Prof. Dr.-Ing. A. Bruhn
Institute for Visualization and Interactive Systems
Computer Vision and Intelligent Systems Group
University of Stuttgart

Assignment C2 – Classroom Work

Problem 1 (Histogram Equalisation)

Let the following 8×5 image be given. Its grey values are quantised to 3 bits per pixel:

0	1	0	4	4	3	4	4
1	2	1	2	2	2	1	1
0	7	6	2	1	3	3	2
2	2	3	2	2	1	2	2
4	3	2	1	2	2	3	2

(a) Perform an affine grey scale transformation to the interval $[0, 7]$.

(b) Apply the discrete histogram equalisation to this image.

Assignment H2 – Homework

Problem 1 (Image Pyramids) (8 points)

Let the following 1-D signal be given:

$$f := (20, 16, 24, 6, 18, 8, 16, 8)^\top$$

(a) Calculate the Gaussian pyramid of the signal.

(b) Calculate the Laplacian pyramid of the signal.

(c) Reconstruct the initial signal from the Laplacian pyramid.

(d) The Laplacian pyramid requires more pixels than the original signal.
Where is the redundancy hidden?

Problem 2 (Discrete Fourier and Wavelet Transform) (6 points)

Let the following 1-D signal $\boldsymbol{f_1}$ and its shifted variant $\boldsymbol{f_2}$ be given:

$$\boldsymbol{f_1} := (0, 20, 8, 12)^\top , \qquad \boldsymbol{f_2} := (20, 8, 12, 0)^\top .$$

(a) Compute the Discrete Fourier Transform (DFT) and the power spectra of both signals.

(b) Compute the Discrete Wavelet Transform (DWT) of both signals, using Haar wavelets.

(c) What are your findings with respect to the results obtained in (a) and (b)?

Problem 3 (Huffman Coding) (6 points)

(a) Let the following Huffman code be given: E : 0, N : 10, S : 110, T : 111

- (i) Draw the corresponding Huffman tree.
- (ii) Decode the word "01011101011000'.

(b) Consider the word "Jalalabad".

- (i) How many bits are needed to encode this word in a naive way, assuming that we know which letters are used?
- (ii) Apply Huffman coding to this word and state the encoded version. How many bits are needed for the encoded version if we neglect the costs for transmission of the Huffman tree?

Problem 4 (Discrete Cosine Transform) (8 points)

Please download the archive `is20_ex02.tgz` from ILIAS into your own directory. You can unpack them with the command `tar xvzf is20_ex02.tgz`.

The programme `dct.c` computes the discrete cosine transform (DCT) of an input image. Apart from writing out the logarithmically transformed DCT spectrum, it also allows to modify the DCT coefficients in different ways before the backtransform. There are 6 menu options in total.

(a) Supplement the routines `DCT_2d` and `IDCT_2d` for the DCT and the inverse DCT with the missing code. The programme can then be compiled using the command

```
gcc -O2 -o dct dct.c -lm
```

(b) **Menu options 1 and 2:** Use the test image `boats.pgm` and compute the DCT for the whole image as well as separately for image blocks of size 8×8 (8x8 DCT). Compare the resulting spectra. What are your findings? What can you say about the runtime?
(*Since no DCT coefficients are modified the original images are obtained via the backtransform*).

(c) **Menu options 3 and 4:** These options are identical to the ones in (b) apart from the fact that approximately 90 % of all frequencies are removed before the backtransform. This can be seen from the percentage of DCT coefficients which are zero. Take a look at the obtained spectra. Have the high or low frequencies been removed? Compare the backtransformed images with respect to their visual quality. Does the DCT or the 8x8 DCT give better results?

(d) **Menu options 5 and 6:** These options allow you to compare two different quantisation strategies for the 8x8 DCT. While the first strategy treats all frequencies equal, the second one uses the weighting matrix from JPEG before quantisation. Compare once again the obtained spectra as well as the visual quality. Which strategy yields better results?

(This programming exercise is related to the basic idea of the JPEG compression standard.)

Problem 5 (Affine Rescaling) (4 points)

The program `pointtransTemplate.c` contains the subroutine `rescale`. It is supposed to perform an affine greyscale transformation such that the rescaled image has the greyscale range $[a, b]$.
As for all point transformations, this can be realised by specifying the entries of an integer 1-D mapping array `g[i]` that assigns to each possible input grey value `i` $\in [0, 255]$ its new output grey value `g[i]`.

(a) Supplement the missing code and compile your program with

```
gcc -O2 -o pointtransTemplate pointtransTemplate.c -lm
```

Note that the image `u[i][j]` is defined in the index range `i=1,...,nx` and `j=1,...,ny`.

(b) Test the routine with the image `machine.pgm`. In order to determine if the rescaling works correctly use at least one setting with $a \neq 0$.

Problem 6 (Gamma Correction) (4 points)

Gamma correction is an important nonlinear point operation. For an image with greyscale range $[0, 255]$ it has the structure

$$\phi(f(x,y)) := 255 \left(\frac{f(x,y)}{255} \right)^{1/\gamma} .$$

(a) Implement the method in the subroutine `gamma_correct` of program `pointtransTemplate.c`.

(b) Validate it with the image `asbest.pgm`. What values for γ give reasonable results ?

Submission

Please remember that up to three people from the same tutorial group can work and submit their results together. Note that in order to submit results as a group you have to create a submission group in ILIAS. A submission result will only be accepted for the submitting group as created in ILIAS!

The solutions have to be submitted in two parts:

1. The solutions to the theoretical problems 1, 2, and 3 have to be submitted in a single `pdf` file, which can be either digitally created or contain a scanned document.
2. The solutions to the practical problems 4, 5 and 6 have to be submitted in a single archive file:
   - Rename the main directory `Ex02` to `Ex02_<your_name>` and use the command

     ```
     tar czvf Ex02_<your_name>.tgz Ex02_<your_name>
     ```

     to pack the data. The directory that you pack and submit should contain the following files:
     - the source code for `dct.c` with implemented DCT and IDCT.
     - the DCT spectra as well as the backtransformed images,
     - the source code for `pointtransTemplate.c` with the subroutines for the problems 5–6
     - the corresponding test images with applied point operations
     - a text file README that contains answers to all questions of the problems 4–6 as well as information on all people working together for this assignment.
   - The file format can be a gzipped tar archive (`.tgz`) or a zip archive (`.zip`). No other file formats are accepted.
   - Please make sure that only your final version of the programmes and images are included.

Submit the two files via ILIAS.
(Remark: Please do **not** use the button "Upload Multiple Files as Zip-Archive").

Deadline for submission: Friday, June 19th, 23:59 am