

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**PROTÓTIPO VESTÍVEL DE SISTEMA DE  
NAVEGAÇÃO MULTIMODAL E INTERATIVO  
PARA AUXÍLIO CUSTOMIZÁVEL A PESSOAS  
COM DEFICIÊNCIAS VISUAIS**

**ANTONIO MIGUEL BATISTA DOURADO**

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

São Carlos – SP

Março/2020

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**PROTÓTIPO VESTÍVEL DE SISTEMA DE  
NAVEGAÇÃO MULTIMODAL E INTERATIVO  
PARA AUXÍLIO CUSTOMIZÁVEL A PESSOAS  
COM DEFICIÊNCIAS VISUAIS**

**ANTONIO MIGUEL BATISTA DOURADO**

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

Março/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 Antonio Miguel Batista Dourado, realizada em 09/03/2020:

---

Prof. Dr. Emerson Carlos Pedrino  
UFSCar

---

Prof. Dr. José Hiroki Saito  
UFSCar

---

Prof. Dr. Roberto Santos Inoue  
UFSCar

---

Prof. Dr. Valter Obac Roda  
UFRN

---

Profa. Dra. Andréa Poletto Sonza  
IFRS - Porto Alegre

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

---

Prof. Dr. Emerson Carlos Pedrino

Dedico este trabalho a meus pais, minha noiva e meus amigos.

## AGRADECIMENTOS

Agradeço a Deus, por ter me permitido chegar até aqui em meio às dificuldades enfrentadas em diversos períodos da minha vida e me dado forças para sempre continuar.

Aos meus pais, minha eterna gratidão por sempre estarem ao meu lado e preocupados comigo, mesmo nas vezes em que eu tentei poupá-los das minhas próprias dores. Ser o orgulho de vocês foi e é, por muitas vezes, a motivação para continuar me tornando uma pessoa cada vez melhor.

Agradeço minha noiva, por ter me acompanhado e apoiado durante os momentos mais difíceis deste trabalho, ouvido-me bravo ou triste, limpado minhas lágrimas e ter sido compreensiva nos momentos em que precisei estar ausente. Obrigado por compreender meus medos e anseios.

Agradeço ao meu orientador, por todo conhecimento adquirido durante esses cinco anos de convivência e pela oportunidade de poder ajudar as pessoas por meio deste trabalho. Agradeço todas as dicas dadas e críticas feitas durante o trabalho, que farão parte de mim enquanto professor e pesquisador.

À ONG PARA-DV, da cidade de Araraquara-SP, eu agradeço por me receber de portas abertas e pelos valiosos ensinamentos sobre a deficiência visual. Agradeço a Maria Helena e o Alex por toda a gentileza durante as dezenas de visitas que fiz à ONG e agradeço também todos os membros que participaram, direta ou indiretamente, deste trabalho. As contribuições de vocês foram essenciais para este trabalho e despertaram em mim uma nova motivação acadêmica para ajudar a sociedade.

Agradeço aos meus amigos, pelo apoio e conselhos durante todo o trabalho, em especial a Amanda Fabron, Leonardo Botega, Juliana Rizzi e André Tarallo. Vocês contribuíram, de diferentes maneiras, para a realização deste trabalho. Agradeço também o Renan Alboy, por ter me ajudado nas reuniões finais na ONG PARA-DV.

Agradeço o Instituto Federal de São Paulo, por me conceder o afastamento remunerado para capacitação, permitindo que eu me dedicasse a uma área de pesquisa totalmente nova.

Como agradecimento, vou procurar trazer toda a minha experiência adquirida para dentro da instituição e fortalecer seu nome.

Por fim, agradeço todos aqueles que contribuíram de uma forma ou outra com este trabalho e não foram citados aqui. Obrigado.

*Os sonhadores podem morrer, mas os sonhos continuam vivos.*

Bruce Dickinson

## RESUMO

A deficiência visual atinge milhões de pessoas em todo o mundo atualmente. Não podendo contar com a visão, as pessoas utilizam outros sentidos para compensar a falta da capacidade de enxergar, como a audição e o tato, na realização de tarefas cotidianas. A substituição sensorial é oferecida como uma alternativa para auxiliar pessoas com perda de visão, por meio de dispositivos capazes de reconhecer o ambiente, obstáculos e caminhos para então fornecer respostas que uma pessoa com deficiência visual possa entender. Entretanto, a literatura contém trabalhos que apresentam diversos sistemas de navegação que estão muito distantes de sua adoção e, principalmente, sem grandes preocupações com a individualidade de preferências de cada usuário ou até mesmo suas necessidades específicas. Frente a esses problemas, o objetivo geral desta tese é fornecer uma alternativa para auxílio customizável à navegação autônoma de pessoas com deficiências visuais, desde a concepção de um sistema de navegação até sua montagem na forma de um protótipo. Para tanto, nesta tese foi desenvolvido um modelo abstrato para sistemas de navegação interativos e customizáveis com o intuito de servir como base para o projeto e desenvolvimento desse tipo de sistema. Foi também criada uma metodologia para customização de recursos funcionais de sistemas de navegação, com o intuito de oferecer uma maneira de especificar como é possível customizar cada recurso do sistema. O modelo abstrato e a metodologia para customização resultaram na criação do protótipo vestível de sistema de navegação multimodal, interativo e customizável. Esta tese também inclui o desenvolvimento do NSGA2CGP, um método de otimização multiobjetivo evolucionário e cartesiano para geração automática de filtros morfológicos otimizados, minimizando o erro e a complexidade, para corrigir imagens de profundidade de câmeras RGB-D que apresentam áreas de distâncias desconhecidas e afetam o processo de segmentação de sistemas de navegação baseados em visão computacional. O método se mostrou sólido e capaz de gerar filtros otimizados para a correção de imagens de profundidade com boa diversidade. O protótipo vestível com o sistema embarcado de navegação foi validado e posteriormente avaliado em experimentos com sete membros da ONG PARA-DV, da cidade de Araraquara/SP, que utilizaram o protótipo em ambientes reais, como as ruas da cidade e o Departamento de Computação da Universidade Federal de São Carlos, ambiente desconhecido pelos membros da ONG. O resultado foi a aprovação do protótipo, com indicativos importantes para o prosseguimento do trabalho com novos recursos.

**Palavras-chave:** Deficiência Visual, Visão Computacional, Sistemas Embarcados, Sistemas de Navegação, Tecnologia Assistiva, Substituição Sensorial

# ABSTRACT

Visual impairment affects millions of people around the world nowadays. By not being able to rely on their vision, people often use other senses to compensate for the lack of vision, such as hearing and touch, when performing daily tasks. Sensory substitution is offered as an alternative to assist people with vision loss, through devices capable of recognizing the environment, obstacles, paths and then providing feedbacks that a visually impaired person can understand. However, the literature contains works that present navigation systems that are very distant from their adoption and without great concerns regarding the individuality of each user's preferences or even their specific needs/limitations. Based on these problems, the main goal of this thesis is to provide an alternative to autonomous and customizable navigation of visually impaired people, from the conception of a navigation system to its build as a functional prototype. Hence, in this thesis, it was developed an abstract model for interactive and customizable navigation systems to provide a baseline for the design and development of this type of system. This work also created a customization methodology for functional features of navigation systems, to offer a way to specify how it is possible to customize each feature of a customizable system. The abstract model and customization methodology resulted in a wearable prototype of a multimodal, interactive and customizable navigation system. This thesis also proposes the NSGA2CGP, an evolutionary and Cartesian multi-objective optimization method for automatic generation of optimized morphological filters, minimizing error and complexity to correct depth images from RGB-D cameras with areas of unknown distances that affect the segmentation process of vision-based navigation systems. NSGA2CGP method proved to be solid and capable of generating optimized filters to fix depth images with good diversity. The wearable prototype with the embedded navigation system was validated and then evaluated in experiments with members of the NGO PARA-DV, from the city of Araraquara / SP, that used the prototype in real environments such as the city streets and the Computing Department of Federal University of São Carlos, a total unknown environment to the members of the NGO. The result of the evaluation was the approval of the prototype, with important indications for the next steps of this work with new resources.

**Keywords:** Visual Impairment, Computer Vision, Embedded Systems, Navigation Systems, Assistive Technology, Sensory Substitution

## LISTA DE FIGURAS

2.1	Estrutura Simplificada de um Sistema de Navegação. . . . .	35
2.2	Microsoft Kinect for XBox 360 (esquerda) e ZED Stereo Camera (direita) (MICROSOFT, 2017; STEREO LABS, 2017). . . . .	36
2.3	Dispositivos de <i>feedback</i> tátil. . . . .	37
2.4	Exemplo de segmentação utilizando limiarização (Chen, Guo e SUN (2010)). . . . .	40
2.5	Arquitetura de uma rede neural convolucional (Adaptado de Lecun et al. (1998)). . . . .	42
2.6	Exemplo de cromossomo cartesiano. . . . .	43
2.7	Ciclo do algoritmo NSGAI (Adaptado de Deb et al. (2002)). . . . .	45
4.1	Exemplos de imagens de profundidade em escala de cinza. . . . .	56
4.2	Representação da estrutura do método NSGA2CGP. . . . .	58
4.3	Representação do fenótipo e genótipo de um cromossomo cartesiano. . . . .	60
4.4	Resultados de treinamento exibidos pela fase de Apresentação. . . . .	64
5.1	Modelo Abstrato de Sistemas de Navegação Interativos e Customizáveis. . . . .	68
5.2	Estrutura Resumida da Customização de Recursos (Geral, Interadores e <i>Feedbacks</i> ). . . . .	70
5.3	Exemplo de organização de especificação de recursos customizáveis. . . . .	74
5.4	Diagrama Geral do "Sistema Embarcado Multimodal e Interativo para Auxílio Customizável à Navegação de Pessoas com Deficiências Visuais". O sistema implementa todos os módulos e recursos auxiliares previstos no modelo abstrato da Seção 5.1. . . . .	75

5.5	Teclado braile para interação com o protótipo e padrão de comandos definidos pelos membros da ONG PARA-DV. . . . .	77
5.6	Ilustração sobre o processo de segmentação de caminhos livres. . . . .	84
5.7	Exemplo de Detecção de Obstáculo sobre o Caminho Livre. . . . .	85
5.8	Primeira versão do colete. Foi vestido em uma pessoa sem perda significativa de visão apenas para fins demonstrativos. . . . .	91
5.9	Equipamentos utilizados na primeira versão do protótipo: (a) Baterias, (b) NVIDIA Jetson TX1, (c) Kinect XBOX 360 e (d) <i>headset</i> de condução óssea. . . .	92
5.10	Segunda versão do protótipo vestível em uma pessoa com deficiência visual. . .	93
5.11	Modelo final do protótipo vestível confeccionado exclusivamente para esta tese. . .	94
5.12	Posicionamento aproximado dos motores de vibração em relação ao corpo humano para estímulo via <i>feedback</i> tátil. . . . .	95
5.13	Equipamentos utilizados na segunda versão do protótipo. . . . .	97
6.1	Planta Baixa do Piso Superior do Departamento de Computação da UFSCar. . .	99
6.2	Planta Baixa do Piso Inferior do Departamento de Computação da UFSCar. . .	100
7.1	Exemplos de resultados de processamento do protótipo vestível com sistema embarcado. . . . .	109
7.2	Protótipo indicando direções seguras ao cruzar com pessoas nas calçadas. Imagens capturadas pelo próprio protótipo. . . . .	111
7.3	Protótipo indicando caminhos para evitar obstáculos perigosos segundo relatos de PDVs. Imagens capturadas pelo próprio protótipo. . . . .	112
7.4	Médias das respostas de cada participante sobre as questões de customização. .	113
7.5	Médias das respostas de cada participante sobre as questões de métodos de <i>feedback</i> . . . . .	114
7.6	Médias das respostas de cada participante sobre as questões referentes à ergonomia. . . . .	115
7.7	Média individual de cada questão aplicada com o formulário. . . . .	116

7.8	Participantes da ONG PARA-DV de Araraquara/SP realizando experimento do Departamento de Computação da UFSCar. . . . .	117
7.9	Participantes do experimento lidando com obstáculos no Departamento de Computação da UFSCar. . . . .	118
7.10	Imagem de entrada e respectivas saídas das soluções factíveis do <i>setup</i> S4. Áreas em branco destacam as distâncias desconhecidas que não foram corrigidas. . . . .	121
7.11	Resultados de desempenho dos conjuntos S1, S2, S3 e S4 de acordo com seus Hipervolumes ( <i>a-d</i> ) e IGDs( <i>e-h</i> ). . . . .	125

## LISTA DE TABELAS

3.1	Técnicas empregadas nos trabalhos da literatura. . . . .	53
6.1	Parâmetros utilizados. . . . .	103
7.1	Percentuais de probabilidade dos obstáculos classificados. . . . .	110
7.2	Resultado dos treinamentos para cada função de erro. . . . .	120
7.3	Consumo de energia do filtros morfológicos gerados pelo NSGA2CGP e da literatura, <i>Inpainting</i> e RMF, em miliwatts (mW). . . . .	122
7.4	Tempo de execução dos filtros otimizados e da literatura, <i>Inpainting</i> e RMF, em milissegundos (ms). . . . .	123
E.1	Equipamentos, valores e sites. . . . .	150

## LISTA DE QUADROS

1.1	Informações gerais sobre os membros participantes da ONG PARA-DV. . . . .	29
4.1	Parâmetros de Execução do NSGA2CGP. . . . .	59
4.2	Modos de operação do tipo de gene <i>Free</i> . . . . .	61
5.1	Recursos Customizáveis do Sistema. . . . .	76
7.1	Principais parâmetros ou <i>setups</i> de treinamento dos quatro melhores resultados do NSGA2CGP. . . . .	119
7.2	Solução factível escolhida do conjunto S4 treinado com a função de erro $f_5$ . Nó é uma referência à identificação dos nós usados* e operação é a tarefa do nó. . .	125

## LISTA DE ALGORITMOS

1	Fase de Inicialização do NSGA2CGP . . . . .	59
2	Fase de Evolução do Método NSGA2CGP. . . . .	62
3	Fase de Apresentação do Método NSGA2CGP . . . . .	63
4	Fase de Visualização do Método NSGA2CGP . . . . .	65
5	Fase Exportar e Implantar do Método NSGA2CGP . . . . .	65
6	Módulo de Customização . . . . .	79
7	Módulo de Carregamento do Sistema . . . . .	80
8	Módulo de Aquisição de Dados . . . . .	81
9	Módulo de Pré-Processamento de Dados . . . . .	82
10	Módulo de Extração de Informações . . . . .	83
11	Módulo de Tradução de Informações para <i>Feedbacks</i> . . . . .	87
12	Módulo de Despacho de <i>Feedback</i> . . . . .	89

# SUMÁRIO

<b>CAPÍTULO 1 – INTRODUÇÃO</b>	<b>21</b>
1.1 Contexto e Motivações . . . . .	21
1.2 Questões de Pesquisa . . . . .	24
1.3 Hipóteses . . . . .	25
1.4 Objetivos . . . . .	26
1.5 Contribuições da Tese . . . . .	26
1.6 Metodologia de Pesquisa . . . . .	27
1.6.1 Natureza e Abordagem da Pesquisa . . . . .	27
1.6.2 Método de Pesquisa e Instrumentos de Coleta de Dados . . . . .	28
1.6.3 Método de Amostragem e Seleção . . . . .	28
1.6.4 Análise dos Dados . . . . .	29
1.6.5 Atividades Metodológicas . . . . .	29
1.6.5.1 Análise do Estado da Arte . . . . .	30
1.6.5.2 Método de Otimização Multiobjetivo Evolucionário Cartesiano para Correção de Imagens de Profundidade . . . . .	30
1.6.5.3 Metodologia para Customização de Sistemas de Navegação para PDVs . . . . .	30
1.6.5.4 Desenvolvimento de Sistema de Navegação para Pessoas com Deficiência Visual . . . . .	31
1.6.5.5 Implantação de Sistema de Navegação em Hardware Embarcado e Vestível . . . . .	31

1.6.5.6	Validação e Avaliação de Protótipo . . . . .	31
1.7	Organização do Texto . . . . .	32
<b>CAPÍTULO 2 – FUNDAMENTAÇÃO TEÓRICA DE SISTEMAS DE NAVEGAÇÃO PARA AUXÍLIO A PESSOAS COM DEFICIÊNCIAS VISUAIS</b>		<b>34</b>
2.1	Sistemas de Navegação . . . . .	34
2.1.1	Sensores de Entrada . . . . .	35
2.1.1.1	Câmeras . . . . .	35
2.1.1.2	IMUs . . . . .	36
2.1.1.3	Outros Sensores . . . . .	36
2.1.2	Tipos de <i>Feedback</i> . . . . .	37
2.1.2.1	<i>Feedback</i> Tátil . . . . .	37
2.1.2.2	<i>Feedback</i> Acústico ou Sonoro . . . . .	38
2.1.3	Pré-Processamento . . . . .	38
2.1.3.1	Morfologia Matemática . . . . .	39
2.1.4	Segmentação . . . . .	39
2.1.4.1	<i>Floodfill</i> . . . . .	39
2.1.5	Classificação . . . . .	40
2.1.5.1	Redes Neurais Convolucionais . . . . .	41
2.2	Otimização Multiobjetivo e Programação Genética . . . . .	42
2.2.1	Programação Genética . . . . .	42
2.2.1.1	Programação Genética Cartesiana . . . . .	43
2.2.2	Otimização Multiobjetivo . . . . .	44
2.2.2.1	Algoritmo Genético de Ordenamento Não dominado – NSGAI . . . . .	44
2.3	Considerações Finais . . . . .	45
<b>CAPÍTULO 3 – ESTADO DA ARTE EM SISTEMAS DE NAVEGAÇÃO PARA AU- XÍLIO A PESSOAS COM DEFICIÊNCIAS VISUAIS</b>		<b>47</b>

3.1	Sistemas de Navegação e Classificação para PDVs . . . . .	47
3.2	Considerações Finais . . . . .	54
<b>CAPÍTULO 4 – MÉTODO DE OTIMIZAÇÃO MULTIOBJETIVO EVOLUCIONÁRIO E CARTESIANO PARA CORREÇÃO DE IMAGENS DE PROFUNDIDADE</b>		<b>55</b>
4.1	Imagens de Profundidade . . . . .	55
4.2	Visão Geral . . . . .	57
4.3	Inicialização . . . . .	57
4.4	Evolução . . . . .	61
4.5	Apresentação . . . . .	63
4.6	Visualização . . . . .	64
4.7	Exportar e Implantar . . . . .	65
4.8	Considerações Finais . . . . .	66
<b>CAPÍTULO 5 – PROTÓTIPO VESTÍVEL COM SISTEMA DE NAVEGAÇÃO MULTIMODAL E INTERATIVO PARA AUXÍLIO CUSTOMIZÁVEL A PESSOAS COM DEFICIÊNCIAS VISUAIS</b>		<b>67</b>
5.1	Modelo Abstrato de Sistemas de Navegação Interativos e Customizáveis . . . . .	67
5.2	Metodologia para Customização de Sistemas de Navegação para PDVs . . . . .	69
5.2.1	Geral . . . . .	71
5.2.2	Interadores . . . . .	71
5.2.3	<i>Feedbacks</i> . . . . .	72
5.3	Sistema Embarcado Multimodal e Interativo para Auxílio Customizável à Navegação de PDVs . . . . .	74
5.3.1	Preferências de Usuário . . . . .	75
5.3.2	Interadores . . . . .	77
5.3.3	Módulo de Customização . . . . .	78
5.3.4	Módulo de Carregamento do Sistema . . . . .	80

5.3.5	Módulo de Aquisição de Dados . . . . .	81
5.3.6	Módulo de Pré-processamento de Dados . . . . .	81
5.3.7	Módulo de Extração de Informações . . . . .	82
5.3.7.1	Detecção de Caminhos Livres . . . . .	84
5.3.7.2	Detecção de Obstáculos . . . . .	85
5.3.7.3	Classificação e Distância de Obstáculos . . . . .	86
5.3.8	Módulo de Tradução de Informações para <i>Feedbacks</i> . . . . .	87
5.3.9	Módulo de Despacho de <i>Feedback</i> . . . . .	89
5.4	Protótipo Vestível . . . . .	90
5.4.1	Primeira Versão: Colete . . . . .	90
5.4.2	Segunda Versão: Mochila . . . . .	92
5.4.2.1	<i>Feedback</i> Tátil . . . . .	94
5.4.2.2	Equipamentos . . . . .	95
5.5	Considerações Finais . . . . .	96
<b>CAPÍTULO 6 – EXPERIMENTOS</b>		<b>98</b>
6.1	Validação e Avaliação do Protótipo Vestível de Auxílio a PDVs . . . . .	98
6.1.1	Validação . . . . .	98
6.1.2	Avaliação . . . . .	99
6.1.2.1	Formulário de Avaliação . . . . .	101
6.2	Método NSGA2CGP . . . . .	102
6.2.1	Parâmetros de Execução . . . . .	102
6.2.2	Funções de Objetivo . . . . .	104
6.2.3	Outras Abordagens e Métricas de Performance . . . . .	105
6.3	Considerações Finais . . . . .	106
<b>CAPÍTULO 7 – RESULTADOS E DISCUSSÕES</b>		<b>108</b>

7.1	Protótipo Vestível para Auxílio a PDVs . . . . .	108
7.1.1	Validação . . . . .	108
7.1.2	Avaliação . . . . .	110
7.2	NSGA2CGP . . . . .	119
7.3	Considerações Finais . . . . .	126
<b>CAPÍTULO 8 – CONCLUSÕES</b>		<b>127</b>
8.1	Atendimento às proposições de pesquisa . . . . .	128
8.2	Limitações e Trabalhos Futuros . . . . .	130
8.3	Produção Científica . . . . .	131
<b>REFERÊNCIAS</b>		<b>133</b>
<b>APÊNDICE A – FORMULÁRIO DE PESQUISA</b>		<b>141</b>
<b>APÊNDICE B – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO</b>		<b>143</b>
<b>APÊNDICE C – EXEMPLO DE ESPECIFICAÇÃO DE RECURSO</b>		<b>145</b>
<b>APÊNDICE D – FILTRO MORFOLÓGICO GERADO PELO NSGA2CGP</b>		<b>148</b>
<b>APÊNDICE E – PREPARAÇÃO E CONSTRUÇÃO DO SISTEMA</b>		<b>150</b>
E.1	Preparação do Ambiente . . . . .	151
E.1.1	NVIDIA Jetson Nano . . . . .	151
E.1.2	Bibliotecas . . . . .	151
E.2	Código-fonte e compilação . . . . .	153
<b>GLOSSÁRIO</b>		<b>155</b>

# Capítulo 1

## INTRODUÇÃO

---

---

### 1.1 Contexto e Motivações

Em um mundo cada vez mais interconectado e globalizado, milhões de pessoas ainda enfrentam dificuldades na realização de simples tarefas cotidianas, como se locomover em um ambiente qualquer. Parte dessas pessoas tem sua mobilidade e orientação limitadas devido à deficiência visual, impossibilitando-as de caminhar de forma segura em um ambiente. De acordo com a Organização Mundial da Saúde (OMS) (WHO, 2019), a falta de visão atinge cerca de 285 milhões de pessoas em todo o mundo, sendo que 13,68% têm perda total de visão e 86,32% são pessoas com perda parcial de visão. Ainda segundo a OMS, países em desenvolvimento ou de baixa renda correspondem a 89% dos casos de deficiência visual. O Brasil, de acordo com levantamento feito pelo Instituto Brasileiro de Geografia e Estatística (IBGE) em sua última pesquisa nacional de saúde (IBGE, 2015), possuía cerca de 7.2 milhões de pessoas com deficiências visuais no ano de 2013, correspondendo a 3,6% da população total daquele ano.

A orientação e mobilidade de uma pessoa com deficiência visual (PDV) são severamente prejudicadas e limitadas, uma vez que ela está impossibilitada de ver um obstáculo em seu caminho ou em qual direção está caminhando. Com essa preocupação, o tema "Orientação e Mobilidade" para PDVs é objeto de estudos, a fim de proporcionar a compreensão sobre quais são as necessidades e os desafios em determinados cenários, sob dois aspectos: o da percepção, voltado inicialmente à PDV, em que são buscadas maneiras de fornecer informações sobre um ambiente; e o conceitual, voltado inicialmente a um determinado ambiente, com o intuito de entender como é o local, sua disposição física (paredes, portas, etc.) e quais os possíveis caminhos que serão realizados no ambiente (LAHAV, 2014).

A área de visão computacional pode ser utilizada para auxiliar no problema de orientação e mobilidade causado pela perda da visão, oferecendo a possibilidade de navegação autônoma e independente em ambientes desconhecidos ou transformados, graças a técnicas cada vez mais avançadas de extração de características e processamento de imagem dos ambientes (PAIVA, 2020). Projetos em visão computacional podem ajudar PDVs em tarefas cotidianas, como acesso à informação, ao deslocamento e à orientação, uso de computadores e dispositivos computacionais e outras tarefas que vão além da navegação, mas também na sua localização dentro de um ambiente e na classificação de objetos que podem representar obstáculos em sua trajetória (AL-KHALIFA; AL-RAZGAN, 2016; NIU et al., 2018). Além da visão computacional, as redes neurais também podem ser usadas em conjunto com a visão computacional, em benefício daqueles com perda de visão, na classificação de objetos, obtendo ótimos resultados e tendo a capacidade de operar em tempo real com baixa taxa de erros, como é o caso das Redes Neurais Convolucionais (POGGI; MATTOCCIA, 2016).

A mobilidade é um fator importante em sistemas de navegação para PDVs, fazendo com que embarcar sistemas em hardware próprio se torne essencial para garantir a boa mobilidade. Alguns autores embarcaram seus projetos, fornecendo alto grau de independência para os usuários de seus sistemas (POGGI; MATTOCCIA, 2016; SIDDHARTHA; CHAVAN; UMA, 2018; HE et al., 2015; KUMAR et al., 2013; KOMMEY; HERRMAN; ADDO, 2019). Projetos mais recentes utilizam hardwares portáteis com placas de vídeo embarcadas, permitindo a implementação de algoritmos de visão computacional com relativa facilidade, em relação às placas *Field-Programmable Gate Array* (FPGA), sem grande impacto na velocidade do processamento em tempo real (DOURADO; PEDRINO, 2018; JAFRI; KHAN, 2018).

Os sistemas de assistência à navegação para pessoas com deficiências visuais podem ser desenvolvidos de maneiras distintas, de acordo com os sensores de entrada de dados e de como a informação processada é enviada para o usuário. Na literatura, os sensores de entrada de dados mais comuns são, dentre outros, GPS, câmeras, sensores inerciais (*Inertial Measurement Units* - IMUs), leitores de radiofrequência (*Radio-Frequency Identification* - RFID) e lasers infravermelhos (SHAHIRA; TRIPATHY; LIJIYA, 2019; TAPU; MOCANU; ZAHARIA, 2017; AL-KHALIFA; AL-RAZGAN, 2016; SIDDHARTHA; CHAVAN; UMA, 2018). Os dados oriundos de cada sensor podem ser usados ou não em conjunto com outros sensores, com o objetivo de entregar ao usuário uma informação precisa sobre o ambiente.

O resultado dos processamentos realizados, ou seja, uma direção a ser seguida, um obstáculo detectado e outros resultados similares devem ser informados ao usuário de forma a estimular sua percepção sobre o seu entorno e auxiliar na tomada de uma decisão. O sistema,

portanto, fornece um *feedback* com o que é necessário informar à PDV. Os dois principais tipos de *feedback* encontrados em publicações especializadas são o *feedback* sonoro ou acústico, em que sons (tons mono ou polifônicos, musicais, falas, etc.) são gerados e enviados via fone de ouvido ou algum dispositivo capaz de reproduzir sons, e o *feedback* tátil, que consiste na estimulação física do usuário, por meio de dispositivos com motores de vibração ligados em luvas, por exemplo, para indicar uma informação (POGGI; MATTOCCIA, 2016).

Entretanto, a maioria dos trabalhos encontrados na literatura foi desenvolvida de forma a ter seus funcionamentos de maneira fixa, fazendo com que diferentes usuários com perda de visão, com diferentes características, preferências, dificuldades e facilidades tenham que se adaptar ao método de interação oferecido pelo sistema, mesmo que seu desenvolvimento tenha sido centrado no usuário (JAFRI; KHAN, 2018). Tal imposição tende a ser um agente dificultador no uso de um sistema, uma vez que o usuário fica impedido de modificar o funcionamento desse sistema a fim de melhorar sua própria experiência, como no caso de usuários idosos que podem necessitar um método diferente de *feedback* ou adaptar o atual para suas necessidades e limitações (WEINBERGER et al., 2019). A prática do diálogo bidirecional entre sistema e PDV pode ser encontrada em alguns trabalhos, porém apenas poucos deles contemplam a personalização do sistema de acordo com os comandos do usuário, ainda que de forma limitada (DONATI et al., 2020; KATZ et al., 2012; PATIL et al., 2014; PRIYADARSHANA; WIMALARATNE, 2014). Não há abordagens, até onde foi possível pesquisar, que deem ao usuário uma grande gama de opções para customizar o funcionamento de um sistema de auxílio à navegação para PDVs e que permitam ao próprio usuário configurar características que vão desde o sistema de distância (métrico ou passos) até o tipo de *feedback*, além das distâncias de detecção de obstáculos, caminhos seguros e qual o tipo de navegação deseja utilizar.

Em sistemas de navegação baseados em visão computacional, o uso de câmeras RGB-D possibilita a captura de imagens de profundidade em escala de cinza e também, como resultado do seu processamento, extrair diversas informações relevantes de um ambiente, como caminhos livres e obstáculos (DOURADO; PEDRINO, 2018; POGGI; MATTOCCIA, 2016; AL-KHALIFA; AL-RAZGAN, 2016). Porém, esse tipo de imagem frequentemente apresenta um problema comum, independente da câmera RGB-D utilizada, de áreas de distâncias desconhecidas, apresentadas na cor preta. Alguns trabalhos da literatura tentam resolver esse problema empregando técnicas de reconstrução de imagens ou filtros recursivos, mas sem levar em consideração que tais técnicas podem ter alto impacto negativo na performance geral de sistemas de navegação, caso o *hardware* tenha capacidade limitada de processamento (QI et al., 2013; GARDUÑO-RAMÓN et al., 2017).

## 1.2 Questões de Pesquisa

Os sistemas de auxílio à navegação para pessoas com deficiências visuais são, atualmente, desenvolvidos baseados em diversas necessidades e preferências individuais, com o objetivo de atender a maior quantidade possível de usuários sob um perfil unificado, indicando que necessidades conflitantes, como o tipo de *feedback* ou a sua velocidade, podem ser escolhidas por maior abrangência ou, no melhor caso, parcialmente atendidas (JAFRI; KHAN, 2018). Isso pode resultar em sistemas menos adequados para um grupo de pessoas em relação a outros, devido às limitações e necessidades de cada grupo de PDVs. Além disso, a deficiência visual não se apresenta exclusivamente em um intervalo específico de idade. Com isso, alguns recursos em um sistema de navegação podem ser mais eficientes para usuários mais novos, enquanto outros podem atender melhor os mais idosos, indicando que, além da perda de visão, a diminuição de outros sentidos (como a audição) pode afetar a maneira como o sistema é utilizado (WEINBERGER et al., 2019). *Questão 1: Como prover um conjunto de recursos funcionais e adaptáveis de forma autônoma por PDVs, em sistemas de auxílio à navegação, que seja capaz de operar com as diferentes necessidades e preferências individuais, mesmo as conflitantes, e também de mitigar as limitações provenientes de outras deficiências sensoriais, além da visão?*

Os métodos de *feedback* dos sistemas de navegação são, em geral, por vias acústicas e/ou táteis. Em diversas abordagens (ADVANI et al., 2017; LI et al., 2018), cada um dos métodos é utilizado para apresentar diferentes informações para PDVs, indicando que uma mesma informação não pode, atualmente, ser representada por diferentes meios de *feedback*, a critério do próprio usuário. Assim como os métodos de *feedback*, os métodos de interação entre usuário e sistemas também podem assumir diferentes modalidades, principalmente via voz e meios táteis, como botões. Após diversas buscas na literatura, foi observado que os trabalhos que implementam mais de um meio de interação geralmente atribuem funcionalidades distintas para cada meio e que, assim como para os métodos de *feedback*, não é concedida a liberdade de escolha para o usuário decidir como prefere interagir com os sistemas (LI et al., 2018). *Questão 2: Como as interações de entrada e informações de feedback podem ser representadas simultaneamente, metodologicamente, por métodos acústicos e táteis, de forma a coexistirem dentro de um sistema, como opções para as interações e/ou feedbacks serem acústicos, táteis ou ambos, de acordo com a escolha individual de cada usuário?*

Os sistemas embarcados de navegação para PDVs baseados em visão computacional utilizam, na maioria dos casos, câmeras RGB-D para captar imagens de profundidade do ambiente e extrair informações. Apesar das diferentes técnicas de cálculo de disparidade, as imagens de profundidade oriundas dessas câmeras apresentam um problema comum de áreas com pro-

fundidade desconhecida, sendo um forte indicativo de risco para o processo de segmentação de caminhos e/ou obstáculos. Algumas abordagens tratam desse problema com algoritmos conhecidos, porém há o custo computacional a se considerar devido à limitada fonte de energia desses sistemas(QI et al., 2013; GARDUÑO-RAMÓN et al., 2017). *Questão 3: Como fazer a correção dos dados de distâncias desconhecidas em imagens de profundidade sem que haja impacto na performance do sistema ou em seu consumo energético?*

## 1.3 Hipóteses

Dadas as motivações e questionamentos apresentados nas seções anteriores, as seguintes hipóteses são apresentadas para futura comprovação:

- A possibilidade de customização de um sistema de navegação por parte de um usuário com perda de visão pode ser um fator que torne positivas as experiências do usuário, fazendo com que a adaptação de uma pessoa seja mais rápida ao sistema, ao customizá-lo para funcionar da maneira mais conveniente às suas próprias percepções, por meio de uma metodologia de customização;
- A redundância de formas de interação de entrada do usuário por diferentes métodos, como voz e tátil, torna o sistema de navegação mais acessível aos diferentes perfis de usuários com deficiência visual, já que passa a ser possível escolher o método mais conveniente para a interação;
- A representação de informações por diferentes métodos de *feedback* permite que PDVs com outras deficiências sensoriais, como baixa audição, possam utilizar o sistema e receber as mesmas informações que são enviadas via som, utilizando a customização do sistema para alterar o método de *feedback*;
- As imagens de profundidade, oriundas de câmeras RGB-D usadas por sistemas de navegação para PDVs, contendo distâncias desconhecidas, podem ser eficientemente corrigidas por meio de filtros morfológicos, otimizados em relação à qualidade e à complexidade, e que previnam problemas de segmentação de caminhos e/ou obstáculos que poderiam representar risco à confiabilidade do sistema.

## 1.4 Objetivos

Este trabalho tem como objetivo principal prover a independência de pessoas com perda de visão na realização de percursos seguros em ambientes abertos ou fechados, por meio de um sistema embarcado de navegação, altamente customizável de maneira autônoma pelos próprios usuários, de acordo com suas necessidades e limitações individuais, que garanta mobilidade sem prejudicar a performance de processamento.

Para alcançar o objetivo principal indicado, os seguintes objetivos específicos são apresentados:

- Desenvolver um modelo abstrato modularizado, para servir de base para implementação de sistemas embarcados de navegação, com suporte à customização autônoma para auxílio a PDVs;
- Desenvolver um método de otimização multiobjetivo evolucionário e cartesiano, para geração de filtros morfológicos otimizados para correção de imagens de profundidade de câmeras RGB-D, visando baixo erro e baixa complexidade;
- Desenvolver uma metodologia para customização de funcionalidades de sistemas embarcados de navegação, por meio da especificação detalhada de comandos multimodais de usuários para cada funcionalidade customizável do sistema e de como elas podem ser alteradas a partir de cada entrada do usuário;
- Desenvolver um protótipo de sistema embarcado de navegação totalmente customizável pelo próprio usuário, via comandos de voz ou teclado braille, de forma a ser possível detectar obstáculos e calcular caminhos livres no percurso de PDVs.

Apesar dos objetivos citados, este projeto não pretende substituir a bengala ou o cão-guia que pessoas com perda de visão utilizam para se locomover atualmente, mas sim complementá-los com o uso de recursos computacionais.

## 1.5 Contribuições da Tese

As contribuições desta tese de doutorado são:

- **Modelo Abstrato de Sistemas Multimodais, Interativos e Customizáveis para Auxílio à Navegação Segura de PDVs:** Define um modelo abstrato para nortear projetos de

sistemas interativos e customizáveis, com o foco em auxiliar a navegação segura de pessoas com deficiências visuais, fazendo uso de um ou mais meios de entrada e saída de dados. É baseado em módulos e recursos auxiliares genéricos, podendo ser aplicado para diferentes tecnologias e hardwares;

- **Metodologia para Customização de Sistemas de Navegação para PDVs:** É uma metodologia para definição de recursos customizáveis em sistemas de navegação para PDVs. A partir de um recurso customizável, a metodologia oferece etapas para a especificação de como customizar determinado recurso, partindo de dispositivos de entrada do usuário;
- **Método de Otimização Multiobjetivo Evolucionário e Cartesiano para Correção de Imagens de Profundidade (NSGA2CGP):** Método criado com a finalidade de gerar e otimizar filtros morfológicos para a correção de imagens de profundidade incompletas, contendo áreas com distâncias desconhecidas, utilizadas em sistemas de navegação baseados em visão computacional;
- **Protótipo Vestível com Sistema de Navegação Multimodal e Interativo para Auxílio Customizável a Pessoas com Deficiências Visuais:** Como resultado da união das contribuições anteriores, foi criado um protótipo vestível contendo um sistema embarcado multimodal e interativo capaz de ajudar pessoas com deficiências visuais a caminhar de forma segura em diversos ambientes, podendo interagir com o sistema ou customizá-lo de acordo com suas próprias necessidades.

## 1.6 Metodologia de Pesquisa

### 1.6.1 Natureza e Abordagem da Pesquisa

Este trabalho de pesquisa pode ser definido como de natureza majoritariamente aplicada, quali-quantitativo e também exploratório. A classificação da pesquisa como quali-quantitativa justifica-se pelo fato de não ser possível mensurar quantitativamente o objetivo principal, ou seja, não é possível calcular objetivamente a independência de uma pessoa ao caminhar em um determinado ambiente. Porém, é possível quantificar elementos que são essenciais para o desenvolvimento do trabalho e ajudam a atingir o objetivo principal.

### 1.6.2 Método de Pesquisa e Instrumentos de Coleta de Dados

O método de pesquisa empregado para a execução deste trabalho é o de estudo de caso. Por possuir uma metodologia de pesquisa aplicada, estudos de caso preveem a aplicação de conhecimentos visando resolver problemas (PROVDANOV; FREITAS, 2013). Para tanto, são trabalhadas condições contextuais específicas de um grupo, como o cotidiano de PDVs em deslocamentos com e sem obstáculos, permitindo, segundo Yin (2001), evidenciar variáveis significativas que poderiam não ser observadas em ambientes controlados.

A coleta de dados e evidências pode ser feita de diversas formas em um estudo de caso. Entre elas, entrevistas, levantamentos, análise de documentos, observações diretas e indiretas, etc.(GERHARDT; SILVEIRA, 2009). Para o presente trabalho, a coleta de dados pôde ser realizada principalmente por meio de entrevistas não estruturadas, com a finalidade de conhecer melhor o domínio do problema e procurar evidenciar suas principais variáveis.

No decorrer da execução do projeto e desenvolvimento do protótipo apresentado neste trabalho, a coleta de dados foi feita via observação direta intensiva, uma vez que os participantes foram observados assistematicamente e também frequentemente entrevistados de forma não estruturada acerca de suas impressões em relação aos aprimoramentos do sistema de auxílio à navegação do protótipo.

Para a avaliação do protótipo vestível, foram empregadas técnicas de observação direta extensiva junto aos participantes, com formulários contendo perguntas fechadas (Apêndice A), após a finalização das sessões de experimentos práticos em ambientes não controlados.

### 1.6.3 Método de Amostragem e Seleção

O método de amostragem adotado neste trabalho foi o não probabilístico, em que não há aleatoriedade na escolha dos participantes, com amostras por acessibilidade, haja vista que foram selecionados todos os voluntários disponíveis, ou seja, a amostra foi composta de todos os voluntários a que se teve acesso. O método não probabilístico por acessibilidade, portanto, é justificado pela falta de uma população que permita a seleção aleatória e também pela aceitação de voluntários em relação à participação no projeto (PROVDANOV; FREITAS, 2013).

As sessões de experimentos foram realizados com pessoas com deficiência visual, frequentadoras da ONG PARA-DV na cidade de Araraquara-SP. Os participantes utilizaram o protótipo vestível desenvolvido com os conceitos apresentados nesta tese e o avaliaram ao final do período de experimentos, com um formulário sobre suas percepções no decorrer do processo. Os par-

ticipantes também preencheram o Termo de Consentimento Livre e Esclarecido (Apêndice B), concordando com os termos da participação. Sete PDVs participaram dos experimentos e contribuíram com a evolução do sistema de navegação e protótipo vestível. No Quadro 1.1 são listados os participantes e suas respectivas informações sobre grau de cegueira, se é congênita ou adquirida, idade, tempo de deficiência visual, gênero, escolaridade, deficiências associadas e se fazem uso de outras tecnologias, como leitores de tela e/ou comandos de fala.

**Quadro 1.1: Informações gerais sobre os membros participantes da ONG PARA-DV.**

Participante	Cego ou Baixa Visão	Congênita ou Adquirida	Idade	Tempo de Def. Visual (Anos)	Gênero	Escolaridade	Outras Deficiências	Usa Leitor de Tela	Usa Comandos de Voz
P1	Cego	Adquirida	64	39	Feminino	Médio Incompleto	Baixa Audição	Sim	Não
P2	Cego	Adquirida	55	05	Feminino	Médio Incompleto	Baixa Audição	Sim	Não
P3	Baixa Visão	Adquirida	29	02	Feminino	Médio Incompleto	Não	Sim	Não
P4	Cego	Adquirida	53	11	Masculino	Médio Completo	Não	Sim	Não
P5	Cego	Adquirida	64	24	Feminino	Médio Incompleto	Não	Não	Não
P6	Cego	Congênita	32	32	Masculino	Superior Completo	Não	Sim	Sim
P7	Baixa Visão	Adquirida	45	18	Feminino	Médio Completo	Não	Sim	Sim

#### 1.6.4 Análise dos Dados

A análise de dados utilizada para este trabalho foi a descritiva, sendo observadas as evidências, relacionando-as com os objetivos do trabalho em busca de comprovação das hipóteses, e estatística, com a ponderação das médias obtidas das respostas de cada questão e grupo de questões do formulário de avaliação. Há também a abstração dos dados coletados para que seja viável explicar possíveis relações de causa e efeito. A confiabilidade dos dados obtidos via formulário foi calculada com o Coeficiente Alfa de Cronbach (TABER, 2018).

#### 1.6.5 Atividades Metodológicas

O desenvolvimento deste trabalho envolve determinadas atividades metodológicas que visam contribuir com a qualidade da pesquisa. Assim, cada atividade tem um objetivo específico, prático ou teórico, e fornece embasamento (objetivo teórico), aplicações (objetivo prático) ou resultados (objetivo prático). São propostas seis atividades metodológicas: Análise do Estado

da Arte, Método de Otimização Multiobjetivo Evolucionário Cartesiano para Correção de Imagens de Profundidade, Metodologia para Customização de Sistemas de Navegação para PDVs, Desenvolvimento de Sistema de Navegação para Pessoas com Deficiência Visual, Implantação de Sistema de Navegação em Hardware Embarcado e Vestível e Validação e Avaliação de Protótipo.

#### **1.6.5.1 Análise do Estado da Arte**

A literatura foi continuamente explorada com foco em sistemas de auxílio à navegação para PDVs, tanto em termos de software e hardware quanto em termos teórico e metodológico, por meio de buscas nas principais bases de dados disponíveis. Técnicas que envolvam a segmentação de caminhos livres e objetos foram investigadas em busca de novas ideias e tendências, bem como a verificação de sistemas que permitem a customização de suas regras e como esses sistemas enviam respostas para seus usuários. Abordagens sobre a correção de imagens de profundidade capturadas de câmeras RGB-D também foram verificadas e analisadas na literatura.

#### **1.6.5.2 Método de Otimização Multiobjetivo Evolucionário Cartesiano para Correção de Imagens de Profundidade**

Antes do início do desenvolvimento do sistema de navegação, foi criado um método de otimização multiobjetivo evolucionário cartesiano com base no algoritmo NSGA-II, utilizando cromossomos de programação genética cartesiana para otimizar filtros morfológicos capazes de corrigir imagens de profundidade incompletas. O objetivo é minimizar o erro e a complexidade dos elementos estruturantes utilizados pelo filtro, disponibilizando diversas soluções factíveis com diferentes níveis de qualidade e complexidade.

#### **1.6.5.3 Metodologia para Customização de Sistemas de Navegação para PDVs**

A metodologia para customização foi criada como uma formalização do suporte que sistemas embarcados de navegação para PDVs podem ter para a customização de seu funcionamento a partir do próprio usuário. Essa metodologia parte do princípio de que a customização será recebida na forma de um comando via dispositivos de entrada, como microfones, teclados, etc., e que a origem dos comandos é o usuário. Desta forma, a metodologia define a especificação de cada funcionalidade customizável no sistema e como elas podem ser alteradas.

#### **1.6.5.4 Desenvolvimento de Sistema de Navegação para Pessoas com Deficiência Visual**

As interações com potenciais usuários do sistema de navegação se iniciaram justo com o desenvolvimento do próprio sistema. Em visitas à ONG PARA-DV, os participantes voluntários responderam diversas perguntas acerca do problema da navegação em ambientes internos e externos, ao passo que tais entrevistas foram realizadas com o intuito de extrair requisitos para possíveis alterações e/ou adaptações na arquitetura do sistema em desenvolvimento, bem como no protótipo vestível. Foram levantados os principais problemas que PDVs têm quando precisam se locomover em um ambiente e seus relatos levaram à reflexão sobre as funcionalidades do projeto.

Com base na contínua análise do estado da arte e dos requisitos levantados junto aos participantes entrevistados, diversos recursos foram incluídos tanto no sistema de navegação quanto no modelo do protótipo vestível, visando sempre atender as necessidades reveladas pelos potenciais usuários. Além disso, os participantes também relataram necessidades sobre dificuldades pontuais de navegação que serão incluídas em futuras versões do protótipo.

#### **1.6.5.5 Implantação de Sistema de Navegação em Hardware Embarcado e Vestível**

A etapa de "Implantação de Sistema de Navegação em Hardware Embarcado e Vestível" consistiu na implantação do sistema de navegação em hardware a ser embarcado em um artigo vestível. O artigo escolhido foi uma mochila totalmente customizada para o projeto, contendo compartimentos para acomodar hardware, fios, fonte de energia, motores de vibração, câmera e outros elementos sem prejudicar as características de uma mochila comum. O hardware foi devidamente preparado para embarcar o sistema de navegação, tendo as bibliotecas necessárias instaladas ou compiladas para garantir o funcionamento correto do sistema. Com a implantação em hardware embarcado, ciclo de refinamento e melhorias de código foi realizado para diminuir gargalos de processamento, redundâncias e erros.

#### **1.6.5.6 Validação e Avaliação de Protótipo**

Por fim, nas etapas de "Validação do Protótipo" e "Avaliação de Protótipo", o protótipo criado no decorrer deste trabalho foi submetido à navegação em três ambientes diferentes. A validação foi realizada pelo autor do projeto e por um aluno de mestrado, enquanto a avaliação foi feita com participantes da ONG PARA-DV e serviu para verificar a qualidade do protótipo desenvolvido, bem como avaliar se as hipóteses apresentadas neste podem ser comprovadas.

## 1.7 Organização do Texto

No Capítulo 1 foram apresentados o contexto geral no qual esta pesquisa se insere, além das motivações que levam à realização deste trabalho. Foram apresentadas as questões de pesquisa, hipóteses que se pretende provar ao fim desta tese, os objetivos e a metodologia de pesquisa que foi empregada no desenvolvimento do trabalho.

No Capítulo 2, "Fundamentação Teórica de Sistemas de Navegação para Auxílio à Pessoas com Deficiências Visuais", são apresentados os conceitos e dispositivos que podem estar presentes na criação de um sistema embarcado de navegação, técnicas de customização e *feedbacks* e como estas podem trabalhar em conjunto para o benefício do usuário. São apresentadas também as plataformas nas quais um sistema pode ser embarcado, incluindo as *Graphics Processing Units* (GPUs) embarcadas.

No Capítulo 3, "Estado da Arte em Sistemas de Navegação para Auxílio à Pessoas com Deficiências Visuais", são apresentados metodologias e sistemas conhecidos na literatura, que podem ser utilizados no cotidiano de PDVs com o intuito de obter autonomia para ir de um ponto a outro, em ambientes fechados ou não. Ainda, são apontados trabalhos que permitem a customização do funcionamento do sistema de navegação, bem como os métodos de entrada de usuário para a customização. Ao fim, as limitações são apresentadas juntamente com as oportunidades existentes que não são contempladas na literatura.

No Capítulo 4, "Método de Otimização Multiobjetivo Evolucionário Cartesiano para Correção de Imagens de Profundidade", é apresentado o método de otimização multiobjetivo de filtros morfológicos para correção de imagens de profundidade baseado no NSGA-II e na Programação Genética Cartesiana, com divisão em fases.

No Capítulo 5, "Protótipo Vestível com Sistema de Navegação Multimodal e Interativo para Auxílio Customizável a Pessoas com Deficiências Visuais", o modelo abstrato de sistema embarcado para navegação de PDVs é apresentado e explicado, seguido da metodologia para customização de sistemas de navegação utilizada no sistema desenvolvido nesta tese. Em seguida, a estrutura lógica do sistema é introduzida e ilustrada em etapas. Por fim, o protótipo vestido é apresentado com sua estrutura física explicada, juntamente com o hardware utilizado.

No Capítulo 6, "Experimentos", é apresentada a metodologia adotada para as atividades realizadas a fim de validar o protótipo em ambiente controlado e o modo como os participantes com deficiência visual da ONG PARA-DV realizaram experimentos em diferentes ambientes, controlados e não controlados, para avaliar o protótipo. Também é apresentada a metodologia para avaliação do método NSGA2CGP, especificamente na Seção 4, e os parâmetros utilizados

para isso.

No Capítulo 7, "Resultados e Discussões", os resultados obtidos com as atividades de validação e avaliação do protótipo e também do método NSGA2CGP são apresentados. Os resultados são analisados e discutidos, buscando elucidar se os dados apresentados como resultados corroboram as hipóteses apresentadas nesta tese.

Por fim, no Capítulo 8, "Conclusões", a tese é concluída, ressaltando as contribuições deste trabalho, bem como suas limitações. São apresentados os diversos trabalhos futuros que serão realizados a fim de dar continuidade à pesquisa iniciada por esta tese.

# Capítulo 2

## FUNDAMENTAÇÃO TEÓRICA DE SISTEMAS DE NAVEGAÇÃO PARA AUXÍLIO A PESSOAS COM DEFICIÊNCIAS VISUAIS

---

---

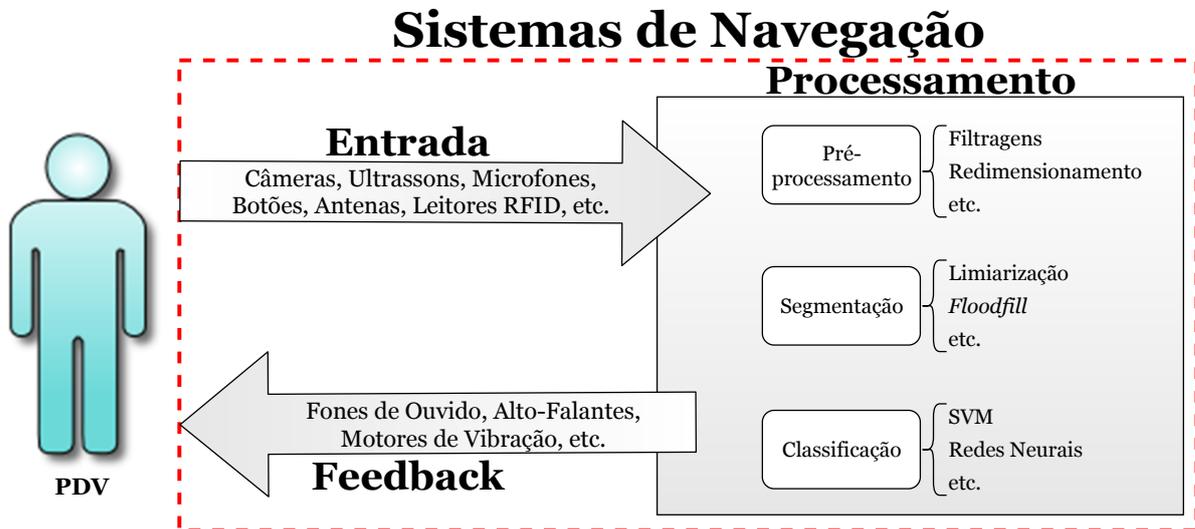
Neste capítulo serão apresentados os principais conceitos que norteiam os sistemas de navegação que auxiliam pessoas com deficiência visual em tarefas cotidianas, como realizar percursos, evitar e classificar obstáculos, interagindo com o sistema e recebendo respostas dele. Além dos sistemas de navegação, será apresentada a fundamentação básica sobre algoritmos de otimização multiobjetivo e programação genética, com foco em seus subtipos específicos, NSGAI e Programação Genética Cartesiana, respectivamente, que são utilizados para a criação de um método para geração automática de filtros morfológicos para o pré-processamento de imagens de profundidade.

### 2.1 Sistemas de Navegação

Os sistemas de navegação para pessoas com deficiência visual são também conhecidos como *Electronic Travel Aids* (ETAs) e auxiliam as pessoas com deficiências visuais em diversas tarefas que envolvem, direta ou indiretamente, seu deslocamento de forma segura e autônoma em qualquer tipo de ambiente, de forma não invasiva, ou seja, mantendo o acesso de seu usuário aos sinais do ambiente, oferecendo boa ergonomia e mobilidade, entre outros (PAIVA, 2020). Os ETAs são responsáveis por realizar a substituição sensorial ao transformar as informações oriundas do ambiente em torno de seu usuário, outrora percebidas pela visão, em estímulos que uma pessoa sem a visão pode compreender (DAKOPOULOS; BOURBAKIS, 2010).

De maneira geral, os sistemas de navegação podem ser simplificados de acordo com a

Figura 2.1. Nela, é possível observar que a pessoa com deficiência visual, por meio de sensores de entrada, envia sinais para um dispositivo que, após processar os dados recebidos, envia respostas ou *feedbacks* para o usuário.



**Figura 2.1: Estrutura Simplificada de um Sistema de Navegação.**

### 2.1.1 Sensores de Entrada

Existem diversos tipos de sensores de entrada que podem ser utilizados em um sistema de navegação para pessoas com deficiência visual. É possível classificar os sensores de entrada em baseados em visão e não baseados em visão (POGGI; MATTOCCIA, 2016). Os baseados em visão são sensores como câmeras, estereoscópicas ou não, que podem fazer uso de sensores ativos para captar imagens como o Kinect, que utiliza infravermelho para captar imagens de profundidade. Os sensores não baseados em visão, por sua vez, não trabalham com imagens, mas com sinais imperceptíveis aos olhos humanos, como sensores inerciais, sensores ultrassônicos, sensores de posicionamento global (GPS), entre outros. Além desses, microfones e botões, embora não sejam normalmente listados como sensores na literatura, também serão considerados como sensores de entrada de dados para este trabalho.

#### 2.1.1.1 Câmeras

As câmeras são sensores baseados em visão que captam imagens do ambiente e as enviam para a plataforma de processamento. Nos sistemas de navegação é comum encontrar abordagens com câmeras simples, duas câmeras e câmeras RGB-D (*Red, Green, Blue and Depth*, como o Kinect da Microsoft e a ZED da Stereo Labs (NEWCOMBE et al., 2011; PILLAI; RAMALINGAM; LEONARD, 2016). A Figura 2.2 exibe as duas câmeras RGB-D.



Figura 2.2: Microsoft Kinect for XBox 360 (esquerda) e ZED Stereo Camera (direita) (MICROSOFT, 2017; STEREO LABS, 2017).

### 2.1.1.2 IMUs

Os sensores inerciais, também chamados de *Inertial Measurement Units* (IMUs), são sensores que contêm acelerômetros, capazes de calcular a velocidade do movimento em três eixos (X, Y e Z), e giroscópios, que calculam a mudança angular também nos três eixos, como o acelerômetro (KING, 1998). Alguns IMUs também contam com magnetômetros, que inferem o magnetismo em um dado momento. Com um IMU, é possível estimar e corrigir a direção, por exemplo, de uma aeronave em curso. Há também a união de câmera RGB-D com um IMU embarcado, que, além de capturar imagens, também registra o posicionamento no momento da captura, como faz a câmera ZED e a Intel Realsense D435i (STEREO LABS, 2017; INTEL, 2018). Entretanto, nos IMUs embarcados em câmeras, o funcionamento da captura de imagens e dos sensores inerciais são independentes e seu uso em conjunto pode ser programado via software.

### 2.1.1.3 Outros Sensores

Além das câmeras e IMUs, a literatura contém projetos com a utilização dos seguintes sensores de entrada:

- **Global Positioning System (GPS):** Sensor baseado em satélite que recupera a localização global do usuário (POGGI; MATTOCCIA, 2016).
- **Lasers infravermelhos:** Sensor de baixo alcance (distância máxima de 6 metros) que emite iluminação no espectro infravermelho. Sua utilização está ligada à detecção de obstáculos, pelo reflexo obtido pela emissão do laser (MEKHALFI et al., 2016).
- **Ultrassom:** Assim como os lasers, os sensores ultrassônicos emitem um sinal, no caso, ultrassônico, em uma faixa específica do espectro eletromagnético e seu eco é refletido e recebido pelo receptor. Com isso, é possível a detecção de obstáculos (KUMAR et al., 2013).
- **Radio-Frequency Identifier (RFID):** Sensor de baixo/médio alcance em que *tags* são emissores estáticos de sinais de radiofrequência e um receptor é utilizado para captar os

dados de cada emissor e, desta forma, obter dados sobre a localização (IPIÑA; LORIDO; LÓPEZ, 2011; CEIPIDOR et al., 2006).

- **Bluetooth e Wi-Fi:** Similar à prática com RFID, o uso de sensoriamento baseado em *Bluetooth* ou *Wi-Fi* utiliza a força do sinal dos emissores de sinais para definir sua posição por meio da triangulação dos sinais (DUARTE, 2014).

## 2.1.2 Tipos de *Feedback*

O *feedback* do sistema é o resultado do processamento, realizado com os dados obtidos dos sensores de entrada, transmitido para o usuário. Não podendo contar com a visão, o *feedback* pode, portanto, ser enviado de duas maneiras para o usuário: via tátil e/ou via acústica, de acordo com os trabalhos encontrados na literatura.

### 2.1.2.1 *Feedback* Tátil

O *feedback* tátil refere-se a estímulos físicos que o sistema envia a atuadores para representar uma informação à PDV. Os atuadores normalmente são vestimentas (coletes, luvas, pulseiras, etc.) que carregam pequenos motores de vibração de corrente contínua, como o C1026B002F e o C1030B028F, que podem vibrar de acordo com a informação representada (MANCINI; FRONTONI; ZINGARETTI, 2018; DAKOPOULOS; BODDHU; BOURBAKIS, 2007). Dois exemplos de *feedback* tátil são a luva, apresentada por Poggi e Mattoccia (2016), que vibra em três dedos diferentes, indicando obstáculo à esquerda, no dedo indicador, à frente, no dedo médio, e à direita, no dedo anelar, e o cinto vibratório, de Dakopoulos e Bourbakis (2008), que fornece o *feedback* vibrando em baixa intensidade para obstáculos distantes e aumentando a intensidade relativamente à proximidade do obstáculo. Ambos os dispositivos são exibidos na Figura 2.3.



(a) Adaptado de Poggi e Mattoccia (2016)



(b) Dakopoulos e Bourbakis (2008)

**Figura 2.3: Dispositivos de *feedback* tátil.**

### 2.1.2.2 *Feedback* Acústico ou Sonoro

Enquanto a técnica de *feedback* tátil envia estímulos físicos como resposta à PDV, o *feedback* acústico ou sonoro envia as respostas do sistema como sons. Os sons podem ser construídos de diversas maneiras, por meio de frequências e intensidades, polifônicos e também como frases sintetizadas em voz (LI et al., 2016; KATZ et al., 2012). Entretanto, segundo Pissaloux (2002), uma das premissas dos sistemas de navegação é que não deve haver a imersão total do usuário no sistema. Desta forma, é essencial que o volume do *feedback* permita que o usuário tome conhecimento das respostas do sistema sem perder o contato com os sons do ambiente. Idealmente, o próprio usuário deveria ter o controle do nível de imersão com o sistema, de acordo com as suas necessidades. Em sistemas com esse tipo de resposta, é comum a utilização de fones de ouvido auriculares, intra-auriculares, supra-auriculares e circumaural (KUMAR et al., 2013; SAINARAYANAN; NAGARAJAN; YAACOB, 2007). Nos últimos anos, os fones de ouvido de condução óssea têm sido utilizados como uma alternativa que previne a imersão do usuário à resposta do sistema (POGGI; MATTOCCIA, 2016; TAPU; MOCANU; ZAHARIA, 2017). Neste trabalho, assim como nos demais encontrados na literatura, os sons acústicos são gerados via software com bibliotecas como OpenAL, eSpeak, entre outras. Há também a possibilidade de utilização de sons estáticos, carregados de arquivos mp3, wav, etc.

### 2.1.3 Pré-Processamento

O pré-processamento dos dados obtidos pelo(s) sensor(es) de entrada tem um papel crucial para a garantia do pleno funcionamento dos sistemas de navegação. Algoritmos e métodos específicos para cada tipo de dado (imagem, áudio, etc.) são responsáveis por preparar/adequar tais dados de forma que seja possível processá-los da melhor maneira dentro do contexto do sistema (WANG; TIAN, 2012).

Em imagens, o pré-processamento pode ser composto de diversas funções que submetem as imagens a filtros que melhoram o contraste, atenuam ruídos, realçam bordas, etc., bem como funções que aumentam ou diminuem a escala das imagens (GONZALEZ; WOODS, 2006). O principal método de pré-processamento empregado neste trabalho é o de filtragem morfológica, que utiliza a teoria de Morfologia Matemática para a criação de filtros capazes de corrigir e/ou eliminar imperfeições presentes em imagens de profundidade.

### 2.1.3.1 Morfologia Matemática

A Morfologia Matemática (MM) é uma área teórica de estudos sobre estruturas e formas que fornece um conjunto de operadores lógicos e aritméticos que podem ser utilizados como ferramentas de processamento de imagens (PEDRINO et al., 2015). A teoria relacionada à MM descreve duas operações básicas, dilatação e erosão, cujo uso pode ser feito para alcançar operações complexas de análise e processamento. A dilatação aplica o operador de união entre uma imagem e um elemento estruturante, enquanto a erosão aplica o operador de interseção (SOILLE, 2004). Elementos estruturantes são pequenas estruturas usadas por operadores morfológicos que podem assumir diversas formas, de acordo com as propriedades da imagem analisada (SERRA; SOILLE, 1994). A operação morfológica posiciona o centro de um elemento estruturante sobre cada pixel de uma imagem e, por meio de dilatações e erosões, essa imagem é alterada de acordo com os elementos estruturantes usados.

Por meio da Morfologia Matemática também é possível elaborar conjuntos de operações morfológicas complexas, com múltiplos elementos estruturantes, capazes de lidar com vários problemas clássicos da área de processamento de imagens, como eliminação de ruído, preenchimento de bordas, entre outros (TI et al., 2017), em aplicações voltadas para áreas como a de tecnologia assistiva<sup>1</sup> (LESZCZUK; BARAN; SKOCZYLAS, 2014).

### 2.1.4 Segmentação

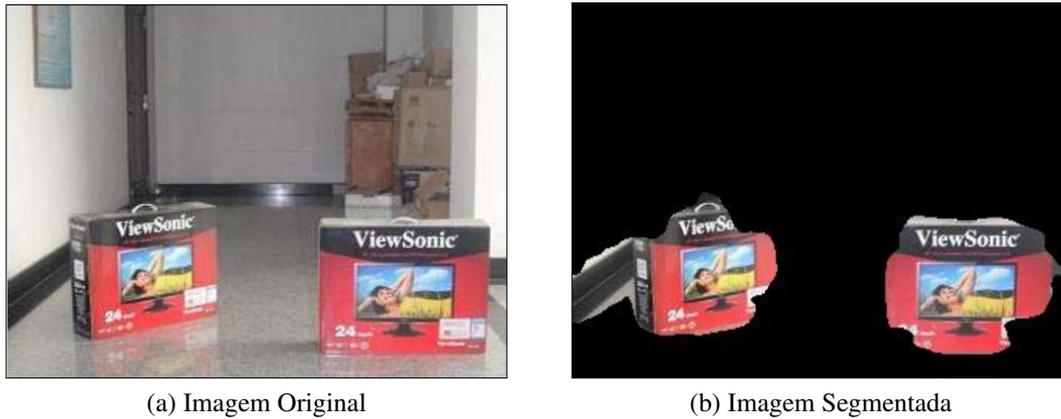
A segmentação é uma das operações clássicas de processamento de imagem que pode ser utilizada nos sistemas de navegação para diversas finalidades, dentre elas segmentação de caminhos livres e/ou objetos/obstáculos (DOURADO; PEDRINO, 2018; OTT; EVERINGHAM, 2015). Há uma variedade de métodos que podem ser utilizados para segmentar uma imagem, a título de exemplos: *Template Matching*, *Watershed*, limiarização, entre outros (LIN; DAVIS, 2010; CHEN; GUO; SUN, 2010; ALADREN et al., 2014). A Figura 2.4 exibe um exemplo de segmentação com a técnica de limiarização.

#### 2.1.4.1 *Floodfill*

A técnica de segmentação aplicada neste trabalho para segmentação de obstáculos é a *flood-fill*, que, a partir de um ponto de origem e limiares de tolerância inferiores e superiores, repinta

---

<sup>1</sup>Segundo a Lei 13.146 (BRASIL, 2015) – art. 3º [...] "III - tecnologia assistiva ou ajuda técnica: produtos, equipamentos, dispositivos, recursos, metodologias, estratégias, práticas e serviços que objetivem promover a funcionalidade, relacionada à atividade e à participação da pessoa com deficiência ou com mobilidade reduzida, visando à sua autonomia, independência, qualidade de vida e inclusão social".



**Figura 2.4:** Exemplo de segmentação utilizando limiarização (Chen, Guo e SUN (2010)).

os componentes 4-conectados ou 8-conectados (TORBERT, 2016). Assim, seja  $(sx, sy)$  as coordenadas do pixel origem/semente, o preenchimento dos vizinhos 4-conectados ou 8-conectados serão preenchidos com a mesma cor da origem como mostrado na Eq. 2.1, sendo  $(x, y)$  os vizinhos conectados da origem.

$$IMG_{(sx, sy)} - limiarInf \leq IMG_{(x, y)} \leq IMG_{(sx, sy)} + limiarSup \quad (2.1)$$

O algoritmo será executado para cada vizinho conectado e preenchido de acordo com a Eq. 2.1, até que nenhum outro pixel esteja dentro do limiar de tolerância do pixel origem. A escolha desse algoritmo para a tarefa de segmentação de obstáculos foi baseada em testes empíricos realizados preliminarmente em laboratório, sendo que a técnica *Floodfill* foi a que obteve melhores resultados em termos de performance em relação às técnicas mencionadas anteriormente nesta subseção.

### 2.1.5 Classificação

Classificar uma imagem ou uma porção dela é um processo que implica o reconhecimento de tal imagem como uma entre diversas classes. Os algoritmos de classificação, por sua vez, se estendem desde simples cálculos de distâncias Euclidianas até complexas redes neurais de muitas camadas (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). As características das imagens são utilizadas pela grande maioria dos métodos de classificação e são extraídas com base nos chamados "descritores", também conhecidos como extratores de características. As características das imagens são usadas para o treinamento dos classificadores, que podem ser estatísticos e probabilísticos, como redes bayesianas, redes neurais, comparador de características, entre outros (GONZALEZ; WOODS, 2006). Para fins meramente explicativos, as abordagens de classificação serão divididas em dois grupos: as clássicas, considerando os métodos que reque-

rem o processo manual de extração de características, e as modernas, que serão consideradas as abordagens que englobam os métodos baseados em redes neurais profundas.

As abordagens clássicas para classificação de objetos/obstáculos, neste trabalho, são aquelas em que ainda é necessário fazer a aplicação e o refinamento de técnicas de extração das características de um conjunto de imagens para então submetê-las ao treinamento de um classificador. Classificadores como Máquina de Vetor de Suporte ou *Support Vector Machines* (SVM) e *K-Nearest Neighbors* (KNN) (GHILARDI; JACQUES; MANSSOUR, 2018; CHENG et al., 2018) e detectores e descritores como SURF e SIFT (TAPU; MOCANU; ZAHARIA, 2017; THAKOOR et al., 2013; BAY et al., 2008; LOWE, 1999) foram muito explorados para várias finalidades, incluindo para ajudar PDVs.

São consideradas abordagens modernas de classificação, neste trabalho, os métodos de classificação e detecção de obstáculos que não necessitam da aplicação de extratores de características como base de um treinamento e são baseados nas redes neurais profundas. A literatura sobre os sistemas de navegação para PDVs tem contado cada vez mais com trabalhos que aplicam classificadores modernos e sofisticados, como as Redes Neurais Convolucionais (RNCs) e as *Long Short-Term Memory* (LSTMs) (IAKOVIDIS et al., 2020; OHN-BAR et al., 2018). Neste trabalho, foi feito uso das Redes Neurais Convolucionais para a classificação de obstáculos.

### 2.1.5.1 Redes Neurais Convolucionais

As Redes Neurais Convolucionais são um tipo de rede neural profunda, categorizadas pela existência de múltiplas camadas intermediárias (GOODFELLOW; BENGIO; COURVILLE, 2016). A popularidade das redes neurais convolucionais aconteceu em 2012, no desafio de classificação ILSVRC, quando uma rede neural convolucional chamada AlexNet venceu o desafio com ampla margem e ótimos resultados (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

As redes neurais convolucionais são redes neurais compostas de operações de convolução e *pooling* ou *subsampling* (redução das dimensões da imagem) sucessivos, com uma rede completamente conectada ao seu final para realizar a classificação. Em outras palavras, uma RNC é treinada para evoluir filtros sucessivos e enviar o resultado a um classificador (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). Tudo isso em uma mesma rede, formando uma arquitetura como a vencedora do desafio ILSVRC de 2012 ou a primeira arquitetura de RNC, proposta em 1998 (LECUN et al., 1998), apresentada na Figura 2.5.

A rede apresentada por Lecun et al. (1998) é um exemplo básico em relação à complexidade

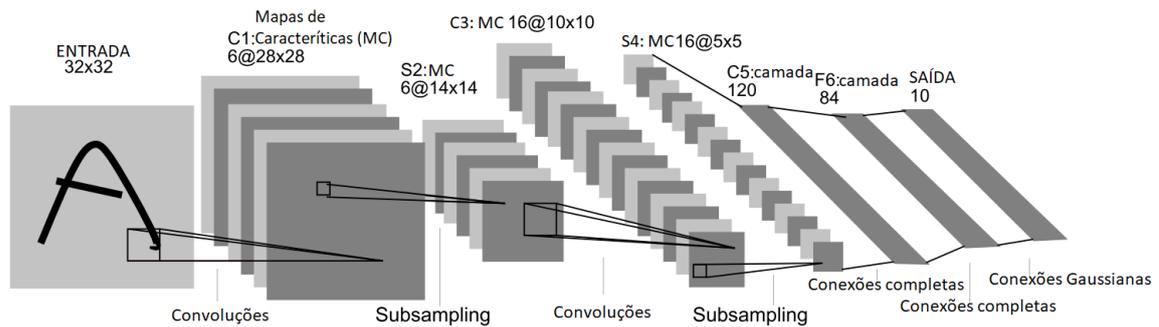


Figura 2.5: Arquitetura de uma rede neural convolucional (Adaptado de Lecun et al. (1998)).

das RNCs utilizadas nos dias atuais. Com a evolução da capacidade de processamento, as redes neurais convolucionais são projetadas com dezenas e até centenas de camadas (HE et al., 2015; SZEGEDY et al., 2015). RNCs com centenas de camadas podem levar dias ou semanas para ter seus treinamentos concluídos, dependendo do tamanho do *dataset*.

Desde 2012, as RNCs vêm evoluindo e obtendo resultados cada vez melhores. No ano de 2015, um modelo de RNC desenvolvido pelo time da Microsoft venceu o desafio ILSVRC, obtendo um percentual de 3.57% no Top-5 de erros, ou seja, em apenas 3.65% dos testes a classe correta não estava entre as cinco melhores classificadas entre as mil classes possíveis do desafio (HE et al., 2016).

Em termos de implementação, existem muitos *frameworks* que automatizam o treinamento de modelos de RNCs e oferecem APIs para realização de classificação e até mesmo para estendê-los. Os mais populares são Caffe, TensorFlow, Keras e Torch, porém existem outros, e novos *frameworks* são desenvolvidos com o passar do tempo (TOKUI et al., 2015).

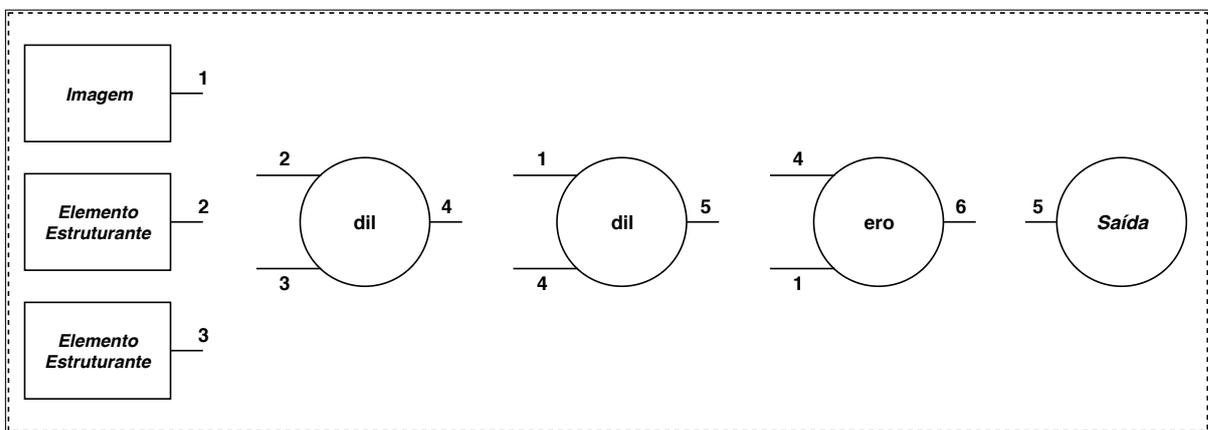
## 2.2 Otimização Multiobjetivo e Programação Genética

### 2.2.1 Programação Genética

O conceito de Programação Genética ou *Genetic Programming* (GP), como é conhecido hoje, foi apresentado em 1992 por Koza (1992). Em GP, a população é composta de programas representando indivíduos, com funções e símbolos terminais, de acordo com o problema a ser resolvido pelo algoritmo. Existem diferentes tipos de GP, como Programação Genética Linear, Programação Genética Baseada em Pilhas, Programação Genética Baseada em Grafos, Programação Genética Baseada em Árvores, entre outros (BRAMEIER; BANZHAF, 2007; PERKIS, 1994; POLI et al., 2008). Neste trabalho, a Programação Genética Cartesiana foi usada como parte de um método apresentado no Capítulo 4.

### 2.2.1.1 Programação Genética Cartesiana

A Programação Genética Cartesiana, em inglês *Cartesian Genetic Programming (CGP)*, é um tipo específico de Programação Genética baseada em Grafos, em que os cromossomos são representados por matrizes bidimensionais de vetores de inteiros, genes, que podem referenciar outros genes ou terminais como entradas e uma função para processar essas entradas e resultar em uma ou mais saídas (MILLER, 2011). Os cromossomos cartesianos podem ser representados como genótipos, um vetor unidimensional de inteiros (genes) para cada cromossomo e as saídas; ou fenótipos, que são o resultado da decomposição de um genótipo em um grafo direcionado, exibindo as conexões entre entradas, genes e saídas. Um cromossomo simples de um filtro morfológico é ilustrado na Figura 2.6.



**Figura 2.6: Exemplo de cromossomo cartesiano.**

O cromossomo cartesiano da Figura 2.6 é composto de três entradas (uma imagem e dois elementos estruturantes), três genes com operações de erosão (ero) e dilatação (dil) e uma saída. Com exceção da saída, a imagem, elementos estruturantes e genes podem ser referenciados pelos números à direita (ex.: a imagem de entrada é referenciada por 1, o primeiro gene por 4, etc.). Os genes, por sua vez, podem receber duas entradas, referenciando as entradas do cromossomo ou outros genes, e executam a operação informada em seu centro. A saída indica de qual gene será a saída final do cromossomo.

De forma geral, um cromossomo CGP deve ser construído de acordo com um número predefinido de linhas, colunas e *levels-back* (a quantidade de colunas anteriores que um gene pode referenciar como entrada). A população evolui por meio de mutações sem *crossover*, pois experimentos anteriores com recombinação via *crossover* mostraram que os resultados não são bons o suficiente para justificar seu uso em CGP (MILLER, 2011). A maioria das estratégias de mutação em CGP é baseada em mutação pontual ou *point mutation*, de acordo com a literatura (POLI et al., 2008). Essas estratégias são combinadas com a popular estratégia de evolução

para CGP conhecida como  $1 + \lambda = 4$ .

## 2.2.2 Otimização Multiobjetivo

Os algoritmos de Otimização Multiobjetivo ou *Multi-Objective Optimization* (MOO) visam resolver problemas envolvendo dois ou mais objetivos conflitantes, que devem ser minimizados ou maximizados, normalmente dentro de uma margem aceitável de *trade-off* entre cada um dos objetivos, para encontrar uma ou um conjunto de soluções factíveis que atendam todos os objetivos de forma satisfatória (BRANKE et al., 2008). Essas soluções são conhecidas como soluções ótimas de Pareto, um grupo de soluções com *trade-offs* claros e que permitem que os usuários escolham a melhor solução que resolva os problemas envolvidos na otimização. Nesta tese, um algoritmo evolucionário MOO foi usado, porém é importante destacar que existem outras abordagens de otimização multiobjetivo, como as interativas (MIETTINEN; RUIZ; WIERZBICKI, 2008). Um problema de otimização multiobjetivo pode ser apresentado de acordo com a Eq. 2.2.

$$\begin{aligned} & \text{Minimizar/Maximizar} \quad \{f_1(x), f_2(x), \dots, f_n(x)\} \\ & \text{sujeito à} \quad x \in S \end{aligned} \tag{2.2}$$

onde  $f_1, f_2$  e  $f_n$  são as funções de objetivo a serem minimizadas ou maximizadas,  $x$  é o vetor contendo  $x_1, x_2, \dots, x_n$  soluções e  $S$  é o espaço de busca ou região factível e suas restrições.

### 2.2.2.1 Algoritmo Genético de Ordenamento Não dominado – NSGAI

O Algoritmo Genético de Ordenamento Não dominado, originalmente *Nondominated Sorting Genetic Algorithm* (NSGAI), é um algoritmo evolucionário de otimização multiobjetivo proposto por Deb et al. (2002). Esse algoritmo aplica uma abordagem elitista para criar diversas fronteiras ou *fronts* não dominados e ordena-os internamente de acordo com sua densidade ou *crowding-distance*. Na Figura 2.7 é exibido o ciclo do NSGAI. A população  $R_t$  é composta pelos melhores indivíduos da geração anterior  $P_t$  mais os filhos gerados  $Q_t$ .

Os *fronts* não dominados são definidos por um conceito de Dominância, em que soluções podem dominar outras soluções enquanto podem ou não serem dominadas simultaneamente. Desta forma, dadas as soluções  $x$  e  $y$ , é dito que  $x$  domina  $y$  ( $x \prec y$ ) se, para cada função objetivo,  $x$  é igual ou melhor que  $y$  e, para pelo menos um objetivo,  $x$  é melhor que  $y$ . Com isso, uma Matriz de Dominância é criada e as soluções com as mesmas quantidades de dominância são agrupadas em *fronts* (ALBA; COTTA, 2017).

O Ordenamento de Densidade, ou *Crowding-distance*, é um método para estimar a densi-

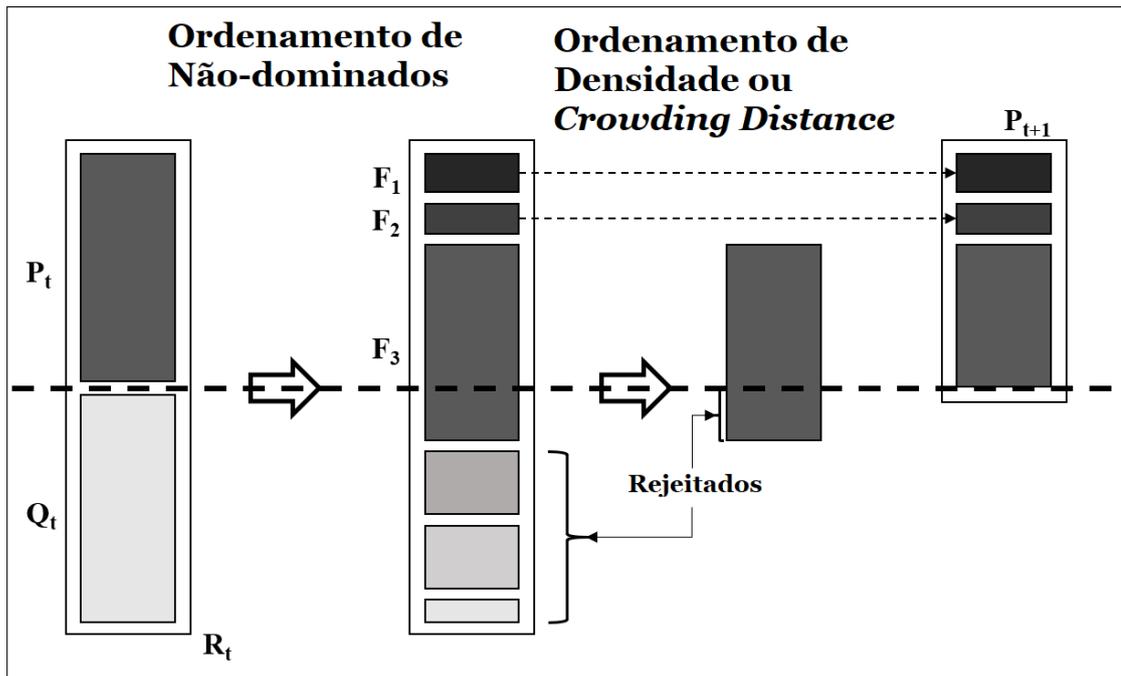


Figura 2.7: Ciclo do algoritmo NSGAI (Adaptado de Deb et al. (2002)).

dade de soluções de um *front* ao redor de cada uma delas. A estimativa é feita com o cálculo da distância de cada objetivo de uma solução e seus valores adjacentes e somando todas as distâncias normalizadas dos objetivos de uma solução para chegar ao valor final dela. Os maiores valores são atribuídos às soluções mais isoladas dentro do *front* para preservar a diversidade da população (ALBA; COTTA, 2017).

O NSGAI foi escolhido para ser adaptado devido à sua popularidade e também à facilidade de implementação. Além disso, em comparação preliminar com outros algoritmos, foi observado que o NSGAI permitia melhor adaptação do seu cromossomo de algoritmo genético original para um cromossomo de programação genética cartesiana. Alguns autores na literatura também apontaram que o NSGAI tende a apresentar melhores resultados com grandes populações e gera soluções com maior diversidade, o que é ideal ao se considerar que sistemas de navegação podem ser implantados em vários dispositivos com capacidades diferentes, e a maior diversidade pode produzir filtros morfológicos que atendam tanto placas de baixo processamento quanto dispositivos de estado da arte com SoCs de alto processamento.

## 2.3 Considerações Finais

Neste capítulo foram apresentados os principais conceitos que fazem parte dos sistemas de navegação para auxílio a PDVs. Oportunamente, foram citadas as técnicas aplicadas no desenvolvimento do sistema embarcado de navegação apresentado no Capítulo 5. No próximo

capítulo serão apresentadas algumas das abordagens encontradas na literatura e que se relacionam com este projeto, discutindo algumas de suas limitações que procuraram ser resolvidas nesta tese.

# Capítulo 3

## ESTADO DA ARTE EM SISTEMAS DE NAVEGAÇÃO PARA AUXÍLIO A PESSOAS COM DEFICIÊNCIAS VISUAIS

---

---

Neste capítulo será apresentado o estado da arte das diferentes abordagens da literatura sobre sistemas de navegação para PDV, seus objetivos, características e limitações, com foco na análise sobre como diferentes perfis de usuários podem ser atendidos nesses trabalhos.

### 3.1 Sistemas de Navegação e Classificação para PDVs

Os sistemas de navegação para PDV são aqueles que têm como objetivo auxiliar seus usuários a realizar percursos, planejados ou não, preocupando-se com questões como detecção de passagem segura e obstáculos. Também há a preocupação com a interação entre usuário e sistema, além da capacidade de customização do mesmo. Por fim, as plataformas nas quais são implementadas esse tipo de sistema são uma preocupação, quando se leva em consideração o desempenho do processamento de imagens.

Mocanu, Tapu e Zaharia (2015) utilizam a clássica técnica de extração de características para a classificação de obstáculos dinâmicos (pedestres, veículos e bicicletas) e estáticos (cercas, postes, árvores, lixeiras, semáforos, galhos de árvore, degraus, entre outros), com uma adaptação do método *Histogram of Oriented Gradients* (HOG) e representam globalmente as imagens por técnicas como *Locally Aggregated Descriptor* (VLAD) e *Bag of Visual Words*(BoVW)(DALAL; TRIGGS, 2005; JÉGOU; DOUZE; SCHMID, 2011; CSURKA et al., 2004). As imagens são classificadas por um método de ranking próprio em conjunto ao *Support Vector Machines* (SVM)(TONG; CHANG, 2001). A implementação é feita em um smartfone

e o processamento alcança, no máximo, 5fps (*frames per second* ou quadros por segundo). É importante ressaltar que, embora 5fps seja uma performance aceitável para pessoas com deficiência visual, os autores não propuseram a indicação de caminhos seguros e o autor não deixa claro como o deficiente visual é informado sobre os obstáculos ou se o sistema é capaz de detectar e/ou diferenciar obstáculos suspensos de obstáculos no chão.

Deb et al. (2013) desenvolveram uma abordagem que visa predominantemente a navegação segura do deficiente visual pela estimação da direção segura em que este pode caminhar. No projeto, os autores utilizam uma câmera simples para a aquisição de imagens e aplicam técnicas como detecção de bordas e segmentação piramidal para então definir uma zona segura utilizando *Template Matching* e transmitir, via tons musicais, se o caminho seguro está à direita ou à esquerda. A motivação do uso de tons musicais não é apresentada. Segundo os autores, a área verificada é central e localizada na parte inferior da imagem. Além de não alertar o usuário de possíveis obstáculos acima do chão ou classificá-los, a ideia proposta considera apenas a área inferior central e despreza grande parte dos lados esquerdo e direito da imagem, que podem ser alternativas de rotas em alguns casos. Por fim, os autores também afirmam que, em caso de dois caminhos, o sistema sempre escolherá o lado esquerdo, sequer apresentando periodicamente as alternativas para o deficiente visual.

Dingli e Mercieca (2016) empregam duas técnicas para a classificação de objetos para pessoas com deficiência visual, comparando-as. A primeira técnica faz uso de redes neurais convolucionais com o modelo AlexNet pré-treinado. A segunda técnica utiliza descritores SIFT e SURF para extrair características do *dataset* das cenas e a classificação é realizada pelo algoritmo FLANN. Como esperado, a rede neural obteve acurácia maior que o método tradicional, com os descritores e o algoritmo FLANN. Os autores, porém, não especificaram o hardware utilizado e apenas mencionaram que o processamento é realizado em um servidor, necessitando um canal de comunicação.

Aladren et al. (2014) trabalham especificamente a segmentação de caminhos livres. Sua abordagem utiliza câmera RGB-D e se baseia tanto na nuvem de pontos de profundidade quanto na imagem RGB-D para extrair a imagem apenas do chão e definir se há um obstáculo no caminho e, em caso positivo, se há um caminho livre ou formas de evitá-lo. Os autores também utilizam a mesma técnica piramidal que Deb et al. (2013), em conjunto com a nuvem de pontos para separar obstáculos de caminho livre. O *feedback* é feito de forma vocal para o usuário, de acordo com a segmentação. A direção também é mapeada em som estéreo 3D, com a diferença de frequências para os intervalos de distâncias de 30 cm. As técnicas dos dois autores são similares à empregada na segmentação de caminho livre deste trabalho, com a diferença que

neste trabalho a distância máxima pode ser escolhida pelo usuário e não há a busca na imagem toda, ou seja, não utiliza segmentação piramidal, mas somente o quarto inferior da imagem apenas de profundidade.

Abboud et al. (2014) não fazem a segmentação de caminhos, porém propõem uma nova forma de representar a forma e a cor de objetos com o projeto EyeMusic. O sistema realiza o agrupamento de cores dos objetos em seis: preto, azul, verde, vermelho, branco e amarelo, em que cada cor é representada por um tipo distinto de instrumento e a cor preta é o silêncio. A forma dos objetos, por sua vez, é determinada pelo tom do som e sua iluminação, pelo volume. A abordagem de *feedback* via instrumentos musicais propicia, segundo os autores, uma experiência mais agradável no uso de um sistema de navegação. Embora os autores tenham feito um estudo para determinar quão agradável pode ser o *feedback* em relação a outros trabalhos semelhantes, o áudio resultante do sistema é reproduzido para cada coluna da imagem, o que torna impraticável o uso dessa abordagem em um ambiente que necessita de respostas em tempo real.

O conceito de sistemas embarcados é explorado por Bangar, Narkhede e Paranjape (2013) no reconhecimento de objetos e suas cores, fazendo uso de uma placa PCM-9550F que embarca um processador de 266MHz e 128MB de memória. Nesse trabalho, os autores realizam a extração das imagens de um sensor de vídeo acoplado em um óculos e realizam operações como extração de *background*, detecção de bordas e *pixel clustering*, para determinar um objeto e também sua cor, por meio dos esquemas de cores, e então informa, via áudio estéreo, o resultado do processamento ao usuário. A detecção de cor, após a detecção de um objeto, é um fator importante para a classificação, porém os autores não realizam a classificação completa do objeto e, embora mencionem áudio estéreo, não citam em seu trabalho se o áudio é construído de forma a indicar a direção do objeto. O trabalho também não visa ou elucida a possibilidade de segmentação de caminhos livres.

Kumar et al. (2013) propuseram um sistema embarcado para a detecção de pedestres usando uma webcam, fones de ouvido, sensor ultrassônico e uma placa eBox 2300. O sensor é o responsável por detectar a presença de um obstáculo, enquanto a webcam faz a captura da imagem para a classificação de um pedestre baseando-se em sua face e padrões da roupa. O sistema é bastante limitado, haja vista que não há a preocupação com caminhos, por exemplo.

Poggi e Mattoccia (2016) também utilizam sistemas embarcados para a implantação de um sistema inteligente para pessoas com deficiência visual. Em seu trabalho, os autores usam uma placa Odroid U3 em comunicação com um smartfone, fones de ouvido, luva de *feedback* tátil e óculos com câmera RGB-D e classificam obstáculos utilizando redes neurais convolucionais,

mas sem a aceleração de uma GPU. Há navegação somente via GPS do smartfone e o sistema proposto prevê a segmentação de caminhos, apenas com a classificação de obstáculos e seus posicionamentos ou informando ao usuário as indicações do GPS. A interação entre PDV e sistema, porém, se limita a comandos de voz do usuário e não há a opção de outro meio. O *feedback* oferecido pelo sistema também não é adaptável, uma vez que os obstáculos são notificados apenas via vibração nas luvas e o caminho indicado por mensagens de voz no smartfone.

Patil et al. (2014) propuseram um sistema de navegação e localização customizável para ambientes fechados baseado em smartfones Android e triangulação dos sinais de emissores *bluetooth* para definição de rotas de acordo com mapas predefinidos. Sua estrutura é composta de um módulo *desktop*, no qual um administrador é responsável por configurar e distribuir os emissores, determinando a ligação entre eles, e um módulo *bluetooth*, que calcula a posição do usuário com três, dois ou um emissor ao alcance. Após determinar a origem e o destino de sua rota, o módulo *desktop* calcula a rota mais rápida com o algoritmo de Dijkstra. Entretanto, essa abordagem não discorre sobre como o usuário recebe o *feedback* e especifica apenas que a customização do sistema é dada pela possibilidade de escolher diferentes mapas para navegar, com diferentes pontos de interesse (ex: mapas para funcionários e mapas para clientes de uma loja), mas sem detalhar como é feita a escolha.

Priyadarshana e Wimalaratne (2014) abordaram a questão da personalização/customização da experiência do usuário em tecnologias vestíveis para PDVs. Em seu trabalho, foi proposto um módulo em forma de cinto, composto de sensores ultrassônicos, um Arduino Mega e motores de vibração, e duas pulseiras contendo motores de vibração e módulo *bluetooth*. Com o objetivo de detectar obstáculos, as informações do ultrassom são enviadas para o Arduino, que classifica o obstáculo entre degraus, escadas, obstáculo frontal, obstáculo à esquerda, obstáculo à direita e envia padrões de vibração para o cinto ou as pulseiras, de acordo com o obstáculo. A customização do sistema é feita por um aplicativo Android e contém poucas opções, como distância máxima, número de vibrações de *feedback*, altura para degrau, necessitando sempre da ajuda de uma pessoa com visão para fazer as customizações, tirando a autonomia de PDVs para fazerem as customizações de forma autônoma.

Li et al. (2018) conduziram aprimoramentos sucessivos em seu projeto chamado ISANA, um sistema de navegação e localização para ambientes internos que utiliza *tablets* com Android Tango e conceitos de mapeamento semântico. Nessa abordagem, o foco do sistema está no contexto do ambiente onde o usuário está inserido e para onde ele quer ir. Também é possível fazer solicitações ao sistema via comandos de voz e fazer perguntas sobre sua localização, indicar destino, entre outros. O ISANA também possui dois tipos de *feedback*, via voz e via vibração,

para diferentes tarefas e foi criado a partir de um perfil de usuário resultante de questionários de preferências e necessidades aplicados em PDVs. Apesar de ser um projeto bastante sofisticado, capaz de classificar contextualmente uma cena, o ISANA não fornece outra forma de interação além da fala e, principalmente, não suporta mudanças de preferências de usuário durante seu uso.

Siddhartha, Chavan e Uma (2018) criaram protótipo vestível em forma de jaqueta inteligente, equipada com sensores ultrassônicos na frente e nas costas que enviam seus dados para um módulo Arduino UNO 328. Nesse trabalho, apenas obstáculos são detectados e o *feedback* é enviado via *bluetooth* para o smartfone do usuário, que reproduz em comando de fala a direção do obstáculo. O projeto, porém, ainda é muito incipiente e não oferece outros recursos ou possibilidade de customização a partir do usuário.

Chen et al. (2019) usam os recursos dos smartfones como base de seu sistema de navegação especializado em detecção de obstáculos em percursos guiados por GPS. A distância dos obstáculos é calculada por um método baseado em visão monocular criado pelos autores e a classificação utiliza redes neurais profundas com a rede SSD-MobileNetV2. O usuário pode interagir com o sistema via voz enquanto o *feedback* é processado e enviado a ele através de fala ou tons, indicando a direção recebida do GPS, a distância do obstáculo e o nível de risco do obstáculo que pode ser pedestres, bicicletas, bicicletas elétricas, motos e carros. O risco é informado por tons sonoros com níveis diferentes e que aumentam de acordo com o risco e, caso haja mais de um obstáculo, os três maiores riscos serão reportados. A abordagem de riscos é bastante relevante no contexto de PDVs, mas o sistema não oferece outro meio de *feedback* que não o sonoro e nenhuma opção de customização do sistema, forçando o usuário a se adaptar ao funcionamento do mesmo.

Rahman, Ullah e Ullah (2019) também desenvolveram um sistema de navegação voltado para smartfones baseado em visão monocular, mas com o objetivo de indicar caminhos livres a partir das imagens do chão. Seu método funciona com a comparação das imagens de referência do chão, sem obstáculos, com o chão obstruído com um ou mais obstáculos comuns. Em um cenário real, o método não vai funcionar quando houver mudanças contínuas nos padrões visuais do chão, como, por exemplo, calçadas com diferentes cores.

Ou, Park e Lee (2019) apresentaram uma arquitetura robusta para navegação de PDVs usando um aplicativo Android para fornecer coordenadas GPS, um servidor de processamento e a Raspberry Pi, equipada com uma câmera, para capturar as imagens do ambiente e enviar *feedback* de áudio. Com as coordenadas GPS do smartfone Android, há a troca de mensagens com a Raspberry que comunica, via fala, sobre as direções para o usuário. O sistema também

faz o reconhecimento de obstáculos usando redes neurais (SSD-MobileNetV1 e Faster R-CNN), processa as coordenadas de GPS e identifica semáforo de pedestres, processando tudo em um servidor ligado à Raspberry e ao smartfone Android. Os autores não esclarecem o tempo de comunicação entre os dispositivos, principalmente no ciclo de envio de imagens ao servidor e recebimento de resposta para *feedback*. A dependência de um agente externo ao hardware local pode ocasionar o não funcionamento do sistema nesse caso.

Kommey, Herrman e Addo (2019) desenvolveram óculos compostos de câmeras frontais e traseiras, fones de ouvido, botões e módulo de processamento com microcontrolador. Os obstáculos presentes na frente e atrás do usuário são detectados e classificados dentro de uma distância de até quatro metros por sistema de detecção de objetos chamado YOLO9000. O resultado é sintetizado em fala e enviado para os fones de ouvido. É um projeto promissor, dado seu tamanho reduzido e capacidade de processamento, mas os autores não deixam claro se todo o processamento é feito nos próprios óculos ou se há a comunicação com um agente externo para sintetizar falas e classificar obstáculos, já que o módulo de processamento, segundo os autores, consta apenas de um microcontrolador ATmega328. Além disso, a detecção de caminhos livres também não é feita pelos óculos, limitando-se apenas à classificação.

Adi, Kusuma e Attamimi (2019) fizeram uso da câmera estereoscópica ZED para a detecção dos obstáculos mais próximos, tanto em ambientes abertos quanto fechados. Com o processo de *downsampling* das imagens de profundidade, os autores detectam a região do pixel que indica o obstáculo mais próximo e processam uma resposta em áudio direcional da esquerda para a direita, de acordo com a direção do obstáculo. O uso de áudio direcional pode ser problemático para usuários que também têm baixa audição e essa é a única forma de *feedback* proposta pelos autores. Não apenas isso, mas o sistema depende de um laptop para o seu funcionamento, o que é inviável para esse tipo de sistema pela dependência de energia devido à baixa duração de bateria, ao peso, à usabilidade, etc.

Donati et al. (2020) desenvolveram uma bengala inteligente que se conecta a um aplicativo Android do smartfone do usuário para guiar PDVs em rotas predefinidas dentro das muralhas históricas da cidade de Lucca, na Itália. A bengala possui botões para interação entre PDV e o sistema, com ações para solicitar mais informações de diferentes aspectos das redondezas da posição do usuário, que é monitorada via GPS. A interação por voz também é oferecida, porém os autores não deixam claro se o usuário pode fazer, via voz, todas as interações dos botões. O *feedback* das rotas é feito via vibração da bengala para sinalizar o caminho seguro predefinido do percurso, variando a intensidade da vibração de forma inversamente proporcional à proximidade do centro do caminho seguro, permitindo que PDVs saibam quando estiverem saindo da

rota. Há também o *feedback* vocal, que envia informações mais específicas sobre o local e as distâncias dos pontos de interesse, caso o usuário solicite. Há a possibilidade de customização de cinco parâmetros no aplicativo Android, porém apenas dois deles dizem respeito à experiência de navegação do usuário e, da forma que os autores apresentaram, nenhum dos parâmetros pode ser alterado diretamente por PDVs.

Como foi possível observar nos trabalhos que compõem o estado da arte em sistemas de navegação para PDVs, existem diversos esforços de pesquisa em variados tópicos, com o objetivo comum de beneficiar usuários que não podem contar com a visão. Um dos tópicos, entretanto, que tem sido pouco explorado na literatura, é o de customização ou personalização autônoma do funcionamento desses sistemas de acordo com as necessidades individuais de PDVs durante o uso. Um comparativo entre as características dos trabalhos citados anteriormente é apresentado na Tabela 3.1.

**Tabela 3.1: Técnicas empregadas nos trabalhos da literatura.**

	DCL	DO	CO	FA	FT	RF	IT	IV	RI	C	V
Mocanu, Tapu e Zaharia (2015)		✓	✓								✓
Deb et al. (2013)	✓			✓							
Dingli e Mercieca (2016)			✓								
Aladren et al. (2014)	✓	✓		✓							✓
Abboud et al. (2014)		✓	*1	✓							✓
Bangar, Narkhede e Paranjape (2013)		✓	*1	✓							✓
Kumar et al. (2013)		✓	✓	✓			✓				✓
Poggi e Mattochia (2016)	✓	✓	✓	✓	✓			✓			✓
Patil et al. (2014)										*2	
Priyadarshana e Wimalaratne (2014)		✓	✓		✓					*2	✓
Li et al. (2018)	✓	✓	✓	✓	✓			✓			✓
Siddhartha, Chavan e Uma (2018)		✓		✓							✓
Chen et al. (2019)		✓	✓	✓				✓			
Rahman, Ullah e Ullah (2019)	✓			✓							
Ou, Park e Lee (2019)		✓	✓	✓							
Kommey, Herrman e Addo (2019)		✓	✓	✓			✓				✓
Adi, Kusuma e Attamimi (2019)		✓		✓							
Donati et al. (2020)				✓	✓		✓	✓		*2	✓
Esta Tese	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Legenda: Detecção de Caminhos Livres (DCL), Detecção de Obstáculos (DO), Classificação de Obstáculos (CO), *Feedback* Acústico (FA), *Feedback* Tátil (FT), Redundância de *Feedbacks* (RF), Interação Tátil (IT), Interação por Voz (IV), Redundância de Interação (RI), Customizável (C) e Vestível (V).

\*1 — Classificação apenas de cores.

\*2 — Customização parcial com auxílio de terceiros.

## 3.2 Considerações Finais

Neste capítulo foram apresentadas diversas abordagens de sistema de navegação para pessoas com deficiências visuais. Além dos problemas citados no decorrer do capítulo, é possível observar que os trabalhos não permitem a customização das preferências do usuário em tempo real, durante o uso do sistema de navegação. A literatura também não contém trabalhos com mais de um método de entrada específica do usuário para interagir com o sistema de navegação ou representa as mesmas informações por métodos diferentes de *feedback*, a critério de seu usuário. Assim, com base nessas limitações, foi desenvolvido um protótipo vestível contendo um sistema embarcado de navegação que permite ao usuário interagir com ele de duas maneiras diferentes, para modificar seu funcionamento de acordo com as próprias preferências individuais. Por trabalhar com imagens de profundidade, o protótipo vestível precisa lidar com um problema comum existente nesse tipo de imagem: as áreas de distâncias desconhecidas. No próximo capítulo será apresentado o método NSGA2CGP, que foi criado com o intuito de gerar automaticamente filtros morfológicos otimizados para corrigir imagens de profundidade.

# Capítulo 4

## MÉTODO DE OTIMIZAÇÃO MULTIOBJETIVO EVOLUCIONÁRIO E CARTESIANO PARA CORREÇÃO DE IMAGENS DE PROFUNDIDADE

---

---

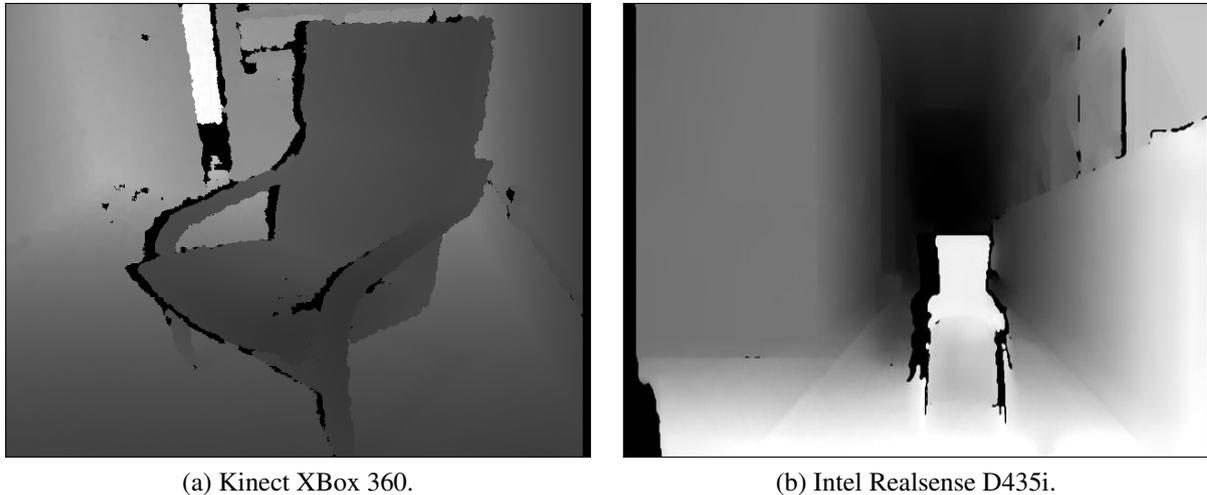
Neste capítulo será apresentado o "Método de Otimização Multiobjetivo Evolucionário e Cartesiano para Correção de Imagens de Profundidade", denominado NSGA2CGP. Este método tem como objetivo gerar e otimizar filtros morfológicos capazes de corrigir imagens de profundidade capturadas por câmeras RGB-D, que apresentam áreas sem distância estimada. O método é resultado da adaptação do algoritmo NSGA-II com a população composta de cromossomos de Programação Genética Cartesiana, realizando a otimização de dois objetivos: minimização de erro em relação aos resultados esperados e minimização da complexidade do filtro por meio dos tamanhos dos elementos estruturantes.

### 4.1 Imagens de Profundidade

Os sistemas de navegação para PDV baseados em visão computacional costumam ter como principal dispositivo de entrada de dados as câmeras RGB-D. Tais câmeras fazem a captura de imagens coloridas, mas também são capazes de capturar os dados de distância sobre cada objeto em um ambiente. Como mencionado na Seção 2.1.1.1, existem diversas técnicas que podem ser encontradas em câmeras RGB-D para estimar a distância de um ponto, resultando em imagens de profundidade em escala de cinza, representando as distâncias por meio de variações dos tons.

Independentemente da técnica de estimação de distância, todas as câmeras RGB-D possuem um problema em comum, em relação às imagens de profundidade e, conseqüentemente, à estimação de distâncias, que são áreas em que não foi possível estimar a distância. Com isso,

as imagens de profundidade capturadas apresentam regiões com distância nula ou zero, fazendo com que os pixels dessas regiões representados em escala de cinza sejam da cor preta. Na Figura 4.1 são exibidas duas imagens de profundidade de duas câmeras diferentes.



**Figura 4.1: Exemplos de imagens de profundidade em escala de cinza.**

As regiões sem distâncias estimadas são potencialmente prejudiciais para o funcionamento dos sistemas de navegação, fazendo com que o processo de segmentação de objetos e/ou caminhos livres não seja confiável, podendo, conseqüentemente, colocar os usuários em perigo de quedas ou colisões. A correção dessas áreas de desconhecidas distâncias pode ser feita com algumas abordagens encontradas na literatura, como *Inpainting*, *Bi-lateral Filter*, *Recursive Median Filter* e *Hole Filling* (KEAOMANEE; YOUNGKONG, 2019; RAVIYA et al., 2019; TELEA, 2004; LAI et al., 2011; VARSHNEY; TYAGI, 2015). Entretanto, tais métodos não levam em consideração as possíveis e diferentes capacidades de hardware que podem implementá-los, fazendo com que suas complexidades computacionais sejam fatores limitantes para garantir performance satisfatória em execuções em tempo real. Abordagens baseadas em filtros morfológicos já foram propostas antes na literatura, mas também sem levar em consideração a complexidade do filtro e a diversidade de dispositivos que poderiam usar tais filtros (TI et al., 2017). Desta forma, o método NSGA2CGP foi criado para gerar e otimizar automaticamente filtros morfológicos levando em consideração não somente a qualidade, mas também a complexidade dos filtros. O método é baseado na união da Programação Genética Cartesiana e o algoritmo de otimização multiobjetivo NSGA-II.

Dentro do escopo desta tese, o desenvolvimento do método NSGA2CGP supre uma necessidade encontrada durante a criação do sistema de navegação a ser apresentado no Capítulo 5. Por ser um sistema baseado em visão computacional, o sistema depende de imagens de profundidade fornecidas por uma câmera RGB-D e, portanto, há a necessidade de correção de

imperfeições presentes em tais imagens. Assim, o método NSGA2CGP foi proposto como uma solução para gerar filtros de pré-processamento de imagens de profundidade a serem utilizados dentro do sistema de navegação desenvolvido.

## 4.2 Visão Geral

O "Método de Otimização Multiobjetivo Evolucionário e Cartesiano para Correção de Imagens de Profundidade", também chamado de NSGA2CGP, foi baseado na combinação entre o algoritmo de otimização multiobjetivo NSGA-II e a programação genética cartesiana. O NSGA-II foi adaptado para utilizar, ao invés de um algoritmo genético clássico, a programação genética cartesiana para compor sua população. Seu objetivo é gerar e otimizar, de forma automática, filtros morfológicos que corrijam imagens de profundidade visando minimizar a relação *erro x complexidade*, de modo a oferecer diversas soluções com *trade-offs* flexíveis que permitam a escolha de filtros adequados à capacidade do hardware alvo e suas limitações (poder de processamento, capacidade de bateria, etc.). Desta forma, os objetivos de otimização do método são minimizar o erro entre saída obtida e saída esperada e minimizar a somatória dos tamanhos dos elementos estruturantes do filtro. O método é composto de cinco fases, que vão desde a inicialização dos parâmetros de execução até a geração de código pronto para ser implantado. Na Figura 4.2 são apresentadas as cinco fases: Inicialização, Evolução, Apresentação, Visualização e Exportar/Implantar.

O NSGA2CGP foi implementado no MATLAB e Octave em um ambiente misto com Windows 10 e Linux Mint 19.1, porém o método é independente de linguagem de programação ou sistema operacional. Isso permite que implementações futuras sejam feitas utilizando *frameworks* e *toolkits* que, por exemplo, acelerem a execução da otimização dos filtros em GPUs ou construam uma interface gráfica para auxiliar nas configurações de execução do método.

## 4.3 Inicialização

A fase de Inicialização contempla todas as tarefas necessárias para preparar a população inicial para ser otimizada por meio dos processos evolutivos. Essa fase é iniciada com o recebimento e análise dos parâmetros de execução definidos pelo usuário e que guiarão todo o funcionamento do NSGA2CGP. No Quadro 4.1 são apresentados os parâmetros de execução, que serão discutidos no decorrer deste texto, e seus tipos básicos. Os parâmetros, após inicializados, são armazenados em uma estrutura paramétrica que é acessível por todas as fases do

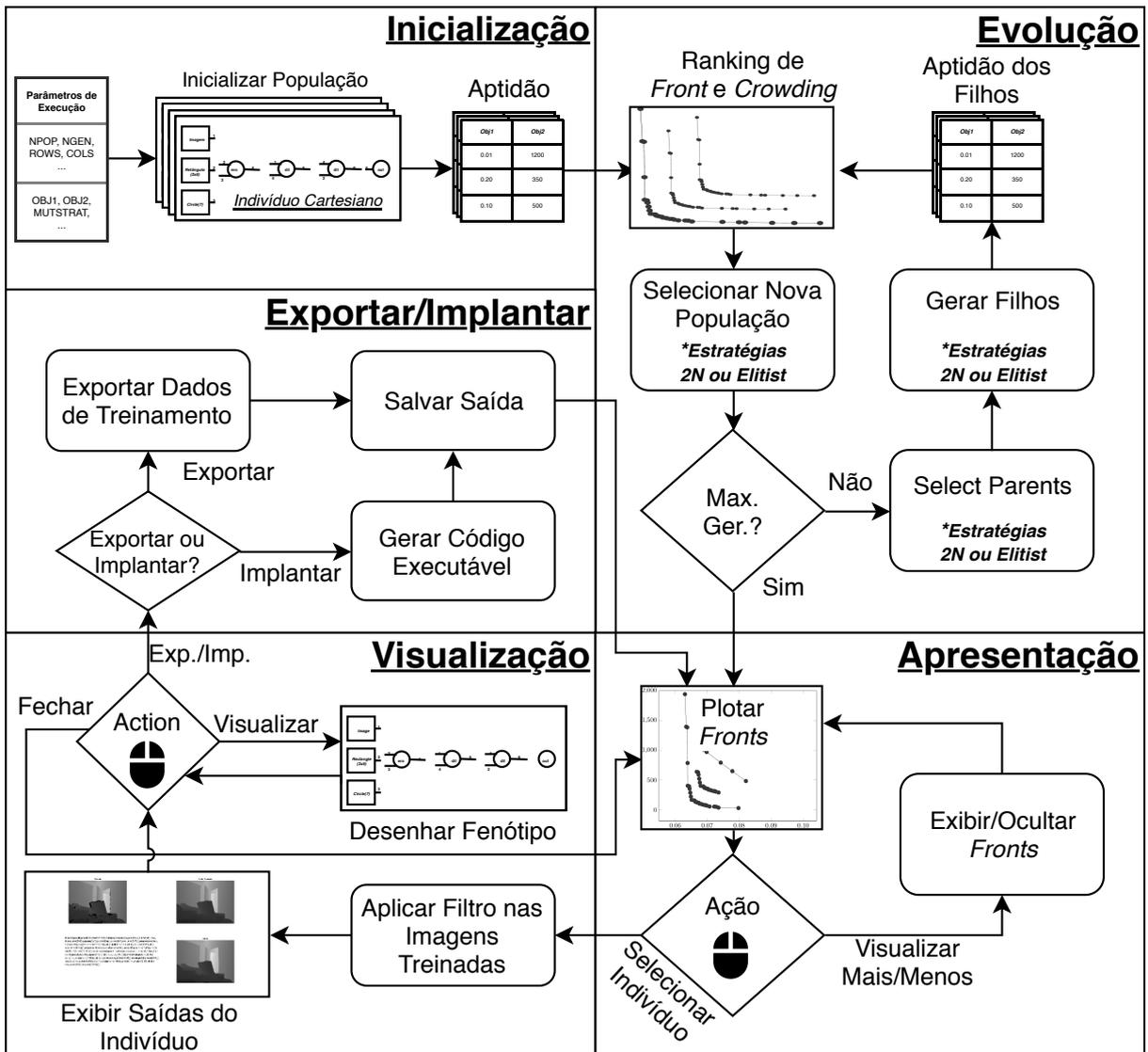


Figura 4.2: Representação da estrutura do método NSGA2CGP.

método.

Nessa fase, uma nova população é criada através da geração aleatória de indivíduos ou é utilizada uma população pré-treinada como população inicial, importando os dados de treinamento armazenados no caminho especificado pelos parâmetros *INITPOP*. Além disso, caso um treinamento tenha sido interrompido, o parâmetro *RESUTRAIN* pode ser utilizado para indicar o caminho de um arquivo de estado de treinamento que tenha sido salvo pelo parâmetro *SAVESTATE*, fazendo com que a otimização dos filtros continue de onde foi interrompida. No Algoritmo 1 é descrito, de forma sucinta, o funcionamento da fase de Inicialização.

Após a leitura e inicialização dos parâmetros de execução na linha 1, o Algoritmo 1 verifica se a população deve ser carregada de uma pré-existente, se o treinamento está retomando a partir de uma interrupção anterior ou se a população deve ser gerada totalmente (linhas 2-8).

**Quadro 4.1: Parâmetros de Execução do NSGA2CGP.**

Parâmetro	Descrição	Parâmetro	Descrição
NGEN	(int) Número máximo de gerações.	TRAINDATA	({Mat},{Mat}) Par de <i>arrays</i> contendo imagens de entrada e suas saídas esperadas.
NPOP	(int) Tamanho da população.	STRELS	({SE1,...,SEn}) <i>Array</i> de elementos estruturantes.
COLS	(int) Número total de colunas do cromossomo CGP.	ERROR	(string) Função de aptidão para avaliar os cromossomos.
ROWS	(int) Total de linhas do cromossomo CGP.	CODEWRPS	({@function}) <i>Wrappers</i> usados para gerar código executável para cada linguagem.
LBACK	(int) <i>Levels-back</i> dos genes.	USELOGICS	(logical) Permite o uso de operadores lógicos como AND/OR/NOT.
TMUT	(float) Taxa de mutação.	GENTYPE	(string) Define como os genes recebem as entradas.
MUTSTRAT	(string) Estratégia de mutação.	BPWEIGHT	(logical) Permite aplicar penalidade de pixels pretos.
INITPOP	(string) Usar população pré-treinada em um novo treinamento.	RESUTRAIN	(string) Resume um treinamento parado anteriormente.
GENESEL	(string) Estratégia de escolha de gene para mutação.	SAVESTATE	(int) Salvar o estado de treinamento a cada $n$ gerações.

Em seguida, são calculados os objetivos para cada indivíduo da população, na linha 10.

Com o processo de criação de uma nova população, são gerados um número  $NPOP$  de indivíduos/cromossomos aleatoriamente de acordo com o parâmetro de execução. Cada cromossomo é estruturado em um *array* unidimensional de tamanho  $COLS * ROWS + 1$ , sendo cada posição um gene e a última, a saída do cromossomo. Cada gene, por sua vez, é composto de três números inteiros, que representam a operação a ser executada e dois argumentos de entrada do gene, respectivamente, como exibido na Figura 4.3 com a representação do fenótipo e genótipo, sendo esse último um *array* de inteiros.

---

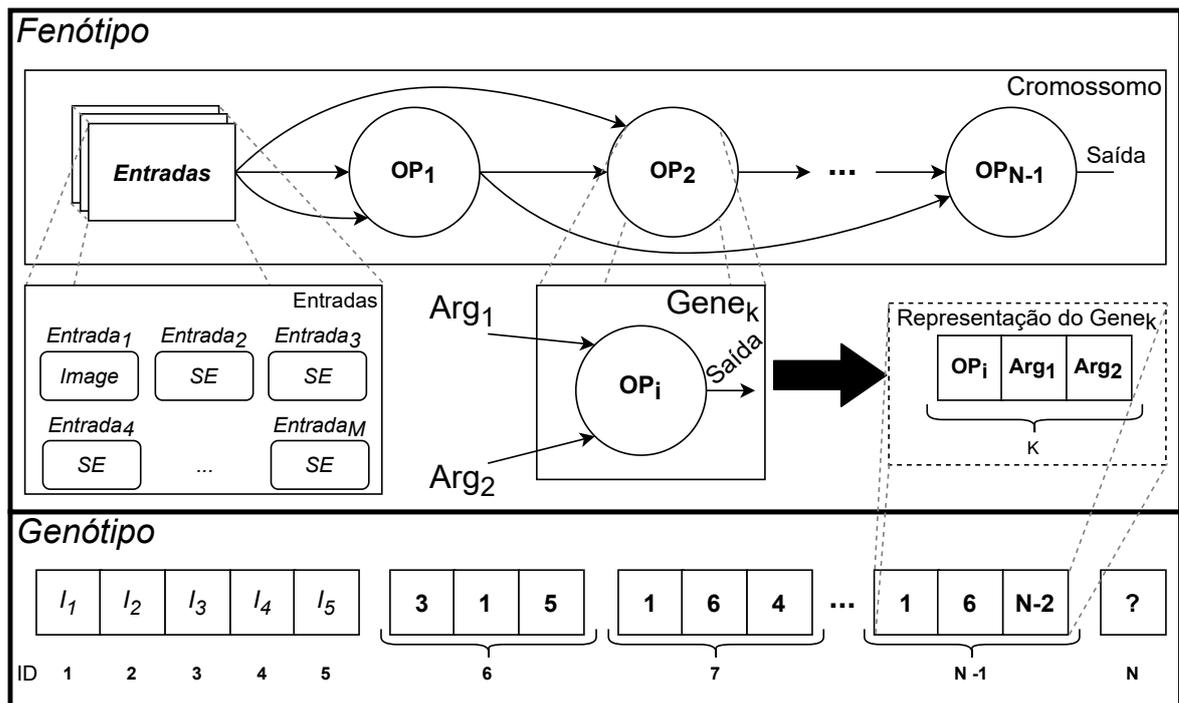
**Algoritmo 1:** Fase de Inicialização do NSGA2CGP

---

**Entrada:** EXEC\_PARAMS  
**Saída:** {POPULATION, FITARRAY}

- 1 PARAMS  $\leftarrow$  parseInput(EXEC\_PARAMS)
- 2 **if** INITPOP **then**
- 3 | POPULATION  $\leftarrow$  loadTrainedPopulation(INITPOP)
- 4 **else if** RESUTRAIN **then**
- 5 | PARAMS  $\leftarrow$  loadExecutionState(RESUTRAIN)
- 6 | POPULATION  $\leftarrow$  loadPopulationState(RESUTRAIN)
- 7 **else**
- 8 | POPULATION  $\leftarrow$  generateRandomPop(PARAMS)
- 9 **end**
- 10 FITARRAY  $\leftarrow$  calculateFitness(POPULATION,PARAMS)

---



**Figura 4.3: Representação do fenótipo e genótipo de um cromossomo cartesiano.**

O fenótipo (Figura 4.3a) segue o formato apresentado anteriormente na Figura 2.6, com as ligações entre as entradas, operações/genes ( $OP_1$ ,  $OP_2$ ,  $OP_{N-1}$  e Saída). As entradas, por sua vez, são compostas de uma imagem e uma série de elementos estruturantes (Figura 4.3b), enquanto as operações são compostas de duas entradas, o operador e uma saída (Figura 4.3c). Na Figura 4.3d, o cromossomo é representado por seu genótipo, tendo suas entradas como elementos unitários no início do *array*, operações como trios de inteiros e a saída como um elemento único no final do *array*.

Os genes de um cromossomo podem assumir dois tipos, de acordo com o valor do parâmetro *GENTYPE*. O tipo *MorphPair* impõe uma restrição aos genes, fazendo com que suas entradas sejam sempre compostas da imagem de entrada/treinamento ou a saída de outro gene, também imagem, como primeira entrada e um elemento estruturante do parâmetro *STRELS*, como segunda entrada. O segundo tipo, *Free*, não impõe qualquer restrição às entradas dos genes, permitindo que um gene receba tanto imagens (IMG) quanto elementos estruturantes (SEL) em qualquer combinação como entrada e aplicando sua função de acordo com o Quadro 4.2.

Por fim, além das operações morfológicas de dilatação, erosão e operadores lógicos, os genes também podem executar a operação *NOP*, um operador de repasse que simplesmente envia como saída a entrada 1 ou entrada 2, se o número do gene for ímpar ou par, respectivamente.

Quadro 4.2: Modos de operação do tipo de gene *Free*.

Entrada 1	Entrada 2	Dil & Ero	AND, OR, etc.	NOT
IMG	SEL	Aplicar	Replicar SEL e aplicar em IMG	Aplicar a IMG
SEL	IMG	Aplicar	Replicar SEL e aplicar em IMG	Aplicar a IMG
IMG	IMG	Repassar 1 se gene ímpar, 2 caso contrário	Aplicar	Aplicar em 1 se gene ímpar, 2 caso contrário
SEL	SEL	Aplicar	Aplicar	Aplicar em 1 se gene ímpar, 2 caso contrário

## 4.4 Evolução

O processo de Evolução se inicia após a formação da população inicial e o cálculo de suas aptidões, seja um novo treinamento ou a continuação de um interrompido. A fase de Evolução repete-se pela quantidade de vezes definida no parâmetro *NGEN*, tendo seu progresso salvo a cada número de gerações informado pelo parâmetro *SAVESTATE*.

A população, juntamente com seus dados de aptidões, é processada em relação à dominância de cada indivíduos sobre outro, resultando em uma matriz de dominância. Com ela, são definidos os *fronts* não dominados, sendo que as melhores soluções para aquela geração serão colocadas no primeiro *front*, ou seja, no *front* com indivíduos que não são dominados por ninguém e, teoricamente, possuem as melhores relações de erro e complexidade. Uma vez definidos, cada um dos *fronts* é submetido a um processo de ranqueamento interno por meio de um operador de densidade chamado *crowding distance* ou distância de aglomeração, atribuindo melhores valores aos indivíduos de regiões menos aglomeradas como forma de preservar a diversidade.

O método NSGA2CGP prevê duas estratégias de mutação como resultado da junção entre NSGA-II e programação genética cartesiana. A escolha entre elas é definida pelo parâmetro *MUTSTRAT* entre a estratégia '2N' ou 'ELITIST'. No Algoritmo 2 é apresentada uma descrição da fase de Evolução.

De acordo com o Algoritmo 2, a população é ranqueada em *fronts* de acordo com os valores dos seus objetivos (linha 3) e, em seguida, internamente organizada pela *crowding-distance* (linha 4). O algoritmo verifica se a estratégia de mutação é 2N ou *Elitist* e as aplica na população para gerar os filhos (linhas 6 e 15). Essa rotina é repetida *NGEN* vezes. Caso a estratégia de mutação seja 2N, a fase de Evolução ainda seleciona os *NPOP* melhores indivíduos (linha 28).

**Algoritmo 2:** Fase de Evolução do Método NSGA2CGP.

---

```

Entrada: {POPULATION, FITARRAY}
Saída: {POPULATION, FITARRAY, RANKEDPOP}
1 GEN ← 0
2 while GEN < NGEN do
3   FRONTS ← organizeFronts(POPULATION, FITARRAY)
4   RANKEDPOP ← crowdingDistance(POPULATION, FITARRAY, FRONTS)
5   switch MUTSTRAT do
6     case 2N do
7       POPULATION ← POPULATION[RANKEDPOP[1 → NPOP]]
8       for i = 1 to NPOP do
9         PARENT ← selectRandomParent(POPULATION)
10        OFFSPRING ← mutateParent(PARENT)
11        OFFSFIT ← calculateFitness(OFFSPRING)
12        POPULATION[NPOP + i] ← OFFSPRING
13        FITARRAY[NPOP + i] ← OFFSFIT
14      end
15     case ELITIST do
16       for i = 1 to TMUT% do
17         PARENT ← RANKEDPOP[i]
18         OFFSPRING ← mutateParent(PARENT)
19         OFFSFIT ← calculateFitness(OFFSPRING)
20         if OFFSFIT is better or equal fitness than RANKEDPOP[i] then
21           POPULATION[RANKEDPOP[i]] ← OFFSPRING
22           FITARRAY[RANKEDPOP[i]] ← OFFSFIT
23         end
24       end
25     end
26   end
27 end
28 if MUTSTRAT is 2N then /* Seleciona os melhores NPOP para a próxima fase */
29   POPULATION ← POPULATION[RANKEDPOP[1 → NPOP]]
30   FITARRAY ← FITARRAY[RANKEDPOP[1 → NPOP]]
31   RANKEDPOP ← RANKEDPOP[1 → NPOP]
32 end

```

---

As estratégias de mutação serão apresentadas a seguir.

A estratégia *ELITIST* é uma operação de mutação pontual que, sendo *TMUT* o percentual de mutação definido pelo parâmetro de mesmo nome, seleciona os melhores *TMUT*% indivíduos ranqueados e faz a mutação *TMUT*% de seus genes, gerando assim um filho para cada indivíduo. Os filhos gerados têm suas aptidões calculadas e substituem seus pais na população, caso tenham suas aptidões melhores ou iguais às de seus pais, para garantir a diversidade.

A estratégia *2N* é baseada na abordagem clássica de mutação do NSGA-II, que seleciona *NPOP* indivíduos e gera filhos aleatórios. Na estratégia *2N*, todos os indivíduos selecionados têm *TMUT*% dos seus genes alterados por mutação para um valor aleatório, respeitando o tipo do gene definido por *GENTYPE*, gerando *NPOP* filhos e calculando suas aptidões. Assim, a população passa, temporariamente a ter o dobro do tamanho definido pop *NPOP*, ou seja, *2NPOP*.

Para a próxima geração, a população de tamanho  $2N$  é ranqueada em *fronts* e pela distância de aglomeração e os melhores  $NPOP$  indivíduos são escolhidos como a nova população, repetindo o processo até que o número máximo de gerações seja alcançado.

As duas estratégias de mutação, *ELITIST* e  $2N$ , usam uma forma adaptada da técnica "*Single Active Mutation*" para selecionar os genes que podem sofrer mutações aleatórias (GOLDMAN; PUNCH, 2013). Essa abordagem diminui a quantidade de testes, pois permite a seleção apenas de genes ativos do cromossomo, fazendo com que seja menos provável acontecer a mutação de genes inativos. A adaptação feita consiste em, ao invés de selecionar apenas um gene ativo como na proposta da literatura, o NSGA2CGP seleciona  $TMUT\%$  genes ativos para mutação, sem desrespeitar as restrições dos genes e seus tipos.

## 4.5 Apresentação

Com a população evoluída e otimizada na fase anterior, a fase de Apresentação é responsável por exibir a população otimizada para o usuário. Essa tarefa consiste em apresentar os indivíduos divididos em cada *front*, fazendo a distinção entre cada um deles e possibilitando exibir mais ou menos *fronts* e também selecionar um indivíduo específico para visualização específica, de acordo com as ações que o usuário realizar, como apresentado no Algoritmo 3.

---

### Algoritmo 3: Fase de Apresentação do Método NSGA2CGP

---

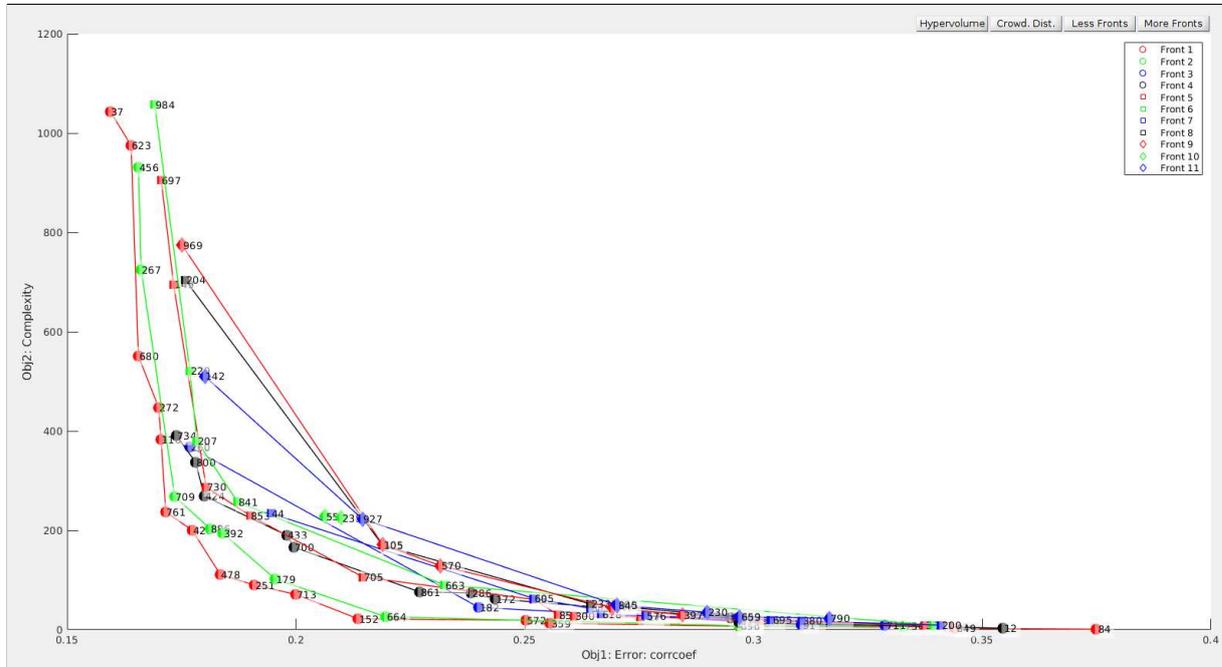
```

Entrada: {PARAMS, POPULATION, FITARRAY, RANKEDPOP}
Saída: {SAVED_TRAIN, FRONTS_PLOT}
1 while Presentation do
2   TOTALFRONT  $\leftarrow$  maxRank(RANKEDPOP)
3   MAXPLOT  $\leftarrow$  min(FRONTPRES, TOTALFRONT)
4   for  $i = 1$  to MAXPLOT do
5     | plotFront(RANKEDPOP[FRONT ==  $i$ ])
6   end
7   switch UserInput do
8     | case Less Fronts AND Not at Minimum Front do
9       | deleteFront(MAXPLOT)
10      | MAXPLOT  $\leftarrow$  MAXPLOT - 1
11     | case More Fronts AND Not at Maximum Front do
12       | MAXFRONT  $\leftarrow$  MAXFRONT+1
13       | plotFront(RANKEDPOP[FRONT == MAXFRONT])
14     | case Visualize Individual do
15       | INDIVIDUAL  $\leftarrow$  userInput.parse()
16       | visualization(INDIVIDUAL)
17     | case Export Training do
18       | SAVED_TRAIN  $\leftarrow$  deployExport(Input)
19     end
20   end
21 end

```

---

No Algoritmo 3, cada *front* é plotado respeitando o número máximo de *fronts* que o usuário quiser visualizar. Em relação às interações do usuário, é possível diminuir ou aumentar os números de *fronts* exibidos (linhas 8 e 11), clicar em um indivíduo específico para visualizá-lo detalhadamente (linha 14) ou exportar o treinamento (linha 17).



**Figura 4.4: Resultados de treinamento exibidos pela fase de Apresentação.**

O método NSGA2CGP utiliza a forma gráfica clássica para exibir os *fronts*. Cada conjunto de indivíduos de um *front* é plotado de cores e formas diferentes dos demais. Além disso, cada indivíduo, ao ser selecionado, envia o usuário à próxima fase, Visualização, que exibe informações detalhadas da solução encontrada naquele indivíduo. Na Figura 4.4 é exemplificado como a fase de Apresentação exibe os resultados de um treinamento, incluindo os botões que permitem exibir mais ou menos *fronts* entre 1 e o número máximo de *fronts*.

## 4.6 Visualização

A fase de Visualização é uma extensão da fase de Apresentação, cuja função é permitir a exibição dos detalhes específicos de cada solução e visualizar informações como valores de aptidão, imagens de saída após aplicação do filtro otimizado daquela solução e visualizar o fenótipo do cromossomo. Quando o indivíduo está sendo visualizado, todas as imagens de treinamento podem ser visualizadas junto com a saída esperada e a saída real, para uma comparação lado a lado sobre a qualidade visual. A descrição dessa fase é apresentada no Algoritmo 4.

**Algoritmo 4:** Fase de Visualização do Método NSGA2CGP

---

```

Entrada: {INDIVIDUAL}
Saída: {CODE, PARTIAL_GENE, OUTPUT_IMAGES}
1 plotCGPChromosome(INDIVIDUAL)
2 while Visualization do
3   switch UserInput do
4     case Visualize Gene Output do
5       | visualizeGene(UserInput.parse())
6     case Show Training Images do
7       | trainingImageOutput(INDIVIDUAL)
8     case Generate Deployable Code do
9       | CODE ← exportDeploy(INDIVIDUAL)
10    end
11  end
12 end

```

---

Quando um cromossomo é visualizado a partir de seu fenótipo, é possível compreender cada etapa do cromossomo (linha 5) e não apenas isso, mas também escolher um gene específico do fenótipo e visualizar sua saída específica (linha 7), mesmo que não seja um gene ativo, permitindo o aprimoramento e ajustes finos de elementos estruturantes ou outros parâmetros em treinamentos futuros. Na visualização do indivíduo também é encontrada a opção para gerar código pronto (linha 9) para ser implantado de acordo com o *Wrapper* de linguagem, passado por parâmetro.

## 4.7 Exportar e Implantar

Por fim, a última fase do método NSGA2CGP é a fase de Exportar/Implantar, descrita no Algoritmo 5, que é responsável por gerar filtros morfológicos já implementados e prontos para serem implantados dentro de um sistema de navegação e começar a corrigir imagens de profundidade (linhas 3 e 4). Essa fase também faz a exportação dos dados de treinamento para que esses possam ser reusados em futuras sessões de treinamentos (linha 6).

**Algoritmo 5:** Fase Exportar e Implantar do Método NSGA2CGP

---

```

Entrada: INDIVIDUAL OR {TRAINING_DATA}
Saída: CODE OR SAVED_TRAIN
1 switch Input do
2   case INDIVIDUAL do
3     | WRAPPER ← selectedCodeWrapper(CODEWRPS)
4     | CODE ← @WRAPPER(INDIVIDUAL)
5   case {TRAINING_DATA} do
6     | SAVED_TRAIN ← saveTrainingData(TRAINING_DATA)
7   end
8 end

```

---

Considerando que o NSGA2CGP não é atrelado a qualquer linguagem de programação, *framework* ou *toolkit*, ele permite gerar filtros morfológicos para qualquer linguagem de programação ou biblioteca que suporte processamento de imagens como, por exemplo, o OpenCV. Não importando o ambiente no qual o NSGA2CGP foi implementado, o parâmetro *CODEWRPS* pode ser utilizado para referenciar funções que transformam um indivíduo otimizado em filtros morfológicos para diferentes linguagens de programação.

## 4.8 Considerações Finais

Neste capítulo foi apresentado o NSGA2CGP, um método de otimização multiobjetivo evolucionário e cartesiano, para geração automática de filtros otimizados voltados à correção de imagens de profundidade, em escala de cinza, contendo regiões de distâncias desconhecidas. Foram apresentadas as cinco fases que compõem o método e suas tarefas individuais. Mais detalhes de como o NSGA2CGP foi utilizado, bem como as funções de aptidão empregadas para otimizar filtros neste trabalho, serão apresentados no Capítulo 6. No próximo capítulo será apresentado o Protótipo Vestível com Sistema Embarcado de Navegação Multimodal e Interativo para Auxílio Customizável a Pessoas com Deficiências Visuais.

# Capítulo 5

## PROTÓTIPO VESTÍVEL COM SISTEMA DE NAVEGAÇÃO MULTIMODAL E INTERATIVO PARA AUXÍLIO CUSTOMIZÁVEL A PESSOAS COM DEFICIÊNCIAS VISUAIS

---

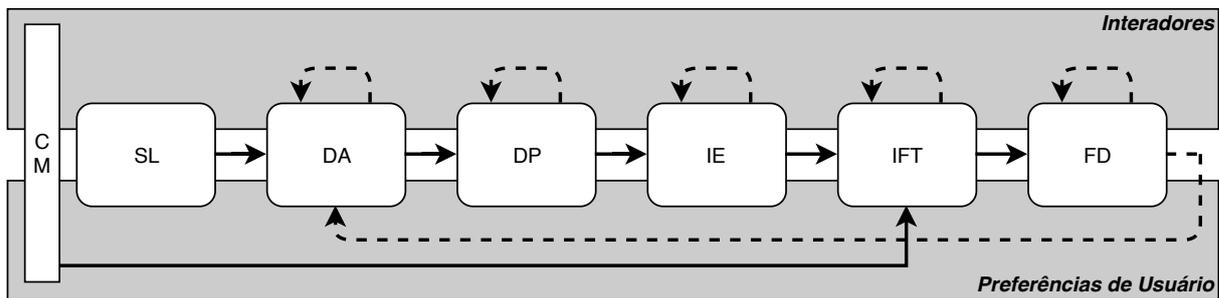
---

Dado o objetivo principal deste trabalho, que é prover autonomia às pessoas com deficiências visuais na realização segura de percursos, neste capítulo são apresentados quatro importantes contribuições que se incrementam, visando o objetivo principal. Assim, são apresentados o "Modelo Abstrato de Sistemas de Navegação Interativos e Customizáveis", que é uma abstração geral que serve como base para construção de sistemas de navegação com meios genéricos de entrada e saída; a "Metodologia para Customização de Sistemas de Navegação para PDVs", que compreende os principais aspectos que envolvem a especificação de comandos que os usuários podem enviar para os sistemas de navegação e como eles serão alterados; o "Sistema Embarcado Multimodal e Interativo para Auxílio Customizável à Navegação de PDVs", que é construído sobre o modelo abstrato e implementa a metodologia para customização; e "Protótipo Vestível para Auxílio à Navegação Segura de PDVs", que apresenta as versões anteriores do protótipo vestível e também a atual, contendo a última versão do sistema e que foi utilizada para os experimentos desta tese.

### 5.1 Modelo Abstrato de Sistemas de Navegação Interativos e Customizáveis

Sistemas de auxílio à navegação para pessoas com deficiências visuais podem fazer uso de diversas tecnologias diferentes para capturar dados, processá-los e enviar uma resposta aos

usuários. Entretanto, independentemente das tecnologias empregadas, é possível realizar a abstração das atividades realizadas por esses sistemas e, no caso deste trabalho, adicionar recursos que promovam a interatividade entre usuário e sistema e também a liberdade de customização de recursos por parte do usuário. Assim, foi criado um "Modelo Abstrato de Sistemas de Navegação Interativos e Customizáveis", dividido em módulos e recursos auxiliares, que pode ser usado como base para o desenvolvimento de sistemas de navegação, independente de tecnologia, que sejam customizáveis e interativos. Os módulos e recursos auxiliares são ilustrados na Figura 5.1.



**Figura 5.1: Modelo Abstrato de Sistemas de Navegação Interativos e Customizáveis.**

O modelo abstrato, apresentado na Figura 5.1, é definido por sete módulos e recursos auxiliares que compreendem, de forma genérica, as principais tarefas de sistemas de navegação interativos e customizáveis. Cada módulo engloba um ou mais grupos de tarefas que garantem o funcionamento do sistema, enquanto os recursos auxiliares podem ser utilizados pelos módulos ou pelo usuário a qualquer momento durante o uso do sistema para interação e customização. Os módulos são:

- **CM** (*Customization Module* ou Módulo de Customização) contém as tarefas para customizar o funcionamento do sistema. Pode ser acessado a qualquer momento durante o uso do sistema e permite que o usuário altere suas preferências em tempo real. O acesso é feito via comando específico de um ou mais Interadores;
- **SL** (*System Loader* ou Carregador de Sistema) reúne tarefas iniciais do sistema que recuperam as preferências de usuário armazenadas em disco, disponibiliza-as globalmente para todo o sistema e define os principais parâmetros necessários para a execução dos demais módulos;
- **DA** (*Data Acquisitor* ou Aquisitor de Dados) responsável por fazer conexão com o(s) dispositivo(s) de entrada e fazer a captura dos dados que serão posteriormente processados pelo sistema;

- **DP** (*Data Preprocessor* ou Pré-processador de Dados) é responsável por receber os dados capturados e prepará-los para a extração de informações. São realizadas tarefas de pré-processamento adequadas para cada tipo de entrada;
- **IE** (*Information Extractor* ou Extrator de Informação) realiza o processamento dos dados em busca de informações, levando em consideração as características de funcionamento do sistema e o tipo dos dados;
- **IFT** (*Information to Feedback Translator* ou Tradutor de Informação para *Feedback*) verifica as preferências de *feedback* do usuário e transforma as informações extraídas no padrão de *feedback* preferencial;
- **FD** (*Feedback Dispatcher* ou Expeditor de *Feedback*) é responsável por realizar a conexão com dispositivo de *feedback*, intermediar a comunicação entre sistema e dispositivo e enviar as informações obtidas para o usuário.

Os recursos auxiliares fornecem maneiras de tornar o sistema interativo e customizável através de dois recursos:

- **Interadores** definem como é possível, com o uso de dispositivos específicos de entrada, como microfones, teclados, etc., interagir com o sistema a fim de alterar sua maneira de funcionar e adequá-lo às necessidades e preferências que o usuário do sistema tem. Além disso, os Interadores também podem exercer funções de solicitação, que permitem ao usuário buscar mais informações ou fornecer dados para para o sistema ao invés de customizá-lo;
- **Preferências de Usuário** é um recurso no qual o sistema pode recuperar as configurações individuais de um usuário, bem como alterá-las por meio dos Interadores, via Módulo de Customização. Contém todos os recursos customizáveis de um sistema, cuja especificação é feita com a "Metodologia para Customização de Sistemas De Navegação para PDVs".

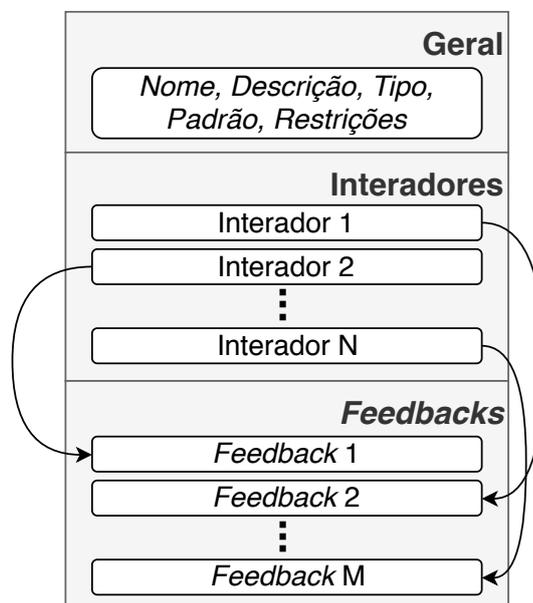
## 5.2 Metodologia para Customização de Sistemas de Navegação para PDVs

A possibilidade de customização, em tempo de execução, do sistema de navegação por parte do usuário com deficiência visual ainda é, segundo a literatura, um dos recursos que

foram pouco explorados nos projetos de pesquisa, cuja importância é salientada tanto pelos autores quanto pelas PDVs entrevistadas (WEINBERGER et al., 2019; JAFRI; KHAN, 2018; LI et al., 2018). Com isso, faz-se necessária a criação de uma metodologia que seja o ponto inicial para os esforços de pesquisa sobre a customizabilidade de sistemas que auxiliam PDVs em seu cotidiano.

O objetivo da Metodologia para Customização de Sistemas de Navegação para PDVs é auxiliar a especificação formal das tarefas de customização de recursos que compõem o funcionamento desse tipo de sistema. Ela contempla os elementos básicos das interações entre usuário e sistema, cujo objetivo é, por um ou mais meio(s) de entrada, alterar como o sistema se comporta, desde quais meios de entrada podem ser utilizados até o tipo de *feedback* a ser enviado pelo sistema.

A metodologia divide as customizações em três etapas: Geral, com informações gerais sobre o recurso que será customizado; Interadores, que definem como o recurso será alterado, o dispositivo e seus dados, e o comando que será executado através do dispositivo; e *Feedbacks*, que especificam qual(is) sinais de confirmação serão enviados para os usuários. Na Figura 5.2 é apresentada uma ilustração com a estrutura resumida de customização de um recurso.



**Figura 5.2:** Estrutura Resumida da Customização de Recursos (Geral, Interadores e *Feedbacks*).

Cada recurso customizável pode ser um ou diversos Interadores, assim como um ou mais *Feedbacks* que são referenciados por um ou mais Interadores. Cada campo especificável tem nome único para futura implementação de *parsers* que interpretem especificações de customizações e adicionem as funcionalidades aos sistemas automaticamente. O nível de detalhamento das informações de especificação depende totalmente dos desenvolvedores do sistema.

### 5.2.1 Geral

A definição geral do recurso customizável a ser especificado é feita na etapa Geral da metodologia. Nessa etapa, devem ser especificados os dados básicos de um recurso que pode ser customizável pelo usuário, desde o nome e descrição até nome de preferência de usuário, um recurso auxiliar. Os dados especificáveis nessa etapa são:

- **RID** - Identificador único do recurso customizável;
- **RNAME** - Nome do recurso;
- **RDESCRIPTION** - Descrição do recurso customizável;
- **UPNAME** - Referência do recurso customizável nas preferências de usuário. Pode ser especificado para referência de implementação;
- **RCONSTRAINTS** - Descrição textual das restrições de customização do recurso. Ex.: Volume máximo 100% e volume mínimo 20%;
- **RTYPE** - Tipo de dado do recurso;
- **RDEFAULT** - Valor padrão do recurso.

### 5.2.2 Interadores

Um recurso pode ser customizado por um ou mais Interadores. A especificação de Interadores prevê como e por onde ocorrerá a customização de um recurso, quais as condições para que cada Interador realize a customização e os modificadores necessários para compor a interação de forma válida. Diferentes Interadores também podem realizar a mesma tarefa de customização, normalmente como alternativas entre um meio e outro, como voz e teclado. Assim, cada recurso customizável pode ter vários Interadores para alterá-lo de diferentes maneiras a partir do usuário. O comando do Interador é condicionado à restrição especificada nele. Um exemplo de Interador é um comando de voz recebido pelo microfone com palavras específicas ou o ato de pressionar o botão de um teclado. A especificação de um Interador é feita pelos seguintes campos:

- **ITID** - Identificador único do Interador;
- **ITNAME** - Nome do Interador;

- **ITDESC** - Descrição da ação do Interador em relação ao recurso;
- **ITCONSTRAINT** - Restrição para uso do Interador;
- **ITYPE** - Tipo de dispositivo do Interador;
- **ITMANUMODEL** - Informações sobre fabricante e modelo do dispositivo do Interador;
- **ITINTERFACE** - Interface de comunicação com o dispositivo do Interador;
- **ITHWID** - Referência de *hardware* para o dispositivo do Interador;
- **ITCOMMAND** - Comando base do Interador;
- **ITMODIFIERS** - Modificadores do comando base para completar o comando do Interador:
  - **ITMODNAME** - Nome/tipo do modificador;
  - **ITMODVALUE** - Valor referência para o modificador.
- **ITFSUCCESS** - Lista de identificadores de *Feedbacks*, separados por vírgula, a serem invocados como confirmação **positiva** que o Interador executou o comando com **sucesso**;
- **ITFRESTRICTED** - Lista de identificadores de *Feedbacks*, separados por vírgula, a serem invocados como confirmação **negativa**, quando o Interador executou o comando e entrou em conflito com a restrição do Interador e/ou do recurso;
- **ITFERROR** - Lista de identificadores de *Feedbacks*, separados por vírgula, a serem invocados como confirmação **negativa**, indicando que o Interador não conseguiu executar o comando devido a um erro.

### 5.2.3 *Feedbacks*

Os *Feedbacks* contemplam as especificações das confirmações via *feedback* dos comandos de customização enviados por meio de um Interador. Com o fim da interação de alteração de um recurso do sistema, um ou mais *feedbacks* podem ser invocados a fim de notificar o usuário sobre a situação da tarefa de customização. A confirmação do comando, entretanto, pode assumir tanto a forma de sucesso quanto de erro/falha na execução da customização ou conflito com restrições, sendo cada confirmação especificada de maneira independente. Os *Feedbacks* dos Interadores são especificados pelos seguintes campos:

- **FID** - Identificador único do *feedback*;
- **FNAME** - Nome do *feedback*;
- **FDESCRIPTION** - Descrição sobre o *feedback*;
- **FCONSTRAINT** - Descrição textual das restrições para que o *feedback* seja invocado;
- **FTYPE** - Tipo de *feedback* (Sucesso, Erro ou Restrição);
- **FSIGNAL** - Sinal de *feedback*. Pode ser um som, uma vibração, entre outros;
- **FPTIMES** - Número de vezes que o sinal se repetirá para formar o *feedback*;
- **FPDURATION** - Quantidade de tempo (em milissegundos) que cada sinal irá durar em relação ao *feedback*;
- **FMSG** - Mensagem a ser transmitida, caso aplicável, pelo *feedback*;
- **FMANUMODEL** - Informações sobre fabricante e modelo do dispositivo do responsável por transmitir o *feedback*;
- **FINTERFACE** - Interface de comunicação com o dispositivo que transmitirá o *feedback*;
- **FHWID** - Referência de *hardware* para o dispositivo de *feedback*.

A especificação dos recursos customizáveis, interadores e *feedbacks* pode ser feita de várias maneiras diferentes. Na Figura 5.3 é apresentado um exemplo de organização de especificação no formato de planilha, com a customização do recurso *MinDistance* com um Interador para aumentar em 0.5 metro a distância mínima de detecção de caminhos livres ao pressionar um botão e dois *Feedbacks*, um acústico e outro tátil. A especificação possui outros Interadores para botões e voz e outros *Feedbacks*, sendo apresentado apenas um exemplo simplificado.

É sugerido o desenvolvimento de uma interface web ou *desktop* que implemente os campos de especificação de recursos customizáveis com suporte à criação e alteração de Recursos, Interadores e *Feedbacks*, e que permita exportar uma lista de especificações em um formato popular como JSON ou XML, para ser interpretado e criar, de forma automática, rotinas de customização dentro do próprio sistema de navegação para PDVs.

Customização RID0005						
Geral	Nome:	Distância Mínima / Navegação			Nome UP:	MinDistance
	Descrição:	Define a distância para encontrar caminhos livres			Tipo:	float
	Restrição:	MinDistance >= 1; MinDistance <= MaxDistance			Padrão:	1.5
Interador ITID0001						
Nome:	Tecla Aumentar Distância		Tipo:	Botão		
Descrição:	Ao apertar a tecla, aumenta em 0.5 metro		Restrições:	MinDistance + 0.5 <= MaxDistance		
Fab./Modelo:	KNUP KP-2003A		Interface:	USB		
Hardware:	1a2c:2124		Sucesso:	FID0001; FID0004; FID0007		
Restrito:	FID0003; FID0006; FID0009		Erro:	FID0002; FID0005; FID0008		
Comando:	KP_KEY8					
Feedback FID0001						
Nome:	Conf. Voz Sucesso		Descrição:	Envia mensagem de áudio confirmando a nova distância		
Restrição:	(FBMethod = Acoustic     FBMethod = Both) && FBAcousticType = Mono			Tipo:	Sucesso	
Sinal:	Áudio Mono		Repetições:	0	Duração:	-
Mensagem:	Nova distância					
Fab./Modelo:	Aftershokz Sportz Titanium	Interface:	USB	Hardware:	1a2c:2124	
Feedback FID0004						
Nome:	Conf. Tátil Sucesso		Descrição:	Envia padrão de vibração sinalizando nova distância		
Restrição:	FBMethod = Tactile     FBMethod = Both			Tipo:	Sucesso	
Sinal:	Vibração		Repetições:	-	Duração:	-
Mensagem:	Nova distância em padrões de vibração					
Fab./Modelo:	HG7881	Interface:	GPIO	Hardware:	gpio73, gpio200, gpio12	

Figura 5.3: Exemplo de organização de especificação de recursos customizáveis.

### 5.3 Sistema Embarcado Multimodal e Interativo para Auxílio Customizável à Navegação de PDVs

O Sistema Embarcado Multimodal e Interativo para Auxílio Customizado à Navegação de PDVs foi projetado considerando todos os aspectos sistemáticos e metodológicos do Modelo Abstrato apresentado na Seção 5.1 e da Metodologia para Customização apresentada na Seção 5.2. Trata-se de um sistema baseado em visão computacional, uma vez que o principal dispositivo no qual o sistema busca dados para extrair informações é uma câmera RGB-D que seja capaz de fornecer tanto imagens comuns quanto dados de distâncias por meio de imagens de profundidade. Seu objetivo é servir como um sistema base, que ofereça ao usuário opções de detecção de caminhos livres e/ou obstáculos, além de solicitar a identificação ou distância de dado obstáculo, de forma interativa e multimodal, já que é possível utilizar tanto fala quanto botões para interagir com o sistema que, por sua vez, pode enviar uma resposta ao usuário via som e/ou via meio tátil. Assim, sua multimodalidade é dada pela possibilidade da comunicação entre usuário e sistema por mais de um meio diferente e até mesmo simultâneos. O diagrama apresentado na Figura 5.4 ilustra o funcionamento, em alto nível, do sistema que foi projetado e desenvolvido.

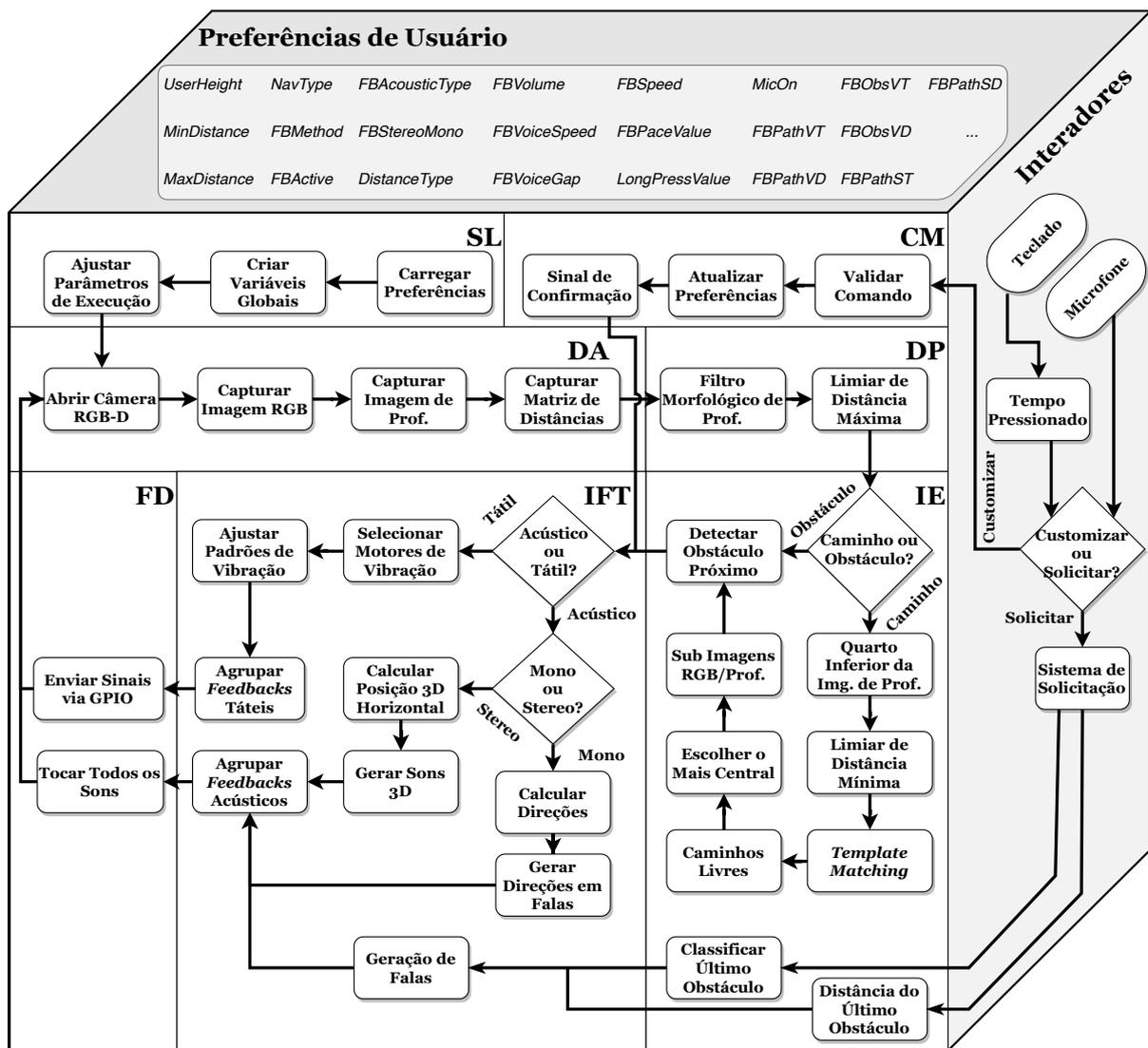


Figura 5.4: Diagrama Geral do "Sistema Embarcado Multimodal e Interativo para Auxílio Customizável à Navegação de Pessoas com Deficiências Visuais". O sistema implementa todos os módulos e recursos auxiliares previstos no modelo abstrato da Seção 5.1.

### 5.3.1 Preferências de Usuário

A customizabilidade do sistema foi projetada visando refletir as observações feitas na ONG PARA-DV, juntamente com as informações coletadas em entrevistas iniciais com os frequentadores. O resultado foi um complexo conjunto de possibilidades de customização, que abrange diversos aspectos do funcionamento do sistema e oferece à pessoa com deficiência visual a capacidade de adaptar o modo de operação às suas próprias necessidades e preferências individuais.

As Preferências de Usuário (PUs) são um recurso auxiliar do sistema, que contém trinta e quatro opções de customização, ou recursos customizáveis, para nortear o funcionamento das principais tarefas de navegação e também aspectos secundários à navegação, como a al-

tura do usuário. Os recursos customizáveis e uma breve descrição deles são apresentados no Quadro 5.1. O acesso às PUs é livre para qualquer módulo do sistema consultar uma ou mais preferências específicas, enquanto a modificação de um recurso das PUs só pode ser feito especificamente pelo Módulo de Customização, cujo acesso é feito pelos Interadores.

**Quadro 5.1: Recursos Customizáveis do Sistema.**

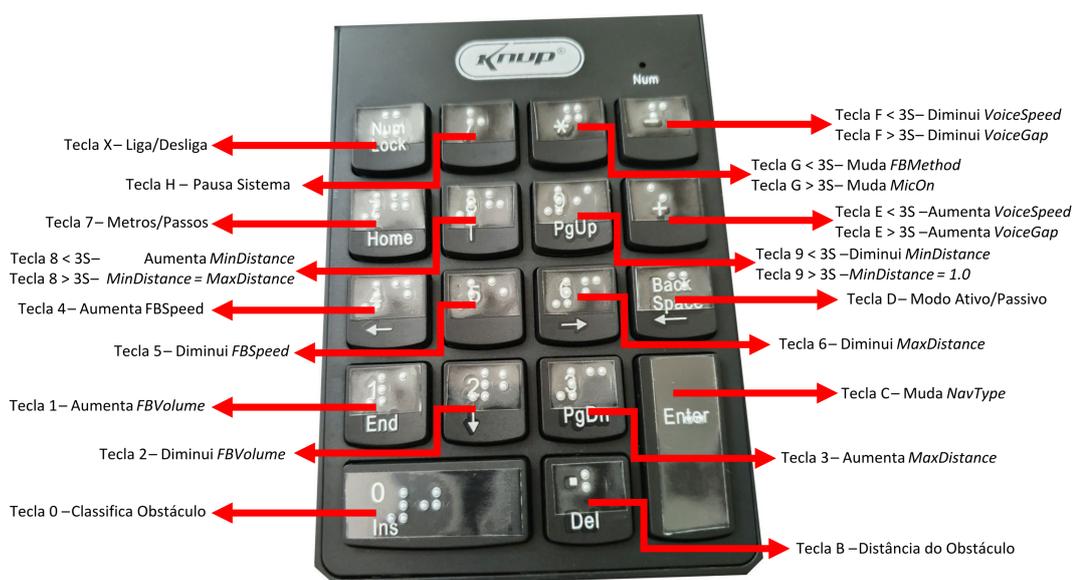
<b>Recurso</b>	<b>Descrição</b>	<b>Recurso</b>	<b>Descrição</b>
<i>UserHeight</i>	Altura do usuário do sistema.	<i>FBVolume</i>	Percentual de volume de <i>feedback</i> .
<i>MinDistance</i>	Distância mínima em mm.	<i>FBVoiceSpeed</i>	Velocidade da fala de <i>feedback</i> .
<i>MaxDistance</i>	Distância máxima em mm.	<i>FBVoiceGap</i>	Espaçamento entre palavras de fala no <i>feedback</i> .
<i>NavType</i>	Modo de navegação do sistema.	<i>FBSpeed</i>	Tempo entre um <i>feedback</i> e outro.
<i>FBMethod</i>	Método de <i>feedback</i> .	<i>FBPaceValue</i>	Valor em mm equivalente a um passo.
<i>FBActive</i>	Modo ativo ou passivo.	<i>FBLongPressVal</i>	Tempo mínimo de pressionamento longo de uma tecla.
<i>FBAcousticType</i>	Tipo de <i>feedback</i> acústico.	<i>MicOn</i>	Habilita o reconhecimento de fala.
<i>FBStereoMono</i>	Tipo de saída de áudio.	<i>DistanceType</i>	Tipo de distância entre metros e passos.
<i>FBPathVT</i>	Número de vibrações para caminho livre.	<i>FBPathVD</i>	Duração de cada vibração para caminho livre.
<i>FBObstVT</i>	Número de vibrações para obstáculo.	<i>FBObstVD</i>	Duração de cada vibração para obstáculo.
<i>FBObstST</i>	Número de <i>beeps</i> para obstáculo.	<i>FBObstSD</i>	Duração de cada <i>beep</i> para obstáculo.
<i>FBPathST</i>	Número de <i>beeps</i> para caminho livre.	<i>FBPathSD</i>	Duração de cada <i>beep</i> para caminho livre.
<i>K0-K9</i>	Customização das teclas de 0 a 9 do teclado.		

Como é possível observar no Quadro 5.1, as possibilidades de personalização de várias áreas diferentes do sistema é grande. O usuário com deficiência visual é capaz de definir, por meio de Interadores, até mesmo o padrão de *feedback* que irá receber para obstáculos e/ou caminhos livres, definindo a quantidade e duração de sinais que receberá. O usuário também pode alterar as funções de customização das teclas entre 0 e 9 de um dos dispositivos Interadores, o teclado braile, que será apresentado na Seção 5.3.2. Com isso, há a liberdade de customizar um dos próprios dispositivos Interadores e organizar os comandos da maneira mais conveniente para o uso individual, considerando as dez teclas disponíveis para customização como recursos customizáveis também. Ao customizar um botão, as opções de recursos são baseadas em seus Interadores, ou seja, a customização de teclas consiste em alterar não apenas o recurso, mas também, especificamente, o Interador do recurso designado à tecla.

### 5.3.2 Interadores

A interação de customização entre PDV e sistema de navegação é feita a partir da captura de dados de dois dispositivos de entrada específicos para interação com o usuário. O primeiro deles é um microfone embutido ao fone de ouvido, que não necessita estar em uso como meio de *feedback*, podendo ser usado para capturar frases do usuário e reconhecê-las como comandos de fala associados a um Interador. Por limitações técnicas, o reconhecimento de fala é feito apenas em inglês atualmente, uma vez que são utilizadas as bibliotecas Sphinx e Pocketsphinx para o processamento da fala e seu suporte em português é limitado e apresentou falhas em testes preliminares. Porém, a inclusão do reconhecimento do português está prevista como um trabalho futuro, quando o sistema for integrado à internet. Cada áudio extraído pelo microfone passa por um processador de hipóteses para construir a sentença proferida pelo usuário e a submete para um validador de comandos. Os comandos devem obedecer uma sintaxe definida durante a implementação, seguindo a especificação dos Interadores, e o usuário do sistema deve ser informado dos comandos previamente. Cada comando pode ou não fornecer um *feedback* para customizações, enquanto solicitações normalmente enviam uma resposta para o usuário.

O segundo dispositivo é um teclado numérico adaptado para braile exclusivamente para esta tese. Trata-se de um teclado numérico comum, com dezoito teclas, que recebeu etiquetas braile em todos os seus botões, podendo assim ser mapeado e utilizado para o envio de comandos ao sistema. Por meio de entrevistas e reuniões com os membros da ONG PARA-DV, o *layout* básico de comandos do teclado e a posição de cada comando foram definidos como exibido na Figura 5.5.



**Figura 5.5:** Teclado braile para interação com o protótipo e padrão de comandos definidos pelos membros da ONG PARA-DV.

A quantidade de teclas, entretanto, é uma das limitações do dispositivo, haja vista que são trinta e quatro recursos customizáveis para apenas dezoito botões. Essa limitação é mitigada com o uso dos modificadores de Interadores, que permitem usar um botão de várias maneiras para diferentes finalidades, e também da customizabilidade das teclas 0-9 para alterar quais recursos essas teclas customizarão quando pressionadas. Por padrão, para customizar as teclas de 0 a 9, basta pressioná-las por 10 segundos e o sistema entrará em modo de customização de tecla. Para cada vez que a tecla em customização for apertada, uma nova função será atribuída a ela e o *feedback* acústico com o nome da função será enviado para o usuário. Dada a quantidade de comandos possíveis, a resposta desse tipo de customização é dada apenas via som. Para salvar o novo comando da tecla, basta pressioná-la novamente por 10 segundos ou apertar qualquer outra tecla para abandonar qualquer alteração.

Portanto, os Interadores procuram fornecer uma interface para comunicação bidirecional com o usuário do sistema de navegação, enviando comandos de voz ou táteis para o sistema com o objetivo de solicitar uma informação em tempo real ou customizar o sistema de acordo com as suas necessidades. Ao receber um comando, o sistema verificará se tal entrada é uma solicitação (classificar um obstáculo, por exemplo) ou a alteração do funcionamento do sistema, uma customização de preferências do usuário.

### 5.3.3 Módulo de Customização

As tarefas de customização do sistema de navegação são gerenciadas pelo Módulo de Customização. Esse módulo é invocado por um Interador após uma ação sobre um dos dispositivos de interação. Nesse caso, como mencionado anteriormente neste capítulo, os dois dispositivos utilizados para tal finalidade foram um microfone e um teclado braile. Ao ser invocado, o módulo recebe o comando de customização junto com seus parâmetros e uma referência às preferências do usuário, que são acessíveis por qualquer módulo. O pseudocódigo do Módulo de Customização é apresentado no Algoritmo 6.

O comando recebido, após verificado sobre sua validade perante o sistema (linha 1), é submetido a uma checagem de restrições junto com os parâmetros. Nessa tarefa, é verificado se a customização irá violar uma restrição imposta pela especificação do recurso (linha 17), como diminuir a distância mínima de um metro e meio para um metro, quando a restrição desse recurso indica que o valor mínimo é um metro e meio. Assim, é gerada a informação de conflito de restrição como confirmação (linha 14), que é enviada para o Módulo de Tradução de Informação para *Feedback*. No caso de uma customização realizada com sucesso, o recurso auxiliar Preferências de Usuário é atualizado com os novos dados e a confirmação também é

enviada *feedback*.

---

**Algoritmo 6: Módulo de Customização**


---

```

Entrada: {COMMAND, PARAMS, USERPREFS}
Saída: INFO
1 if isValid(COMMAND) then
2   if COMMAND is UserHeight then
3     UserHeight  $\leftarrow$  readUserHeightFromInput()
4     while IMUAngle  $\neq$  idealAngle(UserHeight) do
5       if IMUAngle  $>$  idealAngle(UserHeight) then
6         INFO["CONFIR"]  $\leftarrow$  LOW_CAMERA
7       else
8         INFO["CONFIR"]  $\leftarrow$  HIGH_CAMERA
9       end
10      informationToFeedbackModule(USERPREFS, INFO)
11    end
12  end
13  if constraintCheck(COMMAND, PARAMS) then
14    OUTPUTMSG  $\leftarrow$  updateUserPreferences(USERPREFS, COMMAND, PARAMS)
15    INFO["CONFIR"]  $\leftarrow$  success(OUTPUTMSG)
16  else
17    INFO["CONFIR"]  $\leftarrow$  constraint(getConstraint(COMMAND, PARAMS))
18  end
19 else
20   INFO["CONFIR"]  $\leftarrow$  error('Invalid Command')
21 end
22 informationToFeedbackModule(USERPREFS, INFO)

```

---

Existe um caso especial de customização que abrirá caminho para outros tipos de customizações cada vez mais individuais, que é da altura do usuário do protótipo (linha 2). Essa customização é feita com o objetivo de definir o ângulo da câmera RGB-D, considerando que ela estará presa 30 cm abaixo da altura total do usuário. Com o ângulo ideal, o sistema embarcado do protótipo deve ser capaz de capturar imagens a partir de 1 metro de distância. Pessoas com alturas diferentes exigem ângulos diferentes da câmera e isso pode ser definido com ajuda do IMU interno, caso o dispositivo possua.

O usuário pode informar sua altura em centímetros digitando-a no teclado braile e, após o último dígito, o sistema entrará em modo de ajuste de câmera para verificar se o ângulo da câmera RGB-D é o ideal para a altura informada. Caso não seja, o usuário será informado por meio acústico (frase) ou tátil (vibrações) se é necessário erguer ou abaixar mais a câmera. Vibrações na esquerda significam que a câmera está muito baixa, enquanto na direita, que está muito alta. Ao acertar o ângulo, três vibrações em todos os motores indicarão que o ângulo ideal foi encontrado e o sistema voltará ao funcionamento normal. Esse modo específico de customização atualmente só pode ser acessado na inicialização do sistema, durante o Módulo de Carregamento do Sistema, ao pressionar a tecla C por 10 segundos ou mais, e não está documentado na lista de comandos, pois menos da metade dos participantes da ONG PARA-

DV testaram esse modo experimental.

### 5.3.4 Módulo de Carregamento do Sistema

O primeiro modo a ser invocado na execução do sistema é o Módulo de Carregamento do Sistema. Em sua chamada são informados: o nome do arquivo que armazena as Preferências de Usuário, a lista de dispositivos físicos e/ou virtuais que serão utilizados durante a execução do sistema, o modelo já treinado da rede neural convolucional que classificará os obstáculos sob demanda e o modelo de voz treinado que será utilizado para reconhecer os comandos de fala do usuário. No Algoritmo 7 é apresentado o pseudocódigo do Módulo de Carregamento do Sistema.

---

#### Algoritmo 7: Módulo de Carregamento do Sistema

---

```

Entrada: {USR_PREF_FILE, IODEVICES, CNN_MODEL, VOICE_MODEL}
Saída: {EXEC_PARAMS, USERPREFS}
1 foreach userPref field of USER_PREF_FILE do
2   | USERPREFS ← loadPreference(userPref)
3 end
4 foreach device in IODEVICES do
5   | if checkDevice(device) then
6     | EXEC_PARAMS[device.name] ← device
7   | else
8     | throw error('Device device.name not found')
9   | end
10 end
11 EXEC_PARAMS["CNN_MODEL"] ← CNN_MODEL
12 EXEC_PARAMS["VOICE_MODEL"] ← VOICE_MODEL
13 setupModuleThreads(EXEC_PARAMS, USERPREFS)
14 dataAcquisitorModule(USERPREFS)

```

---

Após acessar e carregar os arquivos de entrada, o módulo realiza a leitura das preferências de usuário e as armazena em uma cópia na memória (linha 1), que servirá de referência para todos os outros módulos e terá suas alterações replicadas para o armazenamento em disco. Os dispositivos utilizados também serão submetidos a uma verificação de acesso, buscando garantir que todos eles estejam disponíveis para o início do sistema (linha 4). Por fim, as referências aos dispositivos e os modelos treinados para classificação de obstáculos e fala são agrupados em um pacote de parâmetros de execução (linhas 11 e 12) que é enviado junto com as Preferências de Usuário para a inicialização das tarefas de cada módulo (linha 13), e, em seguida, inicia-se o processamento do sistema, ao invocar o início da aquisição de dados (linha 14).

### 5.3.5 Módulo de Aquisição de Dados

O processo de aquisição de dados faz a captura das imagens oriundas de uma câmera RGB-D, como o Kinect da Microsoft ou a Realsense da Intel, após verificar sua disponibilidade. Duas imagens são capturadas: uma imagem colorida e uma representação de profundidade em escala de cinza. Além delas, uma matriz de distâncias também é capturada e contém a distância em milímetros de cada pixel da imagem de profundidade. O Algoritmo 8 contém o pseudocódigo do Módulo de Aquisição de Dados, que, após a obtenção dos dados da câmera, invoca o Módulo de Pré-processamento de Dados.

---

#### Algoritmo 8: Módulo de Aquisição de Dados

---

**Entrada:** *USERPREFS*  
**Saída:** {*RGBIMG, DEPTHIMG, DEPTHMAT*}

```

1 refreshUserPreferences(USERPREFS)
2 CAMERA ← getRGBDCamera()
3 while CAMERA.waitForFrames() do
4   | continueWaiting();
5 end
6 RGBIMG ← CAMERA.getRGB()
7 DEPTHIMG ← CAMERA.getDepthImage()
8 DEPTHMAT ← CAMERA.getDepthMatrix()
9 dataPreprocessorModule(RGBIMG, DEPTHIMG, DEPTHMAT)

```

---

### 5.3.6 Módulo de Pré-processamento de Dados

O pré-processamento dos dados do sistema consiste em tarefas que preparam as imagens para a extração de informações. Essa preparação é dividida em três etapas subsequentes de processamento de imagens. O Módulo de Pré-processamento de Dados é apresentado no Algoritmo 9. A primeira etapa é a aplicação de um filtro morfológico para correção das áreas de distância desconhecida nas imagens de profundidade em escala de cinza (linhas 1 e 2). O filtro utilizado pelo sistema desta tese foi gerado automaticamente pelo "Método de Otimização Multiobjetivo Evolucionário e Cartesiano para Correção de Imagens de Profundidade", NSGA2CGP, apresentado no Capítulo 4, tendo sido escolhido pelo autor como solução factível para as especificações de equipamentos utilizados.

Com a imagem de profundidade corrigida, o módulo faz a equivalência de valores de escala de cinza dos pixels com as distâncias na matriz de distâncias e também realiza a correção dessa matriz com os valores das áreas corrigidas. Assim, tanto a imagem de profundidade quanto sua respectiva matriz de distâncias ficam corrigidas para uso posterior pelo sistema.

Para concluir o pré-processamento, o módulo recupera das Preferências de Usuário a infor-

**Algoritmo 9:** Módulo de Pré-Processamento de Dados

---

**Entrada:** {*USERPREFS*, *RGBIMG*, *DEPTHIMG*, *DEPTHMAT*}  
**Saída:** {*RGBIMG*, *DEPTHIMG*, *DEPTHMAT*}

- 1 *DEPTHIMG* ← *NSGA2CGPMorphologicalFilter*(*DEPTHIMG*)
- 2 *DEPTHMAT* ← *fillUnknownDistancesFromDepthImg*(*DEPTHIMG*, *DEPTHMAT*)
- 3 *MAXDISTANCE* ← *USERPREFS*["MaxDistance"]
- 4 *DEPTHMAT*(*DEPTHMAT* > *MAXDISTANCE*) ← *max*(*DEPTHMAT*)
- 5 *DEPTHIMG*(*DEPTHMAT* > *MAXDISTANCE*) ← 1 // 1(0-255) é a maior distância  
cinza
- 6 *informationExtractorModule*(*USERPREFS*, *RGBIMG*, *DEPTHIMG*, *DEPTHMAT*)

---

mação sobre a distância máxima de processamento, *MaxDistance* (linha 3), ou seja, até qual distância o sistema deve considerar a existência de objetos para buscar caminhos livres e aplica um operador de limiar, definindo com distância máxima todos os pixels que apresentarem distância maior que *MaxDistance* (linhas 4 e 5). Isso faz com que o sistema considere livre e sem obstáculos tudo que estiver acima dessa distância.

### 5.3.7 Módulo de Extração de Informações

O Módulo de Extração de Informações concentra as tarefas centrais do sistema de navegação desenvolvido nesta tese e é demonstrado no pseudocódigo do Algoritmo 10. Seu funcionamento, após o recebimento das imagens pré-processadas da câmera RGB-D, verifica as Preferências de Usuário sobre o modo Ativo ou Passivo, pela preferência *FBActive* (linha 1). Esses modos definem como os sucessivos quadros recebidos da câmera serão processados.

O modo Ativo processa os quadros e extrai as informações continuamente para *feedback* e o tempo entre uma resposta e outra é definido pela preferência *FBSpeed*, cujo valor é um multiplicador para o intervalo mínimo de 100 milissegundos entre respostas e assume valores a partir de 0 (zero, sem intervalo entre quadros) ou mais (ex.: *FBSpeed* igual a 5 significa que o sistema aguardará 500 milissegundos entre um ciclo e outro de processamento). O modo Passivo, diferente do Ativo, não extrai informações para *feedback* de forma contínua, mas verifica se a cena capturada mudou em relação às anteriores e, em caso positivo, prossegue com a extração de informações. A detecção de mudança de cena é feita utilizando o Coeficiente de Correlação de Pearson (CCP), comparando o CCP (Eq. 5.1) do quadro atual (*X*) em relação ao anterior (*Y*), com a diferença da média dos CCPs últimos *N* quadros (Eq. 5.2). Para este trabalho foi utilizado *N* = 100.

$$\rho = \frac{\text{cov}(X, Y)}{\sigma_x \sigma_y} \quad (5.1)$$

**Algoritmo 10:** Módulo de Extração de Informações

---

**Entrada:** {*USERPREFS*, *RGBIMG*, *DEPTHIMG*, *DEPTHMAT*}

**Saída:** *INFO*

```

1 if Coming from DP module, USERPREFS["FBActive"] is Passive and scene changed or is Active then
2   if USERPREFS["NavType"] in {"Path","Both"} then
3     FREEPATHS ← freePathSegmentation(USERPREFS,DEPTHIMG,DEPTHMAT)
4     if size(FREEPATHS) = 0 then
5       INFO["FREEPATHS"] = 0
6       informationToFeedback(USERPREFS,INFO)
7       return
8     end
9     if size(FREEPATHS) > 1 then
10      if Multipath feedback within 10 seconds then
11        FREEPATHS ← pickMostCentral(FREEPATHS)
12      else
13        INFO["FREEPATHS"] ← size(FREEPATHS)
14        INFO["FREEPATHS_XPOS"] ← getXPathCentroids(FREEPATHS)
15        informationToFeedback(USERPREFS,INFO)
16        return
17      end
18    end
19    INFO["FREEPATHS"] ← 1
20    INFO["FREEPATHS_XPOS"] ← getXPathCentroids(FREEPATHS)
21    [RGBIMG,DEPTHIMG,DEPTHMAT] ← subImage(RGBIMG, DEPTHIMG, DEPTHMAT, FREEPATHS)
22  end
23  if USERPREFS["NavType"] in {"Obstacle","Both"} then
24    OBSTACLES ← getNearestObstacle(RGBIMG,DEPTHIMG,DEPTHMAT)
25    if size(OBSTACLES) > 0 then
26      INFO["OBSTACLES"] ← size(OBSTACLES)
27      INFO["OBSTACLES_XPOS"] ← getXObstCentroids(OBSTACLES)
28    end
29  end
30  informationToFeedback(USERPREFS,INFO)
31 end
32 if Coming from Interactor then
33   if Classify Obstacle then
34     CLASSIFIER ← loadCNNClassifier(getCNNModel())
35     PREDICTIONS ← CLASSIFIER.classify(getLastDetectedObstacle())
36     if First class ≥ 90% then
37       INFO["CONFIR"] ← {PREDICTIONS[1],Certain}
38     else if First class ≥ 60% and second class ≤ 30% then
39       INFO["CONFIR"] ← {PREDICTIONS[1],Reliable}
40     else if Second class ≥ 30% then
41       INFO["CONFIR"] ← {PREDICTIONS[1],Unreliable}
42     else
43       INFO["CONFIR"] ← {Unknown}
44     end
45     informationToFeedback(USERPREFS,INFO)
46   end
47   if Obstacle Distance then
48     INFO["CONFIR"] ← getObstacleDistance(getLastDetectedObstacle())
49     informationToFeedback(USERPREFS,INFO)
50   end
51 end

```

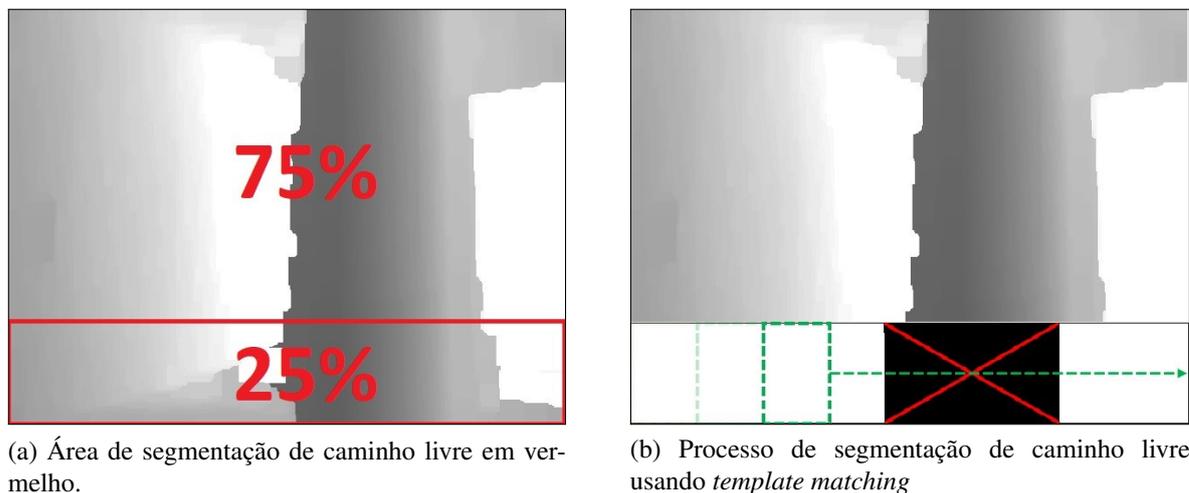
---

$$SC_i = \left| \rho_i - \frac{\sum_{j=1}^N \rho_{i-j}}{N} \right|, \quad i - j > 0 \quad (5.2)$$

A extração de informações pode ocorrer de três modos, de acordo com a definição da preferência *NavType*, das Preferências de Usuário. O primeiro modo (linha 2) é o de Caminhos Livres (Seção 5.3.7.1), em que o sistema busca direções seguras, dentro da distância mínima (*MinDistance*) para a passagem da PDV, e faz a segmentação desses caminhos para informá-los ao usuário. O segundo modo (linha 23), Obstáculos (Seção 5.3.7.1), faz a busca de objetos presentes na imagem de profundidade e segmenta os mais próximos como obstáculos da PDV. O último modo é a junção de Caminhos Livres e Obstáculos (linha 2 seguida de linha 23), que aplica a busca de caminhos livres e, quando um caminho é escolhido, é feita a busca de obstáculos apenas na direção do caminho escolhido.

### 5.3.7.1 Detecção de Caminhos Livres

No modo "Caminho Livre", o sistema faz a segmentação de caminhos livres na frente do usuário com base na distância mínima escolhida por ele. A segmentação é realizada nos 25% inferiores da imagem, utilizando a técnica de *Template Matching* com janela deslizante, como ilustrado na Figura 5.6.



**Figura 5.6: Ilustração sobre o processo de segmentação de caminhos livres.**

É subentendido que os 25% inferiores da imagem têm mais chances de conter os principais dados necessários para o sistema decidir se uma direção é segura para caminhar sem que os pés sejam impedidos por obstáculos como cadeiras, sofás, postes, parquímetros, etc. No caso de múltiplos caminhos, o usuário também é informado e pode se direcionar na direção que preferir ou o sistema escolherá sempre o caminho mais próximo ao centro, para evitar a mudança de

direção.

### 5.3.7.2 Detecção de Obstáculos

No modo "Obstáculos", o usuário é informado apenas do posicionamento dos obstáculos em seu caminho, ignorando totalmente se um caminho é livre ou não. Esse modo leva em consideração o conhecimento do usuário sobre o ambiente no qual está se locomovendo, preocupando-se apenas com os obstáculos. Caso seja necessário, o usuário pode solicitar a classificação de um obstáculo ou sua distância via comando de voz ou pressionando um botão do teclado braile.

Para segmentar os obstáculos, o sistema encontra o(s) mais próximos do usuário com limites progressivos até *MaxDistance*. Ao encontrar pelo menos um obstáculo, seu centro é utilizado como semente para segmentar os componentes conectados ao centro com o método *FloodFill* (CHU et al., 2019). Caso o modo de navegação seja detecção de obstáculos e caminho livre, a segmentação de obstáculos será feita apenas acima da área segmentada como caminho livre. Assim, uma vez que um caminho livre seja escolhido, o sistema deverá efetuar uma busca por obstáculos nos 75% superiores da imagem e, caso encontre um, informar ao usuário. Na Figura 5.7 é exemplificada a detecção de um obstáculo acima da linha do caminho livre.

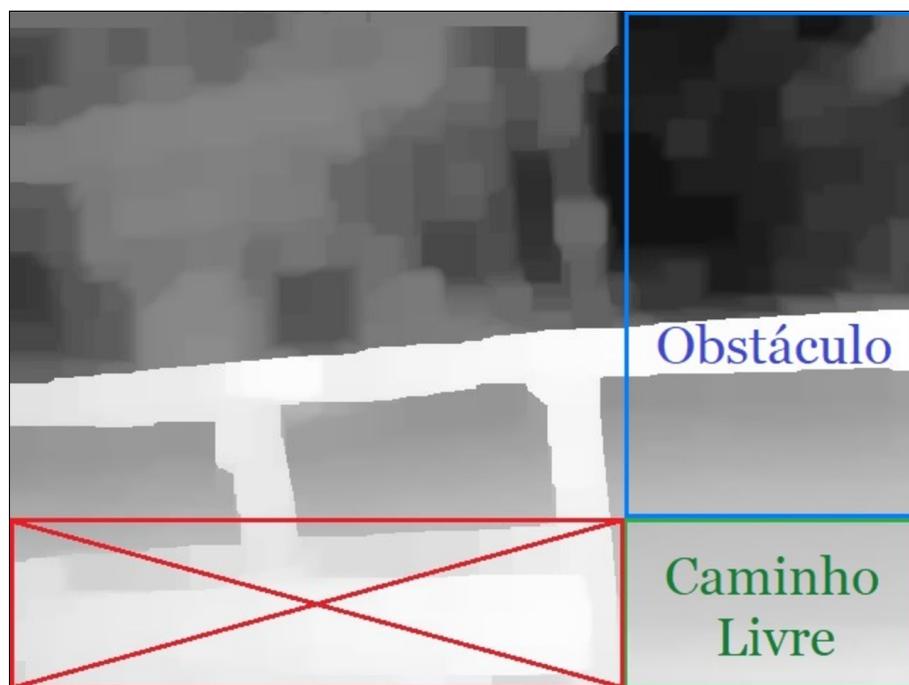


Figura 5.7: Exemplo de Detecção de Obstáculo sobre o Caminho Livre.

### 5.3.7.3 Classificação e Distância de Obstáculos

A classificação dos obstáculos (linha 33 do Algoritmo 10) e suas distâncias (linha 47 do Algoritmo 10) são feitas apenas quando a PDV fizer uma interação de solicitação de identificação de um obstáculo recém-detectado. Com isso, o obstáculo isolado é submetido a uma rede neural artificial de múltiplas camadas intermediárias que realiza convoluções na imagem do obstáculo e então o classifica. Neste trabalho foram utilizadas redes neurais convolucionais para a classificação, treinadas pelo *framework* Caffe e reaproveitando treinamentos anteriores dos modelos AlexNet, GoogLeNet e VGG-16, por meio do *Transfer Learning*, com uma base de imagens extraída do site Image-Net. Apesar do uso de redes neurais convolucionais, novas abordagens de redes neurais profundas podem ser utilizadas em trabalhos futuros, como redes neurais convolucionais recorrentes e/ou unidades de *Long Short-Term Memory* (LSTMs).

Os classificadores utilizados retornam as cinco maiores probabilidades juntamente com suas respectivas classes. Obstáculos parecidos podem gerar impacto no resultado e, conseqüentemente, na confiabilidade do sistema. Assim, a Equação 5.3 utiliza condições para nivelar a confiabilidade da predição feita pelo modelo, visando transmitir tal confiabilidade no *feedback* para o usuário. Na equação são propostos quatro percentuais, de acordo com a probabilidade obtida pela função  $Pred_n(img)$ , em que  $n$  é a  $n$ ésima predição, 1 tem a maior probabilidade, 2, a segunda maior probabilidade e assim sucessivamente. A confiabilidade da predição é considerada total, ou seja, 100%, caso a probabilidade da primeira classe seja maior ou igual a 90%. Já para os casos em que a probabilidade da primeira classe é maior ou igual a 60% e segunda classe menor ou igual a 30%, considera-se 75% de confiabilidade, significando que há uma chance de o obstáculo não pertencer à primeira classe. Caso a segunda predição seja superior a 30%, é considerado que há 50% de confiabilidade, tornando a classificação do obstáculo potencialmente incerta.

$$Conf(img) = \begin{cases} 100\% \text{ ou Certeza,} & \text{se } Pred_1(img) \geq 90\%, \\ 75\% \text{ ou Provável,} & \text{se } Pred_1(img) \geq 60\% \text{ e } Pred_2(img) \leq 30\%, \\ 50\% \text{ ou Incerto,} & \text{se } Pred_2(img) \geq 30\%, \\ 0\% \text{ ou Desconhecido,} & \text{caso contrário.} \end{cases} \quad (5.3)$$

A mensagem que será enviada para *feedback* é criada com base na Equação 5.3, sendo que casos com confiabilidade máxima passam apenas à primeira classe como mensagem (ex: "Televisão"), enquanto para confiabilidades de 75%, a mensagem criada deixa claro que *possivelmente* seja o objeto da primeira classe (ex: "Possível Televisão"). Já para o caso de 50% de

confiabilidade, a mensagem transmitida apresenta incerteza sobre a classificação (ex: "Parece ser Televisão"). Por fim, caso não seja possível classificar o obstáculo (Confiabilidade 0%), o *feedback* informará o usuário de que não foi possível determinar a classe do obstáculo (ex: "Não conhecido").

### 5.3.8 Módulo de Tradução de Informações para *Feedbacks*

O Módulo de Tradução de Informações para *Feedbacks* é responsável por toda a conversão das informações extraídas dos dados processados e por prepará-las para o envio ao usuário de forma compreensível. Esse módulo recebe informações tanto do Módulo de Extração de Informação quanto do Módulo de Customização e faz as adequações necessárias para que as informações brutas se transformem em sinais ou instruções de sinais que, quando enviadas ao usuário, respeitem suas preferências de *feedback*. Assim, o módulo leva em consideração as definições do recurso auxiliar Preferências de Usuário e como os *feedbacks* devem ser constituídos dentre as diferentes possibilidades disponíveis, dependendo das preferências armazenadas. O pseudocódigo do módulo é apresentado no Algoritmo 11.

---

#### Algoritmo 11: Módulo de Tradução de Informações para *Feedbacks*

---

```

Entrada: {USERPREFS,INFO}
Saída: FBPACKAGE
1 switch USERPREFS["FBMethod"] do
2   case Tactile or Both do
3     FBPACKAGE["OBST_MOTORS"] ← getMotors(USERPREFS,INFO,"OBST")
4     FBPACKAGE["OBST_VPATTERN"] ← getVPattern(USERPREFS,"OBST")
5     FBPACKAGE["FPATH_MOTORS"] ← getMotors(USERPREFS,INFO,"FPATH")
6     FBPACKAGE["FPATH_VPATTERN"] ← getVPattern(USERPREFS,"FPATH")
7     FBPACKAGE["CONFIR_MOTORS"] ← getMotors(USERPREFS,INFO,"CONFIR")
8     FBPACKAGE["CONFIR_VPATTERN"] ← getVPattern(USERPREFS,"CONFIR")
9   end
10  case Acoustic or Both do
11    if USERPREFS["FBAcousticType"] = "Stereo" then
12      FBPACKAGE["OBST_SOUND"] ← get3DSound(USERPREFS,INFO,"OBST")
13      FBPACKAGE["OBST_SPATTERN"] ← getSPattern(USERPREFS,"OBST")
14      FBPACKAGE["FPATH_SOUND"] ← get3DSound(USERPREFS,INFO,"FPATH")
15      FBPACKAGE["FPATH_SPATTERN"] ← getSPattern(USERPREFS,"FPATH")
16      FBPACKAGE["CONFIR_SOUND"] ← get3DSound(USERPREFS,INFO,"CONFIR")
17      FBPACKAGE["CONFIR_SPATTERN"] ← getSPattern(USERPREFS,"CONFIR")
18    else
19      FBPACKAGE["OBST_SOUND"] ← genTTSAudio(USERPREFS,INFO,"OBST")
20      FBPACKAGE["FPATH_SOUND"] ← genTTSAudio(USERPREFS,INFO,"FPATH")
21      FBPACKAGE["CONFIR_SOUND"] ← genTTSAudio(USERPREFS,INFO,"CONFIR")
22    end
23  end
24 end
25 feedbackDispatchModule(USERPREFS, FBPACKAGE)

```

---

As informações recebidas de outros módulos podem ser mensagens de *feedback* já predefinidas como confirmações, posicionamento horizontal de um obstáculo ou a direção de um ou mais caminhos livres. De acordo com a preferência *FBMethod*, o módulo pode formatar o *feedback* como padrão de vibração (tátil), som (acústico) ou ambos. Para o método tátil (linha 2), o módulo obtém a relação de motores que irão compor o *feedback* e o padrão de vibração para cada tipo de informação. No caso de caminhos livres e obstáculos, as preferências *FBPathVT*, *FBPathVD*, *FBObstVT* e *FBObstVD* são utilizados para informar a quantidade e duração de vibrações para caminhos e obstáculos, respectivamente.

Um caso especial de padrão de vibração a ser destacado é a representação de informações numéricas em vibrações. Visando atender especialmente PDVs com problemas de audição, foi identificada a necessidade de criar uma maneira de informar ao usuário do sistema números, como distâncias. A solução encontrada para isso foi distribuir as vibrações entre três motores posicionados no ombro esquerdo, peitoral e ombro direito, e limitar a representação numérica, por hora, até 999. Portanto, em *feedbacks* numéricos, o motor do ombro esquerdo representa a centena, o motor do peitoral representa a dezena e o motor do ombro direito representa a unidade. A quantidade de vibrações define o multiplicador do motor. Por exemplo, para representar 1 metro e 63 centímetros em vibrações, 1.63 será transformado em 163 centímetros e o motor da esquerda vibrará uma vez, seguido de seis vibrações no peitoral e três vibrações na direita. Os participantes da ONG PARA-DV auxiliaram na criação do padrão. Mais informações sobre as posições dos motores de vibração serão dadas na Seção 5.4.2.1 deste capítulo.

O *feedback* acústico (linha 10) é dividido em dois tipos de som: estéreo (linha 11) e monoaural (linha 18). No primeiro, um som polifônico representa a posição de um objeto, a direção de um caminho livre ou um padrão predefinido de som como confirmação. O segundo consiste na sintetização de fala a partir de sentenças contendo a direção de um caminho, uma mensagem de confirmação, a classe ou distância do obstáculo, etc. Portanto, o *feedback* acústico pode ser categorizado das seguintes maneiras:

- *Áudio monoaural*, em que o *feedback* é sintetizado como fala
  - Caminhos: Quando um caminho é recebido, a posição  $X$  do caminho é dividida em três regiões horizontais, de acordo com a segmentação realizada, e a direção é sintetizada. As direções podem ser: Esquerda, Frente e Direita.
  - Obstáculos: Por padrão, os obstáculos são mapeados em som estéreo para não confundir o usuário, principalmente quando os caminhos livres e obstáculos estão sendo detectados juntos. Porém, via dispositivo interador há a classificação do obstáculo, que, por sua vez, é sintetizado como fala;

- **Áudio Estéreo**, em que as informações de direção são mapeadas horizontalmente no intervalo  $[-1,1]$ , sendo que  $-1$  representa a extrema esquerda e  $1$  representa extrema direita. É realizada uma simples operação de normalização. Os sons são gerados com as preferências *FBPathST* e *FBPathSD* para quantidade de *beeps* e duração de cada *beep* de caminho livre e *FBObstVT* e *FBObstVD* para as mesmas características, porém para obstáculos. Para *feedbacks* que não contenham direção, como confirmações, são utilizados padrões predefinidos internamente.

### 5.3.9 Módulo de Despacho de *Feedback*

O último módulo do sistema é o Módulo de Despacho de *Feedback*, representado pelo Algoritmo 12. Esse módulo é responsável por receber o pacote de *feedbacks* previamente formatados pelo Módulo de Tradução de Informações para *Feedback* e enviá-los para as PDV. Cada *feedback* pode ser enviado por até duas interfaces diferentes, portas GPIO (linha 1) ou saída de áudio (linha 6). As tarefas do módulo consistem no acesso a cada pacote de *feedback*, acessar as interfaces de comunicação deles e, após enviar o sinal de resposta para o usuário, encerrar a conexão com os dispositivos de saída. Com os *feedbacks* enviados, o sistema inicia um novo ciclo, com a aquisição de novas imagens.

---

#### Algoritmo 12: Módulo de Despacho de *Feedback*

---

```

Entrada: {USERPREFS,FBPACKAGE}
Saída: Feedback to the user
1 foreach [motor,pattern] in [FBPACKAGE[*]_MOTORS],FBPACKAGE[*]_VPATTERN]] do
2   | GPIO ← openGPIOpins(motor)
3   | GPIO.sendSignal(pattern)
4   | GPIO.close()
5 end
6 foreach sound in FBPACKAGE[#1]_SOUND]] do
7   | AUDIO ← openAudioInterface()
8   | if FBAcousticType = "Stereo" then
9     | pattern ← FBPACKAGE[#1]_SPATTERN]]
10    | AUDIO.play(sound,pattern)
11  else
12    | AUDIO.play(sound)
13  end
14  | AUDIO.close()
15 end
16 dataAcquisitorModule(USERPREFS)

```

---

## 5.4 Protótipo Vestível

Além do sistema, este trabalho também fez o design e criação de um protótipo vestível de auxílio à navegação segura para pessoas com deficiências visuais. Em sua concepção, o protótipo foi pensado para oferecer boa mobilidade para PDV, sem detrimento relevante da capacidade de processamento. Portanto, a utilização de plataformas embarcadas é pertinente para garantir a relação mobilidade/desempenho. Outra preocupação foi a autonomia de uso do protótipo, uma vez que a mobilidade é garantida com o uso de baterias como fontes de alimentação dos dispositivos e o peso não deve ser excessivo, já que baterias costumam ser pesadas.

Em termos de desempenho, pela natureza do sistema embarcado baseado em visão computacional, diversas operações geométricas são realizadas e, ao utilizar o processador para tais cálculos, o desempenho do sistema tende a cair e/ou aumentar o consumo de energia, mesmo em plataformas embarcadas. Tarefas de classificação de imagem utilizando redes neurais convolucionais, especificamente, realizam diversas operações em matrizes, o que afeta ainda mais negativamente o desempenho. Nesse cenário, o paralelismo auxilia no desempenho do processamento desse tipo de tarefa, explorando GPUs embarcadas, como a Jetson Tegra X1 (NVIDIA, 2016) ou Jetson Nano (NVIDIA, 2019), que são "Sistemas-Em-Um-Chip" (em inglês, *System-on-Chip* (SOC)), pequenas placas de processamento contendo processadores e GPUs embarcadas, do tamanho de um cartão de crédito, que podem ser embarcadas em diversas outras placas controladoras. Durante o desenvolvimento desta tese, foram criados dois protótipos vestíveis, sendo a primeira versão um colete e a segunda versão uma mochila.

### 5.4.1 Primeira Versão: Colete

O primeiro protótipo vestível contendo uma versão antiga do sistema embarcado foi criado sobre um colete de *airsoft*. Por serem de materiais relativamente leves e modularizáveis com compartimentos para baterias e cabos, os coletes foram escolhidos como base do primeiro protótipo vestível. Na Figura 5.8 são exibidos a frente e o verso do colete montado e funcionando em uma pessoa sem deficiência visual para demonstração.

Em sua primeira versão, o protótipo já era capaz de classificar alguns obstáculos e identificar caminhos livres (DOURADO; PEDRINO, 2018). Foi experimentalmente testado com sete participantes da ONG PARA-DV de Araraquara/SP e valiosas observações foram feitas. Os participantes foram os mesmos da segunda versão. Nessa versão, o *feedback* era feito apenas via som estéreo ou monaural, o que obrigava o usuário a utilizar os fones de ouvido o tempo



(a) Frente do protótipo.

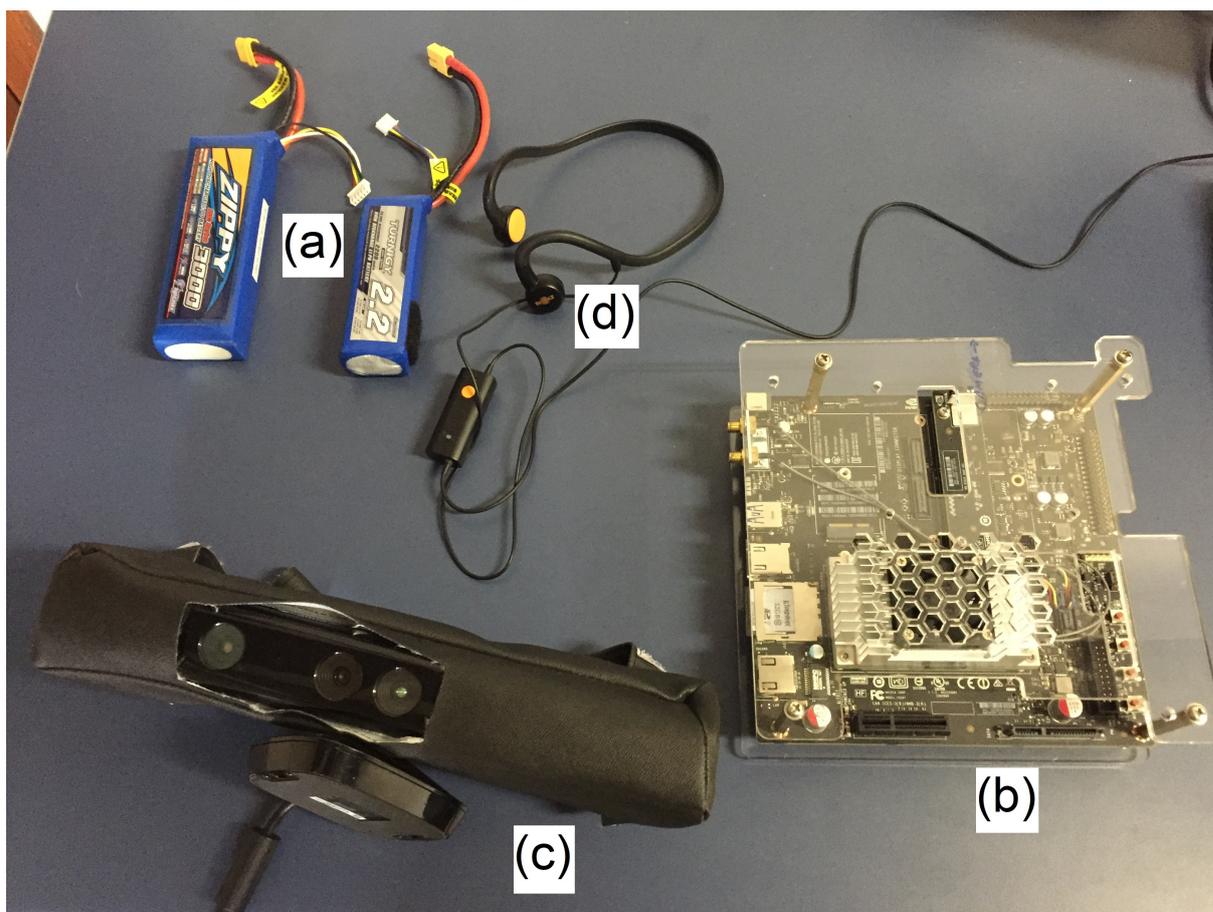
(b) Verso do Protótipo.

**Figura 5.8: Primeira versão do colete. Foi vestido em uma pessoa sem perda significativa de visão apenas para fins demonstrativos.**

todo para que o protótipo funcionasse da maneira esperada e as interações de customização eram apenas por voz e limitadas. Foi observada grande dificuldade de uso por usuários com perda parcial de audição, devido às respostas serem apenas via áudio. Além disso, o colete não era fácil de ser vestido sozinho e seu peso total não era agradável ao usuário. Outro problema observado foi o aumento da sensação de calor, relatado pelas PDVs durante o uso. O protótipo era composto dos seguintes equipamentos:

- **Placa:** Kit de Desenvolvimento NVIDIA Jetson TX1
- **Câmera RGB-D:** Kinect XBOX 360
- **Fone de ouvido:** Fones de condução óssea com microfone Aftershokz
- **Bateria 1:** LiPo 3S 2200mAh (Kinect)
- **Bateria 2:** LiPo 4S 2200mAh (Jetson TX1)

Adicionalmente ao colete, foram confeccionados pequenos bolsos customizados para abrigar a câmera Kinect XBOX 360 e prendê-la junto ao colete na região do tórax e para abrigar o kit de desenvolvimento Jeton TX1 da NVIDIA, prendendo-o no verso do colete. Dois compartimentos adicionais foram utilizados para armazenar as baterias de alimentação da câmera e do kit de desenvolvimento, como observado na Figura 5.8. Na Figura 5.9 são exibidos os equipamentos desmontados e que foram usados na primeira versão do protótipo.



**Figura 5.9:** Equipamentos utilizados na primeira versão do protótipo: (a) Baterias, (b) NVIDIA Jetson TX1, (c) Kinect XBOX 360 e (d) *headset* de condução óssea.

Dado o *feedback* relatado pelos membros da ONG PARA-DV sobre o protótipo, as observações feitas durante as interações experimentais para uso dessa primeira versão do protótipo e os próprios testes feitos pelo autor desta tese em ambiente ainda controlado, foi evidenciada a necessidade de um protótipo mais funcional para um cenário de uso cotidiano. Algumas tarefas não poderiam ser facilmente realizadas por pessoas com deficiências visuais, como recarregar as baterias LiPo corretamente, já que elas precisam de carregadores específicos e a usabilidade desses para pessoas sem a visão é inexistente. As observações também mostraram que era necessário rever os meios de customização e de *feedback* do sistema, considerando os diferentes perfis dos participantes.

#### 5.4.2 Segunda Versão: Mochila

Na segunda versão do protótipo, diversos problemas relatados e observados na primeira versão foram tratados. O novo protótipo foi desenvolvido como uma mochila funcional, tendo embarcado em si o hardware e o software para auxílio à navegação segura. Essa nova versão foi

concebida visando não apenas o aprimoramento tecnológico, com equipamentos e/ou recursos novos, mas também sua capacidade funcional, além do propósito original. Na Figura 5.10 são exibidas duas imagens de um dos membros da ONG PARA-DV vestindo o protótipo.



(a) Frente do protótipo.

(b) Verso do Protótipo.

**Figura 5.10: Segunda versão do protótipo vestível em uma pessoa com deficiência visual.**

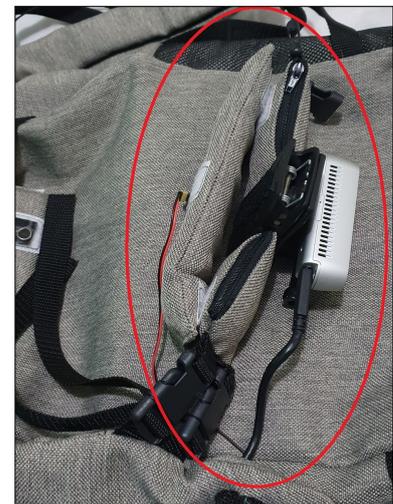
Alguns autores da literatura apontam que os sistemas de auxílio à navegação de PDVs devem garantir que, dentre outras coisas, não haja obstrução à mobilidade do usuário durante o uso do sistema e também que os protótipos vestíveis procurem ser discretos (JAFRI; KHAN, 2018; KATZSCHMANN; ARAKI; RUS, 2018; WEINBERGER et al., 2019). Baseado nisso, uma mochila comum de passeio foi adaptada para receber o hardware, dispositivos e cabos da maneira mais discreta possível e ainda permitindo utilizar a mochila para sua função original: carregar objetos pessoais como roupas, cadernos, etc. As adaptações incluíram adicionar um fundo falso para passagem e manutenção dos cabos e fios quando necessário, deixando livre o uso do espaço interno da mochila, e a adição de uma capa modular sobre uma das alças para passagem dos mesmos materiais. Isso permitiu, por meio de entrevistas e observações, ajustar o projeto dessa versão do protótipo para o seu modelo final, que foi confeccionado exclusivamente para esta tese e é apresentado em detalhes na Figura 5.11.



(c) Motores de Vibração das Alças.



(d) Motor de Vibração das Costas.



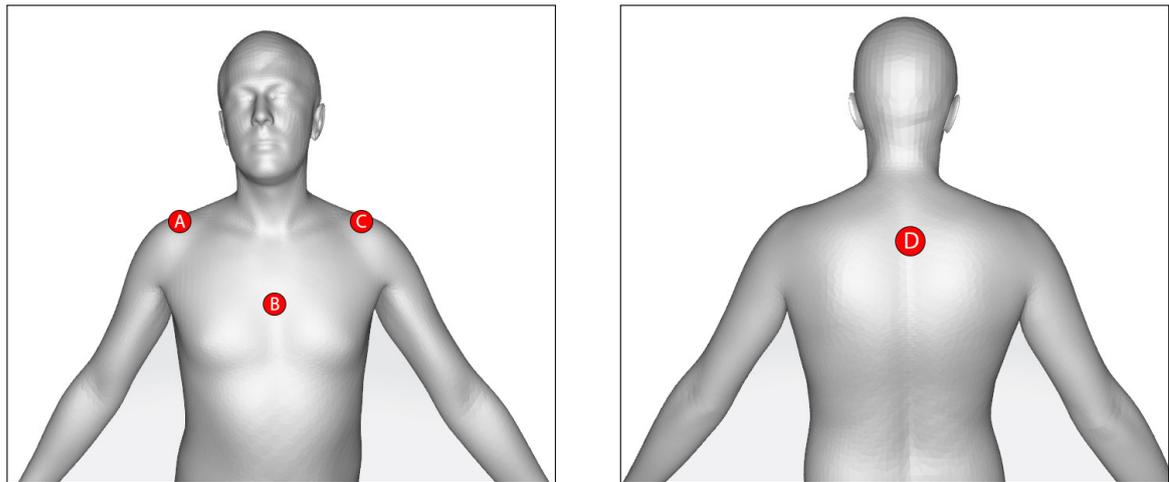
(e) Motor de Vibração do Peitoral.

**Figura 5.11: Modelo final do protótipo vestível confeccionado exclusivamente para esta tese.**

#### 5.4.2.1 Feedback Tátil

As observações feitas nos experimentos preliminares com a ONG PARA-DV deixaram claro que, para PDVs com diminuições de audição, *feedbacks* acústicos podem ser difíceis de serem compreendidos e isso implica diretamente a utilidade do protótipo. Dessa forma, optou-se por adicionar uma alternativa de *feedback* ao sistema e protótipo, o *feedback* tátil. A literatura propõe o uso de luvas, cintos, óculos e outros dispositivos com motores de vibração para sinalizar direções de caminhos e/ou obstáculos, mas isso significaria mais um equipamento para o usuário vestir/carregar (KOMMEY; HERRMAN; ADDO, 2019; POGGI; MATTOCCIA, 2016; PRIYADARSHANA; WIMALARATNE, 2014). A solução encontrada foi integrar os motores de vibração na própria mochila, em posições distintas que não obriguem o usuário a ter que executar tarefas adicionais ao vestir o protótipo para ter acesso ao *feedback* tátil.

O *feedback* tátil do protótipo se dá com a ativação de padrões para quatro motores de vibração posicionados na mochila para estimular quatro partes do corpo. Cada alça lateral contém um motor que fica em contato com os ombros do usuário, enquanto os dois outros motores são posicionados nas costas da mochila, de modo a estimular o usuário na altura do começo da coluna torácica, e na parte posterior da alça horizontal na frente da mochila, ficando em contato com a região peitoral do usuário, como exibido na Figura 5.12.



(a) Área de contato dos motores de vibração da frente.

(b) Área de contato do motor de vibração das costas.

**Figura 5.12: Posicionamento aproximado dos motores de vibração em relação ao corpo humano para estímulo via *feedback* tátil.**

#### 5.4.2.2 Equipamentos

Novos equipamentos do estado da arte foram integrados ao novo protótipo. O hardware base para o processamento do sistema foi substituído pela então recém-lançada NVIDIA Jetson Nano (NVIDIA, 2019), uma SoC de relativo baixo custo (US\$ 99,00) lançada em abril de 2019 com a proposta de popularizar os projetos de computação inteligente e robótica de baixo custo. Essa placa possui um modo de baixo consumo de 5 Volts e picos de, no máximo, 2 Amperes (com periféricos), o que faz com que seja possível alimentar a placa via Micro-USB e utilizar *PowerBanks* como fonte de energia ao invés de baterias LiPo, dando autonomia para o usuário sem visão recarregar sozinho o *PowerBank* após treinamento prévio.

A câmera RGB-D do Kinect XBOX 360 continha diversas restrições. Além do tamanho e peso, a necessidade de uma fonte de energia específica para a câmera é um fator de impacto negativo, já que o usuário precisaria carregar consigo mais uma bateria, apenas para alimentar a câmera. O Kinect também inviabilizava o uso do protótipo em ambientes abertos, pois a técnica de luz estruturada implementada na câmera é severamente afetada pela luz natural do

sol ou outras fontes de infravermelho. A câmera usada passou a ser a Intel Realsense D435i, que, além de ser pequena, tem sua alimentação via porta USB pela mesma conexão de dados e implementa a técnica de infravermelho passivo, o que a habilita para uso em ambientes internos e externos sem grandes problemas. Ela também possui um sensor IMU interno que é usado atualmente para definir o ângulo ideal da câmera e será usado em futuras expansões do projeto, que incluirão o monitoramento da localização do usuário para guiá-lo a destinos específicos.

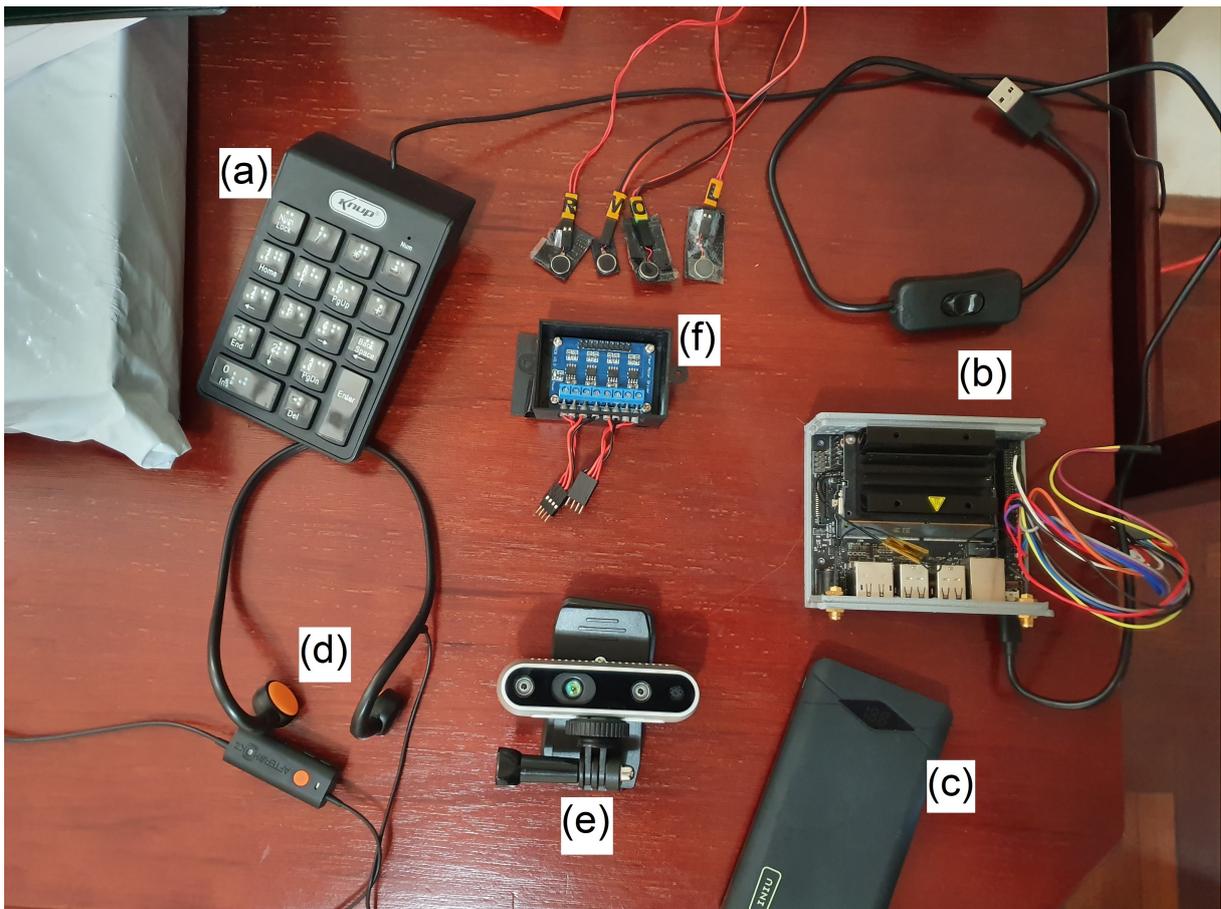
Para controlar os quatro motores de vibração, o protótipo contém uma ponte H modelo HG7881 que é manipulada via sinais da porta GPIO do SoC e ativa/desativa cada um dos motores ligados a ela. Os motores são pequenos, de vibração Vibracall 1027, normalmente encontrados dentro dos aparelhos celulares atuais. Portanto, a segunda versão do protótipo é composta dos equipamentos na Figura 5.13, listados abaixo:

- **Placa:** Kit de Desenvolvimento NVIDIA Jetson NANO
- **Câmera RGB-D:** Intel Realsense D435i
- **Fone de ouvido:** Fones de condução óssea com microfone Aftershokz
- **Bateria:** *Powerbank* INIU 10000Mah
- **Teclado Braille:** Teclado numérico KNUP KP-2003A com adesivos braille
- **Motores:** 4 x Vibracall 1027
- **Ponte H:** HG7881 de quatro canais

Na Figura 5.13, os equipamentos estão dispostos da seguinte forma: Teclado braille (Figura 5.13a), NVIDIA Jetson NANO e botão liga/desliga (Figura 5.13b), *Powerbank* (Figura 5.13c), *Headset* de condução óssea (Figura 5.13d), Câmera RGB-D (Figura 5.13e), Ponte H e motores (Figura 5.13f). Os custos dos equipamentos, bem como instruções sucintas de como preparar e configurar o ambiente para o sistema de navegação desenvolvido nesta tese, onde baixar o código-fonte e como compilá-lo, podem ser encontrados no Apêndice E.

## 5.5 Considerações Finais

Neste capítulo foram apresentados os esforços para alcançar os objetivos descritos no Capítulo 1 e os meios de comprovar as hipóteses lá apresentadas. Foram introduzidos o Modelo Abstrato de Sistemas de Navegação Interativos e Customizáveis e cada um de seus módulos e



**Figura 5.13: Equipamentos utilizados na segunda versão do protótipo.**

recursos auxiliares, com o objetivo de guiar o desenvolvimento de sistemas que auxiliem o cotidiano de PDV; a Metodologia para Customização de Sistemas de Navegação para PDV, com os principais elementos para especificação formal de recursos customizáveis pelo usuário; o Sistema Embarcado Multimodal e Interativo para Auxílio Customizável à Navegação de PDV, que foi desenvolvido baseado no Modelo Abstrato e na Metodologia para Customização, tendo resultado em um sistema concreto para auxiliar pessoas com deficiências visuais no dia a dia e que também aplica soluções geradas pelo método NSGA2CGP, apresentado no Capítulo 4; e, por fim, as duas versões do protótipo vestível desenvolvido nesta tese, os equipamentos e recursos empregados neles. No próximo capítulo serão apresentados os experimentos realizados com o intuito de validar e avaliar o protótipo vestível com o sistema de navegação e o método NSGA2CGP.

# Capítulo 6

## EXPERIMENTOS

---

---

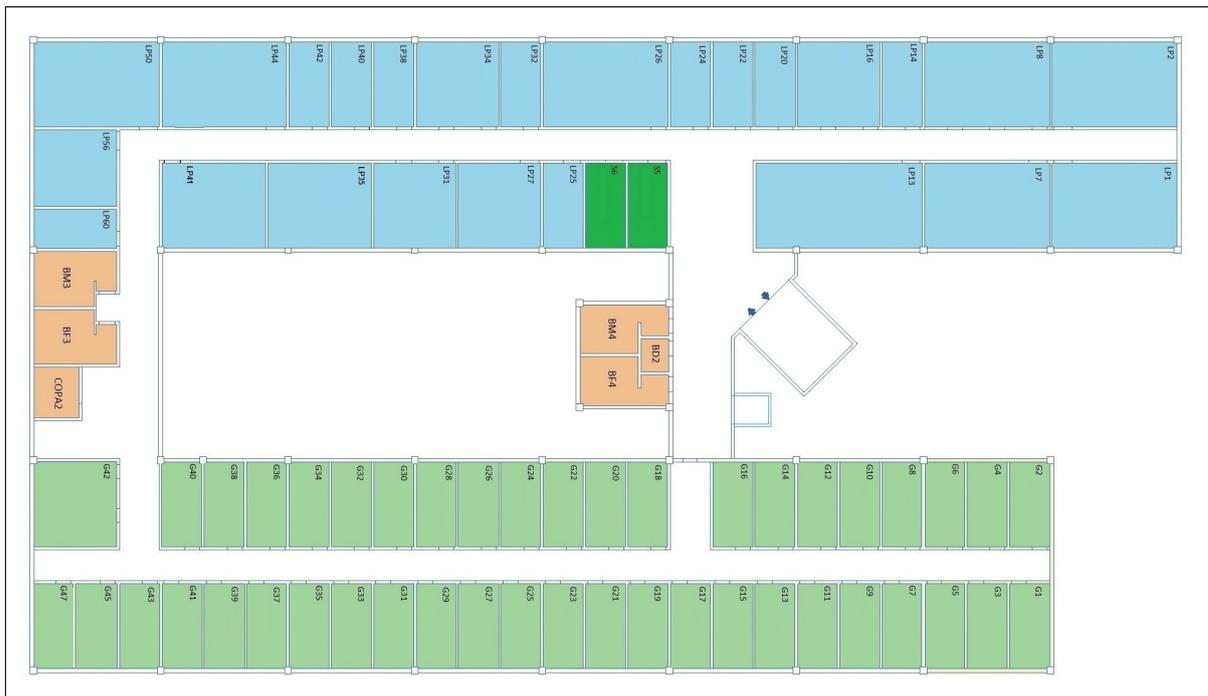
Neste capítulo serão apresentadas as metodologias utilizadas para a realização dos experimentos do Protótipo Vestível para Auxílio a PDVs e do método NSGA2CGP. Os experimentos do protótipo consistem em uma etapa de validação e uma etapa de avaliação. Para o método NSGA2CGP, serão apresentados os parâmetros utilizados, funções de erro, elementos estruturantes e métricas de desempenho para avaliação.

### **6.1 Validação e Avaliação do Protótipo Vestível de Auxílio a PDVs**

Os experimentos realizados com o protótipo tiveram o propósito de servir como validação do sistema embarcado desenvolvido com base no Modelo Abstrato e Metodologia de Customização e como avaliação, em ambientes não controlados, por pessoas com deficiências visuais. Para fins explicativos, o protótipo utilizado para os experimentos foi a segunda versão desenvolvida, a mochila.

#### **6.1.1 Validação**

Na etapa de validação, o protótipo vestível foi submetido a percursos em ambientes controlados, realizados pelo próprio autor deste trabalho e um estudante voluntário. Os ambientes nos quais a validação ocorreu foram os pisos inferior e superior do Departamento de Computação da Universidade Federal de São Carlos (UFSCar), com caminhos e obstáculos naturalmente presentes nos ambientes. As Figuras 6.1 e 6.2 exibem as plantas baixas dos pisos superior e inferior, respectivamente, do Departamento de Computação, que foram utilizados para a validação.



**Figura 6.1: Planta Baixa do Piso Superior do Departamento de Computação da UFSCar.**

Com os percursos realizados nos corredores do departamento, foi verificada a capacidade do protótipo em fornecer *feedbacks* que ajudassem a evitar colisões e mantivessem o usuário do protótipo em direções seguras. Durante os percursos, o protótipo coletou imagens e dados como imagens RGB, imagens de profundidade, direções de caminhos livres e posição de obstáculos segmentados para análise posterior. Também foi contabilizado o número de *feedbacks* errados, como caminhos não encontrados, obstáculos inexistentes e colisões.

O protótipo também foi submetido a tarefas de classificação dos obstáculos presentes pelo departamento, sendo testado com uma rede neural convolucional treinada com o modelo GoogLeNet. A rede foi treinada e usada como parte do trabalho de mestrado do aluno Renan Galeane Alboi (ALBOY, 2019), com algumas classes de obstáculos divididas em estados, especificamente para servir como classificador do protótipo desta tese. Dez classes foram treinadas com uma média de 500 imagens para cada classe. As classes foram: Armário, Cadeira, Escada, Fogão, Gaveta Aberta, Gaveta Fechada, Geladeira, Mesa, Porta Aberta e Porta Fechada.

### 6.1.2 Avaliação

A avaliação do protótipo ocorreu em ambientes não controlados reais com pessoas com deficiências visuais. Graças à parceria com a Organização Não Governamental (ONG) PARA D.V. - Associação para o Apoio e a Integração do Deficiente Visual, aprovada pelo Comitê de Ética em Pesquisa (CEP) da UFSCar, sob número CAAE 68940017.0.0000.5504, foi possível



de Caxias e finalizando o percurso no ponto de partida. Durante os percursos, todos os recursos customizáveis estavam disponíveis para a customização. Nenhum detalhe do ambiente foi modificado para o experimento e as calçadas continham barracas de ambulantes, carrinhas com frutas, parquímetros, pessoas sentadas no chão, etc. Foi solicitado que tais cenários continuassem exatamente como estavam, visando submeter o protótipo a situações cotidianas. Os experimentos aconteceram durante os meses de outubro, novembro e dezembro de 2019.

O segundo ambiente não controlado foi inserido para avaliar a sensação de segurança que o protótipo transmite. Desta forma, dos sete participantes do experimento, quatro (P3, P4, P6 e P7) aceitaram o convite de ir até o Departamento de Computação da UFSCar para um experimento em local totalmente desconhecido e em outra cidade. Para que os participantes chegassem ao local do experimento, o traslado foi realizado com recursos próprios, sem qualquer ônus financeiro à ONG ou participantes dela. O experimento foi realizado em parte do piso inferior (Figura 6.2), entre a entrada principal e as escadas para o piso superior, e no piso superior inteiro (Figura 6.1). Assim como no primeiro experimento, não houve nenhuma modificação para a realização dos percursos e os transeuntes foram orientados a prosseguir normalmente pelos corredores.

Ao final do primeiro experimento, todos os participantes responderam um formulário de avaliação do protótipo e seus diversos aspectos. Aqueles que participaram do segundo experimento, no Departamento de Computação da UFSCar, foram questionados sobre a necessidade de mudar alguma resposta no formulário, porém nenhum deles sentiu necessidade de fazê-lo.

#### 6.1.2.1 Formulário de Avaliação

Como instrumento de avaliação, um formulário foi aplicado com o intuito de obter o *feedback* de cada participante após os experimentos avaliativos. É baseado na escala de Likert, cuja estrutura apresenta afirmações ao invés de perguntas e o entrevistado pode dizer seu grau de concordância com essas afirmações (JOSHI et al., 2015). Normalmente são usadas cinco respostas para as afirmações: Discordo Fortemente, Discordo, Indiferente, Concordo e Concordo Totalmente. Essas respostas podem ser quantificadas ao atribuir pesos a elas, como, por exemplo, 1 para "Discordo Fortemente", 2 para "Discordo", 3 para "Indiferente", 4 para "Concordo" e 5 para "Concordo Totalmente".

O formulário de avaliação contém vinte e seis itens divididos em três seções diferentes. No final, uma seção com cinco recursos customizáveis foi aplicada para registrar as preferências individuais de cada participante. As três seções são "Customização", "Meios de Saída/*Feedback*" e "Ergonomia". Treze questões compõem a primeira seção e avaliaram a percepção dos par-

participantes sobre suas experiências com a customizabilidade do protótipo. A seção "Meios de Saída/Feedback" é composta de oito questões sobre as formas de *feedback* do protótipo. Os aspectos físicos do protótipo que envolvem conforto, facilidade de vestir e utilizar e a sua aceitação foram avaliados nas cinco questões da seção "Ergonomia". O formulário de avaliação aplicado para os participantes está no Apêndice A. A análise das respostas será quantificada com os pesos das alternativas, além de ter sua confiabilidade calculada com o coeficiente  $\alpha$  de Cronbach, que realiza a correlação entre as variâncias das respostas coletadas (HORA; MONTEIRO; ARICA, 2010).

## 6.2 Método NSGA2CGP

O objetivo do experimento foi gerar filtros morfológicos otimizados para correção de imagens de profundidade para serem usados no sistema embarcado do protótipo vestível, ajudando a evitar problemas de segmentação posteriormente. Por se tratar de um protótipo vestível, alimentado por uma fonte de energia limitada, a otimização também foi feita em relação à complexidade dos filtros. Portanto, os filtros necessitaram ser otimizados para dois objetivos: baixo erro ( $f_e$ ) entre entrada e saída esperada e baixa complexidade ( $f_c$ ) dos elementos estruturantes, para minimizar o impacto no custo computacional. Assim, o NSGA2CGP foi aplicado para o problema de otimização definido na Eq. 6.1, sendo  $X$  o vetor composto das variáveis de decisão  $X = (x_1, x_2, \dots, x_n)^T$ , com  $n = 2$ , por se tratar de um problema de dois objetivos.

$$\begin{aligned} \text{minimizar} \quad & F(X) = (f_e(X), f_c(X))^T \\ \text{sujeito a} \quad & g_1(X) \equiv 0 \leq f_e(X) \leq 1, \\ & g_2(X) \equiv f_c(X) > 0 \end{aligned} \tag{6.1}$$

### 6.2.1 Parâmetros de Execução

O método foi implementado no software Octave tanto no Windows 10 quanto no Linux Mint 19.1. As sessões de otimização foram executadas em um computador de mesa com processador Intel Core i7 8700 K a 5 Ghz e 32 GB de memória a 3000 MHz. Os principais parâmetros e suas variações são apresentados na Tabela 6.1.

Alguns parâmetros foram definidos empiricamente após diversas execuções preliminares do método implementado. Foi observado, por exemplo, que os operadores lógicos AND/OR/-NOT não agregaram qualquer benefício nos resultados preliminares, portanto não foram usados nos experimentos principais. De maneira semelhante, o número de linhas de cada cromossomo

**Tabela 6.1: Parâmetros utilizados.**

<b>Parâmetro</b>	<b>Utilizado</b>
<b>Imagens</b>	10 pares
<b>Elementos Estruturantes</b>	40 EEs
<b>População</b>	100 e 1000
<b>Gerações</b>	100 e 1000
<b>Colunas</b>	100, 500 e 1000
<b>Levels-back</b>	2, COLS/2 e COLS
<b>Taxa de Mutação</b>	5%, 15% e 25%
<b>Tipo de Gene</b>	<i>Free</i> e <i>MorphPair</i>
<b>Operadores Lógicos</b>	<i>NO</i>
<b>Estratégia de Mutação</b>	2N e <i>Elitist</i>
<b>Funções de Erro</b>	SSIM, RMSE; CCOEF, RMSECC, RMSESSIM

não resultou em melhorias nos resultados, sendo definido como um. Os números de gerações e tamanhos da população testados preliminarmente variaram entre 50 e 2000, tendo seus resultados agrupados entre os valores 100 e 1000, e, assim, ambos valores foram submetidos aos experimentos.

O conjunto de operadores do experimento foi composto de dilatação, erosão e *nop*, um operador customizado que repassa uma das suas entradas baseado no gene ser ímpar ou par, como explicado na Seção 4.3. O conjunto de elementos estruturantes, por sua vez, foi composto de quarenta formas, entre quadrados, círculos, losangos, octágono, retângulo e linha. Cada uma das formas teve seu tamanho variando em tamanhos ímpares entre 1 e 7, enquanto as linhas também tiveram seus ângulos variados em 0, 45, 90 e 135 graus.

Para o processo de otimização foi usado um conjunto de treinamento de dez pares de imagens de profundidade com dimensão 640x480 pixels. As imagens de entrada foram capturadas pelo Kinect XBOX 360 e pela Intel Realsense D435i e cada uma delas foi duplicada e manualmente corrigida como resultado esperado. O conjunto de pares de imagens de treinamento refletiu os cenários nos quais o protótipo vestido seria utilizado, buscando maximizar seu funcionamento. As imagens capturadas, entretanto, não se baseiam em uma localidade apenas. O protótipo foi testado tanto em ambiente interno quanto ambientes externos, ao passo que, durante análise preliminar, foi observado que diferentes perspectivas de uma mesma cena não alteram positivamente a qualidade dos filtros gerados ou sua complexidade. Portanto, dez imagens foram suficientes para atender ambientes internos da universidade (corredores, obstáculos baixos e elevados, etc.) e externos pela cidade (calçadas com transeuntes, orelhões, parquíme-

tros, etc.).

## 6.2.2 Funções de Objetivo

O experimento otimizou dois objetivos, minimizando a relação "erro x complexidade". O primeiro objetivo foi o erro entre as imagens de saída dos filtros e as imagens esperadas. Por serem tarefas de minimização, o erro ideal é 0, ou seja, os filtros seriam capazes de produzir exatamente a saída esperada quando aplicados às imagens de entrada defeituosas. Cinco funções de erro foram usadas para o experimento, três delas já conhecidas na literatura (CHAI; DRAXLER, 2014; WANG et al., 2004) e duas novas funções introduzidas pelo autor desta tese e baseadas nas funções já existentes. A primeira função usada,  $f_1$ , foi a de Similaridade Estrutural ou *Structural Similarity* (SSIM), que avalia a proximidade das estruturas entre a imagem de saída  $\mathbf{x}$  e saída esperada, normalizadas por  $C_1$  e  $C_2$ ,  $\mathbf{y}$  (Eq. 6.2).

$$f_e(X) = f_1 = SSIM(x, y) = 1 - \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (6.2)$$

A segunda função usada,  $f_2$ , realiza o cálculo da raiz da média quadrada, também conhecida como *Root Mean Squared Error* (RMSE), entre o par de imagens (Eq. 6.3).

$$f_e(X) = f_2 = RMSE(x, y) = \sqrt{\frac{\sum_{i=1}^N (x_i - y_i)^2}{N}} \quad (6.3)$$

A terceira função,  $f_3$ , é uma normalização do coeficiente de correlação (CCOEF) dentro do intervalo [0,1] entre as imagens de saída e esperada (Eq. 6.4).

$$f_e(X) = f_3 = CCOEF(x, y) = 1 - \frac{(x \cdot y)}{\sqrt{x \cdot x} \sqrt{y \cdot y}} \quad (6.4)$$

A quarta função,  $f_4$ , foi proposta pelo autor desta tese e é baseada na raiz quadrada do produto entre RMSE e CCOEF. A motivação dessa função é dada pelo fato de CCOEF ter um bom desempenho com imagens binárias, de acordo com a literatura. Ao juntá-la com RMSE, será verificado se é possível estender o uso de CCOEF em imagens de escala de cinza. A função proposta é chamada  $R^2MSECC$  (Eq. 6.5).

$$f_e(X) = f_4 = R^2MSECC(x, y) = \sqrt{RMSE(x, y) * CCOEF(x, y)} \quad (6.5)$$

Por fim, foi proposta uma segunda função como  $f_5$ , baseada na raiz quadrada do produto entre a similaridade estrutural e a raiz da média quadrada ( $R^2MSESSIM$ ). Essa função foi criada

como resultado de observações empíricas dos desempenhos de RMSE e SSIM em diferentes cenários e isso pode ser um fator positivo quando diferentes imagens são avaliadas (Eq. 6.6).

$$f_e(X) = f_5 = R^2 MESSIM(x, y) = \sqrt{RMSE(x, y) * SSIM(x, y)} \quad (6.6)$$

Por se tratar de uma otimização de filtros morfológicos que devem eliminar as áreas de distâncias desconhecidas que apresentam pixels pretos, o parâmetro de execução *BPWEIGHT* foi definido como verdadeiro e, dessa forma, o NSGA2CGP aplica uma penalidade proporcional para cada indivíduo da população que gerar uma saída com pixels pretos. Assim, para cada função de erro  $f_i$ , o valor final é acrescido com seu próprio percentual de pixels pretos em relação ao total  $N$  (Eq. 6.7). É importante destacar que, apesar de serem cinco funções para o primeiro objetivo, o NSGA2CGP foi executado com cada função como primeiro objetivo para todas as combinações de parâmetros de execução.

$$f_i(x) = f_i(x) * \left( \left( \frac{\sum_{j=1}^N x_j = 0}{N} \right) + 1 \right) \quad (6.7)$$

O segundo objetivo otimizado foi a complexidade,  $COMPLEX(f_c)$ . A complexidade de um filtro morfológico é definida pela soma dos tamanhos de todos os elementos estruturantes usados pelo filtro (Eq. 6.8), sendo  $S$  o número de elementos estruturantes usados pelo filtro  $x$  e  $SE$  é um elemento estruturante específico.

$$f_c(X) = COMPLEX(x) = \sum_{i=1}^S |SE(x_i)| \quad (6.8)$$

A complexidade é muito importante no processo de seleção de um indivíduo para ser usado como filtro em um dispositivo real. Quanto maior a complexidade, mais operações matemáticas serão executadas, consumindo mais tempo e energia. Ao considerar uma fonte de energia limitada, o *trade-off* entre erro/qualidade e complexidade deve ser bem pensado, o que pode justificar o uso de filtros otimizados com baixa complexidade, mas com pequenos defeitos de correção de profundidade.

### 6.2.3 Outras Abordagens e Métricas de Performance

Junto com os filtros morfológicos otimizados, dois outros algoritmos normalmente encontrados na literatura para, entre outras aplicações, corrigir imagens de profundidade foram implementados e submetidos à captura de erro, consumo de energia e tempo de execução. O *Inpainting* foi implementado baseado no artigo original de Telea (TELEA, 2004), que aplica o

conceito de reconstrução de áreas danificadas considerando como tal os pixels pretos. O filtro recursivo mediano também pode ser usado para corrigir dados de profundidade desconhecida, aplicando, recursivamente, o enchimento dos pixels pretos (LAI et al., 2011; VARSHNEY; TYAGI, 2015).

Para avaliar a performance dos filtros otimizados, foram empregadas duas métricas: Hipervolume ou *Hypervolume* (HV) e *Inverted Generational Distance* (IGD). O hipervolume é o cálculo dos volumes entre cada solução não dominada de um *Pareto Front* (PF) em relação a um ponto de referência que corresponde ao pior cenário possível. O valor de HV (Eq. 6.9) é a soma de cada hypervolume  $v \in PF$  (LIAO et al., 2018). Considerando que o caso desta tese é um problema de minimização, os melhores conjuntos não dominados devem apresentar os maiores valores, indicando que as soluções estão mais distribuídas entre os objetivos. O IGD é uma modificação da métrica *Generational Distance* (GD), com o intuito de calcular a distância média entre o vetor de objetivos de um conjunto de Pareto  $P$  e os objetivos de um conjunto aproximado de Pareto  $A$  (Eq. 6.10). Nesse caso, quanto menor o valor do IGD, melhor é a sua convergência e diversidade (BEZERRA; LÓPEZ-IBÁÑEZ; STÜTZLE, 2017)

$$HV = \sum_{i=1}^{|PF|} v_i \quad (6.9)$$

$$IGD(A, P) = \frac{\sum_{i=1}^{|A|} dst(A_i, P)}{|A|}, \text{ onde} \quad (6.10)$$

$$dst(A_i, P) = \min_{j=1}^{|P|} \sqrt{\sum_{m=1}^M \left( \frac{obj_m(A_i) - obj_m(P_j)}{obj_m^{max} - obj_m^{min}} \right)^2}$$

## 6.3 Considerações Finais

Este capítulo apresentou os experimentos realizados para validar e avaliar o protótipo contendo o sistema embarcado multimodal e interativo para auxílio customizável à navegação de PDVs. As atividades de validação e avaliação ocorreram em dois ambientes diferentes, sendo que os experimentos de avaliação foram feitos em ambientes não controlados e reais por PDVs. Também foram apresentados os experimentos conduzidos com o método NSGA2CGP, parâmetros utilizados, funções de objetivo e métricas de avaliação. No próximo Capítulo serão apresentados os resultados obtidos com esses experimentos e observações relacionadas, juntamente

com as discussões dos resultados.

# Capítulo 7

## RESULTADOS E DISCUSSÕES

---

---

Neste capítulo serão apresentados os resultados obtidos durante e após a realização dos experimentos descritos no Capítulo 6. A apresentação dos resultados será dada em duas partes, assim como nos experimentos, começando pelo protótipo vestível e seguido pelo método NSGA2CGP. Os resultados serão discutidos ao longo do capítulo, de acordo com sua apresentação.

### 7.1 Protótipo Vestível para Auxílio a PDVs

#### 7.1.1 Validação

As sessões de validação do protótipo verificaram o funcionamento do sistema desenvolvido com base no Modelo Abstrato e Metodologia de Customização. Os percursos realizados nos ambientes de experimento permitiram atestar a capacidade do protótipo de informar direções de caminhos livres e/ou obstáculos, com baixa ocorrência de erros. Os percursos foram feitos duas vezes e cada um durou cerca de 15 minutos. Alguns exemplos de processamento do protótipo são exibidos na Figura 7.1. O sistema embarcado no protótipo gerou as imagens após segmentar os caminhos livres e enviar o *feedback*.

Foram coletadas imagens de 2.229 *feedbacks* de caminhos livres e/ou obstáculos durante o experimento de validação. Cada *feedback* gerou sete imagens (colorida, profundidade, segmentações, etc.) para análise posterior, totalizando pouco mais de 15.600 imagens coletadas. A análise de erros dos caminhos livres e obstáculos foi feita manualmente pelo autor nas 2.229 imagens coloridas com o(s) caminho(s) desenhado(s), como as exibidas na Figura 7.1. Os obstáculos detectados foram extraídos e armazenados já segmentados, sem a caixa de marcação na imagem principal, que, na existência de obstáculos, ficava sem caixa alguma, indicando ao



**Figura 7.1: Exemplos de resultados de processamento do protótipo vestível com sistema embarcado.**

autor que um obstáculo foi detectado, e então a imagem segmentada do obstáculos era analisada. Durante a validação do protótipo não houve nenhuma colisão, como esperado, tendo a análise posterior das imagens capturadas revelado que o número de obstáculos inexistentes e inexistência errada de caminhos é baixa. A análise contabilizou 63 respostas de caminho não encontrado, quando na verdade havia 89 obstáculos laterais inexistentes. Portanto, em 2.229 *feedbacks*, o protótipo apresentou 152 respostas erradas ou 6.8% de erro. Considerando que não houve nenhuma colisão, o percentual de 93.2% de sucesso na detecção de caminhos livres e/ou obstáculos é satisfatório para afirmar que o protótipo cumpre essa tarefa.

Em relação à classificação de obstáculos, o protótipo foi usado para classificar os dez obstáculos listados na Seção 6.1.1, que estão presentes no departamento. Cada obstáculo foi classificado conforme o percurso passava por ele, resultando, dessa forma, nos percentuais médios de probabilidade apresentados na Tabela 7.1.

Durante todas as passagens pelos obstáculos, o protótipo foi capaz de classificá-los corretamente, variando apenas o percentual de probabilidade da classe. Nos obstáculos com estados,

**Tabela 7.1: Percentuais de probabilidade dos obstáculos classificados.**

<b>Classe</b>	<b>Média</b>
<b>Armário</b>	74.61%
<b>Cadeira</b>	66.25%
<b>Escada</b>	99.96%
<b>Fogão</b>	65.06%
<b>Gaveta Aberta</b>	65.93%
<b>Gaveta Fechada</b>	61.04%
<b>Geladeira</b>	82.82%
<b>Mesa</b>	70.31%
<b>Porta Aberta</b>	80.86%
<b>Porta Fechada</b>	63.87%

como portas e gavetas, o nível de certeza do protótipo diminuiu, levando o protótipo a inverter os estados das classes, porém ainda acertando a classe e classificando o obstáculo entre "Parecer", "Provável" e apenas dizer o nome do obstáculo, que representa probabilidades altas, como no caso das escadas.

A customização do sistema embarcado no protótipo foi explorada durante o experimento. Todos os recursos customizáveis foram alterados, tanto via fala quanto via teclado braile, durante os percursos. Foi observado que o aumento das distâncias mínimas e máximas torna o sistema embarcado mais sensível às movimentações corporais do usuário, já que quanto maior o alcance, mais adiantado o sistema tende a ficar em relação aos obstáculos laterais da direção que o usuário estará indo. Porém, com um ritmo de caminhada maior, foi observado que distâncias mínimas de 2 metros dão mais segurança contra colisões. De forma geral, o protótipo mostrou a capacidade esperada de customizabilidade.

### 7.1.2 Avaliação

Na primeira avaliação, cada participante da ONG PARA-DV usou o protótipo em ambiente aberto nas ruas de Araraquara/SP, como explicado no capítulo anterior, por cerca de duas horas cada. Durante o percurso, os participantes sentiram a necessidade de fazer a customização dos seguintes recursos:

- Distância Máxima, para adiantar ou atrasar a detecção de obstáculos – Distâncias: 1 metro, 1.5 metro e 2 metros;
- Distância Mínima, para adaptar o campo de detecção de caminhos livres com o ritmo de

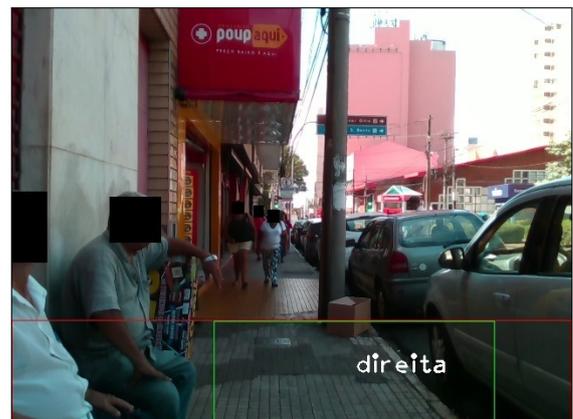
caminhada individual – Distâncias: 1 metro, 1.5 metro e 2 metros;

- *Feedback* Ativo ou Passivo, para verificar o melhor tipo para cada um;
- Velocidade de *Feedback* no modo ativo, para ajustar o conforto sensorial;
- Modo de Navegação, de acordo como cada participante prefere o funcionamento;
- Método de *Feedback*, com todos preferindo o método tátil.

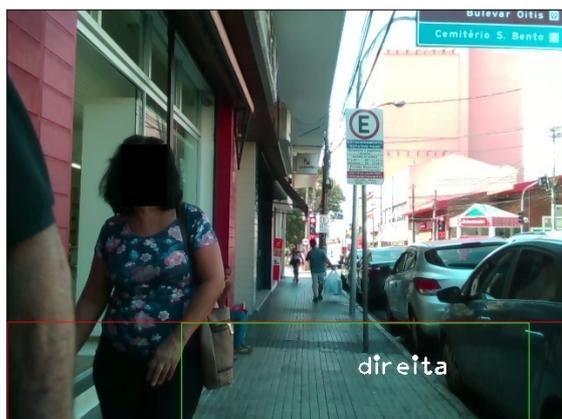
Os experimentos com PDVs em percursos de calçada, durante o horário comercial, sem qualquer preparação prévia do ambiente, demonstraram que o sistema é capaz de lidar com diversos elementos normalmente presentes quando se está caminhando em um ambiente assim. Pessoas atravessando a frente ou cruzando dos lados, em sentido contrário, foram cenas recorrentes durante cada sessão. Felizmente, o protótipo foi capaz de interpretar a situação e lidar da melhor maneira individualmente, como mostrado na Figura 7.2. Rostos foram ocultados para manter a privacidade dos transeuntes.



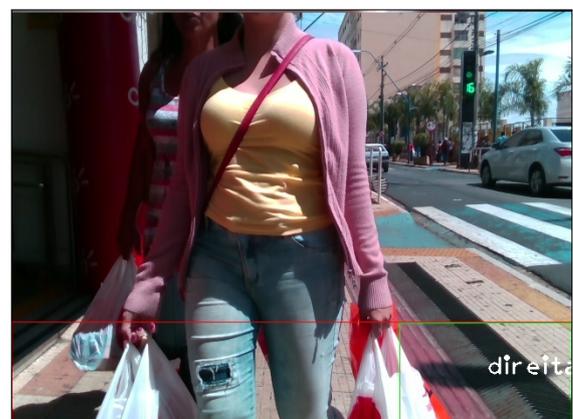
(a) Pessoa cruzando pela esquerda.



(b) Pessoas sentadas na esquerda.



(c) Pessoas cruzando pela esquerda.



(d) Pessoas atravessando a frente.

**Figura 7.2: Protótipo indicando direções seguras ao cruzar com pessoas nas calçadas. Imagens capturadas pelo próprio protótipo.**

Nos cenários de pessoas atravessando a frente de um lado para o outro, o protótipo momentaneamente indicava que não havia nenhum caminho disponível ou a presença de um obstáculo e voltava a indicar o caminho livre em seguida, após a passagem da pessoa. Obstáculos frequentemente mencionados por PDVs como problemáticos em seu cotidiano, como postes largos e finos, barracas, parquímetros, etc., foram detectados e evitados com sucesso pelo protótipo, conforme mostrado na Figura 7.3.



(a) Evitando suporte de panfleto à esquerda.



(b) Evitando poste de metal à direita.



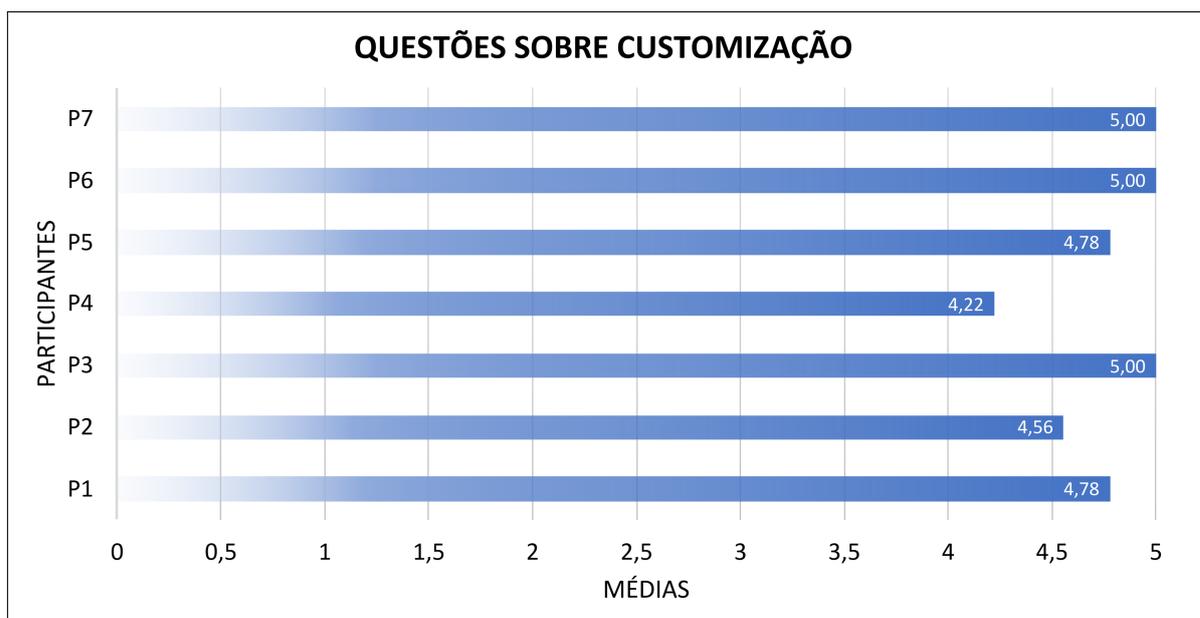
(c) Evitando poste de metal à direita.



(d) Evitando poste à direita e fachada de loja à esquerda.

**Figura 7.3: Protótipo indicando caminhos para evitar obstáculos perigosos segundo relatos de PDVs. Imagens capturadas pelo próprio protótipo.**

Ao final de cada sessão de experimento, o formulário de avaliação (Apêndice A) foi aplicado nos participantes com o objetivo de avaliar sua experiência com o protótipo na rua, os recursos oferecidos por ele, entre outros. As treze primeiras perguntas formaram o grupo de questões sobre os recursos customizáveis do sistema embarcado do protótipo e as formas de customizá-los. Na Figura 7.4 são apresentadas as médias de cada um dos participantes sobre as treze perguntas de customização. As respostas foram quantificadas de 1 a 5, conforme explicado na Seção 6.1.2.1, dado que os três grupos de questões são baseados na escala de Likert. Os participantes serão identificados de P1 a P7, para manter suas identidades anônimas.



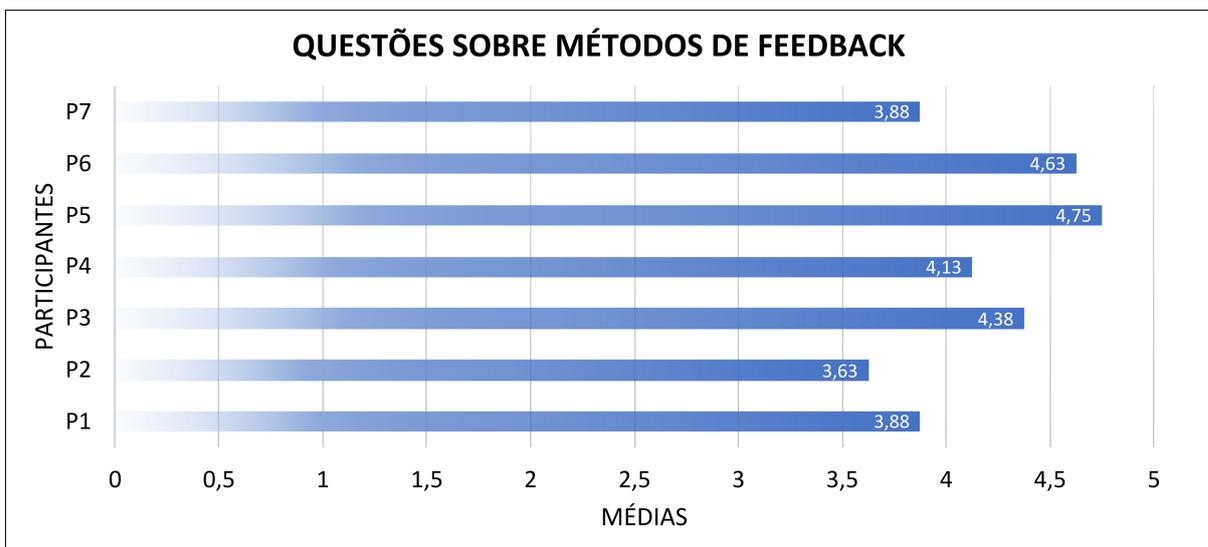
**Figura 7.4:** Médias das respostas de cada participante sobre as questões de customização.

De acordo com as respostas apresentadas na Figura 7.4, três dos sete participantes (P3, P6 e P7) concordaram, em média, totalmente com as questões apresentadas, afirmando que, ao customizar o funcionamento do protótipo, perceberam uma melhoria do uso. Os demais participantes variaram suas médias entre 4.22 e 4.78, indicando que os aspectos relacionados à customização do protótipo de fato são positivos quanto à pluralidade de preferências individuais.

Após cada pergunta, os participantes puderam falar livremente sobre o tópico que acabaram de responder e complementar sua resposta em forma de contribuição. O participante P4 complementou diversas respostas desse grupo de questões com destaque para suas respostas das questões 1.7 e 1.9, as quais foram "Discordo" e "Indiferente", respectivamente. Segundo o participante, por já existir um comando para pausar o funcionamento do protótipo, não haveria necessidade de ter um modo passivo. Já em relação à distância de detecção de caminhos livres, o participante disse ser indiferente por sempre andar no mesmo ritmo e questionou a possibilidade do próprio sistema detectar o ritmo ao invés de ser necessário informar. Essa sugestão foi muito importante e será incluída na seção sobre trabalhos futuros. O participante também chamou a atenção para a necessidade de memorização dos comandos de voz, o que pode ser um problema se houver o esquecimento. Ainda em relação aos comandos de voz, durante os experimentos alguns participantes questionaram o meio de entrada que, por limitações ferramentais, só reconhece comandos em inglês. Apesar de não terem conhecimento do idioma, os participantes informaram o que queriam customizar e o autor proferiu os comandos de fala, que prontamente customizaram o sistema. Segundo os participantes, é importante que os comandos funcionem em português também.

Os participantes P1, P3 e P5 afirmaram que poder mudar o jeito que o sistema funciona a qualquer momento é muito importante, principalmente para cenários de compartilhamento da mochila. Entretanto, os três participantes disseram que o treinamento realizado com eles antes dos experimentos foi essencial para a compreensão do protótipo, já que ele dispõe de muitos recursos. Além disso, o participante P3 sugeriu melhorar o tempo levado para customizar todos os recursos desejados quando a maioria está definida para preferências totalmente diferentes. Essa limitação já havia sido prevista e já conta com uma ideia para resolvê-la: criação de perfis de usuários. Por limitação de tempo, esse recurso será adicionado em uma versão seguinte do sistema embarcado.

O segundo grupo de questões foi sobre os métodos de *feedback* oferecidos pelo protótipo, os padrões apresentados a eles para representação de informações, etc. Esse grupo contou com oito questões e as médias de respostas por participante são apresentadas na Figura 7.5.



**Figura 7.5:** Médias das respostas de cada participante sobre as questões de métodos de *feedback*.

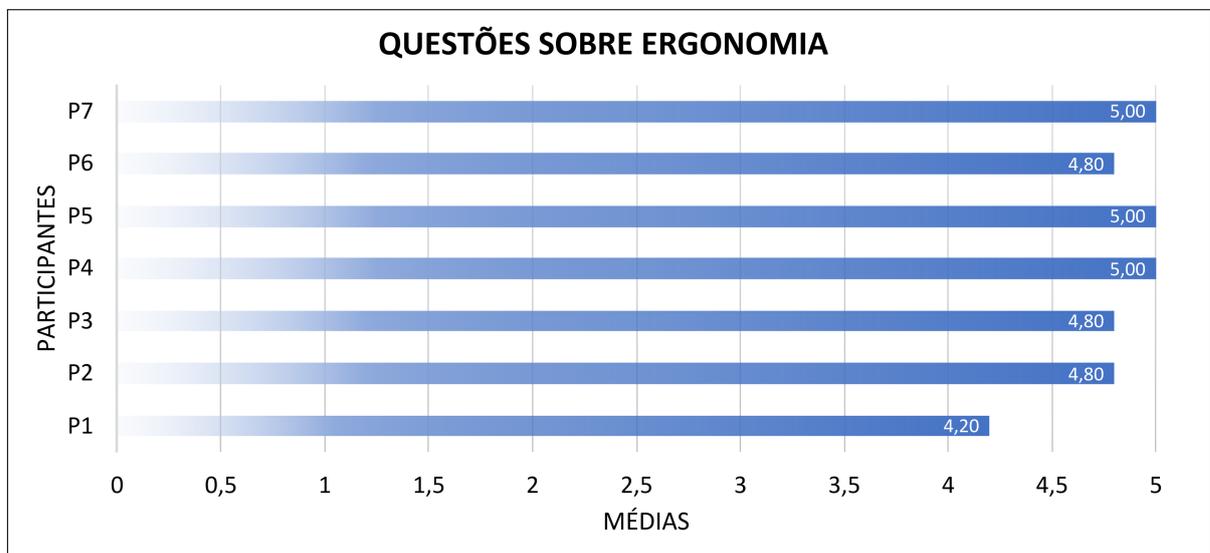
O contraste entre as médias das Figuras 7.5 e 7.4 é grande. Os participantes se mostraram muito satisfeitos em relação ao *feedback* tátil via motores de vibração, principalmente no que diz respeito ao seu posicionamento. De acordo com os participantes P7 e P2, o motor de vibração das costas pode ser mais difícil de sentir devido à sensibilidade de cada um na região onde o motor fica posicionado. Para resolver isso, o segundo modelo da segunda versão do protótipo passou a contar com pequenas almofadas para o motor das costas, fazendo com que ele seja levemente pressionado na região e possa transmitir o *feedback*, como mostrado na Figura 5.11.

De forma geral, os participantes não se adaptaram bem aos fones de ouvido. Segundo a maioria, os *feedbacks* sonoros podem ser úteis, mas, mesmo com fones de ouvido de condução óssea que não obstruem os ouvidos, prestar atenção nas respostas sonoras do protótipo e

nos sons do ambiente torna a experiência confusa e até mesmo estressante. O participante P6 exemplificou a divisão de atenção entre *feedbacks* e ambientes comparando com a sensação de falar ao telefone enquanto presta atenção na televisão. Com exceção da identificação de um obstáculo, os participantes não consideram que os sons possam complementar as vibrações e agregar benefícios, mas apenas ser uma alternativa.

Os participantes P1 e P2 têm problemas auditivos e contam com uma audição extremamente baixa. Eles externaram grande dificuldade em compreender as respostas sonoras do protótipo, principalmente aquelas em que o som é mapeado espacialmente entre esquerda e direita. Apesar disso, acreditam que os *feedbacks* sonoros devem ser oferecidos como alternativa. O participante P1 relatou, durante os percursos na rua e ao responder esse grupo de questões, que o protótipo ajudou a manter o equilíbrio em caminhar, já que a baixa audição somada à falta de visão constantemente o fazem perder o equilíbrio e eventualmente tropeçar para os lados. Esse foi um resultado inesperado e muito positivo.

O ultimo grupo de questões foi dividido em cinco perguntas relacionadas à ergonomia física do protótipo em sua segunda versão, sua aceitabilidade e facilidade de manuseio. As médias das respostas dos participantes são apresentadas na Figura 7.6.



**Figura 7.6:** Médias das respostas de cada participante sobre as questões referentes à ergonomia.

A versão em mochila do protótipo melhorou muito a percepção dos participantes, passando a ser um dispositivo passível de adoção no cotidiano. O fato de ser possível usar a mochila normalmente enquanto o sistema embarcado funciona foi elogiado pela maioria dos participantes, que também afirmaram ser um protótipo confortável e relativamente discreto.

Alguns participantes demonstraram, porém, preocupação em relação à segurança contra

roubos ou furtos. Os participantes P6 e P3, que concordaram parcialmente com a questão 3.5, levantaram uma importante questão sobre a segurança da câmera RGB-D, que fica presa à alça horizontal no peitoral do usuário. Segundo eles, há a preocupação da câmera ser roubada já que, em teoria, basta desconectar o cabo dela e puxá-la. Uma solução que já havia sido pensada para outra finalidade seria incorporar a câmera na própria alça horizontal. Entretanto, isso se trata de um trabalho futuro.

Individualmente, quase todas as vinte e seis questões obtiveram boa média entre os sete participantes. Na Figura 7.7 são apresentadas as médias individuais para cada questão do formulário. Nela, é possível observar que vinte e quatro questões tem médias 4 ou superior.

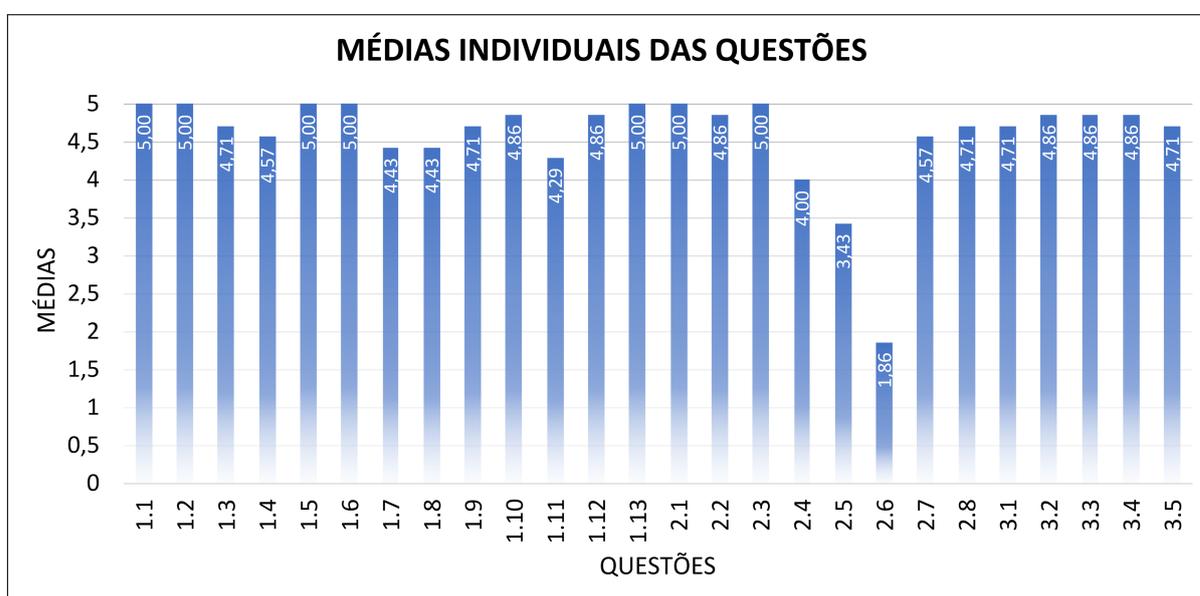


Figura 7.7: Média individual de cada questão aplicada com o formulário.

As questões 2.5 e 2.6 apresentam médias de 3,43 e 1,86, respectivamente. Na questão 2.5, é afirmado que "As respostas sonoras são de fácil compreensão". Segundo os participantes, a voz sinteticamente gerada pelo protótipo muitas vezes é "robotizada" e, mesmo customizando a velocidade das falas e/ou o espaço entre as palavras, o resultado pode ser confuso, principalmente para os participantes com baixa audição, que relataram ser impraticável para eles o uso desse método de *feedback*. Já na questão 2.6, os participantes discordaram que haja benefício no uso de respostas sonoras junto com as táteis, visto que apenas a classificação de obstáculos não tem uma representação tátil. Assim, o *feedback* sonoro não foi considerado como complementar ao tátil, mas sim alternativo.

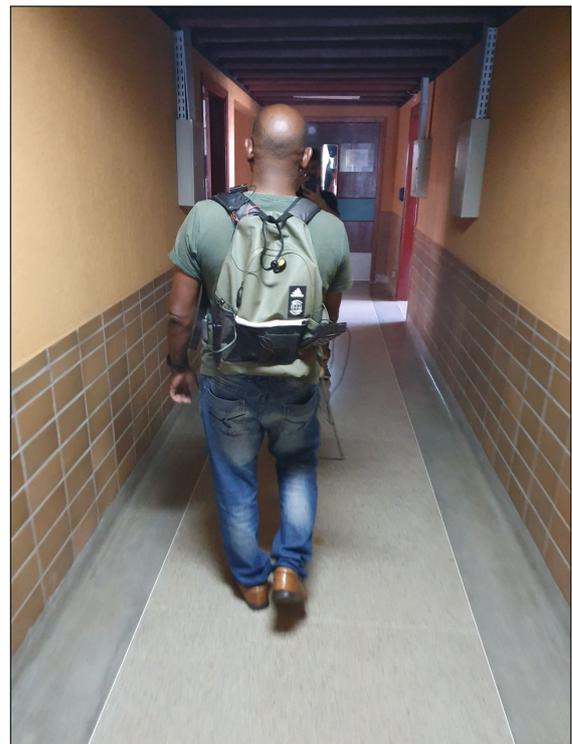
Quanto à confiabilidade dos dados levantados pelo formulário, o coeficiente alfa de Cronbach foi calculado com base em todas as questões com variância maior que zero entre os participantes. As questões 1.1, 1.2, 1.5, 1.6, 1.13, 1.14 e 1.16 obtiveram variância zero, devido

à unanimidade de concordância total. Portanto, dezenove respostas de cada um dos sete participantes foram submetidas ao teste de confiabilidade. Segundo Taber (2018), valores de  $\alpha > 0.70$  indicam boa confiabilidade interna e boa relação entre as variáveis. Para as respostas do formulário, foi obtido  $\alpha = 0.72$ , reforçando a confiabilidade dos resultados extraídos com os formulários.

O segundo experimento de avaliação foi realizado no Departamento de Computação da UFSCar. Quatro dos sete participantes do primeiro experimento foram levados até o departamento e a atividade durou pouco mais de três horas no total, com percursos concentrados no piso superior. Cada percurso foi iniciado e finalizado no mesmo local e os participantes foram instruídos sobre as direções que iriam tomar no final de cada corredor, para dar uma volta e chegar no mesmo ponto. Foi solicitado que eles escolhessem o método de *feedback* preferido e, caso desejassem, poderiam alterá-lo durante o percurso. Todos os participantes optaram por *feedback* tátil. Em relação ao ambiente, nenhum dos participantes conhecia o local do experimento e suas instruções eram de, ao final de um corredor ou bifurcação abrindo para esquerda, ir para a esquerda. Na Figura 7.8 são exibidas imagens do segundo experimento.



(a) Participante realizando percurso.



(b) Participante realizando percurso.

**Figura 7.8:** Participantes da ONG PARA-DV de Araraquara/SP realizando experimento do Departamento de Computação da UFSCar.

Os percursos foram realizados com sucesso por todos os participantes. Entretanto, foi observado que barreiras transparentes como vidros não são detectáveis pela câmera RGB-D. Uma

solução pensada para as próximas versões do protótipo é incluir um sensor ultrassônico para fazer parte dos sensores de entrada e complementar a câmera RGB-D. Um dos participantes solicitou subir e descer as escadas do departamento com o protótipo e relatou que este indicou quando ele poderia continuar subindo/descendo e quando seria necessário virar à esquerda/direita para continuar. Outro participante lidou com um obstáculo totalmente inesperado, inclusive para o autor desta tese: uma escada aberta no meio de um corredor com uma pessoa em cima dela. Aproveitando o experimento, o participante indicou as direções que era possível passar e, com auxílio da detecção de obstáculos, conseguiu atravessar a escada com sucesso. As imagens da Figura 7.9 mostram os dois momentos.



(a) Participante com obstáculo à sua frente.



(b) Participante subindo escadas com auxílio do protótipo.

**Figura 7.9: Participantes do experimento lidando com obstáculos no Departamento de Computação da UFSCar.**

Quando todos os participantes concluíram seus percursos, foi perguntado se eles gostariam de alterar alguma resposta do formulário de avaliação com base nas atividades realizadas naquele dia. Os participantes afirmaram que suas respostas continuavam as mesmas e que estavam satisfeitos com isso. Desta forma, o segundo experimento foi finalizado.

Os resultados obtidos por meio de observações e dos formulários avaliativos foram muito positivos. Existem diversas melhorias já planejadas para as próximas versões do protótipo, grande parte graças aos relatos oferecidos pelos participantes durante as duas sessões de expe-

rimentos. Todas as interações realizadas entre autor e participantes forneceram dados e informações que contribuíram com a evolução do protótipo como um todo, no sentido de tornar o projeto viável para uso cotidiano.

## 7.2 NSGA2CGP

Durante o Módulo de Pré-processamento do sistema embarcado no protótipo, as imagens de profundidade passaram pelo processo de correção das áreas de distâncias desconhecidas. As correções foram feitas por um filtro morfológico gerado e otimizado pelo método NSGA2CGP, apresentado no Capítulo 4.

O método NSGA2CGP foi avaliado sob múltiplos indicadores. Tempo de execução e consumo de energia dos filtros otimizados foram calculados para compor o processo de análise de resultados. Os parâmetros de execução utilizados foram apresentados na Seção 6.2.1, e, devido ao grande número de combinações, apenas os parâmetros dos quatro melhores treinamentos serão apresentados e discutidos. Assim, no Quadro 7.1 são apresentados os parâmetros ou *setups* dos quatro melhores resultados para cada função de objetivo/erro.

**Quadro 7.1: Principais parâmetros ou *setups* de treinamento dos quatro melhores resultados do NSGA2CGP.**

Nome	Pop.	Ger.	Cols.	Mut.	Estrat.	Levels	Gene
S1	100	1000	500	25%	2N	500	Free
S2	1000	100	500	25%	Elitist	500	Free
S3	100	1000	500	25%	Elitist	500	Free
S4	1000	100	500	25%	2N	500	Free

O melhor valor para o número de colunas foi 500. Não foram observadas melhorias significativas quando foram utilizadas 1.000 colunas por cromossomo, apenas tendo levado mais tempo para concluir o processo de treinamento. De maneira similar, a combinação de 1.000 gerações e 1000 indivíduos por população não apresentou avanços quando comparados com apenas um dos parâmetros assumindo o número maior. Entretanto, a melhor taxa de mutação foi a de 25%, com resultados melhores em relação aos demais parâmetros. Empiricamente, foi testado um treinamento com 35% de taxa, porém não gerou resultados positivos.

Cada um dos *setups* da Tabela 7.1 foi executado com cada funções de erro  $f_{(1-5)}$  apresentadas na Seção 6.2.2 e com a função de complexidade  $f_c$ , resultado em vinte treinamentos considerados os melhores resultados quando comparados às demais combinações de parâmetros. Na Tabela 7.2 são apresentados os resultados de erro e complexidade para cada *setup* e também para os erros das imagens corrigidas com *Recursive Median Filter* (RMF) e *Inpaint-*

ting. Para cada treinamento, foram extraídos três indivíduos que representam o melhor resultado apenas para o primeiro objetivo (erro), apenas para o segundo objetivo (complexidade) e uma solução factível intermediária escolhida pelo autor. Os indivíduos são identificados por  $MO_1$ ,  $MO_2$  e  $Fct$ , respectivamente. Para cada par de resultados de erro e complexidade, uma coluna é identificada como  $f_{(1-5)}$ , com  $O_1$  sendo o valor do erro e  $O_2$  a complexidade.

**Tabela 7.2: Resultado dos treinamentos para cada função de erro.**

		$f_1$		$f_2$		$f_3$		$f_4$		$f_5$	
		$O_1$	$O_2$								
S1	$MO_1$	0.0474	1387	0.0652	1561	0.1393	721	0.0899	904	0.0606	765
	$MO_2$	0.0930	10	0.1115	10	0.2753	10	0.1694	10	0.0997	10
	Fct.	0.0588	265	0.0679	221	0.1550	235	0.0999	303	0.0618	251
S2	$MO_1$	0.0584	564	0.0718	702	0.1595	1043	0.0989	932	0.0639	1565
	$MO_2$	0.0889	19	0.0968	18	0.2555	13	0.1799	26	0.1018	12
	Fct.	0.0616	237	0.0730	297	0.1715	237	0.1043	284	0.0669	248
S3	$MO_1$	0.0607	401	0.0719	432	0.1736	230	0.1034	826	0.0647	299
	$MO_2$	0.0961	19	0.1228	15	0.2555	13	0.1918	12	0.1048	10
	Fct.	0.0607	401	0.0894	169	0.1917	131	0.1142	176	0.0682	183
S4	$MO_1$	0.0494	3200	0.0701	1009	0.1566	342	0.0797	5065	0.0631	1940
	$MO_2$	0.0930	10	0.1115	10	0.2753	10	0.1694	10	0.0997	11
	Fct.	0.0578	410	0.0710	557	0.1733	110	0.0985	532	0.0639	404
	RMF	0.0558	-	0.0858	-	0.2031	-	0.1320	-	0.0692	-
	<i>Inpainting</i>	0.0662	-	0.0860	-	0.2592	-	0.0722	-	0.0647	-

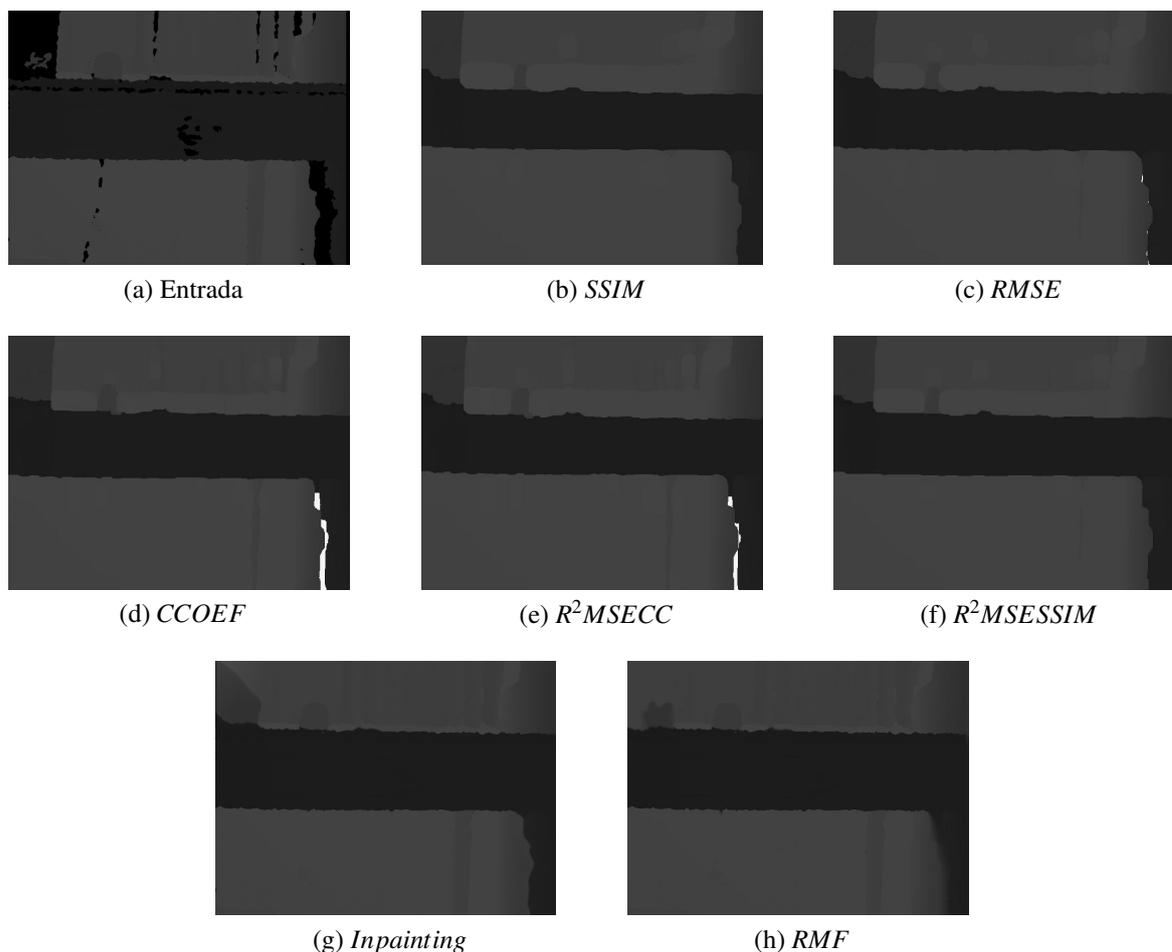
De acordo com os resultados da Tabela 7.2, foi observado que a estratégia de mutação exerce grande influência sobre a minimização dos erros, fazendo com que a estratégia 2N, representada pelos *setups* S1 e S4, tenha gerado resultados com erros menores do que a estratégia *Elitist*, com S2 e S3. A complexidade dos treinamentos apresentou grande variação entre os conjuntos e, embora a solução factível (*Fct*) de S4 tenha obtido o mesmo erro que  $MO_1$  para a função  $f_5$ , sua complexidade foi cerca de 75% menor.

O comportamento de superioridade da estratégia 2N em relação à *Elitist* se repetiu para todos os erros  $f_i$  e foi observado que a estratégia *Elitist* produziu conjuntos muito pequenos de soluções não dominadas. Isso fez com que ocorressem casos como o do *Fct* do *setup* S3 de  $f_1$ , que também é a melhor solução para o primeiro objetivo.

Em comparação com os métodos RMF e *Inpainting*, os resultados obtidos pelas soluções otimizadas com o NSGA2CGP apresentaram menores erros, embora a diferença não seja tão

grande. A complexidade desses dois métodos não foi calculada, já que não são filtros morfológicos. Além disso, a função  $f_3$  demonstrou estar totalmente fora do escopo deste trabalho, com taxas de erro altas, indicando não funcionar bem com imagens que contenham distâncias desconhecidas.

A qualidade visual dos resultados pode ser vista na Figura 7.10, tendo como exemplo uma imagem de entrada defeituosa que não foi parte do conjunto de treinamento e as saídas após ser processada pelas soluções factíveis do *setup* S4 de cada uma das funções de erro  $f_{(1-5)}$ . Os filtros gerados com *SSIM* (Figura 7.10b) e  $R^2MSESSIM$  (Figura 7.10f) atingiram o objetivo de corrigir todas as distâncias desconhecidas, enquanto *RMSE* (Figura 7.10c), *CCOEF* (Figura 7.10d) e  $R^2MSECC$  (Figura 7.10e) apresentaram falhas, que foram destacadas em branco para melhor visualização. Os filtros *Inpainting* (Figura 7.10g) e *RMF* (Figura 7.10h) também obtiveram sucesso na correção das áreas com distância desconhecida, como esperado.



**Figura 7.10:** Imagem de entrada e respectivas saídas das soluções factíveis do *setup* S4. Áreas em branco destacam as distâncias desconhecidas que não foram corrigidas.

Considerando que o método NSGA2CGP foi criado com o propósito principal de oferecer suporte à resolução de um problema específico que afeta os sistemas de navegação para PDVs,

as imagens de profundidade defeituosas, todas as soluções apresentadas na Tabela 7.2 foram exportadas para código executável em C++ via fase de "Exportar e Implantar" do NSGA2CGP. Foi desenvolvido um *wrapper* para o OpenCV em C++ que permitiu a geração de código pronto para ser executado a partir de cada solução. Os filtros foram compilados em um SoC NVIDIA Jetson TX2, da mesma família do SoC utilizado no protótipo, e executados isoladamente, enquanto os sensores internos de consumo de energia eram monitorados. Cada filtro foi executado três vezes e suas médias de consumo são apresentadas na Tabela 7.3.

**Tabela 7.3: Consumo de energia do filtros morfológicos gerados pelo NSGA2CGP e da literatura, *Inpainting* e RMF, em miliwatts (mW).**

		Otimizado				Outros	
		$f_1$	$f_2$	$f_4$	$f_5$	Inpnt.	RMF
S1	$MO_1$	1114	699	947	<b>465</b>		
	$MO_2$	48	48	48	48		
	Fct.	337	393	353	<b>289</b>		
S2	$MO_1$	721	<b>497</b>	767	1154		
	$MO_2$	48	48	48	48		
	Fct.	425	392	257	<b>361</b>	353	5833
S3	$MO_1$	473	562	1177	<b>234</b>		
	$MO_2$	48	48	48	48		
	Fct.	251	425	377	<b>226</b>		
S4	$MO_1$	1599	659	1410	<b>586</b>		
	$MO_2$	48	48	48	48		
	Fct.	273	274	400	<b>97</b>		

A função *CCOEF*, originalmente presente nos testes, foi excluída devido aos seus resultados ruins. As medições de energia reforçam os resultados da Tabela 7.2, confirmando que a estratégia *Elitist* é inferior a 2N sob as mesmas condições, já que o conjunto S4 apresentou, em geral, consumos relativamente menores que S2, enquanto S1 e S3 foram mais próximos no consumo.

Como esperado, os melhores resultados para o segundo objetivo, a complexidade, apresentaram consumos mínimos devido à baixa complexidade, porém praticamente nenhuma melhoria nas imagens de profundidade. Em média, o consumo foi igual para todos os casos, 48 mW, que é o consumo base de um programa no SoC. Foi observado que complexidades abaixo de 40 não consomem mais energia do que 48 mW, mas são filtros inúteis para o contexto deste projeto.

A função de erro proposta junto com o método NSGA2CGP,  $R^2MSESSIM$  obteve o melhor consumo de energia entre todos os filtros, consumindo apenas 97 mW na solução factível do

*setup* S4. Essa função também foi a que produziu o menor consumo entre as soluções factíveis de S1, S3 e S3, além de apresentar menor consumo em para  $MO_1$  em S1, S3 e S4, ficando atrás apenas de *RMSE* para o conjunto S2.

O método *Inpainting* consumiu uma média de 353 mW para corrigir imagens de profundidade, enquanto o RMF teve um consumo médio muito alto, 3853 mW ou 3.85W, devido a seu comportamento recursivo para corrigir as áreas desconhecidas. Apesar dos bons resultados visuais, o consumo de energia de ambos os filtros da literatura não são ideais em um cenário de fonte de energia limitada.

Além do consumo, o tempo de execução dos filtros é de suma importância para os sistemas de navegação. Em um ciclo entre aquisição de dados e envio de *feedback*, o filtro morfológico corresponde apenas a uma das várias tarefas que são executadas pelo sistema, que precisa lidar com todas elas em tempo hábil para enviar respostas para o usuário. Considerando processos como segmentação, classificação e outros, um filtro morfológico de alta performance para corrigir imagens de profundidade é o mais adequado para esse caso. Na Tabela 7.4 são apresentados os tempos de execução, em milissegundos, de cada filtro implementado no SoC NVIDIA Jetson TX2.

**Tabela 7.4: Tempo de execução dos filtros otimizados e da literatura, *Inpainting* e RMF, em milissegundos (ms).**

		Otimizado				Outros	
		$f_1$	$f_2$	$f_4$	$f_5$	Inpnt.	RMF
S1	$MO_1$	555	300	466	<b><u>236</u></b>		
	$MO_2$	4	4	5	4		
	Fct.	199	156	155	<b><u>152</u></b>		
S2	$MO_1$	286	<b><u>234</u></b>	446	545		
	$MO_2$	4	4	4	4		
	Fct.	181	177	380	<b><u>173</u></b>	244	6530
S3	$MO_1$	177	293	580	<b><u>151</u></b>		
	$MO_2$	4	4	4	5		
	Fct.	156	188	166	<b><u>146</u></b>		
S4	$MO_1$	1245	<b><u>235</u></b>	933	537		
	$MO_2$	6	3	5	5		
	Fct.	142	148	197	<b><u>67</u></b>		

Os resultados obtidos com os tempos de execução dos filtros ajudaram a escolher uma solução para se tornar parte do sistema embarcado do protótipo, apresentado no Capítulo 5, dadas as restrições de complexidade já discutidas anteriormente. A solução ideal deve consumir menos energia, mantendo bom tempo de execução e boa qualidade na correção das imagens de profundidade. A maioria das soluções com as melhores taxas de erro não apenas consomem

mais energia, como também levam mais tempo para serem executados. Por exemplo, filtro com melhor qualidade  $MO_1$  para a função de erro  $SSIM$  no *setup* S4 levou aproximadamente 1.24 segundos para corrigir uma imagem. Esse desempenho torna o filtro totalmente inviável para o cenário deste trabalho.

Todas as soluções factíveis, escolhidas pelo autor, dos *setups* de treinamento tiveram tempos de execução menores que suas respectivas soluções  $MO_1$ . As soluções otimizadas com a função  $R^2MSESSIM$  mais uma vez superaram todas as outras, agora no quesito tempo de execução, com uma média de 67 ms na solução factível do conjunto S4, que foi o melhor tempo geral entre todas as soluções factíveis. A função também produziu as melhores soluções factíveis e também  $MO_1$  para S1, S2 e S3, ficando apenas atrás de  $RMSE$  na melhor solução  $MO_1$  para o conjunto S4.

Os métodos da literatura não apresentaram bons resultados. O método *Inpainting* levou, em média, cerca de 244 ms para completar o processo de filtragem, enquanto o RMF levou 6.5 segundos. Desta forma, é possível afirmar que ambos métodos não são adequados, até onde foi possível testá-los, para serem usados em um sistema de navegação para PDVs. Portanto, a solução escolhida para ser parte do sistema de navegação desta tese foi a solução factível do *setup* S4, otimizada com a função  $f_5$ ,  $R^2MSESSIM$ . No Quadro 7.2 é apresentada a lista de entradas, elementos estruturantes, nós e operações usadas pela solução escolhida. O código C++ para OpenCV gerado pelo NSGA2CGP dessa solução pode ser encontrado no Apêndice D.

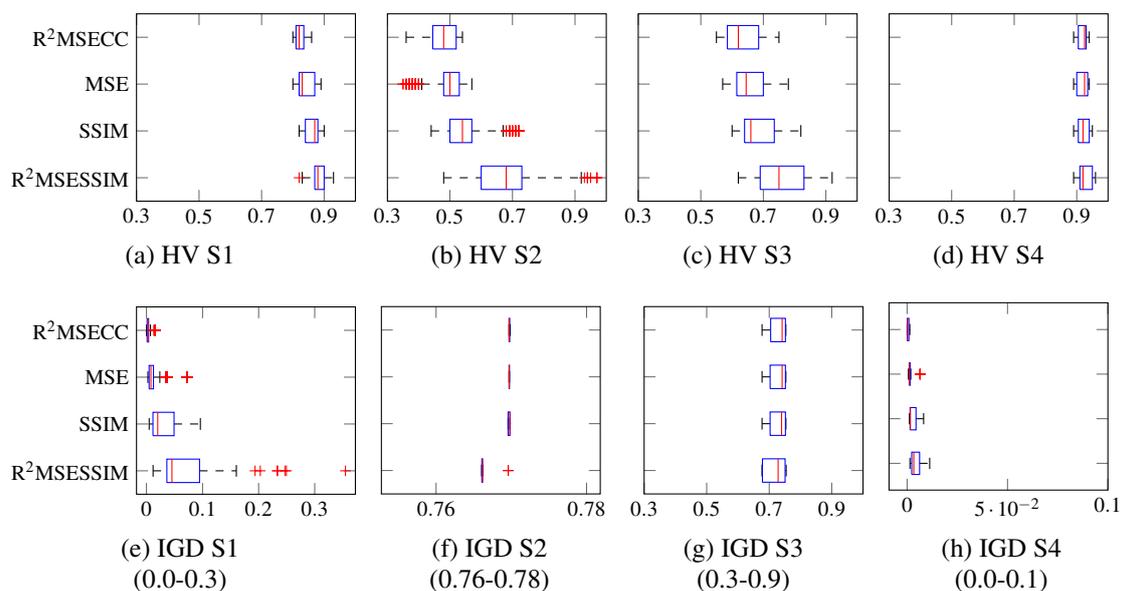
Por fim, os conjuntos de treinamento S1, S2, S3 e S4 para cada uma das funções de erro foram submetidos à avaliação de desempenho com as métricas de Hipervolume e IGD, como explicado na Seção 6.2.3, sobre os testes de desempenho do NSGA2CGP. Novamente, a função  $CCOEF$  não fez parte da avaliação, pois já se sabia que seu desempenho seria baixo. Os resultados são apresentados na Figura 7.11, com os valores normalizados no intervalo [0,1] para que seja possível compreender melhor as proporções.

Como esperado, as soluções de S1 e S4 acompanharam as tendências dos demais resultados e apresentaram performances melhores que S2 e S3, tanto para o Hipervolume quanto para o IGD. No caso do Hipervolume, foi observado que S1 e S4 apresentaram grande distribuição entre seus indivíduos devido aos altos valores de hipervolume para todas as funções de erro. O *setup* S4 se destacou com uma média acima de 0.9 para todos os casos, tendo a função  $R^2MSESSIM$  obtido os melhores valores entre todas as outras funções e todos os outros *setups*. Para o IGD, S2 e S3 também foram superados por S1 e S4, com valores muito próximos de zero. Entretanto, a função  $R^2MSESSIM$  se apresentou pior do que as outras funções dentro de

**Quadro 7.2: Solução factível escolhida do conjunto S4 treinado com a função de erro  $f_5$ . Nó é uma referência à identificação dos nós usados\* e operação é a tarefa do nó.**

Nó	Operação	Nó	Operação
1	Imagem de Entrada	45	nop(33,41)
2	quadrado(1x1)	51	ero(44,34)
4	losango(2)	53	nop(45,2)
5	retângulo(1x3)	55	dil(53,42)
12	quadrado(3x3)	66	dil(45,55)
20	linha(3,90°)	69	nop(42,22)
22	quadrado(5x5)	73	nop(51,69)
28	linha(5,0°)	83	ero(20,5)
32	quadrado(7x7)	111	nop(20,5)
33	círculo(7,9)	117	nop(111,66)
34	losango(8)	126	dil(32,66)
41	linha(7,135°)	127	nop(83,73)
42	ero(1,28)	163	dil(126,127)
43	ero(12,4)	384	dil(117,163)
44	dil(42,43)	<b>Saída</b>	<b>384</b>

\*Nós não usados pela saída foram omitidos.



**Figura 7.11: Resultados de desempenho dos conjuntos S1, S2, S3 e S4 de acordo com seus Hiper-volumes (a-d) e IGDs(e-h).**

cada *setup* individual. Apesar da diferença de IGD entre todas as funções de S1 e S4 ser muito baixa, a investigação desse comportamento ainda poderá ser feita futuramente.

## 7.3 Considerações Finais

Este capítulo apresentou os resultados obtidos durante os experimentos apresentados no Capítulo 6. O protótipo vestível para auxílio de pessoas com deficiências visuais foi validado, após ter sido desenvolvido com base no Modelo Abstrato e Metodologia de Customização introduzidos por essa tese, no Departamento de Computação da UFSCar, tendo todos os seus recursos sido testados. Também foi avaliado via relatos e formulário de avaliação após experimentos com PDVs da ONG PARA-DV nas ruas de Araraquara/SP e no Departamento de Computação da UFSCar. Os resultados revelaram que o protótipo foi capaz de auxiliar diferentes perfis de PDVs em ambientes variados. Por fim, o método NSGA2CGP apresentou ótimos resultados no cumprimento de seu objetivo de gerar filtros morfológicos de baixa taxa de erro e baixa complexidade para correção de imagens de profundidade, garantindo baixo consumo e tempo de execução com a função de erro proposta nesta tese,  $R^2MSESSIM$ . No próximo Capítulo serão apresentadas as conclusões deste trabalho, suas limitações técnicas e os trabalhos que serão realizados após este.

# Capítulo 8

## CONCLUSÕES

---

---

Abordar tecnologicamente as limitações ocasionadas pela falta de visão, com o intuito de melhorar a vida de uma pessoa com deficiência visual, é um desafio grande quando se leva em consideração a adoção e o uso de uma solução proposta. O cotidiano de PDV possui uma variedade de tarefas consideradas triviais para aqueles que podem contar com a visão, mas que podem causar acidentes a eles pela falta da mesma, como, por exemplo, desviar de obstáculos suspensos. Com ajuda da tecnologia, esse tipo de problema pode ser mitigado aplicando técnicas e sensores para detectar obstáculos e/ou caminhos seguros e ajudar as pessoas a evitar colisões. Isso evidenciou outros temas importantes, não explorados antes dentro do contexto de sistemas assistivos de navegação, como a pluralidade de preferências individuais de PDV e as formas de resolver isso.

A literatura sobre sistemas de navegação para PDV apresenta vários trabalhos com foco em detecção de caminhos livres, obstáculos, localização e mapeamento, entre outros. Também há trabalhos com o desenvolvimento focado em sugestões coletadas de PDV, mas a solução final apresenta pouca ou nenhuma flexibilidade para atender diferentes necessidades e preferências individuais de cada usuário.

Esta tese apresentou os esforços de pesquisa no desenvolvimento de um Protótipo Vestível Multimodal e Interativo para Auxílio Customizável à Navegação de Pessoas com Deficiências Visuais, baseado em um Modelo Abstrato de Sistemas de Navegação Customizáveis e uma Metodologia de Customização de Recursos, também apresentados nesta tese. O objetivo é fornecer maior grau de independência à PDV, de acordo com suas necessidades individuais, na forma de um sistema embarcado em uma mochila, que pode ser customizado diretamente pelo usuário por um ou vários meios de entrada e possui mais de um método de *feedback*.

Adicionalmente, foi explorado o problema de áreas de distâncias desconhecidas em ima-

gens de profundidade, que podem colocar em risco a confiabilidade dos sistemas de navegação. Para resolver esse problema, foi criado o método multiobjetivo evolucionário NSGA2CGP, que gera filtros morfológicos otimizados para corrigir tais áreas, minimizando o erro e a complexidade dos filtros sem exercer impacto no desempenho geral de um sistema. Também foram propostas duas novas funções de objetivo/erro,  $R^2MSESSIM$  e  $R^2MSECC$ . O método e as funções foram implementados e passaram por uma série de experimentos que revelaram que o NSGA2CGP é apto para gerar filtros morfológicos otimizados em relação ao erro (qualidade) e à complexidade, com boa diversidade e convergência, principalmente usando a função  $R^MSESSIM$ .

O protótipo foi submetido a experimentos de validação, para atestar seu funcionamento considerando o desenvolvimento baseado no Modelo Abstrato e Metodologia de Customização. Para a avaliação, foi firmada uma parceria com a ONG PARA-DV de Araraquara/SP, que ajudou na evolução deste projeto, e alguns de seus membros fizeram dois experimentos de percursos em ambientes não controlados, seguidos do preenchimento de um formulário de avaliação sobre o protótipo e suas funcionalidades.

Os resultados mostraram que o protótipo é capaz de prover independência à PDV de acordo com suas necessidades individuais, haja vista que as questões específicas sobre a customizabilidade do protótipo alcançaram nota máxima e que todos os grupos de questões obtiveram médias acima de 4 pontos, numa escala de 5, com boa confiabilidade das respostas, de acordo com o coeficiente alfa de Cronbach. Os experimentos realizados também mostraram que o protótipo consegue lidar com diferentes ambientes e diferentes iluminações, já que a primeira série de experimentos foi realizada nas ruas, com iluminação natural abundante, e a segunda no Departamento de Computação da UFSCar, um ambiente totalmente desconhecido para os participantes e mais escuro.

## 8.1 Atendimento às proposições de pesquisa

Em relação às hipóteses apresentadas nesta tese, é possível afirmar que, com exceção da segunda, todas foram comprovadas após o desenvolvimento do protótipo. Sobre a primeira hipótese, "*A possibilidade de customização de um sistema de navegação por parte de um usuário com perda de visão pode ser um fator que torne positiva as experiências do usuário, fazendo com que a adaptação de uma pessoa seja mais rápida ao sistema ao customizá-lo para funcionar da maneira mais conveniente às suas próprias percepções, por meio de uma metodologia de customização*", foi visto que os usuários avaliaram positivamente o quesito de customizabi-

lidade do sistema. Ao customizar o sistema da forma desejada, os participantes relataram maior facilidade para usar o protótipo.

A segunda hipótese apresentada, "*A redundância de formas de interação de entrada do usuário por diferentes métodos, como voz e tátil, torna o sistema de navegação mais acessível para os diferentes perfis de usuários com deficiência visual, já que passa a ser possível escolher o método mais conveniente para a interação*", pôde ser parcialmente comprovada. Devido a limitações de qualidade de software, o reconhecimento de fala *offline* no idioma português apresentou baixa qualidade e optou-se pelo reconhecimento de falas em inglês. Os participantes da ONG PARA-DV afirmaram não conhecer o inglês, porém avaliaram que é muito importante ter opções diferentes para interagir com o protótipo, já que nem todos sabem ler o braile no teclado do protótipo, e as interações de fala realizadas pelo autor a pedido dos participantes se mostrou útil.

A terceira hipótese, "*A representação de informações por diferentes métodos de feedback permite que PDVs com outras deficiências sensoriais, como baixa audição, possam utilizar o sistema e receber as mesmas informações que são enviadas via som, utilizando a customização do sistema para alterar o método de feedback*", foi comprovada. Apesar de a maioria dos participantes ter optado pelo *feedback* tátil, para dois deles a questão foi de necessidade, pois ambos apresentam graus severos de baixa audição e não eram capazes de compreender o *feedback* acústico do protótipo. Outro resultado importante no sentido da comprovação dessa hipótese veio da melhoria do equilíbrio de um dos participantes com baixa audição, que relatou perder o equilíbrio na rua ocasionalmente, e o *feedback* tátil mitigou esse problema inesperadamente.

Por fim, a última hipótese apresentada, "*As imagens de profundidade, oriundas de câmeras RGB-D usadas por sistemas de navegação para PDVs, contendo distâncias desconhecidas, podem ser eficientemente corrigidas por meio de filtros morfológicos, otimizados em relação à qualidade e complexidade, e podem prevenir problemas de segmentação de caminhos e/ou obstáculos que poderiam representar risco à confiabilidade do sistema*", foi comprovada após os experimentos realizados indicarem que o método NSGA2CGP, criado para corrigir o problema das distâncias desconhecidas em imagens de profundidade, é capaz de produzir soluções otimizadas tanto para corrigir as imagens com qualidade quanto para manter a baixa complexidade e não gerar impacto significativo no consumo de energia ou na performance geral de um sistema de navegação.

## 8.2 Limitações e Trabalhos Futuros

Esta tese apresenta algumas limitações devido ao escopo do projeto, limitação de ferramenta ou tempo. As limitações encontradas serão resolvidas por meio de trabalhos futuros, apresentados no decorrer desta seção. Propostas de trabalhos futuros também serão apresentadas a partir de pedidos de PDVs.

A primeira limitação deste projeto diz respeito às rotas dos percursos realizados com o protótipo vestível. O sistema embarcado do protótipo, por ora, não é capaz de traçar rotas até um local predefinido e guiar PDVs até um destino. Baseado nessa limitação, futuramente será adicionado o conceito de Interadores Remotos, cuja função será de receber dados de uma fonte externa, como a internet ou um módulo de GPS interno, processá-los e enviar, caso necessário, o *feedback* para o usuário. Desta forma, será possível para as próprias PDVs solicitarem a navegação até um destino específico e serem guiados, em segurança, até lá. Os Interadores Remotos também poderão servir para outros fins além de rotas, visando atender às solicitações de funcionalidades que os membros da ONG PARA-DV fizeram, como, por exemplo, receber informações de quando o seu ônibus está chegando, entre outros. Logo, os Interadores Remotos podem ser vistos como a futura expansão de conectividade do protótipo vestível.

O protótipo também apresenta a limitação de não detectar as sarjetas. Essa é uma limitação perigosa, já que os *feedbacks* podem levar os usuários para a rua, com carros transitando. Essa limitação já era conhecida pelos autores e os participantes dos experimentos foram orientados a não deixar de prestar atenção na bengala, que, além de indicar obstáculos imediatos e fora do alcance mínimo do protótipo, também indicaria a sarjeta. Após o fim deste trabalho, a detecção de sarjetas será desenvolvida e incorporada ao protótipo via projeto de iniciação científica na instituição onde o autor leciona. Paralelo a esse projeto, será também proposto um projeto para adicionar a funcionalidade de reconhecimento de faixas de pedestres e orientação para atravessá-las.

O reconhecimento de comandos de fala atualmente suporta apenas o inglês, uma vez que o desenvolvimento do protótipo foi feito para funcionar totalmente independente da internet, e as ferramentas disponíveis de detecção e reconhecimento de sentenças em português não possuem a qualidade desejada. Como trabalho futuro para resolver essa limitação, será explorada a introdução de conectividade no protótipo vestível e o reconhecimento de fala poderá ser feito remotamente, em português, com o uso de *Application Programming Interfaces* (APIs), por exemplo, do Google.

Esta tese também não aborda a detecção de trajetória de objetos dinâmicos. Esse tema de

pesquisa foi desenvolvido paralelamente a esta tese por um outro aluno de doutorado do mesmo grupo. Futuramente, essa pesquisa com foco em objetos dinâmicos poderá ser integrada ao protótipo vestível e aumentar o grau de segurança do usuário.

O compartilhamento do protótipo pode requerer a customização de muitos recursos do sistema para atender às necessidades do usuário a cada vez que elas mudarem. Futuramente será implementada a criação e alteração de perfis de usuário, permitindo a troca de perfil quando uma pessoa com preferências diferentes utilizar o protótipo.

Várias câmeras RGB-D, como a utilizada neste projeto, contêm sensores inerciais internos capazes de fornecer dados úteis sobre deslocamento em diferentes eixos. Como trabalho futuro, esses sensores inerciais serão explorados para ajustar, de forma automática, a distância mínima e máxima do protótipo, sem que seja necessária a interação do usuário para aumentar ou diminuí-la. Os sensores inerciais também farão parte de um mecanismo motorizado de ajuste automático do ângulo de elevação da câmera baseado nas distâncias mínimas.

## 8.3 Produção Científica

O desenvolvimento desta tese gerou as seguintes publicações:

Cordeiro, N. H., Dourado, A. M. B., Da Silva Quirino, G. & Pedrino, E. C. (2017). *A data fusion architecture proposal for visually impaired people*. **In Proceedings - 2016 29th SIBGRAPI Conference on Graphics, Patterns and Images**, SIBGRAPI 2016.

Dourado, A. M. B. & Pedrino, E. C. (2018). *Embedded Navigation and Classification System for Assisting Visually Impaired People*. **13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018)**.

Dourado, A. M. B. & Pedrino, E. C. (2020). *Automatic generation of high performance morphological filters to fix missing data in depth images on real-time embedded systems for visually impaired people*. **PRZEGLĄD ELEKTROTECHNICZNY**, 2020(1), 113–117.

Dourado, A. M. B. & Pedrino, E. C. (2020). *Multi-objective Cartesian Genetic Programming optimization of morphological filters in navigation systems for Visually Impaired People*. **Applied Soft Computing**, Volume 89, Abril/2020.

A. M. B. Dourado & E. C. Pedrino, "Evolutionary Approach for Automatic Generation of Multi-Objective Morphological Filters for Depth Images in Embedded Navigation Systems," in IEEE Latin America Transactions, vol. 18, no. 07, pp. 1320-1326, Julho/2020.

## REFERÊNCIAS

---

---

- ABBOUD, S. et al. EyeMusic: Introducing a "visual" colorful experience for the blind using auditory sensory substitution. *Restorative Neurology and Neuroscience*, IOS Press, Department of Medical Neurobiology, The Institute for Medical Research Israel-Canada, Faculty of Medicine, The Hebrew University of Jerusalem, Jerusalem, Israel., v. 32, n. 2, p. 247–257, jan 2014.
- ADI, I. P.; KUSUMA, H.; ATTAMIMI, M. Blind People Guidance System using Stereo Camera. In: *2019 International Seminar on Intelligent Technology and Its Applications (ISITIA)*. [S.l.]: IEEE, 2019. p. 298–303. ISBN 978-1-7281-3749-0.
- ADVANI, S. et al. A Multitask Grocery Assist System for the Visually Impaired: Smart glasses, gloves, and shopping carts provide auditory and tactile feedback. *IEEE Consumer Electronics Magazine*, v. 6, n. 1, p. 73–81, jan 2017. ISSN 2162-2248.
- AL-KHALIFA, S.; AL-RAZGAN, M. Ebsar: Indoor guidance for the visually impaired. *Computers & Electrical Engineering*, Elsevier Ltd, v. 54, p. 26–39, aug 2016. ISSN 00457906.
- ALADREN, A. et al. Navigation Assistance for the Visually Impaired Using RGB-D Sensor With Range Expansion. *IEEE Systems Journal*, p. 1–11, jan 2014.
- ALBA, E.; COTTA, C. Multi-objective Optimization. In: *Evolutionary Algorithms*. 1. ed. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2017. cap. 5, p. 165–182.
- ALBOY, R. G. *Técnicas de reconhecimento de imagem para incorporação em ferramentas de auxílio a deficientes visuais*. 83 p. Tese (Mestrado) — Universidade Federal de São Carlos, 2019.
- BANGAR, S.; NARKHEDE, P.; PARANJAPE, R. Vocal Vision for Visually Impaired People. *The International Journal Of Engineering And Science (IJES)*, v. 2, n. 3, p. 1–7, jan 2013.
- BAY, H. et al. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, v. 110, n. 3, p. 346–359, jun 2008. ISSN 10773142.
- BEZERRA, L. C.; LÓPEZ-IBÁÑEZ, M.; STÜTZLE, T. An empirical assessment of the properties of inverted generational distance on multi- and many-objective optimization. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 10173 LNCS, p. 31–45, 2017. ISSN 16113349.
- BRAMEIER, M.; Banzhaf, W. *Linear Genetic Programming*. Boston, MA: Springer US, 2007. 323 p. (Genetic and Evolutionary Computation). ISSN 1098-6596. ISBN 978-0-387-31029-9.

- BRANKE, J. et al. *Multiobjective optimization: Interactive and evolutionary approaches*. [S.l.]: Springer Science & Business Media, 2008. v. 5252. 489 p.
- BRASIL. *Lei Brasileira de Inclusão da Pessoa com Deficiência*. 2015. 1–31 p. Disponível em: <[http://www.planalto.gov.br/ccivil\\_03/\\\_ato2015-2018/2015/lei/l13146.htm](http://www.planalto.gov.br/ccivil_03/\_ato2015-2018/2015/lei/l13146.htm)>. Acesso em: 15/04/2020.
- CEIPIDOR, U. et al. RadioVirgilio/Sesamonet: An RFID-based navigation system for visually impaired. *Mobile Guide '06 . . .*, p. 1–6, 2006.
- CHAI, T.; DRAXLER, R. R. Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, v. 7, n. 3, p. 1247–1250, 2014. ISSN 19919603.
- CHEN, L.; GUO, B.-l.; SUN, W. Obstacle Detection System for Visually Impaired People Based on Stereo Vision. *2010 Fourth International Conference on Genetic and Evolutionary Computing*, IEEE, p. 723–726, jan 2010. ISSN 978-0-7695-4281-2.
- CHEN, Q. et al. Smartphone Based Outdoor Navigation and Obstacle Avoidance System for the Visually Impaired. In: *International Conference on Multi-disciplinary Trends in Artificial Intelligence*. [S.l.: s.n.], 2019. p. 26–37.
- CHENG, R. et al. Visual Localization of Key Positions for Visually Impaired People. In: *2018 24th International Conference on Pattern Recognition (ICPR)*. [S.l.]: IEEE, 2018. v. 2018-Augus, p. 2893–2898. ISBN 978-1-5386-3788-3. ISSN 10514651.
- CHU, P. M. et al. Flood-fill-based object segmentation and tracking for intelligent vehicles. *International Journal of Advanced Robotic Systems*, v. 16, n. 6, p. 1–11, 2019. ISSN 17298814.
- CSURKA, G. et al. Visual categorization with bags of keypoints. In: *In Workshop on Statistical Learning in Computer Vision, ECCV*. [S.l.: s.n.], 2004. p. 1–22.
- DAKOPOULOS, D.; BODDHU, S. K.; BOURBAKIS, N. A 2D Vibration Array as an Assistive Device for Visually Impaired. In: *2007 IEEE 7th International Symposium on BioInformatics and BioEngineering*. [S.l.]: IEEE, 2007. p. 930–937. ISBN 1-4244-1509-8.
- DAKOPOULOS, D.; BOURBAKIS, N. Preserving visual information in low resolution images during navigation of visually impaired. In: *Proceedings of the 1st ACM international conference on Pervasive Technologies Related to Assistive Environments - PETRA '08*. New York, New York, USA: ACM Press, 2008. p. 1. ISBN 9781605580678.
- DAKOPOULOS, D.; BOURBAKIS, N. G. Wearable Obstacle Avoidance Electronic Travel Aids for Blind: A Survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, v. 40, n. 1, p. 25–35, jan 2010.
- DALAL, N.; TRIGGS, B. Histograms of Oriented Gradients for Human Detection. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. [S.l.]: IEEE, 2005. v. 1, p. 886–893. ISBN 0-7695-2372-2.
- DEB, K. et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 2, p. 182–197, 2002. ISSN 1089778X.

- DEB, S. et al. A novel approach of assisting the visually impaired to navigate path and avoiding obstacle-collisions. In: *2013 3rd IEEE International Advance Computing Conference (IACC)*. [S.l.]: IEEE, 2013. p. 1127–1130. ISBN 978-1-4673-4529-3.
- DINGLI, A.; MERCIÉCA, I. Multimedia Interfaces for People Visually Impaired. In: *Advances in Design for Inclusion*. [S.l.: s.n.], 2016. p. 487–495. ISBN 9783319419619.
- DONATI, M. et al. An Aid System for Autonomous Mobility of Visually Impaired People on the Historical City Walls in Lucca, Italy. In: *22nd Mediterranean Conference on Control and Automation*. [S.l.]: IEEE, 2020. p. 379–411. ISBN 978-3-030-16450-8\_16.
- DOURADO, A. M. B.; PEDRINO, E. C. *Embedded Navigation and Classification System for Assisting Visually Impaired People*. 2018. 516–523 p.
- DUARTE, K. S. *SmartGuia: Shopping Assistant for Blind People*. 265 p. Tese (Doutorado) — University of Coimbra, 2014.
- GARDUÑO-RAMÓN, M. A. et al. A new method for inpainting of depth maps from time-of-flight sensors based on a modified closing by reconstruction algorithm. *Journal of Visual Communication and Image Representation*, v. 47, p. 36–47, aug 2017. ISSN 10473203.
- GERHARDT, T. E.; SILVEIRA, D. T. *Métodos de Pesquisa*. 1. ed. Porto Alegre: Editora da UFRGS, 2009. 120 p. ISSN 1677-5449. ISBN 9788538600718.
- GHILARDI, M. C.; JACQUES, J. C. S.; MANSSOUR, I. Crosswalk Localization from Low Resolution Satellite Images to Assist Visually Impaired People. *IEEE Computer Graphics and Applications*, v. 38, n. 1, p. 30–46, jan 2018. ISSN 0272-1716.
- GOLDMAN, B. W.; PUNCH, W. F. Reducing wasted evaluations in cartesian genetic programming. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. [S.l.: s.n.], 2013. v. 7831 LNCS, n. 3, p. 61–72. ISBN 9783642372063.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006. ISBN 013168728X.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016.
- HE, H. et al. Wearable Ego-Motion Tracking for Blind Navigation in Indoor Environments. *IEEE Transactions on Automation Science and Engineering*, v. 12, n. 4, p. 1181–1190, 2015. ISSN 15455955.
- HE, K. et al. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 770–778, 2016. ISSN 1664-1078.
- HORA, H. R. M. da; MONTEIRO, G. T. R.; ARICA, J. Confiabilidade em Questionários para Qualidade: Um Estudo com o Coeficiente Alfa de Cronbach [Reliability in Questionnaires for Quality: a study with the Cronbach's alpha Coefficient]. *Produto & Produção*, v. 11, n. 2, p. 85–103, 2010.
- IAKOVIDIS, D. K. et al. *Digital Enhancement of Cultural Experience and Accessibility for the Visually Impaired*. [S.l.: s.n.], 2020. 237–271 p. ISBN 9783030164508.

- IBGE. *Pesquisa nacional de saúde 2013: Ciclos de vida - Brasil e grandes regiões*. Rio de Janeiro, 2015. 90 p. Disponível em: <<https://biblioteca.ibge.gov.br/index.php/biblioteca-catalogo?view=detalhes&id=294525>>.
- INTEL. *Intel® RealSense™ Depth Camera D435i*. 2018. Disponível em: <<https://www.intelrealsense.com/depth-camera-d435i/>>. Acesso em: 11/12/2019.
- IPIÑA, D. López-de; LORIDO, T.; LÓPEZ, U. Indoor Navigation and Product Recognition for Blind People Assisted Shopping. In: *Lecture Notes in Computer Science*. [S.l.: s.n.], 2011. v. 6693 LNCS, p. 33–40. ISBN 9783642213021.
- JAFRI, R.; KHAN, M. M. User-centered design of a depth data based obstacle detection and avoidance system for the visually impaired. *Human-centric Computing and Information Sciences*, Springer Berlin Heidelberg, v. 8, n. 1, p. 14, dec 2018. ISSN 2192-1962.
- JÉGOU, H.; DOUZE, M.; SCHMID, C. Product Quantization for Nearest Neighbor Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 33, n. 1, p. 117–128, jan 2011. ISSN 0162-8828.
- JOSHI, A. et al. Likert Scale: Explored and Explained. *British Journal of Applied Science & Technology*, v. 7, n. 4, p. 396–403, jan 2015. ISSN 22310843.
- KATZ, B. F. G. et al. NAVIG: augmented reality guidance system for the visually impaired. *Virtual Reality*, v. 16, n. 4, p. 253–269, jun 2012. ISSN 1005501202136.
- KATZSCHMANN, R. K.; ARAKI, B.; RUS, D. Safe local navigation for visually impaired users with a time-of-flight and haptic feedback device. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, v. 26, n. 3, 2018. ISSN 15344320.
- KEAOMANEE, Y.; YOUNGKONG, P. RGB-D Depth inpainting with Color Guide Inverse Distance Weight. In: *2019 4th International Conference on Information Technology (InCIT)*. [S.l.]: IEEE, 2019. p. 249–253. ISBN 978-1-7281-1019-6.
- KING, A. D. Inertial navigation - Forty years of evolution. *Gec Review*, v. 13, n. 3, p. 140–149, 1998. ISSN 0267-9337.
- KOMMEY, B.; HERRMAN, K.; ADDO, E. O. A Smart Vision Based Navigation Aid for the Visually Impaired. *Asian Journal of Research in Computer Science*, v. 4, n. 3, p. 1–8, nov 2019. ISSN 2581-8260.
- KOZA, J. R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992. ISBN 0-262-11170-5.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems 25 (NIPS2012)*, p. 1–9, 2012.
- KUMAR, A. et al. An embedded system for aiding navigation of visually impaired persons. *Current Science*, v. 104, n. 3, p. 302–306, jan 2013.
- LAHAV, O. Improving orientation and mobility skills through virtual environments for people who are blind : Past research and future potential. *International Journal of Child Health and Human Development*, v. 7, n. 4, p. 349–355, jan 2014. ISSN 9780704915459.

- LAI, K. et al. A large-scale hierarchical multi-view RGB-D object dataset. In: *2011 IEEE International Conference on Robotics and Automation*. [S.l.]: IEEE, 2011. p. 1817–1824. ISBN 978-1-61284-386-5. ISSN 10504729.
- LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, 1998. ISSN 00189219.
- LESZCZUK, M.; BARAN, R.; SKOCZYLAS, L. Public Transport Vehicle Detection Based on Visual Information. *Communications in Computer and Information Science*, Springer International Publishing, Cham, v. 429, n. Chapter 2, p. 16–28, jan 2014. ISSN 978-3-319-07568-6.
- LI, B. et al. ISANA: Wearable Context-Aware Indoor Assistive Navigation with Obstacle Avoidance for the Blind. In: *Lecture Notes in Computer Science*. [S.l.: s.n.], 2016. v. 9914 LNCS, p. 448–462. ISBN 9783319488806.
- LI, B. et al. Vision-Based Mobile Indoor Assistive Navigation Aid for Blind People. *IEEE Transactions on Mobile Computing*, v. 18, n. 3, p. 702–714, 2018. ISSN 15580660.
- LIAO, Q. et al. A comparative study on evolutionary multi-objective optimization algorithms estimating surface duct. *Sensors (Switzerland)*, v. 18, n. 12, 2018. ISSN 14248220.
- LIN, Z.; DAVIS, L. Shape-Based Human Detection and Segmentation via Hierarchical Part-Template Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 32, n. 4, p. 604–618, apr 2010. ISSN 0162-8828.
- LOWE, D. Object recognition from local scale-invariant features. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. [S.l.]: IEEE, 1999. cs, p. 1150–1157 vol.2. ISBN 0-7695-0164-8. ISSN 0-7695-0164-8.
- MANCINI, A.; FRONTONI, E.; ZINGARETTI, P. Mechatronic System to Help Visually Impaired Users During Walking and Running. *IEEE Transactions on Intelligent Transportation Systems*, v. 19, n. 2, p. 649–660, feb 2018. ISSN 1524-9050.
- MEKHALFI, M. L. et al. Recovering the sight to blind people in indoor environments with smart technologies. *Expert Systems with Applications*, Elsevier Ltd, v. 46, p. 129–138, mar 2016. ISSN 09574174.
- MICROSOFT. *Kinect Sensor*. 2017. Disponível em: <<https://developer.microsoft.com/en-us/windows/kinect>>. Acesso em: 25/08/2017.
- MIETTINEN, K.; RUIZ, F.; WIERZBICKI, A. P. Introduction to Multiobjective Optimization: Interactive Approaches. In: *Multiobjective {Optimization}*. [S.l.: s.n.], 2008. p. 27–57. ISBN 978-3-540-88907-6 978-3-540-88908-3.
- MILLER, J. F. Cartesian Genetic Programming. In: *Cartesian Genetic Programming*. [S.l.]: Springer, 2011. p. 17–34.
- MOCANU, B.; TAPU, R.; ZAHARIA, T. An Obstacle Categorization System for Visually Impaired People. *2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, p. 147–154, 2015.

NEWCOMBE, R. a. et al. KinectFusion: Real-time dense surface mapping and tracking. *2011 10th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2011*, IEEE, v. 29, n. 1, p. 127–136, jan 2011. ISSN 9781457721830.

NIU, L. et al. A Wearable Assistive Technology for the Visually Impaired with Door Knob Detection and Real-Time Feedback for Hand-to-Handle Manipulation. In: *Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017*. [S.l.: s.n.], 2018. v. 2018-Janua, p. 1500–1508. ISBN 9781538610343. ISSN 2473-9936.

NVIDIA. *DATA SHEET NVIDIA Jetson TX1 System-on-Module*. [S.l.], 2016. 1–44 p. Disponível em: <<http://developer.nvidia.com/embedded/dlc/jetson-tx1-module-data-sheet>>. Acesso em: 24/02/2017.

NVIDIA. *NVIDIA Jetson Nano*. 2019. Disponível em: <<https://www.nvidia.com/pt-br/autonomous-machines/embedded-systems/jetson-nano/>>. Acesso em: 12/02/2019.

OHN-BAR, E. et al. Modeling Expertise in Assistive Navigation Interfaces for Blind People. In: *Proceedings of the 2018 Conference on Human Information Interaction & Retrieval - IUI 18*. New York, New York, USA: ACM Press, 2018. p. 403–407. ISBN 9781450349451.

OTT, P.; EVERINGHAM, M. Implicit color segmentation features for pedestrian and object detection. In: *2009 IEEE 12th International Conference on Computer Vision (ICCV)*. [S.l.]: IEEE, 2015. p. 723–730. ISBN 1550-5499.

OU, S.; PARK, H.; LEE, J. Implementation of an Obstacle Recognition System for the Blind. *Applied Sciences*, v. 10, n. 1, p. 282, dec 2019. ISSN 2076-3417.

PAIVA, S. *Technological Trends in Improved Mobility of the Visually Impaired*. Cham: Springer International Publishing, 2020. 421 p. (EAI/Springer Innovations in Communication and Computing). ISBN 978-3-030-16449-2.

PATIL, T. et al. Customizable Indoor Navigation and Location System. *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) ICONET*, v. 2, n. 04, p. 364–366, 2014.

PEDRINO, E. C. et al. Automatic construction of image operators using a genetic programming approach. In: *2011 11th International Conference on Intelligent Systems Design and Applications (ISDA)*. [S.l.]: IEEE, 2015. p. 636–641. ISBN 2164-7143.

PERKIS, T. Stack-based genetic programming. In: *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*. [S.l.]: IEEE, 1994. p. 148–153. ISBN 0-7803-1899-4.

PILLAI, S.; RAMALINGAM, S.; LEONARD, J. J. High-performance and tunable stereo reconstruction. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.]: IEEE, 2016. v. 2016-June, p. 3188–3195. ISBN 978-1-4673-8026-3. ISSN 10504729.

PISSALOUX, E. A characterization of vision systems for blind people mobility. In: *IEEE International Conference on Systems, Man and Cybernetics*. [S.l.]: IEEE, 2002. vol.4, p. 6. ISBN 0-7803-7437-1.

- POGGI, M.; MATTOCCIA, S. A Wearable Mobility Aid for the Visually Impaired based on embedded 3D Vision and Deep Learning. In: *First IEEE Workshop on ICT Solutions for eHealth (IEEE ICTS4eHealth 2016) in conjunction with the Twenty-First IEEE Symposium on Computers and Communications*. [S.l.: s.n.], 2016.
- POLI, R. et al. *A Field Guide to Genetic Programming*. [S.l.]: Lulu.com, 2008. 250 p.
- PRIYADARSHANA, Y.; WIMALARATNE, G. Sensing Environment through Mobile: A Personalized Wearable Obstacle Detection System for Visually Impaired People. *International Journal of Soft Computing and Engineering*, v. 3, n. 6, p. 81–88, 2014.
- PROVDANOV, C. C.; FREITAS, E. C. D. *Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico*. 2. ed. Novo Hamburgo: Editora Feevale, 2013. 276 p. ISBN 9788577171583.
- QI, F. et al. Structure guided fusion for depth map inpainting. *Pattern Recognition Letters*, v. 34, n. 1, p. 70–76, jan 2013. ISSN 01678655.
- RAHMAN, S. ur; ULLAH, S.; ULLAH, S. A Mobile Camera Based Navigation System for Visually Impaired People. In: *Proceedings of the 7th International Conference on Communications and Broadband Networking - ICCBN 2019*. New York, New York, USA: ACM Press, 2019. p. 63–66. ISBN 9781450362474.
- RAVIYA, K. et al. Real Time Depth Hole Filling using Kinect Sensor and Depth Extract from Stereo Images. *Oriental journal of computer science and technology*, v. 12, n. Issue 3, p. 115–122, 2019. ISSN 09746471.
- SAINARAYANAN, G.; NAGARAJAN, R.; YAACOB, S. Fuzzy image processing scheme for autonomous navigation of human blind. *Applied Soft Computing*, v. 7, n. 1, p. 257–264, jan 2007.
- SERRA, J.; SOILLE, P. *Mathematical Morphology and Its Applications to Image Processing*. Dordrecht: Springer Netherlands, 1994. v. 2. 390 p. (Computational Imaging and Vision, v. 2). ISBN 978-94-010-4453-0.
- SHAHIRA, K. C.; TRIPATHY, S.; LIJIYA, A. Obstacle Detection, Depth Estimation And Warning System For Visually Impaired People. In: *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*. [S.l.]: IEEE, 2019. p. 863–868. ISBN 978-1-7281-1895-6.
- SIDDHARTHA, B.; CHAVAN, A. P.; UMA, B. V. An electronic smart jacket for the navigation of visually impaired society. *Materials Today: Proceedings*, Elsevier Ltd, v. 5, n. 4, p. 10665–10669, 2018. ISSN 22147853.
- SOILLE, P. *Morphological Image Analysis*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. v. 49. 391 p. ISSN 09574271. ISBN 978-3-642-07696-1.
- STEREOLABS. *ZED Stereo Camera*. 2017. Disponível em: <<https://www.stereolabs.com/documentation/overview/getting-started/introduction.html>>. Acesso em: 10/09/2017.
- SZEGEDY, C. et al. Going deeper with convolutions. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.]: IEEE, 2015. p. 1–9. ISBN 978-1-4673-6964-0. ISSN 10636919.

- TABER, K. S. The Use of Cronbach's Alpha When Developing and Reporting Research Instruments in Science Education. *Research in Science Education*, Research in Science Education, v. 48, n. 6, p. 1273–1296, 2018. ISSN 15731898.
- TAPU, R.; MOCANU, B.; ZAHARIA, T. A computer vision-based perception system for visually impaired. *Multimedia Tools and Applications*, Multimedia Tools and Applications, v. 76, n. 9, p. 11771–11807, may 2017. ISSN 1380-7501.
- TELEA, A. An Image Inpainting Technique Based on the Fast Marching Method. *Journal of Graphics Tools*, IEEE, v. 9, n. 1, p. 23–34, jan 2004. ISSN 1086-7651.
- THAKOOR, K. A. et al. Attention biased speeded up robust featureS (AB-SURF): A neurally-inspired object recognition algorithm for a wearable aid for the visually-impaired. In: *2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*. [S.l.]: IEEE, 2013. p. 1–6. ISBN 978-1-4799-1604-7.
- TI, C. et al. Depth Recovery for Kinect Sensor Using Contour-Guided Adaptive Morphology Filter. *IEEE Sensors Journal*, v. 17, n. 14, p. 4534–4543, jul 2017. ISSN 1530-437X.
- TOKUI, S. et al. Chainer: a Next-Generation Open Source Framework for Deep Learning. *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*, p. 1–6, 2015.
- TONG, S.; CHANG, E. Support vector machine active learning for image retrieval. *Proceedings of the ninth ACM international conference on Multimedia MULTIMEDIA 01*, ACM Press, New York, New York, USA, v. 54, n. C, p. 107, jan 2001. ISSN 1581133944.
- TORBERT, S. *Applied Computer Science*. Cham: Springer International Publishing, 2016. 539 p. ISBN 978-3-319-30864-7.
- VARSHNEY, P.; TYAGI, A. An Enhanced Recursive Median Filter for Noise Reduction based on Randomness in Pixel Values of an Image. *International Journal of Computer Applications*, v. 113, n. 8, p. 8–10, mar 2015. ISSN 09758887.
- WANG, S.; TIAN, Y. Camera-Based Signage Detection and Recognition for Blind Persons. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. [S.l.: s.n.], 2012. v. 7383 LNCS, n. PART 2, p. 17–24. ISBN 9783642315336.
- WANG, Z. et al. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, v. 13, n. 4, p. 600–612, apr 2004. ISSN 1057-7149.
- WEINBERGER, N. et al. Public Participation in the Development Process of a Mobility Assistance System for Visually Impaired Pedestrians. *Societies*, v. 9, n. 2, p. 32, 2019. ISSN 2075-4698.
- WHO. *World report on vision*. [S.l.: s.n.], 2019. 180 p. ISBN 9789241516570.
- YIN, R. K. *Estudo de Caso - Planemaneto e Métodos*. [S.l.: s.n.], 2001. v. 2. 205 p. ISSN 1751-8113. ISBN 8573078529.

# Apêndice A

## FORMULÁRIO DE PESQUISA

Questão		Discordo Fortemente	Discordo	Indiferente	Concordo	Concordo Fortemente
		1	2	3	4	5
1.1	A possibilidade de customização do sistema pelo próprio usuário é um diferencial benéfico.	<input type="radio"/>				
1.2	O recurso de customização do sistema aumenta o grau de independência do usuário.	<input type="radio"/>				
1.3	A customização via comandos de teclado braile é fácil e útil.	<input type="radio"/>				
1.4	O teclado braile é de fácil compreensão sobre a disposição de teclas.	<input type="radio"/>				
1.5	Os comandos disponíveis no teclado braile atendem pessoas com diferentes necessidades e preferências.	<input type="radio"/>				
1.6	A livre customização entre <u>deteção de caminhos, obstáculos ou ambos</u> , beneficia o usuário.	<input type="radio"/>				
1.7	A livre customização entre modo <u>ativo ou passivo</u> beneficia o usuário.	<input type="radio"/>				
1.8	A livre customização entre resposta por <u>vibração, som ou ambos</u> , beneficia o usuário.	<input type="radio"/>				
1.9	Aumentar ou diminuir a <u>distância de deteção de caminho livre</u> permite ao usuário adaptar o sistema ao seu ritmo e ambiente.	<input type="radio"/>				
1.10	Aumentar ou diminuir a <u>distância de deteção de obstáculos</u> fornece maior sentimento de segurança ao usuário.	<input type="radio"/>				
1.11	Aumentar ou diminuir a <u>velocidade de respostas</u> oferece maior opção de conforto sensorial.	<input type="radio"/>				
1.12	A interface de <u>comandos por voz</u> é uma importante alternativa ao teclado braile.	<input type="radio"/>				
1.13	Os <u>comandos de voz</u> permitem que usuários não familiares ao braile customizem o sistema.	<input type="radio"/>				

2 – Meios de Saída/ <i>Feedback</i>		1	2	3	4	5
2.1	Os motores de vibração estão <u>posicionados</u> em lugares satisfatórios.	<input type="radio"/>				
2.2	As <u>direções de cada motor</u> , juntamente com o das costas indicando nenhum caminho/obstáculo, são suficientes.	<input type="radio"/>				
2.3	O <u>padrão de vibração para caminho livre</u> é de fácil compreensão.	<input type="radio"/>				
2.4	O <u>padrão de vibração para obstáculo</u> é de fácil compreensão.	<input type="radio"/>				
2.5	As <u>respostas sonoras</u> são de fácil compreensão.	<input type="radio"/>				
2.6	As <u>respostas sonoras complementam</u> as respostas táteis (vibração).	<input type="radio"/>				
2.7	As <u>respostas sonoras são alternativas</u> às respostas táteis (vibração).	<input type="radio"/>				
2.8	A representação métrica em vibração esquerda/frente/direita é clara e fácil de entender.	<input type="radio"/>				
3 – Ergonomia		1	2	3	4	5
3.1	O protótipo (mochila) é <u>fácil</u> de ser vestido.	<input type="radio"/>				
3.2	O protótipo (mochila) não apresenta elementos <u>obstrutivos</u> para o cotidiano.	<input type="radio"/>				
3.3	O protótipo (mochila) cumpre seu objetivo tecnológico sem perder o <u>conforto</u> .	<input type="radio"/>				
3.4	A possibilidade de utilizar o <u>compartimento interno</u> do protótipo (mochila) normalmente torna sua adoção mais fácil no cotidiano.	<input type="radio"/>				
3.5	O formato e arranjo atual da mochila é <u>socialmente aceitável</u> e <u>discreto</u> .	<input type="radio"/>				

Preferências		
1	Tipo de <i>feedback</i> .	( ) Vibração ( ) Som ( ) Ambos
2	Modo de navegação.	( ) Caminho ( ) Obst. ( ) Ambos
3	Entrada de comandos.	( ) Teclado ( ) Voz ( ) Ambos
4	Tipo de <i>feedback</i> acústico.	( ) Fala/TTS ( ) Tons
5	Modo ativo/passivo.	( ) Ativo ( ) Passivo

# Apêndice B

## TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO



### TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Prezado(a) Sr.(a) \_\_\_\_\_, portador do RG \_\_\_\_\_; lhe foi informado detalhadamente sobre a pesquisa intitulada: “Desenvolvimento de Dispositivo de Substituição Sensorial Nacional Inteligente e de Baixo Custo para a Navegação Autônoma de Deficientes Visuais”. Foi também esclarecido que a pesquisa tem objetivo estritamente acadêmico e visa o desenvolvimento de um dispositivo de substituição sensorial que permita a navegação autônoma de deficientes visuais em ambientes fechados, sendo sua participação incluída nas fases de testes de protótipos.

Uma vez participante da pesquisa, está garantida a sua desistência em qualquer momento, se assim desejar. Fica esclarecido que a participação é voluntária e não envolve qualquer vantagem financeira para nenhuma das partes (pesquisador e participante) e/ou favores de qualquer natureza que possam favorecer e/ou financiar quaisquer atividades de ambas as partes.

O projeto, bem como suas sessões de testes de protótipo, não apresenta riscos à saúde do participante, não tem cunho biológico ou médico e não são invasivos psicologicamente, limitando-se a pequenos percursos, sons e sentenças de voz, em português ou inglês. Esclarece-se também que o participante será acompanhado pelo pesquisador durante toda a sessão de testes, recebendo instruções sobre o funcionamento do sistema e qualquer outro auxílio relativo ao teste.

Os resultados desta pesquisa poderão ser divulgados para fins científicos em congressos, conferências e revistas especializadas na área e áreas relacionados, incluindo o uso de imagens descritivas, garantindo que a identidade do participante será preservada. Os testes serão gravados coletivamente ou individualmente e sua divulgação fora de meios científicos fica condicionada à permissão do participante. Além das imagens, qualquer outro dado referente à pesquisa será tratado de forma sigilosa e privada, sendo que o(a) Sr.(a) poderá solicitar informações sobre a pesquisa, inclusive após sua publicação.

Na condição de participante, declaro que entendi os objetivos, riscos e benefícios de minha participação na pesquisa e concordo em participar. Declaro também que compreendi que o projeto, por meio de seus protótipos, pode apresentar leve por ser um objeto estranho, na forma de um colete revestido que acrescentará cerca de 1Kg distribuídos entre sua frente e suas costas, para fins de equilíbrio e diminuição da sensação do pequeno peso, que o feedback sonoro do sistema é customizável à velocidade e periodicidade que cada participante desejar, entretanto casos de customizações falhas podem gerar desconforto de repetição sonora ao participante e que a qualquer momento posso interromper as sessões. Declaro que compreendi que os comandos do sistema são no idioma inglês e isso pode gerar desconforto. Por fim, estou ciente que riscos de colisões com obstáculos e paredes são extremamente baixos, mas o responsável pelos experimentos se postará próximo a mim durante os experimentos para impedir qualquer tipo de colisão.

O pesquisador me informou que o projeto foi aprovado pelo Comitê de Ética em Pesquisa em Seres Humanos da UFSCar que funciona na Pró-reitoria de Pós-Graduação e Pesquisa da Universidade Federal de São Carlos, localizada na Rodovia Washington Luiz, Km. 235 - Caixa Postal 676 - CEP 13.565-905 – São Carlos - SP – Brasil. Fone (16) 3351-8110. Endereço eletrônico: cephumanos@power.ufscar.br

São Carlos, \_\_\_\_\_ de \_\_\_\_\_ de \_\_\_\_\_

\_\_\_\_\_  
Participante da pesquisa

\_\_\_\_\_  
Me. Antonio Miguel Batista Dourado  
antonio\_dourado@dc.ufscar.br  
Depto. De Computação - UFSCar  
Rod. Washington Luís, Km 235  
Caixa Postal 676  
13565-905 São Carlos-SP

# Apêndice C

## EXEMPLO DE ESPECIFICAÇÃO DE RECURSO

Customização RID0005					
Geral	Nome:	Distância Mínima / Navegação		Nome UP:	MinDistance
	Descrição:	Define a distância para encontrar caminhos livres		Tipo:	float
	Restrição:	MinDistance >= 1; MinDistance <= MaxDistance		Padrão:	1.5
<b>Interador ITID0001</b>					
<b>Nome:</b> Tecla Aumentar Distância		<b>Tipo:</b> Botão			
<b>Descrição:</b> Ao apertar a tecla, aumenta em 0.5 metro		<b>Restrições:</b> MinDistance + 0.5 <= MaxDistance			
<b>Fab./Modelo:</b> KNUP KP-2003A		<b>Interface:</b> USB			
<b>Hardware:</b> 1a2c:2124		<b>Sucesso:</b> FID0001; FID0004; FID0007			
<b>Restrito:</b> FID0003; FID0006; FID0009		<b>Erro:</b> FID0002; FID0005; FID0008			
<b>Comando:</b> KP_KEY8					
<b>Interador ITID0002</b>					
<b>Nome:</b> Tecla Diminuir Distância		<b>Tipo:</b> Botão			
<b>Descrição:</b> Ao apertar a tecla, diminui em 0.5 metro		<b>Restrições:</b> MinDistance - 0.5 >= 1.0			
<b>Fab./Modelo:</b> KNUP KP-2003A		<b>Interface:</b> USB			
<b>Hardware:</b> 1a2c:2124		<b>Sucesso:</b> FID0001; FID0004; FID0007			
<b>Restrito:</b> FID0003; FID0006; FID0009		<b>Erro:</b> FID0002; FID0005; FID0008			
<b>Comando:</b> KP_KEY9					
<b>Interador ITID0003</b>					
<b>Nome:</b> Tecla Aumentar Distância até Max		<b>Tipo:</b> Botão			
<b>Descrição:</b> Ao apertar a tecla, iguala a distância com Distância Máxima		<b>Restrições:</b> -			
<b>Fab./Modelo:</b> KNUP KP-2003A		<b>Interface:</b> USB			
<b>Hardware:</b> 1a2c:2124		<b>Sucesso:</b> FID0001; FID0004; FID0007			
<b>Restrito:</b> -		<b>Erro:</b> -			
<b>Comando:</b> KP_KEY8					
<b>Modificadores</b>					
Nome:		KEYPRESS_THRESHOLD	Valor:		3s
<b>Interador ITID0004</b>					
<b>Nome:</b> Tecla Distância Mínima Absoluta		<b>Tipo:</b> Botão			
<b>Descrição:</b> Ao apertar a tecla, define a distância mínima como 1.0		<b>Restrições:</b> -			
<b>Fab./Modelo:</b> KNUP KP-2003A		<b>Interface:</b> USB			
<b>Hardware:</b> 1a2c:2124		<b>Sucesso:</b> FID0001; FID0004; FID0007			
<b>Restrito:</b> -		<b>Erro:</b> -			
<b>Comando:</b> KP_KEY9					
<b>Modificadores</b>					
Nome:		KEYPRESS_THRESHOLD	Valor:		3s
<b>Interador ITID0005</b>					
<b>Nome:</b> Frase para Aumentar Distância		<b>Tipo:</b> Microfone			
<b>Descrição:</b> Frase para aumentar a distância em 0.5 metro		<b>Restrições:</b> MinDistance + 0.5 <= MaxDistance ; MicOn = true			
<b>Fab./Modelo:</b> Aftershokz Sportz Titanium		<b>Interface:</b> USB			
<b>Hardware:</b> 1a2c:2124		<b>Sucesso:</b> FID0001; FID0004; FID0007			
<b>Restrito:</b> FID0003; FID0006; FID0009		<b>Erro:</b> FID0002; FID0005; FID0008			
<b>Comando:</b> "SYSTEM NAVIGATION DISTANCE INCREASE"					

Interador ITID0006				
<b>Nome:</b>	Frase para Diminuir a Distância	<b>Tipo:</b>	Microfone	
<b>Descrição:</b>	Frase para diminuir a distância em 0.5 metro	<b>Restrições:</b>	MinDistance - 0.5 >= 1.0 ; MlcOn = true	
<b>Fab./Modelo:</b>	Aftershokz Sportz Titanium	<b>Interface:</b>	USB	
<b>Hardware:</b>	1a2c:2124	<b>Sucesso:</b>	FID0001; FID0004; FID0007	
<b>Restrito:</b>	FID0003; FID0006; FID0009	<b>Erro:</b>	FID0002; FID0005; FID0008	
<b>Comando:</b>	"SYSTEM NAVIGATION DISTANCE DECREASE"			
Interador ITID0007				
<b>Nome:</b>	Frase para Aumentar Distância até Max	<b>Tipo:</b>	Microfone	
<b>Descrição:</b>	Frase para aumentar a distância para Distância Máxima	<b>Restrições:</b>	MlcOn = true	
<b>Fab./Modelo:</b>	Aftershokz Sportz Titanium	<b>Interface:</b>	USB	
<b>Hardware:</b>	1a2c:2124	<b>Sucesso:</b>	FID0001; FID0004; FID0007	
<b>Restrito:</b>	-	<b>Erro:</b>	-	
<b>Comando:</b>	"SYSTEM NAVIGATION DISTANCE MAXIMUM"			
Interador ITID0008				
<b>Nome:</b>	Frase para Distância Mínima Absoluta	<b>Tipo:</b>	Microfone	
<b>Descrição:</b>	Frase para diminuir a distância para 1.0	<b>Restrições:</b>	MlcOn = true	
<b>Fab./Modelo:</b>	Aftershokz Sportz Titanium	<b>Interface:</b>	USB	
<b>Hardware:</b>	1a2c:2124	<b>Sucesso:</b>	FID0001; FID0004; FID0007	
<b>Restrito:</b>	-	<b>Erro:</b>	-	
<b>Comando:</b>	"SYSTEM NAVIGATION DISTANCE MINIMUM"			
Feedback FID0001				
<b>Nome:</b>	Conf. Voz Sucesso	<b>Descrição:</b>	Envia mensagem de áudio confirmando a nova distância	
<b>Restrição:</b>	(FBMethod = Acoustic    FBMethod = Both) && FBAcousticType = Mono	<b>Tipo:</b>	Sucesso	
<b>Sinal:</b>	Áudio Mono	<b>Repetições:</b>	0	<b>Duração:</b> -
<b>Mensagem:</b>	Nova distância			
<b>Fab./Modelo:</b>	Aftershokz Sportz Titanium	<b>Interface:</b>	USB	<b>Hardware:</b> 1a2c:2124
Feedback FID0002				
<b>Nome:</b>	Conf. Voz Erro	<b>Descrição:</b>	Envia mensagem de áudio relatando erro	
<b>Restrição:</b>	(FBMethod = Acoustic    FBMethod = Both) && FBAcousticType = Mono	<b>Tipo:</b>	Erro	
<b>Sinal:</b>	Áudio Mono	<b>Repetições:</b>	0	<b>Duração:</b> -
<b>Mensagem:</b>	Descrição do erro			
<b>Fab./Modelo:</b>	Aftershokz Sportz Titanium	<b>Interface:</b>	USB	<b>Hardware:</b> 1a2c:2124
Feedback FID0003				
<b>Nome:</b>	Conf. Voz Restrição	<b>Descrição:</b>	Envia mensagem de áudio acusando restrição	
<b>Restrição:</b>	(FBMethod = Acoustic    FBMethod = Both) && FBAcousticType = Mono	<b>Tipo:</b>	Restrito	
<b>Sinal:</b>	Áudio Mono	<b>Repetições:</b>	0	<b>Duração:</b> -
<b>Mensagem:</b>	Detalhes da restrição			
<b>Fab./Modelo:</b>	Aftershokz Sportz Titanium	<b>Interface:</b>	USB	<b>Hardware:</b> 1a2c:2124
Feedback FID0004				
<b>Nome:</b>	Conf. Tatil Sucesso	<b>Descrição:</b>	Envia padrão de vibração sinalizando nova distância	
<b>Restrição:</b>	FBMethod = Tactile    FBMethod = Both	<b>Tipo:</b>	Sucesso	
<b>Sinal:</b>	Vibração	<b>Repetições:</b>	-	<b>Duração:</b> -
<b>Mensagem:</b>	Nova distância em padrões de vibração			
<b>Fab./Modelo:</b>	HG7881	<b>Interface:</b>	GPIO	<b>Hardware:</b> gpio73, gpio200, gpio12
Feedback FID0005				
<b>Nome:</b>	Conf. Tatil Erro	<b>Descrição:</b>	Envia padrão de vibração sinalizando erro. Duas vibrações em todos os motores.	
<b>Restrição:</b>	FBMethod = Tactile    FBMethod = Both	<b>Tipo:</b>	Erro	
<b>Sinal:</b>	Vibração	<b>Repetições:</b>	2	<b>Duração:</b> 200ms
<b>Mensagem:</b>	-			
<b>Fab./Modelo:</b>	HG7881	<b>Interface:</b>	GPIO	<b>Hardware:</b> gpio73, gpio200, gpio12, gpio168
Feedback FID0006				
<b>Nome:</b>	Conf. Tatil Restrito	<b>Descrição:</b>	Envia padrão de vibração sinalizando restrição. Vibração longa em todos os motores.	
<b>Restrição:</b>	FBMethod = Tactile    FBMethod = Both	<b>Tipo:</b>	Restrito	
<b>Sinal:</b>	Vibração	<b>Repetições:</b>	0	<b>Duração:</b> 500ms

<b>Mensagem:</b>	-		
<b>Fab./Modelo:</b>	HG7881	<b>Interface:</b>	GPIO
		<b>Hardware:</b>	gpio73, gpio200, gpio12, gpio168
<b>Feedback FID0007</b>			
<b>Nome:</b>	Conf. Som Sucesso	<b>Descrição:</b>	Envia padrão de som confirmando a nova distância. Beep duplo.
<b>Restrição:</b>	(FBMethod = Acoustic     FBMethod = Both) && FBAcousticType = Stereo	<b>Tipo:</b>	Sucesso
<b>Sinal:</b>	Som Stereo	<b>Repetições:</b>	1
		<b>Duração:</b>	250ms
<b>Mensagem:</b>	-		
<b>Fab./Modelo:</b>	Aftershokz Sportz Titanium	<b>Interface:</b>	USB
		<b>Hardware:</b>	1a2c:2124
<b>Feedback FID0008</b>			
<b>Nome:</b>	Conf. Som Erro	<b>Descrição:</b>	Envia padrão de som sinalizando erro. Beep três vezes.
<b>Restrição:</b>	(FBMethod = Acoustic     FBMethod = Both) && FBAcousticType = Stereo	<b>Tipo:</b>	Erro
<b>Sinal:</b>	Som Stereo	<b>Repetições:</b>	2
		<b>Duração:</b>	300ms
<b>Mensagem:</b>	Descrição do erro		
<b>Fab./Modelo:</b>	Aftershokz Sportz Titanium	<b>Interface:</b>	USB
		<b>Hardware:</b>	1a2c:2124
<b>Feedback FID0009</b>			
<b>Nome:</b>	Conf. Som Restrição	<b>Descrição:</b>	Envia padrão de som acusando restrição. Beep longo.
<b>Restrição:</b>	(FBMethod = Acoustic     FBMethod = Both) && FBAcousticType = Stereo	<b>Tipo:</b>	Restrito
<b>Sinal:</b>	Som Stereo	<b>Repetições:</b>	0
		<b>Duração:</b>	500ms
<b>Mensagem:</b>	Detalhes da restrição		
<b>Fab./Modelo:</b>	Aftershokz Sportz Titanium	<b>Interface:</b>	USB
		<b>Hardware:</b>	1a2c:2124



```
cv::Mat node45;
node45=node33.clone();
node45=node33.clone();
cv::Mat node53;
node53=node45.clone();
cv::Mat node1=inputimage.clone();
float node28_data[5]={1,1,1,1,1};
cv::Mat node28=cv::Mat(1,5,CV_32F,node28_data);
node28.convertTo(node28,CV_8UC1);
cv::Mat node42;
cv::morphologyEx(node1,node42,cv::MORPH_ERODE,node28);
cv::Mat node55;
cv::morphologyEx(node42,node55,cv::MORPH_DILATE,node53);
cv::Mat node66;
cv::morphologyEx(node55,node66,cv::MORPH_DILATE,node45);
cv::Mat node126;
cv::morphologyEx(node66,node126,cv::MORPH_DILATE,node32);
float node5_data[3]={1,1,1};
cv::Mat node5=cv::Mat(1,3,CV_32F,node5_data);
node5.convertTo(node5,CV_8UC1);
cv::Mat node83;
cv::morphologyEx(node5,node83,cv::MORPH_ERODE,node20);
cv::Mat node127;
node127=node83.clone();
cv::Mat node163;
cv::morphologyEx(node126,node163,cv::MORPH_DILATE,node127);
cv::Mat node384;
cv::morphologyEx(node163,node384,cv::MORPH_DILATE,node117);
cv::Mat imgout=node384.clone();
```

# Apêndice E

## PREPARAÇÃO E CONSTRUÇÃO DO SISTEMA

---

---

O processo de preparação e construção do sistema a ser embarcado em SoCs Jetson Nano leva em consideração diversos aspectos, que vão desde a instalação do sistema operacional até a construção do arquivo executável do sistema de navegação propriamente dito. Os equipamentos utilizados na versão corrente do protótipo vestível, apresentados na Seção 5.4.2.2, foram adquiridos com recursos próprios do doutorando nos diferentes sites, listados na Tabela E.1.

**Tabela E.1: Equipamentos, valores e sites.**

Equipamento	Preço	Site
NVIDIA Jetson Nano	US\$ 99,00	<a href="https://developer.nvidia.com/embedded/jetson-nano-developer-kit">https://developer.nvidia.com/embedded/jetson-nano-developer-kit</a>
Intel Realsense D435i	US\$ 199,00	<a href="https://store.intelrealsense.com/buy-intel-realsense-depth-camera-d435i.html">https://store.intelrealsense.com/buy-intel-realsense-depth-camera-d435i.html</a>
Aftershokz Sportz Titanium	US\$ 59,90	<a href="https://aftershokz.com/products/sportz-titanium-with-mic">https://aftershokz.com/products/sportz-titanium-with-mic</a>
Bateria INIU 10.000MAh	US\$ 16,99	<a href="https://www.amazon.com/gp/product/B07H6LB4J4">https://www.amazon.com/gp/product/B07H6LB4J4</a>
Teclado KNUP KP-2003A	R\$ 16,99	<a href="https://www.americanas.com.br/produto/47280579">https://www.americanas.com.br/produto/47280579</a>
Ponte-H HG7881	R\$ 6,56	<a href="https://pt.aliexpress.com/item/32959624541.html">https://pt.aliexpress.com/item/32959624541.html</a>
Motor Vibracall 1027	R\$ 7,00	<a href="https://www.casadoledgarca.com.br/motor-de-vibracao-1027-arduino">https://www.casadoledgarca.com.br/motor-de-vibracao-1027-arduino</a>

A mochila foi construída exclusivamente para esta tese e não houve custos de montagem além dos materiais utilizados, que custaram cerca de R\$ 100,00, já considerando os cabos e conectores necessários para comunicação entre dispositivos. Portanto, o custo total da versão final do protótipo é de US\$ 374,89 dólares e R\$ 150,55 reais.

## E.1 Preparação do Ambiente

As relações de procedimentos, bibliotecas e versões apresentadas neste apêndice seguem as mesmas das utilizadas no protótipo vestível. Os itens que não apresentarem versões específicas podem ter sua versão mais recente adotada, uma vez que não há dependência de versão neles.

### E.1.1 NVIDIA Jetson Nano

A placa Jetson Nano deve receber a versão 4.2 do Jetpack, contendo a distribuição Linux For Tegra 32.1 (L4T 32.1), com o kernel 4.9.140-tegra. O Jetpack 4.2 pode ser baixado a partir do seguinte link: <[https://developer.nvidia.com/jetpack-4\\_2](https://developer.nvidia.com/jetpack-4_2)>. O passo a passo para instalação do Jetpack está descrito no link: <<https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit>>.

### E.1.2 Bibliotecas

O sistema de navegação que compõe o protótipo vestível usa uma extensa lista de bibliotecas em sua operação. Atualmente, o sistema oferece suporte para duas câmeras RGB-D diferentes, sendo a Stereolabs ZED V1 e a Intel Realsense D435i, que tem suas bibliotecas incluídas na relação deste Apêndice.

Algumas das bibliotecas utilizadas têm dependências de outras bibliotecas. Nesse caso, recomenda-se a leitura dos arquivos *README* e/ou *INSTALL*, normalmente presentes nos diretórios de bibliotecas que requerem a compilação na própria Jetson Nano. A lista a seguir contém as bibliotecas e versões, caso necessárias, usadas pelo sistema de navegação.

- **libglu1-mesa-dev** (via apt-get)
- **freeglut3-dev** (via apt-get)
- **cuda** versão 10.0.166
- **cudnn** versão 7.3.1
- **opencv** versão 3.3.1
- **nvcaff** versão 0.16.6 (necessário criar link simbólico em /usr/local/include e /usr/local/lib para cabeçalhos e binários, respectivamente)

- **sphinxbase** versão customizada disponível em <https://github.com/antioniodourado/sphinxbase>
- **pocketsphinx** versão customizada disponível em <https://github.com/antioniodourado/pocketsphinx>
- **libusb-1.0.0-dev** (via apt-get)
- **libopenal-dev** (via apt-get)
- **libespeak-dev** (via apt-get)
- **libconfig++-dev** (via apt-get)
- **libalut-dev** (via apt-get)
- **libboost-dev** (via apt-get)
- **libgoogle-glog-dev** (via apt-get)
- **numlockx** (via apt-get)
- **libncurses5-dev** (via apt-get)
- **libinput-dev** (via apt-get)
- **libudev-dev** (via apt-get)
- **libpulse-dev** (via apt-get)
- **oss-compat** (via apt-get)
- **alsa-oss** (via apt-get)
- **libasound2-dev** (via apt-get)
- **librealsense2** tutorial disponível em <https://www.jetsonhacks.com/2019/05/16/jetson-nano-realsense-depth-camera>
- **sl\_zed** disponível em [https://www.stereolabs.com/developers/release/#sdkdownloads\\_anchor](https://www.stereolabs.com/developers/release/#sdkdownloads_anchor)

## E.2 Código-fonte e compilação

O código-fonte do sistema de navegação, disponibilizado sob a licença GPLv2, pode ser acessado e baixado do seguinte link:

<https://github.com/antoniourado/afsvip/>

No pacote, três diretórios são encontrados:

- **Assets** - Contém o arquivo de configuração, onde as customizações e outras configurações do sistema são salvos, o modelo treinado de rede neural convolucional e o modelo treinado pelo Sphinx, sendo:
  - DLModel - Diretório do modelo treinado de rede neural convolucional com os arquivos:
    - \* `deploy.prototxt` - Estrutura da rede neural;
    - \* `model.caffemodel` - Binário com os pesos treinados;
    - \* `labels.txt` - Lista de classes;
    - \* `mean.binaryproto` - Médias das imagens de treinamento.
  - SphinxModel - Diretório do modelo treinado do Sphinx para reconhecimento de fala, tendo como principais arquivos:
    - \* `en-us/afsvip.lm` - Modelo de idioma do inglês;
    - \* `en-us/afsvip.dic` - Dicionário de palavras.
- **build** - Contém o arquivo Makefile, com as diretrizes de compilação do sistema, e um link simbólico para o diretório Assets;
- **src** - Contém os arquivos de código-fonte com todo o desenvolvimento realizado para o sistema de navegação.

Para a compilação do sistema, é necessário verificar os diretórios das bibliotecas e cabeçalhos especificados em *build/Makefile* e garantir que eles contenham as localizações de todas as bibliotecas necessárias. Caso haja erro na compilação, pode ser necessário criar links simbólicos nos diretórios especificados pelo *Makefile* na variável `$INCPATH` e nos caminhos no final da variável `$LIBS`. O comando *make*, executado dentro do diretório *build/*, inicia a sequência de compilação do sistema. Caso nenhum erro ocorra, o executável **sysnavip** estará disponível para ser executado ao final do processo de compilação.

Futuramente, o sistema, suas bibliotecas e dependências serão empacotados dentro de um contêiner Docker, a fim de facilitar a sua implantação em diferentes plataformas e *hardwares*. A partir de uma imagem, será possível criar um contêiner autossuficiente sobre suas dependências, que poderá ser rapidamente executado.

## GLOSSÁRIO

---

---

- API** – *Application Programming Interfaces / Interface de Programação de Aplicativos*
- BoVW** – *Bag of Visual Words / Bolsa de Palavras Visuais*
- CCP** – *Coeficiente de Correlação de Pearson*
- CEP** – *Comitê de Ética em Pesquisa*
- CGP** – *Cartesian Genetic Programming / Programação Genética Cartesiana*
- ETA** – *Electronic Travel Aids / Auxílios Eletrônicos de Percurso*
- FLANN** – *Fast Library for Approximate Nearest Neighbors / Biblioteca Rápida para Aproximação de Vizinhos Próximos*
- FPGA** – *Field-Programmable Gate Array / Arranjo de Portas Programáveis em Campo*
- FSP** – *Frames Per Second / Quadros Por Segundo*
- GD** – *Generational Distance / Distância Geracional*
- GPS** – *Global Positioning System / Sistema Global de Posicionamento*
- GPU** – *Graphics Processing Units / Unidades de Processamento Gráfico*
- GP** – *Genetic Programming / Programação Genética*
- HOG** – *Histogram of Oriented Gradients / Histograma de Gradientes Orientados*
- HV** – *Hypervolume*
- IBGE** – *Instituto Brasileiro de Geografia e Estatística*
- IGD** – *Inverted Generational Distance / Distância Geracional Invertida*
- ILSVRC** – *ImageNet Large Scale Visual Recognition Competition / Competição de Reconhecimento Visual em Larga Escala do ImageNet*
- IMU** – *Inertial Measurement Unit / Unidade de Medida Inercial*

- KNN** – *K-Nearest Neighbors / K Vizinhos Mais Próximos*
- LSTM** – *Long Short-Term Memory / Memória de Curto Prazo Longa*
- MM** – *Morfologia Matemática*
- MOO** – *Multi-Objective Optimization / Otimização Multiobjetivo*
- NSGA-II** – *Nondominated Sorting Genetic Algorithm / Algoritmo Genético de Ordenamento Não dominado*
- OMS** – *Organização Mundial da Saúde*
- ONG** – *Organização Não Governamental*
- PDV** – *Pessoa com Deficiência Visual*
- RFID** – *Radio-Frequency Identifier / Identificador de Rádio-Frequência*
- RGB-D** – *Red, Green, Blue and Depth / Vermelho, Verde, Azul e Profundidade*
- RMF** – *Recursive Median Filter / Filtro Mediano Recursivo*
- RMSE** – *Root Mean Squared Error / Raíz de Média Quadrada*
- RNC** – *Rede Neural Convolucional*
- SOC** – *System-on-Chip / Sistema no Chip*
- SSIM** – *Structural Similarity / Similaridade Estrutural*
- SURF** – *Speeded Up Robust Features / Características Robustas Aceleradas*
- SVM** – *Support Vector Machines / Máquinas de Vetores de Suporte*
- VLAD** – *Locally Agregated Descriptor / Descritor Agregado Localmente*