

Universidade de Brasília - UnB  
Faculdade UnB Gama - FGA  
Engenharia Eletrônica

**Sistema de Navegação de Cadeiras de Rodas  
através de Reconhecimento de Pisos Táteis por  
Visão Computacional**

Autor: Débora Fernandes Gonçalves  
Orientador: Prof. Dr. Daniel Maurício Muñoz Arboleda

Brasília, DF  
2023





Débora Fernandes Gonçalves

# **Sistema de Navegação de Cadeiras de Rodas através de Reconhecimento de Pisos Táteis por Visão Computacional**

Monografia submetida ao curso de graduação em Engenharia Eletrônica da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Dr. Daniel Maurício Muñoz Arboleda

Brasília, DF

2023

---

Débora Fernandes Gonçalves

Sistema de Navegação de Cadeiras de Rodas através de Reconhecimento de Pisos Táteis por Visão Computacional/ Débora Fernandes Gonçalves. – Brasília, DF, 2023-

72 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Daniel Maurício Muñoz Arboleda

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB  
Faculdade UnB Gama - FGA , 2023.

1. Piso Tátil. 2. Visão Computacional. I. Prof. Dr. Daniel Maurício Muñoz Arboleda. II. Universidade de Brasília. III. Faculdade UnB Gama. Sistema de Navegação de Cadeiras de Rodas através de Reconhecimento de Pisos Táteis por Visão Computacional

CDU 02:141:005.6

---

Débora Fernandes Gonçalves

# **Sistema de Navegação de Cadeiras de Rodas através de Reconhecimento de Pisos Táteis por Visão Computacional**

Monografia submetida ao curso de graduação em Engenharia Eletrônica da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica.

Trabalho aprovado. Brasília, DF, 13/12/2023:

---

**Prof. Dr. Daniel Maurício Muñoz  
Arboleda**  
Orientador

---

**Prof. Dr. Diogo Caetano Garcia  
(FGA/UnB)**  
Convidado 1

---

**Prof. Dr. Gerardo Antonio Idrobo  
Pizo (FGA/UnB)**  
Convidado 2

Brasília, DF  
2023



# Agradecimentos

Ao concluir esta significativa etapa da minha trajetória acadêmica e profissional, percebo que cada obstáculo superado contou com o apoio de diversos agentes em todas as esferas da minha vida.

Gostaria de expressar minha eterna gratidão a Deus por me conceder vida, saúde e força, capacitando-me a superar os desafios ao longo de toda a minha jornada acadêmica.

Ao meu orientador, Daniel Muñoz, a quem admiro profundamente como profissional desde o primeiro contato em sala de aulas, dedico meu agradecimento pela orientação, incentivo, paciência e ensinamentos transmitidos ao longo desses anos.

Agradeço de forma especial a minha irmã Laura, por seu constante incentivo, encorajamento e apoio inestimável, especialmente nesta fase final da graduação. Obrigada por ser minha companheira nas jornadas de testes; sem você, nada seria possível.

À minha mãe Neuza, agradeço pelo suporte financeiro e emocional constante, além de seu papel fundamental em impulsionar meus estudos e me encaminhar para o melhor caminho.

Ao meu pai Dogival, meu agradecimento pelo apoio, compreensão das minhas ausências e por fazer parte da base sólida da minha vida.

Aos meus familiares, pelo apoio, compreensão das minhas ausências e por serem acolhedores na minha presença.

Aos amigos de graduação, meu reconhecimento pela ajuda, escuta e incentivo ao longo de toda a trajetória acadêmica.

Estendo minha gratidão a todos que, direta ou indiretamente, contribuíram para a minha formação.



# Resumo

Este estudo propõe um sistema de navegação para cadeiras de rodas, utilizando visão computacional para reconhecimento de pisos táteis, com foco em uma solução de baixo custo. O sistema visa identificar e distinguir tipos de pisos táteis, como alerta e direcional, e retornar o resultado para o usuário tomar decisões de movimento da cadeira de rodas. Durante o desenvolvimento, enfrentou-se o desafio de equilibrar desempenho e tempo de processamento nos algoritmos de processamento de imagem, cruciais para a viabilidade da solução embarcada. Foram empregadas técnicas como conversão de escala de cinzas, equalização do histograma, suavização da imagem, segmentação e extração de atributos com base nos pixels de interesse na imagem binarizada. Os resultados demonstram a eficácia na identificação e distinção de tipos de pisos táteis.

O trabalho destaca aplicações promissoras da tecnologia assistiva e visão computacional, proporcionando autonomia de forma acessível para pessoas com deficiência motora. Além disso, sugere possibilidades de aprimoramento e aponta caminhos para futuras pesquisas na área.

**Palavras-chaves:** Cadeira de Rodas. Piso Tátil. Processamento de Imagem. Visão Computacional. Sistema Embarcado.



# Abstract

*This study proposes a navigation system for wheelchairs, using computer vision to recognize tactile floors, focusing on a low-cost solution. The system aims to identify and distinguish types of tactile floors, such as alert and directional, and return the result for the user to make wheelchair movement decisions.*

*During development, the challenge of balancing performance and processing time in the image processing algorithms, crucial to the solution of the embedded solution, was planned. Techniques such as grayscale conversion, histogram equalization, image smoothing, segmentation and attribute removal based on the pixels of interest in the binarized image were employed. The results demonstrate the effectiveness in identifying and distinguishing types of tactile floors.*

*The work highlights promising applications of assistive technology and computer vision, convenience and autonomy in an accessible way for people with motor disabilities. Furthermore, it suggests possibilities for improvement and points out paths for future research in the area.*

**Key-words:** *Wheelchair. Tactile Floor. Image Processing. Vision Computational. Embedded System.*



# Lista de ilustrações

Figura 1 – Cadeira de Rodas Ortobras. . . . .	20
Figura 2 – Piso tátil de alerta e direcional, respectivamente (OLIVEIRA; SOLEDADE, 2016). . . . .	24
Figura 3 – Modelo de corel RGB [Fonte:(GONZALEZ; WOODS, 2010)]. . . . .	25
Figura 4 – Disco Cromático HSV [Fonte: (SIMÕES, 2012)]. . . . .	27
Figura 5 – Conversão entre imagem monocromática e RGB. [Fonte: (MARTINS, 2019)]. . . . .	27
Figura 6 – Algoritmos de Segmentação. . . . .	33
Figura 7 – Máscara geral de filtro 3 x 3. . . . .	34
Figura 8 – Máscaras 3 x 3. . . . .	35
Figura 9 – Detecção de ponto isolado. (a) Imagem original da radiografia da uma lâmina de turbina com porosidade de um único pixel preto. (b) Imagem filtrada, resultante da convolução da máscara com a imagem. (c) Imagem após aplicação do limiar, resultado da utilização da Equação 2.11. [Fonte: (GONZALEZ; WOODS, 2010)] . . . . .	36
Figura 10 – Detecção de linhas. (a) Imagem original. (b) Imagem resultante do filtro laplaciano, e parte ampliada desta imagem. (c) Valor absoluto do filtro laplaciano. (d) Valores positivos do filtro laplaciano. [Fonte: (GONZALEZ; WOODS, 2010)] . . . . .	36
Figura 11 – Máscaras de detecção de linhas. Adaptado de (GONZALEZ; WOODS, 2010). . . . .	37
Figura 12 – Modelos de borda e seus perfis de intensidade. [Fonte: (GONZALEZ; WOODS, 2010)] . . . . .	38
Figura 13 – Máscaras unidimensionais. Adaptado de (GONZALEZ; WOODS, 2010). . . . .	39
Figura 14 – Máscaras de filtros. Adaptado de (GONZALEZ; WOODS, 2010). . . . .	39
Figura 15 – Histogramas de intensidade que podem ser divididos (a) por um limiar único e (b) por limiares duplos. [Fonte: (GONZALEZ; WOODS, 2010)] . . . . .	41
Figura 16 – Exemplo de Limiarização Global Simples.(a) Impressão digital ruidosa. (b) Histograma. (c) Segmentação resultante usando um limiar global. [Fonte: (GONZALEZ; WOODS, 2010)] . . . . .	42
Figura 17 – Exemplo de Limiarização global ótima com o método de Otsu. (a) Imagem original. (b) Histograma.(c) Resultado da segmentação utilizando o algoritmo global básico. (d) Resultado obtido pelo método de Otsu. [Fonte: (GONZALEZ; WOODS, 2010)] . . . . .	44

Figura 18 – Exemplo de segmentação utilizando crescimento de regiões.(a) Imagem de raios X de uma solda defeituosa. (b) Histograma. (c) Imagem inicial com as sementes. (d) Imagem final com as sementes. (e) Valor absoluto da diferença entre (a) e (c). (f) Histograma de (e). (g) Imagem da diferença após uma limiarização utilizando limiares duplos. (h) Imagem da diferença limiarizada com o menor dos limiares duplos. (i) Resultado da segmentação obtido pela técnica de crescimento de região. [Fonte: (GONZALEZ; WOODS, 2010)] . . . . .	46
Figura 19 – Exemplo de segmentação por divisão e fusão de regiões. (a) Imagem da supernova Cygnus Loop, tomada na banda de raios X pelo telescópio Hubble da Nasa. (b) a (d) Resultados de limitar a menor quadrirregião permitida aos tamanhos de $32 \times 32$ , $16 \times 16$ , e $8 \times 8$ pixels, respectivamente.[Fonte: (GONZALEZ; WOODS, 2010)] . . . . .	47
Figura 20 – Imagem representada graficamente como uma superfície. [Fonte: (GONZALEZ; WOODS, 2010)] . . . . .	47
Figura 21 – Exemplo de segmentação usando watersheds morfológicas.(a) Imagem original. (b) Vista topográfica. (c) e (d) Duas fases da inundação. (e) Resultado de novas inundações. (f) Começo da fusão da água de duas watersheds. (g) Barragens maiores. (h) Linhas finais da watershed.[Fonte: (GONZALEZ; WOODS, 2010)] . . . . .	48
Figura 22 – Raspberry Pi 3 Model B+ [Fonte: (LTD, a)]. . . . .	53
Figura 23 – Mudança de direção por pisos táteis [Fonte: (RACHID, 2013)]. . . . .	55
Figura 24 – Etapas do processamento de imagem. . . . .	56
Figura 25 – Raspberry Pi Câmera Rev 1.3 [Fonte: (LTD, b)]. . . . .	56
Figura 26 – Visão lateral da cadeira de rodas [Fonte: (BRANCO, 2018)]. . . . .	57
Figura 27 – Câmera acoplada à cadeira de rodas. . . . .	58
Figura 28 – Pisos táteis amarelos e vermelhos do prédio UAC do Campus Gama da Universidade de Brasília. . . . .	58
Figura 29 – Pisos táteis pretos do prédio UED do Campus Gama da Universidade de Brasília. . . . .	59
Figura 30 – Interface gráfica composta por uma barra deslizante para ajuste do tamanho do <i>kernel</i> . . . . .	60
Figura 31 – Interface gráfica composta por uma barra deslizante para ajuste do valor limiar ( <i>thresh</i> ). . . . .	61
Figura 32 – Equalização do Histograma. . . . .	63
Figura 33 – Suavização da Imagem com aplicação do filtro média, filtro gaussiano e filtro mediana, respectivamente, utilizando um kernel de $5 \times 5$ . . . . .	64
Figura 34 – Suavização da Imagem com aplicação do filtro média, filtro gaussiano e filtro mediana, respectivamente. . . . .	65

Figura 35 – Imagem original acima e imagem binária resultante abaixo. . . . .	66
Figura 36 – Teste com captura de vídeo assíncrona em pisos táteis amarelos e vermelhos. . . . .	67
Figura 37 – Resultado em tempo real em pisos táteis pretos. . . . .	67
Figura 38 – Tempo médio de processamento de um frame. . . . .	68
Figura 39 – Representação do tipos de piso por LEDs. À esquerda, tem-se a representação de um piso direcional, indicado pelo LED verde, enquanto à direita, tem-se a representação de um piso de alerta, identificado pelo LED vermelho. . . . .	68



# Lista de tabelas

Tabela 1 – Características da Raspberry Pi Câmera Rev 1.3 . . . . .	57
Tabela 2 – Custos do protótipo. . . . .	62
Tabela 3 – Demanda computacional para suavização da imagem com kernel 5x5. .	64
Tabela 4 – Demanda computacional para suavização da imagem com kernel 15x15.	65



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>19</b>
<b>1.1</b>	<b>Descrição do Problema</b>	<b>20</b>
<b>1.2</b>	<b>Objetivos</b>	<b>21</b>
1.2.1	Objetivo Geral	21
1.2.2	Objetivos Específicos	21
<b>1.3</b>	<b>Justificativa</b>	<b>21</b>
<b>1.4</b>	<b>Aspectos Metodológicos</b>	<b>22</b>
<b>1.5</b>	<b>Organização do Documento</b>	<b>22</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>23</b>
<b>2.1</b>	<b>Pisos Táteis</b>	<b>23</b>
<b>2.2</b>	<b>Processamento de Imagens Digitais</b>	<b>24</b>
2.2.1	Modelo de Cores	25
2.2.1.1	Modelo RGB	25
2.2.1.2	Modelo HSV	26
2.2.1.3	Escala de Cinza	27
2.2.2	Equalização do histograma	28
2.2.3	Suavização da Imagem	29
2.2.3.1	Filtro de média aritmética	30
2.2.3.2	Filtro gaussiano	30
2.2.3.3	Filtro de mediana	31
2.2.4	Segmentação da Imagem	32
2.2.4.1	Detecção de ponto, linha e borda	33
2.2.4.2	Detecção de pontos isolados	35
2.2.4.3	Detecção de linhas	36
2.2.4.4	Detecção de bordas	37
2.2.4.5	Limiarização	40
2.2.4.6	Limiarização global simples	42
2.2.4.7	Limiarização global ótima com o método de Otsu	42
2.2.4.8	Limiarização variável	43
2.2.4.9	Segmentação baseada na região	44
2.2.4.10	Segmentação por crescimento de região	45
2.2.4.11	Segmentação por divisão e fusão de região	46
2.2.4.12	Segmentação usando <i>watersheds</i> morfológicas	47
2.2.5	Extração de atributos	48

<b>2.3</b>	<b>Visão computacional</b> . . . . .	<b>49</b>
2.3.1	OpenCV . . . . .	50
<b>2.4</b>	<b>Estado da Arte</b> . . . . .	<b>50</b>
<b>3</b>	<b>IMPLEMENTAÇÃO</b> . . . . .	<b>53</b>
<b>3.1</b>	<b>Plataforma computacional</b> . . . . .	<b>53</b>
<b>3.2</b>	<b>Algoritmo</b> . . . . .	<b>54</b>
<b>3.3</b>	<b>Aquisição da Imagem</b> . . . . .	<b>55</b>
<b>3.4</b>	<b>Pré-processamento da Imagem</b> . . . . .	<b>57</b>
<b>3.5</b>	<b>Segmentação da Imagem</b> . . . . .	<b>60</b>
<b>3.6</b>	<b>Extração dos atributos</b> . . . . .	<b>61</b>
<b>3.7</b>	<b>Metodologia dos testes</b> . . . . .	<b>62</b>
<b>3.8</b>	<b>Custo do projeto</b> . . . . .	<b>62</b>
<b>4</b>	<b>RESULTADOS</b> . . . . .	<b>63</b>
<b>4.1</b>	<b>Testes iniciais em capturas de imagens</b> . . . . .	<b>63</b>
4.1.1	Histograma . . . . .	63
4.1.2	Suavização da Imagem . . . . .	64
<b>4.2</b>	<b>Testes em capturas de vídeo assíncrona</b> . . . . .	<b>66</b>
<b>4.3</b>	<b>Testes em tempo real</b> . . . . .	<b>67</b>
<b>4.4</b>	<b>Representação Visual</b> . . . . .	<b>68</b>
<b>5</b>	<b>CONSIDERAÇÕES FINAIS</b> . . . . .	<b>69</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>71</b>

# 1 INTRODUÇÃO

A deficiência motora ocorre quando partes do corpo sofrem mudanças parciais ou completas que afetam o funcionamento físico-motor e o desempenho das funções de uma pessoa. Um dos tipos mais conhecidos de deficiência motora é a paraplegia, que resulta na perda dos movimentos dos membros inferiores. Os paraplégicos não possuem comprometimento dos membros superiores e conseguem uma independência satisfatória com modelos de cadeiras de rodas manuais. Por outro lado, a tetraplegia afeta tanto os membros superiores quanto os inferiores, devido a uma lesão medular alta. Uma pessoa com tetraplegia possui sua independência física afetada, necessitando de ajuda para realizar atividades básicas, logo, o uso de cadeiras de rodas automatizadas e dispositivos de apoio é essencial para a qualidade de vida deste grupo.

De acordo com a Organização Mundial de Saúde, mais de um bilhão de pessoas em todo o mundo convivem com alguma deficiência, algo próximo de 15% da população mundial ([World Health Organization \(WHO\), The World Bank, 2012](#)).

No Brasil, a população com deficiência foi estimada em 18,6 milhões de pessoas com 2 anos ou mais, o que representa 8,9% da população nessa faixa etária. Esses dados são provenientes da Pesquisa Nacional por Amostra de Domicílios (PNAD): Pessoas com Deficiência 2022, resultado de uma parceria entre a Secretaria Nacional dos Direitos da Pessoa com Deficiência do Ministério dos Direitos Humanos e da Cidadania (SNDPD/MDHC) e o Instituto Brasileiro de Geografia e Estatística (IBGE). Os resultados da PNAD também indicam que as pessoas com deficiência enfrentam maior dificuldade de inserção no mercado de trabalho e nas instituições de ensino, o que impacta negativamente no acesso a renda ([Instituto Brasileiro de Geografia e Estatística \(IBGE\), 2022](#)).

A Pesquisa Nacional de Saúde (PNS), realizada em 2019 pelo Ministério da Saúde em parceria com o IBGE, amostra as condições de saúde da população brasileira. A pesquisa detalha que 7,783 milhões, ou 3,8 % da população acima de dois anos, apresentam deficiência motora nos membros inferiores, enquanto 5,457 milhões (2,7%) dos brasileiros possuem deficiência nos membros superiores. Neste estudo foi constatado que a deficiência afeta desproporcionalmente as populações vulneráveis, existindo uma prevalência maior de deficientes em locais de baixa renda ([Instituto Brasileiro de Geografia e Estatística \(IBGE\), 2019](#)).

Diante deste cenário, uma nova área do conhecimento, denominada Tecnologia Assistiva (TA), se consolidou com o objetivo de desenvolver soluções inovadoras e adaptativas que auxiliem em desafios enfrentados por pessoas com deficiência, buscando integrá-las de maneira mais completa na sociedade, melhorando sua qualidade de vida ([Subsecre-](#)

taria Nacional de Promoção dos Direitos da Pessoa com Deficiência, Comitê de Ajudas Técnicas, 2009).

Para garantir a autonomia e reinserção social de pessoas com deficiências motoras em situação de vulnerabilidade econômica, torna-se necessário o estudo acerca de tecnologias assistivas de baixo custo.

Nesse contexto, uma equipe composta por alunos e professores da Universidade de Brasília desenvolveu projetos de adaptação de uma cadeira de rodas comum em uma cadeira de rodas elétrica, especialmente projetada para atender usuários tetraplégicos. Este trabalho se propõe a utilizar essa cadeira como base para a implementação de um sistema de navegação por meio do reconhecimento de pisos táteis, utilizando visão computacional. O modelo utilizado é uma cadeira de baixo custo e fácil aquisição da Ortobras, conforme ilustrado na Figura 1.



Figura 1 – Cadeira de Rodas Ortobras.

## 1.1 Descrição do Problema

Os tetraplégicos não conseguem conduzir uma cadeira de rodas manual por falta de controle dos membros superiores. O ideal para a autonomia e independência dessas pessoas é uma cadeira de rodas motorizada autônoma ou semi-autônoma com controle eletrônico, seja por *joystick*, *eye tracking*, controle de voz, entre outros. Entretanto, os produtos comerciais disponíveis não são adequados para atender às diversas necessidades e demandas específicas. Em geral, as cadeiras de rodas autônomas não são encontradas facilmente no mercado brasileiro, e as encontradas são inviáveis financeiramente para a maioria da população nessas condições de saúde.

O acionamento por *joystick* ainda é uma barreira para os tetraplégicos. Neste trabalho pensou-se então em aproveitar a sinalização visual e tangível do piso tátil, obrigatório em espaços de uso público, para indicar a direção que a cadeira de rodas deve se deslocar através de algoritmos de reconhecimento visual. Os pisos táteis são guias de

borracha antiderrapante com texturas e cores diferentes do piso, que servem para orientar pessoas com os diversos graus de deficiência visual.

Em termos de *hardware*, o principal desafio é embarcar um algoritmo de visão computacional capaz de reconhecer os tipos de pisos táteis, visando o melhor desempenho computacional no menor tempo de processamento. Em termos de *software*, a dificuldade é validar diferentes tipos de algoritmos que sejam eficientes para a aplicação.

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

Projetar um sistema de reconhecimento de pisos táteis, visando uma solução de baixo custo computacional, que possa ser aplicado à navegação de cadeira de rodas. O sistema deve reconhecer e detectar as mudanças dos tipos de pisos táteis e retornar o resultado para o usuário tomar decisões de movimento da cadeira de rodas.

### 1.2.2 Objetivos Específicos

- Projetar um modelo computacional capaz de detectar e reconhecer os pisos táteis em tempo real através da visão computacional.
- Embarcar o algoritmo de processamento em um microcontrolador ou microprocessador de baixo custo.
- Representar a saída do tipo de piso tátil de forma que um sistema de navegação possa usá-lo para localização e definição de rotas.
- Efetuar testes de bancada que permitam medir o desempenho do sistema e sua viabilidade.

## 1.3 Justificativa

Os pisos táteis são essenciais para garantir a segurança, orientação e mobilidade de deficientes visuais. Pensando nisso, em conjunto com a lacuna de acessibilidade para os tetraplégicos, teve-se a ideia de aproveitar os guias táteis como orientação de percursos para cadeiras de rodas em locais públicos. O objetivo é reconhecer os pisos táteis e os diferenciarem, do tipo direcional ou alerta, para a correta tomada de decisão do movimento da cadeira de rodas.

## 1.4 Aspectos Metodológicos

Conforme será descrito no Capítulo 3, este trabalho será classificado quanto aos seus objetivos e aos procedimentos técnicos nele empregados. A pesquisa se desenvolve em etapas, desde a formulação do problema até a apresentação e discussão dos resultados.

A pesquisa é de natureza aplicada, ou seja, há o conhecimento e aplicação prática para a solução do problema proposto, com objetivo descritivo, que visa estabelecer relação entre as variáveis de estudo e utiliza um procedimento técnico de simulação.

As etapas são divididas em:

I) Definição do problema de estudo, levantamento de requisitos e definição dos objetivos;

II) Estudo do estado da arte e revisão bibliográfica acerca de cadeiras de rodas motorizadas, pisos táteis e algoritmos de visão computacional;

III) Implementação de um algoritmo de visão computacional para detecção e reconhecimento de pisos táteis;

IV) Simulação do algoritmo para testes;

V) Validação da consistência dos dados obtidos nas simulações;

VI) Apresentar conclusões e etapas futuras.

## 1.5 Organização do Documento

A estrutura documental do trabalho é dividida em capítulos.

O Capítulo 2 apresenta uma fundamentação teórica e um levantamento do estado da arte de cadeiras de rodas motorizadas e algoritmos de visão computacional para reconhecimento de pisos táteis, além de outros aspectos importantes para a aquisição e processamento de dados futuros.

O Capítulo 3 exhibe a proposta metodológica do projeto, como o algoritmo a ser utilizado e embarcado, os componentes a serem utilizados, bem como todos os aspectos de hardware e software que a implementação demanda.

O quarto Capítulo discorre sobre os resultados obtidos nos testes de bancada.

Por fim, o último Capítulo traz as considerações finais do projeto, bem como dificuldades e soluções encontradas ao decorrer do trabalho, apontando sugestões de melhorias para uma implementação futura.

## 2 Fundamentação Teórica

Neste capítulo, serão abordados os algoritmos de visão computacional para reconhecimento de pisos táteis, plataformas computacionais e um levantamento do estado da arte, como os trabalhos já realizados sobre cadeiras de rodas motorizadas e reconhecimento de guias táteis.

### 2.1 Pisos Táteis

Os pisos táteis são elementos de acessibilidade utilizados para auxiliar pessoas com deficiência visual ou com baixa visão a se deslocarem com segurança e autonomia em espaços públicos e/ou privados. Eles possuem uma superfície com textura diferenciada e cor contrastante em relação ao piso. A instalação é feita em locais estratégicos, como calçadas, plataformas de embarque e escadas, e para indicar rotas, direções, obstáculos e informações de segurança.

As normas que regulamentam os pisos táteis variam de acordo com a localidade. No Brasil, as normas técnicas que orientam a instalação e a padronização dos pisos táteis são definidas pela Associação Brasileira de Normas Técnicas (ABNT). Por exemplo, a norma ABNT NBR 9050:2015 ([ABNT, 2015](#)) estabelece as diretrizes para a acessibilidade em edificações, mobiliário, espaços e equipamentos urbanos. Além disso, o Conselho Nacional dos Direitos da Pessoa com Deficiência (CONADE) também desenvolve normas e diretrizes para a acessibilidade em diversos ambientes.

As normas dos pisos táteis definem suas características técnicas, tais como as dimensões, as cores, as texturas, a altura e a distância entre as saliências, em busca de garantir a eficácia e padronização dos pisos. Por outro lado, orientam a instalação adequada, considerando aspectos como localização, altura, inclinação, sinalização e iluminação, a fim de evitar acidentes. Existem diferentes tipos de pisos táteis para finalidades específicas, variando em sua forma, cor, textura e material utilizado. Os principais tipos de pisos táteis são ([ABNT, 2016](#)):

- Piso tátil de alerta: É um tipo de piso que possui uma textura de bolinhas em relevo e é utilizado para indicar a proximidade de um obstáculo ou perigo, como escadas, elevadores ou rampas.
- Piso tátil direcional: É um tipo de piso que possui uma textura em linhas retas ou onduladas em relevo, e é utilizado para indicar o sentido de direção ou de trajeto, como em corredores.

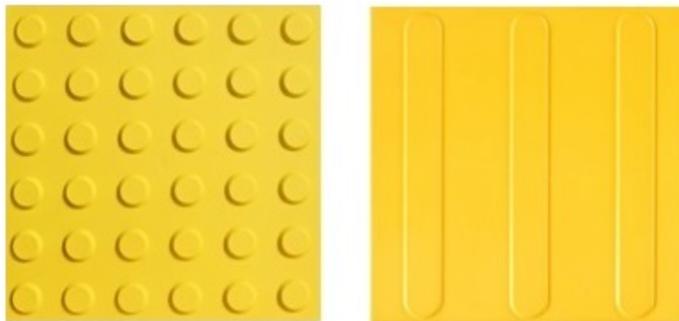


Figura 2 – Piso tátil de alerta e direcional, respectivamente (OLIVEIRA; SOLEDADE, 2016).

Os materiais utilizados na fabricação dos pisos táteis também podem variar. Alguns exemplos de materiais são o concreto, a borracha e o plástico. Vale ressaltar que a instalação correta e a manutenção dos pisos táteis são fundamentais para garantir a eficácia e a segurança desses elementos de acessibilidade, promovendo a inclusão e a igualdade de acesso a todos os indivíduos.

## 2.2 Processamento de Imagens Digitais

A representação de imagens digitais envolve a descrição da informação visual contida em uma imagem de maneira que possa ser processada por um computador. Uma imagem digital é comumente expressa como uma matriz bidimensional de pixels, sendo estes os elementos fundamentais que compõem a estrutura visual da imagem. Cada pixel, como a menor unidade visual, é identificado por suas coordenadas  $(x, y)$  na matriz e possui um valor que indica sua intensidade ou cor, dependendo se a imagem é em escala de cinza ou colorida (GONZALEZ; WOODS, 2010).

O processamento de imagem refere-se ao conjunto de técnicas e algoritmos utilizados para manipular e extrair informações de imagens digitais. O pré-processamento de imagem é uma etapa inicial que envolve a aplicação de técnicas como remoção de ruídos, ajuste de contraste, correção de distorções e realces de características importantes da imagem, com o intuito de melhorar sua qualidade e torná-la mais adequada para as etapas subsequentes de processamento. O processamento de imagem, propriamente dito, refere-se à aplicação de técnicas para extrair informações, realizar análises ou obter resultados desejados a partir da imagem pré-processada (GONZALEZ; WOODS, 2010).

Há muitas técnicas de processamento de imagem, cada uma com características e aplicações específicas. A escolha da técnica mais adequada para a aplicação depende de fatores como os requisitos da tarefa, as características da imagem, as restrições computacionais e conhecimento do domínio. Nas seções a abaixo serão descritas as principais etapas do processamento de imagem e suas respectivas técnicas e características.

## 2.2.1 Modelo de Cores

Um modelo de cores é uma especificação de um sistema de coordenadas e um subespaço dentro desse sistema no qual cada cor é representada por um único ponto. A escolha do modelo de cores depende das necessidades específicas da aplicação e das características das cores que estão sendo manipuladas. Cada modelo oferece uma abordagem única para representar e manipular as cores com base em diferentes propriedades, como luz, pigmentação e percepção humana (GONZALEZ; WOODS, 2010).

### 2.2.1.1 Modelo RGB

O modelo RGB utiliza componentes espectrais primários, sendo vermelho (R, de *red*), verde (G, de *green*) e azul (B, de *blue*) para representar imagens coloridas, em virtude das características de absorção do olho humano.

O espaço de cores de interesse é representado por um cubo, conforme mostrado na Figura 3. Nesse cubo, os valores RGB primários estão localizados nos vértices correspondentes a vermelho, verde e azul. As cores secundárias ciano, magenta e amarelo ocupam outros três vértices. O preto está na origem do cubo, enquanto o branco está no vértice mais distante da origem. No contexto desse modelo, a escala de cinza é representada pelos pontos com valores RGB iguais e se estende do preto até o branco ao longo de um segmento de reta que conecta esses dois pontos. Cada cor no modelo é representada por pontos no interior ou na superfície do cubo e é definida por vetores que se originam na origem e se estendem até o ponto que representa a cor (GONZALEZ; WOODS, 2010).

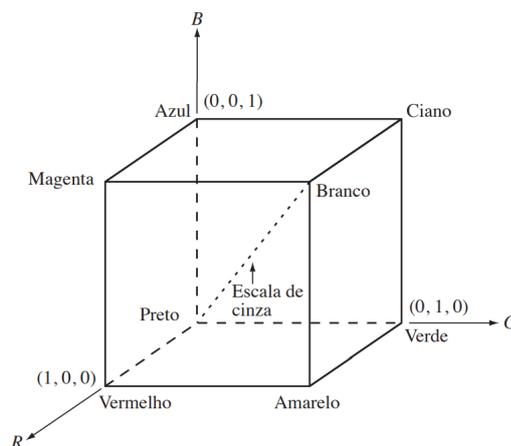


Figura 3 – Modelo de cores RGB [Fonte:(GONZALEZ; WOODS, 2010)].

Cada pixel de uma imagem RGB possui três componentes, que podem ser organizados na forma de um vetor coluna

$$z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

Onde  $z_1$   $z_2$   $z_3$  representam a intensidade de pixels na imagem vermelha, verde e azul, respectivamente.

### 2.2.1.2 Modelo HSV

O Modelo de Cores HSV (*Hue, Saturation e Value*), que representa Matiz, Saturação e Valor, é um modelo de representação de cores que visa descrever de forma mais intuitiva e perceptual a informação cromática. Diferentemente do Modelo RGB (*Red, Green, Blue*), que é baseado nas intensidades de luz vermelha, verde e azul, o modelo HSV é mais próximo da percepção humana de cores.

Os componentes principais do modelo HSV são:

- **Matiz (*Hue*):** Matiz é um atributo associado ao comprimento de onda dominante em uma mistura de ondas de luz. Assim, o matiz representa a cor dominante percebida por um observador, geralmente representado como um ângulo em um círculo de cores, variando de 0 a 360 graus.
- **Saturação (*Saturation*):** Indica a pureza da cor. Uma saturação mais alta significa cores mais vibrantes, enquanto uma saturação mais baixa leva a cores mais desbotadas ou tons de cinza. A saturação geralmente é representada por números inteiros em uma escala de 0 a 255.
- **Valor (*Value*):** Refere-se à luminância ou brilho da cor. Um valor mais alto resulta em cores mais claras, enquanto um valor mais baixo leva a cores mais escuras, também costuma ser representado por números inteiros em uma escala de 0 a 255 (GONZALEZ; WOODS, 2010).

Este modelo de cores pode ser representado por um cilindro (Figura 4), onde o *Hue, Saturation e Value* são representados pela dimensão angular, radial e vertical, respectivamente.

O Modelo de Cores HSV é frequentemente utilizado em aplicações gráficas, processamento de imagens e design, pois proporciona uma representação mais intuitiva e separada das características perceptuais das cores. Ele facilita o ajuste de propriedades como brilho, saturação e tonalidade de forma mais intuitiva do que outros modelos de cores, especialmente para usuários não especializados em processamento de imagens.

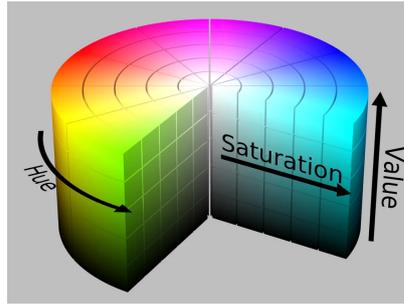


Figura 4 – Disco Cromático HSV [Fonte: (SIMÕES, 2012)].

### 2.2.1.3 Escala de Cinza

A conversão de uma imagem colorida para escala de cinza é uma etapa comumente realizada em processamento de imagens e visão computacional. Esta conversão reduz a complexidade da informação visual. Uma imagem colorida utiliza o modelo de cores RGB (*Red, Green e Blue*). Ao converter para escala de cinza, essas três componentes são combinadas em uma única intensidade de brilho, representada em um único canal. Isso simplifica e reduz a dimensionalidade da imagem e, conseqüentemente, a quantidade de informações a serem processadas. Além disso, a conversão para escala de cinza pode normalizar a influência da iluminação na imagem. Em imagens coloridas, a intensidade de cada canal de cor é afetada pela iluminação. Já na escala de cinza, não há essa influência, assim os algoritmos se concentram nas características estruturais da imagem, independentemente das condições de iluminação (GONZALEZ; WOODS, 2010).

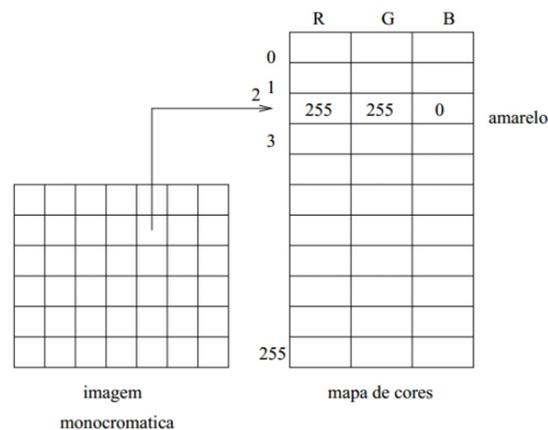


Figura 5 – Conversão entre imagem monocromática e RGB. [Fonte: (MARTINS, 2019)].

Em muitos casos, a informação de cor não é relevante para a aplicação. Por exemplo, em tarefas de detecção de bordas, reconhecimento de padrões ou segmentação, a informação de cor não contribui significativamente para a solução, como nos casos de reconhecimento de cores ou detecção de objetos com base em informações de cores especí-

ficas. Isso é benéfico quando se lida com grandes conjuntos de dados ou em sistemas com recursos computacionais limitados.

## 2.2.2 Equalização do histograma

Um histograma é uma representação gráfica da distribuição de intensidades de uma imagem. Ele mostra a frequência com que cada nível de intensidade ocorre na imagem, ou seja, quantos pixels possuem determinado valor de intensidade.

Formalmente, Gonzalez e Woods (2010) definem que o histograma de uma imagem digital, que possui níveis de intensidade no intervalo  $[0, L - 1]$ , é uma função discreta representada por  $h(r_k) = n_k$ , onde  $r_k$  é o  $k$ -ésimo valor de intensidade e  $n_k$  corresponde ao número de pixels na imagem que possuem a intensidade  $r_k$ .

A equalização do histograma é uma técnica de processamento de imagem que visa melhorar o contraste global da imagem. Ela redistribui os valores de intensidade da imagem de forma a tornar o histograma mais uniforme, maximizando a utilização da faixa dinâmica disponível.

O histograma de uma imagem digital pode ser visto como uma função de distribuição de probabilidade, que estima a probabilidade de ocorrência do nível de cinza na imagem. Cada elemento do conjunto é calculado por:

$$P(r_k) = \frac{n_k}{MN} \quad (2.1)$$

Onde  $r_k$  é o número de nível de cinza da imagem, podendo variar entre 0 (preto) e 255 (branco) em uma representação de 8 bits,  $n_k$  é o número de pixels na imagem com o nível de cinza,  $MN$  é número total de pixels na imagem e  $P(r_k)$  é a probabilidade de ocorrência do nível de cinza  $r_k$ .

Após a análise do histograma, pode-se equalizar a imagem, isto é, ajustar os níveis de cinza, para garantir um brilho e contraste balanceados. Para calcular os valores da função de transformação de equalização de histograma, é proposta a seguinte equação (GONZALEZ; WOODS, 2010):

$$\begin{aligned} s_k &= T(r_k) = (L - 1) \sum_{j=0}^k P(r_j) \\ &= \frac{(L - 1)}{MN} \sum_{j=0}^k n_j, \forall k = 1, 2, 3, \dots, L - 1; \end{aligned} \quad (2.2)$$

A aplicação da equalização do histograma envolve os seguintes passos:

1. Cálculo do histograma: O histograma da imagem original é calculado, mostrando a distribuição de intensidades.
2. Cálculo da função de distribuição acumulada: Uma função é criada a partir do histograma, representando a distribuição acumulada das intensidades.
3. Mapeamento dos valores de intensidade: Cada valor de intensidade da imagem original é mapeado para um novo valor, usando a função de distribuição acumulada. Isso redistribui os valores de intensidade de acordo com a distribuição acumulada.
4. Criação da imagem equalizada: Uma nova imagem é criada, onde os valores de intensidade foram substituídos pelos valores mapeados. Essa nova imagem possui um histograma mais uniforme e, conseqüentemente, um melhor contraste.

A equalização do histograma pode ser aplicada em várias situações, como na melhoria do contraste de imagens com distribuição desigual de intensidades, realce de detalhes em áreas escuras ou claras da imagem e pré-processamento para técnicas de segmentação e detecção de bordas.

### 2.2.3 Suavização da Imagem

Um filtro do tipo passa-baixas deixa passar as baixas freqüências e elimina os valores relacionados às altas freqüências. Portanto, o efeito deste filtro é o de suavização da imagem, uma vez que as altas freqüências que correspondem às transições abruptas são atenuadas.

No contexto de processamento de sinais, suavizar uma imagem refere-se a aplicar um filtro ou técnica para reduzir o ruído ou as variações abruptas de intensidade nos pixels da imagem. A suavização, também conhecida como filtragem ou *blur* em inglês, pode ser usada para melhorar a qualidade visual da imagem, reduzir imperfeições e preparar a imagem para outras etapas de processamento, como detecção de bordas ou reconhecimento de objetos (GONZALEZ; WOODS, 2010).

A suavização é frequentemente realizada por meio de filtros lineares, como o filtro de média aritmética e o filtro Gaussiano. Esses filtros calculam uma média ponderada das intensidades dos pixels vizinhos para determinar o valor do pixel suavizado. A suavização também pode ser feita por meio de filtros não lineares, como o filtro de mediana, que em vez de calcular uma média ponderada dos valores dos pixels na vizinhança, ele classifica esses valores e seleciona o valor mediano.

Ao aplicar um filtro de suavização em uma imagem, as variações abruptas de intensidade, como pequenos detalhes ou ruídos, são suavizadas, resultando em uma imagem mais uniforme. Vale ressaltar que, ao aplicar uma suavização, há perda de detalhes finos e

informações de alta frequência na imagem. Portanto, é necessário encontrar um equilíbrio entre a suavização desejada e a preservação dos detalhes importantes, dependendo da aplicação em questão.

### 2.2.3.1 Filtro de média aritmética

O filtro de média aritmética é uma técnica que calcula a média dos valores dos pixels em uma área específica da imagem, chamada de janela de subimagem ou vizinhança. Para a aplicação do filtro, primeiro é definido o conjunto de coordenadas  $S_{xy}$  de tamanho  $m \times n$  e centrado no ponto  $(x, y)$ , que representa os pixels dentro da vizinhança e então calcula-se a média aritmética dos valores desses pixels na imagem  $g(s, t)$  (GONZALEZ; WOODS, 2010):

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t) \quad (2.3)$$

Um filtro de média suaviza as variações locais em uma imagem, resultando na redução do ruído devido ao efeito de borramento. A técnica de suavização com filtro de média é geralmente a menos exigente em termos de desempenho e tempo computacional. Isso ocorre porque o filtro de média realiza uma operação de média simples dos pixels vizinhos, sem ponderação adicional. O tempo de processamento é proporcional ao tamanho  $m \times n$  da vizinhança, mas mesmo com tamanhos maiores, o filtro de média tende a ser rápido em comparação com outras técnicas mais complexas, como o filtro gaussiano (GONZALEZ; WOODS, 2010).

Se a prioridade da aplicação for a eficiência computacional e tempo de processamento, sem priorizar a preservação de detalhes finos e o ruído em questão não for de alta frequência, a aplicação do filtro de média para suavização de imagens pode ser uma boa escolha.

### 2.2.3.2 Filtro gaussiano

O filtro gaussiano aplica uma operação de convolução na imagem utilizando um *kernel* gaussiano, que é uma função de distribuição de probabilidade gaussiana.

A equação do filtro gaussiano para suavização de imagens é dada por:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \cdot \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (2.4)$$

Onde  $G(x, y)$  representa o valor do pixel na imagem suavizada,  $(x, y)$  são as coordenadas do pixel na imagem original e  $\sigma$  é o desvio padrão que controla a largura da distribuição gaussiana.

Ao aplicar o filtro gaussiano, cada pixel na imagem é substituído pelo valor médio ponderado dos pixels em sua vizinhança, sendo que os pesos são determinados pela distribuição gaussiana. Isso significa que os pixels mais próximos ao pixel central têm um peso maior na média ponderada, enquanto os pixels mais distantes têm um peso menor.

O filtro gaussiano é especialmente eficaz em remover ruídos de alta frequência, como o ruído aleatório e granular. Ele realiza uma suavização gradual, preservando as bordas e detalhes importantes da imagem. A largura do kernel gaussiano, controlada pelo parâmetro  $\sigma$ , determina a quantidade de suavização aplicada, quanto maior seu valor, maior será a área de influência do filtro e mais suave será a imagem resultante.

O filtro gaussiano faz uma suavização mais delicada, preservando mais os contornos que o filtro de média. Ainda assim, pode causar perda de detalhes e borramento em bordas mais fortes.

A complexidade computacional do filtro gaussiano está relacionada principalmente ao tamanho do kernel gaussiano utilizado, quanto maior o tamanho do kernel, mais operações de multiplicação e adição são necessárias. Além disso, o filtro gaussiano envolve a operação de convolução, o que requer cálculos mais complexos. A técnica é amplamente utilizada em aplicações que demandam suavizações delicadas e preservação de detalhes.

### 2.2.3.3 Filtro de mediana

A ideia central do filtro de mediana é substituir o valor de um pixel pela mediana dos níveis de intensidade dos pixels em sua vizinhança. Para cada pixel na imagem, uma janela retangular é definida em torno dele, e os valores dos pixels nessa janela são classificados em ordem crescente. O valor mediano, que é o valor do meio dessa sequência classificada, é então atribuído ao pixel central da vizinhança (GONZALEZ; WOODS, 2010).

Ao usar a mediana em vez da média para determinar o novo valor do pixel, o filtro de mediana consegue preservar melhor as bordas e detalhes da imagem. Isso ocorre porque a mediana é menos sensível a valores extremos do que a média, evitando assim a suavização excessiva e a perda de informações importantes. Ele oferece uma abordagem eficiente e robusta para reduzir o ruído, preservando ao mesmo tempo a nitidez e os detalhes da imagem original.

$$\hat{m}(x, y) = \text{mediana}_{(s,t) \in S_{xy}} g(s, t) \quad (2.5)$$

O filtro de mediana demanda maior desempenho computacional em comparação com os filtros anteriores devido à natureza do cálculo da mediana, que envolve a classificação dos valores dos pixels na vizinhança. Esta classificação envolve comparar os valores dos pixels repetidamente e rearranjá-los em ordem crescente ou decrescente. Esse processo

pode ser computacionalmente custoso, especialmente para janelas maiores ou imagens de alta resolução.

Vale ressaltar que a eficiência de cada filtro pode ser aprimorada com algoritmos otimizados e técnicas de processamento paralelo.

## 2.2.4 Segmentação da Imagem

A segmentação de imagens, o próximo passo após o pré-processamento de imagens, é um processo complexo que busca replicar no computador um processo cognitivo altamente sofisticado realizado através da visão humana (BEUCHER; MEYER, 1982). A segmentação consiste em subdividir uma imagem de entrada em suas partes ou objetos constituintes. Essas partes individuais são caracterizadas por serem uniformes e homogêneas em relação a propriedades específicas da imagem, como cor e textura.

No processo de segmentação, busca-se estabelecer uma distinção clara entre os pixels presentes na imagem, bem como diferenciá-los do fundo. Essa distinção é essencial para que o programa computacional possa interpretar os pixels adjacentes e agrupá-los em regiões (SANTOS, 2002).

Segundo Gonzalez e Woods (2010), se  $R$  representa toda a região espacial ocupada por uma imagem, a segmentação da imagem é um processo que particiona  $R$  em  $n$  sub-regiões,  $R_1, R_2, \dots, R_n$  de tal forma que:

- (a)  $\bigcup_{i=1}^n R_i = R$
- (b)  $R_i$  é um conjunto conectado,  $i=1, 2, \dots, n$ .
- (c)  $R_i \cap R_j = \emptyset$  para todo  $i$  e  $j$ ,  $i \neq j$
- (d)  $Q(R_i) = \text{VERDADEIRA}$  para  $i=1, 2, \dots, n$ .
- (e)  $Q(R_i \cup R_j) = \text{FALSA}$  para quaisquer regiões adjacentes  $R_i$  e  $R_j$ .

A propriedade  $Q$  refere-se a uma função ou critério que define algum aspecto desejado ou uma característica específica que as sub-regiões resultantes da segmentação devem possuir. Essa função é uma métrica ou medida que avalia a qualidade da segmentação em termos dos objetivos específicos da aplicação, podendo ser uma medida de homogeneidade, onde regiões semelhantes em termos de intensidade de cor ou textura são agrupadas. Ou pode ser uma medida relacionada à forma, ao tamanho ou a qualquer outra propriedade que seja relevante para a análise de imagem em questão (GONZALEZ; WOODS, 2010).

A condição (a) estabelece que a segmentação deve ser completa, o que significa que cada pixel da imagem deve estar em uma região. A condição (b) requer que os pontos dentro de uma região estejam conectados de acordo com uma forma predefinida. A condição (c) assegura que as regiões sejam separadas uma das outras. A condição (d) refere-se

às propriedades que os pixels dentro de uma região segmentada devem satisfazer, por exemplo,  $Q(R_i)=\text{VERDADEIRA}$  se todos os pixels em  $R_i$  possuírem o mesmo nível de intensidade. Por fim, a condição (e) indica que duas regiões adjacentes,  $R_i$  e  $R_j$ , devem ser distintas em termos da propriedade  $Q$  (GONZALEZ; WOODS, 2010).

Centenas de técnicas de segmentação estão presentes em literatura, mas não há um método único que seja considerado bom para todas as imagens, nem todos os métodos igualmente bons para um determinado tipo de imagem. No entanto, a maioria dos métodos de segmentação desenvolvidos para uma classe de imagens pode ser facilmente aplicada/estendida a outra classe de imagens (PAL; PAL, 1993).

Conforme abordado por Gonzalez e Woods (2010), a maioria dos algoritmos de segmentação se fundamenta em uma das propriedades básicas de valores de intensidade: descontinuidade e similaridade, conforme ilustrado na Figura 6. Na primeira categoria, o enfoque é dividir a imagem com base em mudanças abruptas de intensidade, como as bordas. Já na segunda categoria, a abordagem principal é dividir a imagem em regiões que sejam semelhantes com base em um conjunto de critérios predefinidos. Exemplos dessa abordagem incluem limiarização, crescimento de região e a divisão e fusão de regiões.

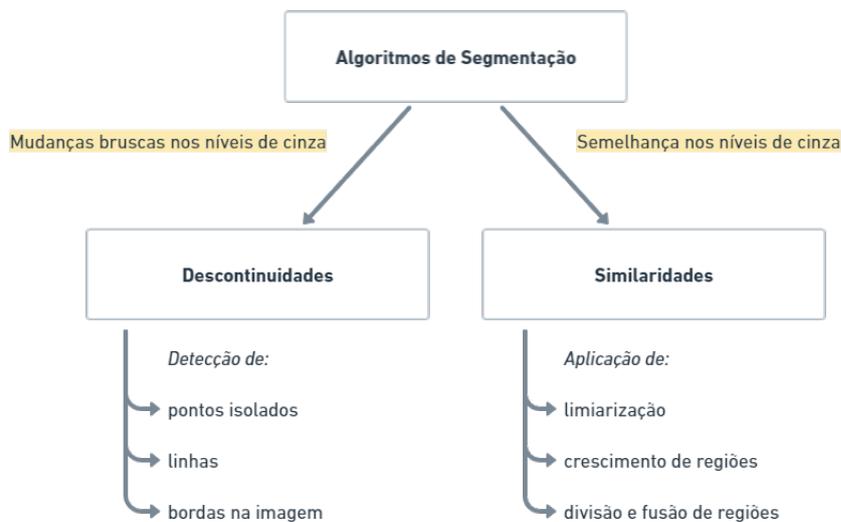


Figura 6 – Algoritmos de Segmentação.

Adaptado de (GONZALEZ; WOODS, 2010).

#### 2.2.4.1 Detecção de ponto, linha e borda

A detecção de pontos, linhas e bordas é realizada para identificar regiões de uma imagem que apresentam mudanças abruptas de intensidade ou descontinuidades, fornecendo informações importantes sobre os contornos e características dos objetos presentes na cena (GONZALEZ; WOODS, 2010).

As bordas são regiões onde ocorrem transições abruptas de intensidade, indicando

mudanças significativas nos níveis de cinza. Uma linha, por sua vez, pode ser considerada como um segmento de borda em que a intensidade do fundo de cada lado da linha é significativamente maior ou menor do que a intensidade dos pixels que compõem a própria linha. De maneira semelhante, um ponto isolado pode ser interpretado como uma linha em que o comprimento e a largura são reduzidos a poucos pixels (GONZALEZ; WOODS, 2010).

As mudanças locais abruptas na intensidade podem ser detectadas usando derivadas, em especial as derivadas de primeira e segunda ordem. Uma aproximação para a derivada de primeira ordem no ponto  $x$  de uma função bidimensional  $f(x)$  é obtida através da expansão da função  $f(x+\Delta x)$  em uma série de Taylor sobre  $x$ , assumindo que  $\Delta x=1$  mantendo apenas os termos lineares. O resultado é a diferença digital:

$$\frac{\partial f}{\partial x} = f'(x) = f(x+1) - f(x) \quad (2.6)$$

A partir da diferenciação da Equação 2.6 com relação a  $x$ , obtém-se uma expressão para a segunda derivada:

$$\frac{\partial^2 f}{\partial x^2} = f''(x) = f(x+1) + f(x-1) - 2f(x) \quad (2.7)$$

Segundo Gonzalez e Woods (2010), uma abordagem eficaz para calcular as derivadas de primeira e segunda ordem em cada posição de pixel em uma imagem é o uso de filtros espaciais. Para o filtro com máscara  $3 \times 3$  da Figura 7, calcula-se a soma dos produtos dos coeficientes da máscara com os valores de intensidade na região abrangida pela máscara. Logo, a resposta da máscara no ponto central da região é:

$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9 = \sum_{k=1}^9 w_k z_k \quad (2.8)$$

Na qual  $z_k$  representa a intensidade do pixel localizado espacialmente na posição do  $k$ -ésimo coeficiente da máscara.

$W_1$	$W_2$	$W_3$
$W_4$	$W_5$	$W_6$
$W_7$	$W_8$	$W_9$

Figura 7 – Máscara geral de filtro  $3 \times 3$ .

Adaptado de (GONZALEZ; WOODS, 2010).

## 2.2.4.2 Detecção de pontos isolados

A detecção de pontos baseia-se nas técnicas que utilizam as derivadas de segunda ordem, o que leva à utilização do laplaciano:

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (2.9)$$

Obtendo as derivadas parciais a partir da Equação 2.7, o laplaciano é, então:

$$\nabla^2 f(x, y) = f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1) - 4f(x, y) \quad (2.10)$$

Essa expressão pode ser implementada usando a máscara da Figura 8 (a) ou, a partir da extensão da Equação 2.10 para incluir os termos diagonais, a máscara da Figura 8 (b).

0	1	0	1	1	1
1	-4	1	1	-8	1
0	1	0	1	1	1
a			b		

Figura 8 – Máscaras 3 x 3.

Adaptado de (GONZALEZ; WOODS, 2010).

Gonzalez e Woods (2010) explicam que um ponto é identificado no local  $(x, y)$  em que a máscara está centrada se o valor absoluto da resposta da máscara nesse ponto exceder um limiar predefinido. Esses pontos são rotulados como 1 na imagem resultante, enquanto todos os outros pontos são rotulados como 0, resultando em uma imagem binária. Em termos mais simples, a saída é calculada usando a seguinte expressão:

$$g(x, y) = \begin{cases} 1 & \text{se } |R(x, y)| \geq T \\ 0 & \text{caso contrário} \end{cases} \quad (2.11)$$

Na qual  $g$  é a imagem de saída,  $T$  é um limiar não negativo, e  $R$  é dado pela Equação 2.8.

A Figura 9 ilustra a segmentação de pontos isolados em uma radiografia de uma lâmina de turbina de um motor de um jato, a partir da utilização da máscara da Figura 8 (b).

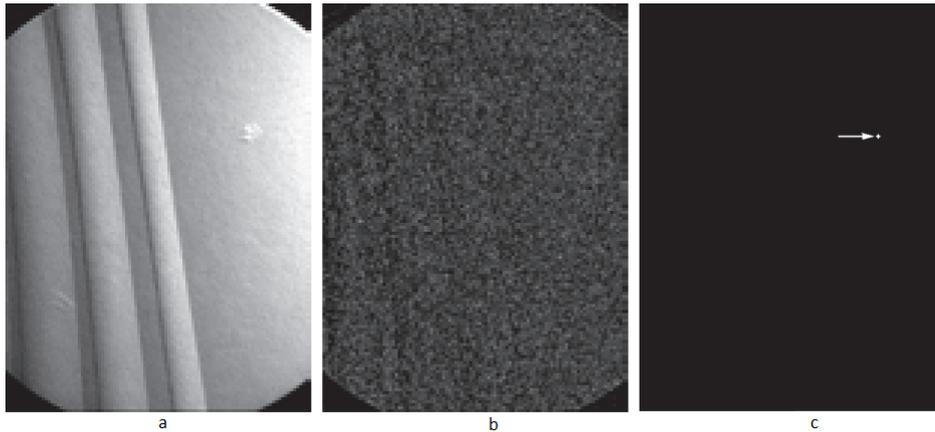


Figura 9 – Detecção de ponto isolado. (a) Imagem original da radiografia da uma lâmina de turbina com porosidade de um único pixel preto. (b) Imagem filtrada, resultante da convolução da máscara com a imagem. (c) Imagem após aplicação do limiar, resultado da utilização da Equação 2.11. [Fonte: (GONZALEZ; WOODS, 2010)]

### 2.2.4.3 Detecção de linhas

Para a detecção de linha, espera-se que as derivadas de segunda ordem, em detrimento das derivadas de primeira ordem, resultem em uma resposta mais forte e produzam linhas mais finas (GONZALEZ; WOODS, 2010). Isso permite, portanto, a utilização da máscara laplaciana da Figura 8 (b) para a detecção de linhas também.

A Figura 10 ilustra a utilização do filtro laplaciano para a detecção de linhas.

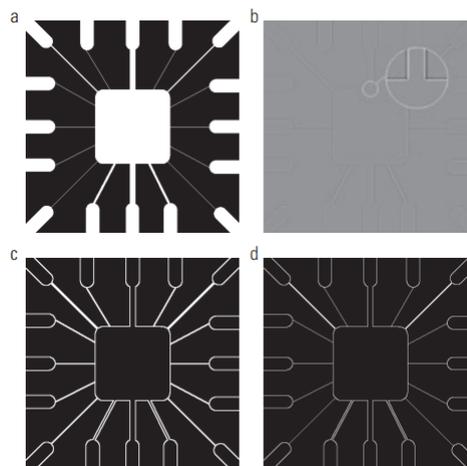


Figura 10 – Detecção de linhas. (a) Imagem original. (b) Imagem resultante do filtro laplaciano, e parte ampliada desta imagem. (c) Valor absoluto do filtro laplaciano. (d) Valores positivos do filtro laplaciano. [Fonte: (GONZALEZ; WOODS, 2010)]

Gonzalez e Woods (2010) ressaltam que o detector laplaciano é isotrópico e, logo,

sua resposta independe da direção. Ocasionalmente, o interesse reside na detecção de linhas em direções específicas e, para isso, devem ser consideradas as máscaras da Figura 11.

-1	-1	-1	2	-1	-1	-1	2	-1	-1	2	
2	2	2	-1	2	-1	-1	2	-1	2	-1	
-1	-1	-1	-1	-1	2	-1	2	-1	2	-1	
Horizontal			+45°			Vertical			-45°		

Figura 11 – Máscaras de detecção de linhas. Adaptado de (GONZALEZ; WOODS, 2010).

#### 2.2.4.4 Detecção de bordas

A detecção de bordas é o método mais comumente utilizado para realizar a segmentação de imagens, identificando regiões onde ocorrem mudanças abruptas e locais de intensidade (GONZALEZ; WOODS, 2010).

Gonzalez e Woods (2010) classificam os modelos de borda (Figura 12) de acordo com seus perfis de intensidade:

- Borda em degrau: envolve uma transição entre dois níveis de intensidade que ocorrem idealmente com uma distância de 1 pixel, conforme ilustrado na Figura 12 (a);
- Borda em rampa: bordas desfocadas e com ruído, cujo grau de indefinição se deve, em especial, às limitações no mecanismo de focalização e o nível de ruído é proveniente dos componentes eletrônicos do sistema de imagens. Nesses casos, é mais apropriado modelar as bordas como um perfil de rampa na intensidade, como a borda da Figura 12 (b);
- Borda em telhado: são representações modeladas de linhas que atravessam uma região, onde a largura da base de uma borda em forma de telhado é determinada pela espessura e nitidez da linha. Essa representação visual descreve uma borda com uma base mais ampla, que gradualmente se estreita em direção ao centro da linha, semelhante ao formato de um telhado, conforme a Figura 12 (c).

Gonzalez e Woods (2010) abordam a existência de três passos fundamentais a serem considerados na detecção das bordas:

1. Suavização da imagem para redução do ruído;
2. Detecção dos pontos de borda;
3. Localização da borda.

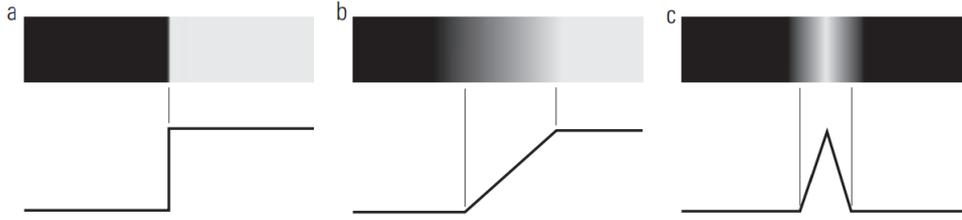


Figura 12 – Modelos de borda e seus perfis de intensidade. [Fonte: (GONZALEZ; WOODS, 2010)]

Mudanças na intensidade podem ser detectadas, com a finalidade de encontrar bordas, a partir de derivadas de primeira ou de segunda ordem. A ferramenta ideal para encontrar a intensidade e a direção da borda na posição  $(x, y)$  de uma imagem,  $f$ , é o gradiente, denotado por  $\nabla f$ , e definido como o vetor:

$$\nabla f = \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (2.12)$$

A magnitude do vetor  $\nabla f$ , denominado  $M(x, y)$ , é o valor da taxa de variação na direção do vetor gradiente:

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2} \quad (2.13)$$

A direção do vetor gradiente é dada pelo ângulo  $\alpha$ , medido em relação ao eixo  $x$ :

$$\alpha(x, y) = \arctan \left[ \frac{g_y}{g_x} \right] \quad (2.14)$$

Para a obtenção do gradiente de uma imagem, faz-se necessário o cálculo das derivadas parciais  $\frac{\partial f}{\partial x}$  e  $\frac{\partial f}{\partial y}$  em qualquer posição da imagem:

$$g_x = \frac{\partial f(x, y)}{\partial x} = f(x + 1, y) - f(x, y) \quad (2.15)$$

$$g_y = \frac{\partial f(x, y)}{\partial y} = f(x, y + 1) - f(x, y) \quad (2.16)$$

Essas duas equações podem ser implementadas para todos os valores pertinentes  $x$  e  $y$ , filtrando  $f(x, y)$  com as máscaras 1-D na Figura 13.

Quando os objetos de interesse apresentam bordas inclinadas (diagonais), é necessário utilizar máscaras 2-D específicas. Máscaras de tamanho  $2 \times 2$  são conceitualmente simples, mas não são tão adequadas para calcular a direção da borda quanto às máscaras

$$\begin{array}{c} \text{a} \\ \boxed{\begin{array}{c} -1 \\ 1 \end{array}} \end{array} \quad \begin{array}{c} \text{b} \\ \boxed{\begin{array}{cc} -1 & 1 \end{array}} \end{array}$$

Figura 13 – Máscaras unidimensionais. Adaptado de (GONZALEZ; WOODS, 2010).

simétricas em relação ao ponto central, sendo as menores máscaras geralmente de tamanho 3x3. Essas máscaras levam em consideração a natureza dos dados nos lados opostos ao ponto central, o que significa que carregam informações mais precisas sobre a direção da borda (GONZALEZ; WOODS, 2010).

A Figura 14 ilustra as máscaras de filtro usadas para calcular as derivadas necessárias para o gradiente e são frequentemente chamadas de operadores de gradiente, operadores de diferença, operadores de borda ou detectores de borda.

$$\begin{array}{c} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \text{Roberts} \\ \\ \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{Prewitt} \\ \\ \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{Sobel} \\ \\ \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \text{Prewitt} \\ \text{diagonais} \\ \\ \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \text{Sobel} \\ \text{diagonais} \end{array}$$

Figura 14 – Máscaras de filtros. Adaptado de (GONZALEZ; WOODS, 2010).

O operador Roberts é um filtro de convolução que calcula a magnitude do gradiente em duas direções ortogonais. Ele é relativamente simples, mas menos robusto em relação a ruídos e bordas finas.

Os operadores Prewitt e Sobel são filtros de convolução que calculam as derivadas parciais horizontais e verticais da imagem para obter a magnitude do gradiente. O operador Prewitt possui pesos fixos para os pixels vizinhos, enquanto o operador Sobel atribui pesos maiores aos pixels mais centrais, proporcionando uma resposta mais suave. Ambos são amplamente utilizados devido à sua eficácia na detecção de bordas.

Os operadores Prewitt diagonais e Sobel diagonais são variações dos operadores Prewitt e Sobel, respectivamente. Eles calculam as derivadas parciais diagonais para re-

alçar bordas diagonais ou inclinadas.

Gonzalez e Woods (2010) abordam a existência de técnicas mais avançadas utilizadas para a detecção de bordas, que levam em consideração fatores como ruído da imagem e a natureza das próprias bordas, como, por exemplo, o detector de borda Marr-Hildreth e o detector de Bordas de Canny. Esses detectores geralmente exigem maior demanda computacional em comparação com os demais. Isso se deve à natureza mais complexa desses algoritmos e às etapas adicionais envolvidas no processo de detecção de bordas.

O detector de borda Marr-Hildreth é baseado na convolução da imagem com um kernel Laplaciano de Gaussiana (LoG), seguido pela detecção de cruzamentos de zero. A convolução com o LoG é uma operação computacionalmente intensiva, pois envolve a combinação de operações de convolução e aplicação da função Gaussiana. Além disso, a detecção de cruzamentos de zero requer o processamento adicional para identificar pontos onde a derivada segunda do LoG atravessa zero (GONZALEZ; WOODS, 2010).

O detector de bordas de Canny é conhecido por ser um dos mais sofisticados e eficazes. Ele envolve várias etapas, incluindo a aplicação do filtro gaussiano para suavização, cálculo do gradiente da imagem, supressão de não-máximos para afinar as bordas e aplicação de um limiar duplo para selecionar as bordas finais. Cada uma dessas etapas requer operações computacionalmente intensivas, como convoluções e cálculos de gradientes (GONZALEZ; WOODS, 2010).

#### 2.2.4.5 Limiarização

Devido às suas propriedades intuitivas, facilidade de implementação e eficiência computacional, a limiarização de imagens desempenha um papel fundamental nas aplicações de segmentação de imagem (GONZALEZ; WOODS, 2010).

Ao contrário da detecção de bordas, onde as regiões são segmentadas a partir da identificação dos segmentos de borda e, em seguida, da vinculação dos segmentos com as fronteiras, na limiarização, a divisão de imagens em regiões é feita com base nos valores de intensidade e/ou nas propriedades desses valores (GONZALEZ; WOODS, 2010).

Considerando que o histograma de intensidade na Figura 15 (a) represente uma imagem,  $f(x, y)$ , composta por objetos claros em um fundo escuro, onde os pixels do objeto e do fundo tenham valores de intensidade agrupados em dois grupos dominantes (modos). Uma abordagem óbvia para extrair os objetos do fundo é selecionar um limiar  $T$ , que separa estes modos. Assim, qualquer ponto  $(x, y)$  na imagem em que  $f(x, y) > T$  é considerado um ponto do objeto e, caso contrário, o ponto é chamado ponto de fundo. Em outras palavras, a imagem segmentada,  $g(x, y)$ , é dada por:

$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) > T \\ 0 & \text{se } f(x, y) \leq T \end{cases} \quad (2.17)$$

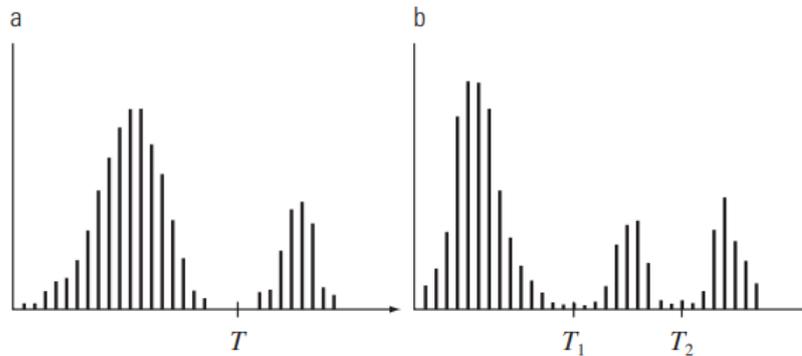


Figura 15 – Histogramas de intensidade que podem ser divididos (a) por um limiar único e (b) por limiares duplos. [Fonte: (GONZALEZ; WOODS, 2010)]

No caso da Figura 15 (b), o problema de limiarização se torna mais difícil, por envolver um histograma com três modos dominantes que correspondem, por exemplo, a dois tipos de objetos claros sobre um fundo escuro. Neste cenário, a limiarização múltipla classifica um ponto  $(x, y)$  como pertencente ao fundo se  $f(x, y) = T_1$ , para uma classe de objeto se  $T_1 < f(x, y) \leq T_2$ , e para a classe de outro objeto se  $f(x, y) > T_2$ . A imagem segmentada, portanto, é dada por:

$$g(x, y) = \begin{cases} a & \text{se } f(x, y) > T_2 \\ b & \text{se } T_1 < f(x, y) \leq T_2 \\ c & \text{se } f(x, y) \leq T_1 \end{cases} \quad (2.18)$$

Na qual  $a$ ,  $b$  e  $c$  são três valores quaisquer com diferentes níveis de intensidade.

Gonzalez e Woods (2010) destacam o uso de alguns termos para a definição de diferentes tipos de limiarização, embora não sejam universais:

- Limiarização global simples: Quando  $T$  é uma constante aplicável a uma imagem inteira;
- Limiarização variável: Quando o valor de  $T$  muda ao longo da imagem;
- Limiarização local ou regional: limiarização variável na qual o valor de  $T$  em qualquer ponto  $(x, y)$  em uma imagem depende das propriedades de uma vizinhança de  $(x, y)$ ;

- Limiarização dinâmica ou adaptativa: limiarização variável na qual  $T$  depende das coordenadas espaciais  $(x, y)$ .

#### 2.2.4.6 Limiarização global simples

Quando há uma distinção clara entre as distribuições de intensidade dos pixels do fundo e dos objetos, é viável utilizar um único limiar global que seja aplicado a toda a imagem (GONZALEZ; WOODS, 2010).

Gonzalez e Woods (2010) definem o algoritmo iterativo a seguir para a limiarização global:

1. Selecionar uma estimativa inicial para o limiar global,  $T$ ;
2. Segmentar a imagem usando  $T$  na Equação 2.17;
3. Calcular os valores de intensidade média de  $m_1$  e  $m_2$  para os pixels em  $G_1$  e  $G_2$ , respectivamente.
4. Calcular um novo valor de limiar:

$$T = \frac{1}{2}(m_1 + m_2) \quad (2.19)$$

5. Repetir as etapas 2 a 4 até que a diferença entre os valores de  $T$  em iterações sucessivas seja menor que o parâmetro predefinido  $\Delta T$ .

A Figura 16 ilustra um exemplo de segmentação baseada na estimativa do limiar utilizando o algoritmo descrito acima.

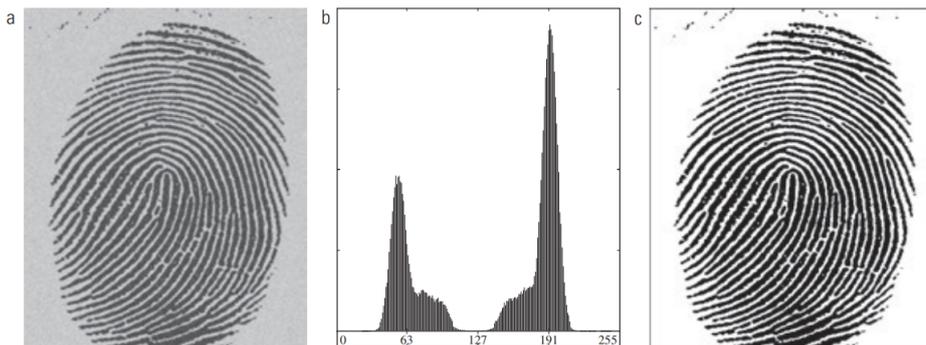


Figura 16 – Exemplo de Limiarização Global Simples. (a) Impressão digital ruidosa. (b) Histograma. (c) Segmentação resultante usando um limiar global. [Fonte: (GONZALEZ; WOODS, 2010)]

#### 2.2.4.7 Limiarização global ótima com o método de Otsu

O método de Otsu utiliza a função de densidade de probabilidade para determinar um limiar ótimo que maximize a variância entre as classes. Em outras palavras, ele

busca encontrar um valor de limiar que maximize a separabilidade entre os objetos e o fundo, levando em consideração as distribuições de intensidade dos pixels (GONZALEZ; WOODS, 2010).

Segundo Gonzalez e Woods (2010), o algoritmo de Otsu pode ser assim resumido:

1. Calcular o histograma normalizado da imagem de entrada. Designar os componentes do histograma como  $p_i$ ,  $i = 0, 1, 2, \dots, L - 1$ .

2. Calcular as somas acumuladas,  $P_1(k)$ , para  $k = 0, 1, 2, \dots, L - 1$ , usando a Equação 2.20:

$$P_1(k) = \sum_{i=0}^k p_i \quad (2.20)$$

3. Calcular as médias acumuladas  $m(k)$ , para  $k = 0, 1, 2, \dots, L - 1$ , usando a Equação 2.21:

$$m(k) = \sum_{i=0}^k ip_i \quad (2.21)$$

4. Calcular a intensidade média global,  $m_G$ , usando a Equação 2.22:

$$m_G = \sum_{i=0}^{L-1} ip_i \quad (2.22)$$

5. Calcular a variância entre classes,  $\sigma_B^2(k)$ , para  $k = 0, 1, 2, \dots, L - 1$ , usando a Equação 2.23:

$$\sigma_B^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]} \quad (2.23)$$

6. Obter o limiar de Otsu,  $k^*$ , como o valor de  $k$  para o qual  $\sigma_B^2(k)$  é máxima. Se a máxima não for única, obter  $k^*$  pela média dos valores de  $k$  que correspondem aos diversos valores máximos detectados.

7. Obter a medida de separabilidade,  $\eta^*$ , avaliando a Equação 2.24 em  $k = k^*$ .

$$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_G^2} \quad (2.24)$$

A Figura 17 (a) apresenta uma imagem de microscópio ótico de células polimerossomas e a utilização da limiarização global ótima usando o método de Otsu com o objetivo de segmentar as moléculas do fundo.

#### 2.2.4.8 Limiarização variável

Conforme abordado por Gonzalez e Woods (2010), fatores como ruído e iluminação não-uniforme desempenham um papel crucial no desempenho de algoritmos de limiarização. Embora a suavização da imagem e a utilização da informação das bordas

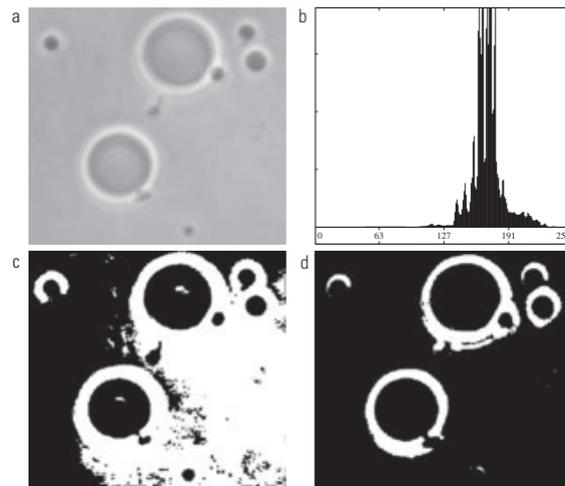


Figura 17 – Exemplo de Limiarização global ótima com o método de Otsu. (a) Imagem original. (b) Histograma. (c) Resultado da segmentação utilizando o algoritmo global básico. (d) Resultado obtido pelo método de Otsu. [Fonte: (GONZALEZ; WOODS, 2010)]

possam ser eficazes para melhorar os resultados, muitas vezes há casos em que esses tipos de pré-processamento são impraticáveis ou simplesmente não proporcionam uma melhora significativa da situação.

Nesses casos, surge a necessidade de um nível mais complexo de limiarização, conhecido como limiarização variável. Gonzalez e Woods (2010) elencam algumas técnicas para a escolha dos limiares variáveis:

- Particionamento da imagem: subdividir uma imagem em retângulos que se sobrepõem;
- Limiarização variável baseada nas propriedades locais da imagem: um limiar é calculado para cada ponto  $(x, y)$  da imagem, com base em uma ou mais propriedades calculadas em sua vizinhança;
- Usando médias de movimento: o limiar é calculado com base na média de uma janela linear com  $n$  pixels que se move ao longo de cada linha da imagem.

#### 2.2.4.9 Segmentação baseada na região

A segmentação baseada na região utiliza de técnicas de segmentação que se concentram em encontrar as regiões de forma direta, sem usar as informações de bordas ou limiares. Existem várias abordagens tradicionais para segmentar regiões em uma imagem, incluindo o crescimento de região e a divisão e fusão de regiões. Esses métodos têm sido amplamente utilizados para extrair regiões coesas e homogêneas na segmentação baseada

em região. No entanto, além dessas técnicas, existem outras abordagens que também são empregadas no processo de segmentação, mas que não serão abordadas neste trabalho em particular, como a segmentação usando algoritmos de clusterização, segmentação usando superpixels e a segmentação usando particionamento ótimo em grafos.

#### 2.2.4.10 Segmentação por crescimento de região

O crescimento de região é um procedimento que busca agrupar pixels ou sub-regiões em regiões maiores, baseado em critérios predefinidos para o crescimento. Como o próprio nome sugere, a abordagem básica envolve iniciar com um conjunto de pontos "semente" e, a partir dessas sementes, permitir que as regiões cresçam ao anexar pixels vizinhos que possuam propriedades semelhantes às das sementes, como intervalos específicos de intensidade ou cor (GONZALEZ; WOODS, 2010).

Supondo que  $f(x, y)$  seja uma matriz representando a imagem de entrada,  $S(x, y)$  seja uma matriz contendo as sementes, onde os valores 1 indicam as posições das sementes e os valores 0 indicam as demais localizações, e  $Q$  seja uma propriedade a ser aplicada em cada posição  $(x, y)$ . As matrizes  $f$  e  $S$  têm o mesmo tamanho. O algoritmo básico de crescimento de região é baseado na conectividade-8 e pode ser definido da seguinte forma (GONZALEZ; WOODS, 2010):

1. Encontrar todos os componentes conectados em  $S(x, y)$  e erodir cada componente conectado a um pixel; rotular todos os pixels encontrados com o número 1. Todos os outros pixels em  $S$  recebem 0.
2. Formar uma imagem  $f_Q$  que, em um par de coordenadas  $(x, y)$ , deixe  $f_Q(x, y) = 1$  se a imagem de entrada satisfaz a propriedade determinada,  $Q$ , nessas coordenadas; caso contrário, deixe  $f_Q(x, y) = 0$ .
3. Digamos que  $g$  é uma imagem formada anexando a cada semente em  $S$  todos os pontos rotulados com o número 1 em  $f_Q$  que estão 8-conectados a essa semente.
4. Rotular cada componente conectado em  $g$  com uma diferente etiqueta de região (por exemplo, 1, 2, 3, ...). Esta é a imagem segmentada obtida pelo crescimento de região.

A Figura 18 (a) ilustra a imagem de raios X de 8 bits de uma solda, contendo várias fissuras e porosidades. O crescimento de região é utilizado para segmentar as regiões com defeito de solda.

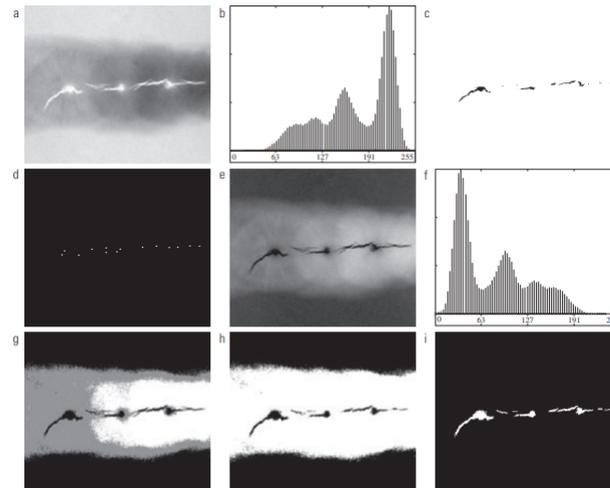


Figura 18 – Exemplo de segmentação utilizando crescimento de regiões. (a) Imagem de raios X de uma solda defeituosa. (b) Histograma. (c) Imagem inicial com as sementes. (d) Imagem final com as sementes. (e) Valor absoluto da diferença entre (a) e (c). (f) Histograma de (e). (g) Imagem da diferença após uma limiarização utilizando limiares duplos. (h) Imagem da diferença limiarizada com o menor dos limiares duplos. (i) Resultado da segmentação obtido pela técnica de crescimento de região. [Fonte: (GONZALEZ; WOODS, 2010)]

#### 2.2.4.11 Segmentação por divisão e fusão de região

Como uma alternativa ao método de segmentação por crescimento de região, existe a abordagem de subdivisão da imagem. Nessa abordagem, a imagem é inicialmente dividida em um conjunto de regiões distintas e arbitrárias. Posteriormente, são realizadas operações de fusão e/ou divisão dessas regiões na tentativa de satisfazer as condições desejadas de segmentação (GONZALEZ; WOODS, 2010).

Gonzalez e Woods (2010) destacam o algoritmo de divisão e fusão de região como:

1. Dividir em quatro quadrantes separados qualquer região  $R_i$  para a qual  $Q(R_i) = \text{FALSO}$ .
2. Quando não for possível continuar dividindo, fundir as regiões adjacentes  $R_j$  e  $R_k$  para as quais  $Q(R_j \cup R_k) = \text{VERDADE}$ .
3. Parar quando a fusão não for mais possível.

A Figura 19 (a) mostra uma imagem de raio X de  $566 \times 566$  pixels da Cygnus Loop. O exemplo tem por objetivo retirar da imagem o “anel” de matéria menos densa em torno do centro denso.

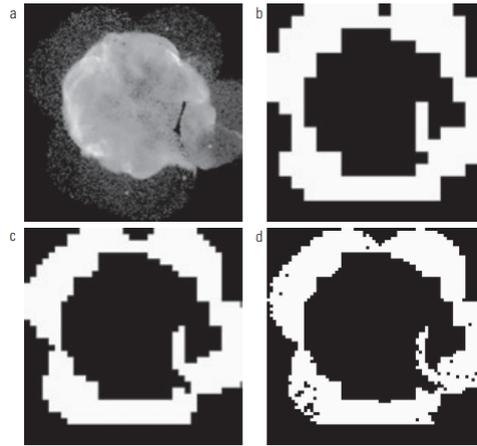


Figura 19 – Exemplo de segmentação por divisão e fusão de regiões. (a) Imagem da supernova Cygnus Loop, tomada na banda de raios X pelo telescópio Hubble da Nasa. (b) a (d) Resultados de limitar a menor quadrirregião permitida aos tamanhos de  $32 \times 32$ ,  $16 \times 16$ , e  $8 \times 8$  pixels, respectivamente. [Fonte: (GONZALEZ; WOODS, 2010)]

#### 2.2.4.12 Segmentação usando *watersheds* morfológicas

O conceito de *watershed* (bacias hidrográficas) baseia-se na visualização de uma imagem em três dimensões: duas coordenadas espaciais versus intensidade, como na Figura 20.

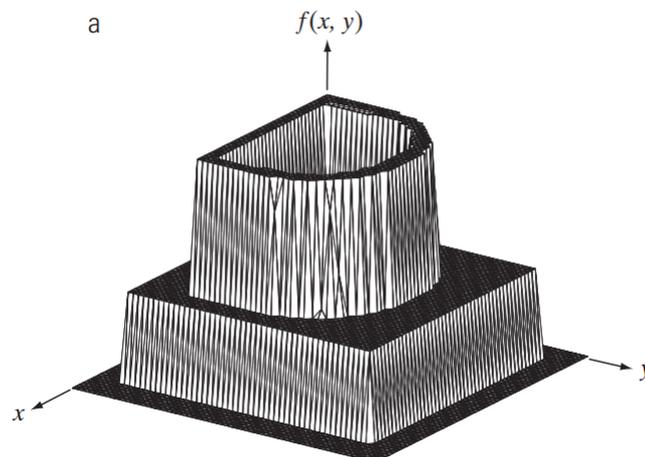


Figura 20 – Imagem representada graficamente como uma superfície. [Fonte: (GONZALEZ; WOODS, 2010)]

Em uma interpretação “topográfica” desta, são considerados três tipos de pontos distintos:

- (a) Pontos que pertencem a um mínimo regional, ou seja, pontos em que a inten-

cidade é menor do que todos os seus vizinhos.

(b) Pontos em que uma gota d'água, se despejada em sua localização, cairia quase certamente em um mínimo simples, ou seja, apenas em um único ponto de mínima intensidade.

(c) Pontos em que a água tem a mesma probabilidade de cair em mais de um mínimo regional.

Para um mínimo regional específico, o conjunto de pontos que satisfaz a condição (b) é conhecido como bacia hidrográfica, *catchment basin* ou *watershed* desse mínimo. Por outro lado, os pontos que satisfazem a condição (c) formam linhas de crista na superfície topográfica e são chamados de linhas de divisão ou linhas de *watershed*. Nos algoritmos de segmentação baseados nesses conceitos, o principal objetivo é encontrar as linhas de *watershed*, ou seja, identificar as fronteiras entre as diferentes regiões da imagem (GONZALEZ; WOODS, 2010).

A utilização deste método é ilustrada na Figura 21.

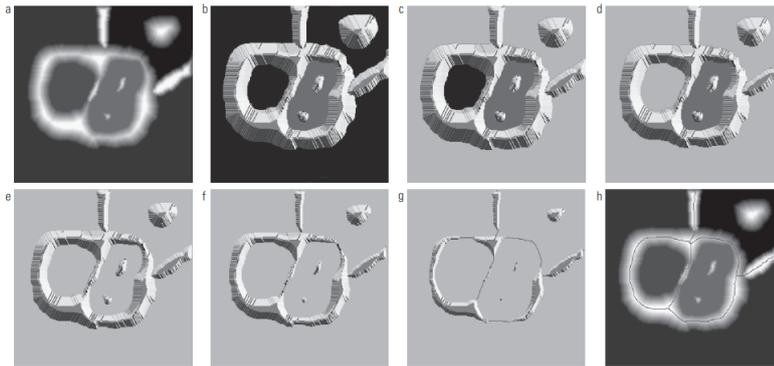


Figura 21 – Exemplo de segmentação usando watersheds morfológicas. (a) Imagem original. (b) Vista topográfica. (c) e (d) Duas fases da inundação. (e) Resultado de novas inundações. (f) Começo da fusão da água de duas watersheds. (g) Barragens maiores. (h) Linhas finais da watershed. [Fonte: (GONZALEZ; WOODS, 2010)]

A segmentação de *watershed* é uma técnica adequada para separar objetos com textura quase uniforme do fundo em imagens. Essa abordagem é especialmente eficaz quando aplicada à imagem gradiente, pois os mínimos regionais estão diretamente relacionados aos baixos valores de gradiente nos objetos de interesse (GONZALEZ; WOODS, 2010).

### 2.2.5 Extração de atributos

Depois da segmentação da imagem, é comum extrair características específicas dos objetos segmentados. Isso envolve a identificação e a descrição de atributos relevantes,

como forma, cor, textura, entre outros, que possam ser usados para caracterizar os objetos de interesse.

Com base nas características extraídas, é possível aplicar algoritmos de classificação ou reconhecimento de padrões para atribuir rótulos ou realizar a identificação dos objetos segmentados. Essa etapa pode envolver o uso de técnicas de aprendizado de máquina, como redes neurais, SVM (Support Vector Machines) ou árvores de decisão.

Por fim, os resultados obtidos após a segmentação e as etapas subsequentes podem ser visualizados e interpretados. Isso pode envolver a geração de imagens ou gráficos que destacam os objetos segmentados ou a apresentação de informações relevantes de forma compreensível.

## 2.3 Visão computacional

A visão computacional é um campo da inteligência artificial que se concentra no desenvolvimento de algoritmos e técnicas para permitir que computadores possam interpretar o mundo visual. A visão computacional envolve a captura, o processamento e a análise de imagens, assim como a extração de informações úteis a partir delas.

A visão computacional tradicional é uma abordagem para a análise de imagens digitais por meio de técnicas de processamento de imagem e análise estatística. Essa abordagem é baseada na extração de características visuais e na identificação de padrões e estruturas na imagem. A visão computacional tradicional geralmente envolve várias etapas de processamento de imagem que consiste no pré-processamento de imagem, segmentação de imagem, extração de características, classificação e reconhecimento de objetos. Essa sequência de etapas executadas em ordem é nomeada de *pipeline*, onde cada etapa é projetada para transformar a imagem em uma representação mais útil para a próxima etapa do processo.

Embora a visão computacional tradicional seja amplamente utilizada em diversas aplicações, sendo um conjunto de técnicas de complexidade variável, faz-se necessária uma abordagem manual para ajustar os parâmetros e selecionar recursos, além de possuir limitações em ambientes não controlados.

Com o desenvolvimento de técnicas de aprendizado profundo e redes neurais, a visão computacional tem evoluído significativamente para uma abordagem mais automatizada e eficaz (SANTOS et al., 2019). As redes neurais são modelos matemáticos inspirados no funcionamento do cérebro humano que são utilizados na inteligência artificial para realizar tarefas de aprendizado de máquina. Essas redes são compostas por camadas de "neurônios artificiais" interconectados, que processam e transformam dados de entrada em dados de saída.

As redes neurais são capazes de lidar com grandes volumes e variedades de dados complexos e permitem que recursos sejam aprendidos automaticamente. Por este motivo, frequentemente superam as técnicas tradicionais em termos de precisão. Entretanto, é uma abordagem que requer grandes quantidades de dados de treinamento para produzir resultados precisos e eficazes, além de exigir recursos de computação de alto desempenho.

### 2.3.1 OpenCV

O *OpenCV* (*Open Source Computer Vision*) é uma biblioteca de software amplamente utilizada para visão computacional e processamento de imagens ([OpenCV Team, 2023](#)). Ele fornece uma ampla gama de algoritmos e funções eficientes para a manipulação, análise e compreensão de imagens e vídeos em tempo real. Sua facilidade de uso, documentação abrangente e ampla comunidade de desenvolvedores o tornam uma escolha popular para projetos que envolvem visão computacional.

Esta biblioteca possui algoritmos para realizar tarefas comuns, como filtragem de imagens, detecção de bordas, segmentação, correspondência de pontos-chave, entre outras. Inclui algoritmos para detecção e reconhecimento de objetos, abrangendo a detecção de rostos, identificação de objetos em tempo real, rastreamento de objetos em vídeos e classificação de características distintas.

O *OpenCV* é altamente otimizado para executar operações de processamento de imagens em tempo real, tornando-o adequado para aplicativos de visão computacional que exigem baixa latência.

## 2.4 Estado da Arte

Existem variados estudos que utilizam técnicas de processamento de imagem para detecção de piso tátil em diferentes aplicações e localidades. Essas técnicas incluem o uso de aprendizado de máquina, redes neurais convolucionais e visão computacional tradicional, a fim de identificar e classificar os padrões dos pisos táteis.

Um estudo de [EINLOFT; GHILARDI; MANSSOUR](#) teve como objetivo apresentar um modelo, baseado em técnicas de visão computacional, para auxiliar deficientes visuais a localizar, identificar e seguir as superfícies de pavimentação tátil em ambientes internos. O método utilizado no estudo analisa as imagens capturados por uma câmera em busca de uma possível área de superfície tátil. Se essa área for identificada, o algoritmo analisa sua textura e decide se existe, de fato, uma superfície de pavimentação tátil. Em caso positivo, utiliza-se uma matriz de ocorrência de níveis de cinza (GLCM) e um modelo de aprendizado supervisionado chamado máquina de vetores de suporte (SVM, do inglês *Support Vector Machine*) para detectar a textura e diferenciar entre superfícies

de pavimentação táteis direcionais e de alerta. Em seguida, um feedback do usuário é fornecido em formato de áudio (EINLOFT; GHILARDI; MANSSOUR, 2016).

O trabalho de KASSIM et al. aborda o desempenho de um algoritmo de detecção tátil baseado em visão computacional, utilizado para o reconhecimento das formas de pisos táteis para fins de navegação. Um cálculo métrico é usado para detectar as formas. A métrica na codificação funciona calculando a área e o perímetro de qualquer componente conectado em uma imagem binária após o pré-processamento. (KASSIM et al., 2018).

De acordo com o estudo de AKTAŞ; DEMIR; DOĞAN realizado na Turquia, metodologias de aprendizado profundo, sub-ramo da inteligência artificial, podem ser utilizadas para detecção em tempo real de superfícies de pavimentação táteis. É utilizado o modelo YOLO-V3, combinado com o modelo DenseNet para criar o modelo YOLOV3Dense, a fim de projetar um sistema de tecnologia assistiva que possa ser usado por pessoas com deficiência visual, veículos autônomos e robôs (AKTAŞ; DEMIR; DOĞAN, 2020).

O estudo de VIEIRA é referente à um sistema embarcado inteligente para a navegação autônoma de uma cadeira de rodas motorizada em um ambiente conhecido. A técnica de navegação utilizada é a odometria e reconhecimento de marcos artificiais através de pisos táteis, para a determinação do posicionamento absoluto da cadeira de rodas. (VIEIRA, 2019).



## 3 Implementação

Este capítulo destina-se à exposição detalhada do modelo geral proposto e à apresentação dos elementos necessários para a implementação do sistema. Apresenta-se a justificativa de escolha da plataforma computacional, o detalhamento do algoritmo proposto bem como as técnicas específicas de processamento de imagem a serem empregadas. O objetivo central consiste no desenvolvimento de um algoritmo eficiente e viável para ser implementado em um sistema embarcado.

### 3.1 Plataforma computacional

Existe uma diversidade de placas de desenvolvimento disponíveis no mercado, variando em complexidade e robustez. A escolha apropriada para um projeto depende de vários fatores, incluindo o nível de expertise do desenvolvedor, a disponibilidade de recursos, a escalabilidade desejada e a relação custo-benefício. Tendo isso em vista, neste trabalho o sistema embarcado será composto por uma *RaspBerry Pi 3 Model B+*, que será responsável pela captura, modelagem, síntese e processamento da imagem. A *RaspBerry* é considerada um microprocessador, mais especificamente um mini-computador de placa única (*SBC - Single Board Computer*). Embora possa ser considerada uma plataforma computacional embarcada, a *RaspBerry Pi* não é um microcontrolador, pois não é otimizada para tarefas específicas ou de baixo nível, esta placa destaca-se por sua capacidade de suportar elevada demanda de processamento e armazenamento. Essa capacidade é crucial para a execução eficiente de algoritmos mais complexos, como os requeridos neste projeto.



Figura 22 – Raspberry Pi 3 Model B+ [Fonte: (LTD, a)].

As principais especificações técnicas da Raspberry Pi 3 Model B+, de acordo com seu datasheet (LTD, a) , incluem:

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 1GB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header
- Full-size HDMI
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- 4-pole stereo output and composite video port
- Micro SD port for loading your operating system and storing data
- 5V/2.5A DC power input
- Power-over-Ethernet (PoE) support (requires separate PoE HAT)

Além dessas especificações, a Raspberry Pi 3 B+ possui suporte a protocolos de comunicação como I2C, SPI e UART.

## 3.2 Algoritmo

Segundo a ABNT NBR 16537:2016 (ABNT, 2016), quando houver mudança de direção com ângulo entre  $90^\circ$  e  $150^\circ$ , deve haver sinalização tátil de alerta, formando áreas de alerta com dimensão equivalente ao dobro da largura da sinalização tátil direcional, como mostra a Figura 23. Considerando este caso como objeto de estudo, pode-se diferenciar o piso direcional do piso de alerta através da comparação de área entre ambos.

Para a aplicação deste projeto embarcado, faz-se necessário utilizar uma técnica com baixa complexidade computacional, visto que deve ser embarcado em um microprocessador, e o tempo de processamento do algoritmo deve ser o menor possível. Sendo assim, serão utilizadas técnicas de visão computacional tradicionais, evitando técnicas de aprendizagem de máquina ou de inteligência artificial. O algoritmo utilizado deve localizar e identificar, de forma autônoma, pisos táteis em ambientes conhecidos, utilizando operadores simples que evitem cálculos complexos.

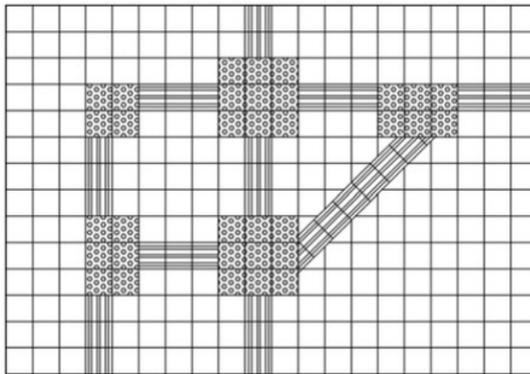


Figura 23 – Mudança de direção por pisos táteis [Fonte: (RACHID, 2013)].

As linguagens de programação de baixo nível, como C e C++, são comumente utilizadas para processamento de imagens em sistemas embarcados, devido ao seu desempenho computacional e acesso direto a recursos de baixo nível. Entretanto, *Python* é uma das linguagens mais populares para programar na Raspberry Pi. É uma linguagem de alto nível, de fácil aprendizado e possui uma grande comunidade de desenvolvedores com bibliotecas e frameworks que facilitam o desenvolvimento de projetos na Raspberry Pi. Por este motivo, a linguagem utilizada foi Python em conjunto com a biblioteca *OpenCV*.

As etapas do algoritmo são descritas na Figura 24. O primeiro passo é a aquisição da imagem, obtida através da captura de uma câmera digital, após isso é feito um pré-processamento da imagem, afim de melhorar suas condições, para assim efetuar a etapa de segmentação, onde separa-se a imagem em partes de interesse, neste caso, delimitando os pisos táteis. Por fim, analisa-se a proporção entre o número de pixels brancos e pretos em uma região para distinguir entre os diferentes tipos de pisos táteis, extraíndo os atributos e reconhecendo os padrões e suas classificações.

### 3.3 Aquisição da Imagem

Nesta etapa, o objetivo é adquirir imagens em tempo real, utilizando a Raspberry Pi Câmera Rev 1.3 como dispositivo de captura. O foco é minimizar a latência entre a captura e o processamento, atendendo às necessidades específicas do projeto. O módulo utilizado é leve (3g), compacto (25 x 24 x 9 mm), pode gerar fotos com resolução de até 2896 x 1944 pixels e vídeos com resolução de até 1080p a 30 fps com codec H.264 (AVC) (LTD, b). O modelo da câmera é ilustrado na Figura 25 e um melhor detalhamento das especificações da câmera é mostrado na Tabela 1.

A cadeira de rodas utilizada neste projeto já foi objeto de estudo em outras pesquisas da Universidade de Brasília. Na pesquisa conduzida pela BRANCO, foram instalados suportes laterais de madeira projetados à frente dos pneus dianteiros da cadeira, conforme visto na Figura 26. Aproveitando essa estrutura preexistente, a câmera foi acoplada à parte

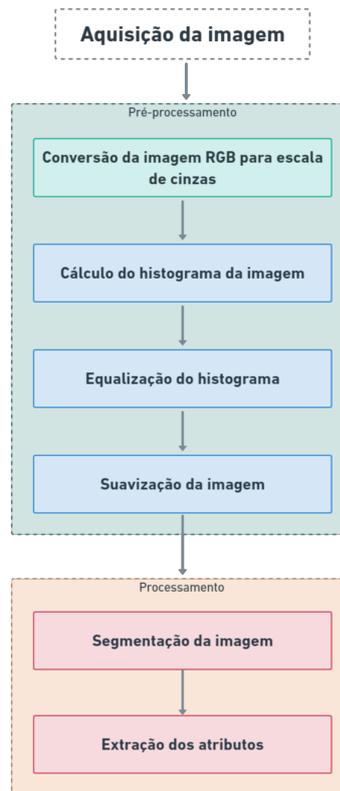


Figura 24 – Etapas do processamento de imagem.



Figura 25 – Raspberry Pi Câmera Rev 1.3 [Fonte: (LTD, b)].

frontal dos suportes laterais, como observado na Figura 27. A angulação da câmera em relação ao piso foi ajustada manualmente até alcançar a região de captura desejada. Dessa forma, essa configuração proporcionou uma solução prática e eficiente para a integração da câmera ao sistema, aproveitando a estrutura já existente na cadeira de rodas.

Em termos de software, a aquisição das imagens é realizada por meio de um método em *OpenCV* projetado para ler continuamente frames de um objeto de captura de vídeo associado à entrada da câmera. Essa abordagem é frequentemente empregada em estruturas de repetição para capturar frames sucessivos e executar operações de processamento de imagem em tempo real. O *loop* continuará até que não haja mais frames disponíveis para leitura ou até que o programa seja interrompido manualmente.

Tabela 1 – Características da Raspberry Pi Câmera Rev 1.3

Especificação	Valor
Sensor	OV5647
Resolução	5 MP
CCD Size	1/4 polegada
Abertura (F)	1.8
Comprimento focal	3.6mm (ajustável)
Angulo de visão (Diagonal)	75.7 graus
Dimensão	25 x 24 x 9 mm

Fonte: (LTD, b)



Figura 26 – Visão lateral da cadeira de rodas [Fonte: (BRANCO, 2018)].

### 3.4 Pré-processamento da Imagem

Como explícito no Capítulo 2.1, os pisos táteis podem variar em termos de cor e padrão de acordo com as normas e regulamentações locais. A escolha do modelo de cores para o processamento de imagens digitais pode depender das características específicas dos pisos táteis. Cada modelo oferece uma abordagem única para representar e manipular as cores com base em diferentes propriedades, como luz, pigmentação e percepção humana.

A Raspberry Pi Câmera Rev 1.3 captura imagens no modelo de cores RGB. Nos testes iniciais, realizados com vídeos do ambiente externo do prédio UAC do Campus Gama da Universidade de Brasília, observou-se que os pisos táteis apresentam uma mescla de cores, variando entre amarelo e vermelho (Figura 28). Diante desse cenário, optou-se por utilizar o modelo de cores HSV para ajustar os limiares de matiz, abrangendo a gama de cores que inclui tanto o amarelo quanto o vermelho. Esta escolha se mostrou mais eficaz em relação à escala de cinza, que não produziu os resultados esperados.

No entanto, este projeto está focado em infraestruturas *indoor*, e os testes finais foram conduzidos no primeiro andar do prédio UED do Campus Gama da Universidade de Brasília. Nesse local, os pisos táteis são exclusivamente pretos, como visto na Figura

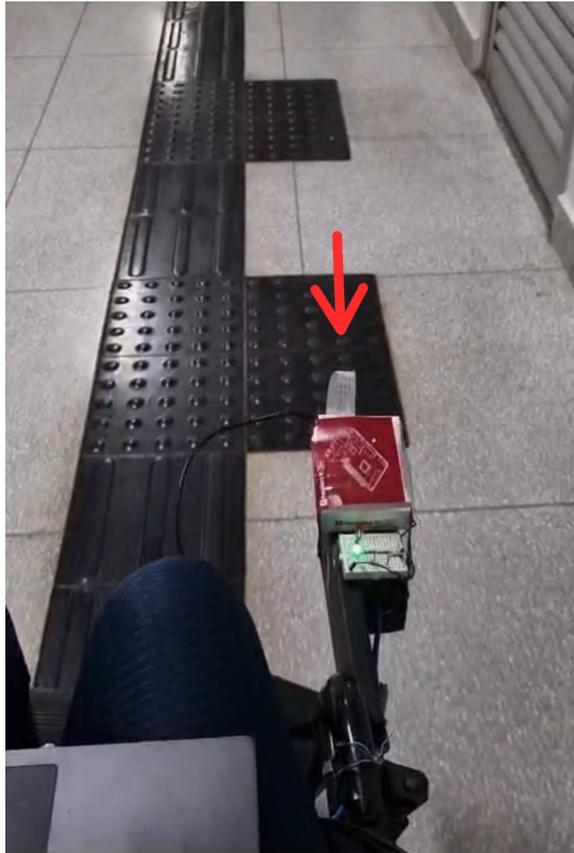


Figura 27 – Câmera acoplada à cadeira de rodas.



Figura 28 – Pisos táteis amarelos e vermelhos do prédio UAC do Campus Gama da Universidade de Brasília.

29 o que tornou mais viável a conversão para escala de cinza, uma vez que as informações de cor tornaram-se menos relevantes. A conversão para escala de cinza simplifica a análise de imagem e reduz a complexidade computacional.

A análise do histograma é necessária para equalizar a imagem, isto é, ajustar os níveis de cinza, garantindo um brilho e contraste balanceados. Outra etapa importante é aplicar a suavização na imagem, isto é, desfoque da imagem por meio de vários filtros passa-baixa, para minimizar o contraste em suas bordas e garantir condições ideais para a etapa de segmentação. A biblioteca OpenCV fornece as seguintes funções para



Figura 29 – Pisos táteis pretos do prédio UED do Campus Gama da Universidade de Brasília.

suavização da imagem:

- *cv.blur()*: Técnica de desfoque realizada através da convolução de uma imagem com um filtro de média normalizado, devendo-se especificar a altura e largura desejada do *kernel*.
- *cv.GaussianBlur()*: Técnica utilizada para aplicar um filtro gaussiano na imagem. Deve-se especificar o tamanho da vizinhança e o desvio padrão do filtro gaussiano.
- *cv.medianBlur()*: Função utilizada para aplicar um filtro de mediana na imagem. O parâmetro necessário é o tamanho do kernel do filtro de mediana, devendo ser um número ímpar positivo.

Conforme discutido no Capítulo 2.2.3 e confirmado pelas simulações realizadas, o filtro de média demonstrou demandar menor complexidade computacional em comparação com o filtro gaussiano e o filtro mediana. Isso se deve ao fato de que o filtro de média envolve operações matemáticas simples.

Diante disso, optou-se pela utilização da técnica *cv.blur()* para realizar a suavização das imagens. O tamanho do *kernel*, parâmetro essencial que influencia a intensidade da suavização aplicada à imagem, foi ajustado dinamicamente por meio de uma interface gráfica composta por uma barra deslizante (Figura 30). Ao modificar essa barra, o tamanho do *kernel* é atualizado como uma tupla, garantindo que seja sempre quadrado e com um tamanho ímpar, requisito comum para operações de suavização.

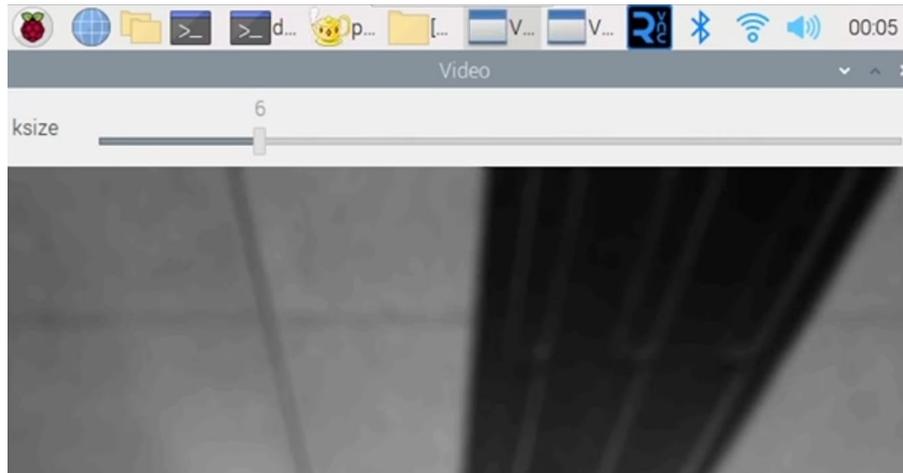


Figura 30 – Interface gráfica composta por uma barra deslizante para ajuste do tamanho do *kernel*.

### 3.5 Segmentação da Imagem

A limiarização (*thresholding*) é o método mais simples de segmentação de imagens e pode ser utilizado no processo de binarização para segmentar uma imagem em tons de cinza. Esta operação pressupõe que os componentes da imagem estejam separados em classes (como por exemplo, objetos em tons claros de cinza e com o fundo em tons escuros). Sabendo que, neste estudo, o piso a ser capturado é preto e o chão é mais claro, tem-se uma situação aplicável.

A função *cv.threshold* em *OpenCV* possui a seguinte assinatura:

```
retval, threshold = cv2.threshold(src, thresh, maxval, type, [dst])
```

Sendo:

- *src*: É a imagem de origem, que deve estar em escala de cinza.
- *thresh*: É o valor limiar para o qual pixels com valores acima e abaixo são classificados de maneira distinta.
- *maxval*: Valor a ser atribuído aos pixels que são maiores que o valor limiar (*thresh*).
- *type*: Tipo de técnica de binarização a ser aplicada.
- *dst* (opcional): A imagem de destino, do mesmo tamanho e tipo que *src*. Se não fornecido, a função cria uma imagem de destino.

Para a escolha correta do valor limiar (*thresh*), foi utilizada uma barra deslizante para ajuste manual com base nas imagens capturadas (Figura 31). Esse valor é sensível à qualidade da imagem, níveis de iluminação, contraste, angulação da câmera e outros fatores relevantes. O tipo de técnica de binarização escolhido foi a binarização simples,

onde os pixels com intensidade acima do valor de limiar (*thresh*) são definidos como *maxval* (255) e outros como zero (0). O resultado é uma imagem binarizada, onde os pixels são ajustados para serem pretos ou brancos, dependendo da comparação com o limiar, tendo assim, uma imagem de alto contraste.

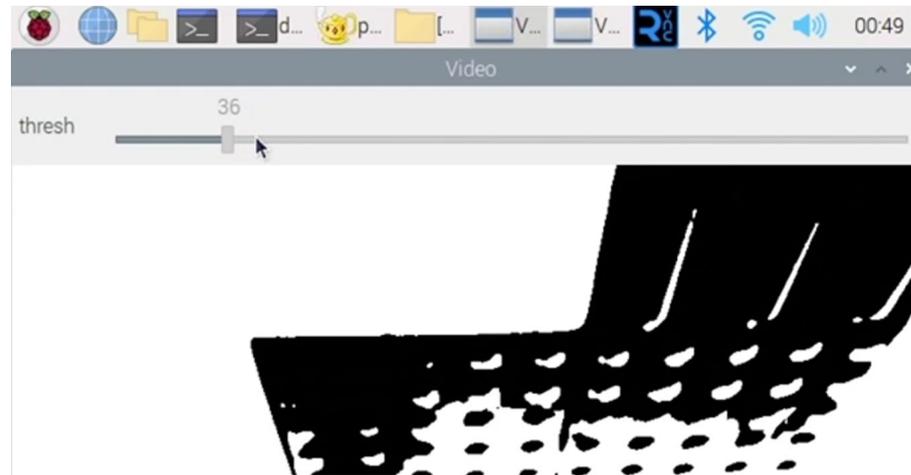


Figura 31 – Interface gráfica composta por uma barra deslizante para ajuste do valor limiar (*thresh*).

Essa abordagem proporciona flexibilidade na adaptação dos limiares, permitindo otimizar o processo de binarização para diferentes condições de captura de imagem.

## 3.6 Extração dos atributos

A partir da imagem binarizada, os pisos táteis são representados por pixels pretos, enquanto o fundo é representado por pixels brancos, conforme visto na Figura 31. A diferenciação entre pisos de alerta e direcionais é possível devido à maior largura dos pisos de alerta. Os frames com uma maior quantidade de pixels pretos são associados aos pisos de alerta, enquanto aqueles com quantidades menores correspondem aos pisos direcionais. A escolha do valor limiar, que distingue os pisos, é uma decisão manual e depende das características específicas de cada captura e cenário. Observa-se o valor médio de pixels pretos para cada tipo de piso, e a partir disso, determina-se o valor limiar. Abaixo desse limiar são identificados os pisos direcionais, e acima, os pisos de alerta. Para evitar erros e ruídos, estabeleceu-se que a transição entre os tipos de pisos (aumento ou diminuição dos pixels pretos) ocorre somente após 10 frames atingirem o valor limiar.

Em todas as etapas do algoritmo, o objetivo foi utilizar técnicas com o melhor *tradeoff* entre desempenho e tempo de execução.

### 3.7 Metodologia dos testes

Para validar a proposta de implementação, conduziram-se testes utilizando a câmera acoplada à cadeira de rodas. Inicialmente, os testes concentraram-se em imagens capturadas em condições ideais, visando a comparação de diferentes técnicas de processamento para a escolha daquelas que demandassem menor tempo de processamento. Posteriormente, a atenção voltou-se para os testes em capturas de vídeo, inicialmente de forma assíncrona. Após validar essa abordagem, os testes foram replicados de maneira síncrona.

Antes desses experimentos específicos, foram realizados testes individuais do módulo da câmera. A *Raspberry* foi configurada adequadamente, contemplando a configuração da linguagem de programação, a criação de um ambiente virtual, o *upload* de bibliotecas necessárias e a preparação de todos os recursos essenciais para o funcionamento do algoritmo.

### 3.8 Custo do projeto

A Tabela 2 apresenta uma estimativa do custo para o sistema embarcado integrado à cadeira de rodas. Foi considerado apenas o valor gasto com os materiais principais, sem incluir a cadeira de rodas e os suportes previamente instalados.

Tabela 2 – Custos do protótipo.

Componente	Quantidade	Valor unitário	Valor total
Raspberry Pi 3 Model B+	1	R\$ 545,00	R\$545,00
Raspberry Pi Câmera Rev 1.3	1	R\$ 59,24	R\$59,24
Cabo Micro USB 2.0	1	R\$ 19,50	R\$19,50
Mini Protoboard 170 pontos	1	R\$ 5,25	R\$5,25
LED Difuso	2	R\$ 1,89	R\$3,78
Resistor 100 Ohms	3	R\$ 0,15	R\$0,30
<b>TOTAL</b>			<b>R\$ 631,03</b>

## 4 Resultados

Neste capítulo, são apresentados os resultados dos testes e simulações, englobando imagens capturadas em condições ideais e capturas de vídeo, realizadas de forma síncrona e assíncrona. Os resultados foram registrados para demonstrar a execução simultânea dos testes e seus resultados. Para visualizar os registros completos, acesse a playlist no YouTube [aqui](#). Adicionalmente, todos os códigos implementados estão disponíveis no GitHub, [acesse o link](#).

### 4.1 Testes iniciais em capturas de imagens

#### 4.1.1 Histograma

A Figura 32.A representa a imagem de entrada convertida para escala de cinzas e a Figura 32.B ilustra seu histograma. Observa-se uma distribuição desigual dos pixels, logo, a equalização do histograma foi aplicada afim de espalhar as intensidades mais frequentes ao longo de todo o intervalo, resultando em uma imagem com maior riqueza de detalhes visuais, como pode ser visto nas Figuras 32.C e 32.D. No entanto, é importante ressaltar que a equalização do histograma intensificou os ruídos, o que será tratado nas etapas posteriores.

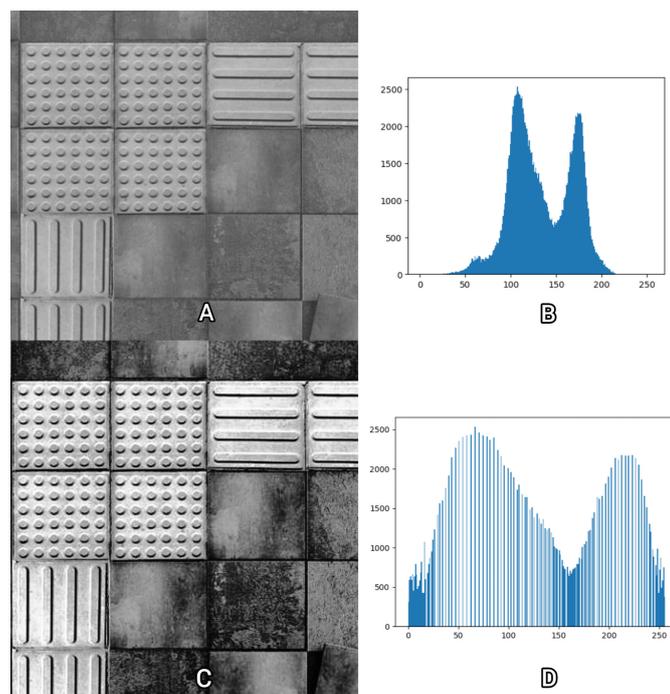


Figura 32 – Equalização do Histograma.

### 4.1.2 Suavização da Imagem

A Figura 33 ilustra as imagens suavizadas com a aplicação do filtro média, filtro gaussiano e filtro mediana a partir de uma kernel de 5x5. Pode-se analisar que a imagem suaviza um pouco, mas a mudança não é drástica, isto por que, um kernel pequeno considera uma área de vizinhança menor ao realizar a operação de suavização. Como resultado, apenas informações localizadas próximas ao pixel atual são consideradas, o que resulta em um efeito de suavização mais sutil.

Em relação a demanda computacional de cada filtro, foi utilizada a função *default timer* fornecida pela biblioteca *timeit* para medir o tempo de execução de cada filtro de forma pontual em um computador. É uma estratégia útil para medir o tempo de forma portátil e precisa, garantindo resultados consistentes em diferentes plataformas. Desta forma, os resultados das medições comprovam que o filtro de mediana requer maior tempo de processamento, como pode ser visto na Tabela 3.

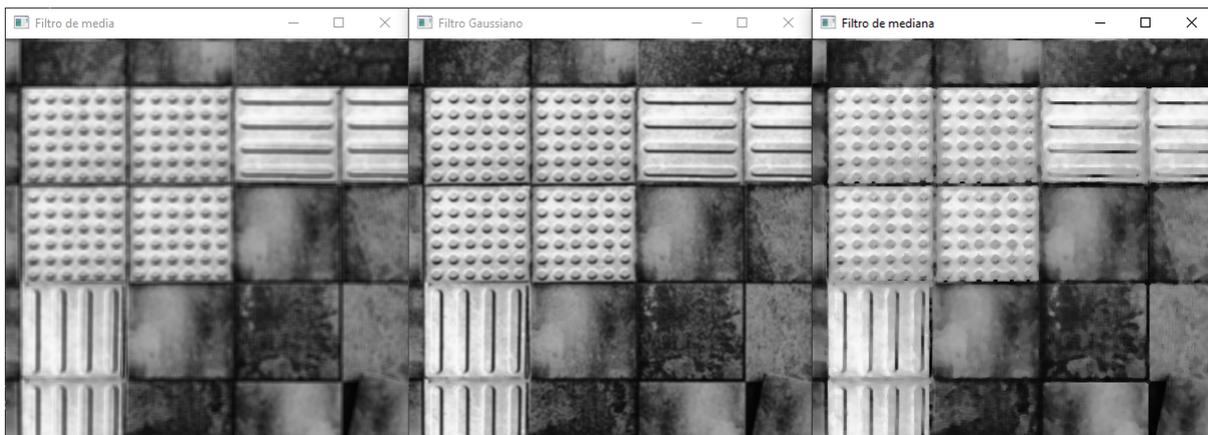


Figura 33 – Suavização da Imagem com aplicação do filtro média, filtro gaussiano e filtro mediana, respectivamente, utilizando um kernel de 5x5

Tabela 3 – Demanda computacional para suavização da imagem com kernel 5x5.

Tempo de processamento ( $\mu s$ ) com kernel 5x5			
Compilação n <sup>o</sup>	Filtro Média	Filtro Gaussiano	Filtro Mediana
1	404,4	623,1	2148,6
2	391,6	614,5	2415,2
3	330,3	633,8	2551,9
4	403,8	668,5	2520,4
5	444,8	612,9	2430,3
6	391,3	553,5	2300,3
7	400,9	584,4	2329,4
8	577,2	573,9	2210,2
9	391,2	615,7	2313,7
10	411,9	577,1	2246,9

Ao aumentar o tamanho do kernel para 15x15, uma área de vizinhança maior é considerada durante a operação de suavização. Isso leva a uma suavização mais forte, maiores perdas de detalhes e bordas mais nítidas, como pode ser visto na Figura 34. O desfoque gaussiano não pesa cada pixel igualmente, quanto mais próximo um pixel estiver do centro, mais ele afetará a média ponderada usada para calcular o novo valor do pixel central, isso resulta em uma suavização mais delicada que as demais. Em relação a demanda computacional (Tabela 4), o aumento no tamanho do kernel influencia principalmente na aplicação do filtro mediana, devido à natureza da operação envolvida, pois a ordenação de uma janela maior exige mais operações de comparação e troca de elementos.

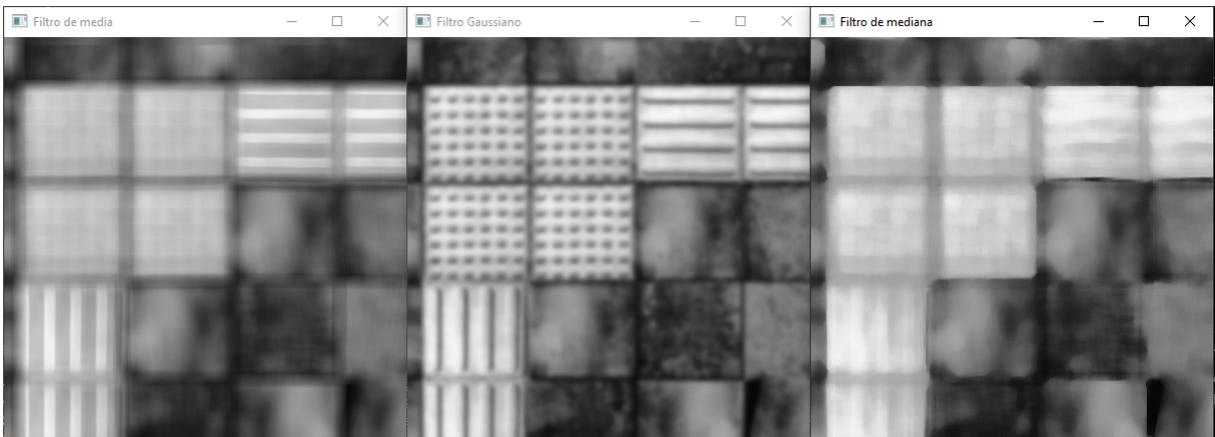


Figura 34 – Suavização da Imagem com aplicação do filtro média, filtro gaussiano e filtro mediana, respectivamente.

Tabela 4 – Demanda computacional para suavização da imagem com kernel 15x15.

Tempo de processamento ( $\mu s$ ) com kernel 15x15			
Compilação nº	Filtro Média	Filtro Gaussiano	Filtro Mediana
1	344,9	659,7	8942,9
2	340,5	699,8	9128,8
3	628,6	898,9	10787,9
4	349,5	903,1	9153,9
5	346,6	631,0	9215,8
6	341,7	664,8	9039,3
7	396,3	691,3	9448,2
8	342,4	677,8	9026,1
9	352,5	652,2	9097,3
10	341,5	639,3	9020,6

Outro parâmetro do filtro gaussiano que influencia diretamente na intensidade do desfoque e no desempenho computacional é o  $\sigma$ , que é o desvio padrão que controla a largura da distribuição gaussiana e determina a quantidade de suavização aplicada, quanto maior seu valor, maior será a área de influência do filtro e mais suave será a

imagem resultante. Se o  $\sigma$  for 0, ele será calculado automaticamente a partir do tamanho do kernel. Se não for 0, aumentar o sigma aumentará o desfoque.

## 4.2 Testes em capturas de vídeo assíncrona

Nesta etapa, a imagem foi convertida para o espaço de cores HSV para facilitar a segmentação de cores específicas, neste caso, amarelo e vermelho. Uma máscara foi criada para filtrar os pixels que estão dentro do intervalo de cores desejado, de acordo com os valores limiares para a matiz (H), saturação (S) e valor (V). Na imagem binária resultante, os pixels brancos correspondem aos pixels dentro do intervalo de cores amarelo e vermelho, enquanto os pixels pretos representam as demais cores, como pode ser visto na Figura 35.

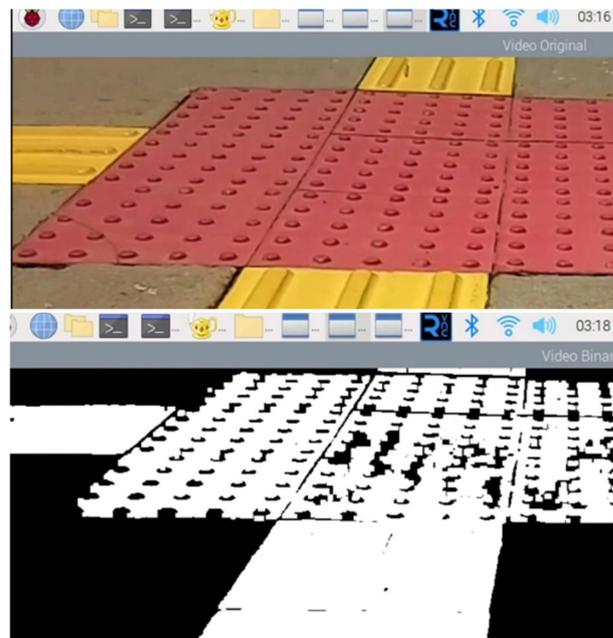


Figura 35 – Imagem original acima e imagem binária resultante abaixo.

Para aprimorar a qualidade da máscara e remover ruídos, foi aplicada uma operação de abertura morfológica. Essa operação contribui para suavizar contornos e eliminar pequenos detalhes indesejados na máscara.

Por fim, realizou-se a contagem de pixels brancos como região de interesse. Desta forma, é possível distinguir entre o piso direcional e piso de alerta, dependendo da quantidade de pixels brancos identificados na imagem. O resultado pode ser visto na Figura 36.

Em resumo, esse processo combinou a segmentação de cores no espaço HSV, o uso de uma máscara binária para isolar a região de interesse e a aplicação de operações

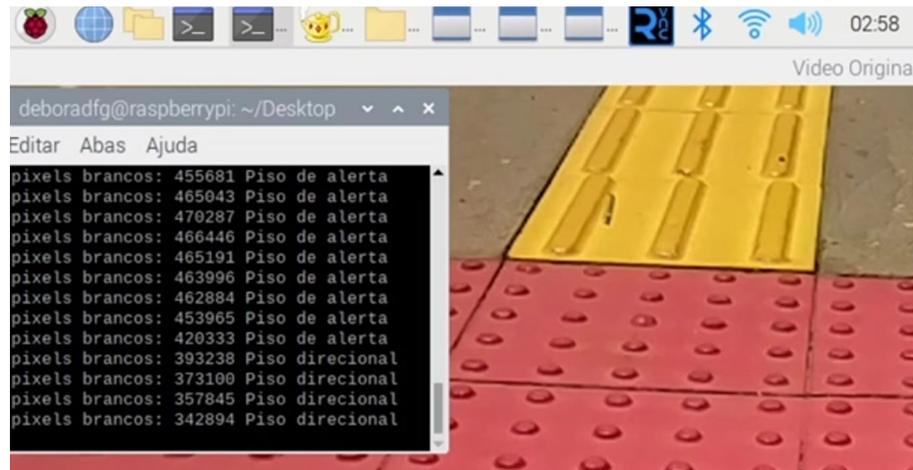


Figura 36 – Teste com captura de vídeo assíncrona em pisos táteis amarelos e vermelhos.

morfológicas para aprimorar a qualidade da máscara, permitindo uma análise eficaz para imagens que contém mais de uma cor de interesse no processamento.

### 4.3 Testes em tempo real

Os testes finais foram conduzidos em tempo real em um ambiente interno, proporcionando condições controladas. Isso permitiu a manipulação de variáveis ambientais, como a iluminação, enquanto se minimizavam possíveis interferências. Nesse cenário, o piso tátil apresentava uma coloração totalmente preta, o que viabilizou a execução de todas as etapas descritas no Capítulo 3.

A Figura 37 ilustra os resultados obtidos. O algoritmo é capaz de detectar e diferenciar os pisos táteis em curtos períodos de tempo.

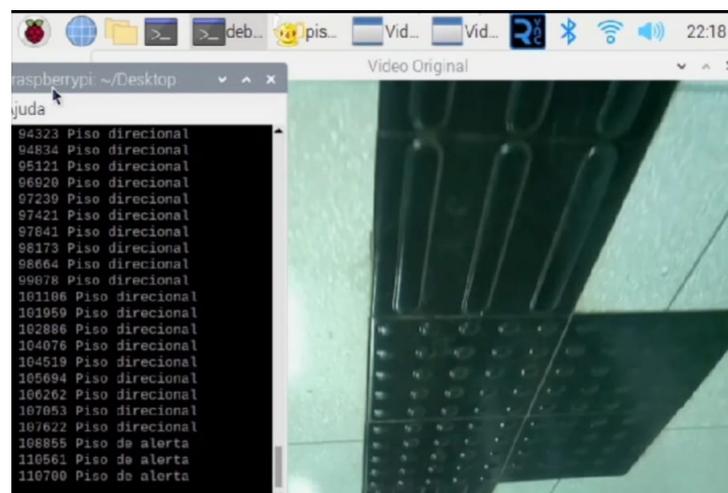


Figura 37 – Resultado em tempo real em pisos táteis pretos.

O tempo médio de processamento por frame, calculado com base na aquisição e

processamento de uma imagem em uma média de 10 frames, é de 75 milissegundos (ms), como pode ser visto na Figura 38. Este indicador é crucial para avaliar a adequação de outra plataforma computacional para a aplicação em questão.

```
deboradfg@raspberrypi:~/Desktop $ python temp_process.py
Média do tempo de processamento para 10 frames: 0.07490795840003557 segundos
```

Figura 38 – Tempo médio de processamento de um frame.

## 4.4 Representação Visual

Para proporcionar uma representação visual ao usuário sobre qual tipo de piso está à frente da cadeira de rodas, foram utilizados GPIOs (*General Purpose Input/Output*) conectados a LEDs. Um LED verde foi associado aos pisos direcionais, indicando a direção segura a ser seguida, enquanto um LED vermelho foi vinculado aos pisos de alerta, alertando sobre possíveis obstáculos ou situações especiais, essa representação pode ser observada na Figura 39. Essa abordagem visa fornecer uma indicação clara e intuitiva ao usuário, facilitando a navegação em ambientes com pisos táteis variados.



Figura 39 – Representação do tipos de piso por LEDs. À esquerda, tem-se a representação de um piso direcional, indicado pelo LED verde, enquanto à direita, tem-se a representação de um piso de alerta, identificado pelo LED vermelho.

## 5 Considerações finais

Os resultados obtidos nos testes indicam que é possível projetar um sistema embarcado para navegação de uma cadeira de rodas por reconhecimento de pisos táteis, em uma solução de baixo custo computacional. O sistema foi capaz de reconhecer e detectar as mudanças dos tipos de pisos táteis e retornar o resultado ao usuário para futuras decisões de movimento da cadeira de rodas. Dessa forma, abre-se a possibilidade de planejar rotas para uma navegação autônoma.

Em todas as etapas do projeto, foram feitas escolhas criteriosas das técnicas a serem empregadas com o melhor *tradeoff* entre desempenho e tempo de execução, de modo que, o algoritmo seja eficiente o suficiente para ser executado em tempo real, mas sem comprometer a precisão e a qualidade do reconhecimento dos pisos táteis. A avaliação do desempenho em tempo real foi fundamental para validar a eficácia e confiabilidade do sistema, revelando desafios que foram enfrentados e superados durante o desenvolvimento, permitindo ajustes e otimizações para melhorar a precisão e robustez do sistema.

Recomenda-se como trabalhos futuros, a integração do sistema de reconhecimento de pisos táteis ao sistema de controle da cadeira de rodas, o desenvolvimento de um sistema de estimação de localização de uma cadeira de rodas usando pisos táteis e odometria e o desenvolvimento de um algoritmo de navegação baseado em mapas de Voronoi e algoritmo A\* usando os pisos de alerta como nós da configuração de espaço livre.

Em conclusão, o sistema se mostrou confiável e eficiente para atender às necessidades práticas dos usuários, contribuindo para a inclusão e melhorando a qualidade de vida de pessoas com deficiências. Além disso, o projeto oferece uma contribuição significativa para o avanço da pesquisa em visão computacional e tecnologias assistivas.



# Referências

ABNT. *NBR 9050: Acessibilidade a edificações, mobiliário, espaços e equipamentos urbanos*. Rio de Janeiro: ABNT, 2015. Citado na página 23.

ABNT. *ABNT NBR 16537: Acessibilidade — Sinalização tátil no piso — Diretrizes para elaboração de projetos e instalação*. 2016. ABNT, Rio de Janeiro. Citado 2 vezes nas páginas 23 e 54.

AKTAŞ, A.; DEMİR, ; DOĞAN, B. Derin öğrenme yöntemleri ile dokunsal parke yüzeyi tespiti. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, Journal of the Faculty of Engineering and Architecture of Gazi University, v. 35, n. 3, p. 1685–1700, 2020. Disponível em: <<http://dx.doi.org/10.17341/gazimmfd.652101>>. Citado na página 51.

BEUCHER, S.; MEYER, F. Image analysis and mathematical morphology - vol. 1. In: \_\_\_\_\_. London: Academic Press, 1982. cap. 12, p. 433–481. Citado na página 32.

BRANCO, M. D. C. C. TCC, *Instrumentação de cadeira de rodas motorizada para usuários com tetraplegia*. Brasília, DF: [s.n.], 2018. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Eletrônica). Citado 3 vezes nas páginas 12, 55 e 57.

EINLOFT, D. C.; GHILARDI, M. C.; MANSSOUR, I. H. Automatic detection of tactile paving surfaces in indoor environments. PUCRS, p. 1–4, 2016. Citado 2 vezes nas páginas 50 e 51.

GONZALEZ, R. C.; WOODS, R. E. *Processamento Digital de Imagens*. 3. ed. São Paulo: Pearson, 2010. Citado 27 vezes nas páginas 11, 12, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47 e 48.

Instituto Brasileiro de Geografia e Estatística (IBGE). *Pesquisa Nacional de Saúde (PNS)*. 2019. Realizada pelo Ministério da Saúde. Disponível em: <<https://www.pns.iciet.fiocruz.br/wp-content/uploads/2021/12/liv101846.pdf>>. Citado na página 19.

Instituto Brasileiro de Geografia e Estatística (IBGE). *Pesquisa Nacional por Amostra de Domicílios (PNAD): Pessoas com Deficiência 2022*. 2022. Citado na página 19.

KASSIM, A. B. M. et al. Vision-based tactile paving detection method in navigation systems for visually impaired persons. In: *Advances in Human and Machine Navigation Systems*. [S.l.]: IntechOpen, 2018. cap. 3, p. 18. Citado na página 51.

LTD, R. P. *Raspberry Pi 3 Model B+*. <<https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/>>. Acessado em 30 de setembro de 2023. Citado 2 vezes nas páginas 12 e 53.

LTD, R. P. *Raspberry Pi Documentation*. <<https://www.raspberrypi.com/documentation/accessories/camera.html>>. Acessado em 30 de fevereiro de 2023. Citado 4 vezes nas páginas 12, 55, 56 e 57.

- MARTINS, S. B. *Introdução ao Processamento Digital de Imagens: Parte 1 - Definições Básicas, Espaço de Cores e Histogramas*. Campinas, SP: [s.n.], 2019. Citado 2 vezes nas páginas 11 e 27.
- OLIVEIRA, S.; SOLEDADE, A. *Acessibilidade: o piso tátil*. 2016. Acesso em 04 de outubro de 2023. Disponível em: <<https://oprofessorweb.wordpress.com/2016/09/16/acessibilidade-o-piso-tatil/>>. Citado 2 vezes nas páginas 11 e 24.
- OpenCV Team. *OpenCV*. 2023. Copyright © 2023, OpenCV Team. Acesso em 04 de julho de 2023. Disponível em: <<https://opencv.org/>>. Citado na página 50.
- PAL, N. R.; PAL, S. K. A review on image segmentation techniques. *Pattern Recognition*, v. 26, n. 9, p. 1277–1294, 1993. Citado na página 33.
- RACHID, M. *Acessibilidade: Composição da sinalização tátil de alerta e direcional*. 2013. Citado 2 vezes nas páginas 12 e 55.
- SANTOS, M. K. et al. Artificial intelligence, machine learning, computer-aided diagnosis, and radiomics: advances in imaging towards precision medicine. *Radiologia Brasileira*, 2019. Disponível em: <<https://doi.org/10.1590/0100-3984.2019.0049>>. Citado na página 49.
- SANTOS, V. T. *Segmentação de imagens mamográficas para detecção de nódulos em mamas densas*. Tese (Tese de Doutorado), 2002. Citado na página 32.
- SIMÕES, M. *Cálculo: Matemática para todos*. v. 16, p. 56, maio 2012. Citado 2 vezes nas páginas 11 e 27.
- Subsecretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência, Comitê de Ajudas Técnicas. *Tecnologia Assistiva*. Brasília: CORDE, 2009. 138 p. Citado na página 20.
- VIEIRA, L. V. C. *Navegação autônoma de cadeira de rodas motorizada em ambiente previamente conhecido*. Dissertação (TCC) — Universidade de Caxias do Sul, 12 2019. Citado na página 51.
- World Health Organization (WHO), The World Bank. *Relatório mundial sobre a deficiência*. São Paulo: SEDPcD, 2012. 334 p. Citado na página 19.