UNIVERSIDADE FEDERAL DE ITAJUBÁ - UNIFEI
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Sistema de localização topológica utilizando visão computacional

Natalia Sánchez Sánchez

Itabira, 20/04/2023

UNIVERSIDADE FEDERAL DE ITAJUBÁ - UNIFEI
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Natalia Sánchez Sánchez

Sistema de localização topológica utilizando visão computacional

Dissertação submetida ao Programa de Pós-Graduação em ciência e tecnologia da computação como parte dos requisitos para obtenção do Título de Mestre em Ciência e Tecnologia da Computação.

Área de Concentração: Matemática da Computação

Orientador: Prof. Dr. Giovani Bernardes Vitor.

20/04/2023
Itabira

UNIVERSIDADE FEDERAL DE ITAJUBÁ - UNIFEI
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Sistema de localização topológica utilizando visão computacional

Natalia Sánchez Sánchez

Dissertação aprovada por banca examinadora em 20 de abril de 2023, conferindo ao autor o título de **Mestre em Ciência e Tecnologia da Computação.**

Banca Examinadora:
Prof. Dr. Rafael Francisco dos Santos
Prof. Dr. Ruben Dario Hernandez Beleño
Prof. Dr. Giovani Bernardes Vitor

Itabira
2023

Natalia Sánchez Sánchez
Sistema de localização topológica utilizando visão computacional/ Natalia Sánchez Sánchez. – Itabira, 20/04/2023-
140 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Giovani Bernardes Vitor.

Dissertação (Mestrado)
Universidade Federal de Itajubá - UNIFEI
Programa de pós-graduação em ciência e tecnologia da computação, 20/04/2023.

1. Localização topológica. 2. Visão computacional. I. Prof. Dr. Giovani Bernardes Vitor. II. Universidade Federal de Itajubá (Campus Itabira). III. Ciência e Tecnologia da Computação. IV. Sistema de localização topológica utilizando visão computacional.

CDU 07:181:009.3

Natalia Sánchez Sánchez

Sistema de localização topológica utilizando visão computacional

Dissertação submetida ao Programa de Pós-Graduação em ciência e tecnologia da computação como parte dos requisitos para obtenção do Título de Mestre em Ciência e Tecnologia da Computação.

Trabalho aprovado. Itabira, 20 de abril de 2023:

Prof. Dr. Giovani Bernardes Vitor.
Orientador

Prof. Dr. Rafael Francisco dos Santos

Prof. Dr. Ruben Dario Hernandez Beleño

Itabira
20/04/2023

Agradecimentos

Com muita gratidão em meu coração, gostaria de expressar minha imensa gratidão a Deus por me permitir alcançar mais um marco em minha carreira, cheio de novas experiências e conhecimentos valiosos. Gostaria também de agradecer à minha família, que tem sido meu apoio inabalável em todos os meus empreendimentos pessoais e acadêmicos. Seu amor incondicional e encorajamento constante me inspiram a perseverar em busca dos meus sonhos, não importa os obstáculos que possam surgir. Eles são a razão pela qual eu nunca perdi a esperança e a determinação, e me guiaram nos momentos mais desafiadores de aprendizado. Sem dúvida, minha família é meu maior tesouro e sou muito abençoada por tê-los como meus guias na vida.

Hoje, ao concluir meus estudos, sinto-me profundamente grata por esta conquista e dedico-a àqueles que me apoiaram em minha jornada. Em especial, dedico esta conquista ao meu pai que partiu para o céu, pois foi ele quem me inspirou a buscar meus sonhos e a nunca desistir diante das dificuldades. Agradeço também ao meu melhor amigo, que se tornou um pilar de apoio incondicional em minha vida. Sua presença constante e seus conselhos sábios foram essenciais para que eu pudesse chegar até aqui. Obrigada por todas as horas compartilhadas e pelo suporte inabalável. Juntos, vocês são os responsáveis por essa vitória, e é com muita gratidão que compartilho este momento com vocês.

Da mesma forma, agradeço ao Prof Dr. Ruben, que desde que o conheci, sempre me apoiou incondicionalmente, e me ensinou a crescer profissionalmente. Obrigada por ser uma fonte inesgotável de inspiração e conhecimento. Ele foi meu guia e mentor nesta emocionante jornada de aprendizado e crescimento profissional.

Também é muito importante expressar gratidão e reconhecimento a quem nos ajuda e apoia em nossos desafios. Por isso, quero expressar meu sincero agradecimento ao Prof Dr. Rafael, que tem sido uma figura essencial na minha jornada acadêmica e profissional. Seu conhecimento, habilidades e paciência são incomparáveis e sem dúvida contribuíram significativamente para meu crescimento pessoal e profissional. Agradeço por sempre estar disposto a compartilhar seus conhecimentos e ideias, por me encorajar a buscar soluções inovadoras e por ser um modelo inspirador de profissionalismo e dedicação. Você é uma pessoa maravilhosa e que Deus te abençoe sempre.

Por outro lado, nessa aventura de explorar e conhecer uma nova cultura, um novo país e pessoas que estiveram sempre presentes, sou grata a Deus por tê-los colocado em meu caminho. Agradeço em especial ao engenheiro Rafael, que sempre me apoiou e compartilhou ideias para que eu pudesse melhorar continuamente. Além de me acompanhar durante experimentos e me oferecer seu tempo e conhecimento, ele foi uma presença constante e incondicional em minha vida. Admiro profundamente o profissionalismo e a dedicação dele, e não há momento melhor para expressar minha gratidão.

Assim, agradeço a família Pinto, Whiskas, Sr. Antonio, Wagner, Gleice, Moisés e outras pessoas que sempre estiveram dispostas a me ajudar e deram seu tempo para me ouvir-me quando mais precisava. Agradeço também ao meu orientador Prof Dr. Giovani, por me dar a oportunidade de trabalhar com ele e me ensinar seus conhecimentos.

Meus amigos e companheiros de viagem, funcionários da Unifei e novos amigos no Brasil, eu lhes digo que hoje esta maravilhosa aventura termina e obrigada por estarem comigo. Sei que, mesmo após o término deste capítulo, continuaremos a enfrentar novas aventuras e desafios juntos. Mais uma vez, obrigada por fazerem parte desta jornada tão enriquecedora.

Finalmente, obrigada demais professores envolvidos em minha jornada universitária, agradeço por transmitirem os conhecimentos necessários para que eu chegasse aqui hoje. Simplesmente, obrigada por tudo, espero que possamos continuar juntos nesta jornada chamada vida.

Nada disto teria sido possível sem vocês. Este trabalho é o resultado de esforço, dedicação e perseverança diariamente.

Simplesmente ***muito obrigada****.*

Atenciosamente,
Natalia Sánchez Sánchez.

"O tempo é o recurso mais precioso que temos e não pode ser comprado nem recuperado. Cada segundo que passa é uma oportunidade única para alcançar nossos objetivos e realizar nossos sonhos. Portanto, é importante valorizá-lo e usá-lo sabiamente. Nunca desista, porque o sucesso é reservado para aqueles que perseveram e mantêm a determinação, mesmo diante dos desafios. Lembre-se sempre de que cada passo em direção aos seus objetivos é um passo em direção ao seu sucesso."

Resumo

O presente trabalho apresenta uma metodologia inovadora que utiliza técnicas de visão computacional para realizar a localização topológica de um veículo autônomo. A grande vantagem desta técnica é que ela dispensa o uso de GPS ou qualquer sensor de posição em tempo contínuo, o que pode aumentar significativamente a segurança em regiões onde os sensores de localização são limitados ou ausentes. A metodologia consiste na construção de um mapa topológico da região de interesse, onde os pontos de interesse são definidos. Para isso, várias imagens de cada coordenada são coletadas e passam por filtros e processamentos para formar um banco de imagens georreferenciadas. A partir daí, o sistema recebe como entrada um vídeo, onde as imagens são comparadas com as imagens do banco de imagens, utilizando o algoritmo SURF, para definir se há correspondência com as coordenadas de interesse. Se a correspondência for identificada, o algoritmo define a localização do veículo no mapa topológico. Os resultados dos experimentos realizados mostram uma taxa de acerto de 91,5% na detecção dos pontos de interesse dentro do mapa topológico, indicando que esta metodologia pode complementar o sistema de navegação de um veículo autônomo de forma eficiente e precisa.

Palavras-chaves: Localização topológica, visão computacional, navegação e veículo autônomo.

Abstract

This paper presents an innovative methodology that uses computer vision techniques to perform the topological localization of an autonomous vehicle. The great advantage of this technique is that it eliminates the use of GPS or any continuous time position sensor, which can significantly increase safety in regions where location sensors are limited or absent. The methodology consists of building a topological map of the region of interest, where the points of interest are defined. To do this, several images of each coordinate are collected and go through filters and processing to form a georeferenced image bank. From there, the system receives as input a video, where the images are compared with the images in the image bank, using the SURF algorithm, to define if there is a correspondence with the coordinates of interest. If a match is identified, the algorithm defines the location of the vehicle on the topological map. The results of the experiments performed show a 91.5% accuracy rate in detecting the points of interest within the topological map, indicating that this methodology can complement the navigation system of an autonomous vehicle efficiently and accurately.

Key-words: Topological location, computer vision, navigation and autonomous vehicle.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

AV	*Veículo autônomo*	18
Distância	*DIS*	85
GCS	*Sistema de coordenadas geográficas*	55
GNSS	*Sistema Global de Navegação por Satélite*	24
GPS	*Sistema de Posicionamento Global*	19
KI	*Application Program Interface / KPImage*	79
KNN	*K Nearest Neighbors*	36
KP	*Pontos chave, KPobjeto, Keypoints*	51
LIDAR	*Light Detection and Ranging ou Laser Imaging Detection and Ranging*	19
MaxNo	*Maior número normalizado de correspondências*	87
MH	*Matriz Hessiana*	45
MNMR	*Máximo normalizado*	79
MR	*Ajuste de Matches Rate*	74
MT	*Matches das imagens*	80
ORB	*Oriented FAST and rotated BRIEF*	33
OSM	*OpenStreetMap*	11
SIFT	*Scale-invariant Feature Transform*	11
SURF	*Speeded-Up Robust Features*	11
TH	*Threshold*	72
UNIFEI	*Universidade Federal de Itajubá*	21
UTM	*Universal Transverse Mercator*	55
VAM	*Veículo micro-aéreo*	26
VC	*Visão Computacional*	35
vSLAM	*Visual simultaneous localization and mapping (vSLAM)*	43
WGS84	*Coordenadas WGS84 (WGS = World Geodetic System)*	55
ZED	*Câmera ZED*	65

Sumário

1 Introdução

Segundo a Organização Mundial da Saúde (OMS) [4], até o ano 2022, mais de 3.500 pessoas morrem diariamente nas estradas do mundo, o que significa que quase 1,3 milhões de mortes e cerca de 50 milhões de feridos por ano poderiam ser evitados. Isto faz dos acidentes rodoviários, a principal causa de morte de crianças e jovens em todo o mundo [4]. Da mesma forma, a OMS diz haver pelo menos 2,2 bilhões de pessoas no mundo com deficiências visuais, pelo motivo de degeneração macular relacionada à idade, catarata, retinopatia diabética, erros de refração não corrigidos, entre outros. Pessoas com deficiências visuais que não estão com seus exames em dia ou que não usam óculos, podem causar acidentes de trânsito [5].

Com o avanço da tecnologia dos AV (*Veículo autônomo*), e com a implementação de algumas destas tecnologias os veículos atuais, tem gerado um grande impacto na redução do número de acidentes fatais, pois **"autonomia"** refere-se à capacidade de dirigir um veículo sem a necessidade de controle físico ativo ou monitoramento por uma pessoa. Ele pode ser classificado de acordo com seu nível de autonomia, onde o nível **0** é quando o motorista controla totalmente o ambiente de direção, até o nível **5** de autonomia, onde o veículo tem um sistema de direção totalmente automatizado [6].

Entretanto, a condução autônoma de veículos apresenta diversas vantagens, tais como a segurança, eficiência, conveniência e acessibilidade. No entanto, também há algumas desvantagens, como o alto custo, a dependência da tecnologia, problemas de infraestrutura e questões éticas. Além disso, a autonomia em AV traz benefícios importantes, como a redução da emissão de gases poluentes, economia de espaço e maior segurança nas estradas, mas também traz riscos, como falhas na tecnologia, roubo de dados, ataques cibernéticos, responsabilidade em caso de acidentes e impactos na economia. Portanto, é necessário considerar cuidadosamente todos esses fatores ao avaliar a adoção em larga escala de AV.

O foco neste campo tecnológico é essencial para promover uma maior exploração em um ambiente que requer pesquisa constante devido à complexidade e controvérsias que ela suscita [7]. No mais, vários autores afirmam que a segurança oferecida por este tipo de veículo precisa ser investigada, pois mesmo o menor erro pode ameaçar a vida das pessoas [6, 8, 9, 10]. Além disso, os sistemas de visão por computador cresceram nos últimos anos [11, 12, 13], onde podem se tornar uma ferramenta muito importante para o desenvolvimento de aplicações autônomas em qualquer área do conhecimento.

Controle de movimento, detecção do ambiente e identificação de trajetória são fundamentais para a implementação da autonomia no veículo, sendo cruciais para a se-

gurança do usuário. Uma maneira de monitorar isto é obter a aquisição e localização dos dados de posição do veículo, que devem ser tão precisos quanto possível.

Os AV também contam com vários sistemas para se moverem suavemente quando surgem obstáculos em seu caminho. Um desses sistemas é aquele que controla sua navegação, que consiste na capacidade de mudar ou seguir uma rota livre de colisões dentro de seu espaço de trabalho [14, 15, 16]. As formas pelas quais o problema de navegação é resolvido podem ser variadas, porém, a solução escolhida depende do tipo de espaço de trabalho em que o AV opera, sendo eles conhecidos, ou desconhecidos [8, 15, 17]. Com base nas duas formas de espaço de trabalho, existem duas estratégias de navegação diferentes: global e local (baseada no comportamento) [18, 16, 15, 17].

A navegação local e global são dois tipos de sistemas de posicionamento essenciais para a locomoção de pessoas e objetos em diversas situações. A navegação local é utilizada para movimentação em pequenas áreas, enquanto a navegação global é utilizada em áreas maiores. Por o lado da navegação local caracteriza-se por envolver a identificação de pontos de referência próximos, enquanto a navegação global utiliza sistemas de posicionamento global como o GPS para determinar a localização. Ambas as técnicas utilizam sensores para obter informações sobre o ambiente. A navegação local pode utilizar sensores de proximidade, tais como sensores de ultrassom e infravermelho para detectar a presença de obstáculos próximos. Já a navegação global, além de utilizar o GPS, pode utilizar sensores de visão computacional para identificar características do ambiente, como marcos de referência ou áreas perigosas. Além disso, sensores de giroscópio e acelerômetro podem ser usados em ambos os sistemas para determinar a orientação e a velocidade do veículo. Em resumo, a principal diferença entre os dois tipos de navegação está na área de atuação e no uso de sensores específicos para cada técnica.

Entretanto, para qualquer tipo de navegação, a localização em tempo real é necessária e, para isso, o uso combinado de dados de sensores e informações ambientais armazenadas em um mapa pode ajudar a estimar a posição e orientação de um AV em seu ambiente (GPS (*Sistema de Posicionamento Global*), LIDAR (*Light Detection and Ranging ou Laser Imaging Detection and Ranging*), Radar, IMU (Inertial Measurement Unit), odômetro, etc). A localização é essencial para a navegação, pois permite saber a localização atual e planejamento da rota para chegar ao seu destino com eficiência. Tais sensores precisam obter dados confiáveis para realizar a navegação, a fim de reduzir a incerteza de seu posicionamento, permitindo que eles se movam de um ponto inicial para um ponto final desejado. Um exemplo é o GPS, utilizado para posicionamento de veículos. A operação confiável e segura dos sensores GPS é um fator chave para aumentar a aceitação de AV [19].

Baseado na localização atual dada pelo GPS, é possível fazer um algoritmo para a geração automática de rotas [20], a fim de escolher a rota mais curta e otimizada de

um lugar para outro. Isto é essencial para os veículos funcionarem corretamente e de forma autônoma com segurança, sem qualquer interação humana [21]. Entretanto, o GPS é suscetível a ruídos, tais como interferências que geram imprecisões, afetando assim a localização.

Por esta razão, quanto mais complexo e completo for um sistema autônomo, mais erros e riscos podem ocorrer no tempo de resposta, isto leva a novos problemas que vão desde o mapeamento de um ambiente não estruturado até a prevenção de obstáculos em diferentes direções do ambiente. Para isso, tem que levar em consideração elementos estáticos e dinâmicos; seu posicionamento e localização para a perfeita execução da missão, planejamento flexível da trajetória no contexto da operação e muitos outros desafios surgem.

A seguir, são apresentados a proposta de uma metodologia que permite a implementação de uma visão computacional integrada com localização topológica e os objetivos deste trabalho.

1.1 Contextualização, motivação e questão de pesquisa

O desafio deste projeto de pesquisa é fazer uma proposta metodológica para o projeto e desenvolvimento de um algoritmo que consiga ser implementado em um AV, que seja capaz de retornar a localização topológica do AV em um mapa, utilizando a percepção e captura de imagens de uma câmera, sem a necessidade do uso de um sensor GPS.

O desafio surgiu porque pretende-se integrar um sistema que possa auxiliar o sistema de localização de um veículo onde possa existir falhas no sinal do GPS. Alguns autores dizem que existem situações em que o GPS não está disponível, devido a interferências, blindagem ou efeitos multipath, reflexão do sinal, sempre dependendo do meio de navegação [22, 23, 24]. Além disso, a baixa taxa de amostragem é um problema que afeta a disponibilidade de informações de navegação para ações de controle no veículo. Seu desempenho é afetado negativamente, mau tempo e estruturas de construção que bloqueiam a força do sinal para o receptor GPS [25, 24]. Este aspecto faz com que o GPS não seja totalmente confiável para a localização e implementação da navegação.

Por outro lado, as câmeras também são utilizadas para a localização de veículos e até mesmo de robôs móveis. Os métodos baseados em câmeras podem alcançar um posicionamento relativamente preciso com baixo custo, baixo peso e baixo consumo de energia [24]. Os dados dos sensores visuais das câmeras fornecem uma riqueza de informações essenciais para a localização, mapeamento, detecção de obstáculos e planejamento de rotas sem a necessidade de fundir outros sensores. Entretanto, as câmeras sofrem a influência das condições de iluminação, especialmente em um ambiente externo, e a distorção de-

vido ao movimento rápido, bem como sua visão estreita [25, 24]. Para isso, esta proposta propõe um algoritmo que consegue identificar o ambiente através do processamento de imagens com a integração de uma localização topológica, para complementar o processo de navegação em um AV quando outros sensores tiverem problemas de imprecisão ou falhas.

Essa proposta surgiu no campus Itabira da UNIFEI (*Universidade Federal de Itajubá*), no grupo RobSIC, onde existe um carrinho de golfe financiado pela VALE cujo objetivo principal é se transformar em um AV. Durante este trabalho se trabalharou na parte de localização topológica com a integração de visão computacional, que ajudará a identificar o caminho a seguir e perceber onde se encontra o veículo. Além disso, a intenção deste trabalho é contribuir para a melhoria da segurança e propor uma metodologia que responda à seguinte pergunta de pesquisa: *Pode ser implementada uma localização topológica utilizando uma câmera?*.

Esta questão de pesquisa surge de uma revisão sistemática [1], cujo objetivo era conhecer a situação dos algoritmos de navegação topológica que integram o processamento de imagens e ver quais sensores de baixos custos são implementados para este tipo de sistema. Como resultado da revisão, foram mostradas as vantagens e desvantagens dos métodos que integram algoritmos de processamento de imagem e navegação, considerando a compilação de um banco de dados de 315 artigos publicados nos últimos 5 anos, que pode ser um guia útil para pesquisadores e engenheiros na área. Além disso, está revisão utilizou a metodologia PICO (Patient, Intervention, Comparison and Outcome) [26] para estabelecer as questões de pesquisa e os critérios de exclusão e inclusão.

1.2 Objetivos

1.2.1 Objetivo General

Fazer um sistema capaz de mapear uma localização, através de um banco de dados georreferenciado e imagens, para garantir a localização topológica do veículo em um ambiente externo.

1.2.2 Objetivos específicos

- Implementar um algoritmo de visão computacional, por meio da detecção de pontos de interesse de uma imagem, para identificar pontos de passagem do mapa, a fim de definir a localização do veículo.

[1] https://drive.google.com/file/d/1QTnrMn8OYZTbk2hGzdClEDG127YBRg8w/view?usp=share$_l ink$

- Estimar a localização topológica do veículo implementando a integração de um banco de dados georreferenciado e imagens, para garantir o reconhecimento da trajetória de um ponto de partida até um ponto final em um mapa.
- Validar o algoritmo de visão computacional proposto por meio de um ambiente externo para garantir que o sistema gere a localização por meio de um mapa.

1.3 Contribuições

Durante o período de mestrado houve participação no XXIV Congresso Brasileiro de Automação (CBA) com a publicação intitulada *"Digital Twins: A 3D simulation approach to test validation with an autonomous vehicle"*, este artigo toma como referência o carrinho de golfe, onde é projetado e implementado em ambiente Gazebo com o campus da UNIFEI Itabira, essa integração é implementada com ROS para validar um algoritmo de localização e detecção. Esta simulação é realizada para minimizar possíveis erros em termos de sua integração na realidade.

Por outro lado, foi feito um artigo de revisão da literatura, que está em processo de revisão na revista *"Engineering Applications of Artificial Intelligence"* ISSN: 0952-1976, intitulado "Revisão sistemática de acordo com abordagens de localização topológica implementadas para veículo autônomo".

Além disso, este é um trabalho que contribui para o desenvolvimento da navegação porque contém uma integração do processamento de imagens e da localização topológica. Sendo que, o desenvolvimento desta proposta rendeu 91,5% de sucesso na realização desta integração. Da mesma forma, este algoritmo abre um campo de investigação, para determinar se no futuro a metodologia proposta pode ser integrada com algum método de navegação para um AV, pois nos testes que foram realizados é possível mostrar que o algoritmo funciona a qualquer hora do dia, tendo em conta que é robusto à variação ambiental.

1.4 Estrutura do trabalho

Este trabalho é dividido em 5 capítulos, além do presente. Cada capítulo é descrito a seguir:

O capítulo 2 apresenta um estado da arte das terminologias principais que devem ser consideradas para compreender a proposta deste trabalho.

O capítulo 3, descreve a metodologia proposta, que está subdividida em diferentes fases, que, considera a integração da localização topológica e do processamento de imagens.

O Capítulo 4 apresenta os resultados tanto qualitativos como quantitativos obtidos através da metodologia proposta. Por fim, o capítulo 5 e 6 apresenta uma discussão, conclusões e contribuições deste trabalho, como as diferentes linhas de pesquisa, geradas em decorrência das contribuições apresentadas neste trabalho. Além disso, um conjunto de apêndices foram anexados, o que ajudará o leitor a complementar sua visão deste trabalho, tem evidência de resultados quantitativos e qualitativos, onde por meio de vídeos mostram o funcionamento da metodologia implementada.

2 Revisão teórica

Hoje, os veículos atuais são equipados com inúmeros sensores que integram diferentes sistemas para melhorar, ajustar ou automatizar a segurança do veículo e a experiência de direção. Estes sistemas ajudam os motoristas a fornecer medidas preventivas para reduzir a exposição ao risco ou colaborar na automatização das tarefas de direção para minimizar os erros humanos [27, 28]. Eles normalmente se baseiam apenas em medições de sensores internos ("proprioceptivos"), mas não são suficientes para fornecer aplicações de segurança e alerta associadas com o ambiente externo. Com sensores externos (exteroceptivos), os veículos conseguem adquirir informações sobre o ambiente ao redor, reconhecendo outros fatores e objetos que coexistem no mesmo espaço. A detecção externa de veículos está ganhando importância especialmente com a proliferação de câmeras que, combinada com um melhor processamento e análise da imagem, permitem uma ampla gama de aplicações [27, 29, 30].

A Figura 1 mostra uma classificação adaptada de como se subdividem os sensores, onde eles podem ser divididos em proprioceptivo, exteroceptivo, ativo, passivo, absoluto ou relativo. Basicamente, esta classificação é escolhida dependendo da aplicação com a qual o sensor será utilizado. Os sensores mais importantes para AV são: câmera, IMU (Unidade de medição inercial), giroscópio, acelerômetro, odômetros, LIDAR, radar, GNSS (*Sistema Global de Navegação por Satélite*), GPS, entre outros. Estes sensores permitem o reconhecimento do ambiente, onde eles podem receber ou emitir informações de tudo o que os rodeia. O GPS é um sistema de posicionamento por satélite específico dos Estados Unidos, enquanto o GNSS é um termo genérico que se refere a qualquer sistema de navegação por satélite que utiliza várias constelações de satélites. Sem a implementação destes dispositivos não seria possível mover ou interagir com objetos, pessoas ou o ambiente. Por este motivo, antes de escolher um bom sensor deve saber como se classificam e quais poderiam ser implementados.

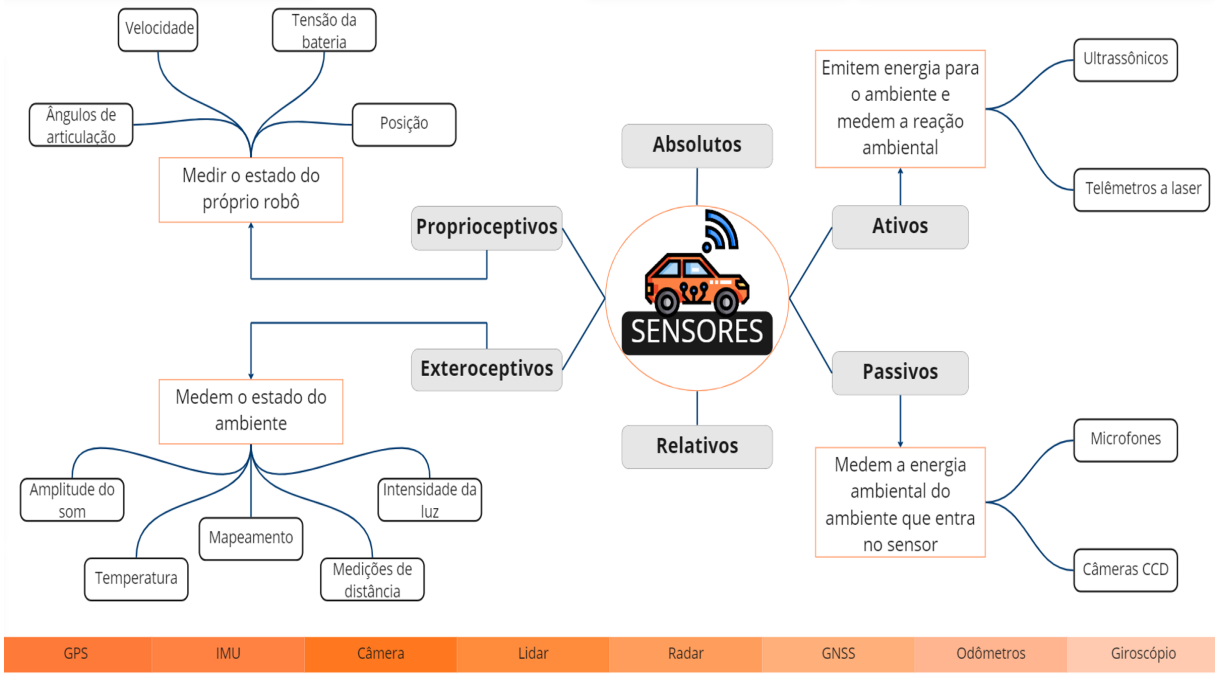

Figura 1 – Classificação de sensores.

Por outro lado, a visão computacional e a detecção são áreas de pesquisa em que se busca extrair informações úteis das imagens e vídeos capturados por câmeras e outros dispositivos. Com essas informações, é possível realizar uma série de tarefas, como a detecção de objetos, reconhecimento de faces e identificação de padrões em imagens médicas. Além disso, a visão computacional e a detecção também são fundamentais para a localização de objetos e veículos em ambientes autônomos. A partir da análise das imagens capturadas, os algoritmos de visão computacional podem identificar características distintas do ambiente, como linhas de demarcação, placas de sinalização, obstáculos e outros objetos, permitindo que o AV possa se orientar e se deslocar com segurança. Assim, a visão computacional e a detecção são tecnologias chaves para o desenvolvimento de sistemas autônomos, que têm o potencial de revolucionar a maneira como nos deslocamos e interagimos com o mundo ao nosso redor [31, 32, 33]. Por isso, uma técnica ideal de detecção de características deve ser robusta às transformações de imagem, tais como rotação, escala, iluminação, ruído e transformações afins [34, 35, 36]. Além disso, é importante que as características selecionadas sejam altamente distintas, a fim de que seja possível realizar sua correspondência correta com alta probabilidade. Para alcançar esse objetivo, geralmente é necessário realizar diferentes processos de seleção e processamento para cada característica individual, garantindo que cada uma delas seja suficientemente única e discriminante. Isso é particularmente importante em áreas como reconhecimento de padrões, onde a identificação de características distintas é fundamental para a precisão e eficácia dos modelos de classificação.

Nesta seção, apresentamos trabalhos que serviram como base para o desenvolvi-

mento da pesquisa, bem como aqueles que possuem alguma relação com o objetivo proposto. As ideias desses trabalhos são discutidas com o intuito de oferecer soluções para o problema em questão. Além disso, conceitos simples e essenciais para o desenvolvimento da metodologia ou experimentos também são abordados.

2.1 Técnicas de navegação

Com diferença da navegação, a localização buscar responder a perguntas como: **Qual é a sua posição atual? Onde você tem estado? Onde você está indo? Como você pode alcançar uma meta de forma mais eficiente? Qual é o melhor caminho para chegar lá? Como evitar obstáculos e perigos na rota? Como atualizar a rota, se necessário?** entre outras perguntas. Diferentes autores [37, 38] definem a navegação como um conjunto de ações realizadas para alcançar uma meta ou destino atribuído a partir de um determinado ponto de origem. Da mesma forma, vários trabalhos [39, 40, 41, 42] tentaram explicar como as pessoas estão localizadas e como chegam a um destino.

Um dos principais desafios da navegação é a criação de mapas e o planejamento de rotas durante a condução, pois deve considerar percepção sensorial, controle de movimento e detecção de objetos, entre outros aspectos. A navegação necessita ter uma modelagem de ambiente, um sistema sensorial, planejamento de rotas, um sistema de controle e um sistema de posicionamento [16, 43, 44]. Vários autores sugerem soluções como localização, mapeamento simultâneo e localização topológica como solução para estas situações [14, 43].

A navegação pode ser implementada usando mapas geométricos, mapas de decomposição, mapas topológicos e mapas semânticos [45, 46, 47]. Recentemente, o mapa topológico tem sido utilizado na literatura para simplificar o espaço de busca. Um exemplo é o caso de C.Wang [44] onde os autores propõem um método para construir um mapa topológico do ambiente usando uma câmera de profundidade montada em um robô. O robô explora o ambiente e constrói um mapa do espaço circundante, usado para planejar as trajetórias futuras do robô. Os autores também propõem uma estratégia de exploração eficiente do ambiente baseada no ganho de informação, que envolve a seleção das áreas mais informativas para exploração com base no estado atual do mapa.

Em contraste, Oleynikova [48] propõe um método para planejar as trajetórias de um VAM (*Veículo micro-aéreo*) em um ambiente 3D usando um grafo topológico esparso. Os autores usam uma combinação de câmeras LIDAR e RGB-D para construir um mapa 3D do ambiente, que é então simplificado em um grafo topológico esparso. O grafo é usado para planejar as trajetórias do VAM, levando em consideração fatores como desvio de obstáculos e eficiência energética.

No geral, ambos os trabalhos usam mapas topológicos para navegação em ambientes 3D, mas diferem em termos das técnicas específicas usadas e do tipo de robô ou veículo usado para exploração. C.Wang [44] se concentra em um robô terrestre, enquanto Oleynikova [48] é adaptado para um veículo micro-aéreo.

Por outro lado, por várias décadas, muitos pesquisadores e cientistas forneceram inúmeras metodologias em abordagens de navegação. Vários métodos empregados para a navegação de um robô móvel são amplamente classificados em duas categorias, segundo B.K. Patle [25], clássica e reativa. Na abordagem clássica, o robô é equipado com um mapa do ambiente e um caminho predefinido é planejado com base nesse mapa. O robô segue esse caminho até chegar ao seu destino. No entanto, essa abordagem é limitada pela precisão do mapa e pela capacidade do robô de se localizar no mapa. Na abordagem reativa, o robô reage ao seu ambiente em tempo real usando dados dos sensores para evitar obstáculos e chegar ao seu destino. No entanto, esta abordagem é limitada pela complexidade do ambiente e dos sensores utilizados. Em resumo, a navegação de robôs móveis pode ser interrompida por vários fatores, como a precisão do mapa na abordagem clássica, a capacidade do robô de se localizar dentro do mapa, a complexidade do ambiente na abordagem reativa e os sensores usado na abordagem reativa.

Na abordagem reativa para a navegação e localização de veículos autônomos, os algoritmos mais comuns são aqueles que utilizam informações sensoriais em tempo real para tomar decisões e realizar ações imediatas de controle. Dentre esses algoritmos, podemos citar o algoritmo de campo potencial artificial, que utiliza campos de potenciais para guiar o veículo em direção ao objetivo, evitando obstáculos; o algoritmo de campos de vetoriais, que utiliza vetores de força para orientar o veículo em direção ao objetivo, combinando informações de múltiplos sensores; e o algoritmo de navegação baseado em comportamentos, que utiliza regras comportamentais para tomar decisões em diferentes situações, como evitar obstáculos, seguir uma linha de trajetória ou alcançar um objetivo específico.

Esses algoritmos são capazes de realizar a navegação e localização em tempo real, utilizando informações sensoriais para tomar decisões de controle em cada momento. Isso significa que o veículo é capaz de se adaptar a mudanças no ambiente e realizar manobras de evasão de obstáculos de forma autônoma e eficiente. A escolha do algoritmo mais adequado para cada problema depende de uma série de fatores, como o tipo de ambiente em que o veículo irá operar e o nível de precisão e robustez necessários para a navegação e localização.

Além disso, existem diversas técnicas de algoritmos que podem ser utilizadas para a implementação de localização e navegação em sistemas autônomos. O algoritmo genético [49, 50] é uma abordagem inspirada na seleção natural, onde soluções são avaliadas e selecionadas através de processos de reprodução e mutação, buscando encontrar a melhor

solução para o problema. A lógica difusa [51] é outra técnica que pode ser aplicada para representar o conhecimento impreciso e incerto, permitindo que o sistema faça inferências e tome decisões baseadas em critérios difusos. A rede neural [52] é uma técnica inspirada no funcionamento do cérebro humano, que consiste em estruturas de processamento de informações capazes de aprender padrões e fazer predições.

Por outro lado, a navegação e localização de veículos autônomos são tarefas complexas que exigem a percepção do ambiente, tomada de decisão e controle do veículo. Existem diversos algoritmos que podem ser utilizados para lidar com esses desafios, como o algoritmo genético [49, 50], lógica difusa [51], rede neural [52], algoritmo de vaga-lume [53], a otimização de enxame de partículas [54], a otimização de colônia de formigas [50, 55], a otimização de forrageamento bacteriano [56], a colônia artificial de abelhas [57], o algoritmo de salto de sapo embaralhado [58], a otimização invasiva de ervas daninhas [59], o algoritmo de busca de harmonia [60], o algoritmo de morcego [61] e o algoritmo de evolução diferencial [62], entre outros.

Esses algoritmos têm diferentes inspirações biológicas e podem ser aplicados para diversas tarefas, como a otimização de trajetórias e rotas, a modelagem da incerteza em tarefas de navegação e localização, a localização e reconhecimento de objetos e a busca de caminhos ótimos em ambientes complexos. Cada algoritmo tem suas próprias vantagens e limitações, e a escolha do melhor algoritmo dependerá das necessidades e condições específicas de cada situação.

Alguns exemplos de algoritmos incluem o algoritmo de salto de sapo embaralhado, que pode ser utilizado para fazer escolhas aleatórias de trajetórias em uma região desconhecida para explorar e encontrar a melhor rota possível, e a colônia artificial de abelhas, que pode ser utilizada para otimizar a rota e encontrar o melhor caminho para o veículo, levando em consideração as condições do ambiente, a presença de obstáculos e outras variáveis. Em resumo, a navegação e localização de veículos autônomos são complexas, mas a utilização de algoritmos pode ajudar a tornar essas tarefas mais eficientes e seguras.

Em conclusão, a integração de localização e navegação em um AV envolve o uso de diversos sensores, para criar um mapa detalhado do ambiente em que o veículo está operando. Esses sensores coletam dados em tempo real sobre a posição do veículo em relação ao seu entorno e permitem que o sistema de navegação faça atualizações constantes em relação ao caminho que o veículo deve seguir.

A localização é usada para determinar a posição do veículo em relação ao mundo externo e é fundamental para a navegação autônoma. Por meio do uso de tecnologias de localização como GPS, o veículo é capaz de determinar sua posição exata na estrada, enquanto sensores como câmeras e lidars são usados para monitorar as características do ambiente em que o veículo está operando, como a presença de outros veículos ou obstáculos na estrada.

A navegação, por sua vez, é responsável por determinar a melhor rota para o veículo seguir com base em informações do ambiente, do destino e de outros fatores relevantes. O sistema de navegação deve ser capaz de interpretar dados de sensores e mapas para identificar a melhor rota a seguir, levando em consideração a posição atual do veículo e o destino desejado.

Em um AV, a integração de localização e navegação é crucial para permitir que o veículo opere de forma segura e eficiente. O sistema deve ser capaz de tomar decisões em tempo real com base nos dados coletados pelos sensores, para que possa adaptar a trajetória do veículo conforme necessário e evitar colisões com outros veículos ou obstáculos na estrada. No geral, a localização e a navegação são componentes cruciais da capacidade de um AV operar com segurança e eficiência no mundo real [63].

2.2 Técnicas de localização

A diferença da navegação a localização intenta responder perguntas como **Onde estou em relação ao meu ambiente imediato? Como posso determinar minha posição com precisão? Como posso ajustar meu comportamento para cumprir meu objetivo atual? Como posso atualizar minha posição em tempo real enquanto me movo? Como posso usar dados de localização para melhorar o desempenho geral e tomar decisões mais inteligentes e eficazes?** Da mesma forma, a questão que este trabalho tenta responder é **Como posso lidar com situações em que minha localização pode ser afetada por fatores externos, como a interferência de sinais GPS?**.

A percepção do ambiente é importante para AV porque é a base para outras tarefas, como localização, mapeamento e planejamento. Sem uma percepção precisa, o veículo pode não conseguir determinar com precisão sua posição ou navegar pelo ambiente [64]. Por exemplo, um AV pode usar a percepção para identificar obstáculos como pedestres, outros veículos ou bloqueios de estradas. A percepção também pode ser usada para identificar sinais de trânsito, marcações de faixas e outras características importantes do ambiente. Essas informações podem ser usadas para tarefas de planejamento e controle, como decidir quando acelerar, frear ou virar. Além disso, a percepção é essencial para que os AV operem com segurança e confiabilidade, pois precisam ser capazes de perceber seus arredores com precisão e rapidez para tomar decisões em tempo real e evitar colisões.

A relação entre percepção e localização, em um AV, é que a percepção depende dos dados de localização para criar uma representação precisa do ambiente circundante. Sem dados precisos de localização, o sistema de percepção não consegue gerar uma representação confiável do ambiente, o que pode resultar em decisões incorretas por parte do veículo. A localização em tempo real estima sua localização e orientação em seu ambiente

operacional, alinhando dados de uma combinação de sensores e informações ambientais armazenadas em um mapa [24]. Este processo determina a posição e orientação em relação ao ambiente, o cálculo preciso e robusto da localização, bem como a orientação, por meio de um mapa, fornece uma base essencial para um veículo determinar as ações adequadas em determinadas situações.

Uma das vantagens de realizar uma localização ao ar livre é a possibilidade de utilizar o GPS. Este sistema fornece uma estimativa da posição absoluta, especificamente latitude e longitude, mas está sujeito a algumas limitações. A precisão da estimativa depende do número de satélites detectados pelos sensores [24], o que pode afetar a precisão da localização. No entanto, atualmente, a maioria dos algoritmos de localização utiliza mapas métricos para auxiliar no processo. Esses mapas possuem uma escala métrica, que permite a representação do espaço de forma a preservar as distâncias entre os pontos. Quer dizer que o mapa métrico resultante é uma representação do ambiente que inclui informações sobre a localização de objetos, obstáculos e outros recursos em coordenadas contínuas [65].

O uso de um mapa métrico em algoritmos de localização permite uma estimativa precisa da localização de um agente comparando as medições do sensor com o mapa. Aproveitando o sistema de coordenadas contínuas do mapa, os algoritmos de localização podem usar técnicas como filtro de Kalman ou filtro de partículas para estimar a localização do agente com alta precisão e robustez. Da mesma forma, existe a aproximação GPS baseada no filtro de partículas, este é um filtro bayesiano que representa a crença por um conjunto de amostras ponderadas (chamadas de partículas). Os filtros são aplicáveis a sistemas que admitem um modelo e muitos autores o implementam ao realizar uma localização [66, 67, 68, 69].

O filtro de Kalman opera recursivamente em fluxos de dados ruidosos (observações), para produzir um estado estatisticamente ótimo do sistema subjacente cujo estado se deseja estimar [70, 71, 72]. Este tipo de ferramenta ajuda a obter uma melhor localização ao implementá-lo em qualquer mecanismo.

O algoritmo de localização de Monte Carlo, também conhecido como filtro de partículas, é um algoritmo probabilístico usado para estimar o estado de um robô com base nas medições do sensor e nas entradas de controle. O algoritmo representa a crença sobre o estado do robô como um conjunto de partículas, onde cada partícula representa um possível estado do robô. Tem a vantagem de não necessitar de informação de GPS, mas, por outro lado, requer um mapa prévio do espaço. Outro exemplo é o algoritmo MCL **modificado** para estimativa de posição ao ar livre [24, 73, 74]. Da mesma forma, alguns autores dividem a localização em duas:

- **Localização relativa (local):** Usa sensores internos (como codificadores de roda,

sensores de inércia e bússolas digitais) para estimar a posição do robô. A estimativa da posição e orientação do robô são informações produzidas por diferentes sensores através da integração de informações, geralmente encoders ou sensores inerciais. A integração começa a partir da posição inicial e é continuamente atualizada ao longo do tempo. O posicionamento relativo só pode ser usado por um curto período de tempo [24, 75, 76].

- **Localização absoluta (global):** Utiliza sensores externos para calcular a direção do local em relação ao ambiente [77]. Este método permite que o robô busque sua localização diretamente do domínio do sistema móvel. Existem vários métodos que geralmente dependem de navegação, waypoints ativos ou passivos, correspondência de mapas ou sinais baseados em satélite, como o GPS [77, 24, 67].

Por outro lado, a localização em um AV é um aspecto crítico para o seu correto funcionamento. Existem diferentes técnicas e tecnologias utilizadas para obter a localização do veículo em tempo real, incluindo sistemas de posicionamento global (GPS), odometria, sensores de proximidade, mapas digitais e outros.

O GPS é uma das principais tecnologias utilizadas para a localização de um AV. Ele fornece coordenadas de latitude e longitude que podem ser usadas para determinar a posição do veículo em relação a um mapa digital. No entanto, o GPS pode ser afetado por interferência do sinal, como edifícios altos ou túneis, o que pode reduzir sua precisão.

Pois a odometria implica em erro qualitativo, entre outros fatores ou imperfeições no terreno que podem causar imprecisões [78, 79, 80]. A odometria é outra técnica utilizada para a localização em um AV. Ela envolve a medição da distância percorrida pelas rodas do veículo e a velocidade com que estão girando. Isso pode ser usado para calcular a posição do veículo em relação a sua posição inicial.

Os sensores de proximidade, como sensores de ultrassom ou lidar, podem ser usados para detectar a presença de objetos próximos ao veículo e ajudar a determinar sua posição em relação a esses objetos.

Os mapas digitais também são usados para auxiliar na localização do veículo. Eles fornecem informações detalhadas sobre as estradas, faixas, sinais de trânsito e outros elementos do ambiente em que o veículo está se movendo. Isso pode ajudar a reduzir a margem de erro na determinação da posição do veículo.

Em resumo, a localização em um AV é um processo complexo e envolve a integração de várias tecnologias e técnicas para garantir uma precisão adequada. Cada uma dessas tecnologias tem suas próprias vantagens e limitações, e é importante que o sistema de localização do veículo seja capaz de lidar com diferentes condições e situações para garantir um desempenho confiável e seguro.

2.2.1 Mapa topológico

Um mapa topológico é uma representação de um ambiente por meio de um grafo de forma que cada nó corresponda a um ponto de referência (waypoints) do ambiente e aos arcos que representam as rotas. Esses mapas reduzem o problema de navegação ao encontrar uma rota de um nó inicial até um nó final, percorrendo os nós que compõem o caminho um a um. A vantagem dos mapas topológicos é que, para localizar a posição de um veículo em um ambiente, não depende apenas da odometria, simplesmente precisa de **waypoints** suficientes para desenhar o mapa.

Por outro lado, a navegação é classificada pelos autores conforme o tipo de informação utilizada pelo veículo para se deslocar no ambiente, dentre estes se tem:

- **Navegação topológica, também chamada navegação qualitativa:** É a abordagem mais natural para modelar um sistema de navegação em um robô. A navegação utiliza referências de características ambientais importantes ou marcos (waypoints/marcos). Por exemplo: Siga o corredor e entre pela primeira porta à direita. Geralmente, a representação do conhecimento do ambiente é modelada por grafos de conectividade [81, 82, 83, 84, 85].
- **Navegação topo-geométrica ou navegação híbrida:** Combina informações geométricas simples e informações simbólicas. O modelo de ambiente representa a conectividade entre elementos geométricos e simbólicos [86, 87].
- **Navegação geométrica, também chamada navegação quantitativa:** Este modelo de navegação é baseado em medições geométricas. A posição do robô, os elementos do ambiente e o objetivo são definidos por suas coordenadas. O modelo do ambiente é representado por mapas geométricos e/ou mapas de grade [88, 89, 90].

O mapa topológico é às vezes referido como um caminho de imagem, gráfico visual, mapa de aparência ou mapa topológico baseado em aparência [82]. Comparados aos mapas métricos, os mapas topológicos são simples e compactos, ocupando menos memória do computador e, consequentemente, podem agilizar os processos de navegação computacional. Agora, se uma localização topológica é integrada com imagens, dá origem a uma posição visual [91].

A localização topológica em AV refere-se à capacidade do veículo de se localizar dentro de seu ambiente usando mapas topológicos. A localização topológica normalmente envolve o uso de sensores como câmeras, LIDAR ou radar, que capturam dados sobre o ambiente circundante e algoritmos que analisam esses dados para identificar recursos e pontos de referência distintos que podem ser usados para determinar a localização do veículo no mapa topológico. O veículo pode usar essas informações para planejar sua rota e evitar obstáculos.

Alguns autores utilizam técnicas de localização topológica, que envolvem a criação de um mapa do ambiente e sua utilização para estimar a localização do robô. Por meio de uma câmera, as informações visuais podem ser utilizadas para navegar em ambientes internos ou externos. Por exemplo, Shalev O. [92], concentra-se no uso de uma câmera de visão superior para localizar um robô móvel em um pomar. Os autores usam uma abordagem de localização de Monte Carlo que aproveita a estrutura da copa das árvores como referência para a localização. Esta técnica pode estimar a posição do robô com precisão e baixo custo computacional.

Ming Liu [93] se concentra no desenvolvimento de um mapa topológico de um ambiente usando uma câmera omnidirecional. Os autores propõem um descritor de cores claras para representar diferentes lugares no ambiente e usam uma abordagem baseada em grafos para construir um mapa topológico. Esta técnica pode reconhecer a localização do robô com alta precisão em ambientes difíceis.

Song Xu [94], centra-se na navegação em ambiente interior com uma câmara monocular. Os autores propõem um esquema de Monte Carlo baseado em imagens que combina uma rede de proposta de região eficiente com um algoritmo de prova de importância. Esta técnica pode alcançar uma localização precisa em um ambiente interno complexo. Ele propõe uma estrutura eficiente e robusta para navegação topológica interna que usa um esquema de Monte Carlo baseado em imagens.

Leandro B. [95], propõe uma nova abordagem para a localização de robôs móveis baseada em mapas topológicos e no uso de imagens omnidirecionais. Os autores usam um algoritmo de classificação com opção de rejeição para distinguir entre diferentes regiões topológicas e localizar o robô dentro delas.

Outros autores [95, 92, 93, 94] reúnem algumas técnicas de localização topológica usando o método de Monte Carlo, mapeamento topológico e reconhecimento de cena, para estimar a localização do robô. O reconhecimento de cena envolve a extração de recursos de imagens para identificar a cena ou local. Essas técnicas são baseadas em informações visuais das câmeras, seja na forma de imagens de visão superior ou visualizações omnidirecionais. Em relação às diferenças entre esses autores, cada autor propõe uma abordagem ou método diferente para construir e usar mapas topológicos para localização. Em conclusão, [95] propõe um algoritmo de classificação de opção de rejeição, enquanto [92] usa um algoritmo de localização de Monte Carlo que considera a geometria da copa. Além disso, [93] propõe uma representação de mapa topológico baseada em reconhecimento de cena, enquanto [94] propõe um esquema de Monte Carlo baseado em imagem para navegação topológica interna.

Por outro lado, existem diversos algoritmos que são utilizados em sistemas de localização autônoma, tais como o SIFT, SURF, ORB (*Oriented FAST and rotated BRIEF*), entre outros [96, 97, 98, 99, 100]. Esses algoritmos geralmente utilizam segmentos inva-

riantes e transformação de funções invariantes em escala para detectar pontos de interesse no ambiente e estimar a pose do robô em relação a um mapa pré-construído. Além disso, a segmentação semântica do ambiente é outra técnica utilizada para melhorar a precisão da localização [101, 102, 103]. Mais recentemente, o uso de *Deep Learning* tem sido explorado para aprimorar a segmentação e a detecção de objetos em tempo real [104, 105, 106]. Esses avanços na área de processamento de imagens têm permitido que sistemas de localização autônoma sejam mais precisos e eficientes em diferentes ambientes e situações.

Dito isso, neste trabalho será implementado o algoritmo SURF, na próxima seção seu estudo será expandido, mais especificamente na detecção de vizinhos mais próximos. Ou seja, o processo de combinar duas imagens para determinar sua localização topológica normalmente envolve uma série de etapas:

1. Extração de características: Nesta etapa, características distintas são extraídas de ambas as imagens. Esses recursos podem ser cantos, núcleo, bordas, bolhas ou outros padrões exclusivos de cada imagem, que podem ser usados para estabelecer correspondências entre as duas imagens.

2. Correspondência de características: Em seguida, os recursos extraídos de ambas as imagens são comparados e combinados com base em sua similaridade. Isso geralmente é feito usando algoritmos como o SIFT ou SURF. As correspondências entre as feições nas duas imagens são estabelecidas pela comparação de seus descritores, que são representações numéricas das feições.

3. Estimativa da transformação: Uma vez identificadas as feições correspondentes, estima-se a transformação que mapeia uma imagem na outra. Isso pode ser uma translação, rotação, dimensionamento ou uma combinação deles. A transformação pode ser uma transformação afim 2D, uma transformação rígida 3D ou uma transformação não linear mais complexa.

4. Verificação: No algoritmo SURF, os descritores de pontos de interesse extraídos das imagens de referência e consulta são comparados usando a técnica de vizinhos mais próximos no passo de verificação. Para determinar a similaridade, busca-se o vizinho mais próximo de cada descritor da imagem de referência na imagem de consulta, utilizando distâncias como a Euclidiana ou a Hamming. Se a distância entre o descritor de referência e seu vizinho mais próximo estiver abaixo de um limite pré-definido, considera-se que os pontos de interesse coincidem. Essa verificação permite avaliar o quão bem as imagens se alinham entre si, utilizando recursos adicionais como segmentos de linha ou padrões de textura, e métricas de qualidade, como correlação ou informação mútua.

Em conclusão, se a transformação for precisa, as duas imagens podem ser equivalentes e sua localização topológica pode ser determinada. Esse processo é frequentemente usado em aplicativos de visão computacional, como mosaico de imagem, em que várias imagens são mescladas para criar uma visão panorâmica. Também é usado em aplicativos como rastreamento de objetos, onde a localização de um objeto é determinada com base em sua aparência em vários quadros de um vídeo.

2.3 Técnica Visão Computacional

A VC (*Visão Computacional*) é um campo da engenharia cuja finalidade é extrair informações sobre o mundo físico de imagens por computador. O primeiro passo em cada tarefa atribuída a aplicações baseadas em visão é obter uma descrição das características de uma cena. Isto se refere à representação de aspectos visualmente específicos, tais como: pontos, contornos, contrastes, cantos, bordas do objeto, entre outros.

Desde sua criação, a VC tem sido uma área de pesquisa em expansão a qual sempre está evoluindo [34], permitindo o desenvolvimento de aplicações muito diversas em diferentes campos, tais como indústria, robótica, medicina, aplicações de segurança ou no desenvolvimento de sistemas para a melhoria da qualidade de vida de pessoas com algum tipo de deficiência visual.

Vários autores [31, 32, 36, 34, 35] dizem que a estrutura para qualquer processo de reconhecimento da implementação da VC pode ser dividida como mostra a Figura 2, onde cada um é executado consecutivamente para obter o objetivo desejado.

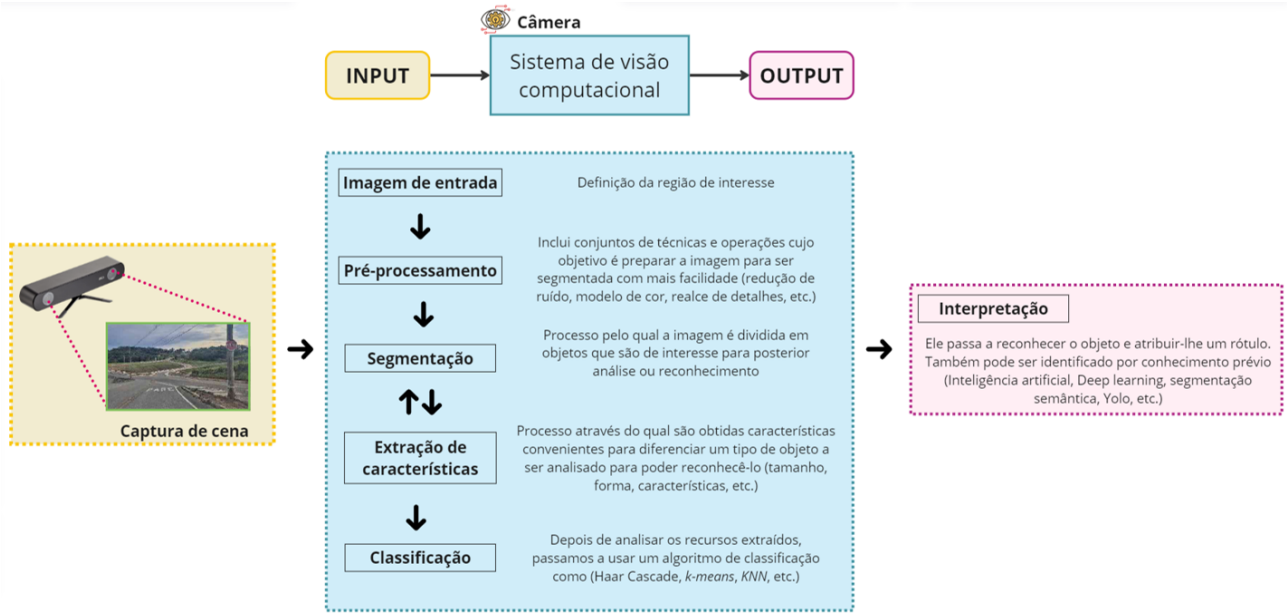

Figura 2 – Etapas da visão computacional.

Estas etapas variam conforme a aplicação a ser realizada, outras etapas ou mesmo

outros métodos podem ser necessários. Em alguns casos, é necessário realizar uma segmentação, cujo objetivo é a separação dos objetos para posterior identificação e descrição após a classificação. Isto pode ser usado para dividir a imagem em diferentes regiões de acordo com uma ou mais características que tenham uma forte relação com objetos reais. Da mesma forma, existem diferentes abordagens para realizar a segmentação, por meio de técnicas baseadas em thresholding, detecção de contorno, crescimento da região, segundo a cor ou textura (segmentação semântica) ou a técnica de correspondência são algumas das mais comumente utilizadas [34, 107, 108, 109].

Para identificar um objeto, é essencial ter uma descrição precisa do mesmo. Para solucionar esse desafio, há uma variedade de algoritmos disponíveis para serem implementados [110, 111, 112]. O algoritmo classificador que, graças às descrições dos diferentes objetos previamente obtidos, atribui um elemento de entrada na etiquetagem a uma categoria específica conhecida. Atualmente, existem várias técnicas de classificação, que podem ser divididas, em princípio, em aprendizagem supervisionado e não supervisionado [113, 114, 115].

Aprendizagem é um termo geral que se refere a um conjunto de técnicas e algoritmos utilizados para ensinar um computador a executar uma tarefa específica. A aprendizagem pode ser supervisionada ou não supervisionada, dependendo da disponibilidade de dados rotulados. Na aprendizagem supervisionada, o algoritmo é treinado com um conjunto de dados rotulados. Isso significa que cada amostra do conjunto de dados é acompanhada de uma etiqueta que indica qual é a classe a que ela pertence. O objetivo é que o algoritmo aprenda a associar os recursos (ou características) de cada amostra com suas respectivas classes, a fim de ser capaz de classificar novas amostras corretamente. A aprendizagem supervisionada é frequentemente utilizada em tarefas de classificação e regressão.

Na aprendizagem não supervisionada, por outro lado, o algoritmo é treinado com um conjunto de dados não rotulados. O objetivo é que o algoritmo aprenda a identificar padrões e estruturas nos dados sem a orientação de um conjunto de rótulos. A aprendizagem não supervisionada é frequentemente utilizada em tarefas de agrupamento (clustering), redução de dimensionalidade e detecção de anomalias.

Além disso, existem muitos algoritmos de classificação que podem ser usados no processo de aprendizado supervisionado. Esses algoritmos são capazes de aprender a partir de um conjunto de dados rotulados para classificar novas instâncias com base nas características aprendidas. Entre os algoritmos mais populares de classificação, temos o KNN (*K Nearest Neighbors*), SVM, Naive Bayes, Árvores de Decisão, Regressão Logística, Rede Neural Artificial, Boosting, Random Forest, Gradient Boosting e XGBoost. Todos esses algoritmos são considerados classificadores, pois são usados para classificar instâncias em diferentes classes com base em características específicas extraídas dos dados. Cada um

desses algoritmos possui suas próprias características e limitações, o que torna importante escolher o algoritmo mais adequado para o problema em questão [116, 117]. E, por outro lado, existem diferentes algoritmos de aprendizado não supervisionado que podem ser utilizados na análise de dados. O K-means é um dos mais conhecidos e utilizados, ele tem como objetivo agrupar dados em clusters de acordo com sua similaridade. Outro algoritmo é o Hierarchical Clustering, que é uma técnica de agrupamento hierárquico onde os dados são organizados em uma estrutura de árvore. O DBSCAN é um algoritmo de clustering que se baseia em densidades de pontos para identificar grupos de dados. Já o t-SNE é uma técnica de redução de dimensionalidade que tem como objetivo representar dados de alta dimensão em um espaço bidimensional ou tridimensional, preservando as relações de similaridade entre os pontos. Por fim, temos o PCA (Principal Component Analysis), que é uma técnica de redução de dimensionalidade que visa encontrar as principais características que explicam a maior variação nos dados, permitindo a representação em um espaço com menos dimensões. Todos esses algoritmos são considerados de aprendizado não supervisionado, pois não requerem a intervenção de um especialista para rotular ou classificar os dados.

No entanto, é importante notar que, embora o SURF em si seja um algoritmo não supervisionado, sua implementação em um sistema de localização pode envolver técnicas de aprendizado supervisionado, como classificação de imagens e detecção de objetos, para melhorar a precisão e robustez do sistema de localização.

Os AV não precisam apenas saber quais objetos percebem em imagens de câmera, mas também precisam saber onde estão localizados. Esse problema é conhecido como detecção e localização de objetos. Durante este trabalho o algoritmo SURF será implementado, com o fim de realizar a detecção e extração de características no ambiente determinado. Da mesma forma, o SURF não se encarrega de localizar os objetos, neste caso só serve para extrair as características comuns, e depois integra outra série de passos explicados na seção 2.4.3 a seguir.

2.3.1 Modelos de cores

Os espaços de cores no processamento de imagens destinam-se a facilitar a especificação de cores de alguma forma padrão. Além disso, os dispositivos de captura e exibição de imagens capturam e geram cores misturando as cores primárias de luz, vermelho, verde e azul, com diferentes intensidades, modelo RGB. Porém, a relação entre a quantidade de cada cor não é intuitiva ao olho humano. Por esse motivo, surgiram outros modelos de cores, como o HSV, baseado na forma como as cores são organizadas quando são percebidas pelas pessoas em termos de atributos que definem cor, tom, saturação e brilho [118, 119].

Um modelo de cores é uma representação do espectro de cores que utiliza múltiplas dimensões. A maioria dos modelos modernos tem três dimensões, como o RGB, e podem

ser visualizados em formas 3D, enquanto outros modelos possuem mais dimensões, como o CMYK. Existem vários modelos de cores amplamente utilizados em ferramentas de design digital e linguagens de programação, incluindo RGB, tons de cinza, CIE XYZ, CIE1976 LUV, YcbCr e CIE Lab. Embora todos eles usem as mesmas cores primárias RGB, eles exibem o espectro de cores em dimensões diferentes, o que permite uma ampla gama de opções de cores. Além disso, os modelos de cores são úteis para melhorar a aparência de imagens.

Transformar e saber os modelos de cores pode ser uma etapa essencial em muitos algoritmos de visão computacional porque ajuda a normalizar os dados de entrada e torná-los mais consistentes em diferentes condições de iluminação, configurações de câmera e fontes de imagem [120]. Por exemplo, considere um algoritmo de reconhecimento de objetos que precisa identificar um objeto específico em uma imagem. Se as condições de iluminação da imagem forem diferentes das condições de iluminação usadas para treinar o algoritmo, as cores da imagem podem ser alteradas ou distorcidas. Isso pode dificultar a identificação correta do objeto pelo algoritmo. Ao transformar as cores da imagem, o algoritmo consegue normalizar os valores das cores e torná-los mais consistentes com os dados de treinamento. Isso pode melhorar a precisão e a robustez do algoritmo e torná-lo mais eficaz em uma ampla gama de situações.

As transformações de cores também podem ser usadas para outras finalidades, como aprimorar ou suprimir determinados recursos de cores em uma imagem, ajustar o equilíbrio ou contraste de cores ou reduzir ruídos ou artefatos na imagem. No geral, a transformação de cores é uma ferramenta importante da visão computacional e pode ajudar a melhorar o desempenho e a confiabilidade de muitos algoritmos de visão [121, 1].

2.3.1.1 Modelos RGB: HSV, HSL, HSI e HSY

RGB é um modelo de cor com três dimensões (vermelho, verde e azul) que se misturam para produzir uma cor específica. Ao definir as cores nessas dimensões, deve-se saber a sequência de cores no espectro de cores, por exemplo, que uma mistura de 100% de vermelho e verde produz o amarelo. O modelo de cores RGB é geralmente representado como um cubo, atribuindo as dimensões vermelho, verde e azul aos eixos x, y e z no espaço 3D [119, 122].

Braheem [119] fala que os modelos HSV, HSL, HSI e HSY não são incluídos como seus próprios espaços de cores. No entanto, eles aparecem nos modos de mesclagem e seletores de cores, portanto, uma breve visão geral (RGB-HSV):

- **Matiz/Hue**: O tom de uma cor, ou seja, vermelho, amarelo, verde, etc. O matiz do Krita é medido em 360 graus, sendo 0 o vermelho, 120 o verde e 240 o azul.

- **Saturação**: A saturação da cor é a pureza e a intensidade de uma cor exibida em uma imagem. A saturação varia de 0 (cinza) a 100 (cor pura).
- **Valor**: Às vezes conhecido como brilho, é uma medição que representa quanto o píxel precisa iluminar, medido de 0 a 100.
- **Leveza**: Onde uma cor se alinha entre branco e preto. Esse valor é não linear e coloca todas as cores mais saturadas possíveis em 50. Vária de 0 a 100.
- **Intensidade**: Semelhante à leveza, exceto que reconhece que o amarelo (1,1,0) é mais claro que o azul (0,0,1). Vária de 0 a 100.
- **Luma**: Semelhante à luminosidade e intensidade, exceto que pondera os componentes vermelho, verde e azul com base em medições da realidade de quanta luz uma cor reflete para determinar sua luminosidade. Vária de 0 a 100.

2.3.1.2 Modelo de cor em tons de cinza

Este espaço de cores registra apenas valores de cinza. Isso é útil, pois registrando apenas valores de cinza, ele precisa apenas de um canal de informação, o que no que lhe concerne significa que a imagem fica muito mais leve no consumo de memória [119, 122, 118].

Isso é útil para texturas, mas também para qualquer outra aplicação que precise permanecer em tons de cinza, como quadrinhos em preto e branco.

2.3.1.3 Modelo de cor CIE XYZ

Em 1931, o CIE (Instituto de Cor e Luz) estudava a percepção humana das cores. Ao fazer isso, criaram os primeiros espaços de cores, sendo o XYZ o que melhor se aproxima da visão humana. **Y** representa verde, **Z** azul e **X** vermelho [119, 123].

XYZ é usado como referência de linha de base para todos os outros perfis e modelos. Todas as conversões de cores são feitas em XYZ e todas as coordenadas dos perfis correspondem a XYZ.

2.3.1.4 Modelo de cor CIE 1976 LUV

O espaço de cores CIE 1976 (LUV), utilizado na colorimetria, é um modelo adotado pela Comissão Internacional de Iluminação (CIE) em 1976. Ele foi desenvolvido como uma transformação do espaço de cor 1931 CIE XYZ, com o objetivo de criar uniformidade perceptual. O modelo é amplamente utilizado em aplicações como gráficos de computador que envolvem luzes coloridas. Embora as misturas aditivas de diferentes luzes coloridas caiam em uma linha no diagrama de cromaticidade uniforme do CIELUV (conhecido como CIE 1976 UCS), tais misturas não caem, ao contrário da crença popular, ao longo

de uma linha no espaço de cores CIELUV, a menos que as misturas sejam constantes em leveza. O espaço de cores CIELUV é considerado um modelo de cor simples para cálculo e ajuda a alcançar uma melhor aparência da imagem [124].

2.3.1.5 Modelo de cor YcbCr

Representa luminosidade, vermelho-croma e azul-croma. Onde YCrCb significa:

- **Y**: Luma/Luminosidade, portanto, a quantidade de luz que uma cor reflete.
- **Cr**: Croma Vermelho. Esse valor mede quão vermelha é uma cor em relação ao quão verde ela é.
- **Cb**: Croma Azul. Esse valor mede quão azul é uma cor contra quão amarela ela é.

Este espaço de cores é frequentemente usado em fotografia e em implementações (corretas) de JPEG. Como o ser humano, é muito mais sensível à claridade das cores, portanto, o JPEG tenta compactar os canais Cr e Cb e deixar o canal Y com qualidade total [119, 122, 123].

2.3.1.6 Modelo de cor CIE Lab

O modelo de cor CIE Lab representa

- **L**: Leveza, semelhante à luminosidade.
- **a**: É a medida de quão magenta é uma cor em relação a quão verde ela é.
- **b**: Uma medida de quão amarela é uma cor em relação a quão azul é uma cor.

Supõe-se que Lab seja uma variedade mais compreensível de XYZ e o mais *completo* de todos os espaços de cores. É frequentemente usado como um espaço intermediário de cores na conversão, mas ainda mais como o espaço de cores correto para fazer o balanceamento de cores. É muito mais fácil ajustar o contraste e o tom de cores em Lab [119, 122, 124, 118, 123].

Neste trabalho serão feitas diferentes comparações de modelos de cores (Seção 4.1.1. de resultados) para identificar se a transformação de uma imagem RGB em outro modelo aumenta o número de características ao implementar a metodologia proposta que utiliza visão computacional.

2.3.2 Pontos característicos e descritores

Para descrever objetos em imagens, é comum utilizar descritores que se baseiam na extração de pontos característicos ou pontos-chave. No entanto, para que esses pontos sejam úteis, é preciso que atendam a certos requisitos. Primeiramente, eles devem ser representativos, ou seja, capazes de serem localizados mesmo diante de transformações geométricas ou fotométricas. Além disso, é importante que esses pontos sejam distinguíveis entre si para que possam ser usados para identificar objetos em diferentes imagens [125]. A detecção de pontos característicos implica a individualização de pontos que, devido às suas características, ajudam a definir as imagens. Um detector ideal localiza esses pontos repetidamente, independentemente do ponto de vista, sendo confiável antes das transformações da imagem [126, 127].

A descrição desses pontos consiste na extração de informações relevantes e distintas da região que os rodeia. Isso é fundamental para que a estrutura do objeto possa ser reconhecida se for encontrada em outra imagem [128, 129]. Dessa forma, a descrição desses pontos é uma etapa crucial na correspondência de pontos-chave entre imagens, que permite a localização e o reconhecimento de objetos. Por meio da extração e descrição dos pontos característicos, é possível realizar tarefas como a detecção e o rastreamento de objetos em vídeos, bem como a criação de mapas 3D a partir de imagens [130, 131].

Em outras palavras, o descritor visa fornecer uma descrição única e robusta de um conjunto e descreve a distribuição da intensidade do conteúdo dentro do ponto de interesse dos pontos vizinhos. O descritor é gerado com base na área circundante de um ponto de interesse; portanto, um vetor descritivo é obtido para o ponto de interesse. Após obter a imagem com a obtenção de pontos característicos (descritores), qualquer algoritmo de classificação e detecção pode ser implementado.

De acordo com SHIKROT e Intel [1, 121] os pontos e descritores característicos são usados no processo de extração de características em visão computacional e processamento de imagens. No entanto, eles servem a propósitos diferentes e possuem características distintas.

Os pontos característicos, também conhecidos como pontos-chave ou pontos de interesse, são locais específicos em uma imagem que são facilmente reconhecíveis e distinguíveis de seus arredores. Esses pontos geralmente são selecionados com base em determinados critérios, como alto contraste, alta curvatura ou padrões de textura distintos. Os pontos característicos são usados para representar os recursos exclusivos de uma imagem e podem ser usados para alinhar imagens, rastrear objetos e executar tarefas de reconhecimento de imagem.

Os descritores, por outro lado, são representações numéricas da estrutura da imagem local em torno de um ponto característico. Eles são usados para descrever a aparência

da região ao redor de um ponto característico e fornecem uma maneira de comparar e combinar imagens com base em seus recursos locais. Os descritores são calculados analisando os valores de intensidade ou orientações de gradiente dos píxeis na região ao redor do ponto característico e podem ser representados usando várias técnicas, como histogramas, wavelets ou transformada de Fourier.

Em resumo, os pontos característicos são localizações específicas em uma imagem que representam suas características únicas, enquanto os descritores são representações numéricas da estrutura local da imagem em torno de um ponto característico. Juntos, os pontos característicos e os descritores formam a base para muitas aplicações de visão computacional e processamento de imagens, como reconhecimento de objetos, alinhamento de imagens e rastreamento de movimento.

2.3.3 Extração de características

Uma imagem precisa ser descrita (apontar as características mais importantes que permitem distingui-las de outras imagens) a fim de compará-la com outras imagens e obter a semelhança que tem (partes comuns). Uma das formas de descrever imagens é detectando e extraindo os "pontos de interesse"(ou características) visuais que elas contêm. Existem muitos tipos de recursos dependendo do método utilizado.

Na literatura, diferentes soluções são propostas para a extração de características, dependendo do tipo de recurso e da abordagem utilizada. Por exemplo, o algoritmo Canny [132, 133] é utilizado para detecção de bordas em imagens, enquanto o YOLO [37, 134, 135] é um algoritmo de detecção de objetos em tempo real. Além disso, existem os descritores SIFT e SURF [136, 97, 98], que extraem pontos de interesse ou pontos invariantes nas imagens.

Uma técnica comumente utilizada para aplicações em tempo real é o SURF, reconhecido como um método eficiente [136, 97, 137]. O SURF permite eliminar a informação de fundo, sendo útil quando há interferência na detecção de objetos de interesse. Além disso, é capaz de detectar pontos característicos repetidamente, mesmo com variações de iluminação e perspectiva, sendo valioso para rastreamento e reconhecimento de objetos. No entanto, a detecção de pontos característicos enfrenta desafios em ambientes com ambiguidade visual, levando à proposição de soluções para garantir uma detecção eficaz de objetos em tempo real.

O ambiente pode causar problemas de ambiguidade, representando um grande desafio na implementação de softwares capazes de detectar mudanças na luz. Para lidar com esses problemas, diferentes autores implementam diversas soluções. Hailing Zhou et.al [138] apresenta um sistema de rastreamento de UAV e usa um método de corte gráfico para extrair a estrada especificada no ROI. A técnica de recurso rápido é usada para coletar

as características numéricas e o rastreador de recursos Kanade-Lucas-Tomasi (KLT) é aplicado para estimar o movimento. Por sua vez, Aguilar et.al [139] propuseram um sistema de detecção de objetos e prevenção de colisões baseado em micro-áreas em tempo real. Neste método, o descritor SURF é utilizado para extrair os pontos característicos do obstáculo. As características extraídas são então comparadas entre as imagens do banco de dados sem aumentar o custo computacional. Esse método é testado em tempo real em UAVs de baixo custo e os resultados mostram eficácia na detecção e prevenção de colisões.

Por outro lado, Kaur & Bathla [140] propôs um módulo para estabilização de vídeo e detecção de objetos em movimento. SIFT e SURF usam-no como descritores. Ao estimar parâmetros da câmera, este método tende a encontrar objetos em movimento usando filtro de Kalman. Reconhece o objeto e reorganiza o movimento para mover o objeto para uma posição estabilizada. Aqui SIFT é usado para estabilizar o vídeo e detectar objetos em movimento. A extração de características e o algoritmo de correspondência de descritores são usados para estimativa de movimento da câmera.

Também na área da medicina, existem trabalhos utilizando esse tipo de algoritmos, Liu et al. [141], propôs um método para implementar algoritmos como SIFT, SURF, ORB para realizar uma cirurgia minimamente invasiva, o cirurgião pode alcançar uma reconstrução tridimensional através das imagens obtidas pelo endoscópio no intestino grosso, permitindo restaurar a cena tridimensional da área a ser operada e rastrear o movimento da superfície do tecido. Como resultado desta pesquisa, obteve-se que o melhor algoritmo para identificar os pontos de características correspondentes com sucesso foi o SURF.

Além disso, este algoritmo também é implementado no uso de detectores de características no vSLAM (*Visual simultaneous localization and mapping (vSLAM)*) para cenários subaquáticos [142] e em outras aplicações [143, 144, 145, 79]. No entanto, a questão permanece se esse algoritmo pode ser implementado em uma aplicação para um AV. Em seguida é detalhado seu procedimento.

2.3.3.1 Algoritmo SURF

A correspondência de imagens é um campo em constante evolução, e os métodos de correspondência de características são amplamente utilizados devido à sua capacidade de lidar com variações na escala, rotação e iluminação (FAST (Features from Accelerated Segment Test) [146], HARRIS [147], SIFT, SURF). Entre esses métodos, o SIFT foi proposto em 1999 e melhorado em 2004, tornando-se um dos algoritmos mais populares para descrever características invariantes de escala. No entanto, devido à sua alta complexidade de tempo e grande quantidade de dados computacionais, muitos pesquisadores propuseram melhorias no algoritmo [142, 148].

Uma dessas melhorias foi o algoritmo SURF, proposto por Bay et al. em 2006

[149]. O SURF utiliza o conceito de características Harr e imagem integral, o que acelera significativamente a velocidade de execução do programa em comparação com o SIFT. O SURF também apresenta desempenho semelhante em relação à mudança de iluminação e invariância da mudança do ângulo de visão em relação ao SIFT e são várias vezes mais rápido [150, 151, 152].

O algoritmo SURF também é amplamente utilizado na localização topológica em ambientes desconhecidos. A extração de características robustas de imagens do ambiente é uma etapa crucial para a construção de um mapa topológico, e o SURF apresenta características invariantes a escala, rotação e mudanças na iluminação, tornando-o ideal para essa tarefa. Os descritores de características baseados em Haar wavelets e na matriz Hessiana também contribuem para a robustez do algoritmo em relação a ruído e variações de contraste na imagem.

Em resumo, o algoritmo SURF é uma melhoria significativa em relação ao SIFT e é amplamente utilizado em várias aplicações, incluindo correspondência de imagens e localização topológica. Assim, o SURF pode ser utilizado como uma ferramenta para a construção de um sistema de localização topológica, sendo capaz de detectar e descrever características de forma robusta e invariante em imagens do ambiente. Na Figura 3, são mostrados os passos que utiliza para executar o algoritmo de navegação para tentar detectar as características de uma cena.

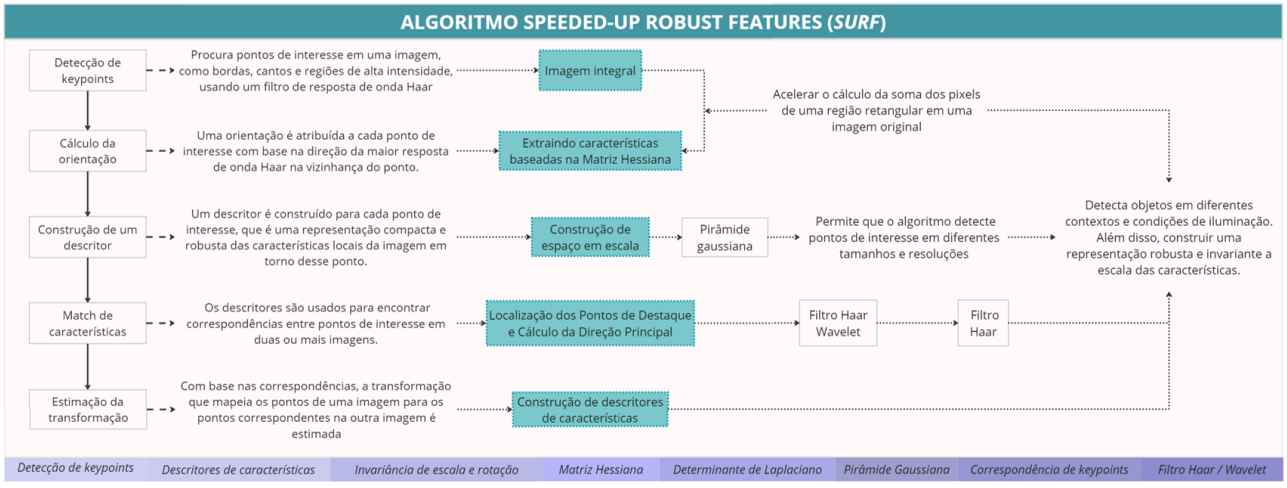

Figura 3 – Etapas do algoritmo SURF.

Independentemente de ser um algoritmo proposto por anos anteriores, hoje ainda está sendo trabalhado, pois promete ótimos resultados, em diferentes aplicações. O principal interesse da abordagem SURF está no seu rápido cálculo dos operadores que utilizam filtros de caixa, permitindo aplicações em tempo real, como rastreamento e reconhecimento de objetos, calibração da câmera, reconstrução 3D, gravação de imagens, entre outros. Este algoritmo busca imagens discretas e aplica imagens integrais, a extração de pontos característicos baseia-se na teoria do espaço de escala, permitindo encontrar as

coordenadas dos pontos característicos, detectando os pontos extremos locais da imagem, ou seja, o ponto local mais brilhante ou mais escuro, segundo isso, cumpre uma série de passos como:

1. Busca por pontos de interesse através da construção da MH (*Matriz Hessiana*): é uma matriz que organiza todas as derivadas parciais de segunda ordem de uma função, neste caso as características distintas da imagem são diminuídas.
2. Invariância de escala: a característica de cada ponto de interesse é representado por um vetor de características, responsável por finalizar a igualdade das características da imagem.
3. Seleção da direção principal dos pontos característicos: Após a obtenção dos descritores, vetores são combinados entre diferentes imagens. Essa coincidência é muitas vezes baseada em uma distância entre os vetores.
4. Operador de descrição de ponto característico.

Diante disso, a seguir estão as principais características que o SURF implementa para realizar a extração e detecção de características, proposta na Figura 3.

2.3.3.1.1 Imagem integral

A imagem integral ou tabela de áreas somadas foi introduzida em 1984 [153]. É usado como uma forma rápida e eficiente de calcular a soma de valores (valores de píxel) em uma determinada imagem, ou um subconjunto retangular de uma grade (a imagem dada). Ela também pode ser usada para calcular a intensidade média numa determinada imagem.

A Figura 4 mostra um exemplo da representação de píxeis de uma imagem, onde uma coordenada de píxel (2,2) é tomada, ao aplicar a imagem integral que cada píxel corresponde o número 38, representa a soma de todos os valores superiores esquerdos.

Explicado matematicamente, permite um cálculo rápido de filtros de convolução do tipo caixa, os quais são usados para suavizar e desfocar imagens calculando a média dos píxeis vizinhos. A entrada de uma imagem integral $I_{\sum}(x)$ em um local $x = (x, y)^T$ representa a soma de todos os píxeis da imagem de entrada numa região retangular formada pela origem e x (Equação 2.1).

$$I_{\sum}(x) = \sum_{i=0}^{i \leq x} \sum_{i=0}^{i \leq x} I(i, j) \tag{2.1}$$

Com $I_{\sum}$ calculado, apenas quatro adições são necessárias para calcular a soma das intensidades sobre qualquer área retangular vertical, independentemente do seu tamanho.

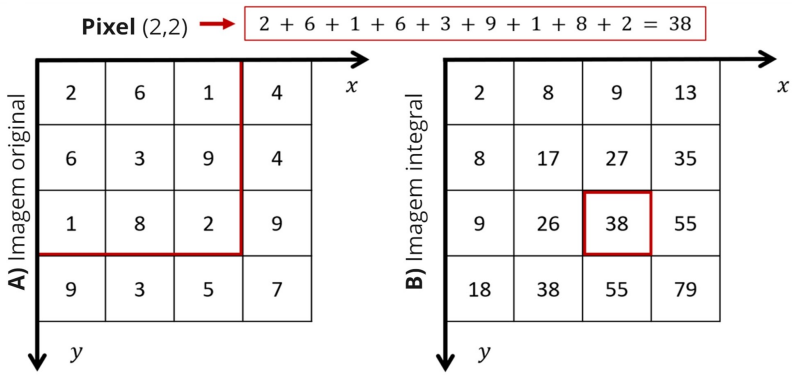

Figura 4 – Representação de imagem integral em um píxel de (2,2).

Além disso, se quiser encontrar o valor para qualquer píxel, se pode calcular pela Equação 2.2:

$$s(x, y) = I(x, y) + s(x - 1, y) + s(x, y - 1) - s(x - 1, y - 1) \tag{2.2}$$

onde $s(x, y)$ é um ponto dentro da imagem integral e $I(x, y)$ é o mesmo ponto dentro da imagem original.

Em conclusão, a imagem integral é uma matriz bidimensional com as mesmas dimensões da imagem original, na qual cada elemento armazena a soma acumulada de todos os pixels acima e à esquerda dele na matriz da imagem original. Essa soma acumulada é calculada em tempo constante, independentemente do tamanho da área da imagem considerada.

O cálculo da imagem integral é usado pelo SURF para acelerar o cálculo de características locais de interesse, como descritores de pontos-chave, a partir de uma janela retangular. Em vez de calcular a soma de intensidades dos pixels dentro da janela para cada ponto de interesse, o SURF usa a imagem integral para calcular essa soma em tempo constante, o que torna o algoritmo mais eficiente e escalável.

2.3.3.1.2 Extraindo características baseadas na matriz Hessiana

Após a obtenção da imagem integral, é feita uma aproximação da matriz Hessiana para detectar pontos de interesse na imagem. Essa matriz é representada por um array numérico bidimensional de tamanho m**x**n, onde cada dois elementos representam um píxel, e o valor desses elementos é dado por uma faixa de 0 a $a(2^n - 1)$, em que a é um fator de escala e n é o número de bits envolvidos. A matriz Hessiana é construída a partir das segundas derivadas parciais de uma função e é usada no SURF para detectar características invariantes a escala, fornecendo informações sobre a curvatura da função em cada ponto.

Essa representação numérica da matriz Hessiana permite que oSURF detecte características robustas e invariantes a escala em imagens de grande porte. Ao encontrar pontos de interesse invariantes a escala, o SURF consegue realizar tarefas como reconhecimento de objetos, correspondência de imagens e navegação visual.

Além disso, a representação dos píxeis da imagem e suas cores, transparências e outras propriedades é realizada por meio de matrizes bidimensionais transpostas onde cada uma delas expressa a intensidade de uma propriedade. A mais comum é a representação de uma matriz simétrica que se representa como uma matriz quadrada $m = n$ e é igual à sua transposição (Equação 2.3).

$$A = A^T \tag{2.3}$$

SURF utiliza a MH devido ao seu bom desempenho em tempo de computação e precisão. Em vez de usar uma medida diferente para selecionar a localização e a escala (detector de Hessian-Laplace), SURF depende do determinante da MH para ambos. Seja $f(x, y)$ uma função diferenciável de segunda ordem, a MH é uma função e as derivadas parciais são compostas da seguinte forma, Equação 2.4:

$$H(f_{x,y}) = \begin{bmatrix} \frac{\delta^2 f}{\delta x^2} & \frac{\delta^2 f}{\delta x \delta y} \\ \frac{\delta^2 f}{\delta x \delta y} & \frac{\delta^2 f}{\delta y^2} \end{bmatrix} \tag{2.4}$$

Obtendo como discriminantes da MH o seguinte (Equação 2.5):

$$det(H) = \frac{\delta^2 f \delta^2 f}{\delta x^2 \delta y^2} - \frac{\delta^2 f}{\delta x \delta y} \tag{2.5}$$

O valor discriminante é o autovalor da matriz H, e o símbolo discriminante usado para determinar se é um valor extremo. Se $det(H) > 0$, pode-se obter que o ponto (x, y) é um ponto de valor extremo local.

Para adaptá-la a qualquer escala, filtra a imagem por um Kernel Gaussiana, então dado um ponto $X = (x, y)$, a MH $H(x, \sigma)$ em x escala σ é definida como, Equação 2.6:

$$H_{(x,\sigma)} = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \tag{2.6}$$

onde $L_{xx}(x, \sigma)$ esta a convolução da derivada Gaussiana de segunda ordem com imagem I no ponto x, e o mesmo para $L_{xy}(x, \sigma)$ e $L_{yy}(x, \sigma)$. Os filtros Gaussiana são ideais para análise de espaço em escala, mas, na prática, devem ser discretos. Isso leva a uma perda de repetibilidade de rotações de imagem em torno de múltiplos ímpares de $\frac{\pi}{4}$. Essa fraqueza vale para os detectores baseados em Hessian, em geral. No entanto, os detectores ainda funcionam bem e a ligeira queda no desempenho não compensa a vantagem das convoluções rápidas proporcionadas pela discretização.

O algoritmo SURF utiliza a detecção de pontos de interesse em uma imagem por meio de uma análise baseada em filtros de escala e na detecção de descritores de características locais. Para a análise dos descritores de características locais, SURF utiliza um método de comparação de vetores chamado "Descritores SURF"que se baseia em uma combinação de filtros Gaussianos e diferenciais para extrair informações da imagem em diferentes escalas e orientações. Esse método é robusto a ruídos e mudanças de iluminação, permitindo a identificação de pontos de interesse em diferentes condições. Bay et al [149] apontaram que a análise Gaussiana requer discretização e recorte da imagem e, mesmo que a imagem seja amostrada com filtragem Gaussiana, ocorrerá aliasing. Assim, se pode usar filtragem de quadros em vez de filtragem Gaussiana e usar imagens integrais para aumentar a convolução para acelerar a computação.

O cálculo do determinante da MH no algoritmo SURF envolve a aplicação da convolução com o kernel Gaussiano, seguida pela derivada de segunda ordem. Além disso, o SURF utiliza filtros de caixa, que calculam a média dos valores dos pixels em uma janela retangular ao redor de cada pixel da imagem. Essa filtragem é realizada para suavizar a imagem, eliminando ruídos e irregularidades. O tamanho da janela considerada para o cálculo da média é conhecido como tamanho do filtro. Uma vantagem do SURF é que as derivadas Gaussianas de segunda ordem podem ser aproximadas computacionalmente com baixo custo usando imagens abrangentes. Essas imagens abrangentes são aquelas que incluem uma variedade de elementos ou informações relevantes em um único quadro, permitindo uma compreensão mais completa e profunda do que está sendo retratado. Essa capacidade contribui para a velocidade e eficiência do algoritmo SURF (Figura 5 e 6).

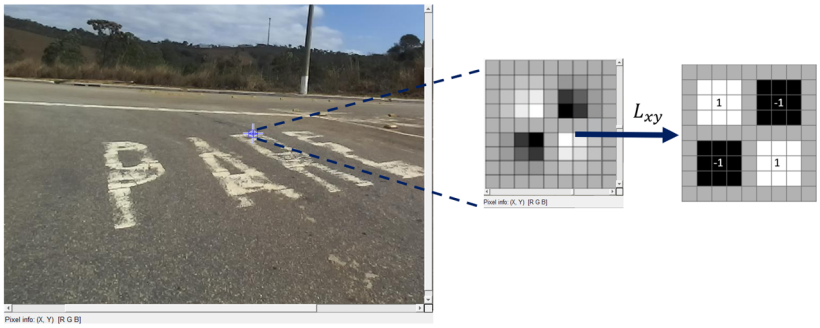

Figura 5 – Derivativo parcial Gaussiana em XY. Adaptado de [1].

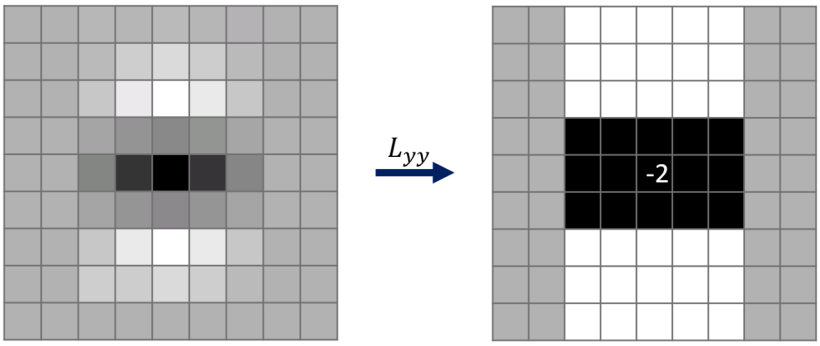

Figura 6 – Derivativo parcial Gaussiana em Y. Adaptado de [1].

.

Por outro lado, para equilibrar o erro entre o valor exato e o valor aproximado, são utilizados pesos que variam conforme a balança. Os filtros de caixa de tamanho 9x9 nas Figuras 5 e 6 são aproximações das derivadas gaussianas de segunda ordem com um desvio padrão (σ) de 1.2. Essas aproximações são denotadas como D_{xx}, D_{yy} e D_{xy}. Agora podemos representar o determinante discriminante da matriz H como mostrado na Equação 2.7.

$$det_{(H_{approx})} = D_{xx}D_{yy} - (wD_{xy})^2; w = 0.9 \tag{2.7}$$

Isso significa que, em aplicações práticas, se usa um 0.9 constante para indicar que seu coeficiente de peso relativo não terá um grande impacto nos resultados. Da mesma forma, o parâmetro da MH pode ser modificado dependendo da aplicação, pois quanto mais aumenta, mais pontos característicos da imagem poderá reconhecer, segundo esses autores [154, 155, 144].

2.3.3.1.3 Construção de espaço em escala

Para que a escala da imagem seja invariável e adapte-se à mudança de escala objetiva em diferentes imagens, é necessário construir um espaço de escala para extrair os pontos característicos do SURF. A pirâmide de imagem é uma forma de representação de imagem em múltiplas escalas. Para obter os pontos extremos da imagem em diferentes escalas através do discriminante da MH, uma pirâmide de imagens dimensionadas é construída por um método semelhante ao SIFT (Scale-invariant feature transform), e o espaço de escala é dividido em várias ordens (oitava), cada estágio armazena imagens com diferentes graus de desfoque obtidos após filtrar a imagem de entrada filtrando blocos de diferentes tamanhos. No entanto, o tamanho da imagem no algoritmo SURF é sempre o mesmo, mas o tamanho do modelo do filtro de quadro em diferentes estágios é diferente.

Além disso, pirâmides gaussianas de imagens são comumente usadas para implementar espaços de escala (Figura 7). As imagens são repetidamente suavizadas como uma

Gaussiana e depois reduzidas para atingir um nível de pirâmide mais alto. O SURF não precisa aplicar iterativamente o mesmo filtro à saída de uma camada filtrada anteriormente devido a filtros de quadro (são uma técnica de processamento de imagens utilizada para filtrar ou destacar informações específicas de uma imagem, ou vídeo. Esses filtros podem ser aplicados em tempo real ou em imagens/vídeos pré-gravados) e imagens integrais; em vez disso, pode aplicar esses filtros de qualquer tamanho exatamente na mesma velocidade diretamente à imagem original, mesmo simultaneamente. Como resultado, o espaço de escala é examinado aumentando o tamanho do filtro em vez de reduzir iterativamente o tamanho da imagem.

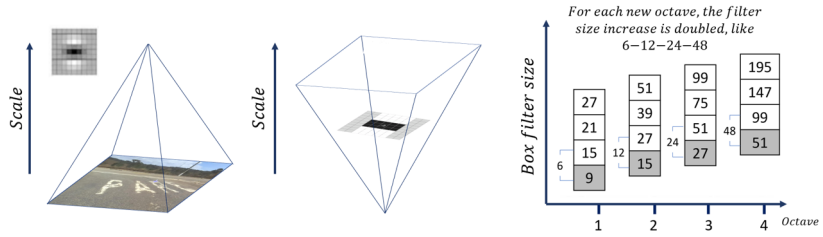

Figura 7 – Pirâmide gaussiana para fazer função de extração, a primeira pirâmide é a base do algoritmo SIFT convertido para a segunda escala implementando SURF. Evidenciando assim a caixa de filtro da imagem.

Ou seja, para cada oitava adicional, o tamanho do filtro é duplicado, e os intervalos de amostragem para extrair os pontos de interesse também podem ser duplicados, permitindo que o filtro seja ampliado a um custo constante. A Supressão não máxima (Non-Maximum Suppression - NMS) é uma técnica usada em visão computacional e detecção de objetos para reduzir a quantidade de detecções redundantes ou sobrepostas, é realizada em uma vizinhança de 3**x**3**x**3 para identificar pontos de interesse na imagem e nas escalas.

Os números no fundo, cinza, representam o tamanho do modelo do filtro da caixa. Caso o tamanho da imagem seja consideravelmente maior do que o tamanho do modelo, é possível aumentar a ordem do filtro para obter resultados mais precisos. Para um modelo com tamanho NxN, a escala correspondente é determinada por $\sigma = 1.2x9/N$.

Após obter o valor extremo da escala utilizando a MH, é realizada uma comparação nas proximidades tridimensionais de 3**x**3**x**3 para verificar se o ponto de valor extremo é um ponto característico candidato. Se o valor extremo permanecer como o valor máximo ou mínimo, então ele é considerado um ponto característico. Em seguida, operações de interpolação são realizadas no espaço de escala e espaço de imagem para obter posições estáveis dos pontos característicos e valores de escala.

Em contraste com a abordagem de redução iterativa do tamanho da imagem (à esquerda na figura), o uso de imagens integrais permite que o filtro seja dimensionado a um custo constante (à direita na figura). Isso torna o processo mais eficiente e escalável

para imagens maiores. Essa abordagem evita a necessidade de reduzir repetidamente o tamanho da imagem, resultando em uma análise mais rápida e eficaz.

2.3.3.1.4 Localização dos pontos de destaque e cálculo da direção principal

A criação do descritor SURF é realizada em duas etapas. O primeiro passo é definir uma orientação reprodutível com base em informações de uma região circular ao redor do KP (*Pontos chave, KPobjeto, Keypoints*). Em seguida, constrói uma região quadrada alinhada com a orientação selecionada e extrai o descritor SURF a partir dele.

Para ser invariante à rotação, SURF tenta identificar uma orientação reprodutível para pontos de interesse (Figura 9). Para conseguir isso:

1. No algoritmo SURF, o filtro Haar Wavelet é usado para realizar a detecção de características. O filtro Haar é aplicado em diferentes posições da imagem, e a resposta nas direções x e y é usada para determinar a presença ou ausência de características em uma determinada posição. Quando o filtro é aplicado em uma posição onde a resposta nas direções x e y corresponde ao branco 1 e preto -1, isso significa que há uma mudança brusca no valor do pixel naquela posição. Essa mudança pode ser interpretada como uma borda ou um canto na imagem, que são características importantes para a detecção de pontos de interesse (Figura 8).

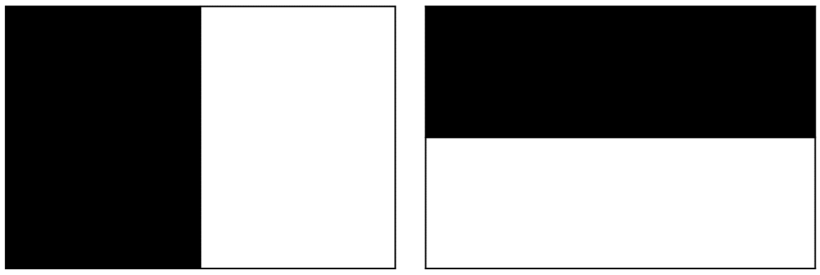

Figura 8 – Características do Haar com diferentes posições onde a resposta nas direções x (esquerda) e y (direita) corresponde ao branco 1 e preto é -1. Representação do Haar Wavelet no SURF.

2. Na etapa de detecção de KP, o SURF utiliza uma aproximação da resposta da convolução com filtros Gaussianos em diferentes escalas e orientações para identificar pontos de interesse (Figura 9). Para cada KP, é estimada uma orientação dominante para que o descritor de características seja invariante a rotações da imagem.

   Para estimar a orientação dominante, é construído um histograma das orientações dos gradientes em torno do KP. SURF utiliza o operador de Sobel para calcular os gradientes na horizontal e vertical em uma janela ao redor do KP. Os gradientes são ponderados pela magnitude do gradiente e por uma função gaussiana que atenua a contribuição dos gradientes mais distantes do KP.

O histograma é construído a partir dos gradientes ponderados, com bins de orientação que vão de 0 a 360 graus. A orientação dominante é escolhida como o pico do histograma, ou seja, a orientação que tem a maior soma das magnitudes dos gradientes ponderados.

Caso haja mais de um pico no histograma (por exemplo, se a distribuição dos gradientes é bimodal), SURF estima múltiplas orientações dominantes. Para cada orientação dominante, é calculado um novo KP com a mesma posição e escala, mas com a orientação alterada para a orientação dominante estimada. O descritor de características é calculado em torno desse novo KP, resultando em uma descrição invariante as rotações da imagem.

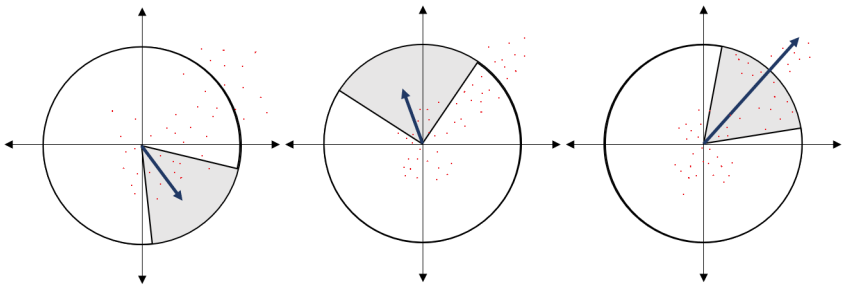

Figura 9 – Orientação dos pontos a serem detectados, SURF. Adaptado de [1].

Por outro lado, construir uma representação robusta e invariante a escala das características significa que o algoritmo é capaz de detectar e descrever as características de uma imagem de forma consistente, independentemente da escala em que elas aparecem na imagem. Isso é importante porque as características de uma imagem podem aparecer em diferentes escalas, dependendo de fatores como a distância da câmera, o ângulo de visão, entre outros. Além disso, as imagens também podem ser redimensionadas ou apresentar variações de iluminação, o que pode afetar a aparência das características. Para construir uma representação robusta e invariante a escala das características, o SURF utiliza uma técnica chamada de escala espacial. Essa técnica envolve a aplicação do filtro Haar Wavelet em diferentes escalas da imagem, de forma a detectar características em diferentes tamanhos e resoluções.

Em seguida, o algoritmo utiliza uma técnica de descrição baseada em pontos-chave para descrever as características detectadas em termos de suas propriedades geométricas e de intensidade. Essa descrição é invariante a rotação e escala, o que significa que ela pode ser usada para comparar características em diferentes imagens, mesmo que elas tenham sido capturadas em diferentes escalas ou orientações.

Finalmente, SURF utiliza a resposta do filtro Haar em várias posições da imagem para construir uma representação robusta e invariante a escala das características.

2.3.3.1.5 Construção de descritores de características

Agora é hora de extrair o descritor (Figura 10):

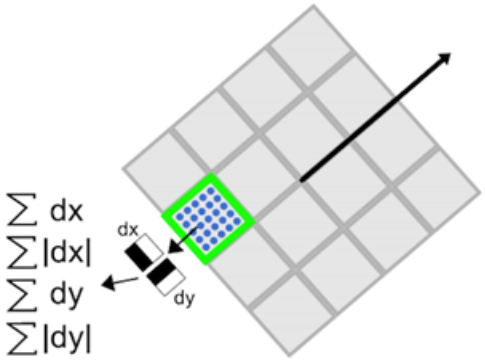

Figura 10 – Componente descritor SURF. Da esquerda para a direita: Um descritor SURF consiste em uma região com divisões de 4×4, onde é filtrada com o Haar Wavelet para a área interna desta região. Em seguida, os 4 valores mencionados são adicionados e obtidos, gerando um vetor de um tamanho de 64. Adaptado de [1].

1. O primeiro passo é construir uma região quadrada centrada em torno do KP e orientada ao longo da orientação que foi calculada anteriormente.

2. A região é então regularmente dividida em sub-regiões quadradas menores de $4x4$. Para cada sub-região, calcula algumas características simples em pontos amostrais regularmente espaçados de $5x5$. Para simplificar, se chama dx de resposta da onda Haar na direção horizontal e dy a resposta da onda Haar na direção vertical (tamanho do filtro 2s). Para aumentar a robustez contra deformações geométricas e erros de localização, as respostas dx e dy são primeiramente ponderadas com um Gaussiana $\sigma = 3.3s$ centrado no KP.

Em seguida, as respostas de Wavelet d_x e d_y são resumidas em cada sub-região e formam um primeiro conjunto de entradas para o vetor característico. Para trazer informações sobre a polaridade das mudanças de intensidade, também extrai a soma dos valores absolutos das respostas, $|d_x|$ e $|d_y|$. Assim, cada sub-região tem um vetor de descritor quadridimensional v para sua estrutura de intensidade subjacente $V = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$. Segundo [155, 156], isso resulta em um descritor vetorial para todas as sub-regiões de 4×4 de comprimento 64.

Em conclusão, a construção de descritores de características serve para extrair informações relevantes das características detectadas nas etapas anteriores do algoritmo, de forma a torná-las mais robustas e discriminantes.

Esses descritores de características são construídos com base em informações locais em torno de cada ponto de interesse detectado pelo algoritmo. Essas informações locais incluem orientação, escala e forma da região em torno do ponto de interesse.

Os descritores de características são usados posteriormente para realizar a correspondência entre pontos de interesse em diferentes imagens, permitindo a detecção e rastreamento de objetos em sequências de imagens. Em resumo, a construção de descritores de características é fundamental para a eficácia e robustez do algoritmo SURF. Finalmente, estes são os passos usados pela biblioteca SURF em OpenCV implementada no algoritmo proposto.

2.3.3.2 Algoritmo Brute-force feature matching

O Brute-force feature matching é amplamente utilizado para realizar a correspondência de recursos em imagens binárias [157],. Este método utiliza uma abordagem de força bruta, comparando todas as características de uma imagem com todas as características de outra para determinar sua similaridade. Seu objetivo principal é encontrar correspondências robustas e precisas entre características, sendo muito útil em aplicações como reconhecimento e rastreamento de objetos, bem como alinhamento de imagens [158, 159].

Um exemplo relevante nesse campo é o artigo de Bostanci [157], que se concentra na correspondência de recursos em imagens binárias. Nesse estudo, é proposta uma abordagem baseada em lógica fuzzy para encontrar correspondências entre recursos extraídos de imagens binárias, buscando obter resultados precisos e robustos nesse tipo de imagens.

Por outro lado, o artigo [160] utiliza a correspondência de características de força bruta para correspondência de características em imagens históricas. A pesquisa se concentra no uso da geometria de quadriláteros como meio de aprimorar a correspondência entre características nesse tipo de imagens.

Quanto ao artigo [159], o foco está na detecção e reconhecimento de objetos em imagens. A proposta do estudo é utilizar a correspondência de recursos de força bruta para encontrar correspondências entre recursos em uma imagem de consulta e um banco de dados de recursos, com o objetivo principal de detectar e reconhecer objetos nas imagens.

Apesar das diferenças nas aplicações específicas, todos esses artigos destacam a utilidade e eficácia da correspondência de recursos de força bruta. Essa abordagem é considerada robusta e precisa, sendo adequada para uma variedade de aplicações, como reconhecimento de objetos, detecção de objetos e alinhamento de imagens. Além disso, são discutidas abordagens adicionais, como o uso de lógica fuzzy e consideração da geometria de quadriláteros, para aprimorar os resultados da correspondência de recursos em contextos específicos.

A correlação entre o algoritmo SURF e o método de Brute-force feature matching é estreita e eles se complementam no campo da correspondência de recursos. Ao combinar essas duas técnicas, é possível obter correspondências precisas e confiáveis entre recursos de diferentes imagens, o que é crucial em aplicações como reconhecimento de objetos, rastreamento de objetos e alinhamento de imagens. O SURF fornece os descritores necessários e o Brute-force feature matching realiza a busca exaustiva de correspondência, permitindo obter resultados precisos e robustos na correspondência de recursos.

Em resumo, a combinação do algoritmo SURF com o método de Brute-force feature matching oferece uma solução eficaz e confiável para a correspondência de recursos em imagens [159, 161]. Essa abordagem tem sido amplamente utilizada em várias aplicações, como reconhecimento de objetos [162], rastreamento de objetos [163], alinhamento de imagens e muito mais [164, 165]. Ao explorar as características únicas de cada método e aproveitar sua complementaridade, é possível obter resultados precisos e robustos na correspondência de recursos.

2.3.3.3 Homografia

Uma homografia é uma transformação projetiva que determina uma correspondência entre duas figuras geométricas planas [166, 167]. A homografia entre duas vistas pode ser estimada com correspondências entre os pontos da imagem, como se mostra na seguente Equação 2.8, de modo a obter uma matriz H que mapeia pontos de uma vista.

$$P' = (H)(P) = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \tag{2.8}$$

Onde P' é um ponto da primeira visualização e P é um ponto da segunda. Existe um fator de escala, $h_{33} = 1$. A matriz tem apenas 8 graus de liberdade, portanto, pode ser estimada se conhece pelo menos quatro correspondências entre os pontos das diferentes vistas.

A aplicação da homografia permite verificar se os agrupamentos realizados na etapa anterior correspondem ao objetivo almejado, ao determinar a correspondência entre dois planos.

2.4 Sistemas de coordenadas

Conhecer a nomenclatura das coordenadas exatas de um robô autônomo, especificamente o GCS (*Sistema de coordenadas geográficas*), UTM (*Universal Transverse Mercator*) e WGS84 (*Coordenadas WGS84 (WGS = World Geodetic System)*), é essencial por vários motivos.

Primeiro, esses sistemas de coordenadas fornecem uma forma padronizada de representar a localização do robô, o que permite fácil comunicação e integração com outros sistemas. Isso é particularmente importante em ambientes com vários robôs ou ao trabalhar com dados geoespaciais.

Em segundo lugar, entender esses sistemas de coordenadas é fundamental para uma navegação precisa do robô. O GCS, UTM e WGS84 fornecem um método universal de expressar localizações geográficas, e entender as relações entre eles é essencial para navegar de forma autônoma.

Um GCS inclui uma unidade de medida angular, um meridiano principal e um dado (com base em um esferóide). Nesse caso, um ponto é referenciado por seus valores de longitude e latitude, são ângulos medidos do centro da terra até um ponto na superfície da terra. Os ângulos são geralmente medidos em graus, na Figura 11 A) pode ver a representação do globo terrestre, que mostra as coordenadas definidas por WGS84 (Latitude e longitude).

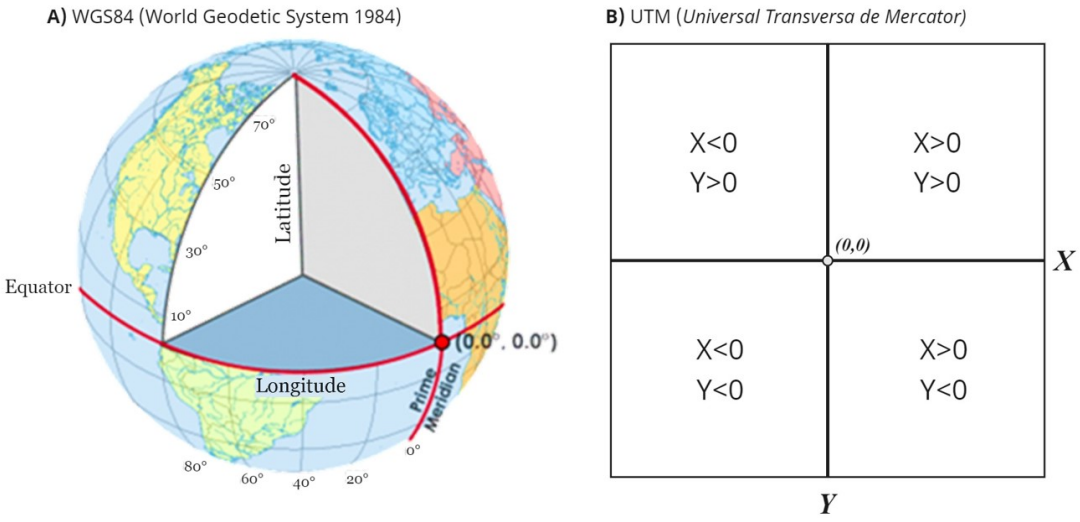

Figura 11 – Representação das coordenadas WGS84 e UTM.

No entanto, a leitura dos dados deve ser em coordenadas projetadas, ou seja, coordenadas UTM. Posto que a diferença neste tipo de coordenadas é que os sistemas de coordenadas geográficas (WGS84) são baseados em um esferóide e usam unidades (graus) e sistemas de coordenadas projetadas são baseados em um plano (o esferóide projetado em uma superfície 2D) e usam unidades lineares (pés, metros, etc.).

Outra razão pela qual as coordenadas UTM serão usadas é porque elas garantem uma precisão de cerca de 50 cm [168, 169, 2]. Portanto, deve-se implementar a derivação matemática para o caso esférico, segundo ou plano que mostra a Figura 12, para a projeção transversal de mercator em sua forma "mais pura", a continuação são explicadas cada umas das equações, implementadas para realizar a conversão, implementado no algoritmo.

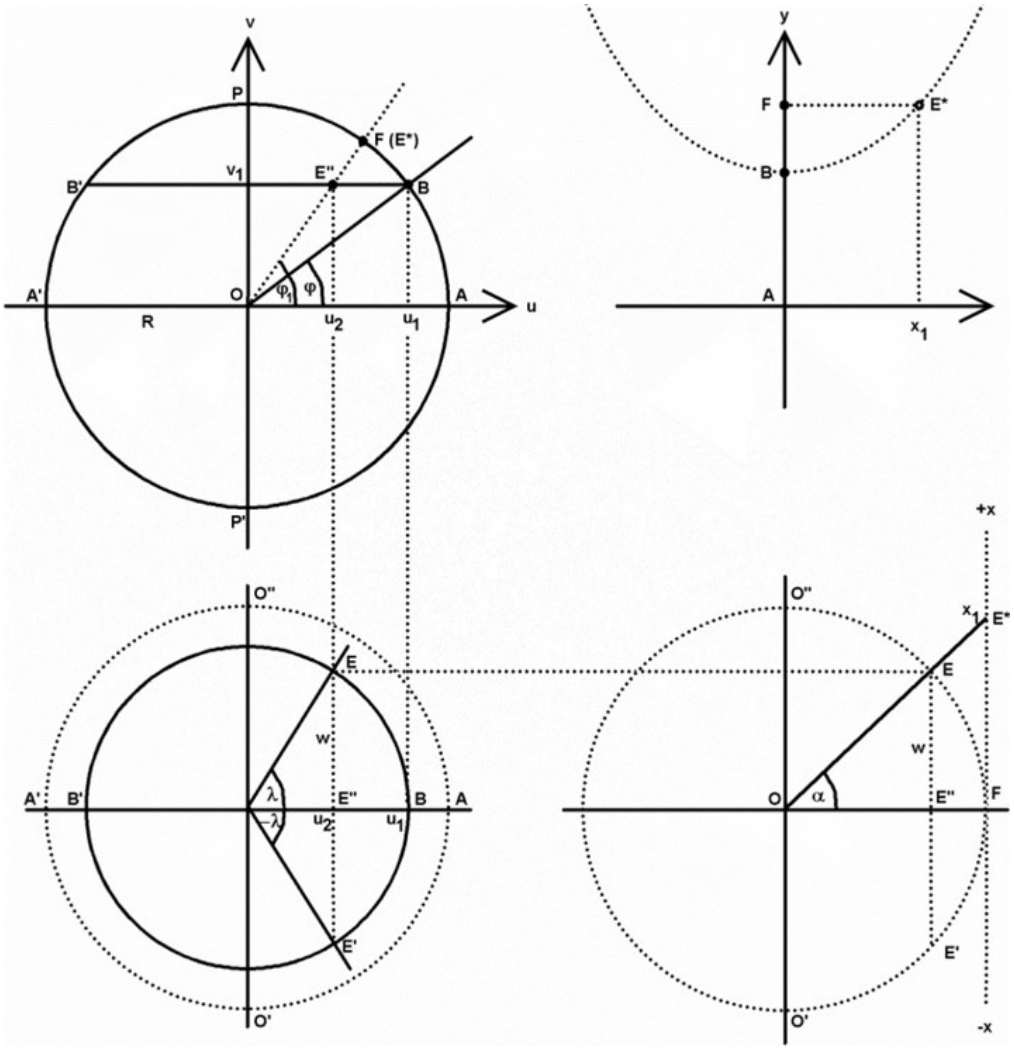

Figura 12 – Representação esferóide, de acordo com suas projeções. Adaptado de [2, 3].

Equações:

$$
\begin{aligned}
u_1 &= R\cos_{\varphi} \\
v_1 &= R\sin_{\varphi} \\
u_2 &= u_1\cos_{\lambda} \\
w &= u_1\sin\lambda \\
\alpha &= \arcsin(\frac{W}{R}) = \arcsin(\cos\varphi\sinh\lambda) \\
x_1 &= R\tan(\arcsin(\cos\varphi\sin\lambda)) \\
\tan\varphi_1 &= \frac{v_1}{u_2} = \frac{R\sin\varphi}{R\cos\varphi\cos\lambda} = \frac{\tan\varphi}{\cos\lambda} \\
\varphi_1 &= \arctan(\frac{\tan\varphi}{\cos\lambda})
\end{aligned}
\tag{2.9}
$$

O UTM norte da localização EeF é o arco de ângulo ϕ_1 do equador para o local F multiplicado pelo raio da terra R e o fator de escala $ko = 0.9996$, Equação 2.10.

$$y = R \arctan(\frac{\tan \varphi}{\cos \lambda}) \tag{2.10}$$

A Equação 2.9 para o valor x1 é convenientemente calculada usando a relação, Equação 2.11:

$$\arcsin z = \arctan \frac{z}{(z - z^2)^{\frac{1}{2}}} \tag{2.11}$$

Simplificado, sem um fator de escala Equação 2.12 :

$$x = \frac{R \cos \varphi \sin \lambda}{(1 - \cos^2 \varphi \sin^2 \lambda)^{\frac{1}{2}}} \tag{2.12}$$

Esse tipo de projeção pode ser visto evidenciado na precisão direcional (importante no transporte marítimo), área igual e angular (ortogonalidade). Assim, as curvas λ e ϕ que são constantes em qualquer ponto são perpendiculares umas às outras (são ortogonais), o produto de suas encostas deve ser igual a -1 em todos os lugares (Equação 2.13):

$$(\frac{\frac{\delta_y}{\delta_\varphi}}{\frac{\delta_x}{\delta_\varphi}})(\frac{\frac{\delta_y}{\delta_\lambda}}{\frac{\delta_x}{\delta_\lambda}}) = -1 \tag{2.13}$$

Com base nessas fórmulas, duas funções principais são criadas no código, que convertem um par latitude/longitude (WGS84) para a zona UTM/leste/norte. Que neste caso se pode especificar uma zona; se nenhum valor for fornecido, ou seja, $zona == 0$, então o correto será calculado. Em seguida, ele retorna a área usada e coloca leste e norte (em metros) em x e y, ou seja, em latitude/longitude.

Finalmente se tem a leitura e escrita das coordenadas UTM em latitude e longitude, esses valores são essenciais para desenhar o mapa topológico. Na seção 4 explica melhor o procedimento.

3 Desenvolvimento proposto

Nesse contexto, surge a proposta de uma metodologia inovadora que combina localização topológica e visão computacional para criar uma abordagem mais robusta para navegação autônoma. Esta metodologia consiste em sete etapas que permitem uma detecção e reconhecimento do ambiente, através da análise de imagens e comparação com um mapa topológico pré-existente.

A metodologia proposta tem o potencial de melhorar a confiabilidade dos veículos autônomos, o que pode ser um passo importante para um futuro com mobilidade mais segura e eficiente. Neste capítulo, serão apresentados os sete passos da metodologia, suas implicações serão discutidas e seu desempenho será avaliado por meio de diversos experimentos e testes.

A Figura 13 representa o fluxo do trabalho desenvolvido (Metodologia de Pesquisa). Onde, **A)** o módulo de processamento de dados tem a responsabilidade de obter um arquivo .OSM e transformá-lo em uma estrutura de dados para posterior análise. **B)** O módulo construção de mapa topológico utiliza os dados estruturados do módulo anterior para construir a representação visual do mapa. **C)** O módulo de geração de dataset, realiza a obtenção do banco de imagens de cada ponto composto por 30 imagens. **D)** O módulo de aquisição de imagens, neste é gerado um vídeo que toma cada 20 frames, obtendo 100 imagens que serão usadas para fazer a detecção de cada ponto no mapa, **E)** módulo de imagem matching, é responsável pela implementação do SURF, após a implementação do vizinhos mais próximos por meio de Brute-force feature matching, para obter os descritores de imagem para implementar uma homografia, durante o processo de correspondência. **F)** O módulo de localização topológica é responsável por integrar o dataset de imagens com o mapa implementado pelos dados do OSM e finalmente **G)** é responsável por propor uma análise em três experimentos diferentes para responder à questão de pesquisa.

A seguir será apresentado em detalhes cada módulo apresentado na Figura 13.

3.1 Módulo processamento de dados OSM

A localização é um dos módulos-chave de um sistema de condução autônoma. Saber a localização do veículo é de vital importância para a percepção e controle. É por isso que se propõe uma metodologia que permite adquirir dados geográficos e transformá-los em um mapa, com a possibilidade de evidenciar para onde o veículo pode ir, e onde o veículo está.

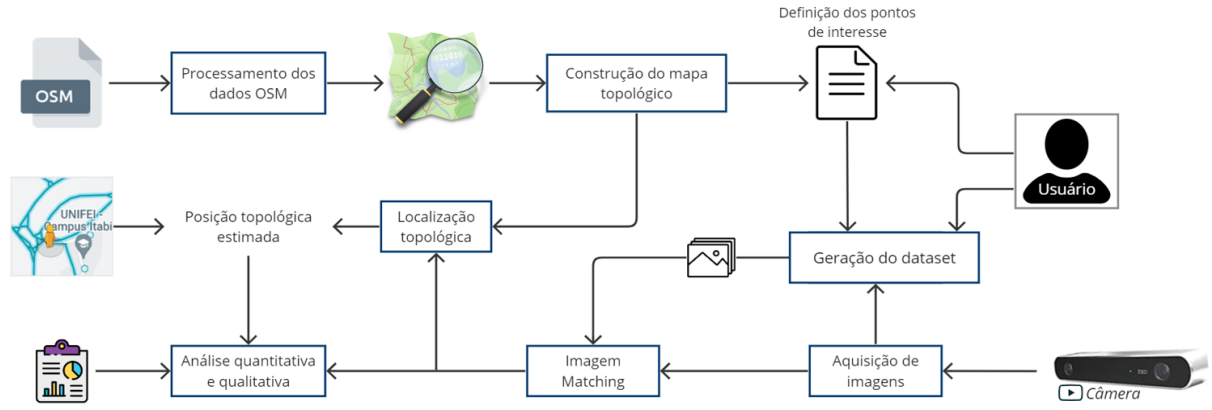

Figura 13 – Proposta metodológica para a integração de localização topológica e visualização computacional.

Para começar, os mapas mais acessíveis e disponíveis publicamente hoje são mapas 2D de visualização superior, como OpenStreetMaps(OSM) e mapas de satélite. OSM é um desses tipos de mapa, fornecendo formas 2D da maioria das principais estruturas e marcadas com dados de geolocalização. A metodologia proposta é utilizar um arquivo .OSM, que fornece informações geográficas e topológicas sobre vias, endereços, informações sobre uso do solo ou edificações.

Este arquivo está disponível em [1], deve selecionar a seção do mapa que deseja analisar, neste caso, a seção de dados vetoriais de norte -19,66923, sul -19,67522, oeste -43,21630 e leste -43,20827 é analisada. Um arquivo .OSM é gerado, o qual é um formato ordenado, estruturado e codificado em XML. Ele contém quatro elementos para armazenar a estrutura de dados topológicos:

- Nós: Representa a posição geográfica armazenada como pares de latitude e longitude. É usado para representar recursos de mapa sem tamanho, como picos de montanhas.
- Caminhos: Listas ordenadas de nós, significando uma polilinha ou um polígono. Representa recursos lineares, como estradas e rios, e zonas, como estacionamentos, natureza e parques.
- Relacionamentos: As listas ordenadas de nós e estradas representam suas relações como barreiras e curvas nas estradas, as rodovias abrangem diferentes estradas existentes e áreas com desníveis.
- Rótulos/tags: Armazena metadados sobre objetos de mapa. Sempre ligado a qualquer nó, caminho ou relação. Considerando que eles são usados para caracterizar as características físicas do solo (edifícios e estradas etc.) no OSM. Cada rótulo relaciona uma feição geográfica à feição representada por aquele nó ou relacionamento específico.

[1] https://www.openstreetmap.org/exportmap=17/-19.67213/-43.21141layers=ND

Esses dados vêm em coordenadas WGS84 [170], o que significa que é um sistema de coordenadas geográficas (GCS) usa uma superfície esférica tridimensional para definir localizações na Terra, isso foi explicado em na secção 2.5. Para este trabalho, este tipo de coordenadas será utilizado para processar os dados obtidos pelo OSM.

3.2 Módulo construção do mapa topológico

Após se ter a entrada de dados do OSM e realizar a conversão de dados UTM, é realizado a interface e o desenho dos dados obtidos pelo .OSM, ou seja, as informações geográficas sobre as relações entre os objetos geométricos.

Foram utilizadas bibliotecas em Qt que permitem a confecção de elementos gráficos, permitindo assim o mapeamento e representação de dados geográficos, tais como: elementos, tags (Etiquetas) e elementos semânticos.

Para construir um mapa topológico a partir de dados do OSM, é possível utilizar diversas ferramentas e bibliotecas disponíveis. Neste sentido, uma opção é criar um mapa topológico por meio de uma interface utilizando a linguagem de programação C++ e os dados do OSM. A seguir, apresentamos como essa abordagem pode ser implementada.

- Passo 1, download de dados de OSM: Construir um mapa topológico usando dados OSM é baixar os dados. Os dados de OSM são imprescindíveis descarregar os dados em formato XML. Isso foi explicado na seção anterior 3.1.
- Passo 2, análise dos dados do OSM: Após baixar os dados do OSM, é necessário analisá-los para extrair informações relevantes sobre eles. Esta informação inclui as coordenadas dos nós, as conexões entre os nós (vias) e os metadados associados aos nós e vias. Para isso, use umas bibliotecas para divulgar na interface gráfica cada característica do mapa topológico, a parte visual.
- Passo 3, transforme os dados, na criação do mapa topológico: Depois de analisar os dados do OSM, o próximo passo é criar um gráfico (A interface). Neste caso, escolha 5 pontos que são as interseções da área de interesse. Esses pontos são representados por suas coordenadas e as arestas são representadas por suas conexões.
- Passo 4, divulgue os possíveis caminhos do mapa: Uma vez que se interpreta o gráfico, é essencial encontrar os caminhos mais curtos entre os pontos, para ter em conta que direção pode ir ao veículo, para esta região de interesse, o carro só vai em uma direção.
- Passo 5, construção da interface gráfica: Após ter armazenado todas as informações dos dados de OSM, o passo final é construir uma interface para mostrar o mapa topológico. Isso pode ser feito usando uma biblioteca GUI como Qt ou wxWidgets.

- Passo 6, integração do banco de imagens georreferenciadas: Este passo será explicado com mais detalhes na próxima seção 3.3. Baseia-se na integração de um banco de imagens que representa cada ambiente de cada ponto.

Após estes passos também são implementados outros 3 passos (seção 3.4, 3.5 e 3.6) que consistem na integração de localização e visão por computador. A importância dos dados de OSM são essenciais porque é um mapa colaborativo, o que significa que os usuários podem contribuir com os dados do mapa agregando, editando e atualizando informações. Os dados são abertos e são impulsionados pela comunidade, o que significa que uma grande comunidade de colaboradores garante que os dados sejam atualizados e sejam precisos.

Por outro lado, pretende-se que este programa funcione para qualquer computador, para alcançar esse resultado, as bibliotecas do QT (Tabela 2) também ajudaram na integração do usuário com a interface gráfica. Neste caso, a interação com o mouse é fundamental, pois o usuário poderá escolher qualquer arquivo .OSM e poderá controlar como deseja vê-lo, neste caso as configurações de zoom e clique também estão programadas.

Biblioteca	Implementação
#include "QGraphicsItem"	Elementos gráficos
#include "QWheelEventQXmlStreamReader"	Controle do cursor e parâmetros do mouse (evento da roda)
#include "QXmlStreamWriter"	Funciona para carregar arquivos XML, tanto para leitura quanto para escrever
#include "iostream"	Faça entradas e saídas
#include "math.h"	Operações matemáticas básicas

Tabela 2 – Bibliotecas implementadas para a representação de dados geográficos por meio de um mapa.

Uma vez que foram criadas funções que estabelecem características que um mouse deve ter, movimento, zoom ou executar um clique, em qualquer lugar na tela. Foi considerado que a maioria das marcas existentes de fabricação de mouses operam em etapas de 15^o, nesse caso o valor delta é um múltiplo de 120; ou seja, 120 unidades, $\frac{1}{8} = 15$ isso ajuda a fazer o programa funcionar para qualquer tipo de mouse, neste caso funciona para fazer a seleção do arquivo baixado e permite controlar o zoom e a seleção dos pontos no mapa.

Depois disso, é feito o desenho dos polígonos e linhas do mapa, conforme as informações fornecidas pelo arquivo .OSM, ele mostra a relação de cada caminho e ponto. Também é programada a leitura de etiquetas compostas por dois elementos, uma 'chave (k)' e um 'valor (v)'. Ambos os elementos são campos de texto, mas geralmente represen-

tam elementos numéricos ou outros elementos estruturados. Neste caso, se tem tags como: amenity (natural), grama, madeira, mato, uso do solo (industrial), brownfield, residencial, floresta, estrada de ferro (railway) e highway (trilha, bicicleta, pista, degraus, área), via aérea (aeroway), avental, pista, hidrovia, entre outros. Todos eles têm o propósito de representar os elementos dentro do mapa. Na Figura 14 são mostrados os modelos de dados do OSM. A Figura 14. A) apresenta a parte visual da plataforma do arquivo .OSM o qual é baixado. Essa área de trabalho corresponde um lugar próximo à UNIFEI, campus Itabira e B) é o resultado da construção do mapa topológico implementado no ambiente gráfico do Qt Creator. Foram escolhidos 5 pontos de interesse a serem reconhecidos no mapa, onde estes pontos representam interseções de vias e possuem características, como placas ou faixas, que apresentam informações que se destacam em relação ao resto das imagens.

Figura 14 – Modelos de dados do OSM e resultado da construção topológica do mapa.

Após obter a construção do mapa topológico, a interpretação dos 5 pontos é realizada. Se caracteriza-se cada ponto, onde o ponto pode ser selecionado evidenciando um ponto de partida com cor vermelha e o usuário pode selecionar o ponto de chegada, para realizar o planejamento de um ponto **A** a **B**.

O processo começa com a validação dos dados do ponto de partida, representados por **5** pontos escolhidos como interseções. O código solicita que o usuário insira o ponto

inicial e o ponto de destino. Após ter essas informações, o sistema começa a explorar todos os caminhos possíveis entre os pontos e se posiciona no ponto inicial, tendo em conta que o caminho escolhido só tem a possibilidade de interagir em uma só direção.

A partir do ponto inicial, o sistema busca um caminho para chegar ao ponto de destino. Ele examina cada uma das rotas possíveis a partir do ponto inicial até os pontos seguintes com os quais ele tem conexão. O percurso é sempre feito para frente, após encontrar os pontos próximos ao ponto inicial, o sistema compara a distância de cada um desses pontos em relação ao ponto de destino e seleciona o possível caminho como candidato.

Esse ponto é então atribuído como o novo ponto inicial e o processo se repete até que o ponto inicial seja escolhido igual que o ponto de destino. Isso significa que um possível caminho foi encontrado e pode ser armazenado como informações em um vetor de posições. Essas informações serão usadas para desenhar o caminho.

Em resumo, o sistema explora todos os caminhos possíveis entre os pontos escolhidos e seleciona o caminho para chegar ao ponto de destino. Depois, armazena as informações sobre o caminho encontrado para posterior uso. Além disso, permite saber as coordenadas de cada ponto, com o fim de saber a localização do veículo.

3.3 Módulo geração do banco de imagens

O ser humano está sempre captando e observando o que se vê ao seu redor, porém, como é que **um lugar pode ser reconhecido após várias passagens?** e **como objetos que nunca vimos antes são associados a objetos já conhecidos?** neste caso, **o computador poderia fazer o mesmo e tornar um veículo autônomo por meio da câmera na hora de saber onde está localizado, com seus respectivos informações geográficas?**

Hoje em dia as câmeras estão no nosso dia a dia, por exemplo, pode ser usada em sistemas de vigilância ou como ferramenta de trabalho para fazer videoconferências ou até mesmo trabalhar para poder se conectar com pessoas que estão em qualquer lugar do mundo. Neste projeto é utilizada uma ZED (*Câmera ZED*) que consegue capturar vídeo em tempo real, processar informação e oferecer ao usuário informações e detalhes de uma cena. Neste caso ela sera a ferramente de visão do veículo.

Consequentemente, a implementação desta proposta começa com a captura de uma imagem, que pode ser definida como uma função bidimensional $f(xy)$ onde x e y são as coordenadas espaciais, e o valor de f em qualquer par de coordenadas (xy) é a intensidade da imagem naquele momento.

Uma imagem pode ser contínua em relação a x e y, e em intensidade (imagem

analógica). A conversão dessa imagem em formato digital requer que as coordenadas e a intensidade sejam digitalizadas. A digitalização das coordenadas é chamada de amostragem, enquanto a intensidade de digitalização é chamada de quantização [121]. Então, quando todas as quantidades são discretas, se chama a imagem de imagem digital.

Antes de fazer a geração do dataset, a implementação da calibração da câmera foi considerada, pois é passo necessário para obter uma confiabilidade e precisão dos parâmetros geométricos do processo de formação de imagens. Ou seja, se uma alta precisão for obtida, significa que a câmera deve fornecer a mesma saída para a mesma entrada em todos os tipos de condições ambientais. De outra forma, uma alta resolução significa que a câmera deve conseguir decifrar a menor mudança possível. Portanto, a câmera precisa ser calibrada para maior precisão e baixa distorção, ajudando a alcançar uma representação mais precisa do mundo real nas imagens capturadas.

A precisão da calibração determinará posteriormente a exatidão das medições feitas a partir das imagens. É por essa razão que é essencial realizar a calibração da câmera com garantias completas de que os parâmetros obtidos são os mais semelhantes aos reais. No trabalho isso foi feito pela captura de imagens com a câmera calibrada pelo aplicativo da ZED.

A ZED permite a variação de diferentes parâmetros como: distância focal ($f_x e f_y$), pontos principais ($c_x e c_y$), distorção da lente ($k_1, k_2, k_3, p_1 e p_2$), visão horizontal, vertical e diagonal e calibração estéreo (rotação e tradução entre o olho esquerdo e direito). Neste caso, funções como:

- Calibração.
- Distância focal da lente esquerda em píxeis.
- Coeficiente de distorção radial.
- Tradução entre a lente esquerda e à direita no eixo z.
- Campo horizontal de visão da lente esquerda em graus.

Permitem a modificação dos parâmetros conforme a cena. No entanto, por padrão, ele integra uma calibração automática constituída pelo método de Zhang, o qual é um método de calibração da câmera de xadrez de plano único, e este método fica entre o método de calibração fotográfica e o método de autocalibração, que supera as deficiências dos objetos de calibração de alta precisão exigidos pelo método tradicional de calibração e resolve o problema da baixa robustez do método de autocalibração [171, 172]. O método tradicional de calibração requer três planos verticais de dois por dois. Precisa-se usar um tabuleiro de xadrez impresso e tirar vários conjuntos de fotos de diferentes direções. Em

comparação com a autocalibração, a precisão é melhorada e a operação é conveniente [173, 174, 79].

Neste caso, por padrão, um processo de calibração é integrado à ZED (Figura 15), este modo é muito flexível do ponto de vista que a câmera pode ser movida livremente para reconhecer padrões no monitor. Ao realizar essa calibração deve-se considerar que as condições do ambiente devem ser controladas, como que não tenha reflexo na tela e não tenha luz brilhante, pois poderia dar dados errôneos no momento da calibração.

O arquivo de calibração fornecido pelo software da ZED contém informações sobre a posição precisa da câmera direita e esquerda e suas características ópticas. A Figura 15 mostra o tabuleiro de xadrez, neste caso é devido apontar a ZED para o monitor aparece um ponto azul onde se deve apontar. O objetivo é fazer com que o ponto azul corresponda ao ponto vermelho, em diferentes posições. Depois disso, o programa lança um arquivo .txt que contém todos os valores correspondentes à calibração. Este arquivo é lido no início do código para tirar fotos de cada ponto, com a câmera calibrada. Nessa hora, é feita a geração do dataset. Neste caso, como são 5 pontos plotados no mapa, são geradas 30 imagens para cada ponto, obtendo-se assim um dataset de 150 imagens. Estas imagens possuem o mesmo tamanho (640x480), captadas em diferentes horários do dia. Estas imagens foram salvas a mão, as imagens foram associadas aos pontos de acordo as características do ambiente.

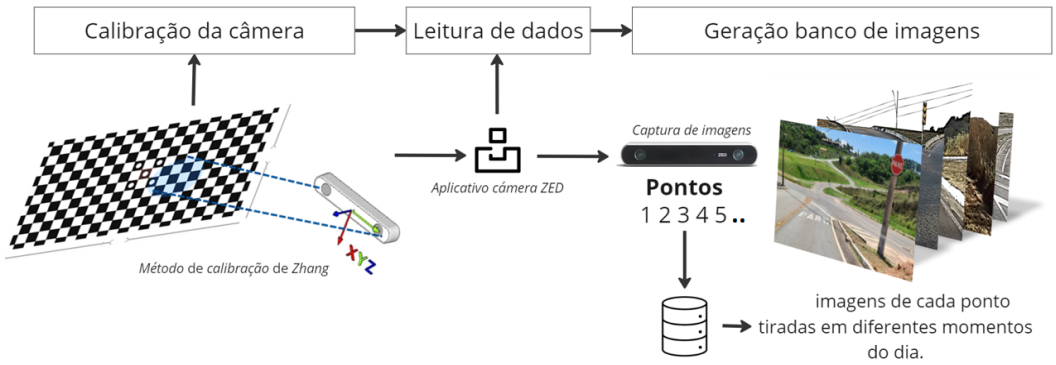

Figura 15 – Processo de geração do banco de imagens tendo em conta a aplicação disponibilizada pela câmara ZED, que permite a captação de imagens, com a câmara calibrada.

3.4 Módulo aquisição de imagens

Após obter o dataset das imagens de cada ponto, é feito um vídeo em diferentes horários do dia. Os vídeos caracteriza-se como uma sequencia de imagens. Quando o vídeo é gravado, sabe-se quantos quadros ou imagens tem por segundo e isso se chama FPS (quadros por segundo). Neste caso, o video é a entrada a ser analisada. Para não analisar muitos frames do vídeo, são consideradas apenas 100 imagens, onde essas imagens serão rotuladas manualmente, onde cada imagem é classificada com o nome de cada ponto a que pertence, ou seja, é feita uma tabela no Excel onde são numeradas as 100 imagens e colocadas se a imagem pertence a um dos pontos de interesse, caso contrário, 0 é colocado.

Se deve considerar a duração do vídeo, para depois passá-lo por um programa que permita a obtenção de quadros de vídeo (frames). UAo se obter um vídeo às 7h da manhã com duração de 1h25min, isso corresponde a um tempo total de 85 minutos, o que equivale a 85 segundos. Sabendo que o vídeo foi adquirido com uma frequência de 30 quadros por segundo (FPS), temos um total de 2550 quadros. No entanto, para a análise, foi extraída uma imagem a cada 2 quadros, resultando em 100 imagens selecionadas para compor a análise.

Para obter os frames, foi utilizado o software VLC media player, o qual é um reprodutor multimídia e framework. Ele começa criando uma pasta para salvar os frames do vídeo, depois copia ou anda por ele. A Figura 16 mostra o procedimento.

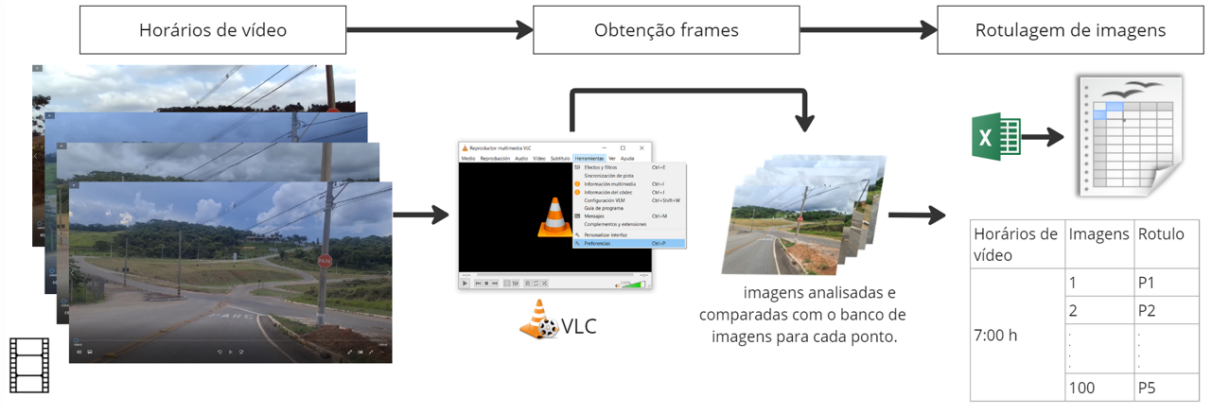

Figura 16 – Procedimento de análise de vídeo de entrada, para fazer a comparação com o dataset implementado para cada ponto.

Porém, em trabalhos futuros seria interessante realizar rotulagem automático, onde as imagens do video podem ser registradas de acordo com cada ponto. Esse algoritmo seria capaz de analisar cada quadro e rotular a imagem de acordo com o ponto de interesse, evitando erros na rotulagem e decidir se a imagem pertence ao ponto desejado.

3.5 Módulo imagem matching

No mundo da visão computacional, o algoritmo SURF provou ser uma ferramenta valiosa para identificar objetos e padrões em imagens [175, 176]. Neste capítulo será apresentado o módulo Imagem Matching, onde será aprofundado o funcionamento deste algoritmo e explicado detalhadamente o funcionamento do SURF e como será utilizado na proposta metodológica. O SURF é capaz de analisar as características únicas de uma imagem e, a partir delas, fazer a descrição de seus KP. Essas informações serão de grande ajuda para a identificação e análise de objetos em imagens, o que permitirá a obtenção de resultados precisos e eficientes para contexto de este projeto.

Para descrever os pontos-chave (KP) em uma imagem, é necessário extrair as características de cada ponto ou píxel. Existem diversos algoritmos que realizam essa descrição de formas diferentes. Neste contexto, o algoritmo SURF se destaca por conseguir detectar, descrever e reconhecer os KP, atribuindo a cada vetor descritor. É importante destacar que o algoritmo SURF processa as imagens em níveis de cinza [175, 176]. A seguir, serão abordadas as etapas do algoritmo SURF para uma melhor compreensão de seu funcionamento.

3.5.1 SURF

Este algoritmo busca imagens discretas e aplica imagens integrais, a extração de pontos característicos baseia-se na teoria do espaço de escala, permitindo encontrar as coordenadas dos pontos característicos, detectando os pontos extremos locais da imagem, ou seja, o ponto local mais brilhante ou mais escuro, diante disso, a seguir estão as principais características que o SURF implementa neste trabalho para realizar a extração de características Figura 17, cada bloco foi explicado na seção 2.3.3.1. e na Figura 3.

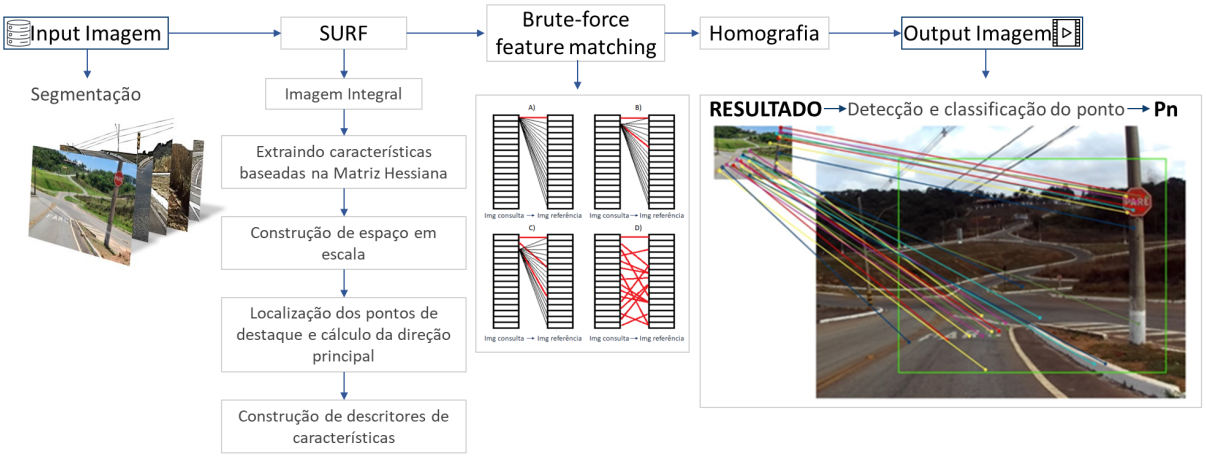

Figura 17 – Processo para detectar e classificar o ponto.

Nesta Figura 17, inicia-se com a aquisição de uma imagem, que seria o vídeo, depois é aplicado o algoritmo SURF, logo Brute-force feature matching que trabalha

para encontrar correspondências entre os descritores. Na correspondência (matching) o objetivo é encontrar os K vizinhos mais próximos de um determinado ponto ou vetor em um conjunto de dados. O algoritmo encontra os vizinhos mais próximos calculando a distância entre os pontos no espaço de características. Logo é aplicado o processo de homografia, que è a que encontra a transformação afim que relaciona pares de pontos. O desenvolvimento de cada um será explicado na seção 4.

3.5.2 Correspondência de recursos por Brute-force feature matching

Após a extração dos descritores de características utilizando o algoritmo SURF, o próximo passo consiste em realizar a correspondência de recursos para encontrar correspondências significativas entre os pontos de interesse no mapa e os pontos de interesse na base de dados. Nesta etapa, utilizamos o algoritmo de correspondência de recursos por Brute-force feature matching.

O algoritmo de correspondência de recursos por Brute-force feature matching é uma abordagem simples, mas eficaz, para identificar correspondências significativas entre as características extraídas de diferentes imagens [1, 121]. Ele compara todos os pontos de interesse do vídeo com todos os pontos de interesse na base de dados. A distância entre os descritores de características é calculada por meio de métricas de similaridade, como a distância.

Para realizar a correspondência de recursos utilizando o algoritmo de correspondência de recursos por Brute-force feature matching, foi usado a função cv.BFMatcher() da biblioteca OpenCV. Essa função recebe os descritores de características das imagens e da base de dados como entrada e retorna as correspondências encontradas.

A escolha do valor de 'k' (KNN) é importante nessa etapa. O parâmetro 'k' na função cv.BFMatcher.knnMatch() determina o número de melhores correspondências a serem encontradas para cada ponto de interesse no mapa. Um valor adequado de 'k' deve ser selecionado com base no contexto do problema e nas características dos dados. Um valor muito baixo pode resultar em correspondências incompletas, enquanto um valor muito alto pode aumentar o tempo de processamento [177, 154, 178].Nesse caso, se fez uma escolha de um valor de 'k=2' para limitar a pesquisa às duas melhores correspondências para cada ponto de interesse no mapa. No entanto, é importante observar que o valor ótimo de 'k' pode variar dependendo do cenário.

Este processo pode ser visualizado na Figura 17, onde é demonstrado o algoritmo de correspondência de recursos por Brute-force feature matching, Hadisukmana [158] explica melhor o conceito e a imagem é adaptada de seu artigo. Na primeira etapa Figura 17 A), é apresentado o processo de encontrar a menor distância entre uma característica na imagem de consulta e todas as características na imagem de referência. Cada linha na

imagem representa o cálculo da distância. Enquanto isso, a linha vermelha grossa indica a distância mínima encontrada. As etapas Figura 17 A) até C) mostram, respectivamente, a primeira, segunda e terceira característica na imagem de consulta. Por fim, na última etapa Figura 17 D), cada característica na imagem de consulta já possui sua correspondência ou encontrou sua distância mínima.

Além disso, é relevante destacar que o algoritmo de correspondência de recursos por Brute-force feature matching pode ser utilizado em conjunto com outras técnicas de filtragem ou refinamento, como a utilização da proporção de distância entre as duas melhores correspondências para rejeitar correspondências ambíguas [159]. Ao implementar o módulo de correspondência de recursos com o algoritmo de correspondência de recursos por Brute-force feature matching após a extração de características usando o algoritmo SURF, é capaz de obter correspondências confiáveis entre os pontos de interesse no mapa e na base de dados. Essas correspondências são essenciais para a próxima etapa da metodologia, que é a estimativa da transformação espacial (homografia) para realizar a detecção e classificação dos pontos de interesse.

3.5.3 Homografia

Após obter a implementação do SURF e do Brute-force feature matching, é realizada a homografia, que primeiro corresponde à estimação robusta dos parâmetros do modelo na presença de outliers, ou seja, pontos de dados ruidosos e incorretos. Este se encarrega de encontrar a transformação entre os KP coincidentes.

Se passar o conjunto de pontos das duas imagens que estão sendo comparadas, ele encontrará a transformação de perspectiva daquele objeto. Pode então usar uma função de OpenCv [179] (findHomography) para encontrar a relação. Pelo menos um ponto correto é necessário para encontrar a transformação de perspectiva.

Essa matriz de transformação de perspectiva pode ser calculada a partir de um conjunto de quatro pontos de referência na imagem original e os pontos correspondentes na imagem transformada. Esta técnica é utilizada como correção de perspectiva na imagem de entrada do vídeo.

Portanto, as boas correspondências que fornecem uma estimativa correta são chamadas de valores internos e as restantes são chamadas de outliers. Se aplica a mesma biblioteca [179] a qual retorna uma máscara que especifica os pontos internos e externos, ela permite estimar a homografia entre dois conjuntos de pontos correspondentes em duas imagens.

Dando como saída a detecção da cena conforme a imagem apresentada no dataset. Neste ponto ele localiza se detecta as características principais de cada ponto e traz as linhas boas de coincidências na imagem de saída (Figura 18).

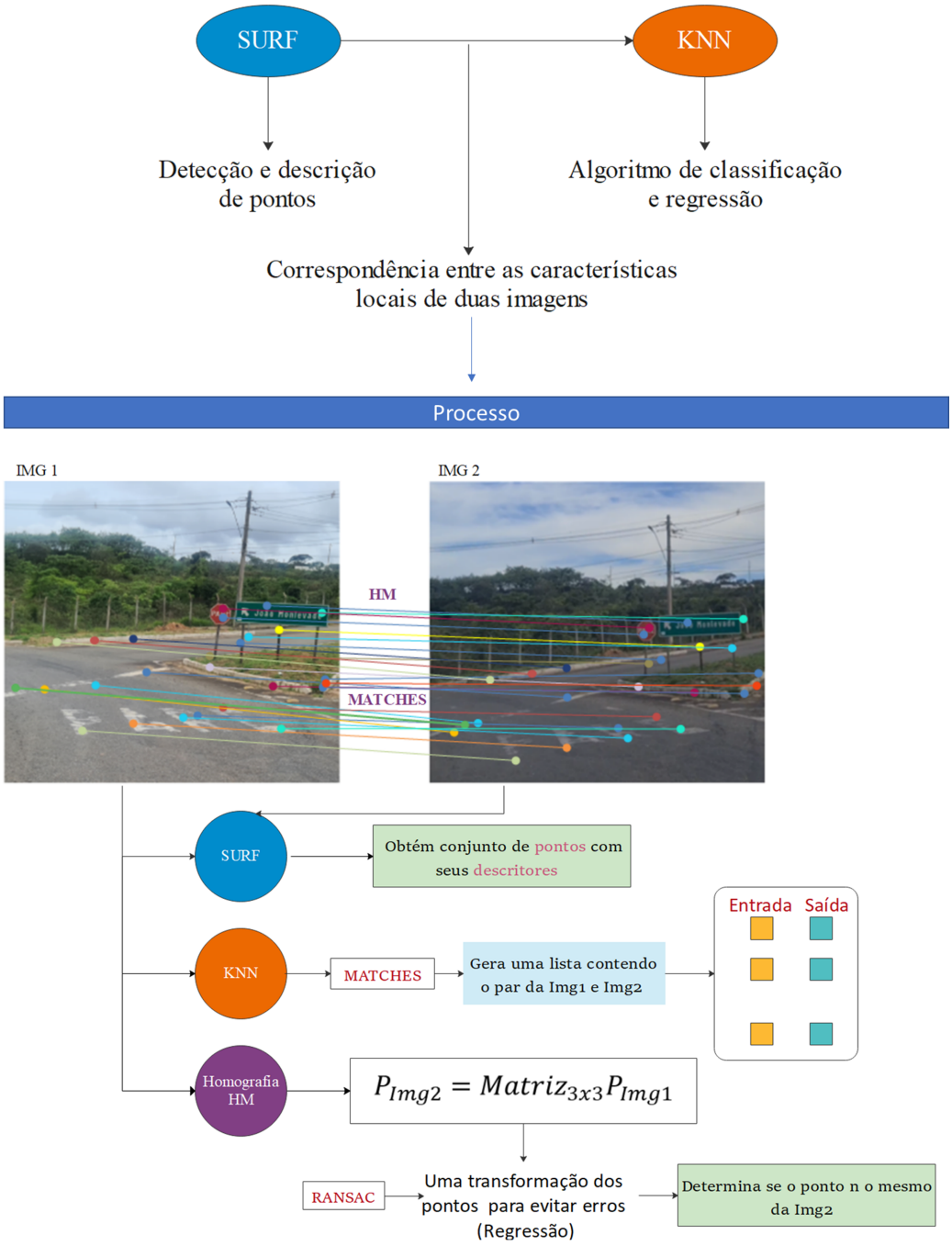

Figura 18 – Processo usado na implementação da parte de visão computacional.

3.6 Módulo localização topológica

Durante a localização topológica, o veículo registra as imagens para procurar os KP usando a representação de características baseada em SURF. Um limite de similaridade TH (*Threshold*) é adotado para julgar se a imagem da consulta corresponde aos KP no dataset.

Ao adicionar uma imagem de consulta ao dataset, é importante garantir que ela esteja conectada aos conhecimentos prévios KP já existentes, ou seja a informação adqui-

rida da latitude e longitude de acordo o mapa. Se a imagem não corresponder a nenhum dos KPs no dataset, é necessário criar um TH, para que o programa descarte esse ponto, e continue analisando as imagens. Por outro lado, se a imagem já tiver uma correspondência com um KP isso significa que está no ponto de interesse.

É importante lembrar que essa abordagem cria um loop na estrutura topológica do mapa, o que é altamente benéfico para o sistema como um todo. Isso permite que as informações sejam acessadas e atualizadas com mais facilidade, o que melhora significativamente a eficiência e a precisão do sistema. Portanto, é fundamental garantir que todas as imagens de consulta estejam conectadas aos KP relevantes para que a estrutura topológica funcione corretamente e forneça resultados precisos e úteis, da localização.

No processo de localização topológica, o veículo se localiza no ponto mais próximo comparando a imagem atualmente capturada com os KP do banco de imagens dos pontos topológicos, ou seja comparando KP com KP. Portanto, a seleção dos KPs é a questão principal na construção do mapa topológico, que deve obedecer aos seguintes princípios:

1. A similaridade entre os pontos-chave selecionados deve ser maior que um certo limite para garantir discriminabilidade e representatividade.

2. Para imagens coletadas com poucas características locais, KP devem ser selecionados para evitar instabilidade durante o processo de localização, que depende da MH e da distância, colocada no módulo de correspondência de imagem.

Para obter uma representação KP compacta, são caracterizadas as imagens por um conjunto das características mais relevantes do ambiente com o auxílio do algoritmo Brute-force feature matching que é encarregado da classificação.

A Figura 19 mostra o processo e fluxograma de localização topológica. Para a imagem atual capturada pelo veículo, calcula a similaridade entre a imagem atual e os KP existentes no mapa topológico conforme o método de correspondência de imagens do módulo de imagens matching.

Constrói um novo parâmetro para observar a semelhança entre a imagem atual e os KP existentes no mapa de topologia, se for maior que determinado limite TH, inserindo a imagem atual como um novo KP.

As imagens com KP de cada coordenada são responsáveis por fornecer as informações numéricas da cena local, como as características visuais observados pelo carro. As características visuais das imagens são transformadas em códigos hash binários para facilitar a comparação eficiente de imagens em tempo real para localização global. Desta forma, obtém-se como resposta a detecção do ponto e as coordenadas em que se encontra.

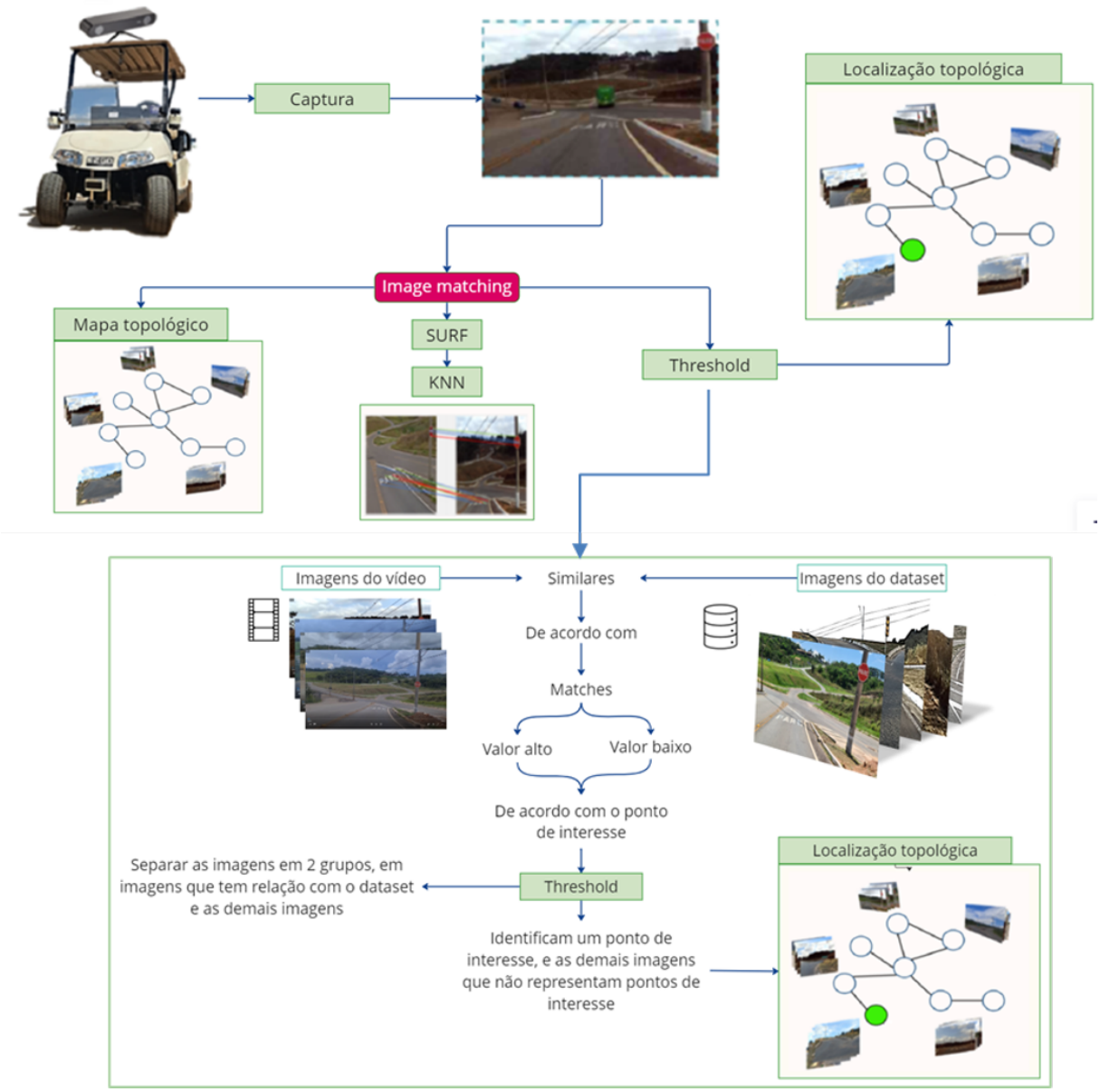

Figura 19 – Processo e fluxograma de localização topológica com visão computacional. E explicação de implementação de Threshold

3.7 Módulo análise quantitativa e qualitativa

Para saber se a proposta metodológica funciona e responde à questão de pesquisa, propõe-se uma análise onde se tem três parâmetros iniciais $K_{Pimagem}$[2], $K_{Pobject}$[3] e *matches*[4] são gerados pelo software. Esses dados contém informações ao comparar as 30 imagens de cada ponto com as 100 imagens do vídeo. Com base nessas informações é feito um ajuste neste caso denominado **MR (*Ajuste de Matches Rate*)**, que na literatura [180, 144, 154] é considerado um índice de correspondência que representa a porcentagem

[2] $K_{Pimagem}$: Pontos-chaves da imagem do banco
[3] $K_{Pobject}$: Pontos-chave das imagens do video
[4] *Matches*: Correspondência de características (ou pontos-chave) entre duas imagens (video e banco de imagens)

de correspondências em um conjunto de dados que pode corresponder com sucesso aos membros de um conjunto de dados, para isso é utilizada a seguinte Equação 3.1.

$$MatchRate = (\frac{Matches}{\frac{K_{Pimagem}+K_{Pobjeto}}{2}})100 \tag{3.1}$$

Após ajustados os valores do MR, é calculado o erro do algoritmo. Este erro é retirado da condição se as correspondências forem menores $K_{Pimagem}$ e $K_{Pobjeto}$, então, se insere 1 se há erro ou 0 caso contrário, assim sabe-se se o programa teve alguma falha na identificação dos matches.

Depois disso, é realizada a classificação do dataset das 100 imagens, onde cada ponto é correlacionado com as 30 imagens. Neste caso o estudo é realizado pelos valores máximos sendo realizada uma normalização desses valores, com a seguinte Equação 3.2:

$$NormalizedMax = \frac{Maximum - MinimumDoMaximum}{MaximumDoMaximum - MinimumDoMaximum} \tag{3.2}$$

Onde as incógnitas representam:

- NormalizedMax: Número máximo normalizado
- Maximum-MinimumDoMaximum: Número máximo do número máximo
- MinimumDoMaximum: Número mínimo do número máximo
- MaximumDoMaximum: Número máximo do número máximo
- MinimumDoMaximum: Menor número do maior número.

Os dados são processados para analisar os resultados do algoritmo que faz uma rotulação para cada ponto e comparado com a rotulação manuais das imagens. Os objetivos dos testes são descritos em seguida:

1. **Teste conforme a variação de modelo de cor durante o pre-processo:** Escolher o modelo de cor que possui maior porcentagem de matches com a imagem RGB original, para poder usá-lo durante a implementação do SURF; isto é para saber se tem alguma variação, na hora de ter uma imagem com modelo de cor diferente.
2. **Teste conforme a variação de parâmetros:** O objetivo é fazer variações da MH e o parâmetro de distância, os quais são 2 parâmetros a variação com respeito a porcentagem de correspondência entre imagens. Isto serve para analisar se essas variações de parâmetros fixos podem ser definidos para todos os testes em qualquer hora do dia.

3. **Testes em diferentes horários do dia:** Conhecer os acertos do programa, através do maior número obtido de cada coordenada, tendo em conta os parâmetros de MH e distância, fixos. Além disso, se faz a implementação de um TH para realizar uma análise.

Na Figura 20 se pode visualizar melhor a proposta de análise de acordo com cada etapa. Considerando a implementação de um valor de TH, limitando este valor em um range de 0 a 1, este é utilizado para realizar uma separação linear conforme os grupos de imagens de cada ponto reconhecido. Este limite é calculado dependendo se o maior número normalizado é maior que o número escolhido como TH, então ele pertence a P0, o que significa que ele só aceitará essa quantidade de porcentagem conforme o TH. Da mesma forma, é feito um gráfico para observar o que acontece em cada ponto detectado, e também é feita uma matriz de confusão. Isso será explicado com maior detalhe, no capítulo seguinte.

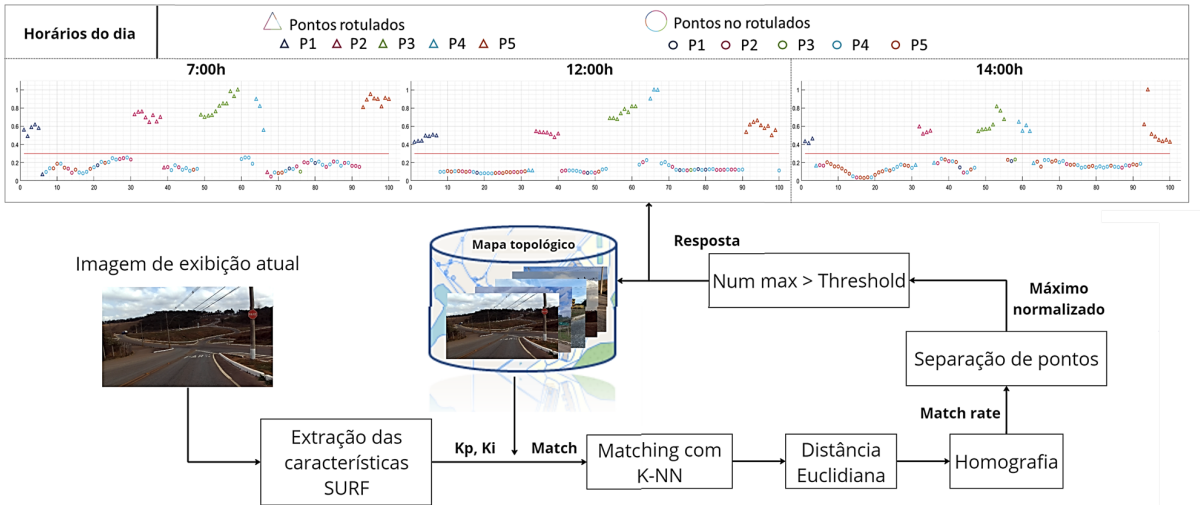

Figura 20 – O fluxograma do mapeamento topológico e análise quantitativa.

Da mesma forma, foi feita uma matriz de confusão onde é considerado aquele TH que define a rotulagem e a separação dos dados. A matriz de confusão nos ajuda a ter uma ferramenta que nos permite visualizar o desempenho do algoritmo, neste caso ele tem as seguintes características (Figura 21):

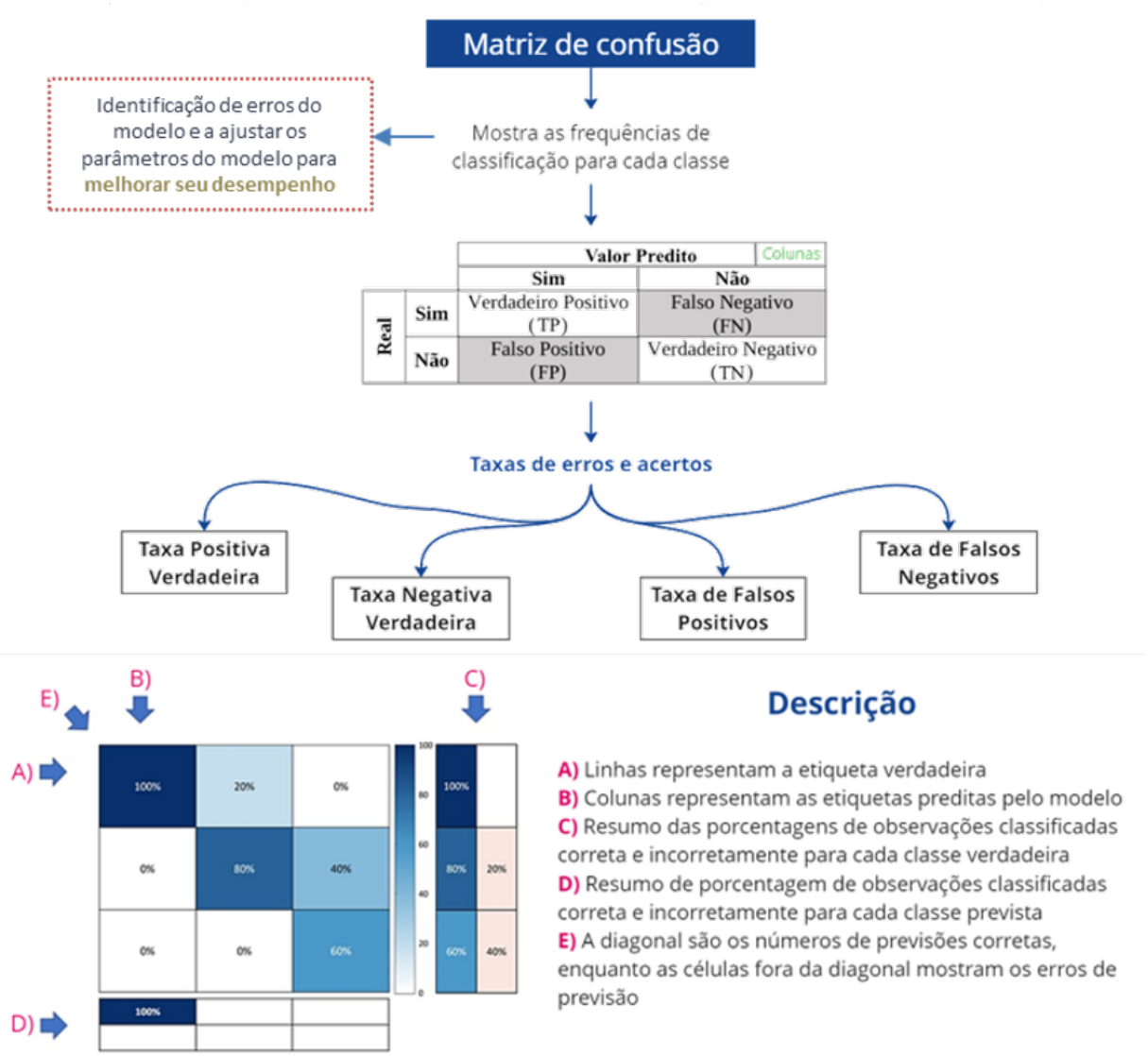

Figura 21 – Desenho da matriz de confusão e explicação.

4 Resultados Obtidos

Este capítulo é uma continuação do desenvolvimento da pesquisa onde são mostrados os resultados dos experimentos propostos a fim de responder à questão de pesquisa e demonstrar se o método proposto pode ser realmente utilizado em veículos autônomos. Cada experimento e seus respectivos resultados são descritos com mais detalhes em seguida.

4.1 Resultados quantitativos

O objetivo deste capítulo é apresentar os resultados obtidos através da implementação de uma proposta metodológica que integra localização topológica e visão computacional utilizando o algoritmo SURF. Neste sentido, serão apresentados os resultados quantitativos obtidos através da realização de vários testes. Para isso, serão tomados como referência 5 pontos específicos, que serão selecionados para comparação da base de dados composta por 30 imagens, através da implementação do SURF a partir da entrada de 3 vídeos capturados em horários diferentes (Figura 22).

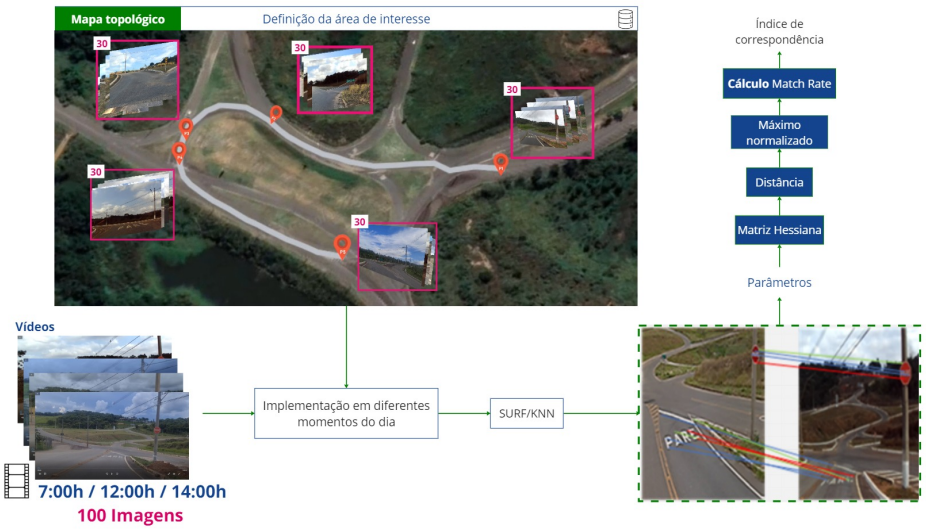

Figura 22 – Execução de proposta metodológica, segundo a integração de localização topológica e visão computacional.

A apresentação destes resultados permitirá avaliar a eficácia e eficiência da metodologia proposta e sua aplicação na identificação da localização e detecção de cada ponto do cenário. Estes resultados são fruto de um trabalho rigoroso e dedicado de vários meses de investigação e desenvolvimento. Nas seções seguintes, serão apresentados detalhadamente os resultados obtidos e sua respectiva análise, foram realizados três testes, eles são mostrados em seguida.

4.1.1 Teste conforme a variação de modelo de cor durante o pré-processo:

Para implementar um modelo de cores que ajude a identificar mais números de correspondências da imagem de entrada, existe uma imagem base no modelo de cores RGB de cada ponto, então é feita uma comparação com 7 modelos de cores diferentes (GrayScale, HSV, CIE XYZ, HLS, CIE 1976 LUV, YcbCr e CIE Lab), para saber se existe alguma diferença em trabalhar com um modelo de cor específico, pois visa detectar mais correspondências durante o pré-processamento.

É realizado um teste onde cada modelo de cor é comparado com relação à imagem original, obtendo o KP que representa os pontos característicos da imagem original, e o KI (*Application Program Interface / KPImage*) que neste caso seriam os pontos das imagens dos 7 modelos de cores diferentes, após a obtenção disso o código mostra as correspondências das duas imagens.

Tendo em conta estes três valores, aplica-se o MR e analisa-se o valor máximo obtido em cada modelo de cor da imagem e depois é realizada uma normalização de cada valor, obtendo como resultado que o modelo de cor em tons de cinza obtém mais partidas de cada ponto. A Figura 23 mostra os resultados segundo a análise do maior número de correspondências.

		Modelo de Cor	Kp Objeto RGB	Max matches rate (MR)	Max normalizado (MNMR)
Comparação de imagens RGB	1	GrayScale	1945	85,39	1
	2	HSV	1945	36,59	0,38
	3	CIE XYZ	1945	82,73	0,61
	4	HLS	1945	54,98	0,57
	5	CIE 1976 LUV	1945	18,56	0,15
	6	YcbCr	1945	28,95	0,28
	7	CIE Lab	1945	13,82	0,09

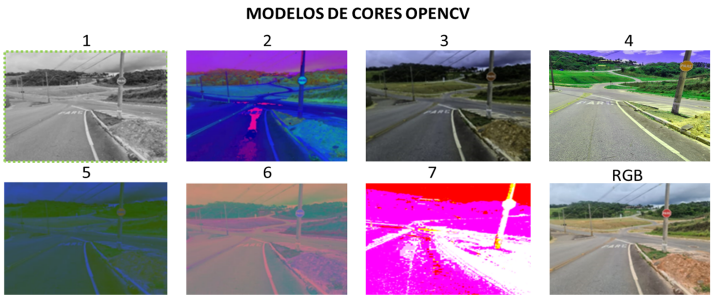

Figura 23 – Resultado da comparação dos modelos de cores conforme o maior número de correspondências.

Os valores da coluna KpObject RGB representam o KP que a imagem que está sendo comparada contém, o MR representa os valores máximos aplicando o erro e o MNMR (*Máximo normalizado*) representa o número máximo normalizado . O que significa é que quanto maior o MR, melhor o resultado, pois detecta-se mais KP.

Isso significa que todas as imagens que entrarão no sistema serão em tons de cinza, já que é o modelo de cores que possui maior número de correspondências, também por ter apenas um canal, é aquele que leva menos tempo para ser processado.

4.1.2 Teste conforme a variação de parâmetros:

Este teste será executado em um dataset de imagens, bem como em cada frame de um vídeo. O resultado deste experimento é um conjunto de pontos-chave com seus respectivos descritores, a fim de analisar a variação de alguns parâmetros.

Para a detecção dos MT (*Matches das imagens*) e extração dos descritores, o primeiro passo consiste na detecção dos pontos-chave para extrair seus descritores do objeto KP do dataset e imagens KI do vídeo, sendo os números de pontos que serão encontrados e descritos pelo algoritmo. Essa detecção dos MT de correspondências e descritores é feita pelo algoritmo SURF mencionado acima e usado para realizar a correspondência pelo algoritmo Brute-force feature matching.

A Figura 24 mostra o fluxograma considerado para obtenção dos resultados, onde está disponível a base de dados e a entrada do vídeo, é aplicado o algoritmo SURF, que permite a modificação de parâmetros como matriz hessiana e distância, depois o algoritmo Brute-force feature matching é implementado, o que permite obter as correspondências de comparação das informações de entrada e informações de saída. Em seguida é implementada a equação TH, responsável por fazer uma linha que representa a separação dos pontos que foram rotulados e não rotulados, onde o TH considerou os pontos de interesse que são coincidentes e as imagens que não representam pontos de interesse. Obtendo assim a posição da localização do veículo.

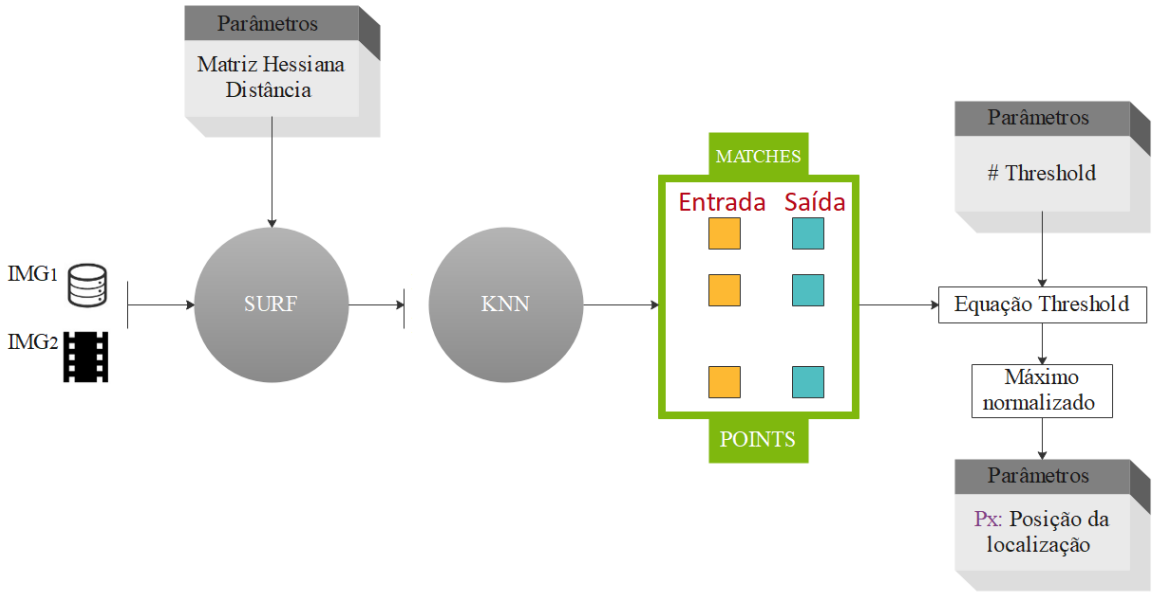

Figura 24 – Fluxograma para analisar a variação dos parâmetros de distância e matriz Hessiana, conforme a detecção de pontos-chave.

Levando isso em consideração, é feita uma análise da modificação do MH visto que ele é utilizado devido a sua boa relação entre precisão e custo de tempo [178, 154, 144]. Nesse caso, o determinante do MH é utilizado para localizar os pontos e para determinar

a escala, ou seja, se for determinado um valor que garanta um número maior de pontos, o sistema pode se tornar mais eficiente e em algum ponto pode tornar-se autônomo ao implementá-lo em diferentes momentos do dia. Neste teste foi variado com três valores dos quais deu a seguinte Figura 25 de resultado.

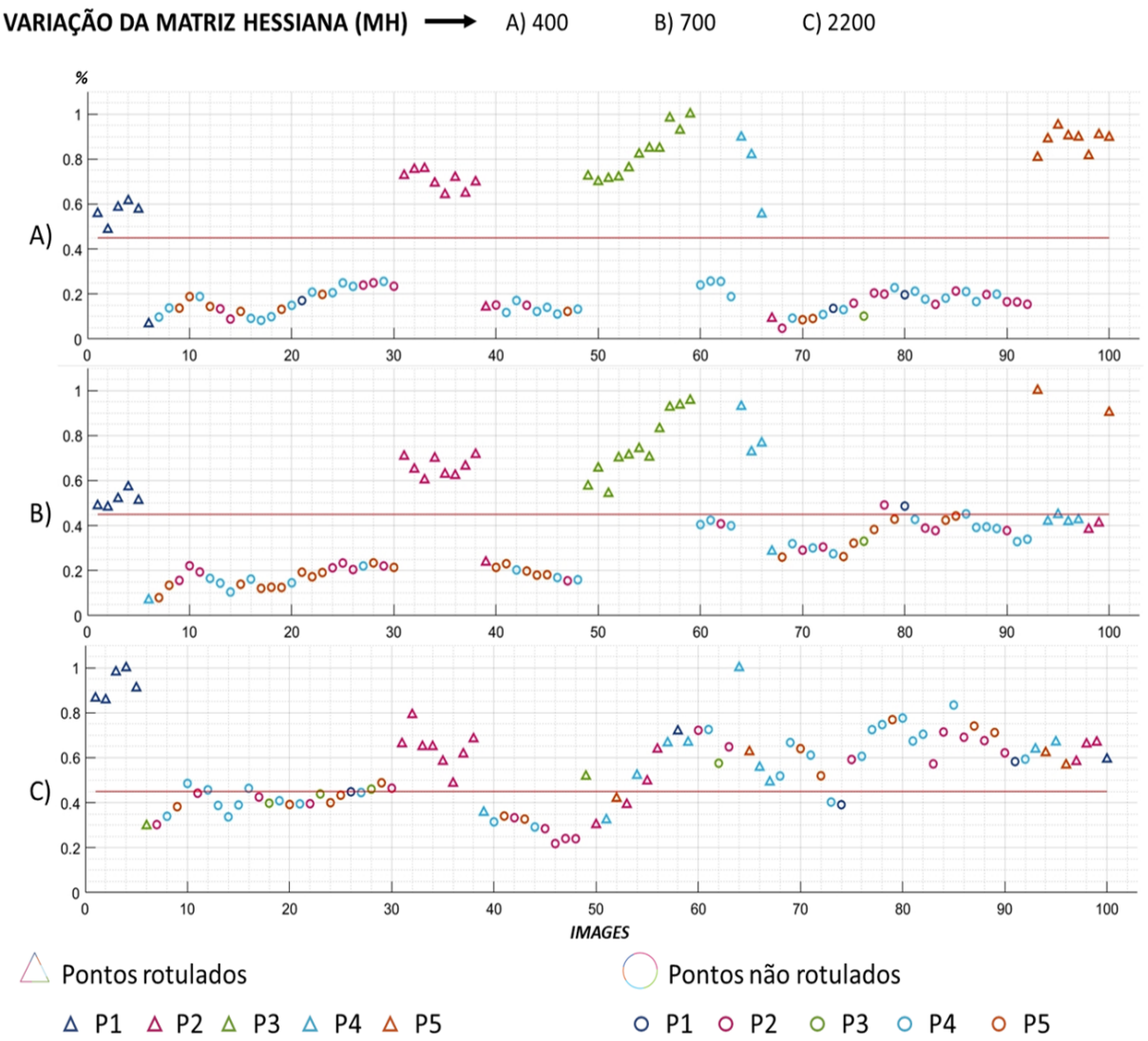

Figura 25 – Resultado da detecção de cada ponto segundo a detecção das 100 imagens de vídeo, dependendo da variação da matriz Hessiana. A) 400, B) 700 e C) 2200.

Esta representação saiu conforme a análise do seguinte pseudocódigo 1:

Algorithm 1 Pseudocódigo implementado para análise da detecção de pontos característicos das imagens, conforme variação de parâmetros da matriz Hessiana.

Start
    Carregar dataset de imagens para 5 pontos.
        100 imagens dadas por um vídeo.
        30 imagens de dataset.
    Rotulagem manual de imagens de vídeo.
        As imagens que pertencem a cada ponto, classificado como P1, P2, P3, P4 e P5.
    Obtenha resultados do SURF, conforme a comparação da imagem de vídeo com o dataset de imagens.
        Objeto KP, imagem KI, matches MT.
    Análise das 1500 possibilidades, quais imagens foram detectadas e com que ponto.
    Fazer um ajuste, implementando match rate.
        Um ajuste é feito e o erro % do algoritmo é obtido.
    O valor máximo e mínimo das 100 imagens é obtido para cada ponto.
        Comparando as 1500 possibilidades.
    O valor mínimo do máximo e do máximo do máximo é obtido.
    A normalização do valor máximo é realizada.
    Três parâmetros são obtidos para rotulagem.
        - VM: Valor máximo normalizado das 100 imagens.
        - RA: Rotulagem de algoritmo.
        - RM: Etiquetagem manual.
    VM, RA e RM são representados graficamente.
        Considerando o valor máximo de onde não deve detectar.
    Um treshold (TH) é analisado, o qual é a separação de cada um dos pontos.
    Uma matriz de confusão é implementada.
        Tendo em conta o P0.
    End

Como resultado, na Figura 25 o triângulo representa os máximos normalizados reconhecidos referentes à análise de MR e o círculo representa os pontos máximos reconhecidos pelo maior número, a linha vermelha representa os TH que poderiam ser usados para separar os conjuntos. Nesse experimento conclui-se que quanto maior o MH, mais pontos ele encontrará, ou seja, ele também tem maior possibilidade de errar, visto isso, ele opta por usar um MH de 400.

Surge a dúvida quanto à precisão da detecção em alguns pontos, uma vez que o algoritmo detecta com um número baixo ou em outro ponto, levantando a hipótese de um erro manual na rotulagem das imagens. Para investigar essa questão, a Figura 26 apresenta uma análise do comportamento da borda de erro do algoritmo em relação à detecção de cada ponto e imagem. Os pontos detectados são representados por um triângulo e os não detectados por um círculo na seção A) da Figura 25, com variação de MH de 400. A linha laranja representa a região detectada pelo algoritmo e a linha azul representa os pontos rotulados manualmente. Em relação a imagem 6, o programa detectou que corresponde ao ponto P1, mas com um valor baixo. Analisando a imagem anterior, indica que o algoritmo detectou mais pontos e, embora possa parecer um erro na rotulagem, a avaliação humana conclui que o ponto pertence a P1, mesmo que o algoritmo não possua características suficientes para detectá-lo.

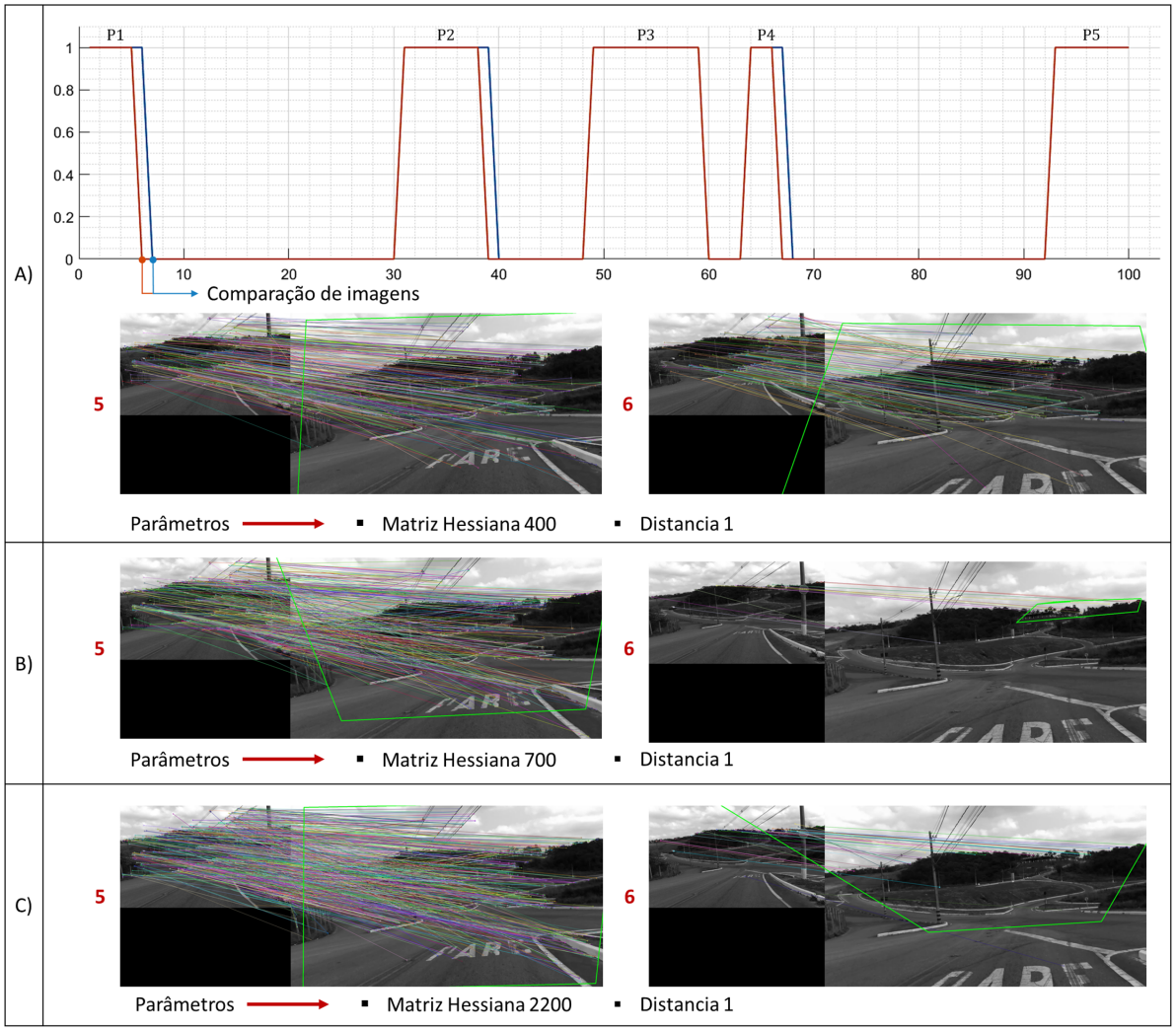

Figura 26 – Comparação da figura conforme as imagens 5 e 6 do dataset, onde a) mostra o gráfico que detectou mais pontos, a representação do programa é mostrada ao detectar os pontos mais próximos nos pontos 5 e 6 segundo a matriz Hessian B) 700 e C) 2200.

Por outro lado, a seção B) das Figuras 25 e 26 mostra que uma porcentagem maior é detectada como falso positivo, neste caso pode-se observar que a imagem está detectando muito poucos pontos e está detectando os pontos onde não representa nada realmente característico, ao contrário da seção C) (Figuras 25 e 26) onde detecta bem todos os pontos, mas tem a desvantagem de que nos pontos seguintes não detecta o que tem que detectar, isso se deve ao fato de que quanto mais pontos, maior a possibilidade de haver erro, isso é evidenciado melhor por meio da matriz de confusão, a qual é a visualização do desempenho do algoritmo em cada variação do MH (Figura 27), pensando em uma separação linear mais simples, é classificado por P0, ou seja, o TH, onde existem 2 representações, uma é a baixíssima, pontos que não seriam considerados não rotulados e a outra séria o detectado.

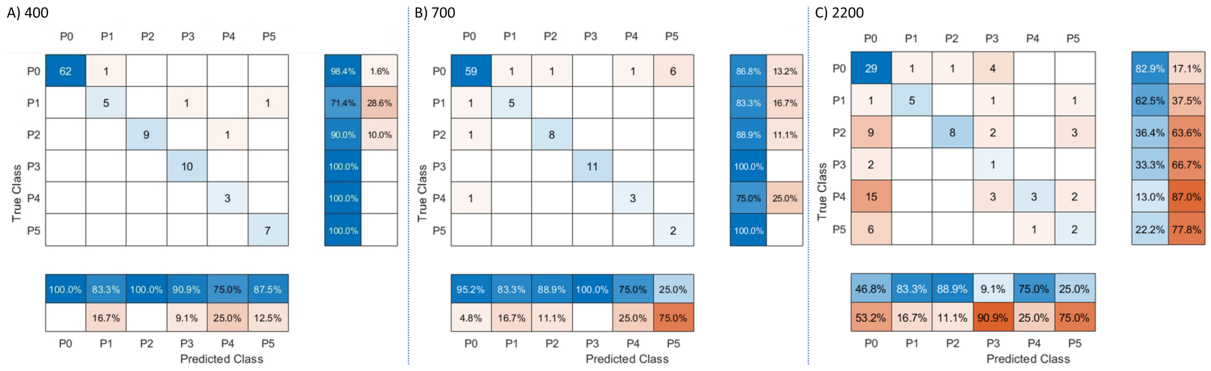

Figura 27 – Matriz de confusão para cada variação da matriz Hessiana A) 400, B) 700 e C) 2200.

Considerando estes resultados, será utilizado um MH de 400. Em seguida, é realizada a variação de **Distância (*DIS*)**, onde após utilizar uma busca KNN, calcula a "distância"entre um descritor de consulta e todos os descritores e retorna os k pares com a menor distância. Aqui manteremos k=2. Portanto, agora tem 2 pares correspondentes para cada descritor de consulta. Basicamente, a pesquisa Brute-force feature matching nos deu 2 correspondências para cada descritor de consulta. O importante é decidir quais dessas partidas são "boas". Para isso, a estratégia utilizada é pegar os dois vizinhos mais próximos e comparar suas distâncias com o ponto-chave de consulta.

Varia-se esse parâmetro **Distância** de 0,75, 0,85 e 1, para observar se tem alguma variação ao detectar os pontos, a Figura 28 mostra os resultados, com a representação da matriz de confusão.

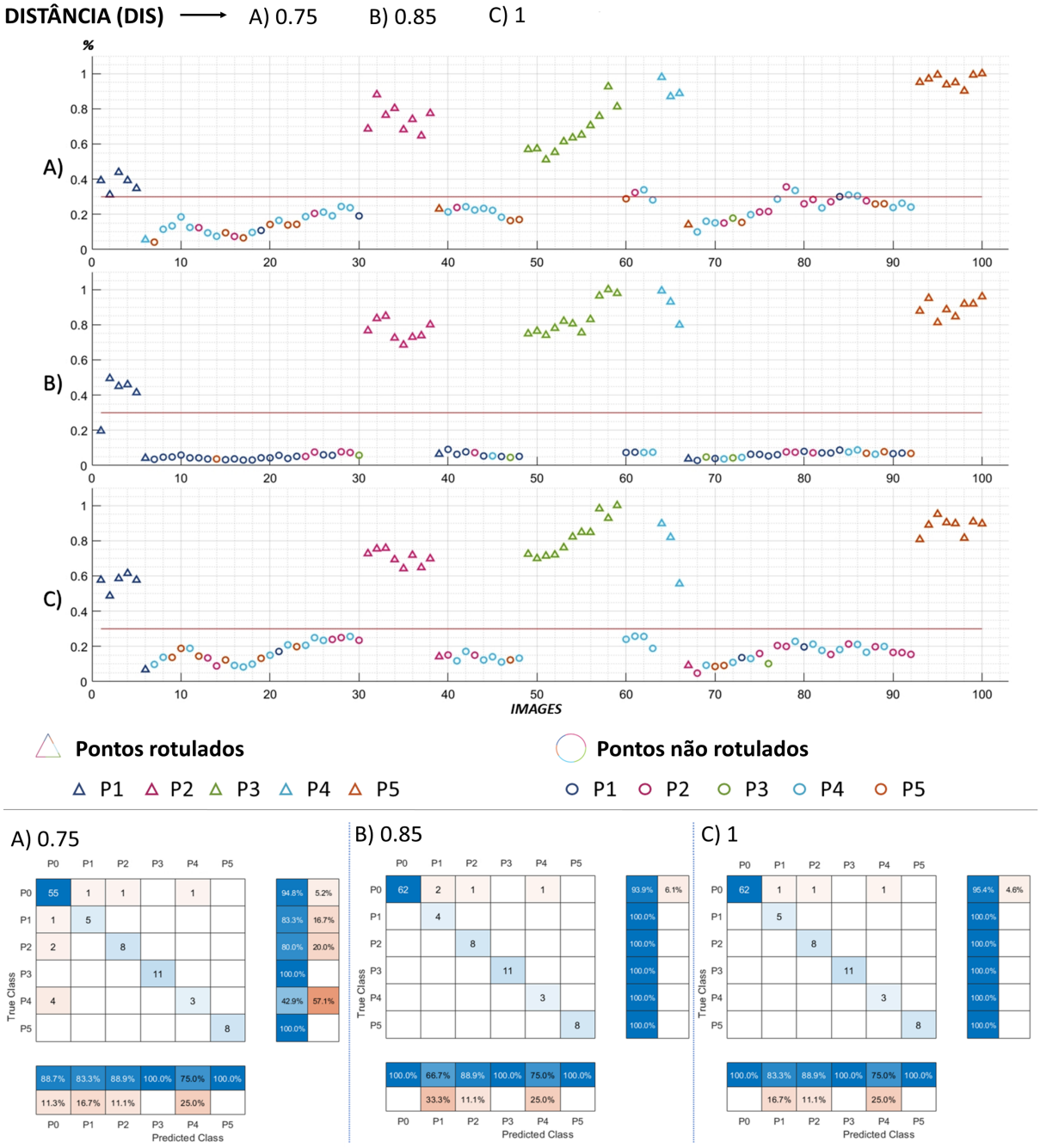

Figura 28 – Representação graficamente e por meio de matriz de confusão, conforme a variação da distância em A) 0,75, B) 0,85 e C)1.

Conforme a Figura 28, pode-se observar que quanto maior o número de 0 a 1, melhor será o resultado, pois neste caso estaria analisando 100% dos píxeis da imagem. No entanto, o TH não é muito bem visualizado em que faixa ele poderia ser variado, por isso, considerando que o MH 400 e o Distância 1 vão ser usados, ele é obtido tomando um TH com um intervalo de **[0,26-0,42]**, sendo o resultado original do algoritmo que mostra que os números abaixo de 0,26 são P0 que correspondem às imagens não marcadas, faz-se também uma média de cada um dos pontos e depois uma média ponderada que representa a porcentagem de acertos representada no gráfico da Figura 29 .

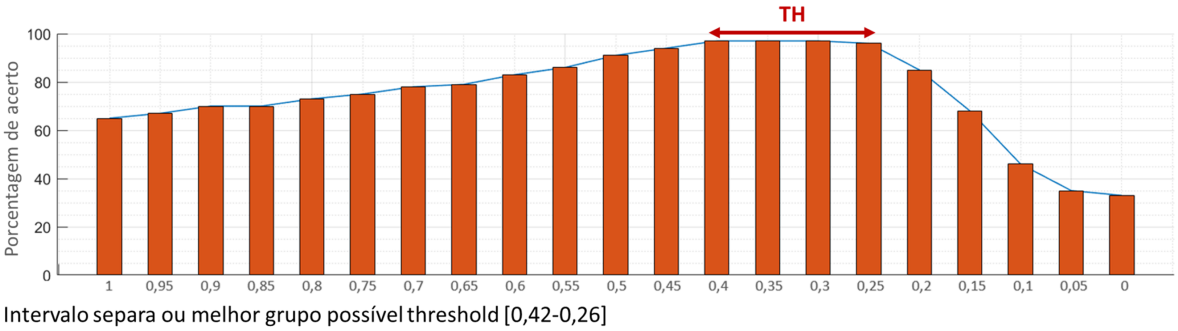

Figura 29 – Representação do intervalo threshold segundo a variação de 0 a 1.

Em conclusão, se o número de MaxNo (*Maior número normalizado de correspondências*) for maior que o TH, algum ponto corresponde, se P0 não for colocado, ou seja, esse TH não for selecionado intuitivamente, a matriz de confusão e a representação gráfica ajudam a encontrar esse intervalo, busca-se também que com esse intervalo possa subdividir o grupo de pontos em diferentes horários do dia.

4.1.3 Testes em diferentes horários do dia:

Após saber que com um MH de 400, um Distância 1 e um intervalo de TH 0.4, bons resultados são obtidos, e é evidente que a porcentagem de acertos melhora, neste teste são usados diferentes horários do dia: 7h00, 12h00 e 14h00. A Figura 30 mostra os resultados obtidos.

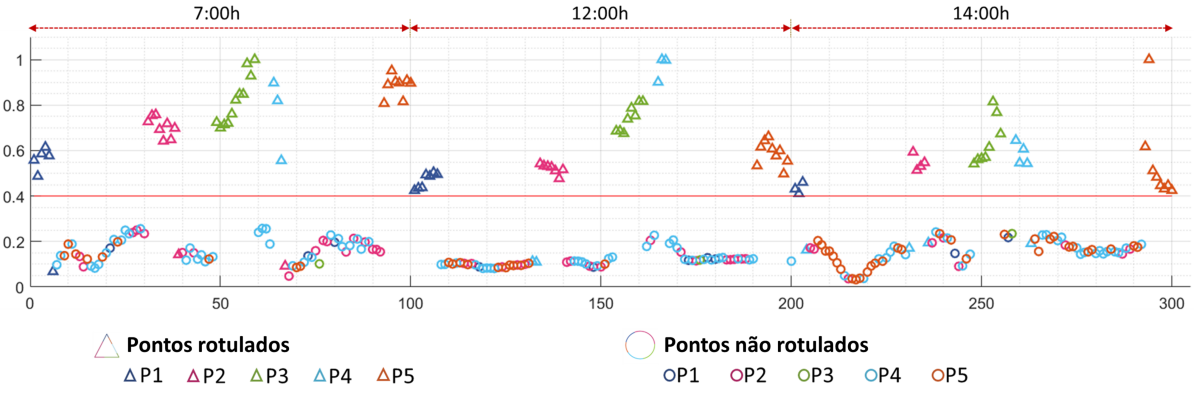

Figura 30 – Resultado da detecção de pontos durante diferentes horários do dia com um matriz Hessiana 400, uma distância de 1 e um threshold 0.4.

Conforme o acordo TH de 0,4, é aplicada média onde os valores dados pela matriz de confusão são considerados de acordo com cada hora do dia, neste caso é aplicada uma média, resultante em qual algoritmo detectou $91.48333333\% \approx 91,48\%$ correto com um erro de $8{,}516666667\% \approx 8,52\%$ (Tabela 3).

Horários	Acertos						Média	**Média de acertos**
7:00am	100	83,3	88,9	100	75	100	91,2	91,48333333
12:00pm	100	100	77,8	100	100	100	96,3	
14:00pm	100	75	66,7	100	80	100	86,95	
Horários	Erros						Média	**Média de erros**
7:00am	0	16,7	11,1	0	25	0	8,8	8,516666667
12:00pm	0	0	22,2	0	0	0	3,7	
14:00pm	0	25	33,3	0	20	0	13,05	

Tabela 3 – Resultado da matriz de confusão conforme o número de acertos e erros em diferentes horários do dia.

Em conclusão, consegue-se um bom número de acertos implementando um TH de 0.4, MH 400 e um Distância 1, em diferentes momentos do dia, para validar este teste, a integração de todos os módulos apresentados da topologia com a integração de imagens é implementado.

4.2 Resultados qualitativos

A importância dos dados qualitativos reside no fato de que eles permitem obter um conhecimento mais profundo sobre certas realidades subjetivas, que estão ocorrendo no algoritmo proposto. Para isso, a Figura 31 mostra como é criada a interface gráfica, que se divide em três etapas principais A) Aquisição dos dados B) Obtenção do arquivo .OSM e C) Seleção do ponto **A** até o ponto **B**. Em seguida o algoritmo implementa a metodologia proposta, onde identifica os 5 pontos propostos por meio de VC.

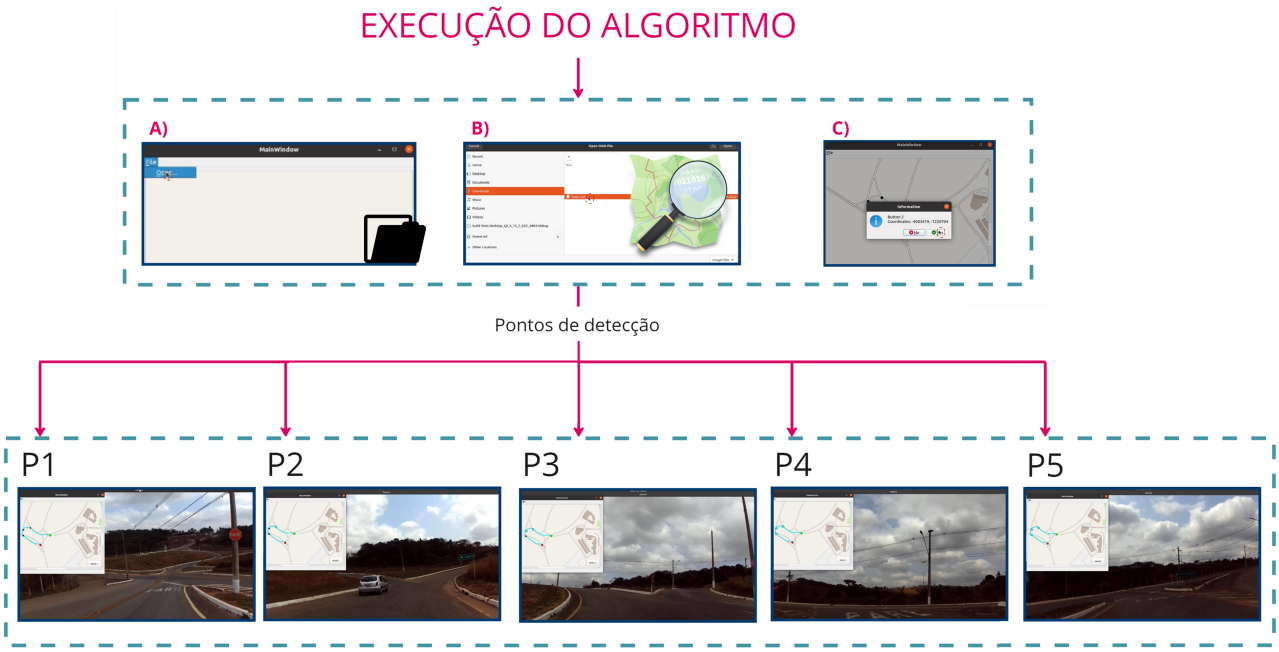

Figura 31 – Execução do programa de acordo com cada ponto detectado, A) Aquisição dos dados, B) Seleção do arquivo .OSM, C) Seleção do ponto inicial e ponto final.

Como resultado dos testes realizados, foi disponibilizado um vídeo publicamente[1] demonstrando a execução do algoritmo em diferentes horários, às 7h00, 12h00 e 14h00. Por meio disso visualiza-se que nos três momentos do dia, foi possível detectar e realizar a integração de VC e localização topológica. Tendo em conta que com ajuda dos parâmetros MH de 400, um Distância 1 e um intervalo de TH 0.4 detecto todos os pontos de localização propostos.

4.3 Resultados de robustez do algoritmo

Essa proposta nasce em resposta ao aparecimento de um certo problema, neste caso, é para responder se **puder ser implementado uma localização topológica usando um câmera?**. O algoritmo implementado é composto por uma série finita de etapas que convergem na solução, explicadas no capítulo 3 (metodologia), no entanto,

[1] https://youtu.be/9L0lpsMWs3Y

qualquer virada inesperada do ambiente em que está funcionando pode ser um problema, ou seja, o algoritmo implementado deve ser flexível para alterações e variações do ambiente.

Deve funcionar de maneira robusta, resolvendo problemas de ruído, padrões de entrada, aprendizado incompleto e adaptável. Essa abordagem encontra características mais distintas da imagem e, em seguida, combina a previsão para estimar e se adaptar às novas mudanças no ambiente.

Atualmente, estão realizando uma remodelação a UNIFEI, portanto, o meio ambiente (estradas, sinais de trânsito, rua, calçada, etc.) estão mudando (Figura 32).

Figura 32 – Variação da cena principal para a atual, há uma grande mudança nos pontos 2 e 3, onde não tem sinalização.

Como mostrado na Figura 32, os pontos 2 e 3 são afetados porque não tem a sinal de trânsito, neste caso a questão que se coloca é **se é possível que o algoritmo detecte estes pontos sem a necessidade de qualquer modificação, ou se alguma metodologia pode ser implementada para aumentar a porcentagem de sucesso em cada ponto?** Para isso foi gravado um vídeo às 10:00h, e prosseguimos com a execução do código, neste caso obtivemos o gráfico mostrado na Figura 33 A), é evidente que ao analisar os pontos 2 e 3 detecta apenas um ponto com um TH de 0,3 e dá um percentual de 68,83% de sucesso durante a trajetória, dado pela média apresentada na matriz de confusão.

Para melhorar este número de acertos, procedemos a aumentar a base de dados com cinco imagens dos pontos 2 e 3, como mostrado na Figura 33 B), é evidente que o número de acertos aumentou em 14,99%. Em conclusão, o algoritmo se mostrou robusto no que diz respeito ao aumento da base de dados, independentemente da mudança de ambiente.

Por outro lado, foi disponibilizado um vídeo publicamente [2], é apresentado a visualização da resposta do algoritmo, onde é evidenciado que detecta cada ponto sem a necessidade que o ambiente seja igual, que no anterior estudado. Além disso, com a implementação das imagens, foi possível identificar todos os pontos da cena. Da mesma forma,

[2] https://youtu.be/O82Gf2GUCFA

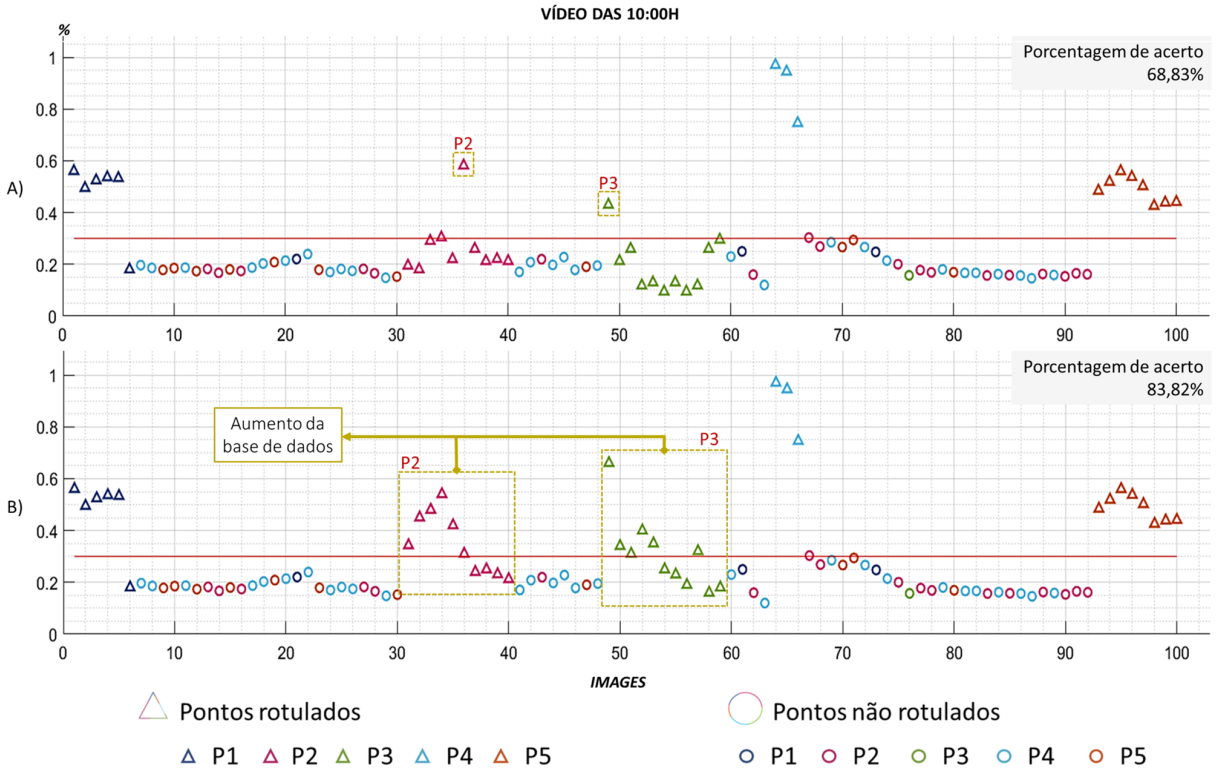

Figura 33 – Resultado da robustez do algoritmo, vídeo feito às 10:00, A) sem modificação no dataset e B) com modificação no dataset.

outro vídeo é gravado às 16h e o algoritmo é executado, ele detecta todos os pontos, sem nenhum problema.

5 Discussão dos resultados

Durante este trabalho, foi desenvolvido um software que implementa uma metodologia proposta (Figura 13) para responder à questão de pesquisa: **Uma localização topológica pode ser implementada usando um conjunto de imagem e técnicas de processamento de imagens?**

A resposta é se a localização topológica pode ser realizada através da implementação de técnicas de processamento de imagem. Uma vez que, conforme os resultados obtidos, foi possível analisar três horários diferentes durante o dia, resultando la implementação de uma localização topológica. Obtendo-se, assim, mais de 91,5% de acertos em cada ponto. Por outro lado, ao implementar o TH um novo programa é feito para retificar, sim, realmente o número do TH pode ser padronizado para implementar a qualquer hora do dia antes do anoitecer. Para isso, é gravado um vídeo às 10h e 18h onde é implementada toda a proposta metodológica. Este resultado pode ser evidenciado em um vídeo publicamente disponível[1], isso mostra a integração da localização topológica e também a detecção de cada ponto, obtendo assim que o algoritmo detecte pelo menos um ponto em diferentes momentos do dia.

Mostrando que essa técnica é interessante ser aplicada e pode definir uma localização topológica, surge a questão se **pode ser viável utilizar essa metodologia proposta para implementar uma navegação topológica?**. Argumenta-se que, dependendo do tipo de navegação, é extremamente importante localizar pelo menos um ponto local preciso, o que significa que, para esse espaço de trabalho dos cinco pontos em que é trabalhado, ele pode ser implementado, uma vez que essa localização topológica necessária não precisa ser precisa mais de uma vez.

Neste caso se poderia dizer que sim, é viável realizar uma navegação topológica, uma vez que mesmo que o código tenha erros, isso não significa que seja ruim, significa que o algoritmo só não consegue alcançar imagem alguma que não tem tantos pontos característicos, mas neste caso não afetaria nada, porque para realizar uma navegação topológica, só precisa detectar pelo menos um único ponto.

Pelo menos para este circuito analisado pode-se dizer que é realmente viável fazer uma navegação topológica, uma vez que está detectando uma de todas as possibilidades de chegada ao ponto, de n inconveniente.

Em conclusão, mesmo que se tenha algum erro no algoritmo, na navegação pode aceitar que, é necessário ser um valor temporário, e como evidenciado, neste caso não tem falsos positivos acima do TH, o que significa que para esses resultados nesta área de

[1] https://drive.google.com/file/d/1eZl7aaq-BBCAyICQjUs2fhgQtwN2n4Te/view?usp=share$_l ink$

trabalho, esta ferramenta poderia ser atacada em uma navegação topológica, porque ele realmente consegue identificar em alguns setores onde o carro está localizado, e qual é a região esperada.

Ou seja, a informação dada pelo algoritmo pode ser traduzida em que ele realmente identifica no ponto onde o veículo está, então, uma trajetória pode ser feita. Esta informação semântica se traduz em uma representação precisa naquele mapa local, há uma navegação segura, segundo as informações topológicas não precisas.

No entanto, pode ter problemas em um ambiente maior ou em um ambiente onde é difícil encontrar características comuns das imagens, mas para isso, em um trabalho futuro outra técnica poderia ser implementada, assim mesmo, aplicar métodos mais complexos, colocar as coisas com maior semelhança e ver se realmente continua a ser eficiente. No entanto, à medida que problemas ou dificuldades aparecem, pode acoplar e integrar parte sensorial ou até mesmo aplicar um filtro Kalman, para resolver possíveis problemas de instabilidade.

6 Conclusões

Na visão computacional, o desempenho dos algoritmos utilizados afeta diretamente os resultados das aplicações em tempo real. Isso ocorre na área de veículos autônomos, onde o processamento é feito continuamente, exigindo um tempo de resposta dos algoritmos o mais rápido possível, pois atrasos no processamento podem causar problemas como colisão e, consequentemente, insegurança.

Uma das funções utilizadas neste trabalho foi a implementação da identificação de correspondência de ponto-chave, responsável por identificar pontos congruentes entre duas imagens diferentes para realizar a correspondência e definir a posição com a ajuda de localização e integração topológica. Para isso foi necessário identificar e descrever os pontos-chave. Foi proposta uma metodologia para saber se um dataset contendo a estruturação de imagens pode ser realmente gerado e transformado em dados topológicos com a integração de uma câmera.

Como resultado, obteve-se que o número de acertos para a base de dados utilizada foi de 91,5%, ou seja, quantitativamente, teve uma taxa de acerto média superior a 90% com os parâmetros implementados.

Por outro lado, o comportamento do algoritmo foi analisado em diferentes momentos do dia, adicionando um intervalo de TH [0,42-0,26] que representa a separação dos pontos detectados e não detectados. Da mesma forma, conclui-se que mesmo que haja um erro de 8,5% no algoritmo, na navegação pode se aceitar isso, porque é necessário detectar até mesmo um ponto para saber onde ele está. O que significa que para estes resultados e este entorno estudado, esta ferramenta poderia ser implementada em uma navegação topológica, pois ela realmente consegue identificar em alguns setores onde está o carro, e qual a região esperada.

Da mesma forma, em termos de robustos do algoritmo, é possível mostrar que, independentemente da variação do ambiente, a incrementação do dataset contendo a estruturação de imagens pode ser realizada, a fim de continuar detectando a posição onde está localizado.

Em outras palavras, as informações fornecidas pelo algoritmo podem ser traduzidas no que ele realmente identifica no ponto onde o veículo está localizado, para ser feita uma trajetória do veículo. Essas informações semânticas podem ser traduzidas em uma representação precisa naquele mapa local, então, há uma navegação segura, baseada em informações topológicas imprecisas.

Finalmente, foi sim respondido à pergunta de pesquisa, dando um resultado posi-

tivo. Além disso, é possível realizar uma navegação topológica, na área estudada, porque mesmo que o código tenha erros, isto não significa que seja ruim, significa que o algoritmo simplesmente não pode alcançar nenhuma imagem que os tenha. Em conclusão, mesmo que o algoritmo não detecte um ponto, na navegação pode aceitar isso, porque é necessário que seja um valor temporário, e como evidenciado, neste caso não há falsos positivos acima do TH, o que significa que por estes resultados e esta área de trabalho, esta ferramenta poderia ser atacada em uma navegação topológica, porque ela realmente consegue identificar em alguns setores onde o carro está, e qual é a região esperada. Em outras palavras, as informações dadas pelo algoritmo podem ser traduzidas no que ele realmente identifica no ponto em que o veículo está localizado, para uma trajetória poder ser feita.

Referências

1 OPENCV. Opencv: Introduction to surf (speeded-up robust features). 2022. Disponível em: <https://docs.opencv.org/3.4/df/dd2/tutorial_py_surf_intro.html>. 11, 38, 41, 48, 49, 52, 53, 70

2 HELENA, B. Umrechnung zwischen geographischen koordinaten (länge und breite) und dem utm-format - pdf kostenfreier download. 2017. Disponível em: <https://docplayer.org/24003599-Umrechnung-zwischen-geographischen-koordinaten-laenge-und-breite-und-\dem-utm-format.html>. 11, 56, 57

3 MONDEJAR, M. T. Development of calculation applications for topography in utm projection. p. 0–76, 2015. Disponível em: <https://upcommons.upc.edu/bitstream/handle/2117/77269/memoria.pdf>. 11, 57

4 ORGANIZATION, W. H. *Road traffic injuries*. 2022. Disponível em: <https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>. 18

5 ORGANIZATION, W. H. *Blindness and vision impairment*. 2022. Disponível em: <https://www.who.int/es/news-room/fact-sheets/detail/blindness-and-visual-impairment>. 18

6 ROJAS-RUEDA, D. et al. Autonomous vehicles and public health. *https://doi.org/10.1146/annurev-publhealth-040119-094035*, Annual Reviews, v. 41, p. 329–345, 4 2020. ISSN 15452093. Disponível em: <https://www.annualreviews.org/doi/abs/10.1146/annurev-publhealth-040119-094035>. 18

7 LUIS, A.; RODRíGUEZ, T. Una aproximación general al desarrollo de los coches autónomos * uma abordagem geral para o desenvolvimento de carros autônomos a general approach to the development of autonomous cars. *Revista CTS*, v. 16, p. 153–175. 18

8 BONDAREV, A.; PRYSTAJ, A.; PRONENKO, V. Gps-synchronization optimization process of autonomous data collection systems. *Proceedings - 16th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET 2022*, Institute of Electrical and Electronics Engineers Inc., p. 486–490, 2022. 18, 19

9 DOGRAMADZI, M.; KHAN, A. Accelerated map matching for gps trajectories. *IEEE Transactions on Intelligent Transportation Systems*, Institute of Electrical and Electronics Engineers Inc., v. 23, p. 4593–4602, 5 2022. ISSN 15580016. 18

10 WINOTO, A. S. et al. Proposed autonomous vehicle framework for safe driving. *2019 7th International Conference on Cyber and IT Service Management, CITSM 2019*, Institute of Electrical and Electronics Engineers Inc., 11 2019. 18

11 LARASATI, H. T.; LE, T. T. H.; KIM, H. Trends of quantum computing applications to computer vision. *2022 International Conference on Platform Technology and Service, PlatCon 2022 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., p. 7–12, 2022. 18

12 HU, X. et al. The 2020 low-power computer vision challenge. *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems, AICAS 2021*, Institute of Electrical and Electronics Engineers Inc., 6 2021. 18

13 VALKO, N. V.; OSADCHYI, V. V. Review of state of computer vision technologies development in the world and ukraine. *Science and Technology Education Journal of Physics: Conference Series*, IOP Publishing, v. 2288, p. 12002, 2022. 18

14 DANIUšIS, P. et al. Topological navigation graph framework. *Autonomous Robots*, 2021. ISSN 15737527. 19, 26

15 XIAN, Z. et al. A bionic autonomous navigation system by using polarization navigation sensor and stereo camera. *Autonomous Robots*, Springer New York LLC, v. 41, p. 1107–1118, 6 2017. ISSN 15737527. 19

16 ZHENG, X. et al. Analysis of autonomous underwater vehicle (auv) navigational states based on complex networks. *Ocean Engineering*, v. 187, 2019. ISSN 00298018. 19, 26

17 NAKHAEINIA, D.; KARASFI, B. A behavior-based approach for collision avoidance of mobile robots in unknown and dynamic environments. *J. Intell. Fuzzy Syst.*, v. 24, p. 299–311, 2013. 19

18 CHEN, C.-H.; SHIOU-YUN, J.; LIN, C.-J. Using an adaptive fuzzy neural network based on a multi-strategy-based artificial bee colony for mobile robot control. *Mathematics*, v. 8, p. 1223, 07 2020. 19

19 JAISWAL, R. K.; JAIDHAR, C. D. A performance evaluation of location prediction position-based routing using real gps traces for vanet. *Wireless Personal Communications*, Springer New York LLC, v. 102, p. 275–292, 9 2018. ISSN 1572834X. Disponível em: <https://link.springer.com/article/10.1007/s11277-018-5839-6>. 19

20 ARABI, A. A. et al. Autonomous rover navigation using gps based path planning. *AMS 2017 - Asia Modelling Symposium 2017 and 11th International Conference on Mathematical Modelling and Computer Simulation*, Institute of Electrical and Electronics Engineers Inc., p. 89–94, 8 2018. 19

21 REN, K. et al. The security of autonomous driving: Threats, defenses, and future directions. *Proceedings of the IEEE*, Institute of Electrical and Electronics Engineers Inc., v. 108, p. 357–372, 2 2020. ISSN 15582256. 20

22 FERREIRA, R. et al. Effective gps jamming techniques for uavs using low-cost sdr platforms. *6th Global Wireless Summit, GWS 2018*, Institute of Electrical and Electronics Engineers Inc., p. 27–32, 7 2018. 20

23 HU, C.; CHEN, Y.; WANG, J. Fuzzy observer-based transitional path-tracking control for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, Institute of Electrical and Electronics Engineers Inc., v. 22, p. 3078–3088, 5 2021. ISSN 15580016. 20

24 PEAVY, M. et al. Integration of real-time semantic building map updating with adaptive monte carlo localization (amcl) for robust indoor mobile robot localization. *Applied Sciences*, MDPI AG, v. 13, p. 909, 1 2023. ISSN 20763417. 20, 21, 30, 31

25 PATLE, B. K. et al. A review: On path planning strategies for navigation of mobile robot. *Defence Technology*, Elsevier, v. 15, p. 582–606, 8 2019. ISSN 2214-9147. 20, 21, 27

26 NISHIKAWA-PACHER, A. Research questions with pico: A universal mnemonic. *Publications*, v. 10, n. 3, 2022. ISSN 2304-6775. Disponível em: <https://www.mdpi.com/2304-6775/10/3/21>. 21

27 ORTIZ, F. M. et al. Vehicle telematics via exteroceptive sensors: A survey. p. 18, 08 2020. 24

28 ORT, T.; GILITSCHENSKI, I.; RUS, D. Autonomous navigation in inclement weather based on a localizing ground penetrating radar. *IEEE Robotics and Automation Letters*, PP, p. 1–1, 02 2020. 24

29 CHIANG, K.-W. et al. The performance analysis of ins/gnss/v-slam integration scheme using smartphone sensors for land vehicle navigation applications in gnss-challenging environments. *Remote Sensing*, v. 12, p. 1732, 05 2020. 24

30 VIVET, D. Extracting proprioceptive information by analyzing rotating range sensors induced distortion. In: . [S.l.: s.n.], 2019. 24

31 KAZEROUNI, A. et al. *An Intelligent Modular Real-Time Vision-Based System for Environment Perception*. 2022. 25, 35

32 AHMED, M. I. B. et al. A real-time computer vision based approach to detection and classification of traffic incidents. *Big Data and Cognitive Computing*, v. 7, p. 22, 1 2023. ISSN 2504-2289. Disponível em: <https://www.mdpi.com/2504-2289/7/1/22>. 25, 35

33 HANDWIKI. Feature detection (computer vision) | encyclopedia mdpi. 2022. Disponível em: <https://encyclopedia.pub/entry/28619>. 25

34 IFTIKHAR, S. et al. Deep learning-based pedestrian detection in autonomous vehicles: Substantial issues and challenges. *Electronics (Switzerland)*, MDPI, v. 11, 11 2022. ISSN 20799292. 25, 35, 36

35 GUERRIERI, M.; PARLA, G. Deep learning and yolov3 systems for automatic traffic data measurement by moving car observer technique. *Infrastructures*, MDPI, v. 6, 9 2021. ISSN 24123811. 25, 35

36 ZEPEDA, A. R.; DUKE, A. M. R.; CASTRO, R. C. Applied computer vision on advanced driving systems. In: . [S.l.]: Latin American and Caribbean Consortium of Engineering Institutions, 2022. v. 2022-July. ISBN 9786289520705. ISSN 24146390. 25, 35

37 BLUE, S. T.; BRINDHA, M. Edge detection based boundary box construction algorithm for improving the precision of object detection in yolov3. *2019 10th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2019*, Institute of Electrical and Electronics Engineers Inc., 7 2019. 26, 42

38 EMMI, L. et al. A hybrid representation of the environment to improve autonomous navigation of mobile robots in agriculture. *Precision Agriculture*, Springer US, v. 22, p. 524–549, 2021. ISSN 15731618. Disponível em: <https://doi.org/10.1007/s11119-020-09773-9>. 26

39 ZHANG, K. et al. Stratified control strategy of vehicle longitudinal active collision avoidance. *Journal of Physics: Conference Series*, v. 1735, p. 14–17, 2021. ISSN 17426596. 26

40 ORTIZ, F. M. et al. Vehicle telematics via exteroceptive sensors: A survey. p. 1–18, 2020. Disponível em: <http://arxiv.org/abs/2008.12632>. 26

41 LE, A. V. et al. Coverage path planning using reinforcement learning-based tsp for htetran — a polyabolo-inspired self-reconfigurable tiling robot. *Sensors*, v. 21, 2021. ISSN 14248220. 26

42 KIM, M. S.; HAN, J.; LEE, J. Optimal trajectory control for capturing a mobile sound source by a mobile robot. *https://doi.org/10.1142/S0219843617500256*, World Scientific Publishing Company, v. 14, 11 2017. ISSN 02198436. 26

43 CHEN, Q. et al. From topological map to local cognitive map: a new opportunity of local path planning. *Intelligent Service Robotics*, Springer Berlin Heidelberg, v. 14, p. 285–301, 2021. ISSN 18612784. Disponível em: <https://doi.org/10.1007/s11370-021-00352-z>. 26

44 WANG, C. et al. Efficient autonomous exploration with incrementally built topological map in 3-d environments. *IEEE Transactions on Instrumentation and Measurement*, v. 69, p. 9853–9865, 2020. ISSN 15579662. 26, 27

45 SILVA, S. Pires Pinheiro da et al. A new approach to navigation of unmanned aerial vehicle using deep transfer learning. p. 222–227, 10 2019. 26

46 MONGUS, D.; JURIC, S. Generation of traversability maps based on 3d point-clouds. *2nd International Conference on Next Generation Computing Applications 2019, NextComp 2019 - Proceedings*, 2019. 26

47 LUO, R. C.; SHIH, W. Topological map generation for intrinsic visual navigation of an intelligent service robot. *2019 IEEE International Conference on Consumer Electronics, ICCE 2019*, IEEE, p. 1–6, 2019. 26

48 OLEYNIKOVA, H. et al. Sparse 3d topological graphs for micro-aerial vehicle planning. *IEEE International Conference on Intelligent Robots and Systems*, p. 8478–8485, 2018. ISSN 21530866. 26, 27

49 ZHOU, R. et al. Genetic algorithm-based challenging scenarios generation for autonomous vehicle testing. *IEEE Journal of Radio Frequency Identification*, Institute of Electrical and Electronics Engineers Inc., v. 6, p. 928–933, 2022. ISSN 24697281. 27, 28

50 ANJALI, A.; AMRITHA, S. A comparative study of ant colony optimization and genetic algorithm for path planning of aerial food delivery system. *Proceedings of the 2022 3rd International Conference on Intelligent Computing, Instrumentation and Control Technologies: Computational Intelligence for Smart Systems, ICICICT 2022*, Institute of Electrical and Electronics Engineers Inc., p. 580–585, 2022. 27, 28

51 SUPRAPTO, B. Y. et al. Position control system on autonomous vehicle movement using fuzzy logic methods. Institute of Electrical and Electronics Engineers (IEEE), p. 744–749, 3 2023. 28

52 SARKAR, S. et al. Comparison between deterministic and deep neural network based real-time trajectory prediction of an autonomous surface vehicle. *Oceans Conference Record (IEEE)*, Institute of Electrical and Electronics Engineers Inc., v. 2022-October, 2022. ISSN 01977385. 28

53 ZHOU, W.; ZHU, D.; YAN, X. The tracking control of autonomous underwater vehicle based on fa - model predictive control. *Proceedings - 2021 2nd International Conference on Artificial Intelligence and Computer Engineering, ICAICE 2021*, Institute of Electrical and Electronics Engineers Inc., p. 435–439, 2021. 28

54 ZHANG, W.; ZHANG, W. Efficient uav localization based on modified particle swarm optimization. *2022 IEEE International Conference on Communications Workshops, ICC Workshops 2022*, Institute of Electrical and Electronics Engineers Inc., p. 1089–1094, 2022. 28

55 LI, Z.; CHEN, K.; WANG, J. An adaptive environmental sampling path planning technique for autonomous underwater vehicles based on ant colony optimization algorithm. *Proceedings of 2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology, ICCASIT 2022*, Institute of Electrical and Electronics Engineers Inc., p. 727–730, 2022. 28

56 LIU, X. et al. Parameters optimization and adaptive control for uav by quantum bacteria foraging algorithms and local refinement strategy. *Chinese Control Conference, CCC*, IEEE Computer Society, v. 2022-July, p. 2451–2458, 2022. ISSN 21612927. 28

57 LIN, S. et al. Improved artificial bee colony algorithm based on multi-strategy synthesis for uav path planning. *IEEE Access*, Institute of Electrical and Electronics Engineers Inc., v. 10, p. 119269–119282, 2022. ISSN 21693536. 28

58 LI, X.; FANG, Y.; FU, W. Uav path planning based on shuffled frog-leaping algorithm and dubins path. *Chinese Control Conference, CCC*, IEEE Computer Society, v. 2020-July, p. 3990–3995, 7 2020. ISSN 21612927. 28

59 NIE, X. et al. An improved deep neural network model of intelligent vehicle dynamics via linear decreasing weight particle swarm and invasive weed optimization algorithms. *Sensors*, v. 22, n. 13, 2022. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/22/13/4676>. 28

60 LIU, L. et al. Harmony search method with global sharing factor based on natural number coding for vehicle routing problem. *Information*, v. 11, n. 2, 2020. ISSN 2078-2489. Disponível em: <https://www.mdpi.com/2078-2489/11/2/86>. 28

61 LEI, T. et al. A bat-pigeon algorithm to crack detection-enabled autonomous vehicle navigation and mapping. *Intelligent Systems with Applications*, v. 12, p. 200053, 2021. ISSN 2667-3053. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2667305321000429>. 28

62 SU, J. li; WANG, H. An improved adaptive differential evolution algorithm for single unmanned aerial vehicle multitasking. *Defence Technology*, v. 17, n. 6, p. 1967–1975, 2021. ISSN 2214-9147. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2214914721001306>. 28

63 SEO, H.; LEE, K.; LEE, K. Investigating the improvement of autonomous vehicle performance through the integration of multi-sensor dynamic mapping techniques. *Sensors*, v. 23, n. 5, 2023. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/23/5/2369>. 29

64 PENDLETON, S. D. et al. Perception, planning, control, and coordination for autonomous vehicles. *Machines*, MDPI AG, v. 5, 3 2017. ISSN 20751702. 29

65 NASHED, S. B.; ILSTRUP, D. M.; BISWAS, J. Localization under topological uncertainty for lane identification of autonomous vehicles. p. 6000–6005, 2018. 30

66 KATWE, S. et al. Particle filter based localization of autonomous vehicle. p. 1–6, 10 2021. 30

67 BEGHDADI, A. et al. Vehicle tracking using projective particle filter. *Faculty of Informatics - Papers*, 09 2009. 30, 31

68 ELFRING, J.; TORTA, E.; MOLENGRAFT, R. van de. Particle filters: A hands-on tutorial. *Sensors*, v. 21, n. 2, 2021. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/21/2/438>. 30

69 ADURTHI, N. Scan-matching based particle filtering approach for lidar-only localization. 02 2023. 30

70 SMIESZEK, M.; DOBRZAńSKA, M. Application of kalman filter in navigation process of automated guided vehicles. *Metrology and Measurement Systems*, v. 22, 09 2015. 30

71 FARAG, W. Kalman-filter-based sensor fusion applied to road-objects detection and tracking for autonomous vehicles. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, v. 235, p. 095965182097552, 12 2020. 30

72 RUGGABER, J.; BREMBECK, J. A novel kalman filter design and analysis method considering observability and dominance properties of measurands applied to vehicle state estimation. *Sensors*, v. 21, n. 14, 2021. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/21/14/4750>. 30

73 JO, H.; KIM, E. New monte carlo localization using deep initialization: A three-dimensional lidar and a camera fusion approach. *IEEE Access*, v. 8, p. 74485–74496, 2020. 30

74 CHEN, R. et al. Deep samplable observation model for global localization and kidnapping. *IEEE Robotics and Automation Letters*, v. 6, n. 2, p. 2296–2303, 2021. 30

75 DEBEUNNE, C.; VIVET, D. A review of visual-lidar fusion based simultaneous localization and mapping. *Sensors (Switzerland)*, MDPI AG, v. 20, 4 2020. ISSN 14248220. 31

76 WOLF, D. F.; SUKHATME, G. S. Localization and mapping in urban environments using mobile robots. *Journal of the Brazilian Computer Society*, Sociedade Brasileira de Computação, v. 13, n. J. Braz. Comp. Soc., 2007 13(4), p. 69–80, Dec 2007. ISSN 0104-6500. Disponível em: <https://doi.org/10.1007/BF03194257>. 31

77 ALATISE, M. B.; HANCKE, G. P. Pose estimation of a mobile robot based on fusion of imu data and vision data using an extended kalman filter. *Sensors (Switzerland)*, MDPI AG, v. 17, 10 2017. ISSN 14248220. 31

78 N., U. S. A. Obstacle avoidance and distance measurement for unmanned aerial vehicles using monocular vision. *International Journal of Electrical and Computer Engineering (IJECE)*, v. 9, p. 3504–3511, 2019. ISSN 2088-8708. 31

79 ZHANG, Z. et al. Study on reconstruction and feature tracking of silicone heart 3d surface. *Sensors*, MDPI, v. 21, 11 2021. ISSN 14248220. 31, 43, 67

80 ZHOU, D. et al. Reasoning graph: A situation-aware framework for cooperating unprotected turns under mixed connected and autonomous traffic environments. *Transportation Research Part C: Emerging Technologies*, Pergamon, v. 143, p. 103815, 10 2022. ISSN 0968-090X. 31

81 LUO, R. C.; SHIH, W. Autonomous mobile robot intrinsic navigation based on visual topological map. *IEEE International Symposium on Industrial Electronics*, Institute of Electrical and Electronics Engineers Inc., v. 2018-June, p. 541–546, 8 2018. 32

82 MA, J.; ZHAO, J. Robust topological navigation via convolutional neural network feature and sharpness measure. *IEEE Access*, Institute of Electrical and Electronics Engineers Inc., v. 5, p. 20707–20715, 9 2017. ISSN 21693536. 32

83 RAVANKAR, A. A. et al. A hybrid topological mapping and navigation method for large area robot mapping. *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan, SICE 2017*, Institute of Electrical and Electronics Engineers Inc., v. 2017-November, p. 1104–1107, 11 2017. 32

84 WIYATNO, R. R.; XU, A.; PAULL, L. Lifelong topological visual navigation. *IEEE Robotics and Automation Letters*, Institute of Electrical and Electronics Engineers Inc., v. 7, p. 9271–9278, 10 2022. ISSN 23773766. 32

85 XU, S.; ZHOU, H.; CHOU, W. Visual topological mapping and navigation for mobile robot in large-scale environment. *IEEE International Conference on Robotics and Biomimetics, ROBIO 2019*, Institute of Electrical and Electronics Engineers Inc., p. 2589–2594, 12 2019. 32

86 GU, Y.; CHIDSIN, W.; GONCHARENKO, I. Ar-based navigation using hybrid map. *LifeTech 2021 - 2021 IEEE 3rd Global Conference on Life Sciences and Technologies*, Institute of Electrical and Electronics Engineers Inc., p. 266–267, 3 2021. 32

87 GUAN, B. F.; LI, S. H.; FU, Q. W. Research on rotation scheme of hybrid inertial navigation system with three rotating axes. *27th Saint Petersburg International Conference on Integrated Navigation Systems, ICINS 2020 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 5 2020. 32

88 NAGATA, S.; MIYAGAWA, I.; MURAKAMI, K. Flexible flight navigation for quadcopters based on geometric formation using motion sensing. *2018 International Workshop on Advanced Image Technology, IWAIT 2018*, Institute of Electrical and Electronics Engineers Inc., p. 1–4, 5 2018. 32

89 VASILYUK, N. N.; TOKAREV, D. K. Identification of geometric displacements of odometers in a gnss inertial navigation system installed on a land vehicle. *27th Saint Petersburg International Conference on Integrated Navigation Systems, ICINS 2020 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 5 2020. 32

90 TSUJIMURA, T.; AOKI, R.; IZUMI, K. Geometrical optics analysis of projected-marker augmented reality system for robot navigation. *Proceedings - 2018 12th France-Japan and 10th Europe-Asia Congress on Mechatronics, Mecatronics 2018*, Institute of Electrical and Electronics Engineers Inc., p. 83–88, 10 2018. 32

91 AHN, D.; CLARIDADES, A. R. C.; LEE, J. Integrating image and network-based topological data through spatial data fusion for indoor location-based services. *Journal of Sensors*, Hindawi Limited, v. 2020, 2020. ISSN 16877268. 32

92 SHALEV, O.; DEGANI, A. Canopy-based monte carlo localization in orchards using top-view imagery. *IEEE Robotics and Automation Letters*, Institute of Electrical and Electronics Engineers Inc., v. 5, p. 2403–2410, 4 2020. ISSN 23773766. 33

93 LIU, M.; SIEGWART, R. Topological mapping and scene recognition with lightweight color descriptors for an omnidirectional camera. *IEEE Transactions on Robotics*, Institute of Electrical and Electronics Engineers Inc., v. 30, p. 310–324, 2014. ISSN 15523098. 33

94 XU, S.; ZHOU, H.; CHOU, W. Erf-imcs: An efficient and robust framework with image-based monte carlo scheme for indoor topological navigation. *Applied Sciences 2020, Vol. 10, Page 6829*, Multidisciplinary Digital Publishing Institute, v. 10, p. 6829, 9 2020. ISSN 2076-3417. Disponível em: <https://www.mdpi.com/2076-3417/10/19/6829/htmhttps://www.mdpi.com/2076-3417/10/19/6829>. 33

95 MARINHO, L. B. et al. A novel mobile robot localization approach based on topological maps using classification with reject option in omnidirectional images. *Expert Systems with Applications*, Elsevier Ltd, v. 72, p. 1–17, 4 2017. ISSN 09574174. 33

96 LIN, K.; WANG, Z. Traffic sign classification by using learning methods: Deep learning and sift based learning algorithm. *2022 IEEE 14th International Conference on Computer Research and Development, ICCRD 2022*, Institute of Electrical and Electronics Engineers Inc., p. 239–243, 2022. 33

97 LINDQVIST, C. Comparing sift and surf performance on patent drawings. 2017. Disponível em: <http://www.teknat.uu.se/student>. 33, 42

98 LIU, J.; WEI, Q.; BAI, Y. Fast stitching of uav images based on improved surf algorithm. *Proceedings of 2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology, ICCASIT 2021*, Institute of Electrical and Electronics Engineers Inc., p. 1310–1313, 2021. 33, 42

99 SHI, Z.; TANG, K.; ZHOU, T. Research on real-time feature extraction algorithm for image processing. *Proceedings - 2021 6th International Symposium on Computer and Information Processing Technology, ISCIPT 2021*, Institute of Electrical and Electronics Engineers Inc., p. 371–376, 2021. 33

100 WANG, N.; KARIMI, H. R. Successive waypoints tracking of an underactuated surface vehicle. *IEEE Transactions on Industrial Informatics*, IEEE Computer Society, v. 16, p. 898–908, 2 2020. ISSN 19410050. 33

101 KIM, S.; HA, J.; JO, K. Semantic point cloud-based adaptive multiple object detection and tracking for autonomous vehicles. *IEEE Access*, Institute of Electrical and Electronics Engineers Inc., v. 9, p. 157550–157562, 2021. ISSN 21693536. 34

102 YI, D. et al. Improving synthetic to realistic semantic segmentation with parallel generative ensembles for autonomous urban driving. *IEEE Transactions on Cognitive and Developmental Systems*, Institute of Electrical and Electronics Engineers Inc., 2021. ISSN 23798939. 34

103 XU, J. et al. Speed bump recognition for autonomous vehicles based on semantic segmentation. *2021 IEEE 3rd International Conference on Communications, Information System and Computer Engineering, CISCE 2021*, Institute of Electrical and Electronics Engineers Inc., p. 387–393, 5 2021. 34

104 BAYHAN, E. et al. Deep learning based object detection and recognition of unmanned aerial vehicles. *HORA 2021 - 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications, Proceedings*, Institute of Electrical and Electronics Engineers Inc., 6 2021. 34

105 LI, Z.; LUO, X. Autonomous underwater vehicles (auvs) path planning based on deep reinforcement learning. *11th International Conference on Intelligent Control and Information Processing, ICICIP 2021*, Institute of Electrical and Electronics Engineers Inc., p. 125–129, 2021. 34

106 TIMMIS, I.; PAUL, N.; CHUNG, C. J. Teaching vehicles to steer themselves with deep learning. *IEEE International Conference on Electro Information Technology*, IEEE Computer Society, v. 2021-May, p. 419–421, 5 2021. ISSN 21540373. 34

107 LI, Y.; SHI, J.; LI, Y. Real-time semantic understanding and segmentation of urban scenes for vehicle visual sensors by optimized dcnn algorithm. *Applied Sciences (Switzerland)*, MDPI, v. 12, 8 2022. ISSN 20763417. 36

108 CAKIR, S. et al. Semantic segmentation for autonomous driving: Model evaluation, dataset generation, perspective comparison, and real-time capability. 7 2022. Disponível em: <http://arxiv.org/abs/2207.12939>. 36

109 AUDEBERT, N.; SAUX, B. L.; LEFèVRE, S. Segment-before-detect: Vehicle detection and classification through semantic segmentation of aerial images. *Remote Sensing*, MDPI AG, v. 9, 4 2017. ISSN 20724292. 36

110 ALFALOU, A. et al. *A comparative study of CFs, LBP, HOG, SIFT, SURF, and BRIEF for security and face recognition.* [S.l.: s.n.], 2018. 36

111 SULTANI, Z. N.; DHANNOON, B. N. Modified bag of visual words model for image classification. *Al-Nahrain Journal of Science*, v. 24, p. 78–86, 6 2021. ISSN 26635453. Disponível em: <https://anjs.edu.iq/index.php/anjs/article/view/2402/1868>. 36

112 LIU, W. et al. *A Review of Image Feature Descriptors in Visual Positioning.* 2021. 36

113 CHIARONI, F. et al. Self-supervised learning for autonomous vehicles perception: A conciliation between analytical and learning methods. 2019. Disponível em: <https://opencv.org/>. 36

114 HUSSAIN, S. et al. A comparative study of supervised machine learning techniques for diagnosing mode of delivery in medical sciences. *International Journal of Advanced Computer Science and Applications*, v. 10, 01 2019. 36

115 WANG, J. et al. Bridge the gap between supervised and unsupervised learning for fine-grained classification. 3 2022. Disponível em: <http://arxiv.org/abs/2203.00441>. 36

116 SALIM, S.; AL-TUWAIJARI, J. Skin disease classification system based on machine learning technique: A survey. *IOP Conference Series: Materials Science and Engineering*, v. 1076, p. 012045, 02 2021. 37

117 ALMEIDA, J. R. D. Transfer learning e convolutional neural networks para a classificação de imagens e reconhecimento de objetos no âmbito da perícia criminal. 2020. Disponível em: <https://repositorio.unb.br/handle/10482/40286>. 37

118 BORA, D. J.; GUPTA, A. K.; KHAN, F. A. International journal of emerging technology and advanced engineering comparing the performance of l*a*b* and hsv color spaces with respect to color image segmentation. *Certified Journal*, v. 9001, 2008. Disponível em: <www.ijetae.com>. 37, 39, 40

119 IBRAHEEM, N. et al. Understanding color models: A review. *ARPN Journal of Science and Technology*, v. 2, 01 2012. 37, 38, 39, 40

120 MINZ, P. S.; SAINI, C. S. Comparison of computer vision system and colour spectrophotometer for colour measurement of mozzarella cheese. *Applied Food Research*, Elsevier, v. 1, p. 100020, 12 2021. ISSN 2772-5022. 38

121 SHIKROT, R.; ESCRIVá, D. M. *Mastering OpenCV 4: A comprehensive guide to building computer vision and ... - Roy Shilkrot, David Millán Escrivá - Google Libros*. Third edition, Packtpub, 2018. ISBN 978-1-78953-357-6. Disponível em: <https://books.google.com.br/books?id=FtCBDwAAQBAJ&pg=PA243&dq=opencv&hl=es-419&sa=X&ved=2ahUKEwiD8af3u4b7AhWYs5UCHa-GABUQ6AF6BAgHEAI#v=onepage&q=opencv&f=false>. 38, 41, 66, 70

122 ABBADI, N. E.; SALEEM, E. Automatic gray images colorization based on lab color space. *Indonesian Journal of Electrical Engineering and Computer Science*, v. 18, p. 1501, 06 2020. 38, 39, 40

123 PIERRE, F.; AUJOL, J.-F. *Recent Approaches for Image Colorization*. [S.l.: s.n.], 2020. 39, 40

124 HISOUR. Cieluv – hisour arte cultura exposição. 2023. 40

125 XU, C. et al. Multiview image matching of optical satellite and uav based on a joint description neural network. *Remote Sensing*, v. 14, n. 4, 2022. ISSN 2072-4292. Disponível em: <https://www.mdpi.com/2072-4292/14/4/838>. 41

126 HU, Z. et al. Vehicle re-identification based on keypoint segmentation of original image. *Applied Intelligence*, v. 53, 05 2022. 41

127 LI, J. et al. Real-time keypoints detection for autonomous recovery of the unmanned ground vehicle. 07 2021. 41

128 SHEN, J. et al. Joint metric learning of local and global features for vehicle re-identification. *Complex Intelligent Systems*, v. 8, 03 2022. 41

129 ZOU, T. et al. Kam-net: Keypoint-aware and keypoint-matching network for vehicle detection from 2d point cloud. *IEEE Transactions on Artificial Intelligence*, PP, p. 1–1, 09 2021. 41

130 GHASEMIEH, A.; KASHEF, R. 3d object detection for autonomous driving: Methods, models, sensors, data, and challenges. *Transportation Engineering*, v. 8, p. 100115, 2022. ISSN 2666-691X. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2666691X22000136>. 41

131 STILLA, U.; XU, Y. Change detection of urban objects using 3d point clouds: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, v. 197, p. 228–255, 2023. ISSN 0924-2716. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0924271623000163>. 41

132 LIU, R. B. J. X. Y. T. Q. L. X. Edge-cooperative privacy-preserving object detection over random point cloud shares for connected autonomous vehicles. 2022. Disponível em: <https://ieeexplore.ieee.org/document/9928424>. 42

133 TANG, S. et al. Vecframe: A vehicular edge computing framework for connected autonomous vehicles. Institute of Electrical and Electronics Engineers (IEEE), p. 68–77, 2 2022. 42

134 HU, Y. et al. Object detection of uav for anti-uav based on improved yolo v3. *Chinese Control Conference, CCC*, IEEE Computer Society, v. 2019-July, p. 8386–8390, 7 2019. ISSN 21612927. 42

135 LIANG, S. et al. Edge yolo: Real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, Institute of Electrical and Electronics Engineers Inc., 2022. ISSN 15580016. 42

136 BRALET, A.; KECHICHIAN, R.; VALETTE, S. Local surf-based keypoint transfer segmentation. *Proceedings - International Symposium on Biomedical Imaging*, IEEE Computer Society, v. 2021-April, p. 1390–1393, 4 2021. ISSN 19458452. 42

137 PEDERSEN, J. T. Surf:feature detection description. p. 1–12, 2011. 42

138 ZHOU, H. et al. Efficient road detection and tracking for unmanned aerial vehicle. *IEEE Transactions on Intelligent Transportation Systems*, Institute of Electrical and Electronics Engineers Inc., v. 16, p. 297–309, 2 2015. ISSN 15249050. 42

139 AGUILAR, W. G.; CASALIGLLA, V. P.; PóLIT, J. L. Obstacle avoidance based-visual navigation for micro aerial vehicles. *Electronics (Switzerland)*, MDPI AG, v. 6, 3 2017. ISSN 20799292. 43

140 KAUR, J.; BATHLA, A. K. Video stabilization for an aerial surveillance system using sift and surf. *Proceedings on 2016 2nd International Conference on Next Generation Computing Technologies, NGCT 2016*, Institute of Electrical and Electronics Engineers Inc., p. 742–747, 3 2017. 43

141 LIU Y., T. J. H. R. Y. B. L. S. Y. L. . Z. W. Improved feature point pair purification algorithm based on sift during endoscope image stitching. *Frontiers in Neurorobotics*, p. 840594, 16 2022. 43

142 HIDALGO, F.; BRäUNL, T. Evaluation of several feature detectors/extractors on underwater images towards vslam. *Sensors (Switzerland)*, MDPI AG, v. 20, p. 1–16, 8 2020. ISSN 14248220. 43

143 CHOI, J. et al. Development of an autonomous surface vehicle and performance evaluation of autonomous navigation technologies. *International Journal of Control, Automation and Systems*, v. 18, p. 535–545, 2020. ISSN 20054092. 43

144 PUI, S. T.; MINOI, J. L. Keypoint descriptors in sift and surf for face feature extractions. In: . [S.l.]: Springer Verlag, 2018. v. 488, p. 64–73. ISBN 9789811082757. ISSN 18761119. 43, 49, 74, 80

145 YABO, A. et al. *VEHICLE CLASSIFICATION AND SPEED ESTIMATION USING COMPUTER VISION TECHNIQUES*. 2016. 43

146 HUANG, S.; SUN, G.; LI, M. Fast and flann for feature matching based on surf. p. 1584–1589, 2021. 43

147 NALCACI, G.; YILDIRIM, D.; ERMIS, M. Selective harmonic elimination for light-rail transportation motor drives using harris hawks algorithm. p. 1–6, 2020. 43

148 LI, T. et al. Scale-invariant localization of electric vehicle charging port via semi-global matching of binocular images. *Applied Sciences (Switzerland)*, MDPI, v. 12, 5 2022. ISSN 20763417. 43

149 BAY, H.; TUYTELAARS, T.; GOOL, L. V. Surf: Speeded up robust features. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer, Berlin, Heidelberg, v. 3951 LNCS, p. 404–417, 2006. ISSN 03029743. Disponível em: <https://link.springer.com/chapter/10.1007/11744023_32>. 44, 48

150 DAS, J.; SHAH, M.; MARY, L. Bag of feature approach for vehicle classification in heterogeneous traffic. *2017 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems, SPICES 2017*, Institute of Electrical and Electronics Engineers Inc., 10 2017. 44

151 HSIEH, J. W.; CHEN, L. C.; CHEN, D. Y. Symmetrical surf and its applications to vehicle detection and vehicle make and model recognition. *IEEE Transactions on Intelligent Transportation Systems*, v. 15, p. 6–20, 2 2014. ISSN 15249050. 44

152 TEDJOJUWONO, S. M. Vehicle recognition systems using speed-up robust features and non-maxima suppression. *Proceedings of 2019 International Conference on Information Management and Technology, ICIMTech 2019*, Institute of Electrical and Electronics Engineers Inc., p. 42–47, 8 2019. 44

153 GONZALEZ, R. C.; WOODS, R. E.; MASTERS, B. R. Digital image processing, third edition. *Journal of Biomedical Optics*, v. 14, p. 029901, 2009. ISSN 1083-3668. Disponível em: <https://www.academia.edu/33225191/Digital_Image_Processing_Third_Edition>. 45

154 WANG, J.; WATADA, J. Panoramic image mosaic based on surf algorithm using opencv. *WISP 2015 - IEEE International Symposium on Intelligent Signal Processing, Proceedings*, 06 2015. 49, 70, 74, 80

155 BOUZIADY, A. E. et al. Vehicle speed estimation using extracted surf features from stereo images. *2018 International Conference on Intelligent Systems and Computer Vision, ISCV 2018*, Institute of Electrical and Electronics Engineers Inc., v. 2018-May, p. 1–6, 5 2018. 49, 53

156 LEE, E. S.; KUM, D. Feature-based lateral position estimation of surrounding vehicles using stereo vision. *IEEE Intelligent Vehicles Symposium, Proceedings*, Institute of Electrical and Electronics Engineers Inc., p. 779–784, 7 2017. 53

157 BOSTANCI, E. et al. A fuzzy brute force matching method for binary image features. *arXiv preprint arXiv:1704.06018*, 2017. 54

158 HADISUKMANA, N.; YUDIANTO, A. Paper money recognizer using feature descriptor. *Indonesian Journal of Electrical Engineering and Computer Science*, v. 12, 10 2018. 54, 70

159 JAKUBOVIC, A.; VELAGIC, J. Image feature matching and object detection using brute-force matchers. p. 83–86, 09 2018. 54, 55, 71

160 MAIWALD, F. et al. Feature matching of historical images based on geometry of quadrilaterals. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2, p. 643–650, 05 2018. 54

161 FLORES, M. et al. Efficient probability-oriented feature matching using wide field-of-view imaging. *Engineering Applications of Artificial Intelligence*, v. 107, p. 104539, 2022. ISSN 0952-1976. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0952197621003870>. 55

162 HEMA, M. D.; KANNAN, S. Object detection and learning via feature descriptors. 2021. 55

163 RUI-JUAN, T. Surf algorithm and its detection effect on object tracking. *Journal Of Southwest University Of Science And Technology*, 2011. 55

164 XIN, M.; LI, S. W.; ZHANG, M. H. Robust object tracking by particle filter with scale invariant features. *Appl. Mech. Mater.*, v. 151, p. 458–462, 2012. 55

165 AN, M.-S. et al. Robust multi-object tracker based on feature point particle filter algorithm. *Proceedings Of KIIT Summer Conference*, 2011. 55

166 ZHU, M. et al. Monocular 3d vehicle detection using uncalibrated traffic cameras through homography. 3 2021. Disponível em: <http://arxiv.org/abs/2103.15293>. 55

167 TOURANI, A. et al. Motion-based vehicle speed measurement for intelligent transportation systems. *International Journal of Image, Graphics and Signal Processing*, MECS Publisher, v. 11, p. 42–54, 4 2019. ISSN 20749074. 55

168 ALEPHNULL. Utm/ wgs84 conversion in c++. 2022. Disponível em: <https://alephnull.net/software/gis/UTM_WGS84_C_plus_plus.shtml>. 56

169 NENA. Latitude and longitude – utm conversion | nenadsprojects. 2016. Disponível em: <https://nenadsprojects.wordpress.com/2012/12/27/latitude-and-longitude-utm-conversion/>. 56

170 GREENWADE, G. D. The Comprehensive Tex Archive Network (CTAN). *TUGBoat*, v. 14, n. 3, p. 342–351, 1993. 61

171 GAO, Z. et al. Stereo camera calibration for large field of view digital image correlation using zoom lens. *Measurement*, Elsevier, v. 185, p. 109999, 11 2021. ISSN 0263-2241. 66

172 BURGER, W. Zhang's camera calibration algorithm: In-depth tutorial and implementation. 16 2016. Disponível em: <https://www.researchgate.net/publication/303233579_Zhang's_Camera_Calibration_Algorithm_In-Depth_Tutorial_and_Implementation>. 66

173 ZHANG, X. Comprensión simple de la máscara rcnn | de xiang zhang | medio. 2018. Disponível em: <https://alittlepain833.medium.com/simple-understanding-of-mask-rcnn-134b5b330e95>. 67

174 RARES, J. F. I. V. V. G.; WALTER, G. R. S. A. G. M. Self-supervised camera self-calibration from video. 2022. Disponível em: <https://sites.google.com/ttic.> 67

175 NOVAIS, J. P. *Aplicação dos Algoritmos SIFT e SURF na Classificação de Sub-Imagens por Discriminação de Textura*. [s.n.], 2016. Disponível em: <https://www.researchgate.net/publication/343797480_Aplicacao_dos_Algoritmos_SIFT_e_SURF_na_Classificacao_de_Sub-Imagens_por_Discriminacao_de_Textura>. 69

176 SILVA, A.; HEINEN, M.; JURASZEK, G. Identificação de produtos por imagem utilizando o algoritmo surf um comparativo entre redes perceptron multicamadas e máquinas de vetor de suporte. In: . [s.n.], 2013. Disponível em: <https://www.researchgate.net/publication/275959295_Identificacao_de_Produtos_por_Imagem_Utilizando_o_Algoritmo_SURF_Um_Comparativo_Entre_Redes_Perceptron_Multicamadas_e_Maquinas_de_Vetor_de_Suporte>. 69

177 MADAN, R. et al. Traffic sign classification using hybrid hog-surf features and convolutional neural networks. In: . [S.l.]: SciTePress, 2019. p. 613–620. ISBN 9789897583513. 70

178 POP, C.; MOGAN, G.-L.; BOBOC, R. Real-time object detection and recognition system using opencv via surf algorithm in emgu cv for robotic handling in libraries. *International Journal of Modeling and Optimization*, v. 7, p. 265–269, 10 2017. 70, 80

179 HOWSE, J.; MINICHINO, J.; JOSHI, P. *Learning OpenCV 4 Computer Vision with Python 3: Get to Grips with Tools, Techniques, and Algorithms for Computer Vision and Machine Learning, 3rd Edition*. [S.l.]: Packt Publishing, 2019. 71

180 KARAMI, E.; PRASAD, S.; SHEHATA, M. *Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images*. 2017. 74

Anexos

ANEXO A – Artigo publicado e certificado de congresso

Congresso Brasileiro de Automática

2022

FORTALEZA-CE

CERTIFICADO DE PARTICIPAÇÃO

Certificamos que **Natalia Sánchez Sánchez** participou do XXIV Congresso Brasileiro de Automática - CBA 2022, realizado de 16 a 19 de outubro de 2022, na cidade de Fortaleza-CE.

FABRÍCIO GONZALEZ NOGUEIRA
Coordenador Geral

BISMARK CLAURE TORRICO
Coordenador Geral

COORDENAÇÃO

REALIZAÇÃO

ORGANIZAÇÃO

PATROCÍNIO

APOIO INSTITUCIONAL

AG. DE TURISMO

Congresso Brasileiro de Automática

2022

FORTALEZA-CE

CERTIFICADO DE APRESENTAÇÃO

Certificamos que **Natalia Sanchez Sanchez** apresentou o trabalho intitulado **"Digital Twins: A 3D simulation approach to test validation with an autonomous vehicle"** no XXIV Congresso Brasileiro de Automática - CBA 2022, realizado em Fortaleza-CE, de 16 a 19 de outubro de 2022.

FABRÍCIO GONZALEZ NOGUEIRA
Coordenador Geral

BISMARK CLAURE TORRICO
Coordenador Geral

COORDENAÇÃO

REALIZAÇÃO

ORGANIZAÇÃO

PATROCÍNIO

APOIO INSTITUCIONAL

AG. DE TURISMO

Digital Twins: A 3D simulation approach to test validation with an autonomous vehicle

Nalia Sánchez * **Marcos V. Cruz** * **Rafael F. Santos** *
Rubén Hernandez ** **Willian G. Almeida** * **Giovani Bernardes** *

* *Federal University of Itajuba, Institute of technological Sciences - ICT/UNIFEI, Laboratory of Robotics, intelligent and Complex Systems - RobSic, 35903-087 Itabira-MG*
** *Universidad Militar Nueva Granada, Colombia*

Abstract:
Within the Digital Twins context, efforts are being made to minimize costs, increase safety, and speed up tests within a specific application, based on computer graphics tools for three-dimensional simulations. Thus, as a way to mitigate mainly the risks with the safety issue involving activities with autonomous vehicles, the present work proposes the modeling and structuring of an electric golf cart in a 3D virtual environment, so that it can serve for studies in the areas of perception, navigation, and control. Therefore, taking as a reference the proper vehicle existing at the university, Blender software was used together with the Gazebo simulator to perform the validation of this simulation environment. Different open source Environment Mapping (SLAM), Pattern Recognition (YOLO), and Autonomous Navigation algorithms were integrated to minimize possible errors regarding their integration in the actual vehicle. Finally, validation tests were performed on the Yolo algorithm, resulting in an accuracy of 98.9% with a margin of error of 1.1% in the identification of objects.
Resumo:
Dentro do contexto de *Digital Twins*, esforços estão sendo realizados para minimizar os custos, aumentar a segurança e agilizar ensaios dentro de uma aplicação específica, fundamentados em ferramentas de computação gráfica para simulações tridimensionais. Sendo assim, como forma de mitigar principalmente riscos com a questão da segurança envolvendo atividades com veículos autônomos, o presente trabalho propõe a realização da modelagem e estruturação de um carro de golfe elétrico em ambiente virtual 3D, de maneira que este possa servir de estudos nas áreas de percepção, navegação e controle. Portanto, tomando como referência o veículo real existente na universidade, foi utilizado o software Blender juntamente com o simulador GAZEBO para que, não somente o veículo, mais também os principais pontos do ambiente por onde o veículo irá ser ensaiados no cenário real fossem incluídos e representados no ambiente de simulação 3D. Para validação deste ambiente de simulação, diferentes algoritmos de código aberto para Mapeamento de Ambiente (SLAM), Reconhecimento de Padrões (YOLO) e Navegação Autônoma foram integrados afim de minimizar possíveis erros quanto sua integração no veículo real. Por fim, foi realizado testes de validação no algoritmo Yolo, obtendo como resultado uma precisão de 98,9% com uma margem de erro de 1,1% na identificação de objetos.

Keywords: Digital Twins, simulation, 3D modeling, autonomous Navigation, object detection.
Palavras-chaves: Gêmeos Digitais, simulação, modelagem 3D, navegação autônoma, detecção de objetos.

1. INTRODUCTION

The implementation of low-cost technologies in software and hardware has given rise to many new automation and autonomy solutions. Since, their application in the industry increases productivity and improves different processes, for example, IoT, AI, Industry 4.0, or Big Data (Redeker et al. (2021); Faz-Mendoza et al. (2020)); demonstrates additional benefits in the use of resources and improved productivity.

Given this, when undertaking a project, it is necessary to carry out a series of tests, not only to start it up but also to periodically reevaluate it. Consequently, there are simulation tools such as the software CoppeliaSim, Webots, Gazebo, OpenRave, and RodoDK, among others (Jakubiec (2018)), which provide a realistic representation of the physical environment of the virtual environment. Given this, there are great challenges today, regarding the final design, since, if you know how to implement these tools, can reduce time, and costs, optimize the manufacture of the product and even allow the identification of problems and errors that could occur in real life.

These simulation tools are currently being used for the realization of autonomous vehicles. However, this is a topic

that requires a lot of research, since the implementation of a well-designed simulator will allow us to quickly test algorithms, design collisions, and even perform tests of different real-world scenarios.

Gazebo is a simulator that provides realistic sensor information and interactions between potentially plausible objects, including an accurate simulation of rigid body physics. This simulator can be integrated with ROS (Robot Operating System), which provides necessary interfaces to develop robots, environments, and development of intelligent algorithms for planning, navigation, and testing of the environment with specific applications in various areas: hospital, industrial, business, etc (Xu et al. (2021); Sai Sahith Velamala (2017); Hussein et al. (2018); Marian et al. (2020); Mario Gluhaković (2020); Rivera et al. (2019); Quang et al. (2019)).

In the literature, different authors have used such tools. Banjanovic-Mehmedovic et al. (2021b) used ROS, Gazebo, and Rviz in an application where the robot navigates in an environment that is responsible for solving problems related to disinfection in a Covid-19 contaminated environment. On the other hand, Ahmed et al. (2019) performed an application in which they perform activities related to industrial maintenance considered high risk and difficult to access, subject to chemicals, hazardous substances, or biological substances, used in an unmanned aerial vehicle. Also, Arango et al. (2020) presents the development process of the ROS-based Drive-By-Wire system, designed for an open-source autonomous electric vehicle prototype, which implements a manual/automatic interchange system, allowing the driver to activate autonomous driving and safely take control of the vehicle at any time. This and other authors (Villa et al. (2020); Tian et al. (2021); Yang and Chi (2021)) have implemented these new technologies utilizing simulations to ensure better control of the project. In addition, allows tasks to be carried out in efficient ways in terms of latency such as energy consumption, among others.

However, when working with autonomous vehicles, this type of driving without human intervention requires continuous processing of the images captured by the vehicle during its journey, to accurately determine the path to follow, as well as the possible presence of obstacles along the way. Given the large amount of data to be processed and the essential response of the vehicle in real-time, it is essential to explore new architectures to perform these tasks efficiently both in terms of latency and energy consumption. For this reason, this work proposes to implement an initial phase that allows simulations describing the interaction of an autonomous vehicle with the study environment, whose models are focused on scenarios of the Federal University of Itabira (UNIFEI), where the autonomous vehicle is being designed and where real experiments will be reproduced. Based on the above, this research contributes to the following results:

- Blender modeling of an environment that includes the structure of the Unifei campus and the vehicle (golf cart) of which it is in the process of integrating sensors that will be used in simulations in a real environment.
- They carried validation of the Yolo algorithm out through the calculation of a confusion matrix, which allows demonstrating the performance of the algorithm with its respective margin of error. Likewise, the integration with the RQT application was performed, which evidences the open-source object detection method and the Ros integration.
- It generated a map from the integration of the SLAM cartographer that provides simultaneous real-time location and mapping on the 2D plane of the University. In addition, they carried integration out "Nav2" package (Stack (2020)) which results in the desired route in the virtual environment, integrating a particle filter and the localization algorithm (AMCL, adaptive localization Monte Carlo).

This article is divided into the construction of the robotic system and the simulation environment, application implementation, results, and conclusions. The system architecture, modeling and environment configuration are shown as a ROS-based solution that can be transferred to different simulation platforms from which they implement autonomous navigation, perception, and 2D mapping.

2. 3D MODELLING AND STRUCTURING OF THE ROBOTIC PLATFORM

2.1 Context of development

The figure 1 presents the background of the development of this work, divided into two stages. The first stage includes the design of the Unifei University and the golf cart, which were designed in Blender software, allowing the modeling, rendering, and creation of simulations, which in this case allows the integration with Gazebo, which allows the second stage. The second stage includes the integration of the model to Gazebo, which allows the transfer of information about the interaction of the vehicle to the physical environment, it also allows the integration of different applications, in this case, three applications are implemented, which serve to validate the design of the car and the environment. From these two stages, they expected to obtain results of the mapping, navigation, visualization, and pattern recognition, which are intended to be implemented in the real golf cart.

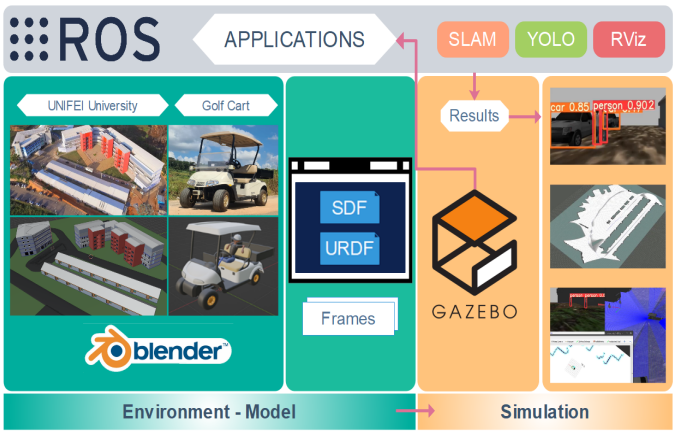

Figure 1. Phases for the implementation of a simulated autonomous vehicle.

The integration through Gazebo uses the URDF and SDF files. These formats are files are formats established to describe the robot structure in the RViz visualization software and the Gazebo physical simulation software. Both files satisfy the simulation needs in a Gazebo environment, resulting in the implementation of SLAM (navigation) and Yolo (sensing). In this way, these applications can be implemented in a real environment from which the simulations generated in the software will be validated, they describe each stage below:

2.2 Environment

To create the simulation environment, we started with two steps. First, modeling the Unifei Itabira campus (Figure 2). Second, modeling the golf cart, mentioned earlier. Both models were modeled in Blender, the university was modeled using real images of the environment. Similarly, the golf cart (model EZGO RXV) was modeled using the images and measurements of the actual vehicle present at the university.

The modeling of the environment and vehicle through the Blender software allows these designs to be saved in different extensions such as .obj, .stl, .fbx, among others. Such files allow the integration of the models (environment and vehicle) in the Gazebo platform that will be used as visualization and collision components.

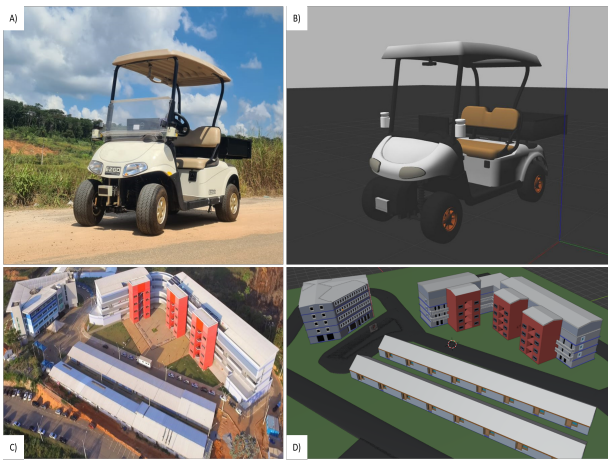

Figure 2. Real environment and physical environment. A) golf cart, B) golf cart modeling, C) Unifei University where the project is being carried out, and D) modeling of the university environment.

2.3 Modeling

The construction of the physical model of a robot starts from the SDF and URDF files. Both files contain information about the fixed and moving parts that make up the robot, as well as the connections between them and the sensing that is defined through the concepts of links, joints, and plugins that correspond to the robot members, joints, and sensor integration, respectively.

The URDF (Unified Robot Description Format) file is obtained from the transformation frames of each member of the vehicle, ie, the position of the base of the vehicle about lasers, camera, IMU, etc. Figure 3 shows the golf cart in the Gazebo environment, where the integration of different sensors was carried out and contains the visual representation of the concepts of the physical model of the vehicle and the transformations between the frames generated in the RViz software.

In addition, once the SDF and URDF files are generated, both are executed in the Gazebo and Rviz environments. In Figure 3, the vehicle is in the Gazebo environment with all the sensing specifications that were defined in the real vehicle. As shown the vehicle has two 1-layer lasers (located in the front region of the vehicle) and two 4-layer lasers (located in the front region near the headlight and the rear region under the body of the vehicle), the two 1-layer lasers perform a scanning whose angle involves an aperture from 0 to 270°. And the other two 4-layer lasers scan from an angle of 50° to -60°. The other sensors are for orientation (IMU and GPS), stereo camera, and encoders.

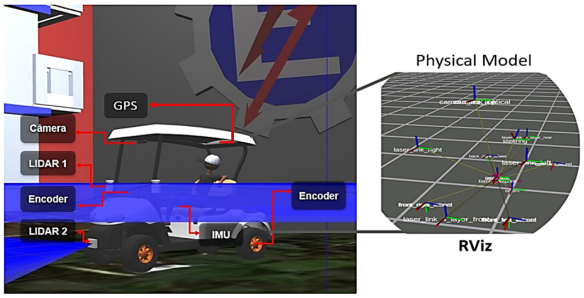

Figure 3. Integration of sensors and visual part of the physical model of the vehicle.

Another way to represent and analyze the transformation frame and the relationship between the vehicle members is through the transformation tree link [1] which has a graphical representation whose aspect is a tree of nodes (frames) that obeys a hierarchy of which establishes the relationship between each part of the vehicle, for example, the relationship between the base of the vehicle (base_link), camera (camera-link) and sensors (imu link, camera link).

2.4 Simulation (environment configuration)

The simulations involve applications focused on mapping, navigation, and pattern recognition. The simultaneous localization and mapping (SLAM) in the simulation environment is done using the vehicle's lasers and odometry sensors. As the vehicle moves through the environment and detects its surroundings, the region detected in its surroundings is recorded and this information contributes to the navigation of the vehicle in the environment. The same is true for pattern recognition (YOLO) via the stereo camera.

3. DEPLOYED APPLICATIONS

When obtaining the simulation model, different applications that can be implemented in this system were found, which are the following:

[1] https://drive.google.com/file/d/1pjIkLkda203lOi2utAYFF5zzxFxgQPQo/view

3.1 Simultaneous localization and mapping (SLAM)

To create the map of the environment, a package called SLAM cartographer developed by Google was used. The generation of the 2D map is based on the data generated by lidar sensors, odometry, and IMU, whose sensors are necessary for the implementation of this package, therefore first the vehicle was modeled and then this algorithm was used. SLAM refers to the problem of constructing a map of an unknown environment by a mobile robot, while allowing the use of the map for localization and navigation in the environment (Banjanovic-Mehmedovic et al. (2021a)). The main function of SLAM is to analyze the input data to determine the position of the vehicle and build a map of the environment where it can move autonomously.

SLAM, on the other hand, allows the marking of trajectories, which can be developed from different filtering or smoothing methods, such as Kalman filters (EKF, UKF) and particle filters (Xuexi et al. (2019); Valencia et al. (2011a,b); Burger et al. (2019)). In this case, they divided the implementation of the mapping package into two types: local SLAM and global SLAM. The local SLAM consists of the construction of a time-varying sequence of local submaps, and the global SLAM allows for finding closed-loop constraints, which are demonstrated when a car has to pass a traffic circle, generating a map containing information about the surrounding environment and its obstacles (Figure 5).

3.2 Detection Yolo (You Only Look Once)

For autonomous vehicles, perception is necessary since it is responsible for analyzing the environment surrounding the vehicle, detecting, and recognizing the objects in it, for the acquisition of information, this is usually implemented through various types of sensors, such as cameras, LIDAR, radar, ultrasonic devices, etc. The camera is a sensor that can provide more detailed information about the vehicle environment in terms of high resolution and texture information. Given this, algorithms were created that allow the detection of obstacles and objects in the environment, allowing to perform better control of the environment information.

Similarly, object detection is an advanced form of image classification in which a neural network predicts objects in an image and points to them in the form of bounding boxes (Bjelonic (2016–2018)), i.e., it can perform generalized detection, recognition, or localization tasks in real-world scenarios. There is an algorithm called Yolo (You Only Look Once) that uses classifiers to perform detection using end-to-end neural networks that make bounding box predictions and class probabilities at the same time, it achieves state-of-the-art results outperforming other real-time objects detection algorithms by a wide margin (Corović et al. (2018); Bjelonic (2016–2018)). From this, the integration of this algorithm with Ros allows performing a simulation of the vehicle, to detect obstacles encountered on the road.

This project uses YoloV5 which is previously trained with the COCO dataset, which will identify objects such as cars, people, and trucks, among others. In addition, a ZED stereoscopic camera is simulated which generates the real-time position of the camera and allows to perform environment detection. Figure 4 shows the process followed for the implementation of this algorithm, which is divided into three parts, which are part of the image processing. In the capture stage, the visual image of the environment is obtained to perform a preprocessing that includes noise reduction techniques and detail enhancement. In this case, the segmentation that divides the image according to the objects to be detected, people and cars, is also implemented. After this, comes the description and detection part, which is the process where the characteristics of the image such as size and shape are obtained to determine the recognition and implement it in the graphic part of Gazebo.

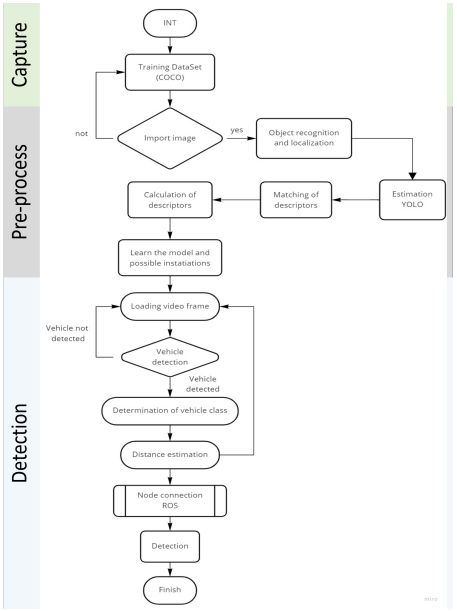

Figure 4. Algorithm implemented for object detection using the Yolo implementation.

4. RESULTS

As a result of the implementation of the simulation, they carried out two validation tests, where the environment model is static, and they generate the robot model in this environment to navigate. They explain these tests below.

4.1 Map and trajectory creation

By navigating the environment, the map shown in Figure 5 was generated. The contour includes the 2D profile of the environment. The white region is the occupancy of the modeled environment (university) in space and the black contours indicate the boundaries of the environment. As the vehicle moves through the university, the map measurements become more accurate through optimization algorithms that assemble sub-maps and cluster into a global map. In addition, during vehicle navigation it was necessary to choose a trajectory (blue curve in Figure 7), speed (from 0 to 5 m/s) and laser data publication rate (30Hz) that could provide a global map with the best assembly of all submaps. Xuexi et al. (2019) demonstrated in practical experiments that map generation is influenced by the velocity of the robot in the environment, the laser data publication rate and the trajectory. The proper choice

of the trajectory that provides the best loop is also responsible for the quality and clustering of the submap maps that make up the overall map generation.

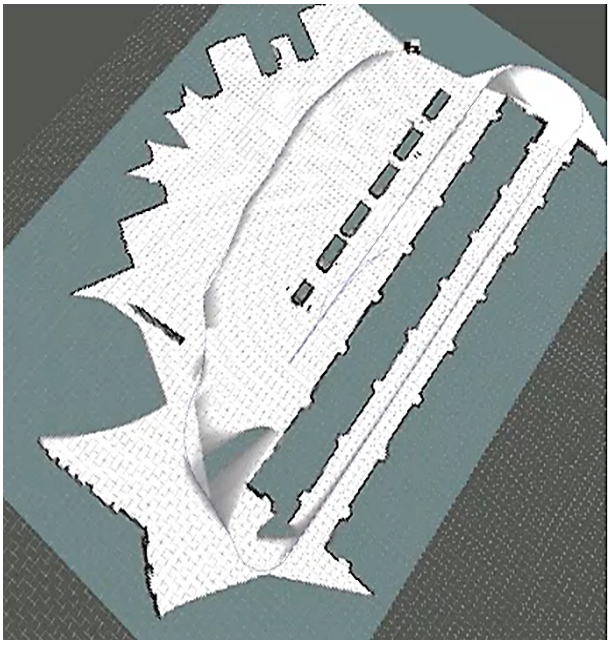

Figure 5. Result of the map obtained according to UNIFEI

The following Figure 6 generated by gui rqt graph shows the result given according to the relationship between the mapping, Gazebo, Rviz, Yolo and teleoperation nodes, as well as other packages responsible for the operation of the SLAM cartographer software.

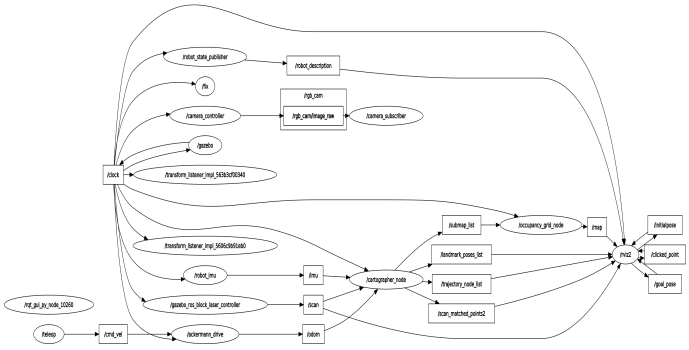

Figure 6. Visualization of the nodes connected to Yolo and Slam.

4.2 Navigation

Once the global map was generated, it was used to make the vehicle navigation in the simulation environment. In this simulation, the Navigation2 software package shows the delimitations of the blue environment that are considered obstacles for vehicle navigation. In addition, the software uses the AMCL (adaptive Monte Carlo) particle filter which has the location of the vehicle on the map and helps the robot to move to the desired point. Figure 7 shows the result of vehicle navigation leaving the map origin and moving to the desired point.

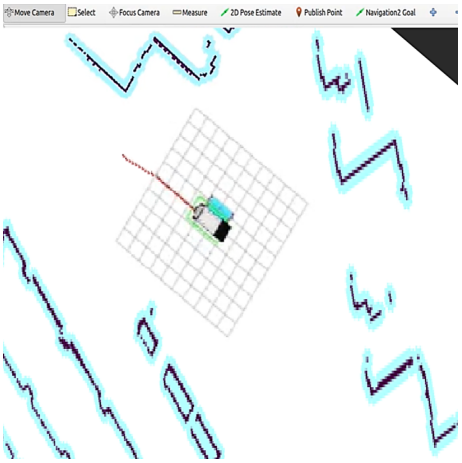

Figure 7. Map showing obstacles and trajectory of the vehicle.

On the other hand, a graph generated by RQT is also obtained as a result, which shows the relationship between the navigation nodes, Gazebo, Rviz, and other packages responsible for the operation of the nav2 software (Operation of the Nav2 software through the node map, link [2]). In the following link [3] the result of the simulation and integration is presented.

4.3 Detection

The image recognition algorithm reports the objects present in the environment, in this case vehicles and people. This information is extremely important for making decisions about vehicle navigation and the safety of the user and people in the environment. The combination of this information, with the map created from the environment, allows better decisions to be made in the autonomous navigation of the vehicle within the University. Figure 8 shows the obtained sensing and the generated map in a teleoperated navigation.

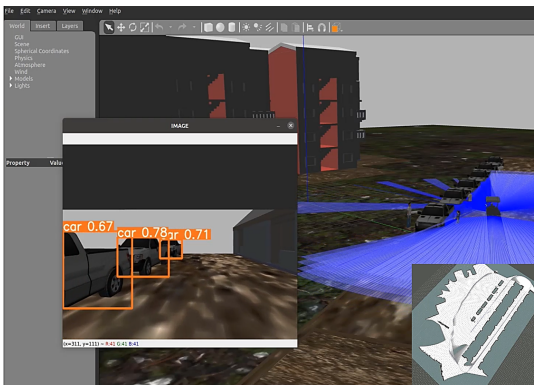

Figure 8. Integration of gazebo and slam for navigation and object detection.

On the other hand, in order to verify that the Yolo algorithm worked correctly, an experiment was carried out consisting of recording a one-minute video, where the car

[2] https://drive.google.com/file/d/1N4mIiThGUH9CZVYMBvj15kdWL4j6PfKj/view?usp=sharing
[3] https://youtu.be/aBmsR5X1hQc

navigates at three different speeds (1, 3 and 6.8 m/s). These videos were analyzed for each second giving the possibility of identifying false positives or false negatives, in order to make a confusion matrix (Khan et al. (2021); Sommer et al. (2020)) that will allow visualizing the performance of the algorithm in terms of object detection.

A confusion matrix was used for each of these objects (cars and people), to obtain precision results that give the proportion of correctly labeled objects to the whole sample. Metrics such as Recall, which is responsible for analyzing how many objects are detected and correctly detected, and the F-measure that determines the harmonic mean of precision and recall at the time of detection are used. The results of the videos are compiled in Figure 9, according to their corresponding precision, error%, recall (TPRate), PPV and F-measure.

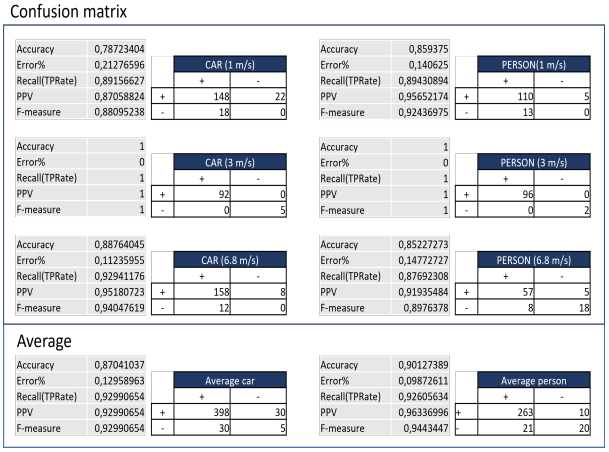

Figure 9. Results of the confusion matrix according to the cars and people detected in three times, obtaining as a result different parameters.

The results of averaging the matrices in Figure 9 show that the YOLO software works since the measured parameters do not reflect a great difference from the results obtained through the average. However, it is clear that, at the time of detection, it has a greater margin of error in cars, since this algorithm works through distance and shape, this means that it identifies cars when it finds something similar to their contour. On the other hand, it is analyzed that the higher the speed, the lower the margin of error. The following link [4] presents an experiment with navigation and object detection.

5. CONCLUSION

The experiments showed that it is possible to perform a simulation of a vehicle at the university (Unifei) through the ROS frameworks and Gazebo software, in addition, it was possible to add some of the sensors that are being implemented in the real golf cart.

As well, they implemented a detection algorithm that allows the use of Yolo, which is capable of detecting objects in different positions, obtaining an accuracy of 98.9% with a margin of error of 1.1% when identifying objects according to their shape, these results were validated with the confusion matrix.

In future work, these applications will be improved and implemented on the vehicle currently in Unifei's facilities. In order to test the simulation, works in the proper environment, which allows you to reduce costs and time.

[4] https://youtu.be/45qVR0e$_Rwg$

REFERENCES

Ahmed, A.A., Olumide, A., Akinwa, A., and Chouikha, M. (2019). Constructing 3d maps for dynamic environments using autonomous uavs. In *Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MobiQuitous '19, 504–513. Association for Computing Machinery, New York, NY, USA. doi:10.1145/3360774.3368200. URL `https://doi.org/10.1145/3360774.3368200`.

Arango, J.F., Bergasa, L.M., Revenga, P.A., Barea, R., López-Guillén, E., Gómez-Huélamo, C., Araluce, J., and Gutiérrez, R. (2020). Drive-by-wire development process based on ros for an autonomous electric vehicle. *Sensors*, 20(21). doi:10.3390/s20216121. URL `https://www.mdpi.com/1424-8220/20/21/6121`.

Banjanovic-Mehmedovic, L., Karabegovic, I., Jahic, J., and Omercic, M. (2021a). Optimal path planning of a disinfection mobile robot against covid-19 in a ros-based research platform. *Advances in Production Engineering And Management*, 16, 405–417. doi:10.14743/APEM2021.4.409.

Banjanovic-Mehmedovic, L., Karabegović, I., Jahic, J., and Omercic, M. (2021b). Optimal path planning of a disinfection mobile robot against covid-19 in a ros-based research platform. *Advances in Production Engineering Management*, 16, 405–417. doi:10.14743/apem2021.4.409.

Bjelonic, M. (2016–2018). YOLO ROS: Real-time object detection for ROS. `https://github.com/leggedrobotics/darknet_ros`.

Burger, P., Naujoks, B., and Wuensche, H.J. (2019). Map-aware slam with sparse map features. *IEEE International Conference on Intelligent Robots and Systems*, 347–353. doi:10.1109/IROS40897.2019.8968466.

Corović, A., Ilić, V., Duric, S., Marijan, M., and Pavković, B. (2018). The real-time detection of traffic participants using yolo algorithm. In *2018 26th Telecommunications Forum (TELFOR)*, 1–4. doi:10.1109/TELFOR.2018.8611986.

Faz-Mendoza, A., Gamboa-Rosales, N.K., Medina-Rodríguez, C.E., Casas-Valadez, M.A., Castorena-Robles, A., and López-Robles, J.R. (2020). Intelligent processes in the context of mining 4.0: Trends, research challenges and opportunities. In *2020 International Conference on Decision Aid Sciences and Application (DASA)*, 480–484. doi:10.1109/DASA51403.2020.9317095.

Hussein, A., García, F., and Olaverri-Monreal, C. (2018). Ros and unity based framework for intelligent vehicles control and simulation. In *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, 1–6. doi:10.1109/ICVES.2018.8519522.

Jakubiec, B. (2018). Application of simulation models for programming of robots. *SOCIETY. INTEGRATION. EDUCATION. Proceedings of the International Scientific Conference*, 5, 283. doi:10.17770/sie2018vol1.3214.

Khan, M.Z., Khan, M.U.G., Saba, T., Razzak, I., Rehman, A., and Bahaj, S.A. (2021). Hot-spot zone detection to tackle covid19 spread by fusing the traditional machine learning and deep learning approaches of computer vision. *IEEE Access*, 9, 100040–100049. doi:10.1109/ACCESS.2021.3094720.

Marian, M., Stîngă, F., Georgescu, M.T., Roibu, H., Popescu, D., and Manta, F. (2020). A ros-based control application for a robotic platform using the gazebo 3d simulator. In *2020 21th International Carpathian Control Conference (ICCC)*, 1–5. doi:10.1109/ICCC49264.2020.9257256.

Mario Gluhaković, M.H. (2020). *2020 Zooming Innovation in Consumer Technologies Conference (ZINC) : Online, 26-27 May 2020.*

Quang, H.D., Manh, T.N., Manh, C.N., Tien, D.P., Van, M.T., Kim, D.H.T., Thanh, V.N.T., and Duan, D.H. (2019). *Mapping and Navigation with Four-wheeled Omnidirectional Mobile Robot Based on Robot Operating System.* IEEE.

Redeker, M., Klarhorst, C., Göllner, D., Quirin, D., Wißbrock, P., Althoff, S., and Hesse, M. (2021). Towards an autonomous application of smart services in industry 4.0. In *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1–4. doi:10.1109/ETFA45728.2021.9613369.

Rivera, Z.B., De Simone, M.C., and Guida, D. (2019). Unmanned ground vehicle modelling in gazebo/ros-based environments. *Machines*, 7(2). doi:10.3390/machines7020042. URL `https://www.mdpi.com/2075-1702/7/2/42`.

Sai Sahith Velamala, Devendra Patil, X.M. (2017). *IEEE ROBIO 2017 : 2017 IEEE International Conference on Robotics and Biomimetics : December 5-8, 2017, Macau SAR, China.*

Sommer, N.M., Velipasalar, S., Hirshfield, L., Lu, Y., and Kakillioglu, B. (2020). Simultaneous and spatiotemporal detection of different levels of activity in multidimensional data. *IEEE Access*, 8, 118205–118218. doi:10.1109/ACCESS.2020.3005633.

Stack, R.N. (2020). Nav2. URL `https://navigation.ros.org/about/index.html#about`.

Tian, S., Liu, D., He, X., and Wang, C. (2021). A visual perception method for autonomous navigation at sea based on ros. In *2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, 149–153. doi:10.1109/ICCASIT53235.2021.9633563.

Valencia, R., Andrade-Cetto, J., and Porta, J.M. (2011a). Path planning in belief space with pose slam. 78–83. doi:10.1109/ICRA.2011.5979742.

Valencia, R., Andrade-Cetto, J., and Porta, J.M. (2011b). Path planning in belief space with pose slam. *Proceedings - IEEE International Conference on Robotics and Automation*, 78–83. doi:10.1109/ICRA.2011.5979742.

Villa, J., Vallicrosa, G., Aaltonen, J., Ridao, P., and Koskinen, K.T. (2020). Model-based guidance, navigation and control architecture for an autonomous underwater vehicle. In *Global Oceans 2020: Singapore – U.S. Gulf Coast*, 1–6. doi:10.1109/IEEECONF38699.2020.9389247.

Xu, Y., Zhang, T., and Bao, Y. (2021). *Analysis and Mitigation of Function Interaction Risks in Robot Apps*, 1–16. Association for Computing Machinery, New York, NY, USA. URL `https://doi.org/10.1145/3471621.3471854`.

Xuexi, Z., Guokun, L., Genping, F., Dongliang, X., and Shiliu, L. (2019). Slam algorithm analysis of mobile robot based on lidar.

Yang, L. and Chi, H. (2021). Slam self - cruise vehicle based on ros platform. In *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, 6–11. doi:10.1109/ICCECE51280.2021.9342204.

ANEXO B – Artigo revisão sistemática

Engineering Applications of Artificial Intelligence

Systematic review according to topological navigation approaches implemented to autonomous vehicle.

--Manuscript Draft--

Manuscript Number:	EAAI-22-3724
Article Type:	Research paper
Keywords:	Topological navigation Autonomous vehicles Space navigation Geospatial analysis Image processing.
Corresponding Author:	Natalia Sánchez Sánchez COLOMBIA
First Author:	Natalia Sánchez Sánchez
Order of Authors:	Natalia Sánchez Sánchez
	Rafael Francisco dos Santos
	Ruben Dario Hernandez Beleño
	Adler Diniz de Souza
	Giovani Bernardes Vitor
Abstract:	The topological maps allow a concise representation of the world by keeping only information about the relevant places and navigation allows determining the position, course, time, speed, and distance. Given this, this paper proposes a systematic review, which aims to know the status of topological navigation algorithms integrating image processing and look at what low-cost sensors can be implemented in these types of systems. As a result, shows the advantages and disadvantages that methods that image processing integrate with topological navigation algorithms, considering the collection of a database of 315 articles published in the last 5 years, which can be a useful guide for researchers and engineers in the field. Another result is that a PICO (Patient, intervention, comparison, and outcome) methodology is used to establish the research questions, and exclusion and inclusion criteria were also implemented, resulting in 109 articles that meet the requirements, according to the implementation of tools that help to unify, plan, and report the review, according to the databases investigated, this to obtain reliable information. Evidence that 35% of the publications are based on projects based on low-cost simulators and sensors, which can perform tests when implementing topological navigation. Finally, 50% of these authors implement cameras for the detection of the environment and vision of the system, and 20% use lidar sensors to determine the distance from the vehicle to an object or surface, which allows the recognition of the environment.
Suggested Reviewers:	Paola Andrea Niño Suarez Instituto Politecnico Nacional Secretaria de Investigacion y Posgrado pninos@ipn.mx
	Oscar Fernando Avilés Sánchez, Doctor Pilot University of Colombia oscar-aviles@upc.edu.co

Declaration of interests

☐The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Natalia Sanchez Sanchez reports financial support and article publishing charges were provided by Federal University of Itajubá. Natalia Sanchez Sanchez reports a relationship with Federal University of Itajubá that includes: non-financial support.

Systematic review according to topological navigation approaches implemented to autonomous vehicle.

Natalia S. Sanchez[1,1], Rafael Santos[1], Ruben Hernandez[b], Adler Diniz de Souza[c], Giovani Bernardes[1]

[a]*Federal University of Itajuba, Institute of technological Sciences - ICT/UNIFEI, Laboratory of Robotics, intelligent and Complex Systems - RobSic, 35903-087 Itabira-MG.*
[b]*Universidad Militar Nueva Granada, Colombia.*
[c]*Instituto de Matemática e Computação (IMC), Universidade Federal de Itajubá, Itajubá 37500-903, Brazil.*

Abstract

The topological maps allow a concise representation of the world by keeping only information about the relevant places and navigation allows determining the position, course, time, speed, and distance. Given this, this paper proposes a systematic review, which aims to know the status of topological navigation algorithms integrating image processing and look at what low-cost sensors can be implemented in these types of systems. As a result, shows the advantages and disadvantages that methods that image processing integrate with topological navigation algorithms, considering the collection of a database of 315 articles published in the last 5 years, which can be a useful guide for researchers and engineers in the field. Another result is that a PICO (Patient, intervention, comparison, and outcome) methodology is used to establish the research questions, and exclusion and inclusion criteria were also implemented, resulting in 109 articles that meet the requirements, according to the implementation of tools that help to unify, plan, and report the review, according to the databases investigated, this to obtain reliable information. Evidence that 35% of the publications are based on projects based on low-cost simulators and sensors, which can perform tests when implementing topological navigation. Finally, 50% of these authors implement cameras for the detection of the environment and vision of the system, and 20% use lidar sensors to determine the distance from the vehicle to an object or surface, which allows the recognition of the environment.

Keywords:
Topological navigation, autonomous vehicles, space navigation, geospatial analysis, image processing.

1. Introduction

The World Health Organization (WHO) has prepared a report that found that approximately 1.3 million people die on the world's roads every year and between 20 and 50 million suffer non-fatal injuries and ensure that one of the main reasons are traffic accidents. Furthermore, more than half of all traffic fatalities occur among vulnerable road users: pedestrians, cyclists, and motorcyclists Organization (OMS). The WHO says that these accidents represent 12% of deaths caused by road traffic worldwide (Organization, 2021). This is due to driving at high speeds, recklessness, and being under the influence of alcohol or psychoactive substances, among others.

Technologies such as forward collision warning (FCW), automatic emergency braking (AEB), lane departure warning (LDW), lane keep assist (LKA) and blind-spot monitoring (BSM) have the potential to help prevent approximately 14% of all vehicle crash fatalities (Benson et al., 2018). These technologies are currently sought after as they promote the decrease of death rates due to automobile accidents, in addition to allowing faster and safer travel for people. An example of implementation of these technologies is the following brands: Tesla Motors Inc, Amazon (Zoox), Uber, Google (Waymo), Toyota, Baidu Inc, Nissan Motor Company Ltd, The Volvo Group, Apple Inc, and Alphabet Inc, among others. These large companies bring constant innovation as they focus on conducting various validation tests with different prototypes of autonomous vehicles on all types of roads and scenarios, to improve the quality of life through autonomous driving.

However, these vehicles are not complete, as they lack full autonomy. The big challenge is to implement efficient autonomous navigation because it is one of the most complex tasks that autonomous vehicles may encounter. After all, although it may seem like a trivial task for humans it is composed of a variety of sub-tasks and problems that make automatic navigation difficult (Francisco Borja y Arnau, 2016). Moreover, the main problem for autonomous vehicles is to answer the question: **"What is its current location and, what is in the surround?"**; this means that the vehicle must be able to find its position in the environment in which it finds itself and perceives it.

Robot autonomy is one of the main goals set by robotics researchers in recent decades. Currently, research has been carried out to realize navigation systems, which aim for a robot to have the ability to determine its own position in its frame of reference and then be able to plan a path to go to an objec-

Email addresses: natalia.sanchez.sanchez@outlook.com (Natalia S. Sanchez), giovanibernardes@unifei.edu.br (Giovani Bernardes)

tive location. On the other hand, to navigate in its environment, the robot or any other mobility device requires a representation, i.e., a map of the environment and the ability to interpret. These representations are performed by some authors with the integration of image processing or topological analysis, which allow for recognizing the space where it is located and the detection of objects in real-time, to retrieve or extract information quickly and flexibly (Zhang et al., 2021).

Consequently, there are different systems capable of being used in autonomous navigation, among the best known are the GPS and IMU sensors (Laoufi et al., 2019; Palafox et al., 2019), which allow the continuous calculation of position, orientation, and speed, among other variables. However, these sensors face problems in different operating scenarios. Therefore, they are not entirely reliable in their measurement, moreover, they can become expensive (Lee y Kum, 2020) systems. Other solutions for efficient navigation are based on visual learning (Joo et al., 2020), location systems (simultaneous mapping) (Eriksen y Frandsen, 2018) and topological location (Zhang et al., 2021).

Given this, different studies have been carried out that implement some type of navigation, which allows a robot to move from one point to another safely and efficiently (Feraco et al., 2020; Laoufi et al., 2019; Jia et al., 2020b; Imad et al., 2021), however, there is a gap in the collection of information from the literature, that addresses the advantages and disadvantages of the algorithms that implement this, as well as that, demonstrates that the integration of topology navigation can be carried out which allows the combination of fundamental competencies such as location, mapping, recognition, perception, route planning, and movement control. Consequently, this research focuses on topological navigation and image processing, where two research questions are posed which follow the PICO methodology (Patient, intervention, comparison, and result), in order to provide value questions, give an order and structure (Eriksen y Frandsen, 2018); these questions are:

QP1: *Which sensors are best suited to be used in an autonomous vehicle navigation app, based on precision, price, and size ratio?*

QP2: *What are the most convenient topological navigation and image processing integration approaches for an application in autonomous vehicles, considering their advantages and disadvantages?*

According to these questions, this study performs a systematic review that collects and analyzes multiple studies allowing us to obtain answers to these questions, to help give better guidance in decision making in future research using navigation. Some of the key contributions of this systematic review paper can be summarized as follows:

- It compiles the main simulators from 2016 to 2021, which are used to perform validation tests of autonomous vehicles, being able to implement navigation algorithms.

- This systematic review covers the contemporary literature on topological navigation problems and takes up more than 30 algorithms implemented in 2016 until 2021, grouped by different databases.

- Provides a thorough review and insightful analysis of different aspects of topological algorithms that integrate different sensors such as laser, camera, gyroscopic, radar, and lidar, among others.

- Offers several challenges and possible future directions for topological navigation based on image processing. Shows a collection of algorithms with their advantages and disadvantages.

- Systematic review offers decision support when choosing a sensor, in terms of accuracy and cost.

This article is subdivided into five sections: Section II describes the research methodology used to collect and compare the different articles and research found, and a bibliometric network is elaborated according to exclusion and inclusion criteria; in Section III, the main results are presented and discussed, defining what topological navigation is and which sensors and algorithms have been implemented over time; in Section IV, the research questions are answered and finally, in Section V, the corresponding conclusions are presented.

2. Materials and Methods

A research methodology was used for the preparation of this article, as shown in Figure 1. This methodology is divided into 5 phases where a subdivision is made in three blocks. The first block integrates the formulation of the research questions. The second one is composed of the location, selection, and evaluation of studies, which indicate the planning process for the systematic review. In this block, different databases are used, inclusion and exclusion criteria are defined, and a search equation is composed to achieve a reliable database. On the other hand, the third block is the analysis and results of the research.

Graph A, in Figure 1, shows the progress made during the last years, and graph B shows the rejected and accepted articles according to block 2.

Inclusion criteria and exclusion criteria are said to be an important part of a systematic review since if they are correctly defined, the inclusion and exclusion criteria increase the probability that the review will generate reliable results and minimize the risks or biases of the information (Leng et al., 2019; Yurtsever et al., 2020). For these criteria, a bibliometric network was created to identify whether this search is valid and current.

The Parsifal tool (Parsifal, 2014), which is a tool that helps in the context of the development of systematic literature reviews to provide some help to plan and organize information, was used to support this methodology. It provides a mechanism to create a quality assessment checklist and data extraction forms. With the help of this tool, we were able to make a quality assessment. These questions are linked to the research questions and according to the answers previously expected by the review. In this case, a scale of 0 to 1 was used, where 0 is literature that does not comply with any of the questions and the information is not reliable, 0.5 corresponds to partially complied with and 1 complies with the requirements for the study.

Table 1 shows the exclusion and inclusion criteria considered when classifying the articles. In this case, of the 315 articles, 109 were included and met the inclusion criteria. Like-

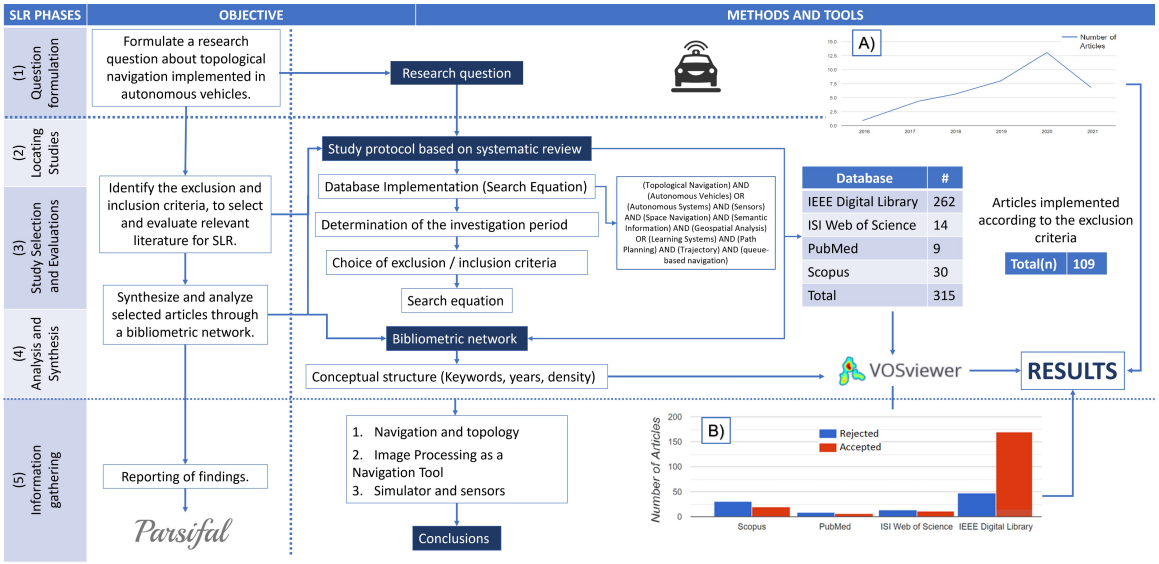

Figure 1: Research methodology according to the phases of Systematic Literature Review (SLR) in which the objectives and results obtained are shown, in the section, it is shown A) Graph of progress in the last 6 years of the literature focused on themes Autonomous Vehicles and Autonomous Navigation B) Graphic that shows the results of acceptance criteria according to four databases.

wise, in this link[1] shows the result issued by Parsifal, to validate all the criteria that were considered, with their corresponding results.

Table 1: Exclusion and inclusion criteria for article selection.

✓ INCLUSION CRITERIA	x EXCLUSION CRITERIA
CI01: Presents a study on technology use.	**CE01:** Articles so far under review.
CI02: Shows results of sensory tools and navigation algorithms.	**CE02:** Studies prior to 2016.
CI03: Includes usage detection tools.	**CE03:** Duplicate studies.
CI04: The results of the implemented techniques are evidenced.	**CE04:** Studies in any language other than English.
CI05: The articles must be published in well-known magazines or conferences.	**CE05:** Does not include the use of sensorial tools and navigation algorithms.
	CE06: Articles in which keywords are not included in the title, abstract and/or text of the publication will not be selected.

Finally, a database of 109 articles was used. These articles meet each of the criteria, considering that 92 are duplicates in the databases, and 114 do not meet the information or exclusion criteria of the 315 base articles.

2.1. *Bibliometric network*

A bibliometric network also called scientific mapping was implemented to visualize scientific reproduction through the extraction of metadata. It contains authors, organizations, countries, documents, source-reviews, keywords, cited references, cited authors, or cited source-reviews, and units of measurement (co-authorship, co-occurrence, citation, bibliographic coupling, or awareness). These metadata are used as nodes that may be analyzed in this type of network relationship. Visualization of similarities (VOSviewer) (VOSviewer, 2022) is used and a clustering technique is also applied.

Figure 2 shows the bibliometric network of the compilation of the previously mentioned databases. The colors are represented by the clustering technique, which is a set of closely related nodes according to the type of link being analyzed. In this case, the colors are selected from most to least important (red, green, blue, yellow, purple, light blue, orange, brown, and pink). For this bibliometric network, all the keywords related to the 109 selected articles were considered. It is concluded that it presents great importance in the area of geospatial analysis, remote sensing, trajectory, robots, vehicles, navigation, topology, mobile robots, among others. Also, the keywords indicate that satisfactory research was carried out using the databases since they show words of interest such as topology, navigation, and remote sensing. Also, the size of the circles signifies the parameter that is being studied the most.

In addition to having the objective of visualizing the conceptual structure (Figure 1), the evolution of the research field is analyzed, which shows great progress. Figure 3 is divided into two sections: **(A)** It represents the trajectory during the impact of these technologies in recent years and **(B)** It shows groupings by colors, in this case, each node in the map is represented with a color ranging from yellow to blue, reflecting the density of the relationships between the descriptors. The higher the density, i.e., the co-occurrence between the descriptors, the closer they will be to the yellow color, whereas if they are closer to the blue color, this indicates a greater dispersion and, therefore, less co-occurrence. Also, the representation of the variation of the font size and its shading is shown; as the last variable, it presents the different proximity between the terms, this analy-

[1] https://drive.google.com/file/d/1Cs42_64yJy3Cimytf7wrOF9eIjvGYUGm/view?usp=sharing

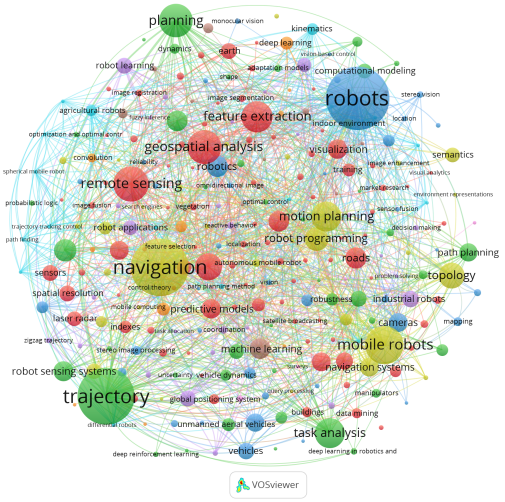

Figure 2: Bibliometric network through IEEE, Scopus, Web of Science and PubMed keywords and databases.

sis is by probabilities with the mathematical algorithm called Kernel (”Density view”tab) (Dwi Fitria Al Husaeni, 2021).

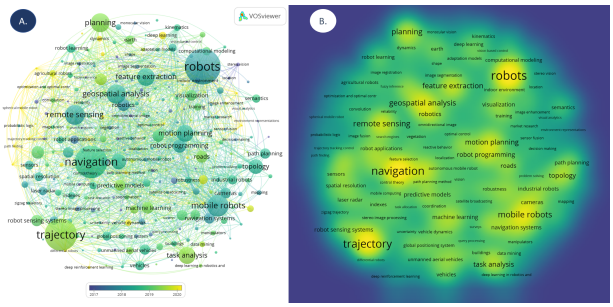

Figure 3: Bibliometric network based on years of trajectory and density.

3. Results

The results of the systematic review show that an automated guided vehicle (AGV), or an autonomous vehicle, is a car that can drive without human intervention. To do this, it needs various characteristics that generate autonomy, among which is the detection of the environment, measurement, and decision-making in real-time, among others. This also depends on the navigation control system, which can change or follow a collision-free path in the work area.

Navigation can be solved in various ways, but the solution to choose depends on the type of workspace in which the autonomous vehicle moves. The work environment may or may not be familiar. Depending on the types of workspaces, some authors (Imad et al., 2021; Feraco et al., 2020; Francisco Borja y Arnau, 2016) say that two different navigation strategies can be found: global and local, or reactive (based on behavior); A global navigation strategy is used when the user knows, provides or perceives the environment in which the robot moves. Local or reactive navigation systems are used to collect information from sensors and modify the vehicle's trajectory when the vehicle's working space is unknown, or its dynamic behavior is uncertain.

On the other hand, the topology and image processing implemented in navigation represents a tool that allows gathering information about the environment where it is located. Next, three sections are shown, which are navigation and topology, image processing as a navigation tool, and finally, tools for real-time simulations are shown since they are intended to integrate different sensors, control configurations, and vehicle navigation.

3.1. Navigation and topology

Navigation is about answering questions like *What is your current position? Where have you been? Where are you going? How can you reach the goal most efficiently? this time?*, among other questions. Different authors (Emmi et al., 2021; Blue y Brindha, 2019) define navigation as a set of actions performed to reach an assigned goal or destination from a given point of origin; Likewise, various investigations (Zhang et al., 2021; Le et al., 2021; Ortiz et al., 2020; Kim et al., 2017) have tried to explain how people are located and how they reach a destination.

One of the main challenges of navigation is the creation of maps and the planning of routes when driving, since they must consider sensory perception, movement control, and object detection, among other aspects. Navigation allows to have an environment modeling, a sensory system, route planning, a control system, and a positioning system (Daniusis et al., 2021; Chen et al., 2021). Several authors suggest solutions such as localization, simultaneous mapping, visual teaching, repetition and topological localization as a solution to these situations (Chen et al., 2021; Sur, 2019; Nashed et al., 2018).

Navigation can be implemented using geometric maps, decomposition maps, topological maps, and semantic maps (Silva et al., 2019; Mongus y Juric, 2019; Luo y Shih, 2019). Recently, the topological map is used in the literature to simplify the search space, an example is the case of C.Wang (Wang et al., 2020a) and Oleynikova (Oleynikova et al., 2018) present a 3D topological map in order to obtain planning of routes, they used some patterns when implementing navigation. Below are the main characteristics that you must consider when making navigation:

*3.1.1. **Location***

One of the tasks of a robot's navigation system is to estimate its own position and orientation in the environment it is embedded in and then plan a path to its destination. The positioning of the vehicle in the environment, also called location, must be the most precise, since this guarantee is necessary so that the vehicle can move safely in the environment, without the need for a driver. Currently, to implement navigation, they use different sensors such as GPS, vision sensor, laser scanner,

ultrasonic sensor, RFID, odometer, IMU, digital compass, and gyroscope, among others.

In some applications, simultaneous localization and mapping (SLAM) must be performed, where the 2D/3D structure of the environment is defined and, at the same time, the position and orientation of the vehicle in this environment is estimated (Choi et al., 2020; Wheeler et al., 2020; Mashoshin y Pashkevich, 2020). G.Li (Li et al., 2018a) says that precise localization is an important task for safe driving, performs a SLAM algorithm based on Kernelized RÃ©nyi Distance (KRD).

SLAM is a technique used by robots and autonomous vehicles to build a map of an unknown environment it is in while estimating its trajectory as it moves within this environment (Park et al., 2011; Blue y Brindha, 2019; Gonzalez et al., 2016; Arroyo et al., 2016). The use of this technique allows to have an automated guided vehicle without infrastructure, which has the advantage of satisfying restrictions that are not easy to model in the controller robot, such as the existence of restricted regions, or the right of way along roads.

Therefore, SLAM is not only used for land roads (Wang et al., 2014; Aldibaja et al., 2020) but also for underwater areas, as is the case with Choi (Choi et al., 2019) who performs methods to provide accurate underwater location based on acoustics without no prior information of source location based on a Gaussian sum filter. This implements inertial sensors with the directional angles of the acoustic sources to estimate the locations of both the vehicle and the acoustic sources. A different approach is used in Q. Zhang (Zhang et al., 2018) which uses an algorithm called UKF-SLAM (Unscented Kalman filter (UKF) - simultaneous localization and mapping) which employs a camera based on imaging sonar features in a structured environment submarine. It also uses random sampling to extract line features from the sonar scan data to build the feature map. On the other hand, Wu (Wu et al., 2018) uses an algorithm called EKF-SLAM (Extended Kalman Filter - SLAM) for autonomous landing of unmanned helicopters, and Rizk (Rizk et al., 2020) adopts an image stitching technique to overcome the limitations of current methods in terms of complexity and memory requirements, to locate a vehicle.

Wen (Wen et al., 2020) introduces a novel visual-inertial stereo system with loop closure detection based on a 3D point cloud hybrid semantic-topological map framework. It performs autonomous navigation by providing a map for route planning. Martini (Martini et al., 2020) realizes a novel two-stage system that integrates candidates to the topological location of a place recognition system, says that using radars, spectral techniques based on landmarks can be implemented, and autonomy can be obtained before changes extremes in appearance or condition. These two authors present different proposals, but at the same time, they use navigation through the location. On the other hand, Valencia (Valencia et al., 2011) does not have the reference point in mind, to obtain optimal navigation by searching the route, the objective is to have the least uncertainty that a robot needs. Also, they not only implement navigation in algorithms, D.Wang (Wang et al., 2014) presents a polarization sensor based on a bionic camera in real-time, this sensor manages to have two working modes: the first is a single point measurement and the second is a multi-point measurement mode.

3.1.2. ***Mapping***

Maps are symbolic representations of a given environment. In navigational approaches, there are two types of mapping paradigms: metric and topological. Metric maps represent the world accurately, capturing in detail much of the metric information and distances of the environment. These maps are usually produced using a global coordinate system. Metric maps are usually considered more appropriate in applications to guide vehicles and avoid obstacles. However, metric maps are more difficult to construct and maintain. The construction of these maps is also considered computationally expensive (Silva et al., 2019; Kessler et al., 2018). On the other hand, topological maps represent the environment more abstractly through a graph. The nodes of the graph represent distinctive or relevant locations in the environment and the arcs model the immediate neighborhood relationships of these locations. Generally, there is no distance information between locations connected by an arc. And the locations or nodes of the graph are usually not associated with a position or a global coordinate system (Silva et al., 2019; Kessler et al., 2018; Sui et al., 2020; Huachao et al., 2020). Nodes are interpreted as places, regions, positions, or landmarks in the environment, and arcs represent the connection between nodes, which may well represent traversable paths or motion control commands.

The topology is evident when a robotic system establishes instructions such as ”go to the right”, ”forward”, ”turn” and so on, geometrically it is when you want to arrive at a map of the environment indicating, according to specific coordinate signals (x, y, *theta*). Burger (Burger et al., 2019) performs sparse optimization maps based on graphs of a trajectory through a line, finding its specific position of it, unlike Ort (Ort et al., 2018) which does autonomous vehicle navigation in rural environments without detailed previous maps. These trajectories are updated to stay in the local frame using the vehicle's odometry, the associated uncertainty based on residual least squares, and a recursive filtering approach, which allows the vehicle to navigate road networks reliably and at high speed, without detailed previous maps. Furthermore, Rambhatla (Rambhatla et al., 2018) proposes a technique to develop (and locate) topological maps from light and range detection data via a lidar sensor. To these authors, locating an autonomous vehicle in relation to a reference map in real-time is crucial for its safe operation.

3.1.3. ***Planning/Design***

Path planning is the key issue for autonomous mobile robots. Path planning and motion control problems are not only important for achieving autonomous navigation and other complex intelligent tasks but also incorporate the perceptual ability and intelligence of robots. W. He (He et al., 2017) says that there are two types of planning, one based on known environmental models or knowledge perception maps, and one based on case-based planning. The above planning is according to the known environmental model or perception map knowledge to plan; the latter is in accordance with the planning of existing

knowledge by using the matching method to solve the planning problem. Case-based planning applies to more complex but relatively fixed environments. Other authors (Peralta et al., 2020; Kazim et al., 2021) divide it into global planning or route planning, and local planning or route tracking. These are important because each of the situations that the robot can face is an extremely complex task due to inaccuracies and lack of information all the way.

Currently, there are many authors working on route planning, such as C.Jia (Jia et al., 2020a) proposes an autonomous vehicle motion planning method based on dynamic programming, K.Lee & Kum (Lee y Kum, 2020) elaborates longitudinal and lateral integrated safe route planning of an autonomous vehicle across the friction boundary. The trajectory planning algorithm proposes to evaluate simulations in a single obstacle scenario by comparing them with the existing AEB (Autonomous Emergency Braking) and AES (Autonomous Emergency Steering) algorithms. H. Jia (Jia et al., 2020b) makes a dynamic lane change trajectory planning scheme for autonomous vehicles on structured highways, as well as a lane change trajectory planning method for autonomous vehicles on structured highways. S. Cai and Y. Wan (Cai y Wan, 2020) implements a local trajectory planning and control method for autonomous vehicles based on the RRT (Quick exploring Random Tree) algorithm, among others (Jia et al., 2020b; Yang et al., 2020; Bitar et al., 2020). All these authors show that there are different methods to achieve planning; Likewise, control strategies can be implemented such as PID controller, neural network, fuzzy logic genetic algorithm, and adaptive controller, these strategies are linked to route planning algorithms, which can be: algorithm A *, histogram algorithm vector field algorithms, Table 5 lists the main algorithms according to their advantages and disadvantages.

3.2. Image Processing as a Navigation Tool

Topological navigation systems are performed through road planning where geometric trajectories directed towards a specific objective are used. These implement stochastic methods and navigation by heuristic methods which makes these systems optimal but very cumbersome and complicated when it comes to adjusting, documenting, modifying, and improving (Patruno et al., 2021; Luo y Shih, 2018). Through the processing of digital images, it allows having a tool that improves the quality of the information contained in an image so that this information can be interpreted and processed automatically.

In topological navigation, the word topology means the study of position or location. Topology is the study of shapes, including their properties, the deformations applied to them, and the mapping of functions between them. Digital topology is defined as the study of topological relationships in a digital image, which contains a rectangular matrix of finite pixels that is modeled using the digital plane (Adams y Franzosa, 2007). Through these tools, sensors can be implemented, which achieve the basic objective of planning a trajectory that allows the robot to move from a starting position to a final position in an environment.

One of the great challenges of the integration of navigation and images is that it must improve robustness, and efficiency in visual navigation indoors and improve the general characteristics since they are susceptible to interference by occlusion, translation, and lighting, which restricts their use. correct operation in the implementation.

S. Xu, Zhou, and Chou (Xu et al., 2020; Song Xu y Chou, 2019) realize a scalable vision-based topological mapping and navigation architecture for a mobile robot to work robustly and flexibly in a large-scale environment by exploiting image fine-tuning. They divide their proposed architecture into three stages, which are described below:

*3.2.1. **Topological mapping***

At this stage, the keyframes are sought to represent the discriminatory environment. In this case, they come by nodes of the topological map, through a neural network. These represent the scenario with distinctive characteristics, according to typical local reference points, in addition to providing semantic visual information for the robot to perform flexible and robust navigation.

To build the topological map you must look for: 1) similarity of frames that must be greater than a certain threshold, to ensure the discriminative and representative capacity of the image; 2) In the case of images with few local visual characteristics, you must follow a digital image process, where the image is captured and the similarity between the current image and the keyframes is calculated, to perform the comparison method through the creation of a new topological node. Considering that when it is greater than a certain threshold, it means that there is a similarity between the current image and the existing frames. Then a pre-process and segmentation is done, which consists of adding a border between the current node and the previous one, which is considered the reference for planning.

*3.2.2. **Global location***

They use a global location based on the Monte Carlo method based on images and particle filters. This helps to estimate the position of the robot and allows the recovery of images, calculated by each sample. It also uses Hash technology to compare images robustly and efficiently.

In addition, the topological location based on images seeks to learn the weights, based on pre-trained models; for detection they extract the characteristics of high-dimensional convolution from the images, implementing a thick filter based on global characteristics and a fine recovery filter, which takes care of the local characteristics, in this filter reorders the characteristics of the local region of the image, and is used through activation groupings by regions, adding an ROI (Region of Interest).

After this it is localized by the Monte Carlo method, estimating the maximum a posteriori over the topological space through a Bayesian filter, then the a posteriori distribution of the sample set is calculated, determining the data association between the current observation and the corresponding topological node, to provide the most general solution for the perceptual aliasing.

The monte Carlo method consists of the following steps:

- Prediction phase: Random samples are captured around the topological nodes, which contain different similarities.

- Update phase: It is based on the probability distribution of the robot, which will be updated according to the sampling, i.e. it is the calculation of the posterior probability distribution consisting of three steps (Prediction, updating, and resampling).

- Adaptive genetic algorithm: Adaptive evolution strategy of the genetic algorithm to increase the diversity of samples and reduce the computational load.

3.2.3. ***Trajectory planning***

At this stage planning algorithms are used, in this case, the Dijkstra algorithm finds the shortest path based on a graph from the current topological node to the target topological node. It is also done by traversing a sequence of images from keyframes for integration into the topological map.

This is a great example of integrating image processing as a topological navigation tool. In this case, this author was able to integrate visual topological localization with the help of neural network-based image retrieval with a probabilistic approach to Monte Carlo localization, showing performance in efficiency and robustness compared to existing navigation methods (Table 3 and 4).

3.3. Simulators

One of the main objectives of a sensor is to perform measurements, which can do different functionalities in an automated manufacturing system. For example, detecting, classifying, monitoring, counting, positioning, determining the orientation of something specific, measuring productivity, and ensuring quality, among others. Sensors are also essential when implementing them in an AGV, as they represent the vehicle's perception. However, these sensors are often too expensive to be integrated on a large scale in the mass production of vehicles (Bitar et al., 2020; Zhang et al., 2020b; Jaen, 2017). Current AGV is mainly based on a combination of light detection and ranging sensors (lidar), radar and camera sensors to build a 3D (semantic) map of a scene, ultrasound, GPS, and laser, among others.

Sensors respond to the presence or absence of any type of object, large or small, transparent, or opaque, shiny, or matte. Extracting depth information is key in autonomous driving. In outdoors scenarios, this task is commonly addressed by using lidar and radar systems, which are active sensors (Camargo et al., 2019; Huang et al., 2020; Gupta y Fernando, 2022) and these sensors are being used to estimate the location of a robot. However, they have a shortcoming when integrated into an urban environment, which presents more challenging situations as in many scenarios, e.g., on secondary roads or some city streets, lane lines are not present or not well enough marked.

Several authors have proposed different strategies to implement in AGV. In this case, Palafox (Palafox et al., 2019) presented a vision-based method to locate a vehicle within the road when there are no lane lines. It uses only RGB images as input, fusing the outputs of semantic segmentation and a monocular depth estimation architecture, to locally reconstruct a 3D semantic point cloud of the viewed scene. D.Martini (Martini et al., 2020) builds Deep Learning method using a truncated Monte Carlo tree search to evaluate parking states and select different motions. This author uses artificial neural networks to provide the search probability of each tree branch, as does J.H. Uhl (Uhl et al., 2020) who uses an automated machine learning-based framework to extract symbols such as buildings and urban areas, from historical topographic maps in the absence of training data, employing contemporary geospatial data as auxiliary data to guide the collection of training samples. As well as using neural networks (Meicen Song et al., 2020) semantic image segmentation, also allows the extraction of human settlement patterns in a geospatial vector data format to be ready for analysis in detecting a trajectory. What is important about these methods is that they integrate information from the robot's internal sensors (e.g., odometry, gyroscopes, accelerometers, etc.) with information from exteroceptive sensors (cameras, lasers, sonars, etc.), all to develop strategies to implement in an AGV.

However, as previously mentioned, a test of AVG is usually costly and high risk at the time of a real-life implementation. For this reason, the need to include simulations in the design cycle of autonomous driving systems arises. Currently, numerous simulation tools offer a variety of options to perform specific tests. These simulators make it possible to create atypical scenarios, which rarely happen in the real world, and test them without the safety risks and high costs of physical testing. Some of these tools are open-source, cross-platform, and customized, and others have aspects of a physical implementation to further simplify the transition of virtual vs real-world scenarios.

To perform virtual tests on these systems, a degree of knowledge in various engineering fields is usually required because they are systems that integrate various hardware and software components, mechanical, electrical, and electronic, converging in a team that must perform satisfactorily in a specific area or workplace. Most of the autonomous driving simulators provide rapid prototyping, physical engines for realistic movements, and detection algorithms. The most of recent simulators are elaborated in Unity or Unreal Engine (Pilz et al., 2019; Tong et al., 2020) platforms, because they are realistic 3D rendering and, dynamic for various programming languages such as Python, Java, C, C++, etc.

Within these simulators, they allow the implementation of different types of sensors that serve for testing before the project reaches the stage where higher costs are created in prototypes or production vehicles. The earlier a problem is identified, the more economical it will be to solve it and it will not delay production (Pilz et al., 2019; Meicen Song et al., 2020; Zhu et al., 2019; Chen et al., 2015; Nacu et al., 2018). Table 2 shows some simulators used by different authors in which the possibility of implementing sensors to an AGV is indicated, in this case, the symbol **x** means that it contains the sensors, and **-** means that it is not known information (Xiong et al., 2017; Raj y Sreekanth, 2017; Gang et al., 2020; tao Zhang et al., 2021; Li et al., 2019; Sushmitha et al., 2019; Nguyen y Nguyen, 2019; Salem et al., 2021; Kamar et al., 2020; Rong et al., 2020; Miura y Azumi, 2020; Wang et al., 2020c; Niemirepo et al., 2019; Stevic et al., 2020).

Table 2: Types of simulators and sensors they implement.

	VIRTUAL SENSORS						
	Camera	Lidar	Radar	IMU	GPS	Others	Ref
Prescan	x	-	x	-	x	Ultrasonic, laser, infra-red and an-tennas for Communi-Cation	[78], [79]
Dspace	x	-	x	-	-	Ultrasonic, mono stereo and Fisheye Cameras	[10]
Ansys	x	x	x	-	-	HiL connecti-vity and MiL connectivity	[80]
Rfpro	x	x	x	-	-	Training AI	[74], [75]
Cognata	x	x	x	-	-	Wide angle and fisheye cameras, noise models	[74], [75]
Carsim	x	x	x	-	-	Ultrasonic	[76], [81], [82]
Carmaker	x	x	x	-	-	Ultrasonic	[83], [84]
Gazebo	x	x	x	x	x	Torque of a sensor	[85]
Issac sim (Nvidia)	x	x	x	x	-	Support ROS	[74], [75]
Torcs	x	-	-	-	-	OpenGL	[86], [87]
Deepdriving	x	-	-	-	-	UnrealEngine Python	[77]
Lgsvl	x	x	x	x	x	Ultrasonic	[88]–[90]
Airsim	x	x	-	x	x	Distance sensor	[91]
Carla	x	x	-	x	x	Depth sensors	[92], [93]
Apollo	x	x	x	x	x	Support ROS	[17]
Morse	x	-	-	-	x	Laser scanner, odometry, actuators	[94]
PTV vissim	x	-	-	-	-	Reduction of Environmental impact	[95]
Unikie	x	x	x	-	-	Sensor fusion	[96]
Syncity	x	x	x	x	x	Optical flow, infrared, thermal, ultrasonic	[74], [75]

With these simulators a great amount of money and time is saved for each design change, authors such as Buhet (Buhet et al., 2019a) show complex urban situations that may arise, as road user positions, road geometry, and traffic light states are used to produce vehicle trajectories and control a real vehicle. Gang (Gang et al., 2020) performs an analysis of engine failures in high-risk situations. S.H. Wang (Wang et al., 2020c) implements motorcycles where he makes detection software to avoid accidents. C. Chen (Chen et al., 2015) proposes to map an input image to a small number of key perception indicators that are directly related to the accessibility of a road/traffic, obtaining to train a neural network, enabling autonomous driving. Laoufi (Laoufi et al., 2019) performs control of the dynamics of speed responses and flow, as well as on the potential of the fuzzy PI regulator of self-adjustment to rule out disturbances in an environment. These authors and many more have implemented these tools using them as an advantage when developing different algorithms for navigation and implementation of sensors. These simulators provide an opportunity for research extension, as they allow for early optimization, and avoid high costs and less delay in the execution of AGV.

4. Discussion

This section discusses and answers the proposed research questions, at the beginning of the systematic review.

QP1: Which sensors are best suited to be used in an autonomous vehicle navigation app, based on accuracy, price, and size ratio?

To answer this question, 21 works were chosen from the database that implement 15 different sensors to perform different navigation tasks. Using the information from the works, the sensors were subjectively classified in terms of accuracy, price, and size on a scale of high, medium, and low (Figure 4). Analyzing the figure, it is possible to verify that topological navigation can be developed using low and medium-cost sensors such as cameras, odometry, ultrasound, infrared, and radiofrequency, among others. Regarding the application of sensors, 50% of these works implement cameras to detect the environment and view the system, and 20% use lidar sensors to determine the distance to an object or surface, which allows for the recognition of the environment. The works highlight that sensors must be accurate when using navigation algorithms.

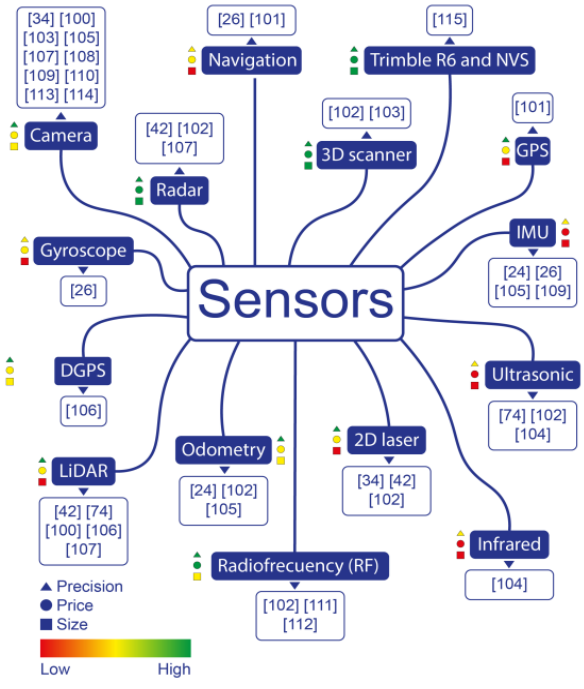

Figure 4: Concept map of sensors implemented for autonomous navigation.

On the other hand, it was also found that not all authors use the same processing system (microcontrollers), as they first establish what requirements they need, whether response time,

computational, cost and power. Some of these microcontrollers are low cost like: Rock Pi, Raspberry Pi, Jetson, Smartfly Tech Asus Tinker Board, among others (Henriksen et al., 2016; Yang y Lan, 2018; Ojanpera et al., 2021; Vijitkunsawat y Chantngarm, 2020; Tao et al., 2021; Pradhan et al., 2020).

QP2: What are the most convenient topological navigation approaches based on image processing integration for an application in autonomous vehicles, considering their advantages and disadvantages?

Fundamentally, when implementing technologies with autonomous navigation, the identification of an environment is needed. According to several authors (Yang et al., 2020; Bitar et al., 2020; Adams y Franzosa, 2007; Luo y Shih, 2018; Xiaoying et al., 2021), they say that images are important as they allow early detection of an object or scene. An alternative currently used is the digital image processing system to perform the detection and recognition of objects in the workspace. These perception systems allow the robot to have a greater degree of autonomy and help in making decisions that depend on the physical environment where it is.

Moreover, topological Navigation using image processing is a strategy, which describes relationships between features in an environment. These features are usually either symbols or recognizable objects in space. The objects to be detected must be easy to identify since these landmarks of one or more perceptually distinctive or interesting features are the ones that allow good navigation, since, by their color, shape, texture, or position, they use landmarks for the system orientation.

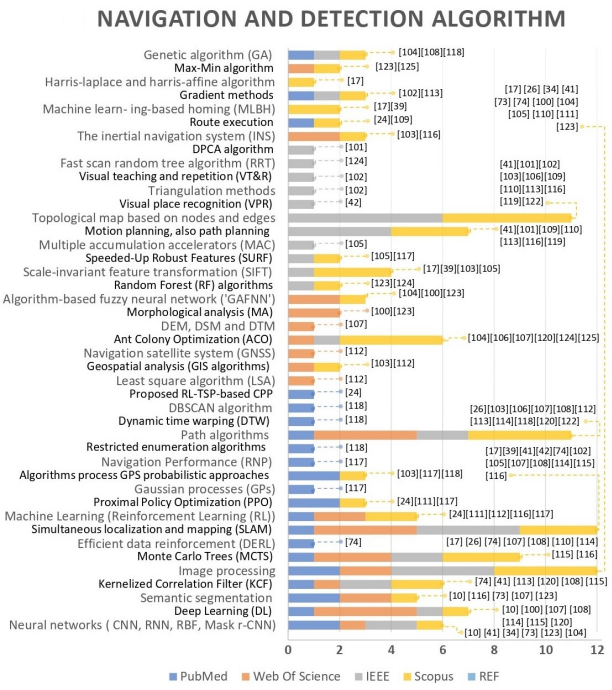

Figure 5: Topological navigation algorithms and other methodologies, implemented by different authors.

Through the fusion of topological navigation and image processing, it is possible to obtain algorithms with a greater perception of a scene, since image processing allows detecting and navigation shows topological parameters according to properties that depend on a specific geometry.

When using digital image processing you must know the environment where the characterization is going to be performed, you can define a distance and then use geometry or distance approximation methods. Based on this, the use of topological navigation provides a mechanism to perform integrity checks on your data and helps you to validate and maintain better representations of the entities in navigation; just as when one processes an image, its main objective is to identify the most representative data according to capture, then it has as a basis to perform a preprocessing, a localization of regions to identify the area of interest. The union of these systems can be defined as systems that provide geographic information about an environment or an object.

Through the collection of information from the databases, a detailed search was made of the advantages and disadvantages of the algorithms that implement topological navigation and image processing. In this context, this discussion covers the approaches used in navigating and recognizing the environment. Therefore, this discussion has been divided into two parts:

In the first part, the main trends, and correlations that the databases have were collected. Figure 5 shows a graph with the main trends, databases, and the number of works published with these algorithms in recent years. The IEEE stands out for having a greater number of implementations in topological navigation tools, image processing, simultaneous location and mapping (SLAM) and algorithms that allow identifying a trajectory and planning routes.

Therefore, it can be said that the review of the articles of the database carried out, more than 70% of the authors have had results related to navigation, regardless of the application, this shows that the search equation has an agreement with the proposed research topic (Pradhan et al., 2020; Buhet et al., 2019b; Seckin, 2020; Hbaieb et al., 2019; Raj y V G, 2021; Yoon y Raychowdhury, 2021; Zhou et al., 2020; Sameen et al., 2017; Xian et al., 2017; Prasetyo et al., 2021; Shi et al., 2021; Lee y Myung, 2019; Liu et al., 2017; Gines et al., 2019; Wang et al., 2020b; Gargees y Scott, 2020; Zhang et al., 2020a; Hedegaard et al., 2021a; Ozturk et al., 2020; Gao et al., 2019; Suzuki, 2022; Bi et al., 2019; Hamieh et al., 2020; Can et al., 2021; Lundberg, 2017; Santos et al., 2016).

In the second part, a summary and compilation of the main advantages and disadvantages that the algorithms present in the 109 articles of the 4 databases was made. The algorithms were grouped and divided into the sections Region-Based Framework, Methods for Locating Objects in Images, and Algorithms of Route Planning Implemented in Navigation.

4.1. Region-based framework

There are deep learning approaches that have been used intensively over the past decade to address some of the most challenging problems related to visual content, such as those related

Table 3: Regional-based framework, advantages, and disadvantages.

REGION PROPOSAL-BASED FRAMEWORKS (NEURAL NETWORKS)

These are algorithms that consist of simulating the behavior of a biological brain using thousands of interconnected artificial neurons that are stored in rows called layers, forming thousands of connections.

METHODS/ALGORITHMS	ADVANTAGES		DISADVANTAGES	REF
Neural networks	✓ High precision, with optimal and effective capture. ✓ Information processing is local (inputs and weights). ✓ Fault-tolerant. ✓ Recognize patterns that have not been learned (prior knowledge).		☒ They need a greater pre-processing of the data, being quite sensitive to the different scales of the variables. ☒ Long learning time.	[7][14] [33][37] [46][72] [77][83] [84] [97] [99] [103] [104][115] [118] [123][130] [134][137]
Region-based convolutional neural network (RCNN)	✓ Performance training in a single stage. ✓ The training updates the information of all the layers of the network. ✓ No disk storage is required for feature caching.	Train time (h): 84 Speedup: 1x Test time/image: 47.0s Test speedup: 1x Map: 66%	☒ Is a slow and time-consuming process (Search algorithm). ☒ High computing cost.	[14][23] [46][72] [77][83] [99] [103] [104][109] [115][118] [123][130] [134][137]
Fast R-CNN / Faster R-CNN	✓ Fast training and efficient backpropagation ✓ Faster than RCNN ✓ Single-stage training, using a multi-task loss ✓ No disk storage is required for feature caching	Train time (h): 9.5 Speedup: 8.8x Test time/image: 0.32s Test speedup: 146x Map: 66.9%	☒ High computing cost ☒ It takes about 2 seconds per image to detect objects, which sometimes doesn't work properly with large real-life data sets. ☒ Using an external candidate region generator creates a bottleneck in the discovery process.	[14][23] [46][72] [77][83] [99][103] [104][109] [115][118] [123][130] [134][137]
Spatial pyramid pooling (SPP-NET)	✓ It can produce a fixed-size output with any input size, while the grouping of sliding windows in the deep network above cannot ✓ Use multi-tier containers ✓ Improves classification accuracy by training the network on images with different scales	Train time (h): 25 Speedup: 3.4x Test time/image: 2.3s Test speedup: 20x Map: 63.1%	☒ The process is slow ☒ The process is carried out in stages, the feature vector needing to be written to disk, and the CNN parameters being unable to be backpropagated during training	[14][109] [137]
Mask-RCNN/ Mask branch	✓ The loss function for the model is the total loss when performing the classification ✓ Predicts the class of the object, refines the bounding box, and generates a pixel-level mask of the object based on the proposal from the first stage		☒ Presents small failures in real-time applications	[14][72] [99][115] [137]

to objects (identification and tracking) (Li et al., 2018b). Recently, deep learning has significantly improved performance in several image processing techniques (DarwishAlzughaibi et al., 2015), and object detectors with convolutional neural networks (CNNs) have reached precision status.

However, there are also other object detection techniques (Joo et al., 2020; Eriksen y Frandsen, 2018; Yurtsever et al., 2020; Ortiz et al., 2020; co et al., 2020; Patruno et al., 2021; Gonzalez et al., 2016; Mohamed et al., 2019; Routray et al., 2017; Hedegaard et al., 2021b; Vazquez Mendez y Casal Urcera, 2016; Rosu et al., 2019; Luo et al., 2021; Zhang, 2018; Law y Deng, 2018; Tahmid, 2020). In this case, the following Table 3 shows some methods that are used with their respective advantages and disadvantages, taking into account that it varies according to the need of usuary.

Neural networks learn through a set of input data and have great learning complexity for large tasks that involve the interpretation of region or some object. The data compiled in Table 3 show the necessary requirements of the algorithms, the computational time, and the results obtained in the image processing. This information can help researchers in choosing the method to be used in their applications.

As seen in Table 3 algorithms such as Faster-RCNN (Faster-Region-based convolutional neural network) and R-FCN (Region-based Fully Convolutional Network) are very accurate, depending on the feature extractor they can be used especially in video preprocessing, because of their ability to detect small objects. Although they do not reach speeds that allow them to be used in real-time systems, they are very efficient in detecting possible threats in images unlike learning in the detection step can significantly improve the performance of a tracking algorithm. CNN is essential in feature extraction: the use of appearance features is also fundamental to a good tracker and CNNs are particularly effective in their extraction. Comparing all the pre-

Table 4: Methods for locating objects in images.

METHODS FOR OBJECT LOCALIZATION IN IMAGES
Object tracking based on deep neural networks

METHODS/ALGORITHMS	ADVANTAGES	DISADVANTAGES	REF
Recurrent YOLO(ROLO)	✓ YOLO and LSTM network implementation (desired route). ✓ It is stable when it comes to blurring.	☒ High computing cost.	[20][70][109]
Hough line transform/ histograms of oriented gradients (HOG)	✓ It is robust to different lighting conditions, changes in the contour of the image, and background differences. ✓ Resistance to noise.	☒ High computational cost is required for the realization of calculations and storage of data.	[14][39][104]
Single-shot multibox (SSD)	✓ Use the SSD300 model at 60fps which is fast and accurate. ✓ Delivery in a single step the input image to the network. ✓ The graph compares SSD to an input sequence of 300x300 (59 fps), with an accuracy of 74.3%.	☒ The SSD512 is somewhat slow (22 fps) and often has trouble detecting small objects. ☒ The SSD300 despite being faster is less accurate than YOLO.	[14][84]
You only look once (YOLO)	✓ Fast and accurate ✓ Detection in real-time. ✓ Calibration without retraining (adjustment) ✓ Higher speed: the computational cost of YOLO with OpenCV is much lower. Each image can be processed in around 220ms versus the 3.2sg it takes on Darknet. ✓ Input speed of 448x448 (45 fps) has an accuracy of 63.4%.	☒ The algorithm is limited in detecting objects presented in groups and small things. ☒ It presents problems in the generalization of objects displayed in different configurations and the process of post-processing errors, treating all errors with the same level of importance.	[23][71][109]
Haar cascade classifier (Viola-Jones object)	✓ Efficient feature selection. ✓ Scale-invariant and localization. ✓ Using Haar filters instead of concordance points. ✓ The training type can be used to detect different objects.	☒ More efficient in images of front faces. ☒ Sensitive to lighting changes. ☒ The sliding cascade can get multiple detections of the same object. ☒ The small number of false positives needs learning.	[14][104]
Holdout method	✓ Only works with the data in the training set, it's very fast to perform. ✓ Fast and accurate.	☒ Is not very demanding in terms of computing power. ☒ Different data sets generate very different results.	[136][137]
K-fold cross-validation	✓ Great precision thanks to combinations created by subsets.	☒ The process is slow.	[136][137]
TLD (Tracking, learning, and detection)	✓ You can use Kalman filters and Markov models to significantly improve the accuracy of the algorithm.	☒ The high rate of false positives. ☒ The process is slow.	[23][31] [97][99][104] [107][109][134]
Kernelized correlation filters (C. Kcf tracker)/ particle filter	✓ Suitable to improve both the speed and accuracy of detection at the same time.	☒ This algorithm cannot retrieve the position of the object of interest, after the total occlusion of the image.	[14][23][33] [71][74] [99][104][109] [114][130]
Edge boxes	✓ It reduces the physical distance that data must travel, and the time required to move.	☒ Complexity and computational cost when recognizing boxes.	[23][41]
Goturn tracker/optimization	✓ It works with RCNN. ✓ Its execution speed is high.	☒ It does not handle the occlusion of the object to follow well.	[107] [108]
Semantic segmentation/SLAM	✓ Allows determining boundaries of each object and its location on the image plane.	☒ Mask RCNN is quite slow and prevents many real-time applications from being used.	[9][14][33] [39][41][74] [103] [105] [107][108] [120]

viously proposed methods, Fast R-CNN/Faster R-CNN would be a good choice as it has a faster training time and guarantees a mapping of 66.9%.

4.2. Methods for locating objects in images (Based on deep neural network tracking)

This section is focused on algorithms that have evolved in such a way that they allow to detect with greater precision and robustness objects, in addition to being able to extract characteristics of the image, making it possible to compare points of interest of a group of images to determine the coincidence of objects. Table 4 shows the methods highlighted by different authors.

Analyzing the advantages and disadvantages of the methods in Table 4, it can be seen that YOLO and SSD (Single shot multibox) are generally faster than two-stage detectors because they prevent preprocessing algorithms from making predictions with fewer candidate regions and make the classification subquadratic completely convolutional. According to H. Law and J. Deng (Law y Deng, 2018), SSD algorithms are considerably faster than those based on region proposals, although they have difficulty detecting small objects, especially if they are clustered. In addition, if YOLO detection is faulty, due to motion blur, object tracking remains, just like using HAAR cascade classifier. Many of the algorithms presented in Table 4 use the OpenCV library which helps with motion detection, object recognition, 3D reconstruction from images, among other functionalities.

4.3. Algorithms of more relevant route planning implemented in navigation

Regarding path planning techniques, Table 5 presents the advantages and disadvantages of several algorithms such as SIFT (Scale-invariant feature transform), SURF (speeded up robust

Table 5: Algorithms of more relevant route planning implemented in navigation, advantages, and disadvantages.

ALGORITHMS IMPLEMENTING PLANNING TECHNIQUES

METHODS/ ALGORITHMS	ADVANTAGES	DISADVANTAGES	REF
Algorithm A*	✓ The search is heuristic and reduces calculation time. ✓ Calculates least-cost routes on a weighted chart. ✓ Estimates the lowest cost.	☒ The resulting path is not continuous, and the heuristic rule is sometimes difficult to find.	[11][12] [31][51] [56][88] [106][113] [116][119] [125]
State lattices	✓ Considers motion and detection uncertainty on a trajectory. ✓ Capable of handling various dimensions.	☒ High computing cost. The planner only has a full resolution (lattice discretization).	[99][117] [131]
Line Drawing Algorithms DDA	✓ Generates points between coordinates. ✓ It's easy to implement. ✓ It ensures a shorter path.	☒ There is additional overhead when using the round function. ☒ The resulting lines are not smooth. ☒ The points are not accurate.	[118][119] [120]
Cornu or Euler spiral (Clotoide)	✓ It has the property that its curvature at any point is proportional to the distance along the curve measured from the origin. ✓ Its radius is infinite and is suitable for local planning.	☒ Natural parameterization complicates a coordinate-based definition, such as curve fitting or interpolation.	[124][132]
Dijkstra's algorithm	✓ It builds paths in increasing order of path length, suitable for overall planning in structures and unstructured environments. ✓ Explore all possible paths.	☒ It is slow and the resulting path is not continuous and is not recommended for real-time applications.	[31] [106][113] [116][119] [122][125]
SIFT (Scale-invariant feature transform)	✓ The descriptors are calculated through the orientation of the gradients of each point. ✓ It is stable to the transformation of the angle of view and noise. ✓ It is invariant in rotation and scale.	☒ It is slow and the ability to extract feature points from soft edges is weak. ☒ High computing cost and mathematically complicated. ☒ It generally doesn't work well with lighting and blur changes.	[14][26] [103][116] [130]
IFC Semantics and Citygml	✓ Generates a city model in 3D multi-resolution. ✓ It has semantic information according to census tracts and points of interest.	☒ It is not an effective method to generate a city in 3D. Its usefulness lies in creating certain buildings with a high level of detail.	[99][116] [130]
SURF (Speeded up robust features)	✓ Extract points of interest in image recognition. ✓ An excellent method for matching objects, faster than SIFT.	☒ Limits the number of descriptors that can be used and rejects much information from the image. ☒ SURF is not only invariant to scale and rotation.	[14][26] [103][116] [130]
D*, AEB, and AES	✓ It saves calculation time in planning. ✓ D* is very effective for replanning in the context of navigation. ✓ They work for Autonomous Emergency Steering and Autonomous Emergency Braking.	☒ It is slow in executing.	[8][106] [113][125]
Anytime Dynamic A *(AD *) Rapidly- exploring random tree (RRT)	✓ Combines replanning and search at any time. ✓ Based on quick scan trees	☒ High computing cost. ☒ Sometimes returns an erroneous trajectory.	[11][31] [33][56] [106][112] [113][124] [125][130]
Monte Carlo tree	✓ Is a heuristic algorithm. ✓ Can operate effectively without any knowledge of the specific domain other than the end rules and conditions and can find its own moves and learn from them by playing random playouts.	☒ High computing cost ☒ Needs many iterations to be able to effectively decide the most efficient path. So, there's a little speed issue there.	[74][113] [120][130]

features), and Dijkstra's algorithm, among others. This information allows the analysis and choice of path planning algorithms that can be used in topological navigation, according to the characteristics of the problem.

A booming topic is also generative adversarial neural networks (GANs) which are a new way of using Deep Learning to generate images that resemble a real environment (Tahmid, 2020). That is, GANs is an unsupervised learning system in which two artificial bits of intelligence compete to achieve a goal, this can be applied to detect roads more efficiently. On the other hand, there are the neural networks CNN, RCNN, and Fast R-CNN, among others, which also allow the generation of regions of interest, classification, and detection of objects. The challenge is to develop algorithms that avoid perceptual aliasing, to perform robust navigation, in addition to improving the quality and effectiveness of data collection, and scene brightness, as these are key points for good detection and transmitting proper georeferencing. Also, one direction for such navigation implementations is to use deep learning and artificial intelligence methods, as well as Monte Carlo methods that implement metric maps based on the topology and navigation of the vehicle.

5. Conclusions

The systematic review made it possible to obtain a bibliometric network, which shows that it is a topic of progress and scientific interest, according to the results of the network. In addition, different research and study topics emerge, making it difficult to establish which is the best topological navigation technique. For this reason, the main advantages and disadvantages of some methods are described as image processing integration and topological navigation.

Once the 315 initial articles were entered, 109 were approved about the exclusion and inclusion criteria, which allowed obtaining the most interesting documents, with relevant and reliable information, of which 35% documented projects based on simulators and low-cost sensors, which achieve safe topological navigation with minimum requirements; from the

remaining 65%, some techniques and algorithms that can be implemented to perform accurate navigation.

In general, it can be established that the techniques and sensors used in autonomous topological navigation manage to be able to identify the location, mapping, and plan appropriate routes for the displacement of the vehicle, being these some of the main characteristics to be considered in the different research works. Since they are the main challenge for the improvement of autonomous navigation, always seeking the minimum error among the detection parameters.

It should be noted that for all the above mentioned, this research topic is still booming because, despite the great progress already made, some different applications and approaches can be established at the time of implementing this technology (topological navigation). Also, the evolution of simulators has allowed multiple tests in different environments, with different sensors, allowing modifications to the environment and system immediately and simply, allowing to use in the same test some of the main navigation sensors. Thus, achieving a timely and guaranteed comparison, without fear of committing an accident, and allowing it to be more efficient, safe, and reliable in the implementation of these systems.

Likewise, a relationship with topological navigation based on image processing was found, since this integration allows the detection of information of interest, which helps the vehicle to be autonomous when interacting with the environment in which it is located, allowing the implementation of navigation in which the components of the scene can be established. In addition, 50% of these authors implement cameras for the detection of the environment and vision of the system, 20% use lidar sensors to determine the distance of an emitting device to an object or surface, which allows the recognition of the environment.

As future work, according to the results of this review, it is possible to identify that the simulators will allow establishing an environment of sensor simulation and development of validation tests where road detection codes and topological navigation can be implemented. This will allow obtaining data to measure and analyze specific areas of land using geographic information systems software, thus achieving approximate information of areas, lengths, basins, geological formations, location, detection, among others.

Finally, currently, autonomous vehicles have evolved in different areas (electronics, automotive, mechanics, engineering, etc) that have contributed to an advance in applied techniques. However, there is a little generation of competencies or qualities in the topological navigation techniques that allow establishing which are the best or worst efficiency, with which it is possible to establish how the future will be in the area of topological navigation of vehicles. As a result of this structured review, the advantages and disadvantages of different algorithms are evidenced, and it is also concluded that if a good algorithm is obtained, a better accuracy, interaction with the environment, reading efficiency, among other characteristics, can be guaranteed. Likewise, there remains the great challenge of improving algorithms and techniques of Deep Learning and artificial intelligence, which are currently research and improvement trends, accompanied by techniques such as GANS, deep networks, particle filters, Monte Carlo method, among others.

Regarding the path planning techniques, the information presented provides a knowledge base for the development of the necessary intelligence so that an autonomous vehicle can use algorithms that allow establishing topological navigation in the environment in which it is located, this is a relevant contribution from according to the search methodology implemented in this article.

In addition, one of the big challenges is to get the right software according to the implementation of real-world data collection, precision motion control, false-positive reduction, easy installation, and implementation. In this process, it is essential to consider different stages, such as the localization part, the robot must always know where it is, in mapping the robot must know how to build a representation of its environment or map, in planning it must know how to calculate the best way to reach its destination and finally it must be able to follow the calculated route avoiding obstacles.

ACKNOWLEDGMENT

This study was financed in part by Vale and in part by UNIFEI.

References

Adams, C., Franzosa, R., 2007. Introduction to topology: Pure and applied. undefined.

Aldibaja, M., Suganuma, N., Yanase, R., Yoneda, K., 2020. Reliable graph-slam framework to generate 2d lidar intensity maps for autonomous vehicles. IEEE Vehicular Technology Conference 2020-May.
DOI: 10.1109/VTC2020-Spring48590.2020.9129063

Arroyo, R., Alcantarilla, P. F., Bergasa, L. M., Romera, E., 2016. Openable: An open-source toolbox for application in life-long visual localization of autonomous vehicles. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, 965–970.
DOI: 10.1109/ITSC.2016.7795672

Benson, A., Tefft, B., Svancara, A.M.Horrey, W.L, 2018. Potential reduction in crashes, injuries and deaths from large-scale deployment of advanced driver assistance systems - aaa foundation for traffic safety.
URL: https://aaafoundation.org/potential-reduction-in-\crashes-injuries-and-deaths-from-large-scale-deployment-\of-advanced-driver-assistance-systems/

Bi, Z., Ma, Y., Yang, X., Zhou, P., 2019. Research on wireless robot path planning under edge computing considering multistep searching and inflection points. Transactions on Emerging Telecommunications Technologies.
DOI: 10.1002/ett.3825

Bitar, G., Martinsen, A. B., Lekkas, A. M., Breivik, M., 2020. Two-stage optimized trajectory planning for asvs under polygonal obstacle constraints: Theory and experiments. IEEE Access 8, 199953–199969.
DOI: 10.1109/ACCESS.2020.3035256

Blue, S. T., Brindha, M., 2019. Edge detection based boundary box construction algorithm for improving the precision of object detection in yolov3, 1–5.
DOI: 10.1109/ICCCNT45670.2019.8944852

Buhet, T., Wirbel, E., Perrotton, X., 2019a. Conditional vehicle trajectories prediction in carla urban environment. Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019, 2310–2319.
DOI: 10.1109/ICCVW.2019.00284

Buhet, T., Wirbel, E., Perrotton, X., 2019b. Conditional vehicle trajectories prediction in carla urban environment, 2310–2319.
DOI: 10.1109/ICCVW.2019.00284

Burger, P., Naujoks, B., Wuensche, H. J., 2019. Map-aware slam with sparse map features. IEEE International Conference on Intelligent Robots and Systems, 347–353.
DOI: 10.1109/IROS40897.2019.8968466

Cai, S., Wan, Y., 2020. Topometric imitation learning for route following under appearance change. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops 2020-June, 4354–4362.
DOI: 10.1109/CVPRW50498.2020.00513

Camargo, A. B., Liu, Y., He, G., Zhuang, Y., 2019. Mobile robot autonomous exploration and navigation in large-scale indoor environments. 10th International Conference on Intelligent Control and Information Processing, ICICIP 2019, 106–111.
DOI: 10.1109/ICICIP47338.2019.9012209

Can, Y. S., Gerrits, P. J., Kabadayi, M. E., 2021. Automatic detection of road types from the third military mapping survey of austria-hungary historical map series with deep convolutional neural networks. IEEE Access 9, 62847–62856.
DOI: 10.1109/ACCESS.2021.3074897

Chen, C., Seff, A., Kornhauser, A., Xiao, J., 2015. Deepdriving: Learning affordance for direct perception in autonomous driving. Proceedings of the IEEE International Conference on Computer Vision 2015 Inter, 2722–2730.
DOI: 10.1109/ICCV.2015.312

Chen, Q., Lu, Y., Wang, Y., Zhu, B., 2021. From topological map to local cognitive map: a new opportunity of local path planning. Intelligent Service Robotics 14, 285–301.
URL: https://doi.org/10.1007/s11370-021-00352-z
DOI: 10.1007/s11370-021-00352-z

Choi, J., Park, J., Jung, J., Choi, H. T., 2019. Gaussian sum filter based slam for autonomous navigation of underwater vehicles using acoustic sources. OCEANS 2019 - Marseille, OCEANS Marseille 2019 2019-June, 2019–2022.
DOI: 10.1109/OCEANSE.2019.8867302

Choi, J., Park, J., Jung, J., Lee, Y., Choi, H. T., 2020. Development of an autonomous surface vehicle and performance evaluation of autonomous navigation technologies. International Journal of Control, Automation and Systems 18, 535–545.
DOI: 10.1007/s12555-019-0686-0

co, P. L., Guerreiro, B. J., Batista, P., Oliveira, P., Silvestre, C., 8 2020. Earth-fixed trajectory and map online estimation: Building on ges sensor-based slam filters. Robotics and Autonomous Systems 130, 103552.
DOI: 10.1016/J.ROBOT.2020.103552

Daniusis, P., Juneja, S., Valatka, L., Petkevicius, L., 2021. Topological navigation graph framework. Autonomous Robots.
DOI: 10.1007/s10514-021-09980-x

DarwishAlzughaibi, A., Hakami, H. A., Chaczko, Z., 7 2015. Review of human motion detection based on background subtraction techniques. International Journal of Computer Applications 122, 1–5.
DOI: 10.5120/21757-4988

Dwi Fitria Al Husaeni, A. B. D. N., 2021. Bibliometric using vosviewer with publish or perish (using google scholar data): From step-by-step processing for users to the practical examples in the analysis of digital learning articles in pre and post covid-19 pandemic.

Emmi, L., Flecher, E. L., Cadenat, V., Devy, M., 2021. A hybrid representation of the environment to improve autonomous navigation of mobile robots in agriculture. Precision Agriculture 22, 524–549.
URL: https://doi.org/10.1007/s11119-020-09773-9
DOI: 10.1007/s11119-020-09773-9

Eriksen, M. B., Frandsen, T. F., 10 2018. The impact of patient, intervention, comparison, outcome (pico) as a search strategy tool on literature search quality: a systematic review. Journal of the Medical Library Association : JMLA 106, 420.
URL: /pmc/articles/PMC6148624//pmc/articles/PMC6148624/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC6148624/
DOI: 10.5195/JMLA.2018.345

Feraco, S., Luciani, S., Bonfitto, A., Amati, N., Tonoli, A., 2020. A local trajectory planning and control method for autonomous vehicles based on the rrt algorithm. 2020 AEIT International Conference of Electrical and Electronic Technologies for Automotive, AEIT AUTOMOTIVE 2020, 1–6.
DOI: 10.23919/aeitautomotive50086.2020.9307439

Francisco Borja, A. F., Arnau, S., 11 2016. Navegacion automatica en un vehiculo electrico ligero con distribucion ackermann. Universitat Politecnica de Valencia.
URL: https://riunet.upv.es/handle/10251/74308

Gang, L., Pingshu, G., Jiaqi, Z., Tao, Z., Kailan, Z., Junjie, L., 2020. Research on failure mechanism for distributed drive vehicle based on co-simulation of carsim and matlab. 2020 4th CAA International Conference on Vehicular Control and Intelligence, CVCI 2020, 530–535.
DOI: 10.1109/CVCI51460.2020.9338637

Gao, F., Wang, L., Zhou, B., Han, L., Pan, J., Shen, S., 7 2019. Teach-repeat-replan: A complete and robust system for aggressive flight in complex environments.
URL: https://arxiv.org/abs/1907.00520v1

Gargees, R. S., Scott, G. J., 2020. Deep feature clustering for remote sensing imagery land cover analysis. IEEE Geoscience and Remote Sensing Letters 17, 1386–1390.
DOI: 10.1109/LGRS.2019.2948799

Gines, J., Martin, F., Vargas, D., RodrÃguez, F. J., MatellÃ¡n, V., 2019. Social navigation in a cognitive architecture using dynamic proxemic zones. Sensors (Switzerland) 19, 1–15.
DOI: 10.3390/s19235189

Gonzalez, D., Perez, J., Milanes, V., Nashashibi, F., 2016. A review of motion planning techniques for automated vehicles. IEEE Transactions on Intelligent Transportation Systems 17, 1135–1145.
DOI: 10.1109/TITS.2015.2498841

Gupta, A., Fernando, X., 2022. Simultaneous localization and mapping (slam) and data fusion in unmanned aerial vehicles: Recent advances and challenges. Drones 6 (4).
URL: https://www.mdpi.com/2504-446X/6/4/85
DOI: 10.3390/drones6040085

Hamieh, A., Makhlouf, A. B., Louhichi, B., Deneux, D., 2020. A bim-based method to plan indoor paths. Automation in Construction 113, 103120.
URL: https://doi.org/10.1016/j.autcon.2020.103120
DOI: 10.1016/j.autcon.2020.103120

Hbaieb, A., Rezgui, J., Chaari, L., 2019. Pedestrian detection for autonomous driving within cooperative communication system, 1–6.
DOI: 10.1109/WCNC.2019.8886037

He, W., Li, Z., Chen, C. L., 2017. A survey of human-centered intelligent robots: Issues and challenges. IEEE/CAA Journal of Automatica Sinica 4, 602–609.
DOI: 10.1109/JAS.2017.7510604

Hedegaard, B., Fahnestock, E., Arkin, J., Menon, A., Howard, T. M., 2021a. Discrete optimization of adaptive state lattices for iterative motion planning on unmanned ground vehicles, 5764–5771.
DOI: 10.1109/IROS51168.2021.9636181

Hedegaard, B., Fahnestock, E., Arkin, J., Menon, A., Howard, T. M., 2021b. Discrete optimization of adaptive state lattices for iterative motion planning on unmanned ground vehicles, 5764–5771.
DOI: 10.1109/IROS51168.2021.9636181

Henriksen, E. H., SchjÃ¸lberg, I., Gjersvik, T. B., 2016. The underwater modular open robot simulation engine. Autonomous Underwater Vehicles 2016, AUV 2016, 261–267.
DOI: 10.1109/AUV.2016.7778681

Huachao, Y., Jingyi, L., Xijun, Z., 2020. Optimal trajectory generation for discretized terminal conditions based on reference lines. Proceedings of 2020 3rd International Conference on Unmanned Systems, ICUS 2020, 484–489.
DOI: 10.1109/ICUS50048.2020.9274971

Huang, Z., Wang, J., Fu, X., Yu, T., Guo, Y., Wang, R., 6 2020. Dc-spp-yolo: Dense connection and spatial pyramid pooling based yolo for object detection. Information Sciences 522, 241–258.
DOI: 10.1016/J.INS.2020.02.067

Imad, M., Hassan, M. A., Junaid, H., Ahmad, I., 2021. Navigation system for autonomous vehicle : A survey journal of computer science and technology studies (jcsts) navigation system for autonomous vehicle : A survey.

Jaen, A. A., 2017. Diseño de infraestructura de nueva planta para la linea de producción de los modelos buller y linner 12 en dina camiones, 168.

Jia, C., Huang, M., Sui, L., 2020a. An autonomous vehicle motion planning method based on dynamic programming. 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing, ICCWAMTIP 2020, 394–398.
DOI: 10.1109/ICCWAMTIP51612.2020.9317333

Jia, H., Zhang, L., Wang, Z., 2020b. A dynamic lane-changing trajectory planning scheme for autonomous vehicles on structured road. 2020 IEEE 9th International Power Electronics and Motion Control Conference, IPEMC 2020 ECCE Asia, 2222–2227.
DOI: 10.1109/IPEMC-ECCEAsia48364.2020.9367649

Joo, S. H., Manzoor, S., Rocha, Y. G., Bae, S. H., Lee, K. H., Kuc, T. Y., Kim, M., 2020. Autonomous navigation framework for intelligent robots based on a semantic environment modeling. Applied Sciences (Switzerland) 10, 1–30.
DOI: 10.3390/app10093219

Kamar, D., Akyol, G., Mertan, A., Inceoglu, A., 10 2020. Comparative analysis of reinforcement learning algorithms on torcs environment. 2020 28th Signal Processing and Communications Applications Conference (SIU), 1–4.
URL: https://ieeexplore.ieee.org/document/9302358/
DOI: 10.1109 / SIU49456.2020.9302358

Kazim, M., Azar, A. T., Koubaa, A., Ibrahim, Z. F., Zaidi, A., Zhang, L., 1 2021. Event-driven programming-based path planning and navigation of uavs around a complex urban environment. Unmanned Aerial Systems, 531–565.
DOI: 10.1016/B978-0-12-820276-0.00028-5

Kessler, T., Minnerup, P., Esterle, K., Feist, C., Mickler, F., Roth, E., Knoll, A., 2018. Roadgraph generation and free-space estimation in unknown structured environments for autonomous vehicle motion planning. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC 2018-Novem, 2831–2838.
DOI: 10.1109/ITSC.2018.8569306

Kim, M. S., Han, J., Lee, J., 2017. Optimal trajectory control for capturing a mobile sound source by a mobile robot. International Journal of Humanoid Robotics 14, 567–570.
DOI: 10.1142/S0219843617500256

Laoufi, C., Abbou, A., Akherraz, M., 2019. Dspace improved direct torque control of induction motor using fuzzy logic self-tuning proportional integral regulator for electric vehicle propulsion chain. Proceedings of the 2nd International Conference on High Voltage Engineering and Power Systems: Towards Sustainable and Reliable Power Delivery, ICHVEPS 2019.
DOI: 10.1109/ICHVEPS47643.2019.9011038

Law, H., Deng, J., 2018. Cornernet: Detecting objects as paired keypoints. European Conference on Computer Vision - ECCV, 765–781.

Le, A. V., Veerajagadheswar, P., Kyaw, P. T., Elara, M. R., Nhan, N. H. K., 2021. Coverage path planning using reinforcement learning-based tsp for htetran â a polyabolo-inspired self-reconfigurable tiling robot. Sensors 21.
DOI: 10.3390/s21082577

Lee, K., Kum, D., 2020. Longitudinal and lateral integrated safe trajectory planning of autonomous vehicle via friction limit. International Conference on Control, Automation and Systems 2020-Octob, 1177–1180.
DOI: 10.23919/ICCAS50221.2020.9268305

Lee, Y. C., Myung, H., 2019. Hierarchical sampling optimization of particle filter for global robot localization in pervasive network environment. ETRI Journal 41, 782–796.
DOI: 10.4218/etrij.2018-0550

Leng, C., Zhang, H., Li, B., Cai, G., Pei, Z., He, L., 2019. Local feature descriptor for image matching: A survey. IEEE Access 7, 6424–6434.
DOI: 10.1109/ACCESS.2018.2888856

Li, G., Bao, H., Wang, B., Wu, T., 2018a. Kernelised rényi distance for localization and mapping of autonomous vehicle. Proceedings - 13th International Conference on Computational Intelligence and Security, CIS 2017 2018-Janua, 69–72.
DOI: 10.1109/CIS.2017.00023

Li, G., Chen, J., Wang, K., Long, Y., Xiong, Z., 2019. A model evaluating complexity of driving environment toward autonomous vehicles using neural network. En: 2019 12th International Symposium on Computational Intelligence and Design (ISCID). Vol. 1. pp. 213–216.
DOI: 10.1109/ISCID.2019.00055

Li, J. H., Park, D. G., Ki, H. S., 2018b. Sonar image processing based underwater localization and path planning for auv's autonomous swimming. 2018 5th International Conference on Control, Decision and Information Technologies, CoDIT 2018, 611–615.
DOI: 10.1109/CoDIT.2018.8394810

Liu, J., Hyyppa, J., Yu, X., Jaakkola, A., Kukko, A., Kaartinen, H., Zhu, L., Liang, X., Wang, Y., Hyyppa, H., 2017. A novel gnss technique for predicting boreal forest attributes at low cost. IEEE Transactions on Geoscience and Remote Sensing 55, 4855–4867.
DOI: 10.1109/TGRS.2017.2650944

Lundberg, M., 2017. Path planning for autonomous vehicles using clothoid based smoothing of a* generated paths and optimal control, 68.

Luo, J. Q., sheng Fang, H., ming Shao, F., Zhong, Y., Hua, X., 8 2021. Multi-scale traffic vehicle detection based on faster râcnn with nas optimization and feature enrichment. Defence Technology 17, 1542–1554.
DOI: 10.1016/J.DT.2020.10.006

Luo, R. C., Shih, W., 2018. Autonomous mobile robot intrinsic navigation based on visual topological map. IEEE International Symposium on Industrial Electronics 2018-June, 541–546.
DOI: 10.1109/ISIE.2018.8433588

Luo, R. C., Shih, W., 2019. Topological map generation for intrinsic visual navigation of an intelligent service robot. 2019 IEEE International Conference on Consumer Electronics, ICCE 2019, 1–6.
DOI: 10.1109/ICCE.2019.8662062

Martini, D. D., Gadd, M., Newman, P., 2020. kradar++: Coarse-to-fine fmcw scanning radar localisation. Sensors (Switzerland) 20, 1–23.
DOI: 10.3390/s20216002

Mashoshin, Pashkevich, 2020. Application of passive underwater landmarks for autonomous unmanned underwater vehicles navigation. Gyroscopy and Navigation 11, 333–340.
DOI: 10.1134/S2075108720040070

Meicen Song, S., Chen, H., Sun, H., Liu, 2020. Data efficient reinforcement learning for integrated.

Miura, K., Azumi, T., 2020. Converting driving scenario framework for testing self-driving systems. Proceedings - 2020 IEEE 18th International Conference on Embedded and Ubiquitous Computing, EUC 2020, 56–63.
DOI: 10.1109/EUC50751.2020.00015

Mohamed, S. A., Haghbayan, M. H., Westerlund, T., Heikkonen, J., Tenhunen, H., Plosila, J., 2019. A survey on odometry for autonomous navigation systems. IEEE Access 7, 97466–97486.
DOI: 10.1109/ACCESS.2019.2929133

Mongus, D., Juric, S., 2019. Generation of traversability maps based on 3d point-clouds. 2nd International Conference on Next Generation Computing Applications 2019, NextComp 2019 - Proceedings.
DOI: 10.1109/NEXTCOMP.2019.8883547

Nacu, C. R., Fodorean, D., Husar, C., Grovu, M., Irimia, C., 2018. Towards autonomous ev by using virtual reality and prescan-simulink simulation environments. SPEEDAM 2018 - Proceedings: International Symposium on Power Electronics, Electrical Drives, Automation and Motion, 401–406.
DOI: 10.1109/SPEEDAM.2018.8445211

Nashed, S. B., Ilstrup, D. M., Biswas, J., 2018. Localization under topological uncertainty for lane identification of autonomous vehicles. En: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 6000–6005.
DOI: 10.1109/ICRA.2018.8461185

Nguyen, K. D., Nguyen, T. T., 2019. Vision-based software-in-the-loop-simulation for unmanned aerial vehicles using gazebo and px4 open source. Proceedings of 2019 International Conference on System Science and Engineering, ICSSE 2019, 429–432.
DOI: 10.1109/ICSSE.2019.8823322

Niemirepo, T., Toivonen, J., Viitanen, M., Vanne, J., 2019. Open-source cithrus simulation environment for real-time 360-degree traffic imaging. 2019 8th IEEE International Conference on Connected Vehicles and Expo, ICCVE 2019 - Proceedings.
DOI: 10.1109/ICCVE45908.2019.8965242

Ojanpera, T., Makela, J., Majanen, M., Mammela, O., Martikainen, O., Vaisanen, J., 2021. Evaluation of lidar data processing at the mobile network edge for connected vehicles. Eurasip Journal on Wireless Communications and Networking 2021, 83–88.
DOI: 10.1186/s13638-021-01975-7

Oleynikova, H., Taylor, Z., Siegwart, R., Nieto, J., 2018. Sparse 3d topological graphs for micro-aerial vehicle planning. IEEE International Conference on Intelligent Robots and Systems, 8478–8485.
DOI: 10.1109/IROS.2018.8594152

Organization, W. H., 2021. Road traffic injuries.
URL: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

Organization(OMS), W. H., 2017. Oms — 10 facts about road safety.
URL: https://www.who.int/home/cms-decommissioning

Ort, T., Paull, L., Rus, D., 2018. Autonomous vehicle navigation in rural environments without detailed prior maps. Proceedings - IEEE International Conference on Robotics and Automation, 2040–2047.
DOI: 10.1109/ICRA.2018.8460519

Ortiz, F. M., Sammarco, M., Costa, L. H. M. K., Detyniecki, M., 2020. Vehicle

telematics via exteroceptive sensors: A survey, 1–18.
URL: http://arxiv.org/abs/2008.12632

Ozturk, G., Koker, R., ElDogan, O., Karayel, D., 2020. Recognition of vehicles, pedestrians and traffic signs using convolutional neural networks. En: 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). pp. 1–8.
DOI: 10.1109/ISMSIT50672.2020.9255148

Palafox, P. R., Betz, J., Nobis, F., Riedl, K., Lienkamp, M., 2019. Semanticdepth: Fusing semantic segmentation and monocular depth estimation for enabling autonomous driving in roads without lane lines. Sensors (Switzerland) 19, 1–20.
DOI: 10.3390/s19143224

Park, D., Cho, K., Park, J., Cho, Y., 2011. IT Convergence and Services. Vol. 107.
URL: http://www.scopus.com/inward/record.url?eid=2-s2.0-81255197561&partnerID=tZOtx3y1

Parsifal, 2014. Parsifal.
URL: https://parsif.al/about/

Patruno, C., Reno, Vito, N., Summa, M., Nitti, M., 2021. A robust method for 2d occupancy map building for indoor robot navigation, 10.
DOI: 10.1117/12.2595511

Peralta, F., Arzamendia, M., Gregor, D., Reina, D. G., Toral, S., 2020. A comparison of local path planning techniques of autonomous surface vehicles for monitoring applications: The ypacarai lake case-study. Sensors (Switzerland) 20.
DOI: 10.3390/s20051488

Pilz, C., Steinbauer, G., Schratter, M., Watzenig, D., 2019. Development of a scenario simulation platform to support autonomous driving verification. 2019 8th IEEE International Conference on Connected Vehicles and Expo, ICCVE 2019 - Proceedings.
DOI: 10.1109/ICCVE45908.2019.8964914

Pradhan, B., Al-Najjar, H. A., Sameen, M. I., Mezaal, M. R., Alamri, A. M., 2020. Landslide detection using a saliency feature enhancement technique from lidar-derived dem and orthophotos. IEEE Access 8, 121942–121954.
DOI: 10.1109/ACCESS.2020.3006914

Prasetyo, E., Suciati, N., Fatichah, C., 2021. Yolov4-tiny and spatial pyramid pooling for detecting head and tail of fish, 157–161.
DOI: 10.1109/ICAICST53116.2021.9497822

Raj, A., Sreekanth, P. K., 2017. Modelling and simulation of srm based automatic transmission system for hybrid vehicles in ansys maxwell. Proceedings of IEEE International Conference on Circuit, Power and Computing Technologies, ICCPCT 2017.
DOI: 10.1109/ICCPCT.2017.8074301

Raj, M., V G, N., 2021. Deep neural network approach for navigation of autonomous vehicles. En: 2021 6th International Conference for Convergence in Technology (I2CT). pp. 1–4.
DOI: 10.1109/I2CT51068.2021.9418189

Rambhatla, S., Sidiropoulos, N. D., Haupt, J., 2018. Tensormap: Lidar-based topological mapping and localization via tensor decompositions, 1368–1372.

Rizk, M., Mroue, A., Farran, M., Charara, J., 2020. Real-time slam based on image stitching for autonomous navigation of uavs in gnss-denied regions. Proceedings - 2020 IEEE International Conference on Artificial Intelligence Circuits and Systems, AICAS 2020, 301–304.
DOI: 10.1109/AICAS48895.2020.9073793

Rong, G., Shin, B. H., Tabatabaee, H., Lu, Q., Lemke, S., MoÅ¾eiko, M., Boise, E., Uhm, G., Gerow, M., Mehta, S., Agafonov, E., Kim, T. H., Sterner, E., Ushiroda, K., Reyes, M., Zelenkovsky, D., Kim, S., 2020. Lgsvl simulator: A high fidelity simulator for autonomous driving. 2020 IEEE 23rd International Conference on Intelligent Transportation Systems, ITSC 2020.
DOI: 10.1109/ITSC45102.2020.9294422

Rosu, H. C., Mancas, S. C., Hsieh, C. C., 8 2019. Generalized cornu-type spirals and their darboux parametric deformations. Physics Letters A 383, 2692–2697.
DOI: 10.1016/J.PHYSLETA.2019.05.040

Routray, S., Ray, A. K., Mishra, C., 2 2017. Analysis of various image feature extraction methods against noisy image: Sift, surf and hog. 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT), 1–5.
URL: http://ieeexplore.ieee.org/document/8117846/
DOI: 10.1109 / ICECCT.2017.8117846

Salem, M., Mora, A. M., Guervos, J. J. M., 2021. Overtaking uncertainty with evolutionary torcs controllers: Combining blx with decreasing operator and grand prix selection. IEEE Transactions on Games 1502, 1–10.
DOI: 10.1109/TG.2021.3072417

Sameen, M. I., Pradhan, B., Shafri, H. Z., Mezaal, M. R., Hamid, H. B., 2017. Integration of ant colony optimization and object-based analysis for lidar data classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 10, 2055–2066.
DOI: 10.1109/JSTARS.2017.2650956

Santos, V. D. C., Osorio, F. S., Toledo, C. F., Otero, F. E., Johnson, C. G., 2016. Exploratory path planning using the max-min ant system algorithm. 2016 IEEE Congress on Evolutionary Computation, CEC 2016, 4229–4235.
DOI: 10.1109/CEC.2016.7744327

Seckin, A. C., 2020. A natural navigation method for following path memories from 2d maps. Arabian Journal for Science and Engineering 45, 10417–10432.
URL: https://doi.org/10.1007/s13369-020-04784-0
DOI: 10.1007/s13369-020-04784-0

Shi, L., Zhou, Z., Guo, Z., 2021. Face anti-spoofing using spatial pyramid pooling, 2126–2133.
DOI: 10.1109/ICPR48806.2021.9412407

Silva, S. P. P. D., Filho, P. H., Marinho, L. B., Almeida, J. S., Nascimento, N. M. M., Rodrigues, A. W. D. O., Filho, P. P. R., 2019. A new approach to navigation of unmanned aerial vehicle using deep transfer learning. Proceedings - 2019 Brazilian Conference on Intelligent Systems, BRACIS 2019, 222–227.
DOI: 10.1109/BRACIS.2019.00047

Song Xu, H. X., Chou, W., 2019. Visual topological mapping and navigation for mobile robot in large-scale environment. IEEE International Conference on Robotics and Biomimetics, ROBIO 2019, 2589–2594.
DOI: 10.1109/ROBIO49542.2019.8961726

Stevic, S., KruniÄ, M., , M. M. D., Kaprocki, N., 2020. Development of adas perception applications in ros and âsoftware-in-the-loopâ validation with carla simulator. Telfor Journal 12, 40–45.
DOI: 10.5937/TELFOR2001040S

Sui, L., Yu, H., Chen, X., Jia, C., Huang, M., 2020. Path planning based on clothoid for autonomous valet parking. 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing, ICCWAMTIP 2020, 389–393.
DOI: 10.1109/ICCWAMTIP51612.2020.9317391

Sur, C., 2019. Ucrlf: unified constrained reinforcement learning framework for phase-aware architectures for autonomous vehicle signaling and trajectory optimization. Evolutionary Intelligence 12, 689–712.
URL: https://doi.org/10.1007/s12065-019-00278-7
DOI: 10.1007/s12065-019-00278-7

Sushmitha, T. V., Deepika, C. P., Uppara, R., Sai, R. N., 2019. Vehicle trajectory prediction using non-linear input-output time series neural network. 1st International Conference on Power Electronics Applications and Technology in Present Energy Scenario, PETPES 2019 - Proceedings, 16–20.
DOI: 10.1109/PETPES47060.2019.9003797

Suzuki, Ryuki y Ji, Y. y. P. S. y. U. K., 2022. Indoor slam based on line observation probability using a hand-drawn map, 695–698.
DOI: 10.1109/SII52469.2022.9708610

Tahmid, M., 2020. Comparative analysis of generative adversarial networks and their variants, 19–21.

Tao, C., He, H., Xu, F., Cao, J., 10 2021. Stereo priori rcnn based car detection on point level for autonomous driving. Knowledge-Based Systems 229, 107346.
DOI: 10.1016/J.KNOSYS.2021.107346

tao Zhang, H., bin Hu, B., Xu, Z., Cai, Z., Liu, B., Wang, X., Geng, T., Zhong, S., Zhao, J., 2021. Visual navigation and landing control of an unmanned aerial vehicle on a moving autonomous surface vehicle via adaptive learning, 1–11.

Tong, K., Ajanovic, Z., Stettinger, G., 2020. Overview of tools supporting planning for automated driving. 2020 IEEE 23rd International Conference on Intelligent Transportation Systems, ITSC 2020.
DOI: 10.1109/ITSC45102.2020.9294512

Uhl, J. H., Leyk, S., Chiang, Y. Y., Duan, W., Knoblock, C. A., 2020. Automated extraction of human settlement patterns from historical topographic map series using weakly supervised convolutional neural networks. IEEE

Access 8, 6978–6996.
DOI: 10.1109/ACCESS.2019.2963213

Valencia, R., Andrade-Cetto, J., Porta, J. M., 2011. Path planning in belief space with pose slam. Proceedings - IEEE International Conference on Robotics and Automation, 78–83.
DOI: 10.1109/ICRA.2011.5979742

Vazquez Mendez, M., Casal Urcera, G., 08 2016. The clothoid computation: A simple and efficient numerical algorithm. Journal of Surveying Engineering 142, 04016005:1–9.
DOI: 10.1061/(ASCE)SU.1943-5428.0000177

Vijitkunsawat, W., Chantngarm, P., 6 2020. Comparison of machine learning algorithmâs on self-driving car navigation using nvidia jetson nano. undefined, 201–204.
DOI: 10.1109/ECTI-CON49241.2020.9158311

VOSviewer, 2022. Vosviewer - visualizing scientific landscapes.
URL: https://www.vosviewer.com/

Wang, C., Ma, H., Chen, W., Liu, L., Meng, M. Q., 12 2020a. Efficient autonomous exploration with incrementally built topological map in 3-d environments. IEEE Transactions on Instrumentation and Measurement 69, 9853–9865.
DOI: 10.1109/TIM.2020.3001816

Wang, C., Ma, H., Chen, W., Liu, L., Meng, M. Q., 2020b. Efficient autonomous exploration with incrementally built topological map in 3-d environments. IEEE Transactions on Instrumentation and Measurement 69, 9853–9865.
DOI: 10.1109/TIM.2020.3001816

Wang, D., Liang, H., Zhu, H., Zhang, S., 2014. A bionic camera-based polarization navigation sensor. Sensors (Switzerland) 14, 13006–13023.
DOI: 10.3390/s140713006

Wang, S. H., Lin, C. X., Tu, C. H., Huang, C. C. J., Juang, J. C., 12 2020c. Autonomous vehicle simulation for asia urban areas with a perspective from education. Proceedings - 2020 International Computer Symposium, ICS 2020, 454–458.
DOI: 10.1109/ICS51289.2020.00095

Wen, S., Zhao, Y., Liu, X., Sun, F., Lu, H., Wang, Z., 2020. Hybrid semi-dense 3d semantic-topological mapping from stereo visual-inertial odometry slam with loop closure detection. IEEE Transactions on Vehicular Technology 69, 16057–16066.
DOI: 10.1109/TVT.2020.3041852

Wheeler, D. O., Koch, D. P., Jackson, J. S., Ellingson, G. J., Nyholm, P. W., McLain, T. W., Beard, R. W., 2020. Relative navigation of autonomous gps-degraded micro air vehicles. Autonomous Robots 44, 811–830.
URL: https://doi.org/10.1007/s10514-019-09899-4
DOI: 10.1007/s10514-019-09899-4

Wu, P., Shi, Z., Yan, P., 2018. Improved ekf-slam algorithm of unmanned helicopter autonomous landing on ship. Chinese Control Conference, CCC 2018-July, 5287–5292.
DOI: 10.23919/ChiCC.2018.8483338

Xian, Z., He, X., Lian, J., Hu, X., Zhang, L., 2017. A bionic autonomous navigation system by using polarization navigation sensor and stereo camera. Autonomous Robots 41, 1107–1118.
DOI: 10.1007/s10514-016-9596-7

Xiaoying, G., Qiaoling, L., Zhikang, Q., Yan, X., 2021. Target detection of forward vehicle based on improved ssd, 466–468.
DOI: 10.1109/ICCCBDA51879.2021.9442550

Xiong, G., Li, H., Ding, Z., Gong, J., Chen, H., 2017. Subjective evaluation of vehicle active safety using prescan and simulink: Lane departure warning system as an example. 2017 IEEE International Conference on Vehicular Electronics and Safety, ICVES 2017, 208–213.
DOI: 10.1109/ICVES.2017.7991927

Xu, S., Zhou, H., Chou, W., 2020. Erf-imcs: An efficient and robust framework with image-based monte carlo scheme for indoor topological navigation. Applied Sciences (Switzerland) 10.
DOI: 10.3390/app10196829

Yang, L., Lan, W., 2018. On secondary development of ptv-vissim for traffic optimization. 13th International Conference on Computer Science and Education, ICCSE 2018, 542–547.
DOI: 10.1109/ICCSE.2018.8468743

Yang, X., Tang, X., Yang, K., Fu, C., Deng, Z., 2020. Local motion planning framework for autonomous vehicle considering position uncertainty in highway. 2020 4th CAA International Conference on Vehicular Control and Intelligence, CVCI 2020, 471–474.
DOI: 10.1109/CVCI51460.2020.9338481

Yoon, J. H., Raychowdhury, A., 1 2021. Neuroslam: A 65-nm 7.25-to-8.79-tops/w mixed-signal oscillator-based slam accelerator for edge robotics. IEEE Journal of Solid-State Circuits 56, 66–78.
DOI: 10.1109/JSSC.2020.3028298

Yurtsever, E., Lambert, J., Carballo, A., Takeda, K., 2020. A survey of autonomous driving: Common practices and emerging technologies. IEEE Access 8, 58443–58469.
DOI: 10.1109/ACCESS.2020.2983149

Zhang, J., Wang, W., Qi, X., Liao, Z., 2020a. Social and robust navigation for indoor robots based on object semantic grid and topological map. Applied Sciences (Switzerland) 10, 1–20.
DOI: 10.3390/app10248991

Zhang, Q., Niu, B., Zhang, W., Li, Y., 2018. Feature-based ukf-slam using imaging sonar in underwater structured environment. 2018 IEEE 8th International Conference on Underwater System Technology: Theory and Application, USYS 2018.
DOI: 10.1109/USYS.2018.8778989

Zhang, X., 2018. Simple understanding of mask rcnn.
URL: https://alittlepain833.medium.com/simple-unders\tanding-of-mask-rcnn-134b5b330e95

Zhang, X., Fang, Z., Lu, Z., Xiao, J., Cheng, X., Zhang, X., 2021. 3D reconstruction of weak feature indoor scenes based on hector slam and floorplan generation. In 2021 IEEE 7th International Conference on Virtual Reality (ICVR). IEEE, 117–126.
DOI: 10.1109/ICVR51878.2021.9483856

Zhang, Z., Gao, P., He, Z., Chen, H., Wang, J., 11 2020b. Autonomous driving system design for formula racing. Journal of Physics: Conference Series 1651.
DOI: 10.1088/1742-6596/1651/1/012125

Zhou, C., Gu, S., Wen, Y., Du, Z., Xiao, C., Huang, L., Zhu, M., 2020. Motion planning for an unmanned surface vehicle based on topological position maps. Ocean Engineering 198, 106798.
URL: https://doi.org/10.1016/j.oceaneng.2019.106798
DOI: 10.1016/j.oceaneng.2019.106798

Zhu, C., Zhao, C., Li, Z., 2019. A joint simulation for electric vehicle design based on matlab/simulink and carsim. 2019 IEEE International Conference on Vehicular Electronics and Safety, ICVES 2019.
DOI: 10.1109/ICVES.2019.8906498