

# Visão Computacional Aplicada em Robótica

---

## Exercício 1: Limiarização e Segmentação

### Instruções Gerais

Os programas devem ser escritos com auxílio da biblioteca OpenCV, e seus códigos devem ser entregues com documentação de uso ou devem ser auto-explicativos, prontos para serem compilados. Os arquivos devem ser entregues compactados em formato ZIP ou GZ (não utilizar outro formato de compactação!). O nome do arquivo ZIP ou GZ deve conter o nome e sobrenome do aluno, exemplo, *lista1\_joaosilva.zip*. Não esquecer de entregar as respostas às perguntas (incluindo as saídas dos programas) em relatório em PDF.

### Imagens

Considere a imagem de cartões coloridos (Fig. 1) e a imagem de moedas (Fig. 2).

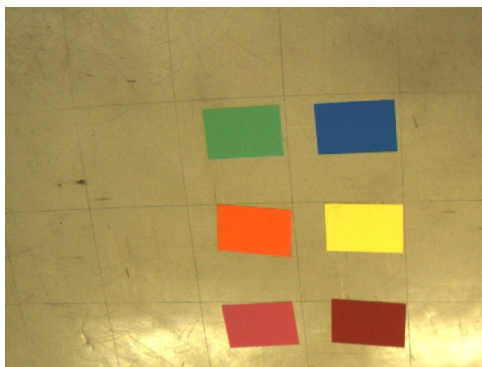


Figura 1: cores.jpeg



Figura 2: moedas.jpg

## Parte A: Limiarização e segmentação manual

O processo de limiarização (thresholding) converte uma imagem em nível de cinza para uma imagem binária. Geralmente é usado para separar os objetos de interesse do restante da imagem quando há um contraste marcante entre esses objetos e o resto da imagem, ou seja, o plano de fundo da imagem (ou *background*). Por exemplo, quando os objetos são escuros em um fundo claro, ou vice-versa. Dessa forma, os pixels do objeto de interesse podem ser marcados por preto e o fundo por branco, ou vice-versa. A separação dos objetos de interesse do restante da imagem é chamada de segmentação. Na Limiarização manual, ajusta-se um valor de limiar (threshold),  $\tau$ , de tal forma que se um valor de pixel da imagem,  $p(i, j)$ , for maior que o limiar, o resultado do processo de limiarização para aquele pixel é 255 (branco), caso contrário é 0 (preto), ou seja:

$$o(i, j) = \begin{cases} 255 & \text{se } p(i, j) > \tau \\ 0 & \text{se } p(i, j) \leq \tau \end{cases} \quad (1)$$

onde  $o(i, j)$  é o pixel resultante do processo.

1. Converta as imagens colorida fornecidas para uma imagem em níveis de cinza.
2. Encontre um valor de  $\tau$  manualmente, de forma que segmente o maior número possível de moedas e cartões nas imagens em nível de cinza.
3. Modificar o algoritmo de limiarização e segmentação, de forma a utilizar dois limiares,  $\tau_1$  e  $\tau_2$ , para definir uma faixa de valores de interesse (ou de não interesse) para a binarização, por exemplo:

$$o(i, j) = \begin{cases} 255 & \text{se } \tau_1 > p(i, j) > \tau_2 \\ 0 & \text{caso contrário} \end{cases} \quad (2)$$

Aplique nas imagens de nível de cinza para tentar melhorar o resultado obtido.

4. Pesquise sobre as operações morfológicas de erosão (erosion), dilatação (dilation), abertura (opening) e fechamento (closing). Aplique essas operações na quantidade e ordem que achar conveniente para tentar melhorar os resultados dos itens anteriores (pode utilizar essas operações nos demais itens da lista). Não se esqueça de documentar quais operadores foram aplicados e a sequência em que foi aplicado.

## Parte B: Limiarização de Otsu

Converta as imagens colorida fornecidas para uma imagem em níveis de cinza.

O método de limiarização de Otsu permite a seleção automática do limiar (threshold) para a segmentação binária da imagem. O método também permite a seleção de múltiplos limiares para que se possa segmentar em múltiplas classes. Neste caso, diferente da binarização, define-se o número de classes (ou valores) que a saída pode assumir.

1. Pesquise o método de Otsu e explique de forma resumida o método.

2. Converta as imagens colorida fornecidas para uma imagem em níveis de cinza.
3. Aplique o método de Otsu para binarização das imagens em nível de cinza.
4. Aplique o método de Otsu para múltiplas classes nas imagens em nível de cinza, e mostre os resultados utilizando um nível de cinza para cada classe. Comente sobre a escolha do número de classes para resolver o problema de segmentação de moedas e dos cartões. Comente também se o uso de múltiplas classes ajudou na segmentação das moedas e dos cartões.

## Parte C: Mudança de espaço de cores

Faça uma pesquisa para aprender mais sobre o espaço de cor RGB e sobre o espaço de cor HSV.

1. Transforme as imagens coloridas fornecidas do espaço de cor RGB para o espaço de cor HSV. No espaço de cor HSV, aplique a segmentação de Otsu para um ou mais níveis no componente H de forma a segmentar as moedas e os cartões coloridos. Mostre o resultado em uma nova imagem do mesmo tamanho da original, onde cada região segmentada aparece em uma cor uniforme colorida artificialmente. Exemplo: cartão verde é colorido com  $RGB = (0; 255; 0)$ , cartão azul com  $RGB = (0; 0; 255)$ , etc. O fundo pode ser colorido como preto pois não há interesse nele  $RGB = (0; 0; 0)$ . Se preferir, você pode também utilizar o valor RGB médio dos pixels originais da região segmentada para colorir artificialmente o resultado.
2. Discuta os resultados da segmentação baseada em cor comparando com o resultado da segmentação baseada em nível de cinza. Se os resultados forem complementares, combine o método de segmentação baseada em cor com o método baseado em nível de cinza.

## Dicas:

- *Funções úteis para esta tarefa:*

- *Mudança de espaço de cores:*

`cv2.cvtColor(src, code) → dst`

**src**      *Imagem colorida (valores tipo "uint" 8-bit ou 16-bit).*

**code**     *Constante que define o tipo de conversão (neste exercício, usar "cv2.COLOR\_BGR2HSV").*

**dst**      *Imagem resultante em outro espaço de cores.*

- *Segmentação por limiarização:*

`cv2.threshold(src, thresh, maxval, type) → retval, dst`

**src**      *Imagem de canal único (valores tipo "float" de 8-bit ou 32-bit).*

**thresh** *Valor limiar.*  
**maxval** *Valor de saída para elementos com valores acima do valor limiar.*  
**type** *Tipo de limiarização. Verifique quais tipos de limiarização pode ser feito pelo OpenCV.*  
**retval** *Retorna o valor limiar.*  
**dst** *Imagem limiarizada.*