

Rastreamento de objeto esférico por cooperação multi-robô

Thulio Guilherme Silva de Amorim



CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, 2019

Thulio Guilherme Silva de Amorim

Rastreamento de objeto esférico por cooperação multi-robô

Monografia apresentada ao curso de Ciência da Computação do Centro de Informática, da Universidade Federal da Paraíba, como requisito para a obtenção do grau de Bacharel em Ciência da Computação

Orientador: Tiago Pereira do Nascimento

Outubro de 2019

Catálogo na publicação
Seção de Catalogação e Classificação

A524r Amorim, Thulio Guilherme Silva de.

Rastreamento de objeto esférico por cooperação
multi-robô / Thulio Guilherme Silva de Amorim. - João
Pessoa, 2019.

66 f. : il.

Orientação: Tiago Pereira do Nascimento.
Monografia (Graduação) - UFPB/CI.

1. Rastreamento visual. 2. Robótica móvel. 3. Sistema
multi-robôs. 4. Fusão de dados. 5. Filtro de
partículas. I. Nascimento, Tiago Pereira do. II. Título.

UFPB/CI



CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

Trabalho de Conclusão de Curso de Ciência da Computação intitulado *Rastreamento de objeto esférico por cooperação multi-robô* de autoria de Thulio Guilherme Silva de Amorim, aprovada pela banca examinadora constituída pelos seguintes professores:

Prof. Dr. Tiago Pereira do Nascimento
Universidade Federal da Paraíba

Prof. Dr. Claurton de Albuquerque Siebr
Universidade Federal da Paraíba

Prof. Dr. Leonardo Vidal Batista
Universidade Federal da Paraíba

Coordenador(a) do Departamento de Informática
Gustavo Henrique Matos Bezerra Motta
CI/UFPB

João Pessoa, 1 de outubro de 2019

“Você não pode ver o que eu vejo porque vê o que você vê. Não pode saber o que sei porque sabe o que você sabe. O que eu vejo e o que sei não podem ser acrescentados ao que você vê e ao que você sabe porque são coisas diferentes. Também não podem substituir o que você vê e o que sabe porque isso seria substituir você mesmo.”

(Douglas Adams)

AGRADECIMENTOS

Desejo agradecer aos meus pais por todo o carinho e compreensão que me foi concedido durante todos os meus anos de universidade e que sem o auxílio deles todo este percurso não seria possível, ao professor Tiago Nascimento pela ajuda no desenvolvimento deste trabalho e a todos as pessoas que tive a honra de conhecer durante esses anos de curso. Tenham certeza que existe um pedaço de cada um de vocês dentro deste trabalho.

RESUMO

Na Robótica Móvel, área na qual o rastreamento visual é um subproblema de extrema importância, uma das soluções encontradas para reduzir os problemas provenientes da oclusão parcial e total é a utilização de múltiplos robôs. Foi implementado um sistema com o intuito de investigar os impactos da fusão de dados no rastreamento visual no espaço tridimensional por multi-robôs na ocorrência de oclusão parcial e total. O sistema conta com duas abordagens de rastreamento visual de um objeto esférico. A primeira consiste no rastreamento visual não cooperativo onde é utilizado apenas os dados oriundos de um único robô. A segunda consiste no rastreamento visual cooperativo onde é utilizado a fusão dos dados oriundos de uma equipe de robôs pelo emprego do filtro de partículas. Para avaliar ambas abordagens, foi simulado um ambiente de rastreamento visual com oclusões parciais e totais onde o rastreamento foi realizado por uma equipe de robôs. O resultado do experimento mostra que a abordagem não cooperativa apresentou um custo computacional inferior ao da abordagem cooperativa contudo a trajetória inferida foi prejudicada pelas oclusões, fato que não ocorreu na abordagem cooperativa devido a fusão dos dados.

Palavras-chave: Rastreamento visual, Robótica móvel, Sistema multi-robôs, Fusão de dados, Filtro de partículas.

ABSTRACT

In Mobile Robotics, where visual tracking is an extremely important subproblem, one of the solutions found to reduce the problems arising from partial and total occlusion is the use of multiple robots. A system was implemented to investigate the impacts of data fusion on multi-robot visual tracking in three-dimensional space on the occurrence of partial and total occlusion. The system has two approaches to visual tracking of a spherical object. The first is non-cooperative visual tracking where only the data from a single robot is used. The second is a cooperative visual tracking where the data from a team of robots is fused using a particle filter. To evaluate both approaches, a visual tracking environment with partial and total occlusions was simulated where the tracking was performed by a team of robots. The result of the experiment shows that the non-cooperative approach presented a lower computational cost than the cooperative approach but the inferred trajectory was impaired by the occlusions, a fact that did not occur in the cooperative approach due to the data fusion.

Key-words: Visual tracking, Mobile robotics, Multi-robot systems, Data fusion, Particle filter

LISTA DE FIGURAS

1	Modelo ideal de câmera <i>pinhole</i> que descreve a relação entre um ponto 3D $(X, Y, Z)^T$ e sua correspondente projeção 2D (u, v) no plano da imagem (adaptado de [30])	22
2	Sistema de coordenadas global e o sistema de coordenadas local do robô [36]	23
3	Sequência padrão de etapas do PADI (adaptado de [20])	24
4	Esquema do cubo de cores RGB. Pontos ao longo da diagonal principal têm valores de cinza, do preto na origem ao branco no ponto $(1, 1, 1)$ [12] .	26
5	Exemplos de elementos estruturantes simples	27
6	Ilustração das definições de choque e encaixe de um elemento estruturante sobre uma imagem binária. O quadrado de cor cinza representa a atribuição do valor um enquanto o quadrado de cor branco representa a atribuição do valor zero	28
7	Fluxograma genérico de um rastreador de objeto	31
8	Representação da evolução de um estado x dentro de um sistema dinâmico	34
9	Turtlebot 2 [9]	36
10	Fluxograma do sistema de rastreamento visual no espaço 3D	38
11	Sequência das operações realizadas sobre a imagem e suas respectivas classificações referente a sequência de etapas do PADI	39
12	Relação entre as distâncias entre o eixo X e Z com a distância do objeto em relação a câmera z	43
13	Distância mensuradas por meio da odometria, sensor de profundidade do Kinect e funções de interpolação durante o experimento	45
14	Rotações realizadas para ir do sistema de coordenadas da tela para a da câmera	45
15	Rotações realizadas para ir do sistema de coordenadas da câmera para a do robô	46
16	Fluxograma do filtro de partículas empregado para o rastreamento visual .	48
17	Configuração do campo de visão	49
18	Informações destacadas sobre o campo de visão	50

19	Organização do ambiente elaborado para a realização do experimento. A linha tracejada representa a trajetória que foi planejada para ser feita pelo objeto de interesse. A variável $C_{Obj\ i}$ representa a posição do centroide do objeto i no sistema de coordenadas global em metros, a variável $D_{Obj\ i}$ representa, respectivamente, a largura e o comprimento do objeto i em metros e $P_i = (x_i, y_i, \theta_i)$ representa a posição x_i e y_i no sistema de coordenadas global em metros e o ângulo de orientação θ_i em radianos do robô i	54
20	Imagem contendo a objeto de interesse fixado em cima de um Turtlebot	55
21	Imagem contendo o ambiente elaborado para a realização do experimento	55
22	A cada 30 milissegundos, contando a partir do início do experimento, o erro de estimação da posição do objeto esférico no espaço do mundo foi calculado como a distância euclidiana, em metros, entre a posição do objeto esférico inferida por meio do rastreamento e a posição do objeto esférico obtida pela odometria. Os gráficos da coluna à esquerda representam o erro no espaço do mundo do rastreamento do objeto esférico na abordagem não cooperativa enquanto os gráficos da coluna a direita representam o erro do rastreamento do objeto esférico na abordagem cooperativa no espaço do mundo. Quando um robô não consegue inferir a posição do objeto, a última posição do objeto de interesse é dada como indefinida, o valor do erro não é computado e nem exibido no gráfico	57
23	Os gráficos na parte superior representam o histograma do erro de estimação da posição do objeto esférico no espaço do mundo dos robôs durante a abordagem não cooperativa. Os gráficos na parte inferior representam o histograma do erro de estimação da posição do objeto esférico no espaço do mundo dos robôs durante a abordagem cooperativa	59

LISTA DE TABELAS

1	Algoritmo do filtro de partículas [42]	35
2	Algoritmo relacionado a etapa de inicialização das partículas do filtro de partículas	51
3	Algoritmo relacionado a etapa de correção dos pesos	52
4	Considerando uma execução como o conjunto de operações que são realizadas sobre as entradas para inferir a posição do objeto de interesse, quantidade média de execuções por segundo do algoritmo de rastreamento de cada abordagem por cada robô	58
5	Porcentagem de execuções nas quais a posição do objeto esférico não pode ser inferida	58

LISTA DE ABREVIATURAS

2D	Bidimensional
3D	Tridimensional
ADI	Análise Digital de Imagens
FDP	Função de Densidade de Probabilidade
PDI	Processamento Digital de Imagens
PADI	Processamento e Análise Digital de Imagens
ROS	Robot Operating System

Sumário

1	INTRODUÇÃO	18
1.1	Definição do Problema	18
1.2	Premissas e Hipóteses	19
1.3	Objetivo geral	19
1.4	Objetivos específicos	19
1.5	Estrutura da monografia	20
2	CONCEITOS GERAIS E REVISÃO DA LITERATURA	21
2.1	Visão Computacional em Robótica	21
2.2	Geometria Projetiva	21
2.3	Sistema de coordenadas cartesianas	22
2.4	Processamento e Análise Digital de Imagens	23
2.5	Espaço de cores	25
2.6	Operações morfológicas	26
2.6.1	Operação morfológica de erosão	28
2.6.2	Operação morfológica de dilatação	28
2.6.3	Operação morfológica de abertura	29
2.7	Conceitos básicos sobre Rastreamento de Objetos	29
2.7.1	Modelo de movimento	30
2.7.2	Modelo de aparência	30
2.7.3	Mecanismo de decisão	32
2.8	Rastreamento de objetos em Robótica	32
2.9	Rastreamento por Filtro de Partículas	33
3	METODOLOGIA	36
3.1	Métodos e equipamentos	36
3.2	Configuração do sistema	37
3.3	Definição do objeto de interesse	37
3.4	Detecção do objeto de interesse	37

3.5	Estimação da posição do objeto	41
3.6	Fusão dos dados	47
3.7	Experimento	53
4	APRESENTAÇÃO E ANÁLISE DOS RESULTADOS	56
5	CONCLUSÕES E TRABALHOS FUTUROS	60
	REFERÊNCIAS	61

1 INTRODUÇÃO

O rastreamento de objetos tem sido um dos tópicos fundamentais de pesquisa dentro da Visão Computacional e possui diversas aplicações dentro da Robótica Móvel.

Dado o estado inicial (por exemplo, posição e dimensões) de um objeto alvo em uma primeira imagem, o objetivo do rastreamento é estimar os estados do alvo nos quadros subsequentes [49]. Dominar a capacidade de perseguir de forma confiável e eficiente um determinado alvo e ser robusto para vários fenômenos incômodos é a chave para muitas aplicações *offline* ou *online* que exploram dados de vídeo [24].

Para aproveitar-se dos recursos disponíveis e, por consequência, aumentar a chance de êxito no rastreamento de um objeto alvo, ou em qualquer outra tarefa que possa ser realizada de forma cooperativa, uma abordagem comumente empregada no rastreamento é a utilização de mais de um robô durante a execução. Conforme [11], muitos esforços têm sido direcionados para a implantação de grupos de robôs móveis autônomos em rede que interagem autonomamente entre si e com o ambiente para melhorar significativamente a eficiência.

Quando existe a interação entre robôs, a comunicação entre eles permite a possibilidade da fusão dos dados coletados do ambiente. A fusão de dados é uma técnica avançada para combinar informações provenientes de várias fontes, a fim de obter resultados mais precisos [2]. Isso permite agrupar as informações de várias fontes para formar uma imagem unificada. Os sistemas de fusão de dados são agora amplamente utilizados em várias áreas, como redes de sensores, Robótica, processamento de vídeo e imagem e design de sistema inteligente [23].

1.1 Definição do Problema

Embora muito esforço já tenha sido colocado na comunidade de rastreamento, o rastreamento robusto de longo prazo ainda é um problema desafiador e um tópico de pesquisa ativo devido à alterações de iluminação e postura, objetos que possuem movimentos complexos, oclusão total ou parcial e cenários do mundo real [18]. Quando se utiliza um único sensor, unidade responsável pela aquisição de informações do ambiente [44], ajustar como o método de rastreamento lida com esses problemas tem grande impacto no seu custo computacional.

Rastrear objetos ou pessoas que se movem em grandes espaços, onde o campo de visão dos sensores comuns é relativamente pequeno quando comparado à área monitorada, é melhor alcançado recorrendo a equipes de sensores cooperativos, estáticos ou dinâmicos, por exemplo, montados em robôs móveis [1]. Em virtude desses fatos, trabalhos recentes sobre rastreamento buscam elaborar métodos que possibilite o uso de múltiplos sensores.

A combinação dos dados traz diversas vantagens envolvendo principalmente o aprimoramento na autenticidade ou disponibilidade de dados. Contudo há vários problemas que tornam a fusão de dados uma tarefa desafiadora [23].

1.2 Premissas e Hipóteses

Como hipótese principal, se tem que a fusão dos dados capturados do ambiente oriundo de sensores distintos torna o rastreamento visual de um objeto no espaço tridimensional ($3D$) mais robusto a oclusões parciais e totais.

1.3 Objetivo geral

O objetivo geral consiste em investigar o impacto da fusão de dados no rastreamento visual no espaço $3D$ por multi-robôs quando ocorre oclusões parciais e totais. Para tal fim deve ser implementado um sistema que inclua duas abordagens de rastreamento. A primeira abordagem consiste no rastreamento não cooperativo onde cada robô de uma equipe busca rastrear o objeto de interesse utilizando apenas os dados obtidos por seus sensores. A segunda abordagem consiste no rastreamento visual cooperativo onde cada membro de uma equipe utiliza os dados obtidos por seus sensores juntamente com os dados dos demais membros para realizar a fusão desses dados.

1.4 Objetivos específicos

Para a implementação do sistema, os objetivos específicos comuns a ambas abordagens são:

- Definir qual será o objeto de interesse e selecionar os atributos que podem ser utilizados para detectar o objeto nas imagens;
- Determinar como será feita a aquisição das imagens e quais as operações que serão realizadas sobre a mesma com o intuito de detectar o objeto de interesse;
- Como o rastreamento deve ser realizado no espaço $3D$, definir o cálculo para transformar a posição do objeto de interesse detectado em imagem para o espaço $3D$;
- Elaborar um experimento que inclua oclusão parcial e total para avaliar as abordagens de rastreamento desenvolvidas.

Alguns objetivos específicos são exclusivos da abordagem cooperativa devido a fusão dos dados, sendo eles:

- Definir qual o mecanismo de comunicação que será utilizado para realização da troca de dados;
- Determinar como será realizado a fusão dos dados provenientes dos robôs envolvidos no rastreamento.

1.5 Estrutura da monografia

Na seção de Conceitos Gerais, como parte inicial é mencionado a importância da Visão Computacional no campo da Robótica na seção 2.1. Em relação a Visão Computacional, os tópicos de geometria projetiva, seção 2.2, e sistema de coordenadas, seção 2.3, são pontuados.

Ainda nos Conceitos Gerais, como o Processamento e Análise Digital de Imagens está comumente presente na Visão Computacional, uma introdução ao Processamento e Análise Digital de Imagens é realizado na seção 2.4. Como tópicos do Processamento e Análise Digital de Imagens, foram mencionados o espaço de cores, seção 2.5, e operações morfológicas, seção 2.6.

Os conceitos básicos sobre Rastreamento de Objetos são mencionados dentro da seção 2.7 juntamente com a importância do rastreamento de objetos em Robótica, seção 2.8, e o rastreamento por Filtro de Partículas, seção 2.9, o qual é o método de rastreamento por fusão de dados utilizado dentro deste trabalho.

Dentro da metodologia, na seção 3.1 se encontra os métodos e equipamentos utilizados para a realização do trabalho. Na seção 3.2 é descrito a configuração do sistema implementado. Na seção 3.3 é descrito o objeto de interesse que será rastreado. Na seção 3.4 é detalhado quais as operações de Processamento e Análise Digital de Imagens são utilizadas para detectar o objeto nas imagens. Na seção 3.5 é especificado como é realizado a estimação da posição do objeto no espaço $3D$ a partir de sua posição na imagem. Na seção 3.6 é descrito como é realizado a fusão dos dados oriundos dos sensores de uma equipe de robôs. Por fim, na seção 3.7 é apresentado como foi realizado o experimento para gerar os resultados.

2 CONCEITOS GERAIS E REVISÃO DA LITERATURA

2.1 Visão Computacional em Robótica

A Visão Computacional é um grande campo relacionado ao processamento de imagens a fim de melhorá-las para benefício humano, interpretar o conteúdo de um ambiente ou criar um modelo $3D$ correspondente à cena [30].

Um robô consiste em uma máquina que atua dentro de um ambiente, planejando suas ações por meio dos dados coletados por seus sensores para atingir um objetivo. Esses dados são cruciais para que o robô alcance suas metas. Conforme [30], os olhos são um sensor muito eficaz para reconhecimento, navegação, desvio de obstáculos e manipulação, de modo que a visão há tempos tem interessado os pesquisadores de Robótica.

2.2 Geometria Projetiva

A Geometria Projetiva possui uma importância significativa para a aquisição de imagens $3D$. Por meio da Geometria Projetiva chegamos ao modelo de câmera *pinhole* que define a relação geométrica entre um ponto $3D$ e sua projeção bidimensional ($2D$) correspondente no plano da imagem. Ao usar um modelo de câmera pinhole, esse mapeamento geométrico de $3D$ para $2D$ é chamado de projeção perspectiva [30].

A projeção perspectiva, também chamada de transformação perspectiva, é formalizada na equação 1. Onde $(X, Y, Z)^T$ representa a posição de um ponto no espaço $3D$, a variável (u, v) representa a projeção do ponto (X, Y, Z) no plano de imagem, a variável f representa a distância focal, as variáveis (c_x, c_y) representam as coordenadas do ponto principal do plano da imagem e $\lambda = Z$ representa o fator de escala homogêneo. Os cálculos realizados na projeção perspectiva que levam pontos do espaço $3D$ para o $2D$ são necessários para realizar o processo inverso.

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (1)$$

As variáveis f, c_x e c_y são tidas como os parâmetros intrínsecos da câmera que variam de acordo com sua configuração e são comumente obtidos por meio da calibração da câmera.

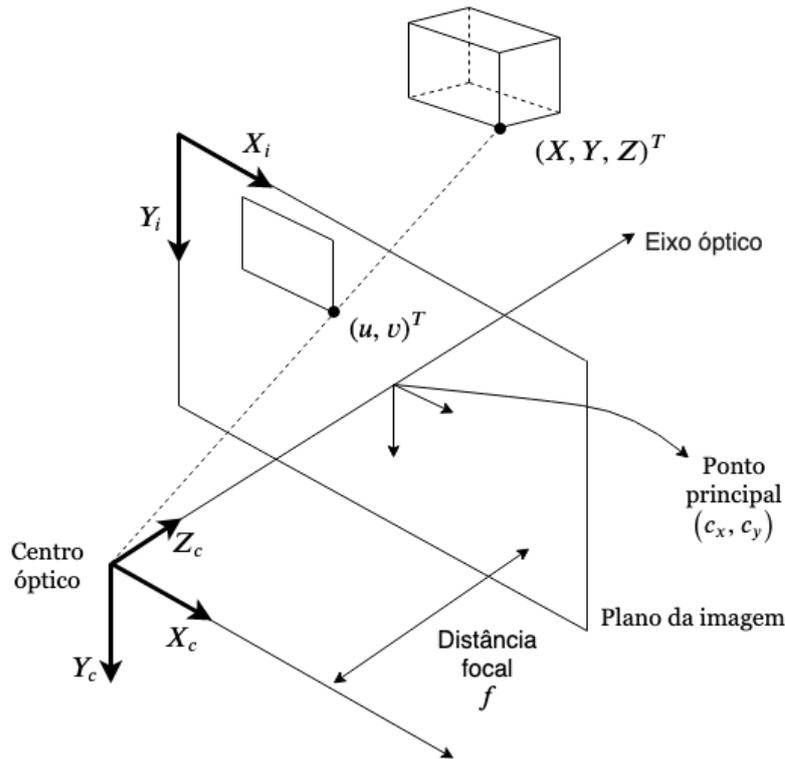


Figura 1: Modelo ideal de câmera *pinhole* que descreve a relação entre um ponto 3D $(X, Y, Z)^T$ e sua correspondente projeção 2D (u, v) no plano da imagem (adaptado de [30])

2.3 Sistema de coordenadas cartesianas

Sistema de coordenadas cartesianas é um conjunto de eixos ortogonais que se cruzam em um ponto conhecido como origem. Sistemas de coordenadas são utilizados para descrever a posição de um objeto por meio de um ponto. Entretanto objetos reais são compostos por uma infinidade de pontos e também possuem uma orientação. Para contornar esse problema, um sistema de coordenadas é anexado ao objeto de interesse onde todos os pontos do objeto são descritos como um vetor constante em relação ao sistema de coordenadas [6].

A posição e orientação de um robô é geralmente definida em dois sistemas de coordenadas: local e global. O sistema de coordenadas local é o sistema de coordenadas anexado ao robô e é utilizado para operações relacionadas a posição e orientação do mesmo, como exemplo a mensuração de deslocamento. Já o sistema de coordenadas global é referente ao ambiente no qual o robô se encontra e serve para relacionar a posição e orientação do robô em relação aos objetos que se encontram em sua volta.

Tendo um robô como um corpo rígido sobre rodas operando em um plano horizontal, os sistemas de coordenadas local e global são descritos na figura 2 onde os eixos X_I e Y_I representam o sistema de coordenadas global com a origem sendo selecionada de

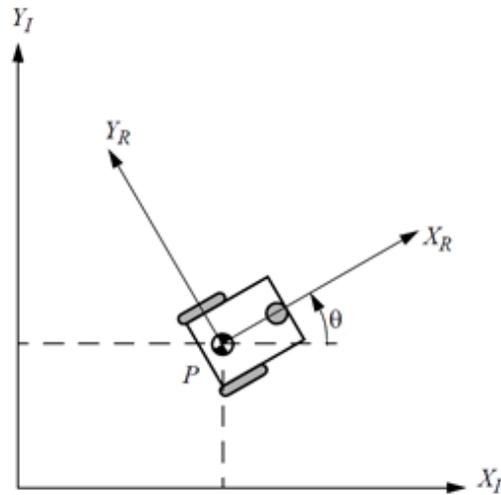


Figura 2: Sistema de coordenadas global e o sistema de coordenadas local do robô [36]

forma arbitrária, os eixos X_R e Y_R representam o sistema de coordenadas local do robô, o ponto P consiste em um ponto do robô que é tido como ponto de referência de distância e a variável θ representa a diferença angular entre os sistemas de coordenadas globais e local do robô [36].

A posição e orientação do robô no sistema de coordenadas global é tido como a pose do robô e é representada pela equação 2 onde as coordenadas (x, y) representam a posição do robô no sistema de coordenadas global e θ representa a diferença angular entre os sistemas de coordenadas global e local do robô.

$$P = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (2)$$

2.4 Processamento e Análise Digital de Imagens

O Processamento e Análise Digital de Imagens (PADI) consiste na junção de duas áreas: o Processamento Digital de Imagens (PDI) e a Análise Digital de Imagens (ADI). Conforme [32], o PDI é um conjunto de técnicas que utiliza operações matemáticas que visa melhorar a imagem, corrigindo defeitos de aquisição e/ou realçando detalhes de interesse enquanto a ADI consiste na extração e tratamento de dados quantitativos a partir de imagens digitais. Comumente o termo PDI é encontrado na literatura referindo-se a ambas as áreas.

De modo geral, PADI abrange os processos cujas entradas e saídas são imagens e, além disso, processos que extraem atributos das imagens, incluindo o reconhecimento de

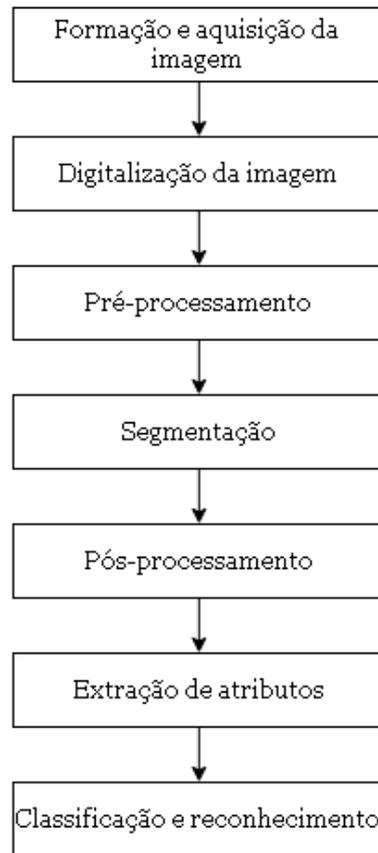


Figura 3: Sequência padrão de etapas do PADI (adaptado de [20])

objetos individuais [12]. A sequência padrão de etapas do PADI é descrita na figura 3. A partir dessa sequência, procedimentos e rotinas de PADI são idealizadas e implementadas. Os conceitos de cada etapa são descritos como:

- Formação e aquisição da imagem: etapa relacionada à capacidade e qualidade do equipamento que realiza a captura da imagem [20];
- Digitalização da imagem: etapa que transforma efetivamente a imagem em sua representação numérica inicial (e original). A partir desta etapa é possível realizar operações em cada pixel da imagem [20];
- Pré-processamento: essa etapa tem como objetivo melhorar a imagem, corrigindo defeitos oriundos da aquisição e realçando detalhes de interesse, de modo a facilitar sua visualização ou segmentação [32]. Alguns exemplos de operações são a redução de ruído, ajuste de brilho e/ou contraste, conversão de espaço de cores, entre outros;
- Segmentação: a segmentação subdivide uma imagem em suas regiões ou objetos constituintes. O nível de detalhe para o qual a subdivisão é transportada depende do problema que está sendo resolvido [12]. Essa etapa é considerada a mais crítica

do PADI pois qualquer erros ou distorções presentes nessa etapa poderão invalidar os resultados [20]. Algumas operações que são realizadas nessa etapa são detecção de pontos, linhas ou contornos, limiarização, entre outros;

- Pós-processamento: essa etapa busca aprimorar o resultado da segmentação onde as características definidas na etapa anterior podem ser aumentadas, realçadas, suavizadas e corrigidas de distorções [32, 20]. Um exemplo de operação realizada na etapa de pós-processamento são as operações morfológicas;
- Extração de atributos: a extração de atributos consiste na etapa de análise onde se transforma as características da imagem em dados quantitativos [20]. Os segmentos dentro da imagem são separados e suas características, como posição ou tamanho, são mensuradas;
- Classificação e reconhecimento: essa etapa consiste na identificação dos segmentos presentes na imagem ou da imagem em si. Conforme [32], nessa etapa se busca transformar as informações obtidas pelas etapas anteriores em conhecimento.

2.5 Espaço de cores

O uso da cor no processamento de imagens é motivado por dois fatores principais. Primeiro, a cor é um descritor poderoso que geralmente simplifica a identificação e a extração de objetos de uma cena. Os seres humanos podem discernir milhares de tonalidades e intensidades de cores, em comparação com apenas duas dúzias de tons de cinza [12].

Um espaço de cores é um método pelo qual podemos especificar, criar e visualizar cores. Como seres humanos, podemos definir uma cor por seus atributos de brilho, matiz e cor. Um computador pode descrever uma cor usando as quantidades de emissão de fósforo vermelho, verde e azul necessárias para corresponder a uma cor. [10].

Entre os principais espaços de cores estão o espaço RGB e o conjunto de espaço de cores HSV.

No modelo RGB, cada cor aparece em seus principais componentes espectrais: vermelho, verde e azul. Este modelo é baseado em um sistema de coordenadas cartesianas. O subespaço de cores de interesse é o cubo mostrado na figura 4, em que os valores RGB estão nos três cantos; ciano, magenta e amarelo estão em outros três cantos; o preto está na origem e o branco está no canto mais distante da origem. Neste modelo, a escala de cinza (pontos de valores RGB iguais) se estende de preto para branco ao longo da linha que une esses dois pontos. As cores diferentes neste modelo são pontos no ou dentro do cubo [12]. Este modelo de cores é fácil de implementar [10] e nenhuma transformação

é necessária para exibir as informações na tela, por esses motivos é considerado como o espaço de cor base para várias aplicações [19].

Em contra partida, o espaço de cor RGB possui dependência de dispositivo, onde a cor resultante é afetada pelas ferramentas usadas para sua exibição [19], e não é adequado para descrever cores em termos práticos para a interpretação humana [12].

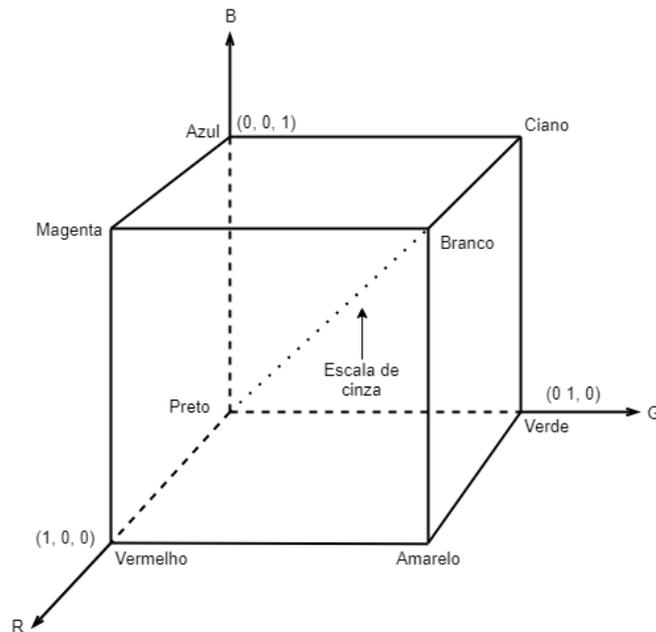


Figura 4: Esquema do cubo de cores RGB. Pontos ao longo da diagonal principal têm valores de cinza, do preto na origem ao branco no ponto $(1, 1, 1)$ [12]

O termo HSV (Hue, Saturation and Value - Matiz, Saturação e Valor), também conhecido como HSB (Hue, Saturation and Brightness - Matiz, Saturação e Brilho), pertence a uma família de espaços de cores alternativos ao espaço de cor RGB. Conforme [10], essa família de espaços de cores são transformações lineares do espaço RGB e são, portanto, dependentes de dispositivos e não lineares. Sua vantagem está na maneira extremamente intuitiva de especificar cores. É muito fácil selecionar um matiz desejado e depois modificá-lo levemente ajustando sua saturação e intensidade.

2.6 Operações morfológicas

A palavra morfologia geralmente denota um ramo da biologia que lida com a forma e a estrutura de animais e plantas. Usamos a mesma palavra aqui no contexto da morfologia matemática como uma ferramenta para extrair componentes de imagem [12]. A morfologia matemática refere-se a um ramo do processamento e análise de imagens não lineares desenvolvido inicialmente por Georges Matheron e Jean Serra que se concentra na estrutura geométrica dentro de uma imagem [8].

A morfologia oferece uma abordagem unificada e poderosa para vários problemas de processamento de imagem. Os conjuntos na morfologia matemática representam objetos em uma imagem [12]. Operações morfológicas sondam uma imagem com uma pequena forma ou *template* chamado elemento estruturante. Conforme [8], a ideia básica do processamento morfológico é investigar uma imagem a partir de um elemento estruturante e quantificar a maneira pela qual o elemento estruturante se encaixa (ou não) na imagem.

Um elemento estruturante consiste em uma matriz de dimensões fixas, geralmente com linhas e colunas de tamanhos iguais e de valor ímpar, composta exclusivamente por zeros e uns cuja a organização desses valores dentro da matriz define a forma do elemento estruturante. Alguns exemplos de elementos estruturantes simples são apresentados na figura 5.

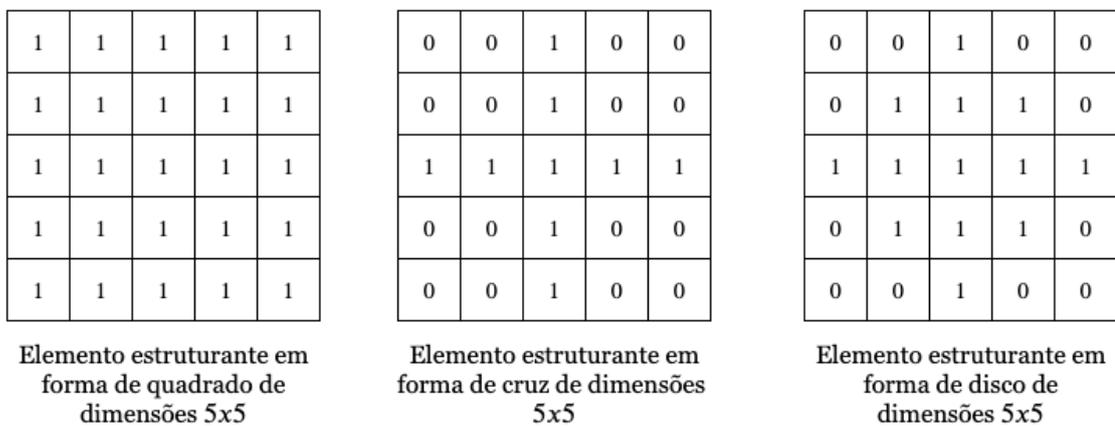


Figura 5: Exemplos de elementos estruturantes simples

Um modo intuitivo de compreender o funcionamento das operações morfológicas básicas são o conceito de encaixe e choque. É dito que o elemento estruturante encaixa com uma certa parte da imagem quando para cada pixel do elemento estruturante definido como um, o pixel correspondente na imagem também é um. Por outro lado, dizemos que o elemento estruturante choco com uma certa parte da imagem quando para todo pixel do elemento estruturante igual à um, existe pelo menos um pixel correspondente da imagem também igual à um.

Uma ilustração das definições de encaixe e choque são demonstradas na figura 6. Tendo a cor cinza como a atribuição do valor um e a cor branco como a atribuição do valor zero. O elemento estruturante resulta em encaixe apenas na posição *A* enquanto resulta em choque nas posições *A* e *B*. Na posição *C* não ocorre choque ou encaixe.

As operações morfológicas apresentam duas operações fundamentais: a erosão e dilatação. As demais operações morfológicas, tendo como exemplo a operação morfológica de abertura, são representadas como uma combinação dessas operações.

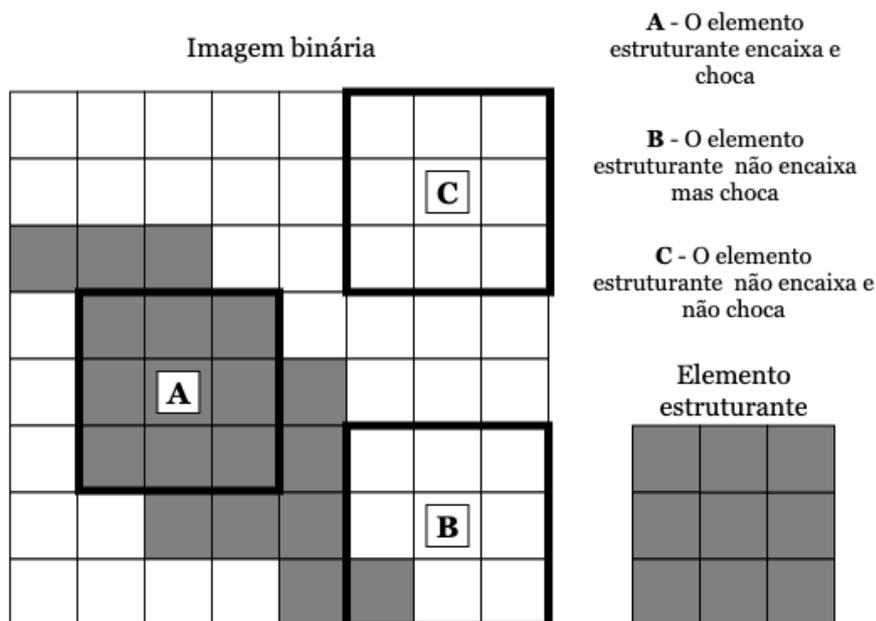


Figura 6: Ilustração das definições de choque e encaixe de um elemento estruturante sobre uma imagem binária. O quadrado de cor cinza representa a atribuição do valor um enquanto o quadrado de cor branco representa a atribuição do valor zero

2.6.1 Operação morfológica de erosão

A operação de erosão é representada pela equação 3 onde f é a imagem binária de entrada, s é o elemento estruturante e g é a imagem binária resultante. A operação de erosão faz com que para cada pixel $p = (x, y)$ da imagem de saída g seja atribuído o valor um se o posicionamento do elemento estruturante s sobre o ponto p da imagem f resulta em encaixe e o valor zero, caso contrário. Conforme [12], a erosão diminui ou afina os objetos em uma imagem binária tendo a erosão como uma operação de filtragem morfológica na qual os detalhes da imagem menores que o elemento estruturador são filtrados (removidos) da imagem.

$$g = f \ominus s \tag{3}$$

2.6.2 Operação morfológica de dilatação

A operação de dilatação é representada pela equação 4 onde f é a imagem binária de entrada, s é o elemento estruturante e g é a imagem binária resultante. A operação de dilatação faz com que para cada pixel $p = (x, y)$ da imagem de saída g seja atribuído o valor um se o posicionamento do elemento estruturante s sobre o ponto p da imagem f resulta em choque e o valor zero, caso contrário. Ainda segundo [12], ao contrário da erosão, que é uma operação de diminui ou afina, a dilatação “cresce” ou “engrossa”

objetos em uma imagem binária. A maneira e extensão específica desse espessamento são controladas pelo formato do elemento estruturador utilizado.

$$g = f \oplus s \quad (4)$$

2.6.3 Operação morfológica de abertura

A operação de abertura é representada pela equação 5 onde f é a imagem binária de entrada, s é o elemento estruturante e g é a imagem binária resultante. A operação morfológica de abertura consiste na aplicação da operação morfológica de erosão usando um elemento estruturante s seguida pela aplicação da operação de abertura utilizando o mesmo elemento estruturante. De acordo com [12], a operação morfológica de abertura geralmente suaviza o contorno de um objeto, quebra os istmos estreitos e elimina saliências finas.

$$g = f \circ s = (f \ominus s) \oplus s \quad (5)$$

2.7 Conceitos básicos sobre Rastreamento de Objetos

O rastreamento de objetos é a determinação do estado de um alvo, como posição, tamanho ou velocidade, em sucessivos quadros de vídeo [26]. O rastreamento de objetos tem sido uma das áreas de pesquisa mais importantes e ativas no campo da Visão Computacional [49]. O crescente interesse no rastreamento de objetos é motivado por um grande número de aplicações promissoras que agora podem ser abordadas por aplicativos em tempo real [40].

De um modo geral, o uso de rastreamento visual é pertinente nas tarefas de reconhecimento baseado em movimento, vigilância automatizada, indexação de vídeo, interação humano-computador e navegação veicular, etc [45].

Embora muito esforço tenha sido feito nas últimas décadas, continua sendo uma tarefa muito desafiadora construir um sistema de rastreamento robusto para lidar com problemas como oclusão, desfoque, movimento rápido, deformação, variação de iluminação, e rotação, etc [17]. Além disto não existe uma única abordagem que lide com sucesso todos os cenários [49].

As abordagens para rastreamento diferem principalmente umas das outras com base na maneira como abordam as seguintes questões: Qual representação do objeto é mais adequada para o rastreamento? Quais os recursos de imagem devem ser usados? Como o movimento, a aparência e a forma do objeto devem ser modelados? [47]. As

respostas a essas perguntas dependem muito do contexto/ambiente específico da tarefa de rastreamento e das informações de rastreamento disponíveis para os usuários [27]. Foi proposto um grande número de métodos de rastreamento que tentam responder a essas perguntas em diferentes cenários [47].

Em razão dos inúmeros cenários e tarefas nos quais se faz necessário o uso de um rastreador, diferentes autores buscaram categorizar as múltiplas abordagens de rastreamento [4, 27, 45, 47], entretanto não existe um consenso em relação a classificação dos rastreadores e a nomenclatura dos componentes. Por este motivo, na descrição dos componentes essenciais para que ocorra o rastreamento, diferentes trabalhos podem usar nomenclaturas distintas para componentes semelhantes ou incluir componentes que podem ou não ser empregados dependendo do contexto da aplicação.

Para uma melhor compreensão do funcionamento de um rastreador, neste trabalho é tido como componentes de um sistema de rastreamento os descrito por [7] que define, de um modo geral, dois componentes fundamentais: um modelo de movimento (ou chamado de modelo dinâmico) e um modelo de aparência (ou chamado de modelo de observação).

Um fluxograma genérico é descrito na figura 7 onde os componentes presentes no rastreamento são representados pelos quatro retângulos sendo eles o modelo de movimento, a representação do objeto, a estratégia de busca e o mecanismo de decisão. As informações necessárias para que os componentes mencionados realizem seu trabalho e as informações geradas por cada componente são representadas pelos paralelogramos.

2.7.1 Modelo de movimento

Modelos de movimento são empregados para prever a localização do objeto em um novo quadro dentro de uma sequência de vídeo com base nas características do seu histórico de movimento. Isso pode melhorar a estabilização de rastreamento e fazer com que o rastreamento sobreviva a algumas oclusões se as trajetórias dos objetos forem corretamente previstas [14]. São exemplos de modelos de movimentos o filtro de partículas [21] e o filtro de Kalman [22].

2.7.2 Modelo de aparência

Um modelo de aparência ou observação é uma representação interna de como o objeto de interesse se parece ou como ele se diferencia do ambiente [5], este modelo gera como saída a confiança que se tem de que um determinado candidato é o alvo [48]. Os modelos de aparência (e, portanto, os rastreadores visuais que usam esses modelos) podem ser categorizados com base nos recursos visuais que eles usam para descrever o objeto [5]. Selecionar as características certas desempenha um papel crítico no rastreamento. Em

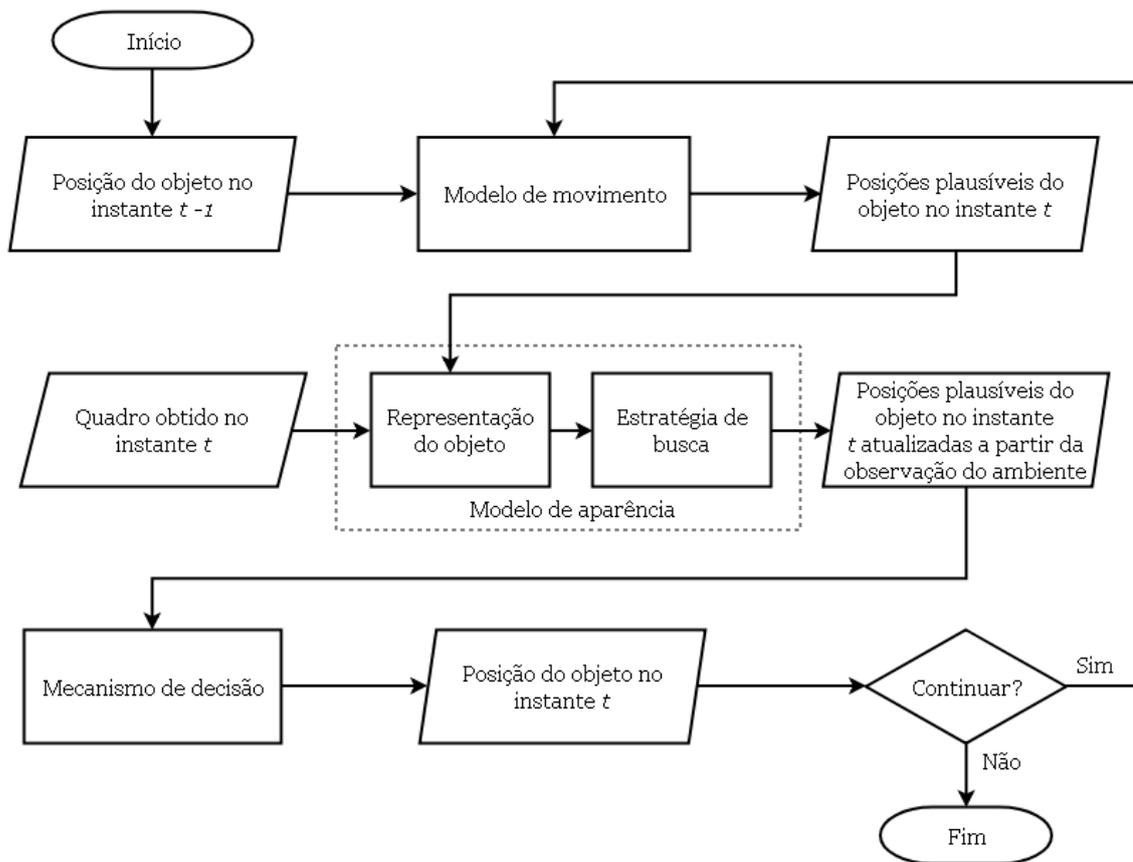


Figura 7: Fluxograma genérico de um rastreador de objeto

geral, a propriedade mais desejável de uma característica visual é sua exclusividade, para que os objetos possam ser facilmente distinguidos no espaço de características [47]. A principal dificuldade é que não é fácil projetar um bom método de modelagem de aparência que também seja robusto as mudanças de aparência como baixo contraste, alterações de iluminação e alterações de escala [28].

Os modelos de aparência são divididos em duas partes: a representação do objeto e a estratégia de busca.

As diferentes representações do objeto são classificadas entre representações globais ou representações locais sendo diferenciadas no níveis de detalhes que são utilizado para modelar a aparência do objeto [5]. Uma representação visual global reflete as características estatísticas globais da aparência do objeto [27]. Em uma representação visual local ou em partes, o modelo de aparência é decomposto em elementos separados, cada elemento descrevendo uma parte do objeto com seu próprio modelo de aparência (local) [5].

Em geral, as representações visuais globais são simples e computacionalmente eficientes para o rastreamento rápido de objetos. Devido às restrições geométricas globais

impostas, as representações visuais globais são suscetíveis a mudanças de aparência globais [27]. A principal vantagem dos modelos de aparência baseados em partes é que eles podem se adaptar a deformações geométricas, o que é especialmente útil ao rastrear objetos não rígidos e articulados [5]. Entretanto é utilizado um grande número de características locais, resultando em um custo computacional muito alto [27].

Em relação as estratégias de busca, um rastreador pode utilizar métodos generativos, discriminativos ou híbridos, junção de uma abordagem generativa com discriminativa, como forma de estratégia de busca. Segundo [38], em relação aos métodos generativos, o problema de rastreamento é formulado como uma busca pelas regiões de imagem mais semelhantes ao modelo do alvo. Apenas as informações do alvo são usadas.

Ao contrário dos métodos generativos, os métodos discriminativos consideram o rastreamento de objetos como um problema de classificação binária para distinguir o alvo do plano de fundo [7]. Geralmente é treinado um classificador para separar o alvo do fundo [48].

Estudos recentes de *benchmarking* mostram que os rastreadores de alto desempenho são geralmente rastreadores discriminativos ou híbridos, principalmente porque os rastreadores puramente generativos não conseguem lidar bem com plano de fundo complicados [48].

2.7.3 Mecanismo de decisão

Em alguns casos, o modelo de aparência pode não retorna a posição exata do objeto em um determinado instante. Isso pode ocorrer quando não é possível determinar de forma objetiva a posição do objeto de interesse. Os mecanismos de decisão servem para estimar a posição do objeto baseando nas posições mais plausíveis geradas pelo modelo de aparência.

2.8 Rastreamento de objetos em Robótica

Estimar a posição de objetos de interesse é um subproblema importante em muitas aplicações robóticas do mundo real. O desafio, entretanto, é lidar com as limitações dos sensores usados para a detecção dos alvos [31].

Na maioria dos casos, na prática, vários sensores *wireless* estáticos são empregados para melhorar a precisão do rastreamento e aumentar o tamanho da área de vigilância. Ao contrário dos sensores estáticos, cuja densidade e alcance de detecção são fixos, os sensores móveis (robôs) podem cobrir áreas maiores ao longo do tempo sem a necessidade de aumentar seu número [50]. Nesse caso, a tarefa se torna a de mover o sensor para que o alvo caia dentro de seu campo de visão. O uso de vários robôs pode aumentar o campo

de visão efetivo do sensor e, portanto, deve, em teoria, permitir um desempenho superior [31].

Com relação a um único robô, equipes de robôs são obviamente menos propensas a falhas, capazes de operar simultaneamente em áreas maiores. A troca de informações entre robôs permite que haja a fusão de dados. Em aplicações de robôs móveis, a fusão refere-se a qualquer estágio do processo de integração em que uma combinação real de diferentes fontes de informações ocorre [16]. A principal vantagem da fusão de dados consiste na melhoria da autenticidade ou disponibilidade de dados [23].

Contudo aplicações que utilizam a fusão de dados devem tomar algumas precauções em relação a problemas que podem surgir a partir da combinação dos dados. Entre esses problemas, temos os dados imperfeitos. Segundo [23], os dados imperfeitos surgem pois os dados fornecidos pelos sensores são sempre afetados por algum nível de imprecisão, bem como incerteza nas medições. Os algoritmos de fusão de dados devem ser capazes de expressar essas imperfeições de maneira eficaz e explorar a redundância de dados para reduzir seus efeitos.

A incerteza surge se o robô não possui informações críticas para realizar sua tarefa. A capacidade de lidar com a incerteza é fundamental para construir robôs de sucesso [42]. Modelos probabilísticos tornaram-se populares devido à sua robustez na presença de incerteza, o que os torna capazes de lidar com grandes quantidades de ruído dos sensores e oclusão [31]. Um dos modelos probabilísticos que é amplamente utilizado para a realização de rastreamento visual dentro da Robótica é o Filtro de Partículas.

2.9 Rastreamento por Filtro de Partículas

Um filtro de partículas é uma implementação não paramétrica do filtro de Bayes e é frequentemente usado para estimar o estado de um sistema dinâmico [39]. Os filtros de partículas funcionam muito bem com problemas de estimação não-lineares e não-gaussianos, provando sua eficiência em lidar com várias dificuldades relacionadas ao rastreamento de objetos [43].

Os filtros de partículas tornaram-se ferramentas populares para resolver o problema de rastreamento. Sua popularidade deriva de sua simplicidade, flexibilidade, facilidade de implementação e sucesso de modelagem em uma ampla gama de aplicações desafiadoras [34] e tem sido extensivamente utilizados nos últimos anos [1, 25, 46, 43].

Para utilizar o filtro de partículas como o rastreador visual no espaço $3D$ algumas suposições devem ser estabelecidas:

- A posição do objeto de interesse $P = (x, y, z)$ no instante de tempo t é tido como o estado x_t . Um estado é visto como uma coleção de informações que podem impactar

o futuro e evolui com o tempo;

- Uma observação z_t representa uma mensuração de um estado no instante t . Observações são obtidas por meio de sensores e fornece informação momentânea do estado;
- Uma ação de controle u_t é descreve uma ação realizada no instante de tempo t sobre o estado x_{t-1} para leva-lo ao próximo estado x_t ;
- A evolução de um estado é dado por um processo de Markov oculto que leva as seguintes suposições: em qualquer instante de tempo t não é possível determinar o estado atual x_t entretanto pode ser estimado, o próximo estado é determinado apenas pelo estado atual, qualquer estado x não é observável tendo sua observação z descreve uma função de probabilidade do estado x e que os estados x , as observações z e as ações de controle u sejam discretas.

A evolução de um estado dentro de um sistema dinâmico é descrito pela figura 8 tendo que a transição de um estado atual x_t para o estado posterior x_{t+1} é obtido a partir da ação de controle u_t realizada sobre o estado x_t e a observação y_t é gerada a partir da estimativa do estado x_t .

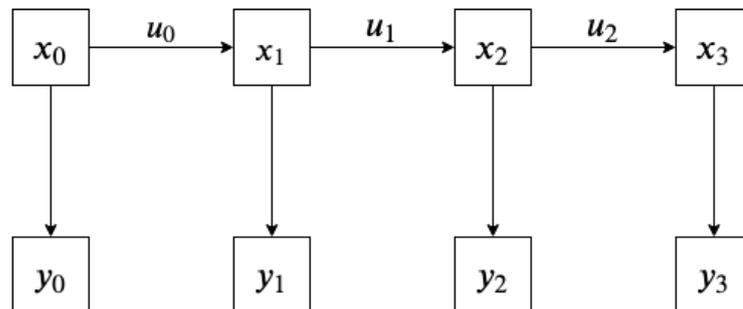


Figura 8: Representação da evolução de um estado x dentro de um sistema dinâmico

O filtro de partículas é, portanto, uma técnica de aproximação para o cálculo de posteriores assumindo que um estado x evolui ao longo de um tempo discreto em um sistema dinâmico estocástico. A ideia é representar o estado posterior por meio de um conjunto de hipóteses X_t onde cada hipótese é representada por uma partícula. Uma partícula x_t consiste em uma instanciação concreta do estado x no instante de tempo t juntamente com o peso w_i que determina o qual próximo do estado verdade é a instância. Esse filtro é apresentado conforme o algoritmo apresentado na tabela 1.

Algoritmo filtro_de_partículas (X_{t-1}, u_t, z_t)

- 1: $\bar{X}_t = X_t = \emptyset$
 - 2: **para** $m = 1$ **até** M **faça**
 - 3: Amostre $x_t^{[m]} \sim p(x_t | u_t, x_{t-1}^{[m]})$
 - 4: $w_t^{[m]} = p(z_t | x_t^{[m]})$
 - 5: $\bar{X}_t = \bar{X}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$
 - 6: **fim para**
 - 7: **para** $m = 1$ **até** M **faça**
 - 8: Extraia i com probabilidade $\propto w_t^{[i]}$
 - 9: Adicione $x_t^{[i]}$ à X_t
 - 10: **fim para**
 - 11:
 - 12: **retorne** X_t
-

Tabela 1: Algoritmo do filtro de partículas [42]

3 METODOLOGIA

3.1 Métodos e equipamentos

O robô utilizado durante todo o trabalho foi o Turtlebot 2, mostrado na figura 9, que consiste em um robô terrestre de código aberto que pode ser adquirido como um conjunto completo ou em partes. Segundo [9], alguns dos principais itens que compõem o modelo Turtlebot 2 são a base móvel que também serve como apoio para as partes superiores do robô, um notebook e o sensor de visão com câmera RGB e sensor de profundidade que consiste em um Microsoft Kinect versão 1.0.



Figura 9: Turtlebot 2 [9]

O Microsoft Kinect é um dispositivo lançado pela Microsoft em outubro de 2010. Ele contém um par de câmeras de profundidade e cores que pode obter a profundidade e a imagem RGB simultaneamente [13]. O baixo custo, a confiabilidade e a velocidade da medição prometem tornar o Kinect os principais dispositivos de medição 3D em robótica interna, reconstrução de cena 3D e reconhecimento de objetos [37]. Mais informações acerca das configurações e limitações do Kinect podem ser encontradas em [37, 33, 15].

Para a realização das operações referentes ao PADI foi utilizada a biblioteca OpenCV¹. Segundo [3], a OpenCV é uma biblioteca de visão computacional de código aberto e foi projetado para eficiência computacional e com um forte foco em aplicativos em tempo real.

Para a obtenção das informações vindas do Turtlebot foi utilizado o *framework* Robot Operating System (ROS)². O ROS não é um sistema operacional no sentido tradicional de gerenciamento e agendamento de processos; em vez disso, fornece uma camada de comunicação estruturada acima dos sistemas operacionais hospedeiros de um *cluster* de

¹opencv.org

²ros.org

computação heterogênea [35]. O ROS é um *framework* amplamente utilizado em robótica. A filosofia é criar um software que funcione em outros robôs, fazendo pequenas alterações no código [29].

3.2 Configuração do sistema

A configuração do sistema que implementa as duas abordagens é descrita por meio do fluxograma da figura 10 onde cada quadrado representa um procedimento realizado dentro do próprio sistema. Cada procedimento ou operação serão discutidos nas próximas seções. O sistema foi implementado utilizando a linguagem de programação Python³.

3.3 Definição do objeto de interesse

Como objeto de interesse foi utilizado um objeto esférico com sua superfície composta quase toda por uma única cor. Este objeto foi selecionado pois o processamento da imagem com o intuito de detectar o objeto tende a ter um custo computacional menor devido a suas características simplórias. Contudo a implementação não fica presa ao objeto de interesse que foi selecionado, tanto é que o objeto de interesse pode ser substituído apenas alterando a etapa de detecção de acordo com as características do novo objeto e realizando pouca ou nenhuma alteração sobre o resto da implementação.

A representação do objeto de interesse consiste em um ponto no espaço 3D que é o centro de massa do objeto esférico. Em relação ao modelo de aparência foram utilizadas as representações visuais globais de cor e contorno do objeto esférico. Como estratégia de busca foram utilizados métodos generativos, ou seja, buscou-se detectar o objeto de interesse apenas determinando o grau de semelhança entre ele e os candidatos encontrados no ambiente.

As configurações do rastreador visual foram as mais simples possíveis para que apresentasse um baixo custo computacional devido ao fato de que a fusão dos dados traria a implementação um aumento de custo computacional e esse aumento não poderia exceder as limitações impostas pelos equipamentos utilizados.

3.4 Detecção do objeto de interesse

Na etapa de detecção do objeto de interesse é tido como entrada a imagem vinda do Kinect sobre a qual deve ser retirada a posição do objeto no sistema de coordenadas de tela caso o mesmo se encontre dentro da imagem. Para tal fim são utilizadas diferentes operações, a figura 11 contém a sequência das operações realizadas sobre imagem com

³python.org

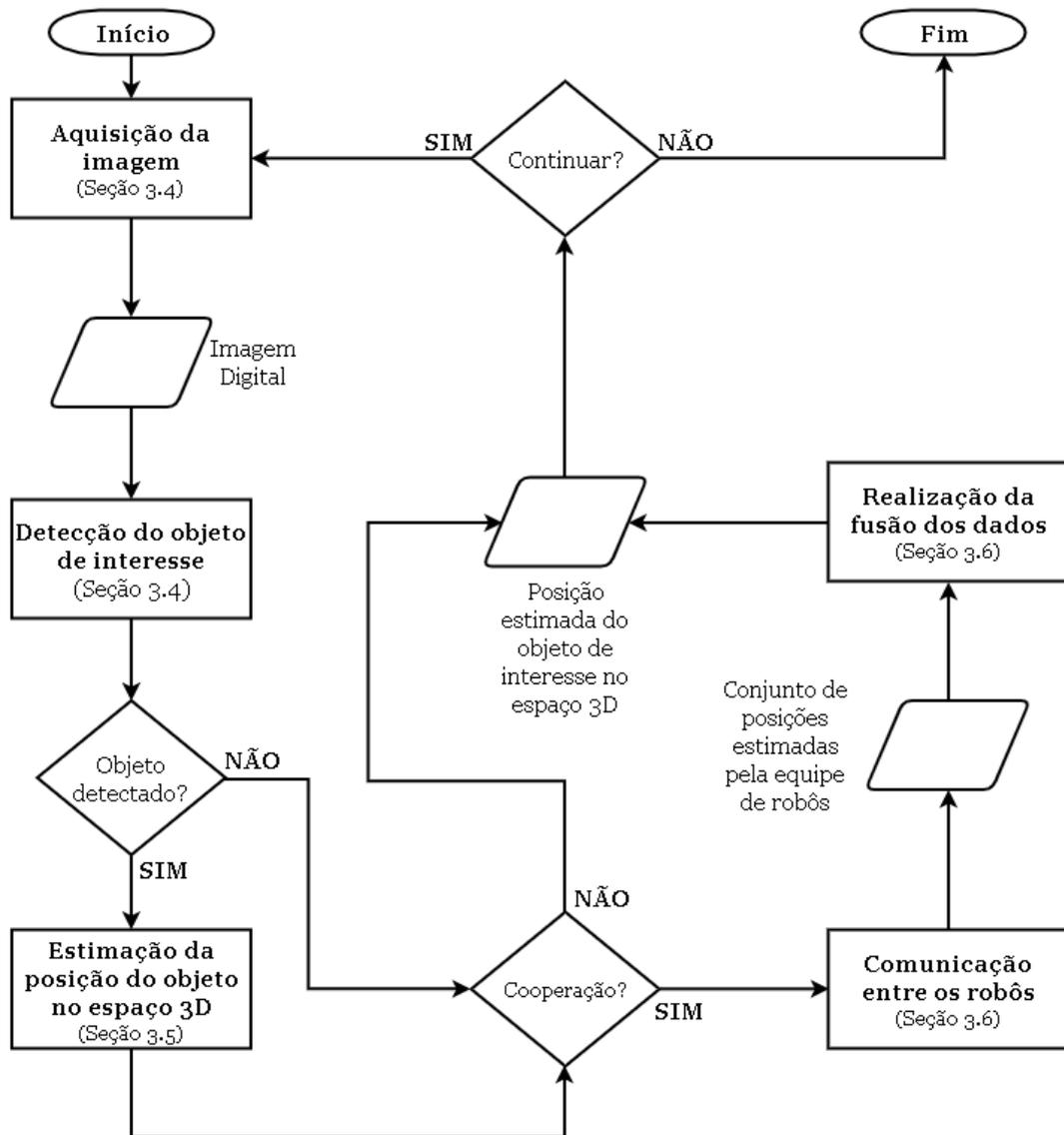


Figura 10: Fluxograma do sistema de rastreamento visual no espaço 3D

o objetivo de detectar o objeto de interesse juntamente com qual etapa da sequência de PADI cada uma das operações pertence.

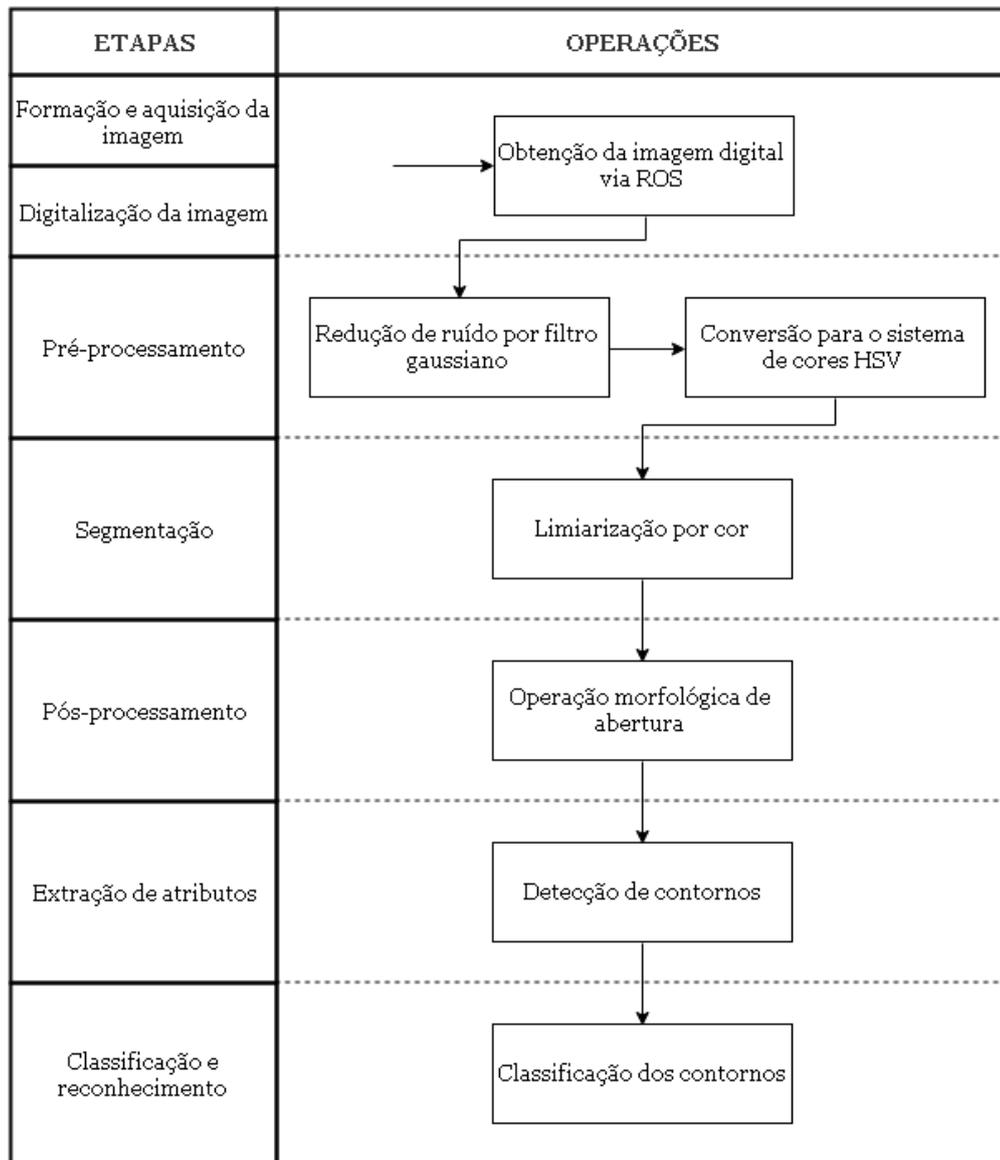


Figura 11: Sequência das operações realizadas sobre a imagem e suas respectivas classificações referente a sequência de etapas do PADI

Como passo inicial, as etapas de formação e aquisição da imagem juntamente com a digitalização é realizado por meio do ROS. Toda imagem gerada pelo sensor é encaminhada para o ROS que por sua vez repassa ao sistema responsável por rastrear objeto de interesse.

Antes de qualquer operação ser executada sobre a imagem, é realizada uma operação de suavização por filtro de gaussiana com o intuito de reduzir o ruído presente na imagem sendo utilizado uma máscara de dimensões 5×5 e $\sigma = 1.1$. Em seguida, a imagem é convertida do espaço de cor RGB para o HSV com o objetivo de facilitar a passagem da

cor percebida do objeto de interesse pelo usuário para o sistema.

Como o objeto de interesse possui uma cor distinta foi utilizado essa informação para diferencia-lo dos demais objetos presentes na cena. Para isso é realizado uma operação de limiarização que é descrita pela equação 6 onde a variável I representa a imagem de entrada, as variáveis $I(x, y)_H$ e $I(x, y)_S$ representam, respectivamente, o valor dos componentes de matiz e saturação do pixel de coordenadas (x, y) da imagem de entrada I e a variável I_{seg} representa a imagem de saída do processo de limiarização. Esta operação recebe a imagem no espaço de cor HSV e retorna uma imagem binária onde cada pixel da imagem de entrada que estiver dentro do limiar gera a atribuição do valor máximo para o pixel da imagem de saída de mesma posição ou a atribuição do valor mínimo caso contrário. Os valores utilizados na limiarização foram obtidos empiricamente.

$$I_{seg}(x, y) = \begin{cases} 1, & \text{se } 65 \leq I(x, y)_H \leq 30 \text{ e } I(x, y)_S \geq 50 \\ 0, & \text{caso contrário.} \end{cases} \quad (6)$$

Como operação da etapa de pós-processamento, é realizada a operação morfológica de abertura como propósito de remover conjuntos de pixels de dimensões inferiores ao elemento estruturante já que se espera que o objeto de interesse sempre possua um tamanho mínimo fixo. Foi utilizado o elemento estruturante descrito na equação 7 que possui o formato de disco.

$$B = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \quad (7)$$

Em seguida, é realizada uma operação para encontrar os contornos dentro da imagem binária. O algoritmo utilizado para tal fim foi o descrito por [41]. Esta operação tem como saída os contornos encontrados na imagem de entrada separados em conjuntos de pixels conectados.

A operação seguinte consiste na classificação dos contornos onde cada contorno gerado pela operação anterior é tratado como um único objeto. Para cada contorno, é encontrado o menor círculo possível que enclausura todos os seus pixels. A partir do raio e a área deste círculo, juntamente com a área delimitada pelo contorno, é obtido um valor que determina o quão próximo o contorno se assemelha ao objeto de interesse, este valor

foi denominado como confiança de observação α e é calculado pela equação 8 onde A_{obs} representa a área observada, que consiste na área formada pela região que é englobada pelo contorno, e A_{esp} representa a área esperada, que é a área do círculo que enclausura todos os pontos do contorno.

$$\alpha = \frac{A_{obs}}{A_{esp}} \quad (8)$$

O valor da área observada A_{obs} é obtido pela fórmula da área do teorema de Green. Já o valor da área esperada A_{esp} é obtido pela equação 9 onde a variável r_{enc} representa o raio do círculo que enclausura todos os pontos do contorno.

$$A_{esp} = \pi \cdot r_{enc}^2 \quad (9)$$

A lógica por trás dessa etapa surge do fato de que, como o objeto de interesse possui atributos que podem ser facilmente observados em imagem e mesmo após as operações anteriores esses atributos se mantêm apesar de sofrerem deformações, podemos comparar atributos extraídos da imagem com os do objeto esférico com o intuito de determinar o quão próximo são uns dos outros. No cálculo descrito anteriormente, foi selecionada a área do objeto e como ela se encontra distribuída na imagem como um atributo de classificação.

Após o cálculo da confiança de observação α de todos os contornos encontrados na imagem, os contornos que tiverem o valor de confiança de observação α inferior à 0.5 são descartados e, dos contornos remanescentes, o contorno com maior valor de confiança de observação α é selecionado como o contorno do objeto de interesse. Como parte final do processo de detecção, a posição do objeto de interesse no sistema de coordenadas da tela $P_t = (x_t, y_t)$ é tida como o centroide do menor círculo possível que engloba todos os pontos do contorno selecionado.

3.5 Estimação da posição do objeto

Nesta etapa, se tem a posição do objeto de interesse no espaço $2D$ no sistema de coordenadas da tela em pixels e é necessária sua conversão para a posição do objeto no espaço $3D$ no sistema de coordenadas global em metros. Esta conversão não é realizada de forma direta sendo obrigatório algumas etapas até chegar ao resultado desejado.

A etapa inicial consiste na conversão do sistema de coordenadas da tela para o sistema de coordenadas da câmera. Esta operação leva a posição do objeto do espaço $2D$ para o espaço $3D$ utilizando as equações 10, 11 e 12, que foram derivadas da equação 1, onde x_c , y_c e z_c representam a posição do objeto no sistema de coordenadas da câmera no

espaço 3D, x_t e y_t representam a posição do objeto no sistema de coordenadas da tela, z representa a distância do objeto em relação a câmera.

$$x_c = \frac{x_t - c_x}{f} \cdot z \quad (10)$$

$$y_c = \frac{y_t - c_y}{f} \cdot z \quad (11)$$

$$z_c = z \quad (12)$$

As variáveis da coordenada do ponto principal do plano da imagem (c_x, c_y) e a distância focal f são os mesmos parâmetros intrínsecos descritos na seção 2.2. As câmeras fixadas nos robôs foram calibradas utilizando o procedimento para calibração de câmeras monoculares do ROS. Após esta transformação, os valores passam a ser representados em metros.

A distância do objeto em relação a câmera z é necessária para a transformação do espaço 2D para o 3D e é comumente obtida por meio de um sensor de profundidade. O valor desta variável pode ser obtido utilizando o sensor de profundidade do próprio Kinect. Contudo, durante a realização dos experimentos, notou-se que o sensor de profundidade do Kinect teve dificuldades em determinar distância do objeto de interesse da câmera tornando inviável a sua utilização.

A abordagem selecionada para contornar este problema foi utilizar informações do objeto obtidas no sistema de coordenadas da tela e o conhecimento prévio que temos sobre o mesmo para determinar sua distância no sistema de coordenadas global. Um exemplo de informação é o raio observado no sistema de coordenadas da tela cujo o tamanho está diretamente relacionado com a distância do objeto em relação a câmera. O cálculo da distância do objeto de interesse em relação a câmera é exposto na figura 12 onde d_x e d_z representam, respectivamente, a distância do objeto em relação a câmera sobre o eixo X e Z . Não é levado em consideração a distância sobre o eixo Y pois o objeto de interesse e a câmera se encontra em uma mesma altura logo distância no eixo é zero.

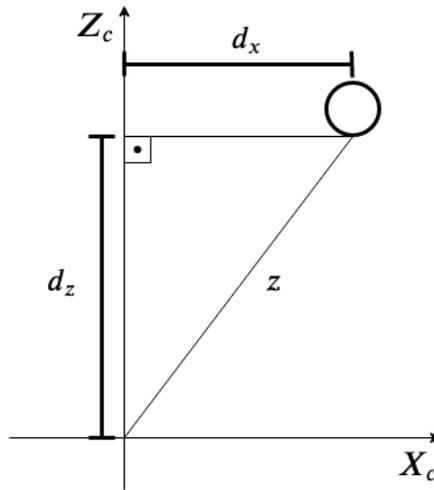


Figura 12: Relação entre as distâncias entre o eixo X e Z com a distância do objeto em relação a câmera z

Para calcular a distância do objeto em relação a câmera foi selecionado informações do objeto que proporciona a distância em relação ao eixo X e Z . Para determinar a distância em relação ao eixo Z foi utilizado o raio formado pelo objeto no sistema de coordenadas da tela enquanto que para a distância em relação ao eixo X foi utilizado a distância do centroide do objeto em relação à coordenada principal da câmera c_x . A partir da escolha das informações foram obtidas funções de interpolação que relacionam as informações obtidas no sistema de coordenadas da tela com a distância do objeto em relação a câmera. Em aplicações matemáticas, as funções de interpolação são utilizadas para fazer previsões de valores de funções dentro de certo intervalo.

Para obter a função de interpolação que relaciona o raio observado com a distância sobre o eixo Z , o objeto de interesse foi posicionado em diferentes distâncias em relação a câmera sobre o eixo Z . Para cada posição foi computado o valor médio do raio em 300 detecções. Com as mensurações foi gerado a função de interpolação descrita pela equação 13.

$$f(x) = 2.8983e^{-6}x^4 - 0.00045117x^3 + 0.026504x^2 - 0.73325x + 9.568 \quad (13)$$

Já para obter a função de interpolação que relaciona a distância do centroide do objeto em relação à coordenada principal da câmera c_x em pixels com a distância do objeto em relação ao eixo X em metros, o objeto de interesse foi posicionado a 2.0 metros de distância em relação ao eixo Z da câmera e foi variando apenas a sua distância sobre o eixo X . Para cada posição foi computado o valor médio da distância entre o valor da coordenada x do centroide do objeto de interesse e a coordenada principal da câmera c_x

em 300 detecções. Com as mensurações foi gerado a função de interpolação descrita pela equação 14.

$$g(x) = 0.0039834x + 0.028457 \quad (14)$$

É válido salientar que este tipo de abordagem faz com que o cálculo da distância se torna totalmente dependente das informações que são obtidas pela detecção e que seja apenas utilizável sobre o mesmo objeto de interesse. Por fim, a distância do objeto em relação a câmera z pode ser calculado como a raiz quadrada da soma do quadrado da distância do objeto em relação a câmera sobre o eixo Z com quadrado da distância do objeto em relação a câmera sobre o eixo X .

Para comparar as distâncias obtidas pelo sensor do Kinect com as distâncias obtidas por meio das funções de interpolação, o objeto de interesse foi fixado encima em um robô e o mesmo realizou a trajetória de um quadrado enquanto um outro robô foi utilizado para realizar o rastreamento visual do objeto. Durante o experimento, após a detecção do objeto de interesse, foi utilizado as funções de interpolação e o sensor de profundidade do Kinect para obter a distância do objeto em relação a câmera. Para verificar ambas abordagens foi utilizado a odometria do robô responsável por movimentar o objeto para determinar a distância do objeto em relação a câmera, a odometria nada mais é que a mudança relativa de posição de acordo com os codificadores das rodas do robô. A figura 13 representa as distâncias mensuradas do objeto de interesse durante o experimento utilizando a odometria, o sensor de profundidade do Kinect e das funções de interpolação.

Pode-se notar que quando utilizado as informações do sensor de profundidade do Kinect não foi possível determinar a distância do objeto mesmo ocorrendo a detecção, fato que não ocorre quando utilizado as funções de interpolação. Outro ponto positivo do uso das funções de interpolação foi a redução da instabilidade das mensurações geradas pelo sensor de profundidade do Kinect que ocorre quando as distâncias mensuradas não são próximas dos valores reais.

Após sair do sistema de coordenadas da tela para o sistema de coordenadas da câmera é necessário aplicar uma rotação para conciliar o sistema de coordenadas da tela com o sistema de coordenadas da câmera. Como demonstrado na figura 14, os sentidos dos eixos x e y são diferentes. Por este motivo, é aplicado uma rotação de 180° no sentido horário em torno do eixo z , como mostra a equação 15, para sanar este problema.

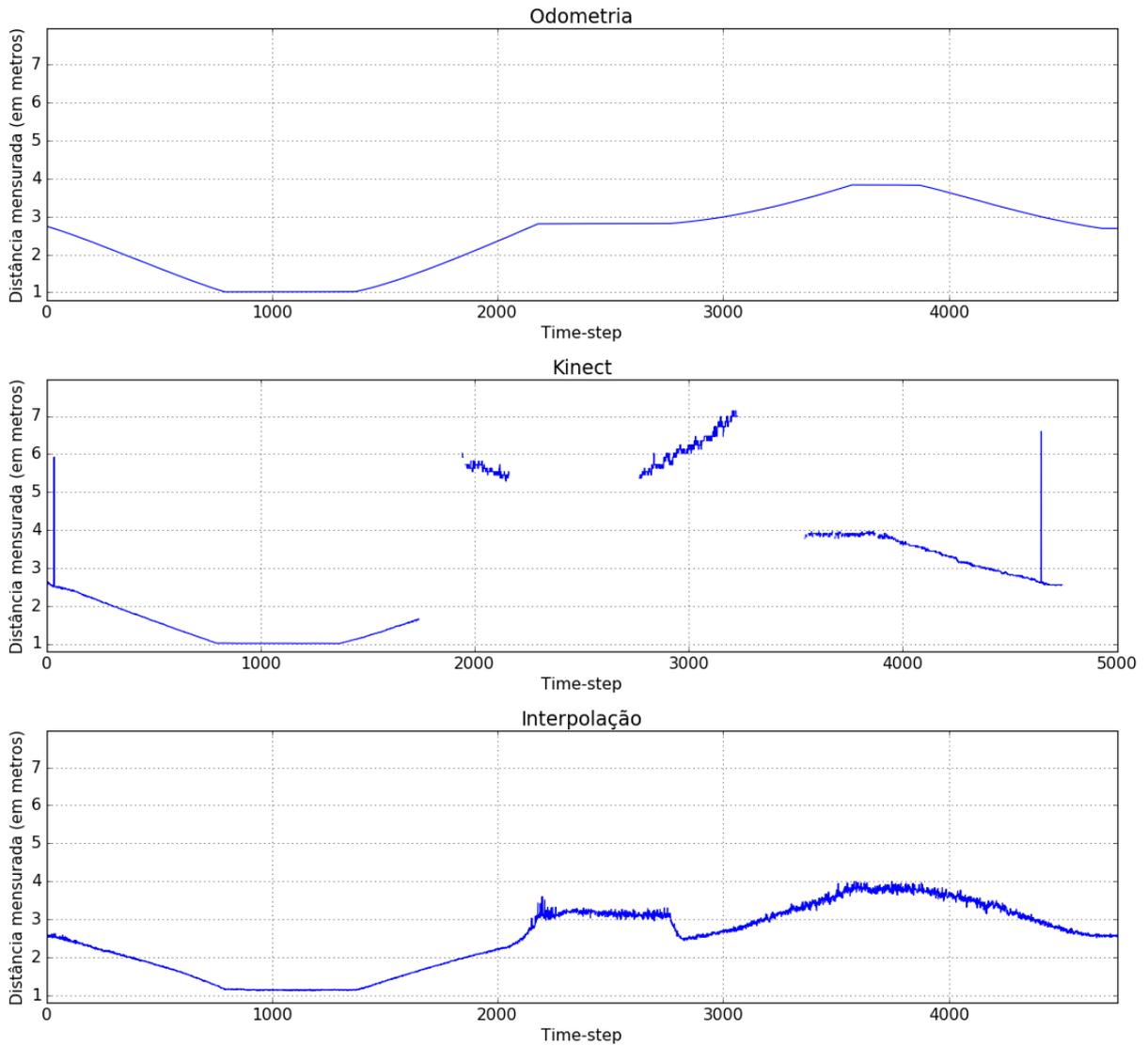


Figura 13: Distância mensuradas por meio da odometria, sensor de profundidade do Kinect e funções de interpolação durante o experimento

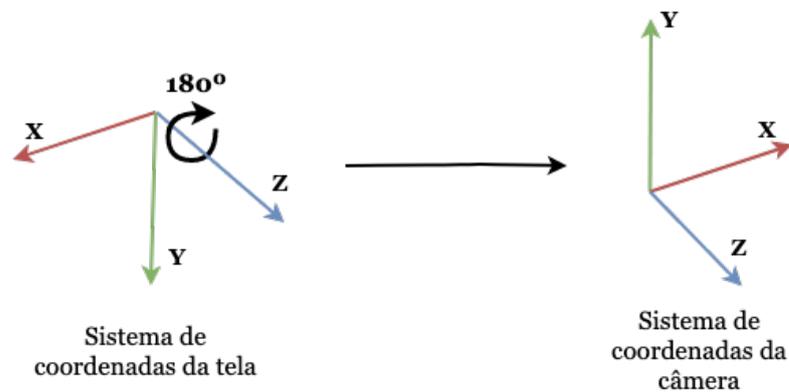


Figura 14: Rotações realizadas para ir do sistema de coordenadas da tela para a da câmera

$$\begin{bmatrix} x'_c \\ y'_c \\ z'_c \\ 1 \end{bmatrix} = \begin{bmatrix} \cos 180^\circ & -\text{sen } 180^\circ & 0 & 0 \\ \text{sen } 180^\circ & \cos 180^\circ & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \quad (15)$$

O próximo passo consiste na passagem da posição do objeto do sistema de coordenadas da câmera para o sistema de coordenadas do robô. Essa operação é necessária pois, quando falamos de câmeras fixadas em robôs, o posicionamento da câmera em relação ao robô pode variar. Um exemplo é o próprio Turtlebot cujo a câmera pode mudar de posição de acordo com a necessidade do usuário. O posicionamento das câmeras em relação à base do Turtlebot é idêntico para todos os robôs utilizados, logo o mesmo cálculo para a mudança do sistema de coordenadas é aplicado para todos os robôs.

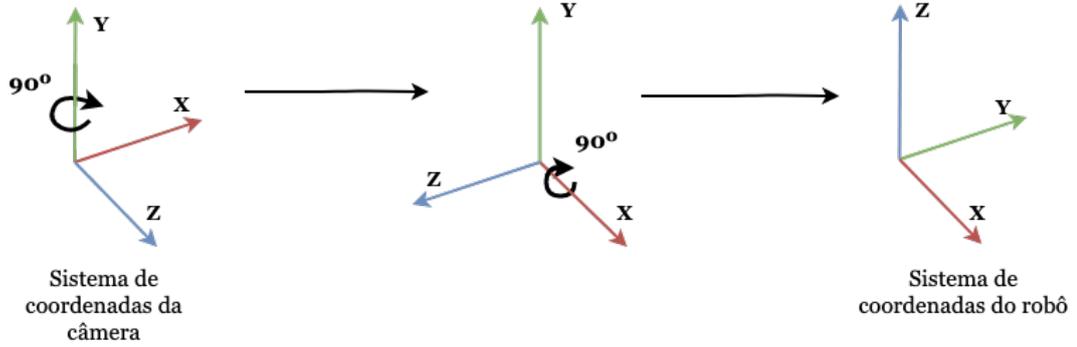


Figura 15: Rotações realizadas para ir do sistema de coordenadas da câmera para a do robô

A figura 15 apresenta os sentidos de cada um dos eixos do sistema de coordenadas da câmera e do robô. Esta operação é um pouco mais complexa pois exige a aplicação de duas rotações. A primeira rotação é uma rotação de 90° no sentido horário em torno do eixo y e a segunda rotação é uma rotação de 90° no sentido horário em torno do eixo x . Esta operação é descrita pela equação 16.

$$\begin{bmatrix} x_r \\ y_r \\ z_r \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 90^\circ & -\text{sen } 90^\circ & 0 \\ 0 & \text{sen } 90^\circ & \cos 90^\circ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 90^\circ & 0 & \text{sen } 90^\circ & 0 \\ 0 & 1 & 0 & 0 \\ -\text{sen } 90^\circ & 0 & \cos 90^\circ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x'_c \\ y'_c \\ z'_c \\ 1 \end{bmatrix} \quad (16)$$

Como a origem dos sistemas de coordenadas da câmera e do robô são diferentes, após as rotações, é aplicado uma translação para levar de uma origem à outra, a translação é descrita pela equação 17. Os valores das variáveis t_{cx} , t_{cy} e t_{cz} foram obtidos

manualmente e são, respectivamente, iguais a -0.125 , 0.0 e 0.25 metros.

$$\begin{bmatrix} x'_r \\ y'_r \\ z'_r \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_{cx} \\ 0 & 1 & 0 & t_{cy} \\ 0 & 0 & 1 & t_{cz} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ z_r \\ 1 \end{bmatrix} \quad (17)$$

Por último, a posição do objeto no sistema de coordenadas do robô (local) é convertida para o sistema de coordenadas global. Essa operação tem como objetivo que todos os robôs envolvidos no rastreamento do objeto de interesse compartilhem o mesmo sistema de coordenadas. Para tal fim, é utilizado a pose do robô. Esta operação é descrita pela equação 18 onde t_{rx} e t_{ry} representam, respectivamente, a posição x e y do robô no sistema de coordenadas do global e θ representa o ângulo de orientação do robô.

$$\begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\text{sen } \theta & 0 & t_{rx} \\ \text{sen } \theta & \cos \theta & 0 & t_{ry} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x'_r \\ y'_r \\ z'_r \\ 1 \end{bmatrix} \quad (18)$$

A implementação referente a abordagem não cooperativa termina nesta seção. A próxima seção consiste na fusão de dados que se encontra apenas na abordagem cooperativa.

3.6 Fusão dos dados

Nessa etapa ocorre a fusão dos dados coletados por todos os robôs de uma mesma equipe de rastreamento. Para tal finalidade, foi utilizado a implementação de um filtro de partículas. A sequência de passos que são realizados é descrita na figura 16, sendo elas a inicialização das partículas, a correção dos pesos, a reamostragem das partículas, a estimação do estado atual e a predição do próximo estado.

Antes de qualquer coisa, para que ocorra a fusão dos dados é necessário que exista um mecanismo de comunicação que possibilite a troca de informações entre os robôs envolvidos no rastreamento. Essa comunicação foi implementada utilizando o mecanismo de comunicação síncrona do ROS conhecido como serviço. Após o término da detecção e estimação da posição do objeto, o robô faz uma requisição contendo as informações de posição x_w , y_w e z_w do objeto de interesse, a confiança sobre a observação α e o identificador k . O trecho de código responsável por executar o serviço armazena os dados gerados pelo robô e retorna as mesmas informações sendo que dos demais robôs envolvidos no rastreamento como resposta.

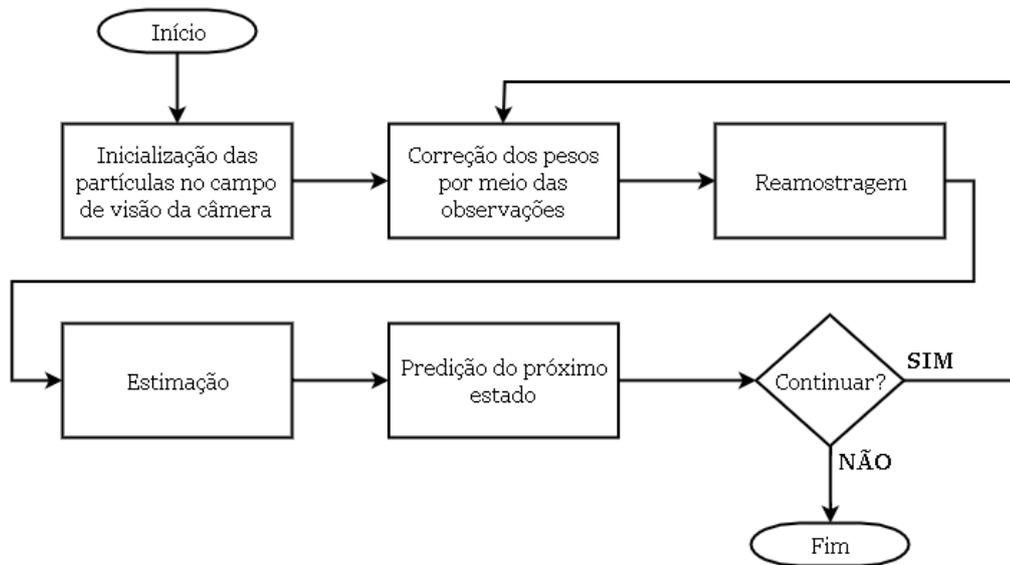


Figura 16: Fluxograma do filtro de partículas empregado para o rastreamento visual

Mesmo quando o robô não consegue obter informações sobre o objeto de interesse a partir dos dados coletados pelos seus sensores, o robô ainda realiza a requisição mas repassa o valor de confiança sobre a observação α igual à zero, fazendo com que os seus dados não afetem a estimação da posição do objeto de interesse durante a fusão dos dados. Ele efetua tal ação com o objetivo de descobrir se algum outro membro de sua equipe tem alguma informação referente ao objeto naquele instante de tempo.

Uma das vantagens da utilização de serviços é a simplicidade de utilização e a velocidade em que ocorre a troca das informações. Em contrapartida, as informações que ficam registradas dentro do serviço precisam ser atualizados com frequência para que estejam o mais próximo do estado atual do ambiente.

Antes entrar em detalhes em relação aos passos da fusão dos dados realizadas por meio do uso de filtro de partículas, algumas definições devem ser esclarecidas. Sendo elas:

- Uma partícula x_t descreve a posição $P_w = (x_w, y_w, z_w)$ do centro de massa do objeto de interesse no espaço $3D$ no instante de tempo t
- A observação β_r representa a posição do objeto de interesse inferida pelo robô r por meio da detecção
- O identificador k consiste em uma variável utilizada para distinguir cada um dos membros da equipe de rastreamento
- As variáveis M e N representam, respectivamente, a quantidade de partículas utilizadas dentro de filtro de partículas e a quantidade de robôs pertencentes a equipe

O primeiro passo do filtro de partículas consiste na inicialização das partículas e dos pesos associados a cada uma delas, o conjunto de partículas iniciais X_0 é gerado dentro do campo de visão da câmera levando em consideração a posição do robô no sistema de coordenadas global. Como demonstrado na figura 17, o campo de visão de uma câmera é determinado pelos ângulos de abertura horizontal a_h e vertical a_v e as distâncias máxima d_{max} e mínima d_{min} de captura. As distâncias máxima e mínima de captura constituem o intervalo de distância no qual é aconselhável utilizar os dados capturados pelo sensor e são geralmente estabelecidos levando em consideração as limitações do mesmo.

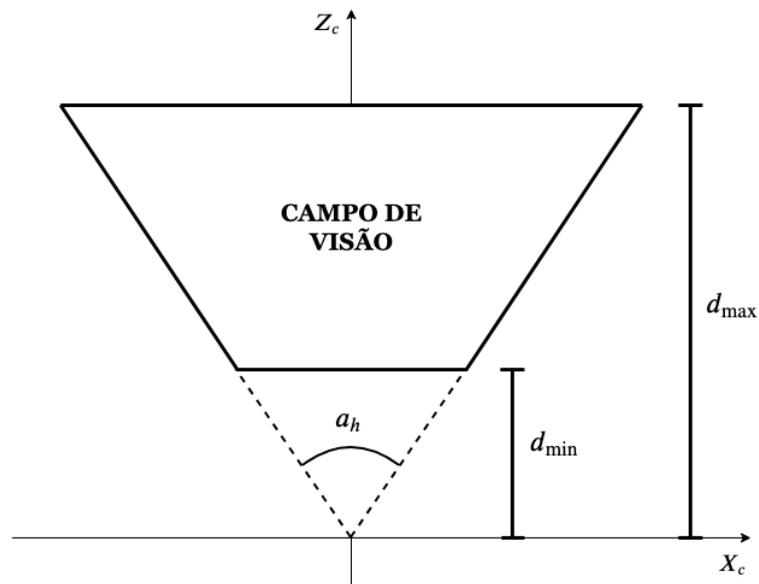


Figura 17: Configuração do campo de visão

Na figura 18, algumas informações sobre o campo de visão foram destacadas. A principal delas é de que, dado um ponto $P = (x, y, z)$ no espaço $3D$, o valor que a variável z assume delimita os possíveis valores que as variáveis x e y podem assumir que façam com que o ponto P ainda se encontre dentro do campo de visão. Com base nas relações trigonométricas, foram obtidas as equações 19 e 20 que foram usadas para gerar partículas no espaço da câmera.

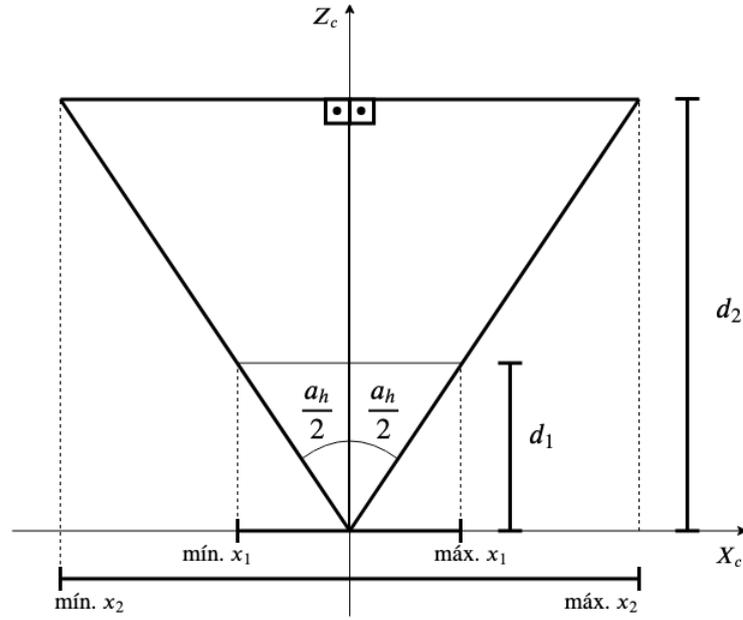


Figura 18: Informações destacadas sobre o campo de visão

$$\text{mín. } x_i = -\tan\left(\frac{a_h}{2}\right) \cdot z_i \quad (19)$$

$$\text{máx. } x_i = \tan\left(\frac{a_h}{2}\right) \cdot z_i \quad (20)$$

A partir dos ângulos de abertura da câmera e as distâncias máxima e mínima de captura é possível determinar se um ponto no espaço $3D$ se encontra dentro do campo de visão. Tendo V como o conjunto de todos os pontos que se encontram dentro do campo de visão de uma câmera e $P = (x, y, z)$ como um ponto qualquer pertencente ao espaço $3D$, o ponto P pertence ao conjunto V se as seguintes afirmações forem verdadeiras:

$$d_{min} \leq z \leq d_{max}$$

$$-\tan\left(\frac{a_h}{2}\right) \cdot z \leq x \leq \tan\left(\frac{a_h}{2}\right) \cdot z$$

$$-\tan\left(\frac{a_v}{2}\right) \cdot z \leq y \leq \tan\left(\frac{a_v}{2}\right) \cdot z$$

Utilizando as afirmações anteriores foi possível gerar partículas dentro do campo de visão da câmera. A tabela 2 descreve o algoritmo de inicialização das partículas. Após gerar as partículas dentro do sistema de coordenadas da câmera é necessário passar essas partículas para o sistema de coordenadas global. Isso pode ser realizado seguindo as equações descritas na seção 3.5. Os ângulos de abertura horizontal a_h e vertical a_v da câmera RGB do Kinect foram retirados de [37] e são, respectivamente, 63° e 50° . Como

distância mínima d_{min} e máxima d_{max} foram utilizados os valores de 1.0 e 4.0 metros, os valores das variáveis de translação da câmera t_{cx}, t_{cy} e t_{cz} e os valores das variáveis de translação do robô t_{rx}, t_{ry} e θ são os mesmos valores que foram descritos na seção 3.5.

Algoritmo fpc_gerar_partículas ($d_{min}, d_{max}, a_h, a_v, t_{cx}, t_{cy}, t_{cz}, t_{rx}, t_{ry}, \theta, M$)

```

1:  $X_0 = \emptyset$ 
2: para  $m = 1$  até  $M$  faça
3:   {Gerar as partículas no sistema de coordenadas da câmera}
4:    $cam_z = rand(d_{min}; d_{max})$ 
5:    $cam_x = rand(-1; 1) \cdot \tan \frac{a_h}{2} \cdot z$ 
6:    $cam_y = rand(-1; 1) \cdot \tan \frac{a_v}{2} \cdot z$ 
7:   {Passa as partículas para o sistema de coordenadas do robô (equação 16)}
8:    $robot_x = cam_z$ 
9:    $robot_y = cam_x$ 
10:   $robot_z = cam_y$ 
11:  {Passa as partículas para o sistema de coordenadas global (equação 18)}
12:   $world_x = \cos \theta \cdot robot_x - \sin \theta \cdot robot_y$ 
13:   $world_y = \cos \theta \cdot robot_x + \sin \theta \cdot robot_y$ 
14:   $world_z = robot_z$ 
15:  {Aplicar as translações}
16:   $world_x = world_x + t_{cx} + t_{rx}$ 
17:   $world_y = world_y + t_{cy} + t_{ry}$ 
18:   $world_z = world_z + t_{cz}$ 
19:   $x_0^{[m]} = (world_x, world_y, world_z)$ 
20:   $X_0 = X_0 + \langle x_0^{[m]}, \frac{1}{M} \rangle$ 
21: fim para
22:
23: retorne  $X_0$ 

```

Tabela 2: Algoritmo relacionado a etapa de inicialização das partículas do filtro de partículas

Após a inicialização, o peso de cada partícula é corrigido com base nas observações de cada um dos robôs. Diferentemente da concepção original do filtro de partículas, mais de uma observação é utilizada durante a processo de correção dos pesos onde cada uma das observações é responsável por uma parte desta correção. Essa parte é determinada pelo grau de confiança sobre a observação.

Como primeiro passo para a correção dos pesos, é realizado a normalização dos graus de confiança sobre a observação α para que cada um deles passe a representar a parte da correção que a observação β é responsável. O algoritmo de correção está detalhado na tabela 3.

Para cada observação β_k é gerada uma função de densidade de probabilidade M_k

Algoritmo fpc.correção($\alpha_k, \beta_k, x_w, w_w, z_w, X_t, M, N, k$)

```

1: {Requisição ao serviço}
2: Envie seu  $\alpha_k, \beta_k, x_w, w_w, z_w$  e  $k$ 
3: {Resposta do serviço}
4: Receba os  $\alpha, \beta, x_w, y_w, z_w$  dos demais robôs
5: {Passo de normalização das confianças}
6: para  $r = 1$  até  $N$  faça
7:    $\alpha_r = \frac{\alpha_r}{\sum_{n=1}^N \alpha_n}$ 
8: fim para
9: {Correção dos pesos das partículas a partir das observações}
10: para  $m = 1$  até  $M$  faça
11:   para  $r = 1$  até  $N$  faça
12:     Obter a probabilidade  $p_r$  de  $x_t^{[m]}$  a partir da FDP  $M_r$  gerada pela observação  $\beta_r$ 
13:   fim para
14:    $w_t^{[m]} = w_t^{[m]} \cdot \sum_{n=1}^N p_n \alpha_n$ 
15: fim para
16: {Normalização dos pesos associados as partículas}
17: para  $m = 1$  até  $M$  faça
18:    $w_t^{[m]} = \frac{w_t^{[m]}}{\sum_{n=1}^M w_t^{[n]}}$ 
19: fim para
20:
21: retorne  $X_t$ 

```

Tabela 3: Algoritmo relacionado a etapa de correção dos pesos

com distribuição normal de média μ igual a posição estimada pela observação β_k e covariância de distribuição σ como descrita na equação 21. A correção do peso de uma partícula $x_t^{[m]}$ de acordo com as observações é obtida pelo somatório das correções geradas por cada observação β onde a correção gerada pela observação β_k é calculada como a probabilidade de $x_t^{[m]}$ obtida pela função de densidade de probabilidade M_k multiplicada pelo o grau de confiança sobre a observação α_k .

$$\sigma = \begin{bmatrix} 1e^{-3} & 0 & 0 \\ 0 & 1e^{-3} & 0 \\ 0 & 0 & 1e^{-3} \end{bmatrix} \quad (21)$$

A correção dos pesos é aplicada sobre a predição por meio da multiplicação da probabilidade obtida por meio das observações pelo peso de cada partícula. Por fim, é realizado a normalização dos pesos. Após a normalização é executado a reamostragem por baixa variação, ou *low variance resampling*, onde foi empregado o algoritmo descrito por [42]. Logo após a reamostragem, a posição mais provável do objeto do interesse é

estimada pela média ponderada das partículas pelo pesos como demonstrado na equação 22.

$$\mu_t = \frac{1}{N} \sum_{i=1}^N x_t^{[i]} w_t^{[i]} \quad (22)$$

Como etapa final do filtro de partículas, é realizada a predição sobre a posição do objeto de interesse no próximo instante de tempo. Neste trabalho, não foi implementado nenhum modelo de movimento que pudesse ser utilizado para estimar a próxima posição do objeto de interesse por meio das instanciações representadas pelas partículas. Por este motivo, dentro da etapa de predição, se busca prever a próxima posição do objeto movendo de forma aleatória a posição atual armazenada em cada partícula.

Todas as etapas anteriores, com exceção da inicialização dos pesos, são repetidas até que a tarefa do rastreamento visual seja encerrada.

3.7 Experimento

Para testar as duas abordagens de rastreamento visual foi realizado um experimento no qual um ambiente foi elaborado para a execução de uma tarefa de rastreamento de um objeto esférico por um grupo de robôs. A organização do ambiente pode ser observada na figura 19 onde a linha tracejada representa a trajetória que foi planejada para ser feita pelo objeto esférico, a variável $C_{Obj\ i}$ representa a posição do centroide do objeto i no sistema de coordenadas global em metros, a variável $D_{Obj\ i}$ representa, respectivamente, a largura e o comprimento do objeto i em metros e $P_k = (x_k, y_k, \theta_k)$ representa a posição x_k e y_k no sistema de coordenadas global em metros e o ângulo de orientação θ_k em radianos do robô k .

A posição dos objetos presentes no ambiente e as poses dos robôs envolvidos no rastreamento visual foram determinados com base nos seguintes critérios:

- Os objetos foram posicionados dentro do ambiente com o intuito de gerar oclusão parcial e total sobre as observações dos robôs em certos pontos da trajetória do objeto esférico visto que a fusão de dados buscar contornar esse tipo de problema
- A relação entre as posições dos objetos com as poses dos robôs foi idealizada de modo que nenhum dos robôs conseguisse obter a trajetória completa do objeto esférico, mas a fusão dos dados conseguiria gerar esta trajetória
- A distância mínima e máxima, que possa ser inferida pelo rastreamento visual durante a trajetória do objeto esférico, entre a posição do objeto e qualquer um dos robôs é, respectivamente, de 1.0 e 4.0 metros.

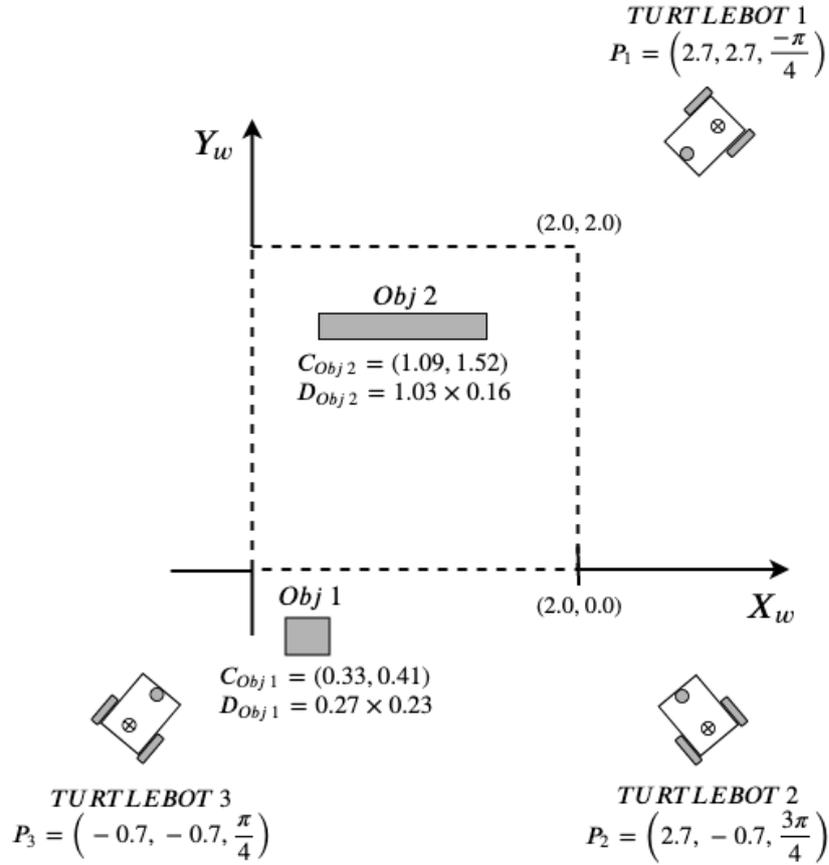


Figura 19: Organização do ambiente elaborado para a realização do experimento. A linha tracejada representa a trajetória que foi planejada para ser feita pelo objeto de interesse. A variável $C_{Obj\ i}$ representa a posição do centroide do objeto i no sistema de coordenadas global em metros, a variável $D_{Obj\ i}$ representa, respectivamente, a largura e o comprimento do objeto i em metros e $P_i = (x_i, y_i, \theta_i)$ representa a posição x_i e y_i no sistema de coordenadas global em metros e o ângulo de orientação θ_i em radianos do robô i

Para evitar erros oriundos da estimação da pose do robô, tendo em vista que a pose do robô é utilizada para a estimação da posição do objeto no espaço $3D$, cada robô foi posicionado no ambiente manualmente e sua pose foi passada para o algoritmo de rastreamento de forma direta, sem a utilização de um algoritmo de inferência. Nenhum dos robôs da equipe de rastreamento se movimentou durante o experimento.

O objeto esférico foi fixado encima de um robô que não pertencia a equipe de rastreamento para que fosse possível realizar a trajetória previamente planejada. Ao robô, foi dado o comando de realizar a trajetória que formasse um quadrado com 2 metros de lado saindo da origem do sistema de coordenadas global em direção ao eixo X^+ rotacionando no sentido anti-horário.

Para comparar com a trajetória do objeto esférico inferida pelo rastreamento visual com trajetória realmente realizada pelo objeto foram obtidos os dados da odometria do

robô que foi responsável por deslocar o objeto. Sobre os dados da odometria é realizada uma pequena translação para que a trajetória passasse a representar a trajetória do objeto. A trajetória obtida por meio da odometria foi considerada como *ground truth* do experimento e são utilizados para validar as trajetórias inferidas por cada robô.



Figura 20: Imagem contendo o objeto de interesse fixado em cima de um Turtlebot

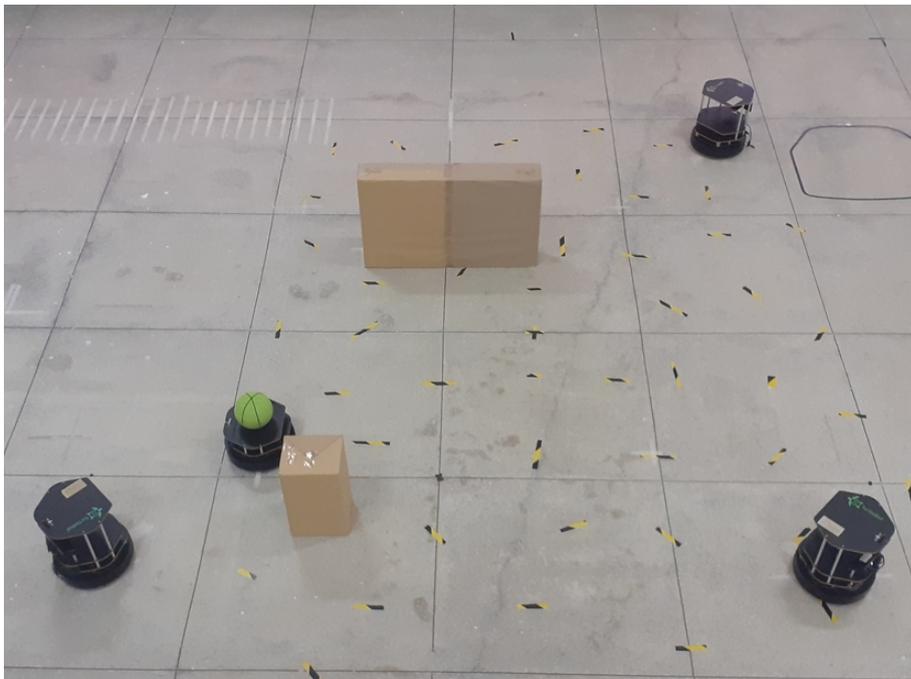


Figura 21: Imagem contendo o ambiente elaborado para a realização do experimento

4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Para gerar os resultados, a cada 30 milissegundos, contando a partir do início do experimento, o erro de estimação da posição do objeto esférico no espaço do mundo foi calculado como a distância euclidiana, em metros, entre a posição do objeto esférico inferida por meio do rastreamento e a posição do objeto esférico obtida pela odometria. O erro de estimação da posição no espaço do mundo de cada robô durante o experimento é mostrado na figura 22. Quando um robô não consegue inferir a posição do objeto, a última posição do objeto de interesse é dada como indefinida, o valor do erro não é computado e nem exibido no gráfico.

Na abordagem não cooperativa, os robôs envolvidos no rastreamento possuem picos de erro quando a sua visão do objeto é prejudicada por causa da oclusão parcial. Isso ocorre devido ao fato de que a oclusão parcial afeta o tamanho do raio do objeto esférico que é observado que por fim afeta a distância que é calculada visto que a distância do objeto em relação a câmera é obtida por meio do raio do objeto esférico no sistema de coordenadas de tela.

Esses picos deixam de acontecer na abordagem cooperativa pois, quando uma observação sofre de oclusão parcial, o valor da variável de confiança sobre a observação α diminui fazendo com que a influência dessa observação sobre a estimação da posição do objeto de interesse durante a fusão dos dados seja menor do que a influência dos demais robôs com uma melhor observação do objeto levando a atenuação do erro.

A principal diferença entre a abordagem não cooperativa e a cooperativa é o fato de que, apesar das oclusões parciais e totais que cada um dos robôs sofre, na abordagem cooperativa todos os robôs conseguem manter o rastreamento do objeto esférico durante todo o experimento enquanto na abordagem não cooperativa há intervalos de tempo nos quais um ou mais robôs não são capazes de estimar a posição do objeto esférico.

Um dos pontos negativos da abordagem cooperativa foi o aumento no custo computacional para a realização do rastreamento. Considerando uma execução como o conjunto de operações que são realizadas sobre as entradas para inferir a posição do objeto de interesse, o algoritmo responsável pela abordagem não cooperativa teve, em média, 29.79 execuções por segundo sendo o máximo de execuções possíveis uma vez que a taxa de aquisição de imagens em RGB do Kinect é de até 30 quadros por segundo. Enquanto o algoritmo responsável pela abordagem cooperativa teve, em média, 4.43 execuções por segundo, como mostra a tabela 4. Essa diminuição é justificada pelo aumento do custo computacional causado pelas operações relacionadas a fusão dos dados que empregou o uso de um filtro de partículas contando com 500 partículas.

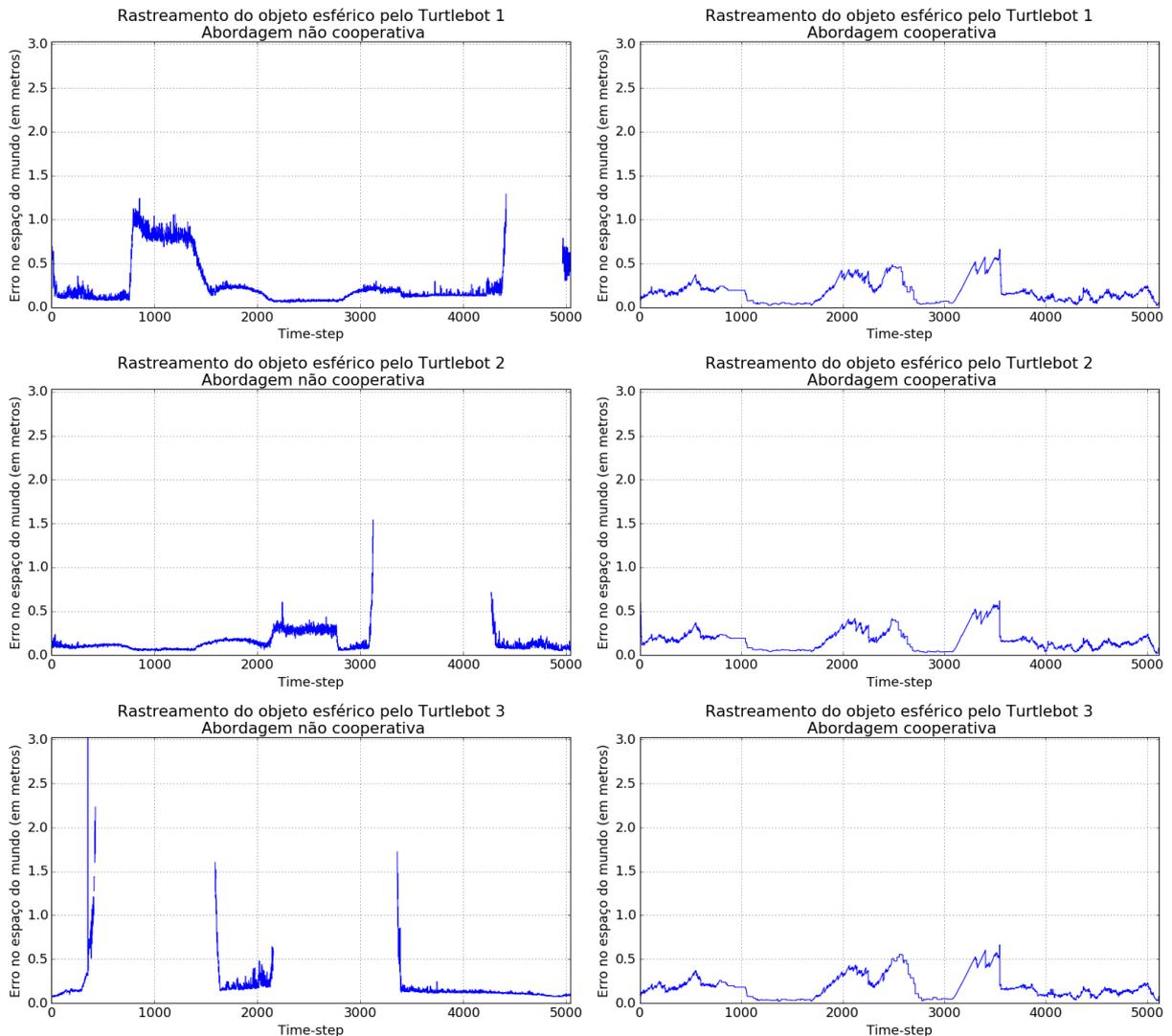


Figura 22: A cada 30 milissegundos, contando a partir do início do experimento, o erro de estimação da posição do objeto esférico no espaço do mundo foi calculado como a distância euclidiana, em metros, entre a posição do objeto esférico inferida por meio do rastreamento e a posição do objeto esférico obtida pela odometria. Os gráficos da coluna à esquerda representam o erro no espaço do mundo do rastreamento do objeto esférico na abordagem não cooperativa enquanto os gráficos da coluna a direita representam o erro do rastreamento do objeto esférico na abordagem cooperativa no espaço do mundo. Quando um robô não consegue inferir a posição do objeto, a última posição do objeto de interesse é dada como indefinida, o valor do erro não é computado e nem exibido no gráfico

	Abordagem não cooperativa	Abordagem cooperativa
Turtlebot 1	29.75	4.23
Turtlebot 2	29.77	4.59
Turtlebot 3	29.84	4.49
Média	29.79	4.43

Tabela 4: Considerando uma execução como o conjunto de operações que são realizadas sobre as entradas para inferir a posição do objeto de interesse, quantidade média de execuções por segundo do algoritmo de rastreamento de cada abordagem por cada robô

Apesar de apresentar um valor médio de execuções por segundos superior ao da abordagem cooperativa, a abordagem não cooperativa obteve uma porcentagem considerável de execuções nas quais a posição do objeto esférico não pode ser inferida, como mostra a tabela 5.

	Abordagem não cooperativa	Abordagem cooperativa
Turtlebot 1	11.37%	0%
Turtlebot 2	22.53%	0%
Turtlebot 3	43.11%	0%
Média	25.67%	0%

Tabela 5: Porcentagem de execuções nas quais a posição do objeto esférico não pode ser inferida

Na abordagem cooperativa, a posição do objeto não foi perdida pois, durante todo o experimento, pelo menos um dos robôs tinha uma observação do objeto esférico sem qualquer tipo de oclusão e esta informação era repassada aos demais. Outro aspecto positivo da abordagem cooperativa foi a diminuição na frequência de erros maiores que 0.6 metros em comparação com a abordagem não cooperativa mostrando que a fusão dos dados gera uma correção na estimação da posição do objeto de interesse. Isso pode ser melhor observado na figura 23 onde os gráficos na parte superior representam o histograma do erro de estimação no espaço do mundo dos robôs durante a abordagem não cooperativa e os gráficos na parte inferior representam o histograma do erro de estimação no espaço do mundo dos robôs durante a abordagem cooperativa.

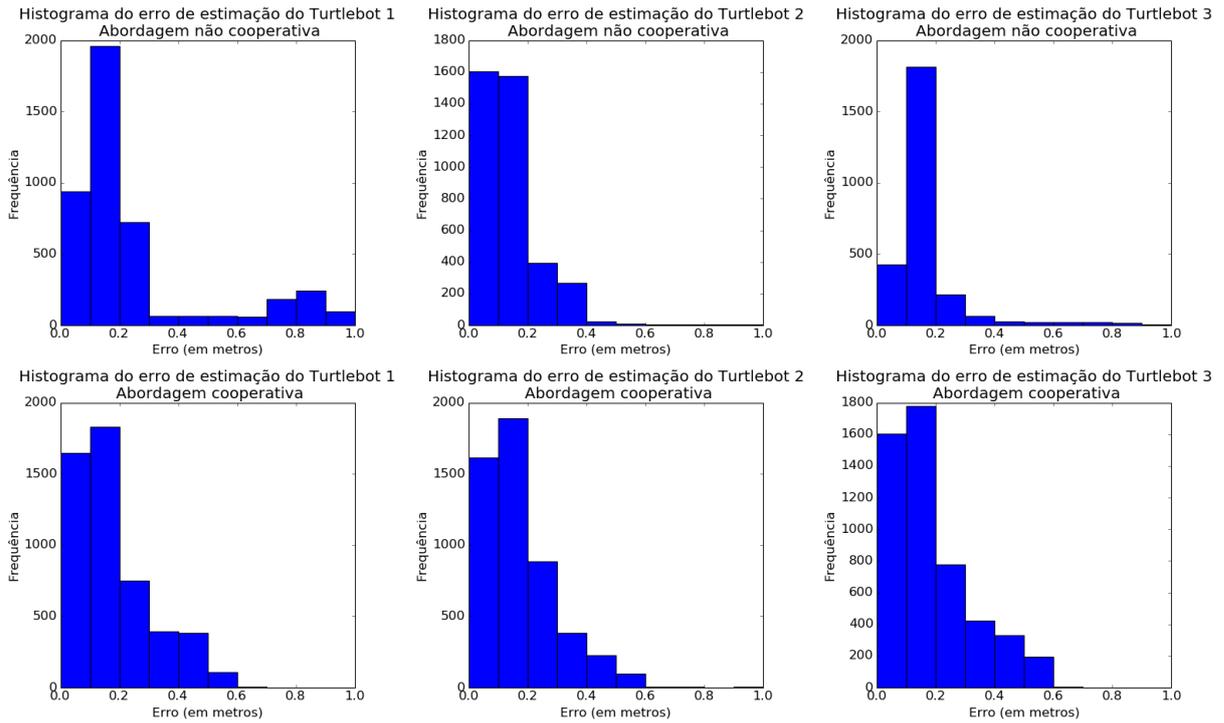


Figura 23: Os gráficos na parte superior representam o histograma do erro de estimação da posição do objeto esférico no espaço do mundo dos robôs durante a abordagem não cooperativa. Os gráficos na parte inferior representam o histograma do erro de estimação da posição do objeto esférico no espaço do mundo dos robôs durante a abordagem cooperativa

5 CONCLUSÕES E TRABALHOS FUTUROS

Foram propostas duas abordagens para a realização do rastreamento visual de um objeto esférico móvel no espaço $3D$ por multi-robôs. A primeira abordagem consistiu no rastreamento visual de forma não cooperativa onde um único robô é responsável por rastrear um objeto esférico sem receber auxílio externo. A segunda abordagem consistiu no rastreamento visual por cooperação multi-robôs onde uma equipe de robôs trocam entre si os dados que são capturados do ambiente. Para melhorar a estimação da posição do objeto esférico, cada robô envolvido no rastreamento de forma cooperativa utilizou a implementação de um filtro de partículas com o intuito de realizar a fusão de seus dados com os dados dos outros membros de sua equipe.

Os resultados do experimento mostraram que o rastreamento visual de forma não cooperativa, apesar de apresentar um baixo custo computacional quando comparado com a forma cooperativa, apresentou dificuldades em estimar a posição do objeto quando se deparava com oclusão parcial ou total. Dificuldades essas que não foram observadas no rastreamento de forma cooperativa dado que a fusão dos dados tornou o rastreamento visual mais robusto para todos os membros da equipe.

Apoiando-se nos resultados, a fusão de dados por meio de filtro de partículas se mostrou ser uma solução viável para o rastreamento de objetos móveis em ambientes que podem ser utilizados dois ou mais sensores para a realização de tal tarefa tendo como algumas de suas vantagens a integração dos dados oriundos de diferentes sensores, a possibilidade de lidar com a incerteza que é gerada pelas mensurações e a fácil escalabilidade embora apresente um aumento significativo do custo computacional.

Em relação aos trabalhos futuros, alguns pontos podem ser destacados:

- A ausência de um mecanismo de predição de movimento do objeto de interesse torna a implementação menos robusta a oclusão parcial ou total e a movimentações abrupta do objeto. A implementação de um modelo de movimento eficaz tanto tornaria a estimação da posição do objeto mais precisa quanto facilitaria a sua detecção
- Sobre o cálculo da confiança sobre a observação, outros atributos além da área observada podem ser utilizados para atribuir a confiança, tendo como exemplo o quão distante se encontra o objeto de interesse e a grau de confiança que o robô tem sobre sua localização. A distância a qual o objeto de encontra pode ser levada em consideração pois quanto mais distante estiver o objeto maior é o erro associado a mensuração dessa variável. Já a confiança do robô sobre sua localização pode ser utilizada visto que a posição e orientação do próprio robô é utilizada para determinar

a posição do objeto de interesse no espaço $3D$, logo um erro na determinação da posição do robô gera um erro sobre a estimação da posição do objeto de interesse

REFERÊNCIAS

- [1] AHMAD, Aamir; LIMA, Pedro. Multi-robot cooperative spherical-object tracking in 3D space based on particle filters. **Robotics And Autonomous Systems**, [s.l.], v. 61, n. 10, p.1084-1093, out. 2013. Elsevier BV. <http://dx.doi.org/10.1016/j.robot.2012.12.008>.
- [2] ALYANNEZHADI, Mohammad M.; POUYAN, Ali A.; ABOLGHASEMI, Vahid. An efficient algorithm for multisensory data fusion under uncertainty condition. **Journal Of Electrical Systems And Information Technology**, [s.l.], v. 4, n. 1, p.269-278, maio 2017. Elsevier BV. <http://dx.doi.org/10.1016/j.jesit.2016.08.002>.
- [3] BRADSKI, Gary; KAEHLER, Adrian. **Learning OpenCV: Computer vision with the OpenCV library**. “O’Reilly Media, Inc.”, 2008.
- [4] CANNONS, Kevin. A review of visual tracking. **Dept. Comput. Sci. Eng., York Univ., Toronto, Canada, Tech. Rep. CSE-2008-07**, v. 242, 2008.
- [5] ĆEHOVIN, Luka. **A hierarchical adaptive model for robust short-term visual tracking**. 2015. 167 f. Tese (Doutorado) - Curso de Computer And Information Science, University Of Ljubljana, Ljubljana, 2015.
- [6] CORKE, Peter. **Robotics, vision and control: fundamental algorithms in MATLAB® second, completely revised**. Springer, 2017.
- [7] DOU, Jianfang; QIN, Qin; TU, Zimei. Robust visual tracking based on generative and discriminative model collaboration. **Multimedia Tools and Applications**, v. 76, n. 14, p. 15839-15866, 2017.
- [8] DOUGHERTY, Edward R.; LOTUFO, Roberto A. **Hands-on morphological image processing**. SPIE press, 2003.
- [9] FAIRCHILD, Carol; HARMAN, Thomas L. **ROS robotics by example**. Packt Publishing Ltd, 2016.
- [10] FORD, Adrian; ROBERTS, Alan. Colour space conversions. **Westminster University, London**, v. 1998, p. 1-31, 1998.
- [11] GHOMMAM, Jawhar; MAHMOUD, Magdi S.; SAAD, Maarouf. Robust cooperative control for a group of mobile robots with quantized information exchange. **Journal Of The Franklin Institute**, [s.l.], v. 350, n. 8, p.2291-2321, out. 2013. Elsevier BV. <http://dx.doi.org/10.1016/j.jfranklin.2013.05.031>.
- [12] GONZALEZ, R. C.; WOODS, R. E. Digital image processing: Pearson prentice hall. **Upper Saddle River, NJ**, v. 1, 2008.

- [13] GUI, Popo et al. Accurately calibrate kinect sensor using indoor control field. In: **2014 Third International Workshop on Earth Observation and Remote Sensing Applications (EORSA)**. IEEE, 2014. p. 9-13.
- [14] HAN, Zhenjun et al. Visual object tracking via sample-based Adaptive Sparse Representation (AdaSR). **Pattern Recognition**, [s.l.], v. 44, n. 9, p.2170-2183, set. 2011. Elsevier BV. <http://dx.doi.org/10.1016/j.patcog.2011.03.002>.
- [15] HONG, Seokmin; SAAVEDRA, Genaro; MARTINEZ-CORRAL, Manuel. Full parallax three-dimensional display from Kinect v1 and v2. **Optical Engineering**, v. 56, n. 4, p. 041305, 2016.
- [16] HSIEH, Ani; LACROIX, Simon. For the JFR special issue on “Multiple collaborative field robots”. **Journal of Field Robotics**, v. 29, n. 5, p. 687-688, 2012.
- [17] HU, Keli et al. A novel object tracking algorithm by fusing color and depth information based on single valued neutrosophic cross-entropy. **Journal Of Intelligent & Fuzzy Systems**, [s.l.], v. 32, n. 3, p.1775-1786, 24 fev. 2017. IOS Press. <http://dx.doi.org/10.3233/jifs-152381>.
- [18] HU, Mengjie et al. Robust object tracking via multi-cue fusion. **Signal Processing**, [s.l.], v. 139, p.86-95, out. 2017. Elsevier BV. <http://dx.doi.org/10.1016/j.sigpro.2017.04.008>.
- [19] IBRAHEEM, Noor A. et al. Understanding color models: a review. **ARPJ Journal of science and technology**, v. 2, n. 3, p. 265-275, 2012.
- [20] IGNACIO, Juliano da Silva. **Processamento e análise digital de imagens em estudos da cinética de recristalização de ligas Al-Mg-X**. 2013. 106 f. Tese (Doutorado) - Curso de Tecnologia Nuclear - Materiais, Instituto de Pesquisas Energéticas e Nucleares, Universidade de São Paulo, São Paulo, 2013.
- [21] ISARD, Michael; BLAKE, Andrew. Condensation—conditional density propagation for visual tracking. **International journal of computer vision**, v. 29, n. 1, p. 5-28, 1998.
- [22] JANG, Dae-Sik; CHOI, Hyung-Il; KIM, Gye-Young. Real-time Tracking with Kalman Filter. In: **MVA**. 1996. p. 10-13.
- [23] KHALEGHI, Bahador et al. Multisensor data fusion: A review of the state-of-the-art. **Information Fusion**, [s.l.], v. 14, n. 1, p.28-44, jan. 2013. Elsevier BV. <http://dx.doi.org/10.1016/j.inffus.2011.08.001>.

- [24] LEANG, Isabelle et al. On-line fusion of trackers for single-object tracking. **Pattern Recognition**, [s.l.], v. 74, p.459-473, fev. 2018. Elsevier BV. <http://dx.doi.org/10.1016/j.patcog.2017.09.026>.
- [25] LEE, Jehoon; SANDHU, Romeil; TANNENBAUM, Allen. Particle filters and occlusion handling for rigid 2D–3D pose tracking. **Computer Vision and Image Understanding**, v. 117, n. 8, p. 922-933, 2013.
- [26] LI, Fan; LIU, Shanshan. Object tracking via a cooperative appearance model. **Knowledge-based Systems**, [s.l.], v. 129, p.61-78, ago. 2017. Elsevier BV. <http://dx.doi.org/10.1016/j.knosys.2017.05.012>.
- [27] LI, Xi et al. A survey of appearance models in visual object tracking. **ACM transactions on Intelligent Systems and Technology (TIST)**, v. 4, n. 4, p. 58, 2013.
- [28] LIU, Wei; WANG, Hui. Visual Tracking Based on Discriminative Compressed Features. **Advances In Multimedia**, [s.l.], v. 2018, p.1-6, 1 ago. 2018. Hindawi Limited. <http://dx.doi.org/10.1155/2018/7481645>.
- [29] MARTINEZ, Aaron; FERNÁNDEZ, Enrique. **Learning ROS for robotics programming**. Packt Publishing Ltd, 2013.
- [30] MORVAN, Yannick. **Acquisition, compression and rendering of depth and texture for multi-view video**. 2009.
- [31] MOTTAGHI, Roozbeh; VAUGHAN, Richard. An integrated particle filter and potential field method applied to cooperative multi-robot target tracking. **Autonomous Robots**, [s.l.], v. 23, n. 1, p.19-35, 4 maio 2007. Springer Nature. <http://dx.doi.org/10.1007/s10514-007-9028-9>.
- [32] GOMES, Otávio da Fonseca Martins. **Microscopia Co-Localizada: Novas Possibilidades na Caracterização de Minérios**. 2007. 105 f. Tese (Doutorado) - Curso de Ciência dos Materiais e Metalurgia, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2007.
- [33] PAGLIARI, Diana; PINTO, Livio. Calibration of kinect for xbox one and comparison between the two generations of microsoft sensors. **Sensors**, v. 15, n. 11, p. 27569-27589, 2015.
- [34] PEREZ, P.; VERMAAK, J.; BLAKE, A.. Data Fusion for Visual Tracking With Particles. **Proceedings Of The Ieee**, [s.l.], v. 92, n. 3, p.495-513, mar. 2004. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/jproc.2003.823147>.

- [35] QUIGLEY, Morgan et al. ROS: an open-source Robot Operating System. In: **ICRA workshop on open source software**. 2009. p. 5.
- [36] SIEGWART, Roland; NOURBAKHSI, Illah Reza; SCARAMUZZA, Davide. **Introduction to autonomous mobile robots**. MIT press, 2011.
- [37] SMISEK, Jan; JANCOSEK, Michal; PAJDLA, Tomas. 3D with Kinect. In: **Consumer depth cameras for computer vision**. Springer, London, 2013. p. 3-25.
- [38] SONG, Zhiguo; SUN, Jifeng; YU, Jialin. Object Tracking by a Combination of Discriminative Global and Generative Multi-Scale Local Models. **Information**, v. 8, n. 2, p. 43, 2017.
- [39] STACHNISS, Cyrill. **Exploration and mapping with mobile robots**. 2006. Tese de Doutorado. Verlag nicht ermittelbar.
- [40] SUGANDI, Budi et al. Object Tracking Based on Color Information Employing Particle Filter Algorithm. **Object Tracking**. InTech, 2011.
- [41] SUZUKI, Satoshi et al. Topological structural analysis of digitized binary images by border following. **Computer vision, graphics, and image processing**, v. 30, n. 1, p. 32-46, 1985.
- [42] THRUN, Sebastian; BURGARD, Wolfram; FOX, Dieter. **Probabilistic Robotics**. [s. l.]: The Mit Press, 2005.
- [43] TRUONG, Mai Thanh Nhat; PAK, Myeongsuk; KIM, Sanghoon. Single object tracking using particle filter framework and saliency-based weighted color histogram. **Multimedia Tools And Applications**, [s.l.], v. 77, n. 22, p.30067-30088, 4 jun. 2018. Springer Nature America, Inc. <http://dx.doi.org/10.1007/s11042-018-6180-5>.
- [44] VIDAL, Flávio de Barros. **Rastreamento Visual de Objetos Utilizando Métodos de Similaridade de Regiões e Filtragem Estocástica**. 2009. 99 f. Tese (Doutorado) - Curso de Engenharia Elétrica, Universidade de Brasília, Distrito Federal, 2009.
- [45] YANG, Hanxuan et al. Recent advances and trends in visual tracking: A review. **Neurocomputing**, [s.l.], v. 74, n. 18, p.3823-3831, nov. 2011. Elsevier BV. <http://dx.doi.org/10.1016/j.neucom.2011.07.024>.
- [46] YI, Shuangyan et al. Single object tracking via robust combination of particle filter and sparse representation. **Signal Processing**, v. 110, p. 178-187, 2015.

- [47] YILMAZ, Alper; JAVED, Omar; SHAH, Mubarak. Object tracking: A Survey. **Acm Computing Surveys**, [s.l.], v. 38, n. 4, p.13, 25 dez. 2006. Association for Computing Machinery (ACM). <http://dx.doi.org/10.1145/1177352.1177355>.
- [48] WANG, Naiyan et al. Understanding and diagnosing visual tracking systems. In: **Proceedings of the IEEE International Conference on Computer Vision**. 2015. p. 3101-3109.
- [49] WU, Yi; LIM, Jongwoo; YANG, Ming-hsuan. Object Tracking Benchmark. **IEEE Transactions On Pattern Analysis And Machine Intelligence**, [s.l.], v. 37, n. 9, p.1834-1848, 1 set. 2015. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/tpami.2014.2388226>.
- [50] ZHOU, Ke; ROUMELIOTIS, S. I.. Multirobot Active Target Tracking With Combinations of Relative Observations. **Ieee Transactions On Robotics**, [s.l.], v. 27, n. 4, p.678-695, ago. 2011. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/tro.2011.2114734>.