

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**SAS-DV: UMA ARQUITETURA DE PREDIÇÃO  
DE RISCOS DE COLISÃO PARA PESSOAS  
COM DEFICIÊNCIA VISUAL**

**NATAL HENRIQUE CORDEIRO**

**ORIENTADOR: PROF. DR. EMERSON CARLOS PEDRINO**

São Carlos – SP

Fevereiro/2020

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**SAS-DV: UMA ARQUITETURA DE PREDIÇÃO  
DE RISCOS DE COLISÃO PARA PESSOAS  
COM DEFICIÊNCIA VISUAL**

**NATAL HENRIQUE CORDEIRO**

Tese apresentada ao Programa de Pós-Graduação em  
Ciência da Computação da Universidade Federal de  
São Carlos, como parte dos requisitos para a obten-  
ção do título de Doutor em Ciência da Computação,  
área de concentração: Processamento de Imagens e  
Sinais

Orientador: Prof. Dr. Emerson Carlos Pedrino

São Carlos – SP

Fevereiro/2020



UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia  
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Tese de Doutorado do candidato Natal Henrique Cordeiro, realizada em 10/02/2020:

Prof. Dr. Emerson Carlos Pedrino  
UFSCar

Prof. Dr. Fredy João Valente  
UFSCar

Prof. Dr. José Hiroki Saito  
UFSCar

Prof. Dr. Valentin Obac Roda  
UFRN

Prof. Dr. Luiz Marcos Garcia Gonçalves  
UFRN

Certifico que a defesa realizou-se com a participação à distância do(s) membro(s) Luiz Marcos Garcia Gonçalves depois das arguições e deliberações realizadas, o(s) participante(s) à distância está(ão) de acordo com o conteúdo e parecer da banca examinadora redigido neste relatório de defesa.

Prof. Dr. Emerson Carlos Pedrino

Dedico o presente trabalho a minha família e aos meus professores que foram peças fundamentais para a minha formação profissional



## **AGRADECIMENTOS**

Gostaria de agradecer em primeiro lugar a Deus, aos meus pais e a minha esposa, que sempre me apoiaram; as minhas filhas que foram o estímulo para os meus estudos; aos meus irmãos, a minha família e amigos que presenciaram o meu esforço; ao Professor Dr. Emerson Carlos Pedrino, pela orientação do trabalho e pelo companheirismo; ao corpo docente da UFSCar, por proporcionar uma formação de alto nível; aos alunos companheiros e docentes do Laboratório GAPIS; aos Professores Doutores Antônio Carlos Sementille (UNESP), João Paulo Papa (UNESP/UFSCar) e Fredy João Valente (UFSCar), pela avaliação do texto da minha qualificação de Doutorado por meio de críticas, correções e sugestões, contribuindo com a qualidade do trabalho; aos Professores Doutores Fredy João Valente (UFSCar), José Hiroki Saito (UFSCar), Luiz Marcos Garcia Gonçalves (UFRN), Paulo Estevão Cruvinel (EMBRAPA/UFSCar), Roberto Santos Inoue (UFSCar), Rodrigo Capobianco Guido (UNESP), Sylvio Barbon Junior (UEL) e Valentin Obac Roda (UFRN), por terem aceitado compor a banca avaliadora da minha defesa de doutorado; a todos aqueles que, de forma direta ou indireta, contribuíram para a realização do presente trabalho.

## RESUMO

As tecnologias desenvolvidas até o momento para a navegação de deficientes visuais (DV) atendem algumas de suas necessidades cotidianas. O trabalho aqui desenvolvido propicia ao deficiente visual fornecer a consciência da situação (*Situation Awareness - SAW*) ao ser gerado um mapa de riscos após a análise das posições, distâncias, tamanhos e ações dos objetos presentes no ambiente no qual esse está inserido. Essa compreensão é refinada por etapas em que se aplica Fusão de Informações de Alto Nível para a projeção de impactos num futuro próximo. Um mapa de riscos é formado após a execução de um conjunto de inferências. Para a execução dessas inferências propõe-se neste trabalho dois conjuntos de dados (*datasets*) específicos para projetar os riscos de colisão em diferentes direções. Os *datasets* são compostos de informações referentes à posição e ao tamanho de: objetos estáticos, passagens livres, objetos dinâmicos e trajetos executados por objetos dinâmicos. Todas essas informações foram mapeadas em um plano 3D. Dessa forma, para demonstrar as vantagens de implementação da Fusão de Informações de Alto Nível no sistema de navegação dedicado ao DV realizaram-se experimentos com a arquitetura proposta e com outros três sistemas de navegação implementados com diferentes abordagens. Com a geração dos resultados, foi possível validar e comparar os sistemas de navegação. Para essa análise comparativa, diferentes situações foram testadas pelos sistemas de navegação, com o objetivo de indicar para o DV uma direção para trafegar com menos riscos de colisão. Além de fornecer um mapa de riscos, com possíveis colisões, o presente estudo apresentou maior confiabilidade para a navegação, quando forem detectados obstáculos mais próximos e trajetos executados por objetos dinâmicos. Nesse mesmo estudo também foram implementadas diferentes técnicas para a detecção e rastreamento de objetos dinâmicos para definir qual delas melhor se adequou ao contexto do DV. Esta pesquisa pode fornecer novas contribuições, como adaptações em técnicas já renomadas, usadas para análise de objetos dinâmicos e, principalmente, com uma metodologia que pode ser usada para comparar a eficiência e o custo computacional de cada técnica para a análise destes objetos.

**Palavras-chave:** Fusão de Informação de Alto Nível, Predição de Risco de Colisão, Deficientes Visuais, Consciência da Situação

# ABSTRACT

The technologies developed so far to help visually impaired people (VIP) navigate meet only some of their everyday needs. This project allows visually impaired people to improve the situation awareness (*SAW*) by generating a risk map following an analysis of the position, distance, size and motion of the objects present in their environment. This comprehension is refined by data fusion steps applied to High Level Information Fusion to predict possible impacts in the near future. A risk map is formed after a set of inferences has been executed. Two specific datasets for projecting the risks of collision in different directions are proposed in this work for the execution of these inferences. These datasets are composed of information regarding the position and size of static objects, free passages, dynamic objects, and paths executed by dynamic objects. All this information was mapped in a 3D plane. Thus, to demonstrate the advantages of implementing High Level Information Fusion in the navigation system dedicated to VIP, experiments were performed with the proposed architecture and with three other navigation systems implemented with different approaches. The results demonstrated it was possible to validate and compare the navigation systems. For this comparative analysis, different situations were tested by the navigation systems so that the VIP could be indicated, namely a direction to travel which includes fewer collision risks. In addition to providing a risk map giving possible collisions, this study provided greater reliability for navigation, especially when obstacles were very close and moving objects were detected. Different techniques for detecting and tracking dynamic objects were also implemented in this study for defining which of them is best suited to the VIP context. This study can provide new contributions such as adaptations to already renowned techniques used for dynamic object analysis and, most notably, with the use of a methodology for comparing the efficiency and computational cost of each technique to analyze such objects.

**Keywords:** High Level Information Fusion, Collision Risk Prediction, Visually Impaired People, Situation Awareness

## LISTA DE FIGURAS

1.1	(a) Simulação de um ambiente <i>indoor</i> contendo o DV e diversos obstáculos. Nesta imagem, o DV é a pessoa que possui uma mochila nas costas e está com uma vestimenta de azul-marinho. (b) Mesmo ambiente da imagem (a) com uma projeção diferente. . . . .	22
3.1	Uma análise dos objetivos do usuário usando a teoria <i>GDTA</i> . . . . .	37
3.2	Arquitetura (SAS-DV) desenvolvida neste trabalho. . . . .	38
3.3	(a) O dispositivo Kinect e seus principais sensores visíveis; (b) A posição em que o dispositivo Kinect foi usado em uma pessoa neste trabalho. . . . .	39
3.4	Sistema de referência. . . . .	42
3.5	Uma simples representação de como aplicar as equações 3.1 e 3.2. . . . .	43
3.6	(a) Contorno dos obstáculos; (b) Características extraídas dos obstáculos ( <i>cm</i> , <i>p1</i> e <i>p2</i> ) da imagem (a). . . . .	44
3.7	(a) Contorno dos obstáculos; (b) Posições de obstáculos convertidas do eixo ( <i>x</i> , <i>y</i> ) da imagem (a) para ( <i>X</i> , <i>Z</i> ). . . . .	45
3.8	(a) Área de abrangência do dispositivo Kinect. (b) Simulação de aquisição de dados. . . . .	50
3.9	Pontos de referência (PRf) definidos para detectar a distância dos obstáculos em várias direções e também usados para calcular os riscos de colisão em algumas direções na área de abrangência que o dispositivo Kinect atua (ver Figura 3.8 (a)). . . . .	51
3.10	Pontos de referência com suas respectivas distâncias. . . . .	52

3.11	Dados que compõem o vetor de característica usado na primeira estratégia de mapeamento do processo de percepção / Base de criação para o primeiro <i>dataset</i> no processo de compreensão. . . . .	53
3.12	Elementos usados para criar a segunda estratégia do processo de percepção e para produzir o segundo <i>dataset</i> no processo de compreensão. . . . .	55
3.13	Gráfico (mapa de riscos) que representa os riscos de colisão que o DV pode ter em cada direção. . . . .	59
4.1	A organização dos submódulos do Sistema de Objetos Dinâmicos da arquitetura SAS-DV. . . . .	61
4.2	Exemplo de ambiente interno usado nos experimentos. Cada símbolo desta imagem representa um elemento relevante no ambiente. . . . .	63
4.3	Fluxograma da metodologia aplicada para analisar as arquiteturas de detecção e rastreamento de objetos dinâmicos e definir a solução adequada para o contexto proposto. . . . .	64
4.4	Organização dos submódulos das Arquiteturas 1 (A1-RP) e 2 (A2-FOF). . . . .	65
4.5	(a) - Demonstração do contorno de uma pessoa detectada. (b) - Cálculo do centro de massa da pessoa detectada na imagem (a). . . . .	66
4.6	Simulação dos vetores de velocidade implementados na técnica de Fluxo Óptico de Farneback com a detecção de movimentos divergentes. . . . .	68
4.7	Organização dos submódulos da Arquitetura 3 (A3-SB). . . . .	70
4.8	Arquitetura 4 (A4-CAM). . . . .	73
4.9	Medindo a efetividade com o Mapa(X , Z). (a) DROD usando A1-RP; (b) DROD usando A2-FOF; (c) DROD usando A3-SB; (d) DROD usando A4-CAM. . . . .	75
4.10	(a) Custo computacional de quatro processadores ( <i>CPUs</i> ) do início ao fim da execução de uma técnica de DROD. (b) Custo computacional da Memória <i>RAM</i> de 8 GB do início ao fim da execução de uma técnica de DROD. . . . .	76
5.1	Representação das distâncias em cores presentes nos mapas de profundidade. . . . .	78

5.2	(a, d, g, j) - imagens fornecidas pelo sensor RGB; (b, e, h, k) - imagens disponibilizadas pelo Sensor IV (mapas de profundidade) com filtros de dilatação, erosão e suavização; (c, f, i, l) - imagens com filtros de dilatação, erosão e suavização mais concentrados. . . . .	79
5.3	(a, d, g, j) - imagens adquiridas num ambiente interno; (b, e, h, k) - segmentação dos objetos estáticos; (c, f, i, l) - detecção do contorno dos objetos estáticos. . .	81
5.4	(a, d, g, j) - imagens fornecidas pelo Sensor RGB; (b, e, h, k) - imagens fornecidas pelo Sensor IV; (c, f, i, l) - detecção da largura da passagem livre. . . . .	82
5.5	(a, b, c, d) - detecção de objetos dinâmicos em diferentes instantes com a técnica de Reconhecimento de Padrões. . . . .	83
5.6	(a, c, e, g, i) - imagens com a detecção de objetos dinâmicos (pessoa) com a técnica de Fluxo Óptico de Farneback; (b, d, f, h, j) - Mapas de profundidade para obtenção da distância do objeto dinâmico; (k) - Detecção do contorno do objeto dinâmico; (l) Contorno produzido para calcular a largura e altura do objeto dinâmico detectado. . . . .	84
5.7	(a, e, i) - imagens do sensor RGB; (b, f, j) - imagens do sensor IV; (c, g, k) - segmentação do objeto dinâmico pela técnica de Subtração de <i>Background</i> ; (d, h, l) - geração do contorno e do centro de massa dos objetos dinâmicos segmentados. . . . .	86
5.8	(a, c, e, g, i, k) - imagens com presença de movimento do vídeo; (b, d, f, h, j, l) - imagens estabilizadas após detecção de movimentos no sensor de aquisição (SA). . . . .	87
5.9	(a, e, i, m) - imagens fornecidas pelo Sensor RGB; (b, f, j, n) - mapas de profundidade estabilizados usando pontos de referências nas regiões periféricas da imagem; (c, g, k, o) - segmentação de objetos dinâmicos usando Subtração de <i>Background</i> com a presença de movimentos no SA; (d, h, l, p) - detecção de contorno. . . . .	89
5.10	(a, b, c, d) - detecção de objetos dinâmicos em diferentes instantes usando a técnica CamShift. . . . .	90
5.11	Trajetos realizados por uma pessoa (objeto dinâmico) em um Mapa(X, Z). No eixo X pode ser visto a largura do objeto e no eixo Z a sua respectiva distância.	91

5.12	(a, d) - imagens com a detecção de objetos dinâmicos (pessoa) com a técnica de Reconhecimento de Padrões; (b, e) - mapas de profundidade; (c, f) - geração do Mapa (X,Z) usando a arquitetura A1-RP. . . . .	92
5.13	(a, d) - imagens com a detecção de objetos dinâmicos com a técnica de Fluxo Óptico de Farneback; (b, e) - mapas de profundidade; (c, f) - geração do Mapa(X , Z) usando a arquitetura A2-FOF. . . . .	93
5.14	(a, d) - imagens com a detecção de objetos dinâmicos com a técnica de Subtração de <i>Background</i> ; (b, e) - mapas de profundidade; (c, f) - geração do Mapa(X , Z) usando a arquitetura A3-SB. . . . .	93
5.15	(a, d) - imagens com o rastreamento de objetos dinâmicos usando a técnica CamShift; (b, e) - detecção dos contornos; (c, f) - geração do Mapa(X , Z) usando a arquitetura A4-CAM. . . . .	94
5.16	Situação 1: Processo de extração de características para experimentos com o Módulo de Fusão. (a) - imagem disponibilizada pelo Sensor RGB; (b) - mapa de profundidade; (c) - detecção do contorno da passagem livre; (d) - detecção de objetos estáticos; (e) - contorno dos objetos estáticos. . . . .	95
5.17	Situação 2: Processo de extração de características para experimentos com o Módulo de Fusão. (a) - imagem disponibilizada pelo Sensor RGB; (b) - mapa de profundidade; (c) - detecção do contorno da passagem livre; (d) - detecção de objetos estáticos; (e) - contorno dos objetos estáticos. . . . .	97
5.18	Processo de extração de características - Quatro situações. (a, f, k, p) - imagens disponibilizadas pelo Sensor RBG; (b, g, l, q) - mapas de profundidade; (c, h, m) - detecção do contorno da passagem livre; (d, i, n) - detecção dos objetos estáticos; (e, j, o) - detecção do contorno dos objetos estáticos; (r) - detecção do objeto dinâmico; (s) - detecção do contorno do objeto dinâmico; (t) - centro de massa do objeto dinâmico. . . . .	100
5.19	(a, b, c) - direções fornecidas pelo $SN_1$ nas três primeiras situações apresentadas na Figura 5.18. . . . .	101
5.20	Direções fornecidas pelo $SN_2$ nas três primeiras situações apresentadas na Figura 5.18. . . . .	101
5.21	Mapa de riscos produzido a partir da Situação 1 apresentada na Figura 5.18. . .	104
5.22	Mapa de riscos produzido a partir da Situação 2 apresentada na Figura 5.18. . .	105

5.23	Mapa de riscos produzido a partir das situações 3 e 4 apresentadas na Figura 5.18.	106
6.1	(a) Ambiente 1. (b) Ambiente 2. . . . .	108
6.2	(a) Média da Velocidade de detecção das arquiteturas; (b) Média da efetividade das arquiteturas; (c) Custo computacional médio das arquiteturas ( <i>CPU</i> e memória <i>RAM</i> ). . . . .	109
A.1	Código Fonte - Detecção de Objetos Estáticos com menos de 2 metros de distância	126
A.2	Código Fonte - Detecção da Passagem Livre com mais de 3 metros de distância	127
A.3	Código Fonte - Detecção de Objetos Dinâmicos usando Reconhecimento de padrões . . . . .	128
A.4	Código Fonte - Detecção de Objetos Dinâmicos usando Fluxo Óptico de Farneback . . . . .	129
A.5	Código Fonte - Método ( <i>drawOptFlowMap</i> ) que detecta objeto dinâmico mesmo com a presença de movimento na câmera. . . . .	130
A.6	Código Fonte - Detecção de Objetos Dinâmicos usando Subtração de <i>Background</i>	131
A.7	Código Fonte - Detecção de Objetos Dinâmicos usando <i>CamShift</i> - Parte 1 . . .	132
A.8	Código Fonte - Detecção de Objetos Dinâmicos usando <i>CamShift</i> - Parte 2 . . .	133



## LISTA DE TABELAS

2.1	Técnicas usadas em Sistemas de Navegação - Visão Computacional (VC), Mapeamento (MAP), Reconhecimento de Padrões (RP), Conversão 2D para 3D (2D-3D), Análise de Objetos Dinâmicos (AOD), Percepção (PE), Compreensão (CO) e Projeção (PRO) . . . . .	28
3.1	Organização dos dados do primeiro <i>dataset</i> . . . . .	57
3.2	Organização dos dados do segundo <i>dataset</i> . . . . .	58
5.1	Dados gerados pelo Submódulo de Percepção da Situação 1 (Figura 5.16) para o modelo 1 - Objetos Dinâmicos (OD), Trajetos realizados por ODs (TRJ), Objetos Estáticos (OE), Passagem Livre (PL), PRf detectados com menos de 1.4m, 1.7m e 2.0m respectivamente (PRf1.4m, PRf1.7m. PRf2m) e Risco de Colisão (RC) . . . . .	96
5.2	Dados gerados pelo sub-módulo de Percepção da Situação 2 (ver Figura 5.17) para o modelo 1 - Objetos Dinâmicos (OD), Trajetos realizados por ODs (TRJ), Objetos Estáticos (OE), Passagem Livre (PL), PRf detectados com menos de 1.4m, 1.7m e 2.0m respectivamente (PRf1.4m, PRf1.7m. PRf2m) e Risco de Colisão (RC) . . . . .	97
5.3	Dados gerados pelo Submódulo de Percepção da Situação 1 (Figura 5.16) para o modelo 2 - Objetos Estáticos (OE), Objetos Dinâmicos (OD), Passagem Livre (PL), Trajetos realizados por ODs (TRJ) e Risco de Colisão (RC) . . . . .	98
5.4	Dados gerados pelo Submódulo de Percepção da Situação 2 (Figura 5.17) para o modelo 2 - Objetos Estáticos (OE), Objetos Dinâmicos (OD), Passagem Livre (PL), Trajetos realizados por ODs (TRJ) e Risco de Colisão (RC) . . . . .	98
6.1	Comparação das técnicas de DROD com o DV paralisado - ambiente 1 . . . . .	108
6.2	Comparação das técnicas de DROD com o DV paralisado - ambiente 2 . . . . .	108

6.3	Comparação das técnicas de DROD com o DV em movimento - ambiente 1 . . .	109
6.4	Comparação das técnicas de DROD com o DV em movimento - ambiente 2 . . .	109
6.5	Comparação entre as Arquiteturas de Detecção e Rastreamento de OD (DRODs)	112
6.6	Análise comparativa entre os quatro Sistemas de Navegação . . . . .	114
6.7	Comparação de classificadores com validação cruzada - (uso de <i>Folds</i> = 10) . . .	116

# SUMÁRIO

<b>LISTA DE ABREVIATURAS E SIGLAS</b>	<b>18</b>
<b>CAPÍTULO 1 – INTRODUÇÃO</b>	<b>20</b>
1.1 Contextualização e Motivação . . . . .	20
1.2 Objetivos e aspectos inovadores . . . . .	24
1.3 Organização do Trabalho . . . . .	25
<b>CAPÍTULO 2 – TRABALHOS RELACIONADOS</b>	<b>27</b>
2.1 Consciência da situação ( <i>Situation Awareness</i> ) . . . . .	30
2.2 Detecção e rastreamento de objetos dinâmicos (DROD) . . . . .	32
<b>CAPÍTULO 3 – SISTEMA DE ANÁLISE SENSORIAL PARA DEFICIENTES VISUAIS (SAS-DV)</b>	<b>36</b>
3.1 Módulo de Entrada e Saída (E/S) . . . . .	38
3.2 Módulo de Visão . . . . .	40
3.2.1 Submódulo de Mapa de Profundidade . . . . .	40
3.2.2 Sistema de Objetos Estáticos . . . . .	41
3.2.2.1 Submódulo de Segmentação de Objetos Estáticos . . . . .	41
3.2.2.2 Submódulo de Posição 3D . . . . .	41
3.2.3 Sistema de Objetos Dinâmicos . . . . .	45
3.2.3.1 A1-RP . . . . .	47
3.2.3.2 A2-FOF . . . . .	47

3.2.3.3	A3-SB . . . . .	48
3.2.3.4	A4-CAM . . . . .	48
3.2.3.5	Versão final do Sistema de Detecção e Rastreamento de Ob- jetos Dinâmicos . . . . .	49
3.3	Módulo de Fusão . . . . .	49
3.3.0.1	Submódulo de Percepção . . . . .	50
	Mapeamento e Pontos de referência . . . . .	50
3.3.0.2	Submódulo de Compreensão . . . . .	55
	Primeiro <i>Dataset</i> (Boolean) . . . . .	56
	Segundo <i>Dataset</i> ( <i>numeric</i> ) . . . . .	57
3.3.0.3	Submódulo de Projeção . . . . .	58
 <b>CAPÍTULO 4 – SISTEMA DE DETECÇÃO E RASTREAMENTO DE OBJETOS DINÂMICOS</b>		<b>61</b>
4.1	Os Objetos Dinâmicos e a mobilidade do Deficiente Visual . . . . .	62
4.2	Detecção e Rastreamento de Objetos Dinâmicos (DROD) . . . . .	64
4.2.1	Arquitetura 1 - Reconhecimento de padrões usando Haar-cascade (A1- RP) . . . . .	65
4.2.1.1	Sensores de Aquisição, Mapa de Profundidade e Posição 3D	66
4.2.1.2	SSOD usando Reconhecimento de Padrões . . . . .	66
4.2.2	Arquitetura 2 - Fluxo Óptico de Farneback (A2-FOF) . . . . .	67
4.2.2.1	SSOD usando FOF . . . . .	67
4.2.3	Arquitetura 3 - Subtração de <i>Background</i> (A3-SB) . . . . .	69
4.2.4	Arquitetura 4 - Continuously Adaptive Meanshift (A4-CAM) . . . . .	73
4.3	Metodologia desenvolvida para comparar as arquiteturas DROD . . . . .	74
 <b>CAPÍTULO 5 – EXPERIMENTOS</b>		<b>77</b>
5.1	Experimentos realizados com o Módulo de Visão . . . . .	77

5.1.1	Experimentos - Submódulo de Mapa de Profundidade . . . . .	78
5.1.2	Experimentos - Submódulo de Segmentação de Objetos Estáticos . . .	80
5.1.3	Experimentos - Submódulo de Segmentação de Objetos Dinâmicos . .	82
5.1.3.1	Reconhecimento de Padrões . . . . .	83
5.1.3.2	Fluxo Óptico de Farneback . . . . .	83
5.1.3.3	Subtração de <i>Background</i> . . . . .	85
	Experimentos com o Submódulo de Controle e Estabilização de	
	Vídeo . . . . .	86
5.1.3.4	CamShift . . . . .	89
5.1.4	Experimentos - Submódulo de Posição 3D . . . . .	90
5.1.4.1	Experimentos do SP3D na A1-RP . . . . .	91
5.1.4.2	Experimentos do SP3D na A2-FOF . . . . .	92
5.1.4.3	Experimentos do SP3D na A3-SB . . . . .	92
5.1.4.4	Experimentos do SP3D na A4-CAM . . . . .	93
5.2	Experimentos realizados com o Módulo de Fusão . . . . .	94
5.2.1	Análise comparativa com outros sistemas de navegação . . . . .	99
<b>CAPÍTULO 6 – RESULTADOS E DISCUSSÃO</b>		<b>107</b>
6.1	Resultados e Discussão - Módulo de Visão . . . . .	107
6.2	Resultados e Discussão - Módulo de Fusão . . . . .	113
6.2.1	Análise comparativa entre os Sistemas de Navegação implementados .	113
<b>CAPÍTULO 7 – CONCLUSÃO</b>		<b>118</b>
<b>REFERÊNCIAS</b>		<b>120</b>
<b>APÊNDICE A – SOBRE O TRABALHO</b>		<b>124</b>
A.1	Financiamento e Agradecimento . . . . .	124
A.2	Imagens . . . . .	124

A.3	Vídeos . . . . .	124
A.4	Publicações . . . . .	125
A.5	Código Fonte . . . . .	126
A.5.1	Objetos Estáticos . . . . .	126
A.5.2	Passagem Livre . . . . .	127
A.5.3	DROD - Arquitetura 1 - Reconhecimento de padrões . . . . .	128
A.5.4	DROD - Arquitetura 2 - Fluxo Óptico de Farneback . . . . .	129
A.5.5	DROD - Arquitetura 3 - Subtração de <i>Background</i> . . . . .	131
A.5.6	DROD - Arquitetura 4 - Continuously Adaptive Meanshift . . . . .	132

## LISTA DE ABREVIATURAS E SIGLAS

---

---

- A1-RP** – *Arquitetura 1 - Reconhecimento de padrões*
- A2-FOF** – *Arquitetura 2 - Fluxo Óptico de Farneback*
- A3-SB** – *Arquitetura 3 - Subtração de Background*
- A4-CAM** – *Arquitetura 4 - Continuously Adaptive Meanshift*
- DI** – *Direção Indicada*
- DROD** – *Detecção e Rastreamento de Objetos Dinâmicos*
- DV** – *Deficientes Visuais*
- IV** – *Sensor Infravermelho*
- OD** – *Objetos Dinâmicos*
- OE** – *Objetos Estáticos*
- PL** – *Passagem Livre*
- PRf** – *Pontos de Referência*
- RFID** – *Radio Frequency IDentification*
- RGB** – *Sensor RGB*
- SAS-DV** – *Sistema de Análise Sensorial para Deficientes Visuais*
- SAW** – *Situation Awareness*
- SA** – *Sistema de Aquisição*
- SCEV** – *Submódulo de Controle e Estabilização de Vídeo*
- SN** – *Sistema de Navegação*
- SP3D** – *Submódulo de Posição 3D*

**SSOD** – *Submódulo de Segmentação de Objetos Dinâmicos*

**SSOE** – *Submódulo de Segmentação de Objetos Estáticos*



# Capítulo 1

## INTRODUÇÃO

---

---

A deficiência visual em pessoas compromete a mobilidade e tem grande impacto na inclusão social. Segundo o relatório da Organização Mundial de Saúde (MARIOTTI, 2014), estima-se a existência de 188,5 milhões de pessoas (no mundo) que apresentam comprometimento visual leve, 217 milhões que apresentam comprometimento de moderado a grave e 36 milhões de pessoas que são cegas. A realização de pesquisas e o consequente desenvolvimento de tecnologias de apoio sensorial e navegação, para pessoas com deficiência visual, estão em crescente expansão (CHAN; ENGELKE; ABHAYASINGHE, 2017; MEKHALFI et al., 2016; TSIRMPAS et al., 2015; MASCETTI et al., 2016). Essas tecnologias têm sido adotadas para suplementar as funções de uma pessoa portadora de deficiência visual e visam proporcionar uma locomoção baseada nas características extraídas do ambiente. Apesar de ser uma área de pesquisa específica, solucionar os problemas enfrentados por essas pessoas exige uma análise abrangente. Dessa forma, é possível que um Deficiente Visual (DV) interaja com outras pessoas, alcance os destinos desejados e consiga compreender o que acontece ao seu redor.

### 1.1 Contextualização e Motivação

Para que o DV compreenda um pouco do ambiente em que está presente e entenda quais impactos ele pode ter em diferentes direções, este trabalho apresenta uma nova arquitetura, contendo técnicas de Visão Computacional e Fusão de Dados que fornecem suporte à tomada de decisão para um DV. Essa arquitetura foi intitulada Sistema de Análise Sensorial para Deficientes Visuais (SAS-DV) e foi projetada por meio de três sistemas. O primeiro sistema detecta Objetos Estáticos (OEs) e passagens livres (PLs). O segundo sistema detecta e executa o rastreamento de Objetos Dinâmicos (ODs). Nos dois sistemas é realizado o mapeamento de todos os objetos, detectados num espaço tridimensional (em milímetros). O terceiro sistema produz

os riscos de colisão que o DV pode ter em determinadas direções. Para o cálculo desses riscos, foram analisadas situações que normalmente surgem com os DVs.

Uma situação pode ser entendida como um conjunto de eventos que evoluem para relações mais sofisticadas entre entidades e seus contextos. Segundo Zhu *et al.* (2014), uma entidade pode ser um objeto, uma pessoa ou uma área. O contexto refere-se a valores correntes e dados específicos que caracterizam o ambiente de uma entidade relevante para interações (ZHU; SHTYKH; JIN, 2014), (LEE; LEE, 2014), (ABOWD *et al.*, 1999). De acordo com esses conceitos, o presente trabalho implementa uma arquitetura usando o modelo da consciência da situação (traduzido para o inglês: *Situation Awareness (SAW)*), proposto, especificamente, para auxiliar na navegação do DV.

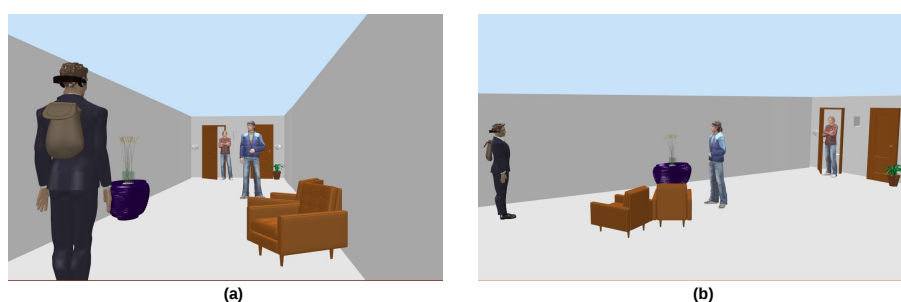
Segundo Endsley *et al.* (2003), o modelo *SAW* pode ser concebido com a execução das seguintes etapas: a percepção dos elementos no ambiente dentro de um volume de tempo e espaço; a compreensão de seu significado e a projeção de seu status no futuro próximo. Esse modelo permite ao usuário compreender o que está acontecendo ao seu redor e quais serão os impactos que ele poderá sofrer, caso realize uma determinada ação. O *SAW*, normalmente, é definido após relacionar as metas e os objetivos que se pretendem atingir em uma determinada situação. Nesta pesquisa, o modelo *SAW* foi aplicado para fornecer informações sobre os obstáculos que estão ao redor do DV e para projetar quais riscos de colisão ele pode ter ao escolher uma determinada direção para se locomover.

A motivação em aplicar o modelo *SAW* é devido ao acúmulo de informações que o DV pode receber a partir de um conjunto de sensores, num curto período de tempo, e ter dificuldades em interpretar e classificar tudo tão rapidamente. Dessa forma, projetou-se a arquitetura *SAS-DV* visando analisar as relações entre entidades detectadas, em determinado ambiente, para compreender a situação e projetar possíveis colisões num futuro próximo (ENDSLEY; BOLTE; JONES, 2003; ANAGNOSTOPOULOS; HADJIEFTHYMIADES, 2009). Se alguma entidade é considerada relevante para analisar os riscos de colisão do DV, então, ela deve ser selecionada para compor uma situação.

A passagem do DV por uma faixa de pedestre é um exemplo de como podem ser propostas algumas situações para o seu tráfego seguro. Nesse cenário, a situação pode ser composta pela relação que as pessoas, carros, bicicletas, semáforos, ruas e placas estabelecem, diretamente, com o objetivo do DV que, no caso, é a travessia da rua com segurança. A locomoção em ambientes desconhecidos que contêm obstáculos e ODs é uma das principais dificuldades com que os DVs devem lidar. Uma variedade de aplicações tem sido produzidas (PHAM; LE; VUILLERME, 2016; IPINA; BOTRÁN; NOVOA, 2011; MEKHALFI *et al.*, 2015; BOURBA-

KIS; MAKROGIANNIS; DAKOPOULOS, 2013; TIAN; HAMEL; TAN, 2014; TAMJIDI; YE; HONG, 2013) para fornecer alternativas de tráfego para o DV. Até o momento, várias alternativas de tráfego foram desenvolvidas para os DVs, só que em áreas específicas. Diferentemente dessas propostas, neste trabalho é priorizado o uso em qualquer área presente em ambientes fechados, com a finalidade de fornecer ao DV um trajeto seguro e livre de colisões, tanto com a presença de ODs quanto OEs.

A Figura 1.1 apresenta um exemplo de ambiente usado para os experimentos realizados durante a pesquisa, nos quais o DV se locomove com a presença de obstáculos e pessoas em movimento. Nas imagens (a) e (b) desta figura, observa-se, no ambiente, a presença de três pessoas, um vaso e duas poltronas. Esta figura visa simular um ambiente frequentado por um DV, onde existem diferentes tipos de obstáculos. Dentre as três pessoas existentes no ambiente, a pessoa que está vestida com uma roupa de cor azul-marinho na imagem (a), é o DV. Ele está com uma mochila nas costas para poder carregar um minicomputador e um o fone de ouvido para emitir informações para navegação.



**Figura 1.1:** (a) Simulação de um ambiente *indoor* contendo o DV e diversos obstáculos. Nesta imagem, o DV é a pessoa que possui uma mochila nas costas e está com uma vestimenta de azul-marinho. (b) Mesmo ambiente da imagem (a) com uma projeção diferente.

Para uma pessoa sem deficiência visual tomar alguma decisão de locomoção, é suficiente analisar diversos tipos de dados disponibilizados pelos seus cinco sentidos. Sendo assim, o ser humano realiza uma análise com as informações mais importantes presentes no ambiente, para compreender e trafegar com segurança e eficiência. Da mesma forma, os DVs necessitam de tecnologias que vão além de apenas indicar o destino desejado. Elas devem detectar padrões sistemáticos, contextualizar os elementos do ambiente e indicar qual ação deve ser tomada para garantir a sua segurança. Baseando-se nessas justificativas, este trabalho apresenta uma nova arquitetura, a partir da Visão Computacional e Fusão de Dados, elaborada para prover ao DV a percepção das entidades em determinado ambiente para, posteriormente, obter a consciência da situação.

Muitos projetos (TANG et al., 2013; ZHU; SHYTYKH; JIN, 2014; LEE; LEE, 2014; ANAG-

NOSTOPOULOS; HADJIEFTHYMIADES, 2009) usam a consciência do contexto para classificar ambientes que são altamente dinâmicos. A consciência do contexto visa habilitar o sistema para perceber, interpretar e reagir às mudanças que ocorrem em determinado local do ambiente (LEE; LEE, 2014), (ANAGNOSTOPOULOS; HADJIEFTHYMIADES, 2009). Neste estudo, a teoria de consciência do contexto é aplicada apenas para estabelecer a compreensão entre entidades e seus contextos para uma implementação mais efetiva do modelo SAW. Desse modo, o SAS-DV visa compreender a relação entre os objetos estáticos, as passagens livres, os ODs e as trajetórias produzidas por ODs, para projetar impactos num futuro próximo. Ressalta-se que este sistema não tem como objetivo substituir qualquer instrumento que o DV usa para se locomover com mais segurança, como uma bengala ou um cão guia. Este sistema visa complementar a compreensão do DV com dados que muitos dispositivos ou acessórios não fornecem.

Para a implementação de uma arquitetura de fusão de dados, o SAS-DV baseou-se no Modelo de Salerno para alto nível de fusão (SALERNO, 2007). Segundo Liggins *et al.* (2008), o Modelo de Salerno incorpora conceitos do modelo *Joint Directors of Laboratories* e do modelo SAW, proposto por Endsley *et al.* (2003). Endsley *et al.* (2003) abordam que o modelo SAW é formado somente quando são executados os três primeiros níveis de um processo de Fusão de Dados, no qual o primeiro nível é a percepção, o segundo a compreensão e o terceiro a projeção. Alkhanifer *et al.* (2014) aplicam o modelo SAW por ser uma abordagem útil para qualquer tipo de sistema de suporte à decisão, porque viabiliza informações de fácil compreensão do ambiente, fornecendo, assim, mais confiança nas ações que serão executadas. No SAS-DV, o nível de percepção atua com a transformação dos dados capturados no ambiente (sensores) em informações mais refinadas. Nesse nível são analisadas as características dos objetos relevantes da cena. A contextualização dessas características fornece os requisitos necessários para os níveis mais altos de abstração do modelo SAW.

Segundo Liggins *et al.* (2008), o nível (percepção) do modelo de Salerno propõe analisar: a existência e a quantidade (*How many?*); a identidade (*What/Who?*) e a cinemática (*Where/When?*). No segundo nível do modelo de Salerno (compreensão), é realizada a análise do comportamento dos objetos, fundamentado em suas ações, intenções, relevância e capacidade. Por fim, o terceiro nível (projeção) prevê ou projeta o impacto num futuro próximo.

Neste trabalho, as informações refinadas produzidas no nível de percepção incluem as posições em três Dimensões (3D) e o tamanho em milímetros das: PLs detectadas; dos OEs; dos ODs e dos trajetos executados por ODs. O nível da compreensão trabalha o entendimento da relação entre os elementos mencionados e o significado de suas ações. O nível da projeção permite projetar uma consequência num futuro próximo baseado nas relações e ações dos ob-

jetos do ambiente (LIGGINS; HALL; LLINAS, 2008), como um aviso indicando que a pessoa poderá colidir com algum objeto.

A projeção de colisões é obtida após a construção de um mapa de riscos, composto por inferências. As inferências são executadas por meio de modelos de aprendizagem, construídos segundo a extração de características de ambientes em que o DV trafega. É importante ressaltar que, para a geração desse mapa, é realizado o mapeamento dos OEs, das PLs e dos trajetos executados por Objetos Dinâmicos em um espaço 3D. Esse mapa permite projetar colisões em diferentes direções.

## 1.2 **Objetivos e aspectos inovadores**

Projetos que desenvolvem soluções tecnológicas de apoio sensorial e navegação para DV têm sido apresentados com o objetivo de auxiliar o DV a descobrir a sua posição, quais elementos estão em seu caminho e qual o local mais seguro para transitar. Na maioria dessas tecnologias têm sido usados diversos tipos de sensores, como os que definem distância, presença, movimentos e cores (CHAN; ENGELKE; ABHAYASINGHE, 2017; MASCETTI et al., 2016; JABNOUN; BENZARTI; AMIRI, 2014; PEI; WANG, 2011; BOURBAKIS; MAKROGIANNIS; DAKOPOULOS, 2013; TIAN; HAMEL; TAN, 2014). Em alguns desses projetos, os sensores mencionados proveem dados para as técnicas de Visão Computacional (MEKHALFI et al., 2015; PHAM; LE; VUILLERME, 2016; XIAO et al., 2015; ANGIN; BHARAT; BHARGAVA, 2011). No entanto, raras vezes são abordados sistemas de Fusão de Dados para integrar todas essas informações e gerar uma tomada de decisão baseada no nível de projeção do modelo *SAW*. Normalmente, esses sensores são usados como fontes de dados, porém, os sistemas não possuem progressões para fases de fusão mais refinadas com a finalidade de corrigir erros, retirar redundâncias e gerar decisões confiáveis.

Este trabalho se enquadra no tipo de pesquisa em que a Fusão de Dados é aplicada como uma técnica fundamental na tomada de decisões. Nesse sentido, foi possível realizar comparações com diferentes técnicas de aprendizagem (sistemas inteligentes, estatísticas e probabilísticas) para verificar quais técnicas seriam as mais indicadas em determinadas situações. Nesse contexto, a fusão é realizada após a integração de todas as características extraídas do módulo de visão para gerar um mapa de riscos composto pelo resultado de um conjunto de inferências. Essas inferências permitem projetar colisões em diferentes direções. É digno de nota que para a geração deste mapa é realizado o mapeamento dos OEs, das PLs, dos ODs e dos seus respectivos trajetos em um plano 3D.

O principal aspecto inovador deste trabalho é a elaboração de uma nova arquitetura, baseada no modelo de Salerno, para a projeção de riscos de colisão. A pesquisa realizada apresenta também outras contribuições, como a implementação de um sistema que detecta OEs, PLs e, principalmente, um outro sistema que detecta ODs e seus trajetos. Para esse último sistema, foram produzidas quatro arquiteturas específicas para o rastreamento de ODs usando diferentes técnicas de detecção. Dessa forma, foi possível realizar nesta tese um outro estudo comparativo para verificar qual arquitetura de detecção e rastreamento de ODs apresentou maior velocidade de detecção, maior eficiência e menor custo computacional.

As técnicas de Visão Computacional e de Fusão de Dados aplicadas no presente estudo foram essenciais para complementar os sentidos de uma pessoa. No entanto, é sempre importante estudar a complexidade dos algoritmos empregados e analisar seu desempenho em diferentes plataformas. Tamjidi *et al.* (2013) afirmam em seu trabalho que uma forma de reduzir erros na navegação é com a adição de sensores e a execução da fusão, porém, afirmam que essa metodologia pode gerar mais peso e aumentar a complexidade do sistema. Segundo o relatório da Organização Mundial de Saúde (MARIOTTI, 2014), 82% das pessoas com deficiências visuais possuem mais de 50 anos de idade. Baseado nessa informação, verificou-se que a estrutura do equipamento de apoio para essas pessoas deve ser leve, para não gerar desconforto, e de pequeno porte, para facilitar a locomoção. Para isso, foram implementados algoritmos contendo um conjunto de técnicas de Visão Computacional, que permitiu obter dados relevantes, sem exigir alto poder de processamento. Ressalta-se que essas escolhas podem viabilizar alternativas para embarcação do sistema em placas de pequeno porte, sem perda de desempenho significativo na geração dos resultados. O emprego das técnicas de Visão Computacional também contribuiu para diminuir a quantidade de sensores físicos. Com isso, é possível diminuir o tamanho, o custo, a energia e o peso do produto final.

Uma outra característica pouco explorada em trabalhos desenvolvidos para o apoio ao DV é a realimentação do sistema para a tomada de decisão. Para o SAS-DV, essa realimentação é muito útil para ajustar e balancear o modelo de aprendizagem no decorrer do seu uso e fornecer um mapa de riscos mais confiável. Estudos como esse também podem ser aplicados em robôs, ou em qualquer tipo de transporte, para fornecer informações auxiliares para locomoção segura.

### **1.3 Organização do Trabalho**

A presente tese está dividida em sete capítulos e um apêndice. O Capítulo 1, como já apresentado, expõe uma breve contextualização dos problemas vivenciados pelo DV e apresenta os

objetivos e os aspectos inovadores do SAS-DV. No Capítulo 2 é realizada uma comparação entre os trabalhos mais recentes relacionados com o estudo, para analisar as possíveis contribuições que o SAS-DV fornecerá aos DVs e para o estado da arte. Essa comparação permitiu definir quais poderiam ser as contribuições deste trabalho, principalmente com relação à área de consciência da situação e da área de detecção e rastreamento de Objetos Dinâmicos. No Capítulo 3 é apresentada em detalhes a metodologia usada no desenvolvimento do SAS-DV, por meio de uma arquitetura composta por métodos e técnicas. No Capítulo 4 são apresentadas em detalhes as arquiteturas desenvolvidas para detecção e rastreamento de objetos dinâmicos. O Capítulo 5 apresenta os experimentos realizados com todos os conceitos abordados nos capítulos 3 e 4. No Capítulo 6 são discutidos os resultados obtidos por meio de comparações realizadas com outras metodologias implementadas, especificamente, para sistemas de navegação para DV e das técnicas e detecção e rastreamento de Objetos Dinâmicos. No capítulo 7 são realizadas as considerações finais. Finalmente, no Apêndice A, são apresentadas as instituições financiadoras, as informações sobre materiais suplementares, as publicações relacionadas com o presente trabalho e os códigos-fontes importantes para a comunidade científica.

# Capítulo 2

## TRABALHOS RELACIONADOS

---

---

O objetivo deste trabalho é fornecer um sistema de apoio à tomada de decisão para a navegação de pessoas que possuem deficiência visual. Dessa forma, para a implementação deste sistema e para contribuir com alguns aspectos inovadores foi realizada uma análise das principais técnicas implementadas em sistemas de navegação. A Tabela 2.1 apresenta, de forma resumida, algumas das técnicas que cada trabalho implementou nos respectivos sistemas de navegação. Por meio dessa tabela foi possível realizar uma análise comparativa entre as técnicas apresentadas e a forma como algumas delas foram implementadas no SAS-DV. Diante disso, nesta seção são abordadas as vantagens e as desvantagens que cada técnica produz no contexto do DV.

Os sistemas desenvolvidos para DV que usam câmeras, geralmente, dependem de técnicas de Visão Computacional e de Reconhecimento de Padrões para obter algum tipo de dado do ambiente. Pode ser observado na Tabela 2.1 que nove projetos mencionados, além do SAS-DV, usam Visão Computacional e sete contando com SAS-DV usam Reconhecimento de Padrões. As técnicas de Visão Computacional podem fornecer uma variedade de informações para o DV independentemente do ambiente. Para obter informações, usando as técnicas de Reconhecimento de Padrões, é exigido um prévio treinamento para classificação e, mesmo assim, pode ser que o sistema não consiga reconhecer todos os objetos de uma determinada classe. Nesse caso, se o sistema se basear apenas na implementação de técnicas de Reconhecimento de Padrões, ele se torna dependente de um amplo treinamento para que garanta a classificação dos objetos desejados e, caso não sejam classificados, os sistemas podem deixar de gerar informações necessárias para a tomada de decisão. No SAS-DV, a técnica de Reconhecimento de Padrões é usada de forma complementar para detectar pessoas, com a finalidade de analisar os trajetos executados por elas. No entanto, outros métodos de detecção e rastreamento de Objetos Dinâmicos também foram implementados usando técnicas da Visão Computacional. Dessa



forma, o SAS-DV não fica dependente da técnica de Reconhecimento de Padrões para obter informações.

**Tabela 2.1: Técnicas usadas em Sistemas de Navegação - Visão Computacional (VC), Mapeamento (MAP), Reconhecimento de Padrões (RP), Conversão 2D para 3D (2D-3D), Análise de Objetos Dinâmicos (AOD), Percepção (PE), Compreensão (CO) e Projeção (PRO)**

	<b>Autores</b>	<b>VC</b>	<b>MAP</b>	<b>RP</b>	<b>2D-3D</b>	<b>AOD</b>	<b>PE</b>	<b>CO</b>	<b>PRO</b>
1	SAS-DV	✓		✓	✓	✓	✓	✓	✓
2	Tsirmpas <i>et al.</i> , 2015		✓						
3	Mekhalfi <i>et al.</i> , 2016	✓	✓	✓					
4	Ipina <i>et al.</i> , 2011			✓					
5	Ando <i>et al.</i> , 2011		✓				✓		
6	Xiao <i>et al.</i> , 2015	✓	✓	✓	✓		✓	✓	
7	Angin <i>et al.</i> , 2011	✓		✓			✓	✓	
8	Pham <i>et al.</i> , 2016	✓		✓	✓				
9	Tapu <i>et al.</i> , 2013	✓		✓	✓	✓			
10	Aladren <i>et al.</i> , 2014								
11	Costa <i>et al.</i> , 2012	✓							
12	Nadia <i>et al.</i> , 2015								
13	Saputra <i>et al.</i> , 2014								

Todos os trabalhos mencionados na Tabela 2.1 usam sensores fixados no DV. Em três desses trabalhos (**2, 3, 4**), alguns dispositivos como sensores, marcadores e *tags* (como *RFID* ou *QR-Code*) foram implantados no ambiente. No trabalho **2** da tabela 2.1, apresentado por Tsirmpas *et al.* (2015), é proposto um sistema de navegação *indoor* que fornece localizações e sugestões para a navegação do deficiente visual. Essa navegação é realizada ao obter dados remotos, por meio de dispositivos denominados etiquetas de *Radio Frequency Identification (RFID)*. Os *RFID* são distribuídos e mapeados no ambiente e fornecem ao deficiente visual informações específicas do local. No trabalho **4**, da mesma tabela, proposto por Ipina *et al.* (2011), é aplicado um conceito semelhante, por meio de *RFID* e *QR-Code*, em que são usados para identificar os produtos de um supermercado. Essa estratégia é interessante para fornecer ao DV a região adequada para trafegar ou para reconhecer objetos, porém, não é interessante para informar possíveis colisões com Objetos Dinâmicos ou com obstáculos que não possuem *RFID*. Outra característica importante que deve ser salientada é a dependência de instalação de várias *RFID* no ambiente. Se não houver *RFID* em determinada região, o sistema passa a não ter informações básicas para tomada de decisão e o sistema perde sua finalidade. Dessa forma, sistemas que usam tecnologias como *RFID*, *QR-Code* ou qualquer outro tipo de símbolo (MEKHALFI *et al.*, 2016; IPINA; BOTRÁN; NOVOA, 2011; TSIRMPAS *et al.*, 2015) devem ser integrados com outros sistemas que complementam as necessidades de locomoção do DV para que não sejam

dependentes apenas de fontes de dados implantadas *a priori*. O SAS-DV foi desenvolvido para não depender de tecnologias implantadas no ambiente e nem de informações solicitadas remotamente.

Na tabela 2.1, são apresentados três itens que são menos explorados entre os sistemas e que melhoram a percepção e a navegação do DV. Os itens abordados são o Mapeamento do Ambiente (MAP), a conversão 2D para 3D (2D-3D) e a Análise de Objetos Dinâmicos. O MAP nem sempre é implementado devido ao alto custo para mapear todas as possíveis regiões em que um DV pode trafegar. De fato, caso o MAP não seja uma técnica complementar do sistema, ele deve ser implementado para todos os ambientes em que um DV pode se locomover (ambientes abertos e fechados). No ambiente aberto, o uso do Global Positioning System (GPS) é essencial para a aplicação, todavia, em ambientes fechados, é necessário criar alternativas devido à possibilidade de obstrução no sinal do GPS. Além do SAS-DV, apenas três trabalhos dessa tabela (6, 8, 9), propostos respectivamente por (XIAO et al., 2015; PHAM; LE; VUILLERME, 2016; TAPU; MOCANU; ZAHARIA, 2013), abordam a conversão 2D para 3D como essencial para uma navegação segura. O plano 2D não fornece medidas reais devido à conversão das coordenadas do mundo para as coordenadas de imagem. Isso pode afetar a navegação do DV. Outra característica pouco abordada nos sistemas de navegação é relacionada com a Análise de Objetos Dinâmicos. Esta técnica pode indicar os possíveis riscos de colisão que o DV pode ter com objetos que estão em movimento, como também fornecer informações úteis para a navegação. É importante destacar que um trajeto executado por uma pessoa pode indicar ao DV regiões que podem ser trafegadas com menor risco de colisão.

Os trabalhos (10, 11, 12, 13), apresentados respectivamente por (ALADREN et al., 2014; COSTA et al., 2012; NADIA et al., ; SAPUTRA; WIDYAWANKA; SANTOSA, 2014), tiveram suas abordagens implementadas para permitir gerar uma comparação com o presente trabalho e destacar as vantagens do modelo de fusão implementado na arquitetura SAS-DV.

As técnicas de Deep Learning também têm sido exploradas para o auxílio à navegação de deficientes visuais. Alguns trabalhos apresentam como é importante fornecer um sistema que classifica objetos. Além disso, esses sistemas são reconhecidos pela alta performance e precisão nos processos de classificação. Os sistemas apresentados por Saleh *et al.* (2019) e Lin *et al.* (2019) podem ser integrados com o SAS-DV, para indicar ao DV vários obstáculos perigosos, o tipo de região onde o DV se locomove, como também fornecer um conjunto de crenças ao *dataset* e produzir mais confiabilidade ao sistema de fusão.

Como a arquitetura apresentada neste trabalho pretende contribuir com melhorias na locomoção dos DV, sugere-se a implementação do modelo SAW. O referido modelo foi adaptado

para a navegação de DV e, com a implementação de um sistema de detecção e rastreamento de objetos dinâmicos, esta seção (trabalhos relacionados) é composta por duas subseções (Consciência da situação e Detecção e Rastreamento de Objetos Dinâmicos) para apresentar os trabalhos desenvolvidos até o momento e realçar as contribuições específicas das técnicas implementadas.

## 2.1 Consciência da situação (*Situation Awareness*)

As técnicas de Fusão de Dados têm sido aplicadas em diversos projetos, entretanto, foram pouco exploradas em sistemas de suporte à navegação e análise sensorial para os DVs. O nível dessa exploração diminui, substancialmente, quando a fusão chega nos níveis de compreensão e de projeção (ver tabela 2.1) do modelo *SAW*. Esses níveis têm como finalidade explorar os dados em busca de relacionamentos consistentes e, conseqüentemente, projetar impactos num futuro próximo. Como já mencionado (na Seção 1), o modelo *SAW* é composto por três níveis: percepção, compreensão e projeção. No SAS-DV, o primeiro nível (percepção) do modelo *SAW* é encarregado de adquirir e organizar as informações de alto nível para formar um conjunto de dados (*datasets*). No segundo nível do modelo *SAW* (compreensão) são definidas as relações e as crenças entre as informações disponibilizadas pelo primeiro nível (percepção). Ainda na compreensão, é realizado também o treinamento do *dataset* com o uso de um classificador e, conseqüentemente, obtém-se um modelo de aprendizagem. Ressalte-se que o modelo de aprendizagem pode ser gerado por meio de diferentes classificadores. A fusão atinge o terceiro nível (projeção) do modelo *SAW* com a geração do mapa de riscos de colisão que, neste caso, é formado por um conjunto de inferências. No SAS-DV, o modelo *SAW* fornece flexibilidade e fácil adaptação para compreender o ambiente em que o DV trafega usando técnicas de visão computacional para extrair dados. Ao aplicar este conceito, permite-se que o sistema não seja dependente de um ambiente específico, o que pode prejudicar o tráfego do DV em diferentes locais.

Os trabalhos (5, 6, 7) da Tabela 2.1, apresentados respectivamente por Ando *et al.* (2011), Xiao *et al.* (2015) e Angin *et al.* (2011), abordam como a fusão de dados foi implementada em seus sistemas. Ando *et al.* (2011) apresentam uma metodologia interessante com o emprego da Fusão de Dados em um sistema de apoio à navegação do DV. Seu objetivo é proporcionar uma comunicação contínua do DV com uma rede de sensores distribuídos pelo ambiente para prover uma locomoção sem colisões. Essa abordagem é apresentada como uma boa estratégia para ser implantada em locais públicos e fechados como escolas, bibliotecas e museus. O dispositivo embarcado no DV, os sensores e a central de processamento são conectados por uma rede sem

fi. A comunicação com o usuário é realizada por meio de sons. Nesse projeto, não foi aplicado um conjunto de técnicas de Visão Computacional para fornecer características, o que implica a necessidade de instalar um conjunto de sensores físicos pelos locais que o DV trafegará.

Xiao *et al.* (2015) apresentam um sistema que adota o sensor de movimentos Kinect para obtenção de mapa de disparidade e abordam o uso de sensores sociais para provimento de dados em sites e redes sociais que ofereçam o suporte ao DV, para descobrir se há perigo de locomoção em uma determinada região. Com essas fontes de dados é possível gerar a consciência da situação (SAW), o que permite produzir uma informação com maior confiabilidade. Entretanto, esse sistema depende da alimentação de dados providos por outras pessoas. Dessa forma, os sistemas de Xiao *et al.* (2015) e Ando *et al.* (2011) possuem uma alta dependência de sensores externos ao DV para realização da fusão de dados. No presente trabalho, a fusão é realizada somente de acordo com dados fornecidos pelo dispositivo Kinect e pelas técnicas de visão computacional. Essa arquitetura não depende de nenhum sensor externo e possui um desempenho satisfatório para obter informações do ambiente e realizar inferências relacionadas à segurança dos DVs.

Angin *et al.* (2011) abordam o uso do modelo CA para apoiar a navegação do DV. Os autores relacionam várias fontes de dados para fornecer uma tomada de decisão ao DV. Nesse trabalho é aplicado um GPS para estimar o posicionamento e uma câmera estéreo para detectar a distância dos objetos. O sistema também realiza a operação de reconhecimento de padrões para disponibilizar mais elementos para o seu sistema de fusão. Porém, não é apresentado se houve a aplicação de técnicas que permitam compreender padrões sistemáticos de contextos, situações ou eventos que forneçam informações mais confiáveis. Dessa forma, não foi possível encontrar uma fusão acima do nível da compreensão do modelo SAW.

No contexto de sistemas de apoio ao DV, não foram encontrados trabalhos que possuam uma arquitetura contendo técnicas de visão computacional e de aprendizado de máquinas embarcadas, que estabeleçam uma fusão em um nível que projete colisões entre o DV e elementos do ambiente. Uma característica dessa pesquisa é fornecer ao DV a possibilidade de realizar um conjunto de inferências que produza um mapa com probabilidades, em diferentes direções, que indique o risco de colisão e de tráfego na direção da passagem livre. A fusão atinge o nível de projeção e a busca por padrões é efetuada por meio de regras de associações. Observa-se que as comparações com as técnicas de classificação, aplicadas nesta pesquisa, podem servir como instrumentos de orientação na escolha da técnica mais adequada para ser adotada em determinada situação.

O aspecto inovador da arquitetura SAS-DV é o estudo e aplicação da fusão de alto nível por

meio dos modelos de Salerno (LIGGINS; HALL; LLINAS, 2008; SALERNO, 2007) e de CA (ANDO et al., 2011; ANGIN; BHARAT; BHARGAVA, 2011; ABOWD et al., 1999), desenvolvidos especificamente para projetar colisões e apoiar a tomada de decisão na navegação de DV em ambientes *indoor*. Esses modelos se apresentam como fundamentais para compreender contextos, situações e providenciar uma tomada de decisão, sem depender de uma variedade de sensores, tanto implantadas no DV, como no ambiente. Apesar desses modelos já terem sido aplicados, é notório que ainda não foram muito explorados em sistemas de navegação e apoio sensorial para os deficientes visuais, principalmente devido às diversas situações existentes nos contextos dos DVs. Provavelmente, isso se deve à complexidade na extração de características devido à implantação de poucos sensores no DV, justificado pelo tamanho e peso que prejudicam o seu conforto. A compreensão do ambiente e de determinadas situações também possui muitas dificuldades devido ao movimento da câmera, como abordado por Chan *et al.* (2017). Outra característica pouco encontrada se refere ao nível de projeção de impactos que o DV pode ter. Isso pode ser justificado pela pouca exploração de situações, que envolvem a análise de comportamentos dos DVs e dos objetos presentes em determinados contextos.

## 2.2 Detecção e rastreamento de objetos dinâmicos (DROD)

Dificilmente é abordada, em tecnologias de apoio ao DV, uma forma específica para analisar o comportamento dos Objetos Dinâmicos e reconstruir os seus trajetos, com intuito de indicar as regiões mais confiáveis para tráfego. No desenvolvimento da arquitetura SAS-DV, foi observado que a DROD pode disponibilizar informações relevantes para o nível projeção do modelo SAW. Para isso, foram implementadas as técnicas DROD, mais abordadas no estado da arte, que, em sua maioria, implementam métodos de subtração de fundo (difundido como subtração de *background*) (WANG et al., 2015; YANG et al., 2017; STAUFFER; GRIMSON, 1999) e de fluxo óptico (HUANG et al., 2018; GANG; XIAOLI; LIRONG, 2012). Essas técnicas são aplicadas em diferentes áreas de atuação, como sistemas usados em: veículos autônomos, robôs, segurança e monitoramento e rastreamento. No entanto, elas devem ser adaptadas para o contexto do DV. Essas adaptações devem ser realizadas devido ao sensor ser instalado em um corpo que realiza movimentos. Diante do exposto, este trabalho teve como desafio detectar ODs mesmo quando o sensor de aquisição sofre algum tipo de movimento. Outras adaptações foram criadas quanto a mudanças de iluminação.

Grande parte dos trabalhos relacionados com detecção de Objetos Dinâmicos tem se preocupado mais especificamente com a qualidade de segmentação (WANG et al., 2015; HUANG et al., 2018; YANG et al., 2017) do objeto dinâmico detectado, e não se preocupa tanto com

a velocidade de detecção e sua efetividade quando o SA sofre algum tipo de movimentação, quando ocorrem fortes mudanças de iluminação e quando não existe iluminação.

Apesar das adaptações citadas, uma das principais contribuições desta tese é a metodologia proposta para analisar a velocidade de detecção e a efetividade de qualquer técnica implementada para a DROD, mesmo com o sensor sofrendo algum tipo de movimento. E, para que essa metodologia seja usada por qualquer outro método de detecção de Objetos Dinâmicos para sistemas específicos para DV, são apresentados na Seção 3.2.3 (Sistemas de objetos dinâmicos) todos os métodos que devem ser implementados para que a metodologia proposta possa analisar qual técnica fornece o melhor desempenho no contexto em que será aplicado. Entre esses métodos, são apresentados como mapear as posições (rastreamento) do OD num plano 3D e como analisar a velocidade de detecção e a efetividade de cada técnica de DROD.

Malireddi *et al.* (2016) apresentam um sistema de segmentação de objetos dinâmicos, baseado em técnicas de geometria computacional, que calcula o movimento do objeto presente em um par de imagens. Essa abordagem possui a premissa de que o objeto a ser segmentado deve ser capturado por duas posições diferentes. Nesse trabalho, os autores garantem a segmentação mesmo com a presença de movimentos na câmera, porém, esses movimentos não podem ser intensos. É uma abordagem interessante, mas, ao implantar esse sistema, é preciso se preocupar com custo computacional exigido em um contexto que necessita de tomadas de decisão em tempo real. Um sistema estéreo somado às técnicas de geometria computacional e de segmentação em pares de imagens, instantaneamente, pode comprometer o sistema de tomada de decisão. Essa mesma preocupação deve ser explorada em sistemas que realizam o reconhecimento de padrões.

Choi *et al.* (2011) descrevem um sistema que realiza o Reconhecimento de Padrões para posteriormente rastrear o objeto. As técnicas de Reconhecimento de Padrões têm sido uma das alternativas de rastreamento de alvo mais usadas em sistemas de análise de objetos dinâmicos. É digno de nota que o custo computacional para reconhecer o padrão (quadro a quadro) é muito alto. O método proposto por Viola *et al.* (2001), baseado em *Haar feature-based cascade classifiers*, foi a primeira técnica implementada neste trabalho. Todos os sistemas que usam Reconhecimento de Padrões podem comprometer o desempenho do sistema de detecção e rastreamento, devido ao processo de reconhecimento ser em tempo real. Esse tipo de sistema necessita de um treinamento supervisionado exaustivo para que garanta o reconhecimento desejado. Se o sistema não reconhecer determinada entidade, a segmentação do OD não é realizada e pode comprometer a tomada de decisão do DV.

Bugeau *et al.* (2009) consideram uma técnica de detecção de Objetos Dinâmicos indepen-

dente de técnicas de Reconhecimento de Padrões. Eles detectam pontos de referências de forma arbitrária e usam a técnica de fluxo óptico para analisar as mudanças. Após detectados os pontos que obtiveram mudanças efetivas, eles aplicaram o *Mean Shift Clustering Algorithm* para segmentar e rastrear a região de interesse. Seguindo a mesma estratégia dessa abordagem, neste trabalho também foi aplicado um conjunto de pontos de forma arbitrária para a análise de movimentos na imagem, independentemente do movimento de câmera. Dessa forma, uma segunda técnica foi adaptada para o DV. Os pontos se basearam na técnica de Fluxo Óptico de Farneback (FOF) e as premissas de mudanças se fundamentaram na diferença entre o deslocamento de uma região comparada com a média de deslocamento da imagem total.

Stauffer *et al.* (1999) apresentam uma técnica de alto desempenho na detecção de Objetos Dinâmicos usando a técnica de subtração de *background* implementada com a *Mixture of Gaussians (MOG)*. Essa técnica foi a terceira escolhida para ser implementada neste trabalho, para compor o estudo comparativo entre as abordagens e verificar qual seria a mais indicada para um sistema com tomadas de decisão em tempo real. É relevante destacar que, para essa técnica fornecer um resultado de segmentação com qualidade, ela exige que a câmera esteja estática. Outro problema recorrente dessa técnica está relacionado com a mudança de iluminação quando aplicada em imagens provenientes de uma câmera RGB. Caso uma luz apague ou acenda, ela detecta a região como sendo um OD. Baseado nos conceitos e dificuldades abordados por essa técnica, esta tese contribui com a implementação de uma adaptação para estabilizar o vídeo para posterior aplicação da técnica de Subtração de *Background*, mesmo com a presença de movimentos na câmera. Esse método é aplicado especificamente em uma imagem formada por sensores infravermelhos. Dessa forma, a imagem não sofre qualquer dano com a alternância da luz em ambientes internos e, conseqüentemente, não prejudica na análise de Objetos Dinâmicos.

A quarta técnica implementada no presente trabalho estabelece relação com a abordagem apresentada por Bugeau *et al.* (2009). Com a detecção de um objeto em movimento, neste caso, usando o Fluxo Óptico de Farneback, foi definido o seu centro de massa para selecionar as cores da vizinhança. Isso permitiu que a técnica *Continuously Adaptive Meanshift (CamShift)*, que possui a mesma finalidade que a aplicada (*Mean Shift Clustering Algorithm*) por Bugeau *et al.* (2009), permitisse o rastreamento do OD e possibilitasse a reconstrução do seu trajeto. Com a CamShift (apresentada por Bradski *et al.* (2008)) foi possível aumentar o desempenho do sistema e ainda não necessitar ou depender de técnicas de Reconhecimento de Padrões para rastreamento de alvos. Vale destacar que essa técnica pode ser aplicada apenas nos casos em que a iluminação não sofra alterações rígidas, pois a região definida para rastreamento pode ser alterada no decorrer do processo.

Diante das abordagens apresentadas, foram implementadas nesta pesquisa quatro arquiteturas de DROD, com o uso das técnicas mencionadas e customizadas para o contexto do DV. No contexto deste trabalho, os sensores (câmeras RGB e IV) estão sempre sofrendo impactos devido aos movimentos produzidos pelo DV e o sistema precisa gerar respostas próximas do tempo real, mesmo com as adversidades do ambiente.

Nesta tese, são apresentadas algumas contribuições baseadas nas dificuldades encontradas na realização dos experimentos com as arquiteturas implementadas para DROD. Entre essas contribuições é possível citar qual técnica se apresenta melhor no processo de detecção e rastreamento, com ausência e presença de movimentos, no sensor de aquisição, mudanças de iluminação e, inclusive, estratégias de fusão entre as técnicas apresentadas para solucionar situações específicas de DROD.



# Capítulo 3

## SISTEMA DE ANÁLISE SENSORIAL PARA DEFICIENTES VISUAIS (SAS-DV)

---

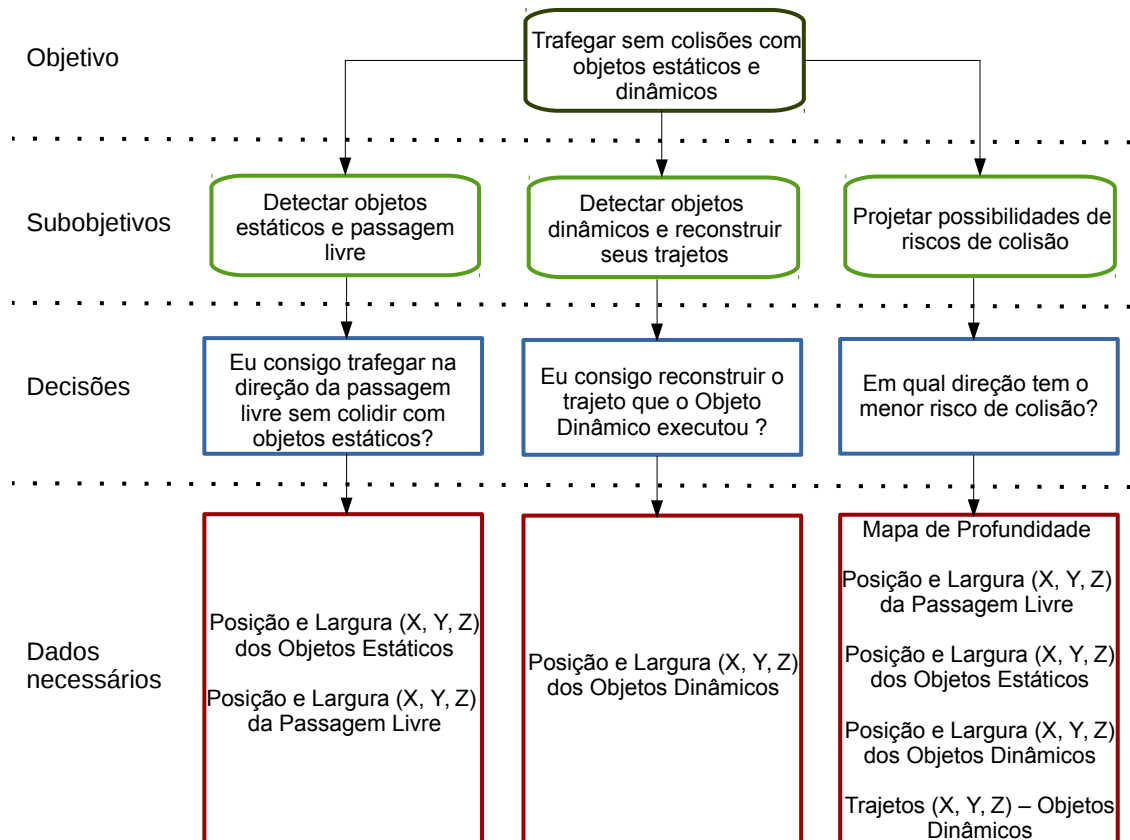
---

O SAS-DV tem como finalidade disponibilizar um sistema que auxilie na navegação do Deficiente Visual, de modo que indique a presença de objetos estáticos e dinâmicos, para informar a direção da passagem livre e a direção que possui o menor risco de colisão. Este sistema é composto por técnicas de Visão Computacional e de Fusão de Dados. As técnicas de visão obtêm um conjunto de características do ambiente, e as técnicas de fusão produzem associações consistentes baseadas no modelo de Salerno.

Para o desenvolvimento de um sistema de fusão com o emprego do modelo *SAW*, é importante que seja realizada uma análise de requisitos capaz de fornecer a percepção, a compreensão e a projeção. Segundo Endsley *et al.* (2003), a *Goal Directed Task Analysis (GDTA)* realiza a análise de tarefas cognitivas com o objetivo de relacionar todas as entidades e suas respectivas funções, assegurando uma tomada de decisão confiável.

A *GDTA* permite estimular o conhecimento do usuário para determinadas tarefas e produzir uma análise com a definição dos objetivos de modo eficaz. Ela é composta pelo objetivo principal do usuário, seus subobjetivos, as questões que devem ser solucionadas e as informações necessárias para as tomadas de decisão. Nessa pesquisa a *GDTA* foi desenvolvida baseada em entrevistas realizadas com dois professores deficientes visuais. As principais dificuldades mencionadas foram registradas por meio dos seguintes questionamentos: quais elementos estão presentes nesse ambiente? Posso trafegar em qual direção? Existe espaço suficiente para trafegar nessa direção? Qual a posição dos obstáculos que não produzem sons? Quais objetos estão se locomovendo? Qual é a sua direção, velocidade e tamanho? Eu posso colidir com algum objeto (estático ou dinâmico)? Após registrar esses questionamentos, definiram-se os objetivos do sistema, relacionados a seguir:

Definir direções alternativas para a locomoção do DV; informar quais são as regiões com possibilidades de ocorrer alguma colisão; alertar quais objetos estão em movimento, considerando a sua distância, direção, velocidade e área; projetar mapas com riscos de colisão em diferentes direções baseadas em obstáculos, direção de passagem livre e nos movimentos de objetos dinâmicos. Os objetivos e as entidades que foram fundamentais para o desenvolvimento do sistema de fusão de dados com o emprego do modelo *SAW* são apresentados na Figura 3.1.



**Figura 3.1:** Uma análise dos objetivos do usuário usando a teoria *GDTA*.

Presente no modelo de Salerno, o modelo *SAW* necessita obter a percepção dos elementos em um ambiente dentro de um determinado tempo e espaço, a compreensão do seu significado e a projeção de suas ações em um futuro próximo (LIGGINS; HALL; LLINAS, 2008). O sistema de fusão do SAS-DV permite a aprendizagem a partir de diferentes classificadores. Para que isso ocorra, um conjunto de dados (*dataset*) foi concebido a partir da detecção de características do ambiente, realizado pelo sistema de visão. Obtidas essas características, definiram-se os respectivos riscos de colisão de forma supervisionada. Com um *dataset* balanceado, foi possível aplicar inferências para cada direção em que o DV pode trafegar e projetar possíveis impactos do DV, com obstáculos presentes nos ambientes do tipo *indoor*.

Após alguns experimentos com o dispositivo Kinect, foi possível verificar que, aliado às

técnicas de Visão Computacional, o sistema fornece uma variedade de características do ambiente e disponibiliza as informações exigidas pela teoria *GDTA*. Dessa forma, elaborou-se uma nova arquitetura composta por módulos, submódulos e sistemas. Esta arquitetura (Figura 3.2) é formada por três módulos: Módulo de Entrada e Saída (E/S) (Seção 3.1), Módulo de Visão (Seção 3.2) e Módulo de Fusão (Seção 3.3). É importante ressaltar que o fluxo de dados dessa arquitetura foi baseado em modelos de Salerno. Esse fluxo de dados tem como finalidade refinar todas as características definidas pela *GDTA* e então disponibilizar os dados requisitados pelos submódulos de percepção, compreensão e projeção do Módulo de Fusão (Seção 3.3).

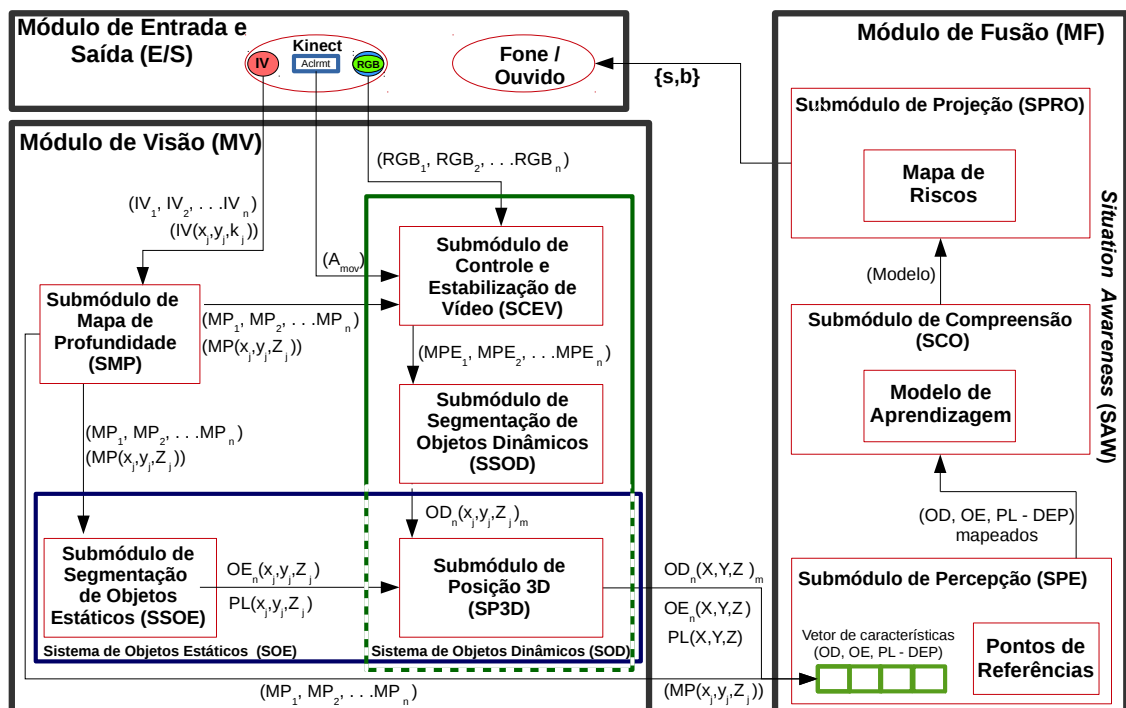


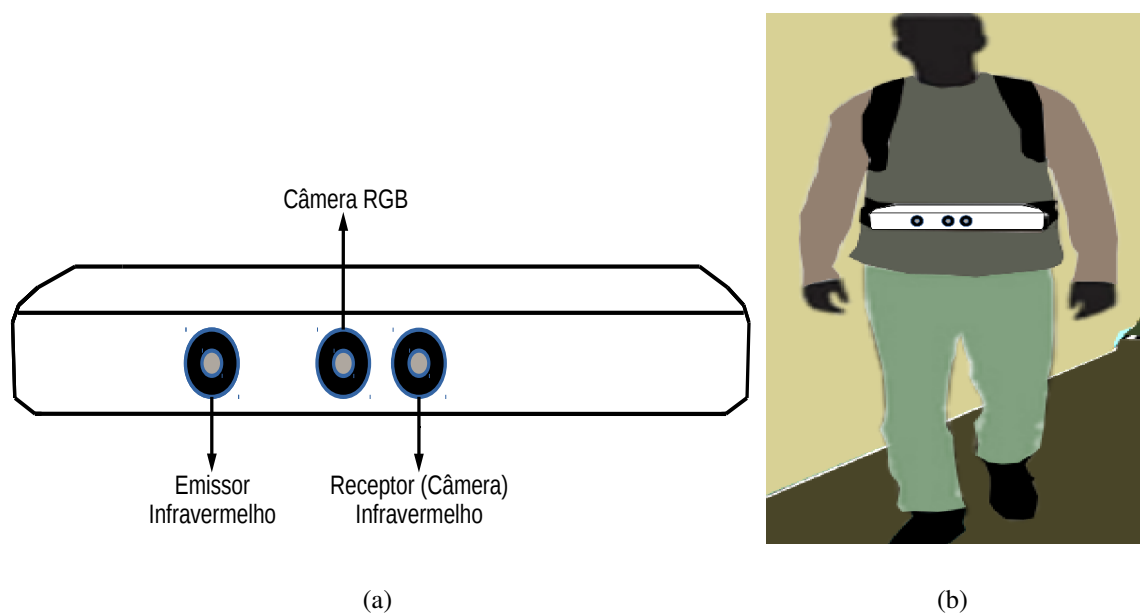
Figura 3.2: Arquitetura (SAS-DV) desenvolvida neste trabalho.

### 3.1 Módulo de Entrada e Saída (E/S)

O Módulo de Entrada e Saída do SAS-DV (Figura 3.2) é composto por um dispositivo Kinect (entrada) e um fone de ouvido estéreo (saída). O Kinect (ver imagem (a) da Figura 3.3) possui, basicamente, cinco componentes, sendo eles: um acelerômetro presente no interior do dispositivo, um emissor infravermelho, duas câmeras de vídeo (uma infravermelha (IV) e outra RGB) com resolução VGA de 30 quadros/s e um microfone. O objetivo da câmera infravermelha é fornecer quadros  $(IV_i, i = 1..30)$  ao Módulo de Visão (MV), em tempo real, contendo mapas de profundidade da cena, que serão utilizados para a detecção de objetos estáticos e dinâmicos presentes no ambiente em que o deficiente está. A câmera RGB fornece quadros

( $RGB_i, i = 1..30$ ) no espectro visível ao MV que, associado com o acelerômetro, realizarão um processo de estabilização de vídeo para o Submódulo de Segmentação de Objetos Dinâmicos (SSOD), a ser explicado no capítulo (4). Por fim, no Módulo de E/S da presente arquitetura, há um *feedback* ao DV por meio de sons (s) ou bipes (b), fornecido pelo Módulo de Fusão. Este *feedback* é realizado por meio de um fone de ouvido que fornece indicações e alertas constantes sobre as características do ambiente, principalmente dos objetos que estão em primeiro plano. Com o uso do fone também é possível informar ao DV as probabilidades de riscos de colisão para diferentes direções. Essas probabilidades são definidas após a fusão da detecção de objetos estáticos e dinâmicos, da detecção da direção de passagem e da reconstrução de trajetos realizados por pessoas.

Para que todos esses sensores sejam utilizados adequadamente pelo DV, é necessário usar uma cinta específica para sustentar o Kinect na região abdominal e usar uma mochila para carregar um computador de pequeno porte e a bateria que alimenta o Kinect. Essa cinta permite ajustar a altura do dispositivo Kinect. Na imagem (b) da Figura 3.3 é possível ver a posição em que o dispositivo Kinect foi usado nos experimentos desse trabalho.



**Figura 3.3:** (a) O dispositivo Kinect e seus principais sensores visíveis; (b) A posição em que o dispositivo Kinect foi usado em uma pessoa neste trabalho.

O presente trabalho foi desenvolvido especificamente para ambientes internos, porém, a arquitetura proposta não impossibilita a sua adaptação para ambientes externos. Neste estudo, usou-se apenas o dispositivo Kinect nos experimentos. Em trabalhos futuros, uma câmera estéreo pode ser implantada e as mesmas técnicas implementadas no SAS-DV podem ser replicadas. A câmera estéreo ZED, usada no trabalho de Adi *et al.* (2019), implementa os princípios

da visão humana (triangulação) e é uma opção para a arquitetura SAS-DV ser adaptada para ambientes externos.

## 3.2 Módulo de Visão

O Módulo de Visão é formado por cinco submódulos que, juntos, são responsáveis pela extração de características, tais como posição 3D dos objetos estáticos, dos objetos dinâmicos e da passagem livre. Na Figura 3.2, é possível observar a existência dos sistemas de objetos estáticos e de objetos dinâmicos. Esses sistemas fazem a integração de alguns submódulos e seus detalhes serão explicados nas próximas seções. Considera-se como objeto estático qualquer objeto (cadeira, mesa, lixo, extintor) próximo e localizado na direção em que o DV trafega. Os objetos dinâmicos mais comuns presentes em áreas internas são as pessoas, os animais domésticos e objetos que possuem rodas. Entre as características extraídas dos objetos estáticos estão a posição, a distância e a área. Para os objetos dinâmicos, o Módulo de Visão realiza a extração dessas mesmas características associadas com o centro de massa e com os vetores de velocidade que fornecem as posições para estimar a velocidade média e a direção do objeto dinâmico. Este módulo tem como finalidade submeter as características ao Módulo de Fusão, para gerar um mapa que projeta os riscos de colisão em diferentes direções.

### 3.2.1 Submódulo de Mapa de Profundidade

O Submódulo de Mapa de Profundidade é o responsável por retornar a distância de qualquer coordenada de imagem (x,y) desejada. O sensor IV do dispositivo Kinect fornece um mapa de profundidade (em cores) no qual as distâncias são convertidas em intensidades (canais RGB). Dessa forma, verificando os valores de intensidade em cada canal RGB, é possível determinar a distância de cada objeto no ambiente. Dessa maneira, criou-se um método que converte essas intensidades em milímetros, baseando-se em uma tabela de referência. Nesse processo, implementou-se um método que retorna todas as distâncias em milímetros. Essas distâncias são essenciais para converter qualquer coordenada 2D para um plano tridimensional. Por meio desse mapa de distâncias é possível segmentar os objetos mais próximos, mais distantes e inclusive os objetos em movimento.

Nesse submódulo, dados os quadros ( $IV_i$ ,  $i=1..30$ ), provenientes do sensor IV do Kinect, seus mapas de cores referentes às distâncias 3D dos objetos da cena são convertidos em matrizes apropriadas ( $MP_i$ ,  $i=1..30$ ) com valores em milímetros associados a essas distâncias  $(x_j, y_j, k_j)$ , em que  $(x_j, y_j)$  é a distância do j-ésimo valor de intensidade  $k_j$  em relação ao DV, que serão

utilizados para o cálculo da posição 3D de cada objeto da cena ( $X_j, Y_j, Z_j$ ) (Equação 3.2), a ser utilizado nos submódulos de segmentação de objetos estáticos e dinâmicos (SSOE e SSOD), a serem descritos no item 3.2.2.1 e no capítulo 4, respectivamente.

## 3.2.2 Sistema de Objetos Estáticos

O Sistema de Objetos Estáticos é formado pelos Submódulos de Segmentação de Objetos Estáticos (Seção 3.2.2.1) e de Posição 3D, (Seção 3.2.2.2). Eles permitem extrair características dos objetos estáticos que estão até a 2 metros de distância do DV e mapeá-los no plano tridimensional. Essas características são fundamentais para o processo de projeção de impactos e consequente formação do mapa de riscos.

### 3.2.2.1 Submódulo de Segmentação de Objetos Estáticos

O Submódulo de Segmentação de Objetos Estáticos (SSOE) tem como finalidade segmentar os objetos estáticos presentes nas cenas, a menos de 2 metros de distância do DV, com o objetivo de evitar colisões. Por essa via, é fornecida a direção de passagem livre de obstáculos. A segmentação desses objetos estáticos é realizada por meio de um Threshold ( $SSOE_{oe}$ ) que percorre a imagem de profundidade ( $MP(x_j, y_j, Z_j)$ ) e elimina o seu *background*, respeitando a premissa de que  $Z_j$  é maior que 2 metros (Algoritmo 1). Esse filtro ( $SSOE_{oe}$ ) possibilita trabalhar apenas com os obstáculos que estão no *foreground* e que geram maiores riscos de impacto num curto período.

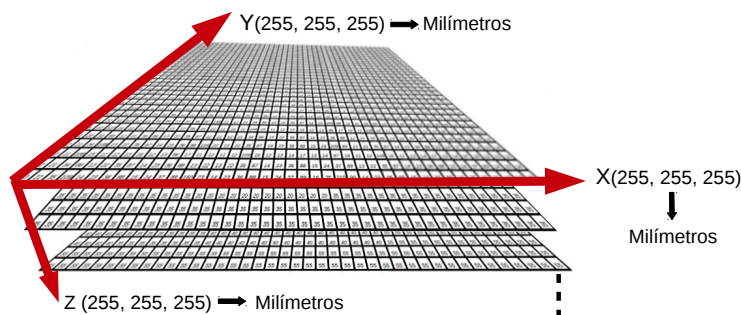
No SSOE também é detectada a posição da passagem livre. Para descobrir qual é a passagem livre, aplicou-se um Threshold ( $SSOE_{pl}$ ) na imagem de profundidade ( $MP(x_j, y_j, Z_j)$ ), respeitando a premissa de que  $Z_j$  é maior que 3 metros. O SSOE é essencial para realizar o cálculo da área da região que apresenta risco de colisão ou passagem livre.

### 3.2.2.2 Submódulo de Posição 3D

A aquisição de dados usando apenas uma câmera RGB gera algumas restrições devido à complexidade relacionada com a movimentação de objetos num ambiente 3D. O cálculo da largura, da altura e da posição absoluta dos objetos estáticos e dinâmicos são algumas dessas restrições. Essas restrições apresentam a necessidade de usar uma câmera IV.

O Submódulo de Posição 3D (SP3D) é responsável por converter o sistema de referência formado em *pixel* pela câmera RGB ( $x, y$ ) e milímetros pela câmera IV ( $Z$ ) para um sistema de

referência (X, Y, Z) baseado apenas em milímetros. O SP3D também é responsável por mapear a posição e a largura da passagem livre, dos objetos estáticos e dinâmicos. A Figura 3.4 ilustra o sistema de coordenadas que o SAS-DV aplica para mapear todos os objetos relevantes do ambiente.



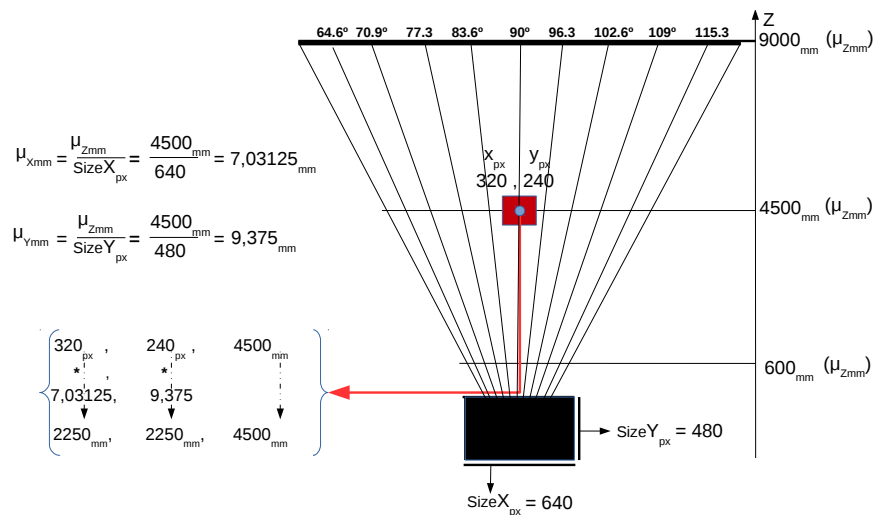
**Figura 3.4: Sistema de referência.**

Esse mapeamento é realizado em três dimensões usando o mesmo sistema de referência (milímetros), a partir dos seguintes parâmetros: Passagem Livre ( $PL(x,y,Z)$ ), Objeto Estático ( $OE_n(x_j,y_j,Z_j)$ ) e Objeto Dinâmico ( $OD_n(x_j,y_j,Z_j)_m$ ). A conversão de um *pixel* presente no plano 2D para o 3D é realizada por meio de medições que correspondem ao tamanho do *pixel* ( $x_j,y_j$ ) na distância ( $Z_j$ ) em que ele se encontra. Ressalte-se que a distância foi convertida para milímetros ( $Z$ ) no item 3.2.1. Todos os objetos relevantes tiveram suas posições e larguras convertidas de *pixels* para milímetros usando as seguintes equações 3.1, 3.2):

$$\mu_{X_{mm}} = x_{px} * \frac{\mu_{Z_{mm}}}{SizeX_{px}} \quad (3.1)$$

$$\mu_{Y_{mm}} = y_{px} * \frac{\mu_{Z_{mm}}}{SizeY_{px}} \quad (3.2)$$

As variáveis  $x_{px}$  e  $y_{px}$  representam a posição do objeto nas coordenadas de imagem (*pixel*) no eixo x e no eixo y, respectivamente. A variável  $\mu_{Z_{mm}}$  é a distância do objeto em milímetros e,  $SizeX_{px}$  e  $SizeY_{px}$  representam a quantidade de *pixels* da imagem nos respectivos eixos (640 x 480). Dessa forma, calcula-se a posição do objeto em milímetros no eixo x ( $\mu_{X_{mm}}$ ) e no eixo y ( $\mu_{Y_{mm}}$ ). A Figura 3.5 apresenta uma forma mais compreensível das Equações 3.1 e 3.2. É possível observar nesta figura que as variáveis  $\mu_{X_{mm}}$  e  $\mu_{Y_{mm}}$  representam o tamanho do *pixel* em milímetros na distância de 4.5 metros. Dessa forma, como o objeto está no centro, a soma dos 320 *pixels* formaram o valor de 2250 milímetros no eixo X. Quando a distância do objeto é diferente, o sistema deve aplicar as equações (3.1 e 3.2) para calcular novamente o tamanho do milímetro relativo a sua distância, para permitir calcular a posição independente do eixo.



**Figura 3.5:** Uma simples representação de como aplicar as equações 3.1 e 3.2.

Com a adoção desse sistema de referência, é possível mapear a posição de qualquer objeto da cena, sem considerar a distância em que ele se encontra. A direção e a velocidade média dos objetos dinâmicos são outras características importantes que podem ser estimadas no plano 3D. É importante destacar a impossibilidade de calcular corretamente os movimentos de um objeto dinâmico no Plano 2D. Outra característica importante a ser destacada é com relação à variação da largura e área do objeto que, com a ausência do eixo Z, é possível obter apenas um valor relativo.

No SP3D, a posição e a largura dos obstáculos são calculadas a partir do método que produz contorno (apresentado por Bradski *et al.* (2008)) aplicado na imagem resultante do SSOE<sub>so</sub> ( $OE_n(x_j, y_j, Z_j)$ ). Para que esse contorno seja executado da maneira esperada, a imagem resultante ( $OE_n(x_j, y_j, Z_j)$ ) passou pelo Detector de Bordas Canny (CANNY, 1986).

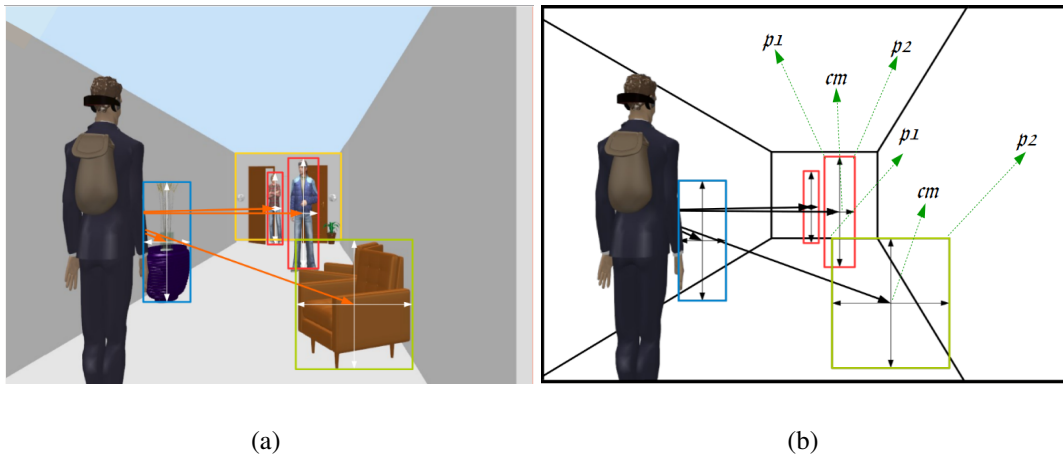
Dessa forma, os contornos são gerados por meio do armazenamento de pontos sequenciais detectados nas bordas. Com os contornos produzidos, é possível verificar a presença de regiões delimitadas por uma curva ou um contorno fechado. Quando isso ocorre, permite-se calcular sua área baseado no teorema de Green (DAWKINS, 2013).

Após a produção dos contornos dos objetos, foram obtidos também os seus cantos (x,y). Sabendo a posição desses cantos é possível calcular a largura e altura do objeto e converter esses pontos em um plano 3D, o que permite o cálculo da largura e da altura absoluta do objeto. Essas características são enviadas ao Submódulo de Percepção do Módulo de Fusão.

Os valores  $x_j$  e  $y_j$  são convertidos para milímetros baseados nas equações (3.1, 3.2). Sendo assim, a função  $CMW_f$  do Algoritmo 1 fornece três pontos para cada objeto estático detectado:



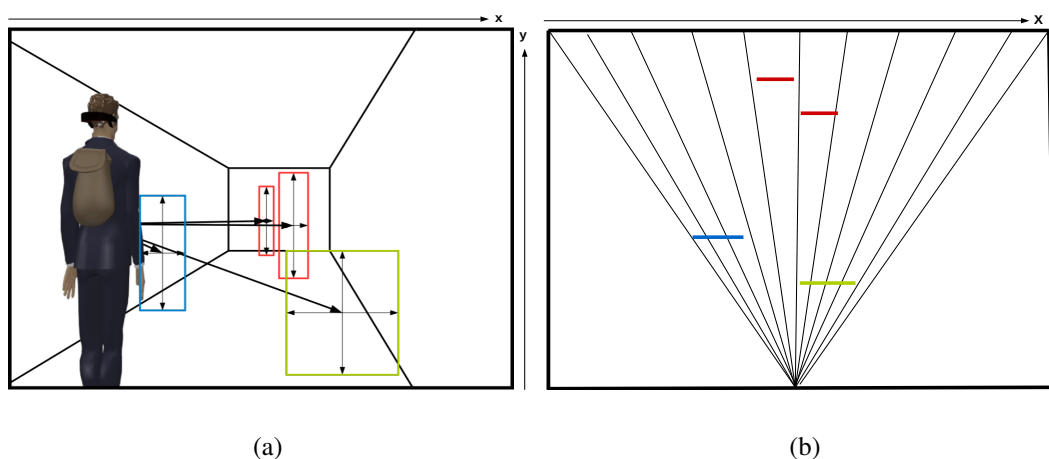
$cm(x_j, y_j, Z_j)$ ,  $p1(x_j, y_j, Z_j)$  e  $p2(x_j, y_j, Z_j)$  (Ver Figura 3.6). O  $cm$  é o centro de massa do contorno calculado sobre o objeto detectado. O  $p1$  indica o canto superior esquerdo do contorno e o  $p2$  indica o canto superior direito do mesmo contorno. Com esses três pontos, é possível mapear a posição 3D do objeto e calcular a sua largura (função  $SP3D_f$  do Algoritmo 1). Esses valores produzem o vetor ( $OE_n(X, Y, Z)$ ).



**Figura 3.6:** (a) Contorno dos obstáculos; (b) Características extraídas dos obstáculos ( $cm$ ,  $p1$  e  $p2$ ) da imagem (a).

Da mesma forma, calculou-se a posição e a largura da passagem. Baseado no contorno da região  $PL(x_j, y_j, Z_j)$ , obtiveram-se os três pontos ( $cm$ ,  $p1$  e  $p2$ ) e, com os mesmos princípios aplicados para os objetos estáticos, essa característica foi submetida ao Módulo de Fusão. A Figura 3.7 apresenta como o mapeamento dos objetos é realizado. Com a conversão de coordenadas  $(x, y)$  para  $(X, Z)$ , é possível compreender como o SAS-DV fornece dados para o sistema de fusão calcular os riscos.

Os submódulos de Mapa de Profundidade, de segmentação de objetos estáticos e de posição 3D formam o Algoritmo 1, que detecta a Passagem Livre e os Objetos Estáticos no plano 3D.



**Figura 3.7:** (a) Contorno dos obstáculos; (b) Posições de obstáculos convertidas do eixo (x, y) da imagem (a) para (X, Z).

### 3.2.3 Sistema de Objetos Dinâmicos

A locomoção para um DV é difícil em cenários que apresentem obstáculos e pessoas em movimento, principalmente quando não são disponibilizados dispositivos que mencionem a presença de elementos que possam gerar colisão. A presença de Objetos Dinâmicos (OD) é um dos principais fatores que comprometem a mobilidade do DV, no entanto, eles são importantes para indicar as áreas mais seguras para se trafegar. O Sistema de Objetos Dinâmicos extrai características relacionadas apenas aos Objetos Dinâmicos presentes no campo de visão do DV e reconstrói a trajetória produzida por ele. O objetivo deste sistema é detectar o objeto que está em movimento, analisar sua direção, estimar a velocidade e calcular sua área. Adquirir e relacionar essas características com o ambiente é importante para o sistema de fusão de dados projetar um caminho seguro para a locomoção do DV. A quantidade de fontes de dados pode refletir paralelamente a qualidade da resposta ou projeção do sistema de fusão. Ressalta-se a importância dessa tecnologia como suporte à análise de objetos dinâmicos para auxiliar o deficiente nas tomadas de decisão.

O Sistema de Objetos Dinâmicos é formado pelos Submódulos de Controle e de Estabilização de Vídeo (SCEV), de Segmentação de Objetos Dinâmicos (SSOD) e de Posição 3D (SP3D). O SCEV é responsável por controlar os dados fornecidos pelo sensor de aquisição (RGB, IV e acelerômetro). A partir desses dados, esse submódulo define a escolha sobre qual técnica de detecção e rastreamento de objetos dinâmicos será usada. Se a técnica selecionada exigir a estabilização do vídeo, é nesse submódulo que essa operação é executada. O SSOD tem como finalidade segmentar os objetos dinâmicos presentes nas cenas e o SP3D, mais especificamente neste sistema, realiza o mapeamento da posição 3D dos ODs.

---

**Algorithm 1:** ALGORITMO DE DETECÇÃO DE OBJETOS ESTÁTICOS (OES) E PASSAGENS LIVRES (PLS)
 

---

**Input:** Vídeo<sub>MP</sub>**Output:** Objetos Estáticos (OE) e Passagem Livre (PL)

```

1 //Obtêm VídeoMP a partir da câmera IV
2 while (Capturar VídeoIV) do
3   //Converte VídeoIV para imagem
4   ( $\langle IV_1, IV_2, \dots, IV_n \rangle$ )  $\leftarrow$  convertFrame(InfraRedVdeo)
5   //Converte o  $k_j$  do  $IV(x_j, y_j, k_j)$  para  $Z_j$  do  $MP(x_j, y_j, Z_j)$ 
6   ( $\langle MP_1, MP_2, \dots, MP_n(x_j, y_j, Z_j) \rangle$ )  $\leftarrow$  MPSf( $\langle IV_1, IV_2, \dots, IV_n(x_j, y_j, k_j) \rangle$ )
7   //Limiarização para detectar objetos com menos de 2 metros
8   ( $\langle OE_1, OE_2, \dots, OE_n(x_j, y_j, Z_j) \rangle$ )  $\leftarrow$  SSOEso( $\langle MP_n(x_j, y_j, Z_j) \rangle$ )
9   while ( $c \leq n$ ) do
10    //Detecta contorno (OE) por Bradski et al. (2008)
11    findContours( $\langle OE_{[c]}(x_j, y_j, Z_j), contours_{so[c]} \rangle$ )
12    //Calcula a posição e a largura dos objetos baseado no contorno)
13    ( $\langle OE_{[c]}(x_j, y_j, Z_j).cm, OE_{[c]}(x_j, y_j, Z_j).p1, OE_{[c]}(x_j, y_j, Z_j).p2 \rangle$ )  $\leftarrow$  CMWf(contoursso[c])
14    //Converte a posição e a largura dos Objetos Estáticos em (mm) (Equações 3.1 e 3.2)
15    ( $\langle OE_{[c]}(X, Y, Z).cm, OE_{[c]}(X, Y, Z).p1, OE_{[c]}(X, Y, Z).p2 \rangle$ )  $\leftarrow$  SP3Df(contoursso[c])
16     $c++$ 
17    //Limiarização para detectar passagem livre mais distante
18     $PL(x_j, y_j, Z_j) \leftarrow$  SSOEpl(MP2, onde (Z))
19    //Detecta contorno (PL) por Bradski et al. (2008)
20    findContours( $PL(x_j, y_j, Z_j), contour_{pl}$ )
21    //Calcula a posição e a largura da passagem baseado no contorno)
22    ( $\langle PL(x_j, y_j, Z_j).cm, PL(x_j, y_j, Z_j).p1, PL(x_j, y_j, Z_j).p2 \rangle$ )  $\leftarrow$  CMWf(contourpl)
23    //Converte a posição e a largura da Passagem Livre em (mm) (Equações 3.1 e 3.2)
24    ( $PL(X, Y, Z) \leftarrow$  SP3Df(contourpl)
25 return  $OE_n(X, Y, Z), PL(X, Y, Z)$ 

```

---

O Sistema de Objetos Dinâmicos exigiu maior complexidade de implementação e funcionamento se comparado ao Sistema de Objetos Estáticos. Devido a isso, foi elaborada uma nova metodologia para comparar as principais características das diferentes técnicas de detecção de ODs. Nesta seção, são apresentadas apenas as técnicas implementadas e a solução aplicada definitivamente no SAS-DV. Para cada técnica foi criada uma arquitetura. Para comparar a efetividade, o tempo de detecção e o custo computacional dessas arquiteturas foi desenvolvida uma nova metodologia. As especificidades de todas as arquiteturas implementadas e comparadas podem ser vistas no Capítulo 4.

Foram verificadas no estado da arte as técnicas mais usadas para detecção e rastreamento de Objetos Dinâmicos. Como já mencionado, para cada técnica uma nova arquitetura foi projetada. Dessa forma, as técnicas mais observadas são apresentadas a seguir com siglas que representam

a sua arquitetura: Arquitetura 1- Reconhecimento de Padrões (A1-RP); Arquitetura 2- Fluxo Óptico de Farneback (A2-FOF); Arquitetura 3- Subtração de *Background* (A3-SB); Arquitetura 4- Continuously Adaptive Meanshift (A4-CAM).

É interessante ressaltar que o DV pode estar tanto parado como em movimento e ainda podendo realizar movimentos leves ou bruscos. Dessa forma, duas técnicas (Fluxo Óptico de Farneback e Subtração de *Background*), normalmente usadas para detecção de Objetos Dinâmicos com a câmera estática, sofreram adaptações para possibilitar a realização de experimentos com DVs.

### 3.2.3.1 A1-RP

A arquitetura A1-RP não teve adaptações na técnica para detectar Objetos Dinâmicos, apenas foi definida uma entidade para que o classificador pudesse detectar o objeto na realização dos experimentos. Neste caso, a detecção de OD foi baseada na teoria de Reconhecimento de Padrões que permite detectar apenas as classes previamente definidas. Essas classes podem ser qualquer elemento como pessoa, cadeira, escada, bicicleta etc. Nesse aspecto, apenas a classe com características do corpo de uma pessoa foi usada. Com a detecção próxima do tempo real, é possível analisar e entender os movimentos que as pessoas realizam. Com a definição de uma classe, as características de um objeto são extraídas e usadas para treinamento, para que possam ser detectadas automaticamente. No entanto, a biblioteca OPENCV já disponibiliza um modelo pronto para classificação de pessoas; dessa forma, não foi necessário extrair características de pessoas e realizar um processo de aprendizado. A biblioteca OPENCV usa a técnica *Haar-cascade* proposta por Viola *et al.* (2001) para permitir o treinamento supervisionado. A principal característica dessa técnica é seu alto desempenho na detecção de uma entidade pré-definida.

### 3.2.3.2 A2-FOF

A arquitetura usando Fluxo Óptico, proposto por Farneback, sofreu adaptações para que os movimentos da câmera não interferissem na detecção de Objetos Dinâmicos. Essa adaptação permitiu a detecção de Objetos Dinâmicos mesmo quando a câmera sofreu movimentos mais intensos. As adaptações podem ser vistas no capítulo 4.

### 3.2.3.3 A3-SB

Para esta arquitetura, um conjunto de adaptações foi implementado. A primeira adaptação ocorreu com implantação de um método de estabilização de vídeo. Esse método foi implementado usando imagens do tipo RGB e do tipo infravermelhas. A estabilização de vídeo ajudou de maneira significativa a detecção de Objetos Dinâmicos quando ocorriam movimentos leves no sensor Kinect. Entretanto, essa estabilização foi prejudicada nos casos em que houve variações na iluminação. Como as imagens RGB são bem sensíveis a essa variação, implementaram-se técnicas que permitiram diminuir essa sensibilidade. Neste caso, foi implementada a conversão do espaço de cor RGB para YCbCr. Essa implementação ajudou, mas, ainda assim, foram mantidas algumas interferências para o método de subtração de *background*. Após essa constatação, excluiu-se o uso das imagens do tipo RGB para os experimentos nesta arquitetura e mantiveram-se apenas as imagens infravermelhas.

As imagens infravermelhas são mais complexas para a detecção de pontos de controle, usados para a estabilização automática do vídeo (ver Seção 4.2.3 do capítulo 4). Em vista disso, foram criados alguns pontos nas bordas das imagens infravermelhas. Essa técnica permitiu que a estabilização ocorresse apesar das falhas da iluminação, o que gerou um custo computacional muito alto e, quando o movimento da câmera era mais intenso, o processo de estabilização era renovado por novos pontos de controle. Devido a isso, transformações geométricas inesperadas ocorriam.

Diante do mencionado, foi necessário verificar diferentes alternativas. Uma delas foi a realização de experimentos com alteração de alguns parâmetros que melhoram a segmentação de um objeto, usando técnicas de MOG (Diferença gaussiana). Com o aumento do parâmetro *varThreshold* a sensibilidade diminuía quando ocorria a detecção de Objetos Dinâmicos, porém, com movimentos intensos na câmera e com a proximidade do objeto dinâmico, essa técnica também se mostrou ineficaz. A alternativa teve início com pesquisas no estado da arte para verificar que outras técnicas poderiam ser usadas. Mesmo assim, essa técnica se manteve como uma das principais implementadas nesse sistema, devido à independência de iluminação e por realizar um rastreamento eficiente quando o sensor de aquisição sofre poucos movimentos.

### 3.2.3.4 A4-CAM

A última arquitetura implementada fez uso da técnica CamShift apresentada por (BRADSKI; KAEHLER, 2008; BRADSKI, 1998). Como essa técnica é invariante quando ocorrem movimentos na câmera, ela foi escolhida para compor o sistema final de detecção de Objetos Di-

nâmicos. Em razão dessa técnica permitir o rastreamento só após a seleção de cores de uma determinada região da imagem, foi necessário usar um método de detecção de objeto dinâmico para selecionar a região a ser operada pelo CamShift e, em seguida, realizar o rastreamento. Para esta arquitetura, foi escolhido o método de Fluxo Óptico de Farneback. Essas duas técnicas podem sofrer com a mudança de iluminação, por outro lado, a pessoa com deficiência visual pode gerar movimentos mais intensos, comparados com os permitidos na técnica de subtração de *background*.

### 3.2.3.5 Versão final do Sistema de Detecção e Rastreamento de Objetos Dinâmicos

Após experimentos com todas as arquiteturas citadas, optou-se por manter na versão final do sistema de detecção de Objetos Dinâmicos três modos de operação. Isso permite que o sistema detecte e rastreie OD em regiões escuras, claras e com movimentos no sensor de aquisição. É interessante acrescentar que, para manter os três modos de operação, as quatro arquiteturas implementadas e experimentadas compõem o sistema. Essa decisão foi baseada nos experimentos realizados e nos resultados obtidos, apresentados no Capítulo 4 e na Seção 6.1.

O sistema inicia-se com o método de Fluxo Óptico de Farneback usando mapas densos, porém, com pontos bem espaçados para não prejudicar o desempenho do sistema. Caso seja detectado algum objeto dinâmico e a câmera não sofra movimentos intensos, automaticamente é executada a Arquitetura A3-SB para rastreamento. Quando ocorrerem movimentos mais intensos, detectados pelo acelerômetro, automaticamente, a Arquitetura A4-Camshift é acionada. A arquitetura A1-RP é executada apenas quando desejada pelo deficiente visual, se o objetivo é detectar algumas entidades pré-definidas. Na arquitetura A1-RP, a entidade pessoa é a única previamente implementada para detectar objetos em movimento.

## 3.3 Módulo de Fusão

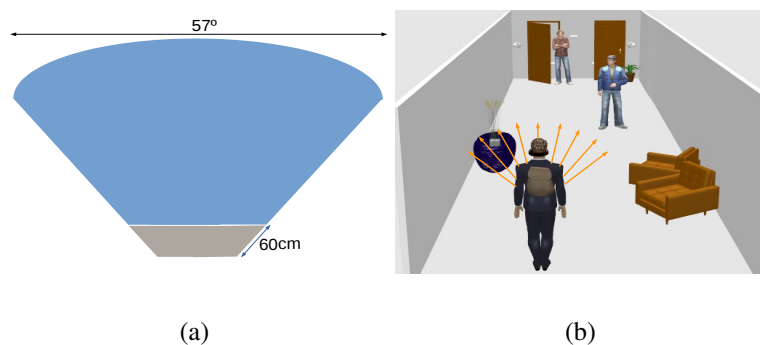
Esta seção apresenta, em detalhes, os dados e a metodologia usados no processo de modelagem, que visa projetar os riscos de colisão para o DV, segundo a abordagem do modelo da consciência da situação (*SAW*). O Módulo de Fusão procura relacionamentos consistentes entre os subobjetivos definidos na *GDTA*, com o intuito de fornecer ao DV uma locomoção livre de colisões. O Módulo de Fusão é formado por três Submódulos: Submódulo de Percepção (3.3.0.1), Submódulo de Compreensão (3.3.0.2) e Submódulo de Projeção (3.3.0.3). Cada processo pertencente ao Módulo de Fusão tem a finalidade de fornecer os dados necessários para

solucionar as decisões dos subobjetivos (3.1). Os dados foram obtidos por meio de técnicas de Visão Computacional e, a partir delas, os *datasets* foram modelados usando diferentes propostas. A partir desses *datasets*, foi possível analisar a precisão que diferentes classificadores puderam fornecer. A apresentação da metodologia aplicada para a geração dos riscos de colisão é dividida entre os três submódulos (Submódulo de Percepção, Submódulo de Compreensão e Submódulo de Projeção). Dessa forma, a leitura desses submódulos permite entender como foi realizada a modelagem dos *datasets* e como foram organizados os dados para a realização de inferências que projetaram os riscos de colisão para os DVs.

### 3.3.0.1 Submódulo de Percepção

No SAS-DV, o Submódulo de Percepção tem como finalidade organizar e integrar as características fornecidas pelo Módulo de Visão (Objeto Dinâmico, Objeto Estático e Passagem Livre (*DEP*)) e produzir informações para os Submódulos de Compreensão e Projeção. A integração dessas características deve ser realizada de modo que seja possível classificar a situação pelo Submódulo de Compreensão e construir um mapa de risco composto por inferências executadas no Submódulo de Projeção.

**Mapeamento e Pontos de referência** Para organizar todos os dados (*DEP*) obtidos pelo Módulo de Visão, criou-se um método baseado em um mapa que armazena a direção (eixo X) e a distância (eixo Z) de cada elemento. As direções e as distâncias foram especificadas de acordo com a área de abrangência do dispositivo Kinect, (ver Figura 3.8 (a)). Baseado nessa área, todos os objetos dinâmicos, objetos estáticos e passagens livres (*DEP*) são mapeados na posição (X, Z). A Figura 3.8 (b) apresenta uma simulação desse processo.

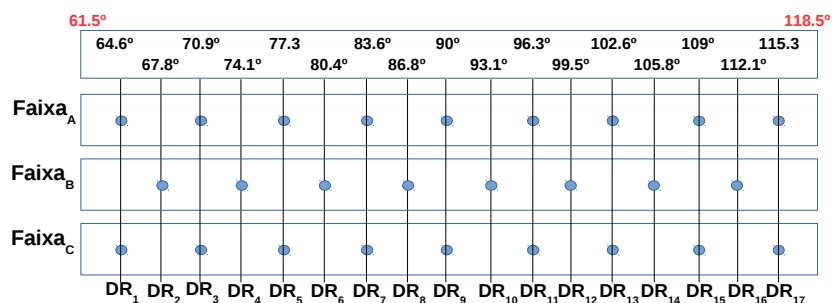


**Figura 3.8:** (a) Área de abrangência do dispositivo Kinect. (b) Simulação de aquisição de dados.

Neste estudo, criaram-se duas estratégias para mapear os elementos presentes no ambiente

com a finalidade de projetar os riscos de colisão. Essas estratégias são detalhadas posteriormente (ver Figuras 3.11 e 3.12).

Além dessas estratégias de mapeamento, implementou-se, também, um método para detectar a distância de 26 Pontos de Referência (PRf), que melhoraram a percepção do ambiente e são importantes para o mapeamento dos elementos. Definiu-se para este método uma análise baseada em três alturas ( $Faixa_A$ ,  $Faixa_B$  e  $Faixa_C$ ) e 17 direções ( $DR_1, DR_2, \dots, DR_{17}$ ) (Ver Figura 3.9). Cada direção (ângulo) apresenta uma região em que o DV pode trafegar. As alturas foram criadas para detectar um obstáculo que pode estar presente em uma direção, contudo, em uma altura específica. Dessa forma, para a definição dos PRf, considerou-se a posição do dispositivo Kinect, fixado à cintura do DV. Esse método foi criado como uma forma complementar para detectar dados em uma área que costuma apresentar mais obstáculos e, conseqüentemente, maior risco de colisão (ver Figura 3.6). Com as distâncias dos PRf, é possível fazer um teste preliminar (união) com o método de mapeamento, com a finalidade de constatar a presença de algum obstáculo que não havia sido detectado. Esse processo permite que as informações escolhidas para compor o *dataset* possuam mais confiabilidade. A Figura 3.9 apresenta todos os ângulos e alturas implementados nos SAS-DV. Supondo que o DV esteja iniciando o seu trajeto, os ângulos  $61.5^\circ$  e  $118.5^\circ$  representam os limites de aquisição de dados, como apresentado na imagem (b) da Figura 3.8.

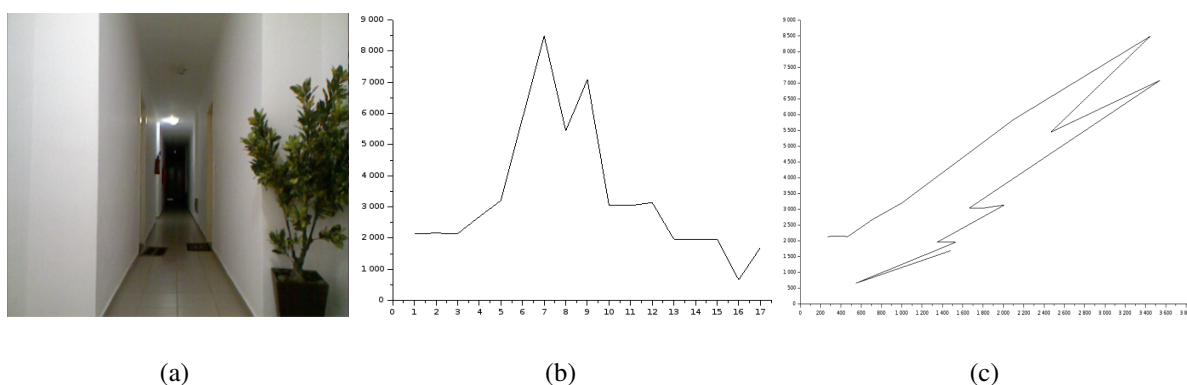


**Figura 3.9: Pontos de referência (PRf) definidos para detectar a distância dos obstáculos em várias direções e também usados para calcular os riscos de colisão em algumas direções na área de abrangência que o dispositivo Kinect atua (ver Figura 3.8 (a)).**

A Figura 3.10 ilustra um exemplo com a aplicação desses PRf em um ambiente interno. Nessa figura, é possível observar 17 distâncias de um total de 26 PRf. Ressalta-se que as faixas  $Faixa_A$  e  $Faixa_C$  possuem os mesmos ângulos, sendo assim, para o processo de percepção, definiu-se manter as menores distâncias detectadas entre elas, pois elas oferecem maiores riscos de colisão.

Na Figura 3.10 (a) é apresentada a imagem de um Ambiente Interno e seus respectivos gráficos (imagem (b) e (c)) que representam as distâncias obtidas dos PRf. Essas distâncias





**Figura 3.10: Pontos de referência com suas respectivas distâncias.**

são usadas para compor os *datasets* modelados no Submódulo de Compreensão. A imagem (b) ilustra as distâncias dos PRfs obtidas da imagem (a) usando apenas os dados do eixo Z (distância) e do eixo (x) que representa as 17 direções. Na imagem (c) é ilustrado um gráfico após a conversão do eixo (x - direções) para as posições dos obstáculos em milímetros (eixo X) com as suas respectivas distâncias (eixo Z). Da mesma forma, pode também ser usado o eixo Y para análise da altura dos obstáculos.

A primeira estratégia para mapeamento dos elementos foi concebida com a criação do primeiro *dataset*. Ele foi criado especificamente para uma decisão realizada de maneira empírica ou com o uso de uma rede bayesiana. Para simplificar o processo de aprendizagem de uma rede bayesiana, foram estabelecidas crenças específicas para a detecção dos seguintes elementos: objeto estático, passagem livre, objetos dinâmicos e intervalos de distâncias (1.4m, 1.7m e 2.0m) a partir dos pontos de referências (PRf) (definidos na Figura 3.9). Esses intervalos são interessantes para criar crenças que definem maiores riscos de colisão baseados na distância do obstáculo.

Para melhorar o entendimento do SPE, é apresentado na Figura 3.11, o formato do vetor de características produzido com informações obtidas em apenas uma faixa de pontos de referência ( $Faixa_A$ ). Na mesma figura, são apresentados os quatro vetores que são usados para indicar as direções que presenciam: objetos estáticos, objetos dinâmicos, passagem livre, trajetos realizados por OD e os três vetores que foram criados para armazenar as distâncias detectadas pelos pontos de referência (1.4m, 1.7m, 2.0m). Sendo assim, o *DEP* é complementado pelo Submódulo de Percepção após analisar os elementos  $(OD_n(X, Y, Z)_m, OE_n(X, Y, Z), PL(X, Y, Z))$  provenientes do Módulo de Visão (Ver Figura 3.5). O SPE produz um vetor para cada uma das três faixas pré-definidas (Pontos de Referência). Esses vetores contêm valores (0 e 1) que indicam a presença ou ausência dos elementos pré-definidos para a tomada de decisão.

Os novos vetores criados no Submódulo de Percepção ( $Faixa_ADEP$ ,  $SFaixa_BDEP$  e  $Faixa_CDEP$ ) são formados pelas respostas aos seguintes questionamentos: existe obstáculo em uma direção e distância específica? Houve pessoas trafegando em uma direção e distância específica? Existe passagem disponível em uma direção e distância específica? Existe algum trajeto realizado por objeto dinâmico? Em quais direções foram detectados obstáculos por meio dos PRfs? Esses cinco questionamentos são realizados em todos os pontos de referências (ver Figura 3.9).

Apesar do Submódulo de Percepção escolher quais pontos produzem maiores riscos entre as faixas ( $Faixa_1$  e  $Faixa_3$ ), a necessidade de mantê-las é alta, pois um obstáculo pode ser detectado em uma direção com uma altura diferente. Dessa forma, é interessante que a quantidade dessas faixas possa aumentar, pois produzirá maior confiabilidade. Em contrapartida, o sistema pode gerar muito gasto computacional e prejudicar o desempenho na geração dos mapas de riscos.

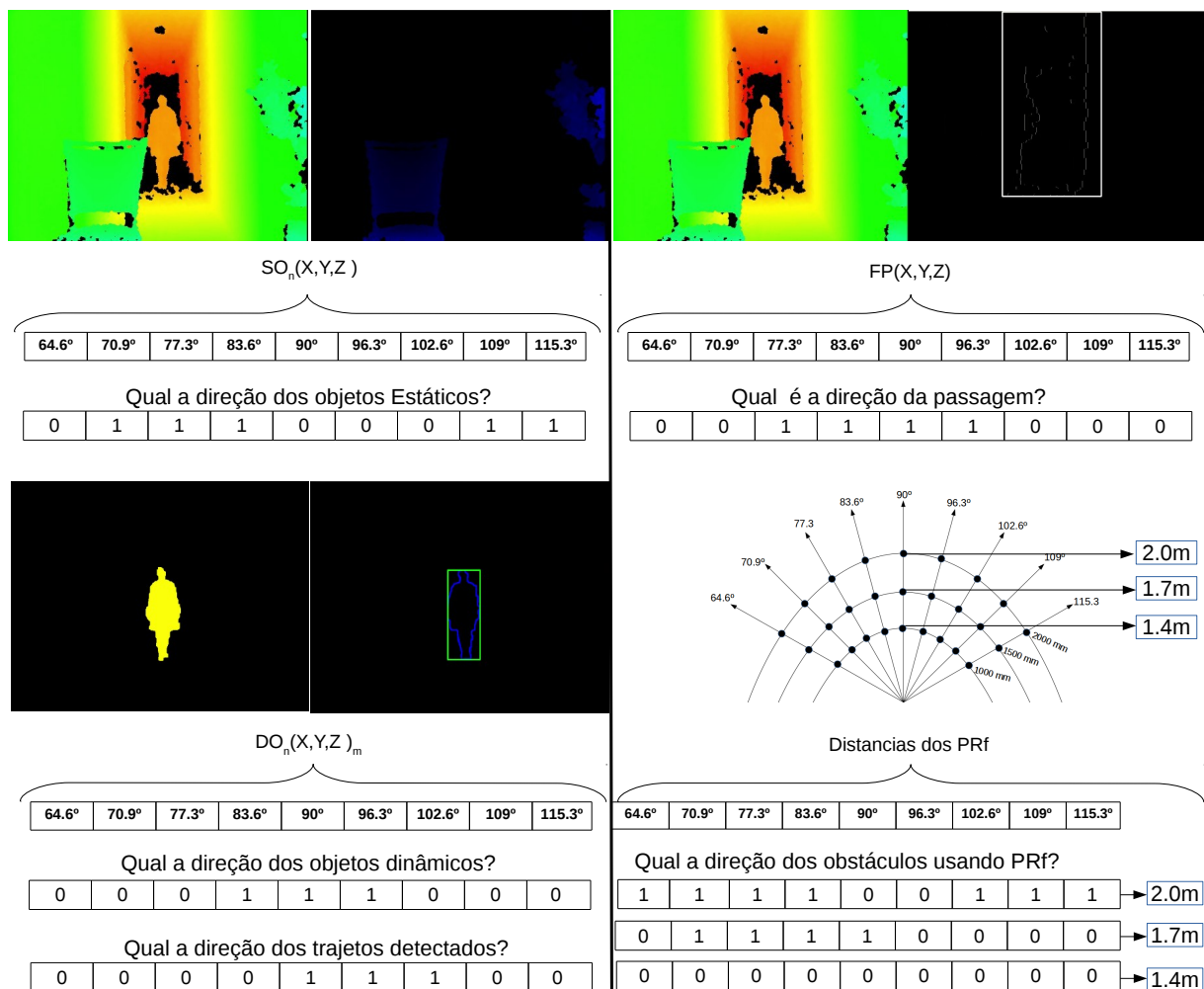
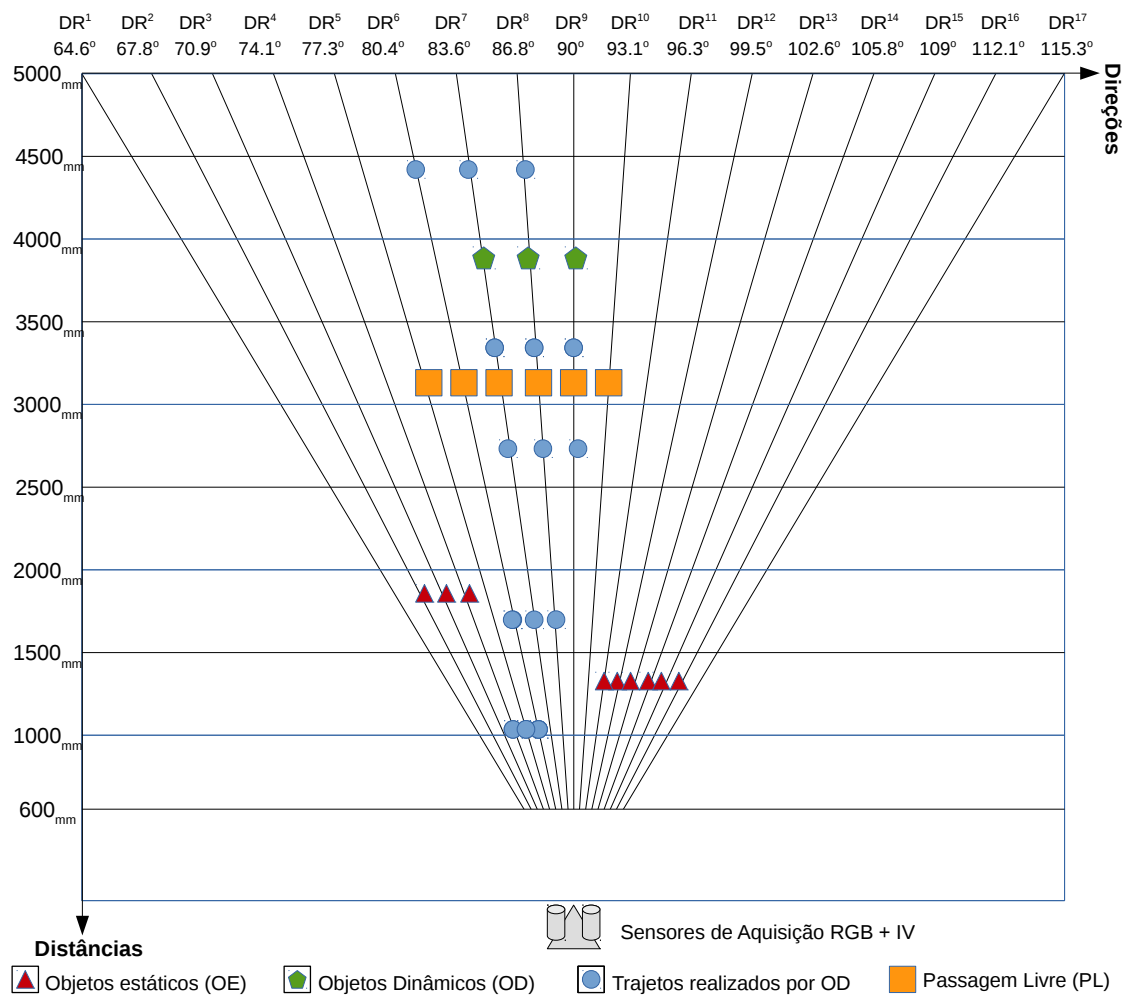


Figura 3.11: Dados que compõem o vetor de característica usado na primeira estratégia de mapeamento do processo de percepção / Base de criação para o primeiro dataset no processo de compressão.

Baseado em conceitos semelhantes aos usados na criação do primeiro *dataset*, surgiu a ideia de criar a segunda estratégia de mapeamento. Para isso, criou-se um segundo *dataset* e os Pontos de Referências (ver Figura 3.9) foram mantidos, porém, em vez de utilizar os intervalos 1.4m, 1.7m e 2.0m onde cada obstáculo poderia estar, foram utilizadas as distâncias em que cada obstáculo foi detectado. Essas distâncias também serão unidas com o vetor de distâncias dos objetos estáticos para aumentar a percepção de obstáculos não detectados, como no primeiro *dataset*. Exemplo de como o *dataset* é composto pode ser visto na Figura 3.12 e o *dataset* pode ser visto na Seção 3.3.0.2 do Submódulo de Compreensão.

A figura 3.12 representa o funcionamento do método proposto, que tem como objetivo analisar os riscos de colisão em diferentes direções (DR) relacionadas com diferentes distâncias (DS). Por meio de intervalos e cruzamentos realizados entre direções e distâncias, foram gerados campos (Map(Direções, Distâncias)) que permitem indicar a posição de elementos importantes para a tomada de decisão. Esses elementos são representados por círculos com as seguintes cores: 1- Obstáculos estáticos (triângulos vermelhos); 2- Obstáculos dinâmicos (Pentágonos verdes); 3- Trajetos criados por ODs (Círculos Azuis); 4- Passagem livre (Quadrados amarelos). No mapa apresentado pela Figura 3.12, é possível observar que, a partir da direção  $DR^1$  ( $67.8^\circ$ ) até a direção  $DR^4$  ( $74.1^\circ$ ), existe um objeto estático a menos de 2 metros de distância. Na direção central, há uma passagem livre ocupando a área de seis direções devido a sua largura. Com os cálculos da posição e do tamanho dos obstáculos, foi possível realizar o mapeamento num plano 3D e analisar o conjunto de dados que compõem uma inferência. Ressalte-se que para calcular a posição e o tamanho dos objetos foram usadas as equações 3.1 e 3.2 apresentadas na Seção 3.2.2.2. Da direção  $DR^7$  ( $83.6^\circ$ ) até a  $DR^9$  ( $90^\circ$ ), é possível observar a detecção de um objeto dinâmico bem próxima do tempo real, a menos de 4 metros de distância, terminando seus movimentos nas direções (de  $DR^4$  à  $DR^6$ ) com apenas 1 metro de distância do DV. Os círculos azuis representam o trajeto já realizado por outro objeto dinâmico.

A figura 3.12 apresenta apenas o mapeamento dos elementos presentes num ambiente em determinado instante, após a captura dos dados por meio de uma câmera RGB e IV. O sistema fica em execução capturando esses dados em tempo real. Todavia, após o mapeamento de todos os elementos, que necessita da execução de várias técnicas de Processamento de Imagens e de Visão Computacional, um atraso é produzido. Mesmo assim, com esse atraso detectado, principalmente quando se tem objetos dinâmicos com alta velocidade, as respostas são interessantes. Após obtenção das respostas em diferentes posições do DV, formou-se o conjunto de dados para treinamento. Com isso, um novo conjunto de características ( $Faixa_ADEP$ ,  $Faixa_BDEP$  e  $Faixa_CDEP$ ) é submetido ao Submódulo de Compreensão, para a análise dos seus comportamentos.



**Figura 3.12:** Elementos usados para criar a segunda estratégia do processo de percepção e para produzir o segundo *dataset* no processo de compreensão.

### 3.3.0.2 Submódulo de Compreensão

O Submódulo de Compreensão executa a segunda fase do Módulo de Fusão, na qual é realizada uma nova fusão, porém, de informações que já foram refinadas pelo SPE. Nesse processo, é realizada a análise do comportamento dos objetos, fundamentado em suas ações, intenções, relevância e capacidade (LIGGINS; HALL; LLINAS, 2008). Dessa forma, os impactos foram obtidos após a análise da relação entre os objetos presentes no ambiente que são relacionados com riscos de colisão.

Baseado no mapeamento (Ver Figuras 3.11 e 3.12) e nos pontos de referência (Ver Figura 3.9) fornecidos pelo SPE, desenvolveu-se no Submódulo de Compreensão dois conjuntos de dados (*datasets*) para compreender o comportamento dos elementos presentes no ambiente, especificamente, para análise de risco de colisão. Esses *datasets* relacionam, de modo consistente, todas as características em busca de detectar padrões sistemáticos no contexto do DV. A partir

desses relacionamentos, é possível realizar classificações que possibilitam o uso de inferências para projetar situações num futuro próximo. Os dois *datasets* produzidos, com tipos de dados variados, permitiram comparar a eficácia com diferentes classificadores.

**Primeiro Dataset (Boolean)** O primeiro *dataset* criado no Submódulo de Compreensão utilizou a Teoria das Redes Bayesianas para considerar uma probabilidade com o grau de certeza da ocorrência de um evento. Segundo Liggins *et al.* (2008), as redes de crenças permitem produzir relacionamentos em torno de uma situação. No parecer de Conrady *et al.* (2015), as Redes Bayesianas podem ser construídas a partir do conhecimento humano ou desenvolvidas a partir de uma combinação entre humanos e a inteligência artificial. O processo de aprendizagem da Rede Bayesiana aplicada no SAS-DV pode ser realizado por um especialista, que define as probabilidades *a priori* ou de forma indutiva, em que se utiliza de um conjunto de dados para construir uma rede de maneira automática. O fato pode ser um problema em aplicações que dificulta essa compreensão, no entanto, a rede do Submódulo de Compreensão foi modelada de uma forma semelhante a uma pessoa com visão que toma decisões para trafegar no caminho menos perigoso. Isso posto, não foi difícil definir as probabilidades iniciais. O modelo produzido (*modelo<sub>Bayes</sub>*) no Submódulo de Compreensão é composto por uma rede Bayesiana em que a variável do domínio, Risco de Colisão, é, condicionalmente, considerada dependente das seguintes variáveis: Objetos Dinâmicos, Trajetos realizados por Objetos Dinâmicos, Objetos Estáticos, Passagem livre e Obstáculos detectados nos intervalos de 1.4m, 1.7m e 2m de distância. Após a construção da rede definiu-se a distribuição das probabilidades *a priori* fundamentada nas suas ações e intenções, relevância e capacidade dos elementos. Para esse processo foi aplicado o teorema de Bayes.

As probabilidades foram definidas baseadas nos seguintes conceitos: a direção que apresenta uma passagem possui um peso positivo para a locomoção, para o DV chegar em determinado destino, visto que o sistema foi desenvolvido apenas para ambientes fechados; a direção que detecta objeto estático e dinâmico em sua faixa aumenta a probabilidade de risco naquele determinado instante. Contudo, após o objeto dinâmico sair da faixa de tráfego do DV, a probabilidade de riscos de colisão diminui, pois, entende-se que se uma pessoa executou determinado trajeto, há menos risco de colisão. A direção que possui obstáculos mais próximos apresenta maiores riscos de colisão, sendo assim, as probabilidades de colisão tiveram maior peso se comparadas com distâncias maiores.

Com a rede treinada, foi possível realizar todas as inferências (pelo Submódulo de Projeção) necessárias para produzir o mapa de riscos. Cada inferência fornece uma probabilidade de risco de colisão (RC) para uma direção (Ver Figura 3.13). Após a produção do mapa de riscos,

o SAS-DV comunica o ângulo que o DV tem mais chances de trafegar sem colisão com obstáculos. O *textitBayesian Network*, editor da ferramenta Weka, foi usado para gerar o primeiro conjunto de dados, por meio das tabelas de probabilidade condicional (presentes no Weka). Esse *dataset* foi proposto com o uso de valores Booleanos (ver Tabela 1). Sendo assim, o programa Weka produziu um *dataset* com repetições de determinadas crenças para que o sistema de aprendizado seja induzido a calcular as probabilidades de risco de colisão, de acordo com as crenças atribuídas nas tabelas de probabilidade condicional. Esse processo se torna semelhante quando calculadas as probabilidades de forma empírica. Os detalhes deste experimento podem ser vistos na tabela a seguir (Tabela 3.1), na qual é apresentado o primeiro *dataset* criado com valores Booleanos e treinado com a teoria das Redes Bayesianas.

**Tabela 3.1: Organização dos dados do primeiro *dataset***

Exemplo	OBS<1.4	OBS>1.4<1.7	OBS>1.7<2.0	OD	PL	TRJ <sub>OD</sub>	RC
1	1	0	0	0	0	1	1
2	0	0	0	0	1	0	0
3	0	1	0	1	0	0	1
4	0	1	1	0	0	0	1
5	0	0	0	0	1	1	0
6	0	0	0	1	1	1	1

O primeiro *dataset* se baseou nas tabelas de probabilidade condicional e, após a realização dos experimentos, foi analisada uma nova forma de produzir um segundo *dataset* com o uso dos Pontos de Referência e por meio dos vetores de dados fornecidos pelo SPE, mas sem o uso dos intervalos de 1.4m, 1.7m e 2.0m. Em lugar de usar os intervalos, foram adicionados no segundo *dataset* os valores das distâncias de cada obstáculo detectado na direção especificada pelos Pontos de Referência (ver Figura 3.9). Sendo assim, mantiveram-se as direções e as mesmas entidades, o que resultou em um formato para armazenar as posições (em milímetros) de todas as entidades detectadas. A captura dos dados foi realizada pela execução do sistema SAS-DV em diferentes instantes e contextos. A figura 3.12 apresenta o método usado para compor o *dataset*.

**Segundo *Dataset* (numeric)** O *dataset* desta segunda etapa de projeção de riscos de colisão é composto por dados do tipo *numeric* (Ver Tabela 3.2), diferentes do tipo usado na primeira etapa (Booleano). Esse tipo de dado permite que a variação de informações seja maior e mais complexa. Devido a isso, é importante que o analista fique atento ao balanceamento e possibilidades de ocorrências das entidades presentes no *dataset*.

Observa-se que, por meio do método proposto neste estudo, é possível mapear todos os

elementos detectados, aproximadamente, em tempo real. Após a aquisição de um conjunto de dados mapeados por esse método, criou-se uma base de dados para a classificação das probabilidades de riscos de colisão presentes em cada direção. Dentro dessa perspectiva, foi possível gerar um modelo de aprendizagem que vai permitir a realização de inferências.

Para verificar as possibilidades de riscos de colisão em cada deslocamento, foi necessário contextualizar a posição dos Objetos Estáticos e dos Objetos Dinâmicos, e a posição da passagem livre e de trajetos realizados por OD. Com esses dados mapeados em um plano 3D, foi possível classificar para cada direção os seguintes riscos de colisão: 1- *Baixo*; 2- *Moderado*; 3- *Alto*; 4- *Muito Alto*; 5- *Colisão próxima* (ver campo class na Tabela 3.2). Nessa mesma tabela é possível interpretar melhor a composição da base de dados que formou o modelo de aprendizagem. Para toda direção em que foi detectado um obstáculo, passagem livre ou trajeto realizado por OD, que permite definir o risco de colisão, foi armazenada a sua respectiva distância em milímetros. O símbolo "?" é usado no *dataset* em situações que o sistema não detecta informação. As colunas OE, OD, PL, TR exibem, respectivamente, as seguintes informações: a distância do objeto estático detectado, a distância do objeto dinâmico detectado, a distância da passagem livre detectada, e a distância do trajeto realizado por algum OD.

**Tabela 3.2: Organização dos dados do segundo *dataset***

<b>Exemplo</b>	<b>OE<sub>mm</sub></b>	<b>OD<sub>mm</sub></b>	<b>PL<sub>mm</sub></b>	<b>TRJ<sub>mm</sub></b>	<b>RC</b>
1	?	?	5015	1570	Baixo
2	?	4530	3105	2030	Moderado
3	1520	2640	?	?	Alto
4	?	1240	?	1430	Muito Alto
5	950	780	?	1050	Colisão próxima

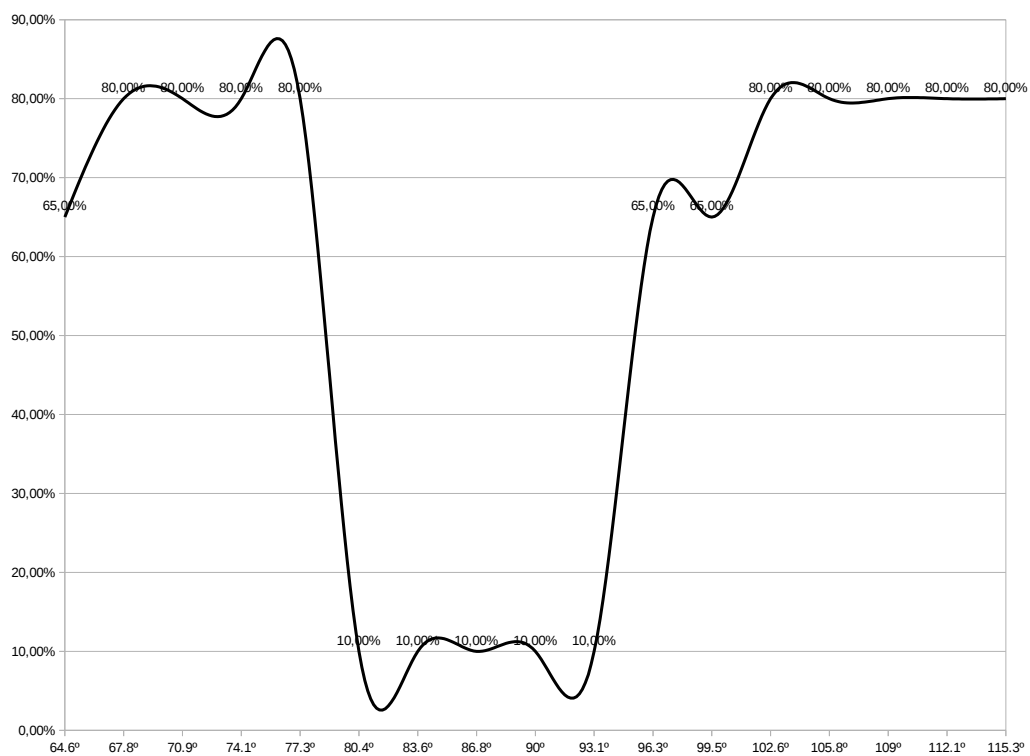
O *dataset* produzido é composto por mais de 100 situações e parte dele pode ser vista na Tabela 3.2 (Linhas 1 a 5). No processo de produção deste *dataset* foi realizado o balanceamento de todas as classes.

### 3.3.0.3 Submódulo de Projeção

O Submódulo de Projeção (Submódulo de Projeção) é a terceira fase do Módulo de Fusão. Esse processo deve realizar a fusão das informações produzidas pelo Submódulo de Compreensão com o intuito de projetar situações num futuro próximo. O Submódulo de Projeção projeta as ações dos objetos em relação ao DV para definir os possíveis impactos em diferentes direções. No Submódulo de Projeção são realizadas inferências em diferentes direções em busca de relacionar obstáculos, passagens disponíveis e trajetos realizados por Objetos Dinâmicos para que

seja possível descobrir os riscos de locomoção em determinadas regiões. A cada movimento do DV, o sistema de inferências pode ser ativado para projetar o mapa de riscos. As posições de maiores relevâncias definidas para projetar o mapa de riscos são apresentadas pelos círculos azuis da Figura 3.9. Cada ponto de referência será útil para organizar as características usadas para inferência. Cada inferência produz uma probabilidade de risco de colisão. O conjunto de inferências forma o mapa de riscos, o qual indica as regiões mais seguras para locomoção. Por fim, o Submódulo de Projeção fornece informações ao DV por meio de sons ou bipes  $\{s, b\}$ .

A Figura 3.13 apresenta um gráfico produzido com auxílio de uma planilha eletrônica. Esse gráfico foi intitulado de mapa de riscos, devido à apresentação da porcentagem de risco de colisão que cada direção possui. Esse mapa foi produzido a partir de um ambiente contendo alguns obstáculos (ver item (a) Figura 3.10). As inferências foram produzidas a partir do primeiro *dataset* (ver Seção 3.3.0.2). Nessa figura, a linha preta indica o resultado das inferências para todas as direções (ângulos definidos na Figura 3.9). É relevante observar que este mapa é formado por 17 ângulos. Para cada ângulo, uma inferência fornece o risco de colisão baseado em todas as características extraídas e relacionadas pelo Módulo de Fusão. Um exemplo semelhante desse mapa usando o segundo *dataset* pode ser visto nas Figuras 5.21, 5.22 e 5.23.



**Figura 3.13:** Gráfico (mapa de riscos) que representa os riscos de colisão que o DV pode ter em cada direção.



É preciso salientar que, apesar de o Submódulo de Percepção definir 26 distâncias, ele submete um vetor contendo apenas 17 distâncias (ver Seção Submódulo de Percepção 3.3.0.1). Por isso, os processos de fusão de nível mais elevado recebem 17 distâncias referentes a objetos estáticos, passagens livres, objetos dinâmicos e trajetões realizados por OD. O mapa de riscos é apresentado com apenas o resultado dessas 17 direções. Isso ocorre por causa das faixas *Faixa<sub>1</sub>* e *Faixa<sub>3</sub>* possuírem os mesmos ângulos, porém, em diferentes alturas. Dessa forma, definiu-se usar no *dataset* os pontos de referências que apresentaram maiores riscos de colisão. Isso se justifica devido à altura do DV ser maior que a diferença entre essas faixas. Isto posto, um obstáculo presente na *Faixa<sub>1</sub>* e ausente na *Faixa<sub>3</sub>* (ou vice-versa) apresentará, de alguma forma, o risco de colisão para o DV.

# Capítulo 4

## SISTEMA DE DETECÇÃO E RASTREAMENTO DE OBJETOS DINÂMICOS

O capítulo de sistemas de Detecção e Rastreamento de Objetos Dinâmicos (DROD) apresenta, em detalhes, as diferentes técnicas implementadas em arquiteturas específicas para a detecção, segmentação e rastreamento de Objetos Dinâmicos. Além disso, neste capítulo é apresentada a concepção de uma nova metodologia para comparar as vantagens e desvantagens das técnicas abordadas. Essa comparação permitiu definir uma arquitetura específica para ser utilizada no Sistema de Objetos Dinâmicos do SAS-DV (Ver Figura 4.1).

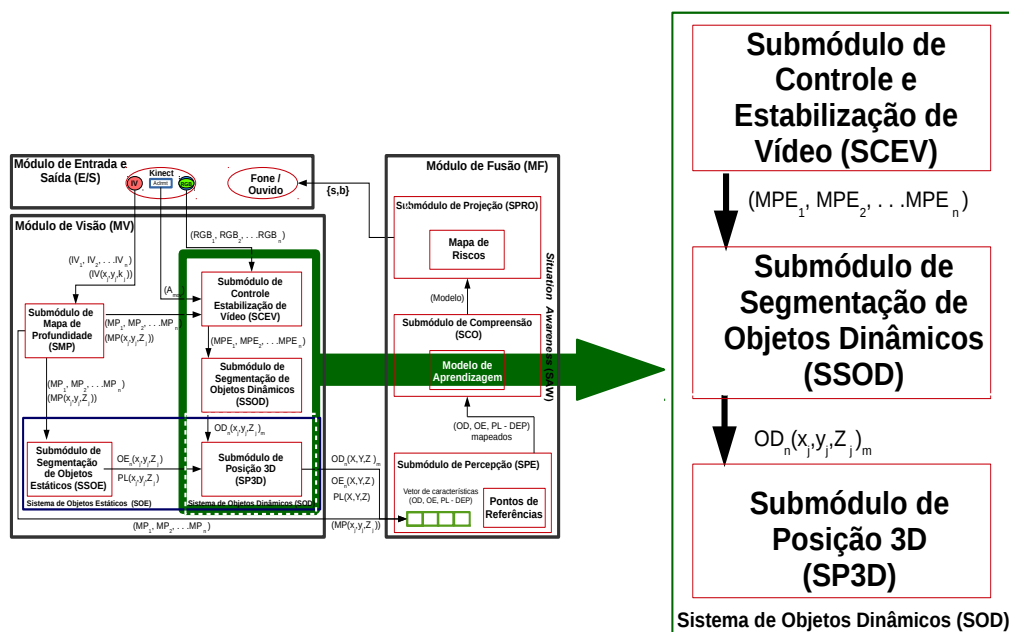


Figura 4.1: A organização dos submódulos do Sistema de Objetos Dinâmicos da arquitetura SAS-DV.

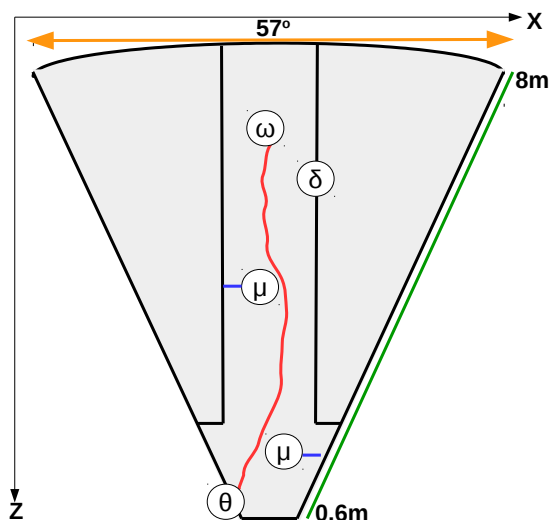
## 4.1 Os Objetos Dinâmicos e a mobilidade do Deficiente Visual

A análise de Objetos Dinâmicos (OD) em imagens é um campo desafiador, principalmente quando o Sensor de Aquisição (SA) de imagens sofre movimentos de translação e rotação. Esse cenário se torna mais complexo quando os movimentos são produzidos de maneira descontrolada, como ocorre quando uma câmera é instalada em um corpo que está sujeito a mudanças de posição. Quando isso acontece, é preciso compreender e diferenciar o movimento do SA com relação aos objetos da imagem. Caso o SA não sofra algum tipo de movimento, as técnicas de análise de ODs podem ser simplificadas. A presença de Objetos Dinâmicos (OD) é um dos principais fatores que comprometem a mobilidade do DV, no entanto, ela pode ser importante para indicar as áreas mais seguras para se trafegar.

Uma variedade de aplicações tem sido produzida (PHAM; LE; VUILLERME, 2016; IPINA; BOTRÁN; NOVOA, 2011; MEKHALFI et al., 2015; TAMJIDI; YE; HONG, 2013) para fornecer alternativas de tráfego para o DV. Apesar disso, uma pequena quantidade tem observado que a análise de OD pode produzir diferentes informações sobre qual caminho trafegar com mais segurança. É preciso salientar que um objeto em movimento pode produzir um alto risco de colisão num determinado instante, mas, em outro momento, a sua trajetória pode indicar uma região para o DV trafegar com mais confiabilidade. Devido às dificuldades que o DV possui no processo de detecção do objeto em movimento, do rastreamento e da reconstrução da sua trajetória, apresenta-se neste estudo uma nova metodologia para ser aplicada em sistemas que detectam e rastreiam OD, com a implementação de diferentes técnicas de Visão Computacional e análise de imagens. Cada técnica foi projetada e implementada em uma nova arquitetura que realiza a detecção e o rastreamento para possibilitar a reconstrução do trajeto em um mapa 3D.

Com intuito de se obter o melhor resultado diante das adversidades de iluminação e movimento nos sensores SA, essa metodologia propõe explorar as melhores características de um conjunto de técnicas de DROD já bem exploradas na literatura. Para isso, foi necessário realizar as devidas adaptações para o contexto dos DVs. Essas adaptações são alguns dos aspectos inovadores da presente metodologia com relação aos projetos propostos na literatura. Muitos desses projetos aplicam essas técnicas mais especificamente em ambientes controlados, com câmeras estáticas e com objetivos diferentes. Em contextos envolvendo DV, os SA, frequentemente, estão em movimento e a detecção de OD possui diferentes complexidades. Sendo assim, cada arquitetura projetada para os DVs possui suas particularidades e envolve vantagens relacionadas com qualidade de segmentação e desempenho.

A Figura 4.2 apresenta o exemplo de um ambiente interno usado para extrair as características com as técnicas implementadas. O símbolo  $\delta$  representa as paredes que formam um corredor. Os símbolos  $\mu$  são os obstáculos estáticos e o  $\omega$  indica o início de um trajeto realizado por um OD, que no caso foi executado por uma pessoa. O  $\theta$  sinaliza o término do processo de rastreamento do OD.



**Figura 4.2:** Exemplo de ambiente interno usado nos experimentos. Cada símbolo desta imagem representa um elemento relevante no ambiente.

A Figura 4.2 também representa um ambiente que pode ser reconstruído usando um sensor infravermelho. O eixo X representa a largura e o eixo Z representa a distância. Em todos os sistemas, implementados para a realização da análise comparativa, implementou-se um Mapa(X, Z), semelhante ao apresentado na Figura 4.2, com a finalidade de verificar os erros e acertos na detecção e a qualidade do rastreamento em diferentes posições. O eixo Y, que forma o mapa 3D, também foi calculado para analisar a altura dos objetos dinâmicos, porém, não está representado nos mapas ilustrados nesta tese.

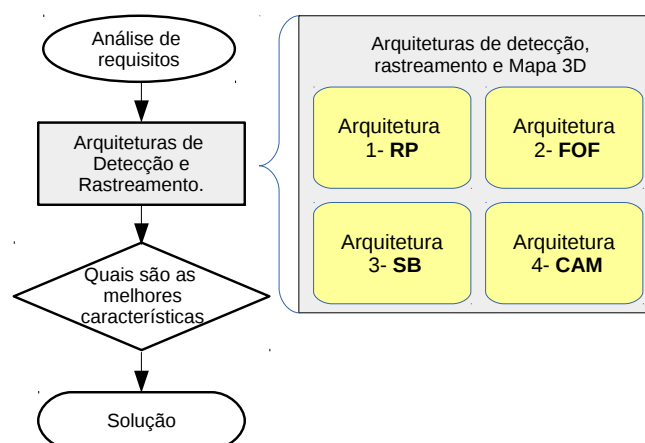
As premissas que são relevantes para a realização de uma análise comparativa entre as técnicas implementadas são as seguintes: (1) eficiência na detecção dos objetos dinâmicos; (2) menor poder de processamento e menor uso de memória. A primeira premissa é um requisito importante para o processo de reconstrução da trajetória produzida pelo OD e permite estimar a sua distância, direção, velocidade e possibilidade de colisão com o DV. A segunda premissa é essencial para que o sistema implementado produza respostas no tempo adequado para o DV poder trafegar com segurança e também para possibilitar a diminuição do consumo de energia e do seu tamanho.

Este capítulo aborda, por meio de uma nova metodologia, um estudo ainda não realizado

e pode ser útil para apresentar quais são as técnicas mais indicadas para implementação em sistemas de apoio para os DVs e em casos que se assemelham, como robôs, carros ou outros tipos de veículos autônomos. Para isso, devem-se considerar os objetivos e os contextos desses sistemas, pois, a escolha das técnicas pode variar de acordo com as suas necessidades. As técnicas devem ser escolhidas após considerar os seguintes aspectos: se o dispositivo será instalado em um corpo móvel ou estático; se ocorrerão movimentos leves ou intensos nos SA e se é possível segmentar os Objetos Dinâmicos num período que permita capturar as posições por onde o objeto se locomove.

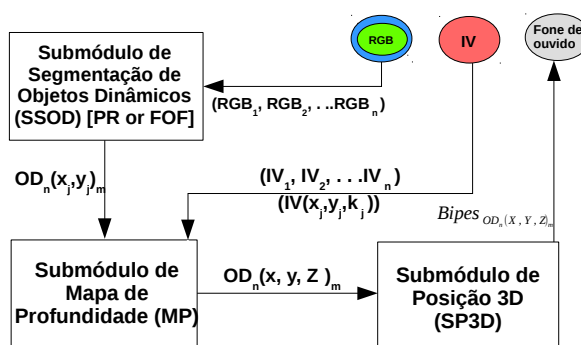
## 4.2 Detecção e Rastreamento de Objetos Dinâmicos (DROD)

Nesta seção é apresentada uma nova abordagem para analisar a detecção e o rastreamento de OD e possibilitar a reprodução e o armazenamento do trajeto executado. Com o armazenamento desse trajeto, é possível fornecer ao DV as posições que podem ser consideradas mais seguras para trafegar. Essa abordagem é composta por: uma análise de requisitos específica para o contexto dos DVs; pelo desenvolvimento de quatro arquiteturas adaptadas para o contexto dos DVs; pela geração de um mapa 3D composto de trajetos realizados por ODs e por uma análise comparativa entre as arquiteturas, visando obter a melhor solução após observar as melhores características de cada arquitetura. Essa abordagem pode ser considerada como uma nova metodologia específica para análise da eficiência e do custo computacional de arquiteturas que detectam ODs. O fluxograma da metodologia proposta é ilustrada na Figura 4.3.



**Figura 4.3:** Fluxograma da metodologia aplicada para analisar as arquiteturas de detecção e rastreamento de objetos dinâmicos e definir a solução adequada para o contexto proposto.

A análise de requisitos baseou-se nas informações obtidas na *GDTA* apresentada no Capítulo 3. A partir dos questionamentos apresentados, decidiu-se analisar e implementar as técnicas



**Figura 4.4:** Organização dos submódulos das Arquiteturas 1 (A1-RP) e 2 (A2-FOF).

mais aplicadas para detecção e rastreamento de objetos dinâmicos. As técnicas implementadas foram customizadas para arquiteturas que permitem o seu funcionamento. As técnicas usadas foram as seguintes: 1- Reconhecimento de Padrões (RP); 2- Fluxo óptico de Farneback (FOF); 3- Subtração de *Background* (SB) e 4- Continuously Adaptive Meanshift (CamShift). Para cada arquitetura, diferentes abordagens foram experimentadas, entretanto, apenas as que apresentaram os melhores resultados foram detalhadas nesta tese. Baseado nas técnicas implementadas e nas modificações visando atender o objetivo do trabalho, uma análise comparativa entre as arquiteturas foi realizada e, por fim, uma solução foi proposta. Todos os módulos presentes nas arquiteturas foram implementados com auxílio da biblioteca OpenCV e da linguagem C++.

### 4.2.1 Arquitetura 1 - Reconhecimento de padrões usando Haar-cascade (A1-RP)

A primeira arquitetura projetada para a detecção e rastreamento de Objetos Dinâmicos se baseou na técnica de Reconhecimento de Padrões. Essa técnica foi implementada no Submódulo de Segmentação de Objetos Dinâmicos (SSOD), apresentado na Figura 4.4.

A arquitetura A1-RP é composta por três módulos, dois sensores de aquisição (RGB e IV) e um dispositivo de saída (Fone de ouvido). Para essa arquitetura, um algoritmo (Algoritmo 2) foi implementado para apresentar a ordem das funcionalidades aplicadas. Cada linha apresenta um cálculo ou uma função com a passagem dos seus parâmetros. Ressalta-se que a Figura 4.4 também apresenta a arquitetura A2-FOF, diferenciando-se da A1-RP apenas pela escolha da técnica de segmentação, implementada no SSOD.

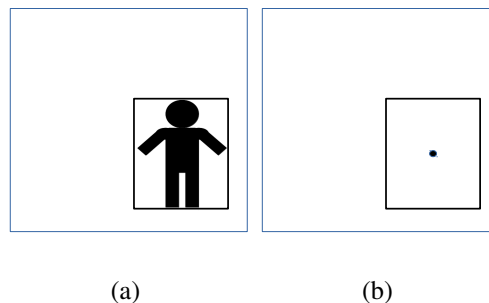
#### 4.2.1.1 Sensores de Aquisição, Mapa de Profundidade e Posição 3D

Todas as arquiteturas apresentadas neste trabalho usam o Kinect como dispositivo de entrada. Na A1-RP usou-se apenas a câmera IV (linha 1 e 2, Alg. 2) e a câmera RGB (linha 1 e 3, Alg. 2).

O Mapa de Profundidade é responsável por retornar a distância ( $Z$ ) de qualquer coordenada de imagem ( $x, y$ ) desejada. Os detalhes desse método são descritos na Seção 3.2.1 do Capítulo 3. O Módulo de Posição 3D é o responsável por mapear a posição de um Objeto Dinâmico a cada instante em que é detectado, realizando o seu percurso. Os detalhes deste método são descritos na Seção 3.2.2.2 do Capítulo 3.

#### 4.2.1.2 SSOD usando Reconhecimento de Padrões

A técnica implementada para realizar a segmentação dos Objetos Dinâmicos foi baseada na teoria de Reconhecimento de Padrões. No entanto, essa técnica detecta apenas as classes previamente definidas. Com a definição de uma classe, é necessário realizar a extração de características e treiná-las para que seja possível realizar a sua detecção de modo automático. No SSOD implementou-se apenas a classe que possui características do corpo de uma pessoa (ver imagem (a) da Figura 4.5). Com uma detecção próxima do tempo real, foi possível analisar e compreender os movimentos produzidos pelas pessoas. Neste sistema aplicou-se a técnica *Haar-cascade* proposta por Viola *et al.* (2001). Essa técnica tem como principal característica o alto desempenho na detecção de uma instância presente em algum ambiente. Na imagem (b) da Figura 4.5 é possível observar o centro de massa calculado da pessoa ilustrada na imagem (a).



**Figura 4.5:** (a) - Demonstração do contorno de uma pessoa detectada. (b) - Cálculo do centro de massa da pessoa detectada na imagem (a).

---

**Algorithm 2:** ALGORITMO DE DETECÇÃO E RASTREAMENTO DE OBJETOS DINÂMICOS USANDO RP OU FOF.

---

**Input:**  $Vídeo_{IV}$  &  $Vídeo_{RGB}$   
**Output:** Detecção e rastreamento de OD

```

1 while (Capture  $Vídeo_{IV}$  &  $Vídeo_{RGB}$ ) do
2    $(IV_1, IV_2, \dots, IV_n) \leftarrow \text{convertFrame}(Vídeo_{IV})$ 
3    $(RGB_1, RGB_2, \dots, RGB_n) \leftarrow \text{convertFrame}(Vídeo_{RGB})$ 
4    $DO_{Segment} \leftarrow \text{Segmentation}([PRorFOF], RGB_1)$ 
5    $\text{findContours}(DO_{Segment}, DO_{Contour})$ 
6    $DO(x_j, x_j, Z_j) \leftarrow \text{CMW}(DO_{Contour})$ 
7    $IV(x_j, y_j, k_j) \leftarrow \text{SSOD}(DO(x_j, x_j))$ 
8    $(MP_1, MP_2, \dots, MP_n(x_j, y_j, Z_j)) \leftarrow MP_{frames}((IV_1, IV_2, \dots, IV_n(x_j, y_j, k_j)))$ 
9    $DO(X, Y, Z)_{mm} \leftarrow \text{SP3D}(DO(x_j, y_j, Z_j))$ 
10 return  $DO(X, Y, Z)_{mm}$ 

```

---

## 4.2.2 Arquitetura 2 - Fluxo Óptico de Farneback (A2-FOF)

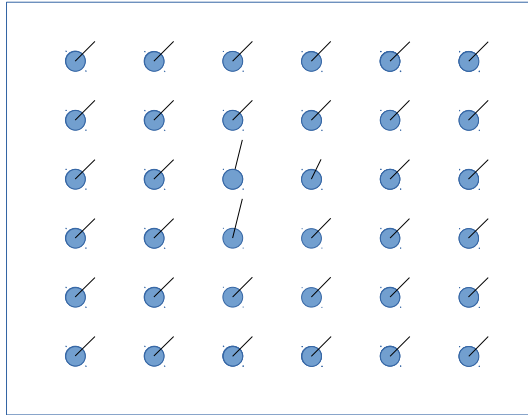
A segunda técnica implementada para a análise de Objetos Dinâmicos foi a do Fluxo Óptico com Mapas densos, de Gunner Farneback (BRADSKI; KAEHLER, 2008; MAYBECK, 1979). Essa é mais uma técnica interessante para detectar diferentes fluxos que ocorrem na imagem. Essa técnica foi adaptada devido aos impactos que o SA sofre na movimentação do DV. Dessa forma, foram criadas soluções para que os Objetos Dinâmicos fossem detectados independentemente do movimento do SA. A Figura 4.4 apresenta a arquitetura da primeira (Reconhecimento de Padrões) e da segunda (Fluxo Óptico de Farneback) abordagem. Isso ocorre devido a todos os dados e módulos possuírem as mesmas características e ações. A única diferença é a técnica aplicada para detecção e rastreamento, exposto no SSOD. Da mesma forma, o Algoritmo 2 também representa as mesmas instruções, com a alternativa de escolha da técnica apresentada na linha 4.

### 4.2.2.1 SSOD usando FOF

Para facilitar a compreensão da adaptação implementada sobre a técnica de Fluxo Óptico de Farneback, foi desenvolvido o Algoritmo 3 que determina a ordem das instruções. O objetivo da primeira etapa dessa técnica é distribuir um conjunto de pontos que permitirão determinar os movimentos (fluxos) existentes, tanto no SA como dos Objetos Dinâmicos. Para a técnica Fluxo Óptico de Farneback detectar os ODs, foram criados vetores para cada ponto na imagem. Esses vetores apresentam a velocidade e a direção, respectivamente, por meio do seu tamanho e do seu ângulo. Quanto maior é o vetor, maior é o fluxo detectado em determinado ponto do mapa denso. A Figura 4.6 apresenta os pontos (linha 3, Alg. 3) azuis distribuídos no frame e os



vetores são apresentados por retas pretas que são geradas (linha 5, Alg. 3) quando detectadas as mudanças provenientes do movimento do objeto. Na Figura 4.6, é possível observar a presença de movimento no SA e de um Objeto Dinâmico. O tamanho e os ângulos dos vetores permitem diferenciar o movimento do SA com relação ao movimento de um Objeto Dinâmico.



**Figura 4.6:** Simulação dos vetores de velocidade implementados na técnica de Fluxo Óptico de Farneback com a detecção de movimentos divergentes.

Para diferenciar a detecção de um movimento proveniente do SA, com relação a de um objeto dinâmico, calculou-se a média dos tamanhos dos vetores usando a distância euclidiana (linha 8, Alg. 3). A distância euclidiana ( $DEuclidian \leftarrow \sqrt{(Vi_y - Vf_y)^2 + (Vi_x - Vf_x)^2}$ ) permitiu calcular o tamanho do vetor, em que  $Vi_y$  e  $Vi_x$  são os valores iniciais dos vetores nos eixos  $(x,y)$  e  $Vf_y$  e  $Vf_x$  são os valores finais. A média se baseou na somatória dos tamanhos dos vetores dividida pela quantidade de pontos ( $AvgDEuclidian \leftarrow \frac{\sum_{i=1} DEuclidian_i}{AmountDEuclidian}$ ).

Da mesma forma, criou-se a média (linha 9, Alg. 3) dos ângulos produzidos com esses movimentos. Os ângulos foram gerados pela função  $atan2$  ( $Angle \leftarrow atan2(Vi_y - Vf_y, Vi_x - Vf_x)$ ) e os parâmetros são os mesmos já citados. Analisando a média das distâncias euclidianas, é possível verificar os pontos (linha 13, Alg. 3) que sofreram maiores fluxos,  $|DEuclidian > AvgDEuclidian|$ . Assim, para cada ponto (linha 14, Alg. 3), também foi verificado se houve uma mudança de fluxo e direção acima da média ( $AvgAngle \leftarrow \frac{\sum_{i=1} Angle_i}{AmountAngle}$ ). Quando o SA sofre algum movimento ou quando é detectado um Objeto Dinâmico, os vetores têm seu tamanho modificado.

Com a mesma linha de raciocínio, quando há movimento de câmera, os ângulos dos vetores de velocidade são similares, porém, caso algum ângulo possua uma alta divergência com relação à média ( $Angle > AvgAngle || Angle < AvgAngle$ ), a probabilidade de ter sido detectado um OD é grande. Para esse método, além das médias, implementaram-se também os conceitos de covariância (linha 15, Alg. 3) para o tamanho e a direção dos vetores produzidos.

O objetivo dessa adaptação é detectar o objeto dinâmico mesmo quando o SA sofra algum tipo de movimento. Após a detecção, é possível analisar sua direção, estimar a velocidade, calcular a área e reconstruir sua trajetória. Essas informações são obtidas após o cálculo do centro de massa e dos vetores de velocidade referentes aos objetos dinâmicos. Da mesma forma que a arquitetura usando Reconhecimento de Padrões e como todas as outras implementadas, os  $Bipes_n OD(X, Y, Z)_m$  são produzidos baseados nas distâncias (em milímetros) dos OD.

---

**Algorithm 3:** ALGORITMO COM ADAPTAÇÕES PARA DETECTAR OD USANDO FLUXO ÓPTICO DE FARNEBACK MESMO COM MOVIMENTOS NO SA.

---

```

Input:  $Vídeo_{RGB}$ 
Output: Detecção de Fluxo
1 while (Capture VídeoRGB) do
2    $((RGB_1, RGB_2, \dots, RGB_n)) \leftarrow \text{convertFrame}(Vídeo_{RGB})$ 
3   CreateDenseMap(Pontos)
4   OpticalFlow  $\leftarrow$  CalcOpticalFlow( $RGB_1, RGB_2, Pontos$ )
5   aux1  $\leftarrow$  1
6   while ( $aux < AmountPoints$ ) do
7     CreateVectors(Pontos[aux1], OpticalFlow)
8      $AVG_{Euclidian} \leftarrow dEuclidian(V_{iy}, V_{ix}, V_{fy}, V_{fx})$ 
9      $AVG_{Angle} \leftarrow atan2(V_{iy}, V_{ix}, V_{fy}, V_{fx})$ 
10    aux1++
11  aux2  $\leftarrow$  1
12  while ( $aux < AmountPoints$ ) do
13    if ( $Point[aux] > AvgEuclidian$ ) then
14      if ( $(Point[aux2] > AVG_{Angle}) \parallel (Point[aux] < AVG_{Angle})$ ) then
15        if ( $CovEuclidian > value\ 1$ )  $\&\&$  ( $CovAngle > value\ 2$ ) then
16          Area  $\leftarrow$  (Point[aux])
17          aux2++
18 return Contour[Area]

```

---

### 4.2.3 Arquitetura 3 - Subtração de *Background* (A3-SB)

A terceira técnica de segmentação aplicada foi a de Subtração de *Background*. Essa técnica é muito usada para identificar os objetos que estão em movimento. Em situações nas quais há presença de rotação, translação ou escala em *frames* subsequentes, deve ser implementada alguma alternativa para que o objeto em movimento seja segmentado com qualidade. A Figura 4.7 apresenta a arquitetura da terceira abordagem implementada. Ela foi adaptada para o DV usando a técnica de Subtração de *Background* mesmo com o sensor de aquisição sofrendo movimento.

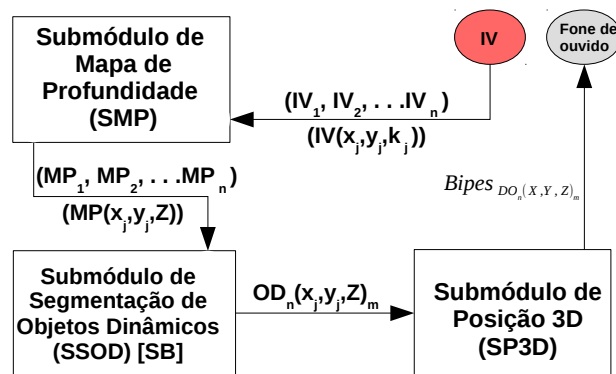


Figura 4.7: Organização dos submódulos da Arquitetura 3 (A3-SB).

O SSOD usando Subtração de *Background* apresentado na Figura 4.7 foi modelado com o uso da técnica de *Mixture of Gaussians (MOG)* (BRADSKI; KAEHLER, 2008). Essa técnica (*SubtracaoBackground<sub>MOG</sub>*) fornece um *foreground* de qualidade e foi escolhida devido a sua característica de alto desempenho e de aplicação em ambientes internos (STAUFFER; GRIMSON, 1999). Nessa arquitetura, dois métodos foram experimentados com o uso da Subtração de *Background*.

O primeiro método foi implementado com a estabilização do vídeo. Para que a estabilização fosse possível, estimaram-se os coeficientes de transformação geométrica entre um *frame* adquirido em um determinado instante com o *frame* subsequente. A princípio, nesse trabalho, foram usadas as imagens RGB para obtenção dos coeficientes de transformação entre uma sequência de imagens. Porém, a cada mudança de iluminação, a segmentação foi prejudicada. Sendo assim, implementou-se nesta arquitetura a Subtração de *Background* sobre o mapa de profundidade, fornecido pelo sensor infravermelho. Verificou-se uma segmentação semelhante ao obtido com a imagem RGB e independente da iluminação, entretanto, com mais dificuldades no processo de estabilização de câmera.

A técnica de detecção de cantos aplicada em conjunto com a técnica de fluxo óptico foi uma escolha interessante para detectar as mudanças de uma sequência de *frames*. Essas mudanças podem ser compreendidas segundo os padrões de intensidade de um conjunto de *pixels*. Dessa forma, o filtro de Kalman foi aplicado aos coeficientes de transformação produzidos entre uma sequência de imagens. Os detalhes de implementação desse método podem ser vistos no Algoritmo 4.

É importante ressaltar que a estimativa de coeficientes de transformação deve ser automática e rápida. Uma outra forma para detectar as mudanças entre uma sequência de *frames* é por meio de uma unidade de medição inercial (*Inertial Measurement Unit*) implantada no SA. Esse sensor

pode fornecer os coeficientes das transformações geométricas para o sistema de estabilização. Essa opção está prevista para ser executada em futuros trabalhos.

---

**Algorithm 4:** ALGORITMO DE DETECÇÃO DE OBJETOS DINÂMICOS USANDO SUB-TRAÇÃO DE *Background* E ESTABILIZAÇÃO DE VÍDEO.

---

**Input:** Acelerômetro, Câmera IV e Câmera RGB

**Output:** Objeto Dinâmico Segmentado

```

1 //Obtém Vídeo IV (Mapa de Profundidade); Vídeo RGB da Câmera e a movimentação pelo Acelerômetro
2 while (Capturar MP, RGB, Amov) do
3   //Converte Vídeo RGB e Vídeo MP para Quadros.
4   ( $\langle RGB_1, RGB_2, \dots, RGB_n \rangle$ )  $\leftarrow$  convertFrame(RGB)
5   ( $\langle MP_1, MP_2, \dots, MP_n \rangle$ )  $\leftarrow$  SSOE
6   //Detecta os cantos da imagem MP com o método de Harris (HARRIS; STEPHENS, 1988)
7   ( $\langle Corners[]_1 \rangle$ )  $\leftarrow$  CornerDetect(MP1)
8   //Calcula o fluxo óptico (TRUCCO; VERRI, 1998) entre as imagens (MP1 e MP2)
9   OpticalFlow( MP1, MP2, Corners[]1, Corners[]2)
10  //Estima os coeficientes de transformação Afim entre as imagens (HARRIS; STEPHENS, 1988)
11  T  $\leftarrow$  estimateAffineTransform(Corners[]1, Corners[]2 )
12  //Aplica o KF nos coeficientes de transformação T e disponibiliza Tk
13  KalmanFilter(T, Tk)
14  //Aplica Tk na imagem de profundidade CurrDmap e estabiliza video
15  ApplyTransformation(Tk, MP2, MP2Stabilized)
16  //Inicializa o modelo de background
17  modelo  $\leftarrow$  BSMOGmodel()
18  //Gera-se o modelo de background
19  BSMOGUpdate( modelo, MP2Stabilized )
20  //Encontra o foreground pelo método MOG
21  BSMOG( modelo, MP2Stabilized )
22  //Detecta OD
23  while ( $n \leq Qtde_{OD}$ ) do
24    //Rastreia 10 posições do OD
25    while ( $m \leq 10$ ) do
26      //Detecta contornos (OD) por Bradski et al. (2008)
27      findContours(MOG, contoursOD)
28      //Calcula a posição e a largura dos objetos (tracking 10)
29      ( $\alpha[\langle OD(x_j, y_j, Z_j).cm, OD(x_j, y_j, Z_j).p1, OD(x_j, y_j, Z_j).p2 \rangle_m]$ )  $\leftarrow$  CMW(contoursOD)
30      m++
31    //Converte a posição e a largura dos objetos dinâmicos em (mm) (Equações 3.1 e 3.2)
32    ( $\langle OD(X, Y, Z)_{[m]}.cm, OD(X, Y, Z)_{[m]}.p1, OD(X, Y, Z)_{[m]}.p2 \rangle$ )  $\leftarrow$  SP3D( $\alpha_{[m]}$ )
33    n++
34 return OD(X, Y, Z)m

```

---

O processo de estabilização de vídeo, com o uso das técnicas de detecção de cantos, de fluxo óptico e de transformação geométrica, gerou um alto custo computacional e passou a interferir na geração dos resultados próximos do tempo real. Devido a esse custo computacional e às

dificuldades em detectar uma diversidade de pontos de apoio nas imagens IV, para análise do fluxo óptico, decidiu-se manter a opção do processo de estabilização de vídeo em segundo plano e usar a Subtração de *Background* de maneira diferente, explicada a seguir.

A segunda forma desenvolvida para aplicação da técnica de Subtração de *Background* foi por meio do aumento do Limiar (*varThreshold*). Esse limiar é a distância quadrática entre um *pixel* da imagem e uma amostra obtida (STAUFFER; GRIMSON, 1999). Esse limiar permite definir se alguns *pixels* detectados pertencem ou não à amostra, fornecendo uma segmentação com maior qualidade. Quanto maior é o limiar, maior é o descarte de correspondência entre os *pixels* vizinhos, o que permite segmentar a região mais central do objeto dinâmico.

Como pode ser visto na arquitetura 3 (4.7), os parâmetros são diferentes, comparados com os parâmetros da arquitetura 1, devido à Subtração de *Background* usar o próprio MP para a realização da segmentação. A A3-SB possui características diferentes das arquiteturas anteriores, pois nela usa-se apenas o sensor IV como SA (linha 1, Alg. 5). O Mapa de Profundidade recebe os dados fornecidos pelo sensor IV (linha 2, Alg. 5) e gera o mapa de profundidade contendo as distâncias de todas as coordenadas ( $MP(x_j, y_j, Z_j)$ ) apresentadas na linha 3, Alg. 5). O SSOD, por sua vez, aplica a Subtração de *Background* para detectar os objetos dinâmicos (linha 4, Alg. 5) para, posteriormente, calcular o seu contorno (linha 5, Alg. 5), o seu centro de massa (linha 6, Alg. 3) e submeter ao SP3D os centros de massa ( $DO_n(x_j, y_j, Z)_m$ ) com a distância ( $Z$ ) já obtida em milímetros (linha 7, Alg. 5). O SP3D é então responsável por converter os valores  $(x_j, y_j)$  em *pixel* para milímetros e fornecer a posição 3D do OD (linha 8, Alg. 1) para ser mapeado em um único sistema de referência. Por fim, o SP3D submete os bipes da posição calculada ( $\text{map}(X, Z)$  em milímetros) para o fone de ouvido.

---

**Algorithm 5:** ALGORITMO DE DETECÇÃO E RASTREAMENTO DE OBJETOS DINÂMICOS USANDO SB

---

**Input:**  $Vídeo_{IV}$

**Output:** Detecção e rastreamento de OD

```

1 while (Capture  $Vídeo_{IV}$ ) do
2    $(\langle IV_1, IV_2, \dots, IV_n \rangle) \leftarrow \text{convertFrame}(Vídeo_{IV})$ 
3    $(\langle MP_1, MP_2, \dots, MP_n(x_j, y_j, Z_j) \rangle) \leftarrow MP_{frames}(\langle IV_1, IV_2, \dots, IV_n(x_j, y_j, k_j) \rangle)$ 
4    $DO_{Segment} \leftarrow \text{Segmentation}([SB], threshold, IV_1)$ 
5    $findContours(DO_{Segment}, DO_{Contour})$ 
6    $DO(x_j, y_j, Z_j) \leftarrow CMW(DO_{Contour})$ 
7    $DO(X, Y, Z)_{mm} \leftarrow SP3D(DO(x_j, y_j, Z_j))$ 
8 return  $DO(X, Y, Z)_{mm}$ 

```

---

#### 4.2.4 Arquitetura 4 - Continuously Adaptive Meanshift (A4-CAM)

A quarta técnica aplicada neste estudo foi a CamShift. Essa técnica é muito interessante para realizar o rastreamento em cores, contudo, necessita de uma outra técnica para detectar o OD e selecionar a área a ser rastreada. Por meio da Subtração de *Background*, do Fluxo Óptico de Farneback e do Reconhecimento de Padrões é possível detectar um OD e criar um marcador. Nessa pesquisa, selecionou-se o centro de massa do objeto detectado para ser rastreado pela técnica CamShift. A Figura 4.8 apresenta a arquitetura da quarta abordagem implementada. Ela foi adaptada para o DV usando a técnica de CamShift mesmo com os SAs sofrendo movimento. O algoritmo 5 apresenta a ordem das instruções e os parâmetros implementados para o seu funcionamento.

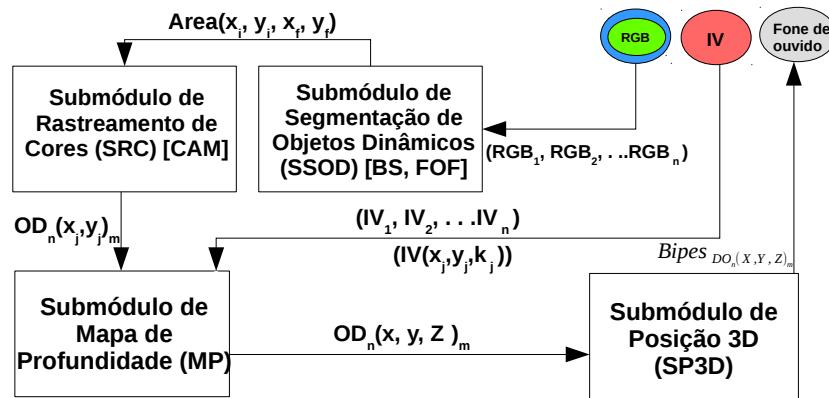


Figura 4.8: Arquitetura 4 (A4-CAM).

A arquitetura 4 (ver Figura. 4.8) é semelhante às arquiteturas 1 e 2. A diferença é que a técnica CamShift necessita de um método (linha 3, Alg. 5) que realiza a detecção de um OD e selecione (marque) uma área ( $\text{Área}(x_i, y_i, x_f, y_f)$ ) para que seja possível rastreá-la. Nessa arquitetura, implementou-se o Fluxo Óptico de Farneback para selecionar a região onde foi detectado o OD. A partir dessa seleção, todos os *pixels* vizinhos que possuem uma coloração semelhante são segmentados (linha 4, Alg. 5) e seu contorno (linha 5, Alg. 5) é efetuado. Da mesma forma que as outras arquiteturas, o centro de massa ( $DO_n(x_j, y_j)_m$ ) é calculado (linha 6, Alg. 5) para que os objetos detectados (n) em diferentes instantes (m) possam ser rastreados e mapeados no mapa 3D. O mapa de profundidade calcula a distância baseada na posição  $(x_j, y_j)$  recebida (linha 8 e 9, Alg. 5) e o SMP (linha 10, Alg. 5) submete ao SP3D o objeto dinâmico ( $DO_n(x_j, y_j, Z_j)_m$ , com as posições  $(x_j, y_j)$  em *pixels* e  $Z_j$  em milímetros). Dessa forma, o SP3D faz a conversão dos valores  $(x_j, y_j)$  em *pixel* para milímetros e fornece a posição 3D do OD.

---

**Algorithm 6:** ALGORITMO DE DETECÇÃO E RASTREAMENTO DE OBJETOS DINÂMICOS USANDO CAMSHIFT.

---

**Input:**  $Vídeo_{IV}$  &&  $Vídeo_{RGB}$   
**Output:** Detecção e rastreamento de OD

```

1 while (Capture  $Vídeo_{IV}$  &&  $Vídeo_{RGB}$ ) do
2    $((RGB_1, RGB_2, \dots, RGB_n)) \leftarrow \text{convertFrame}(Vídeo_{RGB})$ 
3    $Area_{Select} \leftarrow \text{Segmentation}([FOF], RGB_1)$ 
4    $DO_{Segment} \leftarrow \text{Segmentation}([CAM], Area_{Select})$ 
5    $findContours(DO_{Segment}, DO_{Contour})$ 
6    $DO(x_j, x_j) \leftarrow \text{CMW}(DO_{Contour})$ 
7    $((IV_1, IV_2, \dots, IV_n)) \leftarrow \text{convertFrame}(Vídeo_{IV})$ 
8    $IV(x_j, y_j, k_j) \leftarrow \text{SSOD}(DO(x_j, x_j))$ 
9    $((MP_1, MP_2, \dots, MP_n(x_j, y_j, Z_j))) \leftarrow MP_{frames}((IV_1, IV_2, \dots, IV_n(x_j, y_j, k_j)))$ 
10   $DO(X, Y, Z)_{mm} \leftarrow SP3D(DO(x_j, y_j, Z_j))$ 
11 return  $DO(X, Y, Z)_{mm}$ 

```

---

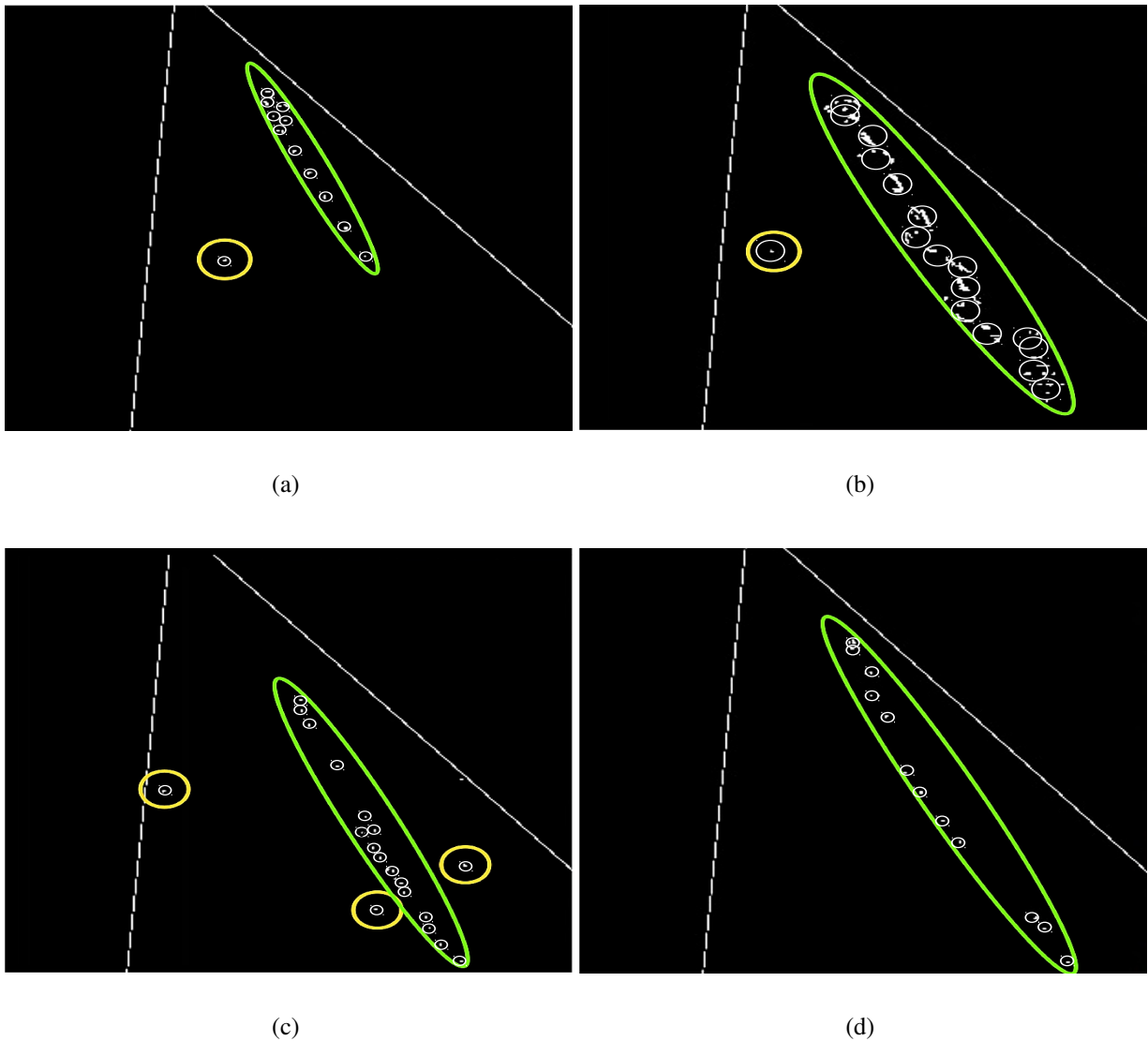
### 4.3 Metodologia desenvolvida para comparar as arquiteturas DROD

Esta seção apresenta em detalhes a metodologia desenvolvida para comparar a efetividade e a velocidade de detecção das arquiteturas implementadas para DROD. Essa comparação permite verificar qual é a arquitetura mais apropriada para um determinado ambiente.

Na Figura 4.9, as imagens apresentam como o ambiente e as posições dos Objetos Dinâmicos detectados podem ser mapeados em milímetros. Nesta figura, as imagens apresentam a metodologia aplicada para comparar as quatro arquiteturas com relação à velocidade de detecção e a efetividade. Nessas imagens (Mapa(X, Z)), é possível visualizar a quantidade de detecção (pontos em branco) criada até o final do rastreamento. Como o sistema registra o tempo inicial e o tempo final do rastreamento, ao término é possível calcular a quantidade de detecções realizadas por segundos (Velocidade de detecção) para cada arquitetura. Os círculos verdes demonstram o rastreamento do objeto dinâmico até sair da área de abrangência do SA. Como o sistema registra o momento (horário) inicial e o final da detecção dos objetos dinâmicos, é possível calcular o número de detecções realizado por segundo. Este mapa também permite a identificação de erros durante o processo de detecção (efetividade). Ressalte-se que é possível identificar, de modo visual, quais detecções foram mapeadas corretamente.

Nas imagens da Figura 4.9, os círculos amarelos representam erros no processo de detecção. As imagens (a, b, c, d) apresentam a aplicação da metodologia com as seguintes técnicas: Arquitetura 1 - Reconhecimento de padrões (A1-PR), Arquitetura 2 - Fluxo óptico de Farneback (A2-FOF), Arquitetura 3 - Subtração de Background (A3-SB) e Arquitetura 4 - Continuously

Adaptive Meanshift (A4-CAM), respectivamente. A planilha que contém os dados brutos obtidos, para comparar a eficácia de cada arquitetura implementada, pode ser obtida nos arquivos suplementares da Revista *Data in Brief* (CORDEIRO; PEDRINO, 2019b). Cada ponto renderizado no Mapa(X , Z) significa que um bipe foi emitido para o DV, para indicar a posição do objeto dinâmico no ambiente.

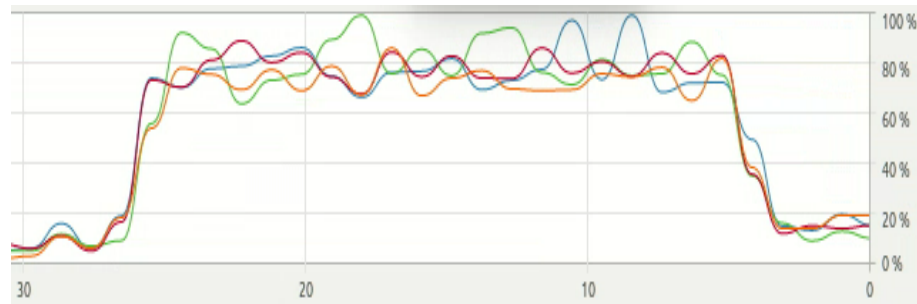


**Figura 4.9:** Medindo a efetividade com o Mapa(X , Z). (a) DROD usando A1-RP; (b) DROD usando A2-FOF; (c) DROD usando A3-SB; (d) DROD usando A4-CAM.

Para analisar o custo computacional de cada arquitetura, foi usado um software auxiliar (*System Monitor*) instalado no sistema operacional (Linux-Ubuntu) para contabilizar os gastos da *CPU* e da memória *RAM*. Para cada tomada de dados, foram adquiridas essas informações. Um exemplo pode ser visto nas imagens (a) e (b) da Figura 4.10. A imagem (a) apresenta os níveis de gasto da *CPU* e a imagem (b) os gastos com a memória *RAM*. Nas imagens é possível



observar que cada tomada durou um pouco mais de 20 segundos. Dessa forma, foi subtraído o gasto computacional inicial (pelo Sistema Operacional) dos gastos advindos do início do sistema de DROD.



(a)



(b)

**Figura 4.10: (a) Custo computacional de quatro processadores (CPUs) do início ao fim da execução de uma técnica de DROD. (b) Custo computacional da Memória RAM de 8 GB do início ao fim da execução de uma técnica de DROD.**

Todos os módulos foram implementados com a ajuda da biblioteca OpenCV (BRADSKI; KAEHLER, 2008). Neste trabalho, o Microsoft Kinect RGB-D foi fixado na cintura de uma pessoa com uma venda nos olhos, simulando um DV. Quando um objeto dinâmico foi detectado, independentemente da velocidade, um bipe foi emitido. Os experimentos realizados com as arquiteturas implementadas estão detalhados na Seção 5 e as discussões relacionadas com os resultados são apresentadas na Seção 6.

# Capítulo 5

## EXPERIMENTOS

---

---

Os experimentos foram divididos em duas seções: Experimentos realizados com o Módulo de Visão (Seção 5.1); Experimentos realizados com o Módulo de Fusão (Seção 5.2). Com os experimentos, foi possível analisar e readequar os sistemas que compõe esses módulos.

Nas seções 5.1 e 5.2 são apresentadas também a realização de experimentos com outras abordagens para comparar a efetividade das técnicas de detecção, do rastreamento de objetos dinâmicos e das técnicas de navegação, respectivamente. Essas comparações foram necessárias tanto para validar as propostas da presente tese como também para definir as técnicas a serem implantadas nos sistemas de visão e de fusão.

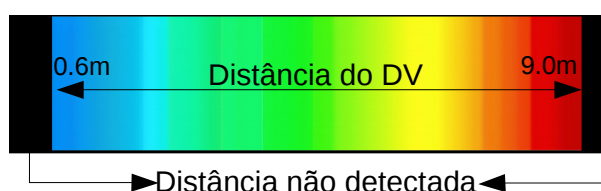
### 5.1 Experimentos realizados com o Módulo de Visão

Nesta seção, foram realizados experimentos com o Módulo de Visão para validar os seguintes submódulos: Submódulo de Mapa de Profundidade (Seção 5.1.1); Submódulo de Segmentação de Objetos Estáticos (Seção 5.1.2); Submódulo de Controle e Estabilização de Vídeo (Seção 5.1.3.3); Submódulo de Segmentação de Objetos Dinâmicos (Seção 5.1.3) e Submódulo de Posição 3D (Seção 5.1.4).

Todas as características foram obtidas em ambientes fechados com a presença de Objetos Estáticos próximos (entre 0.6m a 2.0m), Objetos Dinâmicos e trajetos executados por pessoas com tamanhos diferentes e em posições distintas. Para os Objetos Dinâmicos também foram modificadas as direções dos seus trajetos e a velocidade média no período de detecção e rastreamento. Esses submódulos visam fornecer ao Módulo de Fusão os requisitos que permitem classificar a situação e prover uma tomada de decisão.

### 5.1.1 Experimentos - Submódulo de Mapa de Profundidade

O Submódulo de Mapa de Profundidade fornece as principais informações para o SAS-DV. Esse módulo realiza seu processamento usando apenas dados do Sensor IV. O Sensor IV foi escolhido como a principal fonte de dados devido a sua independência relacionada com a iluminação e por fornecer a distância ( $Z$ ) de qualquer coordenada de imagem ( $x, y$ ) desejada. A Figura 5.1 apresenta um mapa de cores em que cada cor indica uma distância. As cores azuladas representam as menores distâncias (obstáculos próximos do DV) e as cores avermelhadas as maiores. As cores esverdeadas e amareladas representam as distâncias médias e a cor preta representa as regiões que estão fora do alcance do sensor Infravermelho.



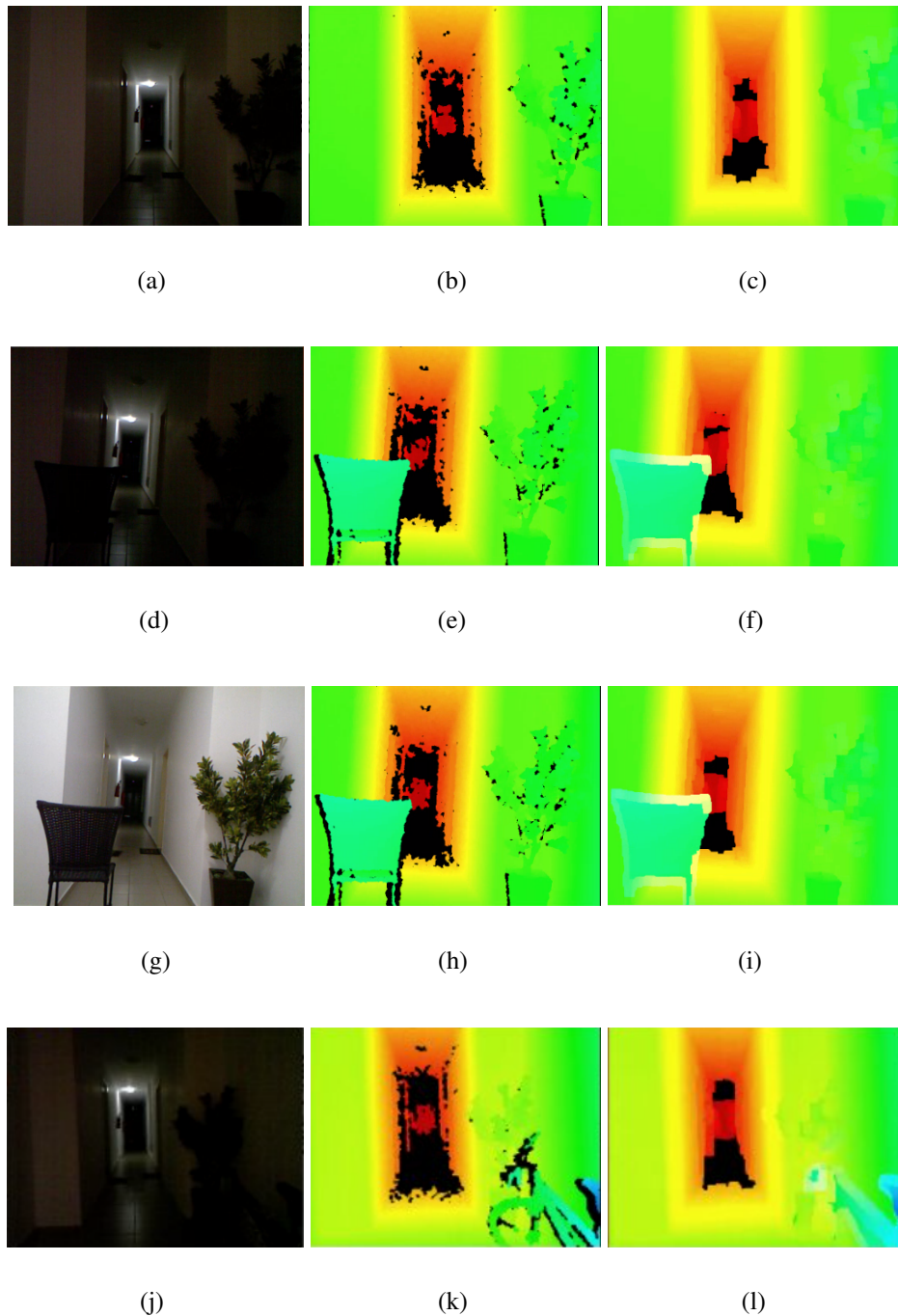
**Figura 5.1: Representação das distâncias em cores presentes nos mapas de profundidade.**

Justificado pela quantidade de características que o Sensor IV permite extrair, é importante analisar diversas formas de reduzir falhas ou distorções produzidas no processo de aquisição. Devido às suas características de aquisição de dados, algumas regiões não são detectadas no eixo  $Z$ . Dessa forma, definiu-se aplicar uma sequência de filtros de suavização para fornecer uma distância aproximada dessas regiões. Esses filtros são aplicados apenas em casos que necessitam detectar a distância de uma coordenada específica que não é obtida. Na Figura 5.2 são apresentadas algumas das imagens adquiridas nos experimentos do SAS-DV.

Os itens (a, d, g, j) da Figura 5.2 apresentam as imagens fornecidas pelo sensor RGB adquiridas em um ambiente interno. Os itens (b, e, h, k), da mesma figura, apresentam as imagens disponibilizadas pelo Sensor IV. Essas imagens (b, e, h, k) já possuem um tratamento simples de suavização. Os itens (c, f, i, l) ilustram imagens com filtros de suavização mais concentrados, inclusive com o uso de técnicas de erosão e dilatação. Por meio das cores apresentadas nessas imagens (itens (c), (f), (i), (l)), é possível notar que essa técnica já permite fornecer uma distância aproximada de regiões que não puderam retornar as distâncias. As regiões pretas apresentadas nas imagens (b), (e), (h) e (k) são posições onde o sensor IV não conseguiu obter suas respectivas distâncias. Os filtros suavizadores aplicados nas imagens (c), (f), (i), (l) permitiram aproximar as distâncias não detectadas e, dessa forma, apresentaram alternativas para solucionar problemas em casos de interferências de iluminação, como reflexão.

As imagens dos itens (d, e) e (g, h) permitem ilustrar a independência relativa à iluminação

que o Sensor IV possui. Nas imagens pode ser observado que os itens (e) e (h) detectam as distâncias mesmo com a alternância (presença ou ausência) de iluminação.

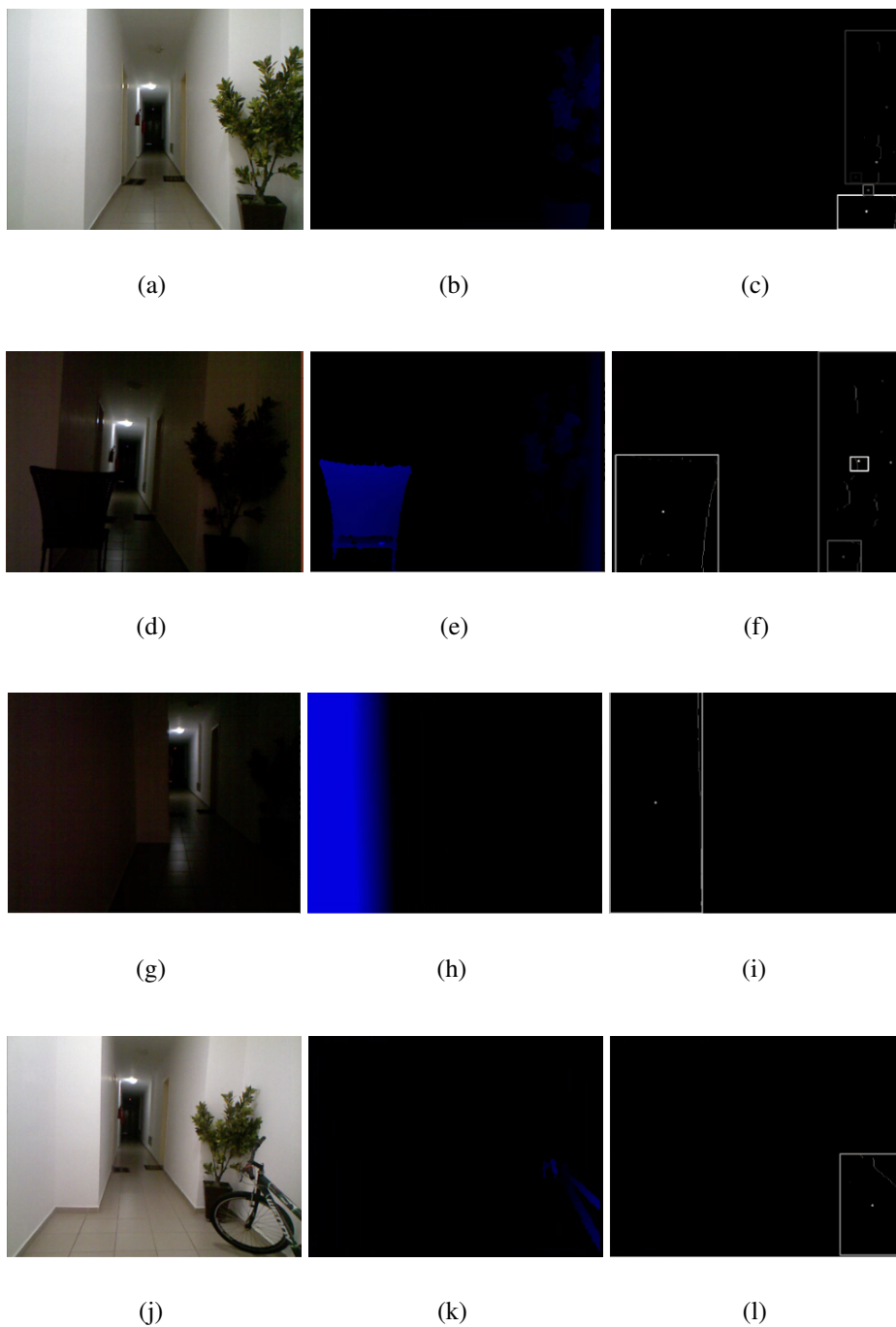


**Figura 5.2:** (a, d, g, j) - imagens fornecidas pelo sensor RGB; (b, e, h, k) - imagens disponibilizadas pelo Sensor IV (mapas de profundidade) com filtros de dilatação, erosão e suavização; (c, f, i, l) - imagens com filtros de dilatação, erosão e suavização mais concentrados.

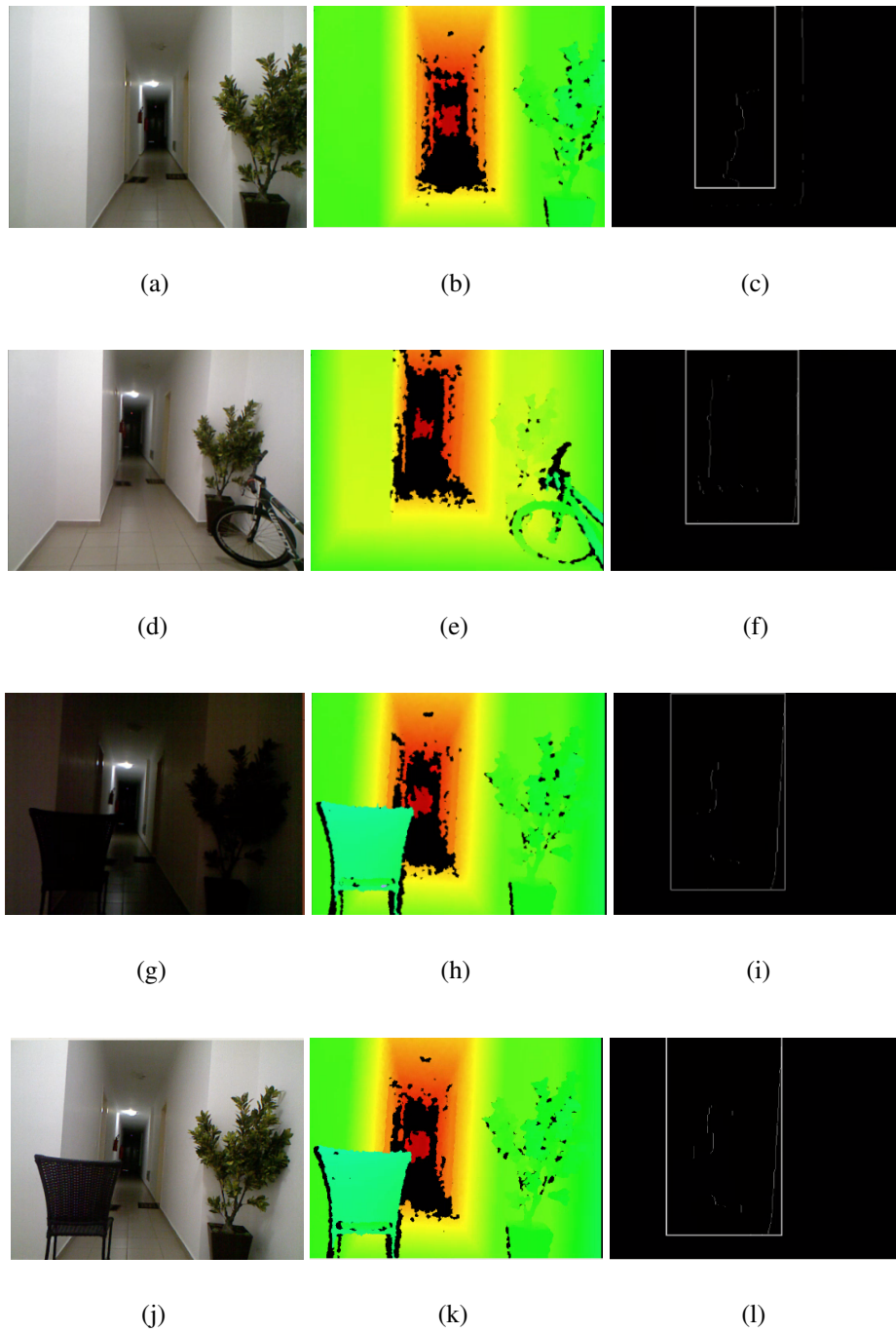
### 5.1.2 Experimentos - Submódulo de Segmentação de Objetos Estáticos

Os experimentos realizados com o SSOE são divididos em duas partes: detecção de objetos estáticos a menos de dois metros de distância do DV e detecção de passagens livres a mais de três metros de distância do DV. A Figura 5.3 apresenta, nos itens (a, d, g, j), imagens adquiridas num ambiente interno e com a presença de objetos estáticos a menos de dois metros de distância do DV. Como a segmentação dos objetos é baseada na distância fornecida pelo Submódulo de Mapa de Profundidade, a iluminação não interfere na segmentação desses objetos. Os itens (b, e, h, k) ilustram a segmentação dos objetos e os itens (c, f, i, l) apresentam a detecção do contorno e, conseqüentemente, o resultado do cálculo do centro de massa ( $CM_{OE}(X, Y, Z)$ ). Com esses dados, é possível calcular a largura e altura dos objetos.

A Figura 5.4 apresenta os experimentos realizados para a detecção da Passagem Livre (PL). Os itens (a, d, g, j) dessa figura apresentam as imagens fornecidas pelo Sensor RGB. Os itens (b, e, h, k) são os mapas de profundidade dos respectivos itens anteriores. Por meio desses mapas, é possível aplicar um filtro de limiarização para segmentar as regiões mais distantes. Dessa forma, os itens (c, f, i, l) ilustram as PLs detectadas pelo SSOE. É possível notar nesses últimos itens a possibilidade de calcular a largura e a altura da passagem, já que os cantos de cada contorno passam pelo SP3D para calcular sua posição no plano 3D. É importante ressaltar que ao detectar uma passagem livre (ver imagem (l)), mesmo com a presença de um objeto estático presente (pequena região) na mesma direção (ver imagem (k)), o sistema prioriza a presença do objeto estático no processo de mapeamento em determinada direção.



**Figura 5.3:** (a, d, g, j) - imagens adquiridas num ambiente interno; (b, e, h, k) - segmentação dos objetos estáticos; (c, f, i, l) - detecção do contorno dos objetos estáticos.



**Figura 5.4:** (a, d, g, j) - imagens fornecidas pelo Sensor RGB; (b, e, h, k) - imagens fornecidas pelo Sensor IV; (c, f, i, l) - detecção da largura da passagem livre.

### 5.1.3 Experimentos - Submódulo de Segmentação de Objetos Dinâmicos

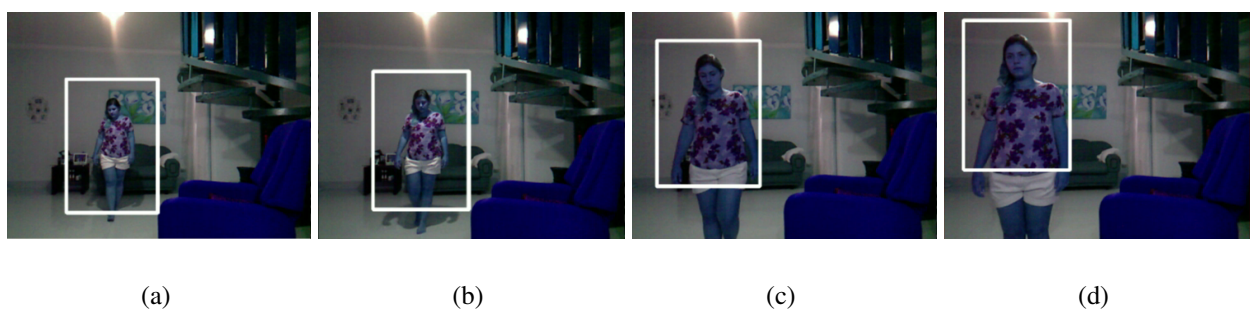
Os experimentos a seguir foram realizados após o Módulo de Visão detectar a presença de Objetos Dinâmicos. A detecção desses objetos ocorre após verificar a presença de movimentos na imagem, divergentes com os realizados pela câmera. No SAS-DV, a análise dos Objetos Dinâmicos foi realizada por quatro técnicas implementadas em arquiteturas específi-



cas: Arquitetura 1 - Reconhecimento de padrões usando *Haar-cascade* (A1-RP); Arquitetura 2 - Fluxo Óptico de Farneback (A2-FOF); Arquitetura 3 - Subtração de *Background* (A3-SB) e Arquitetura 4 - Continuously Adaptive Meanshift (A4-CAM). No capítulo de Resultados serão apresentadas algumas comparações e uma discussão a respeito das características e vantagens dessas técnicas.

### 5.1.3.1 Reconhecimento de Padrões

O Reconhecimento de Padrões com o uso do *Haar-cascade* foi a primeira técnica implementada no SAS-DV para a análise de Objetos Dinâmicos. Essa técnica consegue detectar um objeto dinâmico mesmo que ocorram movimentos na câmera. As técnicas de Reconhecimento de Padrões exigem um alto poder de processamento, porém, essa técnica conseguiu atingir uma quantidade de detecção que permitisse reproduzir o trajeto realizado por um objeto dinâmico. Na Figura 5.5, são apresentadas quatro imagens obtidas em uma pequena sequência de tempo com os experimentos realizados com a técnica de Reconhecimento de Padrões. Todas as imagens apresentam a detecção de objetos dinâmicos com a câmera em movimento. O quadro branco dessas imagens ilustra o contorno usado para calcular o centro de massa. Para esses experimentos, foram colocados os sensores em uma pessoa para simular o dispositivo em um DV. Tanto a pessoa detectada quanto a pessoa que simulou o DV estavam em movimento. Os vídeos dos experimentos usados com todas as técnicas abordadas nesta tese podem ser encontrados nos materiais suplementares dos artigos (CORDEIRO; PEDRINO, 2019b, 2019c).



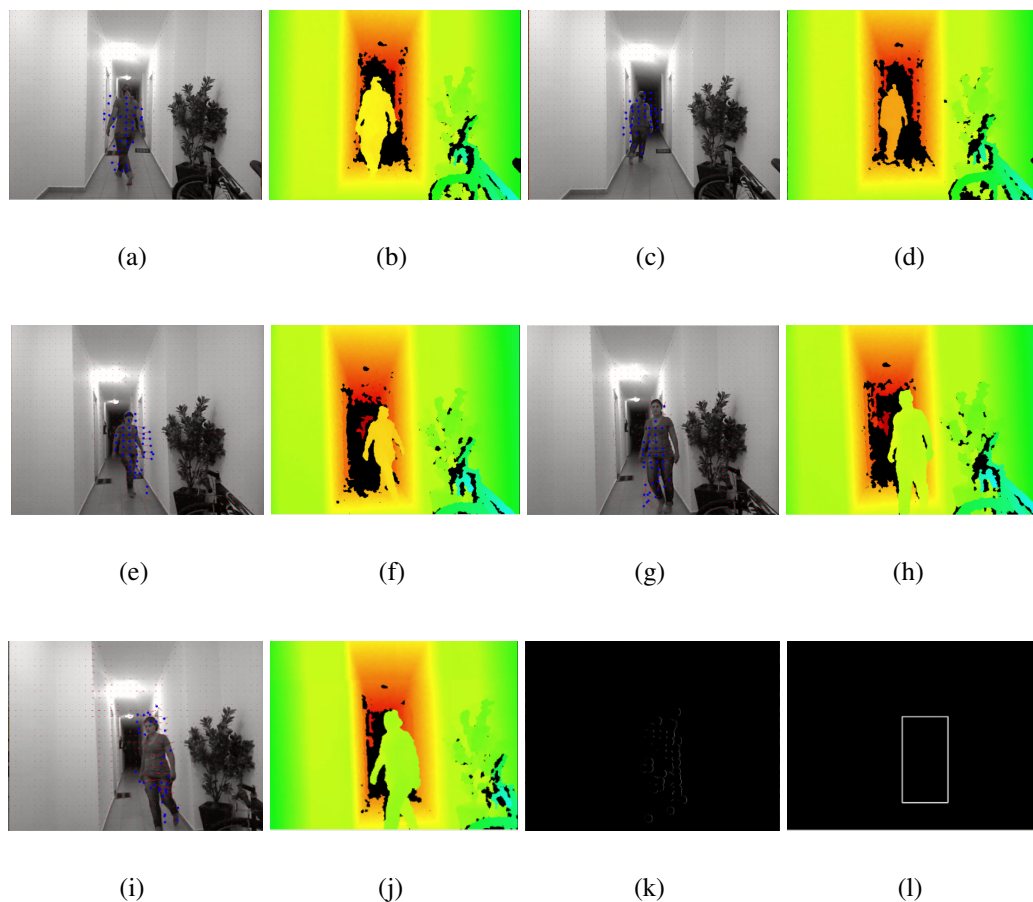
**Figura 5.5:** (a, b, c, d) - detecção de objetos dinâmicos em diferentes instantes com a técnica de Reconhecimento de Padrões.

### 5.1.3.2 Fluxo Óptico de Farneback

O FOP foi a segunda técnica implementada no SAS-DV para a análise de Objetos Dinâmicos. Essa técnica é a responsável por detectar um objeto dinâmico mesmo com a câmera em movimento. Para essa técnica não foi aplicado nenhum processo de estabilização de vídeo e



ela apresentou um bom desempenho computacional. Na Figura 5.6 são apresentadas três faixas obtidas em uma pequena sequência de tempo com os experimentos realizados com a técnica de Farneback. As duas primeiras faixas apresentam a detecção de Objetos Dinâmicos com a câmera em movimento. Cada faixa possui dois pares de imagens ((a, b), (c, d), (e, f), (g, h)) que fornecem as informações essenciais para reconstruir o trajeto executado. A primeira imagem do par apresentado é a detecção de um objeto dinâmico (pontos azuis) por meio da técnica de Farneback, e a segunda, o seu mapa de profundidade. A terceira faixa ((m, n, o, p)) ilustra respectivamente: a técnica de Farneback detectando uma pessoa em movimento; o mapa de profundidade; a detecção de bordas do objeto dinâmico e a detecção do seu contorno. Nas imagens (a, e, i) da Figura 5.6 é possível notar, no mapa denso de Farneback, formado por um conjunto de pontos, a geração de algumas retas vermelhas. Quanto maior é a reta, mais acentuado é o movimento detectado naquela região. A imagem (e) apresentou o movimento de câmera mais acentuado nos experimentos apresentados.



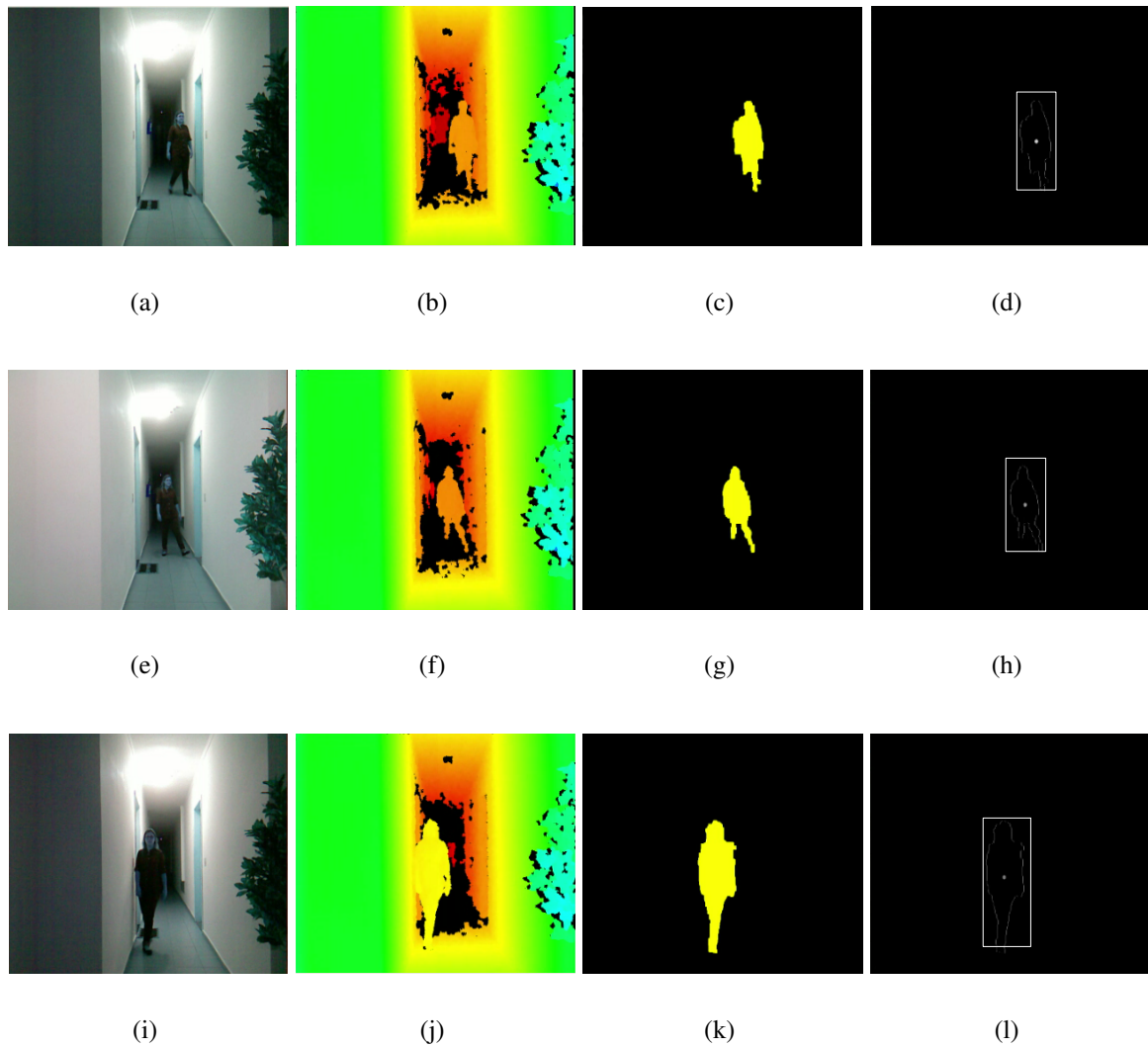
**Figura 5.6:** (a, c, e, g, i) - imagens com a detecção de objetos dinâmicos (pessoa) com a técnica de Fluxo Óptico de Farneback; (b, d, f, h, j) - Mapas de profundidade para obtenção da distância do objeto dinâmico; (k) - Detecção do contorno do objeto dinâmico; (l) Contorno produzido para calcular a largura e altura do objeto dinâmico detectado.

### 5.1.3.3 Subtração de *Background*

Para a análise de Objetos Dinâmicos por Subtração de *Background* é necessário que o DV produza poucos movimentos no sensor de aquisição. Mesmo com a presença de pequenos movimentos no sensor de aquisição, a Subtração de *Background* ainda pode detectar mudanças repentinas e prejudicar na segmentação do objeto dinâmico. Sendo assim, exige-se que seja aplicada uma técnica que diminua essa sensibilidade. Devido a essa situação, o SAS-DV utiliza dados de um acelerômetro para analisar os movimentos e emitir um bipe para o DV realizar poucos movimentos e então a Subtração de *Background* passa a ser executada. No SAS-DV, a Subtração de *Background* é executada com poucos movimentos de duas formas: a primeira com a técnica de estabilização de vídeo e a segunda com modificações (aumento) do parâmetro *Threshold* detalhado na Seção 3.2.3.3.

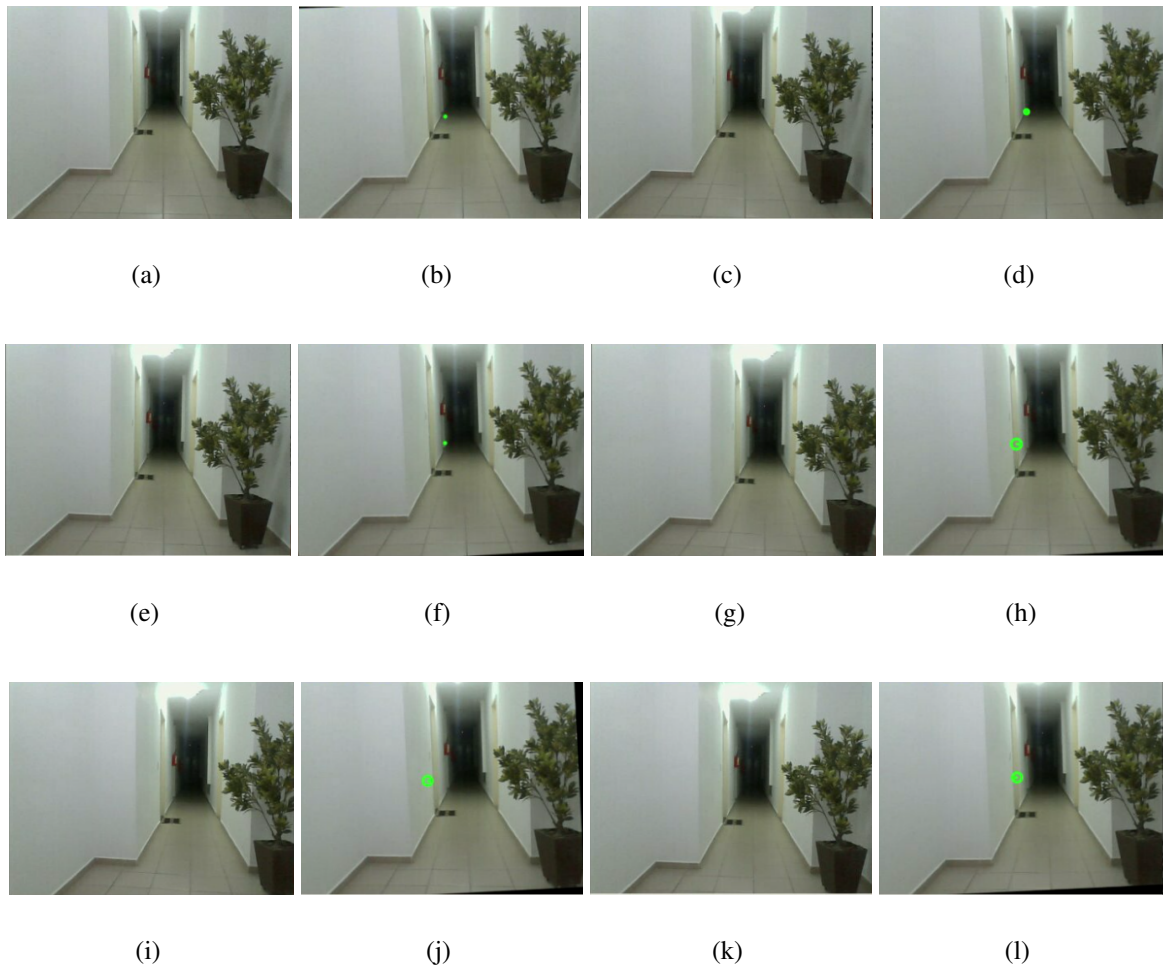
Na Figura 5.7, são apresentadas três faixas ((a, b, c, d), (e, f, g, h) e (i, j, k, l)) obtidas em uma pequena sequência de tempo. Cada faixa é composta por quatro imagens que apresentam, respectivamente: a imagem da câmera RGB; a imagem de profundidade; o objeto dinâmico segmentado; o contorno e o centro de massa do objeto dinâmico segmentado. A partir dessas características, é possível calcular a largura e a altura do objeto dinâmico. Todas as características obtidas nesse processo são enviadas para o Módulo de Fusão.

Com a obtenção da posição dos Objetos Dinâmicos, num curto período, é possível reconstruir o trajeto executado por eles. Esses trajetos são executados com maiores frequências por pessoas e eles são importantes para fornecer as regiões com maiores probabilidades de tráfego sem colisão. Mesmo que a técnica de Subtração de *Background* seja indicada para uma câmera estática, ela também pode ser usada após a estabilização de vídeo e com a definição de um limiar alto, para detectar apenas movimentos mais acentuados. Os experimentos dessa aplicação podem ser vistos na Figura 5.9.



**Figura 5.7:** (a, e, i) - imagens do sensor RGB; (b, f, j) - imagens do sensor IV; (c, g, k) - segmentação do objeto dinâmico pela técnica de Subtração de *Background*; (d, h, l) - geração do contorno e do centro de massa dos objetos dinâmicos segmentados.

**Experimentos com o Submódulo de Controle e Estabilização de Vídeo** Como já mencionado, a estabilização de vídeo é necessária para analisar os movimentos de um Objeto Dinâmico usando a Subtração de *Background*. Sendo assim, o acelerômetro, presente no dispositivo Kinect, é usado para detectar a intensidade dos movimentos e acionar esse processo. Inicialmente, os experimentos foram realizados para estabilizar as imagens RGB, como pode ser visto na Figura 5.8. Para melhor compreensão do processo de estabilização, um exemplo é apresentado num vídeo, disponível no seguinte link: <https://www.youtube.com/watch?v=8wNyRZ3-Y-4>. Um outro exemplo do mesmo processo de estabilização, porém aplicado nos mapas de profundidade, pode ser visto no material suplementar do artigo publicado por (CORDEIRO; PEDRINO, 2019a).



**Figura 5.8:** (a, c, e, g, i, k) - imagens com presença de movimento do vídeo; (b, d, f, h, j, l) - imagens estabilizadas após detecção de movimentos no sensor de aquisição (SA).

Na Figura 5.8 são apresentadas as imagens adquiridas pelo sensor RGB de modo sequencial (em pares) e com presença de movimento. Os itens (a), (c), (e), (g), (i), (k), dessa figura, apresentam as imagens com presença de movimento do vídeo e os itens (b), (d), (f), (h), (j), (l) apresentam as respectivas imagens estabilizadas dos itens anteriores. É possível notar um processo de estabilização com transformações geométricas (translação, rotação e escala) entre todos os pares de imagens. Observou-se, em algumas imagens, que as transformações geométricas são mais acentuadas quando ocorre um movimento mais intenso.

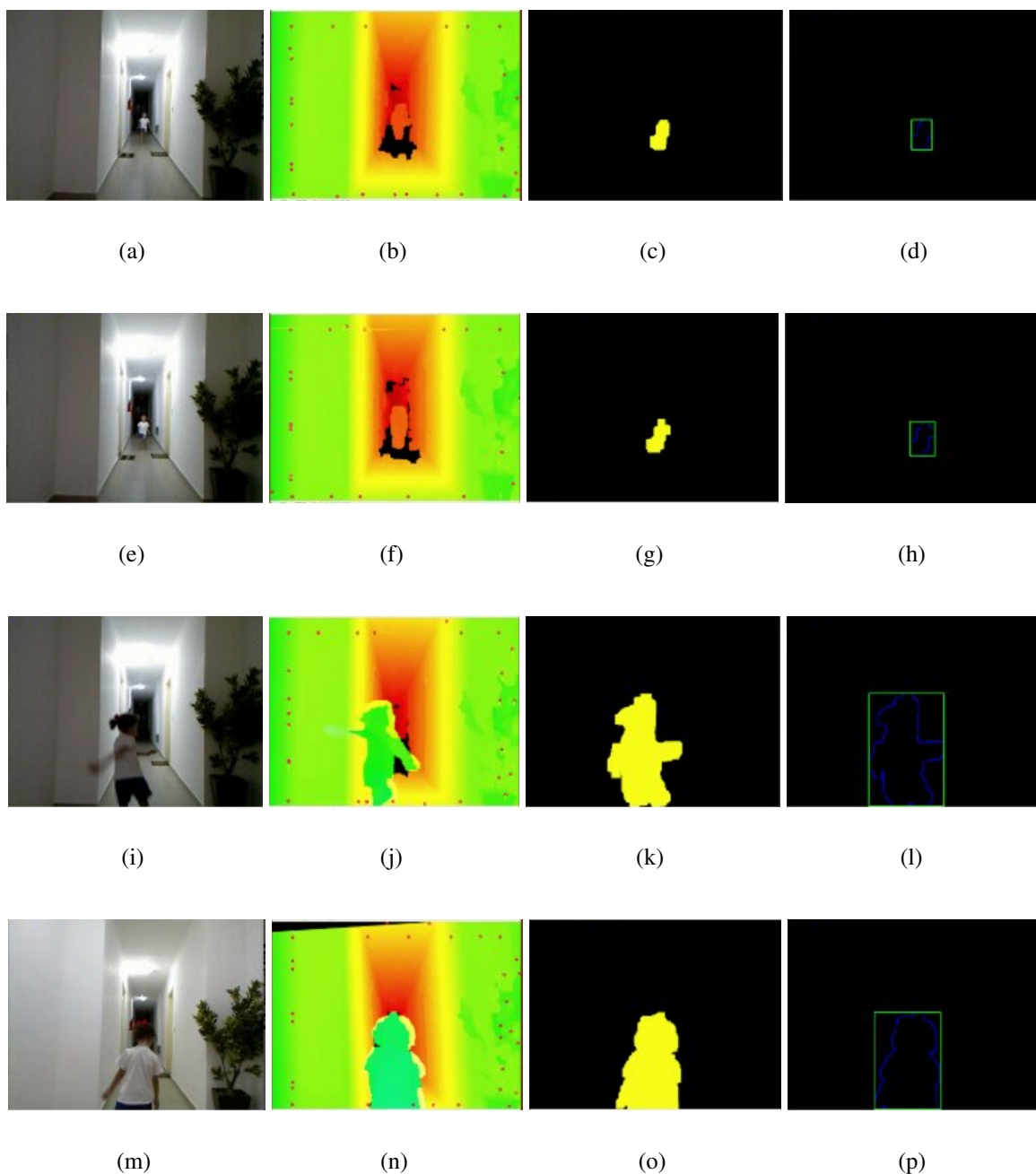
Nesta tese, foram experimentados alguns filtros de iluminação (Radiométrico e espaço de cor YcbCr) para melhorar a qualidade da subtração de *Background*. Porém, como apresentado na Seção 5.1.2, o MP não sofre interferências devido à iluminação. Sendo assim, adotou-se a imagem resultante do Submódulo de Mapa de Profundidade para a realização da segmentação de Objetos Dinâmicos por Subtração de *Background*. As imagens resultantes apresentaram melhor qualidade e ainda foi eliminado um conjunto de métodos de filtragem que impactava

diretamente no desempenho do sistema.

Ao analisar que os dados mais importantes eram fornecidos pelos MP, decidiu-se realizar experimentos para estabilizar os mapas de profundidade. Como já explicado na Seção 4.2.3, o processo de estabilização necessita de pontos de referência para analisar as transformações ocorridas entre uma sequência de *frames*. Entretanto, existe um problema, pois a detecção automática de pontos de referências em mapas de profundidade é mais complexa, o que prejudica o processo de estabilização. A solução para o caso foi obtida com a adição de pontos de referências auxiliares. Com esses pontos é possível prover ao filtro de Kalman uma maior quantidade de posições para verificar as transformações ocorridas entre a sequência de *frames*.

A Figura 5.9 apresenta um conjunto de experimentos realizados com os mapas de profundidade. Nesses experimentos já são apresentados também os resultados da Subtração de *Background*, após o processo de estabilização. Para um desses experimentos, os pontos detectados automaticamente pelo detector de pontos (HARRIS; STEPHENS, 1988), somado com os pontos de referências auxiliares, foram implementados para atuarem apenas nas regiões mais periféricas da imagem (Ver itens (b), (f), (j), (n) da Figura 5.9). Essa escolha se deve à ocorrência de sobreposição dos Objetos Dinâmicos sobre os pontos de referências. A sobreposição prejudica a quantidade e a qualidade das posições enviadas ao filtro de Kalman.

Os itens (a, e, i, m) da Figura 5.9 apresentam as imagens fornecidas pelo Sensor RGB. Os itens (b, f, j, n) são os Mapas de Profundidade dos respectivos itens anteriores e neles são ilustrados os pontos de referências implementados apenas nas regiões periféricas da imagem. Os MPs passaram por processos de estabilização para não prejudicar a qualidade da segmentação dos Objetos Dinâmicos detectada pelas técnicas de Subtração de *Background*. Entre as imagens que apresentam os MPs com estabilização, é possível notar uma transformação mais acentuada no item (n). Isso ocorre devido a um movimento intenso e brusco. Os itens (c, g, k, o), da mesma figura, apresentam os resultados do processo de segmentação por Subtração de *Background* e os itens (d, h, l, p) ilustram a detecção de contorno para, conseqüentemente, calcular a largura e altura do objeto.



**Figura 5.9:** (a, e, i, m) - imagens fornecidas pelo Sensor RGB; (b, f, j, n) - mapas de profundidade estabilizados usando pontos de referências nas regiões periféricas da imagem; (c, g, k, o) - segmentação de objetos dinâmicos usando Subtração de *Background* com a presença de movimentos no SA; (d, h, l, p) - detecção de contorno.

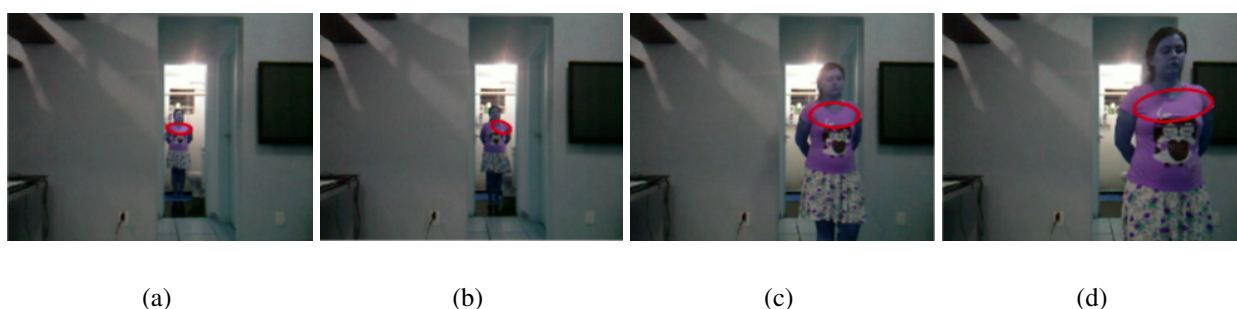
#### 5.1.3.4 CamShift

O CamShift foi a quarta técnica implementada no SAS-DV para a análise de Objetos Dinâmicos. Essa técnica consegue detectar um Objeto Dinâmico independentemente dos movimentos na câmera. É uma técnica que permite um alto desempenho na detecção dos Objetos



Dinâmicos, melhorando a reprodução dos trajetos. Observou-se que é uma técnica muito sensível à iluminação, ou seja, quando a luz apaga ou ocorre uma mudança repentina na iluminação, o sistema de detecção pode perder seu objeto de rastreamento.

Na Figura 5.10 são apresentadas quatro imagens (a, b, c, d). Cada imagem é produzida pelo Sensor RGB e o seu contorno (círculo oval e avermelhado) permite calcular o centro de massa do objeto detectado para o rastreamento.



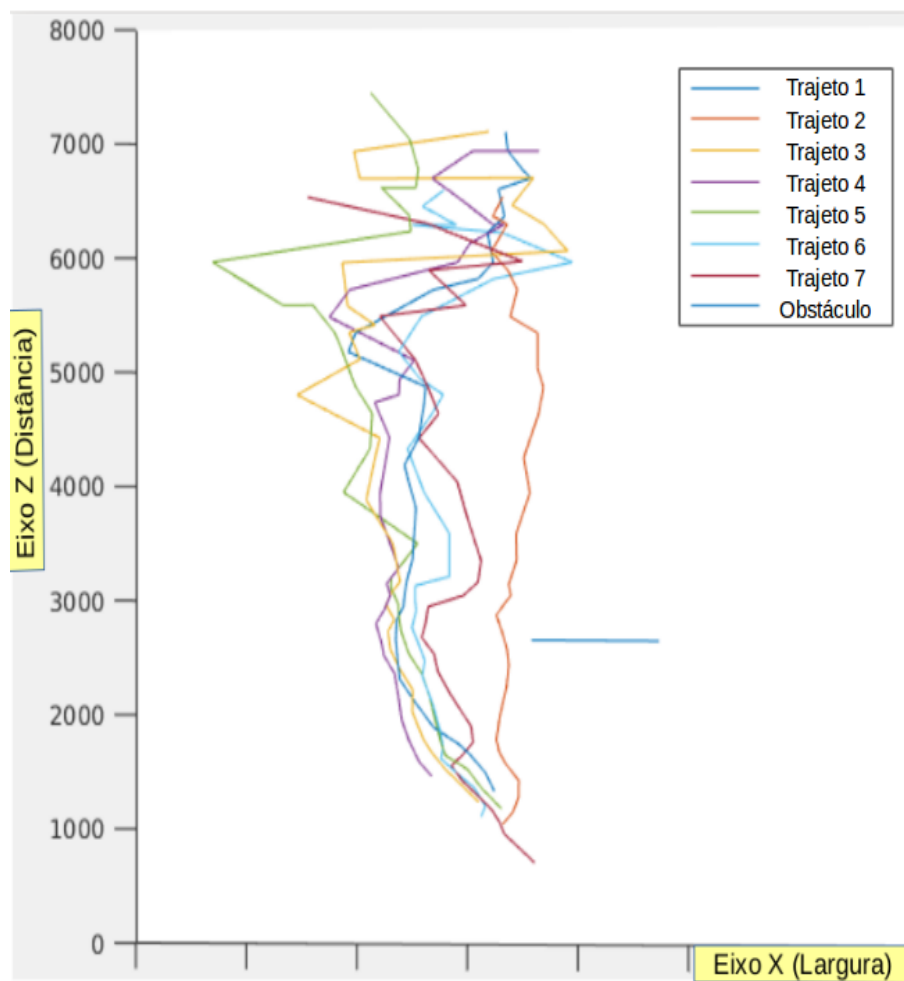
**Figura 5.10:** (a, b, c, d) - detecção de objetos dinâmicos em diferentes instantes usando a técnica CamShift.

#### 5.1.4 Experimentos - Submódulo de Posição 3D

Os experimentos com o SP3D serviram para obter todas as posições  $P(x, y, Z)$  dos objetos detectados no ambiente e converter para posições  $(P(X, Y, Z))$  em milímetros. Essa conversão tem a finalidade de eliminar distorções produzidas no processo de aquisição de dados e que são armazenadas com diferentes sistemas de referências.

No SAS-DV, considerou-se desnecessário calcular a altura ( $Y$ ) nos seguintes casos: obstáculos que não possibilitam o DV trafegar apesar de sua altura e obtenção de distâncias com uma altura previamente planejada. Sendo assim, a maioria das características foi convertida de uma posição  $P(x, y, Z)$  em *pixels*, para  $P(X, Z)$  em milímetros.

Os experimentos iniciais do SP3D foram realizados para a reconstrução de trajetos feitos por Objetos Dinâmicos. Dessa forma, concretizou-se o processo de conversão de  $(x, y, Z)$  para  $(X, Y, Z)$  de dez posições dos Objetos Dinâmicos detectados. Na Figura 5.11 é possível ver o trajeto executado por sete pessoas que desviam de uma cadeira (item (a) da Figura 5.17 e item (p) da Figura 5.18).



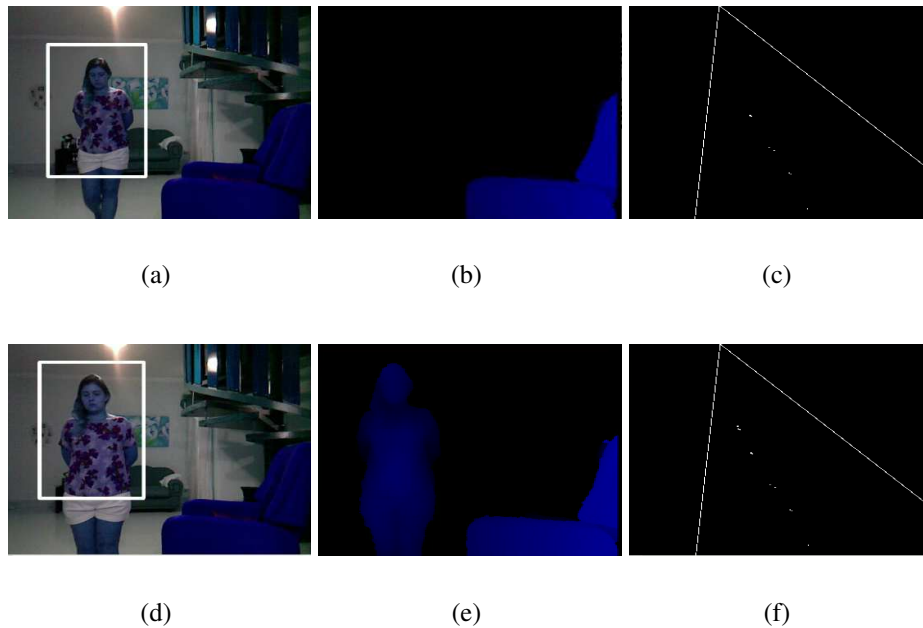
**Figura 5.11:** Trajetos realizados por uma pessoa (objeto dinâmico) em um Mapa(X , Z). No eixo X pode ser visto a largura do objeto e no eixo Z a sua respectiva distância.

#### 5.1.4.1 Experimentos do SP3D na A1-RP

Na Figura 5.12, as imagens (a - f) apresentam duas aquisições de dados para a detecção e rastreamento de Objetos Dinâmicos usando a A1-RP. Nessa arquitetura, a detecção foi realizada usando a técnica de *Haar-cascade*, baseada nos classificadores em cascata (VIOLA; JONES, 2001). Essas aquisições ocorreram com a câmera sofrendo poucos movimentos.

Cada sequência é composta pelas seguintes imagens: detecção de Objeto Dinâmico; Mapa de Profundidade que apresenta os objetos a menos de 2 metros de distância e a posição do objeto no Mapa(X , Z), no mesmo instante da detecção.





**Figura 5.12:** (a, d) - imagens com a detecção de objetos dinâmicos (pessoa) com a técnica de Reconhecimento de Padrões; (b, e) - mapas de profundidade; (c, f) - geração do Mapa (X,Z) usando a arquitetura A1-RP.

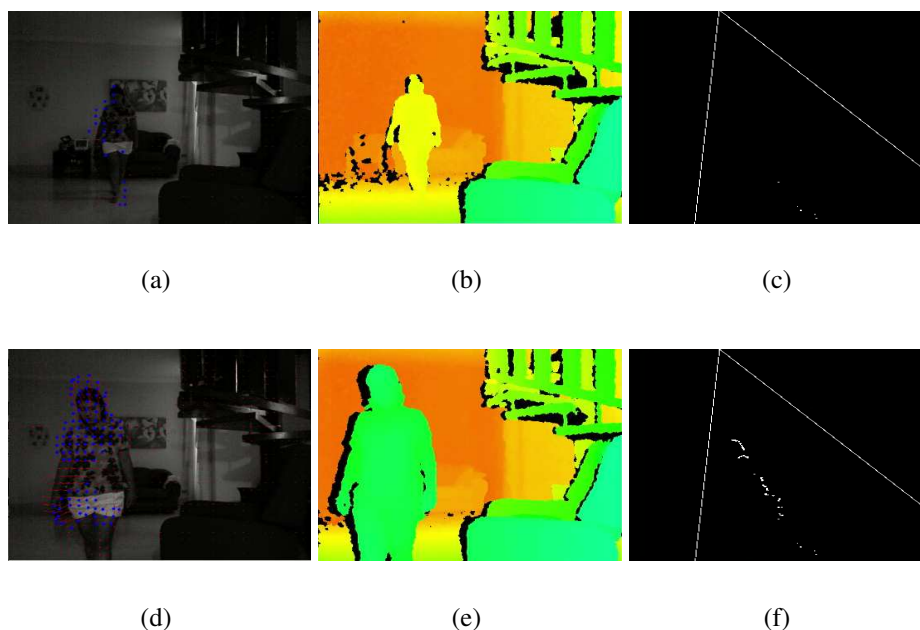
#### 5.1.4.2 Experimentos do SP3D na A2-FOF

Na Figura 5.13, as imagens (a - f) apresentam duas aquisições de dados para a detecção e rastreamento de Objetos Dinâmicos usando a A2-FOF. Essas aquisições ocorreram com a câmera sofrendo poucos movimentos (ver vídeos suplementares no repositório do artigo (CORDEIRO; PEDRINO, 2019b)). Cada sequência é composta pelas seguintes imagens: detecção de Objeto Dinâmico; Mapa de Profundidade e a posição do objeto no Mapa(X , Z), no mesmo instante da detecção.

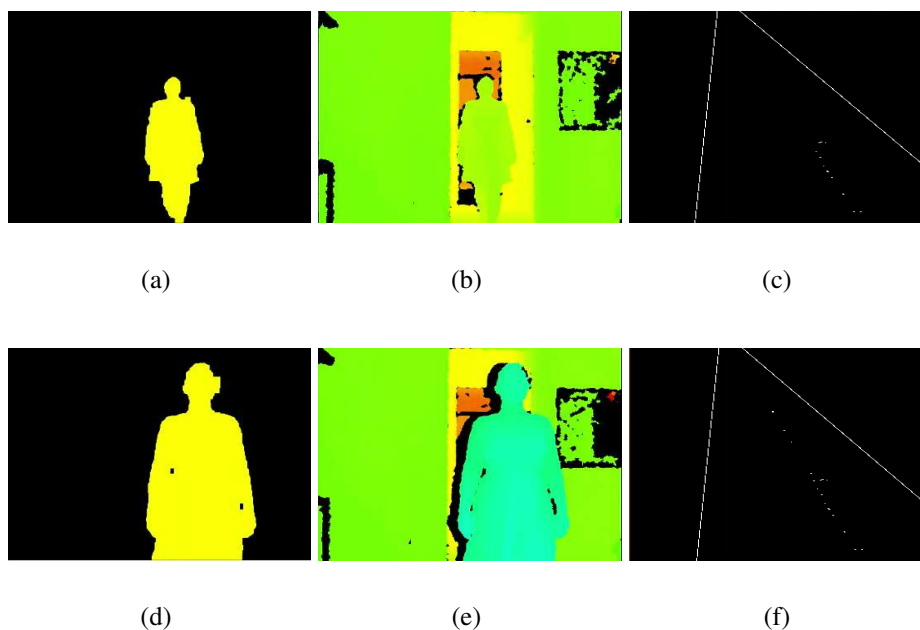
#### 5.1.4.3 Experimentos do SP3D na A3-SB

Na Figura 5.14, as imagens (a - f) apresentam duas aquisições de dados para a detecção e rastreamento de Objetos Dinâmicos usando a A3-SB.

Essas aquisições também foram realizadas com a câmera sofrendo poucos movimentos. Cada sequência é composta pelas seguintes imagens: detecção de Objeto Dinâmico; Mapa de Profundidade e a posição do objeto no Mapa(X , Z), no mesmo instante da detecção.



**Figura 5.13:** (a, d) - imagens com a detecção de objetos dinâmicos com a técnica de Fluxo Óptico de Farneback; (b, e) - mapas de profundidade; (c, f) - geração do Mapa(X , Z) usando a arquitetura A2-FOF.

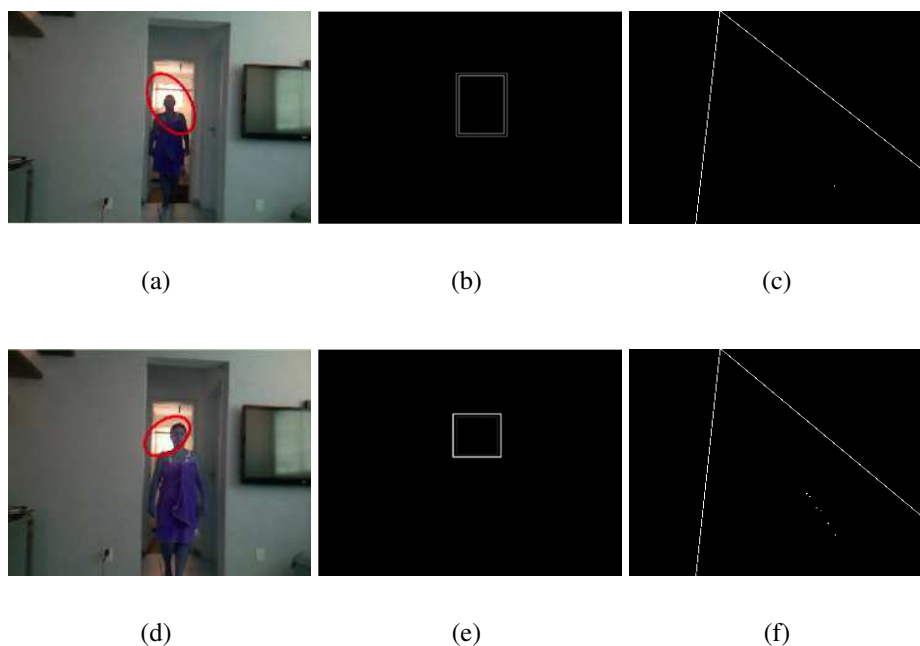


**Figura 5.14:** (a, d) - imagens com a detecção de objetos dinâmicos com a técnica de Subtração de *Background*; (b, e) - mapas de profundidade; (c, f) - geração do Mapa(X , Z) usando a arquitetura A3-SB.

#### 5.1.4.4 Experimentos do SP3D na A4-CAM

Na Figura 5.15, as imagens (a-f) apresentam duas aquisições de dados para a detecção e rastreamento de Objetos Dinâmicos usando a A4-CAM. Esta arquitetura realiza o rastreamento do

objeto baseado em um único matiz. Essas aquisições seguem o mesmo formato dos experimentos anteriores, com a câmera sofrendo poucos movimentos. Cada sequência é composta pelas seguintes imagens: detecção do matiz pré-definido; contorno do objeto detectado e a posição do objeto no Mapa(X , Z) no mesmo instante da detecção.



**Figura 5.15:** (a, d) - imagens com o rastreamento de objetos dinâmicos usando a técnica CamShift; (b, e) - detecção dos contornos; (c, f) - geração do Mapa(X , Z) usando a arquitetura A4-CAM.

Para todos os experimentos executados com as técnicas de detecção e rastreamento de Objetos Dinâmicos foram realizados, em paralelo, o monitoramento do gasto computacional (CPU e Memória RAM) para obter o custo de arquitetura implementada. As comparações entre os gastos são apresentadas no Capítulo 6.

## 5.2 Experimentos realizados com o Módulo de Fusão

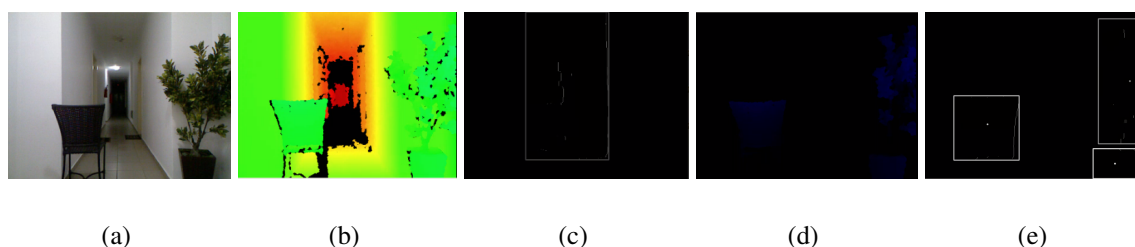
Nesta seção são apresentados alguns experimentos realizados com o Módulo de Fusão para validar os seguintes submódulos: Submódulo de Percepção (Seção 3.3.0.1); Submódulo de Compreensão (Seção 3.3.0.2) e Submódulo de Projeção (Seção 3.3.0.3). Para a primeira versão do SAS-DV (CORDEIRO et al., 2016), foram aplicadas diferentes técnicas de classificação para calcular a probabilidade nas seguintes situações: riscos de colisão com Objetos Dinâmicos; riscos de tráfego em regiões que apresentavam ambientes perigosos (escada, janela baixa, buracos) e possibilidade de locomoção em passagens livres. Um conjunto de características foi extraído para cada uma dessas situações e classificado com as técnicas de Máquina de Vetor de

Suporte (SVM), Rede Neural Perceptron e Redes Bayesianas.

A finalidade deste estudo foi comparar qual das três técnicas poderia ser a mais adequada, considerando a efetividade (quantidade de acertos) para uma determinada situação. Dessa forma, verificou-se que, para situações em que poderia ser adicionado uma crença *a priori*, as Redes Bayesianas se apresentaram como a melhor opção, pela facilidade em produzir um modelo confiável. Nos casos em que houve dificuldade em aplicar conhecimento *a priori* e foi necessário extrair um conjunto de características para compreender melhor o processo de classificação, foi usado o SVM e o Perceptron. Para esses casos em que não houve a adição de crenças *a priori*, usou-se a técnica supervisionada.

O desenvolvimento da primeira versão e o estudo exaustivo dessas técnicas ajudaram a compreender diferentes formas de classificar situações. Desse modo, permitiu-se criar outros formatos para a construção do método de projeção de riscos, baseado em um mapa composto por inferências. Como já mencionado, esse mapa fornece a direção que possui o menor risco de colisão para o DV. Após algumas análises no Submódulo de Compreensão, verificou-se que seria possível adicionar conhecimento *a priori*, dadas as circunstâncias que definiriam as ações, intenções, relevância e capacidade dos elementos usados para a tomada de decisão. Nesse caso, definiu-se, para a segunda versão do SAS-DV, o uso das Redes Bayesianas (CORDEIRO; PEDRINO, 2019a) para executar inferências e produzir um mapa de riscos de colisão.

As Figuras (5.16 e 5.17) apresentam a extração de características realizadas pelo Módulo de Visão e pelo Submódulo de Percepção, em duas situações. Esses experimentos foram realizados com o DV se locomovendo em um ambiente *indoor* sem nenhum Objeto Dinâmico. Como pode ser observado, nessas figuras não são apresentados Obstáculos Dinâmicos na direção do DV. Para os dois experimentos, são apresentadas também, a seguir, as respectivas tabelas (5.1 e 5.2) das figuras (5.16 e 5.17), contendo os resultados obtidos.



**Figura 5.16: Situação 1: Processo de extração de características para experimentos com o Módulo de Fusão.** (a) - imagem disponibilizada pelo Sensor RGB; (b) - mapa de profundidade; (c) - detecção do contorno da passagem livre; (d) - detecção de objetos estáticos; (e) - contorno dos objetos estáticos.

Na Figura 5.16, o experimento foi realizado com um Obstáculo Estático (cadeira) adicio-

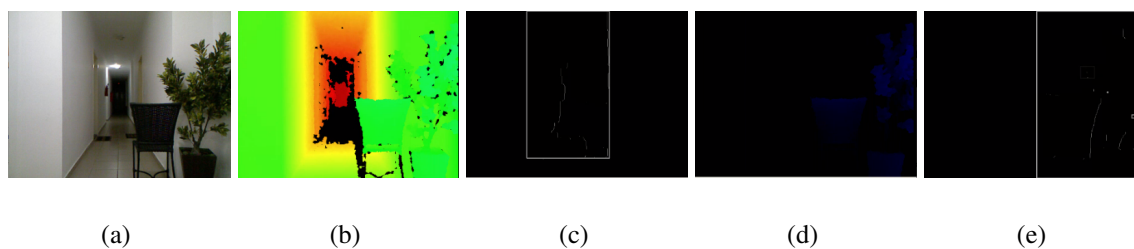
nado, de propósito, na direção do DV. Esse experimento serviu para validar, preliminarmente, os submódulos do Módulo de Fusão. A Tabela 5.1 apresenta os valores calculados pelo SPE. Após essa etapa, o modelo foi produzido (*modelo<sub>Bayes</sub>*) pelo Submódulo de Compreensão e as inferências foram aplicadas pelo Submódulo de Projeção.

**Tabela 5.1: Dados gerados pelo Submódulo de Percepção da Situação 1 (Figura 5.16) para o modelo 1 - Objetos Dinâmicos (OD), Trajetos realizados por ODs (TRJ), Objetos Estáticos (OE), Passagem Livre (PL), PRf detectados com menos de 1.4m, 1.7m e 2.0m respectivamente (PRf1.4m, PRf1.7m, PRf2m) e Risco de Colisão (RC)**

Ângulos	Distância	ODs	TRJ	OE	PL	PRf1.4m	PRf1.7m	PRf2.0m	RC
64.6°	1894	0	0	1	0	0	0	1	65%
67.8°	1877	0	0	1	0	0	0	1	65%
70.9°	1498	0	0	1	0	0	1	1	80%
74.1°	2617	0	0	1	0	0	0	0	65%
77.3°	1487	0	0	1	0	0	1	1	70%
80.4°	5716	0	0	0	1	0	0	0	10%
83.6°	7813	0	0	0	1	0	0	0	10%
86.8°	7852	0	0	0	1	0	0	0	10%
90°	7380	0	0	0	1	0	0	0	10%
93.1°	6772	0	0	0	1	0	0	0	10%
96.3°	4372	0	0	0	1	0	0	0	10%
99.5°	2837	0	0	0	0	0	0	0	45%
102.6°	1713	0	0	1	0	0	0	1	65%
105.8°	1803	0	0	1	0	0	0	1	65%
109°	1849	0	0	1	0	0	0	1	65%
112.1°	1416	0	0	1	0	0	1	1	80%
115.3°	1404	0	0	1	0	0	1	1	80%

Nessa tabela, a coluna ODs apresenta a detecção de Objetos Dinâmicos. Os trajetos realizados por ODs podem ser detectados e apresentados pela coluna TRJ. A coluna de Objetos Estáticos contém valores indicando a direção dos OEs detectados. Na coluna PL, são apresentadas as direções que possuem passagem livre para locomoção. Nota-se que as colunas PR1.4m, PR1.7m e PR2m também fornecem dados para a tomada de decisão. Essas colunas representam os PRf definidos na Figura 3.9. Nesse contexto, é preciso observar que esses pontos possuem uma forte relação com a presença de Objetos Estáticos apresentados na coluna OE. Eles são considerados intervalos de distância e isso permite definir diferentes riscos de colisão. Os riscos de colisão apresentados na coluna RC são maiores nos pontos em que foram detectados os obstáculos estáticos e menores nos pontos em que foi calculada a passagem livre.

Na situação 2, apresentada pela Figura 5.17, é realizado novamente um experimento sem a presença de Objetos Dinâmicos. Todavia, para esse experimento adicionaram-se as posições nas quais foram detectadas a presença de um trajeto realizado por ODs (Ver Figura 5.11).



**Figura 5.17: Situação 2: Processo de extração de características para experimentos com o Módulo de Fusão. (a) - imagem disponibilizada pelo Sensor RGB; (b) - mapa de profundidade; (c) - detecção do contorno da passagem livre; (d) - detecção de objetos estáticos; (e) - contorno dos objetos estáticos.**

A Tabela 5.2 apresenta dados produzidos pelo sub-módulo de Percepção quando analisada a segunda situação (ver Figura 5.17).

**Tabela 5.2: Dados gerados pelo sub-módulo de Percepção da Situação 2 (ver Figura 5.17) para o modelo 1 - Objetos Dinâmicos (OD), Trajetos realizados por ODs (TRJ), Objetos Estáticos (OE), Passagem Livre (PL), PRf detectados com menos de 1.4m, 1.7m e 2.0m respectivamente (PRf1.4m, PRf1.7m, PRf2.0m) e Risco de Colisão (RC)**

Ângulos	Distância	OD	TRJ	OE	PL	PRf1.4m	PRf1.7m	PRf2.0m	RC
64.6°	1894	0	0	1	0	0	0	1	65%
67.8°	1911	0	0	1	0	0	0	1	65%
70.9°	2159	0	0	1	0	0	1	0	80%
74.1°	2650	0	0	1	0	0	0	0	65%
77.3°	3604	0	0	0	1	0	1	0	60%
80.4°	5884	0	0	0	1	0	0	0	10%
83.6°	8452	0	1	0	1	0	0	0	1%
86.8°	7825	0	1	0	1	0	0	0	1%
90°	7352	0	0	0	1	0	0	0	10%
93.1°	6772	0	0	0	1	0	0	0	10%
96.3°	1498	0	0	0	0	0	1	1	80%
99.5°	2820	0	0	1	0	0	0	0	65%
102.6°	1498	0	0	1	0	0	1	1	80%
105.8°	1803	0	0	1	0	0	0	1	65%
109°	1860	0	0	1	0	0	0	1	65%
112.1°	1416	0	0	1	0	0	1	1	80%
115.3°	1411	0	0	1	0	0	1	1	80%

Em vista disso, é possível observar na Tabela 5.2, um menor risco de colisão nas direções (83.6° e 86.8°), locais onde pessoas já haviam trafegado.

A Tabela 5.3 apresenta a aplicação do segundo modelo utilizando as mesmas informações obtidas pelo Submódulo de Percepção, aplicadas na Tabela 5.1. Os detalhes desses modelos podem ser encontrados no Capítulo 3.

**Tabela 5.3: Dados gerados pelo Submódulo de Percepção da Situação 1 (Figura 5.16) para o modelo 2 - Objetos Estáticos (OE), Objetos Dinâmicos (OD), Passagem Livre (PL), Trajetos realizados por ODs (TRJ) e Risco de Colisão (RC)**

Direção	OE <sub>mm</sub>	OD <sub>mm</sub>	PL <sub>mm</sub>	TRJ <sub>mm</sub>	RC
64.6°	1895	?	?	?	Alto
67.8°	1890	?	?	?	Alto
70.9°	1500	?	?	?	Muito Alto
74.1°	?	?	?	?	Moderado
77.3°	1500	?	?	?	Muito Alto
80.4°	?	?	5720	?	Baixo
83.6°	?	?	7815	?	Baixo
86.8°	?	?	7850	?	Baixo
90°	?	?	7380	?	Baixo
93.1°	?	?	6775	?	Baixo
96.3°	?	?	4375	?	Baixo
99.5°	?	?	?	?	Moderado
102.6°	1715	?	?	?	Alto
105.8°	1805	?	?	?	Alto
109°	1850	?	?	?	Alto
112.1°	1420	?	?	?	Muito Alto
115.3°	1465	?	?	?	Muito Alto

A Tabela 5.4 apresenta a aplicação do segundo modelo e utiliza as mesmas informações obtidas pelo Submódulo de Percepção aplicadas na Tabela 5.2.

**Tabela 5.4: Dados gerados pelo Submódulo de Percepção da Situação 2 (Figura 5.17) para o modelo 2 - Objetos Estáticos (OE), Objetos Dinâmicos (OD), Passagem Livre (PL), Trajetos realizados por ODs (TRJ) e Risco de Colisão (RC)**

Direção	OE <sub>mm</sub>	OD <sub>mm</sub>	PF <sub>mm</sub>	TRJ <sub>mm</sub>	RC
64.6°	1900	?	?	?	Alto
67.8°	1915	?	?	?	Alto
70.9°	?	?	?	?	Moderado
74.1°	?	?	?	?	Moderado
77.3°	?	?	3605	?	Baixo
80.4°	?	?	5885	?	Baixo
83.6°	?	?	8455	7200	Baixo
86.8°	?	?	7825	7200	Baixo
90°	?	?	7355	7200	Baixo
93.1°	?	?	6775	?	Baixo
96.3°	1500	?	?	?	Muito Alto
99.5°	?	?	?	?	Moderado
102.6°	1500	?	?	?	Muito Alto
105.8°	1805	?	?	?	Alto
109°	1860	?	?	?	Alto
112.1°	1420	?	?	?	Muito Alto
115.3°	1415	?	?	?	Muito Alto

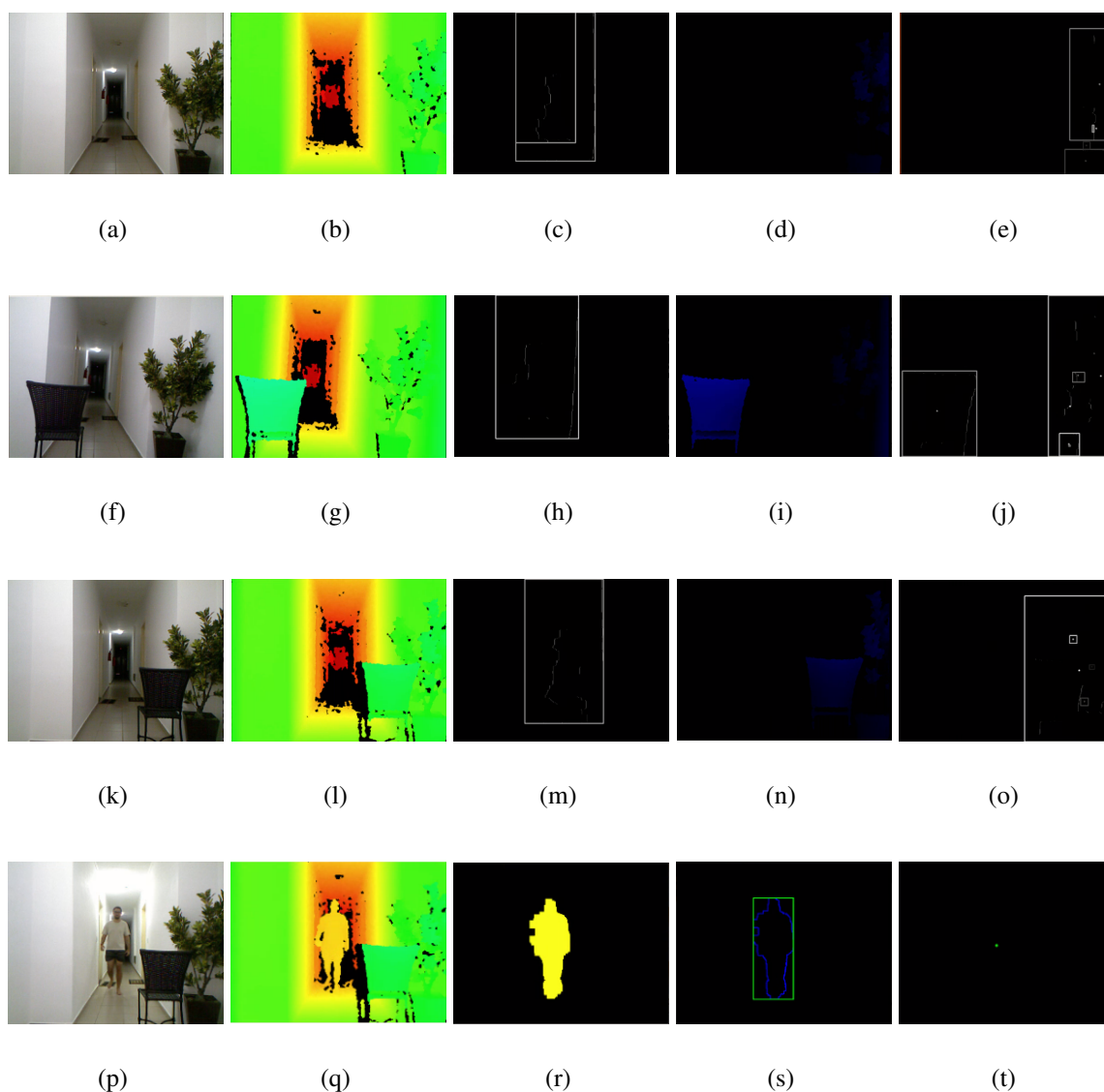
### 5.2.1 Análise comparativa com outros sistemas de navegação

Foram realizadas quatro etapas de experimentos nas quais se aplicou, em cada etapa, um Sistema de Navegação (SN) diferente. O Módulo de Visão da arquitetura SAS-DV foi responsável por fornecer as características básicas para os quatro sistemas de navegação. O Módulo de Visão foi implementado com a linguagem C++, e teve auxílio de técnicas de Visão Computacional e de processamento de imagens, por meio da biblioteca OpenCv (BRADSKI; KAEHLER, 2008). Todas as características foram obtidas em ambientes fechados presenciando Objetos Estáticos e Dinâmicos com diferentes posições e tamanhos. Para os ODs também foram modificadas as direções e a velocidade média no período de detecção.

A Figura 5.18 é composta por quatro situações diferentes e são representadas por quatro imagens (itens (a), (f), (k), (p)), em que elas são fornecidas pela câmera RGB. Para cada situação (itens (a), (f), (k), (p)) são disponibilizadas as seguintes características: os itens (b, g, l, q) apresentam os Mapas de Profundidade; os itens (c, h, m) apresentam os contornos resultantes do método de segmentação da passagem livre ( $PL(x_j, y_j, Z_j).cm$ ,  $PL(x_j, y_j, Z_j).p1$ ,  $PL(x_j, y_j, Z_j).p2$ ); os itens (d, i, n) apresentam os objetos detectados em primeiro plano que estão entre 0.6 à 2.0 metros de distância ( $OE_{[c]}(X, Y, Z).cm$ ,  $OE_{[c]}(X, Y, Z).p1$ ,  $OE_{[c]}(X, Y, Z).p2$ ); os itens (e, j, o) apresentam o contorno dos objetos detectados nos itens (d, i, n); o item (q) fornece segmentação do Objeto Dinâmico (Pessoa); o item (s) disponibiliza o contorno resultante do item (r) e o item (t) apresenta o centro de massa ( $OD_{[n]}(X, Y, Z)_{[m]}.cm$ ) resultante do item (s). É importante destacar que as situações três e quatro (respectivamente itens (k) e (p)) são divergentes apenas pela presença do Objeto Dinâmico. Portanto, sem a presença do Objeto Dinâmico, a situação possui as mesmas características. A partir dessas quatro situações, produziram-se os vetores de características e, em seguida, foram realizados os experimentos para comparar as diferentes abordagens usadas nos sistemas de navegação.

A primeira etapa de experimento foi realizada com o primeiro Sistema de Navegação ( $SN_1$ ) implementado. O  $SN_1$  indica a direção para deslocamento, baseado na posição 3D (X, Y, Z) dos obstáculos que estão a menos de dois metros de distância do DV. Projetadas as posições dos obstáculos no plano 3D, verifica-se a direção de passagem entre os obstáculos. Alguns trabalhos (ALADREN et al., 2014), (COSTA et al., 2012), (LAKDE; PRASAD, 2015), (SAPUTRA; WIDYAWANKA; SANTOSA, 2014) usam uma abordagem semelhante para definir a direção que o DV deve se locomover. O Sistema de Navegação desenvolvido por Aladren et al. (2014) executa bipes com diferentes intensidades que indicam para o DV a presença de obstáculos próximos. O SN apresentado por Costa et al. (2012) também se baseia em obstáculos que estão a menos de dois metros de distância. Nesse SN é fornecido para o DV a posição relativa dos





**Figura 5.18: Processo de extração de características - Quatro situações.** (a, f, k, p) - imagens disponibilizadas pelo Sensor RGB; (b, g, l, q) - mapas de profundidade; (c, h, m) - detecção do contorno da passagem livre; (d, i, n) - detecção dos objetos estáticos; (e, j, o) - detecção do contorno dos objetos estáticos; (r) - detecção do objeto dinâmico; (s) - detecção do contorno do objeto dinâmico; (t) - centro de massa do objeto dinâmico.

obstáculos e cinco direções para locomoção. A Figura 5.19 exibe a aplicação do  $SN_1$  a partir da extração de características das três situações abordadas na Figura 5.18. Cada situação fornece a posição 3D, a largura dos obstáculos e a direção da passagem livre.

Posteriormente, realizou-se a segunda etapa de experimentos com o sistema de navegação ( $SN_2$ ) baseado na implementação de Nadia *et al.* (2015). O  $SN_2$  tem como objetivo fornecer uma passagem livre após obter as distâncias dos cantos detectados na imagem. Com a análise das distâncias, os cantos detectados são exibidos com diferentes cores, e cada cor representa um intervalo de distâncias (Nadia *et al.* (2015)). Os pontos verdes indicam uma região com perigo

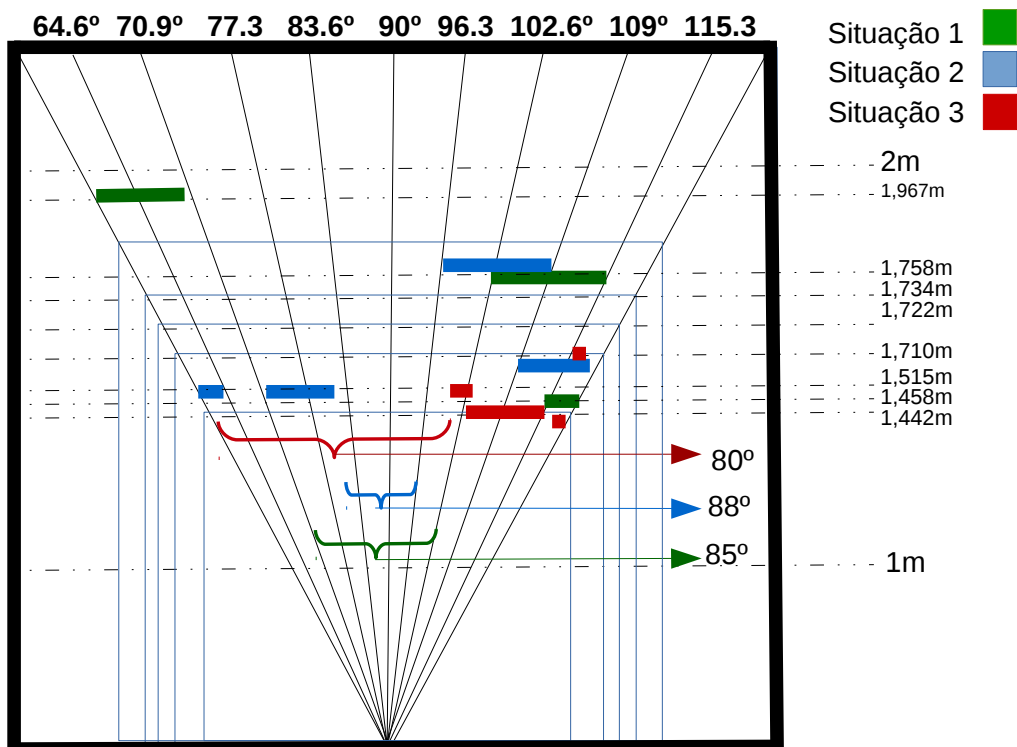


Figura 5.19: (a, b, c) - direções fornecidas pelo  $SN_1$  nas três primeiras situações apresentadas na Figura 5.18.

potencial de colisão na região indicada para tráfego. Os objetos mais distantes apresentam outras cores. Os detalhes da região (linhas azuis, ver Figura 5.20) e das distâncias (pontos coloridos) são melhores especificados no seu trabalho (Nadia *et al.* (2015)). Na mesma figura são apresentados os resultados do  $SN_2$  aplicados nas três situações. A quarta situação não foi gerada em razão desse sistema não analisar trajetos executados por Objetos Dinâmicos.

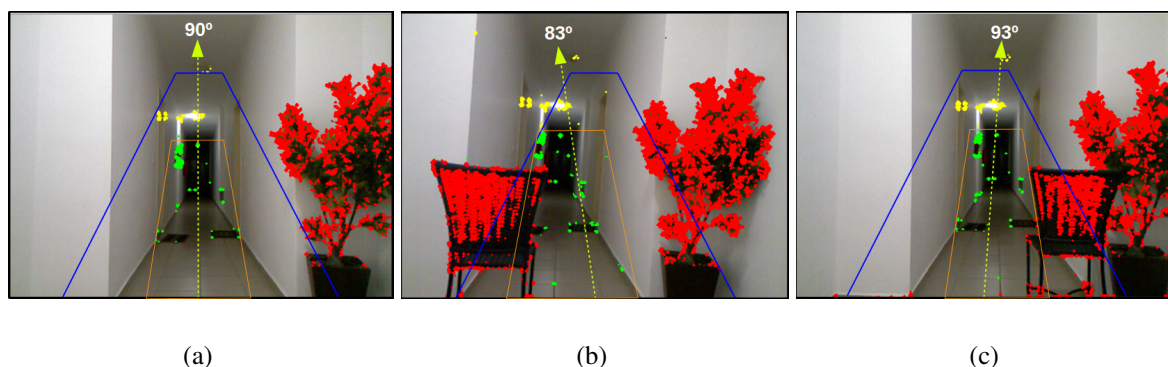


Figura 5.20: Direções fornecidas pelo  $SN_2$  nas três primeiras situações apresentadas na Figura 5.18.

Nadia *et al.* (2015) não fornecem uma direção específica para tráfego, apenas em qual região o DV deve seguir (esquerda, central ou direita). É possível observar que, muitas direções

dentro dessas regiões poderiam ser indicadas. Baseado na implementação do trabalho proposto por Nadia *et al.* (2015) e com a obtenção das distâncias de cada ponto de referência apresentado na Seção 3.3.0.1, um outro Sistema de Navegação ( $SN_3$ ) foi proposto e experimentado, usando apenas o conjunto de distâncias fornecidas pelos PRf (ver Figura 3.9). Essas distâncias são exibidas na Figura 5.21 (área cinza), e foram obtidas a partir dos mesmos ângulos e alturas aplicados no sistema de inferência do SAS-DV. A DI que o  $SN_3$  fornece para a DV respeita a seguinte premissa: gerar o ponto médio entre as duas direções sequenciais (uma ao lado da outra) que possuem as maiores distâncias.

Após analisar os resultados desta abordagem, verificou-se que esta proposta poderia aumentar a confiabilidade do  $SN_{SAS-DV}$ . Sendo assim, foi implementado no SAS-DV a indicação da região mais distante fornecida pelo  $SN_3$ . Essa indicação ocorre apenas quando o SAS-DV fornece mais de uma direção que possua o mesmo risco de colisão. Dessa forma, respeitando as crenças do SAS-DV, é realizada a união entre essas duas abordagens e a indicação do ponto mais distante fornecido pelo  $SN_3$  é enviada para o  $SN_{SAS-DV}$  indicar ao DV apenas uma direção. Com essa união, os resultados dos experimentos realizados pelo  $SN_3$  foram apresentados com os resultados do  $SN_{SAS-DV}$ . A quarta e última etapa de experimentos foi realizada com o Sistema de Navegação do SAS-DV ( $SN_{SAS-DV}$ ) para validar os seguintes submódulos: SSOD; Submódulo de Percepção; Submódulo de Compreensão e Submódulo de Projeção.

Para essa etapa, os experimentos foram iniciados com a análise de Objetos Dinâmicos (SSOD). O SSOD não foi implementado nos três sistemas anteriores ( $SN_1$ ,  $SN_2$  e  $SN_3$ ). Esse experimento teve o objetivo de armazenar trajetos executados por um Objeto Dinâmico próximo do tempo real. A análise desses objetos ocorre após detectar a presença de movimentos divergentes dos realizados pela câmera. Dessa maneira, um bipe é transmitido para o DV parar de se locomover e estabilizar os movimentos bruscos. Ao detectar pequenos movimentos na câmera, é executado o processo de estabilização de vídeo para que seja possível obter diferentes instantes (X,Y,Z) do objeto em movimento e possibilitar a reconstrução dos trajetos. Esses trajetos, executados por pessoas, são importantes para fornecer as regiões com maiores probabilidades de tráfego sem colisão.

Na Figura 5.7, são apresentados alguns exemplos do processo de análise de Objetos Dinâmicos. O mesmo processo foi executado na situação (4) da Figura 5.18. A segmentação por Subtração de *Background* foi a técnica aplicada para obter o posicionamento de todos os Objetos Dinâmicos presentes em cena. Com a obtenção de vários instantes do Objeto Dinâmico, foi possível reconstruir o trajeto produzido por ele. Após a realização de testes com os submódulos do Módulo de Visão, essa etapa se concentrou nos experimentos relacionados aos modelos

propostos para o processo de compreensão do SAS-DV. As aplicações que envolvem Redes Bayesianas têm como necessidade o conhecimento inicial de muitas probabilidades incondicionais (CONRADY; JOUFFE, 2015). Isso pode ser um problema em aplicações que dificultam essa compreensão. Porém, no SAS-DV, a modelagem foi planejada da mesma forma que uma pessoa com visão toma decisões para trafegar com mais segurança. Sendo assim, não foi difícil definir as probabilidades iniciais do primeiro modelo.

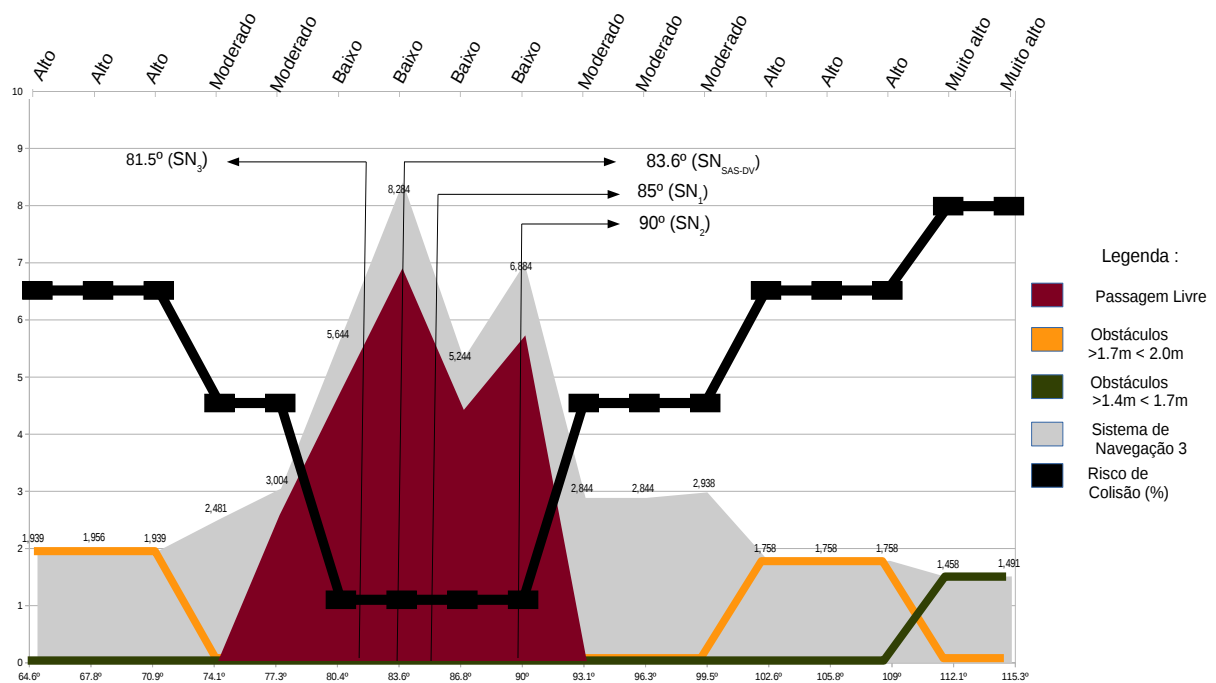
No Submódulo de Compreensão, atribuíram-se probabilidades *a priori* para um conjunto de crenças baseado nas seguintes premissas: uma direção que possui passagem livre produz um peso positivo para a locomoção, pois fornece ao DV condições de atingir um determinado destino; uma direção que possui um objeto dinâmico em tráfego produz maior risco de colisão; uma direção que possui registro de trajeto executado por Objeto Dinâmico produz menos riscos de colisão, pois entende-se que, se uma pessoa executou este trajeto, o DV pode se locomover com maior segurança; a direção que possui obstáculos mais próximos gera maiores riscos de colisão, sendo assim, as probabilidades de riscos de colisão tiveram maior peso. A ferramenta Weka auxiliou na geração desse modelo ( $modelo_{Dataset_1}$ ), baseado em crenças, e no seu treinamento efetuado pelas Redes Bayesianas. Dessa forma, foi possível executar uma inferência (Submódulo de Projeção) para cada ponto de referência definido na Figura 5.18. Com a geração dos mapas de riscos, a direção com menor risco de colisão é indicada.

O *dataset* produzido para os primeiros experimentos usando Redes Bayesianas é de baixa complexidade e composto apenas por valores 0 e 1 que, respectivamente, indicam a ausência e a presença de Objetos Estáticos, Objetos Dinâmicos, Passagem Livre, trajetos realizados por ODs e Risco de Colisão (CR). Mesmo sendo um *dataset* simples e que possui poucas possibilidades de variações, já apresentou benefícios comparados aos trabalhos que não aplicam algum processo de aprendizagem e que permitam realizar o processo de projeção, presente no modelo de *situation awareness*.

Como uma alternativa para melhorar a confiabilidade do sistema de projeção de riscos de colisão, um segundo *dataset* foi concebido com um conjunto de possibilidades muito maior, de forma mais complexa e que permite analisar o risco com mais precisão. O segundo *dataset* possui apenas quatro entradas e uma saída de classificação, mas todas as entradas são compostas por valores do tipo *numeric*. Esse *dataset* calcula o risco de colisão baseado nas posições 3D dos obstáculos. Dessa forma, permite identificar um risco com maior precisão comparado ao primeiro *dataset*. Um exemplo desse modelo pode ser visto no Submódulo de Compreensão do Capítulo 3. Esse segundo *dataset* foi treinado e experimentado por um conjunto de classificadores e os resultados são apresentados no Capítulo 6.

As Figuras 5.21, 5.22, 5.23 apresentam os resultados dos mapas de riscos pelo  $SN_{SAS-DV}$ , nas três situações da Figura 5.18. A linha preta presente nos mapas (a), (b) e (c) representa as probabilidades de risco de colisão (%/10) em 17 direções. A linha verde tracejada presente no mapa (c) apresenta o mesmo tipo de resultado para a situação (4). Devido aos outros sistemas não realizarem a análise de trajetos executados por Objetos Dinâmicos, apenas o  $SN_{SAS-DV}$  produziu esse valor (Linha Laranja da Figura 5.23).

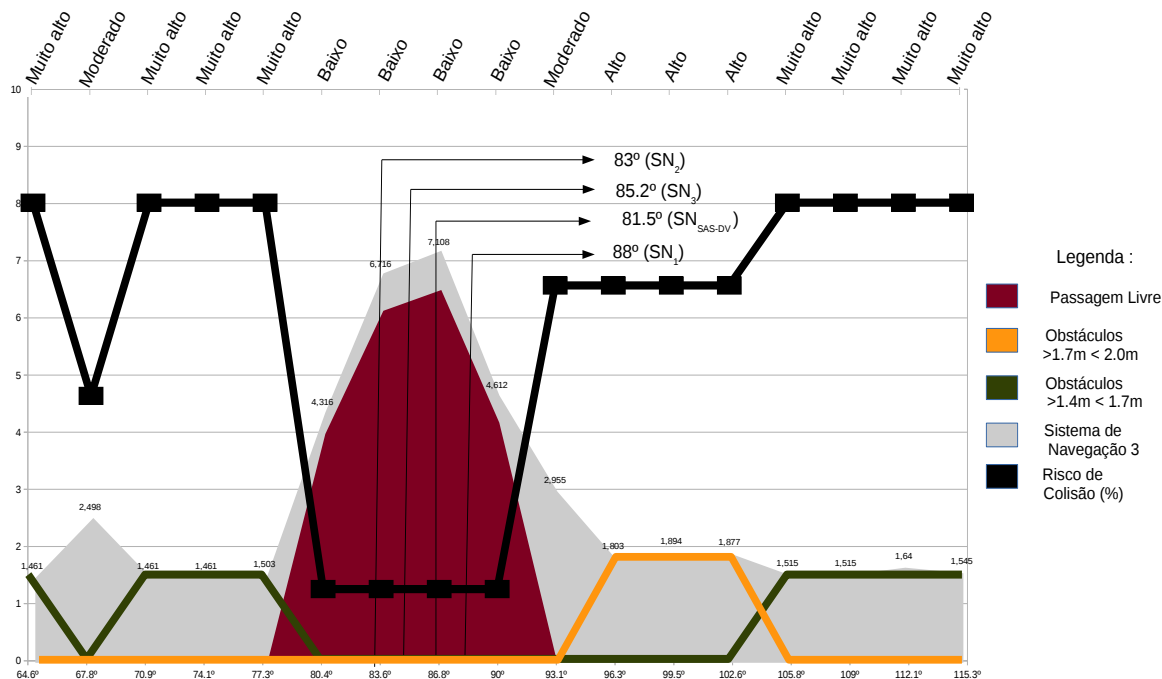
Na área cinza também são apresentados os Objetos Estáticos detectados acima de 2 metros de distância, por meio dos pontos de referências. Esses objetos indicam um risco de colisão menor que os detectados até 2 metros de distância. Observou-se que, na primeira situação, o  $SN_1$  forneceu a direção de  $85^\circ$ , o  $SN_2$   $90^\circ$  e o  $SN_3$   $81.5^\circ$ . Como o  $SN_{SAS-DV}$  indica apenas uma direção entre as 17 possíveis, as direções calculadas pelos sistemas de navegação foram readequadas nas direções mais próximas disponíveis nos mapas (Figuras 5.21, 5.22, 5.23).



**Figura 5.21:** Mapa de riscos produzido a partir da Situação 1 apresentada na Figura 5.18.

No topo desses mapas são apresentados os resultados das inferências usando o segundo *dataset*, por meio do classificador Naive Bayes. O Naive Bayes foi escolhido por ser simples e ter apresentado um baixo índice de erros nas classificações. Os resultados do processo de treinamento usando diferentes classificadores podem ser vistos no Capítulo 6. Essas direções baseiam-se nos 26 pontos de referências (ver Figura 3.9). Como a  $Faixa_1$  possui os mesmos ângulos que a  $Faixa_3$ , mantiveram-se no mapa apenas os resultados das inferências que produziram maiores riscos de colisão. Esses gráficos foram gerados com os valores produzidos pelo

$SN_{SAS-DV}$  ao detectar Objetos Estáticos entre 0.6 a 2 metros de distância do DV.



**Figura 5.22: Mapa de riscos produzido a partir da Situação 2 apresentada na Figura 5.18.**

É possível observar que os mapas apresentam tanto as regiões de Passagem Livre como também as distâncias dos pontos apresentados no  $SN_3$ . Sendo assim, ao verificar as regiões mais seguras para tráfego, o  $SN_{SAS-DV}$  indica a direção resultante da intersecção da região de menos riscos de colisão ( $SN_{SAS-DV}$ ) com a região mais distante do  $SN_3$ . Dessa forma, a contribuição mais significativa deste trabalho é a geração do mapa de riscos baseado em um modelo SAW. O último experimento realizado com o  $SN_{SAS-DV}$  foi com a presença de Objetos Dinâmicos (ver situação 4 da Figura 5.18). É importante lembrar, novamente, que a situação 3 difere da situação 4 apenas pela presença do Objeto Dinâmico. Diante dessas considerações, a única mudança no risco de colisão ocorreu na direção em que o trajeto foi detectado.

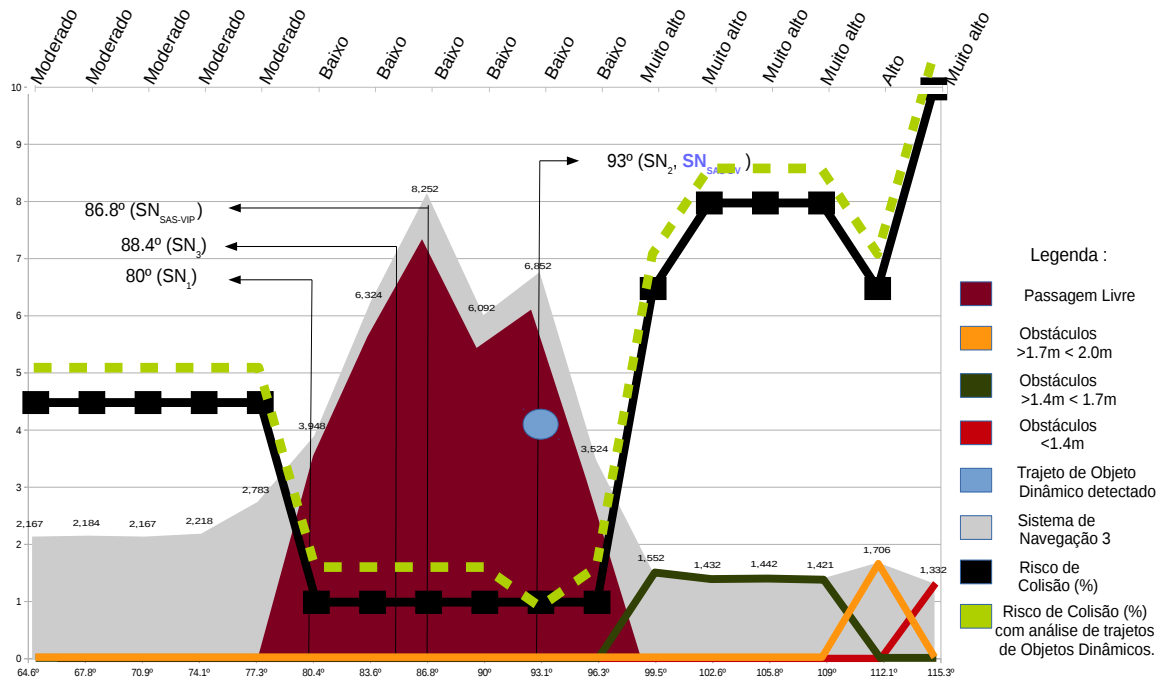


Figura 5.23: Mapa de riscos produzido a partir das situações 3 e 4 apresentadas na Figura 5.18.

# Capítulo 6

## RESULTADOS E DISCUSSÃO

---

---

Este capítulo é composto por duas seções, em que a primeira apresenta os resultados obtidos pelas Arquiteturas de Detecção e Rastreamento e a segunda expõe os resultados dos diferentes sistemas de navegação. Cada seção apresenta também algumas discussões entre as técnicas e os métodos implementados. Ressalta-se que essas implementações possibilitaram realizar um conjunto de análises comparativas e, a partir disso, foram definidas as soluções que compõem o SAS-DV.

### 6.1 Resultados e Discussão - Módulo de Visão

Nesta seção são apresentados, especificamente, os resultados das Arquiteturas de Detecção e Rastreamento (ADRs). Essas arquiteturas produzem pontos em um Mapa( $X, Z$ ) e permitem analisar o trajeto do Objeto Dinâmico em tempo de execução. Com esses mapas produzidos por cada arquitetura, foi possível realizar uma análise comparativa entre algumas características das quatro arquiteturas para Detecção e Rastreamento de Objetos Dinâmicos. Essas características comparadas podem ser vistas nas Tabelas (6.1, 6.2, 6.3 e 6.4), nas quais são apresentados os valores obtidos com o DV paralisado, sem a realização de movimentos na câmera e com o DV em locomoção, em que a câmera sofre movimentos. Os experimentos foram executados em dois ambientes diferentes. As Tabelas 6.1 e 6.3 apresentam os resultados obtidos no ambiente 1, ilustrado no item (a) da Figura 6.1. As Tabelas 6.2 e 6.4 apresentam os resultados relacionados com o ambiente 2, ilustrado no item (b) da Figura 6.1.





**Figura 6.1:** (a) Ambiente 1. (b) Ambiente 2.

A velocidade de detecção foi contabilizada por meio dos pontos apresentados no mapa (X, Z) dividido pelo tempo (detecção por segundos) de aquisição dos dados. Quanto mais pontos o sistema produziu no mapa (ver Figuras 5.12, 5.13, 5.14, 5.15), maior foi a velocidade de detecção da arquitetura. A eficácia também foi contabilizada por meio desses pontos, caso o ponto tenha sido mapeado corretamente. Se o ponto foi mapeado em uma área não desejada, significa que a técnica detectou o OD de maneira equivocada.

**Tabela 6.1:** Comparação das técnicas de DROD com o DV paralisado - ambiente 1

ADRs	Velocidade de detecção	Efetividade	CPU	RAM
A1-RP	2,9	93,50%	64,57%	4,16%
A2-FOF	2,68	85,10%	39,42%	3,74%
A3-SB	3,08	96,10%	49,34%	4,47%
A4-CAM	3,45	95,65%	46,38%	4,13%

**Tabela 6.2:** Comparação das técnicas de DROD com o DV paralisado - ambiente 2

ADRs	Velocidade de detecção	Efetividade	CPU	RAM
A1-RP	2,76	94,90%	66,03%	4,10%
A2-FOF	2,9	78,70%	42,84%	3,80%
A3-SB	3,94	90,70%	49,18%	4,27%
A4-CAM	3,07	100,00%	47,90%	4,26%

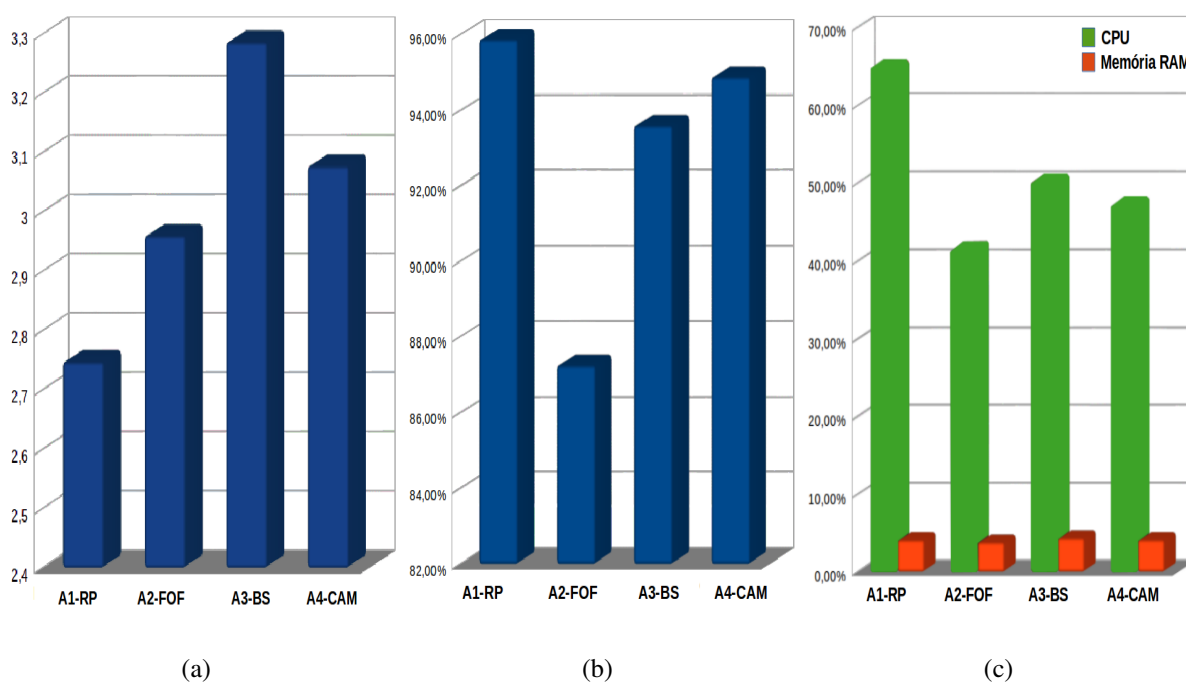
A partir de todos os dados relacionados com a velocidade de detecção, a efetividade e o custo computacional, gerou-se a média geral de cada uma dessas características adquiridas por cada arquitetura implementada. Os gráficos da Figura 6.2 apresentam, respectivamente, as médias obtidas por cada arquitetura e relacionadas com as seguintes características: Velocidade de detecção; Efetividade e Custo computacional.

**Tabela 6.3: Comparação das técnicas de DROD com o DV em movimento - ambiente 1**

ADRs	Velocidade de detecção	Efetividade	CPU	RAM
A1-RP	2,6	97,50%	63,30%	4,06%
A2-FOF	3,06	95,65%	42,60%	3,85%
A3-SB	3	95,60%	51,20%	4,44%
A4-CAM	2,86	90,70%	46,90%	4,20%

**Tabela 6.4: Comparação das técnicas de DROD com o DV em movimento - ambiente 2**

ADRs	Velocidade de detecção	Efetividade	CPU	RAM
A1-RP	2,73	97,60%	65,40%	4,20%
A2-FOF	3,2	89,60%	41,00%	3,94%
A3-SB	3,125	92,00%	50,85%	4,24%
A4-CAM	2,93	93,20%	47,50%	4,00%

**Figura 6.2: (a) Média da Velocidade de detecção das arquiteturas; (b) Média da efetividade das arquiteturas; (c) Custo computacional médio das arquiteturas (CPU e memória RAM).**

No instante em que um OD é detectado, um bipe é emitido com uma intensidade baseada na sua posição 3D e isso indica ao DV que ele deve realizar movimentos menos bruscos, para permitir que o rastreamento do OD seja executado com maior qualidade. Ainda assim, pode ser que o rastreamento não seja produzido com a qualidade esperada, caso o OD esteja numa velocidade muito alta.

Diversos experimentos foram realizados com pessoas andando numa velocidade de 0,7m/s a 2m/s (Ver materiais suplementares do artigo (CORDEIRO; PEDRINO, 2019b)). Os sistemas que usaram as técnicas de Subtração de *Background* (SB), de Fluxo Óptico de Farneback e CamShift foram superiores ao sistema de Reconhecimento de Padrões no processo de detecção e rastreamento.

As técnicas de Reconhecimento de Padrões para a análise de Objetos Dinâmicos têm sido implementadas por alguns projetos (TAPU; MOCANU; ZAHARIA, 2013; SOYLU; CAUS-MAECKER; DESMET, 2009; MOHAMMED; MORRIS, 2015). Essas técnicas apresentam muita qualidade na construção da trajetória do objeto em movimento e nas estimativas de direção e velocidade. Entretanto, se o sistema não conseguir reconhecer um objeto, previamente treinado, como sendo um possível OD, o sistema perde sua finalidade. Dessa forma, qualquer objeto que possa ter movimento autônomo necessita ser reconhecido pelo sistema. Isso envolve um treinamento exaustivo e com uma grande base de dados para cada classe de objeto envolvido.

A técnica de Reconhecimento de Padrões, implementada na Arquitetura A1-RP, apresentou dificuldades em classificar objetos a longa distância (Ver item (a) da Figura 6.2) comparada as outras técnicas. Isso ocorre devido ao sistema reconhecer uma pessoa apenas quando ela está mais próxima do Sensor de Aquisição (SA). O detalhe mais interessante do uso de Reconhecimento de Padrões para esse tipo de análise é com relação à independência do movimento de câmera, pois as mudanças de regiões não geram grandes prejuízos no processo de detecção de um objeto. Pode ser observado nas Tabelas e no item (b) da Figura 6.2 que a técnica de Reconhecimento de Padrões implementada forneceu os menores erros no processo de detecção e rastreamento. Como desvantagem, foi a técnica que mais consumiu processamento e memória (Ver Figura 6.2), devido à realização das comparações com o conjunto de padrões previamente treinados.

A arquitetura A2-FOF, implementada com a técnica de Fluxo Óptico de Farneback, apresentou o menor custo computacional, com o menor processamento e uso de memória RAM (ver item (c) da Figura 6.2). Essa arquitetura detectou OD a partir de uma distância de forma semelhante com as arquiteturas A3-SB e A4-CAM, porém, com maiores erros na detecção (ver item (b) da Figura 6.2). Neste trabalho, também foram realizados testes com a técnica de FOF em imagens infravermelhas. O desempenho melhorou, mas, a quantidade de erros de detecção aumentou de tal forma que não pôde ser considerada como uma alternativa. Com a câmera estática, enquanto a técnica de FOF, usando o Sensor RGB, obteve quase 90,3% de eficácia, a mesma técnica, usando o Sensor IV, obteve quase 47,7%. Com a câmera em movimento, a téc-

nica de FOF, usando o Sensor RGB, obteve quase 84% de eficácia, enquanto a mesma técnica usando o Sensor IV obteve apenas 58.9%. Essa tentativa foi realizada com intuito de produzir um sistema que não sofra nenhum tipo de problema com relação à mudança de iluminação. Porém, após esses testes, esse método foi considerado útil apenas em imagens obtidas pelo Sensor RGB. É importante lembrar que o Sensor RGB é sensível a alterações na iluminação.

A arquitetura A3-SB, que teve a técnica de Subtração de *Background* implementada, apresentou-se como a abordagem que realizou a maior quantidade de detecções de um objeto em movimento comparado com as outras arquiteturas. Para gerar essa análise comparativa, os experimentos realizados com as quatro arquiteturas utilizaram o mesmo trajeto (ver item (a) da Figura 6.2). Ressalta-se que a arquitetura A3-SB teve uma pequena quantidade de erros (ver item (b) da Figura 6.2) e se apresentou como uma ótima abordagem para ser aplicada no SAS-DV. Isso ocorreu mesmo sendo o método que mais consumiu memória RAM e a segunda que gerou maior processamento (ver item (c) da Figura 6.2). Essa arquitetura tem como sua melhor característica a independência da iluminação por ter sido implementada em imagens do Sensor IV.

A técnica de Subtração de *Background* é uma das técnicas que tem sido menos usada nesse tipo de cenário. Essa técnica foi desenvolvida, mais especificamente, para ambientes controlados e para câmeras estáticas. Devido aos movimentos produzidos pelo DV, ao se locomover, a análise de um objeto em movimento pode ser prejudicada. Com a ausência da estabilização do vídeo, com o aumento do limiar que apresenta apenas as regiões que produziram maiores mudanças e com poucos movimentos no SA, foi possível obter os resultados desejados. Ressalta-se que a técnica de Subtração de *Background* não exige nenhum tipo de treinamento para a detecção de Objetos Dinâmicos como ocorre com a técnica de Reconhecimento de Padrões.

A técnica CamShift, implementada na arquitetura A4-CAM, apresentou-se como uma das melhores técnicas de rastreamento de Objetos Dinâmicos por ter um alto desempenho e permitir reconstruir a trajetória com qualidade. No entanto, é uma técnica que pode ser aplicada apenas em cenários que não sofram fortes mudanças de iluminação ou que presenciem objetos com cores semelhantes (*single hue object*). Isso pode gerar confusão e indicar que esses objetos com cores semelhantes podem ser parte do OD e prejudicar no processo de rastreamento. Uma alternativa é restringir sua semelhança, alterando alguns de seus limiares. Outra característica observada nos experimentos com a técnica CamShift foi na impossibilidade de calcular a largura e altura do Objeto Dinâmico da mesma forma quando realizado pelas técnicas de classificação e de Subtração de *Background*. A técnica Fluxo Óptico de Farneback permitiu calcular apenas um valor aproximado.

A arquitetura A2-FOF apresentou bons resultados na detecção de Objetos Dinâmicos, mesmo com a presença de movimentos na câmera. No entanto, não apresentou uma boa qualidade no processo de rastreamento comparado com as outras técnicas. Dessa forma, o Fluxo Óptico de Farneback é indicado no processo de detecção, principalmente, porque permite analisar a diferença entre o movimento da câmera e o movimento do OD. Depois dessa detecção, uma outra técnica pode ser selecionada para realizar o rastreamento, ou seja, quando o FOF detectar um OD, o sistema pode emitir um alerta para o DV diminuir sua movimentação, para que as técnicas de SB ou de CamShift produzam um trajeto com mais qualidade.

Após essa discussão, criou-se uma classificação (*ranking*) (Tabela 6.5) que apresenta as melhores Arquiteturas de Detecção e Rastreamento (ADR) diante das adversidades encontradas nos experimentos, que são: iluminação; movimento de vídeo e Independência de Treinamento Supervisionado (ITS).

**Tabela 6.5: Comparação entre as Arquiteturas de Detecção e Rastreamento de OD (DRODs)**

ADRs	Iluminação	Movimento de Vídeo	ITS
A1-RP	Regular	Excelente	Ruim
A2-FOF	Regular	Bom	Excelente
A3-SB	Excelente	Regular	Excelente
A4-CAM	Ruim	Excelente	Excelente

Diante de todas as tabelas apresentadas, verificou-se que não existe a melhor arquitetura devido a elas possuírem diferentes benefícios em relação aos objetivos e às adversidades do ambiente. Dessa forma, criou-se um protótipo com a união das arquiteturas A3-SB e A4-CAM. Como na arquitetura A4-CAM foi implementada a técnica de Fluxo Óptico de Farneback para detecção do OD, proposta na arquitetura A2-FOF, ela também pode ser considerada como incorporada. O protótipo desenvolvido foi baseado na arquitetura A3-SB devido a sua disponibilidade de atuar com a presença ou ausência de luz. Porém, quando o acelerômetro, integrado no dispositivo Kinect, detectar movimentos bruscos realizados pelo DV, a arquitetura A4-CAM é inicializada. Quando os movimentos no Sensor de Aquisição são mais estáveis, o rastreamento é realizado pela arquitetura A3-SB.

Apesar da técnica de Reconhecimento de Padrões já ter sido implementada, ela não é considerada essencial para essa versão do protótipo devido ao fato de reconhecer apenas pessoas e não detectar objetos distantes. Caso o DV sinta-se mais confortável em ter diferentes classificações, a técnica de Reconhecimento de Padrões pode ser facilmente integrada, necessitando apenas de treinamento supervisionado e um dispositivo que suporte um maior poder de processamento. Dessa forma, conseguiu-se obter um protótipo que realiza a detecção e o rastreamento

de objetos dinâmicos mesmo presenciando mudanças na iluminação, nos movimentos no sensor de aquisição e sem depender de classificação.

Neste trabalho implementou-se a detecção de Objetos Dinâmicos para os dois tipos de sensores (RGB e IV). Dessa forma, caso alguma técnica seja afetada por ruídos, uma técnica alternativa que use o outro sensor pode ajudar na análise.

Entre as principais dificuldades encontradas nesta pesquisa, estão os movimentos rápidos no sensor de aquisição, causados pela forte movimentação do DV. Esses movimentos dificultam a detecção e o rastreamento dos Objetos Dinâmicos. A mudança repentina da iluminação no ambiente também atrapalhou no processo de detecção e classificação de objetos.

## 6.2 Resultados e Discussão - Módulo de Fusão

Nesta seção, são apresentados e comparados os resultados obtidos com o sistema SAS-DV com outros sistemas de navegação. Como foi visto na seção de experimentos, foram gerados alguns mapas de riscos para indicar ao DV as direções com menor risco de colisão. As regiões que presenciavam Objetos Estáticos tiveram o aumento do risco de colisão na suposta direção. Nas direções em que foram detectados trajetos executados por Objetos Dinâmicos, pessoa ou passagem livre, os riscos de colisão diminuíram.

### 6.2.1 Análise comparativa entre os Sistemas de Navegação implementados

A Tabela 6.6 apresenta uma comparação entre os sistemas  $SN_1$ ,  $SN_2$ ,  $SN_3$  e o  $SN_{SAS-DV}$  a partir dos resultados obtidos nas quatro situações apresentadas na Figura 5.18 do Capítulo 5. Os sistemas  $SN_1$  e  $SN_3$  apenas indicaram uma possível direção para locomoção, enquanto o  $SN_2$  disponibilizou uma região e o  $SN_{SAS-DV}$  forneceu um mapa com os riscos de colisão para todas as direções apresentadas na Figura 5.18.

Essa comparação foi efetuada pelas seguintes características: Direção Indicada (DI), Distância da DI (DDI) e Risco de Colisão (RC). O RC se baseou no modelo de crenças do SAS-DV devido aos sistemas  $SN_1$ ,  $SN_2$  e  $SN_3$  não fornecerem essa informação. Com a indicação de apenas uma região central, pelo  $SN_2$ , foi necessário definir uma direção para ser comparada aos demais Sistemas de Navegação.

Para que o  $SN_2$  não ficasse restrito à direção de  $90^\circ$ , verificou-se um outro processo de DI. Segundo Nadia *et al.* (2015), os pontos verdes produzem os riscos mais distantes na região de

locomoção segura em comparação com os pontos de outras cores. Dessa forma, definiu-se uma DI baseada na região em que os pontos verdes estão mais concentrados.

**Tabela 6.6: Análise comparativa entre os quatro Sistemas de Navegação**

	Situação 1			Situação 2		
SN	DI	DDI	RC	DI	DDI	RC
$SN_1$	85°	6.720	10,00%	88°	5.820	10,00%
$SN_2$	90°	6.884	10,00%	83°	6.716	10,00%
$SN_3$	81.5°	6.750	10,00%	85.2°	6.981	10,00%
$SN_{SAS-DV}$	83.6°	8.284	10,00%	81.5°	7.108	10,00%
	Situação 3			Situação 4		
SN	DI	DDI	RC	DI	DDI	RC
$SN_1$	80°	3.800	15,00%	80°	3.800	15,00%
$SN_2$	93°	6.852	10,00%	93°	6.852	3,00%
$SN_3$	88.4°	7.215	10,00%	88.4°	7.215	10,00%
$SN_{SAS-DV}$	86.8°	8.252	10,00%	93°	6.852	3,00%

É possível observar que as Distâncias Indicadas (DI) pelos quatro sistemas divergem devido as suas abordagens de navegação, embora sejam disponibilizadas direções com um Risco de Colisão (RC) próximo. Esses casos divergem com maior frequência quando são detectados Objetos Dinâmicos e obstáculos mais próximos nas regiões de tráfego. Os resultados foram obtidos a partir de todas as situações ilustradas na Figura 5.18.

Entre as quatro situações apresentadas na Tabela 6.6, o  $SN_{SAS-DV}$  apresentou mais confiabilidade graças à combinação de crenças entre a detecção de obstáculos próximos ao DV e a região disponível para passagem livre.

Observou-se que, na primeira situação, o  $SN_1$  forneceu a direção de 85°, o  $SN_2$  de 90° e o  $SN_3$  de 82°. Como o SAS-DV indica apenas uma direção entre as 17 possíveis, as direções calculadas pelos Sistemas de Navegação foram readequadas nas direções mais próximas disponíveis nos mapas (Figuras 5.21, 5.22, 5.23).

Nas situações 1 e 2 (Ver Figuras 5.21 e 5.22) pode ser visto que os riscos de colisão apresentados pelos dois mapas de riscos (*datasets* do tipo *boolean* e do tipo *numeric*), usando diferentes classificadores, foram baixos (10% e *low*). Porém, é possível analisar qual direção pode ser a mais indicada com base nas maiores distâncias de passagem livre e distantes de obstáculos. Dessa forma, fica claro que a abordagem do  $SN_{SAS-DV}$  fornece uma direção mais confiável.

Na situação 3 (Ver Figura 5.23), o  $SN_1$  apresentou o maior risco de colisão comparado com os outros sistemas. A DI (80°) prevista para este caso gerou uma probabilidade de 15% devido ao modelo de treinamento do  $SN_{SAS-DV}$  detectar a presença de obstáculos próximos a 2,7m (2.783mm) e, conseqüentemente, não gerar uma passagem segura. Observa-se no mapa que a

passagem livre dessa direção está bem próxima de obstáculos.

Isto é um caso interessante para validar e analisar a importância de um Sistema de Navegação baseado em crenças. Na situação 4, na qual o caminho executado por um Objeto Dinâmico (pessoa) foi detectado, a probabilidade de Risco de Colisão foi de 3% para o  $SN_{SAS-DV}$ , enquanto sem essa análise, a probabilidade foi de 10% (Situação 3). Este resultado é importante para indicar uma direção diferente daquela que seria fornecida ao deficiente visual (veja a linha tracejada verde na Figura 5.23). Observa-se que, nesta mesma situação, o  $SN_2$  apresentou o mesmo resultado de 3%, porém, esse resultado ocorreu por coincidência, pois o  $SN_2$  não armazena os trajetos executados por objetos dinâmicos.

Nas Figuras 5.21 e 5.22 podem ser vistas algumas classificações diferentes baseadas nos dois modelos (*datasets*) propostos. Na Figura 5.22, na direção  $93.1^\circ$ , a teoria das Redes Bayesianas aplicada no primeiro conjunto de dados de treinamento  $dataset_{boolean}$  indica uma probabilidade de 65% de colisão, da mesma forma que as direções seguintes ( $96.3^\circ$ ,  $99.5^\circ$  e  $102.6^\circ$ ). O classificador *Naive Bayes* classificou, para a mesma direção, mas com o uso do segundo conjunto de dados de treinamento  $dataset_{numeric}$ , uma possibilidade moderada de colisão, diferente das direções seguintes ( $96.3^\circ$ ,  $99.5^\circ$  e  $102.6^\circ$ ) que indicaram uma alta possibilidade de colisão. Na Figura 5.23 pode ser visto algo semelhante na direção  $99.5^\circ$ .

Com os experimentos, verificou-se que o segundo conjunto de dados  $dataset_{numeric}$  apresentou mais confiabilidade por aplicar nas inferências as posições exatas dos obstáculos detectados. Isso permite introduzir mais crenças e produzir classificações mais confiáveis. Exemplos da composição das inferências do primeiro e segundo conjuntos de dados *datasets* estão disponíveis nas Tabelas 5.1 e 5.2 da Seção 5.2.

No topo desses mapas são apresentados os resultados das inferências usando o segundo conjunto de dados  $dataset_{numeric}$  por meio do classificador *Naive Bayes*. O *Naive Bayes* foi escolhido por ser simples e ter apresentado um baixo índice de erros nas classificações. Os resultados do processo de treinamento usando diferentes classificadores pode ser visto na Tabela 6.7.

Na Figura 5.23 pode ser visto o quanto a posição em milímetros é importante para analisar o risco de colisão. Na direção  $93.1^\circ$  foi detectada uma posição referente a um trajeto realizado por um Objeto Dinâmico. Esse objeto foi detectado a 4005 milímetros de distância do DV. Sendo assim, foi possível obter um cálculo mais preciso e um resultado mais confiável que o primeiro conjunto de dados *dataset*, em que apenas identifica a presença ou ausência em determinada direção. É importante ressaltar que, nessa direção, o *Naive Bayes* classificou o risco de colisão como *Baixo*, segundo a distância da passagem livre e a ausência de obstáculos. Ao detectar a



posição de um trajeto já realizado, o classificador deveria aumentar a confiança nesta direção, porém, como já havia sido classificado como *Baixo*, a confiança se manteve. Caso a direção de 93.1° a passagem livre tivesse uma distância menor e fosse classificada como *Moderado*, após a detecção do trajeto realizado pelo OD, essa classificação seria alterada para *Baixo*.

A principal contribuição deste estudo está na confiabilidade de navegação que o SAS-DV propõe. Diversos trabalhos (ALADREN et al., 2014), (COSTA et al., 2012), (NADIA et al., ), (LAKDE; PRASAD, 2015), (SAPUTRA; WIDYAWANKA; SANTOSA, 2014) fornecem direções para o DV poder trafegar, porém, o SAS-DV disponibiliza um mapa e uma direção baseada em crenças. Essas crenças ajudam a estabelecer a direção mais confiável.

A Tabela 6.7 mostra os resultados dos processos de treinamento (no segundo *dataset*) de seis classificadores usando validação cruzada (uso de *Folds* = 10). Essa tabela consiste nos seguintes recursos: os classificadores usados; as Instâncias Corretamente Classificadas (ICC) para cada classificador; *Relative Absolute error (RAerror)* e a precisão média (média-Pc) de cada treinamento.

**Tabela 6.7: Comparação de classificadores com validação cruzada - (uso de *Folds* = 10)**

<b>Classificadores</b>	<b>ICC</b>	<b>RAerror</b>	<b>média-Pc</b>
BayesNet	72.85%	47.85%	77,5%
Multilayer Perceptron	78.57%	41.41%	78,9%
RandomTree	96.42%	63.21%	96,5%
Random Forest	97.14%	63.70%	97,2%
Naive Bayes	98.57%	8.034%	98,6%
HoeffdingTree	99.28%	7.595%	99,3%

A Tabela 6.7 mostra que os classificadores relacionados com Decision Tree Classifier acompanhado do *Naive Bayes* tiveram as maiores ICC e média-Pc. O classificador *Hoeffding Tree* e *Naive Bayes* tiveram as menores taxa de erros. A ICC do *BayesNet* diminuiu muito no instante em que o *dataset* foi modelado com uma alta quantidade de probabilidades, ao usar o tipo de dado *numeric*. Para que esse classificador aumente a sua taxa de acertos, é necessário adicionar muitas informações ao *dataset* até que diminua a quantidade de intervalos que geram dúvidas ao classificador. Esses classificadores puderam validar o processo de modelagem do segundo *dataset*. O presente *dataset* (*dataset<sub>numeric</sub>*) pode ser visto no vídeo dos materiais suplementares do seguinte artigo (CORDEIRO; PEDRINO, 2019a).

O SAS-DV apresentou ser possível compreender a situação e gerar o mapa de riscos em qualquer ambiente *indoor*. O uso da técnica de Reconhecimento de Padrões seria apenas um complemento para melhorar a qualidade da tomada de decisão, mas não uma necessidade. De-

pendendo do reconhecimento de um objeto, uma nova crença pode ser atribuída para a tomada de decisão em determinada direção. Outras crenças podem surgir por meio de *tags* (como *RFID* ou *QR-Code*), sons e sensores. Todas as características obtidas pelo Módulo de Visão se baseiam nos formatos, posições e ações dos objetos em cena. Essas características são importantes para a obtenção da consciência da situação, independentemente de uma entidade específica, facilitando a tomada de decisão a qualquer instante, em qualquer ambiente.

Ressalta-se que a fusão de dados usando o modelo *SAW* para sistemas de apoio ao DV é uma técnica pouco explorada e isso diminui quando esse modelo atinge o nível de projeção. Na Tabela 1 é visível que os projetos não proveem técnicas para projetar impactos num futuro próximo. E essa é uma das principais contribuições do SAS-DV. Ter a percepção dos objetos e suas ações, compreendê-las e poder projetar colisões sem depender da presença de entidades específicas é o que motivou o desenvolvimento do SAS-DV. Consequentemente, exigiu-se a criação de uma nova arquitetura, que possibilita a percepção de diferentes aspectos no ambiente. Outra contribuição possível que esta arquitetura possibilita é a disponibilidade de ajustar as crenças de acordo com a avaliação do DV. Dessa forma, é possível que o mapa de riscos se torne mais eficaz.

Além de fornecer um mapa de riscos e uma direção confiável ao DV, o SAS-DV emite sinais sonoros que representam as mudanças de posição realizadas por Objetos Dinâmicos, o que proporciona compreender esse trajeto próximo do tempo real. O sistema também emite alertas quando são detectados Objetos Estáticos e o DV pode requisitar o mapa de riscos quando desejar.

# Capítulo 7

## CONCLUSÃO

---

---

Muitas pesquisas desenvolvidas com a finalidade de apoiar a locomoção de um DV indicam a existência de obstáculos, suas posições, distâncias e rotas alternativas para tráfego. Apesar disso, poucas têm aplicado a projeção de impactos baseados na compreensão de situações. Esta pesquisa disponibiliza uma arquitetura que implementa algumas técnicas de extração de características para possibilitar os processos de percepção e de compreensão do ambiente em que um DV está. O processo de projeção de impactos desta arquitetura foi concebido após a geração de um mapa de riscos, formado pelo resultado de um conjunto de inferências. Cada inferência foi produzida para cada uma das dezessete direções definidas nesta tese para indicar o risco de colisão. Ressalta-se que as inferências necessitam de informações refinadas, para que os riscos de colisão sejam classificados de modo confiável. Dessa forma, os resultados das técnicas de extração de características implementados no Módulo de Visão puderam ser validados por meio das inferências executadas pelo Módulo de Fusão. Sendo assim, esta tese pode contribuir para o desenvolvimento de Sistemas de Navegação que visam projetar impactos para os DVs.

Os resultados obtidos, após a aplicação das inferências nos dois conjuntos de dados (*datasets*), deixaram claro como o SAS-DV pode fornecer ao DV os riscos de colisão com maior confiabilidade. Constatou-se que, com a inserção de informações, de maneira balanceada nos *datasets*, houve um aumento na precisão média do modelo usado para classificar o risco de colisão em uma determinada direção. Dessa forma, as principais vantagens ao aplicar a fusão da informação de alto nível é o aumento da confiabilidade quando comparado a uma solução proposta de maneira empírica, na qual, normalmente, são usados intervalos de valores e que não apresentam uma precisão satisfatória. Os *datasets* propostos também demonstraram ser flexíveis com a inserção de informações que representam novas crenças. No entanto, o uso da Fusão da Informação de alto nível exige uma análise complexa, necessitando de dados que passam por

processos de refinamento e que são modelados de uma maneira sistêmica, para que a precisão na classificação seja alta. Devido a isso, esse tipo de sistema necessita que um especialista realize a inserção de novas informações nos *datasets* e, conseqüentemente, novos treinamentos. Ao fazer isso, o sistema se torna mais confiável.

Dentre estas contribuições estão: o desenvolvimento de uma arquitetura para analisar o tamanho, a distância e a posição dos Objetos Estáticos e Dinâmicos; a análise de trajetos; a conversão das posições dos obstáculos no plano 2D para o plano 3D; a geração de um mapa de riscos para indicar a direção que possui menos riscos de colisão. Ainda neste trabalho foi desenvolvido um processo de extração de características mais refinado relacionado com a detecção de Objetos Dinâmicos, principalmente, quando há presença de movimentos no Sensor de Aquisição. Além disso, foi desenvolvida também uma metodologia para analisar o desempenho de diferentes técnicas conhecidas na literatura para DROD.

Para validar essa metodologia, foram implementadas quatro novas arquiteturas para a realização de uma análise comparativa. Cada arquitetura foi constituída com diferentes técnicas de DROD e com módulos que permitem reconstruir os trajetos realizados por Objetos Dinâmicos, em um mapa 3D. A principal contribuição desta metodologia foi realizar uma análise comparativa entre as arquiteturas abordadas com as seguintes características: Velocidade de Detecção, Efetividade e Custo computacional. Essa análise pode ser realizada mesmo com as diferentes adversidades relacionadas com a iluminação e com os constantes movimentos produzidos no Sensor de Aquisição. Além disso, essa metodologia permite disponibilizar, para futuros projetos, uma forma de analisar as características abordadas de uma nova técnica de DROD.

Outras contribuições desta tese se referem ao módulo que emite diferentes tipos de bipes. Esses bipes podem indicar a posição de um objeto e o risco de colisão em uma determinada direção com diferentes intensidades e durações. Exemplos da emissão de bipes e do sistema de rastreamento de Objetos Dinâmicos podem ser vistos nos materiais suplementares dos seguintes trabalhos (CORDEIRO; PEDRINO, 2019b, 2019c, 2019a; CORDEIRO; PEDRINO, 2018). Por fim, no presente trabalho, um novo Sistema de Navegação foi proposto e comparações com outras abordagens foram realizadas para mostrar as vantagens de aplicação do modelo *SAW*.

## REFERÊNCIAS

---

---

- ABOWD, G. D. et al. Towards a better understanding of context and context-awareness. Springer Berlin Heidelberg, Berlin, Heidelberg, p. 304–307, 1999. Disponível em: <[https://doi.org/10.1007/3-540-48157-5\\_29](https://doi.org/10.1007/3-540-48157-5_29)>.
- ALADREN, A. et al. Navigation assistance for the visually impaired using rgb-d sensor with range expansion. *IEEE Systems Journal*, PP, n. 99, p. 1–11, 2014. ISSN 1932-8184.
- ANAGNOSTOPOULOS, C.; HADJIEFTHYMIADES, S. Advanced inference in situation-aware computing. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, v. 39, n. 5, p. 1108–1115, Sept 2009. ISSN 1083-4427.
- ANDO, B. et al. A sensing architecture for mutual user-environment awareness case of study: A mobility aid for the visually impaired. *IEEE Sensors Journal*, v. 11, n. 3, p. 634–640, March 2011. ISSN 1530-437X.
- ANGIN, P.; BHARAT; BHARGAVA, K. *Real-time Mobile-Cloud Computing for Context-Aware Blind Navigation*. 2011.
- BOURBAKIS, N.; MAKROGIANNIS, S. K.; DAKOPOULOS, D. A system-prototype representing 3d space via alternative-sensing for visually impaired navigation. *IEEE Sensors Journal*, v. 13, n. 7, p. 2535–2547, July 2013. ISSN 1530-437X.
- BRADSKI, G.; KAEHLER, A. *Learning OpenCV: Computer Vision with the OpenCV Library*. [S.l.]: O'Reilly Media, 2008. ISBN 9780596554040.
- BRADSKI, G. R. Real time face and object tracking as a component of a perceptual user interface. In: *Proceedings Fourth IEEE Workshop on Applications of Computer Vision. WACV'98 (Cat. No.98EX201)*. [S.l.: s.n.], 1998. p. 214–219.
- CANNY, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8, n. 6, p. 679–698, Nov 1986. ISSN 0162-8828.
- CHAN, K. Y.; ENGELKE, U.; ABHAYASINGHE, N. An edge detection framework conjoining with {IMU} data for assisting indoor navigation of visually impaired persons. *Expert Systems with Applications*, v. 67, p. 272 – 284, 2017. ISSN 0957-4174.
- CONRADY, S.; JOUFFE, L. *Bayesian Networks & BayesiaLab, A Practical Introduction for Researchers*. [S.l.]: Bayesia-USA, 2015. ISBN 978-0-9965333-0-0.
- CORDEIRO, N. H. et al. A data fusion architecture proposal for visually impaired people. In: *29th Conference on Graphics, Patterns and Images, SIBGRAPI Sao Paulo, Brazil*. [S.l.: s.n.], 2016. p. 158–165.

- CORDEIRO, N. H.; PEDRINO, E. C. An architecture for collision risk prediction for visually impaired people. In: *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. [S.l.: s.n.], 2018. p. 305–312.
- CORDEIRO, N. H.; PEDRINO, E. C. Collision risk prediction for visually impaired people using high level information fusion. *Engineering Applications of Artificial Intelligence*, v. 81, p. 180 – 192, 2019. ISSN 0952-1976. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0952197619300405>>.
- CORDEIRO, N. H.; PEDRINO, E. C. Data used for detection and tracking of dynamic objects for visually impaired people. *Data in Brief*, v. 26, p. 104403, 2019. ISSN 2352-3409. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S2352340919307589>>.
- CORDEIRO, N. H.; PEDRINO, E. C. A new methodology applied to dynamic object detection and tracking systems for visually impaired people. *Computers & Electrical Engineering*, v. 77, p. 61 – 71, 2019. ISSN 0045-7906. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S004579061830956X>>.
- COSTA, P. et al. Obstacle detection using stereo imaging to assist the navigation of visually impaired people. *Procedia Computer Science*, v. 14, p. 83 – 93, 2012. ISSN 1877-0509. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1877050912007727>>.
- DAWKINS, P. *Calculus III - Green's Theorem*. [S.l.], 2013.
- ENDSLEY, M.; BOLTE, B.; JONES, D. *Designing for Situation Awareness: An Approach to User-Centered Design*. [S.l.]: Taylor & Francis, 2003. ISBN 9780748409662.
- GANG, Z.; XIAOLI, W.; LIRONG, W. Motion analysis and research of local navigation system for visual-impaired person based on improved lk optical flow. In: *Fifth International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*. [S.l.: s.n.], 2012. p. 348–351.
- HARRIS, C.; STEPHENS, M. A combined corner and edge detector. In: *In Proc. of Fourth Alvey Vision Conference*. [S.l.: s.n.], 1988. p. 147–151.
- HUANG, J. et al. Optical flow based real-time moving object detection in unconstrained scenes. *CoRR*, abs/1807.04890, 2018. Disponível em: <<http://arxiv.org/abs/1807.04890>>.
- IPINA, D. L.; BOTRÁN, T. L.; NOVOA, U. L. Blindshopping: Enabling accessible shopping for visually impaired people through mobile technologies. In: *Proceedings of the 9th International Conference on Toward Useful Services for Elderly and People with Disabilities: Smart Homes and Health Telematics*. Berlin, Heidelberg: Springer-Verlag, 2011. (ICOST'11), p. 266–270. ISBN 978-3-642-21534-6.
- JABNOUN, H.; BENZARTI, F.; AMIRI, H. Visual substitution system for blind people based on sift description. In: *Soft Computing and Pattern Recognition (SoCPaR), 2014 6th International Conference of*. [S.l.: s.n.], 2014. p. 300–305.
- LAKDE, C. K.; PRASAD, P. S. Navigation system for visually impaired people. In: *2015 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC)*. [S.l.: s.n.], 2015. p. 0093–0098.

- LEE, W.-P.; LEE, K.-H. Making smartphone service recommendations by predicting users' intentions: A context-aware approach. *Information Sciences*, v. 277, p. 21 – 35, 2014. ISSN 0020-0255.
- LIGGINS, M.; HALL, D.; LLINAS, J. *Handbook of Multisensor Data Fusion: Theory and Practice, Second Edition*. [S.l.]: CRC Press, 2008. (Electrical Engineering & Applied Signal Processing Series). ISBN 9781420053098.
- MARIOTTI, S. P. *World Health Organization (WHO) Visual impairment and blindness*. [S.l.], 2014.
- MASCETTI, S. et al. Robust traffic lights detection on mobile devices for pedestrians with visual impairment. *Computer Vision and Image Understanding*, v. 148, p. 123 – 135, 2016. Special issue on Assistive Computer Vision and Robotics -.
- MAYBECK, P. S. *Stochastic models, estimation, and control*. [S.l.: s.n.], 1979. v. 141. (Mathematics in Science and Engineering, v. 141).
- MEKHALFI, M. L. et al. Toward an assisted indoor scene perception for blind people with image multilabeling strategies. *Expert Syst. Appl.*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 42, n. 6, p. 2907–2918, abr. 2015. ISSN 0957-4174.
- MEKHALFI, M. L. et al. Recovering the sight to blind people in indoor environments with smart technologies. *Expert Syst. Appl.*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 46, n. C, p. 129–138, mar. 2016. ISSN 0957-4174.
- MOHAMMED, A. D.; MORRIS, T. Optical flow estimation using local features. In: *Proceedings of the World Congress on Engineering 2015*. [S.l.: s.n.], 2015. v. 1. ISBN 978-988-19253-4-3.
- NADIA, K. et al. A navigation system for the visually impaired: A fusion of vision and depth sensor. *Applied Bionics and Biomechanics*, v. 2015.
- PEI, S. C.; WANG, Y. Y. Census-based vision for auditory depth images and speech navigation of visually impaired users. *IEEE Transactions on Consumer Electronics*, v. 57, n. 4, p. 1883–1890, November 2011. ISSN 0098-3063.
- PHAM, H.-H.; LE, T.-L.; VUILLERME, N. Real-time obstacle detection system in indoor environment for the visually impaired using microsoft kinect sensor. *Journal of Sensors*, v. 2016, 2016.
- SALERNO, J. J. Where's level 2/3 fusion - a look back over the past 10 years. In: *2007 10th International Conference on Information Fusion*. [S.l.: s.n.], 2007. p. 1–4.
- SAPUTRA, M. R. U.; WIDYAWANKA, W.; SANTOSA, P. I. Obstacle avoidance for visually impaired using auto-adaptive thresholding on kinect's depth image. In: *2014 IEEE 11th Intl Conf on Ubiquitous Intelligence and Computing and 2014 IEEE 11th Intl Conf on Autonomic and Trusted Computing and 2014 IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops*. [S.l.: s.n.], 2014. p. 337–342.
- SOYLU, A.; CAUSMAECKER, P.; DESMET, P. Context and adaptivity in pervasive computing environments: Links with software engineering and ontological engineering. *Journal of Software*, v. 4, n. 9, p. 992–1013, 2009.

- STAUFFER, C.; GRIMSON, W. E. L. Adaptive background mixture models for real-time tracking. In: *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on*. [S.l.: s.n.], 1999. v. 2, p. 252 Vol. 2. ISSN 1063-6919.
- TAMJIDI, A.; YE, C.; HONG, S. 6-dof pose estimation of a portable navigation aid for the visually impaired. In: *Robotic and Sensors Environments (ROSE), 2013 IEEE International Symposium on*. [S.l.: s.n.], 2013. p. 178–183.
- TANG, F. et al. An efficient classification approach for large-scale mobile ubiquitous computing. *Information Sciences*, v. 232, p. 419 – 436, 2013. ISSN 0020-0255.
- TAPU, R.; MOCANU, B.; ZAHARIA, T. A computer vision system that ensure the autonomous navigation of blind people. In: *E-Health and Bioengineering Conference (EHB), 2013*. [S.l.: s.n.], 2013. p. 1–4.
- TIAN, Y.; HAMEL, W. R.; TAN, J. Accurate human navigation using wearable monocular visual and inertial sensors. *IEEE Transactions on Instrumentation and Measurement*, v. 63, n. 1, p. 203–213, Jan 2014. ISSN 0018-9456.
- TRUCCO, E.; VERRI, A. *Introductory Techniques for 3D Computer Vision*. [S.l.]: Prentice Hall, 1998. ISBN 0132611082.
- TSIRMPAS, C. et al. An indoor navigation system for visually impaired and elderly people based on radio frequency identification (rfid). *Information Sciences*, v. 320, p. 288 – 305, 2015. ISSN 0020-0255.
- VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. [S.l.: s.n.], 2001. v. 1, p. I–511–I–518 vol.1. ISSN 1063-6919.
- WANG, Y. et al. Compressive background modeling for foreground extraction. *JECE*, Hindawi Publishing Corp., New York, NY, United States, v. 2015, jan. 2015. ISSN 2090-0147.
- XIAO, J. et al. An assistive navigation framework for the visually impaired. *IEEE Transactions on Human-Machine Systems*, v. 45, n. 5, p. 635–640, Oct 2015. ISSN 2168-2291.
- YANG, Y. et al. Moving object detection for dynamic background scenes based on spatiotemporal model. *Advances in Multimedia*, v. 2017, 2017.
- ZHU, Y.; SHYTYKH, R. Y.; JIN, Q. A human-centric framework for context-aware flowable services in cloud computing environments. *Information Sciences*, v. 257, p. 231 – 247, 2014. ISSN 0020-0255.



# Apêndice A

## SOBRE O TRABALHO

---

---

### A.1 Financiamento e Agradecimento

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior-Brasil (CAPES) - Código de Financiamento 001 e do Instituto Federal de São Paulo.

### A.2 Imagens

Todas as imagens usadas nesta tese são de autoria própria.

### A.3 Vídeos

Esta tese possui cinco vídeos que fornecem um melhor entendimento das técnicas aplicadas. Os links estão disponíveis a seguir:

1 <https://www.sciencedirect.com/science/article/pii/S0952197619300405>

**(Appendix A. Supplementary data)**

2 <https://www.sciencedirect.com/science/article/pii/S004579061830956X>

**(Supplementary Data S3)**

3 <https://www.sciencedirect.com/science/article/pii/S2352340919307589>

**Vídeo 1**

4 <https://www.youtube.com/watch?v=IUSdxCCLznk>

5 <https://www.youtube.com/watch?v=8wNyRZ3-Y-4>

## A.4 Publicações

[1] (Journal - Elsevier) CORDEIRO, N. H.; PEDRINO, E. C. **Collision risk prediction for visually impaired people using high level information fusion. Engineering Applications of Artificial Intelligence**, v. 81, p. 180 - 192, 2019. ISSN 0952-1976.

Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0952197619300405>>.

[2] (Journal - Elsevier) CORDEIRO, N. H.; PEDRINO, E. C. **A new methodology applied to dynamic object detection and tracking systems for visually impaired people. Computers & Electrical Engineering**, v. 77, p. 61-71, 2019. ISSN 0045-7906.

Disponível em: <<http://www.sciencedirect.com/science/article/pii/S004579061830956X>>.

[3] (Journal - Elsevier) CORDEIRO, N. H.; PEDRINO, E. C. **Data used for detection and tracking of dynamic objects for visually impaired people. Data in Brief**, v. 26, p. 104403, 2019. ISSN 2352-3409.

Disponível em: <<http://www.sciencedirect.com/science/article/pii/S2352340919307589>>.

[4] (Conference) CORDEIRO, N. H.; PEDRINO, E. C. **An architecture for collision risk prediction for visually impaired people. In: 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)**. Foz do iguaçu, Brasil. 2018. p. 305-312.

Disponível em: <<https://ieeexplore.ieee.org/document/8614343>>.

[5] (Conference) CORDEIRO, N. H. *et al.* **A data fusion architecture proposal for visually impaired people. In: 29th Conference on Graphics, Patterns and Images (SIBGRAPI)**, São Paulo, Brasil. 2016. p. 158-165.

Disponível em: <<https://ieeexplore.ieee.org/document/7813029>>.

## A.5 Código Fonte

### A.5.1 Objetos Estáticos

```

void objetosEstaticos(){

    // Alocação das matrizes (Imagens) que armazenam a sequência de frames
    Mat prevdepth, curdepth, prevdepth_gray;
    depthMat = Mat(480, 640, CV_8UC3, depth_front);
    prevdepth = depthMat.clone();
    cvtColor(prevdepth, prevdepth_gray, COLOR_BGR2GRAY);

    for(;;){

        curdepth = depthMat.clone();
        imshow("Depth", curdepth);
        erode(curdepth, curdepth, Mat(), Point(-1,-1), 5);
        erode(curdepth, curdepth, Mat(), Point(-1,-1), 5);
        dilate(curdepth, curdepth, Mat(), Point(-1,-1), 3);

        //Extração de Canais - Merge - Análise de distâncias por canais RGB
        Mat Zeross ;
        Zeross = Mat::zeros( prevdepth.size(), CV_8UC1 );
        vector<Mat> rgbChannels(3);
        split(curdepth, rgbChannels);

        {vector<Mat> channels;
        channels.push_back(rgbChannels[0]);
        channels.push_back(Zeross);
        channels.push_back(Zeross);
        merge(channels, BLUE);}

        ##### MUTEX Biblioteca Free Kinect #####
        pthread_mutex_lock(&gl_backbuf_mutex);
        if (current_format == FREENECT_VIDEO_YUV_RGB) {
            .
            .
            .
        }
        pthread_mutex_unlock(&gl_backbuf_mutex);
        ##### FIM MUTEX #####

        //Limiarização - Seleciona apenas as cores que representam distâncias menores
        que 2 metros (Objetos Estáticos)
        for(int iw=0;iw<640;iw++){
            for(int jw=0;jw<480;jw++){
                //curdepth.at<int>(iw,jw)=0;
                Vec3b intensityStatic = BLUE.at<Vec3b>(jw, iw);
                if (((intensityStatic.val[0]) > 130) && ((intensityStatic.val[0]) < 255))
                    Static.at<uchar>(jw,iw)=255;
                else
                    Static.at<uchar>(jw,iw)=0;
            }
        }

        //Método que produz o contorno dos Objetos Estáticos
        blur( Static, src_graySO, Size(3,3) );
        thresh_callbackSO( 0, 0 );

        int keycode = waitKey(30);

        //Troca de Frames
        cv::swap(prevdepth, curdepth);
    }
}

```

Figura A.1: Código Fonte - Detecção de Objetos Estáticos com menos de 2 metros de distância

## A.5.2 Passagem Livre

```

void passagemLivre(){
    // Alocação das matrizes (Imagens) que armazenam a sequência de frames
    Mat prevdepth, curdepth, prevdepth_gray;
    depthMat = Mat(480, 640, CV_8UC3, depth_front);
    prevdepth = depthMat.clone();
    cvtColor(prevdepth, prevdepth_gray, COLOR_BGR2GRAY);

    for(;;){
        curdepth = depthMat.clone();
        imshow("Depth", curdepth);
        erode(curdepth, curdepth, Mat(), Point(-1,-1), 5);
        erode(curdepth, curdepth, Mat(), Point(-1,-1), 5);
        dilate(curdepth, curdepth, Mat(), Point(-1,-1), 3);

        //Extração de Canais - Merge - Análise de distâncias por canais RGB
        Mat Zeross ;
        Zeross = Mat::zeros( prevdepth.size(), CV_8UC1 );
        vector<Mat> rgbChannels(3);
        split(curdepth, rgbChannels);

        {vector<Mat> channels;
        channels.push_back(Zeross);
        channels.push_back(Zeross);
        channels.push_back(rgbChannels[2]);
        merge(channels, RED);}

        ##### MUTEX Biblioteca Free Kinect #####
        pthread_mutex_lock(&gl_backbuf_mutex);
        if (current_format == FREENECT_VIDEO_YUV_RGB) {
            .
        }
        pthread_mutex_unlock(&gl_backbuf_mutex);
        ##### FIM MUTEX #####

        //Limiarização - Seleciona apenas as cores que representam distâncias maiores
        que 3 metros
        Mat Mat_erode, Mat_dilate;
        for(int iw=0;iw<640;iw++){
            for(int jw=0;jw<480;jw++){
                Vec3b intensityREDPassagem = curdepth.at<Vec3b>(jw, iw);
                if ((intensityREDPassagem.val[2]) > 30)
                    Passagem.at<uchar>(jw,iw)=255;
                else
                    Passagem.at<uchar>(jw,iw)=0;
            }
        }
        imshow("Passagem", Passagem);

        //Técnicas de Erosão e Dilatação (Morfologia Matemática) para melhorar o
        processo de contorno da passagem livre
        erode(Passagem, Mat_erode, Mat(), Point(-1,-1), 5);
        erode(Mat_erode, Mat_erode, Mat(), Point(-1,-1), 5);
        erode(Mat_erode, Mat_erode, Mat(), Point(-1,-1), 5);
        dilate(Mat_erode, Mat_dilate, Mat(), Point(-1,-1), 3);
        dilate(Mat_dilate, Mat_dilate, Mat(), Point(-1,-1), 3);
        blur( Mat_dilate, src_graySOPassage, Size(3,3) );

        //Método que produz o contorno da passagem livre
        thresh_callbackSOPassage( 0, 0 );
        imshow("MAPAXZ", MAPAXZ);
        int keycode = waitKey(30);

        //Troca de Frames
        cv::swap(prevdepth, curdepth);
    }
}

```

Figura A.2: Código Fonte - Detecção da Passagem Livre com mais de 3 metros de distância

### A.5.3 DROD - Arquitetura 1 - Reconhecimento de padrões

```

void objetoDinamicoA1PR(){
    struct DO{
        int x,y,X,Z;
    }ListaDOS;

    // Alocação das matrizes (Imagens) que armazenam a sequência de frames
    Mat rgbMat, previmg, prevdepth, curimg, curdepth;
    Mat previmg_gray, curimg_gray;

    rgbMat = Mat(480, 640, CV_8UC3, rgb_front);
    previmg = rgbMat.clone();
    depthMat = Mat(480, 640, CV_8UC3, depth_front);
    prevdepth = depthMat.clone();
    Mat frame;
    string inputName;
    bool tryflip;
    CascadeClassifier cascade, nestedCascade;
    double scale =1;

    //Modelo usado para classificação
    string cascadeName = "haarcascades/haarcascade_mcs_upperbody.xml";
    string nestedCascadeName = "haarcascades/haarcascade_frontalface_alt_tree.xml";

    if( !nestedCascade.load( nestedCascadeName ) ){
        cerr << "ERROR: Could not load classifier cascade" << endl;
        //return -1;
    }

    for(;;){
        curdepth = depthMat.clone();
        curimg = rgbMat.clone();
        curimg.copyTo(frame);

        ##### MUTEX Biblioteca Free Kinect #####
        pthread_mutex_lock(&gl_backbuf_mutex);
        if (current_format == FREENECT_VIDEO_YUV_RGB) {
            .
            .
            .
        }
        pthread_mutex_unlock(&gl_backbuf_mutex);
        ##### FIM MUTEX #####

        //Método que detecta o padrão de uma pessoa e realiza o contorno
        detectAndDraw( frame, cascade, nestedCascade, 1, 1 );

        //Método que cria o MAPA(X,Z)
        Mat MAPAXZ = Mat::zeros(4000,4000, CV_8UC3);
        line( MAPAXZ, Point(1000, 0), Point(5200, 4300), Scalar(255,255,255), 9, 8 ,0);
        line( MAPAXZ, Point(1000, 0), Point(900, 4000), Scalar(255,255,255), 9, 8 ,0);

        //Converte a posição detectada na imagem (CentrodeMassa(x,y)) para o MAPA(X,Z)
        ListaDOS.x = centrodeMassa.x;
        ListaDOS.y = centrodeMassa.y
        ListaDOS.Z=retornadistancia(ListaDOS.y, ListaDOS.x);
        ListaDOS.X = (ListaDOS.Z * 0.0015625) * ListaDOS.x;

        //Desenha a posição do objeto dinâmico no MAPA(X,Z)
        circle(MAPAXZ, Point(ListaDOS.X+1000, ListaDOS.Z), 10, Scalar(255, 255, 255), -1);

        cv::swap(previmg, curimg);
        cv::swap(prevdepth, curdepth);
    }
}

```

Figura A.3: Código Fonte - Detecção de Objetos Dinâmicos usando Reconhecimento de padrões

## A.5.4 DROD - Arquitetura 2 - Fluxo Óptico de Farneback

```

void objetoDinamicoA2FOF(){
    struct DO{
        int x,y,X,Z;
    }ListaDOS;

    // Alocação das matrizes (Imagens) que armazenam a sequência de frames
    Mat curimg, previmg, curimg_gray, previmg_gray, rgbMat, prevdepth, curdepth;
    Mat flow, cflow, frame;
    UMat gray, prevgray, uflow;

    for(;;){

        rgbMat = Mat(480, 640, CV_8UC3, rgb_front);
        curimg = rgbMat.clone();
        cvtColor(curimg, curimg_gray, COLOR_BGR2GRAY);

        depthMat = Mat(480, 640, CV_8UC3, depth_front);
        prevdepth = depthMat.clone();

        if(previmg_gray.empty())
            curimg_gray.copyTo(previmg_gray);
        if(prevdepth.empty())
            curdepth.copyTo(prevdepth);

        if( !previmg_gray.empty() ){
            calcOpticalFlowFarneback(previmg_gray, curimg_gray, uflow, 0.5, 3, 15, 3, 5, 1.2, 0);
            cvtColor(previmg_gray, cflow, COLOR_GRAY2BGR);
            uflow.copyTo(flow);
            //Método que detecta o objeto dinâmico mesmo com a câmera sofrendo
            movimento
            drawOptFlowMap(flow, cflow, 16, 1.5, Scalar(0, 0, 255));
            imshow("flow", cflow);
        }
        imshow("MAPAXZ", MAPAXZ);

        ##### MUTEX Biblioteca Free Kinect #####
        pthread_mutex_lock(&gl_backbuf_mutex);
        if (current_format == FREENECT_VIDEO_YUV_RGB) {
            .
            .
            .
        }
        pthread_mutex_unlock(&gl_backbuf_mutex);
        ##### FIM MUTEX #####

        //Método que cria o MAPA(X,Z)
        Mat MAPAXZ = Mat::zeros(4000,4000, CV_8UC3);
        line( MAPAXZ, Point(1000, 0), Point(5200, 4300), Scalar(255,255,255), 9, 8 ,0);
        line( MAPAXZ, Point(1000, 0), Point(900, 4000), Scalar(255,255,255), 9, 8 ,0);

        //Converte a posição detectada na imagem (CentrodeMassa(x,y)) para o
        MAPA(X,Z)
        ListaDOS.x = centrodeMassa.x;
        ListaDOS.y = centrodeMassa.y
        ListaDOS.Z=retornadistancia(ListaDOS.y, ListaDOS.x);
        ListaDOS.X = (ListaDOS.Z * 0.0015625) * ListaDOS.x;

        //Desenha a posição do objeto dinâmico no MAPA(X,Z)
        circle(MAPAXZ, Point(ListaDOS.X+1000, ListaDOS.Z), 10, Scalar(255, 255, 255), -1);

        cv::swap(previmg_gray, curimg_gray);
        cv::swap(prevdepth, curdepth);
    }
}

```

Figura A.4: Código Fonte - Detecção de Objetos Dinâmicos usando Fluxo Óptico de Farneback

```

static void drawOptFlowMap(const Mat& flow, Mat& cflowmap, int step, double, const Scalar& color){
    Point Pontos1DA, Pontos1DB;
    float pi = 3.1415926535897932384626433832;
    double somaangulos, mediaangulos, somadisteuclidiana, mediadisteuclidiana;
    somaangulos = 0;
    mediaangulos =0;
    somadisteuclidiana=0;
    mediadisteuclidiana=0;
    double matrixAngle[cflowmap.rows][cflowmap.cols];
    double matrixDistEuclidian[cflowmap.rows][cflowmap.cols];
    int matrixDistInicial[40][30], matrixDistFinal[40][30], matrixdiferenca[40][30];

    int contA=0, contB=0;
    for(int y = 0; y < cflowmap.rows; y += step){
        for(int x = 0; x < cflowmap.cols; x += step){
            const Point2f& fxy = flow.at<Point2f>(y, x);
            line(cflowmap, Point(x,y), Point(cvRound(x+fxy.x), cvRound(y+fxy.y)), color);
            Pontos1DA = Point(x,y)*1.f;
            Pontos1DB = Point(cvRound(x+fxy.x), cvRound(y+fxy.y))*1.f;
            float angle;
            angle = atan2( (float) Pontos1DA.y - Pontos1DB.y, (float) Pontos1DA.x - Pontos1DB.x );
            double disteuclidiana;
            disteuclidiana = sqrt(pow((Pontos1DA.y - Pontos1DB.y),2) + pow((Pontos1DA.x - Pontos1DB.x),2));
            float angleFinal;
            angleFinal = abs(angle) * (180/3.14159265359);
            somaangulos+=angleFinal;
            somadisteuclidiana+=disteuclidiana;
            matrixAngle[y][x] = angleFinal;
            matrixDistEuclidian[y][x] = disteuclidiana;
            if (matrixAngle[y][x] != 0){
                somaangulos+=abs(matrixAngle[y][x]);
                contA++;
            }
            if (matrixDistEuclidian[y][x] != 0){
                somadisteuclidiana+=abs(matrixDistEuclidian[y][x]);
                contB++;
            }
        }
    }
    mediaangulos = somaangulos/(contA);
    mediaangulos = somaangulos/(step*step);
    mediadisteuclidiana = somadisteuclidiana/(contB);
    drawing = Mat::zeros( flow.size(), CV_8UC1 );

    int cont =0;
    for(int y = 0; y < cflowmap.rows; y += step){
        for(int x = 0; x < cflowmap.cols; x += step){
            double cov1 = 0.0, cov2 = 0.0, cov3 = 0.0;
            cov1 += matrixAngle[y][x] * matrixAngle[y][x] / (40 - 1);
            cov2 += matrixDistEuclidian[y][x] * matrixDistEuclidian[y][x] / (*cflowmap.rows*/ 40 - 1);
            const Point2f& fxy = flow.at<Point2f>(y, x);
            Pontos1DA = Point(x,y)*1.f;
            Pontos1DB = Point(cvRound(x+fxy.x), cvRound(y+fxy.y))*1.f;
            double angle;
            angle = atan2( (double) Pontos1DA.y - Pontos1DB.y, (double) Pontos1DA.x - Pontos1DB.x );

            double disteuclidiana;
            disteuclidiana = sqrt(pow((Pontos1DA.y - Pontos1DB.y),2) + pow((Pontos1DA.x - Pontos1DB.x),2));
            if (abs(disteuclidiana) > (mediadisteuclidiana /*add limiar*/))
                if ((angle > mediaangulos) || (angle < mediaangulos /*add limiar*/))
                    if (cov1 > 2.5)
                        if(cov2 > 0.5){
                            circle(cflowmap, Point(cvRound(x+fxy.x), cvRound(y+fxy.y)), 3, Scalar(255, 0, 0), -1);
                            circle(drawing, Point(cvRound(x+fxy.x), cvRound(y+fxy.y)), 10, Scalar(255, 0, 0), -1);
                            thresh_callbackDO( 0, 0 );
                        }
        }
    }
    imshow("drawing",drawing);
}

```

**Figura A.5: Código Fonte - Método (drawOptFlowMap) que detecta objeto dinâmico mesmo com a presença de movimento na câmera.**

### A.5.5 DROD - Arquitetura 3 - Subtração de *Background*

```

void objetoDinamicoA3BS(){
    struct DO{
        int x,y,X,Z;
    }ListaDOS;

    // Alocação das matrizes (Imagens) que armazenam a sequência de frames
    Mat prevdepth, curdepth, prevdepth_gray;
    depthMat = Mat(480, 640, CV_8UC3, depth_front);
    prevdepth = depthMat.clone();

    //Inicialização das variáveis para Subtração de Background
    bool update_bg_model = true;
    Mat tmp_frame, bgmask;
    namedWindow("video", 1);
    namedWindow("segmented", 1);

    Ptr<BackgroundSubtractorMOG2> bgsubtractor=createBackgroundSubtractorMOG2();
    bgsubtractor->setVarThreshold(150);

    for(;;){
        curdepth = depthMat.clone();
        Mat Zeross ;
        Zeross = Mat::zeros( prevdepth.size(), CV_8UC1 );

        ##### MUTEX Biblioteca Free Kinect #####
        pthread_mutex_lock(&gl_backbuf_mutex);
        if (current_format == FREENECT_VIDEO_YUV_RGB) {
            .
            .
            .
        }
        pthread_mutex_unlock(&gl_backbuf_mutex);
        ##### FIM MUTEX #####

        curdepth.copyTo(tmp_frame);
        if( tmp_frame.empty() )
            break;

        //Método que detecta o objeto dinâmico pela diferença entre os frames
        bgsubtractor->apply(tmp_frame, bgmask, update_bg_model ? -1 : 0);
        refineSegments(tmp_frame, bgmask, out_frame);
        cvtColor(out_frame, out_frame, COLOR_BGR2GRAY);

        //Método que produz o contorno do objeto dinâmico
        thresh_callbackDOS( 0, 0 );

        //Método que cria o MAPA(X,Z)
        Mat MAPAXZ = Mat::zeros(4000,4000, CV_8UC3);
        line( MAPAXZ, Point(1000, 0), Point(5200, 4300), Scalar(255,255,255), 9, 8 ,0);
        line( MAPAXZ, Point(1000, 0), Point(900, 4000), Scalar(255,255,255), 9, 8 ,0);

        //Converte a posição detectada na imagem (CentrodeMassa(x,y)) para o MAPA(X,Z)
        ListaDOS.x = centrodeMassa.x;
        ListaDOS.y = centrodeMassa.y;
        ListaDOS.Z=retornadistancia(ListaDOS.y, ListaDOS.x);
        ListaDOS.X = (ListaDOS.Z * 0.0015625) * ListaDOS.x;

        //Desenha a posição do objeto dinâmico no MAPA(X,Z)
        circle(MAPAXZ, Point(ListaDOS.X+1000, ListaDOS.Z), 10, Scalar(255, 255, 255), -1);

        //Troca de Frames
        cv::swap(prevdepth, curdepth);
    }
}

```

Figura A.6: Código Fonte - Detecção de Objetos Dinâmicos usando Subtração de *Background*



## A.5.6 DROD - Arquitetura 4 - Continuously Adaptive Meanshift

```

void objetoDinamicoA4CAM(){
    struct DO{
        int x,y,X,Z;
    }ListaDOS;
    // Alocação das matrizes (Imagens) que armazenam a sequência de frames
    Mat cur, prev, cur_gray, prev_gray, rgbMat, previmg, prevdepth, curimg, curdepth;
    Mat previmg_gray, curimg_gray;
    rgbMat = Mat(480, 640, CV_8UC3, rgb_front);
    previmg = rgbMat.clone();
    cvtColor(previmg, previmg_gray, COLOR_BGR2GRAY);
    depthMat = Mat(480, 640, CV_8UC3, depth_front);
    prevdepth = depthMat.clone();
    Mat flow, cflow, frame, previmg_gray2, curimg_gray2;
    UMat gray, prevgray, uflow;

    Rect trackWindow;
    int hsize = 16;
    float hranges[] = {0,180};
    const float* phranges = hranges;
    namedWindow( "CamShift Demo", 0 );
    setMouseCallback( "CamShift Demo", onMouse, 0 );
    createTrackbar( "Vmin", "CamShift Demo", &vmin, 256, 0 );
    createTrackbar( "Vmax", "CamShift Demo", &vmax, 256, 0 );
    createTrackbar( "Smin", "CamShift Demo", &smin, 256, 0 );
    Mat hsv, hue, mask, hist, histimg = Mat::zeros(200, 320, CV_8UC3), backproj;
    bool paused = false;
    for(;;){
        curdepth = depthMat.clone();
        rgbMat = Mat(480, 640, CV_8UC3, rgb_front);
        curimg = rgbMat.clone();
        cvtColor(curimg, curimg_gray, COLOR_BGR2GRAY);

        curimg.copyTo(image);
        CentrodeMassa = Mat::zeros(480, 640, CV_8UC1);
        if( !paused ){
            cvtColor(image, hsv, COLOR_BGR2HSV);
            if( trackObject ){
                int _vmin = vmin, _vmax = vmax;
                inRange(hsv, Scalar(0, smin, MIN(_vmin, _vmax)),
                    Scalar(180, 256, MAX(_vmin, _vmax)), mask);
                int ch[] = {0, 0};
                hue.create(hsv.size(), hsv.depth());
                mixChannels(&hsv, 1, &hue, 1, ch, 1);
                if( trackObject < 0 ){
                    Mat roi(hue, selection), maskroi(mask, selection);
                    calcHist(&roi, 1, 0, maskroi, hist, 1, &hsize, &phranges);
                    normalize(hist, hist, 0, 255, NORM_MINMAX);
                    trackWindow = selection;
                    trackObject = 1;
                    histimg = Scalar::all(0);
                    int binW = histimg.cols / hsize;
                    Mat buf(1, hsize, CV_8UC3);
                    for( int i = 0; i < hsize; i++ )
                        buf.at<Vec3b>(i) = Vec3b(saturate_cast<uchar>(i*180./hsize), 255, 255);
                    cvtColor(buf, buf, COLOR_HSV2BGR);
                }

                calcBackProject(&hue, 1, 0, hist, backproj, &phranges);
                backproj &= mask;
                RotatedRect trackBox = CamShift(backproj, trackWindow,
                    TermCriteria( TermCriteria::EPS | TermCriteria::COUNT, 10, 1 ));
                if( trackWindow.area() <= 1 ){
                    int cols = backproj.cols, rows = backproj.rows, r = (MIN(cols, rows) + 5)/6;
                    trackWindow = Rect(trackWindow.x - r, trackWindow.y - r, trackWindow.x + r,
                        trackWindow.y + r) & Rect(0, 0, cols, rows);
                }
            }
        }
    }
}

```

Figura A.7: Código Fonte - Detecção de Objetos Dinâmicos usando CamShift - Parte 1

```

if( backprojMode)
    cvtColor( backproj, image, COLOR_GRAY2BGR );
    ellipse( image, trackBox, Scalar(0,0,255), 3, LINE_AA );
    ellipse( CentrodeMassa, trackBox, Scalar(255,255,255), 3, LINE_AA );
    thresh_callbackDOS( 0, 0 );
}
else if( trackObject < 0 )
    paused = false;
if( selectObject && selection.width > 0 && selection.height > 0 ){
    Mat roi(image, selection);
    bitwise_not(roi, roi);
}

##### MUTEX Biblioteca Free Kinect #####
pthread_mutex_lock(&gl_backbuf_mutex);
if( current_format == FREENECT_VIDEO_YUV_RGB ) {
    .
    .
    .
}
pthread_mutex_unlock(&gl_backbuf_mutex);
##### FIM MUTEX #####

if(previmg_gray.empty())
    curimg_gray.copyTo(previmg_gray);
if( !previmg_gray.empty() ){
    calcOpticalFlowFarneback(previmg_gray, curimg_gray, uflow, 0.5, 3, 15, 3, 5, 1.2, 0);
    cvtColor(previmg_gray, cflow, COLOR_GRAY2BGR);
    uflow.copyTo(flow);
    drawOptFlowMap(flow, cflow, 16, 1.5, Scalar(0, 0, 255));
    imshow("flow", cflow);
}

//Método que detecta o objeto dinâmico pela diferença entre os frames
bgsubtractor->apply(tmp_frame, bgmask, update_bg_model ? -1 : 0);
refineSegments(tmp_frame, bgmask, out_frame);
cvtColor(out_frame, out_frame, COLOR_BGR2GRAY);

//Método que produz o contorno do objeto dinâmico
thresh_callbackDOS( 0, 0 );

//Método que cria o MAPA(X,Z)
Mat MAPAXZ = Mat::zeros(4000,4000, CV_8UC3);
line( MAPAXZ, Point(1000, 0), Point(5200, 4300), Scalar(255,255,255), 9, 8 ,0);
line( MAPAXZ, Point(1000, 0), Point(900, 4000), Scalar(255,255,255), 9, 8 ,0);

//Converte a posição detectada na imagem (CentrodeMassa(x,y)) para o MAPA(X,Z)
ListaDOS.x = centrodeMassa.x;
ListaDOS.y = centrodeMassa.y;
ListaDOS.Z=retornadistancia(ListaDOS.y, ListaDOS.x);
ListaDOS.X = (ListaDOS.Z * 0.0015625) * ListaDOS.x;

//Desenha a posição do objeto dinâmico no MAPA(X,Z)
circle(MAPAXZ, Point(ListaDOS.X+1000, ListaDOS.Z), 10, Scalar(255, 255, 255), -1);

imshow("MAPAXZ", MAPAXZ);

//Troca de Frames
cv::swap(previmg_gray, curimg_gray);
cv::swap(previmg, curimg);
cv::swap(prevdepth, curdepth);
}
}

```

**Figura A.8: Código Fonte - Detecção de Objetos Dinâmicos usando CamShift - Parte 2**

Os códigos completos, incluindo as bibliotecas, *makefiles* e os arquivos binários estão no seguinte repositório: <<https://github.com/natalhenrique/SAS-DV>>

Além dos códigos-fontes, neste mesmo repositório estão as instruções para configurar as bibliotecas necessárias para compilação do código-fonte e execução do arquivo binário. link: <[https://github.com/natalhenrique/SAS-DV/blob/master/Setup\\_instruction-StepByStep.pdf](https://github.com/natalhenrique/SAS-DV/blob/master/Setup_instruction-StepByStep.pdf)>