

Desenvolvimento de um robô autônomo em cenário educativo

Estanislau de Sena Filho,
 Audryo Aparecido da Terra C. Júnior,
 Italo Gustavo D. Cordeiro, João Paulo G. de Souza,
 Marina Bernardes Diniz, José Antônio C. Avila,
 Flávio Luis Cardeal Pádua, Natália C. Batista

Centro Federal de Educação Tecnológica de Minas Gerais, MG, Brasil
 (e-mails: *estanislau.sena@gmail.com, audryoaj@gmail.com,*
italodonato4@gmail.com, aniram.520@gmail.com,
joao13_souza@hotmail.com, jose.carneiro.avila@gmail.com,
cardeal@cefetmg.br, nataliabatista@cefetmg.br).

Abstract: Robotics, as a technological resource for education, provides the practical demonstration of theoretical concepts from several areas of knowledge. In a creative and playful way, robots can be used in formal learning environments and in interactive scientific spaces to make learning easier and increase the interest of students of all levels of education. This article presents the development of a small autonomous robot capable of navigating the tracks of a small city model, reading traffic signs, finding the edges of lane, detecting obstacles, and obeying traffic lights. The low cost autonomous robot was built using a Raspberry Pi computer, sensors, and other elements that enable the tasks to be performed. Experiments were carried out with the robot's modules and the results indicate the potential of the algorithms in the development of a system for effective demonstrations similar to a real scenario.

Resumo: A robótica, como recurso tecnológico para a educação, propicia a demonstração prática de conceitos teóricos que integram diversas áreas do conhecimento. De forma criativa e lúdica, os robôs podem ser usados em ambientes formais de aprendizagem e em espaços científicos interativos para facilitar o aprendizado e despertar o interesse de estudantes de diferentes níveis de educação. Neste artigo é apresentado o desenvolvimento de um robô autônomo de pequeno porte capaz de navegar em um cenário dinâmico, interpretando alguns elementos da sinalização de trânsito, e de realizar as ações de seguir a pista corretamente, detectar obstáculos e obedecer a um semáforo. O robô autônomo foi desenvolvido utilizando um computador *Raspberry Pi*, sensores e outros elementos que possibilitam a execução das tarefas que o robô precisa realizar no cenário. Experimentos foram realizados com os módulos do robô e os resultados indicam o potencial dos algoritmos na busca por um sistema que possibilite demonstrações com um funcionamento adequado e similares a um cenário real.

Keywords: Mobile robotics; Educational robotics; Computer Vision; OpenCV; Raspberry Pi.

Palavras-chaves: Robótica móvel; Robótica educacional; Visão Computacional; OpenCV; *Raspberry Pi*.

1. INTRODUÇÃO

Um robô autônomo é um sistema que dispõe de sensores e outros sistemas de hardware e software que lhe permitem movimentar de forma independente e sem intervenção humana. As novas possibilidades de serviços, lazer, transporte, dentre outras, que eles podem oferecer, têm chamado a atenção das pessoas para os robôs autônomos e, desse modo, vemos aumentar a coexistência de tais sistemas com os seres humanos a cada dia. São várias as tarefas que os robôs podem executar para ajudar as pessoas ou cooperar com elas, por exemplo, robôs de serviço podem realizar tarefas domésticas como a limpeza de casas, guias em museus, atendimento em recepções e transporte de cargas (Tiddi et al., 2020; Pereira et al., 2013).

Entretanto, mesmo com diversas aplicações em várias situações, são necessárias ações para aumentar o interesse em robótica e ciências em nosso país. A contínua redução do investimento em educação e pesquisa ao longo dos anos (Tanno, 2019) e a quantidade reduzida de espaços científicos interativos (Almeida et al., 2015) são alguns dos entraves a um conhecimento mais amplo e a um aprendizado mais profundo em relação a essa área.

Observa-se em vários países o aparecimento de museus de ciências que exploram a temática da robótica e da computação. Pode-se citar alguns exemplos como *The Robot Museum* (Madrid, Espanha) e *The Tech Museum of Innovation* (San Jose, Estados Unidos). Na cidade de Seul, na Coreia do Sul, há um projeto de construção inovador do *Robot Science Museum*, para apoiar a educação pública

em robótica e aumentar o conhecimento e o interesse do público por robôs (Architects, 2019). No Brasil, pode-se citar o Museu do Computador (São Paulo, Brasil). Esses espaços possuem grande importância para difundir a cultura científica e despertar o interesse pelo conhecimento, assim como mostras de robótica, feiras e oficinas, que tornam mais acessíveis as experiências com tecnologias robóticas (Miller et al., 2008).

De acordo com Pio et al. (2006), a robótica é uma ferramenta multidisciplinar e, no contexto educacional, promove a integração entre diferentes áreas do conhecimento. Em seu trabalho é discutido o uso da robótica como instrumento para apoiar o ensino-aprendizagem de computação, que envolve lógica e resolução de problemas, potencializando a construção de habilidades importantes para estudantes. Kong e Wang (2019) citam que estudos neste campo mostraram que a robótica tem um impacto positivo na aprendizagem dos alunos em diferentes áreas (por exemplo, física, matemática e ciência da computação) e também tem impacto no desenvolvimento pessoal, incluindo pensamento criativo e tomada de decisão.

Dorouka et al. (2020) referem-se à educação em robótica como uma forma bem sucedida de envolver crianças em atividades relacionadas a STEM (*Science, Technology, Engineering, and Mathematics*), pois robôs são atrativos tanto para crianças quanto para adultos. Um caso de estudo no México é apresentado no trabalho de Ponce et al. (2019), onde o uso das plataformas robóticas foi avaliado dentro de escolas de ensino fundamental e universidades. Os resultados mostraram que seu uso em sala de aula ajudou os alunos a melhorar sua aquisição de conhecimento e aumentou sua motivação e atenção, além de desenvolverem um melhor senso de trabalho em equipe, habilidades de comunicação e solução de problemas.

Há múltiplas possibilidades de projetos multidisciplinares que podem ser desenvolvidos no ensino de disciplinas ou utilizados em ambientes informais de aprendizagem, tais como os citados no trabalho de Prol (2006), que recebem três classificações: projetos gerais que reflitam problemas da comunidade (trânsito, construção de cidades, etc), projetos visando um conteúdo específico (tópicos de física, tópicos de matemática, ciências e automação) e projeto de estudo de uma tecnologia (uso de interfaces, computadores, sensores, motores, engrenagens e programação).

Neste contexto, este artigo apresenta o desenvolvimento de um robô autônomo de pequeno porte capaz de navegar em um cenário dinâmico, interpretando alguns elementos da sinalização do Código de Trânsito Brasileiro, e de realizar as ações de seguir a pista corretamente, detectar obstáculos e obedecer a um semáforo. A inserção do protótipo no cenário educativo permite a demonstração e o ensino de conceitos de robótica, inteligência artificial, física, matemática e também o estudo sobre a dinâmica do trânsito.

O robô autônomo foi desenvolvido utilizando um computador *Raspberry Pi*, sensores e outros elementos que possibilitam a execução das tarefas que o robô precisa realizar no cenário. São tarefas bastante desafiadoras, pois ele tem que ser capaz de interpretar corretamente informações do seu ambiente por meio de um sistema de percepção, que é responsável por transformar os dados provenientes de

sensores, tais como imagens de câmeras e leituras a laser, em informações consistentes e úteis para compreender o cenário e detectar obstáculos. Em 2019, o protótipo foi apresentado em uma feira local¹, tendo recebido a visita de vários estudantes e visitantes externos que puderam ver a demonstração dos conceitos de computação e robótica envolvidos de forma lúdica. A Figura 1 mostra o protótipo e o cenário construído.

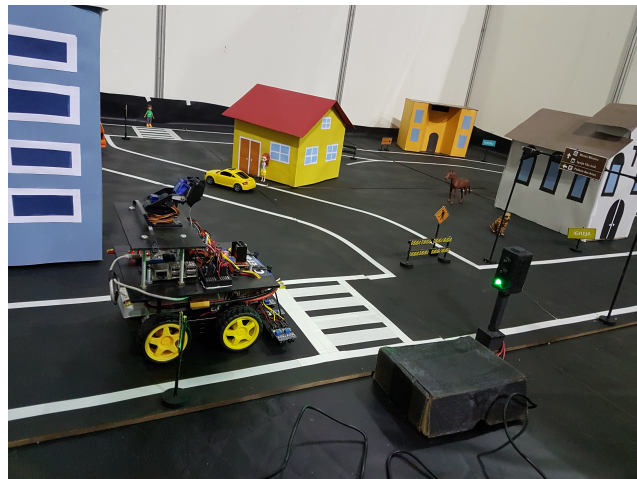


Figura 1. Cenário construído para a navegação do robô.

Neste artigo, é apresentada a arquitetura do protótipo, o funcionamento dos módulos e os algoritmos de visão computacional utilizados, bem como os testes realizados e uma avaliação dos algoritmos. Na próxima seção, será descrito como foi construída a arquitetura física do robô. A Seção 3 apresentará o sistema de visão computacional. Os testes e a análise dos resultados serão mostrados na Seção 4. Finalmente, as considerações finais e os trabalhos futuros serão abordados na Seção 5.

2. PLATAFORMA ROBÓTICA MÓVEL

Para a construção do robô, que é um protótipo de um carro elétrico, foi utilizado um computador embarcado *Raspberry Pi* e diversos componentes eletrônicos, como fototransistores, além de uma câmera e um sensor a laser. Os dados coletados pelos sensores são transmitidos ao computador embarcado através dos pinos *GPIO's* (*General Purpose Input/Output*) e pelo barramento *CSI Camera Port*. De posse das informações sobre o ambiente, o *Raspberry Pi* realiza todos os cálculos necessários para que o veículo elétrico tome uma decisão.

Os componentes de hardware foram fixados sobre duas placas de MDF que estão acopladas sobre um chassi metálico pré-construído com quatro motores e suas respectivas rodas. A rotação das rodas é sincronizada com base nos lados, isto é, as rodas dianteiras esquerda e traseira esquerda giram em sincronia e as rodas dianteira direita e traseira direita também giram em sincronia. Assim, o par de motores de cada lado recebe a mesma entrada digital do módulo Ponte H *L298N*. Isso ajuda o carro a avançar e a recuar quando as duas rodas laterais giram na mesma direção e com a mesma velocidade. O carro gira quando

¹ A 29ª META (Mostra Específica de Trabalhos e Aplicações) é uma feira de caráter científico e tecnológico do CEFET-MG.

as rodas do lado esquerdo giram na direção oposta a das rodas do lado direito.

O protótipo foi projetado buscando-se uma boa distribuição dos componentes eletrônicos visando facilitar a mobilidade do carro elétrico. A seguir, os principais componentes são explicados em mais detalhes.

2.1 Raspberry Pi

O *Raspberry Pi* é um computador de pequeno porte, do tamanho aproximado de um cartão de crédito, que contém todas as funcionalidades básicas de um computador pessoal, incluindo ferramentas para aprendizagem de programação. Além disso, o mesmo permite o desenvolvimento de aplicações para sistemas embarcados. Neste projeto, utilizou-se o modelo 3 B que possui 1 GB de memória RAM e uma CPU *Quad Core 1.2GHz Broadcom BCM2837 64bit*.

2.2 Raspberry Pi Câmera V2 8MP

O módulo da câmera é utilizado para capturar as imagens a serem processadas pelos algoritmos de visão computacional. Este módulo também permite tirar fotos e gravar vídeos em alta definição. A câmera foi acoplada a um suporte fixado na parte mais alta do carro elétrico, obtendo um maior campo de visão.

2.3 Sensor de Distância VL53L0X

O sensor *VL53L0X* é utilizado para detectar a distância dos obstáculos em relação ao robô. O mesmo emite luz invisível e o circuito interno é capaz de detectar e calcular quanto tempo a luz levou para atingir um objeto e retornar para o sensor.

2.4 Ponte H L298N

O módulo Ponte H *L298N* realiza o controle de 4 motores de corrente contínua (DC). Esse módulo é projetado para controlar motores DC e motores de passo, permitindo o controle não só do sentido de rotação do motor, como também da sua velocidade, utilizando os pinos PWM (*Pulse Width Modulation*) do *Raspberry Pi*.

2.5 Conversores de nível lógico

O módulo ponte H *L298N* funciona com sinais de entrada/saída de 5 V, porém o *Raspberry Pi* opera com sinais de 3,3 V. Para que os dois componentes se comuniquem, deve-se utilizar um conversor de nível lógico, para garantir a segurança e a integridade na transmissão dos sinais entre os pinos de operação da Ponte H e do *Raspberry Pi*.

2.6 Baterias

Duas baterias são utilizadas, uma *Pineng* de 5 V e 12000 mAh, para alimentação do *Raspberry Pi*, e outra Lipo de 12,4 V e 2200 mAh, para alimentação da Ponte H. As baterias de alimentação estão presas sobre o chassi metálico.

2.7 Fototransistores TCRT 5000

São sensores do tipo óptico reflexivo que possuem acoplados no mesmo dispositivo um sensor infravermelho (emissor) e um fototransistor (receptor). Esses sensores retornam valores distintos quando estão sobre superfícies claras e escuras. No robô atuam como um complemento aos algoritmos de visão computacional para detecção das bordas da pista.

2.8 Conversor Analógico Digital ADS 1115

O *Raspberry Pi* não possui um ADC (*Analog Digital Converter*) embutido, impossibilitando a interpretação de sinais analógicos. Diante disso, um conversor analógico digital é utilizado para ler informações de sensores analógicos com alta taxa de precisão.

2.9 Esquema dos componentes de hardware e software

O protótipo prioriza o uso de soluções livres tanto em termos de hardware quanto de software e pode ser considerado de baixo custo, pois a soma de todos os componentes utilizados não ultrapassou R\$ 1.500,00, valor inferior se comparado a outros kits disponíveis no mercado. Uma tabela com a relação dos principais componentes utilizados e seus respectivos preços, está disponível na página do trabalho², bem como uma tabela comparativa com outros kits.

Para o funcionamento do computador embarcado *Raspberry Pi*, foi necessária a instalação e configuração de programas como o *Raspbian* (sistema operacional otimizado para o *hardware* do *Raspberry Pi*) e o *VNC Viewer* (possibilita a criação de uma rede virtual privada para acessar remotamente o computador embarcado do robô).

A Figura 2 mostra um esquema contendo os componentes de hardware e software do robô móvel, cujos algoritmos serão explicados na próxima seção. Os algoritmos desenvolvidos foram implementados em módulos, cujo processamento é realizado pelo *Raspberry Pi*.

3. SISTEMA DE VISÃO COMPUTACIONAL

Os dados da câmera do robô são utilizados pelos algoritmos de visão computacional para obter informações sobre o ambiente e permitir a tomada de decisão relativa às ações de seguir a pista corretamente, parar frente a obstáculos e obedecer à sinalização e ao semáforo. A detecção de cada um desses elementos é detalhada nas próximas subseções. A implementação dos módulos de software foi realizada nas linguagens de programação *Python* e *C++* e foi utilizada a biblioteca *OpenCV* (Bradski and Kaehler, 2009), que possui algoritmos otimizados para aplicações relacionadas à visão computacional em tempo real.

3.1 Detecção de placas de sinalização

Para realizar a detecção e identificação das placas de sinalização de trânsito, utilizou-se um algoritmo de Aprendizado

² <https://sites.google.com/view/projetoatena>.

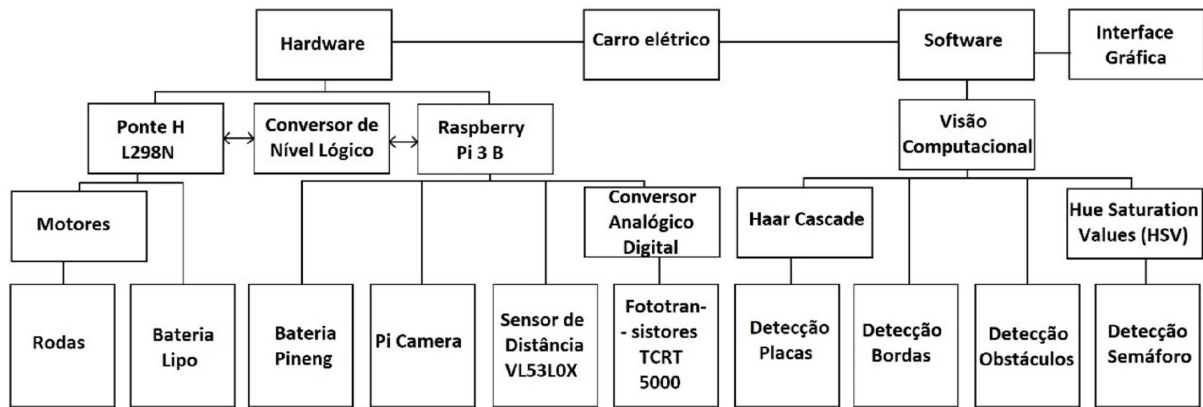


Figura 2. Esquema dos componentes de hardware e software do robô móvel.

de Máquina chamado classificador em cascata (*Haar Cascade*), que detecta objetos de acordo com um modelo obtido de uma base de imagens de treino. Para o classificador conseguir diferenciar os objetos das imagens, é necessário realizar um treinamento com imagens positivas (que contêm o objeto) e imagens negativas (que não contêm o objeto), que irá resultar em um modelo que descreve como são as características do objeto da categoria escolhida. Neste trabalho foram utilizadas 2000 imagens positivas e 5000 imagens negativas. O classificador em cascata consiste de estágios e cada estágio é formado por classificadores mais simples (fracos). As características escolhidas foram as de *Haar* (Viola and Jones, 2001).

A Figura 3 mostra exemplos das placas utilizadas.



Figura 3. Placas utilizadas no cenário.

3.2 Detecção das bordas da pista

Por meio da imagem capturada pela câmera foi possível fazer a detecção da pista e suas bordas. A Figura 4[a] corresponde à imagem original com a visão do robô. Na Figura 4[b], utilizou-se a função de desenho de linhas da biblioteca *OpenCV* para fazer o desenho dos retângulos vermelho e verde. Cada retângulo foi criado a partir da definição de quatro pontos (x, y) .

Os quatro pontos que definem o retângulo vermelho e os quatro pontos que definem o retângulo verde foram associados a vetores e com eles foi possível utilizar as funções de transformação e consolidação perspectiva, que faz transformações geométricas de imagens 3D para uma nova dimensão 2D, permitindo sua exibição e manipulação.

A Figura 5 é a imagem gerada a partir da transformação perspectiva, destacando a região de interesse representada pelos retângulos amarelos. Para fazer a detecção das bordas, utilizaram-se das duas regiões de interesse para criar

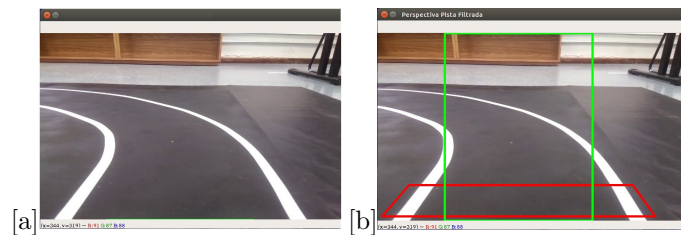


Figura 4. Detecção das bordas da pista. [a] Imagem original. [b] Imagem com os pontos definidos para transformação perspectiva.

duas novas imagens correspondentes às bordas da esquerda e da direita.



Figura 5. Região de interesse na imagem perspectiva.

Com a criação das duas imagens correspondentes às regiões de interesse das bordas esquerda e direita, foi realizada a aplicação de algumas funções para: conversão para escala de cinza, distorção da imagem, binarização da imagem e a detecção de bordas. Por fim, depois de realizado o processo de detecção das bordas, foi efetuada a detecção de contornos para calcular a área e o centroide da imagem. Com isso, duas linhas brancas pequenas, uma horizontal e outra vertical, foram criadas, cujo ponto de interseção entre elas corresponde ao centroide da imagem, como pode ser visto na Figura 6. Como este centro de massa da imagem oscila conforme o robô se movimenta, é possível determinar quando o robô está saindo da pista.

3.3 Detecção de obstáculos

Para detectar impedimentos na pista, também foram utilizadas as imagens da câmera do robô. Para cada quadro (*frame*) obtido, são realizados os passos explicados a seguir.



Figura 6. Pontos de interseção entre as bordas da pista, indicando que o robô está em linha reta.

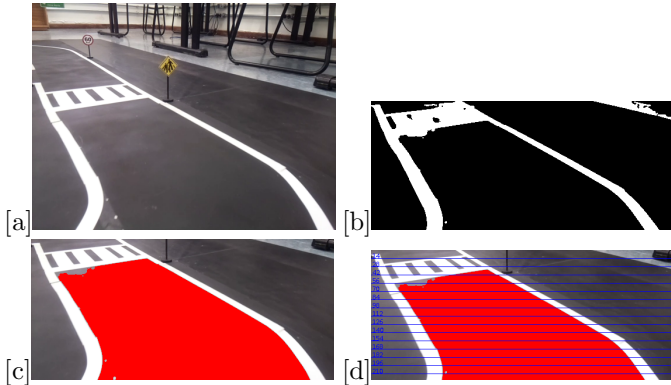


Figura 7. Detecção de impedimentos na pista. [a] Imagem original. [b] Imagem cortada, suavizada e binarizada. [c] Inundação com algoritmo *Flood fill*. [d] Imagem com os perfis que serão analisados.

- (1) Inicialmente, é feito um borramento na imagem utilizando o filtro da mediana, que suaviza a imagem, ou seja, remove pequenos detalhes da imagem e possibilita preencher pequenas discontinuidades em linhas (Gonzalez and Woods, 2003).
- (2) A imagem filtrada é então convertida para escala de cinza utilizando a média ponderada dos canais R (*Red*), G (*Green*) e B (*Blue*).
- (3) É realizada uma binarização simples utilizando um limiar cujo valor é 155.
- (4) A imagem é cortada para eliminar regiões que não são de interesse na detecção, por exemplo regiões mais distantes do robô. O resultado das quatro primeiras etapas podem ser vistas na Figura 7 [b].
- (5) Aplica-se o algoritmo *Flood Fill* para encontrar todos os pixels pretos que estejam ligados, partindo da parte central inferior da imagem. Esses pixels tendem a ser área livre da pista. O *Flood Fill* é um algoritmo que encontra pixels conectados cuja cor é similar ao pixel de origem (semente), gerando uma área que pode ser considerada uniforme na imagem. Nesta etapa obtém-se a imagem da Figura 7[c].
- (6) São analisados então 15 perfis (linhas) da imagem, ou seja, é realizada a contagem de pixels de cada cor seguidos (pixels contíguos de mesma cor formam uma região). Os impedimentos na pista são detectados a partir dessa análise. Por exemplo, uma região vermelha entre duas brancas e com largura suficiente para a passagem do robô pode indicar pista livre. Faixa de pedestre também é detectada pela repetição do padrão da faixa (por exemplo, se houver mais de quatro repetições do padrão preto e branco, com uma largura mínima para cada região e se a medida das regiões não variar muito, há indicativo de faixa de

pedestres). Após a análise de cada perfil, os perfis das metades superior (e da inferior) da imagem são combinados para gerar o resultado final. A Figura 7[d] mostra os perfis utilizados na imagem de exemplo.

O resultado final pode ser impedimento, pista livre ou faixa de pedestres.

3.4 Detecção dos estados do semáforo

O semáforo foi construído em impressora 3D e possui três leds de alto brilho das cores vermelho, verde e amarelo. Os leds são controlados pela plataforma *Arduino UNO R3* que alterna a sequência de acionamento deles de forma a imitar um semáforo real. Os leds vermelho e verde permanecem acesos por 15 segundos, enquanto que o amarelo fica aceso por 5 segundos. A Figura 8 mostra o robô parado em frente ao semáforo.

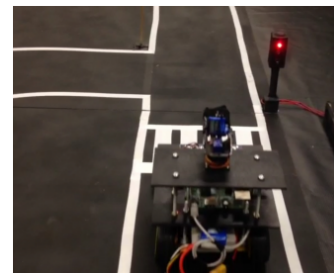


Figura 8. Robô parado em frente ao semáforo.

O algoritmo para detecção dos estados do semáforo utilizou as imagens capturadas pela câmera para delimitar uma região de interesse, onde foi realizada a detecção das cores emitidas pelo semáforo. Os estados foram determinados pela comparação dos pixels da região com intervalos determinados para as cores vermelha e verde usando o espaço de cores HSV (*Hue, Saturation, Value*). A Figura 9 corresponde à imagem da região de interesse onde será feita a detecção dos estados do semáforo.

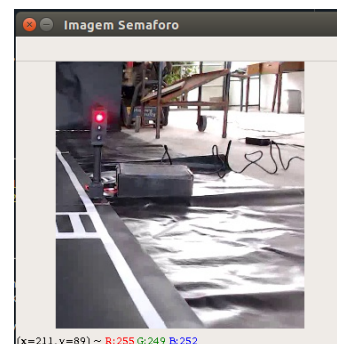


Figura 9. Região de interesse para detecção das cores do semáforo.

4. TESTES E RESULTADOS

Para validar a arquitetura do robô e os módulos implementados, foram realizados testes em um cenário construído por meio do reaproveitamento de materiais recicláveis (como caixas de papelão, palitos e palha de aço) e outros materiais que compuseram o ambiente urbano em miniatura. O cenário contém vias delimitadas com

faixas brancas, faixas de pedestres, placas de sinalização, pedestres, animais e construções.

Os testes realizados visaram verificar a navegação autônoma do robô e a eficácia dos algoritmos de visão computacional nesta aplicação. Os algoritmos não utilizam nenhuma informação sobre o formato da pista e sobre o posicionamento dos elementos do cenário. Os vídeos obtidos da câmera, a uma taxa de 30 fps (*frames* por segundo), foram reduzidos a 10 fps para os testes (dada a baixa velocidade do robô, uma taxa alta de quadros produz muitos quadros praticamente iguais que podem ser descartados). As métricas utilizadas para analisar os resultados dos testes foram a precisão e a revocação. A revocação ou taxa de verdadeiros positivos, neste contexto, é a porcentagem de acerto na identificação dos elementos de uma classe em relação ao número total de elementos dessa classe, ou seja, o número de verdadeiros positivos (VP) identificados pelo detector dividido pelo número total de elementos da classe que foram encontrados manualmente nas imagens da câmeras, que inclui os falsos negativos (FN). A precisão ou valor preditivo positivo é o número de verdadeiros positivos dividido pelo número total de elementos identificados pelo detector, ou seja, a soma de verdadeiros positivos e falsos positivos (FP).

Os testes para detecção das bordas da pista foram realizados com o robô em modo completamente autônomo e tiveram o objetivo de verificar se a navegação ocorreria corretamente sem que o robô saísse da pista. O algoritmo de detecção das bordas da pista foi executado em tempo real e permitiu o controle autônomo do movimento do robô. Para estimar a precisão do algoritmo, foram analisados manualmente 638 quadros de 2 vídeos gravados durante o experimento para verificar se, quando havia necessidade de correção dos motores, o algoritmo forneceu a instrução correta de correção. Por exemplo, ao iniciar uma curva, o robô precisa fazer uma correção no motor direito ou esquerdo para não sair da pista. Dessa forma, a cada quadro, foi anotado se havia essa necessidade de correção (pista em curva) e qual foi a resposta do algoritmo (realizar correção ou não).

Do total de quadros analisados, em 30 havia necessidade de correção dos motores em situação de curva devido à alteração da borda esquerda e 229 devido à borda direita. Em relação à alteração da faixa da esquerda, o algoritmo procedeu corretamente em todos os quadros analisados. Em relação à borda da direita, somente em 4 quadros o algoritmo indicou uma instrução errada de correção, ou seja, havia uma curva e não houve correção dos motores. Todas as vezes que a pista estava reta (sem curvas), o algoritmo não retornou instrução de correção. Esses resultados indicam uma precisão de 100% para as duas situações (borda esquerda e borda direita) e uma revocação de 100% em relação à borda esquerda e 98% em relação à borda direita nos testes realizados. Em vista disso, considera-se que a navegação do robô foi bem sucedida dentro dos limites da pista.

Nos outros testes dos algoritmos de visão computacional, o robô percorreu um trajeto sendo controlado remotamente enquanto gravava os vídeos. Os algoritmos foram executados após o término das gravações. Os experimentos foram

realizados utilizando 10 vídeos com trajetória, iluminação, situação do cenário, trajeto e duração diferentes.

Para os testes com o algoritmo de detecção de placas, 3 vídeos foram usados para definir os parâmetros do algoritmo e 7 vídeos foram usados para avaliar o algoritmo. Foram analisados 2072 quadros, sendo que placas oclusas e cortadas não foram consideradas. Os resultados para cada placa estão na Tabela 1, onde são mostrados o número de placas anotados manualmente (GT, em inglês *ground truth* ou gabarito) e o número de VP, FP e FN encontrados pelos algoritmos seguidos dos valores de precisão e revocação. Esses resultados mostram que, em geral, os detectores tiveram uma boa precisão, indicando que as instâncias retornadas são na maioria corretas, com exceção da placa de Desvio. A revocação foi satisfatória nas placas Pare e Pedestre, indicando que a maior parte das placas existentes no trajeto do robô foram detectadas. Entretanto, para as outras placas, o número de falsos negativos foi alto, mostrando uma necessidade de ampliar o processo de treinamento e ajuste de parâmetros. No caso da placa de Desvio, houve uma dificuldade maior do classificador em discriminar as placas dessa classe e, em parte das imagens analisadas, observou-se que as placas estavam sem iluminação suficiente e algumas apresentaram borramento devido ao movimento do robô.

| | GT | VP | FP | FN | Precisão | Revocação |
|------------|-----|-----|----|----|----------|-----------|
| Pare | 181 | 114 | 0 | 67 | 1,0 | 0,6 |
| Pedestre | 169 | 130 | 10 | 39 | 0,9 | 0,8 |
| Desvio | 38 | 19 | 17 | 19 | 0,5 | 0,5 |
| Velocidade | 105 | 39 | 0 | 66 | 1,0 | 0,4 |
| Obras | 87 | 7 | 2 | 80 | 0,8 | 0,1 |

Tabela 1. Resultados dos testes do algoritmo de identificação de placas de sinalização.

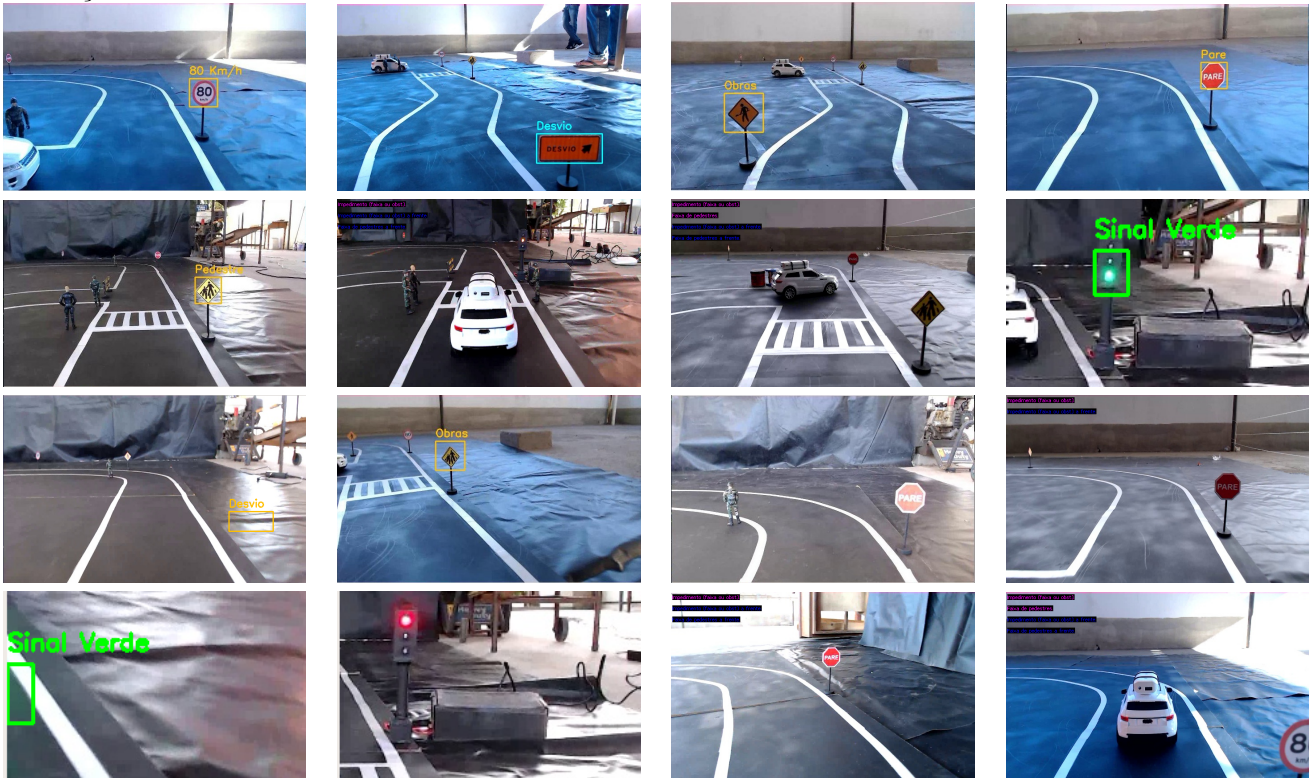
O algoritmo de detecção de impedimentos ou obstáculos na pista foi testado utilizando 1557 quadros obtidos de 8 vídeos (dois foram usados para ajuste de parâmetros). Os quadros em que houve obstrução severa da pista foram removidos da análise, principalmente em situação de curva, resultando em 943 quadros analisados. Como mostrado na Tabela 2, em que as três primeiras colunas formam uma matriz de confusão, o resultado obtido foi satisfatório na identificação de pista livre, com uma precisão de 0,9 e uma revocação de 0,7. Em relação à detecção de impedimentos na pista, o algoritmo teve uma tendência alta de falsos positivos pois qualquer alteração da pista foi entendida como obstrução à passagem do robô, resultando em um baixo valor de precisão. Entretanto, a maioria dos impedimentos existentes foi identificado, resultando em um valor de revocação equivalente a 0,8.

Tabela 2. Resultados dos testes do algoritmo de detecção de impedimentos na pista.

| | Livre | Impedida | Precisão | Revocação |
|----------|-------|----------|----------|-----------|
| Livre | 469 | 219 | 0,9 | 0,7 |
| Impedida | 51 | 204 | 0,5 | 0,8 |

Na detecção de faixas de pedestres, foram analisados os mesmos quadros citados anteriormente, obtendo-se um valor de precisão igual a 0,9 e revocação igual a 0,5, como pode ser visto na Tabela 3. Esses resultados mostram que, embora o número de falsos positivos tenha sido relativamente baixo, em algumas situações a faixa foi confundida

Figura 10. Exemplos de imagens com as detecções de placas, semáforo, faixa de pedestres e impedimentos na pista. As duas primeiras linhas contêm imagens com detecções corretas e as duas últimas linhas contêm imagens com detecções incorretas.



com outros objetos que continham padrões repetidos e em outras ocorreu oclusão da faixa por obstáculos, além da iluminação não uniforme que dificulta a identificação do padrão da faixa em determinados quadros.

Tabela 3. Resultados dos testes de detecção de faixas de pedestres.

| GT | VP | FN | FP | Revocação | Precisão |
|-----|-----|-----|----|-----------|----------|
| 310 | 166 | 144 | 18 | 0,5 | 0,9 |

Nos experimentos com o algoritmo de detecção do semáforo, foram utilizados também 3 vídeos para o ajuste dos parâmetros e 7 vídeos para os testes, sendo 2028 quadros analisados ao todo. Em 270 quadros havia semáforo, sendo que em 150 a luz vermelha estava acesa e em 120 a luz verde estava acesa. A Tabela 4 mostra os resultados do algoritmo de detecção do semáforo. Os valores de precisão e revocação indicam que a detecção do sinal verde foi melhor e, embora nas duas situações o número de falsos positivos tenha sido baixo, ainda há um número considerável de falsos negativos em relação ao sinal vermelho.

Tabela 4. Resultados dos testes de detecção do semáforo.

| | GT | VP | FP | FN | Precisão | Revocação |
|----------|-----|-----|----|----|----------|-----------|
| Vermelho | 150 | 100 | 12 | 50 | 0.9 | 0.7 |
| Verde | 120 | 108 | 4 | 12 | 0.9 | 0.9 |

A Figura 10 mostra exemplos de quadros com as detecções realizadas. Outros exemplos de detecções e vídeos do robô podem ser encontrados na página do trabalho.

Os experimentos executados mostraram que a detecção de objetos em condições variáveis de iluminação e qualidade das imagens não é uma tarefa trivial. Entretanto, a avaliação dos algoritmos apresentou resultados satisfatórios havendo poucos casos de erros em imagens com boas condições de luminosidade. Dessa forma, os resultados indicam o potencial dos algoritmos na busca por um sistema que possibilite demonstrações com um funcionamento adequado e similares a um cenário real.

5. CONSIDERAÇÕES FINAIS

Este artigo apresentou o desenvolvimento de um robô autônomo projetado para navegação em um cenário de ambiente urbano em tamanho reduzido. O protótipo de baixo custo prioriza o uso de soluções livres em termos de hardware e de software e seus módulos proporcionam autonomia para navegar dentro dos limites de uma pista, provendo a detecção de determinadas placas de sinalização, luzes do semáforo e impedimentos na pista.

Os algoritmos de visão computacional testados mostraram resultados satisfatórios nas imagens com boas condições de luminosidade, indicando que as detecções poderão ser mais bem sucedidas em ambientes com iluminação mais uniforme. Além disso, para melhorar os valores de precisão e de revocação da detecção dos elementos, outros algoritmos podem ser testados, como, por exemplo, a utilização de redes neurais.

No cenário educacional, o robô desenvolvido pode ser usado em ambientes formais de aprendizagem e em espaços científicos interativos para realizar demonstrações que faci-

litem o aprendizado e despertem o interesse de estudantes de todos os níveis de educação. Dentre as diferentes áreas das quais se pode mostrar a aplicação prática, incluem-se a física (cinemática, eletromagnetismo), a matemática (equações, álgebra, geometria), a computação (lógica, programação, visão computacional, inteligência artificial, interfaces), controle, eletrônica e mecânica. Além disso, o projeto também permite estudos iniciais sobre a dinâmica do trânsito de forma lúdica, pois simula de forma simplificada mecanismos do trânsito em ambiente compacto.

No âmbito da pesquisa, pode-se citar o estímulo ao desenvolvimento e aprimoramento de algoritmos, do projeto da arquitetura de software e hardware e estudos relativos à extensibilidade dos módulos, como a adição de sensores, novos algoritmos e elementos no cenário. Ademais, o projeto pode ser utilizado como uma forma de divulgar a pesquisa na área de robótica e suas possibilidades em aberto, por exemplo, a busca por detectores mais robustos em visão computacional e soluções referentes a transformações em ambientes urbanos para a utilização de robôs em diversas tarefas.

Como trabalhos futuros, será realizada a integração de todos os módulos e migração para uma plataforma para processamento em tempo real utilizando um modelo cliente-servidor. Novas funcionalidades poderão ser implementadas, tais como a detecção de outras placas de sinalização e testes com a utilização de outros formatos de pista. Além disso, será executada uma avaliação da eficácia do projeto no contexto educacional.

AGRADECIMENTOS

Ao apoio do CEFET-MG por meio do Programa Bolsa de Complementação Educacional e do Programa Institucional de Iniciação Científica Voluntária. À professora Sílvia Albuquerque do CEFET-MG pela impressão 3D do semáforo utilizado na maquete. Às empresas Madeireira Paranaense e PrintBH Produções e Publicidade pela doação de material para a construção da pista.

REFERÊNCIAS

- Almeida, C., Amorim, L., Brito, F., Ferreira, J.R., and Massarani, L. (2015). *Centros e museus de ciência no Brasil 2015*. ABMC, Casa da Ciência e Museu da Vida, Rio de Janeiro.
- Architects, M.A. (2019). Seoul robot science museum. <https://www.melikealtinisik.com/2-index/134-seoul-robot-science-museum/>. Acessado em 18 de junho de 2020.
- Bradski, G.R. and Kaehler, A. (2009). *OpenCV: Open Computer Vision Library*. Disponível em <http://sourceforge.net/projects/opencvlibrary/>. Acessado em 19 de junho de 2020.
- Dorouka, P., Papadakis, S., and Kalogiannakis, M. (2020). Tablets and apps for promoting robotics, mathematics, stem education and literacy in early childhood education. *International Journal of Mobile Learning and Organisation*, 14(2), 255–274.
- Gonzalez, R.C. and Woods, R.E. (2003). *Processamento de Imagens Digitais*. Editora Edgard Blucher, 1 edition.
- Kong, S.c. and Wang, Y.Q. (2019). Nurture interest-driven creators in programmable robotics education: an empirical investigation in primary school settings. *Research and Practice in Technology Enhanced Learning*, 14(1), 20.
- Miller, D.P., Nourbakhsh, I.R., and Siegwart, R. (2008). *Robots for Education*, 1283–1301. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Pereira, F.G., Vassallo, R.F., and Salles, E.O.T. (2013). Human-robot interaction and cooperation through people detection and gesture recognition. *Journal of Control, Automation and Electrical Systems*, 24(3), 187–198.
- Pio, J., Castro, T., and Júnior, A. (2006). A robótica móvel como instrumento de apoio à aprendizagem de computação. *Anais do Simpósio Brasileiro de Informática na Educação*, 1(1), 497–506.
- Ponce, P., Molina, A., Lopez-Caudana, E., Reyes, G., and Mazón, N. (2019). Improving education in developing countries using robotic platforms. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 13(4), 1401–1422.
- Prol, L.C.A. (2006). Diferentes materiais para uso na robótica educacional: A diversidade que pode promover o desenvolvimento de diferentes competências e habilidades. In N. Marcusso, P. Brito, and M. Telles (eds.), *A Tecnologia Transformando a Educação – Casos de Aplicação*, 133–139. Fundação Bradesco.
- Tanno, C.R. (2019). Ministério da educação: Despesas primárias pagas 2014-2018 e impacto da EC no 95/2016 (teto de gastos). Technical Report Informativo Técnico No 6/2019-CONOF/CD, Câmara dos Deputados, Brasília.
- Tiddi, I., Bastianelli, E., Daga, E., d’Aquin, M., and Motta, E. (2020). Robot-city interaction: Mapping the research landscape—a survey of the interactions between robots and modern cities. *International Journal of Social Robotics*, 12(2), 299–324.
- Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, 511–518.