

# Uso da Visão Computacional para Localização do Jogador no Protótipo de uma Máquina Inteligente de Arremesso de Bolas de Tênis

Maurício J.V.Amorim<sup>1</sup>, Flávio Duarte Pedrosa<sup>1</sup>, Lucas Azevedo<sup>1</sup>, Luiz G.L.Moura<sup>1</sup>

<sup>1</sup>Instituto Federal de Educação, Ciência e Tecnologia Fluminense  
Av. Dr. Siqueira, 273 – Campos dos Goytacazes – RJ – Brazil

amorim@iff.edu.br, flaviopedrosa99@hotmail.com,  
lukasiazvedo@hotmail.com, lglmoura@iff.edu.br

**Abstract:** *This article demonstrates the use of Computational Vision techniques in the tennis player's location. The project is one of the bases that will be used in the construction of a prototype of an Intelligent Tennis Ball Throwing Machine. Its operation is based on the acquisition of the images of the court, through a camera; its treatment with techniques of: cutting, smoothing, thresholding and transformations of color and morphological spaces; and finally the detection of objects and consequently detection of the player. The article also discusses the difficulties of execution in low performance computing environments*

**Resumo:** *Este artigo demonstra a utilização de técnicas de Visão Computacional, na localização do jogador de tênis. O projeto é uma das bases que serão utilizadas na construção de um protótipo de uma Máquina Inteligente de Arremesso de Bolas de Tênis. Seu funcionamento baseia-se na aquisição das imagens da quadra, através de uma câmera; seu tratamento com técnicas de corte, suavização, limiarização e transformações de espaços de cor e morfológicas; e por fim a detecção de objetos e conseqüentemente o jogador. O artigo aborda também as dificuldades de execução em ambientes computacionais de pouco desempenho.*

**Palavras Chaves:** *OpenCV, Visão Computacional, Python.*

## 1. Introdução

Vive-se no mundo onde cada vez mais as tarefas são automatizadas. Hoje a limpeza da casa pode ser realizada através de um aspirador robô, dotado de inteligência para otimizar o caminho a ser percorrido. Num carro percebe-se esta automatização através do farol que acende automaticamente ou do limpador que liga quando chove.

Dessa forma, quando um dos autores resolveu aprender a jogar tênis, deparou-se com a necessidade de se construir uma Máquina Inteligente de Arremesso de Bolas de Tênis (MIABT), que lançasse de forma otimizada as bolinhas ao aprendiz. A arquitetura da MIABT pode ser vista na Figura 1.

Pode-se observar pela Figura-1 que a MIABT tem como premissa no fato de seu sensoriamento ser efetuator através das imagens obtidas de uma Câmera de Vídeo acoplada ao um computador. No computador rodaram todos os módulos:

- O Módulo de Detecção tem a finalidade de tratar a imagem obtida da câmera, reconhecendo a posição do jogador (aprendiz) e informar a posição deste ao Módulo de Comando.
- O Módulo de Comando deve calcular os parâmetros de arremesso da bola de acordo com a forma de treinamento pretendida.
- O Módulo de Lançamento receberá os parâmetros a sua direção e velocidade e comandará a parte mecânica para que se execute ação desejada.

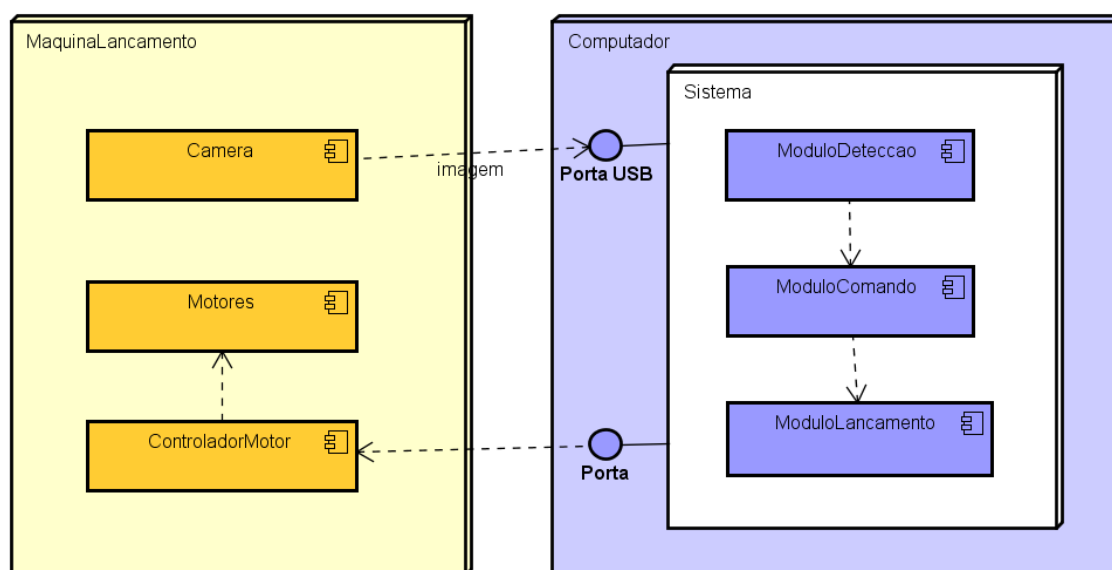


Figura 1 – Arquitetura da Máquina Inteligente de Arremesso de Bolas de Tênis

Este artigo cuidará exclusivamente da aquisição e tratamento da imagem; e da localização do jogador, funções inerentes ao Módulo de Detecção. Inicialmente será apresentada uma descrição da plataforma utilizada no projeto: Raspberry PI, Python, OpenCV. Em seguida é apresentada a solução e suas etapas. Por último os resultados serão resumidos com suas perspectivas positivas e negativas.

## 2. Tecnologia Utilizadas

### 2.1. Raspberry Pi

Dada as considerações do projeto, é necessário que o MIABT seja portátil, para maior facilidade de transporte da máquina de acordo com local de treinamento. Portanto, a escolha do Raspberry como Hardware para ser embarcado o projeto se deu devido ao seu tamanho reduzido, além, do seu baixo custo, com custo de US\$ 25 em média, e facilidade de desenvolvimento com Linux.

O Raspberry Pi é um computador de tamanho reduzido que contempla um microprocessador modelo ARM1176JZF-S de 700 MHz embutido em um chip BCM2835 da Broadcom. Além disso, tem interface Ethernet, entrada para cartão SD (Secure Digital), 512 MB de memória de SDRAM, portas USB, saídas de vídeo RCA e HDMI, saída de áudio, pinos de entrada e saída.

Dito isso, apesar do seu tamanho reduzido, o Raspberry é considerado um computador como outro qualquer, com suas determinadas limitações. Logo, para a elaboração do projeto, o Raspberry é uma excelente opção devido a suas características de processamento em conjunto com seu tamanho reduzido, e pela opção de trabalho com a linguagem Python, que será explicado nos capítulos seguintes.

## 2.2. OpenCV

A OpenCV é uma biblioteca multiplataforma de visão computacional, muito popular em todo o mundo, com grandes comunidades de usuários na China, Japão, Rússia, Europa e Israel. É liberado sob a licença BSD e, portanto, é gratuito tanto para uso acadêmico e comercial. Possui interfaces para C++, C, Python e Java é suportado pelos sistemas operacionais Windows, Linux, Mac OS, iOS e Android. OpenCV foi projetado para ter eficiência computacional e com um forte foco em aplicações em tempo real. Escrito em C / C++ otimizado, a biblioteca também pode tirar proveito do processamento multi-core (BRADSKI e KOEHLER, 2008).

A biblioteca OpenCV apresenta diversas funções para tratar a imagem, e muitas delas foram utilizadas para a execução do projeto. Desde a captura de um frame do vídeo, redefinição do tamanho da imagem, retirada de ruídos, até os cálculos para determinar onde está o jogador a ser localizado.

A escolha da OpenCV se deu por ser de licença livre, de fácil manuseio, pela disponibilidade de bibliografia e pela ampla gama de linguagens de programação suportada.

## 2.3 Python

Python é uma linguagem de altíssimo nível (*VHLL - Very High Level Language*), de sintaxe moderna, orientada a objetos, interpretada via *bytecode*, com tipagem forte (não há conversões automáticas) e dinâmica (não há declaração de variáveis e elas podem conter diferentes objetos), modular, multiplataforma, de fácil aprendizado e de implementação livre.(PYTHON, 2015).

A escolha da linguagem Python para o projeto ocorreu devido a uma das principais características da linguagem, que é o fato de ser uma linguagem de alto nível, mais próximo à linguagem humana, facilitando então, a elaboração do software. Assim, Python foi escolhido pois é extremamente legível, isso significa que é muito fácil compreender programas escritos há algum tempo. É muito comum que os programas em atividades científicas sejam criados a partir da evolução de algoritmos anteriores. Portanto, é extremamente importante ser capaz de entender o que foi feito antes. Uma vez que as palavras-chave da linguagem Python são voltadas para a estruturação dos programas (e não para indicar ao computador como compilar ou interpretar trechos de código), não existem trechos de código que são inúteis para o raciocínio. Além disso, como foi citado

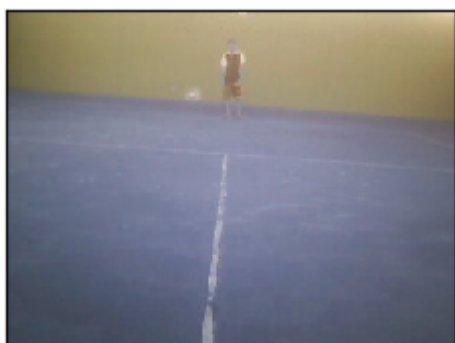
anteriormente, a biblioteca OpenCV que está presente em grande parte da programação referente à localização do jogador é compatível com a linguagem.

### 3. Módulo de Detecção

O módulo de detecção tem como entrada uma câmera que fornece uma imagem da quadra com o jogador em determinada posição. A detecção baseia-se em determinar a posição do jogador dentro da quadra.

A câmera utilizada obtém fotos/vídeos de qualidade \_\_\_\_\_. Processo inicia com a obtenção da imagem. A Figura 1 mostra imagem original com a quadra com o jogador ao fundo. O passo seguinte foi transformar a imagem em tons de cinza (Figura 2), pois os algoritmos de binarização/limiarização utilizados nos últimos passos exigem este formato de imagem e realizando desde o início gera uma otimização de recursos de memória.

Pelo mesmo motivo optou-se pelo corte da imagem (Figura 3), trabalhando apenas com o fundo da quadra. Esta é a área onde espera-se que o jogador se encontre. Para diminuir o contraste a imagem foi suavizada, atenuando a diferença entre cores de objetos próximos (Figura 4). Estas 4 etapas foram aqui denominadas de pré-processamento.



*Figura 1- Imagem Original*



*Figura 2- Imagem em GrayScale*

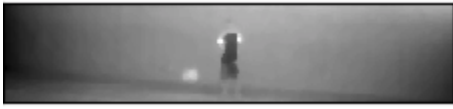


*Figura 3- Imagem Cortada*



*Figura 4- Imagem Suavizada*

Na etapa seguinte foram tentados 3 métodos diferentes de limiarização binária sobre a imagem suavizada, todos trabalhando de forma invertida: Simple Thresholding (Figura 6), Adaptive Thresholding Mean (Figura 7), Adaptive Thresholding Gaussian (Figura 8). Observa-se que neste a Limiarização Adaptativa Gaussiana conseguiu um destaque melhor sobre o objeto jogador, com o mínimo um ruído menor que a Adaptive Thresholding Mean.



*Figura 5- Imagem Suavizada*



*Figura 6- Binarização Invertida*

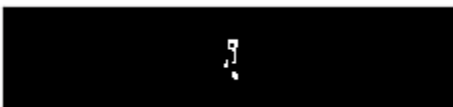


*Figura 7- Binarização Adaptativa Mean*



*Figura 8- Binarização Adaptativa Gaussiana*

É necessário um pós-processamento com objetivo retirar o ruído que ainda restava na imagem. Para isso realizou-se uma erosão (Figura 9) e a seguir uma dilatação (Figura 10).

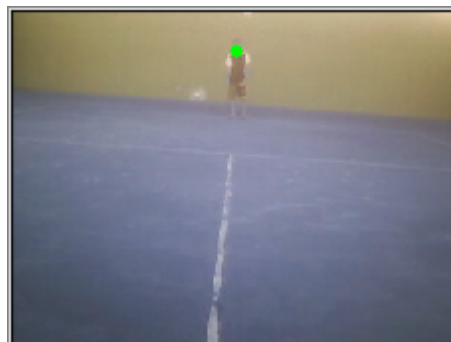


*Figura 9- Erosão*



*Figura 10- Dilatação*

Por último, é utilizada a função “findContours” que retorna o centro dos objetos das formas detectadas na imagem. A Figura 11 mostra um ponto verde pintado onde foi encontrado o jogador.



*Figura 11- Jogador Detectado*

#### **4. Considerações Finais**

Este estudo apresentou a arquitetura de uma Máquina Inteligente de Arremesso de Bolas de Tênis (MIABT). Um dos módulos da máquina é o Módulo de Detecção do Jogador. Ele roda dentro de um computador Raspberry Pi. Este computador é reduzido em dimensões e performance gerando limitações aos métodos utilizados.

O módulo de detecção foi desenvolvido na linguagem Python 3 e utiliza a biblioteca OpenCV. Seu funcionamento é detalhado na seção anterior. Durante os testes observou-

se que o computador é extremamente lento no processamento ao ponto de para teste do algoritmo optou-se pela utilização de um notebook. Por isso não foram colocados neste trabalho os números referentes a performance obtida.

Um passo importante que pode se utilizado em trabalhos futuros é a resolução do mesmo problema através da subtração do fundo. Lembrando que os níveis de iluminação mudam durante o jogo.

## **Referências**

GARRET, Felipe. Como funciona o Raspberry Pi? Entenda a tecnologia.2014 Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2014/11/como-funciona-o-raspberry-pi-entenda-tecnologia-e-sua-aplicabilidade.html>> Acesso em: 27/07/2017.

MENDES, Nilo – Perguntas frequentes sobre o Python. 2015. Disponível em: <<https://wiki.python.org.br/PerguntasFrequentes/SobrePython>>.Acesso em: 27/07/2017.

WIKIDOT. Python: O que é? Por que usar?. Disponível em: <<http://pyscience-brasil.wikidot.com/python:python-oq-e-pq>>. Acesso em: 27/07/2017.

WIKIVERSIDADE. Introdução às Linguagens de Programação/Python. Disponível em: <[https://pt.wikiversity.org/wiki/Introdução\\_às\\_Linguagens\\_de\\_Programação/Python](https://pt.wikiversity.org/wiki/Introdução_às_Linguagens_de_Programação/Python)>. Acesso em: 27/07/2017.