

VEÍCULO AUTÔNOMO GUIADO POR VISÃO COMPUTACIONAL

LUCAS B. PEREZ¹, DANIEL E. RAZERA²

¹ Graduando em Engenharia de Controle e Automação, 1410547, Câmpus São João da Boa Vista, pluqui1@gmail.com.

² Mestre em Engenharia Elétrica, Docente Orientador, IFSP, Campus São João da Boa Vista, daniel.espanhol@ifsp.edu.br

RESUMO: Veículos autônomos já são uma realidade. Diversos grupos de pesquisas vêm desenvolvendo modelos que permitem retirar os erros humanos no transporte veicular. Este trabalho visa implementar um veículo elétrico que caminha por diversas estações seguindo um caminho pré-identificado entre elas. Será solicitado ao veículo que se desloque para uma determinada estação e o mesmo deve tomar a decisão de qual caminho deve seguir a partir de placas de informação. Serão definidos sensores e protocolos que aliados a Visão Computacional irão gerar os parâmetros para a programação interna. A programação, então poderá definir o caminho para o veículo chegar à estação desejada. O projeto eletrônico será desenvolvido utilizando um Raspberry Pi Model B. Desta forma, a programação será em linguagem Python. Isto ampliará o desenvolvimento do projeto na definição dos sensores e os testes de diversas configurações, resultando em um veículo autônomo funcional.

PALAVRAS-CHAVE: opencv. visão computacional. raspberry pi. veículo autônomo.

AUTONOMOUS VEHICLE GUIDED BY COMPUTER VISION

ABSTRACT: Autonomous vehicles are already a reality. Several research groups have been developing models that allow the removal of human errors in vehicle transportation. This work aims to implement an electric vehicle that walks through several stations following a pre-identified path between them. The vehicle will be asked to move to a particular station and the same must make the decision of which way to follow from the information boards. Sensors and protocols will be defined that allied to Computational Vision will generate the parameters for internal programming. The programming, then you can set the field for the vehicle to reach the desired station. The electronic design will be developed using a Raspberry Pi Model B. In this way the programming will be in Python language. This will extend the design development in the definition of the sensors and the tests of diverse configurations, resulting in a functional autonomous vehicle.

KEYWORDS: opencv. computer vision. raspberry pi. autonomous vehicle.

INTRODUÇÃO

Veículos autônomos são aqueles com a capacidade de se locomover sozinhos, ou seja, possuem a capacidade de tomar suas próprias decisões sem a necessidade de uma condução humana. Estes veículos são pesquisados desde 1972, onde o *Stanford Research Institute* (SRI) desenvolveu o robô Shaky, considerado o primeiro robô móvel controlado por inteligência artificial.

Nos meios industriais, os veículos autônomos possuem grande impacto, onde um veículo capaz de se locomover pelos setores de uma indústria de maneira inteligente pode vir a acelerar em muito o tempo de produção, além de levar uma melhora na qualidade do produto final devido ao fato de realizar movimentos seguros e ordenados, evitando assim possíveis acidentes como impactos no material/produto, o que leva a uma possível danificação. Outro ponto é na repetibilidade, essa pode gerar estresse ou ainda uma lesão aos operários, um movimento ou ciclo repetitivo pode ser facilmente adotado em uma rotina do veículo autômato. Também podem ser utilizados em locais de condições

severas, como em uma usina nuclear ou mesmo em uma indústria metalúrgica, onde os altos níveis de radiação e temperatura podem apresentar um fator de risco aos operários, assim melhorando suas condições de trabalho e vida.

Outra aplicação se encontra no caso dos veículos de transporte, com foco na eliminação dos erros e imprudências humanas no trânsito em geral. Somente no Brasil, os acidentes de transporte causam a morte de 43 mil pessoas por ano segundo os dados do Departamento de Informática do Sistema Unido de Saúde (Datasus, 2014). Acidentes de trânsito são causados, na maior parte dos casos, por fatores de imperícia, imprudência ou negligência por parte do condutor humano (Benenson, 2009).

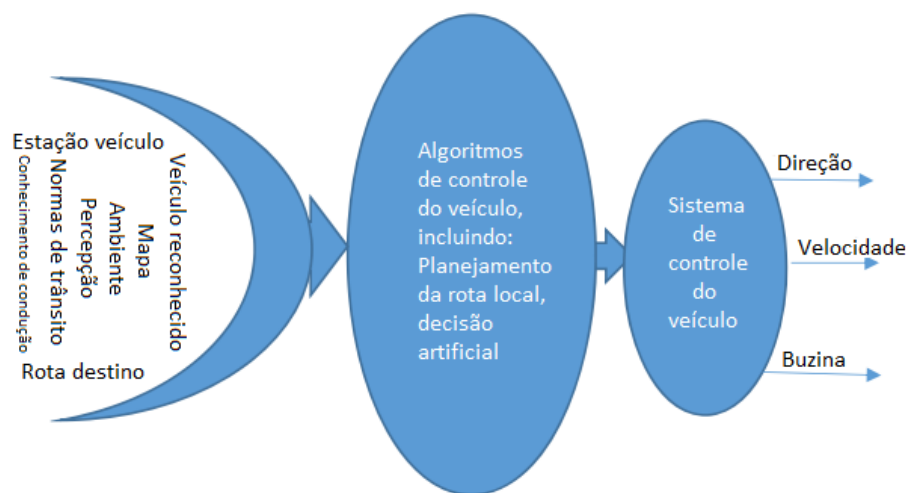
De acordo com Gore et al. (2018), ao se ter um veículo inteligente, capaz de tomar suas próprias decisões e que não está sujeito a realizar imprudências humanas como conduzir cansado, sob efeito de álcool ou drogas ou demais fatores que podem vir a afetar a dirigibilidade, é esperado uma grande redução nos acidentes de trânsito e, por consequência, aumento na qualidade de vida e no conforto de seus usuários.

Veículos autômatos já são uma realidade e diversos grupos de pesquisas vêm desenvolvendo modelos, como é o caso do MIR 100 desenvolvido pelo Mobile-Industrial-Robots (MIR), que se trata de um robô capaz de navegar por um ambiente industrial por meio de câmeras e sensores e assim transportar cargas de até 100 kg e rebocar até 300 kg, o que tende a gerar um aumento na produtividade e ainda reduzir custos.

Inclusive, segundo DAILY et al. (2017), os legisladores alemães já até estabeleceram em 2017 a estrutura legal para permitir os veículos autônomos em vias públicas, desde que um motorista humano esteja no controle, para assumir em situações críticas, ajudando a concretizar essa tecnologia fornecendo legislação clara e responsabilidades legais por seu uso e se espera que até 2020 um veículo de autonomia de nível 5 (totalmente autônomo) possa ser entregue. Tudo isso demonstra que veículos autônomos são possíveis de serem implementados em nosso meio, trazendo inúmeros benefícios para toda a comunidade e meio industrial.

De acordo com ZHAO et al. (2017), veículos autônomos podem ser divididos em 4 partes principais, sendo elas: Sistema de navegação, planejamento do caminho, percepção do ambiente e controle do veículo.

FIGURA 01. Funcionamento de um veículo autônomo.



Fonte: (ZHAO; LIANG; CHEN, 2017, p.11, tradução nossa).

Para o sistema de navegação, todo veículo autônomo deve ser capaz de resolver dois principais problemas, que são sua localização atual e como ir até o destino. Estes problemas podem ser resolvidos com o auxílio de um GPS e com algoritmos de navegação, como o algoritmo de Dijkstra e outros. Estes mesmos algoritmos se encontram no planejamento do caminho e através deles, o veículo sabe qual rota deve ser seguida. A percepção do ambiente é necessária para que o mesmo possua informações suficientes sobre o que está acontecendo ao seu redor e assim tomar as decisões

necessárias para sua locomoção. Já o controle do veículo é onde se encontram dados como o controle sobre a velocidade e direção tomadas, todos os demais fatores influenciam para que o controle seja o mais efetivo e correto possível (ZHAO; LIANG; CHEN, 2017).

A proposta deste projeto é implementar um veículo autônomo que percorre diversas estações seguindo um caminho pré-determinado entre as estações. Será solicitado ao veículo que se desloque de uma determinada estação para outra e esse deve tomar a decisão de qual caminho deve seguir. Serão definidos sensores e protocolos que irão gerar os parâmetros para a programação interna. Sendo assim, a programação poderá definir o melhor caminho para o veículo chegar à estação desejada. Será construída uma pista com diversas estações para testar o funcionamento dele.

O projeto eletrônico será desenvolvido utilizando um Raspberry Pi, o que permite a utilização de uma ampla coleção de sensores graças a sua General Purpose Input/Output (GPIO). Outro fator é pelo menos aceitar diversos tipos de linguagens de programação onde, neste caso, foi optado pela utilização da linguagem Python. Esta linguagem foi selecionada devido a sua compatibilidade com a Open Source Computer Vision Library (OpenCV) e por possuir uma ampla literatura e um maior acesso a exemplos de funcionamento. A OpenCV será melhor descrita em **Material e métodos**.

Proposta do projeto

Sobre o projeto é esperado determinados benefícios como:

- Redução no tempo de produção por meio de acessos mais eficazes aos ambientes da indústria, como a locomoção de uma peça do depósito direto ao local de despacho;
- Substituição dos operários em zonas de riscos e ou de zonas estressantes e repetitivas, como em usinas nucleares, indústrias metalúrgicas ou mesmo com o transporte de possíveis produtos químicos, onde é possível evitar algum provável acidente ao funcionário;
- Diminuição de seu estresse, resultando em uma melhora na sua produção dentro da indústria.

Resumidamente, o projeto deve seguir algumas premissas:

- O veículo deve saber ler informações de placas;
 - Após ler uma placa, esse deve saber se movimentar pelo ambiente;
- O veículo deve seguir uma rota pré-estabelecida até seu destino;
- O veículo deve saber o momento de sua chegada com relação ao destino.

MATERIAL E MÉTODOS

Sobre este projeto: Natureza aplicada, método científico hipotético – Dedutivo, objetivo de estudo exploratório, procedimento técnico por pesquisa bibliográfica e abordagem quantitativa.

O projeto foi pensado a fim de desenvolver um veículo autônomo, que não necessite de nenhum centro de processamento externo. Para reduzir custos e tornar sua locomoção mais efetiva e genérica, quanto ao seu ambiente de trabalho, foi adotado o uso de visão computacional, utilizando a biblioteca OpenCV.

Biblioteca OpenCV

A *Open Source Computer Vision Library* (OpenCV) é uma biblioteca de funções de código aberto, distribuída sob a licença *Berkeley Software Distribution* (BSD). A biblioteca multiplataforma pode ser usada em diversas linguagens de programação, tais como Python, C++ e Java. Sua utilização é principalmente acadêmica e comercial, no desenvolvimento de aplicativos na área de Visão computacional, Processamento de Imagens e Inteligência Artificial.

Composta por mais de 300 funções divididas em diversos módulos, sendo elas com foco no processamento e análise de imagens em tempo real, implementam algoritmos bem estabelecidos na literatura. Tais funções permitem uma simplificação do projeto de aplicações em processamento de imagens, ao mesmo tempo em que permite a confecção de aplicações sofisticadas de tempo real. Para

implantá-la foi necessário o uso de um sistema embarcado, sendo utilizado um Raspberry Pi modelo B.

Raspberry

O Raspberry é um projeto originalmente desenvolvido em Cambridge com a intenção de promover o ensino básico de computação nas escolas com um sistema simples e de baixo custo. O Raspberry Pi é um *Single Board Computer* (SBC) do tamanho de um cartão de crédito. Atualmente o projeto é desenvolvido pela *Raspberry Pi Foundation* que promove o desenvolvimento de projetos com o Raspberry Pi principalmente na área educacional.

O Raspberry Pi 3 B é o segundo modelo mais potente dos Raspberry e conta com o processador Quad Core BCM2837 da Broadcom trabalhando em 1.2GHz e disponibilidade de 1GB de RAM. Possui uma conexão *High-Definition Multimedia Interface* (HDMI) e uma *Display Serial Interface* (DSI) para a conexão com monitores. O armazenamento do sistema operacional e os dados são em cartão micro *Secure Digital* (SD). Conta com 4 portas *Universal Serial Bus* (USB) para uso geral. *Wireless Local Area Network* (WLAN) e Bluetooth, além de um conector para Ethernet. Como se trata de um sistema embarcado para educação, existe uma GPIO de 40 pinos que permite o acionamento e monitoramento de diversos dispositivos eletrônicos.

DESENVOLVIMENTO

A locomoção do veículo foi baseada em robôs seguidores de linha, onde se utiliza sensores infravermelho. Esses sensores podem ser divididos em dois componentes: emissor e receptor, o emissor emite um feixe de luz infravermelha e o receptor capta estes feixes. Sendo assim, sempre em que este sensor se encontra em uma superfície reflexiva, seu receptor detecta o feixe emitido e quando ocorre o contrário, seu receptor nada detecta. Logo, sua saída varia em binário (1 ou 0).

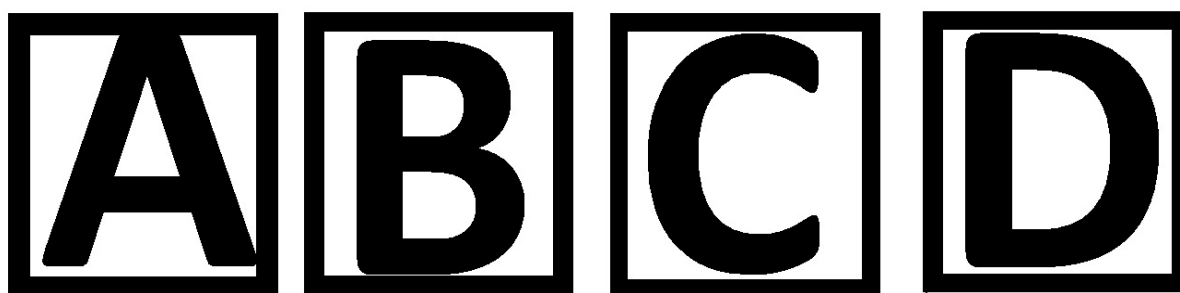
O veículo dispõe de 5 sensores infravermelho localizados na frente dele, eles foram instalados de maneira centralizada, sendo um perfeitamente no meio do veículo e outros 2 de cada lado desse. Para se ter a tomada de decisão foi elaborado um grupo de regras a fim de se estabelecer um erro e assim corrigir a rota de navegação, caso ocorra o desvio de sua trajetória. Este erro varia de -4 até +4, onde o valor negativo para quando o veículo se encontra deslocado para direita, com relação à linha de navegação, e o valor positivo para quando o veículo está deslocado a esquerda, sendo que quando o erro é 0, não existe a necessidade de correção de trajetória.

A leitura de linhas será feita até que seja encontrado um ponto de captura, ou seja, um ponto que contenha placas com informações, esses pontos são marcados por um quadrado do mesmo material das linhas, o que faz com que todos os sensores de leitura disparem informando, assim, sua chegada ao ponto de captura. Neste ponto o sistema irá ativar uma captura da câmera instalada e processar esta imagem para definir qual a rota para a base desejada. Em cada ponto de captura temos uma placa indicativa da direção de uma ou mais bases. O processamento da imagem e identificação da direção foi realizado usando a biblioteca OpenCV.

Durante pesquisas foi descoberto um módulo exemplo da biblioteca OpenCV (*find_obj*) que busca determinados objetos em alguma imagem específica e a partir de estudos feitos através desse módulo foi desenvolvido um padrão de imagens onde, sempre, estarão armazenado todas as informações para a locomoção do veículo.

Para as bases foram criados padrões que fossem representados por letras, como pode ser observado pela Figura 01, onde o primeiro símbolo, da esquerda para a direita, representa a estação “A”, “B”, “C” e “D” respectivamente.


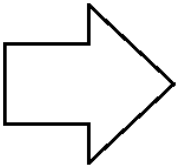

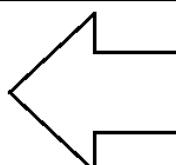

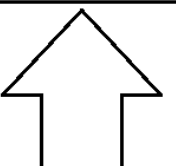

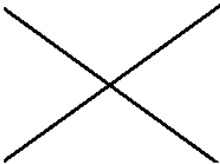
FIGURA 01. Estações padrão.



Fonte: Elaboração própria.

A Figura 02 já representa as direções em que o veículo pode tomar, sendo, de cima para baixo, “virar à direita”, “virar à esquerda”, “seguir reto” e “destino alcançado” respectivamente.

FIGURA 02. Direções padrão.

Símbolos	Direções
	
	
	
	

Fonte: Elaboração própria.

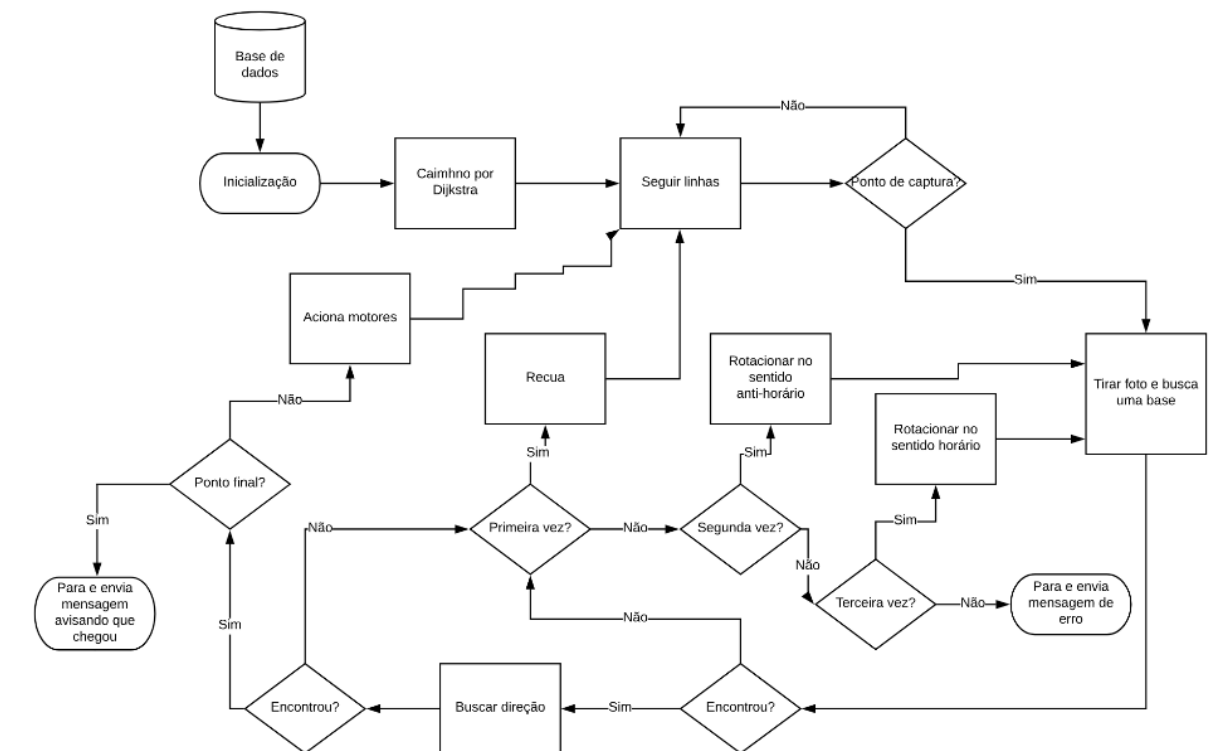
A informação da base e da direção irão compor uma placa localizada em cada cruzamento ou ponto de captura do trajeto.

Para o desenvolvimento foi criada uma classe que utiliza o módulo *find_obj*, onde será recebido uma imagem tirada em tempo real pela câmera. Com a imagem, o módulo será acionado e o Raspberry usará um exemplo dos símbolos padrão contidos em sua memória, feito isso, será gerado uma resposta com os locais em que a placa se encontra na imagem. Caso não tenha uma placa, ele retornará um erro.

A partir dos dados de localização da placa, será feita uma segunda chamada ao *find_obj* que irá procurar por símbolos que representam a direção. Assim será possível identificar os dados de locomoção para o veículo, que por fim será enviado ao módulo responsável pelos controladores dos motores.

Para exemplificar o funcionamento das placas, pode-se utilizar a Figura 03, nela consta duas informações. Primeiro símbolo: Estação “C”. Segundo símbolo: À frente. Logo é possível entender que a estação “C” se encontra à frente, assim, o veículo deve seguir para frente e procurar uma nova linha para seguir.

Figura 04. Fluxograma de processos.



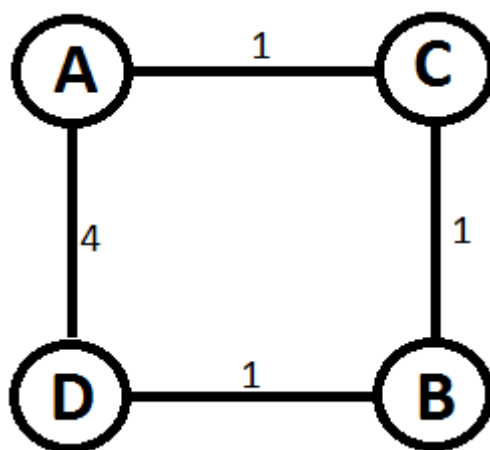
Fonte: Elaboração própria.

Durante testes foi notada uma diferença significativa na taxa de acertos quando a luminosidade era alterada, ao ponto de ser necessária a utilização de dois canhões de luz infravermelho para auxiliar na visão do veículo, o que se mostrou eficiente.

Testes

Para os testes do projeto foram desenvolvidos dois grafos, mapas, a fim de se testar a eficiência de funcionamento do veículo. Estes grafos representam o caminho entre nós, o que pode ser interpretado como cidades e estradas. O número ao lado de cada caminho é um indicador de distância, para o Grafo 01, por exemplo, o nó “A” está interligado ao nó “C” e ao nó “D”, para o primeiro nó existe um caminho com distância 1, enquanto para o segundo a distância é de 4, logo o nó “A” está mais perto do nó “C” do que do nó “D”.

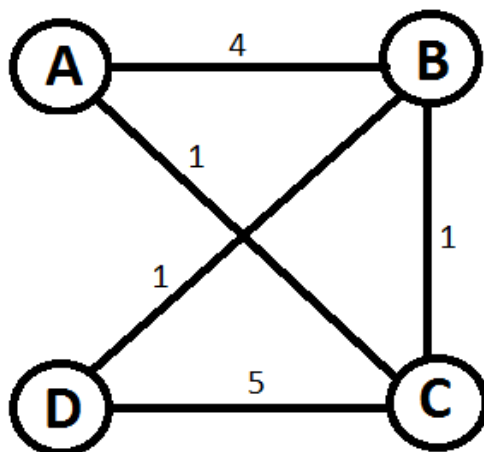
Figura 05. Grafo 01.



Fonte: Elaboração própria.

Para o primeiro grafo foi definido como ponto de partida o nó “A” e ponto de chegada o nó “D”, sendo assim o menor caminho seguindo de “A” para “C”, de “C” para “B” e de “B” para “D”, com um custo total de 3.

Figura 06. Grafo 02.



Fonte: Elaboração própria.

Para o segundo grafo, foram tomados os mesmos nós de partida e chegada e consequentemente, a mesma rota foi estabelecida, visto que o menor caminho continua de “A” para “C”, de “C” para “B” e de “B” para “D”.

A primeira parte dos testes foi em ambiente virtual, a fim de testar a taxa de acerto do algoritmo de Dijkstra, como já esperado, esse conseguiu acertar 200 vezes em 200 testes, o que demonstra uma taxa de acerto de 100%.

Como o algoritmo manteve uma taxa de acerto de 100%, para a montagem do ambiente de testes foram utilizados apenas as placas que seriam utilizadas para o percurso, sendo assim, para o Grafo 01 foram utilizados:

- Uma placa com “virar à direita”;
- Uma placa com “virar à esquerda”;
- Uma placa com “seguir reto”;
- Quatro placas com “destino alcançado”.

Por mais que o Grafo 02 utilize o mesmo percurso, foram adicionados mais pontos de captura, por exemplo, para se alcançar o nó “B”, o veículo deve primeiramente virar à esquerda e depois duas vezes a direita, aumentando assim o numero de placas e dificultando o deslocamento dele. Para este, foram utilizados:

- Duas placas com “virar à direita”;
- Três placas com “virar à esquerda”;
- Duas placas com “seguir reto”;
- Quatro placas com “destino alcançado”.

Para cada grafo foram feitos duas tomadas de 50 testes, sendo a primeira com luz em abundância e a segunda feita durante a noite e sem luz artificial, contando assim apenas com os canhões de infravermelho.

A primeira tomada se mostrou a mais eficiente de todas, visto que havia luz abundante e por se tratar de um menor número de placas, este percurso era mais simples. O veículo se mostrou efetivo, com uma taxa de acerto total de praticamente 100%.

Tabela 01 – Primeira tomada de testes para o Grafo 01 (luminosidade em abundância).

Tipo de placa	Quantidade de placas	Total de testes por placa	Acertos	Taxa de acertos (%)
Virar à esquerda	1	50	50	100
Virar à direita	1	50	49	98
Seguir reto	1	50	50	100
Chegou	4	200	200	100
Total	7	350	349	99,71

Fonte: Elaboração própria.

A luminosidade se mostrou um fator fundamental para uma boa taxa de acertos, na segunda tomada para o Grafo 01, a taxa de acerto total foi de 97,43%, uma queda de 2,28% quando comparado com luminosidade em abundância.

Tabela 02 – Segunda tomada de testes para o Grafo 01 (baixa luminosidade).

Tipo de placa	Quantidade de placas	Total de testes por placa	Acertos	Taxa de acertos (%)
Virar à esquerda	1	50	49	98
Virar à direita	1	50	47	94
Seguir reto	1	50	47	94
Chegou	4	200	198	99
Total	7	350	341	97,43

Fonte: Elaboração própria.

Para o segundo grafo foi notado novamente uma queda da taxa total, acredita-se que isso ocorreu devido ao aumento da complexidade do percurso, visto que como havia um maior número de placas, a chance de ocorrer algum erro era maior. Apesar da queda, a diferença em comparação com a primeira tomada para o Grafo 01 foi algo pequena, apenas 0,44%.

Tabela 03 – Primeira tomada de testes para o Grafo 02 (luminosidade em abundância).

Tipo de placa	Quantidade de placas	Total de testes por placa	Acertos	Taxa de acertos (%)
Virar à esquerda	3	150	149	99,33
Virar à direita	2	100	98	98
Seguir reto	2	100	100	100
Chegou	4	200	199	99,5
Total	11	550	546	99,27

Fonte: Elaboração própria.

Na última tomada, novamente a luminosidade se mostrou um fator importante, porém nesse caso a diferença foi a maior de todas, atingindo 3,09%.

Tabela 04 – Segunda tomada de testes para o Grafo 01 (baixa luminosidade).

Tipo de placa	Quantidade de placas	Total de testes por placa	Acertos	Taxa de acertos (%)
Virar à esquerda	3	150	146	97,33
Virar à direita	2	100	93	93
Seguir reto	2	100	96	96
Chegou	4	200	194	97
Total	11	550	529	96,18

Fonte: Elaboração própria.

Aproveitando todos os dados foi criada uma última tabela, a qual consta todos os dados, incluindo os dois grafos e assim fazendo uma análise total do projeto, sua taxa total de acertos ficou em 98,17%.

Tabela 05 – Relação de testes realizados no total.

Tipo de placa	Total de testes por placa	Acertos	Taxa de acertos (%)
Virar à esquerda	400	396	99
Virar à direita	300	287	95,67
Seguir reto	300	291	97
Chegou	800	793	99,13
Total	1800	1767	98,17

Fonte: Elaboração própria.

Durante algumas tomadas de erro ocorreu a falha de um dos canhões de luz. Quando ocorreram foi feita uma pausa durante as tomadas para a reparação, também se teve a falha de um dos motores, ambos devido a mau contato. Estes erros foram descartados da análise.

CONCLUSÃO

Como pode ser observado nos testes, o veículo se mostrou eficiente, com uma taxa total de acertos de 98,17%. Observou-se que a luminosidade e a angulação com que o veículo parava em relação às placas de sinalização se mostraram um fator importante para o sucesso desse projeto. Quando não existia luz no recinto, a taxa de acerto diminuiu, podendo variar até quase 3%. Vale ressaltar que foram utilizados dois canhões de infravermelho ao lado da câmera, o que permitiu que esta taxa não diminuísse ainda mais.

Outro fator a ser considerado são as placas indicativas que foram impressas em folha sulfite, o que pode ter influenciado em alguns dos erros observados. Devido a pequena espessura e densidade, ocorria uma flambagem dessas criando uma imagem distorcida. Em contrapartida, pequenas rotações das folhas não demonstraram diferenças nos resultados.

Assim, pela análise dos dados e considerando o projeto como um protótipo, os resultados obtidos foram satisfatórios e conclusivos para as possibilidades de aplicação dos sistemas em projetos futuros. Os objetivos iniciais foram alcançados e demonstram a possibilidade de aplicação industrial desse projeto.

Futuras melhorias podem ser encontradas no repositório do projeto, <https://github.com/Fhoust/tcc>, junto com arquivos de *logs* e um vídeo de funcionamento.

REFERÊNCIAS

ABQ - ACADEMIA BRASILEIRA DE QUALIDADE. **Pesquisa: acidentes em rodovias custam R\$ 40 bilhões por ano.** Disponível em: <<http://www.abqualidade.org.br/atualidadesdestaque-abq.php?id=86>>. Acesso em: 11 mai. 2018.

BENENSON, Rodrigo. Perception for driverless vehicles: design and implementation. **École Nationale Supérieure des Mines de Paris**, Paris, França, v. 431, p. 1-219, dez./jun. 2009. Disponível em: <<https://pastel.archives-ouvertes.fr/pastel-00005327/document>>. Acesso em: 19 mai. 2018.

DAILY, M. et al. Self-Driving Cars. **IEEE COMPUTER SOCIETY**, [S.L], v. 50, n. 12, p. 18-23, dez. 2017. Disponível em: <<https://ieeexplore.ieee.org/stamp/stamp.jsp?reload=true&tp=&arnumber=8220479>>. Acesso em: 30 mai. 2018.

GORE, M. D. et al. VEHICLE NAVIGATION USING ADVANCED OPEN SOURCE COMPUTER VISION. **International Journal of Advanced Research in Computer Science**, Pune, Maharashtra, India, v. 9, n. 1, p. 719-725, jan./fev. 2018. Disponível em: <<http://www.ijarcs.info/index.php/Ijarcs/article/view/5350/4584>>. Acesso em: 12 mai. 2018.

JAAFAR, H. et al. INTELLIGENT GUIDANCE PARKING SYSTEM USING MODIFIED DIJKSTRA'S ALGORITHM. **Journal of Engineering Science and Technology**, Selangor, Malaysia, p. 132-141, out. 2014. Disponível em: <http://jestec.taylors.edu.my/Special%20Issue%20SAES2013_9_5_2014/SAES%202013_132_141.pdf>. Acesso em: 23 mai. 2018.

O PORTAL DO TRÂNSITO BRASILEIRO. **Acidentes - causas.** Disponível em: <http://www.transitobr.com.br/index2.php?id_conteudo=8>. Acesso em: 14 abr. 2017.

OPENCV. **Opencvlibrary.** Disponível em: <<https://opencv.org/>>. Acesso em: 03 mai. 2018.

PISSARDINI, Rodrigo De Sousa; WEI, Daniel Chin Min; JÚNIOR, Edvaldo Simões Da Fonseca. VEÍCULOS AUTÔNOMOS: CONCEITOS, HISTÓRICO E ESTADO-DA-ARTE. **Anais do XXVII Congresso de Pesquisa e Ensino em Transportes**, [S.L], v. xxvii, jan. 2013. Disponível em: <https://www.researchgate.net/publication/318897282_VEICULOS_AUTONOMOS_CONCEITOS_HISTORICO_E_ESTADO-DA-ARTE>. Acesso em: 23 abr. 2018.

RASPBERRY PI. **Raspberry pi 3 model b.** Disponível em: <<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>>. Acesso em: 03 mai. 2018.

ROBOTIC. **Mir opens us office, launches autonomous mobile robot into north america.** Disponível em: <<https://roboticsandautomationnews.com/2016/05/23/mir-opens-us-office-launches-autonomous-mobile-robot-into-north-america/4669/>>. Acesso em: 14 abr. 2017.

SRI INTERNATIONAL. **Shakeytherobot.** Disponível em: <<https://www.sri.com/work/timeline-innovation/timeline.php?timeline=computing-digital#!>>. Acesso em: 14 abr. 2017.

TECHTUDO. **Google cria carro autômato, sem volante e para duas pessoas; veja vídeo.** Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2014/05/google-cria-carro-automatico-sem-volante-e-para-duas-pessoas-veja-video.html>>. Acesso em: 14 abr. 2017.

ZHAO, Jianfeng; LIANG, Bodong; CHEN, Qiuxia. The key technology toward the self-driving car. **International Journal of Intelligent Unmanned Systems**, [S.L], v. 6, n. 1, p. 2-20, out. 2017. Disponível em: <<https://www.emeraldinsight.com/doi/pdfplus/10.1108/IJIUS-08-2017-0008>>. Acesso em: 30 mai. 2018.