Exercise Computer graphics – (till October 19, 2006)

Excursus on channel coding and gray-level approximation

Exercise 10: A sender wants to transmit the following 32 bits

10110100 01011011 01010101 10110110

in four chunk-packets, each of which contains 8 bits. Both, sender and receiver use the same random number generation which produces the following bit-stream:

1010 0101 1001 0111

For data transmission, the **Random Linear Fountain Code** from the lecture is used.

<u>Proceeding:</u>

Sender Side
- Divide the message into chunks. Combine the chunks bit-wise according to the bit-merging vector which is taken from the output of the random number generator.
- Stop sending further chunks as soon as a sufficient number of XORed chunks with linear independent merging vectors have been sent.

Receiver Side
- Store the incoming chunks until a sufficient number is received. Sufficient means, that their merging vectors are linearly independent. The merging vectors are taken from the output of the random number generator as was done on the sender side.
- After having gathered enough data, calculate the modulo 2 inverse matrix of the matrix formed by the merging vectors.
- XOR the received chunks according to the inverse matrix.

Exercise Computer graphics

Bi-level display of gray-images

Exercise 11: (a) The following pattern is given for gray-level approximation:

16	5	6	7
15	4	1	8
14	3	2	9
13	12	11	10

A tiny image of 2x2 pixels is to be rendered using the above pattern. However, it is allowed to increase the resolution of the result by the factor of 4 for every side resulting in an output image of 8x8. Which pixels will be set in the output?

1	8
12	4

Exercise Computer graphics

Bi-level display of gray-images

Exercise 11: (b) This time the 8x4 image below is to be rendered using the same pattern as before. However, the image size of 8x4 should be preserved using the modulo-technique. The rule is that a pixel will be set if the value in the pattern is smaller or equal to the gray level which should be displayed.

Hint:

In the modulo-version of the black & white rendering approach, we first calculate the modulo of both the x- and the y-coordinate of a pixel. The result then addresses a grid in the pattern. We actually set the pixel only, if the value within the grid is smaller or equal to the gray-value to be rendered.

Example: We want to render gray value 11 of pixel (6, 3) [first cell is (0,0)]. The corresponding grid-cell in the pattern is (6 % 4, 3 % 4) = (2, 3). The grid-cell (2, 3) of the pattern is 11. Since 11 <= 11 we set the pixel.

Image:

10	4	12	12	10	6	3	7
10	5	12	12	6	15	9	11
8	9	12	12	12	4	10	15
9	10	1	0	1	5	11	1

Solution should look like this:

?	?	?	?	?	?	?	?
?	?	?	?	.	?	?	?
?	*	?	?	?	?	?	?
?	?	?	?	?	?	*	?

pixel set = *
pixel clear = .
fill out yourself = ?

Pattern:

16	5	6	7
15	4	1	8
14	3	2	9
13	12	11	10

Exercise Computer graphics

Bi-level display of gray-images

Exercise 11: (c) In exercise (a), a gray-value to be rendered was blown up by a factor according to the pattern. It is easy to imagine that this does not alter the mean gray value of the B/W image as compared to the original gray image.

Can you explain why this also holds true for exercise (b)? What would you object if someone said, that it is a matter of coincidence whether black or white is set because it mainly depends on the outcome of the modulo operation.