

## Trabalho 4

### 1 Especificação do Problema

Este trabalho está dividido em duas partes, a primeira que calcula algumas medidas de objetos presentes em uma imagem e a segunda que realiza transformações geométricas de escala e rotação em uma imagem.

#### 1.1 Medidas de Objetos

Os principais passos para a obtenção de medidas de objetos em uma imagem são descritos a seguir.

##### 1.1.1 Transformação de Cores

Ler e exibir uma imagem colorida formada por um conjunto de objetos distribuídos em um fundo branco. A imagem colorida deve ser convertida para níveis de cinza (Figura 1).

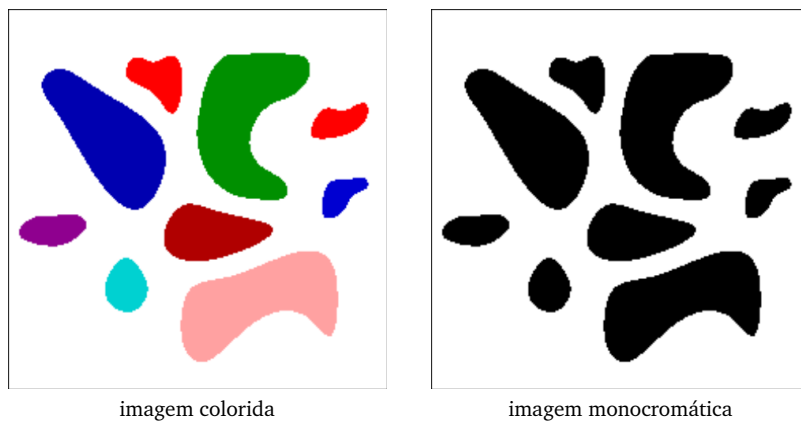


Figura 1: Conversão de imagem colorida para níveis de cinza.

##### 1.1.2 Contornos dos Objetos

Apresentar os contornos (bordas) dos objetos presentes na imagem (Figura 2).

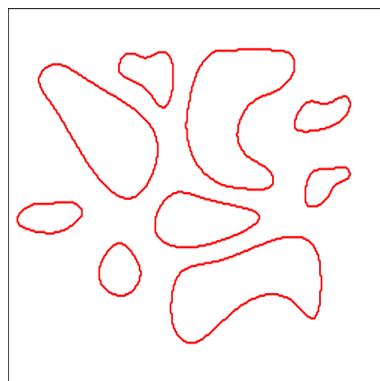


Figura 2: Contornos dos objetos.

### 1.1.3 Extração de Propriedades dos Objetos

Extrair as seguintes propriedades dos objetos: centroide, perímetro e área. Para cada região (objeto), listar o perímetro e a área.

número de regiões: 9

região 0:	área: 3969	perímetro: 313.764502	excentricidade: 0.816362	solidez: 0.747739
região 1:	área: 791	perímetro: 119.982756	excentricidade: 0.741103	solidez: 0.898864
região 2:	área: 3584	perímetro: 259.462987	excentricidade: 0.898073	solidez: 0.977899
região 3:	área: 540	perímetro: 99.254834	excentricidade: 0.889586	solidez: 0.910624
região 4:	área: 438	perímetro: 88.769553	excentricidade: 0.855923	solidez: 0.916318
região 5:	área: 1684	perímetro: 174.124892	excentricidade: 0.868169	solidez: 0.972286
região 6:	área: 642	perímetro: 103.012193	excentricidade: 0.890242	solidez: 0.969789
região 7:	área: 3934	perímetro: 305.421356	excentricidade: 0.910992	solidez: 0.774257
região 8:	área: 675	perímetro: 96.325902	excentricidade: 0.620380	solidez: 0.976845

Mostrar cada região rotulada individualmente na imagem (Figura 3).

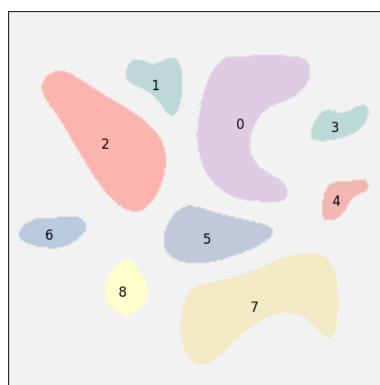


Figura 3: Regiões rotuladas.

### 1.1.4 Histograma de Área dos Objetos

Classificar os objetos de acordo com a propriedade de área. Utilizar os seguintes critérios na classificação:

objeto pequeno: área < 1500 pixels

objeto médio: área  $\geq$  1500 pixels e área < 3000 pixels

objeto grande: área  $\geq$  3000

Exemplo de saída do programa:

número de regiões pequenas: 5

número de regiões médias: 1

número de regiões grandes: 3

Apresentar um histograma com as áreas calculadas dos objetos (Figura 4).

### 1.1.5 Entrada de Dados

As imagens de entrada estão no formato PNG (*Portable Network Graphics*). Alguns exemplos encontram-se disponíveis no diretório: [http://www.ic.unicamp.br/~helio/imagens\\_objetos\\_coloridos/](http://www.ic.unicamp.br/~helio/imagens_objetos_coloridos/).

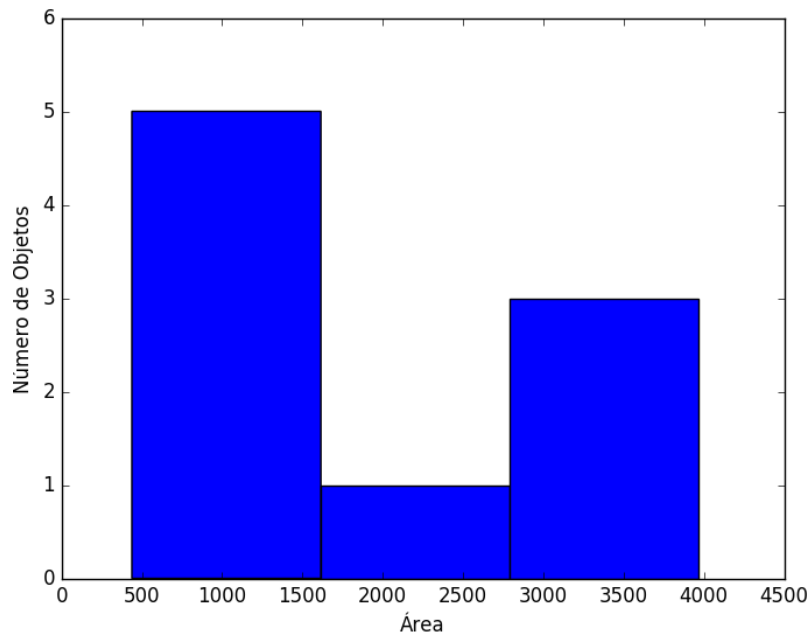


Figura 4: Histograma de áreas dos objetos.

### 1.1.6 Saída de Dados

As imagens de saída devem estar no formato PNG (*Portable Network Graphics*). Resultados intermediários podem ser também exibidos na tela.

## 1.2 Transformações Geométricas

Escrever um código para realizar as transformações geométricas de escala e rotação em uma imagem. O fator de escala e o valor do ângulo de rotação das transformadas devem permitir valores contínuos (ou seja, valores em ponto flutuante).

Um modo de se ampliar uma imagem é mapear cada posição dos pontos da imagem de saída a partir da posição correspondente dos pontos na imagem de entrada. Por exemplo, se o fator de escala é 2.25, então a posição de saída do pixel  $P_o = (10, 23)$  seria mapeado para  $P_i = P_o/s = (10/2.25, 23/2.25) = (4.444, 10.222)$  na imagem de entrada. Para determinar o valor do pixel em  $P_i$ , utilize os métodos de interpolação descritos a seguir.

### 1.2.1 Interpolação pelo Vizinho Mais Próximo

Na interpolação pelo vizinho mais próximo, o valor da intensidade a ser atribuído ao pixel  $(x', y')$  na imagem reamostrada terá o mesmo valor do pixel que estiver mais próximo da posição ocupada pelo pixel  $(x, y)$  na imagem original, conforme mostrado na Figura 5.

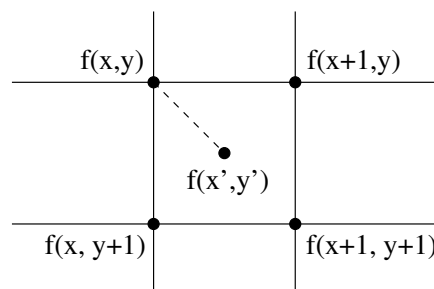


Figura 5: Interpolação pelo vizinho mais próximo.

A interpolação pode ser expressa como

$$f(x', y') = \begin{cases} f(x, y), & \text{para } dx < 0.5 \text{ e } dy < 0.5 \\ f(x + 1, y), & \text{para } dx \geq 0.5 \text{ e } dy < 0.5 \\ f(x, y + 1), & \text{para } dx < 0.5 \text{ e } dy \geq 0.5 \\ f(x + 1, y + 1), & \text{para } dx \geq 0.5 \text{ e } dy \geq 0.5 \end{cases} \quad (1)$$

em que  $dx$  e  $dy$  correspondem às distâncias nas direções  $x$  e  $y$ , respectivamente, entre os pontos  $(x', y')$  e  $(x, y)$ , ou seja, apenas as partes fracionárias do ponto  $(x', y')$

$$\begin{aligned} dx &= x' - x = x' - \lfloor x' \rfloor \\ dy &= y' - y = y' - \lfloor y' \rfloor \end{aligned} \quad (2)$$

A equação 1 pode ser reescrita como

$$f(x', y') = f(\text{round}(x), \text{round}(y)) \quad (3)$$

em que *round* é uma função que aproxima um número para seu valor inteiro mais próximo.

### 1.2.2 Interpolação Bilinear

A interpolação bilinear utiliza uma média ponderada de distância dos quatro pixels vizinhos mais próximos para determinar a intensidade de cada pixel  $(x', y')$  na imagem transformada, como mostrado na Figura 6.

A interpolação é dada por

$$\begin{aligned} f(x', y') = & (1 - dx)(1 - dy) f(x, y) + dx(1 - dy) f(x + 1, y) + \\ & (1 - dx)dy f(x, y + 1) + dxdy f(x + 1, y + 1) \end{aligned} \quad (4)$$

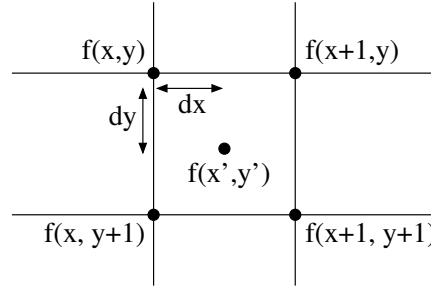


Figura 6: Interpolação bilinear.

### 1.2.3 Interpolação Bicúbica

A interpolação bicúbica utiliza uma vizinhança de  $4 \times 4$  pontos ao redor do ponto em questão para calcular seu valor de intensidade, como mostrado na Figura 7. Uma função comum para calcular as intensidades do pixel na imagem interpolada é a função B-spline cúbica, definida como

$$f(x', y') = \sum_{m=-1}^2 \sum_{n=-1}^2 f(x + m, y + n) R(m - dx) R(dy - n) \quad (5)$$

sendo

$$R(s) = \frac{1}{6}[P(s+2)^3 - 4P(s+1)^3 + 6P(s)^3 - 4P(s-1)^3] \quad (6)$$

$$P(t) = \begin{cases} t, & t > 0 \\ 0, & t \leq 0 \end{cases} \quad (7)$$

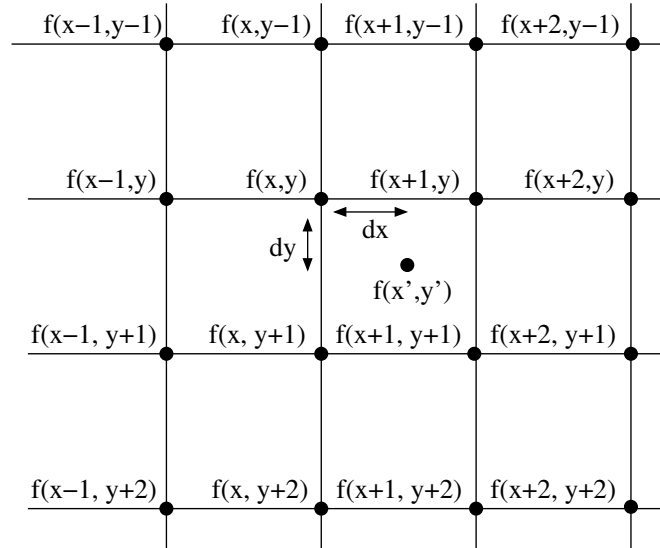


Figura 7: Interpolação bicúbica.

#### 1.2.4 Interpolação por Polinômios de Lagrange

Outro método que utiliza uma vizinhança de  $4 \times 4$  pontos para calcular o valor de intensidade de um pixel  $(x', y')$  é a interpolação por polinômios de Lagrange, definida como

$$f(x', y') = \frac{-dy(dy-1)(dy-2)L(1)}{6} + \frac{(dy+1)(dy-1)(dy-2)L(2)}{2} + \frac{-dy(dy+1)(dy-2)L(3)}{2} + \frac{dy(dy+1)(dy-1)L(4)}{6}$$

sendo

$$L(n) = \frac{-dx(dx-1)(dx-2)f(x-1, y+n-2)}{6} + \frac{(dx+1)(dx-1)(dx-2)f(x, y+n-2)}{2} + \frac{-dx(dx+1)(dx-2)f(x+1, y+n-2)}{2} + \frac{dx(dx+1)(dx-1)f(x+2, y+n-2)}{6}$$

#### 1.2.5 Especificação do Programa

Uma sugestão para execução do programa é:

```
prog [-a ângulo]
      [-e fator de escala]
      [-d largura altura]
      [-m interpolação]
      [-i imagem]
      [-o imagem]
```

em que os parâmetros são:

-a ângulo de rotação medido em graus no sentido anti-horário  
-e fator de escala

-d dimensão da imagem de saída em pixels  
-m método de interpolação utilizado  
-i imagem de entrada no formato PNG  
-o imagem de saída no formato PNG (após a transformação geométrica)

A cada execução, o programa deve realizar apenas uma transformação geométrica (escala ou rotação) na imagem. Os códigos não poderão utilizar comandos disponíveis em bibliotecas para realizar as operações descritas anteriormente, mas implementá-las de acordo com as formulações apresentadas.

### 1.2.6 Entrada de Dados

As imagens de entrada estão no formato PNG (*Portable Network Graphics*). Alguns exemplos encontram-se disponíveis no diretório: [http://www.ic.unicamp.br/~helio/imagens\\_png/](http://www.ic.unicamp.br/~helio/imagens_png/).

### 1.2.7 Saída de Dados

As imagens de saída devem estar no formato PNG (*Portable Network Graphics*). Resultados intermediários podem ser também exibidos na tela.

## 2 Especificação da Entrega

- A entrega do trabalho deve conter os seguintes itens:
  - código fonte: o arquivo final deve estar no formato *zip* ou no formato *tgz*, contendo todos os programas ou dados necessários para sua execução.
  - relatório: deve conter uma descrição dos algoritmos e das estruturas de dados, considerações adotadas na solução do problema, testes executados, eventuais limitações ou situações especiais não tratadas pelo programa.
- O trabalho deve ser submetido por meio da plataforma *Google Classroom*.
- Data de entrega: 13/06/2025.

## 3 Observações Gerais

- Os programas serão executados em ambiente Linux. Os formatos de entrada e saída dos dados devem ser rigorosamente respeitados pelo programa, conforme definidos anteriormente. Não serão aceitos trabalhos após 5 dias da data de entrega.
- Os seguintes aspectos serão considerados na avaliação: funcionamento da implementação, clareza do código, qualidade do relatório técnico.