Lista de Exercícios de Processamento Digital de Imagens

Pós-Graduação

Aluna: Lígia Iunes Venturott
Data: 30 de Junho de 2019

Exercícios Práticos

Atividade 1

Primeiro foi calculado a intensidade da imagem, realizando a média das cores para cada pixel.

A faixa de valores de intensidade foram divididos em 8 regiões de mesmo tamanho. O cálculo foi feito utilizando os valores mínimos e máximos de intensidade da imagem, ao invés de dividir toda a faixa de 256 valores.

Abaixo temos o resultado:

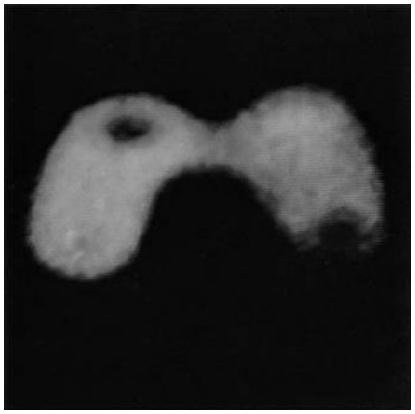


Fig. 1: Intensidade

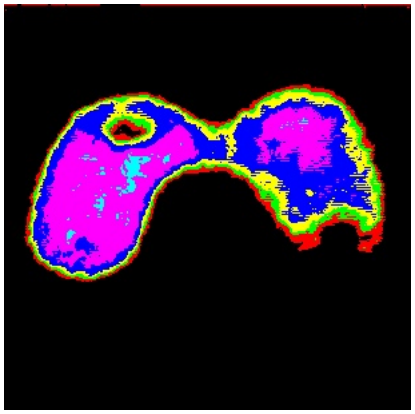


Fig. 2: Sliced

É possível perceber claramente as regiões de intensidades diferentes como no exemplo do livro. Apesar da imagem em escala de cinza parecer constante na Figura 1, percebemos que há uma variação de intensidade na imagem pela Figura 2.

Atividade 2

O primeiro passo a ser realizado foi a conversão da imagem de RGB para HSI. Para isso foram criadas 2 funções, HSI2BGR e BRG2HSI utilizando as equações do livro. A biblioteca OpenCV utiliza o padrão BRG, e não RGB, então a ordem das cores foi alterada para a conversão.

Após a conversão para HSI a faixa de ângulos pertencente à cor vermelha foi selecionada. A essa faixa foram somados aproximadamente 180°. Foi utilizado o resto da divisão por 360 (x%360) para respeitar o limite de 360°do círculo.

Foram criadas 2 variáveis, *ajuste* e *ajuste2*, para ajudar a regular tanto o ângulo do vermelho, quanto o ângulo de adição para transformar em azul.

Por fim a imagem foi convertida novamente para RGB. O resultado se encontra abaixo.

Fig. 3: Original

Fig. 4: Azul

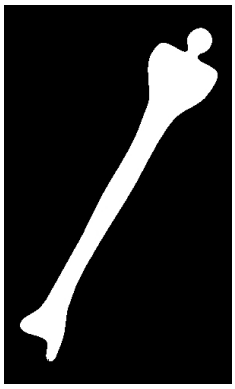
Atividade 3

Nessa atividade foi utilizado o processamento descrito no livro. Foram implementadas as funções *erosao()* e *dilatacao()*. A partir das duas foi implementada a função *abertura()*.

Primeiramente foi encontrado o K_max, q quantidade K máxima de vezes que se pode erodir o objeto antes que este suma.

Para cada k de 0 a K_max foi calculado $S_k = (A \ominus kB) - (A \ominus kB) \circ B$. Para obter $(A \ominus kB)$ realiza-se a operação de erosão do objeto k vezes.

Por fim é feita a união, ou OU lógico, dos S_ks calculados.



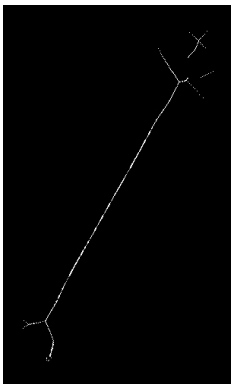


Fig. 5: Original **Fig. 6:** Original

Atividade 5

O primeiro passo desta atividade foi tentar isolar as linhas brancas na imagem. Foram testados 3 métodos.

1. Intensidade > 230: Foram selecionados apenas os pixels cuja intensidade na escala de cinza fossem maior do que 230.
2. Alguma cor > 250: Foram selecionados apenas os pixels que continham alguma cor maior do que 250.
3. Todas as cores > 250: Foram selecionados apenas os pixels em que as 3 cores fossem maior do que 250. Foi feito assim para que uma cor não compensasse a outra, como acontece na primeira opção, garantindo que os pixeis selecionados fossem o mais branco possível.

Os resultados dos 3 métodos estão expostos abaixo:

Fig. 7: Escala de cinza

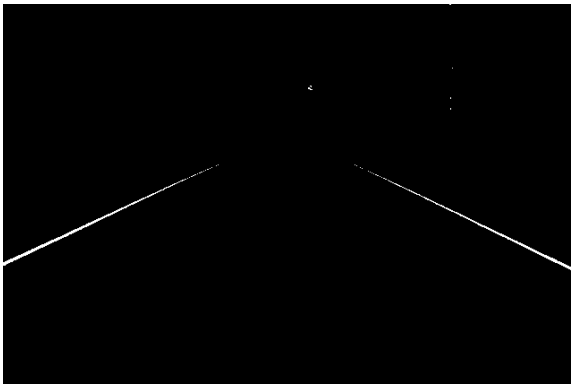


Fig. 8: Método 1

Fig. 9: Método 2

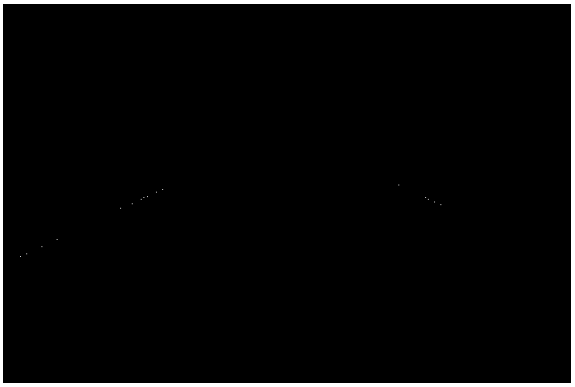


Fig. 10: Método 3

O primeiro e segundo métodos resultam em pontos que não pertencem às linhas da estrada, apesar de mostrarem as linhas de forma mais bem definida.

Já o terceiro método não mostra pontos brancos que não pertençam à estrada, porém resulta em muito poucos pontos. Porém, pela teoria isso já seria o suficiente para a Transformada de Hough.

Foi criada uma função chamada *hough*. Essa função recebe uma imagem e retorna o plano de parâmetros preenchido.

Chamamos de plano de variáveis o plano (X, Y) que contém os pixels da imagem. Chamamos de plano de parâmetros o plano (θ, ρ) que contém as retas formadas a partir dos pontos do plano de variáveis.

Na função *hough* é criada uma matriz de zeros para servir de plano de parâmetros que será preenchida. A partir de cada ponto (x, y) da imagem limiarizada é criada uma função (θ, ρ) e essa função é "desenhada" no plano de parâmetros para a faixa de valores $\theta = [-90, 90]$. Esse "desenho" é feito somando $+1$ à cada pixel do plano de parâmetros por onde a nova função calculada passa. Dessa maneira os pontos onde mais de uma função passou possuirão valores mais altos.

Esse plano de parâmetros preenchido é retornado pela função *hough*.

Depois o algorítimo checa quais os 2 pontos de (θ, ρ) que possuem os maiores valores. Com essas coordenadas (θ, ρ) são calculadas as 2 retas no plano (X, Y).

Essas são plotadas utilizando a biblioteca *matplotlib*.

Os resultados estão expostos abaixo.

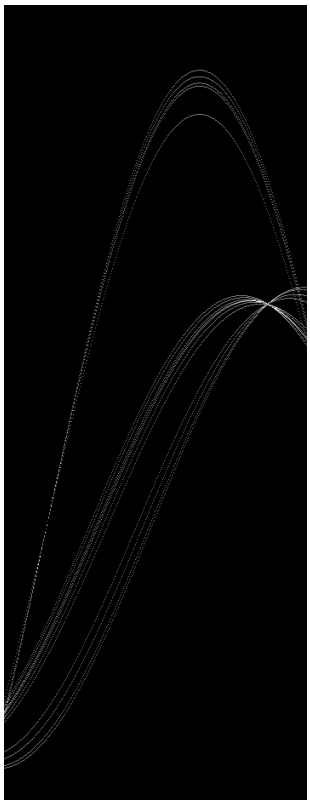


Fig. 11: Plano (θ, ρ) preenchido

	θ	ρ
Reta 1	65,5	262
Reta 2	-65,5	-1

Tabela. 1: Parâmetros encontrados

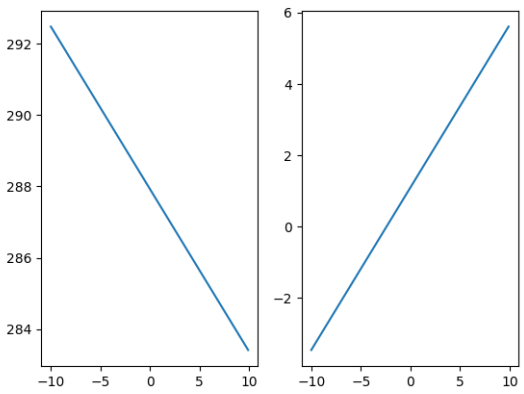


Fig. 12: Retas 2 e 1, respectivamente

Atividade 6

O primeiro passo é produzir as sementes que serão usadas para o crescimento. A imagem original passou por uma limiarização e apenas os pixels com valor 255 foram selecionados.

Depois esses objetos foram erodidos até o limite antes de desaparecerem, porém essa erosão levava a objetos com mais de 1 pixel. Então foi escolhido um pixel aleatório de cada objeto resultante da erosão.

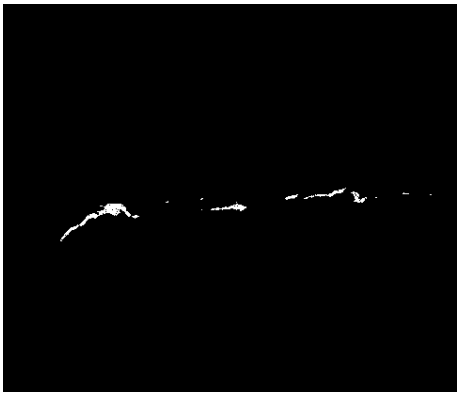


Fig. 13: Limiarizada

Fig. 14: Sementes

Foi criada uma função *grow2()* que realiza 1 etapa de crescimento da imagem. Ela recebe como parâmetros:

1. a imagem original
2. uma imagem com a região atual em branco
3. uma imagem com apenas os pixels a serem analisados em branco
4. o valor T para comparação de intensidade dos pixels

A função itera por todos os pixels de 3) e analisa sua vizinhança. Os pixels que respeitarem as condições descritas são adicionados à região 2).

A função então calcula quais os novos pixels adicionados e retorna tanto a nova região quanto a imagem apenas com os pixels recentemente adicionados.

O algorítimo chama a função *grow2()* em um loop. Os item 3) é a imagem formada pelos últimos pixels adicionados à região, para que a função *grow2()* não tenha que analisar toda a região, apenas as bordas.

O loop segue até que não haja mais incremento na região.

Foi feito o processo para 4 valores de T. Os resultados estão expostos a seguir, tanto em imagem quanto a quantidade de pixels em cada região.

Fig. 15: T = 1, 790 pixels

Fig. 16: T = 3, 836 pixels

Fig. 17: T = 5, 896 pixels

Fig. 18: T = 7, 991 pixels

Atividade 8

Primeiramente foram geradas as imagens para teste.

Fig. 19: Original

Fig. 20: Reduzida

Fig. 21: Rotação 90°

Fig. 22: Rotação 180°

	1	2	3	4	5	6	7
Original	2.87414	8.16792	11.92336	10.93827	17.69278	-15.05568	-22.36949
Reduzido	2.87414	8.16794	11.92326	10.93827	17.69188	-15.05569	-22.36943
Rot 90°	2.87414	8.16792	11.92336	10.93827	-17.71371	-15.05568	-22.36949
Rot 180°	2.87414	8.16792	11.92336	10.93827	-17.69278	-15.05568	-22.36949

Tabela. 2: Momentos Invariantes Calculados

Observa-se que os valores não mudam para a rotação ou redimensionamento da imagem.

Exercícios Teóricos

Questão 1

Não. Consideramos inicialmente que a erosão é uma analise de pixel a pixel se toda a sua vizinhança no formato de B está contida em A.

Considerando o caso mencionado em que a origem de B não está em B. Ao analisar o pixel (x, y), estamos na verdade analisando a vizinhança de outro pixel $(x + m, y + n)$. O resultado dessa análise substituirá o pixel (x, y). Ou seja, (x, y) pode ser um pixel fora de A enquanto $(x + m, y + n)$ é um pixel que está contido em A, e que toda sua vizinhança em formato B também está contido em A.

Dessa maneira o pixel (x, y) será "pintado", mesmo que não pertença a A.

Assim não podemos afirmar que no caso apresentado, a erosão de A por B estará contida em A.

Questão 2

Na Transformada de Hough para retas, sabemos que as retas podem ser definidas por apenas 2 parâmetros θ, ρ. Por isso criamos um plano de parâmetros bidimensional.

Já um círculo é definido por 3 parâmetros a, b, r, em que a e b são as coordenadas do centro da circunferência e r é o raio. Por isso nesse caso o plano de parâmetros será tridimensional.

Temos as equações:

$$(x - a)^2 + (y - b)^2 = r^2 \tag{1}$$

$$x = a + r.\cos\theta \tag{2}$$

$$y = b + r.\cos\theta \tag{3}$$

Sendo que, nas equações 2 e 3 ao calcularmos x e y para todos os θde 0° a 360° desenhamos um círculo. Invertendo estas 2 equações, podemos desenhar um círculo no plano de parâmetros para cada par (x, y) para uma faixa de valores de raios.

Como na transformada para retas, detectamos o ponto no plano que possui a maior intensidade, e esse ponto fornecerá os parâmetros do círculo desejado.

Questão 3

O algorítimo considera uma imagem em escala de cinza como um plano 3D, em que a intensidade de cinza pode ser representada pela altura. Sendo assim temos o que se parece com a topografia de uma região. As regiões com altos valores formam os cumes, e as com valor baixo, os vales.

Considere que água seja despejada nesses vales, ou jorre para fora da terra, em igual velocidade para todos os vales. O objetivo é impedir que a água de vales diferentes se

misture. Para isso, a medida que a água vai subindo devemos construir barreiras no topo dos cumes para impedir que os vales se unam.

Esse processo deve ser repetido até que o nível d'água chegue e ultrapasse o cume mais alto.

Do ponto de vista de imagem, essas barreiras construídas formam as delimitações das regiões da imagem, segmentando-a.

No algorítimo, são localizados os mínimos regionais para servir de ponto de despejo de água.

Questão 4

- Assinatura:

  A assinatura do objeto poderia ser usado para essa tarefa, pois a distância do centro de um circulo para suas bordas é constante. Mesmo que o circulo contenha falhas essa distância não varia muito.

  Já a distância do centro de um quadrado até suas bordas varia de $\frac{l}{2}$ até $\sqrt{2}\frac{l}{2}$, sendo l o comprimento do lado do quadrado.

  Apesar da forma do gráfico da assinatura variar com a rotação ou mudança de escala, ainda pode se checar a variação nesse gráfico para identificar a forma, como a quantidade de picos ou a relação de valores $pico \backslash vale$.

- Descritor Simples - Compacidade

  A compacidade é definida por:

  $$compacidade = \frac{perimetro^2}{area}$$

  Independente de escala, rotação ou translação, a compacidade de um circulo deve ser próxima de 4π, que é aproximadamente *12,56*. Já a de um quadrado deve ser de *16*. Dessa maneira é possível diferenciar os dois.