

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Ricardo Silveira Rodrigues

**DETECÇÃO DE BURACOS EM ESTRADAS: UMA
ABORDAGEM AUTOMATIZADA BASEADA NA
TRANSFORMADA WAVELET DE HAAR**

Santa Maria, RS
2019

Ricardo Silveira Rodrigues

**DETECÇÃO DE BURACOS EM ESTRADAS: UMA ABORDAGEM
AUTOMATIZADA BASEADA NA TRANSFORMADA WAVELET DE HAAR**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação (PGCC) da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do título de **Mestre em Ciência da Computação**.

Orientadora: Prof. Dr^a. Marcia Pasin

Co-orientadora: Prof. Dr^a. Alice de Jesus Kozakevicius

Santa Maria, RS

2019

Silveira Rodrigues, Ricardo

Detecção de buracos em estradas: uma abordagem automatizada baseada na Transformada Wavelet de Haar / por Ricardo Silveira Rodrigues. – 2019.

71 f.: il.; 30 cm.

Orientadora: Marcia Pasin

Co-orientadora: Alice de Jesus Kozakevicius

Dissertação (Mestrado) - Universidade Federal de Santa Maria, Centro de Tecnologia, Pós-Graduação em Ciência da Computação, RS, 2019.

1. Buracos nas rodovias. 2. Detecção de automática de buracos. 3. Vibração. 4. Acelerômetro. 5. Wavelets. I. Pasin, Marcia. II. Kozakevicius, Alice de Jesus. III. Título.

© 2019

Todos os direitos autorais reservados a Ricardo Silveira Rodrigues. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

E-mail: rrodrigues@inf.ufsm.br

Ricardo Silveira Rodrigues

**DETECÇÃO DE BURACOS EM ESTRADAS: UMA ABORDAGEM
AUTOMATIZADA BASEADA NA TRANSFORMADA WAVELET DE HAAR**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação (PGCC) da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do título de **Mestre em Ciência da Computação**.

Aprovado em 26 de Junho de 2019:

Marcia Pasin, Dr^a. (UFSM)
(Presidente/Orientadora)

Renata Hax Sander Reiser, Dr^a. (UFPel)

Benhur de Oliveira Stein, Dr. (UFSM)

Santa Maria, RS

2019

AGRADECIMENTOS

Agradeço à minha namorada Isadora, que esteve ao meu lado durante esta minha jornada. Obrigado por sempre acreditar em mim, por todo apoio, carinho e compreensão. Obrigado por fazer esses dois anos de dedicação muito melhores por estar ao meu lado.

Agradeço à minha família, especialmente aos meus pais, Nilsa e Darlan, que sempre me incentivaram, me apoiaram em todas as decisões da minha vida e me deram o suporte necessário para chegar até aqui.

Agradeço aos meus amigos por me apoiarem e estarem presentes durante esse período.

Por fim, agradeço às minhas orientadoras. À Marcia, por esses três anos de orientação, por tornar esse trabalho possível e sempre estar disposta a enfrentar os desafios que se apresentaram. À Alice, que também aceitou encarar este desafio já em andamento como co-orientadora e sempre acreditou no trabalho que estava sendo feito.

RESUMO

DETECÇÃO DE BURACOS EM ESTRADAS: UMA ABORDAGEM AUTOMATIZADA BASEADA NA TRANSFORMADA WAVELET DE HAAR

AUTOR: RICARDO SILVEIRA RODRIGUES

ORIENTADORA: MARCIA PASIN

CO-ORIENTADORA: ALICE DE JESUS KOZAKEVICIUS

Buracos em rodovias e estradas causam diversos tipos de transtornos aos usuários como desconforto dos passageiros, dano aos componentes dos veículos e podem causar acidentes. No Brasil, o último relatório da Confederação Nacional de Transportes (CNT), de 2018, apontou que 50% dos 107.000 quilômetros de rodovias asfaltadas analisados no país foram classificadas como regular, ruim ou péssima. Essa classificação é feita manualmente e é sujeita a erros ou aberta a interpretações atreladas aos avaliadores. Infelizmente, esta situação não é muito diferente em outros países. Na Índia, os buracos são grandes causadores de graves acidentes de trânsito. Até os motoristas da Inglaterra sentem-se afetados pelos buracos nas estradas. Este trabalho propõe um sistema automatizado para detecção de buracos nas estradas usando uma abordagem wavelet em sinais de acelerômetro. O uso de acelerômetros para a detecção de buracos não é novidade. Tipicamente, soluções que usam acelerômetros são baseadas em limiares, que indicam a existência de buracos, e são manualmente calibrados. A desvantagem da calibração manual é que ela é dependente de um observador expert para realizar um ajuste preciso. Como diferencial, neste trabalho, a abordagem wavelet permite computar esses limiares de forma automática, sem usar calibração manual. Foram realizados experimentos em ambientes controlado e real, que confirmam a eficiência da solução proposta. Este é um importante passo na busca de um processo de plena automatização para a detecção de buracos nas rodovias.

Palavras-chave: Buracos nas rodovias. Detecção de automática de buracos. Vibração. Acelerômetro. Wavelets.

ABSTRACT

DETECTION OF ROAD POTHOLE: AN AUTOMATED APPROACH BASED ON HAAR WAVELET TRANSFORM

AUTHOR: RICARDO SILVEIRA RODRIGUES

ADVISOR: MARCIA PASIN

COADVISOR: ALICE DE JESUS KOZAKEVICIUS

Potholes in highways and roads cause various types of inconvenience to users such as passenger discomfort, damage to vehicle components and can cause accidents. In Brazil, the latest report by the National Transportation Confederation (CNT) in 2018 found that 50% of the 107,000 kilometers of paved highways surveyed in the country were classified as regular, bad or bad. Thus, general condition of roads is not good. Actually, this classification is done manually and is subject to errors or open to interpretations linked to the evaluators. Unfortunately, this situation is not very different in other countries. In India, the potholes are great cause of serious traffic accidents. Even the drivers of England are affected by the potholes in the roads. Thus, this work proposes an automated system for pothole detection using a wavelet approach in accelerometer signals. The use of accelerometers for pothole detection is not new. Typically, solutions that use accelerometers are based on thresholds, which indicate the existence of potholes, and are manually calibrated. The disadvantage of manual calibration is that it is dependent on an expert observer to perform an accurate adjustment. As a differential, in this work, the wavelet approach allows computing these thresholds automatically, without using manual calibration. Experiments were carried out in controlled and real environments, which confirm the efficiency of the proposed solution. This is an important step in the search for a full automation process for pothole detection in roads and highways.

Keywords: Automatic pothole detection. Vibration. Accelerometer. Wavelets.

LISTA DE FIGURAS

Figura 1 –	Representação de um acelerômetro de 3-eixos.....	21
Figura 2 –	Algoritmos para detecção de buracos propostos por (MEDNIS et al., 2011)..	23
Figura 3 –	Processo de detecção de anomalias em estradas. Adaptado de (SATTAR; LI; CHAPMAN, 2018).	26
Figura 4 –	Decomposição do sinal de entrada $x[n]$ em 3 níveis usando o algoritmo Cascata.	30
Figura 5 –	Decomposição do sinal de entrada $x[n]$ em 3 níveis usando o algoritmo à trous.	32
Figura 6 –	Arquitetura do sistema proposto para a detecção de buracos em estradas.	36
Figura 7 –	Dispositivo com acelerômetro construído para a coleta de dados.	37
Figura 8 –	Posicionamento do <i>smartphone</i> em relação aos eixos x, y e z do acelerômetro	38
Figura 9 –	Sinal original C_0	40
Figura 10 –	Médias C_1 e detalhes D_1 resultantes do primeiro nível de decomposição da TWH usando o algoritmo cascata.....	40
Figura 11 –	Três níveis de detalhes usando a THW algoritmo cascata.	41
Figura 12 –	Médias C_1 e detalhes D_1 resultantes do primeiro nível de decomposição da TWH usando o algoritmo à trous.	42
Figura 13 –	Cinco níveis de médias usando a THW algoritmo à trous.	42
Figura 14 –	Algoritmo à trous de 2 passos.....	43
Figura 15 –	Curva da distribuição normal. Fonte (ACTION, 2018).	45
Figura 16 –	Cenário de um conjunto de vias com buracos e indicação de buracos.	46
Figura 17 –	Matriz de confusão.	49
Figura 18 –	Foto do carro-robô Arduino usado no experimento com a pista sintética.	52
Figura 19 –	Foto do carro-robô Arduino sobre um segmento da pista sintética contendo um buraco.....	52
Figura 20 –	<i>Thresholds</i> λ_{sup} e λ_{inf} aplicados aos detalhes D_1 da volta 1 usando a TWH algoritmo cascata	54
Figura 21 –	Sinais complementares de 20 voltas após o truncamento dos sinais D_1 no cenário com pista sintética.	54
Figura 22 –	Mapa da UFSM com indicação da localização do viaduto.....	56
Figura 23 –	Visão geral do viaduto interno da UFSM.....	57
Figura 24 –	Esquema do cenário real onde os experimentos foram conduzidos com a indicação da localização do buraco.	57
Figura 25 –	Buraco no asfalto no viaduto da UFSM e detectado por λ_{sup} e λ_{inf} durante o experimento neste cenário.	58
Figura 26 –	Sinais de detalhes D_1 e complementar D_1^c da volta 25 obtidos pelo algoritmo cascata.	58
Figura 27 –	Sinal original C_0 , médias C_5 e complementar C_5^c da volta 25 obtidos pelo algoritmo à trous.	59
Figura 28 –	Sinal resultante C_5^1 e complementar C_5^{1c} da volta 25 obtidos pelo algoritmo à trous passo 2.....	59
Figura 29 –	Gráfico de verdadeiros positivos (VP) dos resultados obtidos no experimento em ambiente real.....	61
Figura 30 –	Gráfico de verdadeiros negativos (VN) dos resultados obtidos no experimento em ambiente real.....	61

Figura 31 – Gráfico de falsos positivos (FP) dos resultados obtidos no experimento em ambiente real.....	62
Figura 32 – Gráfico de falsos negativos (FN) dos resultados obtidos no experimento em ambiente real.....	62

LISTA DE TABELAS

Tabela 1 –	Valores relativos à pista usada nos experimentos no cenário controlado.	51
Tabela 2 –	Valores de λ_{sup} e λ_{inf} para as 20 voltas percorridas.....	55
Tabela 3 –	Valores relativos aos elementos do cenário real.	56
Tabela 4 –	Gabarito dos experimentos.	60
Tabela 5 –	Resultados para os experimentos no cenário real.	63

LISTA DE ABREVIATURAS E SIGLAS

AIAS	Sistemas de Análise de Imagem Automatizada
ANN	<i>Artificial Neural Network</i>
CCD	<i>Charge-Coupled Device</i>
CNT	Confederação Nacional dos Transportes
CSV	<i>Comma-Separated Values</i>
DTW	<i>Dynamic Time Warping</i>
EVA	<i>Ethylene-Vinyl Acetate</i> ou Acetato-Vinilo de Etileno
GPS	<i>Global Positioning System</i>
HOG	Histograma de Gradiente Orientado
LiDaR	<i>Light Detection And Ranging</i>
SVM	<i>Support Vector Machine</i>
2D	<i>Two-dimensional</i>
3D	<i>Three-dimensional</i>
FN	Falso Negativo
FP	Falso Positivo
TF	Transformada de Fourier
TWH	Transformada Wavelet de Haar
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo

SUMÁRIO

1	INTRODUÇÃO	12
1.1	MOTIVAÇÃO	12
1.2	OBJETIVOS	15
1.3	ESTRUTURA DO TEXTO	16
2	FUNDAMENTAÇÃO	17
2.1	TÉCNICAS DE DETECÇÃO DE BURACOS	17
2.1.1	Processamento de imagens	17
2.1.2	Reconstrução 3D	18
2.1.3	Processamento de sinais de sensores	20
2.2	DESAFIOS DA DETECÇÃO DE BURACOS USANDO SENSORES	27
2.3	TRANSFORMADA WAVELET DE HAAR	28
2.3.1	Algoritmo cascata	30
2.3.2	Algoritmo à trous	32
2.3.3	Operação de Truncamento (Threshold)	33
2.3.4	Conclusão do Capítulo	34
3	METODOLOGIA	35
3.1	COLETA DE DADOS	36
3.2	PROCESSAMENTO	38
3.2.1	Análise dos dados via transformadas wavelet	39
<i>3.2.1.1</i>	<i>Algoritmo cascata</i>	39
<i>3.2.1.2</i>	<i>Algoritmo à trous</i>	41
<i>3.2.1.3</i>	<i>Algoritmo à trous e remoção do sinal médio</i>	43
3.2.2	Thresholding	44
3.3	LOCALIZAÇÃO	46
3.4	CLASSIFICAÇÃO	48
3.5	AVALIAÇÃO DOS RESULTADOS	48
4	EXPERIMENTOS	50
4.1	EXPERIMENTO EM AMBIENTE CONTROLADO	50
4.1.1	Configuração do experimento	50
4.1.2	Processamento e resultados	52
4.1.3	Conclusões parciais	55
4.2	EXPERIMENTO EM CENÁRIO REAL	55
4.2.1	Configuração do experimento	56
4.2.2	Coleta de dados	56
4.2.3	Processamento e resultados	58
4.2.4	Avaliação dos resultados	61
4.2.5	Limitações do experimento	63
5	CONSIDERAÇÕES FINAIS	64
5.1	CONCLUSÕES	64
5.2	TRABALHOS FUTUROS	65
5.3	PUBLICAÇÕES.....	65
	REFERÊNCIAS	67

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Buracos em rodovias e estradas causam diversos tipos de transtornos aos usuários como desconforto dos passageiros, danos aos componentes dos veículos como suspensões, pneus e rodas, além do potencial de provocar acidentes que podem ferir pessoas ou até mesmo serem fatais, como indicam as recentes estatísticas oficiais da União Europeia (COMMISSION, 2018). A presença de buracos em rodovias é um problema comum em muitos países e tem sido um tópico relevante de discussão no Parlamento Europeu (GLEAVE et al., 2014). De acordo com o GESTRATA¹, as estatísticas de 2010 indicaram que a Áustria, apesar de ter as melhores condições de rede rodoviária de alto nível de todos os países europeus, ainda tem, em algumas regiões, apenas 25 % da superfície das estradas classificada como muito bom.

No Brasil, o último relatório da Confederação Nacional de Transportes (CNT) (TRANSPORTES, 2018), de 2018, apontou que 50% dos 107.00 quilômetros de rodovias asfaltadas analisadas no país foram classificadas como regular, ruim ou péssima. Atualmente, o relatório da CNT é feito manualmente com o esforço de um time de pessoas, que é um processo custoso do ponto de vista financeiro, além de ser aberto a subjetividade da interpretação dos avaliadores.

Com o intuito de ser um passo adiante à metodologia manual, este trabalho propõe um sistema automatizado para detecção de buracos das estradas aplicando uma abordagem wavelet em sinais gerados via acelerômetro. O propósito de um sistema de detecção de buracos é automatizar e reduzir tarefas realizadas por humanos, além de auxiliar motoristas na tomada de decisão e também os órgãos responsáveis pelo reparo da pavimentação.

O acelerômetro é um dispositivo que indica a aceleração e também detecta e mede vibrações. É um instrumento amplamente disponível nos *smartphones* e serve para, por exemplo, identificar a orientação do *smartphone* e rotacionar a tela automaticamente. O sinal fornecido pelo acelerômetro pode ser usado para detectar a existência de buracos nas rodovias. Esta ideia já vem sendo amplamente explorada por vários autores (ALLOUCH et al., 2017) e (SOUZA VINÍCIUS GIUSTI, 2018) e será considerada também na metodologia proposta.

Tipicamente, soluções que usam acelerômetros são dependentes da escolha de limiares, ou *thresholds*, que indicam a existência de buracos, e são manualmente calibrados. Assim,

¹ <http://www.gestrata.at/aktuelles>

se os valores medidos pelo acelerômetro (sinal) ultrapassarem um determinado *threshold*, essa variação é classificada como buraco. A desvantagem da calibração manual é que ela é dependente de um observador *expert* para realizar o ajuste do valor desse *threshold* e está sujeito a variabilidades subjetivas.

Neste trabalho, propõe-se uma alternativa para a detecção de buracos utilizando-se a transformada wavelet de Haar aplicada aos dados do acelerômetro. Após a decomposição do sinal captado pela transformada, dados referentes à variabilidade do sinal original ficam evidenciados em uma das componentes (coeficientes wavelets) resultantes da transformação. Então, um *threshold* é calculado a partir desta componente. Com isso, a influência de um *expert* no processo de tomada de decisão é diminuído. A contribuição deste trabalho está: (i) na maneira com que se calcula o *threshold* e (ii) no tipo de transformada considerada.

Existem vários tipos de funções wavelets e, conseqüentemente, várias possibilidades de transformações associadas a estas funções. Neste trabalho, explora-se a utilização de transformada wavelet de Haar não só em sua formulação mais conhecida, dada pelo algoritmo de Cascata (MALLAT, 1989), mas também em sua formulação não-decimada, dada pelo algoritmo *à trous* (KOZAKEVICIUS; SCHMIDT, 2013). A transformada de Haar pode ser aplicada a dados unidimensionais quanto bidimensionais. Isso permite que a proposta de análise inicialmente feita para sinais de entrada possa futuramente ser estendida para imagens. Essa flexibilidade do algoritmo proposto é relevante, pois com a atenção que a detecção automática de buracos está recebendo, tanto da indústria quanto da academia, técnicas de detecção baseadas em imagens começam a ser cada vez mais investigadas.

Na academia, existem atualmente três principais abordagens usadas para detecção de buracos (ENIGOA et al., 2016): baseadas em visão, baseadas em vibração e as que consideram reconstrução 3D. Para este trabalho, a abordagem baseada em vibração será estendida em abordagem baseada em sensores, que engloba também, sensores não baseados em vibração como o sensor ultrassônico. As abordagens baseadas em visão fazem o processamento de fotos e vídeos de estradas para detectar buracos. As abordagens baseadas em sensores utilizam dispositivos como acelerômetro, giroscópio, sensores ultrassônicos, entre outros. Muitos desses dispositivos estão disponíveis em *smartphones* convencionais como o acelerômetro e o giroscópio.

A detecção de buracos usando reconstrução 3D faz uso de sensores ópticos como *Light Detection And Ranging* (LiDaR), Kinect ou imagens stereo. A principal vantagem das abordagens de detecção por visão e reconstrução 3D é a baixa taxa de falsos positivos no reco-

nhecimento de buracos, dependendo do algoritmo de processamento de imagens utilizado. No entanto, essas técnicas necessitam de um grande poder de processamento computacional além de equipamentos externos (como câmeras e sensores ópticos), que ainda não são incluídos na maioria dos veículos, em especial nas versões populares.

A detecção de buracos usando sensores, principalmente o acelerômetro, é feita através da vibração gerada quando um veículo passa por um buraco. Diferentes superfícies ou pavimentações, por exemplo com paralelepípedos ou de chão batido, geram vibrações pelas irregularidades inerentes do tipo da via. Já em estradas asfaltadas, as vibrações captadas são significativamente mais suaves, o que facilita a identificação vibrações mais fortes causadas por anomalias na pavimentação. Assim, na detecção baseada em sensores, a maior dificuldade é determinar se as variações nos dados capturados são causadas por um buraco ou por outras anomalias existentes na superfície. Essa maior sensibilidade com relação aos dados de entrada pode resultar um número maior de falsos positivos em relação às outras técnicas. Por outro lado, os sensores (acelerômetro) são facilmente encontrados em *smartphones* modernos e a abordagem necessita de menor poder computacional (baixo custo) para processar os dados gerados, o que justifica a consideração deste recurso neste trabalho.

Fabricantes de veículos já estão desenvolvendo seus próprios sistemas para detecção de buracos. A Jaguar Land Rover² está desenvolvendo um sistema que detecta buracos usando uma câmera stereo e alerta outros veículos conectados ao mesmo sistema através de uma nuvem computacional. A Ford³ apresentou um sistema de detecção de buracos disponível no modelo de carro Fusion v6. O sistema tem como objetivo antecipar a presença do buraco para que o sistema de suspensão eletrônica do carro se ajuste automaticamente para reduzir o impacto da roda ao cair no buraco. Isso é feito através de sensores localizados logo a frente das rodas dianteiras do veículo. Entretanto, os sistemas desenvolvidos pelas montadoras ainda não estão disponíveis para todos os veículos/motoristas.

Contrastando com soluções proprietárias, uma outra solução que tem ganhado destaque é o sensoriamento participativo ou *crowdsourcing* (BURKE et al., 2006), que pode ser aplicado a detecção de buracos, como em (FOX et al., 2017). Através de sensores instalados nos veículos, a extração de informação é realizada, processada em nuvem e disseminada para outros veículos, com suporte de rede de comunicação. Ao agregar dados de vários veículos, o *crowdsourcing* é

² <https://www.landrover.com/experiences/news/pothole-detection.html>

³ <https://media.ford.com/content/fordmedia/fna/us/en/news/2016/02/18/all-new-ford-fusion-v6-sport-helps-protect-against-potholes.html>

usado para detectar informações ambientais com maior precisão, como a existência de buracos em estradas com várias faixas.

A extração de informações de dados agregados de veículos é ainda um desafio devido aos sensores de pouca amostragem, mobilidade do sensor, operação do sensor assíncrono, ruído do sensor, heterogeneidade do veículo e da estrada e erro de posição do GPS. O erro de posição do GPS é particularmente problemático em ambientes com várias faixas, uma vez que o erro de posição é geralmente maior do que as larguras de pista padrão.

Neste trabalho, além dos dados do acelerômetro, considera-se a tecnologia de GPS (PARKINSON; SPILKER, 1996) (MISRA; ENGE, 2006) e outros sensores utilizados para a localização de buracos. Nos trabalhos relacionados, (SOUZA VINÍCIUS GIUSTI, 2018), (MEDNIS et al., 2011) (ERIKSSON et al., 2008), são considerados sensores similares para detecção de buracos, porém as soluções desenvolvidas foram para cenários específicos, com calibrações dependendo dos cenários e das condições de teste. Neste trabalho, a formulação do cálculo de *threshold* envolvendo apenas os dados analisados faz com que a proposta apresentada seja adaptativa e, portanto, mais robusta para análise de dados independente dos diferentes tipos de superfície.

1.2 OBJETIVOS

O objetivo geral deste trabalho é desenvolver uma metodologia para detecção de buracos e condição de estradas usando uma abordagem wavelet aplicada sobre sinais de acelerômetro. Este trabalho visa suprir um problema comum nas implementações existentes que utilizam sinais de acelerômetro, que é definição manual e estática do valor dos limiares de detecção.

Mais especificamente, este trabalho visa atingir os seguintes objetivos:

- Coletar dados do acelerômetro e de localização posicionados em veículos, tanto em um cenário controlado quanto em condições reais;
- Associar cada amostra captada pelo acelerômetro com uma informação de localização;
- Processar os sinais formatados do acelerômetro usando transformadas wavelet.
- Testar as formulações decimada e não decimada da Transformada de Haar para analisar o sinal;
- Calcular *thresholds* adaptativos para os sinais processados;

- Testar o conceito de dois níveis diferentes de *thresholds*, com o intuito de se distinguir diferentes tipos de buracos;
- Classificar como buraco os valores dos sinais que ficarem acima dos *thresholds* calculados;
- Definir a localização dos buracos pela informação de localização associada aos valores classificados;
- Discutir o conceito de falso positivo para o problema, dado o erro do GPS introduzido na localização dos buracos;
- Validar a solução proposta através de experimentos em ambientes controlado e real.

O uso de wavelets na detecção de buracos é um tópico recente de pesquisa. Em (WANG et al., 2017), transformadas wavelet foram utilizadas no processamento de imagens para identificar buracos. Em contraste, neste trabalho, as transformadas wavelet são aplicadas em sinais (unidimensionais) de acelerômetro e os buracos são detectados por meio de *thresholds* calculados a partir dos coeficientes de detalhe (ou coeficientes wavelets da transformação).

1.3 ESTRUTURA DO TEXTO

Este texto está organizado como segue: O Capítulo 2 descreve a fundamentação teórica, enfocando principalmente as técnicas de detecção de buracos em pavimentos e wavelets, e também apresenta os trabalhos relacionados. O Capítulo 3 descreve o sistema proposto para detecção de buracos em pavimentos e wavelets. O Capítulo 4 apresenta experimentos conduzidos em um cenário controlado, que foi completamente confeccionado para os testes realizados, e em um cenário real, escolhido de tal modo que a localização exata dos buracos reais possam ser dadas com precisão. Finalmente, o Capítulo 5 apresenta as conclusões e sugere trabalhos futuros.

2 FUNDAMENTAÇÃO

Esta Seção apresenta os conceitos teóricos necessários para o entendimento deste trabalho. Inicialmente, é abordada a problemática de detecção de buracos, com ênfase na detecção usando sensores, assunto foco deste trabalho. Depois, são listados os desafios para a detecção de buracos usando sensores. Finalmente, é feita uma breve revisão sobre transformadas wavelet, assunto necessário para o entendimento desta proposta.

2.1 TÉCNICAS DE DETECÇÃO DE BURACOS

Detecção automatizada de buracos é a capacidade um sistema detectar buracos em estradas sem a interferência humana durante o processo. É um tema que obteve muita atenção da academia e também da indústria de automóveis nos últimos anos. Na academia, muitos trabalhos foram publicados a fim de tentar resolver esse problema. Nesse tema é possível dividir os trabalhos em três categorias (ENIGOA et al., 2016): detecção por imagem, reconstrução 3D e detecção por vibração (KIM; RYU, 2014). Esta última categoria é o foco deste trabalho, que será tratada como detecção por sensores, por englobar, além dos sensores de vibração, sensores como o sensor ultrassônico.

2.1.1 Processamento de imagens

Nessa abordagem, a detecção de buracos é realizada fazendo o processamento de imagens ou vídeos com o objetivo de detectar padrões que possam ser classificados como buracos. Esse processamento pode ser feito em tempo real ou após a captação das imagens. (AZHAR et al., 2016) e (LIN; LIU, 2010) são exemplos de trabalhos que utilizaram essa abordagem.

Em (AZHAR et al., 2016), a detecção e localização de buracos do pavimento é realizada usando técnicas de visão computacional. Um dataset contendo 120 imagens de estradas e ruas, com buracos e sem buracos, foi considerado para experimentação. De acordo com a aparência dos buracos, os recursos de Histogramas de Gradientes Orientados (HOG) são calculados para 50 imagens do dataset de entrada. As características são treinadas e classificadas usando Naïve Bayes, resultando na rotulagem da entrada como uma imagem com ou sem buracos. Para localizar o buraco nas imagens, o esquema de segmentação de corte de grafos normalizado é empregado. O esquema proposto foi testado em um dataset com 70 imagens de pavimento.

Os resultados da experimentação apresentaram 90% de acurácia e 86.5% de precisão para a detecção de buracos em imagens.

Em (LIN; LIU, 2010), um algoritmo para reconhecer os buracos do pavimento em imagens é proposto. A medida de textura baseada no HOG é extraída como os recursos da região da imagem, e um classificador Support Vector Machine (SVM) não linear é usado para identificar se uma região-alvo é um buraco. No experimento, 50 imagens foram usadas como treinamento e as outras 30 foram usadas como entrada para o experimento. Os resultados experimentais mostram que o algoritmo pode alcançar uma alta taxa de reconhecimento e que todos os buracos foram identificados nas imagens usadas nos experimentos.

A detecção por imagens possui algumas limitações. A principal é que essa abordagem necessita de um poder de processamento local muito grande para detecções em tempo real, além da necessidade de um dispositivo externo para gravar as imagens. Isso torna inviável sua implementação com as tecnologias atuais presente nos carros.

2.1.2 Reconstrução 3D

A técnica de reconstrução 3D utiliza sensores ópticos para mapear o ambiente a frente ou ao redor do veículo. Depois de pronto o mapeamento, é necessário fazer um processamento semelhante ao processamento de imagens. (KANG; CHOI, 2017), (YU; SALARI, 2011), (VUPPARABOINA et al., 2015) e (WANG et al., 2017) são exemplos de trabalhos que utilizaram técnicas de reconstrução 3D para detectar buracos.

Em (KANG; CHOI, 2017), para melhorar a precisão do sistema de detecção, é proposta a utilização de sensores heterogêneos, combinando LiDAR e câmera 2D. Utilizando o LiDAR 2D, obtém-se a informação de distância e ângulo entre o sensor e a estrada. A câmera 2D obtém imagens da estrada. Mais precisamente, o algoritmo de detecção de buraco inclui pré-processamento de redução de ruído, *clustering*, extração de segmento de linha e gradiente de função de dados do buraco. Em seguida, o método de detecção de buracos baseado em imagem é usado para melhorar a precisão da detecção de buracos e para obter a forma do buraco. O algoritmo que processa imagem inclui as seguintes etapas: filtragem de ruído, controle de brilho, binarização, filtragem de ruído viciante, extração de borda e extração de objetos e detecção de buracos. Para demonstrar a eficiência do sistema de detecção de buracos, experimentos foram realizados. A taxa de erro de detecção foi de até 0.05 cm para os casos analisados. Entretanto, a conexão do sistema usando a visão do LIDAR e a imagem capturada pela câmera não é clara

na bibliografia consultada.

(YU; SALARI, 2011) propõe um sistema de iluminação artificial de alta potência, com a tecnologia de raios laser, para melhorar a qualidade de imagens de buracos. Sistemas de Análise de Imagem Automatizada (AIAS) foram desenvolvidos para análise e gerenciamento de superfícies de pavimento. Câmeras tipicamente usadas em AIAS usam sensores de imagem *Charge-Coupled Device* (CCD), onde um raio visível é projetado. No entanto, a qualidade das imagens capturadas por essas câmeras é limitada. Problemas destacados são iluminação inadequada e sombreamento. Na proposta de (YU; SALARI, 2011), após a captura de imagens, as regiões correspondentes aos buracos são representadas por uma matriz de ladrilhos quadrados e, então, o formato estimado do buraco é determinada. Informações extraídas das imagens são usadas como entrada para uma *Artificial Neural Network* (ANN) de três camadas e são classificadas em buracos e rachaduras (vertical ou horizontal). Para validar o sistema, os autores usaram uma base de dados com 100 imagens de buracos e rachaduras. Os resultados de saída do algoritmo foram comparados com uma classificação manual dessas imagens e foi demonstrado que a classificação foi idêntica. (YU; SALARI, 2011) destacam que o algoritmo demonstrou uma melhor habilidade em classificar áreas escuras, como manchas de óleo, sombras e marcas da faixa, do que outros métodos de classificação de pavimentação 2D existentes.

Muitas vezes os buracos podem estar cobertos por água, dificultando a sua detecção ou impedindo que o condutor tenha noção da profundidade do buraco. (VUPPARABOINA et al., 2015) propõem uma estrutura geométrica baseada na física para detectar buracos secos e cobertos por água usando uma câmera e um dispositivo a *laser*. A câmera e o emissor *laser* são posicionados na frente do veículos, alinhados verticalmente. A detecção dos buracos é feita pela relação geométrica entre a posição do ponto projetado pelo *laser* captado pela câmera e as posições da câmera e do emissor *laser* em si. Em caso de buracos, a posição do ponto projetado pelo *laser* muda de lugar na imagem capturada e então é calculada a profundidade do buraco. Quando estão cobertos por água, a relação geométrica do ponto muda devido a refração causada pela água, assim outra relação geométrica é necessária para calcular a profundidade do buraco. O experimento foi validado usando um carro-robô equipado com uma câmera e dispositivo *laser* acoplado. O carro-robô passou uma pista de metal com buracos secos e buracos cobertos por água. Os resultados mostraram que em buracos secos o erro medido entre a profundidade real e profundidade calculada pelo sistema variou de 1 a 1.33%. Enquanto em buracos cobertos por água o erro variou de 5.33% a 17%.

(WANG et al., 2017) apresenta um método de detecção de buraco em asfalto baseado em campo de energia (*energy field*). O método proposto é dividido em dois processos: construção do campo de energia e processamento. Primeiramente, o campo de energia wavelet da imagem do pavimento é construído para detectar o buraco por processamento morfológico e critérios geométricos. Depois, o buraco detectado é segmentado pelo modelo de campo aleatório de Markov e a borda do buraco é extraída com precisão. A metodologia foi implementada através de um protótipo em MATLAB, treinada e testada com 120 imagens de pavimento. Os resultados mostram que a metodologia distingue buracos de rachaduras, manchas, sujeira oleosa, sombras e tampas de bueiros e segmenta com precisão o buraco. Para a detecção de buracos, o método atinge uma precisão geral de 86,7%. Para a segmentação de buracos, o grau de sobreposição entre a região do buraco extraído e a região do buraco original é maior que 85%, o que representa 88,6% do total das imagens detectadas do buraco do pavimento.

Assim como na detecção por imagem, o alto poder de processamento necessário para usar essa abordagem em tempo real é um fato limitante para o uso dessa técnica. Além disso, o custo dos dispositivos necessários para reconstrução 3D são significativamente maiores do que nas outras abordagens.

2.1.3 Processamento de sinais de sensores

Nessa abordagem é feito o processamento dos sinais produzidos por sensores como o acelerômetro, o giroscópio e o sensor ultrassônico (MADLI et al., 2015) para realizar a detecção de buracos. Neste trabalho o sensor utilizado foi o acelerômetro.

O acelerômetro é um dispositivo que mede a aceleração, que é a taxa de variação da velocidade de um determinado objeto. Um buraco causa uma variação significativa desta medida uma vez que a roda do veículo “cai” dentro dele. A aceleração é medida em metros por segundo ao quadrado (m/s^2) ou em força-g (g). Um objeto em inércia sobre uma superfície na Terra estará sempre sofrendo uma aceleração vertical de aproximadamente $9,8 m/s^2$, que é a força da gravidade, também representada por 1 g.

A Figura 1 esboça o esquema de um acelerômetro de três eixos. Basicamente, o acelerômetro de três eixos mede a aceleração aplicada sobre os eixos x , y e z , e oferece essas medidas como valores de saída.

O método de detecção de buracos usando o acelerômetro baseia-se na instalação do sensor em um lugar do veículo com um dos eixos posicionado de forma perpendicular à superfície.

Quando o veículo passar pelo buraco, a queda da roda gera uma aceleração sobre esse eixo, modificando as medidas obtidas. Existem diversos algoritmos para determinar se o veículo passou por um buraco utilizando essas medições. Alguns exemplos foram descritos no capítulo 2.

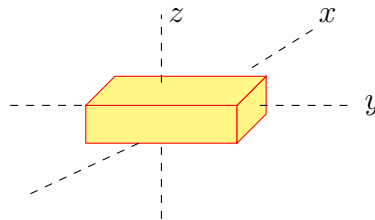


Figura 1 – Representação de um acelerômetro de 3-eixos.

O giroscópio é um dispositivo que mede a orientação de um objeto. Geralmente é utilizado em conjunto com o acelerômetro. O giroscópio indica mudanças de inclinação nos eixos x , y e z quando o veículo passa por um buraco.

O sensor ultrassônico é um dispositivo capaz de detectar e calcular a distância de objetos a sua frente. O dispositivo possui um transmissor e um receptor de ondas de som de alta frequência. O transmissor emite um sinal de alta frequência que é refletido após colidir com algum objeto e recebido de volta pelo receptor do sensor. Conhecendo a velocidade de transmissão desse sinal, é possível calcular a distância que o objeto está do sensor pelo tempo que o sinal levou do instante em que ele foi emitido até o instante que ele foi recebido.

Em (MADLI et al., 2015), o sensor ultrassônico foi utilizado para detecção de buracos em rodovias. Posicionado embaixo do veículo, com o transmissor e receptor apontados para o chão, o sensor ultrassônico mede a distância entre o chassi do veículo e a superfície da rodovia enquanto o veículo se locomove. Devido a profundidade inerente a um buraco, quando o veículo passar por ele, a distância medida pelo sensor ultrassônico tende a ser maior do que a distância padrão entre o veículo e a rodovia.

A literatura também apresenta soluções acadêmicas e industriais para a detecção automatizada de buracos em estradas. Em (RODE et al., 2009), é proposta uma arquitetura com suporte de comunicação Wi-Fi para um sistema de detecção e alerta de avarias na pista, que auxilia motoristas a evitarem buracos nas estradas, oferecendo avisos prévios. O sistema consiste em pontos de acesso colocados nas estradas para a transmissão de dados, que podem ser recebidos em dispositivos usando comunicação Wi-Fi quando entram na área coberta pela influência dos pontos de acesso. Os nós móveis também podem transmitir sua resposta como um *feedback* que, quando recebido pelo ponto de acesso, pode ser utilizado para processamento em um ser-

vidor. A aplicação de detecção de buracos proposta permite que o motorista receba informações sobre os buracos nas estradas próximas ao veículo, enquanto este se desloca rumo ao seu destino. A aplicação pode ser integrada no veículo de modo a alarmar o motorista na forma de um sinal visual, sinal de áudio ou ainda acionar o sistema de frenagem. Simulações demonstram as vantagens de usar esta abordagem para construir sistemas de detecção de buracos. A avaliação foi feita em termos de número de veículos e quantidade de informação trocada. Muitas dessas aplicações semelhantes podem ser associadas na estrutura fornecida pelo sistema para auxiliar na navegação em estradas.

Muitos trabalhos foram publicados na área de detecção de buracos e condição das estradas e pavimentação nos últimos anos. Devido ao aumento de processamento e quantidade de sensores embutidos, muitos desses trabalhos usam *smartphones* para a coleta e até processamento de dados. Os principais trabalhos estão descritos abaixo.

(MEDNIS et al., 2011) propõem uma ferramenta para detecção de buracos em tempo real usando *smartphones* equipados com acelerômetro. O *smartphone* registra e processa os valores medidos pelo acelerômetro de 3-eixos presente. Para isso, os autores desenvolveram quatro algoritmos: Z-THRESH, Z-DIFF, STDEV(Z) e G-ZERO, demonstrados na Figura 2. O algoritmo Z-THRESH estabelece que qualquer valor medido que ultrapasse um determinado *threshold* é considerado um buraco. O algoritmo Z-DIFF calcula a diferença entre dois valores medidos consecutivos e verifica se essa diferença excede um *threshold* estabelecido. Não é claro se o algoritmo analisa os valores do sinal de forma disjunta ou consecutiva. O algoritmo STDEV(Z) calcula o desvio padrão de uma janela de dados medidos pelo acelerômetro e classifica como buraco caso o valor calculado ultrapasse um determinado *threshold*. O algoritmo G-ZERO foi desenvolvido a partir de uma análise visual dos dados. Os autores perceberam que quando o veículo entrava em um buraco, os valores dos três eixos do acelerômetro se aproximavam de zero. Também foi definido um *threshold* para classificar os buracos do algoritmo G-ZERO. A calibragem dos *thresholds* é parte fundamental dos algoritmos apresentados. Ela foi feita de forma empírica. Os valores usados foram os que tiveram maior taxa de buracos classificados corretamente para cada algoritmo no ambiente testado.

Outras soluções foram propostas para a detecção de buracos. Nericell (MOHAN; PADMANABHAN; RAMJEE, 2008) é um sistema para monitorar as condições de tráfego e das estradas usando *smartphones*. O sistema usa sensores embutidos no *smartphone* como acelerômetro, GPS e microfone. O acelerômetro é usado para detectar lombadas e buracos na pista,

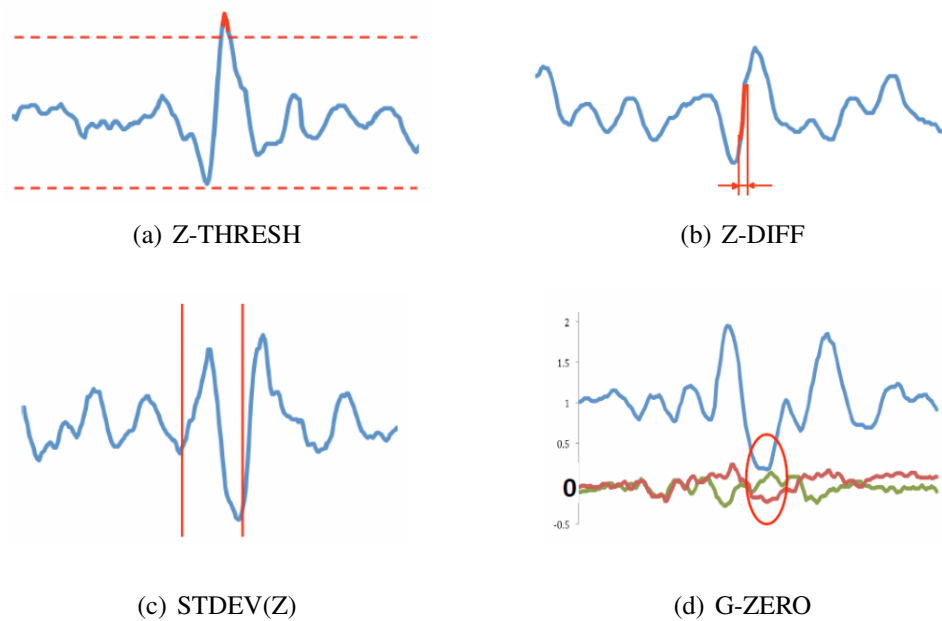


Figura 2 – Algoritmos para detecção de buracos propostos por (MEDNIS et al., 2011).

o algoritmo usado para isso é semelhante ao Z-THRESH proposto por (MEDNIS et al., 2011), que classifica como buraco qualquer leitura do acelerômetro que ultrapassar um certo *threshold* fixado. O acelerômetro também é usado para monitorar a frenagem do carro e para ativar o GPS somente quando o carro estiver em movimento. O microfone é usado para detectar as buzinas do carro, isso serve para monitorar a condição do tráfego. Uma das principais ferramentas propostas pelo sistema é a reorientação virtual do *smartphone*, isto é, diferentemente de outras soluções em que o *smartphone* precisa estar posicionado de uma forma específica durante a medição, usando a reorientação virtual o *smartphone* pode estar posicionado de forma aleatória.

BusNet (ZOYSA et al., 2007) detecta buracos usando dados coletados pelo sistema de transporte público das cidades. A partir de uma rede de sensores projetada para monitorar a poluição ambiental pré-existente, a adição de acelerômetros e dispositivos GPS, permite também monitorar a condição da superfície das ruas. A arquitetura do BusNet é composta por três componentes principais: *Sensor Unit*, *Sub Stations* e uma *Main Station*. São chamados de *sensor units* os ônibus equipados com os sensores. Eles armazenam os valores medidos pelo acelerômetro e GPS. As *sub stations* são localizadas nas paradas de ônibus e recebem os dados coletados pelas *sensor units*. As *sub stations* são conectadas a *main station*, que processa os dados recebidos.

O sistema Pothole Patrol (ERIKSSON et al., 2008) faz a detecção de buracos usando dados coletados a partir do GPS e acelerômetro. O sistema executa com o seguinte suporte

de *hardware*: computador embarcado Soekris 4801 rodando um sistema Linux, módulos de rede, um GPS externo (posicionado no teto do veículo) e um acelerômetro de eixos, montado e posicionado no porta-luvas do veículo. A avaliação do sistema foi realizada com o apoio de uma frota de 7 táxis na cidade de Boston por um período de 10 dias.

No Pothole Patrol, os dados de aceleração coletados via acelerômetro são processados pelo computador embarcado no veículo. Os dados são segmentados em janelas de 256 amostras e coletados em uma frequência de 360 amostras por segundo. Uma série de filtros é aplicada sobre as janelas de dados:

- **velocidade:** Se o carro está parado ou muito devagar, os dados são ignorados.
- **passa-alta:** Remove as frequências baixas dos eixos x e z.
- **z-peak:** Picos de aceleração são as principais características de anomalias na pista. Se o sinal não for maior que um determinado *threshold*, é ignorado.
- **xz-ratio:** A aceleração no eixo-x pode ajudar a diferenciar anomalias na pista. Esse filtro compara os picos dos eixo-x e eixo-z e ignora as amostras se o pico do eixo-x for menor que o pico do eixo-z vezes um determinado fator.
- **velocidade vs. z ratio:** Esse filtro rejeita janelas onde o pico de z é menor que um fator vezes a velocidade.

Os três últimos filtros são dependentes de *thresholds*, que rejeitam eventos que não são buracos. Esses *thresholds* são calculados de acordo com uma base de dados previamente coletada e rotulada à mão que indica a presença ou não de buracos na pista.

Outra métrica utilizada para detecção de buracos foi o agrupamento por localização (*clustering by location*), onde só é considerado um buraco válido se esse ele já foi reportado em um número k de vezes. O agrupamento por localização considera que para um evento ser considerado válido, ele deve ocorrer no mínimo k vezes no mesmo local. Essa técnica é utilizada pra reduzir o número de eventos classificados incorretamente, como vibrações geradas dentro do próprio veículo, que é improvável que aconteça repetidamente no mesmo local em diferentes voltas. Entretanto, os sistemas existentes tem desvantagens. O Pothole Patrol e o BusNet necessitam de *hardware* específico para funcionarem. Outros sistemas, inclusive o aqui proposto, fazem uso de *hardware* convencional, disponível em grande parte dos *smartphones*

da atualidade, que hoje em dia são dispositivos presentes na vida da maioria das pessoas que este artigo visa atingir.

Já os trabalhos propostos por (MEDNIS et al., 2011) e (MOHAN; PADMANABHAN; RAMJEE, 2008) usam *thresholds* estáticos para classificar anomalias nos sinais captados pelos acelerômetros. Os valores dos *thresholds* são calculados com o objetivo de maximizar a faixa de valores que o sinal pode variar sem ser classificado como buraco. Porém, essa abordagem não é adequada para a detecção em diferentes tipos de pavimentações. Diferentes tipos de pavimentações requerem diferentes valores de *thresholds*. Por exemplo, um *threshold* com um valor ótimo para a detecção de buracos em uma estrada asfaltada não pode ser o mesmo usado em uma estrada pavimentada com paralelepípedos, pois a faixa de valores que o sinal pode variar nesse último é muito maior, devido a irregularidade inerente desse tipo de calçamento. No mundo real, cidades e rodovias possuem variados tipos de pavimentações, o que limita o uso desse tipo de abordagem.

(SATTAR; LI; CHAPMAN, 2018) realizaram um estudo sobre o monitoramento de superfícies de estradas usando smartphones. O estudo visa investigar como sensores de smartphones podem ajudar a detectar anomalias sintetizando os maiores problemas e desafios no desenvolvimento atual, e identificar os espaços para futuras pesquisas. O foco do estudo está direcionado exclusivamente para aplicações usando smartphones, implementações usando outros dispositivos foram excluídas do estudo.

(SATTAR; LI; CHAPMAN, 2018) consideram que existem cinco principais etapas nas abordagens para detecção de anomalias: (i) coleta de dados, (ii) pré-processamento, (iii) processamento, (iv) pós-processamento, e (v) avaliação de desempenho. A Figura 3 ilustra o processamento geralmente utilizado para detecção de anomalias usando sensores de smartphones. Os itens destacados em vermelho representam os passos utilizados neste trabalho.

Na primeira etapa, indicada na Figura 3, são considerados dois tipos de dados: baseados em *hardware* (físicos) e baseados em *software* (virtuais). Os dados baseados em *hardware* são adquiridos por sensores físicos como acelerômetro, giroscópio, luz, etc. Os dados baseados em *software* são calculados virtualmente a partir dos valores obtidos pelos dados físicos como a aceleração linear, rotação, gravidade, etc. A etapa de pré-processamento consiste em transformar os dados brutos obtidos em conjuntos de dados organizados e filtrados. Na etapa de processamento, os dados pré-processados são analisados com o objetivo de detectar anomalias das superfícies das estradas. O estudo considera que existem três principais abordagens de

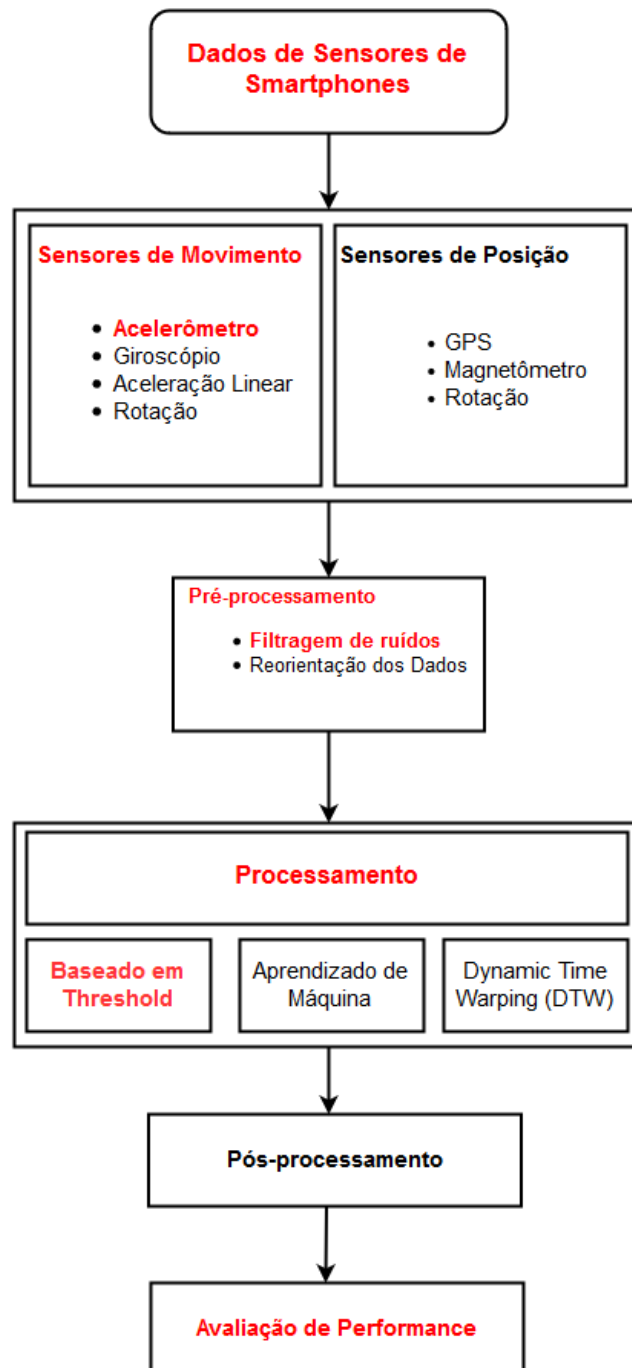


Figura 3 – Processo de detecção de anomalias em estradas. Adaptado de (SATTAR; LI; CHAPMAN, 2018).

processamento: baseadas em *thresholds*, baseadas em aprendizagem de máquina e baseadas no algoritmo *Dynamic Time Warping* (DTW). O pré-processamento e o processamento dos dados podem ser feitos usando dois modos: *online* e *offline*. No modo *offline*, o pré-processamento e o processamento dos dados é feito posteriormente em computadores externos. No modo *online*, a coleta, o pré-processamento e o processamento dos dados são feitos simultaneamente enquanto o veículo passa pelos buracos.

A etapa de pós-processamento inclui o *crowdsourcing* (colaboração coletiva) e a integração de dados de múltiplas fontes (outros veículos) para aumentar a acurácia da detecção. A integração de múltiplas fontes é utilizada para, além de aumentar a acurácia, verificar se um determinado buraco foi consertado. Isso pode ser feito eliminando locais rotulados como buracos que não são registram mais novas detecções.

(SATTAR; LI; CHAPMAN, 2018) concluem que as abordagens apresentadas enfrentam alguns desafios. Apesar dos resultados apresentados pelas diversas implementações presentes no estudo, é difícil comparar a acurácia e performance das abordagens devido a falta de disponibilidade dos algoritmos e conjunto de dados utilizados. Além disso, em muitas abordagens baseadas em *thresholds*, a definição do valor escolhido para esse *threshold* não é clara. Em contraste, nas implementações que possuem métodos de aprendizagem de máquina necessitam de um número grande de dados para treinar os modelos de detecção. Também concluem que uma abordagem ideal deve ser auto-adaptativa e ter auto-aprendizagem.

Este trabalho utiliza uma abordagem baseada em *threshold* com o auxílio de transformadas wavelet, onde o limiar usado é bem definido e auto-adaptativo.

2.2 DESAFIOS DA DETECÇÃO DE BURACOS USANDO SENSORES

A detecção de buracos usando sensores, especificamente o acelerômetro, apresenta alguns desafios e limitações. Primeiro, nesse tipo de abordagem, para que ocorra a detecção do buraco, é necessário que a roda do veículo entre no buraco para que a anomalia no sinal do acelerômetro seja captada. Isso é problemático de maneira que, muitos motoristas ao visualizarem um buraco a frente tentam, se possível, desviá-los, assim o sistema não registraria a existência do mesmo. Implementações baseadas em sensores ópticos não possuem essa limitação, pois a partir do momento em que o buraco é visualizado ele é identificado, independente do motorista desviar ou não. O maior desafio dessa abordagem é minimizar a presença de falsos positivos na detecção. Durante a captação dos dados pelo acelerômetro, existem outros fatores que podem

ser causadores de anomalias nos sinais durante o percurso do veículo como lombadas, bueiros, trilhos, etc. Uma solução para esse problema é a criação de uma lista negra (*black list*), como sugere (ERIKSSON et al., 2008). A lista negra seria preenchida com a localização dessas anomalias conhecidas que não são buracos, nesse caso, o buraco só seria confirmado se a localização da anomalia detectada não constar na lista.

Outro desafio, citado por (SATTAR; LI; CHAPMAN, 2018) é a diferença dos sinais produzidos que ocorre devido aos diferentes modelos de *smartphones*, acelerômetros e veículos, onde varia o tamanho do carro, o sistema de suspensão, o posicionamento do sensor no veículo, entre outras variáveis a serem consideradas. Esse trabalho visa corrigir esse problema, já que, por não usar um *threshold* estático, o valor do *threshold* será calculado considerando todas essas diferenças apontadas.

Para as implementações de detecção de buracos que utilizam sensores e aprendizagem de máquina, outro grande desafio a ser considerado, é a obtenção de uma base de dados para treinar o classificador que será utilizado. Para um classificador ser eficiente, é necessário que uma grande quantidade de dados seja rotulada manualmente como “buraco” e “não buraco”. Em dados providos por sensores essa rotulagem pode ser inviável, já que não há como garantir que a variação do sinal é referente ao buraco sem algum método de localização muito preciso.

2.3 TRANSFORMADA WAVELET DE HAAR

Transformadas são funções aplicadas a dados vetoriais. No caso da transformada wavelet de Haar, tem-se ainda a propriedade de ser uma transformação linear. Com isso, são garantidas propriedades importantes, como a identificação da transformada com uma matriz (a matriz da transformação). Além disso, no caso da wavelet de Haar, a matriz é esparsa do tipo banda, com apenas duas bandas. Ainda tem-se o fato de todos os seus elementos na primeira metade das linhas serem iguais a $1/2$. Na segunda metade $1/2$ e $-1/2$. Isso faz com que a transformada também possa ser calculada a partir de algoritmos rápidos, dispensando a formulação matricial. Desta forma os valores capturados pelo acelerômetro são taxados como sendo o sinal (ou vetor) de entrada a ser transformado.

Diferentes transformadas têm sido frequentemente exploradas para análise de sinais, compressão de dados, e outras aplicações na área de processamento de sinais (PEI; YEH; LUO, 1999): No entanto, a transformada wavelet de Haar, aqui considerada, acaba apresentando propriedades importantes para a análise desejada, como será explorado na metodologia.

Uma das transformadas mais conhecidas em análise de sinais é a transformada de Fourier (TF), que decompõe um sinal em frequências, usando funções de base sinusoidal, ou seja, senos e cossenos. TF é ideal para a análise de sinais estacionários (sinais em que suas propriedades estatísticas não envolvem tempo) (P. M. BENTLEY, 1994), porém não é adequada para análises que precisam manter o domínio de tempo.

A análise de Fourier é uma ferramenta para converter um sinal no domínio de tempo para um sinal no domínio de frequência e vice-versa. A análise usando transformadas wavelet é uma ferramenta para converter um sinal no domínio de tempo em um sinal no domínio de tempo-escala, sendo que as escalas estão associadas a faixas de frequências. Além disso, são transformadas inversíveis, i.e., a partir das componentes pode-se recuperar exatamente os dados de entrada. (JOHN WILEY SONS, 2015).

Segundo (DAUBECHIES, 1992), wavelets são alternativas como bases para representação de funções em $L^2(\mathbb{R})$, que é o espaço das funções que possuem energia finita, ou em termos de sequências, aquelas cujas somas dos seus termos ao quadrado são convergentes. Como exemplos deste espaço estão funções não periódicas para as quais a análise de Fourier deixa de ser a melhor opção.

Neste trabalho, será adotada a Transformada Wavelet de Haar (TWH), que é um caso particular da família de Daubechies. Quando Alfred Haar, em 1910 (HAAR, 1910) definiu as funções pulso e escada, ele não imaginaria que 70 anos depois elas seriam consideradas como base de uma transformação tão eficiente e versátil. E que ainda seriam tratadas de forma tão elegante, quanto feito por Daubechies, ao analisar as transformadas de Fourier das funções de Haar e de outras funções (posteriormente denominadas wavelets) e perceber que, no domínio da frequência, estas possuem propriedades especiais. Na verdade, foi o trabalho de Daubechies que identificou as wavelets com filtros FIR, dando origem aos algoritmos rápidos de decomposição em diferentes níveis de resolução. A principal ideia da decomposição multi-resolução dos sinais realizada pela transformada wavelet é a obtenção de uma componente de aproximação usando a função escala, que no caso de Haar é a função pulso constante. Esta por sua vez está associada a um filtro passa-baixa. Para a segunda componente da transformação são usados os filtros passa-alta, associados à função wavelet, que na TWH é a função escada. Os valores desta componente são chamados de detalhes ou coeficientes wavelets que captam as informações em alta frequência do sinal (JOHN WILEY SONS, 2015; DAUBECHIES, 1992).

Existem dois algoritmos, ou duas formas, para a decomposição do sinal utilizando a

transformada wavelet discreta de Daubechies, e portanto, também para a TWH: a forma decimada (algoritmo cascata) e a forma não decimada (algoritmo à trous). Para formulações referentes à formulação contínua das transformadas wavelets, ver (DAUBECHIES, 1992).

2.3.1 Algoritmo cascata

Na forma decimada, a cada nível da decomposição, o nível seguinte terá a metade da resolução do nível anterior, isto é, terá a metade do número de elementos. Isso se dá porque as médias e os detalhes são calculadas de maneira disjunta. Para exemplificar, Considere um sinal de entrada $x[n] = \{x_1, x_2, x_3, \dots, x_n\}$, onde n é o número de elementos do vetor. As médias, ou aproximações, e os detalhes de primeiro nível são dados por 2.1 e 2.2:

$$C_1 = \left\{ \frac{x_1 + x_2}{2}, \frac{x_3 + x_4}{2}, \dots, \frac{x_{n-1} + x_n}{2} \right\} \quad (2.1)$$

$$D_1 = \left\{ \frac{x_1 - x_2}{2}, \frac{x_3 - x_4}{2}, \dots, \frac{x_{n-1} - x_n}{2} \right\} \quad (2.2)$$

Para calcular J níveis da TWH, sendo j o nível da decomposição e $j = 1, 2, 3, \dots, J$, as médias C_j e os detalhes D_j são calculados sempre a partir do sinal C_{j-1} . Aqui, os coeficientes de filtro passa-baixa $G = [1/2, 1/2]$ são os que produzem as médias em 2.1. Os filtros passa-alta $H = [1/2, -1/2]$ geram as variações em 2.2. A Figura 4 ilustra três níveis da TWH usando o algoritmo Cascata (MALLAT, 1989).

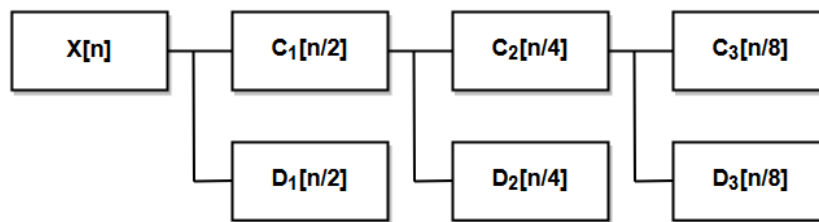


Figura 4 – Decomposição do sinal de entrada $x[n]$ em 3 níveis usando o algoritmo Cascata.

A THW na forma decimada pode ser representada pelo Algoritmo 1, denominado Algoritmo de Cascata, pois em cada nível de decomposição, cada uma das duas componentes geradas possui metade do número de elementos do nível mais fino que as gerou. Assume-se C_0 o vetor do sinal original com N_0 elementos como sendo o vetor de entrada para o algoritmo. O nível de resolução inicial é o mais fino, e o último nível pode conter apenas um elemento em cada bloco (decomposição em todos os níveis possíveis) ou pode parar em um nível intermediário L . Assim, outro dado de entrada é a quantidade L níveis de decomposição. Para cada nível

$i = 0, 1, 2, \dots, L - 1$, o vetor C_i é decomposto em um outro vetor de médias C_{i+1} e um novo vetor de coeficientes wavelets (detalhes) D_{i+1} , ambos contendo $N_{i+1} = N_i/2 = N_0/(2^{i+1})$ elementos.

No algoritmo Cascata, as médias e os detalhes são calculados de forma disjunta. Lembrando que a cada nível de transformação, apenas o vetor com os coeficientes de escala (vetor das médias) é decomposto, então considerando-se o nível mais fino indexado por zero, e ainda L níveis de decomposição, a saída do algoritmo é o vetor de médias do nível mais grosseiro L $[C_L]$ e todos os vetores com coeficientes wavelet (detalhes) $[D_L], [D_{L-1}], \dots, [D_1]$.

Algoritmo 1: ALGORITMO CASCATA

Entrada: vetor do sinal original C_0 , tamanho N_0 , nível L

Saída: $[[C_L], [D_L], \dots, [D_1]]$

início

para $i = 0, 1, 2, \dots, L - 1$ **faça**

$N_i = N_0/(2^{i+1})$

para $j = 0, 1, \dots, \leq N_i - 1$ **faça**

$C_{i+1,j} = 1/2C_{i,2j} + 1/2C_{i,2j+1}$

$D_{i+1,j} = 1/2C_{i,2j} - 1/2C_{i,2j+1}$

fim

fim

fim

Apenas uma observação, $H = (1/2, 1/2)$ e $G = (1/2, -1/2)$ que aparecem no cálculo dos elementos de cada bloco C e D no algoritmos são os filtros associados à TWH não normalizada, uma vez que quando vistos como vetores, H e G não possuem norma 1. Uma das propriedades das transformadas wavelet é que o sinal pode ser recomposto, isto é, a partir dos dados médios no nível L e todas os vetores de detalhes, D_j , nos níveis $L, L-1, \dots, 1$, é possível restaurar o sinal original C_0 .

Algoritmo 2: ALGORITMO CASCATA INVERSA

Entrada: vetores $C_L, D_L, D_{L-1}, \dots, D_1$, tamanho N_L , nível L

Saída: C_0, N_0

início

para $i = L, L - 1, \dots, 1$ **faça**

para $j = 0, 1, 2, \dots, N_i - 1$ **faça**

$C_{i-1,2j} = C_{i,j} + D_{i,j}$

$C_{i-1,2j+1} = C_{i,j} - D_{i,j}$

fim

$N_{i-1} = 2N_i$

fim

fim

Naturalmente, cada nível da inversa do algoritmo cascata possui o dobro de elementos do nível anterior. Informações mais detalhadas sobre as diferentes formulações matriciais

para a TWH estão apresentadas em (SILVEIRA; KOZAKEVICIUS, 2016) e suas versões em aritmética intervalar estão em (SANTOS et al., 2018).

2.3.2 Algoritmo à trous

Na forma não decimada para a TWH, dada pelo algoritmo à trous, os vetores com as médias(coeficientes escala) e os detalhes(coeficientes wavelets) são sempre do mesmo tamanho N_0 que o vetor original. Desta forma, esta transformada apresenta dados redundantes ao invés de disjuntos, como na forma não-decimada.

Para exemplificar, considera-se novamente o sinal de entrada $x[n] = \{x_1, x_2, x_3, \dots, x_n\}$, onde n é o número de elementos do vetor. As médias e os detalhes no algoritmo à trous no primeiro nível são dados por 2.8 e 2.4:

$$C_1 = \left\{ \frac{x_1 + x_2}{2}, \frac{x_2 + x_3}{2}, \dots, \frac{x_{n-2} + x_{n-1}}{2}, \frac{x_{n-1} + x_n}{2} \right\} \quad (2.3)$$

$$D_1 = \left\{ \frac{x_1 - x_2}{2}, \frac{x_2 - x_3}{2}, \dots, \frac{x_{n-2} - x_{n-1}}{2}, \frac{x_{n-1} - x_n}{2} \right\} \quad (2.4)$$

A Equação 2.4 é equivalente à Equação 2.5, ou seja, no algoritmo à trous os detalhes são calculados pela média do nível anterior menos a média calculada do nível atual.

$$D_L = C_{L-1} - C_L \quad (2.5)$$

A Figura 5 ilustra três níveis da TWH usando o algoritmo à trous.

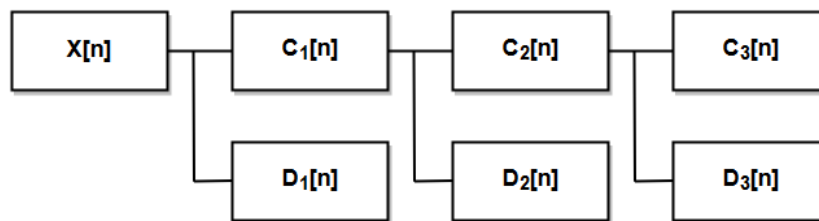


Figura 5 – Decomposição do sinal de entrada $x[n]$ em 3 níveis usando o algoritmo à trous.

No Algoritmo 3, o sinal original C_0 , o número de elementos N_0 e o nível de decomposições L são dados de entrada. Para cada nível $i = 1, 2, \dots, L$, com o vetor C_{i-1} é feita a convolução com o filtro H para produzir o vetor com as médias C_i . O vetor com os detalhes do próximo nível D_i é obtido via diferença entre os blocos com coeficientes de escala de níveis consecutivos C_i e C_{i-1} .

Para o cálculo desta transformada, é necessária a identificação dos valores de fronteira de tal modo que $C_{i,N_0} = C_{i,0}$ em cada nível i .

Algoritmo 3: ALGORITMO À TROUS

Entrada: vetor do sinal original C_0 , tamanho N_0 , nível L

Saída: $[[C_L], [D_L], \dots, [D_1]]$

início

para $i = 1, 2, \dots, L$ **faça**

para $j = 0, 1, 2, \dots, N_0 - 1$ **faça**

$C_{i,j} = 1/2C_{i-1,j} + 1/2C_{i-1,j+1}$

fim

para $j = 0, 1, 2, \dots, N_0 - 1$ **faça**

$D_{i,j} = C_{i-1,j} - C_{i,j}$

fim

fim

fim

Da mesma forma que no algoritmo cascata, é possível calcular a inversa do algoritmo à trous. Para restaurar o sinal original C_0 , temos a seguinte Equação 2.6:

$$C_0 = C_L + (D_L + D_{L-1} + \dots + D_2 + D_1) \quad (2.6)$$

Dessa maneira a inversa pode ser descrita também pela Equação 2.7:

$$C_0 = C_L + (C_{L-1} - C_L) + (C_{L-2} - C_{L-1}), \dots, (C_0 - C_1) \quad (2.7)$$

O Algoritmo 4 descreve a THW à trous inversa, onde é possível calcular o sinal de C_0 apenas com o sinal de C_L .

Algoritmo 4: ALGORITMO À TROUS INVERSA

Entrada: vetor do sinal $C_L, D_L, D_{L-1}, \dots, D_1$, tamanho N_0 e nível L

Saída: C_0

início

para $i = L, L - 1, \dots, 1$ **faça**

para $j = 0, 1, 2, \dots, N_0 - 1$ **faça**

$C_{i-1,j} = C_{i,j} + D_{i,j}$

fim

fim

fim

2.3.3 Operação de Truncamento (Threshold)

Uma vez obtidos os coeficientes wavelets pela TWH, a operação de truncamento \mathcal{T} é imposta apenas aos coeficientes wavelets da transformação: $\hat{D}_i = \mathcal{T}(D)$, sendo responsável pela seleção de coeficientes que serão mantidos na representação após escolhido um valor de *threshold* λ , ou seja, um critério para o truncamento. Assim, segundo (DONOHO; JOHNSTONE,

1995), tem-se:

$$\hat{d}_{i,j} = \begin{cases} d_{i,j}, & \text{if } |d_{i,j}| > \lambda, \\ 0, & \text{if } |d_{i,j}| \leq \lambda \end{cases} \quad (2.8)$$

2.3.4 Conclusão do Capítulo

Neste Capítulo foram apresentados: (i) fundamentação teórica deste trabalho; (ii) algoritmos da TWH, forma decimada e não-decimada; (iii) algoritmo para a operação de truncamento dos coeficientes wavelets.

3 METODOLOGIA

Este Capítulo tem como objetivo detalhar o funcionamento do sistema proposto. Conforme previamente mencionado, diferentemente dos outros trabalhos, que focam no processo de detecção do buraco com o auxílio do acelerômetro, usando *thresholds* manualmente calibrados, este trabalho visa propor um sistema com *threshold* automatizado para detecção de buracos usando uma abordagem wavelet aplicada a sinais de acelerômetro.

A arquitetura do sistema proposto pode ser dividida em quatro partes: (i) coleta de dados, (ii) processamento, (iii) localização e (iv) classificação, como mostrado na Figura 6. Como visto na Seção 2.1.3, em (SATTAR; LI; CHAPMAN, 2018) são descritos cinco etapas presentes nos sistemas de detecção de buracos usando sensores: coleta de dados, pré-processamento, processamento, pós-processamento e avaliação de performance. No sistema proposto neste trabalho, as etapas de pré-processamento e processamento são sumarizadas na etapa de processamento e a etapa de avaliação da proposta é feita no Capítulo 4. Como o sistema não faz uso de *crowdsourcing* e integração de múltiplas fontes, a etapa de pós-processamento não é contemplada.

Essencialmente, os dados são coletados com o auxílio do acelerômetro presente em um *smartphone* ou em um *hardware* próprio, armazenados em arquivos e, posteriormente, processados através de transformadas wavelet e *thresholding* automatizado para detectar buracos em estradas. Um serviço adicional de localização é usado como forma de gabarito. Este serviço não é usado para localizar o buraco no tempo e no espaço, mas serve para definir, no sinal, a posição exata da oscilação gerada pelo buraco de forma imparcial, sem inferir a ocorrência da oscilação pelo resultado esperado e não pelo resultado medido. De fato, um serviço de classificação das condições das estradas precisa contar com um serviço que ofereça a localização precisa de cada buraco. As etapas do sistema proposto são explicadas em mais detalhes nas seções deste capítulo.

Para testar a eficiência da metodologia proposta, o sistema foi implementado primeiro em um cenário controlado e posteriormente em cenário real. O cenário controlado é composto por um veículo-robô, de tamanho reduzido e uma pista sintética esburacada. Após validada a solução no cenário controlado, o sistema foi implementado em um cenário real, que consiste na utilização da solução em veículos reais com o propósito de detectar buracos em estradas.

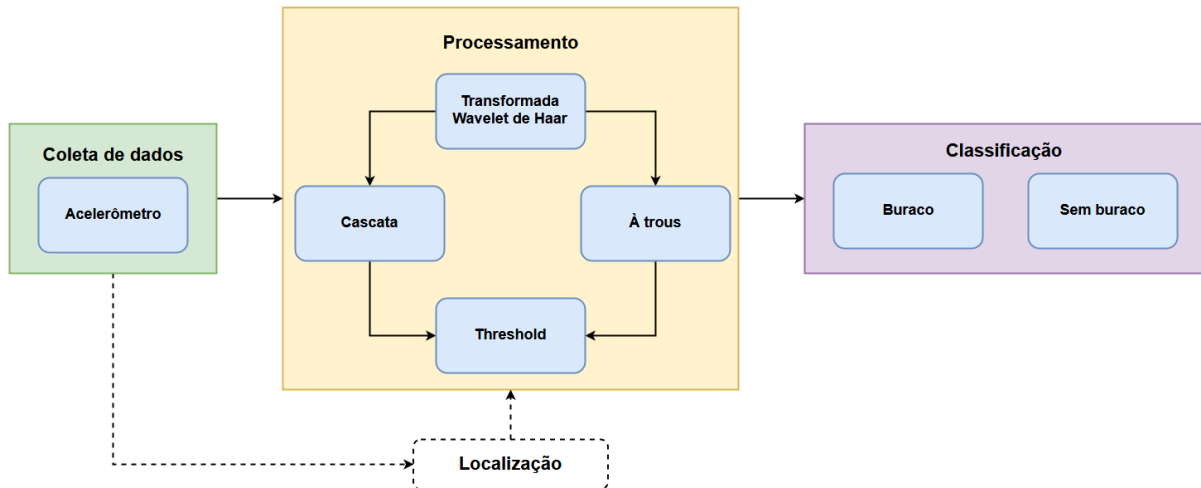


Figura 6 – Arquitetura do sistema proposto para a detecção de buracos em estradas.

3.1 COLETA DE DADOS

A etapa de coleta de dados compreende todo o processo de aquisição e armazenamento dos dados gerados pelo acelerômetro, além da implantação física do sistema desenvolvido no veículo. Para isso, foram exploradas diferentes abordagens: o desenvolvimento de uma aplicação para *smartphones*, a utilização de um aplicativo de coleta de dados de sensores existente em lojas virtuais e o desenvolvimento de uma aplicação e *hardware* próprio usando a Plataforma Arduino.

Inicialmente foi desenvolvida uma aplicação para *smartphones* que utilizam o sistema operacional Android para salvar as leituras do acelerômetro e do GPS embutidos no dispositivo. O acelerômetro e o GPS presentes em alguns *smartphones* são utilizados para diversos tipos de aplicações, como a rotação automática da tela e para aplicativos que utilizam a localização do usuário. A aplicação desenvolvida consiste em uma tela com um botão, que é responsável por iniciar e terminar a gravação dos dados do acelerômetro e do GPS em um arquivo. O arquivo é salvo no formato CSV (*Comma Separated Values*) e cada linha é uma tupla do tipo "*ax, ay, az, long, lat*", onde *ax, ay* e *az* correspondem as leituras dos eixos x, y e z do acelerômetro e *long* e *lat* correspondem as coordenadas de longitude e latitude providas pelo GPS. A aplicação foi desenvolvida na linguagem de programação Java para o sistema operacional Android e funciona apenas em aparelhos que possuem os sensores utilizados.

A segunda abordagem para realizar a coleta de dados é usar o acelerômetro embutido em um *smartphone* e aplicações já existentes. Por exemplo, a gravação dos dados pode ser realizada

por meio do aplicativo Accelerometer Analyzer⁴ para o sistema operacional Android, disponível na loja virtual do sistema. O Accelerometer Analyzer não armazena os dados referentes ao GPS, tornando o uso dessa abordagem restrito à análise dos sinais, desconsiderando o fator localização.

Finalmente, foi desenvolvida uma aplicação e *hardware* próprio usando a Plataforma Arduino. A Figura 7 apresenta uma fotografia do dispositivo desenvolvido. O *hardware* utilizado é composto por uma placa microcontroladora Arduino, um acelerômetro de três eixos, um adaptador para cartão SD e um sensor ultrassônico. As especificações dos componentes utilizados serão melhor descritas posteriormente no Capítulo 4. A proposta da aplicação é armazenar os dados gerados pelo acelerômetro em arquivo, semelhante a aplicação desenvolvida para *smartphones*, juntamente com os valores obtidos pelo sensor ultrassônico. O sensor ultrassônico é utilizado para determinar a posição do buraco no sinal gerado pelo acelerômetro.

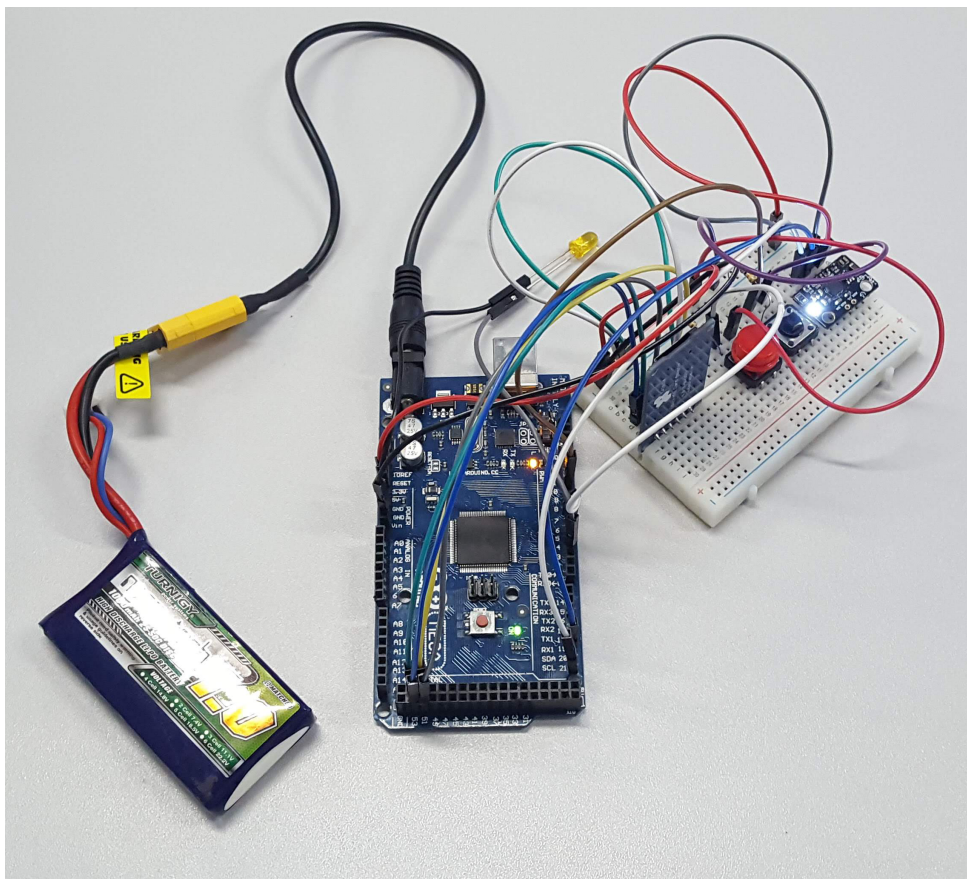


Figura 7 – Dispositivo com acelerômetro construído para a coleta de dados.

Outro fator importante na coleta de dados, é o posicionamento do dispositivo utilizado, *smartphone* ou *hardware* próprio, no veículo. A detecção de buracos por acelerômetro se ba-

⁴ <https://apkpure.com/br/accelerometer-analyzer/com.lul.accelerometer>

seja, basicamente, na vibração causada quando a roda do veículo passa por um buraco. Dessa maneira, o posicionamento do acelerômetro dentro do veículo é essencial para que a leitura da aceleração seja a melhor possível. Por outro lado, o posicionamento do dispositivo precisa ser de fácil manejo pelo usuário para que não haja uma limitação na utilização. Pensando nisso, a solução encontrada foi posicionar o dispositivo no painel do carro, que fica numa posição que capta a aceleração gerada de maneira satisfatória, pois fica próxima ao eixo dianteiro do veículo, e facilita o posicionamento pelo usuário.

Além do posicionamento do dispositivo dentro do veículo, é necessário considerar a posição do dispositivo no painel. Quando a roda do veículo “cai” em um buraco, é exercida uma aceleração vertical no dispositivo. Em um acelerômetro de três eixos, no qual é medida a aceleração nos eixos cartesianos x , y e z , para que a aceleração vertical seja detectada da maneira mais eficiente possível, o eixo que será medido deve estar posicionado perpendicularmente a superfície do painel. A Figura 8 ilustra um *smartphone* e os eixos de orientação do mesmo. Nesse caso, se quisermos usar a aceleração do eixo z no sistema de detecção de buracos, a melhor posição do dispositivo é com a tela para cima, assim, a aceleração vertical exercida sobre o eixo z seria completa.

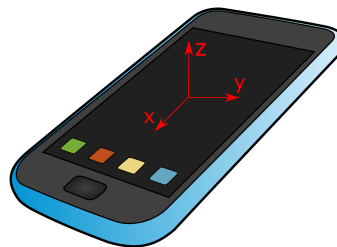


Figura 8 – Posicionamento do *smartphone* em relação aos eixos x , y e z do acelerômetro

3.2 PROCESSAMENTO

A etapa de processamento dos dados engloba a preparação dos dados, ou pré-processamento, a análise dos sinais usando transformadas wavelet e a definição de um *threshold* adaptativo para detecção de buracos. É a etapa mais importante do sistema proposto, na qual se encontra a principal contribuição deste trabalho.

Primeiramente, é necessário fazer um tratamento dos dados adquiridos na etapa anterior, que serão chamados de dados não tratados. Os dados não tratados podem vir em diferentes formatos, em diferentes escalas e com informações extras dependendo da fonte utilizada. Os

dados oriundos do aplicativo Accelerometer Analyzer são gerados em um arquivo texto contendo um cabeçalho com informações sobre a gravação e as medições do acelerômetro logo abaixo, com uma linha por amostra, onde cada linha contém as medições dos eixos x , y e z e o tempo que levou desde a medição da amostra anterior. Neste caso, apenas as medições do eixo z são mantidas. Além disso, os dados precisam ser convertidos da unidade de m/s^2 para unidade gravidade g , que será utilizada na etapa de análise.

Já na aplicação implementada com Arduino, os dados já são salvos no formato correto, com apenas o eixo z , porém com um parâmetro adicional, um valor 0 ou 1 para cada linha de amostra de z , que representa a presença ou não de buracos naquele local.

Outro tratamento necessário é a remoção de amostras não relacionadas à medição nos sinais. Isto é, a remoção de partes do sinal, especificamente parte do início e do fim, que correspondem as medições feitas pelo acelerômetro enquanto o dispositivo estava sendo manuseado quando ligado e desligado, e não necessariamente medindo a presença de buracos. Os dados tratados servirão como entrada para a próxima etapa, a análise dos dados via wavelet.

3.2.1 Análise dos dados via transformadas wavelet

Após tratados, os dados são submetidos a etapa de análise utilizando transformadas wavelet. Como explicado na seção 2.3, as transformadas wavelet são ferramentas de decomposição multi-resolução de sinais, que no contexto deste trabalho, são os sinais captados pelo acelerômetro. Após a aplicação das transformadas wavelet sobre o sinal, é calculado um *threshold*, que servirá como um limiar entre os valores considerados aceitáveis e valores considerados anomalias causadas pela presença de buracos.

Para a análise dos dados, foram implementados três algoritmos que utilizam a Transformada Wavelet de Haar. Sendo dois deles, os algoritmos cascata e à trous puros, já descritos anteriormente e uma variação do algoritmo à trous, que será denominado algoritmo à trous de dois passos ou algoritmo à trous e remoção do sinal médio.

3.2.1.1 Algoritmo cascata

O algoritmo cascata decompõe o sinal de entrada, ou sinal original, em médias e detalhes em L níveis, onde cada nível possui a metade da resolução do nível anterior, isto é, a metade do tamanho. Para melhor entendimento, o sinal original será denominado por C_0 , as médias por C_i e os detalhes por D_i , onde $i = 1, 2, \dots, L$.

Dado um sinal de entrada C_0 , representado na Figura 9, é aplicado o primeiro nível de decomposição usando a TWH com algoritmo cascata. O resultado são os sinais de médias C_1 e detalhes D_1 , que correspondem ao primeiro nível da transformada. A Figura 10 representa os sinais C_1 e D_1 gerados, note que o número de amostras dos sinais é a metade do número de amostras de C_0 .

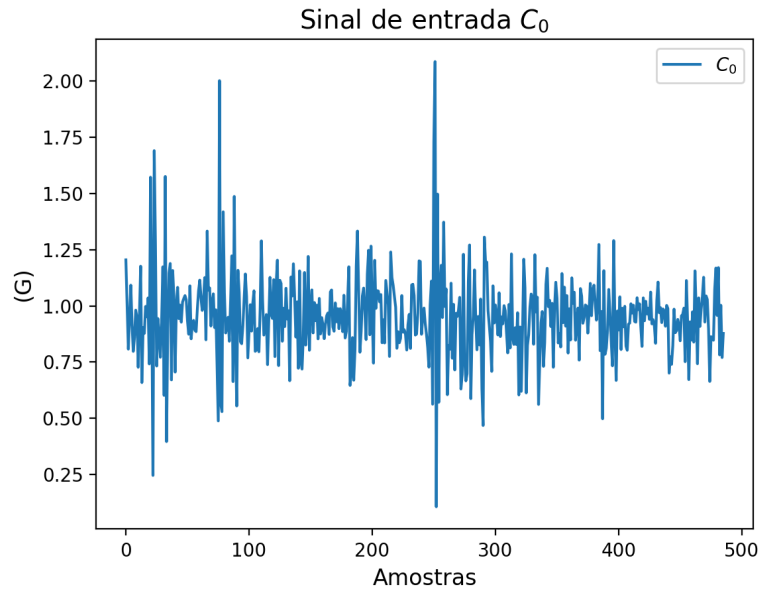


Figura 9 – Sinal original C_0 .

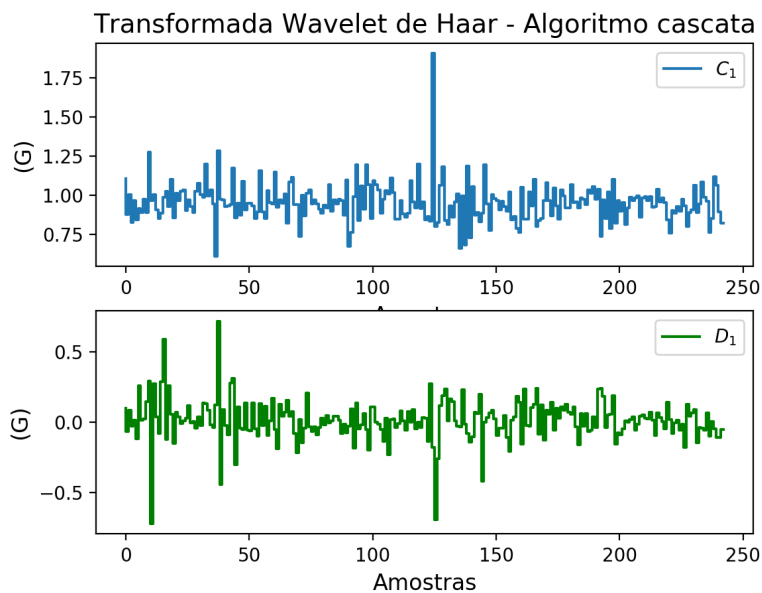


Figura 10 – Médias C_1 e detalhes D_1 resultantes do primeiro nível de decomposição da TWH usando o algoritmo cascata.

Para a análise dos dados usando o algoritmo cascata, são considerados apenas os sinais

de detalhes. Os detalhes de primeiro nível D_1 podem ser interpretados como a diferença absoluta entre o valor de uma amostra e a amostra seguinte. Variações de valores consecutivos muito altas podem indicar um evento causado por uma aceleração repentina captada pelo sensor. Além do primeiro nível de decomposição, podem ser calculados L níveis usando a TWH com algoritmo cascata. A Figura 11 ilustra três níveis de detalhes gerados a partir do sinal C_0 .

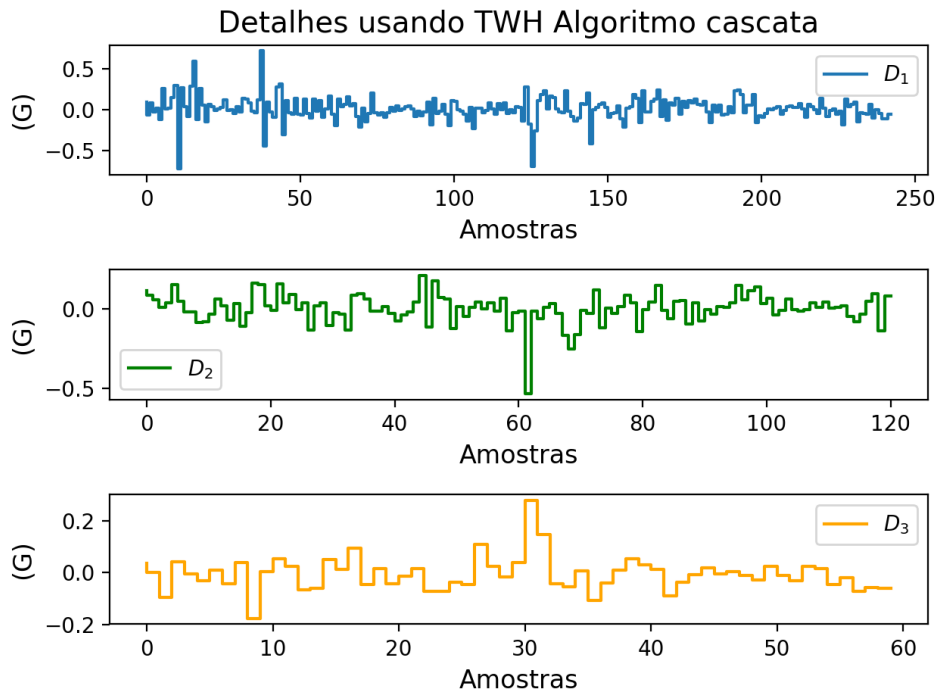


Figura 11 – Três níveis de detalhes usando a TWH algoritmo cascata.

3.2.1.2 Algoritmo à trous

No algoritmo à trous, o sinal de entrada C_0 é decomposto em médias e detalhes em L níveis, porém, diferentemente do algoritmo cascata, os sinais gerados em cada nível mantêm a mesma resolução, ou número de amostras, do sinal original. A Figura 12 mostra o primeiro nível de decomposição de C_0 usando a TWH com algoritmo à trous.

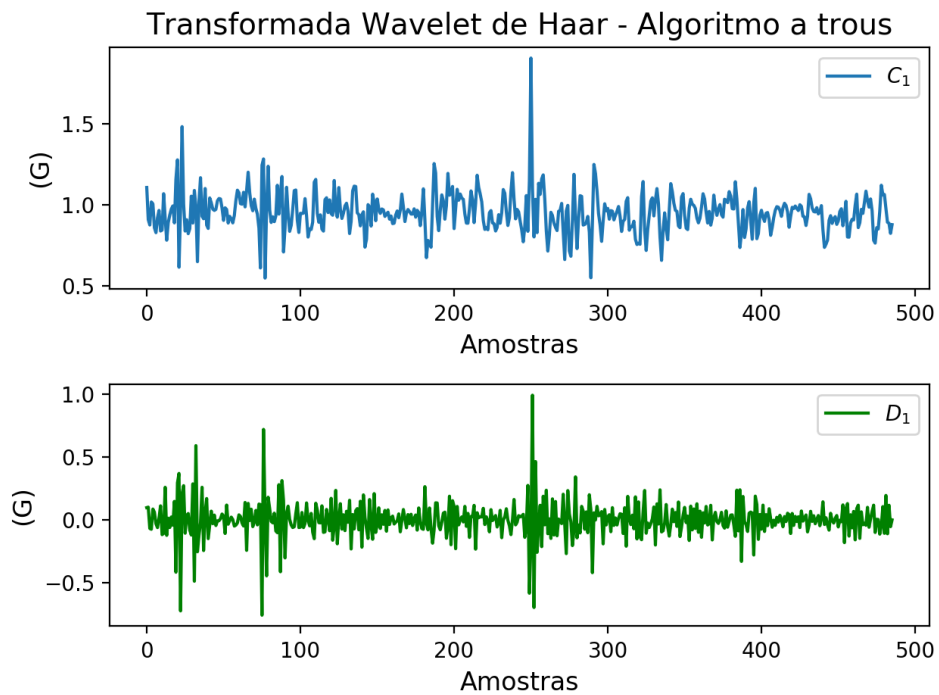


Figura 12 – Médias C_1 e detalhes D_1 resultantes do primeiro nível de decomposição da TWH usando o algoritmo à trous.

Para a análise dos dados usando o algoritmo à trous, são considerados os sinais de médias. As médias calculadas no algoritmo à trous são analisadas de maneira que, a cada nível de decomposição, o sinal é suavizado e as anomalias ficam em evidência. A Figura 13 ilustra as médias de cinco níveis de decomposição do sinal original C_0 .

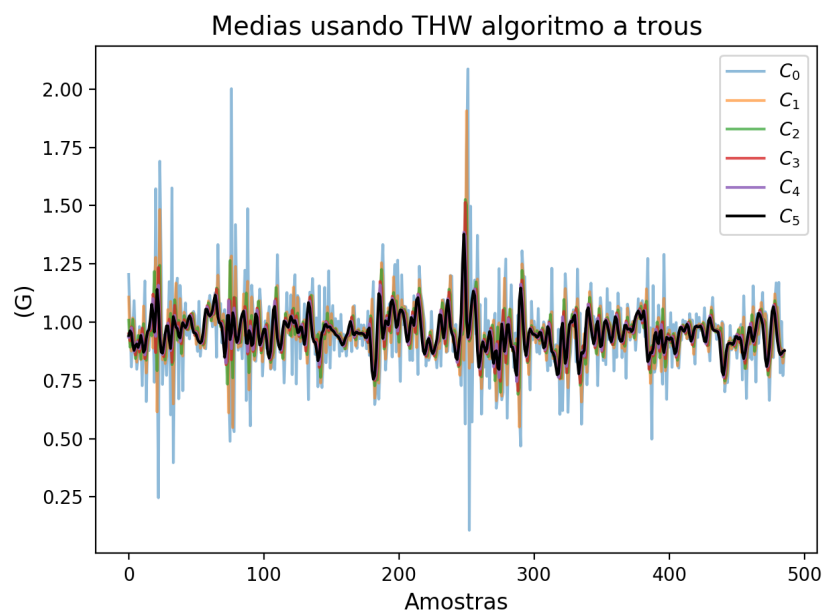


Figura 13 – Cinco níveis de médias usando a THW algoritmo à trous.

3.2.1.3 Algoritmo à trous e remoção do sinal médio

Os dados a serem analisados possuem alta variabilidade intrínseca, pois estão relacionados ao tipo de pavimentação e quantidade de buracos. Neste sentido, uma maneira de realçar as perturbações que realmente interessam na análise é tentar anular as oscilações que são aceitáveis. Para isso, é necessário que se identifique o que são estes dados médios e estes sejam removidos do sinal, antes da operação de threshold ser considerada. Com isso, há uma atenuação do bias causado por estas informações. Desta forma, uma heurística simplificada do algoritmo proposto em (KOZAKEVICIUS; SCHMIDT, 2013) é:

1. Dado o sinal original C_0 , este é decomposto em L níveis para TWH à trous (Algoritmo 3), sendo C_L as médias do último nível;
2. Obtém-se $C_0^1 = C_0 - C_L$. Com isso, o sinal C_0^1 tem uma atenuação das informações médias e pode ser transformado novamente;
3. A transformada à trous é aplicada, mas agora tendo como dado de entrada C_0^1 .

Este processo iterativo pode ser aplicado várias vezes de forma adaptativa. Neste caso, o critério de parada é quando a distância entre C_0^{i+1} e C_0^i for menor do que uma certa tolerância, como feito em (KOZAKEVICIUS; SCHMIDT, 2013). Neste trabalho, o processo de retirada dos dados médios é realizado apenas um número fixo de vezes, são assumidos 2 "passos".

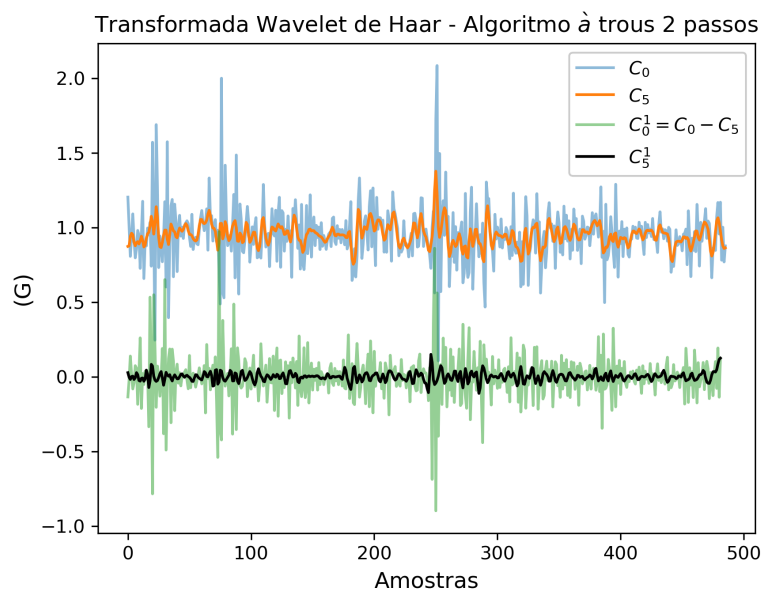


Figura 14 – Algoritmo à trous de 2 passos

A Figura 14 ilustra o sinal original C_0 e o sinal de médias C_5 após a aplicação do

algoritmo à trous com 5 níveis. Na Figura, também tem-se o sinal C_0^1 , que é resultado da remoção do sinal médio C_5 do sinal original C_0 . E finalmente, decompondo o sinal C_0^1 , tem-se o sinal de médias C_5^1 .

3.2.2 Thresholding

A etapa de *Thresholding* é feita logo após a etapa de análise de dados via transformadas wavelet. Um *threshold* nada mais é que um limiar, ou um valor numérico delimitante. No contexto deste trabalho, *thresholds* são utilizados para delimitar quais valores são considerados aceitáveis nas leituras do acelerômetro e quais valores são considerados anomalias causadas pela presença de buracos.

Como visto na Seção 2.1.3, *thresholds* são utilizados em diversos trabalhos na detecção de buracos em estradas. O problema do uso de *thresholds* nos trabalhos revisados é que os valores são estáticos e definidos manualmente através de análises empíricas. Existem diversos fatores que podem influenciar nos valores lidos pelo acelerômetro, sendo assim, leituras de um mesmo evento, ou mesmo buraco, podem apresentar valores divergentes. Os valores podem divergir até mesmo sem a presença de buracos, o que causa a classificação de falsos positivos pelo sistema.

Um dos fatores mais importantes que podem influenciar na diferença de valores lidos pelo acelerômetro é que, diferentes modelos de veículos possuem diferentes sistemas de suspensão, diferentes tamanhos e diversas outras variáveis que fazem com que a vibração sentida pelo acelerômetro em um veículo seja diferente da vibração sentida em outro. Por exemplo, em um veículo com um sistema de suspensão robusto, a vibração gerada pela presença de um buraco será menor que a vibração gerada em um veículo com um sistema de suspensão mais simples. Outro fator a ser considerado é o tipo de pavimentação em que o sistema consegue detectar a presença de buracos. Por exemplo, estradas pavimentadas com paralelepípedos ou de chão batido, apresentam vibrações maiores em comparação com estradas asfaltadas, devido as irregularidades inerentes a esse tipo de pavimentação.

Diante desse cenário, vem a necessidade de se calcular um *threshold* dinâmico, que sirva independente dos fatores apontados. É fato que para detectar uma anomalia em um conjunto de dados, é necessário que haja uma certo padrão nesse conjunto. Esse padrão pode ser um comportamento previsível ou um comportamento aleatório dos dados, desde que dentro de um alcance reconhecível do ponto de vista analítico. Pensando nisso, foi escolhido um *threshold*

com base na distribuição normal.

A distribuição normal, também chamada de distribuição gaussiana, é uma das distribuições probabilísticas mais conhecidas e importantes na área da estatística. É utilizada para calcular a probabilidade de uma variável aleatória estar localizada abaixo da curva de uma função. A distribuição normal é dada pela média aritmética e o desvio padrão de um conjunto de dados, e possui a forma de um “sino“, como mostra a Figura 15. Como pode ser observado, quanto maior o desvio padrão, representado por σ , em relação a média aritmética μ , maior a probabilidade da variável em questão estar dentro da área da curva. Segundo (PATEL; READ, 1982), aproximadamente 99,994% dos valores de uma variável aleatória normal estão dentro de mais ou menos quatro desvios padrão de sua média.

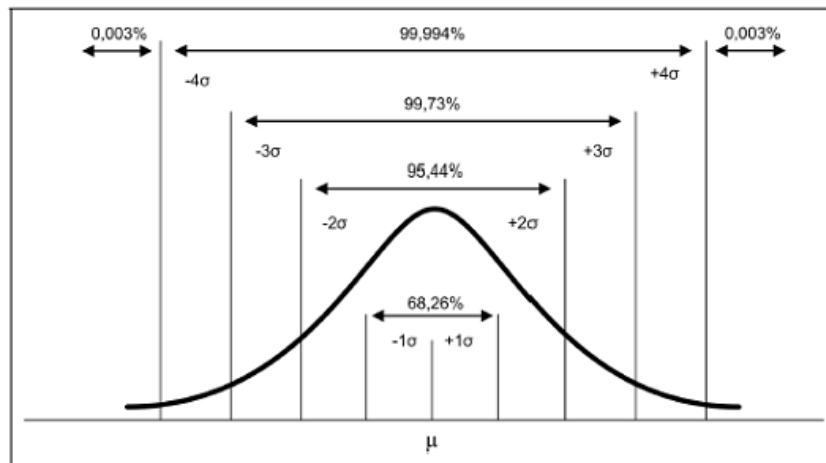


Figura 15 – Curva da distribuição normal. Fonte (ACTION, 2018).

Neste trabalho, foram implementados um *threshold* superior λ_{sup} e um *threshold* inferior λ_{inf} , que detectam a aceleração vertical, tanto positiva quanto negativa, causada pela presença de um buraco. Os dois *thresholds* são representados, respectivamente, pelas Equações 3.1 e 3.2:

$$\lambda_{sup} = \mu + fator \cdot \sigma \quad (3.1)$$

$$\lambda_{inf} = \mu - fator \cdot \sigma, \quad (3.2)$$

sendo μ a média dos dados avaliados. σ é o desvio padrão e *fator* é o valor que multiplica o desvio padrão.

3.3 LOCALIZAÇÃO

Além dos valores medidos pelo acelerômetro, a etapa de coleta de dados também pode registrar a localização dos mesmos, através da longitude e da latitude informadas pelo GPS. Com essas informações é feita a localização dos buracos, o que permite alertar os demais veículos e o próprio veículo (quando na repetição de trajeto) sobre a existência de buracos, além de permitir mapear as estradas de acordo com suas condições. A Figura 16 ilustra um cenário de um conjunto de vias e indicação de buracos. Os buracos precisam ser localizados com precisão e acurácia adequadas para que os veículos possam evitá-los. Entretanto, as implementações de GPS tipicamente encontradas nos dispositivos comerciais não são muito efetivas para o problema de localização de buracos. Elas pecam por não apresentar a precisão e a acurácia necessárias para a localização de buracos.

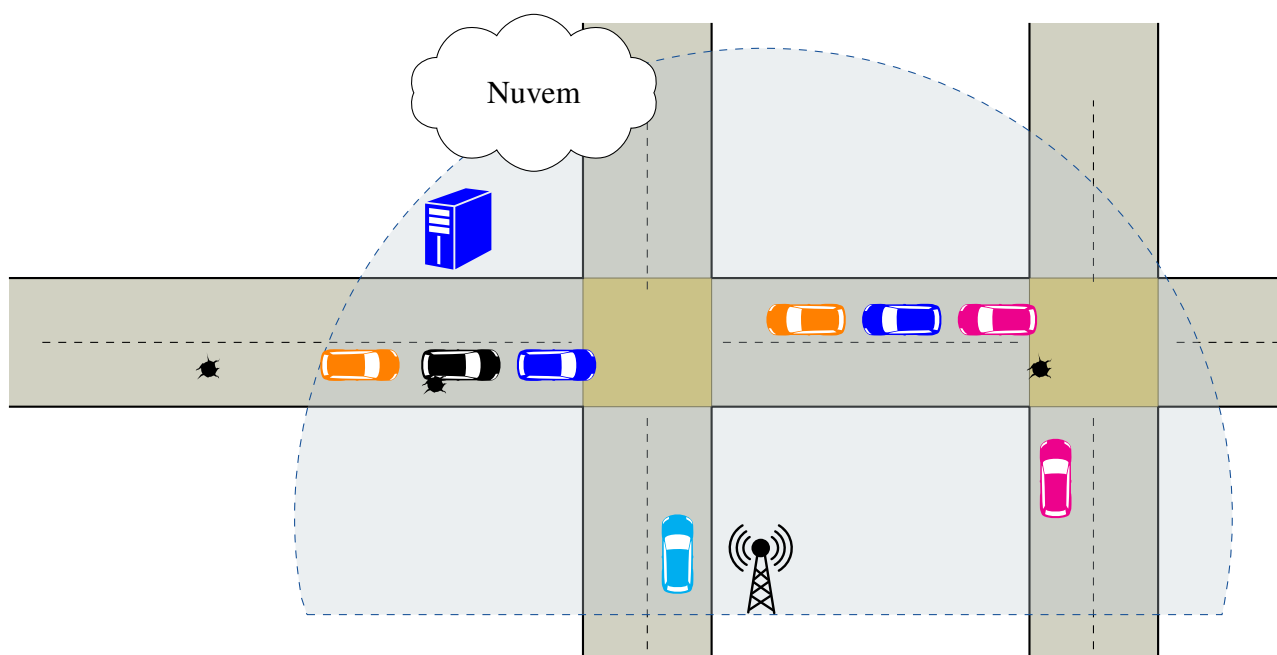


Figura 16 – Cenário de um conjunto de vias com buracos e indicação de buracos.

A precisão e a acurácia são definidas em (TAYLOR, 1997). A acurácia é a quantificação de quão próximas as amostras estão do alvo, que é o valor dito correto. A precisão é a quantificação de quão próximas as amostras replicadas estão umas das outras. A precisão é determinada estatisticamente pelo desvio padrão das diferentes medidas, que traz a informação sobre a diferença entre as medidas. Valores altos para desvio padrão indicam baixa precisão, enquanto que valores baixos indicam alta precisão.

Para determinar se um valor tem acurácia, este precisa ser comparado com o valor dito

correto. Como o valor correto é difícil de ser obtido sem a disponibilidade de um dispositivo de GPS de alta precisão, um conceito chamado **erro percentual** pode ser usado. O erro percentual e é dado por:

$$e = \frac{(a - \hat{a})}{\hat{a}} \cdot 100 \quad (3.3)$$

e indica diferença percentual entre o valor correto *aceito* a e o valor *obtido* \hat{a} no experimento. Para o valor correto *aceito* foi usada a média dos valores obtidos.

A precisão e a acurácia de um sistema podem variar de medição para medição. Existem dois fatores principais pelo qual isso ocorre: limitação dos receptores e interferência nos sinais. A limitação dos receptores ocorre devido a limitações inerentes ao próprio *hardware*. Por exemplo, no caso do GPS, cada satélite que compõe o sistema possui um relógio atômico, que é a forma mais precisa que existe para medir o tempo. Receptores, que possuem relógios atômicos ou outros relógios de alta precisão, podem alcançar milímetros de acurácia. Porém, os relógios encontrados na maioria dos receptores GPS, como os *smartphones*, não possuem essa precisão, principalmente nos receptores de uso civil. *Smartphones* com GPS embutido possuem uma acurácia média de 4,9 m em céu aberto. A acurácia média de dispositivos GPS para aplicações civis é de 12,8 m.

Além da limitação dos receptores, existem fatores que degradam a acurácia do GPS como o sombreamento, que é o bloqueio do sinal por prédios, montanhas, grandes construções, árvores, túneis, etc. Outros fatores que degradam o sinal incluem o uso do GPS em ambientes fechados e o efeito de multipercurso (*multipath*) devido aos efeitos de reflexão e difração que o sinal sofre durante o seu percurso entre o transmissor e o receptor. Outros casos menos comuns que afetam a acurácia são: interferências de outras fontes de rádio, grandes tempestades solares, manutenção dos satélites e falta de padronização: muitos dispositivos receptores não seguem as especificações de interface definidas pelo sistema.

Differential GPS (DGPS) (G. J. MORGAN-OWEN, 1995) é uma técnica que melhora a resolução dos valores computados através do GPS ao adicionar referências baseadas na Terra. A acurácia típica é entre 1–5 m, mas pode chegar a 1 cm nas melhores implementações. Porém, conforme já comentado, os dispositivos móveis disponíveis comercialmente tipicamente não permitem atingir estes valores.

Mais recentemente, *Chen et al.* (CHEN; ZHAO; FARRELL, 2016) desenvolveram uma técnica que maximiza a eficiência do uso de sinais de satélites, considerando algoritmos que reduzem a complexidade dos cálculos e um sensor de inércia acoplado ao dispositivo. Como

resultado, o GPS de um dispositivo móvel pode calcular a posição de um objeto com acurácia de até uma polegada.

Em se tratando de localização *indoor* (HUANG; GARTNER, 2009), recentemente, novas técnicas tem sido propostas (ATIA et al., 2015), (YUDANTO; PETRÉ, 2015), (YANG et al., 2017). Tipicamente, quando voltadas a aplicações com demanda de mobilidade, soluções disponíveis usam o apoio de redes de sensores, o acompanhamento do deslocamento do veículo, e algum processamento para reduzir o *drift*, que é o erro acumulado quando o objeto está em movimento.

3.4 CLASSIFICAÇÃO

Após a etapa de *thresholding*, é feita classificação do sinal como: contém buracos e não contém buracos. Combinando a posição dos buracos detectados no sinal com a informação da localização armazenada na coleta dos dados, sendo ela proveniente do GPS ou do sensor ultrassônico, é possível fazer um mapeamento dos buracos detectados.

A classificação dos buracos pode ser aperfeiçoada com a adição da etapa de pós-processamento no sistema. Usando a integração de múltiplas fontes, é possível implementar técnicas de confirmação e descarte de buracos detectados, seja por uma classificação errônea do sistema ou por conta do buraco detectado não existir mais. Neste trabalho, foi feita uma classificação binária (contém ou não contém) para validar o sistema. O foco do trabalho foi no uso das transformadas wavelet para análise dos dados e a aplicação do *threshold* automatizado.

3.5 AVALIAÇÃO DOS RESULTADOS

Para avaliar os resultados, foram utilizados conceitos de verdadeiros positivos (VP), verdadeiros negativos (VN), falsos positivos (FP) e falsos negativos (FN). São considerados VP quando o algoritmo classifica corretamente a presença de um buraco e VN quando o algoritmo não faz nenhuma detecção em estradas onde não há buracos. FP ocorrem quando o algoritmo aponta a presença de buracos onde não há, e FN quando o algoritmo deixa de apontar a presença de um buraco presente.

Para fins comparativos foram consideradas as seguintes métricas:

- Sensibilidade: a proporção de verdadeiros positivos: a capacidade do sistema em predizer

corretamente a condição para casos que realmente a têm, dada por:

$$\frac{VP}{VP + FN} \quad (3.4)$$

- Especificidade: a proporção de verdadeiros negativos: a capacidade do sistema em prever corretamente a ausência da condição para casos que realmente não a têm, dada por:

$$\frac{VN}{VN + FP} \quad (3.5)$$

- Eficiência, que considera as duas métricas anteriores, e é dada por:

$$\frac{\text{Sensibilidade} + \text{Especificidade}}{2} \quad (3.6)$$

Um algoritmo com sensibilidade de 100%, no escopo deste trabalho, é um algoritmo que consegue classificar todos os buracos por qual o veículo passou. Já a especificidade é capacidade do algoritmo não detectar buracos em regiões que não os possuem. A eficiência é dada pela média entre as duas métricas e quanto mais próximo de 100% melhor a eficiência do algoritmo, ou seja, um algoritmo que detecte todos os buracos presentes e não detecte nenhum buraco que não exista. As métricas e os conceitos utilizados para avaliação podem ser sumarizados pela matriz de confusão representada na Figura 17.

		Valor Verdadeiro		
		Positivo	Negativo	
Valor Previsto	Positivo	Verdadeiro Positivo (VP)	Falso Positivo (FP)	
	Negativo	Falso Negativo (FN)	Verdadeiro Negativo (VN)	
		Sensibilidade VP/(VP + FN)	Especificidade FP/(FP + VN)	Eficácia (Sensibilidade + Especificidade) / 2

Figura 17 – Matriz de confusão.

4 EXPERIMENTOS

Este capítulo descreve os experimentos realizados pra validar a metodologia proposta. Foram realizados dois tipos de experimentos: em cenário sintético, com o uso de um pequeno carro-robô, e em ambiente real, com o uso de um veículo real. No texto que segue, os experimentos serão detalhados.

4.1 EXPERIMENTO EM AMBIENTE CONTROLADO

Para detectar buracos em estradas, experimentos em cenário real requerem uma logística muito custosa do ponto de vista material e de mão de obra. É necessário um veículo, uma rua ou estrada que contenham buracos e que o veículo passe necessariamente pelos buracos, o que pode causar danos aos seus componentes. Além disso, para validar os experimentos é necessário que o processo se repita diversas vezes para obter uma quantidade de dados relevante.

Pensando nisso, foram elaborados testes em ambiente sintético, usando componentes que simulam os componentes de um cenário real. Para isso, foi usado um carro-robô de tamanho reduzido, com uma placa microcontroladora, programado para passar por uma pista sintética de EVA (*AVE* ou *Acetato-Vinilo de Etileno*). A pista contém buracos que serão detectados por um dispositivo equipado com acelerômetro que estará acoplado ao carro-robô.

4.1.1 Configuração do experimento

O carro-robô utilizado, apresentado na Figura 18, foi montado na plataforma de prototipagem Arduino. O carro possui uma placa microcontroladora Arduino Mega 2560, chassi com quatro rodas (duas rodas dianteiras e duas rodas traseiras), quatro motores individuais para as rodas, controlador para cada motor e sensores infravermelhos.

O carro foi programado usando a linguagem C++ na plataforma Arduino, para seguir um trajeto delimitado na pista sintética de EVA. O veículo tem as dimensões de 20cm×20cm. As dimensões das rodas do carro são de 2cm de largura e 3cm de raio cada.

O trajeto do carro-robô foi demarcado por uma fita preta na pista de EVA. Os sensores infravermelhos instalados embaixo do veículo conseguem distinguir a pista da fita pelo contraste de cores entre elas, assim os motores das rodas podem ser controlados individualmente, fazendo com o que o veículo siga o trajeto delimitado. A Figura 19 mostra o carro-robô sobre

a pista montada. Os parâmetros da pista usada neste experimento e os respectivos valores são fornecidos na Tabela 1. Os buracos foram feitos manualmente para simular os buracos reais.

Tabela 1 – Valores relativos à pista usada nos experimentos no cenário controlado.

Parâmetro	Valor (cm)
Tamanho total da pista	400
Trajeto percorrido pelo carro-robô	360
Número de buracos	2
Tamanho do buraco	$4 \times 6 \times 1$
Posição do buraco 1 em relação ao início do trajeto	70
Posição do buraco 2 em relação ao início do trajeto	302

Para a coleta de dados, foi implementada uma aplicação Arduino que executa em um *hardware* adicional, citado na Seção 3.1. Para isso foi utilizado uma placa Arduino 2560, um sensor MPU-6050 com acelerômetro de três eixos embutido, um sensor ultrassônico HC-SR04 e um adaptador de cartão SD com um cartão SD de com 4gb de memória para armazenar os dados. O *hardware* foi posicionado em cima do chassi do carro. O sensor ultrassônico foi usado para marcar a posição exata do buraco juntamente com o sinal do acelerômetro. O sensor foi instalado na lateral do veículo e mede continuamente a distância entre ele e objetos a sua frente, nesse caso, objetos que estão do lado do veículo. No experimento foram utilizados marcadores (caixas de papelão) do lado dos buracos da pista sintética, assim, quando o veículo passar pelos buracos o sensor ultrassônico irá detectar os marcadores enquanto o acelerômetro irá detectar o buraco. Para cada leitura do acelerômetro é armazenada uma leitura do sensor ultrassônico indicando a presença ou não de buracos.

Os dados gerados pelos sensores são armazenados em um arquivo do tipo CSV. Cada linha do arquivo contém uma leitura dos sensores, formatadas como “*eixoZ, buraco*“, onde *eixoZ* é a leitura obtida pelo acelerômetro em *g*, e *buraco* é uma variável que indica se um marcador foi detectado pelo sensor ultrassônico e pode assumir o valor de 0 quando não tem marcador e 1 quando tem.

Para validar o procedimento de detecção proposto no cenário sintético, o experimento foi repetido 20 vezes e, portanto, os dados de 20 voltas diferentes foram armazenados e analisados. O carro-robô leva aproximadamente 9 segundos para percorrer o caminho de 3,6 metros da pista sintética, isso equivale a uma velocidade média de $\approx 0,40$ m/s.

Ainda sobre os detalhes da implementação, referente a coleta de dados, a placa Arduino executa uma aplicação que permite coletar informações do acelerômetro em diferentes taxas

de amostragem por segundo. A taxa escolhida para este experimento foi de 50 amostras por segundo, que de acordo com a velocidade do carro e o comprimento da pista, equivale a 1,25 leituras por centímetro percorrido, valor suficientemente alto para detectar os buracos.

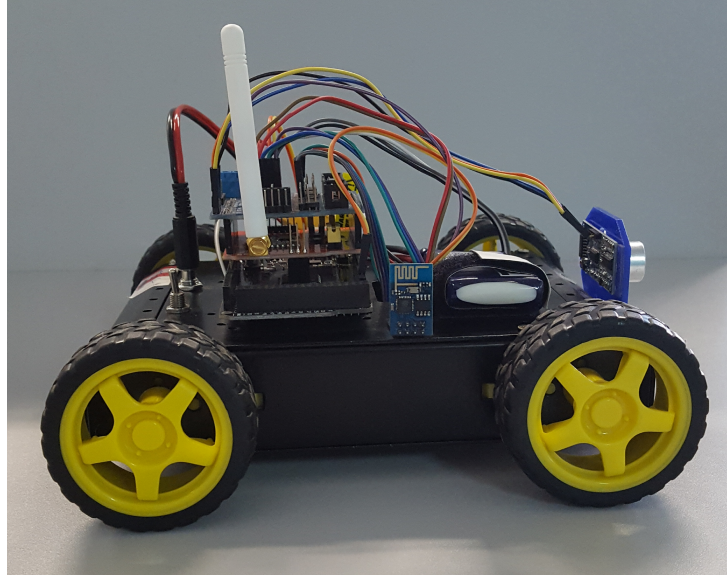


Figura 18 – Foto do carro-robô Arduino usado no experimento com a pista sintética.

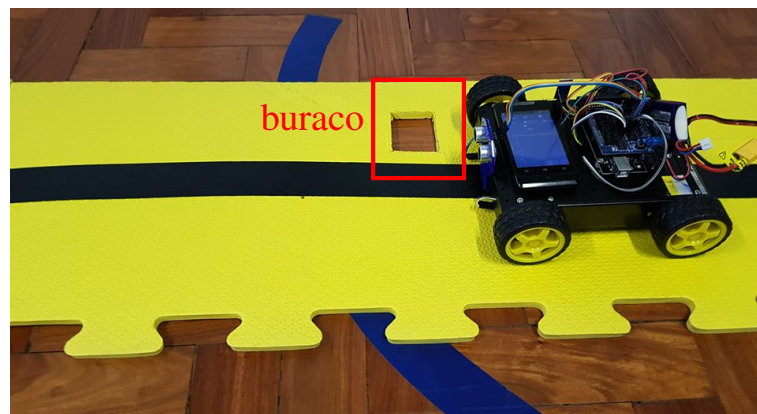


Figura 19 – Foto do carro-robô Arduino sobre um segmento da pista sintética contendo um buraco.

4.1.2 Processamento e resultados

A etapa de processamento consiste em aplicar os algoritmos de TWH nos dados coletados na etapa anterior. Neste experimento, foi aplicado o algoritmo cascata da TWH, visto na Seção 2.3, como prova de conceito. Em seguida da aplicação do algoritmo, os *thresholds* foram calculados e aplicados sobre os sinais resultantes.

Os dados coletados (originais) que foram usados como entrada são representados por C_0 . Aplicando um nível do algoritmo cascata da TWH, C_0 é decomposto nos vetores de médias C_1 e detalhes D_1 . Para esta análise, foi considerado o primeiro nível de detalhes do sinal, o qual contém as informações de alta frequência do sinal original e que são mais provavelmente associadas a fortes variações nos dados.

Os *thresholds* foram calculados usando as Equações 3.1 e 3.2, vistas na Seção 3.2.2. Para as análises feitas ao decorrer deste capítulo, será utilizada a denominação λ_{sup} e λ_{inf} para os *thresholds* superiores e inferiores, respectivamente. Os experimentos foram feitos com o valor do *fator* sendo 4, que garante que 99,994% de uma variável aleatória medida esteja dentro entre λ_{sup} e λ_{inf} .

Para cada volta percorrida pelo carro-robô na pista sintética foi criado um arquivo contendo o sinal C_0 e a localização dos buracos. Esses sinais serviram de entrada para o algoritmo cascata, que foram implementados usando a linguagem de programação Python⁵. Para a análise do sinal, foram gerados gráficos do sinais D_1 com o auxílio da biblioteca gráfica Matplotlib⁶, que assim como o Python, são ferramentas gratuitas.

Junto com o sinal de detalhes D_1 , foram plotados os delimitadores λ_{sup} e λ_{inf} e as posições dos buracos no sinal. A Figura 20 mostra o gráfico da primeira volta percorrida pelo veículo gerado pelo algoritmo. Os buracos são representados por b_{ic} e b_{if} , onde c marca o começo e f o fim do buraco i . É possível perceber pelo gráfico que em dois momentos o sinal extrapola o valor de λ_{sup} , e nos dois momentos o sinal está dentro dos intervalos que marcam a posição do buraco. Dessa forma, é possível garantir que esses dois eventos que extrapolaram o *threshold* foram causados pelos buracos na pista.

⁵ <http://python.org>

⁶ <https://matplotlib.org/>

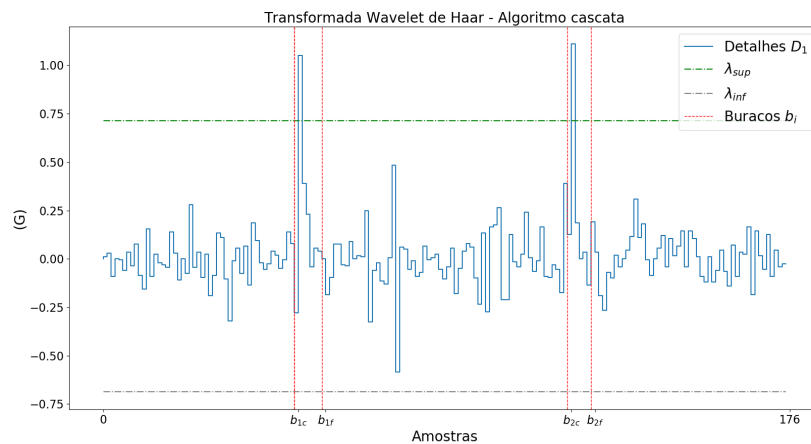


Figura 20 – *Thresholds* λ_{sup} e λ_{inf} aplicados aos detalhes D_1 da volta 1 usando a TWH algoritmo cascata

Finalmente, a Figura 21 mostra os sinais complementares para 20 voltas percorridas pelo carro-robô. Como pode ser observado na figura, todos os eventos detectados que extrapolaram os valores de λ_{sup} e λ_{inf} ocorreram dentro dos intervalos definidos como os locais de buracos, indicando a eficiência do procedimento.

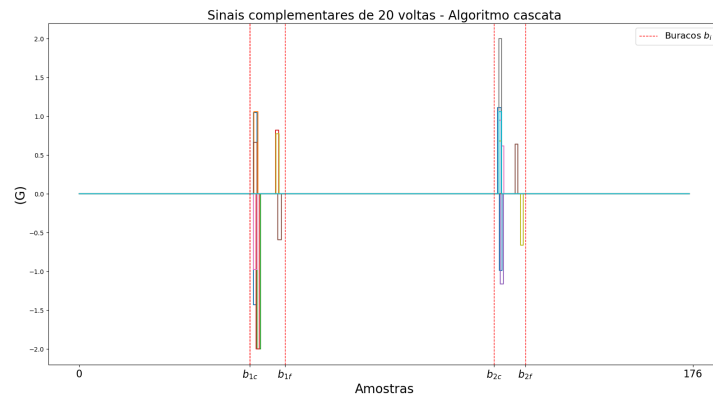


Figura 21 – Sinais complementares de 20 voltas após o truncamento dos sinais D_1 no cenário com pista sintética.

O objetivo do experimento em ambiente sintético foi validar a abordagem usando *thresholds* adaptativos aplicados em sinais processados por transformadas wavelet. A Tabela 2 mostra os diferentes valores de λ_{sup} e λ_{inf} calculados para cada volta sobre os sinais D_1 obtidos através da THW algoritmo cascata. Usando esses valores como delimitadores, os buracos foram detectados em todas as voltas percorridas pelo carro-robô.

Tabela 2 – Valores de λ_{sup} e λ_{inf} para as 20 voltas percorridas.

Volta	$\approx \lambda_{sup}$	$\approx \lambda_{inf}$
1	0.71448	-0.68784
2	0.73989	-0.81408
3	0.69417	-0.67428
4	0.84414	-0.87010
5	0.76274	-0.80520
6	0.60024	-0.63794
7	0.51682	-0.52606
8	0.61728	-0.60340
9	0.82892	-0.78163
10	0.61348	-0.63656
11	0.62516	-0.60283
12	0.83662	-0.84645
13	0.75495	-0.73495
14	0.82607	-0.84042
15	0.75321	-0.72981
16	0.74942	-0.76525
17	0.88971	-0.89230
18	1.06251	-1.02858
19	0.93005	-0.89011
20	0.59693	-0.59068

4.1.3 Conclusões parciais

De forma geral, o experimento com a pista sintética permitiu comprovar que usando *thresholds* adaptativos em sinais processados é possível realizar a detecção de buracos. A utilização do sensores ultrassônico para marcação da localização dos buracos foi essencial para analisar a presença dos buracos sem inferir os resultados desejados.

4.2 EXPERIMENTO EM CENÁRIO REAL

Embora o ambiente controlado facilite a coleta de dados e permita variar os cenários dos experimentos, o sinal coletado não é igual ao sinal obtido em um cenário real. Isso pode acontecer por vários motivos, sendo o principal deles a diferença de escala entre os dois cenários. No cenário de trilha sintética, qualquer vibração causada pela junção dos materiais ou algum ruído é muito mais significativa do que no cenário real. Outra questão é a falta de um sistema de suspensão no carro-robô. Como o carro-robô não tem sistema de suspensão, os sinais coletados pelo acelerômetro tendem a ser mais evidentes. Desta forma, também foram executados experimentos em um cenário real, para verificar a aplicabilidade da metodologia aqui definida.

4.2.1 Configuração do experimento

O experimento em cenário real consistiu em dirigir um veículo por um trecho da estrada que passa por um viaduto existente dentro do nosso campus universitário, onde anteriormente se sabia que havia um conjunto de buracos. Este cenário é apresentado esquematicamente na Figura 22, com indicação da origem O e um destino D . Uma foto real do viaduto é apresentada na Figura 23. Um esboço do viaduto e seus buracos é ilustrado na Figura 24, destacando em especial o buraco que será considerado para detecção. Uma foto indicando a aparência do buraco em questão é apresentada na Figura 25. No total, a pista do cenário real tem extensão de 100,1m. A Tabela 3 sumariza os valores relativos aos elementos do cenário real.

Tabela 3 – Valores relativos aos elementos do cenário real.

Parâmetro	Valor
Tamanho total da pista	100,1
Número de buracos maiores	1
Tamanho do buraco	45cm×45cm × 15cm
Posição do buraco em relação ao início do trajeto	64,8m
Velocidade média do veículo	30km/h

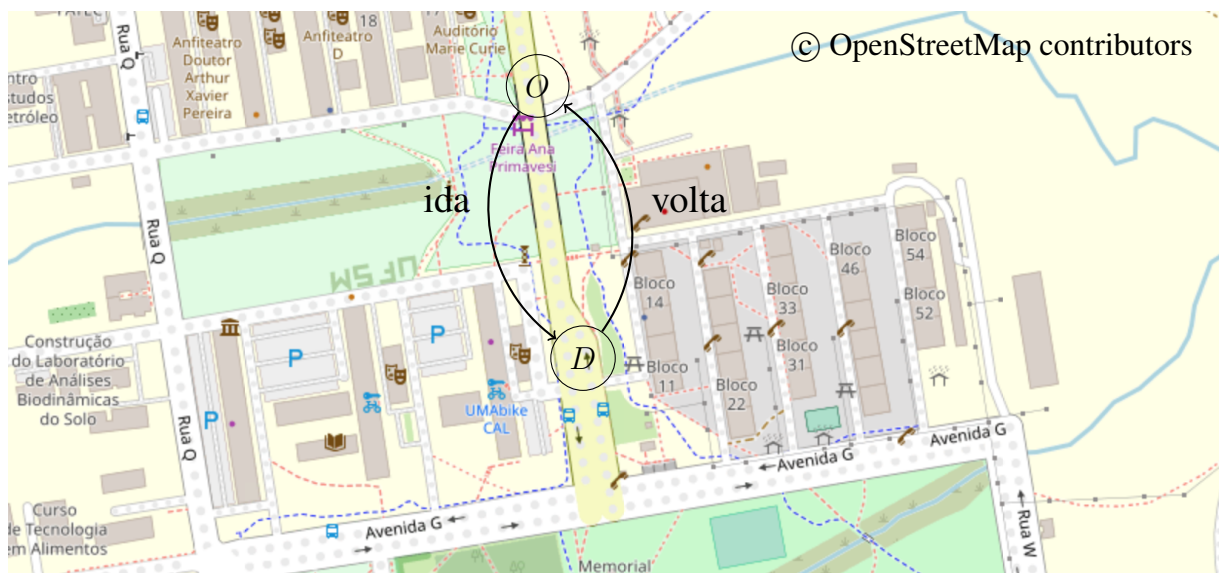


Figura 22 – Mapa da UFSM com indicação da localização do viaduto.

4.2.2 Coleta de dados

Para realizar os experimentos, na fase de coleta de dados, foi utilizado um *smartphone* modelo Samsung Galaxy S8, com acelerômetro embutido. O dados foram coletados com taxa



Figura 23 – Visão geral do viaduto interno da UFSM.

de armazenamento de 50 amostras por segundo. Para tanto, foi utilizado um veículo modelo Toyota Etios Hatchback modelo 2019. O *smartphone* foi posicionado no meio do painel do veículo. A gravação dos dados foi realizada por meio do aplicativo Accelerometer Analyzer para o sistema operacional Android 8.0, disponível na loja virtual Google Play. O aplicativo permite a gravação de dados do acelerômetro integrado ao *smartphone* em um arquivo no próprio *smartphone*.

Durante a coleta de dados para o experimento no cenário real, o veículo manteve velocidade média de 30 km/h. Para o processo de registro, os dados começaram a ser gravados alguns metros antes do veículo iniciar a passagem pelo viaduto e a gravação foi finalizada alguns metros após o término da passagem do veículo pelo viaduto.

Foram realizadas 39 voltas no cenário apresentado na Figura 22, 20 voltas no sentido *O-D* e 19 no sentido *D-0*. O buraco em questão a ser detectado está localizado no sentido *O-D*. Das 20 voltas realizadas nesse sentido, o veículo passou pelo buraco em 12 delas, desviando nas outras 8. Isso foi feito porque em um cenário real alguns motoristas podem desviar dos buracos.

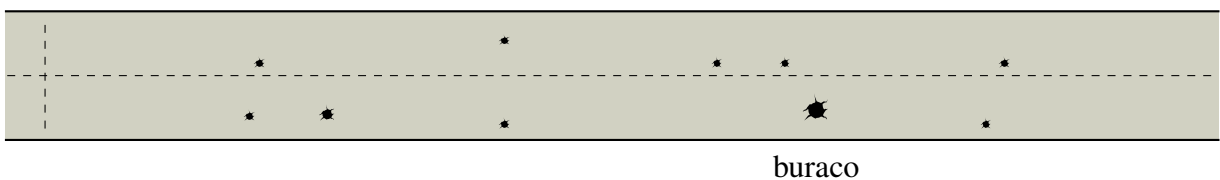


Figura 24 – Esquema do cenário real onde os experimentos foram conduzidos com a indicação da localização do buraco.



Figura 25 – Buraco no asfalto no viaduto da UFSM e detectado por λ_{sup} e λ_{inf} durante o experimento neste cenário.

4.2.3 Processamento e resultados

Os dados das 39 voltas obtidas na etapa de coletas foram processados pelos algoritmos cascata, à trous e à trous de 2 passos, descritos na Seção 3.2.1. Os *thresholds* λ_{sup} e λ_{inf} foram calculados usando o fator de multiplicação 4.

Assim como nos experimentos em cenário sintético, foram produzidos gráficos para cada volta do veículo na pista. Para exemplificar, as Figuras 26, 27 e 28 mostram os gráficos gerados pela volta 25 do experimento, onde os três algoritmos detectaram o buraco em questão. Os sinais complementares foram utilizados para identificar de forma mais clara a posição onde o sinal extrapolou os valores dos *thresholds*.

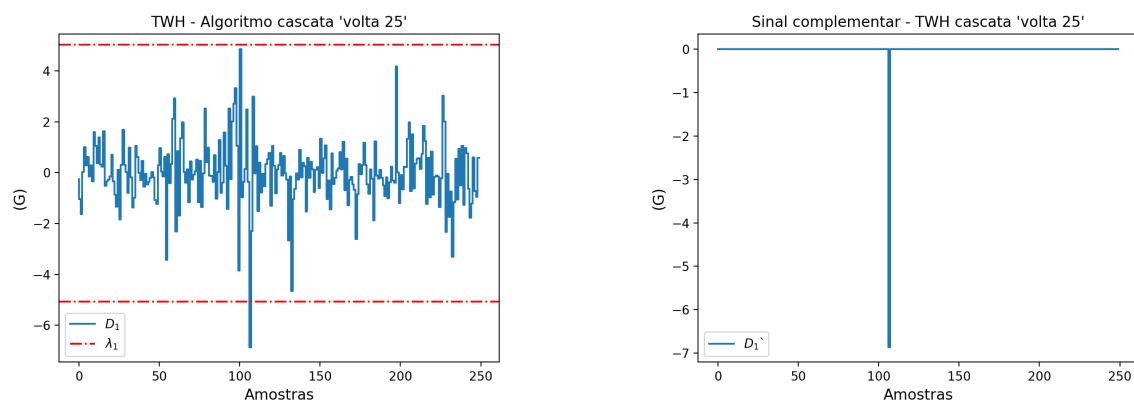


Figura 26 – Sinais de detalhes D_1 e complementar D_1' da volta 25 obtidos pelo algoritmo cascata.

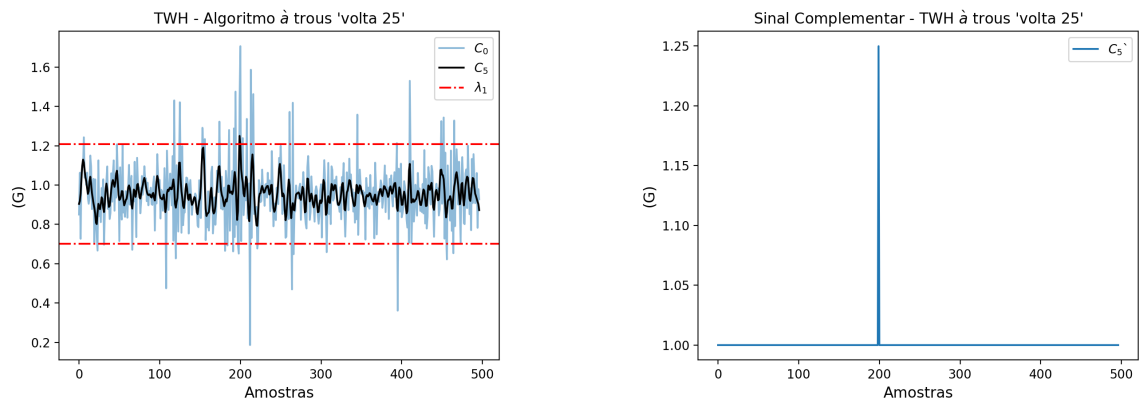


Figura 27 – Sinal original C_0 , médias C_5 e complementar C_5^c da volta 25 obtidos pelo algoritmo à trous.

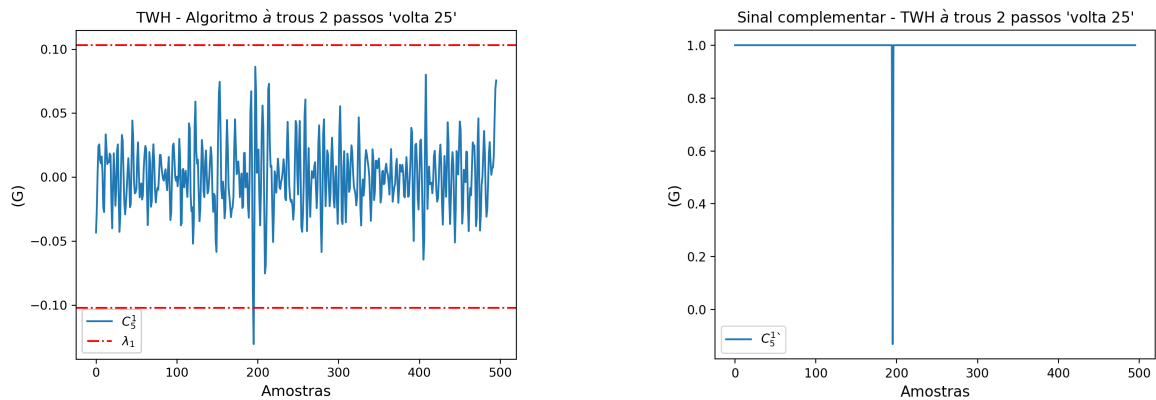


Figura 28 – Sinal resultante C_5^1 e complementar $C_5^1^c$ da volta 25 obtidos pelo algoritmo à trous passo 2.

A Tabela 4 mostra as voltas que o veículo passou pelo buraco, que são representadas pelo marcador ✓ na coluna *Gabarito*. As demais colunas representam em quais voltas os algoritmos aplicados detectaram o buraco.

Tabela 4 – Gabarito dos experimentos.

Voltas	Gabarito	À trous	À trous 2 passos	Cascata
1				
2	✓	✓	✓	✓
3	✓	✓		✓
4				✓
5				✓
6	✓	✓	✓	✓
7	✓	✓	✓	✓
8	✓	✓	✓	✓
9			✓	
10				✓
11	✓	✓	✓	✓
12				✓
13			✓	
14	✓	✓	✓	✓
15	✓		✓	✓
16				
17				✓
18	✓	✓	✓	✓
19				✓
20	✓	✓	✓	✓
21				✓
22				✓
23	✓	✓	✓	✓
24				
25	✓	✓	✓	✓
26				
27				✓
28				✓
29				✓
30				✓
31				✓
32				✓
33				
34				✓
35			✓	
36			✓	
37				
38			✓	✓
39				

4.2.4 Avaliação dos resultados

Para avaliar os resultados obtidos pelos algoritmos implementados, foram utilizadas as métricas descritas na Seção 3.5. Os gráficos representados nas Figuras 29, 30, 31 e 32 mostram os valores de VP, VN, FP e FN de cada algoritmo.

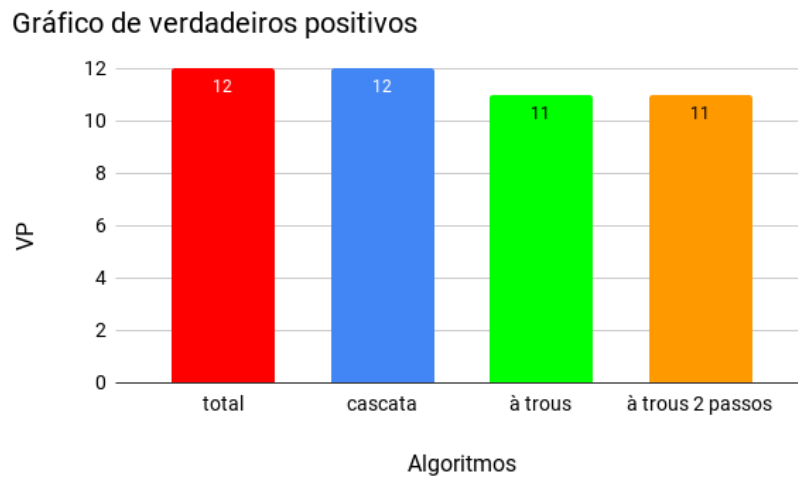


Figura 29 – Gráfico de verdadeiros positivos (VP) dos resultados obtidos no experimento em ambiente real.

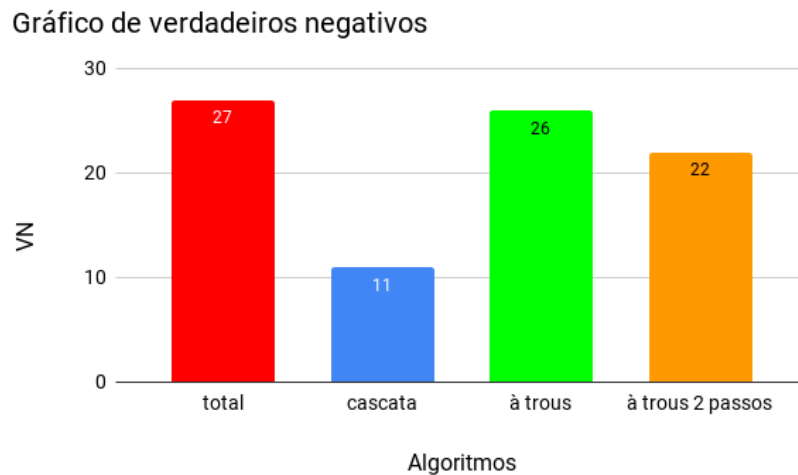


Figura 30 – Gráfico de verdadeiros negativos (VN) dos resultados obtidos no experimento em ambiente real.

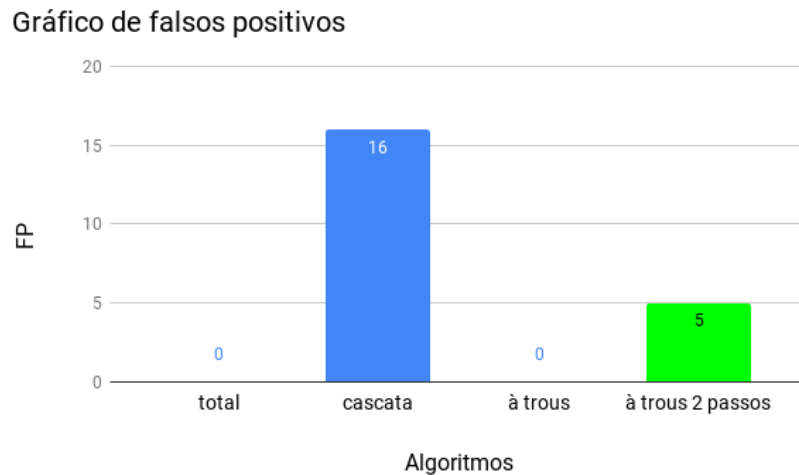


Figura 31 – Gráfico de falsos positivos (FP) dos resultados obtidos no experimento em ambiente real.

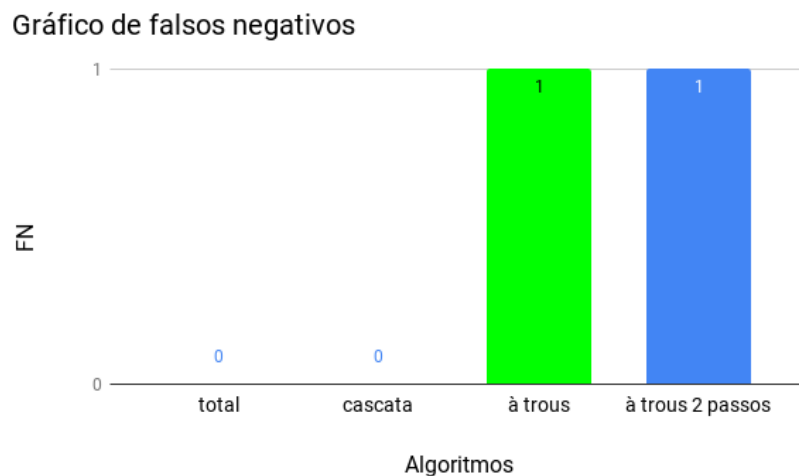


Figura 32 – Gráfico de falsos negativos (FN) dos resultados obtidos no experimento em ambiente real.

A Tabela 5 mostra os resultados obtidos pela análise dos algoritmos implementados segundo as métricas avaliadas. É válido notar que o algoritmo cascata teve uma taxa de acertos de 100% na classificação dos buracos. Porém, o número de FP classificados foi muito alto, o que indica que esse algoritmo é sensível a perturbações menores no sinal, que não necessariamente são causadas por buracos. Isso pode ser observado pela sensibilidade medida de 100% do algoritmo e a especificidade de 69,64%.

Como pode ser observado, o algoritmo à trous foi o que obteve a maior taxa de eficiência entre os algoritmos testados, seguido pelo algoritmo à trous de dois passos. Em ambos os casos, em apenas 1 das 12 voltas que o veículo passou pelo buraco, o mesmo não foi detectado. Apesar

disso, a precisão e a especificidade do algoritmo à trous foi de 100%, isso quer dizer que o algoritmo não classificou nenhum buraco não existente. O algoritmo à trous de dois passos obteve a mesma sensibilidade do algoritmo à trous tradicional porém, com uma especificidade e conseqüentemente uma eficiência menor.

Era esperado que o algoritmo à trous de dois passos tivesse o melhor desempenho entre os aplicados. No entanto, foi notado que a especificidade do mesmo não obteve o resultado esperado. Isso pode ter ocorrido pelo fato de a transformação feita no sinal ter tornado-o muito sensível à falsos positivos. Um estudo mais aprofundado no algoritmo é necessário para que os parâmetros do mesmo possam ser ajustados para um desempenho melhor.

Tabela 5 – Resultados para os experimentos no cenário real.

Métrica	à trous	à trous 2 passos	Cascata
Acertos	11/12	11/12	12/12
Taxa de acertos	92%	92%	100%
VP	11	11	12
VN	26	22	11
FP	0	5	16
FN	1	1	0
Acurácia	94,88%	87%	58,97%
Precisão	100%	73,33%	42,85%
Sensibilidade	91,66%	91,66%	100%
Especificidade	100%	85,18%	69,64%
Eficiência	95,83%	88,42%	69,64%

4.2.5 Limitações do experimento

Os experimentos em ambiente real são desafiadores por diversos motivos. O principal deles é a dificuldade de se obter os dados de forma segura. Para este tipo de experimento, é necessário que o veículo passe deliberadamente por buracos em estradas, o que pode causar danos aos componentes do veículo e até mesmo causar algum acidente decorrente dessa prática. Por este motivo, não é prudente realizar essa coleta em vias de alta velocidade e fluxo de veículos ou pedestres constantes.

Outra limitação é a falta de disponibilidade de veículos para os testes, tanto pelo custo dos mesmos quanto pelo possível dano causado aos seus componentes. O experimento foi realizado com a utilização de apenas um veículo, descrito no início da Seção.

5 CONSIDERAÇÕES FINAIS

5.1 CONCLUSÕES

Este trabalho apresentou uma nova metodologia baseada em vibração para detecção de buracos usando o suporte da Transformada Wavelet de Haar para lidar com sinais de acelerômetros. As wavelets são uma classe de funções usadas para decompor dados na posição e na escala. Essa habilidade transforma as wavelets em ferramentas poderosas no processamento de sinais e imagens, permitindo a associação de faixas de frequência a níveis específicos de resolução da decomposição.

A análise dos coeficientes wavelet obtidos pelo TWH através de um procedimento limiar permite a identificação de fortes variações nos dados. Este tipo de operador de truncamento é amplamente utilizado para a remoção ou compressão de dados, uma vez que permitem a redução de dados em regiões de pequenas variações, mantendo a precisão dos dados. Aqui, o mecanismo de truncagem é considerado para identificar fortes variações nos dados, que são causados pelos buracos, sendo, portanto, a parte principal do nosso método de detecção. Os limiares utilizados comumente são estáticos e definidos manualmente, o que limita o uso para cenários específicos e desconsidera variáveis que alteram os dados de caso para casa. O limiar utilizado neste trabalho é adaptativo, onde não é necessário a calibração manual do mesmo e leva em consideração os dados independentemente de mudanças no ambiente e nos equipamentos usados para a coleta.

Os resultados obtidos em experimentos confirmam o potencial do procedimento de limiar adaptativo na detecção de buracos. Foram testados diferentes algoritmos de transformada wavelet, que foram avaliados em relação as métricas de sensibilidade, especificidade e eficácia. O algoritmo à trous da TWH obteve resultados satisfatórios para a detecção de buracos, obtendo uma eficácia com aproximadamente 96% e uma taxa de especificidade de 100%, o que indica que o algoritmo não classificou nenhum FP.

Esta metodologia possui limitações inerentes à detecção de buracos usando sensores, especificamente, o acelerômetro. Essa abordagem leva em consideração que o veículo passou por um buraco, gerando uma vibração anormal captada pelo acelerômetro e não é possível garantir que o motorista não irá desviar do buraco. Além disso, outros fatores que causam vibrações anormais em estradas precisam ser considerados. Existem soluções que podem ser utilizadas para resolver este problema, como uso de uma lista negra, mencionada neste trabalho.

5.2 TRABALHOS FUTUROS

Para trabalhos futuros, uma extensão natural da metodologia proposta é a sua formulação on-line, uma vez que tanto a TWH quanto a operação de *thresholds* são procedimentos de baixo custo computacional, dependendo apenas dos dados locais. Outra contribuição em off-line analisada é a possibilidade de explorar a natureza bidimensional do problema, seguindo uma abordagem inspirada na proposta em (KOZAKEVICIUS et al., 2015). As muitas voltas consideradas na fase de treinamento poderiam ser analisadas por uma TWH 2D, incorporando informações correlacionadas ao sistema de detecção de buracos. Nesta proposta, voltas diferentes são tratadas como eventos independentes.

Outra possibilidade de trabalhos futuros poderia ser uma mistura do que é proposto aqui com uma heurística de detecção *fuzzy*, como projetado em (WANG et al., 2017). Um classificador poderia, através da associação de dados coletados em diferentes jornadas, melhorar a qualidade da solução. Conforme anteriormente reportado, isso demonstra a necessidade de avaliar a detecção de buracos em um cenário 2D, incorporando novas informações ao sistema de detecção de buracos.

Finalmente, os experimentos aqui realizados foram preliminares. Outros experimentos, com um número maior de buracos e diferentes veículos (com diferentes sistemas de amortecimentos) ainda precisam ser realizados para validar a aplicabilidade real da solução aqui apresentada.

5.3 PUBLICAÇÕES

Artigo aceito em congresso internacional:

- Ricardo Silveira Rodrigues, Marcia Pasin, Alice Kozakevicius, e Vinicius Monego. “Pothole Detection in Asphalt: an Automated Approach to Threshold Computation based on the Haar Wavelet Transform.” *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, Milwaukee USA, July 2019 (Qualis A2).

Publicações durante o período do curso de Mestrado:

- Marcia Pasin, Ricardo Silveira Rodrigues, Leonardo de Abreu Schmidt, e Renato Machado. “Avaliação Experimental da Acurácia e da Precisão de Tecnologias de Comunicação visando Auto-localização em Redes Veiculares.” *VIII Brazilian Symposium on*

Computing Systems Engineering (SBESC 2018), November 2018 - Salvador - Bahia - Brazil (Qualis B2).

- Ricardo Silveira Rodrigues, Marcia Pasin, e Renato Machado. “Indoor Position Tracking: An Application using the Arduino Mobile platform.” *10th IFIP Wireless and Mobile Networking Conference* (WMNC 2017), Valencia, Spain, September 2017, pp. 1–8. (resultado do trabalho de conclusão de Curso de Graduação, Qualis B1)
- Ricardo Silveira Rodrigues, Marcia Pasin. “Avaliação de Desempenho de Tecnologias de Comunicação na Plataforma Arduino no Contexto de VANETs.” *XVIII Escola Regional de Alto Desempenho - Região Sul* (ERAD 2018), Porto Alegre - RS, Abril 2018, pp. 219-220.

REFERÊNCIAS

- ACTION, P. **Distribuição Normal**. Accessed: 2018-08-06, <http://www.portaaction.com.br/probabilidades/62-distribuicao-normal>.
- ALLOUCH, A. et al. RoadSense: smartphone application to estimate road conditions using accelerometer and gyroscope. **IEEE Sensors Journal**, [S.l.], n.10.1109/JSEN.2017.2702739, 2017.
- ATIA, M. M. et al. Integrated Indoor Navigation System for Ground Vehicles With Automatic 3-D Alignment and Position Initialization. **IEEE Transactions on Vehicular Technology**, [S.l.], v.64, n.4, p.1279–1292, April 2015.
- AZHAR, K. et al. Computer vision based detection and localization of potholes in asphalt pavement images. In: IEEE CANADIAN CONFERENCE ON ELECTRICAL AND COMPUTER ENGINEERING (CCECE), 2016. **Anais...** [S.l.: s.n.], 2016. p.1–5.
- BURKE, J. A. et al. Participatory Sensing. **Workshop on World-Sensor-Web (WSW): Mobile Device Centric Sensor Networks and Applications**, [S.l.], 01 2006.
- CHEN, Y.; ZHAO, S.; FARRELL, J. A. Computationally efficient carrier integer ambiguity resolution in multiepoch GPS/INS: a common-position-shift approach. **IEEE Transactions on Control Systems Technology**, [S.l.], v.24, p.1541–1556, 2016.
- COMMISSION, E. **Mobility and Transport ROAD SAFETY IN THE EUROPEAN UNION: trends, statistics and main challenges**. [S.l.]: Luxembourg: Publications Office of the European Union, 2018.
- DAUBECHIES, I. **Ten Lectures on Wavelets**. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1992.
- DONOHO, D. L.; JOHNSTONE, I. M. Adapting to unknown smoothness via wavelet shrinkage. In: JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION. **Anais...** Taylor Francis: Ltd., 1995. v.90, n.432, p.1200–1224.
- ENIGOA, V. S. F. et al. CrowdSourcing based online petitioning system for pothole detection using Android platform. In: FOURTH INTERNATIONAL CONFERENCE ON RE-

CENT TRENDS IN COMPUTER SCIENCE & ENGINEERING (ICRTCSE 2016), *PROCEEDIA COMPUTER SCIENCE*, VOL. 87. **Anais...** [S.l.: s.n.], 2016. p.316–321.

ERIKSSON, J. et al. The Pothole Patrol: using a mobile sensor network for road surface monitoring. In: INTERNATIONAL CONFERENCE ON MOBILE SYSTEMS, APPLICATIONS, AND SERVICES, 6., New York, NY, USA. **Proceedings...** ACM, 2008. p.29–39. (MobiSys '08).

FOX, A. et al. Multi-Lane Pothole Detection from Crowdsourced Undersampled Vehicle Sensor Data. **IEEE Transactions on Mobile Computing**, [S.l.], v.16, p.3417–3430, 2017.

G. J. MORGAN-OWEN, G. T. J. Differential GPS positioning. **Electronics & Communication Engineering Journal**, [S.l.], p.11–21, 1995.

GLEAVE, S. D. et al. **EU Road Surfaces**: economic and safety impact of the lack of regular road maintenance: study. [S.l.]: Directorate General for Internal Policies, Policy Department B: Structural and Cohesion Policies, Transport and Tourism, European Union, 2014.

HUANG, H.; GARTNER, G. A Survey of Mobile Indoor Navigation Systems. **Cartography in Central and Eastern Europe. Lecture Notes in Geoinformation and Cartography**. In: **Gartner G., Ortog F. (eds), Springer, Berlin, Heidelberg**, [S.l.], 2009.

JOHN WILEY SONS, L. . [S.l.: s.n.], 2015. 97-130p.

KANG, B.; CHOI, S. Pothole detection system using 2D LiDAR and camera. In: INT. CONFERENCE ON UBIQUITOUS AND FUTURE NETWORKS (ICUFN), 2017. **Anais...** [S.l.: s.n.], 2017. p.744–746.

KIM, T.; RYU, S. Review and analysis of pothole detection methods. **Journal of Emerging Trends in Computing and Information Sciences**, [S.l.], v.5, n.8, 2014.

KOZAKEVICIUS, A. et al. URL Query String Anomaly Sensor Designed with the Bidimensional Haar Wavelet Transform. **Int. J. Inf. Secur.**, [S.l.], v.14, n.6, p.561–581, November 2015.

KOZAKEVICIUS, A.; SCHMIDT, A. A. Wavelet transform with special boundary treatment for 1D data. **Computer and Applied Mathematics -CAM.**, [S.l.], June 2013.

LIN, J.; LIU, Y. Potholes Detection Based on SVM in the Pavement Distress Image. In: NINTH INTERNATIONAL SYMPOSIUM ON DISTRIBUTED COMPUTING AND APPLICATIONS TO BUSINESS, ENGINEERING AND SCIENCE, 2010. **Anais...** [S.l.: s.n.], 2010. p.544–547.

MADLI, R. et al. Automatic Detection and Notification of Potholes and Humps on Roads to Aid Drivers. **IEEE Sensors Journal**, [S.l.], v.15, n.8, p.4313–4318, Aug 2015.

MALLAT, S. G. A Theory for Multiresolution Signal Decomposition: the wavelet representation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, [S.l.], v.II, n.7, p.674–693, 1989.

MEDNIS, A. et al. Real time pothole detection using Android smartphones with accelerometers. In: INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING IN SENSOR SYSTEMS AND WORKSHOPS (DCOSS), 2011. **Anais...** [S.l.: s.n.], 2011. p.1–6.

MISRA, P.; ENGE, P. **Global Positioning System: signals, measurements and performance**. Lincoln MA, USA: 2nd edition, Ganga-Jamuna Press, 2006.

MOHAN, P.; PADMANABHAN, V. N.; RAMJEE, R. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In: ACM CONFERENCE ON EMBEDDED NETWORK SENSOR SYSTEMS, 6., New York, NY, USA. **Proceedings...** ACM, 2008. p.323–336. (SenSys '08).

P. M. BENTLEY, J. T. E. M. Wavelet transforms: an introduction. **Electronics Communication Engineering Journal**, [S.l.], v.6, n.4, p.175–186, Aug 1994.

PARKINSON, B.; SPILKER, J. **Global Positioning System: theory and applications**. Reston, VA, USA: American Institute of Aeronautics and Astronautics, 1996.

PATEL, J.; READ, C. **Handbook of the Normal Distribution**. [S.l.]: M. Dekker, 1982. (Statistics, textbooks and monographs).

PEI, S.-C.; YEH, M.-H.; LUO, T.-L. Fractional Fourier series expansion for finite signals and dual extension to discrete-time fractional Fourier transform. **IEEE Transactions on Signal Processing**, [S.l.], v.47, n.10, p.2883–2888, Oct 1999.

- RODE, S. S. et al. Pothole Detection and Warning System: infrastructure support and system design. In: INTERNATIONAL CONFERENCE ON ELECTRONIC COMPUTER TECHNOLOGY, 2009. **Anais...** [S.l.: s.n.], 2009. p.286–290.
- SANTOS, V. R. dos et al. **Wavelets: principles, analysis and applications**. chapter 3: int-hwt: interval extensions and optimizations to increase performance and accuracy of haar wavelet transforms. [S.l.]: Nova Publisher, New York, 2018.
- SATTAR, S.; LI, S.; CHAPMAN, M. Road Surface Monitoring Using Smartphone Sensors: a review. **Sensors (Basel)**, [S.l.], v.18(11), p.3845, 2018.
- SILVEIRA, T. T. da; KOZAKEVICIUS, A. **Transformada Wavelet de Haar: conceitos, formulações, aplicações**. [S.l.]: SBM, 2016.
- SOUZA VINÍCIUS GIUSTI, R. . J. B. A. Alves de. Asfalt: a low-cost system to evaluate pavement conditions in real-time using smartphones and machine learning. **Pervasive and Mobile Computing**, [S.l.], n.51. 10.1016/j.pmcj.2018.10.008, 2018.
- TAYLOR, J. R. **An introduction to Error Analysis: the study of uncertainties in physical measurements**, 2nd ed., university science books. [S.l.: s.n.], 1997. 448p.
- TRANSPORTES, C. N. do. **Pesquisa Confederação Nacional do Transportes (CNT) de Rodovias 2018: relatório gerencial**. 2018. 405p.
- VUPPARABOINA, K. K. et al. Laser-based detection and depth estimation of dry and water-filled potholes: a geometric approach. In: NATIONAL CONFERENCE ON COMMUNICATIONS (NCC), 2015. **Anais...** [S.l.: s.n.], 2015. p.1–6.
- WANG, P. et al. Asphalt Pavement Pothole Detection and Segmentation Based on Wavelet Energy Field. **Mathematical Problems in Engineering**, [S.l.], v.2017, Article ID 1604130, p.13, 2017.
- YANG, H. et al. Integrated SINS/WSN Positioning System for Indoor Mobile Target Using Novel Asynchronous Data Fusion Method. **Journal of Sensors, Article ID 7879198, 11 pages**, [S.l.], v.2017, 2017.
- YU, X.; SALARI, E. Pavement pothole detection and severity measurement using laser imaging. In: IEEE INTERNATIONAL CONFERENCE ON ELECTRO/INFORMATION TECHNOLOGY, 2011. **Anais...** [S.l.: s.n.], 2011. p.1–5.

YUDANTO, R. G.; PETRÉ, F. Sensor fusion for indoor navigation and tracking of automated guided vehicles. In: INTERNATIONAL CONFERENCE ON INDOOR POSITIONING AND INDOOR NAVIGATION (IPIN), 2015. **Anais...** [S.l.: s.n.], 2015. p.1–8.

ZOYSA, K. D. et al. A Public Transport System Based Sensor Network for Road Surface Condition Monitoring. In: WORKSHOP ON NETWORKED SYSTEMS FOR DEVELOPING REGIONS, 2007., New York, NY, USA. **Proceedings...** ACM, 2007. p.9:1–9:6. (NSDR '07).