

# Byakugan: Visão Computacional

## RESUMO

A Olimpíada Brasileira de Robótica (OBR) propõe que seus participantes desenvolvam robôs autônomos capazes de superar desafios em uma emulação de cenário de desastre. Com a intenção de resolver o desafio específico de resgate de vítimas, este trabalho propõe tecnologias de reconhecimento e tratamento de imagem, uma forma alternativa ao que geralmente é utilizado por outras equipes participantes. Foi adicionada uma câmera em um robô (previamente desenvolvido) com intuito de capturar imagens do ambiente onde são dispostas as vítimas e uma área para resgate. Em seguida, foi utilizado o Robotic Operating System (ROS) que permitiu a realização de algumas tarefas, tais como: a execução de programas do robô de forma distribuída em várias máquinas, a utilização de outros sensores além da câmera, o acionamento de atuadores presentes no robô e o processamento das imagens, feito com a biblioteca OpenCV. Os testes iniciais indicam resultados promissores na utilização das tecnologias propostas por este trabalho.

Palavras-chave: Competição, Educação, Robótica, Visão computacional.

## ABSTRACT

The Brazilian Robotics Olympiad (OBR – Olimpíada Brasileira de Robótica), proposes their participants to develop robots capable to overcome challenges in a disaster scenario autonomously. This present work proposes an implementation of a computer vision system as an alternative way to those proposed by the OBR's challenge. This work added a camera to a robot (previously developed), aiming visual capture of the environment in which the victims and the safe area are disposed of. Then, was used the Robot Operating System (ROS), that allowed the accomplishment of some tasks like: run the robot programs in a distributed way on several machines, the utilization of some others sensors than the camera, the actuation of the actuators on the robot and the image processing, with the OpenCV library. Initial tests indicate promising results in the use of the technologies proposed by this work.

Keywords: Competition, Computer Vision, Education, Robotics.

## 1. Introdução

Sob um ponto de vista educacional, a robótica é uma tecnologia emergente que gera interessantes experiências. Essas experiências demonstram como a robótica é um agente inclusivo e que traz bons resultados em projetos coletivos que utilizam tecnologias para soluções de problemas. (ORGANIZAÇÃO OBR, 2019)

Dentro desse contexto, um importante agente de desenvolvimento da robótica no Brasil é a Olimpíada Brasileira de Robótica (OBR), organização que realiza competições desde 2007 em vários lugares do país. Acreditando que a robótica é fundamental para a expansão tecnológica, a organização incentiva jovens brasileiros a desenvolverem robôs autônomos que consigam superar desafios na emulação de um cenário de desastre. O primeiro desses desafios é dado por um caminho que o robô deve seguir. Esse caminho pode conter alguns empecilhos, como: redutores de velocidade, obstáculos e encruzilhadas (Figura 1). Logo após seguir esse caminho, o robô deverá encontrar, pegar e salvar as vítimas em um ambiente chamado “sala de resgate” (Figura 2). É nessa etapa da competição que o robô deve capturar as vítimas que são representadas por bolas (prateadas ou pretas, vivas ou mortas, respectivamente) e em seguida deixá-las em um local seguro, representado por um triângulo preto (área de resgate).

**Figura 1 – A arena utilizada nas etapas regionais e estaduais da OBR**



Fonte: Organização OBR

**Figura 2 – Sala de resgate**



Fonte: Autoria Própria

## 1.1 Objetivos específicos

No contexto da competição da OBR, em MARINATO, Gabriela *et al* e AMORIM, Junior Aguilar *et al*, os autores desenvolveram projetos de robôs autônomos baseados em técnicas de visão computacional para realizar os desafios da olimpíada. Este trabalho baseia-se nesses projetos e desenvolve uma proposta semelhante, tendo como objetivos específicos: 1. Utilizar a visão computacional para resolver o desafio do resgate de vítimas com uma maior precisão na identificação dos objetos (vítimas e áreas de resgate); 2. Solucionar alguns problemas encontrados nos trabalhos anteriores, como: a identificação de vítimas de cores distintas e o reconhecimento da área de resgate; 3. Utilizar o Robot Operating System (ROS, Sistema Operacional para Robôs), ferramenta que permitirá controlar o ambiente de execução do robô de forma remota, acompanhar os processos em tempo real e visualizar erros no momento de execução.

## 1.2 Objetivo geral

O desenvolvimento deste trabalho tem como objetivo geral contribuir para a robótica do IFRN – *Campus* Santa Cruz, pois permitirá que novos alunos desenvolvam projetos que serão baseados em visão computacional, promovendo assim, um avanço tecnológico nos robôs utilizados pelo laboratório de robótica do *campus* em competições como a OBR.

Este artigo encontra-se organizado da seguinte forma: a seção 2 apresenta a metodologia do trabalho, o resultado e discussão do projeto são apresentados na seção 3 e as considerações finais apresentadas na seção 4.

---

## 2. Metodologia

A possibilidade de que um robô autônomo com processamento de imagens pudesse ser uma forma eficiente de resolver o desafio de emulação de resgate de vítimas proposto pela OBR, despertou a ideia de acoplar em uma plataforma robótica um sistema de visão computacional. Esse sistema deve ser capaz de auxiliar o robô na localização e resgate de objetos (vítimas e área para resgate) na sala de resgate.

Inicialmente, o grupo estudou as ferramentas que seriam utilizadas: OpenCV e o Sistema Operacional para Robôs (ROS). O OpenCV é uma biblioteca *open-source* que possui várias aplicações desenvolvidas para o reconhecimento de objetos e tratamento de imagens. Essa biblioteca é inteiramente desenvolvida em C++ , sendo possível utilizá-la com Python (utilizado neste trabalho) e outras linguagens (OPENCV, 2018) . O ROS é um *middleware* que cria uma interface entre o Sistema Operacional e *software* do robô, permitindo a distribuição de programas em várias máquinas; o monitoramento desses programas em tempo de execução e uma série de outras aplicações voltadas para a área da robótica. Dois conceitos primordiais do ROS são os nós, representações de programas, e os tópicos, interfaces que permitem a troca de informações entre vários nós (EDNO, 2014). A partir desse momento, todo e qualquer código desenvolvido para o robô, funcionava no ambiente de execução do ROS, seguindo os conceitos de nós e tópicos.

### 2.1 Testes iniciais

Depois do estudo das ferramentas, a primeira etapa de desenvolvimento deste trabalho consistiu em desenvolver dois testes para validar se a proposta do trabalho seria viável. O primeiro teste foi controlar um robô a partir do reconhecimento da movimentação de mãos humanas, utilizando o OpenCV. O segundo, controlar um robô remotamente por um *joystick*, com o objetivo de obter experiência no uso do ROS.

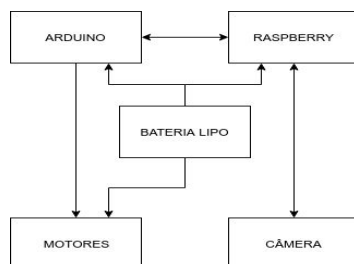
O código desenvolvido no primeiro teste utilizou a biblioteca de visão computacional OpenCV, o que permitiu aplicar as funções para mudança de cor da imagem (`cv2.cvtColor()`) e identificação dos contornos da mão (`cv2.findContours()`). Os códigos desenvolvidos com essas funções comprovaram o poder computacional da biblioteca no uso de programas para a identificação de objetos em tempo real.

Já o segundo teste permitiu identificar que o ROS era uma opção factível para ser utilizada no robô que o trabalho pretendia utilizar. Foram desenvolvidos alguns nós para a comunicação entre duas máquinas, permitindo dividir programas de forma distribuída. Além disso, foi possível visualizar a imagem capturada pela câmera do robô remotamente e em tempo real.

## 2.2 Plataforma robótica

Logo após a realização desses testes, a próxima etapa foi a aplicação do conhecimentos adquiridos em uma plataforma robótica. A plataforma escolhida foi o robô desenvolvido pela equipe Wall-E, do IFRN – Campus Santa Cruz, pois o mesmo já possuía sensores necessários para a realização dos desafios da OBR (ARAÚJO, 2019, no prelo). A Figura 3 exemplifica a estrutura física do robô.

**Figura 3 – Estrutura física simplificada do robô**



Fonte: Autoria Própria

Na elaboração da arquitetura física foram escolhidos dois componentes principais: o Raspberry Pi 3 e o Arduino Mega. Por questões econômicas é comum que equipes utilizem o Arduino como núcleo de todo o processamento. Entretanto, para que este trabalho pudesse usufruir dos processos de visão computacional, o Arduino não seria uma boa opção, pois não possui poder computacional suficiente. Portanto, este trabalho optou em utilizar a plataforma Raspberry Pi 3 para cuidar do processamento das imagens, já que mostra eficiência nessas tarefas, como mostra G. Senthilkuma *et al.* Além disso, o Raspberry Pi 3 possui capacidade suficiente para processar, além das imagens, vários outros dados dos componentes eletrônicos do robô.

## 2.3 Processamento de imagens

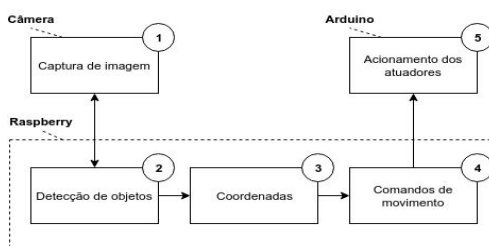
Na terceira fase de desenvolvimento deste projeto, o grupo dividiu as tarefas em duas etapas: identificar as vítimas (bolas) e identificar a área de resgate (retângulo, na visão do robô). Desenvolvidas essas identificações, foi possível diferenciar as cores das vítimas (pretas e prateadas) e identificar a área de resgate em diferentes posições do robô na sala de resgate.

Na identificação das vítimas, foi utilizada uma função do OpenCV chamada *cv2.houghCircle()*, essa função retornava as informações necessárias para o tratamento e tomada de decisões a partir dos dados adquiridos. Para tal, a função utiliza de uma adaptação do método matemático (transformada de Hough), cujo detalhamento foi abstraído pela biblioteca. Essa abstração permite o foco no tratamento específico das coordenadas, cuja composição é formada pelo centro do círculo (x e y) e seu raio. A partir disso, é possível obter a distância e a posição do robô com relação à vítima.

Já com relação à identificação da área de resgate, foram utilizadas outras funções do OpenCV, como: *cv2.cvtColor()* para modificar a cor da imagem para cinza; *cv2.threshold()* para transformação da imagem cinza em apenas duas tonalidades (preto e branco), resultando em uma imagem “binarizada”; *cv2.findContours()* para obter as coordenadas dos contornos identificados na imagem “binarizada”. A partir de todas essas etapas sequenciais e preparatórias, as coordenadas resultantes do *cv2.findContours()* eram interpretadas em vários métodos de tratamento desenvolvidos por este trabalho. Os métodos conseguem distinguir um objeto retangular grande (objeto esperado na identificação da área de resgate) de pequenos contornos resultantes de ruídos na imagem. E, finalmente, depois de interpretar um conjunto de coordenadas como sendo a área de resgate, a coordenada central é disponibilizada.

Na penúltima fase de desenvolvimento, o grupo conseguiu criar a comunicação entre o Raspberry e o Arduino, permitindo que o Raspberry, em conjunto com o Arduino, controle os atuadores e colete informações dos sensores do robô. Essa etapa foi importante para comprovar o poder na estratégia de divisão das tarefas do robô nas duas plataformas: o Raspberry ficava responsável pelo processamento de todas as informações, enquanto o Arduino atuava como “servo” capaz de receber os comandos de movimentação enviados pelo Raspberry. A Figura 4 mostra como essas plataformas agem em conjunto de maneira simplificada.

**Figura 4 – Arquitetura lógica do processamento de imagens**



Fonte: Autoria Própria

O processo começa quando o Raspberry solicita uma imagem da câmera do robô e a disponibiliza para a próxima etapa. Na seção de detecção de objetos, são utilizadas as funções da biblioteca de visão computacional OpenCV. É nessa etapa que imagem alterada é “binarizada”, processo responsável pela transformação das tonalidades em apenas duas cores: preto e branco. Com a imagem “binarizada”, o processo de identificação de objetos se torna menos complexa e mais rápida. Dos contornos identificados nas imagens “binarizadas”, são extraídas as coordenadas importantes para este trabalho: o centro dos objetos e suas posições na imagem. A partir das coordenadas, o Raspberry finalmente consegue enviar os comandos de movimento para o Arduino que serão necessários para realizar a captura e resgate das vítimas.

## 2.4 Etapa final

Na última fase de desenvolvimento, o grupo está desenvolvendo uma estratégia para que o robô consiga da melhor maneira possível executar o desafio do resgate das vítimas. A estratégia inicial está sendo baseada no trabalho de BARBOSA, André *et al.* Entretanto, a complexidade desse desenvolvimento surpreendeu as expectativas e apresenta uma dificuldade para a finalização deste projeto.

## 3. Resultados e Discussões

Nas fases iniciais de desenvolvimento, o uso da visão computacional no robô mostrou-se possível devido o processamento das imagens em tempo real e em paralelo com o uso de motores. Entretanto, em alguns casos foram notados altos níveis de processamento do Raspberry, o que evidenciou um ponto crucial na utilização da visão computacional: as imagens deviam ser de baixa resolução (320x240 pixels) para que fosse possível uma boa taxa de processamento.

### 3.1 Identificação dos objetos

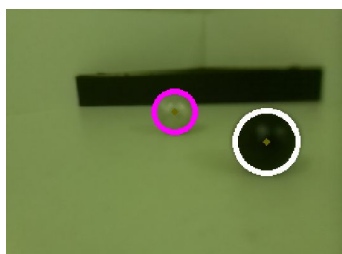
No contexto da identificação de objetos utilizando visão computacional, MARINATO, Gabriela *et al* e AMORIM, Junior Aguiar *et al* fazem uso de vítimas de coloração avermelhada pela facilidade de tratamento, o que evidencia a dificuldade de reconhecimento de vítimas prateadas, cores de pouco contraste com a sala de resgate. O presente trabalho conseguiu a identificação de vítimas com as cores prateadas e pretas (Figura 3.a). Para observar o percentual de acerto da identificação foram realizados testes (ARAÚJO, 2019, no prelo). De acordo com esses testes, as vítimas pretas são mais fáceis de identificar do que as prateadas, provavelmente devido ao contraste entre o branco e preto. A tabela 1 mostra os resultados específicos do experimento de identificação.

**Tabela 1 – Percentual de identificação dos objetos**

Cor	Percentual
Prateada	28%
Preta	76%

Apesar da aparente facilidade de identificação das vítimas mortas na primeira conclusão, essas podem se tornar quase que impossíveis de se detectar quando estão em uma mesma perspectiva da área de resgate (de cor totalmente preta), o que não acontece com as vítimas brancas, que ficam muito evidentes com relação às tonalidades escuras. Também foi notada uma facilidade de identificação quando as vítimas estavam mais próximas da câmera, o que permite uma discussão futura sobre qual seria a distância ideal para identificar as vítimas. Segue na Figura 5 a identificação das vítimas.

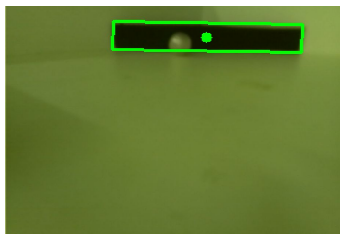
**Figura 5 - (a) Identificação das vítimas prateadas e pretas;**



Fonte: Autoria Própria

MARINATO, Gabriela *et al* deixa claro também a necessidade de um sistema de identificação da área de resgate (região segura para as vítimas). Em fase final de desenvolvimento, a detecção da área de resgate deste trabalho mostra que mesmo uma vítima prateada obstruindo a visão completa da área, essa é identificada na Figura 6 por um grande retângulo preenchido preto (na perspectiva do robô) circundado por um retângulo verde. Em testes quantitativos foi observado que o software consegue identificar a área de resgate em 56% dos casos (ARAÚJO, 2019, no prelo), o que denota a necessidade de melhorias na calibração para aumentar o percentual de identificação.

**Figura 6 - Identificação da área de resgate.**



Fonte: Autoria Própria

#### 4. Considerações Finais

Na OBR 2019, a utilização da visão computacional para a resolução de um dos desafios da competição assegura a intenção deste trabalho: a equipe Wall-E (objeto de trabalho deste projeto) recebeu o prêmio de inovação, evidenciando que a proposta possui relevância tecnológica e que pode inspirar outras equipes participantes a utilizarem robôs desenvolvidos com base no reconhecimento e tratamento de imagens.

Apesar de algumas dificuldades e tarefas incompletas, este trabalho vem contribuindo de forma significativa para o desenvolvimento da robótica no IFRN - *Campus* Santa Cruz, permitindo que as tecnologias propostas possam ser desenvolvidas por novos alunos.

A proposta apresentada abre direções para trabalhos futuros que poderão superar os problemas deste trabalho e aperfeiçoar suas capacidades. Para trabalhos futuros, propomos:

1. Aperfeiçoamento das estratégias de utilização dos algoritmos de tratamento de imagem;
2. Utilizar esta proposta na Olimpíada Brasileira de Robótica (OBR);
3. Desenvolver documentação do uso de técnicas de visão computacional para robôs utilizados na OBR.

#### Agradecimentos

Gostaríamos de agradecer ao Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, por ajudarem no desenvolvimento de milhares de pessoas e por acreditarem que os robôs desenvolvidos em conjunto pela comunidade acadêmica são de extrema importância para a expansão do uso de tecnologias educacionais.

Agradecemos também o nosso orientador, familiares, apoiadores, Olimpíada Brasileira de Robótica e a Mostra Nacional de Robótica.

#### Referências

- MARINATO, Gabriela P.; SOARES, Joyce A. P.; AMARAL, Eduardo M. A. Sistema de detecção e resgate de vítima para um robô autônomo seguidor de linha baseado em visão computacional. **Mostra Nacional de Robótica (MNR)**, Serra - ES - Brasil, 2017. Disponível em: <http://sistemaolimpico.org>. Acesso em: 16 abr. 2019.
- ARAÚJO, Douglas Gabriel; LOURENÇO, Isaac Marlon; BORGES, Paulo Vitor. Byakugan: Visão Computacional. **Anais da Mostra Nacional de Robótica - MNR 2019**, [s. l.], 2019. No prelo.
- AMORIM, Junior Aguilar et al. Implementação de um robô para competição baseado em visão computacional. **Anais da Mostra Nacional de Robótica**, <http://www.mnr.org.br/>, 2018.
- BARBOSA, André; FAGUNDES, Elizandro Gomes; DARÓS, Renan Rocha. Uma metodologia para a resolução do desafio da olimpíada brasileira de robótica – O robô Volk. **5º Simpósio de Integração Científica e Tecnológica do Sul Catarinense – SICT-Sul**, 2016. Disponível em: <http://docente.ifsc.edu.br/werther/publicacoes/2016SITCSUL-2a.pdf>. Acesso em: 10 ago. 2019.
- ARRUDA, Gabriel Augusto; MARIANO, Vinicius Barbosa Pereira Mariano; SALES, Vinicius Gambi. Otimização de detecção de rotas e reconhecimento de objetos para Robocup Junior Rescue por meio de processamento digital de imagem e visão computacional. **MNR 2017**, Londrina – PR, 2017. Disponível em: <http://www.mnr.org.br/wpcontent/uploads/2019/06/MNR-Anais2017.pdf>. Acesso em: 11 ago. 2019.
- EDNO, Francisco. **Tutoriais sobre ROS**. [S. l.], 14 dez. 2014. Disponível em: [http://wiki.ros.org/pt\\_BR/ROS/Tutorials](http://wiki.ros.org/pt_BR/ROS/Tutorials). Acesso em: 15 mar. 2019.
- G. Senthilkuma, K. Gopalakrishnan, V. Sathish Kumar. EMBEDDED IMAGE CAPTURING SYSTEM USING RASPBERRY PI SYSTEM. **International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)**. Volume 3, Issue 2, March – April 2014.
- OPENCV, Dev Team. OpenCV 2.4.13.7 documentation. [S. l.], 12 jul. 2018. Disponível em: <https://docs.opencv.org/2.4.13.7/>. Acesso em: 15 mar. 2019.
- ORGANIZAÇÃO OBR. Por que uma olimpíada de robótica? [S. l.: s. n.], 2018. Disponível em: <http://www.obr.org.br/>. Acesso em: 16 abr. 2019.
- ORGANIZAÇÃO OBR. Manual de Regras e Instruções – Etapa Regional/Estadual. **Manuais OBR**, [s. l.], 2019. Disponível em: [http://www.obr.org.br/manuais/OBR2019\\_MP\\_ManualRegionalEstadual.pdf](http://www.obr.org.br/manuais/OBR2019_MP_ManualRegionalEstadual.pdf). Acesso em: 1 nov. 2019.