EE3206/EE3206E ICVIP

Tutorial Set C – Solutions

Question 1

Part (a)

Compare the first number in the initial list with the second number. If the first number is smaller, interchange the positions; otherwise compare the second number with the third, and so on. At the end of the first iteration, the smallest number wil be found at the bottom of the list. Repeat the procedure for the next iteration. (See illustration below.)

	21		21		21		30
	14		14		30		52
	12		30		52		21
	30		52		14		14
	52		12		12		12
		$\longrightarrow$		$\longrightarrow$		$\longrightarrow$	
Number of comparisons		4		3		2	

Suppose there are N numbers in the list (where $N = k^2$).

Obtaining the smallest number requires $(N-1)$ comparisons

Obtaining the 2nd smallest number requires $(N-2)$ comparisons

Obtaining the 3rd smallest number requires $(N-3)$ comparisons

and so on.

Since we are only interested in the median value, there is no need to sort the entire list. The number of comparisons required to determine the median is

$$\begin{aligned} C_1 &= (N-1) + (N-2) + \ldots + \frac{1}{2}(N-1) \\ &= \frac{3}{8}(N^2-1) \\ &= \frac{3}{8}(k^4-1) \end{aligned}$$

Part (b)

As the window is moved from pixel to pixel, discard those pixel values that are no longer in the window, and insert the new pixel values in the appropriate places in the sequence.

Assume that the initial set of pixel intensities have already been fully sorted. Each time we shift the window, we drop k numbers and take in k new numbers. In the worst case:

> the 1st new number is compared with $(N - k)$ numbers
>
> the 2nd new number is compared with $(N - k) + 1$ numbers
>
> ...
>
> the kth new number is compared with $(N - k) + (k - 1)$ numbers

The total number of comparisons is

$$
\begin{aligned}
C_2 &= k(N-k) + \frac{k}{2}(k-1) \\
&= \frac{k}{2}(k-1)(2k+1)
\end{aligned}
$$

Part (c)

	C_1	C_2	C_1/C_2
$k = 3$	30	21	1.43
$k = 5$	234	110	2.13
$k = 7$	900	315	2.86

Question 2

f (original)

60	60	60	60	60	60	60	60
60	60	60	60	60	60	60	60
60	60	60	60	60	60	60	60
60	60	60	60	60	60	60	60
60	60	60	60	160	160	160	160
60	60	60	60	160	160	160	160
60	60	60	60	160	160	160	160
60	60	60	60	160	160	160	160

f1 (noise added)

60	72	66	63	70	46	46	80
64	72	60	44	76	40	76	50
70	58	48	64	76	78	50	76
64	0	50	68	56	40	74	64
60	54	74	52	158	146	162	152
42	54	60	68	164	140	142	148
66	52	78	54	160	172	174	166
58	40	50	66	156	146	180	142

f2 (nbhd averaging)

63	61	63	62	62	60	
54	52	60	60	63	61	
53	52	72	82	93	94	
51	53	83	99	120	119	
60	61	96	124	158	156	
56	58	95	125	159	157	.

f3 (median)

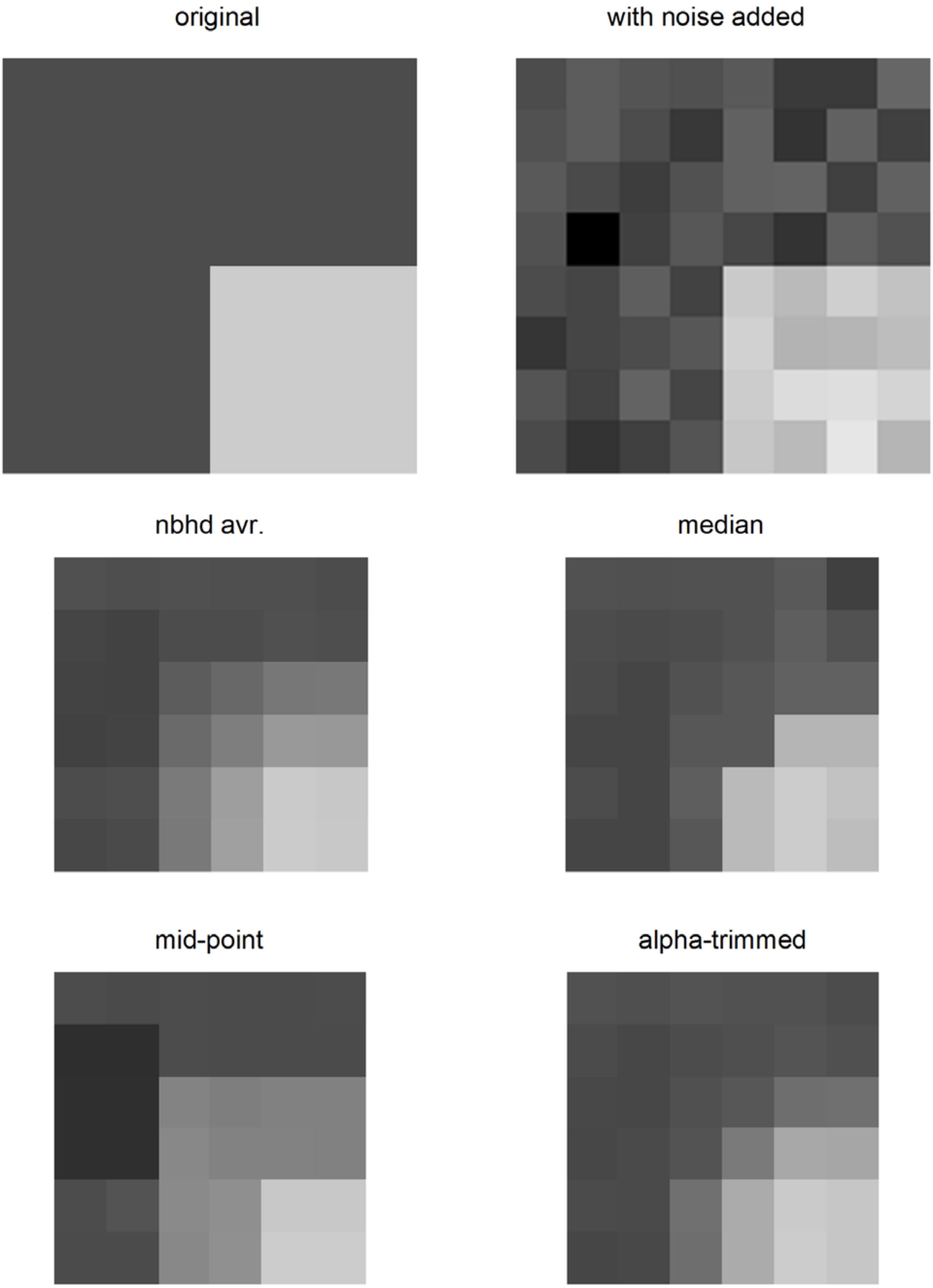
64	63	64	64	70	50	
60	58	60	64	74	64	
58	54	64	68	76	76	
54	54	68	68	142	142	
60	54	74	146	160	152	
54	54	68	146	160	148	.

f4 (mid-point)

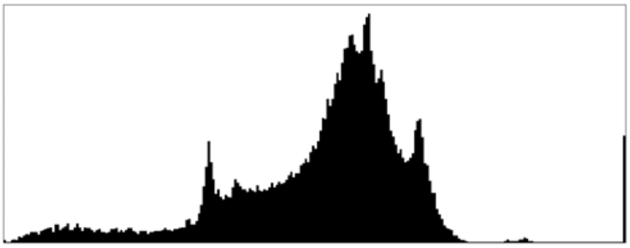
60	58	60	59	59	60	
36	36	60	59	59	59	
37	37	103	99	101	101	
37	37	107	102	102	101	
60	65	108	112	157	157	
59	59	107	113	160	160	.

f4 (alpha-trimmed)

64	62	65	64	64	60	
59	56	60	62	66	63	
57	56	63	68	86	88	
56	58	65	96	132	130	
59	58	88	134	158	155	
55	57	86	134	160	155	.


original
with noise added
nbhd avr.
median
mid-point
alpha-trimmed

Question 3



Part (a)

The output of the MMSE filter is:

$$g(x,y) = f(x,y) - \frac{\sigma_\eta^2}{\sigma_l^2}[f(x,y) - m_l(x,y)]$$

$$\begin{array}{rcl} \sigma_\eta^2 & = & \text{noise variance} \\ \sigma_l^2 & = & \text{local variance (in the window under consideration)} \\ m_l & = & \text{local mean (in the window under consideration)} \end{array}$$

Consider a 3×3 neighbourhood centred at a noise point; one out of nine pixels is an extreme value (255). Compare this local variance with the noise variance, where the latter is determined by the fact that 0.01 (or 1 in 100) of the pixels is a noise pixel (of value 255). Hence, it is likely that the local variance will be much greater than the noise variance:

$$\sigma_l^2 \gg \sigma_\eta^2$$

Thus the output at this point is

$$g(x,y) \approx f(x,y)$$

i.e., there is very little filtering of the noise value. Hence, the MMSE filter will have very little effect on the image.

Part (b)

The image formed by averaging K different noisy images is

$$\bar{g}(x,y) = \frac{1}{K}\sum_{t=1}^{K} g_t(x,y)$$

Consider a pixel at (p,q). Suppose $g_1(p,q) = 255$, i.e., pixel (p,q) in the first image is a noise pixel. This value of 255 is likely to be very different from the noise-free pixel value, $f(p,q)$. For example, if we have 10 frames,

$$\begin{aligned}
g_1(p,q) &= 255 \\
g_2(p,q) &= f(p,q) = 100 \\
g_3(p,q) &= f(p,q) = 100 \\
g_4(p,q) &= f(p,q) = 100 \\
\dots & \\
\\
\bar{g}(p,q) &= \frac{1}{10}(255 + 100 + 100 + ...) = 116
\end{aligned}$$

which is still substantially different from the noise-free value of 100.

Hence, many image frames will be required before $\bar{g}(x,y)$ approaches $f(x,y)$, i.e., image averaging is not suitable.