	Turtle	2,59	4,36	5,70	9,53	13,94	13,90	8,34	2,09	2,47	3,85	10,63	19,19	25,70	10,66
	Walkman	5,64	9,46	12,15	11,89	18,80	20,74	13,11	5,43	4,89	6,45	6,15	13,07	18,57	9,09
	Watering can	10,07	8,38	6,64	6,68	11,62	23,90	11,22	4,93	3,37	9,11	30,30	25,93	18,27	15,32

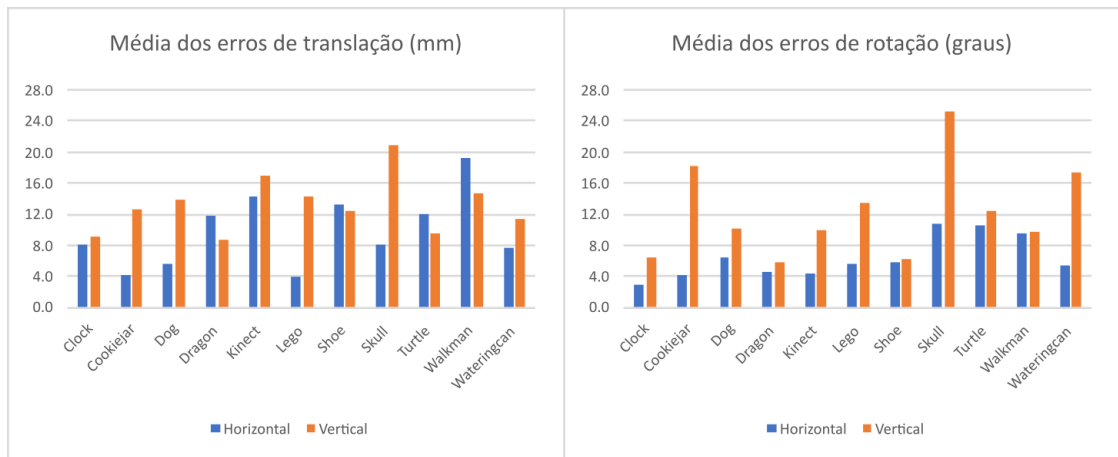
Fonte: Elaborada pelo autor (2021)

resultados obtidos, ou seja, ao comparar os resultados dos experimentos com oclusão na horizontal com aqueles com oclusão na vertical na Tabela 5 e na Tabela 6, é possível perceber uma taxa de crescimento mais acentuado nos erros de rotação e translação, entre os casos de teste com 60% e 75% de oclusão, na tabela que apresenta os resultados com oclusão na vertical.

A Figura 31 apresenta dois gráficos que reforçam essas informações. Neles as médias de erros de rotação e translação considerando todos os níveis de oclusão (exceto os casos de teste com 0% de oclusão) são apresentados em função do modelo do objeto rastreado e da direção do aparato de oclusão (horizontal ou vertical). A partir deles é possível observar que, de forma geral, o rastreador obteve os piores resultados nos casos de teste com oclusão na vertical. Isso fica visível mesmo com o gráfico em questão considerando a média de erro de todos os 5 níveis de oclusão, podendo ser observado principalmente em relação aos erros de rotação.

Uma vez que no estudo de ablação os *rastreadores 4* e *5* apresentaram os melhores resultados em relação às demais, eles foram escolhidos para comparação com as técnicas de rastreamento 6-DOF de objetos arbitrários presentes no estado da arte. A comparação foi realizada em relação a técnicas baseadas em aprendizagem (TAN et al., 2015; GARON; LALONDE, 2017; GARON; LAURENDEAU; LALONDE, 2018; MAROUGKAS et al., 2020) e baseada em otimização (JÚNIOR; LIMA, 2018). Os primeiros resultados dessa comparação, usando os casos de

Figura 31 – Médias dos erros de translação e rotação do *rastreador 4* de todos os níveis de oclusão por caso de teste do cenário *occlusion* da base de dados Garon, Laurendeau e Lalonde (2018), sem os casos de teste com 0% de oclusão.



Fonte: Elaborada pelo autor (2021).

teste do cenário dessa *occlusion*, podem ser vistos na Tabela 8.

Como definido anteriormente, a acurácia foi medida usando as médias dos erros de translação (em mm) e rotação (em graus) obtidos durante o rastreamento dos 11 objetos no cenário *occlusion*. Esses valores foram agrupados de acordo com o nível de oclusão, isto é, de 0% até 75%, sendo 0% aqueles casos em que não há aparato de oclusão na cena. Dessa forma, a partir desses valores, é possível observar que a técnica proposta neste trabalho obteve melhores resultados na maioria dos casos de teste, apresentando maior acurácia em três sequências em relação aos erros de translação (*rastreadores 4* e *5*) e em quatro sequências quando consideradas as médias dos erros de rotação (apenas o *rastreador 4*). Quando calculada a média de erros considerando todos os níveis de oclusão, podemos afirmar que a técnica proposta obteve os melhores resultados tanto em relação à translação quanto em relação à rotação, mesmo considerando os resultados dos *rastreadores 4* e *5* separadamente.

Tabela 8 – Comparação das médias dos erros de rotação e translação de diversas técnicas de rastreamento, avaliadas nos casos de teste do cenário *occlusion* da base de dados de Garon, Laurendeau e Lalonde (2018), considerando todos os 11 objetos alvo e em relação ao nível de oclusão (os melhores resultados aparecem em negrito).

Técnica	Translação (mm)							Rotação (graus)						
	0%	15%	30%	45%	60%	75%	Média	0%	15%	30%	45%	60%	75%	Média
Proposta (rastreador 4)	7,3	7,3	8,1	9,5	14,2	18,8	10,9	4,1	4,2	5,5	7,7	12,5	17,9	8,7
Proposta (rastreador 5)	7,3	7,3	8,1	9,7	13,8	17,9	10,7	4,1	4,3	5,3	9,0	13,1	17,5	8,9
Santos Jr e Lima	7,2	7,7	8,9	11,0	20,6	42,2	16,3	4,6	4,8	6,6	10,1	16,6	27,7	11,7
Garon et al. (object-specific)	7,4	9,8	11,5	12,5	15,5	24,2	13,5	3,6	5,9	7,9	10,0	12,6	22,2	10,4
Garon et al. (multi-object)	23,3	16,9	14,6	14,3	13,2	13,4	16,0	4,0	8,6	12,3	12,1	14,7	15,4	11,2
Garon et al. (generic)	6,7	11,1	18,9	25,9	34,4	47,4	24,1	5,3	8,4	16,1	26,8	43,7	50,6	25,2
Garon e Lalonde	7,4	11,2	18,9	26,8	38,1	55,0	26,2	5,3	8,8	17,7	28,2	41,7	49,8	25,3
Tan et al.	8,2	8,5	15,9	138,0	186,0	213,0	94,9	4,0	7,4	33,1	70,3	89,5	88,0	48,7

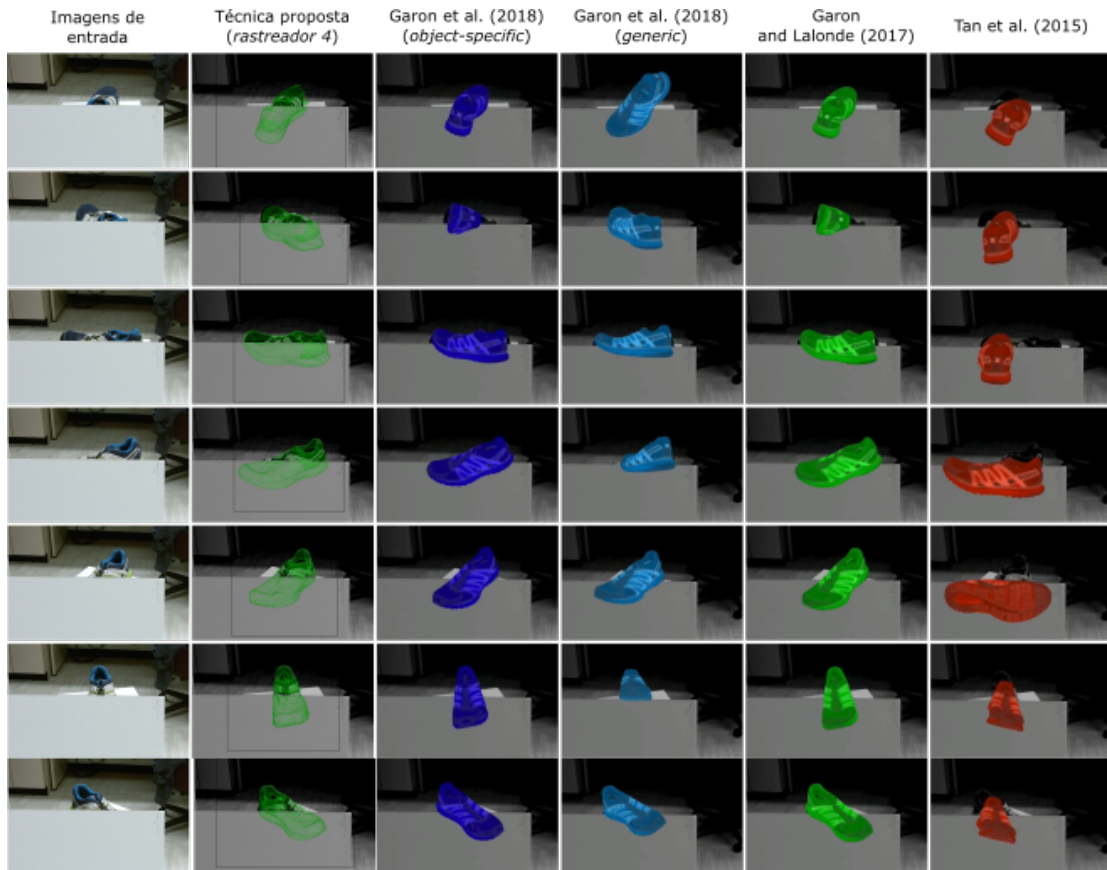
Fonte: Elaborada pelo autor (2021)

Dentre as versões da técnica proposta por Garon, Laurendeau e Lalonde (2018), aquela que obteve os melhores resultados na maioria dos testes (versão chamada de *object-specific*) somente superou o método proposto por este trabalho em relação à rotação da sequência com 0% de oclusão. A versão *multi-object* obteve bons resultados em oclusões maiores, especialmente com 75%, mas apresentou resultados muito inferiores na maioria dos demais casos de teste. Já a versão *generic*, comparada aos demais, somente possibilitou um rastreamento mais acurado na sequência com 0% de oclusão e em relação aos erros de translação. Em relação às técnicas de rastreamento em Garon e Lalonde (2017) e Tan et al. (2015), a técnica proposta se mostrou muito superior. Os casos de teste que mais impactaram nesse resultado foram aqueles com maior oclusão. Uma avaliação qualitativa da técnica proposta em relação aos métodos baseados em aprendizagem pode ser feita através da Figura 32, em que aparecem imagens com resultados de experimentos realizados usando a sequência com 60% de oclusão na horizontal do modelo *Shoe*. Na Figura 33, por sua vez, é possível fazer a comparação entre o *rastreador 4* e o método apresentado em Júnior e Lima (2018). Para a composição de ambas as figuras, no que diz respeito à escolha de cada quadro da imagem, optou-se sempre pelo antecessor do quadro em que o rastreador é reinicializado pela regra descrita no início desta seção. Esse critério foi usado para garantir uma escolha justa, além de ajudar a mostrar os resultados com a maior distância possível dos instantes em que cada rastreador é reinicializado.

No cenário *interaction*, foram realizados experimentos usando todas as sequências do caso de teste *interaction-hard*. Segundo Garon, Laurendeau e Lalonde (2018), esse é o caso de teste que contém as sequências mais difíceis desse cenário. Os resultados do rastreamento são comparados com aqueles obtidos nos experimentos em Maroungkas et al. (2020), em que as sequências do *Dragon* e *Cookie jar* foram usadas para determinar a acurácia e robustez da técnica proposta em relação a Garon, Laurendeau e Lalonde (2018). Assim como em Maroungkas et al. (2020), as métricas usadas na comparação foram o número de falhas do rastreador e as médias dos erros de translação e de rotação para cada sequência. A Tabela 9 apresenta os resultados das técnicas usadas na comparação. Nela é possível observar que a técnica apresentada nesta tese obteve os melhores resultados em relação ao número de falhas e aos erros de translação e rotação em ambos os casos de teste.

Diferente de Maroungkas et al. (2020), os experimentos propostos em Garon, Laurendeau e Lalonde (2018) usando as sequências do caso de teste *interaction-hard* foram realizados usando todos os 11 objetos. Porém, os resultados foram reportados apenas em relação ao total de falhas de cada rastreador em todas as sequências desse caso de teste. Na Tabela 10 é possível

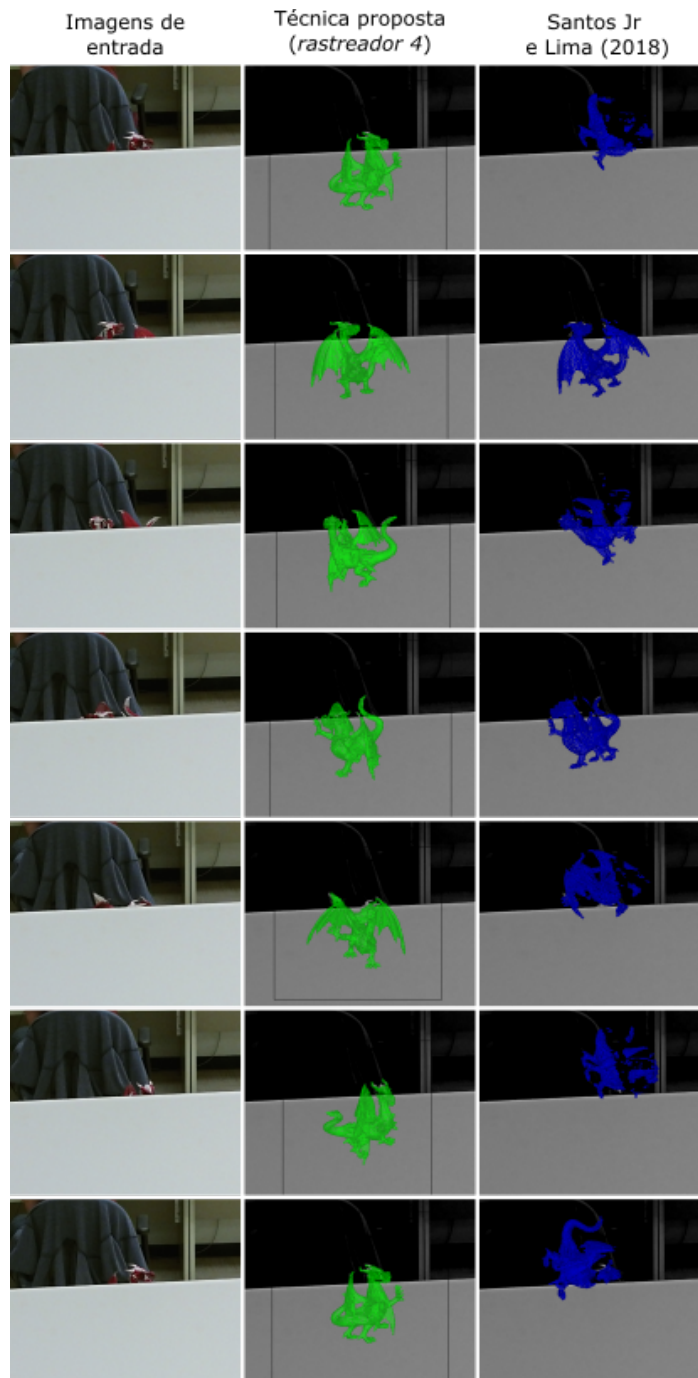
Figura 32 – Comparação dos resultados usando os casos de teste do *Shoe* com 60% de oclusão na horizontal de Garon, Laurendeau e Lalonde (2018). Imagens de entrada (1ª coluna), resultados da técnica proposta usando o *rastreador 4* (2ª coluna), resultados de Garon, Laurendeau e Lalonde (2018) usando *object-specific* (3ª coluna), resultados de Garon, Laurendeau e Lalonde (2018) usando *generic* (4ª coluna), resultados de Garon e Lalonde (2017) (5ª coluna) e resultados de Tan et al. (2015) (6ª coluna).



Fonte: Figura adaptada pelo autor partir de Garon, Laurendeau e Lalonde (2018).

comparar o número de falhas obtidas pelo rastreador proposto em relação às técnicas propostas em Júnior e Lima (2018), Garon e Lalonde (2017), Tan et al. (2015), Garon, Laurendeau e Lalonde (2018). De forma geral, observa-se que a técnica proposta nesta tese obteve o menor número de falhas em relação às demais, principalmente quando observamos os resultados do *rastreador 5*. Considerando essa versão o rastreador, a técnica proposta neste trabalho falhou 78,4% menos que a com melhor resultado entre as demais técnicas (versão *object-specific* de Garon, Laurendeau e Lalonde (2018)). É importante observar também que o rastreador proposto neste trabalho falhou 93,7% menos que a versão *multi-object* do rastreador de Garon, Laurendeau e Lalonde (2018), aquele que obteve bons resultados no rastreamento com 75% de oclusão nos experimentos com o cenário *occlusion*. Uma avaliação qualitativa nos resultados obtidos pelas técnicas citadas pode ser feita através do vídeo resultados etapa 2 do material suplementar no Apêndice A.

Figura 33 – Comparação dos resultados usando os casos de teste do *Dragon* com 75% de oclusão na horizontal de Garon, Laurendeau e Lalonde (2018). Imagens de entrada (1ª coluna), resultados da técnica proposta usando o *rastreador 4* (2ª coluna) e resultados de Júnior e Lima (2018) (3ª coluna)



Fonte: Elaborada pelo autor (2021).

No cenário *stability* foram realizados experimentos com todos os casos de teste. Os resultados obtidos pelos *rastreadores 4* e *5* foram comparados com aqueles apresentados pelas técnicas de rastreamento propostas em Tan et al. (2015), Garon e Lalonde (2017), Garon, Laurendeau e Lalonde (2018) e podem ser vistos na Tabela 11, onde é possível observar que

Tabela 9 – Comparação entre o número de falhas e as médias de erros de translação e rotação de diferentes técnicas de rastreamento (os melhores resultados estão em negrito).

Técnicas	Dragon			Cookie jar		
	Translação	Rotação	Nº de falhas	Translação	Rotação	Nº de falhas
Proposta (rastreador 4)	8,92	6,69	0	4,90	9,55	1
Proposta (rastreador 5)	8,18	5,97	0	5,07	9,47	1
Maraugkas et al.	11,63	8,31	2	14,69	15,00	9
Santos Jr e Lima	13,06	8,36	2	10,74	20,44	14
Garon et al.	34,38	36,38	17	15,78	24,29	15

Fonte: Elaborada pelo autor (2021)

Tabela 10 – Comparação entre o somatório do número de falhas de cada rastreador considerando todas as 11 sequências do caso de teste *interaction-hard* do cenário *interaction* da base de dados de Garon, Laurendeau e Lalonde (2018) (melhor resultado em negrito).

Técnica	Nº de falhas
Proposta (rastreador 4)	21
Proposta (rastreador 5)	8
Santos Jr e Lima	46
Garon et al. (object-specific)	37
Garon et al. (multi-object)	127
Garon et al. (generic)	38
Garon e Lalonde	53
Tan et al.	86

Fonte: Elaborada pelo autor (2021)

duas versões da técnica proposta por Garon, Laurendeau e Lalonde (2018), a *object-specific* e a *multi-object*, obtiveram os melhores resultados em relação às demais. Entre as versões do rastreador proposto nesta tese, os resultados mostraram que o uso da filtragem de poses possibilitou resultados mais acurados nesse tipo de cenário, uma vez que o *rastreador 5* obteve melhores médias em todos os casos de testes em relação ao *rastreador 4*. No geral, a técnica proposta neste trabalho conseguiu melhores resultados apenas em relação às técnicas em Tan et al. (2015), Garon e Lalonde (2017) e a versão *generic* de Garon, Laurendeau e Lalonde (2018), com destaque para a média do erro de rotação no *stability-occluded*, em que o *rastreador 5* obteve o segundo melhor resultado entre todas as técnicas.

5.3.3 Avaliação do Desempenho de Tempo

A análise de desempenho de tempo nessa etapa dos experimentos foi realizada com o objetivo de verificar o impacto do uso de uma ROI nas imagens RGB-D da cena e do processamento

Tabela 11 – Comparação entre o grau de *jitter* de cada rastreador no cenário *stability* de Garon, Laurendeau e Lalonde (2018) (melhores resultados em negrito).

Técnica	Translação (mm/quadro)				Rotação (graus/quadro)			
	Far	Occluded	Near	Média	Far	Occluded	Near	Média
Proposta (rastreador 4)	0,95	0,98	0,98	0,97	1,10	0,94	1,32	1,12
Proposta (rastreador 5)	0,79	0,80	0,70	0,76	1,07	0,77	1,01	0,95
Garon et al. (object-specific)	0,68	0,72	0,56	0,65	0,59	0,76	0,52	0,62
Garon et al. (multi-object)	0,41	0,57	0,38	0,45	0,79	1,09	0,69	0,86
Garon et al. (generic)	0,75	1,19	0,72	0,89	0,98	1,67	0,95	1,20
Garon e Lalonde	1,06	1,24	0,93	1,08	1,23	1,49	1,13	1,28
Tan et al.	1,31	1,53	1,20	1,35	1,44	1,92	1,30	1,55

Fonte: Elaborada pelo autor (2021)

completamente em GPU. Uma comparação foi realizada em relação ao rastreador proposto em Júnior e Lima (2018), que não usa uma ROI e emprega o paralelismo em GPU apenas na função de aptidão no nível dos pontos do modelo. Diferentemente de Júnior e Lima (2018), na versão proposta o tempo de processamento varia de acordo com o nível de oclusão de tipo 3 do objeto rastreado, uma vez que há uma filtragem dos pontos que sofrem tal oclusão a cada quadro. Assim, a avaliação do desempenho de tempo foi realizada usando o caso de teste *interaction-hard*, uma vez que possui sequências com trechos com e sem oclusões do tipo 3.

Os dois modelos escolhidos para essa avaliação foram aqueles com maior e menor número de pontos na base de dados usada, ou seja, *Dragon* e *Lego*, que possuem respectivamente 86.065 e 8.499 pontos. A Tabela 12 mostra que o rastreador com PSO implementado completamente em GPU e que usa ROI possibilitou diminuir consideravelmente o tempo de processamento, sendo mais de 17 vezes mais rápido nos experimentos com o modelo *Dragon* e mais de 5 vezes mais rápido nos experimentos com o modelo *Lego*.

Tabela 12 – Comparação do tempo de processamento entre técnicas de rastreamento 3D baseadas em PSO (melhores resultados em negrito) usando os modelos *Dragon* e *Lego* e o caso de teste *interaction-hard* do cenário *interaction* da base de dados de Garon, Laurendeau e Lalonde (2018).

Modelo	Nº de pontos	Tempo de processamento (s)	
		Técnica proposta	Santos Jr e Lima
Dragon	86.065	0,589	10,503
Lego	8.499	0,265	1,337

Fonte: Elaborada pelo autor (2021)

5.4 METODOLOGIA E RESULTADOS DA TERCEIRA ETAPA

A terceira etapa dos experimentos foi realizada com o intuito de analisar as informações internas do PSO, produzidas durante a execução do algoritmo, tanto em relação à eficiência da busca em si quanto em relação ao método usado para avaliar as partículas através da função de aptidão, e com isso entender melhor critérios relacionados à avaliação, diversidade e convergência das partículas do enxame, possibilitando a correção de possíveis falhas.

A terceira etapa dos experimentos foi realizada no mesmo ambiente da segunda etapa, isto é, em um notebook com processador Intel Core i7-855U @ 1,80 GHz (8 CPUs) 2,0GHz, 8 GB de RAM e placa de vídeo NVIDIA GeForce 930MX. O rastreador usado também foi o mesmo, implementado em C e C++, com o suporte das bibliotecas já citadas, usando um PSO com topologia *lbest*, parâmetros sociais e cognitivos $c_1 = c_2 = 1,496$ e limiar da distância euclidiana entre par de pontos $l = 0,01m$, com amplitudes de rotação e translação com limites máximos respectivamente iguais a $\Delta\omega_{max} = 0,3$ e $\Delta t_{max} = 50mm$ e limites mínimos respectivamente iguais a $\Delta\omega_{min} = 0,1$ e $\Delta t_{min} = 10mm$, porém, dessa vez, usando dois tipos de pesos de inércia, o dinâmico (variável no tempo) e o adaptativo, e com os valores dos parâmetro tamanho de enxame, critério de parada, critério de visibilidade dos pontos do modelo podendo variar de acordo com o objetivo de cada experimento.

Para realizar os testes de convergência e diversidade populacional do enxame foram usadas três métricas: média de aptidão das partículas, a aptidão da melhor partícula e a variância média da população, ao longo das iterações do PSO. A variância média da população representa a diversidade populacional em um determinado instante e pode ser calculada usando

$$D_p = \frac{1}{mn} \sum_{j=1}^n \sum_{i=1}^m (x_{ij} - \bar{x}_j)^2, \quad (5.6)$$

em que D_p representa a variância, x_{ij} a i -ésima partícula do enxame com j dimensões e $\bar{x}_j$ é o centro do enxame, em que

$$\bar{x}_j = \frac{1}{m} \sum_{i=1}^m x_{ij}. \quad (5.7)$$

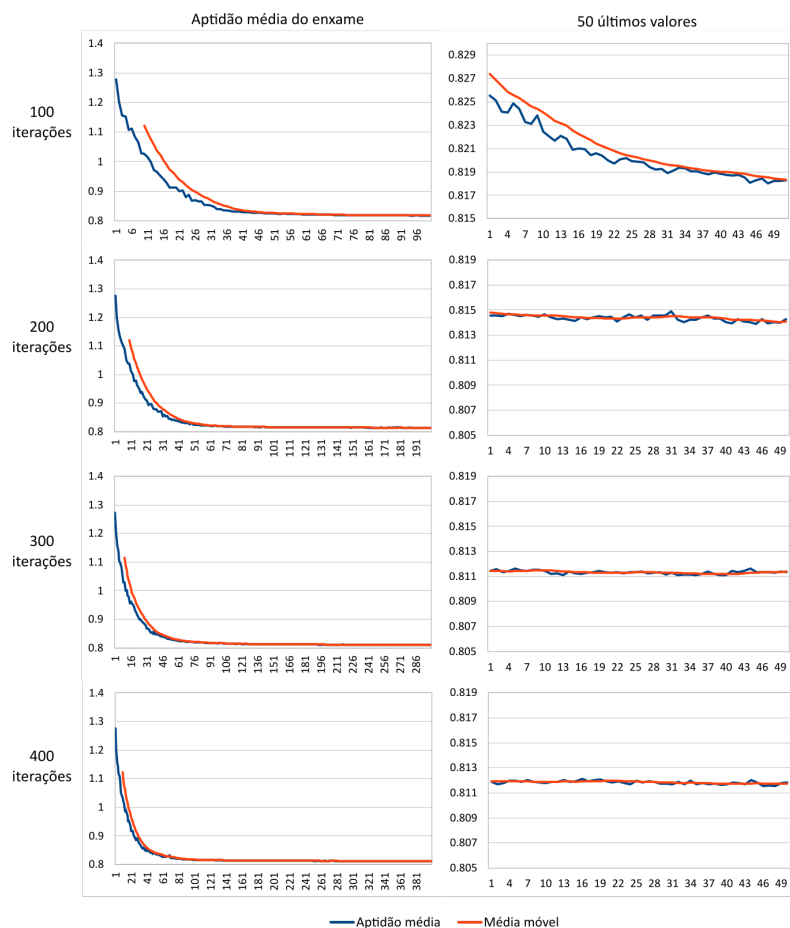
5.4.1 Diversidade Populacional e Teste de Convergência do Enxame

Os experimentos de diversidade e convergência do enxame foram realizados usando o segundo quadro do caso de teste *Dragon* com 0% de oclusão do cenário *occlusion*, com um enxame de tamanho fixo de 128 partículas. Com o objetivo de monitorar a evolução e a perda

de diversidade do enxame ao longo do tempo, foram utilizados quatro diferentes valores para o critério de parada do PSO, a saber, 100, 200, 300 e 400 iterações.

Os gráficos da Figura 34 apresentam os resultados desses testes em relação à aptidão média do enxame a cada iteração. Aqueles à esquerda representam a evolução dos valores atribuídos a essa métrica ao longo de todas as iterações em cada caso de teste. Já aqueles à direita mostram esses valores sempre em relação às 50 últimas iterações de cada experimento, apresentados em escala menor para facilitar a visualização. Como apresentado na Seção 4.5, o valor da aptidão de cada partícula é calculado usando a diferença entre a transformação do modelo, representada pela partícula em questão, e a pose do objeto alvo na cena, assim quanto menor o valor dessa métrica melhor a partícula é avaliada.

Figura 34 – Aptidão média do enxame em experimentos com valores de 100, 200, 300 e 400 iterações para o critério de parada.



Fonte: Elaborada pelo autor (2021).

Ainda na Figura 34, ao observar os gráficos da esquerda é possível verificar que a média das aptidões das partículas cai acentuadamente por volta das 50-60 primeiras iterações, e as variações a partir desse ponto são bem pequenas, para uma escala de décimos apresentada. Assim,

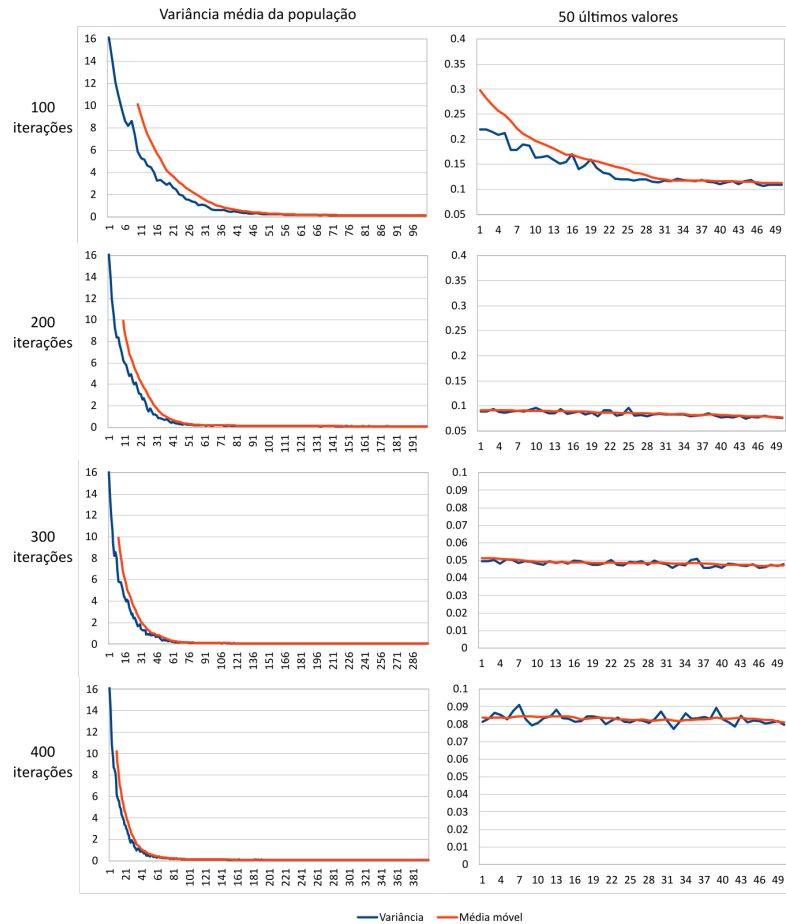
observando-se a média móvel dos gráficos da direita, é possível ver que a aptidão média do enxame parece cair até as últimas iterações, exceto no experimento com 300 iterações, o que significa que nesses casos é possível que o enxame ainda esteja evoluindo em relação à exploração local do espaço de busca. É importante ressaltar que as curvas observadas nesses gráficos, apesar de serem obtidas a partir de experimentos com critérios de parada incrementados de 100 em 100 iterações, não são continuidades umas das outras, uma vez que o decaimento do peso de inércia, que influencia a velocidade das partículas e, portanto, a evolução das aptidões, é calculado em função do número máximo de iterações definido a priori em cada experimento.

Em relação à variância média da população nos quatro experimentos realizados, podemos ver na Figura 35 que os gráficos da esquerda mostram uma diminuição da diversidade da população mais rápida até, aproximadamente, a iteração de número 40-50 em todos os casos de teste, o que evidencia uma busca mais global nesse intervalo, independentemente do número de iterações usadas. A partir de então, é possível verificar que o enxame passa a realizar uma busca local, na qual, de acordo principalmente com a média móvel dos gráficos da direita, as partículas parecem não ter parado de convergir em todos os experimentos, atingindo valores bem próximos de zero. Em relação à essa métrica, é importante observar que o enxame convergiu mais rapidamente no experimento com 300 iterações, em relação a todos os demais.

Já na Figura 36, é possível observar como a aptidão da melhor partícula do enxame evolui a cada iteração. Mais uma vez, nos gráficos da esquerda temos uma visão geral dessa evolução com valores absolutos e média móvel, e do lado direito é possível observar como se apresentaram os dados nas 50 últimas iterações. Dessa forma, nota-se que existe uma variação muito grande na aptidão das melhores partículas no início das iterações, o que pode indicar que o algoritmo consegue escapar de mínimos locais. Por outro lado é possível ver que nas últimas iterações essa alternância na aptidão das partículas ocorre com menor frequência, se estabilizando nas últimas 10 iterações na maioria dos testes, exceto no caso de teste com 400 iterações. Apesar de as mudanças de aptidão parecerem ocorrer com menor frequência nas iterações finais, e com uma tendência de diminuir a aptidão, ainda é possível observar que o enxame nem sempre evolui no sentido positivo, uma vez que em vários instantes ele termina a iteração com a melhor partícula possuindo uma aptidão maior que aquela obtida pela melhor partícula da iteração anterior.

De forma geral, ao final dos experimentos apresentados, observou-se, através das três métricas usadas, duas características principais na evolução do PSO. A primeira é que o

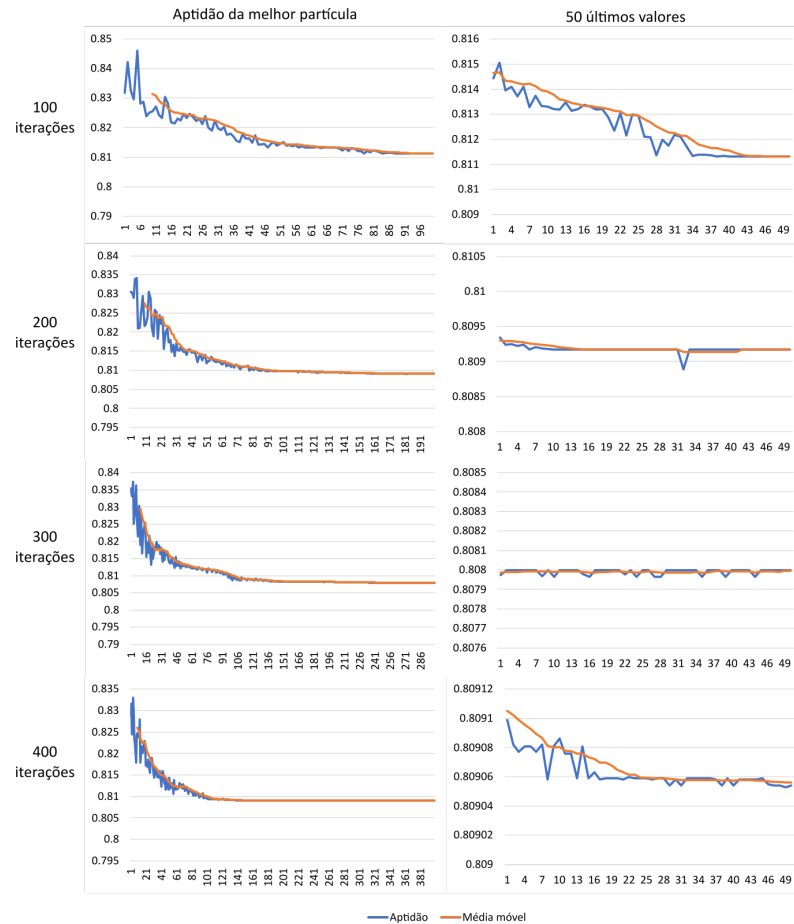
Figura 35 – Variância média da população em experimentos com valores de 100, 200, 300 e 400 iterações para o critério de parada.



Fonte: Elaborada pelo autor (2021).

enxame apresenta uma tendência de exploração mais global em um intervalo de 40-60 primeiras iterações, podendo variar um pouco de acordo com o caso de teste ou com a métrica utilizada. Essa convergência nas iterações iniciais pode em parte ser explicada pela própria natureza do PSO, que tende a convergir rapidamente. Mas também pode ser resultado das estratégias adotadas pela técnica proposta nesta tese, para que o enxame não explore regiões muito distantes da pose inicial encontrada no quadro anterior, tais como o cálculo de amplitudes dinâmicas levando em consideração a inércia do movimento do objeto alvo (ver Seção 4.4), bem como o ato de copiar a melhor partícula do enxame anterior para o enxame atual a cada novo quadro. A segunda característica é que, apesar de uma rápida convergência inicial, ocorreu uma constante evolução das partículas do enxame na maioria dos casos de teste que, aparentemente, não dependeu do número de iterações usadas. Essa característica foi observada principalmente em relação às duas primeiras métricas usadas, ou seja, em relação à aptidão média do enxame e à variância média da população.

Figura 36 – Aptidão da melhor partícula em experimentos com valores de 100, 200, 300 e 400 iterações para o critério de parada.



Fonte: Elaborada pelo autor (2021).

De forma adicional ao que foi observado até aqui, a Tabela 13 apresenta o erro médio de rotação e translação em cada um dos experimentos analisado, nela é possível observar que a convergência apresentada nos gráficos nem sempre se traduz em ganho de acurácia na mesma proporção, uma vez que os erros médios parecem ter estabilizado no caso de teste com critério de parada de 200 iterações. Ou seja, apesar da constante evolução do exame nos três critérios analisados, não foi possível afirmar que, ao usarmos o modelo de função de aptidão criado e avaliado nesta tese, o uso de iterações adicionais sempre irá melhorar os erros de translação e rotação do rastreamento. O que mostra que pode existir uma diferença fundamental entre o que a função de aptidão proposta representa para o exame e o que ela pode de fato medir quando recebe as informações RGB-D da cena.

Tabela 13 – Erro médio de rotação e translação nos experimentos com 100, 200, 300 e 400 iterações.

	Nº de iterações (critério de parada)			
	100	200	300	400
Rotação (graus/quadro)	1,13	0,84	0,82	0,84
Translação (mm/quadro)	1,973	1,34	1,32	1,31

Fonte: Elaborada pelo autor (2021)

5.4.2 Estudo da Função de Aptidão

Os experimentos nesta subseção foram realizados com o objetivo de avaliar a função de aptidão usada pelo PSO proposto nesta tese. Para isso, foram comparadas quantitativamente e qualitativamente as poses encontradas pelo PSO e aquelas definidas pelo *ground truth* da base de dados em diferentes cenários. Nesses experimentos foram usados o modelo *Dragon* no cenário *occlusion* com os casos de teste com 0%, 45% e 60% de oclusão (com o aparato na vertical), o modelo *Watering can* no cenário *occlusion* com o caso de teste de 0% de oclusão e o modelo *Kinect* no caso de teste *stability-far-2* do cenário *stability*, todos pertencentes à base de dados proposta em Garon, Laurendeau e Lalonde (2018). Além destes, foram usados também os modelos *Tide* e *Kinect box* dos casos de testes sintéticos da base de dados em Choi e Christensen (2013). As métricas adotadas para avaliação quantitativa foram: a aptidão da partícula usando média harmônica e média aritmética, o número de pontos projetados, a média das distâncias das coordenadas 3D (Equação 2.6), a média das distâncias das cores (Equação 2.11) e a média das distâncias das normais (Equação 2.12). Em alguns experimentos, com o intuito de observar as distâncias das coordenadas 3D sem adição de valores, nós não usamos a penalidade dos pontos considerados oclusos segundo o limiar l da Equação 2.6, enquanto em outros a penalidade foi usada normalmente com o limiar de $l = 0,01m$.

Os resultados dos experimentos com o *Dragon* usando um PSO de 32 partículas e critério de parada em 100 iterações podem ser vistos na Tabela 14. Nela podemos comparar os valores de cada uma das métricas usadas quando avaliamos as poses encontradas pelo PSO em relação às aquelas apresentadas pelo *ground truth* da base de dados (na tabela abreviada como GT) em cada um dos 5 quadros analisados. No caso de teste com o *Dragon* sem oclusão, as poses encontradas pelo PSO obtiveram os menores valores em quase todas as métricas usadas, exceto na média das distâncias das normais e das cores em alguns quadros. Em relação aos casos de teste *Dragon* com 45% e 60% de oclusão, o PSO obteve os melhores resultados em

todas as métricas e em todos os quadros analisados. Em relação ao valor da aptidão, tanto em média aritmética quanto em média harmônica, as poses encontradas pelo PSO obtiveram as menores médias em todos os testes realizados.

Tabela 14 – Comparação entre as poses encontradas pelo PSO com 32 partículas e 100 iterações em relação àquelas apresentadas pelo *ground truth* da base de dados nos casos de teste *Dragon 0%*, *Dragon 45%* e *Dragon 60%* de oclusão na vertical (melhores resultados em negrito).

		Quadro 1		Quadro 2		Quadro 3		Quadro 4		Quadro 5	
		GT	PSO	GT	PSO	GT	PSO	GT	PSO	GT	PSO
Dragon 0%	Aptidão (méd. harm.)	0,0815	0,0789	0,0793	0,0770	0,0815	0,0776	0,0815	0,0787	0,0792	0,0758
	Aptidão (méd. aritm.)	0,1292	0,1123	0,1253	0,1167	0,1315	0,1193	0,1290	0,1171	0,1328	0,1134
	Média dist. 3D	0,0387	0,0219	0,0358	0,0267	0,0401	0,0299	0,0383	0,0263	0,0416	0,0248
	Média dist. Cores	0,0459	0,0455	0,0450	0,0460	0,0463	0,0449	0,0455	0,0453	0,0468	0,0448
	Média dist. Norm	0,0446	0,0448	0,0445	0,0439	0,0452	0,0445	0,0453	0,0455	0,0444	0,0438
	Nº pontos projet.	24840	24618	24224	22868	24642	23569	25106	24220	24240	22461
Dragon 45%	Aptidão (méd. harm.)	0,0263	0,0257	0,0272	0,0253	0,0269	0,0259	0,0271	0,0258	0,0280	0,0265
	Aptidão (méd. aritm.)	0,0366	0,0349	0,0372	0,0331	0,0368	0,0330	0,0373	0,0333	0,0378	0,0349
	Média dist. 3D	0,0096	0,0089	0,0103	0,0075	0,0094	0,0067	0,0098	0,0070	0,0097	0,0078
	Média dist. Cores	0,0140	0,0131	0,0141	0,0129	0,0140	0,0131	0,0142	0,0133	0,0146	0,0135
	Média dist. Norm	0,0131	0,0129	0,0129	0,0127	0,0134	0,0132	0,0133	0,0130	0,0135	0,0137
	Nº pontos projet.	9578	10618	8808	9802	9796	10703	9446	10294	9267	10361
Dragon 60%	Aptidão (méd. harm.)	0,0289	0,0240	0,0307	0,0260	0,0275	0,0263	0,0293	0,0259	0,0295	0,0248
	Aptidão (méd. aritm.)	0,0436	0,0351	0,0453	0,0364	0,0418	0,0360	0,0431	0,0358	0,0439	0,0360
	Média dist. 3D	0,0146	0,0093	0,0153	0,0093	0,0132	0,0097	0,0131	0,0088	0,0141	0,0093
	Média dist. Cores	0,0144	0,0120	0,0152	0,0125	0,0136	0,0121	0,0138	0,0123	0,0145	0,0121
	Média dist. Norm	0,0146	0,0139	0,0147	0,0146	0,0150	0,0142	0,0162	0,0147	0,0152	0,0145
	Nº pontos projet.	4297	4736	3366	4325	4943	4576	4387	3761	4315	3828

Fonte: Elaborada pelo autor (2021)

Foram realizados também experimentos usando o PSO com 128 partículas e critério de parada em 250 iterações. Os resultados desses experimentos podem ser vistos na Tabela 15. Quando comparados aos resultados dos testes anteriores, o PSO com mais partículas e maior número de iterações obteve as menores médias na maioria dos testes. Ao comparar com as poses do *ground truth*, em relação à aptidão com média harmônica as poses do PSO obtiveram os melhores resultados em todos os quadros. Já em relação à aptidão com média aritmética, as poses do PSO obtiveram os melhores resultados na maioria deles, mostrando um piora em relação aos resultados apresentados na Tabela 14.

O modelo do *Watering can* representa um objeto da base de dados de Garon, Laurendeau e Lalonde (2018) particularmente difícil de ser rastreado, devido à falta de textura e ao alto grau simetria no seu corpo. Devido a essas características, os experimentos realizados anteriormente foram repetidos usando esse objeto, isto é, no caso de teste do *Watering can* com 0% de oclusão do cenário *occlusion*. Como pode ser visto na Tabela 16, nos testes sem limiar para distância euclidiana das coordenadas 3D, dessa vez as poses correspondentes ao *ground truth* obtiveram menores médias das distâncias entre as cores em todos os experimentos, e na maioria deles quando observada as médias das distâncias das coordenadas 3D e aptidão por média

Tabela 15 – Comparação entre as poses encontradas pelo PSO com 128 partículas e 250 iterações em relação àquelas apresentadas pelo *ground truth* da base de dados, no caso de teste *Dragon* 0% de oclusão (melhores resultados em negrito).

		Quadro 1		Quadro 2		Quadro 3		Quadro 4		Quadro 5	
		GT	PSO	GT	PSO	GT	PSO	GT	PSO	GT	PSO
Dragon 0%	Aptidão (méd. harm.)	0,0789	0,0786	0,0770	0,0769	0,0776	0,0774	0,0787	0,0780	0,0758	0,0754
	Aptidão (méd. aritm.)	0,1123	0,1114	0,1167	0,1182	0,1193	0,1165	0,1171	0,1119	0,1134	0,1135
	Média dist. 3D	0,0219	0,0222	0,0267	0,0283	0,0299	0,0275	0,0263	0,0234	0,0248	0,0248
	Média dist. Cores	0,0455	0,0446	0,0460	0,0455	0,0449	0,0447	0,0453	0,0433	0,0448	0,0448
	Média dist. Norm	0,0448	0,0446	0,0439	0,0444	0,0445	0,0443	0,0455	0,0452	0,0438	0,0439
	Nº pontos projet.	24618	24451	22868	22547	23569	22669	24220	23471	22461	21809

Fonte: Elaborada pelo autor (2021)

aritmética. As poses encontradas pelo PSO, por sua vez, obtiveram menores médias em todos os experimentos em relação a duas métricas, aptidão por média harmônica e distâncias entre as normais.

Tabela 16 – Comparação entre as poses encontradas pelo PSO com 32 partículas e 100 iterações em relação àquelas apresentadas pelo *ground truth* da base de dados, no caso de teste *Watering can* 0% de oclusão (melhores resultados em negrito).

		Quadro 1		Quadro 2		Quadro 3		Quadro 4		Quadro 5	
		GT	PSO	GT	PSO	GT	PSO	GT	PSO	GT	PSO
Watering can 0% (sem limiar)	Aptidão (méd. harm.)	0,0675	0,0560	0,0650	0,0556	0,0675	0,0555	0,0694	0,0555	0,0743	0,0554
	Aptidão (méd. aritm.)	0,1052	0,1113	0,0955	0,1007	0,0993	0,1058	0,1020	0,1037	0,1110	0,1073
	Média dist. 3D	0,0320	0,0371	0,0237	0,0289	0,0265	0,0319	0,0286	0,0300	0,0362	0,0336
	Média dist. Cores	0,0366	0,0384	0,0361	0,0369	0,0364	0,0379	0,0363	0,0373	0,0365	0,0378
	Média dist. Norm	0,0366	0,0358	0,0357	0,0350	0,0364	0,0360	0,0371	0,0363	0,0383	0,0359
	Nº pontos projet.	8528	7784	8409	7552	8553	7832	8529	7869	8597	8233
Watering can 0% (l = 0,01 m)	Aptidão (méd. harm.)	0,1150	0,0627	0,1031	0,0625	0,1190	0,0613	0,1438	0,0633	0,2309	0,0643
	Aptidão (méd. aritm.)	0,6071	0,3284	0,5661	0,3070	0,6121	0,3196	0,6820	0,3354	0,8209	0,3646
	Média dist. 3D	0,5338	0,2539	0,4943	0,2348	0,5393	0,2465	0,6086	0,2629	0,7461	0,2906
	Média dist. Cores	0,0366	0,0387	0,0361	0,0369	0,0364	0,0374	0,0363	0,0366	0,0365	0,0383
	Média dist. Norm	0,0366	0,0358	0,0357	0,0352	0,0364	0,0356	0,0371	0,0359	0,0383	0,0357
	Nº pontos projet.	8528	7599	8409	7513	8553	7711	8529	7889	8597	8177
	Pontos oclusos	4531	1906	4133	1741	4590	1878	5173	2051	6403	2354
	Pontos oclusos (%)	53	25	49	23	53	24	60	25	74	28

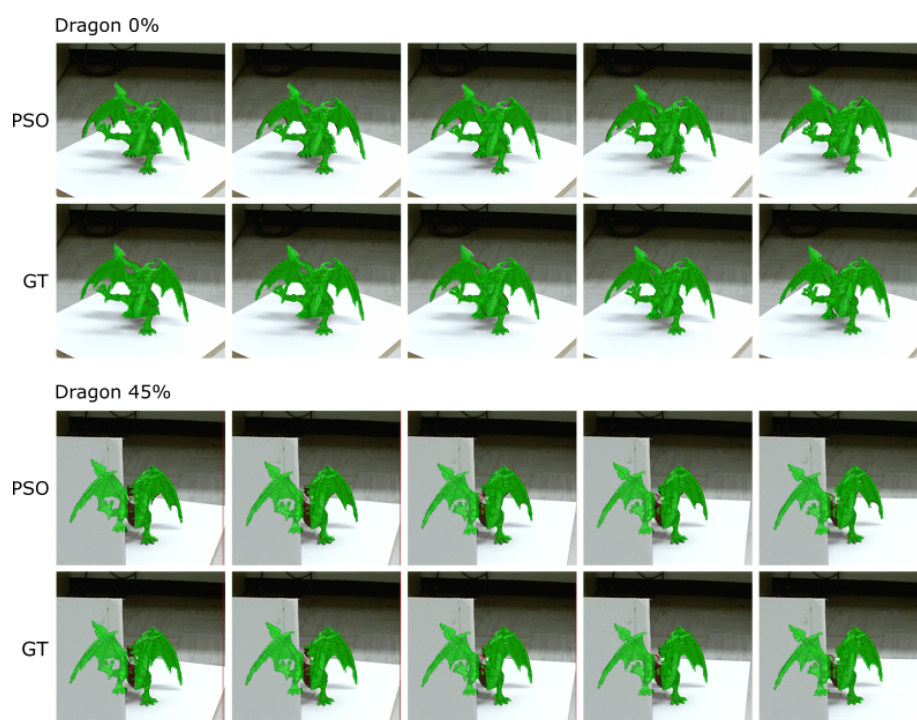
Fonte: Elaborada pelo autor (2021)

Ainda na Tabela 16, é possível observar os resultados dos experimentos com o objeto *Watering can* usando penalidade nas distâncias euclidianas das coordenadas 3D, com limiar como definido na Equação 4.10, isto é, $l = 0.01m$. O PSO passa a obter as menores médias na maioria das métricas em todos os experimentos, ficando com valores maiores em todos os testes apenas em relação à distância entre as cores. Nesses experimentos observa-se ainda que a quantidade de pontos penalizados, de acordo com o limiar l , é menor quando se usa as poses obtidas pelo PSO em todos os quadros analisados.

De forma complementar às análises quantitativa anteriores, é possível realizar uma avaliação qualitativa das poses obtidas pelo PSO através da projeção de cada uma delas sobre a imagem RGB da cena. Na Figura 37, por exemplo, é possível observar respectivamente as

projeções dos pontos do modelo do *Dragon* no cenário *occlusion* com 0% e 45% de oclusão, usando as poses obtidas pelo PSO e comparar com as projeções de seus respectivos *ground truth*. No *Dragon* com 0% de oclusão, apesar das pequenas variações de poses, observadas

Figura 37 – Avaliação qualitativa das poses encontradas pelo PSO com 32 partículas e 100 iterações em relação às aquelas apresentadas pelo *ground truth* da base de dados, nos casos de teste *Dragon* 0% e *Dragon* 45% de oclusão.



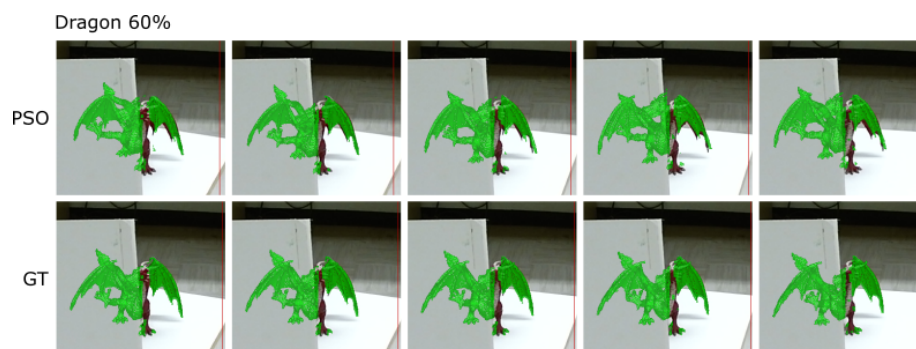
Fonte: Elaborada pelo autor (2021).

principalmente na calda e nas asas do dragão, é difícil de determinar quais são as poses mais acuradas apenas através das projeções. Resultado semelhante ocorre com as projeções com o *Dragon* com 45% de oclusão. É possível observar que nas projeções correspondentes às poses do PSO uma parte muito pequena do contorno da parte de baixo da asa esquerda do dragão parece não estar coberta por pontos verdes, enquanto que nas projeções correspondentes à pose do *ground truth* a parte pintada de verde parece ultrapassar a região da imagem que corresponderia ao contorno da asa. Assim, apesar das imprecisões observadas nas imagens em ambos os casos, é difícil afirmar, apenas observando as projeções, qual deles é mais acurado.

Já na Figura 38, onde podemos comparar as projeções correspondentes ao *Dragon* com 60% de oclusão, é possível ver que as poses encontradas pelo PSO são menos acuradas que aquelas obtidas a partir do *ground truth* da base de dados. Os detalhes das projeções tanto da cabeça, quanto da pata e da asa esquerda exemplificam bem as falhas nas poses do PSO em relação às aquelas do *ground truth*, o que é curioso, uma vez que, segundo os dados da

Tabela 14, as poses do PSO obtiveram as menores médias para todas as métricas avaliadas, independente de o número de pontos projetados ser superior ou não.

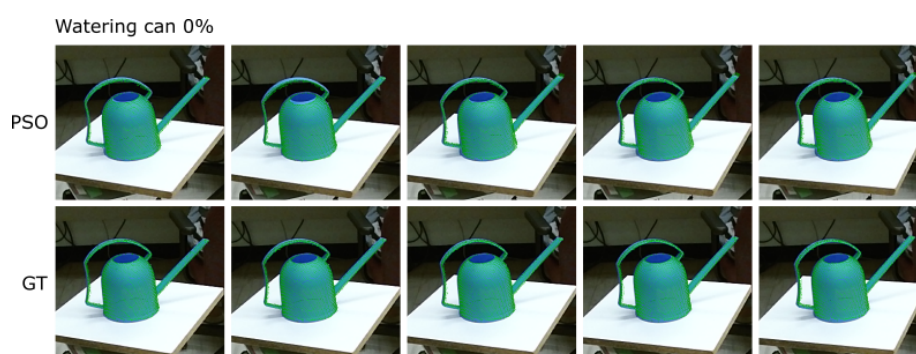
Figura 38 – Avaliação qualitativa das poses encontradas pelo PSO com 32 partículas e 100 iterações em relação às aquelas apresentadas pelo *ground truth* da base de dados, no caso de teste *Dragon* 60% de oclusão.



Fonte: Elaborada pelo autor (2021).

Resultado semelhante ocorre quando são analisadas as projeções do *Watering can* do cenário *occlusion* com 0% de oclusão (Figura 39), em que, apesar de algumas imprecisões das projeções correspondentes ao *ground truth*, principalmente ao longo da alça do regador, é possível observar que as poses obtidas pelo PSO (sem limiar) são menos acuradas. Isso pode ser observado principalmente nas projeções do bico, na parte superior da alça e no canto inferior direito da base do regador.

Figura 39 – Avaliação qualitativa das poses encontradas pelo PSO com 32 partículas e 100 iterações em relação às aquelas apresentadas pelo *ground truth* da base de dados, no caso de teste *Watering can* 0% de oclusão.



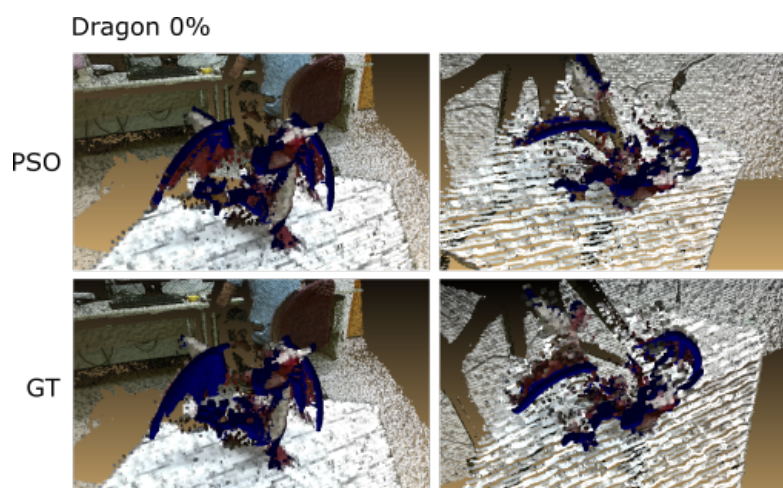
Fonte: Elaborada pelo autor (2021).

Esses resultados aparentemente estão de acordo com o que foi visto na Tabela 16, principalmente em relação às médias das distâncias entre as coordenadas 3D, as médias das distâncias entre as cores e a aptidão calculada por média aritmética. Mas parecem entrar em conflito com os resultados obtidos em relação à aptidão calculada por média harmônica e a média das

distâncias entre as normais para os testes com limiar, que apresenta os menores valores para as poses encontradas pelo PSO.

Uma avaliação qualitativa sobre a pose encontrada pelo PSO em relação à pose do *ground truth* também foi realizada por meio da transformação do modelo do *Dragon* e do *Watering can* sobre a nuvem de pontos da cena (Figuras 40 e 41). Nessas imagens, as regiões em azul escuro correspondem aos pontos do modelo transformado, e as demais regiões aos pontos da cena, com pontos renderizados com suas cores originais. Cada um dos modelos, *Dragon* e *Watering can*, são mostrados sob duas perspectivas, ou seja, observados de frente (imagens à esquerda) e de cima para baixo (imagens à direita), e segundo a pose do PSO (imagens de cima) e do *ground truth* (imagens de baixo).

Figura 40 – Transformação dos pontos do modelo *Dragon* sobre a nuvem de pontos da cena no caso de teste *Dragon* 0% de oclusão, segundo a pose encontrada pelo PSO (acima) e segundo a pose definida no *ground truth* da base de dados (abaixo).



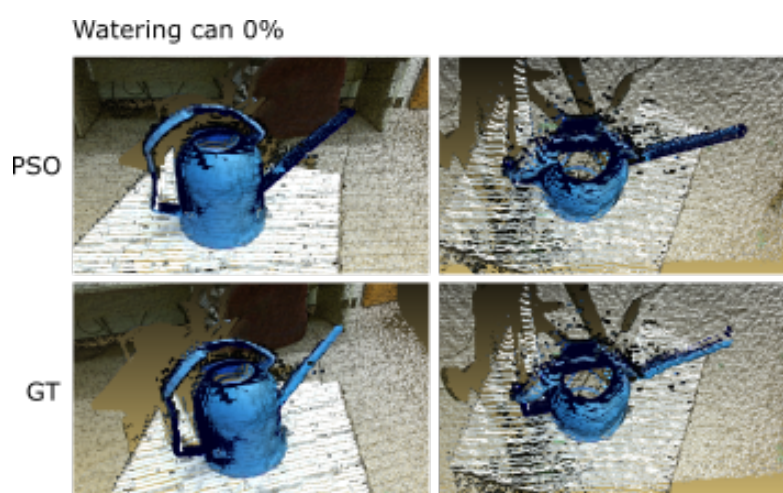
Fonte: Elaborada pelo autor (2021).

Observando as imagens correspondentes ao *Dragon*, é possível notar que muitos dos pontos transformados pela pose do PSO ficam escondidos por trás dos pontos da cena, enquanto que os pontos da transformados pela pose do *ground truth* ficam um pouco à frente dos pontos da nuvem da cena. Na visão superior da nuvem, apesar de parecer que os pontos da asa esquerda transformados pelo modelo estão mais próximos dos pontos da asa esquerda do objeto alvo na cena, é muito difícil afirmar categoricamente qual das transformações ficou melhor devido ao ruído na cena e à dificuldade de avaliar visualmente os possíveis ajustes de rotação e translação que poderiam ocorrer com o modelo, ajustando melhor uma parte da transformação em detrimento de outras.

Na Figura 41, parece ocorrer o contrário, os pontos transformados pela pose do PSO pare-

cem ter ficado um pouco mais à frente dos pontos do objeto alvo do que aqueles transformados pelo *ground truth*, especialmente quando olhamos a região do bico do regador. No corpo do regador, em ambos os resultados, os pontos do modelo parecem ficar em sua maioria por trás dos pontos da cena. Assim como no experimento com o *Dragon*, os resultados com o *Watering can* são inconclusivos devido à dificuldade de determinar qual das poses possibilita as melhores transformações apenas a partir dessas imagens.

Figura 41 – Transformação dos pontos do modelo *Watering can* sobre a nuvem de pontos da cena no caso de teste *Watering can* 0% de oclusão, segundo a pose encontrada pelo PSO (acima) e segundo a pose definida no *ground truth* da base de dados (abaixo).



Fonte: Elaborada pelo autor (2021).

Apesar dos resultados inconclusivos em alguns experimentos qualitativos, houveram testes que mostraram que o PSO encontrou poses visivelmente menos acuradas que as poses do *ground truth* (*Dragon* com 60% de oclusão e *Watering can* com 0%), a despeito dos resultados favoráveis ao PSO nos experimentos quantitativos correspondentes (Tabelas 14 e 16). Uma das hipóteses levantadas durante esses testes para tentar explicar esses resultados é a de que o PSO estaria conseguindo convergir bem em relação às componentes da pose que trata da translação do modelo, por isso os bons resultados em relação às médias das distâncias 3D e o menor número de pontos penalizados nos experimentos com penalidade (Tabela 16), mas não estaria conseguindo alcançar bons resultados em relação às componentes de rotação da pose, o que explicaria parcialmente as poses mal ajustadas em algumas das imagens referentes aos resultados qualitativos.

Nesse caso, se essa hipótese é verdadeira, as componentes correspondentes à rotação nas poses do *ground truth* seriam mais acuradas que as componentes correspondentes à rotação nas poses encontradas pelo PSO. Para verificar isso, foram realizados testes de comparação

das poses encontradas pelo PSO sem modificações (componentes de rotação e translação obtidas pelo PSO) com as poses do PSO modificadas, isto é, contendo as componentes de translação encontradas pelo PSO, mas as componentes de rotação copiadas das respectivas poses do *ground truth*. A Tabela 17 apresenta os resultados desses testes.

Tabela 17 – Comparação entre as poses encontradas pelo PSO com 32 partículas e 100 e àquelas definidas pelo *ground truth* em relação às poses do PSO modificadas (PSO/GT), no caso no caso de teste *Watering can 0%* de oclusão (melhores resultados em negrito).

		Quadro 1		Quadro 2		Quadro 3		Quadro 4		Quadro 5	
		PSO	PSO/GT	PSO	PSO/GT	PSO	PSO/GT	PSO	PSO/GT	PSO	PSO/GT
Watering can 0% (sem limiar)	Aptidão (méd. harm.)	0,0560	0,0567	0,0556	0,0548	0,0555	0,0554	0,0555	0,0549	0,0554	0,0561
	Aptidão (méd. aritm.)	0,1113	0,0976	0,1007	0,0848	0,1058	0,0962	0,1037	0,0895	0,1073	0,0953
	Média dist. 3D	0,0371	0,0255	0,0289	0,0158	0,0319	0,0251	0,0300	0,0173	0,0336	0,0216
	Média dist. Cores	0,0384	0,0358	0,0369	0,0344	0,0379	0,0352	0,0373	0,0356	0,0378	0,0363
	Média dist. Norm	0,0358	0,0364	0,0350	0,0346	0,0360	0,0359	0,0363	0,0367	0,0359	0,0373
	Nº pontos projet.	7784	7309	7552	7010	7832	7397	7869	7234	8233	7550
Watering can 0% (sem limiar)		GT	PSO/GT	GT	PSO/GT	GT	PSO/GT	GT	PSO/GT	GT	PSO/GT
	Aptidão (méd. harm.)	0,0675	0,0567	0,0650	0,0548	0,0675	0,0554	0,0694	0,0549	0,0743	0,0561
	Aptidão (méd. aritm.)	0,1052	0,0976	0,0955	0,0848	0,0993	0,0962	0,1020	0,0895	0,1110	0,0953
	Média dist. 3D	0,0320	0,0255	0,0237	0,0158	0,0265	0,0251	0,0286	0,0173	0,0362	0,0216
	Média dist. Cores	0,0366	0,0358	0,0361	0,0344	0,0364	0,0352	0,0363	0,0356	0,0365	0,0363
	Média dist. Norm	0,0366	0,0364	0,0357	0,0346	0,0364	0,0359	0,0371	0,0367	0,0383	0,0373
	Nº pontos projet.	8528	7309	8409	7010	8553	7397	8529	7234	8597	7550

Fonte: Elaborada pelo autor (2021)

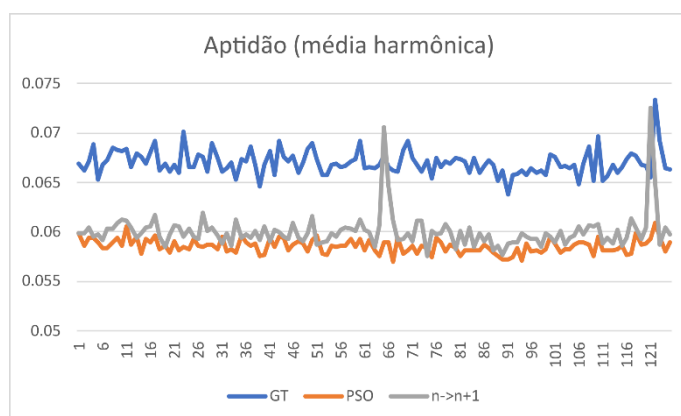
Na primeira parte da Tabela 17, são apresentados os resultados da comparação entre as poses do PSO sem modificações com as poses do PSO modificadas (contendo as componentes da rotação do *ground truth*, por isso PSO/GT para nomear as colunas), assim, é possível observar que as poses modificadas obtêm as melhores médias na maioria dos testes, e em relação a quase todas as métricas, com exceção de alguns testes em relação à média das distâncias das normais, e dois testes em relação à aptidão calculada por média harmônica. Na segunda parte da Tabela 17, são apresentadas as comparações entre as poses do PSO modificado com as poses do *ground truth*. Nesse caso, as poses do PSO modificadas obtiveram as melhores médias em todos os casos de teste e em relação a todas as métricas utilizadas, isto é, quando foram usadas as translações obtidas pelo PSO e as rotações do *ground truth* para compor as poses, estas passaram a ser melhor avaliadas que aquelas com a rotação e translação apenas do *ground truth* em todos os testes. O que parece apontar que realmente o PSO tende a convergir bem em relação às componentes de translação das poses e não obtém resultados tão bons em relação às componentes de rotação.

Os experimentos a respeito da função de aptidão também foram realizados no cenário *stability* da base de dados de Garon, Laurendeau e Lalonde (2018). Nesse cenário, uma vez que os objetos e a câmera ficam estáticos durante toda a filmagem da cena, espera-se que a pose encontrada pelo rastreador em um quadro n qualquer seja tão próxima quanto possível

das poses encontrada no demais quadros e, portanto, que a avaliação realizada pela função de aptidão em ambos os quadros seja semelhante também. Para avaliar isso, foram observadas as aptidões das poses encontradas pelo PSO, bem como aquelas considerando a pose do *ground truth* medidas ao longo de todos os quadros da sequência *Kinect stability-far-2*. Para verificar como o a função de aptidão do rastreador se comporta de um quadro para o outro, foi executado também uma versão do teste em que a pose encontrada no quadro n é avaliada no quadro seguinte $n + 1$.

Os resultados desses experimentos podem ser vistos no gráfico da Figura 42. Nele, assim como nos testes anteriores, observa-se que os valores da aptidão das poses encontradas pelo PSO foram os menores (gráfico laranja) em relação ao *ground truth*. Os resultados referentes à pose do *ground truth* foram os maiores em quase todos os quadros (gráfico azul) em relação aos demais testes. Os valores de aptidão apresentados no gráfico cinza, mostram que quase sempre a aptidão aumenta quando a pose encontrada em um quadro n é avaliada no quadro $n + 1$, com picos nos quadros de número 65 e 121. Observou-se ainda que, mesmo o *ground truth* desse caso de teste sendo composto por uma única pose, isto é, uma pose igual para todos os quadros, o valor da aptidão medida para essa pose ao longo dos quadros varia consideravelmente, o que pode ser causado devido ao ruído inerente ao dispositivo de captura responsável por gerar as nuvens de pontos dos quadros da cena.

Figura 42 – Comparação dos valores das aptidões das poses encontradas pelo PSO em relação aos valores das aptidões das poses definidas pelo *ground truth* no cenário *Stability* no caso de teste *Kinect stability-far-2*.

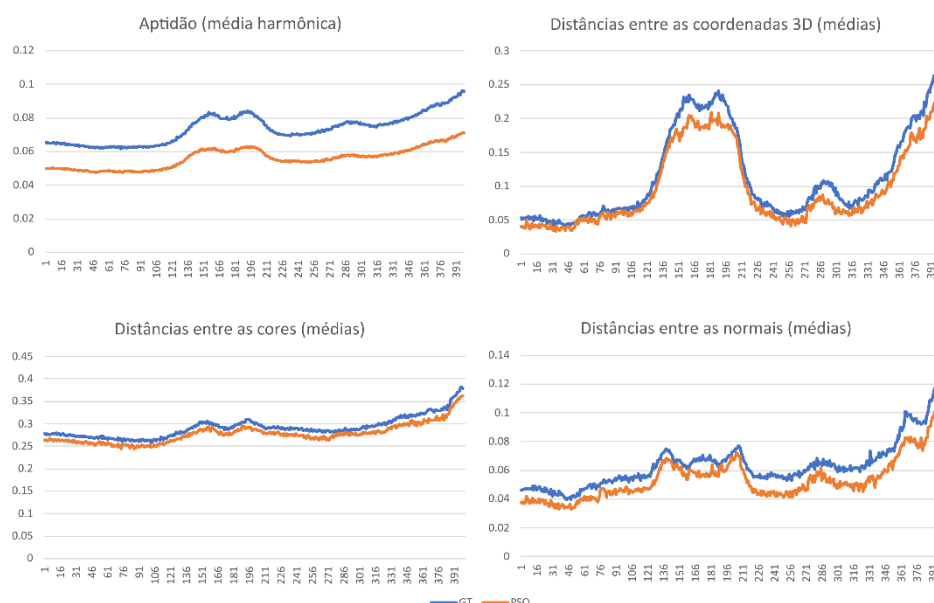


Fonte: Elaborada pelo autor (2021).

Para evitar erros de avaliação devido a possíveis ruídos das nuvens de pontos e imprecisões nos valores de *ground truth*, próprios de caso de testes com nuvens construídas a partir de imagens de ambientes reais, a avaliação da função de aptidão foi realizada também usando os

casos de teste *Tide* e *Kinect box* da base de dados sintética proposta em Choi e Christensen (2013). Nesses testes, inicialmente foram comparadas as aptidões e as médias das distâncias entre as coordenadas 3D, cores e normais, das poses encontradas pelo PSO em relação àquelas do *ground truth* da base de dados ao longo dos 400 primeiros quadros da sequência do *Tide*. Os resultados dessas comparações podem ser vistos na Figura 43. Nela, assim como nos experimentos com cenários reais, é possível observar que os gráficos referentes às poses do *ground truth* ficaram sempre com valores maiores em relação aos gráficos referentes às poses encontradas pelo PSO, ao longo de toda a sequência de 400 quadros e considerando as quatro métricas avaliadas.

Figura 43 – Comparação dos valores das aptidões, distâncias entre as coordenadas 3D, cores e normais das poses encontradas pelo PSO em relação aos respectivos valores das poses definidas pelo *ground truth* nos 400 primeiros quadros do casos de teste *Tide*.

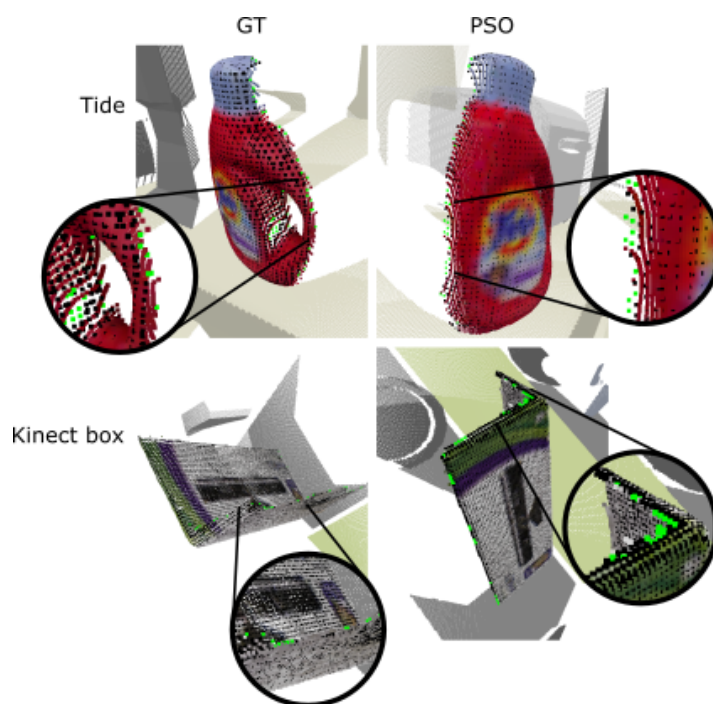


Fonte: Elaborada pelo autor (2021).

Com o intuito de compreender melhor o que acontece na função de avaliação para que as poses do *ground truth* sejam sempre subavaliadas em relação às poses encontradas pelo PSO, foi realizada uma análise qualitativa das transformações do modelo sobre a nuvem de pontos da cena no quadro de número 1 da sequência do *Tide* e do *Kinect box*. A Figura 44 apresenta esses resultados, onde os pontos pretos são pontos do modelo transformados segundo a pose avaliada, os pontos coloridos de acordo com as cores da cena são os pontos da nuvem de pontos da cena e os pontos em verde claro são os pontos do modelo transformado que sofreram penalidade segundo o limiar l da distância euclidiana das coordenadas 3D (ver Equação 4.10). Aqui há duas características a serem observadas. A primeira é que pelo resultado do *Tide* é

possível perceber que mais uma vez a projeção correspondente à pose encontrada pelo PSO faz com que os pontos do modelo fiquem ligeiramente mais à frente do objeto alvo, se comparado ao resultado proporcionado pela pose do *ground truth*. A segunda é que os pontos penalizados (em verde claro em destaque na Figura 44) se encontram em sua maioria nas bordas do objeto alvo ou em regiões prestes a serem perdidas de vista pela câmera.

Figura 44 – Pontos dos modelos *Tide* e *Kinect box* transformados sobre os pontos das nuvens de pontos de suas respectivas cenas, segundo as poses do PSO e do *ground truth*. Nos destaques, e em verde, aparecem os pontos penalizados.



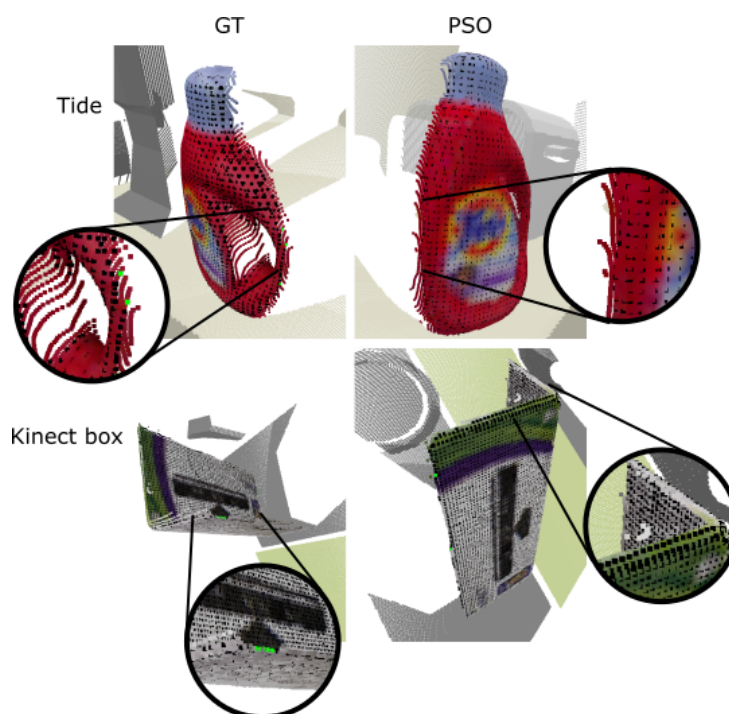
Fonte: Elaborada pelo autor (2021).

Como o valor de penalidade para cada ponto fora do limiar l é alta na função de aptidão, a hipótese levantada aqui é que esse conjunto de pontos penalizados que aparecem em verde estariam influenciando consideravelmente a avaliação das poses, tanto em relação ao *ground truth* quanto em relação às poses encontradas pelo PSO. Por exemplo, a quantidade de pontos penalizados que ocorrem na pose do *ground truth* impediria que o PSO encontre uma pose próxima a pose do *ground truth*, encontrando uma mais afastada, no entanto com menos pontos penalizados, pois tanto nesse experimento quanto naquele usando o *Watering can* (Tabela 16) o número de pontos penalizados em relação ao total de pontos projetados é sempre menor quando usadas as poses do PSO, se comparado com a respectiva pose do *ground truth*.

Com o objetivo de testar essa hipótese, a avaliação qualitativa em que os pontos dos

modelos são transformados sobre a nuvem de pontos da cena, como apresentada na Figura 44, foi repetida, mas dessa vez eliminando os pontos do modelo que pertencessem à borda ou que estivessem no limiar de se tornarem oclusos do ponto de vista da câmera que filma a cena. Para eliminar esses pontos do modelo, foi usado o teste de visibilidade que mede o ângulo θ entre o vetor de direção de vista de cada ponto e sua normal com um limiar $\theta > 110^\circ$. Os resultados dessa avaliação podem ser vistos na Figura 45, onde é possível observar que a quantidade de pontos penalizados diminuiu consideravelmente em relação ao teste anterior.

Figura 45 – Pontos dos modelos *Tide* e *Kinect box* transformados sobre os pontos das nuvens de pontos de suas respectivas cenas, segundo as poses do PSO e do *ground truth*, e usando o limiar $\theta > 110^\circ$ para seleção de pontos na borda. Nos destaques, e em verde, aparecem os pontos penalizados.

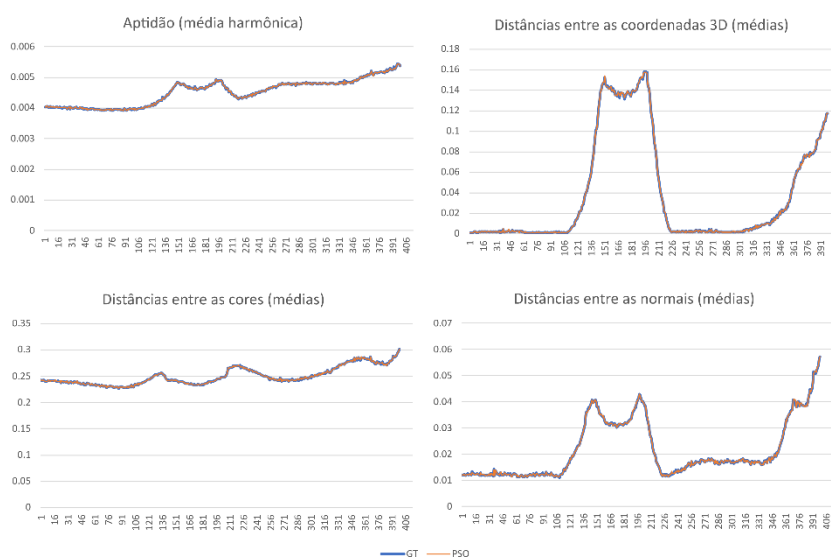


Fonte: Elaborada pelo autor (2021).

Com isso, ao avaliar graficamente mais uma vez as aptidões, as distâncias entre as coordenadas 3D, cores e normais das poses encontradas pelo PSO em relação às poses definidas pelo *ground truth*, dessa vez usando os cortes de pontos da borda do modelo, os resultados são muito diferentes. Primeiro, houve uma queda no valor da aptidão tanto na avaliação das poses encontradas pelo PSO quanto naquelas apresentadas pelo *ground truth*. Isso pode ser visto claramente na diferença de uma ordem de grandeza entre o gráfico da aptidão da Figura 43 e o seu correspondente na Figura 46, naquele a pose do PSO, por exemplo, inicia com um valor entre 0,04 e 0,06 de aptidão enquanto que neste inicia aproximadamente em 0,004. Segundo, diferente dos resultados sem o corte de pontos da borda do modelo, dessa vez os

gráficos referentes às poses encontradas pelo PSO apresentaram valores muito próximos daqueles referentes às poses do *ground truth*, em relação a todas as métricas usadas e ao longo de toda a trajetória utilizada no teste.

Figura 46 – Comparação dos valores das aptidões, distâncias entre as coordenadas 3D, cores e normais das poses encontradas pelo PSO em relação aos respectivos valores das poses definidas pelo *ground truth* nos 400 primeiros quadros do casos de teste *Tide*, usando o limiar $\theta > 110^\circ$ para seleção de pontos na borda.



Fonte: Elaborada pelo autor (2021).

5.4.3 Alterações Propostas e Avaliação de Acurácia

A partir dos resultados experimentais apresentados nas Subseções 5.4.1 e 5.4.2 foi possível observar características a respeito da diversidade e convergência do enxame ao longo do processo de otimização, bem como aspectos de como a função de aptidão avalia as hipóteses de poses nesse processo. Duas considerações importantes podem ser levantadas a respeito desses resultados. Primeiro, como foi observado nos gráficos das Figuras 34 e 35, existe uma rápida perda de diversidade no início do processo de busca, pois tanto a aptidão média quanto a variância média do enxame de partículas cai de forma acentuada nas primeiras iterações, independentemente do número de iterações usadas no experimento. Segundo, as penalidades dos pontos pertencentes à borda do modelo do objeto alvo parecem interferir negativamente na avaliação de cada hipótese de pose realizada pela função de aptidão.

Assim, esta subseção apresenta alguns resultados de experimentos realizados usando o PSO com alterações baseadas nessas considerações. Em todos os experimentos foi usado um PSO

com topologia *lbest*, parâmetros sociais e cognitivos $c_1 = c_2 = 1,496$ e limiar de distância euclidiana entre par de pontos $l = 0,01m$, amplitudes de rotação e translação calculados dinamicamente com limites máximos iguais a $\Delta_{\omega_{max}} = 0,3$ e $\Delta_{t_{max}} = 50mm$ e limites mínimos usando $\Delta_{\omega_{min}} = 0,1$ e $\Delta_{t_{min}} = 10mm$, com um enxame de 32 partículas e critério de parada de 100 iterações. Na tentativa de contornar o problema da perda de diversidade foi testado um PSO com peso de inércia adaptativo, isto é, que considera o estado do enxame a cada interação e não apenas o número de interações, como ocorre no uso do peso de inércia temporal. Quanto ao problema da penalidade dos pontos pertencentes à borda do modelo, assim como nos experimentos da Subseção 5.4.2, foram selecionados quais pontos do modelo transformado podem ser usados no cálculo do erro de reprojeção na função de aptidão a partir do ângulo θ entre o vetor de direção de vista e a normal do ponto avaliado, considerando um limiar $\theta = 110^\circ$, ou seja, para ângulos menores que 110° o ponto não é usado. Os experimentos foram realizados em todos os casos de testes sintéticos da base de dados proposta em Choi e Christensen (2013).

Para possibilitar a comparação com outras técnicas presentes no estado da arte, a métrica utilizada foi a raiz do valor quadrático médio (RMS) dos erros. O RMS dos erros é calculado para cada uma das componentes das poses encontradas ao longo do rastreamento em relação às respectivas componentes das poses do *ground truth*, como apresentado na equação:

$$RMS(k) = \sqrt{\frac{\sum_{i=1}^n (\mathbf{P}_i(k) - \mathbf{G}_i(k))^2}{n}}, \quad (5.8)$$

em que $\mathbf{P}_i$ e $\mathbf{G}_i$ representam respectivamente a i -ésima pose encontrada pelo rastreador e a i -ésima pose do *ground truth*, k corresponde a componente de cada vetor de pose e n o número total de quadros usados ao longo do rastreamento.

Nesses testes foram usadas quatro configurações do PSO. Para uma melhor compreensão dos resultados analisados nessa subseção, segue abaixo o nome e a descrição de cada uma delas:

- *PSO-1*: usa os parâmetros descrito nesta subseção, com peso de inércia temporal sem filtragem de poses;
- *PSO-2*: configurado como o *PSO-1*, mas usando filtragem de poses;
- *PSO-3*: configurado como o *PSO-2*, mas usando cortes dos pontos das bordas do modelo transformado ($\theta = 110^\circ$);

- *PSO-4*: configurado como o *PSO-3*, mas usando peso de inércia adaptativo.

Na Tabela 18 é possível verificar os resultados dos experimentos para cada uma das versões do PSO. Com exceção do *Kinect box*, é possível afirmar que o *PSO-4* obteve os melhores resultados em todos os demais casos de teste, especialmente se for observado as médias dos erros de rotação e de translação, em que o *PSO-4* só não é melhor em relação à médias dos erros de rotação no caso de teste do *Kinect box*. O *PSO-1* e o *PSO-2* conseguiram melhores resultados apenas em algumas componentes de forma isolada, e o *PSO-3* foi melhor que os demais no experimento com o *Kinect box*, especialmente em relação à rotação.

Tabela 18 – Comparação do RMS dos erros obtidas pelas diferentes versões do PSO nos 4 casos de testes sintéticos da base de dados de Choi e Christensen (2013) (melhores resultados em negrito).

Caso de teste	Componente	PSO-1	PSO-2	PSO-3	PSO-4
Kinect box	x	1,041	1,039	1,018	1,023
	y	1,067	1,059	1,023	1,009
	z	0,383	0,384	0,409	0,404
	Roll	0,210	0,237	0,188	0,201
	Pitch	0,153	0,150	0,136	0,139
	Yaw	0,243	0,267	0,200	0,209
	Média trans.	0,830	0,827	0,817	0,812
	Média rot.	0,202	0,218	0,175	0,183
Milk	x	0,839	0,841	0,884	0,881
	y	0,958	0,955	0,991	0,989
	z	0,186	0,173	0,069	0,067
	Roll	0,170	0,167	0,113	0,111
	Pitch	0,096	0,095	0,074	0,075
	Yaw	0,166	0,165	0,121	0,117
	Média trans.	0,661	0,656	0,648	0,646
	Média rot.	0,144	0,142	0,103	0,101
Orange juice	x	0,844	0,845	0,883	0,882
	y	0,929	0,927	0,935	0,926
	z	0,254	0,243	0,060	0,051
	Roll	0,117	0,106	0,071	0,066
	Pitch	0,078	0,072	0,064	0,062
	Yaw	0,133	0,120	0,082	0,077
	Média trans.	0,676	0,672	0,626	0,619
	Média rot.	0,109	0,099	0,072	0,068
Tide	x	0,922	0,921	0,940	0,936
	y	1,007	1,009	0,979	0,978
	z	0,141	0,135	0,045	0,046
	Roll	0,173	0,154	0,134	0,121
	Pitch	0,094	0,090	0,068	0,069
	Yaw	0,138	0,131	0,086	0,080
	Média trans.	0,690	0,688	0,655	0,653
	Média rot.	0,135	0,125	0,096	0,090

Fonte: Elaborada pelo autor (2021)

Os resultados obtidos pelo *PSO-4* ainda foram comparados com técnicas presentes no estado da arte, como pode ser visto na Tabela 19. Nela é possível observar que o *PSO-4* obteve os melhores resultados na maioria das componentes nos experimentos com os casos de teste *Milk*, *Orange juice*, *Tide*, especialmente em relação às componentes de rotação e a componente z de translação. Já uma das versões do rastreador apresentado em Kehl et al. (2017) obteve os melhores resultados na maioria das componentes no experimento com o *Kinect box* apenas. Se for observado as médias dos erros em relação às componentes de rotação e translação (últimas duas linhas de cada caso de teste), é possível verificar que o *PSO-4* obteve os melhores resultados em todos os casos de teste em relação aos erros de rotação e a técnica em Kehl et al. (2017) obteve os melhores resultados em relação à translação.

Se comparado com as demais técnicas (R. Ueda, 2012; CHOI; CHRISTENSEN, 2013; JÚNIOR; LIMA, 2018; TAN; NAVAB; TOMBARI, 2017), o *PSO-4* obteve os melhores resultados em relação a todas as componentes nos experimentos com os casos de teste *Kinect box* e *Milk*, e na maioria das componentes nos experimentos com os casos de teste *Orange juice* e *Tide*. Se observadas as médias de translação e rotação, o *PSO-4* obteve os melhores resultados em relação a todos os casos de testes.

Tabela 19 – Comparação do RMS dos erros obtidas por diferentes técnicas de rastreamento, avaliadas nos 4 casos de testes sintéticos da base de dados de Choi e Christensen (2013) (melhores resultados em negrito).

Caso de teste	Componente	PCL	Choi e Christensen	Santos Jr e Lima	Tan et al.	Kehl et al.	PSO-4
Kinect box	x	43.99	1.84	1.35	2.15	0.76	1.02
	y	42.51	2.23	1.24	1.23	1.09	1.01
	z	55.89	1.36	2.73	1.16	0.38	0.40
	Roll	7.62	6.41	3.58	0.28	0.17	0.20
	Pitch	1.87	0.76	1.55	0.53	0.18	0.14
	Yaw	8.31	6.32	3.54	0.17	0.20	0.21
	Média trans.	47.46	1.81	1.77	1.51	0.74	0.81
	Média rot.	5.93	4.50	2.89	0.33	0.18	0.18
Milk	x	13.38	0.93	1.41	2.09	0.64	0.88
	y	31.45	1.94	2.37	1.02	0.59	0.99
	z	26.09	1.09	1.20	0.68	0.24	0.07
	Roll	59.37	3.83	6.22	0.14	0.41	0.11
	Pitch	19.58	1.41	1.74	0.35	0.29	0.07
	Yaw	75.03	3.26	5.49	0.17	0.42	0.12
	Média trans.	23.64	1.32	1.66	1.26	0.49	0.65
	Média rot.	51.33	2.83	4.48	0.22	0.37	0.10
Orange juice	x	2.53	0.96	1.09	1.57	0.50	0.88
	y	2.20	1.44	0.90	1.43	0.69	0.93
	z	1.91	1.17	1.15	0.62	0.17	0.05
	Roll	85.81	1.32	0.57	0.42	0.12	0.07
	Pitch	42.12	0.75	0.83	0.75	0.20	0.06
	Yaw	46.37	1.39	0.61	0.16	0.19	0.08
	Média trans.	2.21	1.19	1.05	1.21	0.45	0.62
	Média rot.	58.10	1.15	0.67	0.44	0.17	0.07
Tide	x	1.46	0.83	0.93	2.13	0.34	0.94
	y	2.25	1.37	0.70	0.46	0.49	0.98
	z	0.92	1.20	1.36	0.88	0.18	0.05
	Roll	5.15	1.78	0.97	0.17	0.15	0.12
	Pitch	2.13	1.09	0.88	1.08	0.39	0.07
	Yaw	2.98	1.13	0.99	0.59	0.37	0.08
	Média trans.	1.54	1.13	1.00	1.16	0.34	0.65
	Média rot.	3.42	1.33	0.95	0.61	0.30	0.09

Fonte: Elaborada pelo autor (2021)

6 CONSIDERAÇÕES FINAIS

Este capítulo apresenta uma breve análise dos resultados obtidos a partir dos experimentos realizados nesta tese, expõe as principais contribuições e sugere trabalhos futuros que podem ser realizados como continuação da pesquisa apresentada.

6.1 CONCLUSÕES

A primeira etapa dos experimentos realizados no Capítulo 5 mostrou que técnicas de rastreamento baseadas em otimização que usam PSO podem possibilitar um rastreamento com acurácia e robustez a falhas superior a técnicas baseadas em aprendizagem presentes no estado da arte, com a vantagem de não necessitar de longos períodos de treinamento. Essa etapa mostrou ainda que o uso de GPU para paralelização do algoritmo de PSO usado em técnicas de rastreamento baseadas em otimização pode proporcionar o rastreamento em tempo real, com taxas acima de 20 fps, mantendo resultados com boas acurácias.

Os protótipos com as principais mudanças na técnica de rastreamento que impactaram na acurácia e robustez do rastreamento foram criados e testados nos experimentos da segunda etapa. Dentre essas mudanças, se destacaram o uso de média harmônica na função de aptidão do PSO, o método de seleção de pontos visíveis do modelo do objeto, o uso de amplitudes dinâmicas nos subespaços de busca do PSO e a filtragem das poses encontradas pelo rastreador. O uso de média harmônica possibilitou uma função de aptidão mais eficiente, principalmente nos casos de teste com maior oclusão. A seleção dos pontos visíveis do modelo usando a comparação com mapas de profundidade reduziu a quantidade de pontos do modelo utilizados na avaliação de cada pose, porém evitou usar conjuntos de pontos do modelo que não possuem correspondentes na nuvem de pontos da cena, o que diminui o número de penalidades por distância euclidiana e o ruído acrescentado à aptidão calculada para cada partícula pela função de avaliação do PSO. O cálculo das amplitudes de rotação e translação em tempo de execução possibilitou a criação de subespaços de busca menores a cada instanciação do PSO. Esse cálculo levou em consideração ainda a inércia do objeto alvo ao longo da trajetória, o que diminui de forma eficiente a região do espaço de busca que as partículas percorrem para encontrar boas soluções. A filtragem das poses possibilitou a melhoria da acurácia e a robustez a falhas do rastreador. Quando à melhoria na acurácia, o resultado pôde ser visto de forma

mais clara nos experimentos com o cenário *stability*. Já em relação à melhoria na robustez a falhas, os experimentos no cenário *interaction-hard* mostraram que o rastreador pode falhar aproximadamente 61,9% menos quando usada a filtragem de poses. Os resultados dos experimentos com os protótipos mostraram que as mudanças propostas possibilitaram melhorias significativas na acurácia e na robustez do rastreamento da técnica proposta.

Os experimentos da segunda etapa ainda foram usados para verificar a qualidade do rastreador proposto em relação a técnicas de rastreamento 6-DOF presentes no estado da arte, sejam elas baseadas em aprendizagem tais como Tan et al. (2015), Garon e Lalonde (2017), Garon, Laurendeau e Lalonde (2018), Maroungkas et al. (2020) ou baseadas em otimização (JÚNIOR; LIMA, 2018). Nesses experimentos o rastreador introduzido nesta tese apresentou os melhores resultados na maioria das sequências quando avaliado segundo os erros médios de rotação e de translação. Em relação ao número de falhas durante o rastreamento, o rastreador proposto obteve os melhores resultados em todos os testes realizados com os casos de teste *interaction-hard*, que contém sequências com nível de dificuldade elevado, segundo Garon, Laurendeau e Lalonde (2018).

Na terceira etapa dos experimentos, o estudo da diversidade populacional e testes de convergência mostrou que o enxame converge rapidamente nas primeiras iterações, todavia as partículas apresentam uma constante evolução nas últimas iterações, mostrando que ainda há exploração local mesmo em testes com um critério de parada de 400 iterações. No entanto, essa constante convergência local não parece se converter sempre em ganho de acurácia uma vez que os erros médios de rotação e translação parecem se estabilizar já em experimentos com critério de parada de 200 iterações. Um estudo sobre a função de aptidão mostrou quantitativamente que as poses encontradas pelo PSO foram melhor avaliadas pela função de aptidão que aquelas definidas pelo *ground truth* da base de dados na maioria dos testes, mesmo com as poses do *ground truth* apresentando melhores resultados qualitativos em alguns deles. Resultados quantitativos ainda apontaram que o PSO parece convergir bem em relação às componentes de translação da pose mas não consegue obter resultados tão bons em relação às componentes de rotação.

Ainda na terceira etapa dos experimentos, testes realizados em cenário sintéticos também mostraram que a função de aptidão retorna valores menores quando usadas as poses encontradas pelo PSO em relação às respectivas poses do *ground truth*. Entretanto, experimentos eliminando os pontos pertencentes à borda do modelo transformado mostraram que há uma queda no valor das aptidões e a diferença entre as aptidões das poses do PSO e das poses

do *ground truth* diminui significativamente. Na terceira etapa também foram realizados experimentos que mostraram que há melhoria na acurácia do rastreador quando usados os cortes nos pontos pertencentes à borda do modelo transformado, assim como quando usado um PSO com peso de inércia adaptativo no lugar do peso de inércia temporal. Aparentemente essas mudanças contribuíram para aliviar o problema da acurácia em relação às componentes de rotação das poses encontradas pelo PSO, problema conjecturado anteriormente na Subseção 5.4.2. Os resultados desses experimentos foram superiores a rastreadores do estado da arte em grande parte dos casos.

Os experimentos com o modelo *skull* mostraram que a qualidade do rastreamento usando a técnica proposta pode ser afetada pela falta de homogeneidade na distribuição dos pontos do modelo utilizado. Para corrigir esse problema foi realizada uma amostragem dos seus pontos. Os testes com o cenário *occlusion* com aparato de oclusão na vertical expuseram também as dificuldades da técnica em realizar o rastreamento quando a nuvem de pontos da cena apresenta falhas na região onde se encontra o objeto alvo. Esses casos de testes também mostraram a dificuldade do rastreador em lidar com sequências contendo trechos em que o objeto fica totalmente ocluído por outros objetos da cena.

Os experimentos com os casos de teste do cenário *stability* possibilitaram determinar o grau de *jitter* da técnica proposta em relação às demais. Nesse caso, observou-se que a técnica proposta foi inferior ao rastreador apresentado em Garon, Laurendeau e Lalonde (2018), porém conseguiu superar outros dois presentes em Tan et al. (2015), Garon e Lalonde (2017). Uma vez que, nesse tipo de experimento, o objeto alvo fica em repouso durante toda a sequência e o erro é calculado pela diferença entre poses consecutivas p_q e p_{q-1} preditas pela técnica, os resultados encontrados nesses experimentos indicam que ainda pode haver espaço para melhorias na função de aptidão do rastreador, visto que a pose p_q predita para o quadro q sempre usa a pose p_{q-1} como pose inicial.

Os experimentos com avaliação de desempenho de tempo mostraram que as mudanças propostas, tais como a implementação do PSO completamente em GPU e a seleção da ROI das imagens RGB-D da cena, possibilitaram uma melhora significativa no tempo de processamento da técnica, indicando que futuras otimizações podem possibilitar um rastreamento em tempo real. Como reforço dessa hipótese, a primeira etapa dos experimentos já mostrou que o uso de menores ensembles de partículas e modelos com uma quantidade menor de pontos podem possibilitar o rastreamento em tempo real usando a técnica baseada em PSO.

6.2 CONTRIBUIÇÕES

As principais contribuições do trabalho apresentado nesta proposta são:

- Uma nova formulação para calcular a aptidão de cada partícula durante o rastreamento de objetos com base na média harmônica das distâncias entre coordenadas 3D, cores e vetores normais;
- Um método para cálculo dos limites do subespaço de solução do PSO em tempo de execução por componente de pose e levando em consideração a inércia do objeto alvo nos quadros anteriores;
- Uma técnica para seleção de pontos visíveis do modelo 3D do objeto usando renderização de *Z-buffer* e a sobreposição de mapas de profundidade;
- Uma heurística para seleção dinâmica de uma região de interesse da cena (ROI) ao longo do rastreamento;
- Uma heurística para eliminação dos pontos nas bordas do modelo transformado durante as avaliações das partículas do enxame afim de melhorar a acurácia da técnica;
- Um método para filtragem da trajetória encontrada pelo rastreador que possibilitou melhorias na acurácia, na robustez a falhas e no *jitter* da técnica;
- Uma implementação do algoritmo PSO totalmente em GPU com paralelismo no nível de geração de enxame e avaliação/atualização de partículas;
- Um arcabouço que faz uso dos métodos citados anteriormente para rastreamento de objetos 3D arbitrários robusto à oclusão;
- Um estudo de ablação avaliando as melhorias propostas;
- Uma avaliação quantitativa e qualitativa de acurácia, robustez e tempo de execução da técnica proposta, incluindo também a comparação com métodos do estado da arte, usando sequências de imagens obtidas em ambientes reais com oclusões severas dos objetos rastreados;
- Um estudo a respeito da diversidade populacional com testes de convergência e avaliação quantitativa e qualitativa da função de aptidão do PSO, incluindo experimentos em

base de dados sintética que mede a acurácia do método proposto em relação a técnicas presentes no estado da arte.

As publicações realizadas ao longo do doutorado foram as seguintes:

- Artigo completo diretamente relacionado a este trabalho: SANTOS JÚNIOR, José; LIMA, João Paulo. "Particle swarm optimization for 3D object tracking in RGB-D images". **Computers & Graphics**, v. 76, p. 167-180, 2018;
- Artigo completo diretamente relacionado a este trabalho: SANTOS JÚNIOR, José; LIMA, João Paulo; TEICHRIEB, Veronica. "Occlusion-robust method for RGB-D 6-DOF object tracking with particle swarm optimization". **Expert Systems with Applications**, v. 174, p. 114736, 2021;
- Artigo completo indiretamente relacionado a este trabalho: SANTOS JÚNIOR, José; LIMA, Gustavo; PINTO, Adam; LIMA, João Paulo; TEICHRIEB, Veronica; QUINTINO, Jonysberg; SILVIA, Fabio; SANTOS, Andre; PINHO, Helder. "3D Object Reconstruction Using Stationary RGB Camera". **International Conference on Computer Vision Theory and Applications (VISAPP)**, 2022.

6.3 TRABALHOS FUTUROS

Com relação à função de aptidão do PSO para rastreamento de objetos, sugere-se como trabalho futuro um estudo sobre a heurística usada, com o objetivo de evitar má avaliação de poses corretas. Mudanças na função de aptidão que possibilitassem à técnica lidar com situações em que aparecem buracos na nuvem de pontos em regiões que deviam conter os pontos do objeto alvo poderiam melhorar consideravelmente a acurácia do rastreador. Sugere-se também a realização de otimizações no algoritmo visando diminuir o tempo de processamento da técnica, que podem ser acompanhadas de um estudo que verifique a relação entre a escolha dos principais parâmetros do rastreador (tais como tamanho do enxame, número de iterações, tamanho do modelo, etc.) e a acurácia dos resultados.

Uma vez que os experimentos neste trabalho usaram um PSO padrão, e que existem muitos trabalhos propondo melhorias nesse tipo de meta-heurística, sugere-se ainda investigar o impacto do uso de outros tipos de PSO modernos na acurácia, robustez a falhas e desempenho de tempo do rastreamento. É possível testar diferentes versões do algoritmo do PSO, com

diferentes topologias ou usando funções de aptidão multiobjetivo. A utilização do FESPSO (SUN et al., 2013), por exemplo, que reduz a quantidade de chamadas da função de aptidão sem reduzir a qualidade dos resultados, pode ser uma forma de melhorar o tempo de execução das técnicas de rastreamento, uma vez que o tempo de processamento é proporcional ao número de vezes que as partículas são avaliadas durante as iterações do PSO. Outra alternativa que pode melhorar a acurácia e também reduzir o número de avaliações, consequentemente diminuindo o tempo de processamento, é combinar o método proposto com uma técnica baseada em aprendizagem. Estratégias semelhantes que foram usadas com sucesso, tais como Krull et al. (2014), Tan, Navab e Tombari (2017), apontam nessa direção.

REFERÊNCIAS

- AKKALADEVI, S.; ANKERL, M.; HEINDL, C.; PICHLER, A. Tracking multiple rigid symmetric and non-symmetric objects in real-time using depth data. In: IEEE. *2016 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.], 2016. p. 5644–5649.
- AZAD, P.; MÜNCH, D.; ASFOUR, T.; DILLMANN, R. 6-dof model-based tracking of arbitrarily shaped 3d objects. In: IEEE. *2011 IEEE International Conference on Robotics and Automation*. [S.l.], 2011. p. 5204–5209.
- AZUMA, R. T. A survey of augmented reality. *Presence: teleoperators & virtual environments*, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 6, n. 4, p. 355–385, 1997.
- Backstage. 2020. URL: <https://www.backstage.com>. 2020.
- BARIONI, R. R.; CHAVES, T. M.; FIGUEIREDO, L.; TEICHRIEB, V.; NETO, E. V.; GAMA, A. E. D. Arkanoidar: An augmented reality system to guide biomechanical movements at sagittal plane. In: IEEE. *2017 19th Symposium on Virtual and Augmented Reality (SVR)*. [S.l.], 2017. p. 207–214.
- BERKMANN, J.; CAELLI, T. Computation of surface geometry and segmentation using covariance techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 16, n. 11, p. 1114–1116, 1994.
- BRATTON, D.; KENNEDY, J. Defining a standard for particle swarm optimization. In: IEEE. *2007 IEEE swarm intelligence symposium*. [S.l.], 2007. p. 120–127.
- BROCKETT, R. W. Robotic manipulators and the product of exponentials formula. In: SPRINGER. *Mathematical theory of networks and systems*. [S.l.], 1984. p. 120–129.
- CHOI, C.; CHRISTENSEN, H. I. Rgb-d object tracking: A particle filter approach on gpu. In: IEEE. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.], 2013. p. 1084–1091.
- CLERC, M.; KENNEDY, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE transactions on Evolutionary Computation*, IEEE, v. 6, n. 1, p. 58–73, 2002.
- DENG, X.; MOUSAVIAN, A.; XIANG, Y.; XIA, F.; BRETL, T.; FOX, D. Poserbpf: A rao-blackwellized particle filter for 6-d object pose tracking. *IEEE Transactions on Robotics*, IEEE, 2021.
- DEUTSCHER, J.; BLAKE, A.; REID, I. Articulated body motion capture by annealed particle filtering. In: IEEE. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*. [S.l.], 2000. v. 2, p. 126–133.
- EBERHART, R.; KENNEDY, J. A new optimizer using particle swarm theory. In: IEEE. *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. [S.l.], 1995. p. 39–43.

Every Little Achievement Counts. 2020. URL:

<http://everyachievement.blogspot.com/2018/02/creative-corner-forza-motorsport-7.html>. 2020.

GAMA, A. E. F. D.; CHAVES, T. M.; FIGUEIREDO, L. S.; BALTAR, A.; MENG, M.; NAVAB, N.; TEICHRIEB, V.; FALLAVOLLITA, P. Mirrabilitation: A clinically-related gesture recognition interactive tool for an ar rehabilitation system. *Computer methods and programs in biomedicine*, Elsevier, v. 135, p. 105–114, 2016.

GARON, M.; LALONDE, J.-F. Deep 6-dof tracking. *IEEE transactions on visualization and computer graphics*, IEEE, v. 23, n. 11, p. 2410–2418, 2017.

GARON, M.; LAURENDEAU, D.; LALONDE, J.-F. A framework for evaluating 6-dof object trackers. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2018. p. 582–597.

GOKTURK, S. B.; BOUGUET, J.-Y.; TOMASI, C.; GIROD, B. Model-based face tracking for view-independent facial expression recognition. In: IEEE. *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*. [S.l.], 2002. p. 287–293.

GONZALEZ, R. C.; WOODS, R. E. *Processamento de imagens digitais*. [S.l.]: Editora Blucher, 2000.

HARTLEY, R.; ZISSERMAN, A. *Multiple view geometry*. [S.l.]: Cambridge University Press, Cambridge, 2004.

HODAN, T.; HALUZA, P.; OBDRŽÁLEK, Š.; MATAS, J.; LOURAKIS, M.; ZABULIS, X. T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. In: IEEE. *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. [S.l.], 2017. p. 880–888.

HOFF, W. A.; NGUYEN, K.; LYON, T. Computer-vision-based registration techniques for augmented reality. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Intelligent Robots and Computer Vision XV: Algorithms, Techniques, Active Vision, and Materials Handling*. [S.l.], 1996. v. 2904, p. 538–548.

Insurance Institute for Highway Safety. 2020. URL: <https://www.iihs.org/>. 2020.

JORDÀ, S.; GEIGER, G.; ALONSO, M.; KALTENBRUNNER, M. The reactable: exploring the synergy between live music performance and tabletop tangible interfaces. In: *Proceedings of the 1st international conference on Tangible and embedded interaction*. [S.l.: s.n.], 2007. p. 139–146.

JÚNIOR, J. G. dos S.; LIMA, J. P. S. do M. Particle swarm optimization for 3d object tracking in rgb-d images. *Computers & Graphics*, Elsevier, v. 76, p. 167–180, 2018.

JÚNIOR, J. G. dos S.; LIMA, J. P. S. do M.; TEICHRIEB, V. Occlusion-robust method for rgb-d 6-dof object tracking with particle swarm optimization. *Expert Systems with Applications*, Elsevier, v. 174, p. 114736, 2021.

KATO, H.; BILLINGHURST, M. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In: IEEE. *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*. [S.l.], 1999. p. 85–94.

- KEHL, W.; TOMBARI, F.; ILIC, S.; NAVAB, N. Real-time 3d model tracking in color and depth on a single cpu core. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2017. p. 745–753.
- KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: IEEE. *Proceedings of ICNN'95-international conference on neural networks*. [S.l.], 1995. v. 4, p. 1942–1948.
- KOLLER, D.; DANIILIDIS, K.; THORHALLSON, T.; NAGEL, H.-H. Model-based object tracking in traffic scenes. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 1992. p. 437–452.
- KOLLER, D.; KLINKER, G.; ROSE, E.; BREEN, D.; WHITAKER, R.; TUCERYAN, M. Real-time vision-based camera tracking for augmented reality applications. In: *Proceedings of the ACM symposium on Virtual reality software and technology*. [S.l.: s.n.], 1997. p. 87–94.
- KRULL, A.; MICHEL, F.; BRACHMANN, E.; GUMHOLD, S.; IHRKE, S.; ROTHER, C. 6-dof model based tracking via object coordinate regression. In: SPRINGER. *Asian Conference on Computer Vision*. [S.l.], 2014. p. 384–399.
- LEPETIT, V. Monocular model-based 3d tracking of rigid objects: A survey. *Foundations and Trends in Computer Graphics and Vision*, v. 1, n. 1, p. 1–89, 2005.
- LIMA, J. P.; ROBERTO, R.; SIMÕES, F.; ALMEIDA, M.; FIGUEIREDO, L.; TEIXEIRA, J. M.; TEICHRIEB, V. Markerless tracking system for augmented reality in the automotive industry. *Expert systems with applications*, Elsevier, v. 82, p. 100–114, 2017.
- LIMA, J. P. S. d. M. Object detection and pose estimation from rectification of natural features using consumer rgb-d sensors. Universidade Federal de Pernambuco, 2014.
- MAJCHER, M.; KWOLEK, B. 3d model-based 6d object pose tracking on rgb images using particle filtering and heuristic optimization. In: *VISIGRAPP (5: VISAPP)*. [S.l.: s.n.], 2020. p. 690–697.
- MARINI, F.; WALCZAK, B. Particle swarm optimization (pso). a tutorial. *Chemometrics and Intelligent Laboratory Systems*, Elsevier, v. 149, p. 153–165, 2015.
- MAROUGKAS, I.; KOUTRAS, P.; KARDARIS, N.; RETSINAS, G.; CHALVATZAKI, G.; MARAGOS, P. How to track your dragon: A multi-attentional framework for real-time rgb-d 6-dof object pose tracking. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 2020. p. 682–699.
- Matellio. 2020. URL: <https://www.matellio.com/blog/ar-for-customer-engagement-when-and-how-to-use-it-right/>. 2020.
- NEUMANN, U.; YOU, S. Natural feature tracking for augmented reality. *IEEE Transactions on Multimedia*, IEEE, v. 1, n. 1, p. 53–64, 1999.
- Newscast Studio. 2020. URL: <https://www.newscaststudio.com/2017/08/08/augmented-reality-roundup/>. 2020.
- NICKABADI, A.; EBADZADEH, M. M.; SAFABAKHSH, R. A novel particle swarm optimization algorithm with adaptive inertia weight. *Applied soft computing*, Elsevier, v. 11, n. 4, p. 3658–3670, 2011.

OIKONOMIDIS, I.; KYRIAZIS, N.; ARGYROS, A. A. Tracking the articulated motion of two strongly interacting hands. In: IEEE. *2012 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.], 2012. p. 1862–1869.

PRESS, W. H.; TEUKOLSKY, S. A. Savitzky-golay smoothing filters. *Computers in Physics*, American Institute of Physics, v. 4, n. 6, p. 669–672, 1990.

PRICE, E.; LAWLESS, G.; LUDWIG, R.; MARTINOVIC, I.; BÜLTHOFF, H. H.; BLACK, M. J.; AHMAD, A. Deep neural network-based cooperative visual tracking through multiple micro aerial vehicles. *IEEE Robotics and Automation Letters*, IEEE, v. 3, n. 4, p. 3193–3200, 2018.

Professor Josh. 2020. URL: https://professorjosh.com/2014/04/10/augmentative-reality-is-the-real-game-changer-in-education/165516910203404577185138582164106_o-1/.2020.

QIAN, C.; SUN, X.; WEI, Y.; TANG, X.; SUN, J. Realtime and robust hand tracking from depth. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2014. p. 1106–1113.

QR Code Generator. 2020. URL: <https://br.qr-code-generator.com>. 2020.

R. Ueda. 2012. URL: <https://goo.gl/o5rmzU>. 2012.

REHG, J. M.; KANADE, T. Visual tracking of high dof articulated structures: an application to human hand tracking. In: SPRINGER. *European conference on computer vision*. [S.l.], 1994. p. 35–46.

SHI, Y.; EBERHART, R. C. Parameter selection in particle swarm optimization. In: SPRINGER. *International conference on evolutionary programming*. [S.l.], 1998. p. 591–600.

SMINCHISESCU, C.; TRIGGS, B. Covariance scaled sampling for monocular 3d body tracking. In: IEEE. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. [S.l.], 2001. v. 1, p. I–I.

SUN, C.; ZENG, J.; PAN, J.; XUE, S.; JIN, Y. A new fitness estimation strategy for particle swarm optimization. *Information sciences*, Elsevier, v. 221, p. 355–370, 2013.

TAN, D. J.; NAVAB, N.; TOMBARI, F. Looking beyond the simple scenarios: Combining learners and optimizers in 3d temporal tracking. *IEEE transactions on visualization and computer graphics*, IEEE, v. 23, n. 11, p. 2399–2409, 2017.

TAN, D. J.; TOMBARI, F.; ILIC, S.; NAVAB, N. A versatile learning-based 3d temporal tracker: Scalable, robust, online. In: *Proceedings of the IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2015. p. 693–701.

TEICHRIEB, V.; LIMA, J.; APOLINÁRIO, E.; FARIAS, T.; BUENO, M.; KELNER, J.; SANTOS, I. A survey of online monocular markerless augmented reality. *International Journal of Modeling and Simulation for the Petroleum Industry*, Citeseer, v. 1, n. 1, p. 1–7, 2007.

TJADEN, H.; SCHWANECKE, U.; SCHÖMER, E.; CREMERS, D. A region-based gauss-newton approach to real-time monocular multiple object tracking. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 41, n. 8, p. 1797–1812, 2018.

Trenta Minuti. 2020. URL: <https://www.trentaminuti.it/cose-la-realta-aumentata.html>. 2020.

VACCHETTI, L.; LEPETIT, V.; FUA, P. Stable real-time 3d tracking using online and offline information. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 26, n. 10, p. 1385–1391, 2004.

WAGNER, D.; LANGLOTZ, T.; SCHMALSTIEG, D. Robust and unobtrusive marker tracking on mobile phones. In: IEEE. *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*. [S.l.], 2008. p. 121–124.

Wikimedia Commons. 2020. URL: <https://pt.m.wikipedia.org/wiki/Ficheiro:Xbox-360-Kinect-Standalone.png>. 2020.

XIANG, Y.; SCHMIDT, T.; NARAYANAN, V.; FOX, D. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

APÊNDICE A – VÍDEOS E RESULTADOS

Este apêndice lista os vídeos, e seus respectivos hiperlinks, usados para apresentar os resultados dos experimentos descritos nesta tese:

- *Resultados etapa 1*: mostra resultados qualitativos obtidos na primeira etapa dos experimentos, acessível em <https://youtu.be/o_q6mRyEBVM>
- *Resultados etapa 2*: mostra resultados qualitativos obtidos na segunda etapa dos experimentos, acessível em <<https://youtu.be/QC-qusY6TqM>>