DIGITAL IMAGE PROCESSING

Quiz exercises

In the following set of questions, there are, possibly, multiple correct answers (1, 2, 3 or 4). Mark the answers you consider correct.

1. If the spectrum of a continous (not sampled) image is the one in Fig. 1.a), then the spectrum of its sampled version is, most likely:

   a) the one in Fig. 1.b);
   b) the one in Fig. 1.c);
   c) the one in Fig. 1.a);
   d) the one in Fig. 1.d).

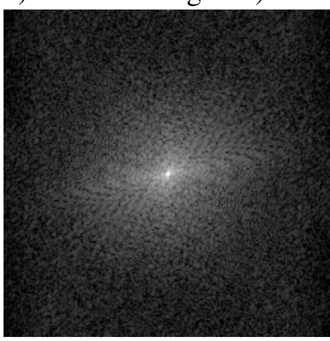


Fig. 1.a)

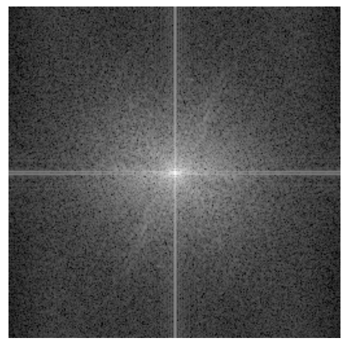


Fig. 1.b)

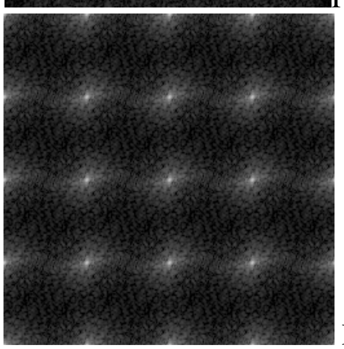


Fig. 1.c)

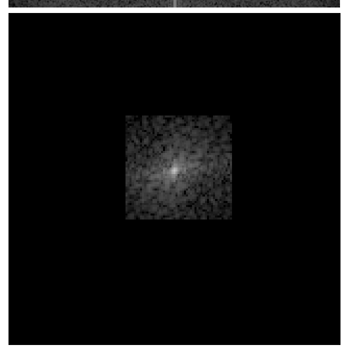


Fig. 1.d)

2. A sampled (but not quantized) image, whose brightness before quantization, in each spatial position, can take values in the range [0mV;250mV], has the linear histogram of its brightness represented (approximately) in Fig. 2.a). After the quantization, the histogram of the resulting digital image is the one in Fig. 2.b). Then, most likely, the quantizer that was used is:

   a) an 8 bit uniform quantizer;
   b) a 2 bit optimal quantizer;
   c) a 4 bit optimal quantizer;
   d) a 2 bit uniform quantizer.

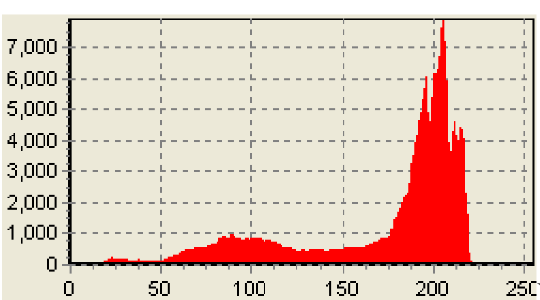


Fig. 2.a)

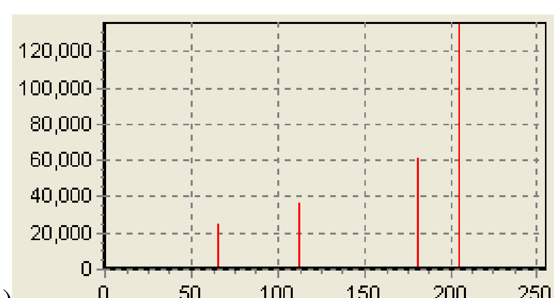


Fig. 2.b)

3. If the result of the uniform quantization at 2 bits (4 quantization levels) of a sampled image is shown in Fig. 3.a), then the result of the quantization of the same image with the same quantizer, but using the pseudo-random noise quantization technique, is most likely:

a) The same as the image in Fig. 3.a), just with more noise added;

b) The one in Fig. 3.b), because in such a scheme, the noise is subtracted after quantization;

c) The one in Fig. 3.c);

d) The one in Fig. 3.d).

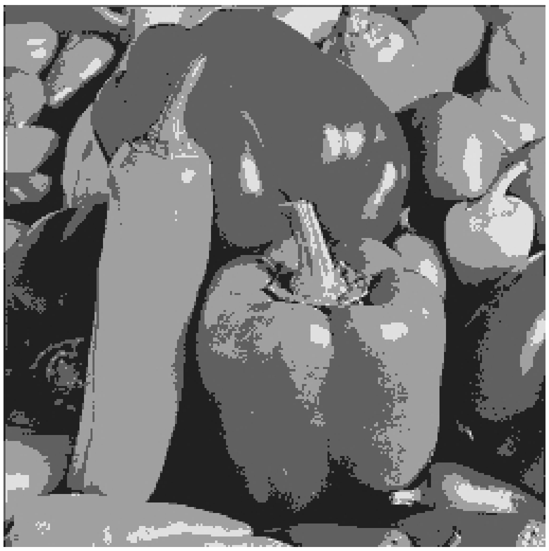


Fig. 3.a)

Fig. 3.b)

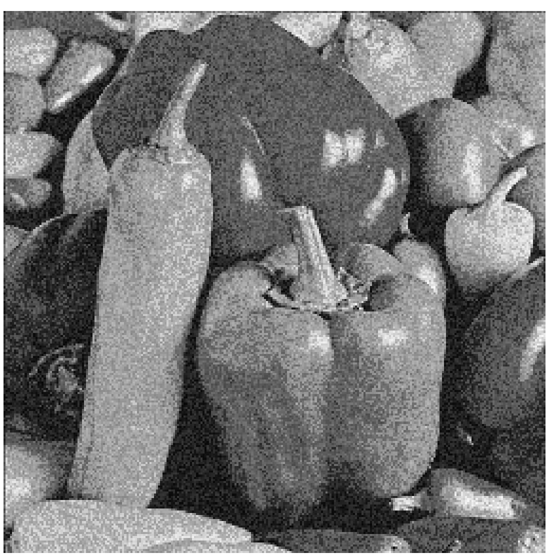


Fig. 3.c)

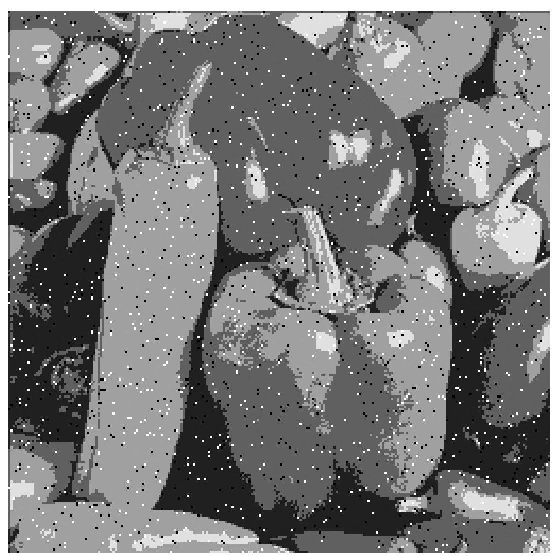


Fig. 3.d)

4. A uniform 2-bits quantizer for the input brightness range [0;220][mV] must have:

a) the decision levels: t_0=0; t_1=55; t_2=110; t_3=165; t_4=220;

b) the decision levels: t_0=00; t_1=01; t_2=10; t_3=11;

c) the reconstruction levels: r_0=27.5; r_1=82.5; r_2=137.5; r_3=192.5;

d) the reconstruction levels: r_0=0; r_1=27.5; r_2=82.5; r_3=137.5; r_4=192.5; r_5=220.

5. Consider the original image from Fig. 4.a) (an original image affected by salt and pepper noise). Most likely, the Fourier amplitude spectrum of this image will look:

a) as in Fig. 4.b);
b) as in Fig. 4.c);
c) as in Fig. 4.d);
d) as in Fig. 4.e).

Fig. 4.a)

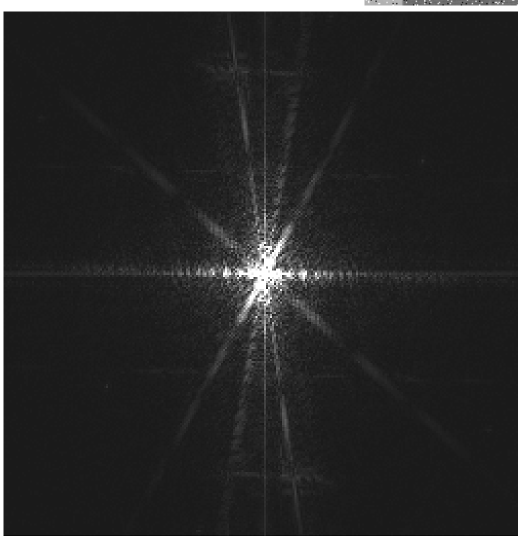


Fig. 4.b)

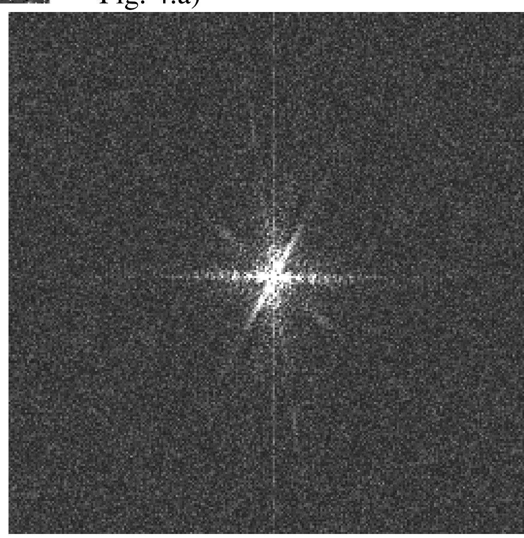


Fig. 4.c)

Fig. 4.d)

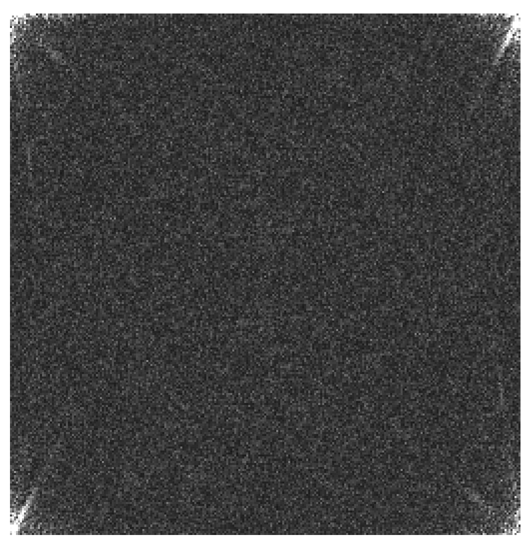


Fig. 4.e)

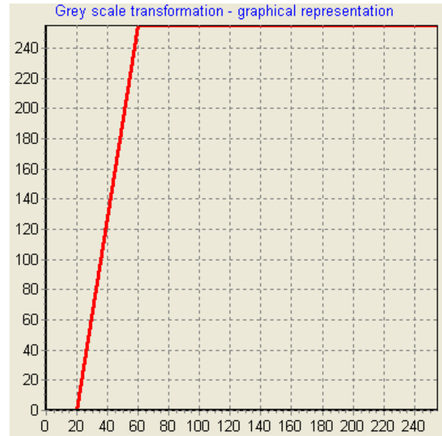
6. The greyscale clipping function in the right is applied on an image having the histogram in Fig. 5.a). Then, most likely, the histogram of the processed image will look:

a) like in Fig. 5.b);

b) like in Fig. 5.c);

c) like in Fig. 5.d);

d) the same as before processing, since we have a linear function.



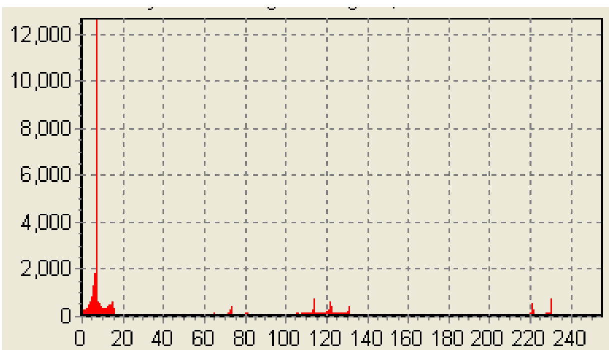


Fig. 5.a)

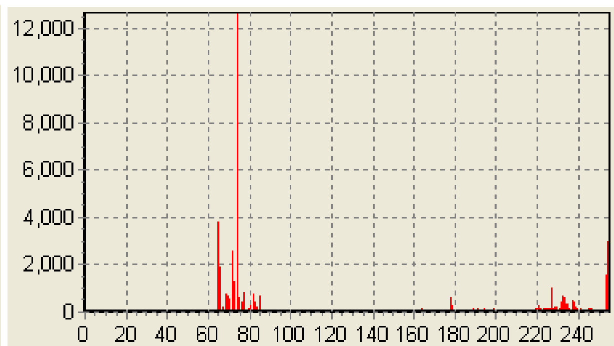


Fig. 5.b)

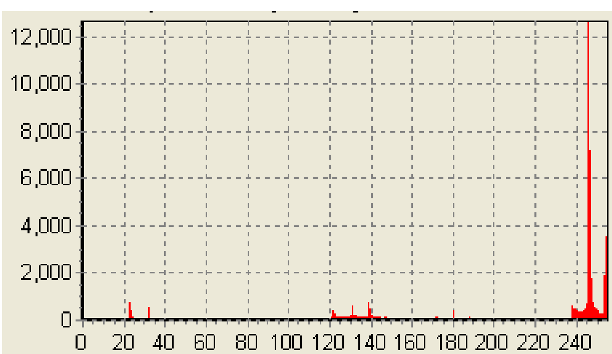


Fig. 5.c)

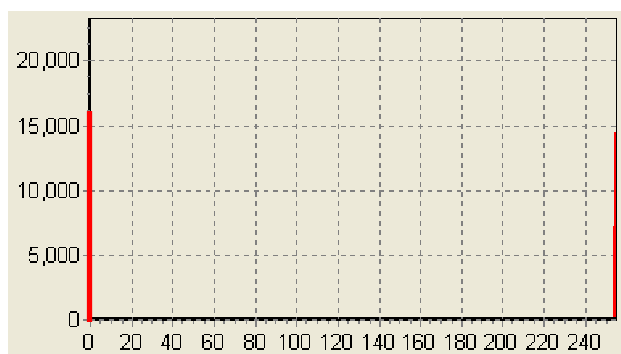


Fig. 5.d)

7. In order to obtain the image in Fig. 6.b) from the original image in Fig. 6.a), the following point processing operation should be applied:

a) contrast compression;

b) negativation;

c) histogram equalization;

d) histogram modification.

Fig. 6.a

Fig. 6.b.

8. On the original grey scale image from Fig. 7. a), which of the following point processing operations could have been applied to obtain the image in Fig. 7.b)?

a) Contrast compression;
b) Negativation (image inversion);
c) Some grey scale slicing operation;
d) Extraction of the most significant bit.

Fig. 7.a

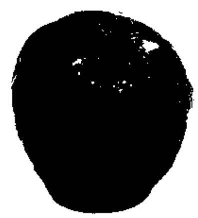


Fig. 7.b

9. Fig. 8.a) represents the grey level histogram of a digital image. After processing this image, one gets another grey level digital image with the grey level histogram shown in Fig. 8.b). What is the most plausible processing applied on the original image from the ones below?

a) Grey scale inversion (negative of the original image);
b) Binary thresholding;
c) Histogram equalization;
d) Some grey scale slicing.

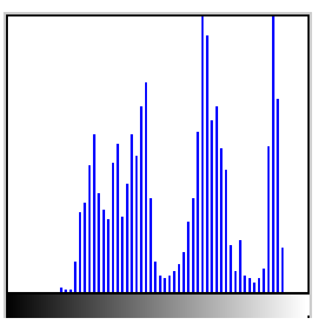


Fig. 8.a)

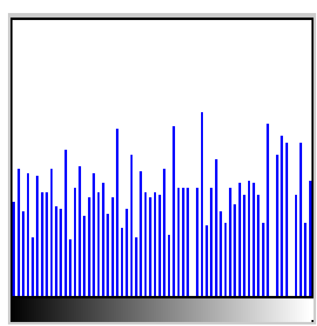


Fig. 8.b)

10. The image in Fig. 9.a) represents an original image of low contrast, having the linear grey level histogram given in Fig. 9.b). If on this image, a point processing (grey scale transformation) is applied, using the transfer function from Fig. 9.c), which of the following will be the resulting image?

a) the same as the original (Fig. 9.a));
b) the image in Fig. 9.d);
c) the image in Fig. 9.e);

d) the image in Fig. 9.f).

Fig. 9.a)

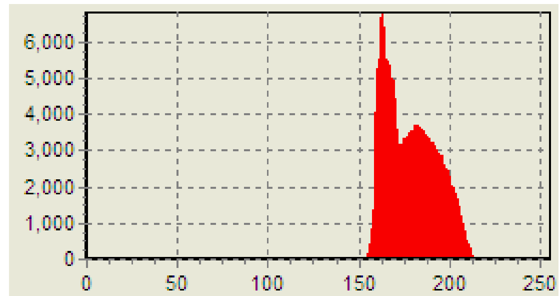


Fig. 9.b)

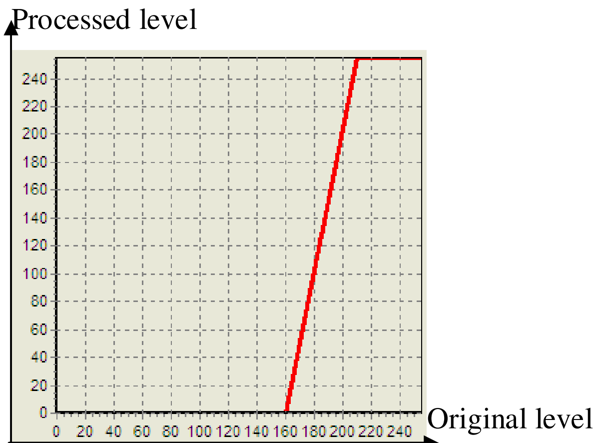


Fig. 9.c)

Fig. 9.d)

Fig. 9.e)

Fig. 9.f)

11. If the original image is the one in Fig. 10.a), and the resulting image after some processing is the one in Fig. 10.b), what is the most likely processing from the list below to give this result?

a) Edge detection by a Laplacian operator;
b) High pass filtering, by subtracting a low pass filtered version of the image from the original image;
c) Median filtering followed by an edge detection;
d) Edge detection followed by a median filtering.

Fig. 10.a)

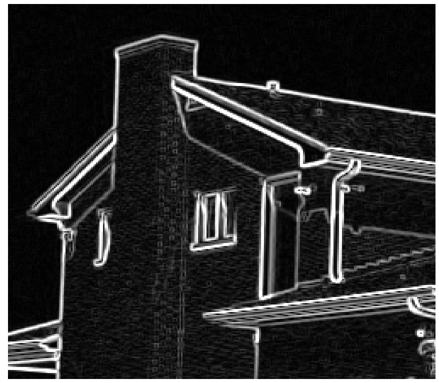


Fig. 10.b)

12. If on the image in Fig. 11.a) one applies a convolution with the 3×3 pixels mask $\mathbf{H} = \frac{1}{8}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$, and the absolute values of the convolution results are displayed, then most likely, the result of this operation is:
a) the image in Fig. 11.b);
b) the image in Fig. 11.c);
c) the image in Fig. 11.d);
d) the image in Fig. 11.e).

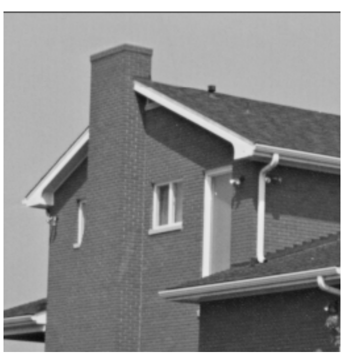


Fig. 11.a)

Fig. 11.b)

Fig. 11.c)

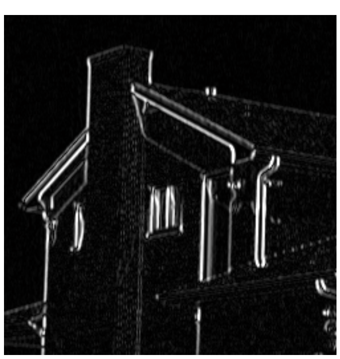


Fig. 11.d)

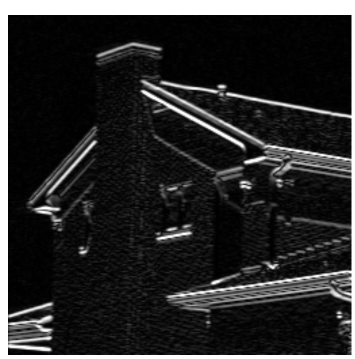


Fig. 11.e)

13. The image in Fig. 12.a) is, most probable, obtained from:

a) The image in Fig. 12.b), after an edge detection
b) Subtracting, pixel by pixel, the image in Fig. 12.c) from the image in Fig. 12.b)
c) The segmentation through amplitude thresholding of the image in Fig. 12.b)
d) The image in Fig. 12.d), after an edge detection


Fig. 12.a)

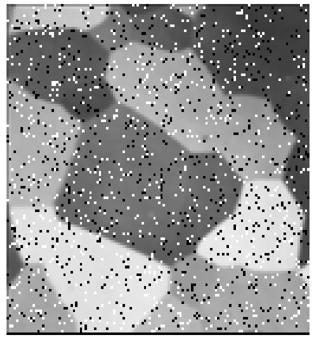


Fig. 12.b)

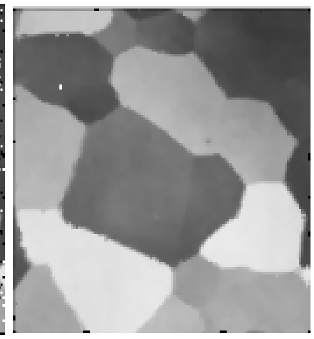


Fig. 12.c)


Fig. 12.d)

14. The image in Fig. 13.b) is obtained by convolving the image in Fig. 13.a) with a 3×3 convolution mask. Which of the following masks could have been used to give this processing result?

a) $\mathbf{H}_1 = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$;

b) $\mathbf{H}_2 = \frac{1}{8}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$;

c) $\mathbf{H}_3 = \frac{1}{4}\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$;

d) $\mathbf{H}_4 = \frac{1}{4}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$.


Fig. 13.a)


Fig. 13.b)

15. Consider a binary image was quad-tree encoded, and the resulting code sequence for this image is: g(g(wwbw)g(wwwg(bbbw))g(g(wbbb)www)g(wg(bbww)ww)), where b is the code for a black uniform pixels block; w is the code for a white uniform pixels block; g is the code for a block containing both black

and white pixels. Considering the object pixels are black and the background pixels are white, what is the area of the object region, in pixels?

a) equals the number of “b” symbols in the code sequence, that is, 9
b) equals the minimum size of a block (that is 4 pixels) multiplied by the number of “b” symbols in the code sequence (9), thus, it is 36
c) cannot be computed, since the direction and the initial block for encoding are not known;
d) is 12 pixels.

16. Consider the binary image from Fig. 14.a). The grey scale image from Fig. 14.b) was more likely obtain from the image in Fig. 14.a) by:

a) The binary erosion with a circular structural element;
b) Applying the distance transform on the object region in Fig. 14.a) (where the distance was scaled in the range [0;255]), considering the black pixels are the object region and the white pixels – the background;
c) Some contrast enhancement algorithm;
d) The representation (by grey levels) of the distances of the object pixels (black pixels) to their nearest boundary (the largest the grey level, the largest the distance of an object pixel to its nearest boundary).

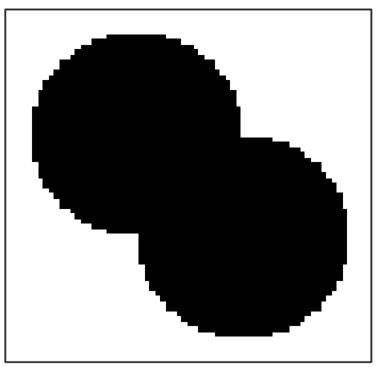


Fig. 14.a)

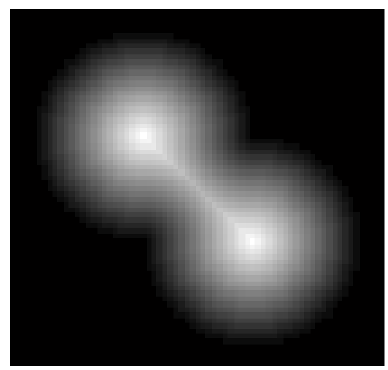


Fig. 14.b)

17. The image in Fig. 15.a) is an original grey scale digital image. Then, the image in Fig. 15.b) is, most probable, the result of:

a) an amplitude thresholding segmentation of the original image;
b) an edge-based segmentation of the original image;
c) a split-and-merge segmentation of the original image;
d) a region growing segmentation of the original image.

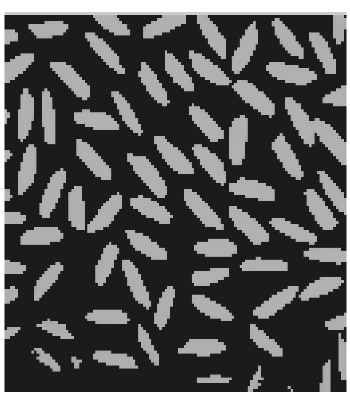


Fig. 15.a)

Fig. 15.b)

18. A binary (black and white) image of size 8×8 pixels, in which the homogeneous square black pixels blocks are encoded by b, the homogeneous square white pixels blocks – by w, and the non-homogeneous square pixels blocks – by g, is described, using the quad tree encoding, by the following sequence of symbols: ggwbbwgbgbwwbwwgbwwwgwwbw. This binary image contains:
   a) 12 white pixels, the remaining being black pixels;
   b) 12 white pixels, 6 grey pixels and 7 black pixels;
   c) 42 white pixels, the remaining being black pixels;
   d) 5 blocks of 2×2 black pixels and other 2 black pixels in non-homogeneous 2×2 pixels blocks.

19. The skeleton of the black object in Fig. 16.a) is, most likely:
   a) the one in Fig. 16.b);
   b) the one in Fig. 16.c);
   c) the one in Fig. 16.d);
   d) the one in Fig. 16.e).

Fig. 16.a)

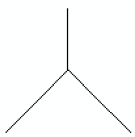


Fig. 16.b)

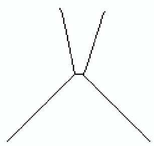


Fig. 16.c)

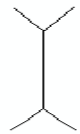


Fig. 16.d)

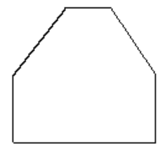


Fig. 16.e)

20. The matrix **V** represents, most likely:
   a) the distance transform of a 9×9 square of object pixels, placed on a uniform background;
   b) the distance transform of a 7×7 square of object pixels, placed on a uniform background;
   c) the amplitude spectrum of the Fourier transform of an image containing a 7×7 pixels black square, placed on a uniform background;
   d) the result of thresholding an image containing a 7×7 pixels black square, placed on a uniform background.

$$\mathbf{V} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 2 & 2 & 2 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 & 3 & 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 & 4 & 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 & 3 & 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 2 & 2 & 2 & 2 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

21. Consider the original grey level image in Fig. 17.a), and its linear grey level histogram in Fig. 17.b). Then, the image in Fig. 17.c) represents, most likely:
   a) the result of the edge detection applied on the image in Fig. 17.a), using compass operators;
   b) the result of the histogram-based segmentation of the image in Fig. 17.a) (using thresholds selected from the histogram);

c) the result of the segmentation of the image in Fig. 17.a) in 3 classes, based on the brightness only;
d) the result of the PCM encoding of the image in Fig. 17.a), using a 1 bit quantizer.

Fig. 17.a)

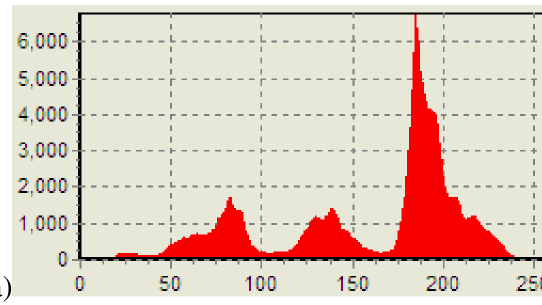


Fig. 17.b)

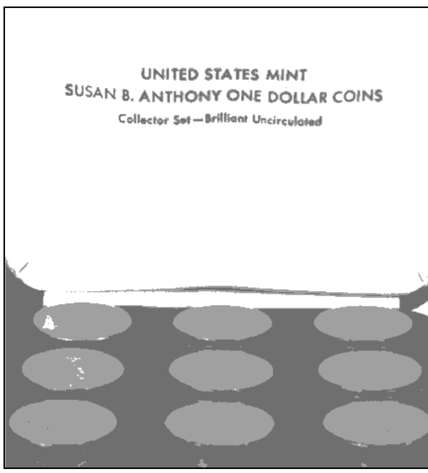


Fig. 17.c)

22. Assume the still image in Fig. 18.a) was compressed by one of the techniques studied in the lecture. If the image reconstructed at the decoder is the one in Fig. 18.b), then most likely, the coding technique was:
a) 1-D DPCM with a predictor of one tact and a 2 bits uniform quantizer;
b) based on the DCT transform coding, applying the DCT on the whole image;
c) transformed-based image compression, using 2-D DCT applied on 8×8 pixels blocks;
d) PCM, using a quantizer with at least 2 bits.

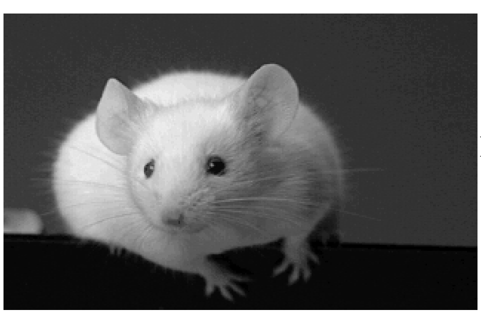


Fig. 18.a)

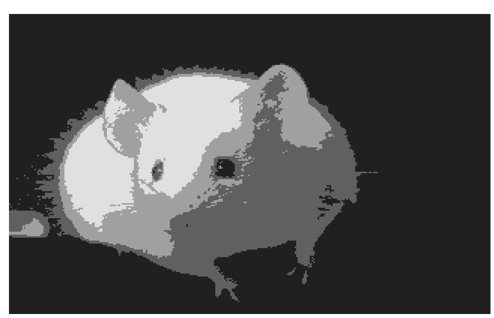


Fig. 18.b)

23. Consider the original grey scale digital image in Fig. 19.a). After applying a lossy encoding procedure and the corresponding decoding, the reconstructed image is the one in Fig. 19.b); one can notice that only 4 grey levels are present in the output image. Most likely, the encoding method was:

a) PCM, using a 4 bits uniform quantizer;
b) Delta Modulation;
c) 2-D DPCM, using for the error quantization – a 2 bits uniform quantizer;
d) PCM, using a uniform quantizer with 4 quantization levels.

Fig. 19.a)

Fig. 19.b)

Quiz solutions

1. According to the 2-D sampling theory, the spectrum of the sampled image is obtained as a set of scaled replicas of the original image spectrum, "placed" around multiples of the sampling frequencies, horizontally and vertically. Therefore, since the only image in Fig. 1 matching this theoretical knowledge is Fig. 1.c) (which looks like a periodical repetition of some scaled version of the original image spectrum shown in Fig. 1.a)), it means the only possible correct answer is **b)**.

2. Let us examine the two figures (Fig. 2.a) – the histogram of the sampled, but non-quantized image, and Fig. 2.b) – the histogram of the sampled and quantized image), and the possible options for the most plausible quantizer used.

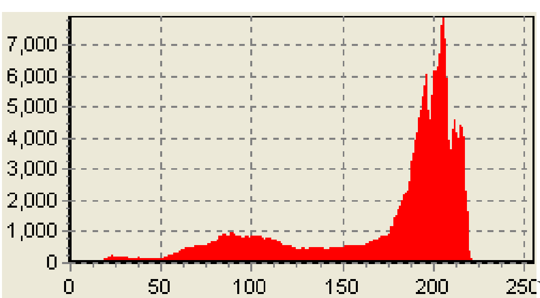


Fig. 2.a)

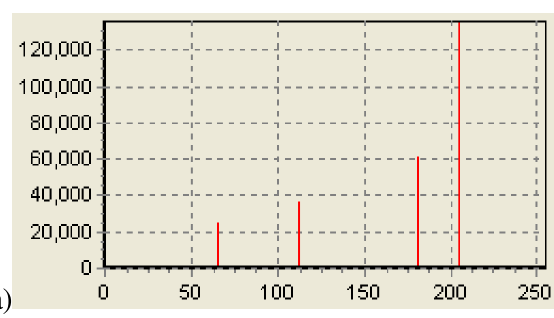


Fig. 2.b)

The first answer, a), sais the quantizer used is an 8 bit uniform quantizer; this would mean we have 2^8 =256 quantization levels ⇔ 256 possible grey levels in the output image (therefore, also 256 possible bins in the resulting histogram), and 256 quantization intervals of equal width. But if such a quantizer would have been used, since the non-quantized brightness range is [0mV;250mV], it would mean we would have the width of each quantization interval of 250/256 = 0.976 mV. Since the histogram of the non-quantized image contains data at least in the range [50mV;225mV], it means we would have use much more quantization intervals (therefore – much more quantization levels) in the output, quantized image, than just 4 levels as seen in the histogram of the quantized image in Figure 2.b). Therefore, the first answer cannot be the correct answer.

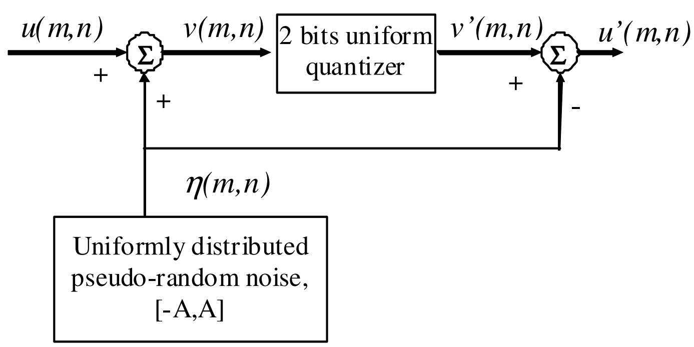
The second answer, b), sais the quantizer used is a 2 bit optimal quantizer; a 2 bit quantizer would give $2^2 = 4$ quantization levels ⇔ 4 possible grey levels in the output image (therefore, also 4 possible bins in the resulting histogram) – as it is the case in the histogram of the quantized image, where we indeed have 4 bins. Furthermore, if the quantizer is optimal, the width of the quantization intervals does not have to be the same for the entire signal range, and therefore also the quantization levels (which are placed exactly in the middle of the quantization intervals) are usually not equally spaced. This also matches the histogram observed in Fig. 2.b). Therefore the second answer is correct, it is possible to use a 2 bit optimal quantizer and to obtain a quantized image having the histogram as in Fig. 2.b)

The third answer, c), sais the quantizer used is a 4 bit optimal quantizer. In that case, we would have $2^4 = 16$ quantization intervals and 16 quantization levels available. Again, as in the first case (option a)), it is impossible in that case to obtain only the four bins in the output histogram, since the quantization intervals would have the width: 250/16 = 15.625mV, therefore the quantization intervals would be: [0;15.625), [15.625; 31.25), [31.25; 46.875), [46.875; 62.5), [62.5; 78.125), [78.125; 93.75) ..., which means, so far, that we should have at least two bins in the output histogram corresponding to the range [50;100], and this is obviously not the case in Fig. 2.b). Therefore the quantizer used couldn't be a 4 bit uniform quantizer.

Finally, the last answer, d), which indicates as possible a 2 bit uniform quantizer, cannot be correct, since this would mean the 4 bins in the histogram in Fig. 2.c) (the quantization levels) should be equally spaced (as they should be in the uniform quantizer's case) – and is obvious that they are not.

Therefore the only possible (correct) answer is **b)**.

3. According to the theory of uniform quantization using pseudo-random noise, the block diagram of the quantizer looks as in the figure below:



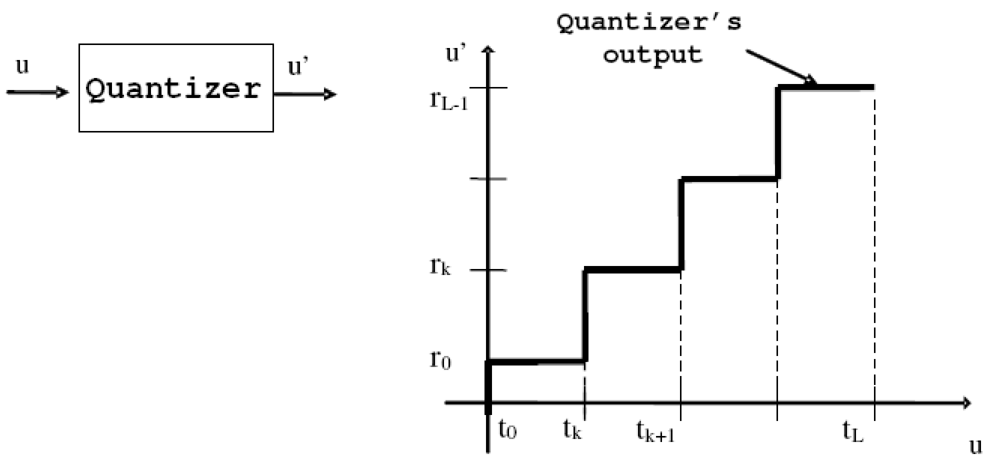
Since a pseudo-random noise is first added to the input image prior to quantization, and then subtracted from the output image after its quantization, it means we can expect some amount of noise to still be visible in the output image. Therefore, the quantized image cannot look like in Figure 3.b), since this is a noise-free image (it actually looks exactly like the original grey level "Peppers" image). Therefore, the answer **b)** cannot be correct.

The first answer, **a)**, again cannot be correct, since the pseudo-random noise is applied on the original image **prior to quantization**; therefore, the brightness in each point changes prior to its input to the quantizer, and this brightness is quantized, and of course, quantizing $u(m,n)+\eta(m,n)$ is not the same as the sum of the quantized $u(m,n)$ and not quantized $\eta(m,n)$ (noise).

As possible answers, we still have **c)** and **d)**. By examining Fig. 3.d), the image looks like the one in Fig. 3.a), just with some impulse noise added; this is not the usual case for a pseudorandom noise quantization scheme, where the noise level is much smaller, and its spatial distribution is more frequent, to allow the desired effect of this quantization scheme, that is: creating variations around the decision thresholds of the quantizer, to avoid for the false contours to appear in the quantized image. As one can see in Fig. 3.d), we still have the false contours, just that we also have some high amplitude noise, thus not achieving the desired goal. Therefore, the answer **d)** is not correct either.

However, the image in Fig. 3.c) looks exactly as one would expect from a 2-bits quantizer with pseudorandom noise: the false contours disappear, there is some noise uniformly distributed on the image (due to its addition prior to quantization and its subtraction after the quantization from the quantized version of the image, one can see more than 4 grey levels present in the output image), but the amplitude of the noise is rather small as compared to the grey level dynamics of the original image. Thus, the correct answer is **c)**: the result of the quantization of the image using a 2 bits uniform quantizer in a pseudorandom noise quantization scheme is, most likely, the image in Fig. 3.c).

4. For a uniform 2-bits quantizer for the input brightness range [0;220][mV], the decision levels and the reconstruction levels are uniformly spaced over the input brightness range. The decision levels of the quantizer define the intervals of input values which are mapped after the quantization to the reconstruction levels, and each reconstruction level is found exactly in the middle of an interval (in the middle between two successive decision levels), as illustrated in the following figure.



The decision levels are typically denoted by t_k, k=0,1,...,L, where L is the number of decision intervals of the quantizer (which of course – equals the number of reconstruction levels of the quantizer). For the quantizer given in this exercise, since it is a 2-bits quantizer, it will allow for $2^2 = 4$ reconstruction levels and 4 decision intervals, therefore, L=4.

This means we will have 5 decision levels (which are needed to define the 4 decision intervals), denoted here by t_0, t_1, t_2, t_3, t_4, and 4 reconstruction levels, denoted here by r_0, r_1, r_2, r_3. Therefore, even just judging by the number of decision levels, one can see that the second option of the quiz, answer **b)**, cannot be correct, since it states that there are only four decision levels instead of five; with four decision levels, one can only define three decision intervals, thus not enough for the given quantizer. By the same reasoning, the fourth option, **d)**, cannot be correct either, since it states that we should have six reconstruction levels, which is impossible for a 2-bits quantizer (this only allows for four reconstruction levels, L=4). Thus, the only candidates as correct answers are the options **a)** and **c)**.

In order to decide if they are correct, let us compute the decision levels and the reconstruction levels of the 2-bits uniform quantizer for the input range [0;220][mV]. The equations that give the expressions of the decision levels and of the reconstruction levels of the uniform quantizer are the following:

$$q = \frac{t_L - t_0}{L},$$
$$t_k = t_{k-1} + q, \quad k = 1,2,...,L;$$
t_0 = the left value of the input data range,

t_L = the right value of the input data range,

$$r_k = t_k + \frac{q}{2}, \quad k = 0,1,2,...,L-1.$$

Therefore, in our case, $t_0=0$; $t_L=t_4=220$; $q=220/4=55$, and the other values result as:

$t_1=t_0+q=0+55=55$;
$t_2=t_1+q=55+55=110$;
$t_3=t_2+q=110+55=165$;
$t_4=t_3+q=165+55=220$, - normal, since $t_4=t_L$=the right value of the input data range

These are exactly the values for the decision levels given by the answer **a)** of the quiz, thus, the answer **a)** is correct.

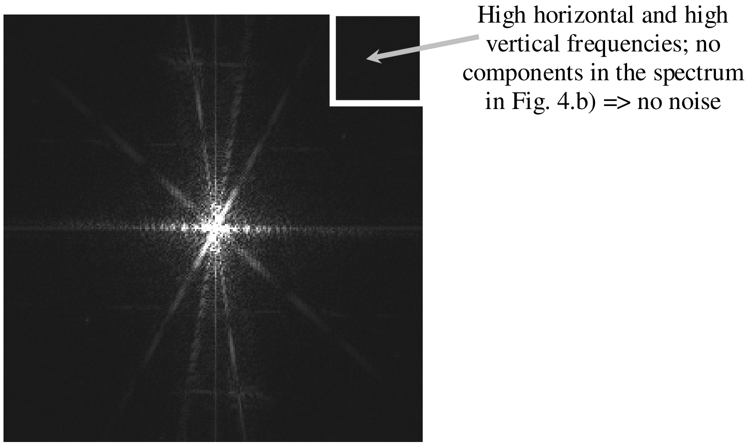
Using the equation for r_k, we compute also the reconstruction levels:

$r_0=t_0+q/2=0+55/2=27.5$;
$r_1=t_1+q/2=55+55/2=55+27.5=82.5$;
$r_2=t_2+q/2=110+55/2=110+27.5=137.5$;
$r_3=t_3+q/2=165+55/2=165+27.5=192.5$.

But these are exactly the values for the reconstruction levels given by the answer **c)** of the quiz, thus, the answer **c)** is as well correct.

This means that **both the answers a) and c) of the quiz are correct,** whereas the answers **b)** and **d)** are false.

5. Examining the original image in Fig. 4.a) (an original image affected by salt and pepper noise), one can expect that the noise component will introduce some random high frequency details in the Fourier spectrum of the image. In respect to this observation, the spectrum in Fig. 4.b) does not correspond to the noise-affected image, but rather to a “clean” (noise-free) image, since it does not have noticeable components in the simultaneously high horizontal and high vertical frequencies:

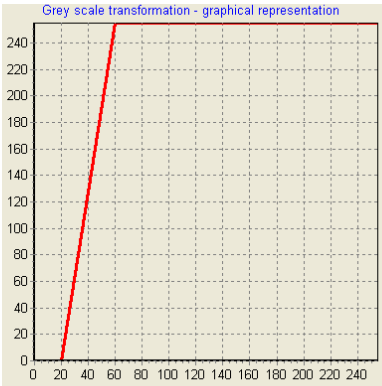


Thus, the first answer, **a)**, cannot be correct. However, the second spectrum, represented in Fig. 4.c), presents high frequency components (corresponding to high horizontal and vertical frequencies), thus the second answer, **b)**, is correct. The same Fourier spectrum (depicted in Fig. 4.c)) is also represented in Fig.

4.e), with the only difference that in the latter case, the representation of the Fourier spectrum considers the origin in the lower left corner, instead of the center (as in Fig. 4.c)); the four corners display the complex conjugate symmetry specific to the Fourier transform. Thus, the last answer, **d)**, is also correct. However the third answer, **c)**, indicating Fig. 4.d), is not correct, since obviously, in Fig. 4.d) we simply have the result of an edge detection applied to the noise-free version of the original image in Fig. 4.a).

In conclusion, **the answers b) and d) of the quiz are correct,** whereas the answers **a)** and **c)** are false.

6. If one looks at the grey scale transformation presented, one can describe it as a grey scale clipping, with the following operation (see below):

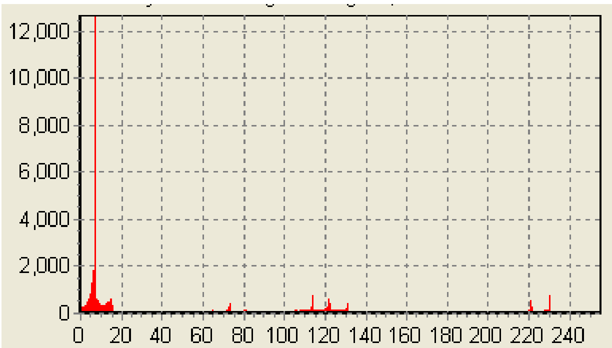


- any grey level lower than 20 in the original image will be set to black;
- any grey level above 60 in the original image will be set to white;
- the contrast in the brightness range [20; 60] in the original image will be increased at maximum, since this input brightness range will be mapped on the output image in the brightness range [0; 255], according to the equation:

$$v = \begin{cases} 0, & if \quad u < 20 \\ \dfrac{u-20}{40} \cdot 255, & if \quad u \in [20;60], \\ 255, & if \quad u > 60 \end{cases}$$

where by u we denote the brightness in a certain spatial position in the original image, and by v – the resulting brightness (after processing) in the same spatial position in the output (processed) image.

Now, if we examine the histogram of the original image, presented in Fig. 5.a):



we can see that **there are no pixels in the brightness range [20; 60],** and all the pixels of the original image are either under 20 (most of them, as we can see), or above 60! Therefore, the only brightness levels that will appear in the resulting image will be 0 and 255 (according to the analytical expression of the transformation, written above). This means that the histogram of the resulting image should have only two bins: one (with most pixels) on 0, and the other (with fewer pixels) on 255. Therefore, Fig. 5.b) and Fig. 5.c) cannot give the correct answer, since they have pixels on other grey levels too, and actually – they have no pixels on 0. Neither the last answer (which states that the histogram of the processed image should be the same as before processing) is not correct, as we could see (anyway this answer couldn't have been correct unless we would have a single straight line as processing function, of slope 1, passing through the origin – that is, unless the processing function could have been described by the equation $v=u$ – which is never the case in a grey scale clipping processing!).

The only histogram that matches our expectations is the one in Fig. 5.d), therefore, the answer indicating Fig. 5.d) as histogram of the processed image is the only correct answer.

Thus, the answers **a)**, **b)** and **d)** are incorrect, and **the answer c) is the correct answer of the quiz.**

7. If we examine comparatively the images in Fig. 6.a) (original) and in Fig. 6.b) (after processing), we can see that the contrast in the processed image (that is, the difference between the lightest grey – above the left flower, up near the image boundary – and the darkest grey – in the lower part of the image, between the two flowers, or, in the upper left corner of the image) is larger than the contrast in the original image; thus the processing should be one that can enhance the contrast, not decrease it. This means the first answer, **a)**, which suggests contrast compression, cannot be correct.

The second answer cannot be correct as well, since it states a negativation can be applied to yield from the image in Fig. 6.a), the image in Fig. 6.b); but negativation means that the lighter areas in Fig. 6.a) would become the darkest areas in Fig. 6.b), and vice-versa; this would mean, e.g., that the middle of the flowers (which are lighter than the petals in Fig. 6.a)) would become darker than the petals in Fig. 6.b), and as we see, this is not the case: the middle of the flowers still stays lighter than the petals in Fig. 6.b) as well as in Fig. 6.a). Thus, **b)** is not correct.

The third answer can indicate the correct processing: the histogram equalization, by its ability of spreading the grey levels widely across the whole brightness range from black to white, can have this contrast enhancement effect that is observed in Fig. 6.b) as compared to Fig. 6.a). Thus, the third answer, **c)**, is correct. So is also the last answer, **d)**, since histogram modification is just a "variant" of histogram equalization, leading to similar dynamic grey scale range modifications (i.e. possibly some contrast enhancement).

Thus, the answers **a)** and **b)** are wrong, **and the answers c) and d) are correct.**


Fig. 6.a


Fig. 6.b.

8. Examining the original grey scale image (Fig. 7.a)) and the result of its processing (Fig. 7.b)), one can notice that the processed image is a binary (pure black and white) image. There are several ways to obtain such an image, so we should examine the possible answers for this quiz, to see which of them indicate a processing that could really lead to the image in Fig. 7.b) and which of them do not.

Fig. 7.a

Fig. 7.b

The first answer indicates as possible operation, the contrast compression. This is definitely wrong, since the contrast of the second image (estimated as the difference between the lightest and the darkest brightness level in the image) is maximum (is actually exactly the difference from white to black), and probably higher than the contrast of the original image (or at least equal to the contrast of the original image). Thus, **a)** is an incorrect answer to this quiz.

The second answer indicates negativation as possible operation; obviously this is again incorrect, because as a result of the negativation, the brightest areas in the image from Fig. 7.a) should become the darkest in Fig. 7.b), and vice-versa, which means, the apple should appear lighter than the background – but this, as it can be seen, is not the case in Fig. 7.b) (the background becomes white – from the brightest in Fig. 7.a), and the apple becomes black – from the darkest in Fig. 7.a)). Thus, **b)** is also incorrect.

Some grey scale slicing operation could indeed lead to the image in Fig. 7.b) from the image in Fig. 7.a), for suitably chosen parameters, as we saw earlier in Quiz 6: if the grey levels inside the apple are under the low threshold of the grey scale slicing function (to be set to black after processing) and the background grey levels are above the high threshold of the grey scale slicing function (to be set to white after processing), then the result would be exactly the one in Fig. 7.b). (In an extreme case, imagine one sets the two thresholds equal – then, one gets simply a thresholding operation, leading to a binary image as the one in Fig. 7.b) as a particular case of grey scale slicing). Thus, the answer **c)** is correct.

The last answer, which states that one could apply the extraction of the most significant bit to obtain the image in Fig. 7.b), is also correct, since according to the definition of this algorithm, the result of processing any grey level u by this algorithm is the new brightness v=0 (black) if the most significant bit of u is 0, or v=255 (white) if the most significant bit of u is 1. Thus, considering an 8-bit representation of each brightness in the original image (which is the typical case for grey scale images), the examination of the most significant bit of the brightness means the examination of the value of u (the brightness) in comparison to 2^7=128: if the most significant bit of u is 0, it means that $u<2^7=128$. However, if the most significant bit of u is 1, then $u\geq128$ (equal only if only the most significant bit is 1, and all the others are 0). Thus, one can say that the extraction of the most significant bit is equivalent to a thresholding with a threshold of 128:

$$v = \begin{cases} 0, & \text{if} \quad u < 128 \\ 255, & \text{if} \quad u \geq 128 \end{cases},$$

where by u we denote the brightness in a certain spatial position in the original image, and by v – the resulting brightness (after processing) in the same spatial position in the output (processed) image.

Such an operation can lead to the image in Fig. 7.b), starting from the original image in Fig. 7.a), so, the answer **d)** is correct as well.

In conclusion, **the answers c) and d) of the quiz are correct,** whereas the answers **a)** and **b)** are false.

9. If the original image's histogram is the one in Fig. 8.a), and the resulting histogram after processing that image is the one in Fig. 8.b), then most likely, the processing applied is histogram equalization (which

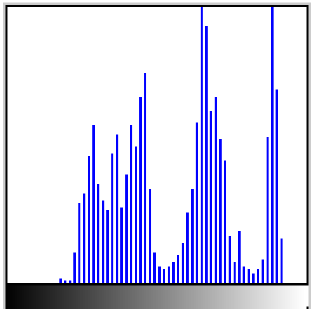


Fig. 8.a)

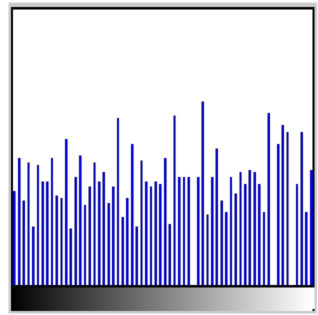


Fig. 8.b)

tends, ideally, to produce an image whose histogram is almost constant over the entire grey levels range). This corresponds to the answer **c)** of the quiz, thus answer **c)** is definitely correct.

Let us examine now the other possible answers: the first one says that the processing was a negativation of the original image; this is not possible, because in this case, the histogram of the resulting image should be a "mirrored" version of the original histogram (with the mirroring done versus the vertical axis). This is because the negativation is a transformation of each grey level u in the original image in a new grey level v given by the relation: $v = 255 - u$, if one considers the natural representation of the grey level images on 8 bits, thus, 0 – corresponds to black and 255 – corresponds to white (the maximum possible grey level in the image representation). Thus, whatever was darker becomes brighter and vice-versa, as a result of negativation, and basically, the number of pixels found on any grey level u in the original image "move" on the new grey level 255-u in the processed (negative) image – leading to the "mirroring" effect of the histogram. Since this is not the case for the histograms in Figures 8.a) and 8.b), it means the answer **a)** is not correct.

The second answer is as well not correct; as a result of binary thresholding, the resulting image is a binary one, thus it contains only grey levels, and we should have only two grey levels with pixels (two non-zero bins) in the resulting histogram. As we can see from Fig. 8.b), there are many grey levels on which we have pixels (obviously much more than 2), therefore the processing cannot be thresholding, thus the answer **b)** is incorrect.

We already saw that the third answer is correct; let's have a look on the last one. In the case of some grey scale slicing, one should have either an image containing just two grey levels (binary image) – in which case the histogram obviously would not look like the one in Fig. 8.b), since only two bins would have non-zero height, either an image which keeps the grey levels unchanged within some brightness range, and sets the others to a constant grey (most commonly – black or white). In the latter case however, one should find that portion of the original histogram in Fig. 8.a) over the plot in Fig. 8.b), and then, a constant grey level which would introduce a bin with a significant height (a large number of pixels on it), and as we can see, this is not the case of the histogram in Fig. 8.b). Therefore, the last answer, **d)**, is not correct: the histogram in Fig. 8.b) does not correspond to an image obtained from the original one (with the histogram shown in Fig. 8.a)) by grey scale slicing.

Thus, the answers **a)**, **b)** and **d)** are incorrect, and **the only correct answer of this quiz is c).**

10. To derive the effect of the grey scale transformation whose plot is shown in Fig. 9.c) on the original image shown in Fig. 9.a), we should examine it in respect to the original image's histogram (shown in Fig. 9.b)). Looking on the grey level histogram, one can see that most of the grey levels in the original image fall in the range [155; 213]. On the other hand, examining the grey scale transformation plotted in Fig. 9.c), we see it represents a grey scale slicing described by the equation:

$$v = \begin{cases} 0, & if \quad u < 160 \\ \dfrac{u-160}{50} \cdot 255, & if \quad u \in [160;210], \\ 255, & if \quad u > 210 \end{cases}$$

where by u we denote the brightness in a certain spatial position in the original image, and by v – the resulting brightness (after processing) in the same spatial position in the output (processed) image.

This means that most of the pixels in the original image from Fig. 9.a) fall inside the range [160;210] which is linearly expanded by the grey scale clipping function to the range [0;255] (that is, from black to white), thus leading to a contrast enhancement effect on the processed image (by significantly increasing the difference between the brightest and the darkest grey levels in the resulting image, as compared to the original image).

Let's recall that we are trying to see how will the resulting image look like, and we have as possible answers the following:

(1) The image will look the same as the original – this is not the case, since we just saw that our transform will produce the "expansion" of the grey level dynamics of our image in the range [0;255] from only [155;213]. Thus, the first answer, **a)**, is not correct.

(2) The image will look like the one in Fig. 9.d) – but in this figure, if we look on the bridge crossing the river, we see that the bridge appears darker than the river; whereas in Fig. 9.a) (original), the bridge was brighter than the river, which means, the image in Fig. 9.d) implies also a negativation of the original image, which is not achievable by the grey scale clipping function shown in Fig. 9.c). So, the second answer, **b)**, is incorrect as well.

(3) The image will look like the one in Fig. 9.e) – this is very likely; one can see that the bright grey levels (as e.g. the bridge, some buildings) become much brighter, the dark ones (as e.g. the river) become much darker, which means the contrast is enhanced, without any negativation. This matches the operation of the grey scale transformation from Fig. 9.c). Therefore, the third answer, **c), is correct.**

(4) The image will look like the one in Fig. 9.f) – this answer cannot be correct, since the image in Fig. 9.f) has even lower contrast than the original image (which means, we can expect an even narrower histogram than the one in Fig. 9.b)) – this does not match the grey scale processing function behavior derived above, which means the last answer is again not correct.

Thus, the answers **a)**, **b)** and **d)** are not correct, and **the only correct answer of this quiz is c).**

Fig. 9.a)

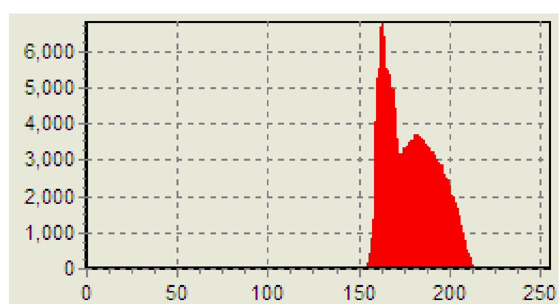


Fig. 9.b)

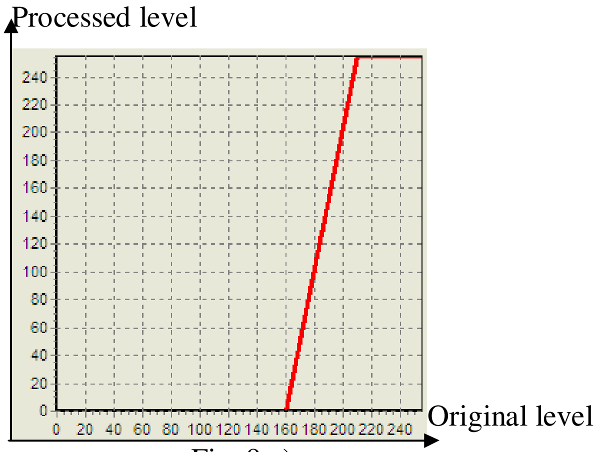


Fig. 9.c)


Fig. 9.d)


Fig. 9.e)


Fig. 9.f)

11. The original image in Fig. 10.a) looks like a grey level image affected by salt & pepper noise. The image in Fig. 10.b) looks like the result of an edge detection (what we see there is the map of the gradient magnitude) applied on the original image from Fig. 10.a), **but without any noise,** because otherwise, one should see large magnitude dots around the noise points (at least, they should be very visible on the uniform areas, like e.g. the background – probably the sky – around the roof).

This means that, if we want to obtain the image from Fig. 10.b) starting from the image in Fig. 10.a), we should apply a salt & pepper noise filter, and only afterwards apply an edge detector on the filtered image.

Let us examine what are the possible answers available and decide, based on the above observation, which of them can lead to the desired result and which cannot:

(1) The first answer suggests the usage of a Laplacian operator; but it is well known that such an operator is rather sensitive to noise, unless a Gaussian low pass filtering is prior applied on the image (leading to a LoG – Laplacian of Gaussian – edge detector). This means that the result of the Laplacian operator would be an edge magnitude image in which the grey level variations around the noise points would clearly appear – and this is not the case in Fig. 10.b). Thus, the first answer, **a)**, is not correct.

(2) The second answer suggests the usage of a high pass filter (that can be obtained by subtracting a low pass filtered version of the image from the original image). However, the low pass filter will attenuate the noise, and when subtracting it from the noisy image, we expect to have the noise points visible in the resulting image – and we see they are not; thus the second answer, **b)**, is not correct either.

(3) The third answer suggests applying a median filter, and then the edge detection of the median filtered image. As we know from the theory, the median filtering is highly efficient for the elimination of the salt & pepper noise. Thus, at the output of the median filter, we can expect to obtain the noise-free original grey scale image. If afterwards we apply an edge detection algorithm to this image, the edge magnitude map should look exactly as the one in Fig. 10.b), which means, **the third answer, c), is the correct answer.**

(4) Finally, the fourth answer suggests switching the order of the operations applied in the case of the third answer: start with an edge detection algorithm and afterwards apply a median filtering. Unfortunately, in this case, although we can expect the median filter to remove the isolated large edge magnitude points introduced in the edge magnitude map by the noise, we can also expect some of the real edge points (white or light grey thin lines surrounded by black background) to be deleted, therefore this procedure is not likely to lead to the desired image shown in Fig. 10.b). Furthermore, chances are for some noise introduced gradients to be kept in the processed image even after the median filtering. Therefore, this solution is not a good answer to the problem, and the answer **d)** should be considered incorrect.

Thus, the **only correct answer to the quiz is c);** the answers **a)**, **b)** and **d)** are not correct.

Fig. 10.a)

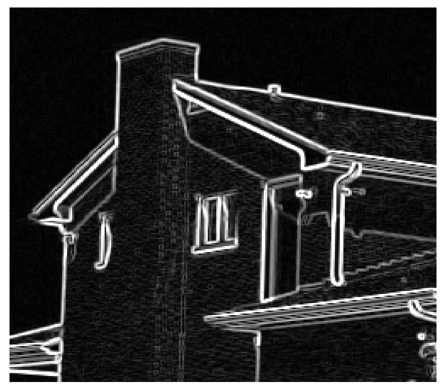


Fig. 10.b)

12. The given convolution mask:

$$\mathbf{H} = \frac{1}{8}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

corresponds to a low pass filter, since all its elements are positive and they sum to 1, which means, they perform a weighted averaging of the pixels' brightness in a local 3×3 pixels neighborhood; the result of this weighted averaging replaces the central pixel brightness. By performing the weighted averaging, one achieves a low pass filtering, since the values of the pixels' grey levels in uniform regions remain the same (i.e., averaging exactly the same values l gives **exactly the same value** l), whereas in 3×3 pixels neighborhoods that exhibit some grey levels variation, the variation is decreased (imagine e.g. the brightness of the central pixel is $l/10$ and the remaining 8 values around it are all l; in that case, as a result of convolving **H** with this neighborhood, one would get the new value for the central pixel as $(4\cdot l/10+l+l+l+l)/8=4/8\cdot(l/10+l)=1.1l/2=0.55l$, much higher than the original value and thus, much closer to the grey level of its neighbors). The reduction of the grey level transitions is equivalent to a decrease in the spatial frequencies in the image, thus, is equivalent to a spatial low pass filtering, and it should be visible on the processed image in the form of "blurring" the edges ("smoothing" the grey level transitions from one region to another). As mentioned before, the uniform grey level regions should remain unchanged (keep the grey level inside them.

Let us examine, based on this analysis, the images given below, as possible answers to the quiz. Actually we can see the examination is simple, since the images in Fig. 11.b), Fig. 11.d) and Fig. 11.e) look like some edge magnitude maps of the original image; none of these images satisfies the expected result of the convolution with **H** in respect to the preservation of the grey level in uniform areas, as e.g. the background around the roof (probably the sky): in the original image, the sky appears medium grey, but in Figs 11.b), d) and e) it appears black (just as the roof!), which means none of these images can be the result of the processing. The only image that matches our expectations is the one in Fig. 11.c): the grey level in the uniform areas is approximately preserved, and the edges (region boundaries) appear "smoothed" (the image appears blurred).

Therefore, the **only correct answer to the quiz is b);** the answers **a)**, **c)** and **d)** are not correct.

(Note that the images from Figs 11.b), d) and e) could be obtain by convolutions with gradient masks, which are equivalent with some spatial high pass filters of the image: Fig. 11.b) corresponds to an anisotropic gradient convolution mask; Fig. 11.d) detects only the vertical edges, therefore it can result through a convolution with an East-West or horizontal gradient mask; Fig. 11.e) detects only the horizontal edges, therefore it can result through a convolution with a North-South or vertical gradient mask).

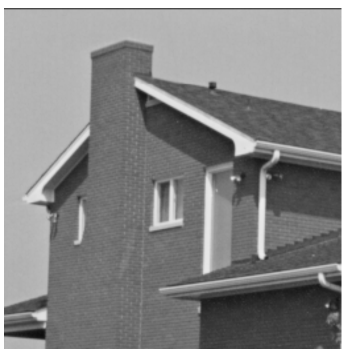


Fig. 11.a)

Fig. 11.b)

Fig. 11.c)

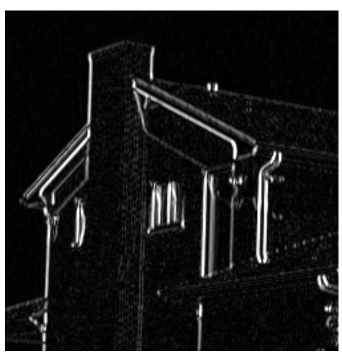


Fig. 11.d)

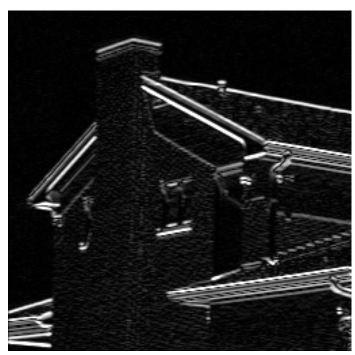


Fig. 11.e)

13. The processed image presented in Fig. 12.a) looks like the negative of some gradient magnitude map, since it displays only some contours; since not all of them are completely black, it means no thresholding was applied on the gradient magnitude map (i.e., the image is not binarized). Let us examine the four possible answers to see which of them describes a possible processing that could lead to this result.

The first answer, a), states that the image from Fig. 12.a) could be obtained from the image in Fig. 12.b) after an edge detection. Considering that the "edge detection" operation denotes a generic process, whose result can be the gradient magnitude map and possibly displayed in its negative form, this answer can be correct. The objects in Fig. 12.b) and the noise present on this image could lead exactly to the image in Fig.12.a), if e.g. one applies a convolution with some gradient masks (as Sobel, Prewitt, Roberts), takes just the magnitude of the resulting gradient and applies a negative to display it. Thus, **answer a) of the quiz is correct**.

The second answer, b), states that it is possible to obtain the image in Fig.12.a) by subtracting pixel by pixel the image in Fig.12.c) from the one in Fig.12.b); however this can't be correct, because in that case, one should have black pixels corresponding to the uniform (noise-free) regions, not white like we have in Fig.12.a). One can observe that the image in Fig.12.b) is a noisy version of the image in Fig. 12.c) (or alternatively, the image in Fig. 12.c) is a low pass filtered – noise filtered – version of the image in Fig. 12.b)). While subtracting a low pass filtered image from the original image can lead to edge detection, in that case, the result should be black in the uniform regions and should exhibit light grey levels corresponding to the edges and to the positions of high frequency details (noise, in our case). Therefore, since the image in Fig. 12.a) corresponds to a negative of this description, the answer b) cannot be correct.

The third answer, c), cannot be correct either: by segmenting the image in Fig. 12.b) using an amplitude thresholding, would be impossible to obtain always darker pixels around the noise boundaries and on the boundaries between regions and white or very bright values in all the uniform regions, as long the grey level variations between the regions and among the noise points are significant. E.g., it would be impossible to have in the resulting image white regions in the upper right corner and in the lower right corner, which differ significantly as grey level, and still to obtain black around the different noise pixels, which vary from black to white. Thus, the answer c) is wrong.

Finally, the last answer, d), can be just as correct as the answer a): the difference between the two answers refers to the image on which the edge detection is assumed to be applied. The image in Fig. 12.d) is just the negative of the image in Fig. 12.b), and since after the process of computing the luminance gradient we are only interested in the gradient magnitude, not its phase (sign), it results that the gradient magnitude (the magnitude of the spatial derivative of the brightness) is exactly the same when applied on an image and on its negative (recall that the negative of a grey scale image with the grey scale range $\{0,1,\ldots,L_{Max}\}$ is obtained by replacing each brightness l in the image with its complement in respect to L_{Max}, $l_{negative}=L_{Max}-l$; L_{Max} is a constant, thus it does not influence the derivative – the gradient; the derivative of $-l$ only differs as sign from the derivative of l). Thus, **answer d) is correct as well**.

Fig. 12.a)

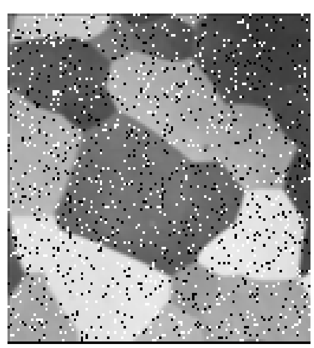


Fig. 12.b)

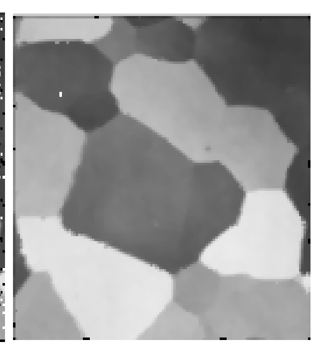


Fig. 12.c)

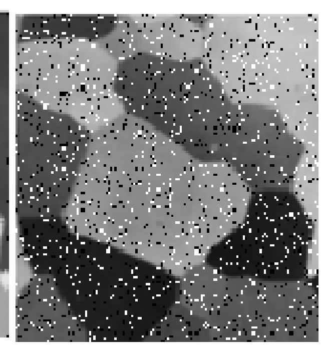


Fig. 12.d)

14. The resulting image, displayed in Fig. 13.b), looks like a "blurred" version of the original image displayed in Fig. 13.a). This blur effect is typically the result of a low pass filtering by spatial averaging in small spatial neighborhoods of the grey levels in the original image. The image in Fig. 13.b) is said to be obtained from the image in Fig. 13.a) by convolving it with a 3×3 mask. Recall that the convolution result in each spatial position is a weighted sum of the grey levels found in the 3×3 spatial neighborhood of that spatial position – having as weights, the elements in the convolution mask. Therefore, in order to have a spatial averaging, the convolution mask must contain only non-negative elements (otherwise the

convolution would lead to a differentiation operation, which is rather equivalent to a high pass, not low pass filtering). Let us examine, in respect to these considerations, the four convolution masks below.

The first mask, $\mathbf{H}_1$, has all the terms equal to 1/9. This means that the resulting grey level obtained by the convolution in any given spatial location will actually be the arithmetic mean of the 9 grey levels in the 3×3 neighborhood of this location. Thus, definitely, by convolving the image in Fig. 13.a) with the convolution mask $\mathbf{H}_1$, the effect will be of blurring the original image, since one can imagine this operation as a smoothing of the sharp edges (grey level transitions from a space location to another). Thus, **the first answer, a), is correct**.

The second mask, $\mathbf{H}_2$, is a symmetrical convolution mask, but as it can be seen, it contains only one positive term in the middle (with the value 1) and all the other eight terms are negative and equal to -1/8. Therefore its effect will be of differentiation of the grey levels in the 3×3 current neighborhood and cannot lead to a blurring effect of the original image. Instead, it will emphasize and display only the edges in the images, because due to the "weights" given by the convolution mask, the resulting grey level in any spatial location is computed as the difference between the brightness in the same spatial location and the arithmetic mean of its nearest eight surrounding neighbors. This means that in all the uniform areas, as e.g. around the apple, the resulting grey level should be zero (thus, it should appear as black) – which is not actually the case in Fig. 13.b) – where the background remains light grey, as in the original image. Therefore, the convolution mask used to obtain the image in Fig. 13.b) from the image in Fig. 13.a) cannot be $\mathbf{H}_2$ – and this means that the answer b) is not correct.

The next two convolution masks are more similar as effect to $\mathbf{H}_2$ than to $\mathbf{H}_1$. Thus, the convolution mask $\mathbf{H}_3$ is symmetrical and it contains only one positive term in the middle (with the value 1), four zero terms on the corners and four negative terms to the left, right, top and below the central position, with the value -1/4. Therefore its effect, the same as of $\mathbf{H}_2$, will be of differentiation of the grey levels in the 3×3 current neighborhood and cannot lead to a blurring effect of the original image. According to the "weights" given by this convolution mask, the resulting grey level in any spatial location is computed as the difference between the brightness in the same spatial location and the arithmetic mean of its four nearest neighbors – above, below, to its right and to its left. This means that in the uniform areas, including around the apple, the resulting grey level will be black (zero brightness) – which is not actually the case in Fig. 13.b) – where the background remains light grey, as in the original image. Therefore, the answer c) is not correct either. A similar situation appears for the last convolution mask, $\mathbf{H}_4$: considering the effect of $\mathbf{H}_4$ on the same uniform light grey area around the apple, one can see that from a pixel in that uniform region, the convolution result will again be zero or very close to zero (since the sum of the elements in $\mathbf{H}_4$ is zero, it follows that in any uniform 3×3 spatial region, having a constant grey level, the result of the convolution – i.e., the result of the weighted sum – will be zero as well). This means that the area around the apple should appear – as a result of convolving the image in Fig. 13.a) with $\mathbf{H}_4$ – black or very close to black. But as we can see in the image from Fig. 13.b), the convolution result in that region is approximately the same as the original region in Fig. 13.a) – a very light grey, therefore one cannot obtain the image in Fig. 13.b) by convolving the image in Fig. 13.a) with the mask $\mathbf{H}_4$. Therefore the answer d) cannot be correct.

The only correct answer to this quiz is a).


Fig. 13.a)


Fig. 13.b)

15. The given binary image is described by its quad-tree code, i.e. by the sequence:

g(g(wwbw)g(wwwg(bbbw))g(g(wbbb)www)g(wg(bbww)ww)).

Decoding this string, we get the following information about the content of our binary image:

- the first symbol in the code is ***g***, therefore the binary image is non-homogeneous, and we expect it to be described by four quadrants of equal size – denoted either by ***g***, or by ***w*** or ***b***;
- looking next in the string, we notice (using the brackets as well) four groups of symbols denoted by ***g***, therefore the four quadrants composing the binary image are as well non-homogeneous:
    1. the first quadrant contains:
        a. three white blocks (encoded by ***w***);
        b. one black block (encoded by ***b***)
    2. the second quadrant contains:
        a. three white blocks (encoded by ***w***);
        b. one non-homogeneous block, encoded by ***g***, which in turn contains three black blocks ***b*** and one white block ***w***
    3. the third quadrant contains:
        a. one non-homogeneous block, inside which we have one white block ***w*** and three black blocks ***b***;
        b. three white blocks, ***w***
    4. the fourth quadrant contains:
        a. three white blocks, ***w***;
        b. one non-homogeneous block, ***g***, which comprises from two black blocks ***b*** and two white blocks ***w***.

We do not have any knowledge about the size of the binary image, but we know that the smallest block is at least 1 pixel. Let us denote the size of the smallest homogeneous block by $s_b \times s_b$ pixels ($s_b \geq 1$), then we can count the number of black pixels (which are known to be object pixels) as follows:

- we have two smallest size black blocks of pixels in the fourth quadrant, thus $2s_b^2$ black pixels in the fourth quadrant;
- we have three smallest size black blocks of pixels in the third quadrant, thus $3s_b^2$ black pixels in the third quadrant;
- we have three smallest size black blocks of pixels in the second quadrant, thus $3s_b^2$ black pixels in the second quadrant;
- we have one larger size black block of pixels in the first quadrant, which means, a black block of size $2s_b \times 2s_b$ pixels, which totals $4s_b^2$ black pixels.

Overall, adding these amounts, we get $2s_b^2 + 3s_b^2 + 3s_b^2 + 4s_b^2 = 12\ s_b^2$ black pixels, that is, $12\ s_b^2$ object pixels, which gives the estimate of the area of the object region, in pixels.

Now let us examine which of the possible answers match this result. The first possible answer, **(a)**, states that the area of the object region should be 9, that is, the number of ***b*** symbols in the code; that is clearly not true, because the size of the block denoted by the first symbol ***b*** is obviously larger than the size of the subsequent ***b*** blocks. ***Thus, the answer (a) is wrong.***

The second answer, **(b)**, is again wrong for a similar reason: it considers the same size of each block of black pixels. Even more, the first part of the statement (i.e., that the minimum size of a block is 4 pixels) is wrong – the minimum size of a pixels' block can be 1, if not mentioned otherwise. ***Thus, the answer (b) is wrong as well.***

The third answer, (c), can't be correct either: the direction and the initial block for encoding are irrelevant in counting the black pixels – we are not interested in the position or compactness of the object region, just in the number of black pixels, which are considered object pixels.

Finally, the last answer can be correct, for s_b=1 pixel – which is a particular case of the counting result that we got. Thus, ***the only correct answer is (d)***.

16. Since the original image in Fig. 14.a) (shown for convenience below) is a binary image, whereas the result from Fig. 14.b) is a grey scale image, it means this result could not been obtain by a binary morphology operation, because binary morphology implies the interaction of the binary input image with a binary structuring element, and the result is still a binary image, not a grey scale one. **Therefore the first answer, (a), cannot be correct**.

The third answer, (c), cannot be correct either: any contrast enhancement operation is a grey scale transformation, which would modify each grey level in the original image to exactly the same grey level in the resulting image. Therefore from an image with only two grey levels, any grey scale transformation (which includes the contrast enhancement algorithms) would lead to an image with two, not more, grey levels – and this is not the case in Fig. 14.b). Anyway the image in Fig. 14.a) contains the white level and black level, therefore its contrast (proportional to the ratio between the brightest and the darkest grey levels in the image) is maximum, and cannot be enhanced more.

The only remaining candidates to be possibly correct are b) and d). Answer (d) states that the image in Fig. 14.b) represents the distance transform of the object region in Fig. 14.a), i.e. the black region in Fig. 14.a), plotted as brightness levels instead of values. Indeed, the distance transform should give an image in which the background values should be zeros (which means, as grey level – black); the maximum distance corresponds to the median axis of the object, which should be represented by the maximum brightness value, i.e., by white; as going from the median axis of the object towards its boundaries, the values in the distance transform decrease, since the distance of the object pixel to the nearest boundary decreases, therefore the grey level should become darker, as is the case in Fig. 14.b). ***According to these considerations, the answer (b) is correct:*** Fig. 14.b) looks like a grey level representation of the distance transform of the original image in Fig. 14.a), if the black pixels are object pixels and the white pixels are background in the binary image.

The last answer, (d), is actually the same as (b), since the distances of the object pixels (black pixels) to their nearest boundary defines the distance transform of the image. ***Therefore, the answer (d) is correct as well.***

Thus, the correct answers for this quiz are (b) and (d).

17. Looking on the original image in Fig. 15.a), one may notice that there are only two dominant grey levels in the image: a lighter grey corresponding to the objects, and a darker grey for the background where our objects lie. After processing this image, one gets the image in Fig. 15.b), where we have all the objects present, but they have different shades of grey, and the background is encoded in another shade of grey.

The question was, what type of processing has been applied on the image in Fig. 15.a) to get the image in Fig. 15.b); all four possible answers indicate some segmentation procedure, therefore the only question is what type of segmentation algorithm was used to render an image separated into background and objects labeled distinctly, by a different grey level each.

Let us examine each possible answer in turn.

The first answer, (a), states that an amplitude thresholding segmentation of the original image was used. However, this cannot be the case, because in the amplitude thresholding segmentation, all the pixels with the same grey level will be assigned to the same class, and therefore they will be encoded by the same grey level in the resulting segmented image – which means, the objects of lighter grey should have the same grey level in the segmented image, not different grey levels, because this segmentation procedure does not take into account the spatial position and spatial neighborhood of the pixels to be segmented. **Therefore, the first answer, (a), is not correct.**

The second answer cannot be correct either: the edge-based segmentation algorithm should lead to an image in which only the boundaries of the regions are visible, and the inner parts of the regions, as well as the background, are represented in black. It is obvious that the image in Fig. 15.b) does not display such an edge information, **therefore the second answer, (b), is not correct.**

The split-and-merge segmentation, mentioned by the third answer, is a possible procedure to get the image in Fig. 15.b), since in this process, one takes into account the spatial information along with the grey level homogeneity in generating the segmented image. In other words, a group of pixels is assigned the same label (in our case, the same grey level in the segmented image) only if they have approximately the same color and they are spatially connected. This would lead to the same label for the background pixels, but different labels for objects that are not connected spatially, as the objects in our original image. ***Therefore the third answer, (c), is correct.***

Finally, the last answer, according to which the segmentation was a region growing procedure, is correct as well: in the region growing segmentation, the pixels are again grouped based on the color homogeneity and spatial compactness, resulting in the same type of segmentation that we can see in Fig. 15.b). ***Therefore the fourth answer, (d), is correct as well.***

In other words, any of the two mentioned segmentation procedures – split&merge or region growing – may be used to give the image in Fig. 15.b) starting from the image in Fig. 15.a), thus the correct answers for this quiz are (c) and (d).

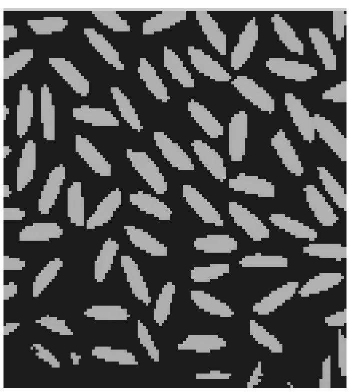


Fig. 15.a)

Fig. 15.b)

18. This quiz asks us detailed information about the content of a binary (black and white) image of size 8×8 pixels, which was quad-tree encoded and yielded the following code: ggwbbwgbgbwwbwwgbwwwgwwbw. We have no information about the numbering of the quadrants in the image, thus we cannot reconstruct graphically unambiguously the image, but we can find information about the distribution of the black pixels and white pixels, based on the known quad-tree code of the image. Let us see exactly what information reveals this code:

- The first symbol in the sequence is a ***g***. This means that our 8×8 pixels binary image is non-homogeneous, and it is formed by four quadrants (of 4×4 pixels each) which are described in the following symbols of the quad-tree code sequence.

- The second symbol in the sequence, after the first ***g*** that just indicated the root node (i.e. the non-homogeneous nature of our binary image) is again a ***g***; this means that the first quadrant (of size 4×4 pixels) is non-homogeneous either, it contains both black and white pixels; we should thus look for at least four symbols after the first ***g*** (if they are just ***w*** and ***b***), or for more if a ***g*** is found in the next four symbols. As we look on the next four symbols after the first ***g***, we notice that they are: a ***w***, a ***b***, another ***b*** and a ***w***. This means that we split the first quadrant, of 4×4 pixels, into four quadrants of 2×2 pixels, and each of these quadrants is homogeneous: two quadrants are homogeneous white, two are homogeneous black. In brief, the first 4×4 pixels quadrant in our 8×8 pixels image contains two white blocks of 2×2 pixels and two black blocks of 2×2 pixels.

- The next symbol, that is, the seventh in the sequence, should represent the second quadrant of 4×4 pixels of our original image (since we finished describing the first quadrant of 4×4 pixels of our image). This symbol is again a ***g***, showing that the second 4×4 pixels quadrant is still non-homogeneous and it is formed by four 2×2 pixels quadrants. Again we look for the sequence of symbols describing this second quadrant – after this seventh ***g*** symbol in the sequence – in the same way we did for the first quadrant. The 8th symbol in the sequence is ***b***, thus the first sub-quadrant of 2×2 pixels in the second quadrant is uniform black. The 9th symbol in the sequence is ***g***, therefore the second sub-quadrant of 2×2 pixels in the second quadrant is non-homogeneous, but is formed by black and white pixels, whose distribution is described by the following four symbols after this ***g***: ***b***, ***w***, ***w***, ***b*** => thus the symbols on the positions 10, 11, 12, 13 show us the presence of two black and two white pixels in the second sub-quadrant of 2×2 pixels of the second quadrant of our binary image. The next (third) sub-quadrant of 2×2 pixels of the second quadrant of our binary image is indicated by the 14th symbol in the sequence – a ***w***, which means it is uniformly white. The last (fourth) sub-quadrant of 2×2 pixels of the second quadrant of our binary image is indicated by the 15th symbol in the sequence – again a ***w***, which means it is as well uniformly white. To summarize our conclusions for the second quadrant of 4×4 pixels in our 8×8 pixels image: this contains two homogeneous 2×2 pixels white blocks, one homogeneous 2×2 pixels black block and one non-homogeneous 2×2 pixels block with two white and two black pixels.

- The following symbol in the sequence, that is, the 16th, indicates the third 4×4 pixels quadrant of our original image. We can see that it is a ***g*** again, which means this quadrant is non-homogeneous, thus it is formed by four 2×2 pixels sub-quadrants, described by the following symbols. The next four symbols after this ***g*** are just ***w*** and ***b***, which means the four 2×2 pixels sub-quadrants are homogeneous (either white or black): we have in the sequence, on the positions 17...20, the letters ***b***, ***w***, ***w***, ***w***, which indicates a uniform black 2×2 pixels block and three uniform white 2×2 pixels blocks in this quadrant.

- We can pass to examine the last 4×4 pixels quadrant of our original image, indicated by the 21st symbol: this is a ***g*** as well, indicating again a non-homogeneous quadrant, split in four 2×2 pixels sub-quadrants, described by the last four symbols in the sequence: ***w***, ***w***, ***b*** and ***w***, which indicates three uniform white 2×2 pixels blocks and one uniform black 2×2 pixels block in this last quadrant.

Since three of the possible quiz answers refer to counting the number of black pixels and the number of white pixels in our binary image, let's now count based on our examination these pixels in the 8×8 pixels binary image:

- we have two white blocks of 2×2 pixels and two black blocks of 2×2 pixels in the first 4×4 pixels quadrant, that is, 2·4 white pixels = 8 white pixels, and 2·4 black pixels = 8 black pixels in the first quadrant; **so, 8 white pixels and 8 black pixels in the first quadrant;**
- we have two 2×2 pixels white blocks, one 2×2 pixels black block and one non-homogeneous 2×2 pixels block with two white and two black pixels in the second 4×4 pixels quadrant, that is, 2·4 white pixels = 8 white pixels, 4 black pixels, and then again 2 white pixels and 2 black pixels in the second quadrant; **so, 8+2 = 10 white pixels and 4+2 = 6 black pixels in the second quadrant;**
- we have one black block of 2×2 pixels and three white blocks of 2×2 pixels in the third 4×4 pixels quadrant, that is, 4 black pixels, and 3·4 white pixels = 12 white pixels in the third quadrant; **so, 12 white pixels and 4 black pixels in the third quadrant;**
- we have again one black block of 2×2 pixels and three white blocks of 2×2 pixels in the fourth 4×4 pixels quadrant, that is, 4 black pixels, and 3·4 white pixels = 12 white pixels in the fourth quadrant; **so, 12 white pixels and 4 black pixels in the fourth quadrant.**

Therefore, the total count of white pixels in our binary image is: 8+10+12+12=**42 white pixels**, and the total count of black pixels in our binary image is: 8+6+4+4=**22 black pixels** (their sum is of course 64, the total number of pixels in our image).

Examining the possible answers in the quiz, we see that **the first one, (a), cannot be correct**, as it indicates a total number of 12 white pixels – when our counting shows we have 42 white pixels.

The second answer, (b), indicates 12 white pixels, 6 grey pixels and 7 black pixels; of course, this is as well wrong – even on a first glance, since we have a binary image, we cannot have grey pixels in it (just black and white pixels); the symbols ***g*** in the code do not indicate a grey pixel, but ***a non-homogeneous block of pixels***. **Thus, the answer (b) is wrong as well.**

The third answer, (c), states that our image contains 42 white pixels, and the remaining are black pixels; this answer is correct, since it matches our computations. ***So, the answer (c) is correct.***

The last answer, (d), does not refer to the total of black and white pixels, but to the distribution of the black pixels inside the image blocks; let us see if the distribution is indicated correctly.

The first part of the answer states that we have 5 blocks of 2×2 black pixels. Analyzing our decoding above, we see that we have: 2 blocks of 2×2 black pixels in the first quadrant + 1 block of 2×2 black pixels in the second quadrant + 1 block of 2×2 black pixels in the third quadrant + 1 block of 2×2 black pixels in the fourth quadrant = indeed, 5 blocks of 2×2 black pixels in our binary image.

The second part of the answer states that we have other 2 black pixels distributed in non-homogeneous 2×2 pixels blocks; this is again correct, since the remaining $22-5\cdot 4=22-20=2$ black pixels, are distributed in the non-homogeneous 2×2 pixels block found in the second quadrant. Therefore, since the two statements in the last answer are correct, it means ***answer (d) of the quiz is correct as well.***

Thus the correct answers for this quiz are (c) and (d).

19. We are asked in this exercise to select the skeleton of the black object from Fig. 16.a).

Fig. 16.a)

We should mention, prior to selecting our answer, that since we can choose from one of four drawings, and since the skeleton is unique (being the median axis of the object), only one of the four can be the correct one.

The possible answers are given by Fig. 16.b) – 16. e) below. Looking on a first glance, we can exclude Fig. 16.e), since it represents the boundary of the object, not its median axis. So, only the answers (a) – (c) remain as candidates; **the answer (d) of the quiz, which indicates Fig. 16.e) as the skeleton, cannot be correct.**

The first answer, (a), indicates the skeleton as being the one in Fig. 16. b); but this would be the skeleton of a triangle, since it corresponds to the median axis of a triangle. We do not see in this plot something we could expect in the skeleton of our object in Fig. 16.a), i.e., something that would correspond to the lower part – which resembles a rectangle – this part must introduce somewhere in the skeleton a horizontal line, since we have several horizontal pixels found at a maximum distance from the closest contour, and we do not see such a horizontal segment in Fig. 16.b). This explanation, together with the fact that the skeleton in Fig. 16.b) corresponds to a triangle and our figure is not a triangle, make us decide that **the answer (a) is wrong**.

The second answer indicates Fig. 16.c) as the skeleton; this answer may be correct, since we see the short horizontal line and those diagonal lines above and below, introduced by the four corners of our figure. Furthermore, the diagonal segments above and below have different slopes, corresponding to the difference in the corners of the figure above and below. Thus, the second answer may be correct.

Let us see also the third answer: it indicates Fig. 16. d), but this is the skeleton of a rectangle, and since our object is not a rectangle, it means **answer (c) cannot be correct either.**

Thus, finally, the only possible correct answer to this quiz is (b): the skeleton of the black object in Fig. 16.a) is the one in Fig. 16.c).

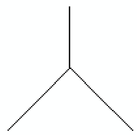


Fig. 16.b)

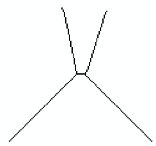


Fig. 16.c)

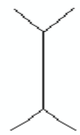


Fig. 16.d)

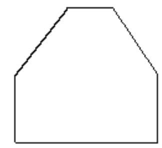


Fig. 16.e)

Comment: If we would like to make sure that Fig. 16.c) is indeed the skeleton of our object, we may build a small object of the same shape, compute its distance transform and then derive its skeleton, by the algorithm presented in the lecture, as follows:

1) Let us assume the smallest sized image that allows us to approximate the object in Fig. 16.a), where the black pixels will be represented by "1", and the background (white pixels) by "0":

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	1	1	0	0
0	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0

2) Let us compute the distance transform of this image; recall from the theory that the distance transform of a binary image **U**[M×N] is computed in an iterative process, starting from a $\mathbf{U}^{(0)}=\mathbf{U}$ (the original image), and computing at each iteration i, a matrix $\mathbf{U}^{(i)}$[M×N], whose values $u^{(i)}(m,n)$, m=0,1,...,M-1; n=0,1,...,N-1, are given by:

$$u^{(i)}(m,n)=\begin{cases}0, & \text{if } u(m,n)=0\\ u(m,n)+\min\left\{u^{(i-1)}(m-1,n),u^{(i-1)}(m+1,n),u^{(i-1)}(m,n-1),u^{(i-1)}(m,n+1)\right\} & \text{if } u(m,n)=1\end{cases},$$

where $u^{(i-1)}(m,n)$ is the value in the spatial location (m,n) in the distance transform at the iteration step $(i$-1).

The iterative process ends when there is no change from one iteration step to the next in the matrix $\mathbf{U}^{(i)}$[M×N], which indicates the convergence.

Applying this algorithm on the binary image above, of size 13×13 pixels, we get:

a. for the first iteration, $\mathbf{U}^{(1)}$[13×13]:

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	1	2	2	2	1	0	0	0	0
0	0	0	1	2	2	2	2	2	1	0	0	0
0	0	1	2	2	2	2	2	2	2	1	0	0
0	1	2	2	2	2	2	2	2	2	2	1	0
0	1	2	2	2	2	2	2	2	2	2	1	0
0	1	2	2	2	2	2	2	2	2	2	1	0
0	1	2	2	2	2	2	2	2	2	2	1	0
0	1	2	2	2	2	2	2	2	2	2	1	0
0	1	2	2	2	2	2	2	2	2	2	1	0
0	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0

b. for the second iteration, $\mathbf{U}^{(2)}[13\times13]$:

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	1	2	2	2	1	0	0	0	0
0	0	0	1	2	3	3	3	2	1	0	0	0
0	0	1	2	3	3	3	3	3	2	1	0	0
0	1	2	3	3	3	3	3	3	3	2	1	0
0	1	2	3	3	3	3	3	3	3	2	1	0
0	1	2	3	3	3	3	3	3	3	2	1	0
0	1	2	3	3	3	3	3	3	3	2	1	0
0	1	2	3	3	3	3	3	3	3	2	1	0
0	1	2	2	2	2	2	2	2	2	2	1	0
0	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0

c. for the third iteration, $\mathbf{U}^{(3)}[13\times13]$:

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	1	2	2	2	1	0	0	0	0
0	0	0	1	2	3	3	3	2	1	0	0	0
0	0	1	2	3	4	4	4	3	2	1	0	0
0	1	2	3	4	4	4	4	4	3	2	1	0
0	1	2	3	4	4	4	4	4	3	2	1	0
0	1	2	3	4	4	4	4	4	3	2	1	0
0	1	2	3	4	4	4	4	4	3	2	1	0
0	1	2	3	3	3	3	3	3	3	2	1	0
0	1	2	2	2	2	2	2	2	2	2	1	0
0	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0

d. for the fourth iteration, $\mathbf{U}^{(4)}[13\times13]$:

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	1	2	2	2	1	0	0	0	0
0	0	0	1	2	3	3	3	2	1	0	0	0
0	0	1	2	3	4	4	4	3	2	1	0	0
0	1	2	3	4	5	5	5	4	3	2	1	0
0	1	2	3	4	5	5	5	4	3	2	1	0
0	1	2	3	4	5	5	5	4	3	2	1	0
0	1	2	3	4	4	4	4	4	3	2	1	0
0	1	2	3	3	3	3	3	3	3	2	1	0
0	1	2	2	2	2	2	2	2	2	2	1	0
0	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0

e. for the fifth iteration, $\mathbf{U}^{(5)}[13\times13]$:

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	1	2	2	2	1	0	0	0	0
0	0	0	1	2	3	3	3	2	1	0	0	0
0	0	1	2	3	4	4	4	3	2	1	0	0
0	1	2	3	4	5	5	5	4	3	2	1	0
0	1	2	3	4	5	6	5	4	3	2	1	0
0	1	2	3	4	5	5	5	4	3	2	1	0
0	1	2	3	4	4	4	4	4	3	2	1	0
0	1	2	3	3	3	3	3	3	3	2	1	0
0	1	2	2	2	2	2	2	2	2	2	1	0
0	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0

f. for the sixth iteration, we will notice that $\mathbf{U}^{(6)}=\mathbf{U}^{(5)}$, therefore the convergence has been reached and $\mathbf{U}^{(5)}=\mathbf{V}$ is the distance transform of **U**.

3) Now using **V**, the distance transform of **U**, we may determine the skeleton as the set of points which are local maxima in cross neighborhoods, i.e., as the locations (k,l), k=0,1,...,M-1, l=0,1,...,N-1 which satisfy the condition:

$$v(k,l)\geq \max\{v(k-1,l),v(k+1,l),v(k,l-1),v(k,l+1)\},\quad v(k,l)=1.$$

If we mark the points that remain by a light grey in the distance transform above (all the other cells are kept white), we get the following matrix – in which obviously the skeleton looks most similar to the one in Fig. 16. c) (even if on a much smaller scale):

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	1	2	2	2	1	0	0	0	0
0	0	0	1	2	3	3	3	2	1	0	0	0
0	0	1	2	3	4	4	4	3	2	1	0	0
0	1	2	3	4	5	5	5	4	3	2	1	0
0	1	2	3	4	5	6	5	4	3	2	1	0
0	1	2	3	4	5	5	5	4	3	2	1	0
0	1	2	3	4	4	4	4	4	3	2	1	0
0	1	2	3	3	3	3	3	3	3	2	1	0
0	1	2	2	2	2	2	2	2	2	2	1	0
0	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0

20. Let us examine the matrix **V** below and then analyze each possible answer in turn, to decide which of them can be correct:

$$\mathbf{V} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 2 & 2 & 2 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 & 3 & 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 & 4 & 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 & 3 & 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 2 & 2 & 2 & 2 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The first answer, (a), states that **V** is the distance transform of a 9×9 square of object pixels, placed on a uniform background; this can be true, since **V** looks like a distance transform of a square object, but we must check for the size of the object: the size of the object should be (according to the definition of the distance transform – recalled in the previous exercise) equal to the number of non-zero values in its distance transform, since all the background pixels (zeros) remain 0, and the object pixels (ones) get values ≥1. According to this, counting the number of lines and columns of non-zero values in **V**, we get 7 lines and 7 columns, thus the size of the square is not 9×9, but 7×7. **Therefore the first answer, (a), is not correct.**

The second answer, (b), states that **V** is the distance transform of a 7×7 square of object pixels, placed on a uniform background. This is indeed correct, since the size of the object matches our counting, and the values in **V** correspond to the distance transform of the square – this can be easily checked if applying the algorithm presented in the previous exercise:

1. Consider the original binary image **U**, and initialize $\mathbf{U}^{(0)}$ as **U**:

$$\mathbf{U} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \mathbf{U}^{(0)}$$

2. Compute, at the iteration step i=1, $\mathbf{U}^{(1)}$ from $\mathbf{U}$ and $\mathbf{U}^{(i\text{-}1)} = \mathbf{U}^{(0)}$, according to the algorithm described in the previous exercise:

$$\mathbf{U}^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 2 & 2 & 2 & 2 & 1 & 0 \\ 0 & 1 & 2 & 2 & 2 & 2 & 2 & 1 & 0 \\ 0 & 1 & 2 & 2 & 2 & 2 & 2 & 1 & 0 \\ 0 & 1 & 2 & 2 & 2 & 2 & 2 & 1 & 0 \\ 0 & 1 & 2 & 2 & 2 & 2 & 2 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

3. Compute, at the iteration step i=2, $\mathbf{U}^{(2)}$ from $\mathbf{U}$ and $\mathbf{U}^{(i\text{-}1)} = \mathbf{U}^{(1)}$, according to the algorithm described in the previous exercise:

$$\mathbf{U}^{(2)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 2 & 2 & 2 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 & 3 & 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 & 3 & 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 & 3 & 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 2 & 2 & 2 & 2 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

4. Compute, at the iteration step i=3, $\mathbf{U}^{(3)}$ from $\mathbf{U}$ and $\mathbf{U}^{(i\text{-}1)} = \mathbf{U}^{(2)}$, according to the algorithm described in the previous exercise:

$$\mathbf{U}^{(3)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 2 & 2 & 2 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 & 3 & 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 & 4 & 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 & 3 & 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 2 & 2 & 2 & 2 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

This is also the last iteration step, since the maximum possible value has been reached in $\mathbf{U}^{(i)}$; if we make i=4 and we attempt to compute $\mathbf{U}^{(4)}$, we will see that nothing changes, $\mathbf{U}^{(4)}= \mathbf{U}^{(3)}$. Since we can notice that our **V** given above is identical to $\mathbf{U}^{(3)}$, it means **V** is indeed the distance transform of a 7×7 square of object pixels, placed on a uniform background, ***therefore the answer (b) is correct.***

The third answer, (c), which states that **V** is the amplitude spectrum of the Fourier transform of an image containing a 7×7 pixels black square, placed on a uniform background, cannot be correct either: one would expect the amplitude spectrum of a square to contain large horizontal and vertical frequency components, apart from the DC component, but this does not appear in **V**: if **V** would be an amplitude spectrum, then we see equal amplitude frequency components all around the DC component (in the center of **V**) which does not correspond to only horizontal and vertical edges. **Therefore this answer, (c), is not correct.**

The last answer, (d), cannot be correct either: the result of thresholding **any** image is a binary image, containing just two grey level values – which is obviously not the case of **V**, where we have 5 distinct values. Even more, since our original image is assumed to be binary (containing just a black square on a uniform background), the same would be the result of thresholding – a binary image, since thresholding implies the comparison of the grey level in any spatial position with some threshold and setting the output grey level in that spatial position to one of two values, depending on its position in respect to the threshold (below or above the threshold). **Therefore, answer (d) is not correct.**

Thus, the only correct answer for this quiz is (b).

21. The processed image in Fig. 17.c), resulting from the original grey level image in Fig. 17.a), looks either like a segmented image or like an image whose brightness was quantized on a small number of quantization levels. If the image was obtained through a segmentation algorithm, then since the six coins are represented by the same grey level (interpreted in the case of a segmentation like a label), it means all the coins were assigned to the same class, although they are not spatially connected. This type of segmentation, which does not take into account the spatial neighborhood of pixels and the spatial connectivity of the regions, must be from the class of grey level segmentation/grey level clustering. We mention that one of the most simple algorithms in this class is the histogram thresholding algorithm – which basically analyzes the linear grey level histogram of an image to find its local maxima and local minima (surrounding the local maxima). Then in this algorithm, the number of significant local maxima gives the number of classes of the segmentation task (the number of classes of grey levels, therefore the number of distinct colors that appear in the segmented image as pixels labels), and the assignment condition of a pixel to a class is given by the position of the pixels' brightness value in respect to some thresholds defined as the local minima surrounding the local histogram maxima. Each local maxima is found between two local minima which define a range (given by the two corresponding thresholds); if the pixels'

brightness value is within the range that surrounds a certain local maximum in the histogram, then it will be assigned the label corresponding to that category (class) and displayed accordingly in the segmented image.

Fig. 17.a)

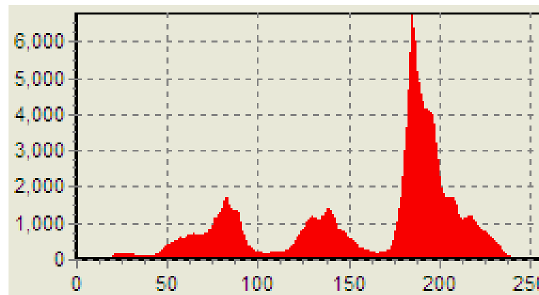


Fig. 17.b)

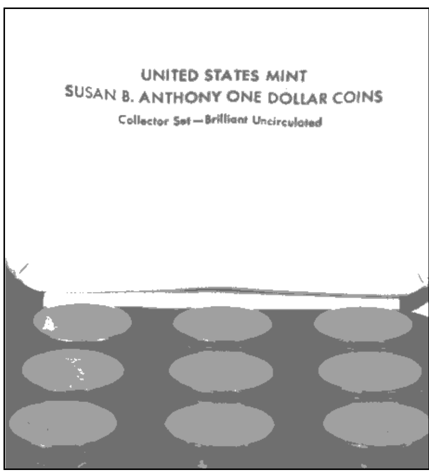


Fig. 17.c)

Now let us return to the quiz and examine the four possible answers, to see which of them may be correct. The first answer states that the image in Fig. 17.c) is the result of the edge detection applied on the image in Fig. 17.a), using compass operators; but no matter what operator is used, we know that the result of the edge detection is an image in which only the boundaries of the objects are present, on a uniform background – and all the uniform patches in the image become background. We clearly see that this is not the case in Fig. 17.c), **therefore answer (a) is not correct.**

The second answer states that the image in Fig. 17.c) is the result of the histogram-based segmentation of the image in Fig. 17.a) (using thresholds selected from the histogram); this matches our observation above, ***therefore answer (b) is correct.***

The third answer is a more general formulation that it is possible indeed to have led to the image in Fig. 17.c) from the image in Fig. 15.a): segmentation of the image in Fig. 17.a) in 3 classes, based on the brightness only. Indeed, we notice three grey levels in Fig. 17.c), i.e., white, a light grey and a darker grey – therefore, three labels assigned to the pixels in Fig. 17.a). Any brightness based segmentation algorithm (not only histogram thresholding) would indeed produce as segmentation result, an image as the one in Fig. 17.c). ***Therefore this answer, (c), is correct as well.***

The last answer, (d), says that the image in Fig. 17.c) may have been obtained from the one in Fig. 17.a) by a PCM encoding, using a 1 bit quantizer. Whereas PCM encoding implies mainly the usage of a uniform quantizer of the brightness, therefore being possible to produce an image that looks segmented, this answer is not correct because using a 1 bit quantizer means using a quantizer with only $2^1 = 2$ quantization levels; however our output image in Fig. 17.c) contains three grey levels, which could not be represented by a 1 bit quantizer. **Therefore answer (d) is not correct.**

Thus, the correct answers for this quiz are (b) and (c).

22. d)

23. d)