

AValiação 03

Semestre: 2020.1 (remoto) Data: 19/10/2020

Nome completo:

Raul Aquino de Araújo

1. (2 pontos) Um modelo de cor é, essencialmente, a especificação de um sistema de coordenadas e um subespaço dentro desse sistema no qual cada cor é representada por um único ponto. Dois modelos de cores amplamente utilizados são o RGB e o HSI. Para cada um destes modelos de cores, explique como a cor é representada/armazenada. Responda sucintamente.

Resposta:

HSI - é a abreviatura para o sistema de cores formadas pelas componentes hue (matiz), saturation (saturação) e value (valor). O HSV também é conhecido como HSB (hue, saturation e brightness — matiz, saturação e brilho, respectivamente). Esse sistema de cores define o espaço de cor conforme descrito abaixo, utilizando seus três parâmetros: Matiz (tonalidade): Verifica o tipo de cor, abrangendo todas as cores do espectro, desde o vermelho até o violeta, mais o magenta. Atinge valores de 0 a 360, mas para algumas aplicações, esse valor é normalizado de 0 a 100%. Saturação: Também chamado de "pureza". Quanto menor esse valor, mais com tom de cinza aparecerá a imagem. Quanto maior o valor, mais "pura" é a imagem. Atinge valores de 0 a 100%. Valor (brilho): Define o brilho da cor. Atinge valores de 0 a 100%. RGB é a abreviatura de um sistema de cores aditivas em que o Vermelho (Red), o Verde (Green) e o Azul (Blue) são combinados de várias formas de modo a reproduzir um largo espectro cromático.

O propósito principal do sistema RGB é a reprodução de cores em dispositivos eletrônicos como monitores de TV e computador, retroprojetores, scanners e câmeras digitais, assim como na fotografia tradicional. Em contraposição, impressoras utilizam o modelo CMYK de cores subtrativas. Uma cor no modelo de cores RGB pode ser descrita pela indicação da quantidade de vermelho, verde, e azul que contém. Cada uma pode variar entre o mínimo (completamente escuro) e máximo (completamente intenso). Quando todas as cores estão no mínimo, o resultado é preto. Se todas estão no máximo, o resultado é branco. Se todos os valores forem iguais ou muito próximos, a resultante é cinzento; à medida que um dos valores se afastar dos outros dois a cor resultante será um tom progressivamente mais "viva" e menos "pastel".

2. (2 pontos) No contexto do processamento morfológico de imagens, alguns operadores são considerados elementares. Isto se deve ao fato da maior parte dos operadores morfológicos serem implementados a partir destes. Neste contexto, explique sucintamente o funcionamento dos operadores a seguir, **quando aplicados em imagens binárias**:
 - a. Erosão
 - b. Dilatação
 - c. Abertura
 - d. Fechamento

Resposta:

Erosão: a erosão pode ser considerada como uma operação de filtragem morfológica em que os detalhes da imagem que são menores que o elemento estruturante são filtrados (removidos) da imagem. Adicionalmente, a erosão também é conhecida como uma operação de afinamento, uma vez que sua aplicação em imagens com objetos longos produz esse efeito. Sejam A e B conjuntos de Z^2 , a erosão de A por B, indicada por $A \ominus B$, é definida como: $A \ominus B = \{z | (B)_z \subseteq A\}$. Neste contexto, B é o elemento estruturante e A é o conjunto de objetos da imagem a sofrer a erosão.

Dilatação: ao contrário da erosão, a dilatação "aumenta" ou "engrossa" os objetos em uma imagem binária, com forma e extensão desse espessamento sendo controladas pelo formato e tamanho do elemento estruturante. Sejam A e B conjuntos de Z^2 , a dilatação de A por B, indicada por $A \oplus B$, é definida como: $A \oplus B = \{z | (B)_z \cap A \neq \emptyset\}$.

Abertura: a abertura geralmente suaviza o contorno de um objeto, rompe os istmos (partes estreitas que ligam duas partes maiores) e elimina saliências finas. A abertura do conjunto A pelo elemento estruturante B, indicado por $A \circ B$, é dada por: $A \circ B = (A \ominus B) \oplus B$. Desta forma, a abertura de A por B é a erosão de A por B, seguida de uma dilatação do resultado por B.

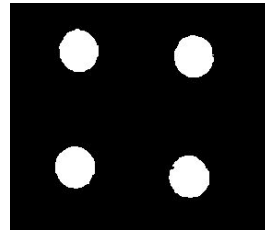
Fechamento: o fechamento também tende a suavizar contornos, mas ao contrário da abertura, geralmente funde as descontinuidades estreitas e alonga os golfos finos, elimina buracos e preenche as lacunas em um contorno. Por sua vez, o

fechamento do conjunto A pelo elemento estruturante B, indicado por $A \square B$, é dado por: $A \square B = (A \oplus B) \ominus B$. Assim, o fechamento de A por B é simplesmente a dilatação de A por B, seguida pela erosão do resultado por B.

3. (3 pontos) A imagem de raios-X de uma placa de circuito foi previamente limiarizada. Deseja-se, a partir desta limiarização, verificar automaticamente se a placa contém 4 buracos e se o diâmetro destes buracos está situado na faixa de tolerância entre 40 e 50 pixels. Utilizando a ferramenta Octave e a imagem 'circuit_holes.bmp', faça o que se pede:



Imagem original



Resultado esperado

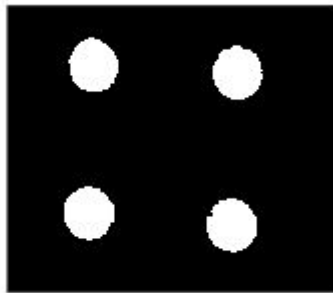
- a. Proponha um método de processamento morfológico capaz de verificar automaticamente a quantidade de buracos existentes na placa de circuito e calcular seus respectivos diâmetros

Resposta:

Fazemos um preenchimento de buraco na imagem original, depois subtraímos as duas imagens, resultando em uma imagem somente com os círculos. No fim usando o fim usamos uma extração de componentes e descobrimos quantos buracos existem e usando os métodos passados como dica abaixo, conseguimos descobrir o diâmetro da imagem.

- b. Implemente esta proposta na ferramenta Octave e apresente os resultados obtidos. *Dica:* use os parâmetros 'minorAxisLength' e 'majorAxisLength' da função regionprops para obter o menor e maior diâmetro aproximado da região conectada.

Imagens geradas:



Código-fonte (em texto ou imagem):

```
octave:4> I= imread('circuit_holes.bmp');
PB = imfill(I,'holes');
I2=PB-I;
figure,
subplot(1,2,1), imshow(I, []), title('Original'),
subplot(1,2,2), imshow(I2, []), title('Preenchimento de buracos');

octave:10> CC = bwconncomp(I2);
octave:11> CC.NumObjects
ans = 4
octave:12> A = regionprops(CC);
```

```
octave:13> A(1)
```

```
ans =
```

```
scalar structure containing the fields:
```

```
Area = 1545
```

```
BoundingBox =
```

```
49.500 156.500 43.000 46.000
```

```
Centroid =
```

```
71.000 179.370
```

```
octave:14> A(2)
```

```
ans =
```

```
scalar structure containing the fields:
```

```
Area = 1529
```

```
BoundingBox =
```

```
53.500 28.500 43.000 47.000
```

```
Centroid =
```

```
75.111 52.670
```

```
octave:15> A(3)
```

```
ans =
```

```
scalar structure containing the fields:
```

```
Area = 1562
```

```
BoundingBox =
```

```
172.500 166.500 44.000 46.000
```

```
Centroid =
```

```
194.88 189.84
```

```
octave:16> A(4)
```

```
ans =
```

```
scalar structure containing the fields:
```

```
Area = 1543
```

```
BoundingBox =
```

```
178.500 35.500 43.000 46.000
```

```
Centroid =
```

```
199.743 58.835
```

```
octave:17> Minor = regionprops(CC, 'MinorAxisLength');
```

```
Major = regionprops(CC, 'MajorAxisLength');
```

```

octave:32> Minor(1)
Major(1)
Minor(2)
Major(2)
Minor(3)
Major(3)
Minor(4)
Major(4)
ans =

    scalar structure containing the fields:

    MinorAxisLength = 43.199
ans =

    scalar structure containing the fields:

    MajorAxisLength = 45.605
ans =

    scalar structure containing the fields:

    MinorAxisLength = 42.150
ans =

    scalar structure containing the fields:

    MajorAxisLength = 46.272
ans =

    scalar structure containing the fields:

    MinorAxisLength = 43.281
ans =

    scalar structure containing the fields:

    MajorAxisLength = 46.115
ans =

    scalar structure containing the fields:

    MinorAxisLength = 42.662
ans =

    scalar structure containing the fields:

    MajorAxisLength = 46.134

```

4. (3 pontos) Deseja-se segmentar uma imagem obtida da digitalização de uma página de texto contendo números. Uma primeira tentativa de limiarização utilizando o método de Otsu produziu um resultado de baixa qualidade. Utilizando a ferramenta Octave e a imagem 'text_page.tif', faça o que se pede:

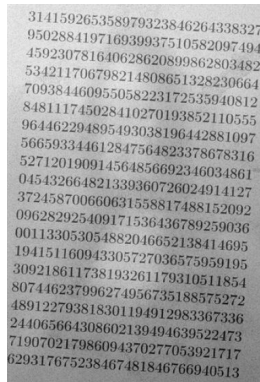


Imagem Original



Limiarização (Otsu)

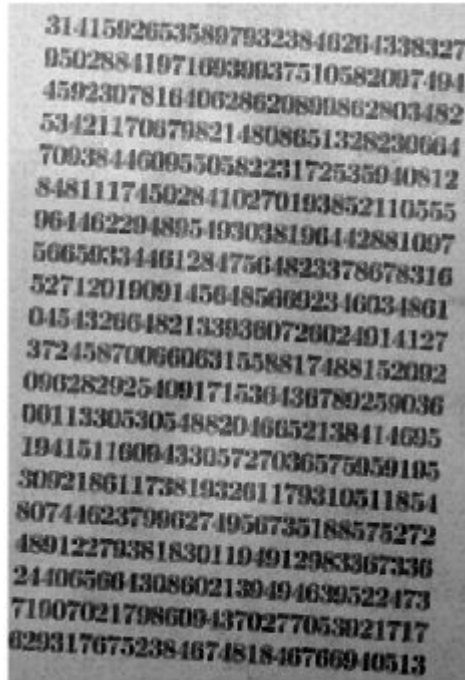
- a. Proponha um método usando processamento morfológico capaz de alcançar um melhor resultado do que o demonstrado acima

Resposta:

Podemos fazer uma simples erosão, que vai dar uma sobrevida nos números para o melhor entendimento.

- b. Implemente esta proposta na ferramenta Octave e apresente os resultados obtidos

Imagens geradas:



Código-fonte (em texto ou imagem):

```
octave:27> I = imread('text.tif');
octave:28> se = strel("square", 3);
octave:29> D = imerode(I, se);
octave:30> imshow(D);
```

"se" é o elemento estruturante para usar na erosão.

Boa sorte!
Prof. Igor.