Answers Exam Image Processing
April 7 2022, 12:15-14:15

Problem 1: Point wise operations [20 points]

Consider the following (small) grey scale image:

8	6	3	1
7	7	2	2
8	5	1	6
1	1	5	3

(a) (5 points) Which global threshold value would be returned by *Otsu's* algorithm if we apply it to the given image? Explain your answer. [Hint: there is no need to simulate the algorithm to answer this question.]

> The histogram h of the image is as follows: $h[1] = 4$, $h[2] = 2$, $h[3] = 2$, $h[4] = 0$, $h[5] = 2$, $h[6] = 2$, $h[7] = 2$, $h[8] = 2$. Clearly, this is a *bimodal* histogram, and the two classes are separated by the threshold $T = 4$ (which Otsu will return).
> [Note: we also accepted, with the correct argumentation, the answers $T = 3$ and $T = 5$. It depends on the implementation of the threshold test ($<$, $\leq$, $>$, $\geq$).]

(b) (5 points) Determine the output image if we use *linear contrast stretching* to change the dynamic range of this image to $[2..16]$. Explain how you arrived at your answer.

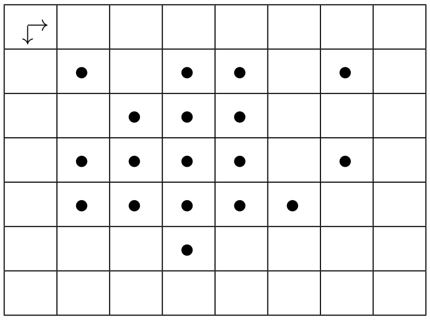
> The formula for linearly contrast stretching a range $[a..b]$ to $[c..d]$ is given by:
>
> $$stretch(x) = \frac{d-c}{b-a} \cdot (x-a) + c$$
>
> In this particular case we have $a = 1$, $b = 8$, $c = 2$, and $d = 16$ which yields:
>
> $$stretch(x) = \frac{16-2}{8-1} \cdot (x-1) + 2 = 2 \cdot (x-1) + 2 = 2 \cdot x$$
>
> Hence, all pixel values are doubled.
>
> The image after linear contrast stretching is
>
> | | | | |
> |---|---|---|---|
> | 16 | 12 | 6 | 2 |
> | 14 | 14 | 4 | 4 |
> | 16 | 10 | 2 | 12 |
> | 2 | 2 | 10 | 6 |

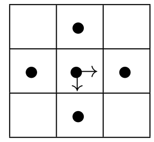
(c) (10 points) Determine the output image if we use *histogram equalization* to change the dynamic range of this image to $[0..8]$. Explain how you arrived at your answer.

> We substitute $L = 9$ and $N = 16$ in $\mathcal{O}(x) = \frac{(L-1)}{N} \sum_{m=0}^{x} h(m) = \frac{\sum_{m=0}^{x} h(m)}{2}$.
>
> | n | $h(n)$ | $H(n) = \sum_{k=0}^{n} h(k)$ | $\mathcal{O}(n)$ |
> |---|---|---|---|
> | 0 | 0 | 0 | 0 |
> | 1 | 4 | 4 | 2 |
> | 2 | 2 | 6 | 3 |
> | 3 | 2 | 8 | 4 |
> | 4 | 0 | 8 | 4 |
> | 5 | 2 | 10 | 5 |
> | 6 | 2 | 12 | 6 |
> | 7 | 2 | 14 | 7 |
> | 8 | 2 | 16 | 8 |
>
> The image after histogram equalization is
>
> | | | | |
> |---|---|---|---|
> | 8 | 6 | 4 | 2 |
> | 7 | 7 | 3 | 3 |
> | 8 | 5 | 2 | 6 |
> | 2 | 2 | 5 | 4 |

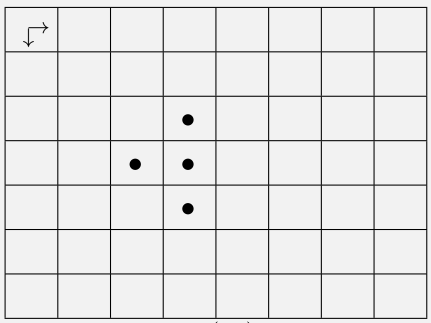
Problem 2: Morphological image processing [25 points]

(a) ($4 \times 2 = 8$ points) Consider the following binary image X (left) and the structuring element B (right).

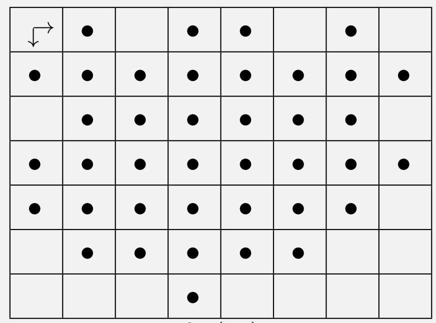




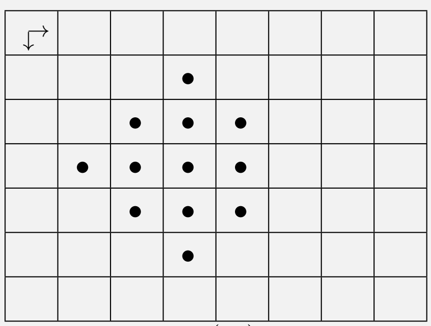
Draw in the empty images the erosion $\varepsilon_B(X)$, the dilation $\delta_B(X)$, the opening $\gamma_B(X)$, and the closing $\phi_B(X)$.



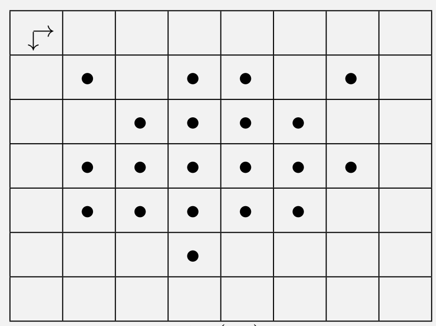
$\varepsilon_B(X)$



$\delta_B(X)$

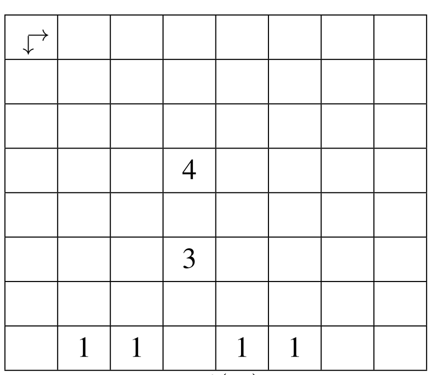


$\gamma_B(X)$

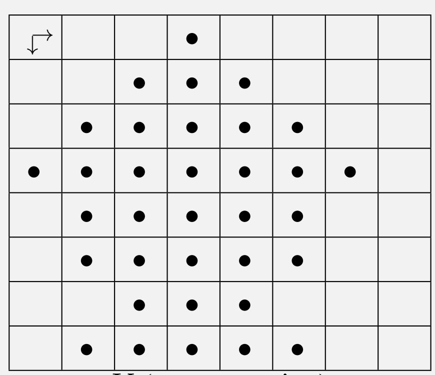


$\phi_B(X)$

(b) (6 points) Below, on the left, you find the *morphological skeleton function* $skf(Y)$ that was obtained with the structuring element that was given in part (a). Draw in the empty image (right) the reconstruction of Y.



$skf(Y)$



Y (reconstruction)

(c) (6 points) For your convenience, the image and structuring element of part (a) is copied to this page:

	•		•	•		•	
		•	•	•			
	•	•	•	•		•	
	•	•	•	•	•		
			•				

X

	•	
•	•	•
	•	

B

Compute the *morphological skeleton function* $skf(X)$ and place it in the empty grid below. Also, give the relevant sets $S_k(X)$ that you encounter in the construction of $skf(X)$ in the answer box.

	1			1		1	
			2				
		2	2			1	
	1		2		1		

The general formula for $S_k(X)$ is $S_k(X) = (X \ominus_k B) \setminus ((X \ominus_k B) \circ B)$, where $X \ominus_0 B = X$ and N is the largest integer such that $S_N(X) \neq \emptyset$.

So, $S_0(X) = X \backslash (X \circ B)$. We computed $X \circ B = \gamma_B(X)$ in part (a). So, it is easy to see that $S_0 = \{(1,1),(1,4),(4,1),(5,4),(6,1),(6,3)\}$ (here, starting from (0,0), the first coordinate is the column number, and the second the row number).

Next, to determine $S_1(X)$, we use that $X \ominus_1 B = \epsilon_B(X) = \{(2,3),(3,2),(3,3),(3,4)\}$ (which we computed in part (a)). Eroding this this set one more time yields the empty set, so the opening is empty. Hence, $S_1(X)$ is simply the erosion $E_B(X)$ itself, so $S_1(X) = \{(2,3),(3,2),(3,3),(3,4)\}$. Also, it is clear that $N = 1$.

This yields the skeleton function $skf(X)$ depicted in the above figure.

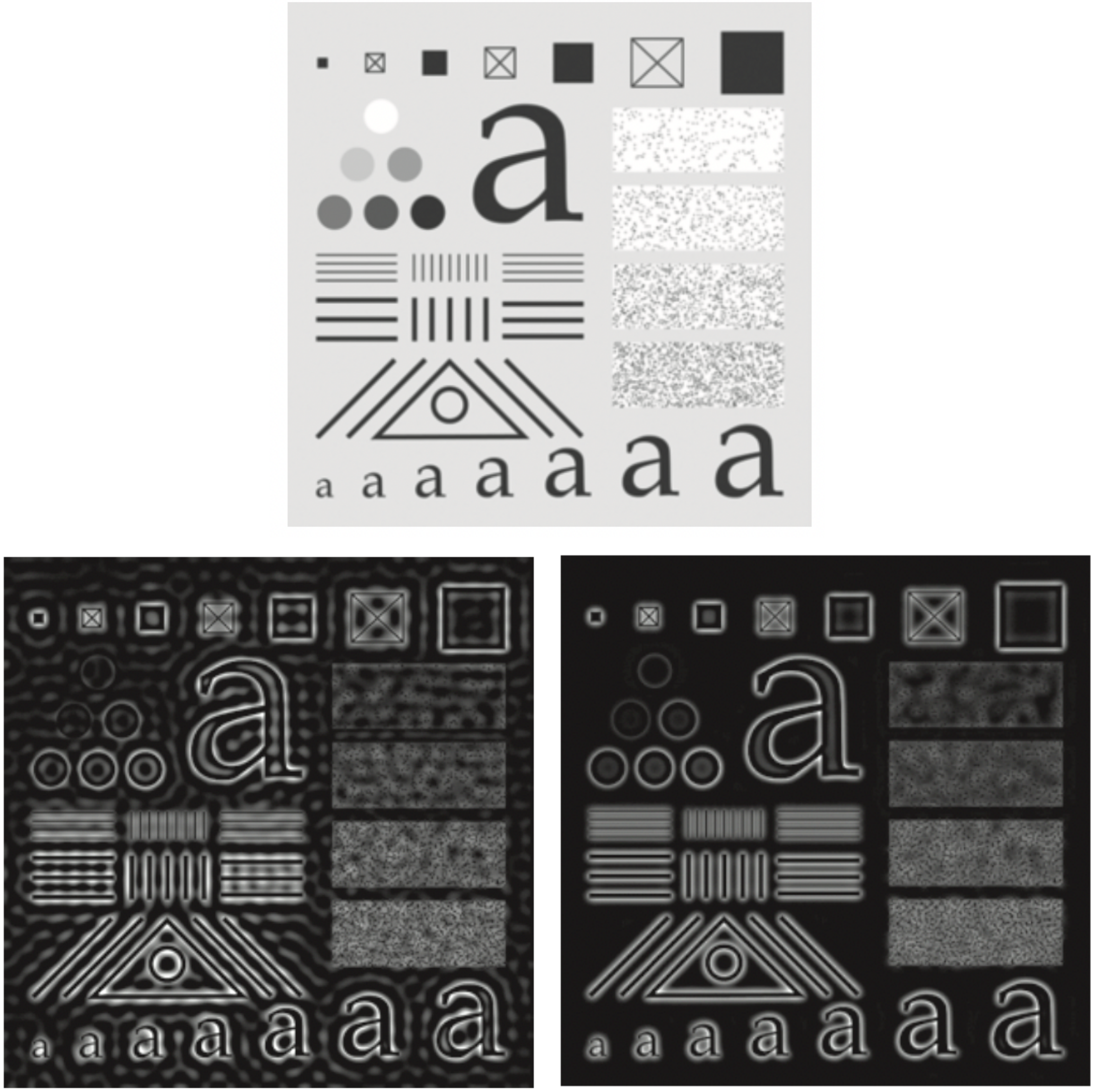
(d) (3 points) Explain what the expression $X \ominus \{(-1,-1)\}$ returns. Given an equivalent expression using the operation $\oplus$.

The erosion is defined as $X \ominus B = \bigcap_{b \in B} X_{-b}$. Since B is in this case a singleton set, this simply reduces to $X \ominus \{(-1,-1)\} = X_{-(-1,-1)} = X_{(1,1)}$. Hence, this particular erosion is simply a translation over the vector $(1,1)$. Clearly, this can also be achieved with the dilation $X \oplus \{(1,1)\}$.

(e) (2 points) Assuming the availability of a procedure that can efficiently compute distance transforms of binary images using a Manhattan metric or a chessboard metric, how can this be used to efficiently compute the erosion of a binary set X with the structuring element that was used in parts (a-c)?

The distance trasform must be used such that the distance of each foreground pixel to the nearest background pixel is computed. Next (for a single erosion) the result should be thresholded such that all pixels with a distance greater than 1 remain. Of course, the metric that we must use is the Manhattan metric.

Problem 3: Frequency domain filtering [15 points]



Top: Original image; Bottom left:IHPF-filtered, $D_0 = 60$; Bottom right:BHPF-filtered, $D_0 = 60$

An example of frequency domain filtering is the ideal highpass filter (IHPF), defined by the transfer function

$$H(\mu, v) = \begin{cases} 0 & \text{if } D(\mu, v) \leq D_0 \\ 1 & \text{if } D(\mu, v) > D_0, \end{cases}$$

$D(\mu, v)$ is the distance of the point (μ, v) to the origin of the frequency domain, and D_0 is is the cut-off radius.

(a) (5 points) What is the purpose of highpass filtering?

> The purpose of highpass filtering is to remove low-frequency structures and preserve the high-frequency structures in the image.

(b) (5 points) What is the artefact caused by IHPF in comparison to Butterworth high pass filter (BHPF)?

IHPF mainly causes the ringing artefacts, such as those blob structures inside/around the letter "a" in the filtered image. This is because the spatial representation of 1D IHPF is a sinc function, of which the small, outer lobs cause the ringing artefacts.

(c) (5 points) What are potential applications for highpass filtering?

The potential applications of highpass filtering are image sharpening, edge detection, etc.

Problem 4: Image compression [15 points]
Consider the following image:

125	80	80	80
125	255	255	80
125	255	255	80
200	255	80	80

(a) (5 points) What is the entropy of this image? What does this value indicate?

Consider the occurrence of the intensity values:

Intensity	Count	Probability
200	1	$\frac{1}{16}$
125	3	$\frac{3}{16}$
255	5	$\frac{5}{16}$
80	7	$\frac{7}{16}$

The entropy of the image is computed as:

$$\begin{aligned}\tilde{H} &= -\sum_{k=0}^{L-1} p_r(r_k) \log_2 \, p_r(r_k) \\ &= -(\frac{1}{16}\log_2\frac{1}{16} + \frac{3}{16}\log_2\frac{3}{16} + \frac{5}{16}\log_2\frac{5}{16} + \frac{7}{16}\log_2\frac{7}{16}) \\ &= 1.75 \; bit/pixel\end{aligned}$$

The minimum number of bits per pixel needed to code the image is 1.75.

(b) (8 points) If we use Huffman coding for this image, what would be the code book? Give the derivation, not only the result.

Huffman coding first orders symbols w.r.t. probability and successively combines lowest pair of symbols into new symbol. Each reduced source is then coded, starting with the smallest source and working back to the original source.

Original Source		**Reduced Source**	
Symbol	Probability	1	2
80	7/16	7/16	9/16
255	5/16	5/16	7/16
125	3/16	4/16	
200	1/16		

Original Source		**Reduced Source**	
Symbol (code)	Probability	1	2
80 (1)	7/16	7/16	9/16 0
255 (00)	5/16	5/16 00	7/16 1
125 (010)	3/16 010	4/16 01	
200 (011)	1/16 011		

The code for symbol 80, 255, 125, and 200 are 1, 00, 010, 011, respectively.
[Note: we also accepted the codes that interchanged the 0 and 1.]

(c) (2 points) This image is originally represented by 8-bit fixed-length encoding. What is the average number of bits per pixel of the image after Huffman coding? Does Huffman coding reduce the redundancy of the image?

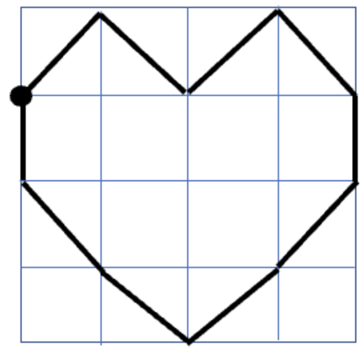
8-bit fixed-length code: $L_{avg} = 8$ bits

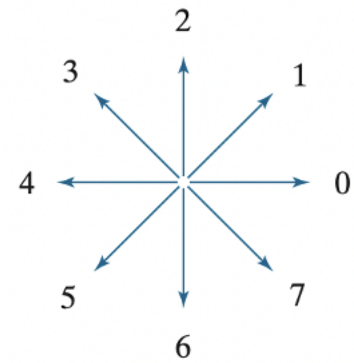
Huffman code:

$$\begin{aligned} L_{avg} &= \sum_{k=0}^{L-1} l(r_k)\, p_r(r_k) \\ &= 1 \times \frac{7}{16} + 2 \times \frac{5}{16} + 3 \times \frac{3}{16} + 3 \times \frac{1}{16} \\ &= 1.81 bits \end{aligned}$$

Yes, Huffman coding reduced the number of bits. The compression ratio is $C = \frac{8}{1.81} = 4.42$ and the redundancy is $R = 1 - \frac{1}{C} = 0.774$.

Problem 5: Boundary descriptor [15 points]





Consider the above figures. On the left, you see the boundary of an object. In the middle, you see a simplified digital version of the boundary. On the right, you see direction numbers that are used in 8-directional chain codes.

(a) (5 points) What is the 8-directional Freeman chain code of the simplified boundary? The dot indicates the starting point.

> The 8-directional Freeman chain code is 1717655332.

(b) (5 points) What is the first difference of the chain code? What is the shape number of this boundary?

> The freeman chain code : 1717655332
> The first difference of the code: 7626770607 (or 6267706077)
> The shape number of the boundary: 0607762677
> [Note: we also accepted the answers that count the number of changes clockwise.]

(c) (5 points) Describe another boundary descriptor and explain how the descriptor works.

> Many options from the slides, e.g. curvature, Fourier descriptors, etc. You name it.