University of Applied Sciences
Department of
Design, Computer Science, Media
Prof. Dr. D. Richter

Transilvania University Brasov
Faculty of
Mathematics and Informatics
Dr. I. Plajer

ISIP 2019
International School on Image Processing
July 15th - 26th, 2019

detlef.richter@HS-RM.de

3. Global Operations in Spatial Domain, Filters

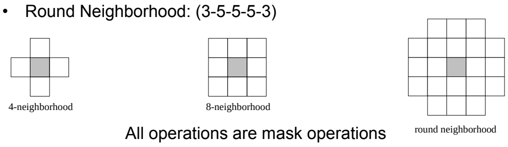
3. Global Operations in Spatial Domain, Filters

Following notation is used :

- Pixel position : p(x, y)
- Grey level at position p(x,y) : g(x, y)
- Modified grey level by an operation : g‘(x, y)
- Derivative of grey level at position p : g_x(x, y), g_y(x, y)
- Source pixel / Target pixel g_{old} / g_{new}

3. Global Operations in Spatial Domain, Filters

- 4-Neighborhood : (adjacent pixels by edges)
- 8-Neighborhood : (adjacent pixels by edges and corners)
- Round Neighborhood: (3-5-5-5-3)



All operations are mask operations
(most commonly used: 8-Neighborhood)

Note : In contrast to the mathematical coordinate system, the positive y-axis directs downwards. This fact originates from the time dependency of the pixel source in video.

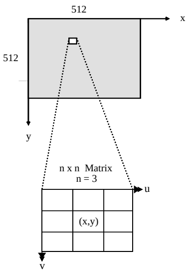
3.1 Global Operations in Spatial Domain, Low-pass Filters

H = h(x,y) =

x-1,y-1	x,y-1	x+1,y-1
x-1,y	x,y	x+1,y
x-1,y+1	x,y+1	x+1,y+1

- n*n filter
- n = 2*k+1, k=1,2,…
- p(x, y) = p(x-u, y-v)
- u and v ∈ {-(n-1)/2,..,0, ..+(n-1)/2}
- In the following :
  n = 3 unless stated otherwise

3.1.1 Global Operations in Spatial Domain, Low-pass Filters



- Mean Filter

$$g'(x,y) = \frac{1}{n^2} \sum_{u=-\frac{n-1}{2}}^{u=+\frac{n-1}{2}} \sum_{v=-\frac{n-1}{2}}^{v=+\frac{n-1}{2}} g(x-u, y-v)$$

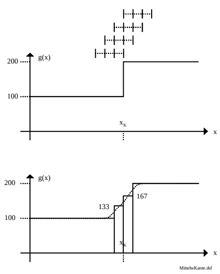
$$g'(x,y) = \frac{1}{n^2} \sum_{u=-\frac{n-1}{2}}^{u=+\frac{n-1}{2}} \sum_{v=-\frac{n-1}{2}}^{v=+\frac{n-1}{2}} g(x-u, y-v) * h(u,v)$$

$$H = h(u,v) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

3.1.1 Global Operations in Spatial Domain, Low-pass Filters

- Problem : margin area (upper and lower line / left and right column)
  - Use unchanged data, in most cases optimal procedure
- Result of operation :
  - No effect on areas with constant grey values
  - No effect on average grey value of image
  - Significant on image noise (desired)
  - Smoothing of edges (undesired)

3.1.1 Global Operations in Spatial Domain, Low-pass Filters



- Example :

  Filter mask 3 x 1 on ideal edge

  Weights of mask elements are 1

  Generation of new grey values

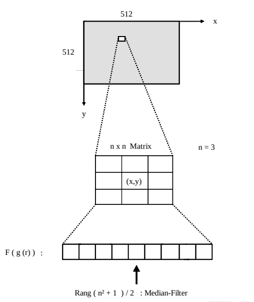
3.1.1 Global Operations in Spatial Domain, Low-pass Filters

- Empirical modification of mask weights (*edge sensitive mask*) :

$$H = \begin{vmatrix} 0,9 & 0,9 & 0,9 \\ 0,9 & 1,8 & 0,9 \\ 0,9 & 0,9 & 0,9 \end{vmatrix}$$

- Average value of mask elements : $h_{average}(u,v) = 1$
- Normalizing requires : $\Sigma\, h(u,v) = n^2$
  - Average grey level remains constant
  - Algorithm requires floating point values
  - Correct rounding and data conversion for output data necessary
  - $g'(x, y) \in \{0, \ldots, 255\}$

3.1.2 Global Operations in Spatial Domain, Low-pass Filters



- Rank order filter
  - Mean filter : $g' \notin \{G^*\}$
  - Rank ordering filter : $g' \in \{G^*\}$
  - G^* : Set of initial grey values
  - F(g(r)) increasing (or decreasing) sequence of grey values
  - $r = \{1, \ldots, n^2\}$; $n > 1$ and odd
  - Def. : $med\,\{F(g(r))\} = g((n^2+1)/2)$

    → Median filter
  - “median value of sorted sequence” instead of Mean value

3.1.3 Global Operations in Spatial Domain, Low-pass Filters

- Gauss filter
- Edge smoothing filter
- One dimensional Gauss function $\qquad h(u) = \frac{1}{\sigma\sqrt{2\pi}} * e^{-\frac{u^2}{2\sigma^2}}$

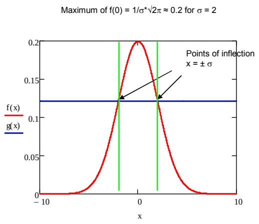
Maximum at h(u=0) $= \frac{1}{\sigma\sqrt{2\pi}}$

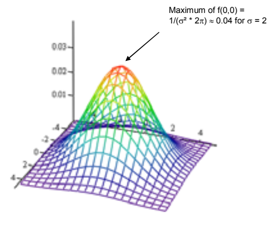
Minimum at h(u→±∞) $= 0$

Point of inflection at h(u=±σ) $= \frac{1}{\sigma\sqrt{2\pi}} * \frac{1}{\sqrt{e}}$

- Two dimensional Gauss function $\qquad h(u,v) = \frac{1}{\sigma^2 * 2\pi} * e^{-\frac{u^2+v^2}{2\sigma^2}}$

3.1.3 One-dimensional and two-dimensional Gauss function





3.1.3 Global Operations in Spatial Domain, Low-pass Filters

Let be σ = 1 and pre-factor = 4, one gets for n = 3 or n = 5 :

H = 1/16 *

oder

H = 1/20 *

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

Integer values

H = 1/21,6 *

0	0	0,5	0	0
0	1,5	2,4	1,5	0
0,5	2,4	4	2,4	0,5
0	1,5	2,4	1,5	0
0	0	0,5	0	0

Float values

Gauss00.DSF

σ defines the range of the filter
The pre-factor is arbitrary and is compensated by the normalizing factor

3.1.4 Global Operations in Spatial Domain, Low-pass Filters

- n-Binomial filter, n even
- Normalized binomial coefficients
  [e.g. n = 4 : $(a+b)^4 \rightarrow {}^1/_{16}$ * { 1 4 6 4 1 }]
- Multiplication column vector * row vector → Binomial mask elements

$$H = h(u, v) = \frac{1}{256} * \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

Question about Low-pass Filters

What are the image processing differences between the algorithms of

- Mean filter
- Median filter

- Gauss filter
- Binomial filter

?

3.1.5 Global Operations in Spatial Domain, Unsharp Masking

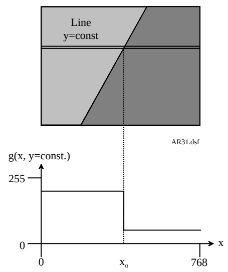
Original image: f(x,y)
Low-pass filtered image: $f_L(x,y)$
Unsharp mask: $g_{mask}(x,y) = f(x,y) – f_L(x,y) \approx 0$
Enhanced image: $f_{enhanced}(x,y) = f(x,y) + k * g_{mask}(x,y)$

If k = 1: unsharp masking
If k ≥ 1: highboost filtering

UnsharpMasking.xmcd

High-pass Filters

3.2 Global Operations in Spatial Domain, High-pass Filters



Edges : Significant change of grey values

Suppose to have a continuous step-function g(x)

How can we find changes of g(x) ?

Hochschule RheinMain - University of Applied Sciences

3.2 Global Operations in Spatial Domain, High-pass Filters

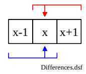
$$\frac{d}{dx}g(x) = \lim_{\Delta x \to 0} \frac{g(x+\Delta x) - g(x)}{\Delta x} = g_x(x)$$

In the discrete domain limit $\Delta x \to 0$ not realizable.
Which limit is realizable ?

$\Delta x \to ?$

We get

$g_x(x) = g(x+1) - g(x)$ forward differences
$g_x(x) = g(x) - g(x-1)$ backward differences



Problem : There exists no position between x and x+1 respectively between x-1 and x.

Prof. Dr. D. Richter
Department [Design>Computer Science>Media]

Hochschule RheinMain - University of Applied Sciences

3.2 Global Operations in Spatial Domain, High-pass Filters

Let be g(x,y) the central pixel of mask H

$$H = \begin{vmatrix} g(x-1,y-1) & g(x,y-1) & g(x+1,y-1) \\ g(x-1,y) & g(x,y) & g(x+1,y) \\ g(x-1,y+1) & g(x,y+1) & g(x+1,y+1) \end{vmatrix}$$

Prof. Dr. D. Richter
Department [Design>Computer Science>Media]

Hochschule RheinMain - University of Applied Sciences

3.2 Global Operations in Spatial Domain, High-pass Filters

The differential in x direction defines the edge in y direction

$$H_y = \begin{vmatrix} 0 & 0 & 0 \\ 0 & -1 & +1 \\ 0 & 0 & 0 \end{vmatrix} \quad \text{or} \quad \begin{vmatrix} 0 & 0 & 0 \\ -1 & +1 & 0 \\ 0 & 0 & 0 \end{vmatrix}$$

forward backward
differences

Prof. Dr. D. Richter
Department [Design>Computer Science>Media]

Hochschule RheinMain - University of Applied Sciences

3.2 Global Operations in Spatial Domain, High-pass Filters

Accordingly we get for edge in x direction :

$$H_x = \begin{vmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & +1 & 0 \end{vmatrix} \quad \text{or} \quad \begin{vmatrix} 0 & -1 & 0 \\ 0 & +1 & 0 \\ 0 & 0 & 0 \end{vmatrix}$$

Prof. Dr. D. Richter
Department [Design>Computer Science>Media]

Hochschule RheinMain - University of Applied Sciences

3.2 Global Operations in Spatial Domain, High-pass Filters

These masks are noise sensitive.

Edges are supposed to have certain lengths.

With distinct probability they continue into adjacent rows or columns.

Use $\Delta x \to 2$ instead of $\Delta x \to 1$ to find a pixel position for the result

We get an empirical extension of the filter mask.

Prof. Dr. D. Richter
Department [Design>Computer Science>Media]

3.2.1 Global Operations in Spatial Domain, High-pass Filters

We define as **Prewitt-Operator** :

$$g_x = H_y = \begin{vmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{vmatrix} \text{ und } g_y = H_x = \begin{vmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{vmatrix}$$

<u>Note:</u>

- H_y finds edges in y-direction, derivative in x direction,
- H_x finds edges in x-direction, derivative in y direction,

All lines / columns are treated with the same weights

3.2.2 Global Operations in Spatial Domain, High-pass Filters

We define as **Sobel-Operator** :

$$H_y = \begin{vmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{vmatrix} \text{ und } H_x = \begin{vmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{vmatrix}$$

The source line / column are weighted double

3.2.2 Global Operations in Spatial Domain, High-pass Filters

The grey value function is a two dimensional function.
We define the **Nabla-Operator** :

$$\nabla = \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix}$$

The Nabla-Operator is a vector operator, g(x, y) is a scalar function.
We define :

$$\nabla g(x, y) = \text{grad } g(x, y) = \begin{pmatrix} g_x(x, y) \\ g_y(x, y) \end{pmatrix}$$

grad g(x, y) is a vector.

3.2.2 Global Operations in Spatial Domain, High-pass Filters

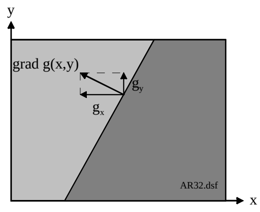
Magnitude of grad g(x,y) : $\left|\text{grad } g(x, y)\right| = \sqrt{g_x^2 + g_y^2}$

Here we lose the information of the direction.

Direction of grad g(x,y) : $\tan \alpha = \frac{g_y}{g_x}$

3.2.2 Global Operations in Spatial Domain, High-pass Filters

grad g(x,y) shows into direction of largest change of luminance.
Representation in mathematical coordinate system :



Mind the use of a mathematical coordinate system

3.2.2 Global Operations in Spatial Domain, High-pass Filters

Visualization as binary image (target image) :

$$g'(x, y) = \begin{cases} 0, & \text{if } \left|\text{grad } g(x, y)\right| \le T \\ 255, & \text{otherwise} \end{cases}$$

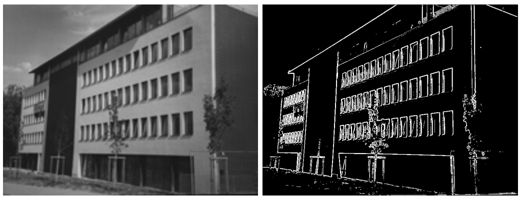
Visualization in grey level image (source image) :

$$g'(x, y) = \begin{cases} g(x, y), & \text{if } \left|\text{grad } g(x, y)\right| \le T \\ \text{const.} & \text{otherwise} \end{cases}$$

Visualization of vector direction (target image) :

$$g'(x, y) = \begin{cases} 0, & \text{if } \left|\text{grad } g(x, y)\right| \le T \\ \text{angle coded as grey value,} & \text{otherwise} \end{cases}$$

3.2.2 Global Operations in Spatial Domain, High-pass Filters



Example of direction-independent Sobel filtering as binary image.
Threshold = 150

HochpassFilter-Sobel.xmcd

3.2.2 Global Operations in Spatial Domain, High-pass Filters

Example of direction-independent Sobel filtering as grey value image

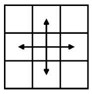
3.2.2 Global Operations in Spatial Domain, High-pass Filters

Example of direction-dependent Sobel filtering

atan2()-function atan()-function

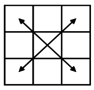
HochpassFilter-Sobel-RA

3.2.3 Global Operations in Spatial Domain, High-pass Filters



Sobelfilter

Roberts.dsf



Robertsfilter

Roberts.dsf

- Roberts filter
- The basis of gradient components are the diagonals

$g_+ = g(x+1,y-1) - g(x-1,y+1)$
$g_- = g(x+1,y+1) – g(x-1,y-1)$

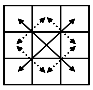
$|\text{grad } g(x,y)| = \text{sqrt}(g_+^2 + g_-^2)$

Difference to Sobel filter ?

3.2.3 Global Operations in Spatial Domain, High-pass Filters

$$H- = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & +1 \\ 0 & +1 & +2 \end{bmatrix} \qquad H+ = \begin{bmatrix} 0 & +1 & +2 \\ -1 & 0 & +1 \\ -2 & -1 & 0 \end{bmatrix}$$

Roberts4.dsf



Robertsfilter

Roberts.dsf

Filter masks, when using main and adjacent diagonals :

$g_+ = 2*[g(x+1,y-1)–g(x-1,y+1)] + g(x,y-1) – g(x-1,y) + g(x+1,y) –g(x,y+1)$

$g_- = 2*[g(x+1,y+1)–g(x-1,y-1)] + g(x,y+1) – g(x-1,y) + g(x+1,y) –g(x,y-1)$

$|\text{grad } g(x, y)| = \text{sqrt}(g_+^2 + g_-^2)$

3.2 Global Operations in Spatial Domain, High-pass Filters

The filters need interactions by thresholding, therefore we use :

Global adaptive threshold

- Calculate entire gradient image : $g'(x,y) = \text{grad } g(x,y)$
- Find maximum of all gradients : $\text{grad}_{max,glob} = \max\{ \text{grad } g(x,y) \}$
- Define adaptive threshold : $T_{adaptiv,glob} = \text{Factor}_{glob} * \text{grad}_{max,glob}$
- Empirical : approx. $0.2 \le \text{Factor}_{glob} \le$ approx. 0.7

3.2 Global Operations in Spatial Domain, High-pass Filters

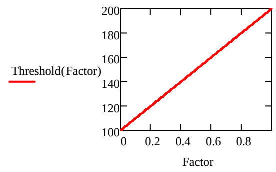
Alternative option : Local adaptive threshold

- Calculate entire gradient image : g‘(x,y) = grad g(x,y)
- Find within n x n – mask
  maximum of gradients : $grad_{max,local}$ = max{ grad g(x,y) }$_{nXn}$
  minimum of gradients : $grad_{min,local}$ = min{ grad g(x,y) }$_{nXn}$
- Define adaptive local threshold
  $T_{adaptiv,local} = Factor_{local} * grad_{max,local} + (1 - Factor_{local}) * grad_{min,local}$
- Empirical : approx. 0,5 ≤ $Factor_{local}$ ≤ approx. 0,7
- Empirical : approx. 11 ≤ n ≤ 25
  according to image content and structure

3.2 Global Operations in Spatial Domain, High-pass Filters

Example for $grad_{max,local}$ = 200 and $grad_{min,local}$ = 100

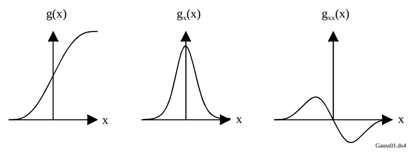
$T_{adaptiv,local} = Factor_{local} * grad_{max,local} + (1 - Factor_{local}) * grad_{min,local}$



3.2 Global Operations in Spatial Domain, High-pass Filters

- This procedure may produce very small threshold values in cases when
  $grad_{max,local} \approx grad_{min,local}$
- Define a minimal threshold, which may not be surpassed.

3.2.4 Global Operations in Spatial Domain, High-pass Filters



Laplace-Filter

The edge is defined by Zero-crossing of 2nd derivative of grey level function.

- Calculate 2nd derivative by applying Nabla operator onto 1st derivative of grey level function.
- Vector operator applied on vector function results in scalar value.

3.2.4 Global Operations in Spatial Domain, High-pass Filters

$$\Delta g(x,y) = \nabla(\nabla g(x,y)) = \nabla \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix} g(x,y) =$$

$$\left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right) \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix} g(x,y) = \frac{\partial^2 g(x,y)}{\partial x^2} + \frac{\partial^2 g(x,y)}{\partial y^2} =$$

$$\frac{\partial}{\partial x}(g(x+1,y) - g(x,y)) + \frac{\partial}{\partial y}(g(x,y+1) - g(x,y)) =$$

$$g(x+1,y) - g(x,y) - g(x,y) + g(x-1,y) + g(x,y+1) - g(x,y) - g(x,y) + g(x,y-1)$$

Note : *backward* und *forward differences* are applied alternately !

3.2.4 Global Operations in Spatial Domain, High-pass Filters

$$H_{math} = const * \begin{vmatrix} 0 & -1 & 0 \\ -1 & +4 & -1 \\ 0 & -1 & 0 \end{vmatrix} \text{ oder } H_{emp1} = const * \begin{vmatrix} -1 & -1 & -1 \\ -1 & +8 & -1 \\ -1 & -1 & -1 \end{vmatrix} \text{ oder } H_{emp2} = const * \begin{vmatrix} -1 & -2 & -1 \\ -2 & +12 & -2 \\ -1 & -2 & -1 \end{vmatrix}$$

- H_{math} is deduced from Math, H_{emp1} und H_{emp2} are empirical deduced masks.
- Note : Σ h(u,v) = 0, i.e. image areas with constant grey level result in g‘ = 0.
- The masks produce symmetrical positive and negative grey values, negative grey values cannot be visualized.
- For visualization use according Look-up-Tables
- Find Zero-crossings (or crossing at 127.5 if scaled)

3.2.4 Global Operations in Spatial Domain, High-pass Filters

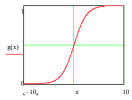
Zero crossing detection :

if ((Ai ≥ Zero point +T AND Bi ≤ Zero point -T) OR-ed for i = 1,2,3,4) = 1
OR
(Ai ≤ Zero point -T AND Bi ≥ Zero point +T)) OR-ed for i = 1,2,3,4) = 1
then g' = 255
else g' = 0

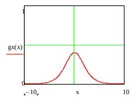
A1	A2	A3
B4	g'	A4
B3	B2	B1

Zero point = 0 or 127,5 if scaled;
Threshold T : {0 ≤ T < ≈10}

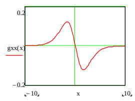
3.2.4 Global Operations in Spatial Domain, Enhancement by Laplace



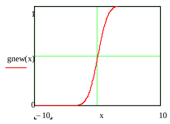
greylevel g(x)



1st derivative gx(x)



2nd derivative gxx(x)



gnew(x) = g(x)-gxx(x)

3.2.4 Global Operations in Spatial Domain, Enhancement by Laplace

Original Image

Enhanced Image by Laplace

3.2.5 Global Operations in Spatial Domain, High-pass Filters

- Laplace filter is extremely sensitive for noise.
- Apply low-pass filter before application of Laplace filter
- Ideal low-pass filter : Gauss filter
- Result : Laplacian of Gaussian (LoG)

3.2.5 Global Operations in Spatial Domain, LoG Filter

- Gauss-Filter : G(u, v, σ) (two-dimensional)

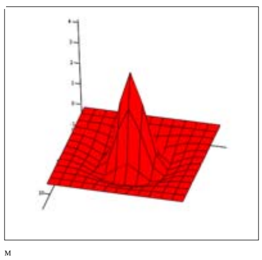
  g'(x,y) = Δ {G(u, v, σ) * g(x,y) } = { Δ G(u, v, σ) } * g(x,y)
  (stated without mathematical proof)

$$\Delta G(u,v,\sigma) = h(u,v) = \frac{1}{2*\pi*\sigma^4} * \left(+\frac{u^2+v^2}{\sigma^2} - 2\right) * e^{\frac{-(u^2+v^2)}{2\sigma^2}}$$

(Mexican hat shaped function)

g'(x,y) = h(u,v) * g(x,y)

3.2.5 Global Operations in Spatial Domain, LoG Filter



M

$M_{i,j}$ = h(u,v), u = 0 … 12, v = 0 … 12

using prefactor -2 to get center element $M_{6,6}$ = 4

$$M_{i,j} := -2 \cdot \left[\frac{\left[(i-6)^2 + (j-6)^2\right]}{\sigma^2} - 2 \right] \cdot e^{\frac{-\left[(i-6)^2+(j-6)^2\right]}{2\cdot\sigma^2}}$$

3.2.5 Global Operations in Spatial Domain, LoG Filter

M=

	0	1	2	3	4	5	6	7
0	-0	-0	-0	-0	-0	-0.01	-0.01	-0.01
1	-0	-0	-0	-0.01	-0.03	-0.06	-0.07	-0.06
2	-0	-0	-0.02	-0.07	-0.16	-0.25	-0.29	-0.25
3	-0	-0.01	-0.07	-0.22	-0.42	-0.53	-0.54	-0.53
4	-0	-0.03	-0.16	-0.42	-0.53	-0.15	0.18	-0.15
5	-0.01	-0.06	-0.25	-0.53	-0.15	1.42	2.49	1.42
6	-0.01	-0.07	-0.29	-0.54	0.18	2.49	4	2.49
7	-0.01	-0.06	-0.25	-0.53	-0.15	1.42	2.49	...

- 2D-example with σ=1,5
- Range : 6 elements resp. 13 x 13 mask
- Cut-out of mask shown, missing elements are symmetrical to h(6,6)
- Element values rounded to 2 decimals
- Mask may be cut-off, if absolute values of h(u,v) / h(6,6) are less then 0.1 ⇒ 9 x 9 mask

3.2.5 Global Operations in Spatial Domain, LoG Filter

For r → ∞ we obtain h(u,v) → 0

Cut-off for filter range assumed as h(u,v) / h(0,0) > -0.1

Range of filter : h(u,v) / h(0,0) =

σ = 1 : h(u,v) / h(0,0) > -0.1 for u ≈ 2,5; 7 x 7 mask
σ = 1,5 h(u,v) / h(0,0) > -0.1 for u ≈ 3,8 9 x 9 mask
σ = 2 : h(u,v) / h(0,0) > -0.1 for u ≈ 5,0; 11 x 11 mask
σ = 3 : h(u,v) / h(0,0) > -0.1 for u ≈ 7,5; 17 x 17 mask
σ = 4 : h(u,v) / h(0,0) > -0.1 for u ≈ 10,0; 21 x 21 mask

3.2.5 Global Operations in Spatial Domain, LoG Filter

cut-off

Let σ = 2 and get for h(u,v)/h(0,0) the following linear mask :

-0,039

-0,093	-0,135	-0,041	0,303	0,772	1.0	0,772	0,303	-0,041	-0,135	-0,093

-0,039

Sum of mask elements / 11 elements = 0.237

Subtract 0.237 from all mask elements to achieve Σ of elements = 0 :

-0,330	-0,372	-0,278	0,066	0,535	0,763	0,535	0,066	-0,278	-0,372	-0,330

Apply mask elements on grey level step from g = 100 to g = 200 :

0	-33	-70	-98	-91	-38	+38	+91	+98	+70	+33	0

Zero-crossing defines position of edge

LoG01 engl.dsf

3.2.5 Global Operations in Spatial Domain, LoG Filter

- Scaling to G ∈ { 0, …, 255 }
- g'(g) = m * g +n

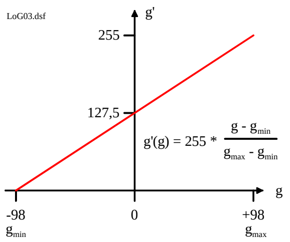
with boundary condition $g(g_{min} = -98) = 0$ und $g(g_{max} = +98) = 255$

$$g'(g) = \frac{255}{g_{max} - g_{min}} * (g - g_{min})$$

- In general in images g_{max} and g_{min} are not symmetrical to zero

3.2.5 Global Operations in Spatial Domain, LoG Filter

LoG03.dsf



⇒ g'(g=-98) = 0 g'(g=0) = 127,5 g'(g=+98) = 255

- Zero-crossing is shifted in this case to 127,5 !

3.2.5 Global Operations in Spatial Domain, LoG Filter

- Define appropriate Mask size
- Calculate the sum of all mask elements, which is ≠ 0
- Divide sum by number of all mask elements n x n
- Subtract result from all mask elements
- Check sum of all mask elements to be 0
- Apply mask on image
- Calculate crossings of average grey value of laplace filtered image.

3.2.5 Global Operations in Spatial Domain, LoG Filter, Results

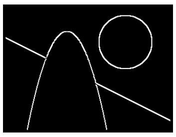


Image:
Synthetic image ParabolaTestImage04 without noise

Average grey value of Laplace filtered image 134.8

Target Image threshold T = 15

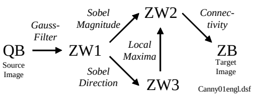
3.2.6 High Pass Filters, Canny Filter

Canny filter is a multi-step operator and generates one-pixel-wide edges

- Gauss Filter (σ = 1 or σ = 1.4)
- Direction dependent and independent Sobel Filter
- Thinning of edges
- Closing of gaps

3.2.6 High Pass Filters, Canny Filter

(www.pages.drexel.edu/~weg22/can_tut.html)



Canny filter is a multi-step operator and generates one-pixel-wide edges

5 x 5 Gauss filter

0.5	1.1	1.4	1.1	0.5
1.1	2.4	3.1	2.4	1.1
1.4	3.1	4.0	3.1	1.4
1.1	2.4	3.1	2.4	1.1
0.5	1.1	1.4	1.1	0.5

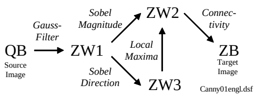
3 x 3 Gauss filter

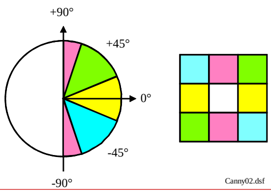
1	2	1
2	4	2
1	2	1

- Apply *low-pass filter on source image (QB)* for noise reduction, e.g. 5*5-Gauss filter with σ = 1 or σ = 1.4, (Action: QB → ZW1)

3.2.6 High Pass Filters, Canny Filter

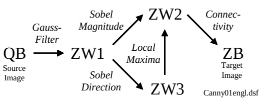
(www.pages.drexel.edu/~weg22/can_tut.html)

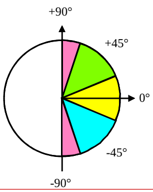




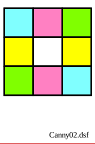
- Apply Sobel *highpass filter on ZW1* for edge extraction.
- Store direction independent magnitude (Action: ZW1 → ZW2)
- Scale ZW2 to values 0, … 255
- Store 4-direction dependent value in ZW3 (e.g. 0, 45, 90, 135 or 0, 1, 2, 3) for the discrete direction (Action: ZW1 → ZW3).

3.2.6 High Pass Filters, Canny Filter





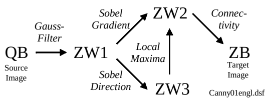
Mask in ZW2



Thinning of edges :

- Scan ZW2 :
- Set actual magnitude value (white element) in ZW2 to 0, if it is **not** the maximum within the direction of the respective mask defined in ZW3 (e.g. white and yellow elements, white and green elements etc).
- Action : (ZW2 → ZW2)

3.2.6 Global Operations in Spatial Domain, Canny Filter

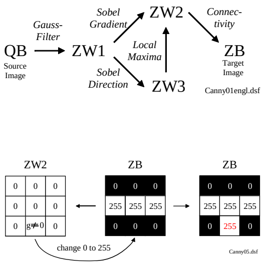


Closing of gaps (Connectivity)

- Define two thresholds $0 < T_1 < T_2 < 255$
- Scan ZW2 :
  - if $g > T_2$: set ZW2 = 0 and ZB = 255
  - if $g < T_1$: set ZW2 = 0
  - if $T_1 < g < T_2$: leave ZW2 = g unchanged

Action : (ZW2 → ZW2 and ZW2 → ZB)

3.2.6 Global Operations in Spatial Domain, Canny Filter



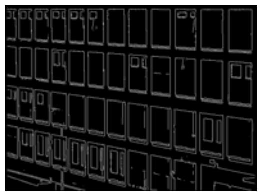
Completion of target image (ZB):

- Scan ZB :
  if in ZB the
  actual pixel = 255 AND
  within the mask in ZW2
  any value g ≠ 0, i.e. $T_1 < g < T_2$,
  → set the corresponding pixel
  in ZB to g = 255.

Action: (ZB → ZB)

3.2.6 Global Operations in Spatial Domain, Canny Filter


Bild


NeuBild

3.3 High Pass Filters, Harris Filter / Corner Filter

- Corners: significant structures in an image
- Applications:
  - Tracking of structures in image sequences
  - Search for corresponding points in stereo vision
  - Reference points for automatic measurements
  - Calibration of cameras
- Widely independent of illumination
- Requirements:
  - Differing between significant and non significant points
  - Independency of noise
  - Real time performance for video-tracking

3.3 High Pass Filters, Harris Filter / Corner Filter (1)

- Edges: significant magnitude of gradients in only one direction
- Corners: significant magnitude of gradients in more than one direction
- Calculate the gradients
  by using Prewitt or Sobel: $\mathrm{Ix}(x,y) = \frac{\partial}{\partial x}\mathrm{I}(x,y) \qquad \mathrm{Iy}(x,y) = \frac{\partial}{\partial y}\mathrm{I}(x,y)$
- Calculate matrices:. $\mathrm{A}(x,y) = \mathrm{Ix}^2 \quad \mathrm{B}(x,y) = \mathrm{Iy}^2 \quad \mathrm{C}(x,y) = \mathrm{Ix} * \mathrm{Iy}$
- Construct matrix M(x,y): $\mathrm{M} = \begin{pmatrix} \mathrm{Ix}^2 & \mathrm{Ix}*\mathrm{Iy} \\ \mathrm{Ix}*\mathrm{Iy} & \mathrm{Iy}^2 \end{pmatrix} = \begin{pmatrix} \mathrm{A} & \mathrm{C} \\ \mathrm{C} & \mathrm{B} \end{pmatrix}$
- Filter the matrices A, B and C by using e.g. 5 x 5 low pass filter:
  etc $\mathrm{A} \rightarrow \overline{\mathrm{A}}$

3.3 High Pass Filters, Harris Filter / Corner Filter (3)

Set $\overline{\mathrm{M}} = \begin{pmatrix} \overline{\mathrm{A}} & \overline{\mathrm{C}} \\ \overline{\mathrm{C}} & \overline{\mathrm{B}} \end{pmatrix}$

Define a „corner response function" Q(x,y):

$$\mathrm{Q}(x,y) = \det \overline{\mathrm{M}} \quad - \mathrm{thr1}\,(\mathrm{trace}\,\overline{\mathrm{M}}\,)^2$$
$$= \mathrm{thr2}\,\left\{(\overline{\mathrm{A}}\,\overline{\mathrm{B}} - \overline{\mathrm{C}}^2) - \mathrm{thr1}\,(\overline{\mathrm{A}} + \overline{\mathrm{B}})^2\right\} \rightarrow \text{large}$$

Default value for thr1⇒ 0.04 … 0.06, max. value 0.25

Check Q(x,y) > thr2 with thr2 ≈ 10^4 … 10^8

3.3 High Pass Filters, Harris Filter (Mathematical Background)

- Let be $\overline{\mathrm{M}} = \begin{pmatrix} \overline{\mathrm{A}} & \overline{\mathrm{C}} \\ \overline{\mathrm{C}} & \overline{\mathrm{B}} \end{pmatrix}$
- The matrices M resp. $\overline{\mathrm{M}}$ are symmetrical and can be diagonalized:

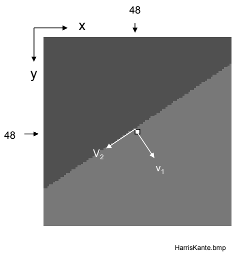
  $\overline{\mathrm{M}}' = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$ in which $\lambda_1 > 0$ and $\lambda_2 = 0$

  λ_i are called eigenvalues of $\overline{\mathrm{M}}'$

  $$\lambda_{1,2} = \frac{\mathrm{trace}\,\overline{\mathrm{M}}}{2} \pm \sqrt{\left(\frac{\mathrm{trace}\,\overline{\mathrm{M}}}{2}\right)^2 - \det\left(\overline{\mathrm{M}}\right)}$$
  $$= \frac{1}{2}\left(\overline{\mathrm{A}} + \overline{\mathrm{B}} \pm \sqrt{\overline{\mathrm{A}}^2 - 2\overline{\mathrm{A}}\,\overline{\mathrm{B}} + \overline{\mathrm{B}}^2 + 4\overline{\mathrm{C}}^2}\right)$$

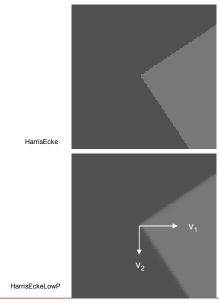
- The eigenvalues define the strength of the edge.
- The eigenvectors v_1 and v_2 define the direction of the corresponding edge.
- Calculation of eigenvectors: http://www.youtube.com/watch?v=i0ggTPAe1GU

3.3 High Pass Filters, Harris Filter (Mathematical Background)



Synthetic Image
if $y < -^2/_3\, x + 80 \rightarrow g = 80$
g = 120 else
g(48,48) = 120
Ix(x,y) = 44,44..
Iy(x,y) = 71,11..
Ix² = 1975,31..
Iy² = 5056,79..
IxIy = 3160,13..
http://www.arndt-bruenner.de/mathe/scripts/eigenwert2.htm
$\lambda_1 = 7031{,}77..$
$\lambda_2 = 0{,}3279..$
v_1 = eigenvector(λ_1) = {0,6249.., 1}
v_2 = eigenvector(λ_2) = {-1,600.., 1}

3.3 High Pass Filters, Harris Filter (Mathematical Background)



Synthetic Image
if $y < -^2/_3\, x + 80 \rightarrow g = 80$
If $y > +^3/_2\, x - 22 \rightarrow g = 80$
g = 120 else
g(48,48) = 120

filtered image by 3x3 Mean filter
g(49,48) = 106
$\overline{A} = 3919 \quad \overline{B} = 879,.. \quad \overline{C} = 36,..$
$\lambda_1 = 4203{,}9.. \quad v_1 = (84{,}45...,\ 1)$
$\lambda_2 = 752{,}0.. \quad v_2 = (-0{,}011..,1)$

3.3 High Pass Filters, Harris Filter (Mathematical Background)

- Both eigenvalues λ_1 and λ_2 should be significant in corner positions or

$$\lambda_1 - \lambda_2 = 2 * \sqrt{\left(\frac{\text{trace}\,\overline{M}}{2}\right)^2 - \det \overline{M}}$$

should be small. Note: $(0.5 * \text{trace}\ \overline{M})^2 > \det\ \overline{M}$

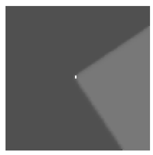
Define a „corner response function“ Q(x,y):

$$Q(x,y) = \det \overline{M} \quad - \text{thr}\ (\text{trace}\,\overline{M}\,)^2$$
$$= (\overline{A}\,\overline{B} - \overline{C}^2) - \text{thr}\ (\overline{A} + \overline{B})^2 \quad \rightarrow \quad \text{large}$$

Default value for thr 0.04 … 0.06, max. value 0.25

Check Q(x,y) > thr2 with thr2 ≈ 10^4 … 10^8

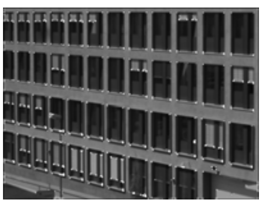
3.3 High Pass Filters, Harris Filter / Corner Filter



Result with th2 = $2*10^6$

3.3 High Pass Filters, Harris Filter / Corner Filter

Source image

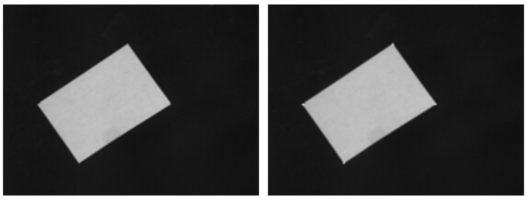


Harris filtered image,
for better visibility the grey intensity is set to 60%

Modellbildung:

Lineare Bewegung ohne Rotation
Rotation bei konstantem Schwerpunkt
Lineare Bewegung des Schwerpunktes mit Rotation
Nichtlineare Bewegung des Schwerpunktes mit Rotation
Zoom-in / zoom-out Bewegung

3.3 High Pass Filters, Harris Filter / Corner Filter



Orginalbild und Harris-gefiltertes Bild zur Bewegungsanalyse

Review of Chapter 3

- Low pass filters (Mean / Median / Gauss / Binomial)
- High pass filters (Prewitt / Sobel / Roberts / Laplace / LoG optional: Canny filter)
- Filtering as mask operation (no pixel operation)
- Filtering with masks of fixed and variable size
- Gradient of grey levels defined as vector
- Threshold operations (fixed / adaptive)
- Direction dependent and direction independent filtering
- Local and global filters
- Corner filter