



# **Blobs e Etiquetagem**

7

# VISÃO POR COMPUTADOR

## Blobs e Etiquetagem

- Blobs e Etiquetagem



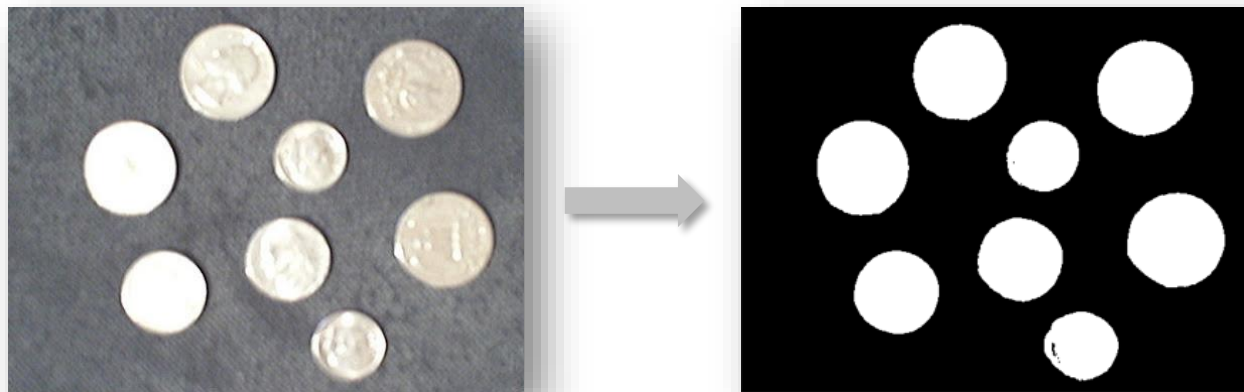
# VISÃO POR COMPUTADOR

## Blobs e Etiquetagem

- **Etiquetagem**

Através da aplicação de técnicas de binarização de imagens é possível segmentar regiões de primeiro plano, com interesse para análise.

Assim, por exemplo, através da binarização é possível definir as regiões que identificam moedas numa determinada imagem:



# VISÃO POR COMPUTADOR

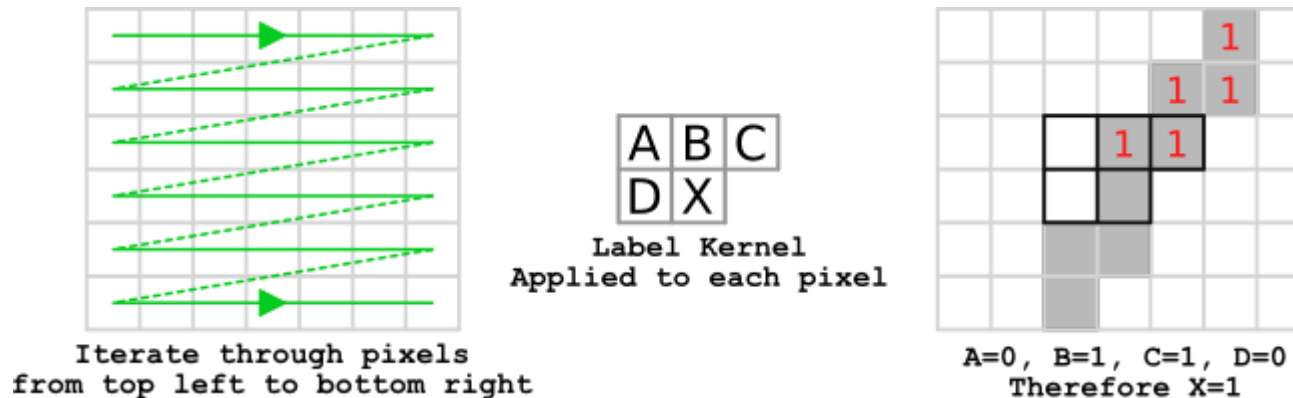
## Blobs e Etiquetagem

- **Etiquetagem**

Após realizar a binarização da imagem, é necessário **etiquetar** cada pixel de acordo com a região a que pertence (*blob*).

Este processo realiza-se verificando, para cada pixel, a etiqueta dos pixéis vizinhos.

A imagem seguinte mostra, de forma simplificada, este processo:



# VISÃO POR COMPUTADOR

## Blobs e Etiquetagem

- **Etiquetagem**

Pseudo-código:

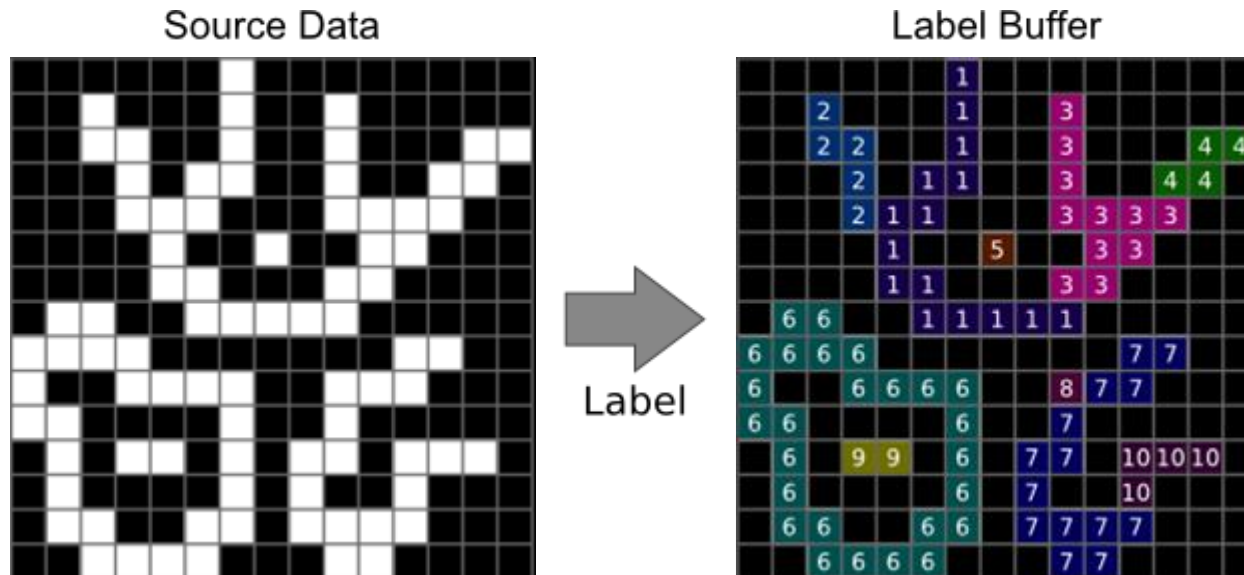
```
label ← 1 // Etiqueta inicial
Para cada pixel
  Se pixel X = 255 Então
    Se vizinhos A,B,C,D = 0 Então
      pixel X ← label // Encontrou um novo objeto (atribui-lhe nova etiqueta)
      label++
    Senão
      pixel X ← etiqueta de vizinho A,B,C,D com menor valor (não incluindo 0)
    Fim_se
  Senão
    pixel X ← 0
  Fim_se
Fim
```

# VISÃO POR COMPUTADOR

## Blobs e Etiquetagem

- **Etiquetagem**

Alguns problemas:



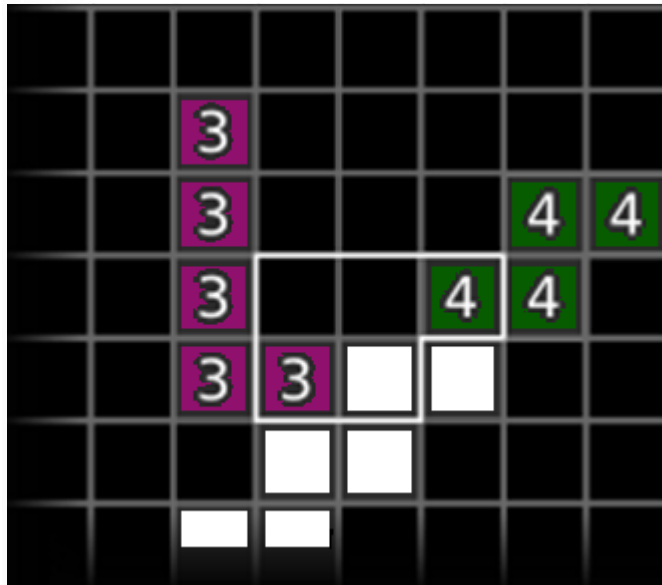
## VISÃO POR COMPUTADOR

### Blobs e Etiquetagem

- **Etiquetagem**

A solução passa por criar e manter uma tabela de etiquetas:

## Label Buffer



A=0, B=0, C=4, D=3  
Therefore X=3

## Label Table

Before

1	2	3	4
1	1	3	4

After

1	2	3	4
1	1	3	3

Updated

# VISÃO POR COMPUTADOR

## Blobs e Etiquetagem

- **Exercícios:**

- Construa a função que realiza a etiquetagem de imagens binárias.

```
int vc_binary_blob_labelling(IVC *src, IVC *dst);
```



# VISÃO POR COMPUTADOR

## Blobs e Etiquetagem

- **Área e Centro de Gravidade de um Blob**

Considere  **$B(x,y)$**  uma região segmentada, isto é, um blob, e  **$L_{max}$**  o valor de nível máximo da imagem binária.

Assim,

- **$B(x,y)=1$**  (ou 255) se o pixel da coordenada  $(x,y)$  pertencer ao blob;
- **$B(x,y)=0$**  se o pixel da coordenada  $(x,y)$  não pertencer ao blob.

Então, a **área** de um blob pode ser obtida por:

$$A = \frac{\sum_{y=0}^{N-1} \sum_{x=0}^{M-1} B(x, y)}{L_{max}}$$

O **centro de gravidade** de um blob é definido por:

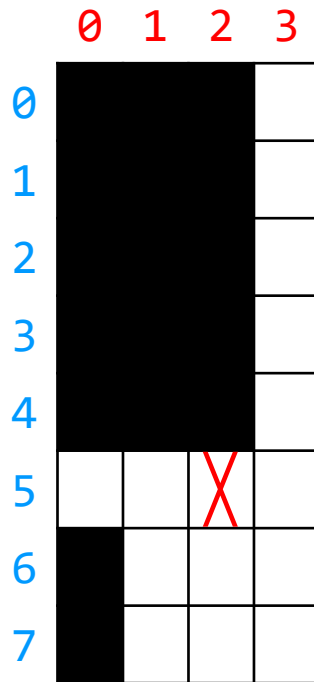
$$C(x_c, y_c) = \left( \frac{\sum_{y=0}^{N-1} \sum_{x=0}^{M-1} x * B(x, y)}{A * L_{max}}, \frac{\sum_{y=0}^{N-1} \sum_{x=0}^{M-1} y * B(x, y)}{A * L_{max}} \right)$$



# VISÃO POR COMPUTADOR

## Blobs e Etiquetagem

- Área e Centro de Gravidade de um Blob



$$\Sigma_x = 1*0 + 3*1 + 3*2 + 8*3 = 33$$

$$\Sigma_y = 1*0 + 1*1 + 1*2 + 1*3 + 1*4 + 4*5 + 3*6 + 3*7 = 69$$

$$A = 15$$

$$C_x = \Sigma_x / A = 33 / 15 = 2.2 \approx 2$$

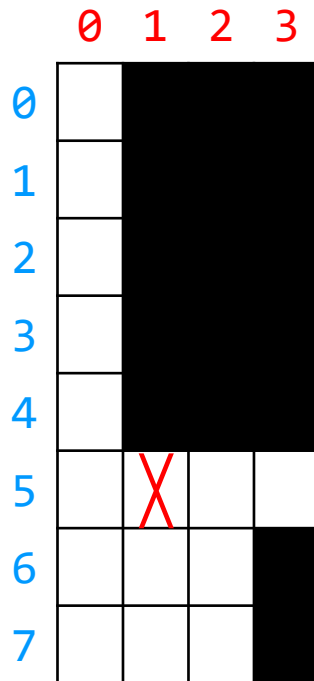
$$C_y = \Sigma_y / A = 69 / 15 = 4.6 \approx 5$$

Para arredondar, utilizar a função `round()` incluída em `math.h`.

# VISÃO POR COMPUTADOR

## Blobs e Etiquetagem

- Área e Centro de Gravidade de um Blob



$$\Sigma_x = 8*0 + 3*1 + 3*2 + 1*3 = 12$$

$$\Sigma_y = 1*0 + 1*1 + 1*2 + 1*3 + 1*4 + 4*5 + 3*6 + 3*7 = 69$$

$$A = 15$$

$$C_x = \Sigma_x / A = 12 / 15 = 0.8 \approx 1$$

$$C_y = \Sigma_y / A = 69 / 15 = 4.6 \approx 5$$

## VISÃO POR COMPUTADOR

Duarte Duque  
dduque@ipca.pt

Shukuria  
Tashakkur  
bolzin  
You  
Gracias  
Thank  
Biyang  
Grazie  
Juspaxar  
Danksheen  
Arigato  
Mehrbani  
Ekhenet  
guzamashu  
Paldies  
Komapsunrida  
Tingli  
Shukria  
Merci  
suksama  
Shukria  
Grazie  
Juspaxar



INSTITUTO POLITÉCNICO  
DO CÁVADO E DO AVE  
ESCOLA SUPERIOR DE TECNOLOGIA