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

Assignment C3 – Classroom Work

Problem 1 (Derivative Filters)

Let (f_i) be a discrete 1-D signal that has been sampled from a sufficiently smooth continuous function $f(x)$ with pixel distance h. Is there, in general, any lower bound of the order p of consistency, if a derivative of order d is approximated with n points?

Assignment H3 – Homework

Problem 1 (Linear Filters) (7 points)

(a) Write down the stencil of a two-dimensional separable binomial filter that is based on the one-dimensional binomial mask $\frac{1}{16}(1, 4, 6, 4, 1)$.

(b) Construct a corresponding highpass filter.

(c) Design a 2-D bandpass filter by means of another binomial filter based on $\frac{1}{4}(1, 2, 1)$.

(d) Compute the stencil that is equivalent to first applying $\frac{1}{4}\cdot$ | 1 | 2 | 1 | to an image and then applying $\frac{1}{18}\cdot$

-1	-1	-1
-1	17	-1
-1	-1	-1

to the result.

Problem 2 (Stencils) (3 points)

The following stencils are the result of two filters each. Bold font entries mark the central pixel. Which are the basic filters in x- and y-direction and what is their type (lowpass, bandpass, highpass, derivative, 2nd derivative...)?

(a) $\frac{1}{64} \cdot \frac{1}{h^2}\cdot$

1	6	15	20	15	6	1
-2	-12	-30	**-40**	-30	-12	-2
1	6	15	20	15	6	1

(b) $\frac{1}{2h} \cdot \frac{1}{h}\cdot$

1	-1
0	**0**
-1	1

(c) $\frac{1}{48}\cdot$

-1	0	2	0	-1
-1	0	**2**	0	-1
-1	0	2	0	-1

Problem 3 (First Order 1-D Derivative Filters) (6 points)

A sufficiently often continuously differentiable function $f(x)$ is sampled with pixel distance h, resulting in a discrete signal (f_i). The goal is to approximate $f'(x)$ in pixel i using the five values $f_{i-2}, f_{i-1}, f_i, f_{i+1}, f_{i+2}$.

(a) Derive the corresponding linear system of equations in matrix notation.

(b) Determine the order of consistency of the approximation

$$f'_i = \frac{f_{i-2} - 8f_{i-1} + 8f_{i+1} - f_{i+2}}{12h}$$

for the first derivative $f'(x)$ in pixel i.

Problem 4 (Mean and Median Filtering) (6 points)

Consider the image $\boldsymbol{f}$:

2	3	7	7	8
8	2	3	7	7
3	3	3	7	7
8	7	7	9	9
7	1	7	7	7

Assume reflecting boundary conditions and apply the following filters to $\boldsymbol{f}$:

(a) Mean filtering (i.e. using a normalised box filter) within a mask of size 3×3.

(b) Median filtering within a mask of size 3×3.

(c) Opening and closing within a mask of size 3×3.

What are your findings?

Problem 5: (Morphological Operations) (6 points)

Using a structuring element B and the elementary operations dilation and erosion, one can define a group of operations as follows:

- $A_B(f) := (f \oplus B) - f$
- $B_B(f) := f - (f \ominus B)$
- $C_B(f) := A_B(f) - B_B(f)$
- $D_B(f) := (f \oplus B) - (f \ominus B)$

(i) Apply those operations by hand to the 1-D signal $\boldsymbol{f} = (\dots, 0, 0, 0, 0, 1, 1, 1, 1, \dots)^\top$ which is continued by repeating zeros to the left and by ones to the right. Use a symmetric structuring element of size 3 ($m = 1$).

(ii) Do you know filters that give similar results as A_B, B_B, C_B, and D_B?

Problem 6 (Canny Edge Detector) (8 points)

Please download the required files from ILIAS into your own directory. You can unpack them with the command `tar xvzf is20_ex03.tgz`.

The file canny.c implements the Canny edge detector for grey value images.

(a) Supplement the missing code in the routine `getderivatives` which is supposed to compute the derivatives by using the Sobel operator. The program can then be compiled using the command

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

(b) Apply the program to the test image `objects.pgm` and try to find reasonable parameters.

(c) Now do the same for the test image `pruebab1.pgm`. Compare the parameter selection of (b) and (c). Try to explain your observations.

(d) To gain further experience you can test the program with other grey value images (e.g. `peppers.pgm`). Just try to create a nice, feasibly looking edge image.

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 – 5 have to be submitted in a single `pdf` file, which can be either digitally created or contain a scanned document.
2. The solution to the practical problem 6 has to be submitted in a single archive file:
   - Rename the main directory `Ex03` to `Ex03_<your_name>` and use the command
     `tar czvf Ex03_<your_name>.tgz Ex03_<your_name>`
     to pack the data. The directory that you pack and submit should contain the following files:
     - the source code for the file `canny.c`
     - your edge images of part (b) to (d).
     - a text file README that contains the parameters you have chosen in (b) and (c) as well as a short explanation why those parameters seem to be reasonable and 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: Thursday, July 9th, 23:59