

Primeira Lista de Exercícios
Capítulos 8, 9, 10, 11

Responda os quesitos completamente em detalhes.

Estudante: Diógenes Wallis de França Silva

1) Compressão:

- a. Descreva a codificação JPEG, demonstre um diagrama desta codificação com todas suas partes e explique cada um destas componentes. Porque JPEG utiliza DCT ao invés da DFT? (Seja completo da sua descrição e explicação).

O padrão de compressão JPEG é bastante popular e abrangente. O JPEG define três tipos de codificação: a codificação baseline com perdas, baseado em DCT; um sistema estendido de codificação para aplicações de maior precisão; e por último um sistema independente sem perdas para realizar compressões reversíveis.

Se tratando do sistema baseline, a precisão (tanto dos dados de entrada, quanto de saída) limita-se a 8 bits, enquanto que os valores da DCT são de 11 bits. A compressão é realizada em 3 passos: cálculo da DCT, processo de quantização e atribuição do código. A imagem é subdividida em blocos de tamanho 8x8, estes blocos são processados de cima para baixo e da esquerda para direita. Cada bloco tem os valores dos seus 64 pixels deslocados da quantidade 2^{k-1} , sendo 2^k o maior número em relação aos níveis de intensidade. Em seguida a DCT do bloco é calculada, quantizada e reordenada para gerar uma sequência 1-D de coeficientes.

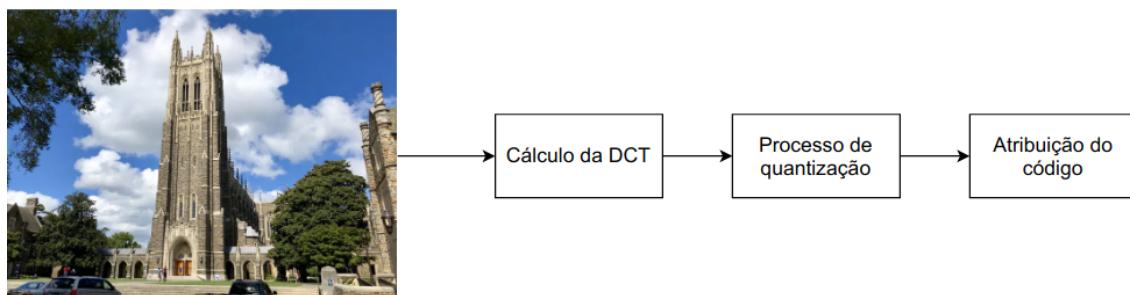


Diagrama de codificação JPEG

O processo é realizado passo a passo através do exemplo a seguir. Seja a seguinte subimagem:

52	55	61	66	70	61	64	73
63	59	66	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	63	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

Note que esta sub-imagem pertence a uma imagem de 8 bits, sendo assim o deslocamento dos níveis de cinza e feito subtraindo cada valor de pixel por 2^{8-1} . O resultado é uma subimagem da seguinte forma:

-76	-73	-67	-62	-58	-67	-64	-55
-65	-69	-62	-38	-19	-43	-59	-56
-66	-69	-60	-15	16	-24	-62	-55
-65	-70	-57	-6	26	-22	-58	-59
-61	-67	-60	-24	-2	-40	-60	-58
-49	-63	-68	-58	-51	-65	-70	-53
-43	-57	-64	-69	-73	-67	-63	-45
-41	-49	-59	-60	-63	-52	-50	-34

Esta subimagem vai ser transformada através da DCT, tornando-se:

-415	-29	-62	25	55	-20	-1	3
7	-21	-62	9	11	-7	-6	6
-46	8	77	-25	-30	10	7	-5
-50	13	35	-15	-9	6	0	3
11	-8	-13	-2	-1	1	-4	1
-10	1	3	-3	-1	0	2	-1
-4	-1	2	-1	2	-3	1	-2
-1	-1	-1	-2	-1	-1	0	-1

O resultado acima é então quantizado pela matriz abaixo:

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Usando a matriz acima, tem-se:

$$\begin{array}{ccccccc}
 -26 & -3 & -6 & 2 & 2 & 0 & 0 \\
 1 & -2 & -4 & 0 & 0 & 0 & 0 \\
 -3 & 1 & 5 & -1 & -1 & 0 & 0 \\
 -4 & 1 & 2 & -1 & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0
 \end{array}$$

Note que os valores obtidos da DCT foram divididos pelos pesos da matriz de normalização. Também vale mencionar que este processo gera um grande número de coeficientes nulos. Usando a seguinte matriz, faz-se a ordenação dos coeficientes em 1-D.

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

O resultado é o seguinte:

[-26 -31 -3 -2 -62 -41 -41 150 200 -1200000 -1 -1 EOB]

Note que o EOB é o símbolo fim de bloco e indica que o restante dos elementos são nulos.

Se tratando da diferença entre a DCT e a DFT, tem-se que as duas transformadas apresentam funções base diferentes, enquanto a DFT tem uma base exponencial, a DCT tem sua base composta por cossenos.

A DCT é geralmente utilizada na compressão de dados, como na JPEG que é citada e exemplificada neste texto. Uma vez que a DCT possui um alto grau de compactação espectral, isto a torna adequada para tarefas de compressão. Comparada com a DFT, pode-se dizer que a DCT consegue concentrar mais energia em um número menor de coeficientes, sendo assim, a DCT precisa de menos memória para armazenar a mesma quantidade de energia da DFT.

**b. Quantos códigos únicos de Huffman existem para um sistema de 5 símbolos?
Construa-os.**

O número de códigos únicos é 2^{5-2} , isto é, tem-se 8 códigos únicos. Isto acontece porque as bifurcações usadas na criação dos códigos possibilitam 2 possibilidades, já que é feita a escolha arbitrária entre 0 e 1. No caso de 5 símbolos existem 3 bifurcações que podem ser determinadas de duas formas diferentes, assim 2^3 possibilidades. Abaixo será exemplificado para um único caso e depois é exibido um diagrama com as 8 possibilidades.

No processo de construção dos códigos, faz-se necessário ordenar as probabilidades de cada símbolo, conforme a tabela abaixo.

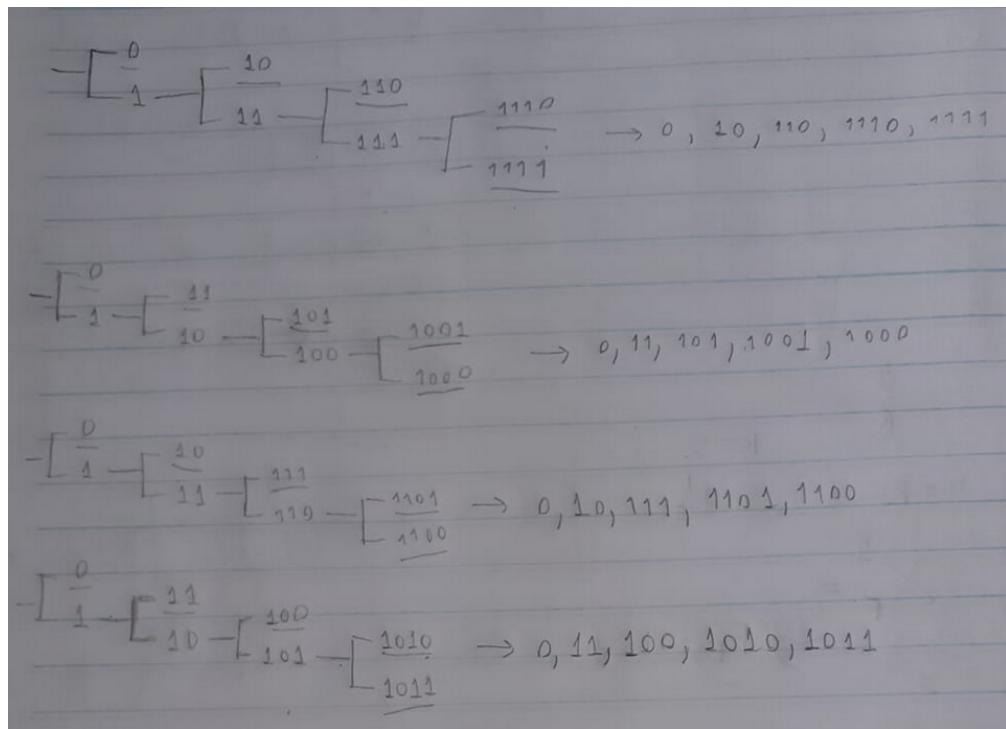
Fonte original		Redução de fonte		
Símbolo	Probabilidade	1	2	3
a1	0,4	0,4	0,4	0,6
a3	0,35	0,35	0,35	0,4
a2	0,2	0,2	0,25	
a4	0,04	0,05		
a5	0,01			

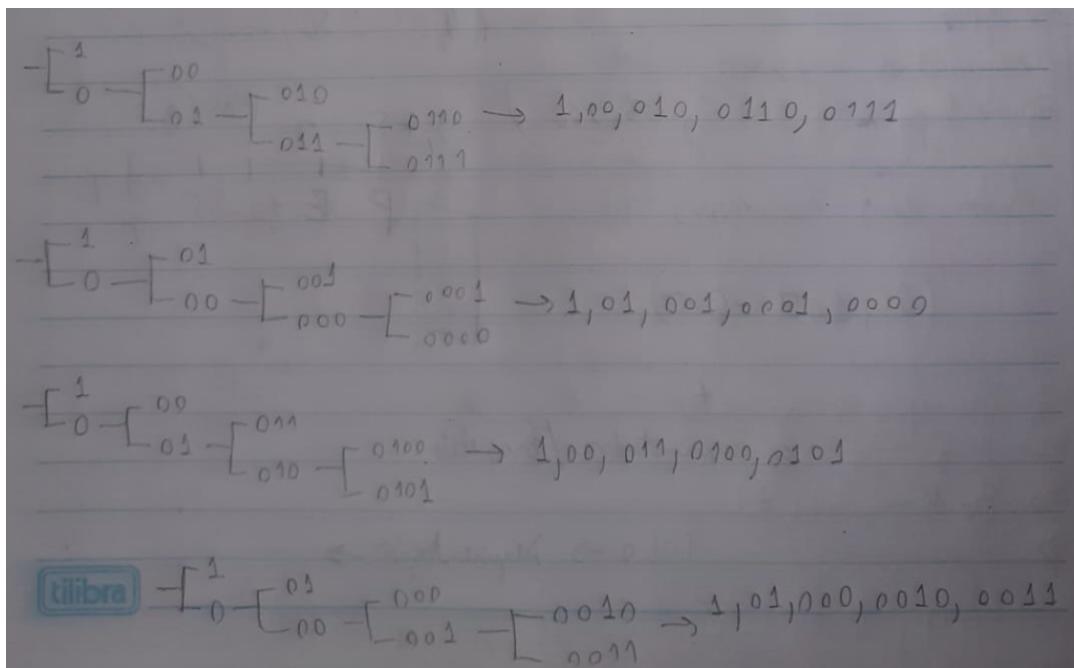
Uma vez as probabilidades ordenadas, faz-se a redução de fonte em que a cada etapa as duas probabilidades de menor valor são combinadas para formar um símbolo composto, cuja probabilidade é a soma das duas probabilidades inferiores. O processo se repete até atingir dois símbolos. Feito isto, agora é codificada cada fonte reduzida. Cada probabilidade composta é reduzida para em duas probabilidades e assim é feita a atribuição dos 0s e 1s. Na etapa 3 da redução de fonte existem dois símbolos que recebem 0 e 1, na segunda etapa 0,6 origina duas

probabilidades que recebem 0 e em seguida 0 ou 1. O processo se repete até que os 5 símbolos estejam definidos. Assim, originam-se então os códigos para cada um dos símbolos.

Fonte original			Reducao de fonte		
Símbolo	Probabilidade	Código	1	2	3
a1	0,4	0,4 - 1	0,4 - 1	0,4 - 1	0,6 - 0
a3	0,35	0,35 - 00	0,35 - 00	0,35 - 00	0,4 - 1
a2	0,2	0,2 - 010	0,2 - 010	0,25 - 01	
a4	0,04	0,04 - 0110	0,05 - 011		
a5	0,01	0,01 - 0111			

Repetindo o processo, mas considerando as diferentes escolhas de 0s e 1s:





Dessa forma, ao todo, geram-se 8 códigos diferentes.

2) Morfologia Matemática:

a. Explique o conceito de Morfologia Matemática. Defina suas operações fundamentais (Dilatação, Erosão, Abertura e Fechamento) tanto para imagem binária como em tons de cinza. Dê exemplos de aplicações.

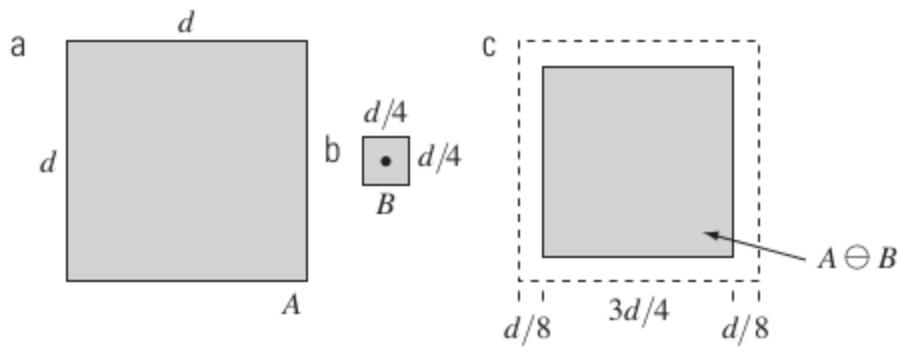
A morfologia num contexto geral significa um ramo da Biologia que tem a forma e a estrutura dos animais como alvo de estudo. Já a morfologia matemática é uma ferramenta utilizada para a extração de componentes das imagens que servem para representar e descrever a forma de regiões, fronteiras, esqueletos. Também vale mencionar a filtragem morfológica, afinamento, poda, uma vez que são técnicas para pré ou pós-processamento. A seguir são abordados os exemplos de morfologia: dilatação, erosão, abertura e fechamento.

Primeiro são abordadas as operações nas imagens binárias:

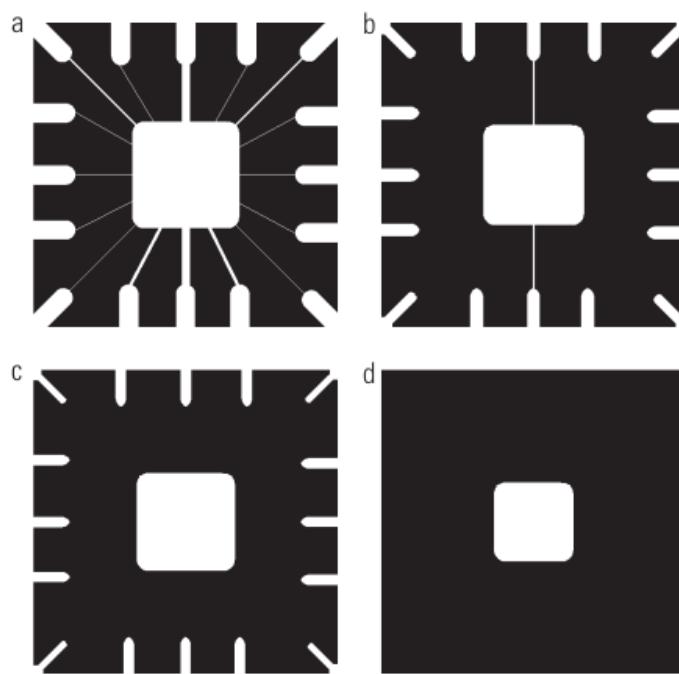
Se tratando do processo de **erosão**, tem-se a seguinte expressão:

$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$

Basicamente, a expressão acima diz que a erosão de A em B nada mais é do que o conjunto dos pontos z, de modo que B transladado por z, deve estar contido em A.



A figura acima contém o processo de erosão, em que B é o elemento estruturante. Note que a linha sólida é o limite de até onde se pode mover a origem de B em que ele ainda continua completamente contido em A. Os pontos da origem de B dentro do limite especificado constituem a erosão sofrida em A por B. Abaixo tem-se um exemplo do uso da erosão para remoção de algumas componentes da imagem.

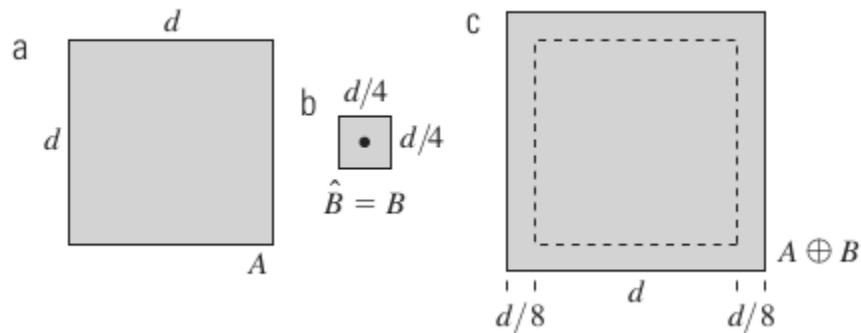


Seja a imagem binária acima, usa-se a erosão para remoção das componentes nas extremidades da imagem, bem como as suas conexões com o elemento do centro. As imagens de b a d são todas tipos de erosão, cujos elementos estruturantes são quadrados de tamanho 11x11, 15x15 e 45x45. No que em d todas as componentes são removidas, ao passo que o elemento no meio também reduziu um pouco de tamanho.

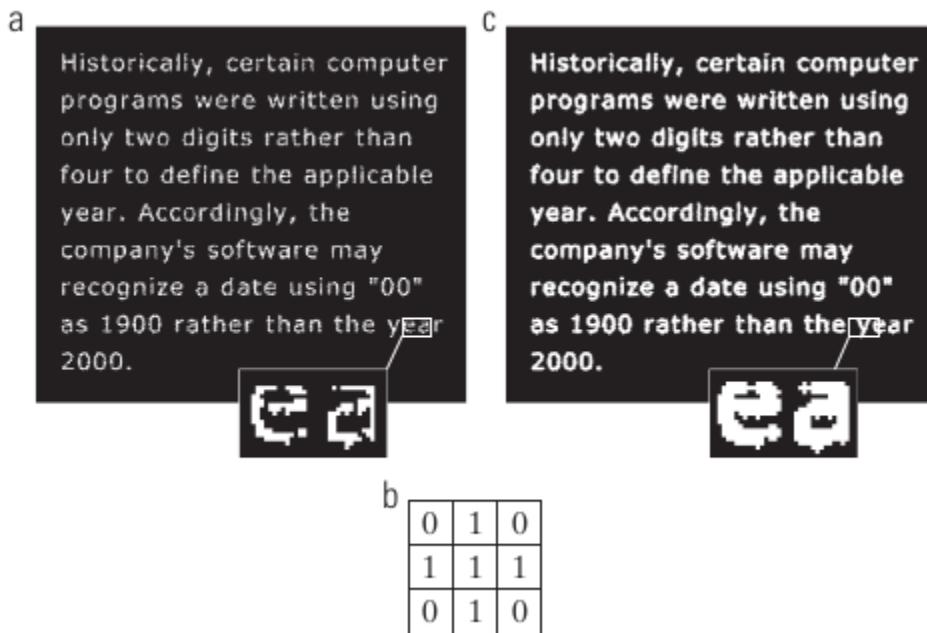
Já se tratando do processo de **dilatação**, tem-se:

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \emptyset\}$$

Sendo assim a dilatação de A em B nada mais é do que o conjunto de todos os deslocamentos z, de modo que A e \hat{B}_z se sobreponham por ao menos um elemento. A imagem abaixo ilustra bem esse conceito:



Diferente da erosão, que diminui o tamanho de A, a dilatação aumenta o elemento A. A linha contínua mostra o limite até onde se pode deslocar a origem de B sem resultar em uma interseção vazia entre B e A. Sendo assim, os pontos no limite especificado acima resultam na dilatação de A por B.



Acima tem-se um exemplo da aplicação do processo de dilatação. Note que a matriz é o elemento estruturante e que a sofre a dilatação. O resultado é a junção dos segmentos quebrados, do ponto de vista visual está bem mais agradável para a leitura.

No processo de **abertura e fechamento**:

A abertura é definida como:

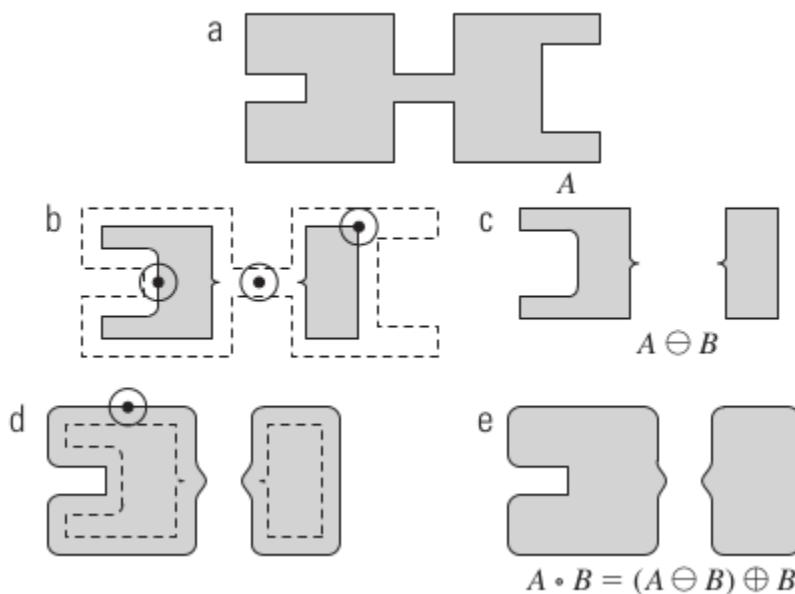
$$A \circ B = (A \ominus B) \oplus B$$

Isto é, a abertura é o processo de erosão de A por B, seguido por uma dilatação por B.

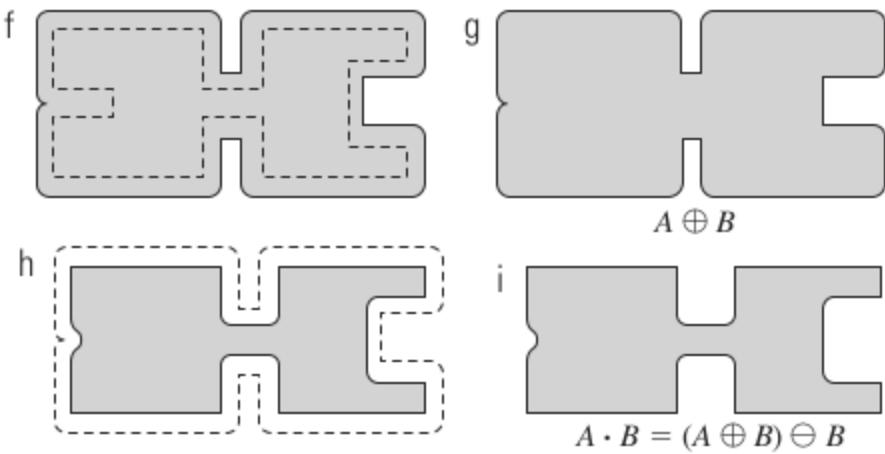
Já o fechamento:

$$A \bullet B = (A \oplus B) \ominus B$$

Trata-se da dilatação de A por B, seguida pela erosão de B. Segue imagem ilustrando os processos citados anteriormente:



Note que o elemento estruturante e o círculo destacado em várias partes diferentes. A figura E representa o processo de abertura aplicado em A. C e D são etapas anteriores ao processo resultante em E.



A figura acima trata do processo de fechamento. O resultado está em I, e novamente as etapas anteriores estão presentes, conforme G e H.

Como exemplo, tem-se a imagem abaixo:



A imagem A representa uma impressão digital em uma imagem binária corrompida por ruído. O objetivo é eliminar o ruído através de um filtro morfológico de abertura, seguido por um de

fechamento. B é o elemento estruturante e os processos que seguem realizam as filtragens especificadas, em que o resultado final é uma imagem sem ruído com o mínimo de distorção.

No contexto das imagens em **nível de cinza**, seguem as análises:

A erosão de f por um elemento estruturante b em qualquer posição (x,y) é dada por:

$$[f \ominus b](x, y) = \min_{(s,t) \in b} \{f(x+s, y+t)\}$$

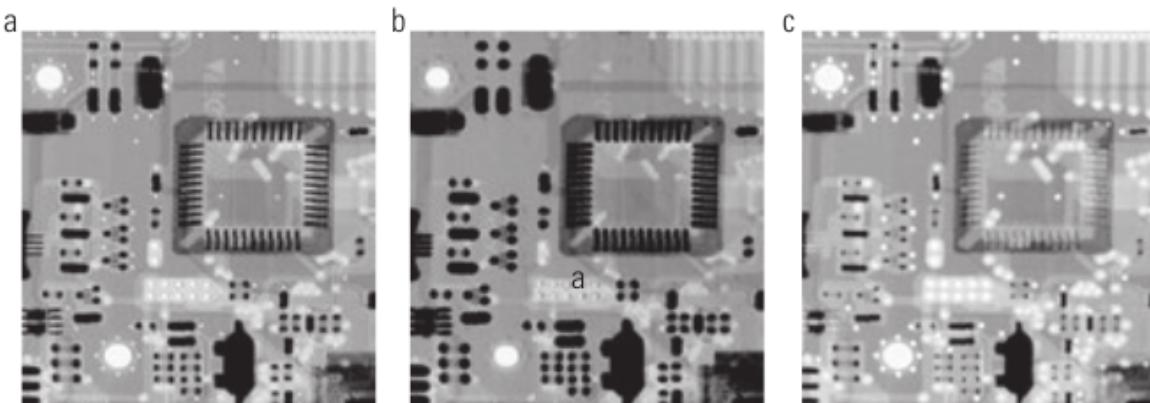
A expressão acima significa que calcular a erosão de f por b, faz-se necessário colocar a origem de b em todos os pixels da imagem. A erosão então é definida através da escolha do valor mínimo de todos os valores de f contidos em b.

Já a dilatação é determinada por:

$$[f \oplus b](x, y) = \max_{(s,t) \in b} \{f(x-s, y-t)\}$$

O processo é o mesmo do anterior, exceto que usa-se a operação de máximo, ao invés de mínimo, e também considera que o elemento estruturante sofre reflexão em relação a sua origem.

A imagem abaixo ilustra o processo:



A imagem A é a original e B e C são o resultado de se aplicar os processos de erosão e dilatação respectivamente. Em B os detalhes mais escuros aumentam de espessura e em C o resultado é o oposto, uma vez que os detalhes escuros foram reduzidos e os mais claros aumentaram.

O processo de abertura é representado por:

$$f \circ b = (f \ominus b) \oplus b$$

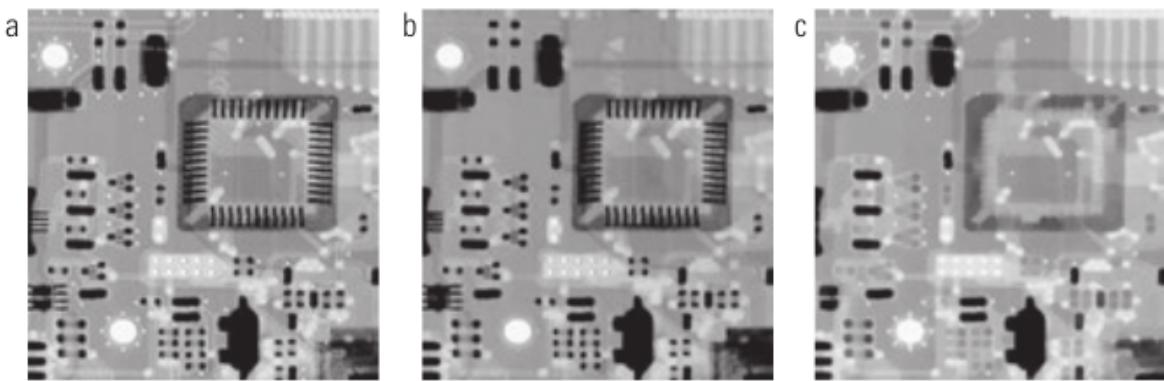
Segue a mesma ideia de antes, erosão de f pelo elemento b, seguido por uma dilatação de b.

Já o fechamento:

$$f \bullet b = (f \oplus b) \ominus b$$

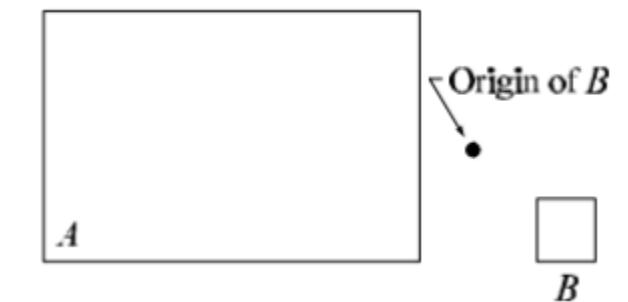
Também segue a mesma ideia apresentada anteriormente, f sofre dilatação por b e o resultado é erodido por b.

Como exemplo, tem-se a imagem abaixo:

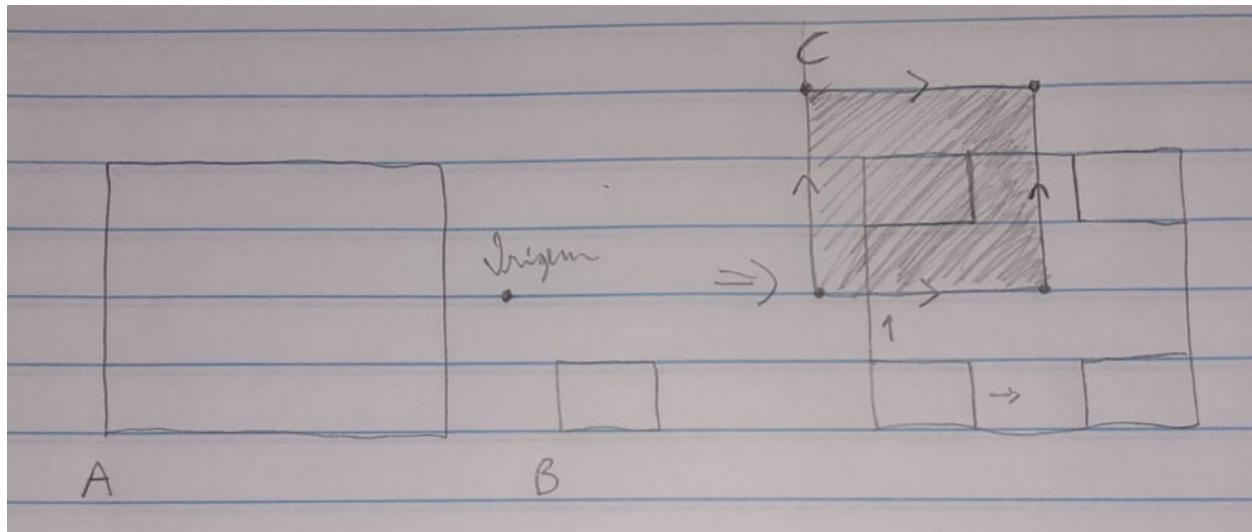


Novamente a imagem A é a imagem original, sendo B o resultado do processo de abertura e C o resultado do processo de fechamento. Em B o efeito nos objetos escuros é mínimo e a intensidade dos objetos claros diminuiu. Já em C os objetos claros e o fundo não tiveram tanta diferença, mas em compensação os objetos escuros foram atenuados. Lembrando que a intensidade dos efeitos depende do tamanho do elemento estruturante.

b. A erosão do conjunto A por um elemento estruturante B é um subconjunto de A se a origem de B está contido em B. Demonstre um exemplo na qual a erosão ($A \ominus B$) fica fora ou parcialmente fora de A. Ou seja, resolva:



Sendo a origem de B conforme a imagem acima, tem-se o seguinte resultado:



A e B são as mesmas do enunciado e C é a resposta após o processo de erosão. Pela definição os pontos considerados em C ocorrem somente quando B está contido em A, sendo assim, obtém-se quatro pontos posicionando B nos quatro cantos de A e em seguida cada localização da origem é interligada para formar C. Apenas destacando que C é o quadrado hachurado.

3) Segmentação:

a. Qual o valor médio do Laplaciano da Gaussiana? Demonstre.

Seja o Laplaciano da Gaussiana: $\nabla^2 G(x, y) =$

$$\nabla^2 G(x, y) = \left[\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Para calcular o seu valor médio:

$$\text{Médio} (\nabla^2 G(x, y)) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right] e^{-\frac{x^2+y^2}{2\sigma^2}} dx dy$$

$$= \frac{1}{\sigma^4} \int_{-\infty}^{\infty} x^2 e^{-\frac{x^2}{2\sigma^2}} dx + \int_{-\infty}^{\infty} y^2 e^{-\frac{y^2}{2\sigma^2}} dy - 2 \int_{-\infty}^{\infty} x^2 y^2 e^{-\frac{x^2+y^2}{2\sigma^2}} dx dy$$

$$\left(\text{Densidade: } \sigma_x^2 = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} z^2 e^{-\frac{z^2}{2\sigma^2}} dz \quad \text{e} \quad \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} z^2 e^{-\frac{z^2}{2\sigma^2}} dz = 1. \right)$$

$$= \frac{1}{\sigma^4} (\sqrt{2\pi}\sigma \cdot \sigma^2) (\sqrt{2\pi}\sigma \cdot \sigma^2) - \frac{2(2\pi\sigma^2)}{\sigma^2}$$

$$= 4\pi - 4\pi = 0$$

b. Explique o algoritmo de binarização global de Otsu.

O objetivo do processo de limiarização é reduzir o erro no processo de atribuição dos pixels para dois ou mais grupos. Para atingir esse objetivo, faz-se necessário o uso de dois parâmetros: a função densidade de probabilidade relacionada aos níveis de cinza de cada classe e a probabilidade de ocorrência da classe em uma dada aplicação. Uma outra abordagem para resolver este problema é o método de Otsu.

O método de Otsu é considerado ótimo no sentido de maximizar a variância entre as classes. A ideia é que as classes bem estabelecidas sejam distintas quando comparadas com as intensidades dos seus pixels. Também vale mencionar que o método de Otsu se baseia em cálculos realizados no próprio histograma da imagem, que pode ser facilmente obtido.

O algoritmo em si pode ser resumido nos seguintes passos:

1. Calcular o histograma normalizado da imagem e designar as componentes de frequência do histograma, que podem ser representadas como: p_i , $i = 0, 1, 2, \dots, L-1$. Em que L é a quantidade de níveis distintos na imagem.

2. Calcular as somas acumuladas para $k = 0, 1, \dots, L - 1$, através da expressão:

$$P_1(k) = \sum_{i=0}^k p_i$$

3. Calcular as médias acumuladas $m(k)$ ($k = 0, 1, \dots, L - 1$), conforme a expressão abaixo:

$$m(k) = \sum_{i=0}^k ip_i$$

4. Calcular a intensidade média global:

$$m_G = \sum_{i=0}^{L-1} ip_i$$

5. Calcular as variâncias entre classes ($k = 0, 1, \dots, L - 1$):

$$\sigma_B^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]}$$

6. Calcular o valor de k^* , como limiar de Otsu, em que:

$$\sigma_B^2(k*) = \max_{0 \leq k \leq L-1} \sigma_B^2(k)$$

Em caso de empate, k^* é obtido pela média dos valores de k que correspondem aos respectivos máximos.

7. Por último, obter a medida de separabilidade:

$$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_G^2}$$

Em que o k deve ser o k^* .

c. Suponha que você possui vários algoritmos de segmentação de imagens. Seu desejo é avaliar qual desses algoritmos é o melhor. Descreva possíveis estratégias para realizar essa tarefa. (Não leve em consideração nem o tempo de processamento nem a memória requerida).

Uma das formas é analisar o processo de segmentação pelas métricas mais apropriadas. Por exemplo, no caso de uma imagem com um dado objeto, no processo de segmentação deseja-se realizar uma classificação em relação a cada pixel. Sendo assim, a acurácia pode não ser uma forma interessante de se avaliar o processo, visto que se o objeto em questão representa uma pequena parcela da imagem, há desbalanceamento nos dados e

consequentemente uma forte tendência de classificar priorizando a classe dominante. Uma forma de lidar com isso é utilizando métricas como o índice de Jaccard ou o F1 score. Em ambos os casos é possível verificar o quanto o modelo consegue discriminar bem entre as classes.

Num panorama geral, acerca do melhor algoritmo, tem-se que não há um método definitivo para escolher, uma vez que os tipos de imagens influenciam no processo. A combinação dos algoritmos pode ajudar no processo.

Por último, pode-se dizer que o melhor algoritmo seria aquele que segmenta melhor, e uma boa segmentação é aquela que o segmento pode ser facilmente composto por suas próprias peças, mas difícil de ser composto com outras partes da imagem [1].

Outra forma também é o uso de técnicas quantitativas, conforme em [2].

[1] Bagon, S., Boiman, O. and Irani, M., 2008, October. What is a good image segment? a unified approach to segment extraction. In *European Conference on Computer Vision* (pp. 30-44). Springer, Berlin, Heidelberg.

[2] Dlugosz, F. L., Rosot, N. C., Rosot, M. A. D., & de Oliveira, Y. M. M. (2009). Índice para a avaliação de segmentação de imagens. *Floresta*, 39(1).

4) Derive o Laplaciano de uma gaussiana (LoG) a partir da definição da gaussiana:

$$G(r) = e^{-r^2/2\sigma^2}$$

onde $r^2=x^2+y^2$:

Seja $G(r) = e^{-\frac{r^2}{26^2}}$, para $r^2 = x^2 + y^2$:

$$G(x, y) = e^{-\frac{(x^2+y^2)}{26^2}}$$

Calcular os gradientes:

$$\nabla^2 G(x, y) = \frac{\partial^2 G(x, y)}{\partial x^2} + \frac{\partial^2 G(x, y)}{\partial y^2}$$

$$= \frac{\partial}{\partial x} \left[-\frac{x}{6^2} e^{-\frac{(x^2+y^2)}{26^2}} \right] + \frac{\partial}{\partial y} \left[-\frac{y}{6^2} e^{-\frac{(x^2+y^2)}{26^2}} \right]$$

$$= \left[\frac{x^2}{6^4} - \frac{1}{6^2} \right] e^{-\frac{(x^2+y^2)}{26^2}} + \left[\frac{y^2}{6^4} - \frac{1}{6^2} \right] e^{-\frac{(x^2+y^2)}{26^2}}$$

$$\nabla^2 G(x, y) = \left[\frac{x^2 + y^2 - 26^2}{6^4} \right] e^{-\frac{(x^2+y^2)}{26^2}}$$

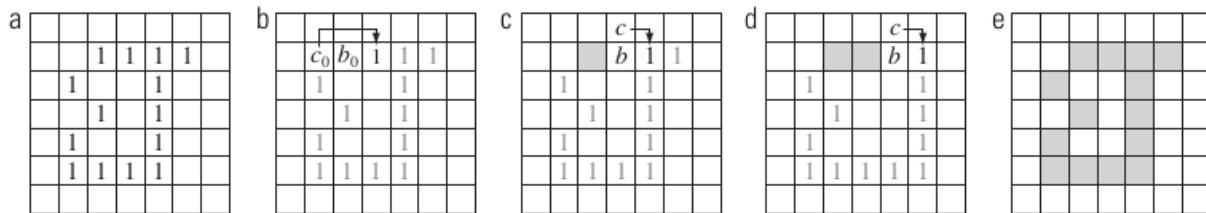
5) Representação e Descrição:

- a. Descreva três métodos de representação de imagens. Qual a sua importância em PI? Dê 3 exemplos de aplicação salientando o uso da representação e descrição de imagem.

A representação é uma das tarefas necessárias para tornar os dados úteis para o computador. Seguida da descrição, estas atividades são cruciais para a representação e descrição de agregados de pixels.

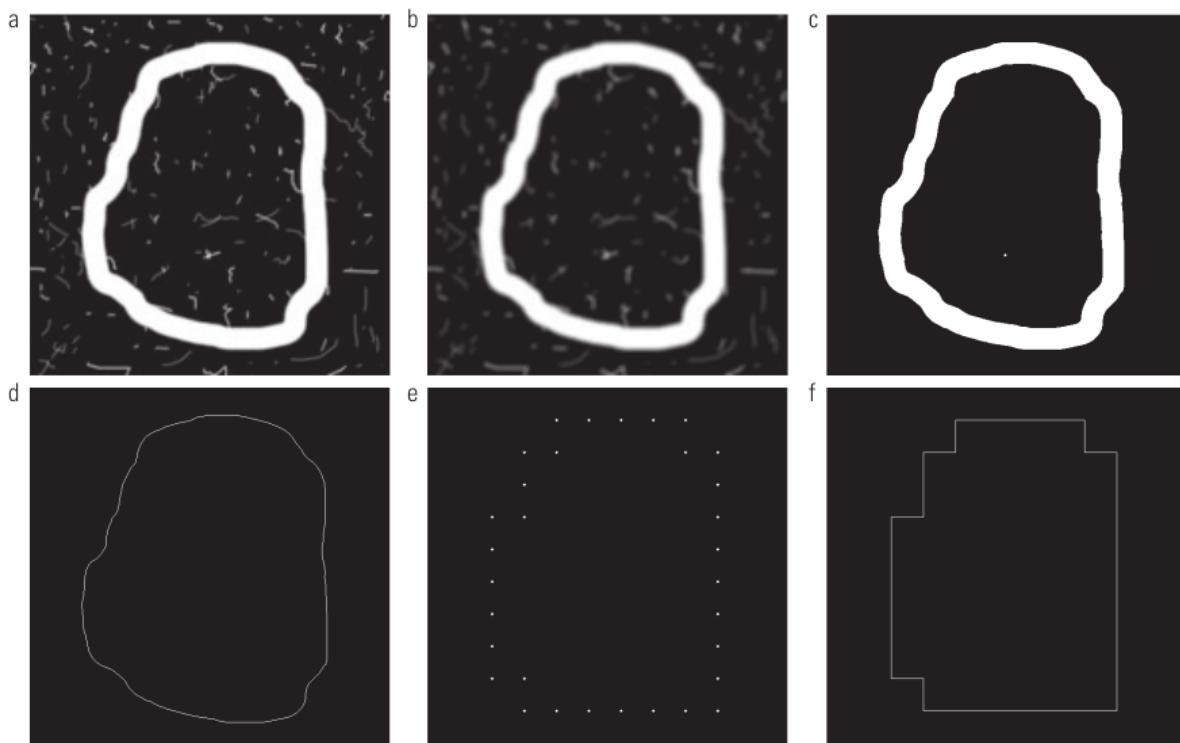
Agora são tratados os métodos, começando pelo seguidor de fronteira. Trata-se de um algoritmo cuja saída é uma sequência ordenada de pontos. Seja uma região binária, um algoritmo para seguir a fronteira da região consiste em 5 etapas. Considere que um ponto de partida vale 1 e está no canto superior esquerdo. Denote o ponto vizinho, partindo deste ponto

examine os vizinhos-8 do ponto de partida. Essa análise é necessária para determinar a localização do próximo ponto de partida. As etapas se repetem e no final é possível obter a devida fronteira.



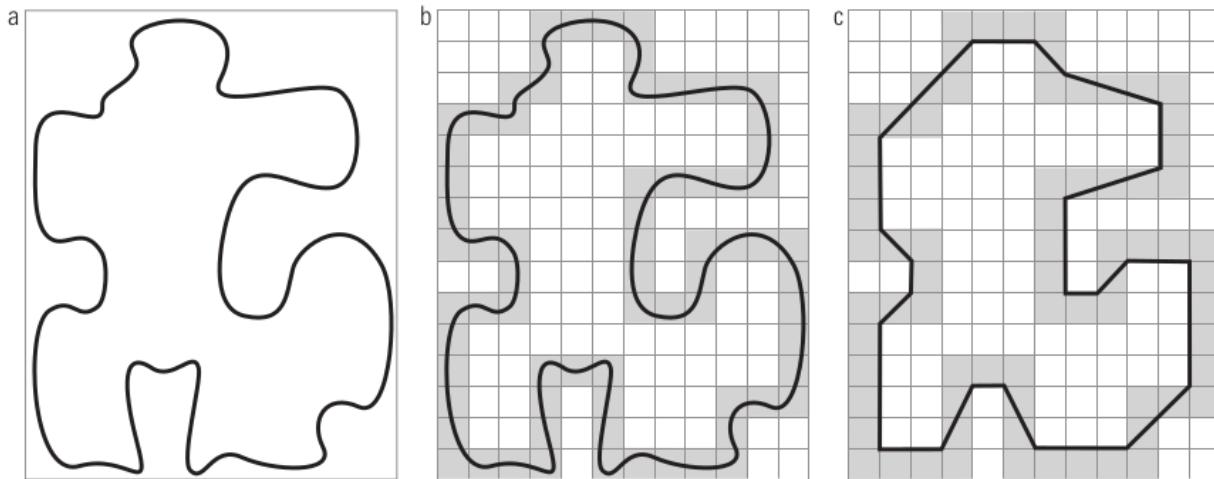
A imagem acima ilustra um exemplo em que seguindo as etapas do processo mencionado, obtém-se a devida fronteira.

Uma outra forma de representar uma fronteira é pelos códigos da cadeia. Basicamente é adotado um sistema de direções, que pode ser 4 ou 8, a depender do contexto, além de verificar a sequência de direções tomadas. À medida que são percorridas a borda da imagem, vai se atribuindo os respectivos números indicando as direções e assim gera-se um código de fronteira. Abaixo tem-se um exemplo:



Após uma série de processos gera-se a imagem em D que é subamostrada em E e em F estão os pontos conectados. Baseando-se em E, calcula-se o respectivo código da cadeia.

E por último tem-se as aproximações poligonais. Uma fronteira pode ser aproximada através do uso de um polígono. Uma das técnicas utilizadas é a fronteira por polígono de perímetro mínimo (MPP). O MPP pode ser gerado através de uma fronteira que cerca um conjunto de células concatenadas conforme a imagem B abaixo. Considerando a fronteira como um elástico, o elástico pode encolher e encontrar seus limites nas regiões delimitadas, conforme na imagem C. Note que os vértices coincidem com os cantos das paredes.

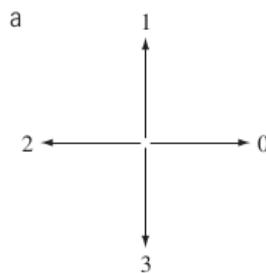


b. Qual a ordem do número de forma da imagem abaixo?

A ordem do número de forma é definida como o número de dígitos necessários para sua representação. No caso abaixo, tem-se 16 dígitos no número de forma, conforme pode ser contabilizado na figura, sendo assim, a ordem de 16.

c. Obtenha o número de forma da imagem.

Considerando um código com quatro direções:



Sendo a imagem abaixo, tem-se:

