

IF61C—Fundamentos de Programação 1

Projeto Computacional - Imagens

1 Processamento de imagens: conceitos básicos

Uma imagem digital é tipicamente definida como uma função bidimensional, $f(x, y)$, em que x e y são coordenadas espaciais (plano), e a amplitude de f no par de coordenadas (x, y) é denominado nível de intensidade ou nível de cinza da imagem naquele ponto. O número de elementos que representam uma imagem, denominados pixels (*picture elements*), é determinado pela taxa de amostragem sendo considerada. Esta é responsável pela resolução da imagem, realizada a uma melhor ou pior visualização dos detalhes de cada objeto.

O valor que cada pixel pode assumir é determinado pela quantização considerada para codificar os níveis de intensidade. Imagens em níveis de cinza tipicamente consideram uma codificação em oito bits, em que a intensidade de um pixel varia de zero (preto) a 255 (branco). As imagens binárias, por sua vez, consideram apenas dois níveis de intensidade: 0 (preto) e 1 (branco). Sendo assim, as imagens podem ser representadas por matrizes de inteiros sem sinal, sendo que cada posição corresponde a um pixel. Assim como é o caso para matrizes, em imagens o canto superior esquerdo tem coordenadas $(0, 0)$.

Outro conceito relevante é o de vizinhança. Um pixel p nas coordenadas (x, y) possui 4 vizinhos horizontais e verticais, cujas coordenadas são dadas por

$$(x + 1, y) \quad (x - 1, y) \quad (x, y + 1) \quad (x, y - 1)$$

Por exemplo, o pixel com coordenadas $(1, 2)$, representado pela cor azul, tem como vizinhança-4 os pixels em vermelho

$$\begin{array}{ccccc} 0 & 0 & \textcolor{red}{0} & 0 & 0 \\ 0 & \textcolor{red}{1} & \textcolor{blue}{1} & \textcolor{red}{1} & 0 \\ 0 & 0 & \textcolor{red}{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}$$

A vizinhança-8 considera também os vizinhos diagonais, cujas coordenadas são dadas por

$$(x + 1, y + 1) \quad (x + 1, y - 1) \quad (x - 1, y + 1) \quad (x - 1, y - 1)$$

Por exemplo, o pixel com coordenadas $(1, 2)$, representado pela cor azul, tem como vizinhança-8 os pixels em vermelho

$$\begin{array}{ccccc} 0 & \textcolor{red}{0} & \textcolor{red}{0} & \textcolor{red}{0} & 0 \\ 0 & \textcolor{red}{1} & \textcolor{blue}{1} & \textcolor{red}{1} & 0 \\ 0 & \textcolor{red}{0} & \textcolor{red}{1} & \textcolor{red}{0} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}$$

Note que a vizinhança não está definida para os pares de índices muito próximos à “borda” da matriz.

1.1 Limiarização

Basicamente a limiarização, é um método que converte uma imagem em níveis de cinza em uma imagem binária, a qual idealmente deve separar os objetos de interesse do fundo. Para tal é preciso especificar um valor, denominado limiar, que define se um determinado pixel pertence ao objeto ou ao fundo. Em suma, qualquer pixel cujo valor de intensidade é maior que o limiar é definido como pertencente ao objeto. Caso contrário, o ponto é classificado como pertencente ao fundo. Em outras palavras, a imagem limiarizada $g(x, y)$ é definida como:

$$g(x, y) = \begin{cases} 1, & \text{se } f(x, y) \geq T \\ 0 & \text{se } f(x, y) < T \end{cases} \quad (1)$$

em que T denota o limiar e os nível de intensidade 1 e 0 representam, respectivamente, objeto (branco) e fundo (preto).

1.2 Filtro da média e filtro da mediana

O filtro da média é uma transformação frequentemente usada para suavizar ruídos em sinais e imagens digitais. Dada uma matriz I de inteiros positivos, com m linhas e n colunas, o filtro da média produz uma matriz transformada M , com as mesmas dimensões que I , definida da seguinte maneira: para cada par de índices (i, j) o elemento $M(i, j)$ da matriz transformada corresponde à média dos elementos da vizinhança-8 de (i, j) na imagem de entrada I .

O filtro da mediana segue a mesma lógica. Para cada par de índices (i, j) o elemento $M(i, j)$ da matriz transformada corresponde à mediana dos elementos da vizinhança-8 de (i, j) na imagem de entrada I .

1.3 O formato PGM

Este trabalho considera imagens com o formato PGM (*Portable Gray Map*) para armazenar imagens em tons de cinza em arquivos. Este tipo de arquivo deve conter um cabeçalho seguido da matriz que armazena os valores dos pixels da imagem. Exemplo:

```
P2
5 4
255
10 20 32 42 55
11 11 25 25 36
10 11 20 22 35
11 11 13 13 13
```

A primeira linha do arquivo contém a palavra-chave P2 ou P5, sendo que a primeira é a utilizada no trabalho. Os dois números da segunda linha correspondem ao número de colunas e linhas da matriz, respectivamente (isso mesmo, colunas vem antes). A terceira linha contém um valor denominado `maxval`, que é maior ou igual ao maior valor da matriz. Os demais números correspondem aos valores da intensidade de cada pixel (neste trabalho, um inteiro entre 0 e 255). Também é possível incluir um comentário, iniciado pelo caracter `#`. Exemplo:

```
P2
# imagem: exemplo.pgm
5 4
255
10 20 32 42 55
11 11 25 25 36
10 11 20 22 35
11 11 13 13 13
```

2 Tarefas

Neste projeto computacional você vai utilizar o arquivo `processaImagem.c`. **Após analisar e entender cada função do código-fonte:**

Nível 01

1. Finalize a implementação das funções `alocaMatriz()` e `desalocaMatriz()`.
2. Implemente a função `void binarizaImagem (int **imagem, int **resultado, int ncolunas, int nlinhas)`. Depois, faça os seguintes testes:
 - (a) Tente segmentar o núcleo presente na imagem `celula.pgm` considerando diferentes limiares. Qual o valor utilizado para obter o melhor resultado?
 - (b) Tente segmentar o texto das imagens `document01.pgm` e `document02.pgm`. Quais os problemas encontrados?

Nível 02 = Nível 01 +

1. Implemente uma função que realiza a filtragem de ruído utilizando o filtro da média. Depois, filtre as imagens `tulipNoise1.pgm` e `tulipNoise1.pgm`.
2. Implemente uma função que realiza a filtragem de ruído utilizando o filtro da mediana. Depois, filtre as imagens `tulipNoise1.pgm` e `tulipNoise1.pgm`.
3. As duas técnicas anteriores apresentam o mesmo desempenho para ambos os tipos de ruído? Justifique.