



Universität Stuttgart

Pattern Recognition

Chapter 3:

Image Acquisition and Preprocessing

Prof. Dr.-Ing. Uwe Sörgel
soergel@ifp.uni-stuttgart.de



Contents

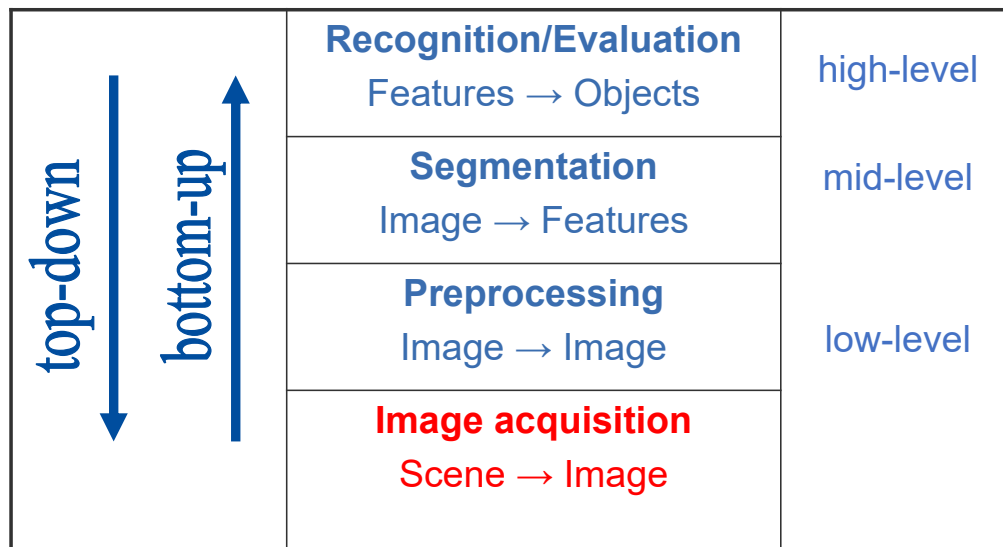
- Image Acquisition
- Preprocessing
 - Image enhancement
 - Image restoration
- Calculation of derivations



Universität Stuttgart



Level model of model-based image analysis



The digital image as a matrix of pixels

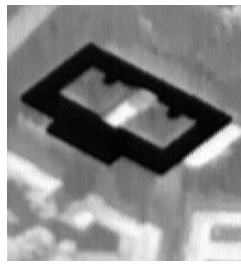
132	129	123	118	113	112	103	90	100	111	128	144	157	160	168	187	193	205	214	210	230	229	239	250	240	247	211	229	224	220	241	238	223	197	190	174	104	155	143	140	154	159	104	174	190	190	197	199	199	198	195	192	
111	107	93	90	92	94	102	130	121	160	163	191	193	193	176	206	207	209	202	224	231	246	226	249	262	227	223	213	216	219	240	241	252	232	234	226	220	212	195	159	159	129	120	90	101	111	136	153	179	177	173	173	
65	90	93	90	123	132	136	116	130	163	166	170	176	186	183	197	216	206	227	233	233	220	236	206	246	242	208	147	201	216	236	237	233	223	237	224	241	243	230	204	194	143	140	117	104	94	69	70	101	114	136	151	
103	116	140	162	162	162	164	116	130	163	166	170	186	187	188	166	176	202	196	231	223	200	184	170	196	236	187	96	103	162	212	226	196	173	221	216	226	243	240	217	214	162	178	162	137	111	95	70	84	63	47	104	
124	142	134	148	163	167	177	147	136	130	132	131	166	182	177	136	122	174	166	130	181	133	111	94	120	116	163	90	82	92	149	147	167	142	166	180	171	223	221	213	202	149	188	168	183	170	164	110	78	63	47	41	
146	132	137	126	146	162	162	174	139	115	114	118	107	106	118	127	99	91	90	72	76	81	82	75	79	82	86	86	95	84	91	96	135	124	107	142	110	154	181	177	183	114	191	194	189	190	171	161	135	100	83		
162	142	110	139	129	131	140	154	104	120	99	102	90	58	39	46	77	69	51	62	68	70	98	91	77	94	112	122	130	111	95	79	94	95	79	90	94	98	120	140	93	71	173	172	177	209	184	152	144	134			
160	159	155	101	120	111	120	106	106	99	59	71	56	52	50	55	52	53	51	59	93	95	91	79	99	101	99	99	96	95	92	95	90	92	112	90	99	95	135	76	40	110	106	127	173	165	161	140	140	136			
147	152	156	151	158	102	87	102	74	53	40	43	37	44	50	64	65	96	63	71	74	61	80	65	79	87	90	93	95	95	90	100	94	90	95	80	79	81	135	176	148	114	120	79	77	87	153	156	156	142	137		
134	120	137	130	110	98	47	61	30	40	48	42	94	46	66	63	147	167	136	92	63	43	71	78	66	87	104	90	64	118	107	98	97	97	66	84	66	81	130	180	196	190	146	66	72	96	68	137	160	176	155	162	
122	106	106	87	96	66	60	30	35	32	37	41	67	101	133	170	185	177	108	57	60	86	91	88	80	75	91	97	96	99	90	91	92	97	96	96	98	98	81	78	190	224	231	208	147	107	62	62	69	129	140	169	162
107	82	71	47	34	35	30	32	27	32	46	84	104	148	194	201	190	143	80	65	73	105	111	90	81	74	78	81	90	81	82	80	75	88	81	81	140	114	88	103	201	236	226	221	175	121	64	69	100	108	133	182	
56	51	37	34	31	24	30	32	33	73	123	190	170	197	203	201	175	87	42	55	70	103	102	95	79	75	73	73	80	66	63	65	77	73	88	151	125	113	82	142	218	219	223	219	204	137	85	91	83	94	125		
36	25	21	23	21	24	41	47	94	146	160	173	192	190	197	190	129	52	56	66	92	92	91	90	76	76	69	50	126	184	110	47	47	51	59	90	119	119	125	90	90	162	220	216	219	211	206	154	126	93	61	91	
30	26	17	26	24	41	67	94	140	163	171	179	183	190	191	170	87	52	62	72	76	71	66	65	92	92	40	60	200	251	170	49	37	35	50	91	99	139	149	95	102	124	207	210	221	212	214	193	165	100	54	95	
11	16	11	27	46	80	115	139	105	170	161	185	192	180	151	60	57	66	77	90	91	66	90	60	62	41	47	142	212	137	40	33	32	35	62	60	125	125	121	106	97	170	225	235	223	215	203	162	154	61	96		
16	18	26	63	89	116	146	166	169	160	171	184	191	193	192	146	68	66	90	111	124	138	93	81	63	36	36	36	36	46	61	44	33	30	26	26	46	81	121	136	138	74	82	146	216	236	232	200	190	170	169	127	60
23	22	44	62	111	131	144	162	163	162	173	182	181	201	200	131	50	72	104	114	120	118	86	70	48	29	50	32	35	36	33	29	24	22	18	28	90	117	127	135	84	62	118	205	228	230	223	201	178	171	140	68	
44	37	58	96	118	127	139	140	161	163	176	184	186	207	202	128	56	80	113	115	112	111	102	104	42	24	26	28	31	31	27	24	20	17	14	16	62	94	138	143	122	95	121	191	224	225	235	214	197	171	147	126	
47	37	55	87	109	121	139	152	100	172	183	192	203	210	204	137	67	75	87	95	93	106	115	110	52	21	23	24	28	29	23	18	13	13	11	12	53	102	130	152	117	115	122	181	218	222	221	213	202	183	109	146	
47	36	43	75	96	110	138	155	169	192	191	198	205	210	205	149	69	71	92	95	104	130	121	126	89	22	18	20	23	22	16	12	9	9	7	16	72	106	106	130	139	100	119	177	215	210	213	205	201	190	170	156	
62	52	52	55	71	96	133	156	173	186	193	200	205	206	205	161	62	58	95	99	119	142	135	121	111	45	65	15	17	17	13	6	4	4	5	22	64	94	123	129	124	111	116	160	211	214	212	205	166	167	175	156	
61	60	60	63	66	76	117	164	176	186	192	198	206	208	207	170	61	67	71	90	111	117	106	123	160	100	23	16	12	11	8	4	1	2	8	34	63	100	136	121	91	86	123	162	206	206	200	164	164	171	162	153	
67	60	80	84	88	90	101	134	166	183	191	196	202	207	208	193	130	76	83	103	93	100	129	140	151	142	76	23	9	6	4	2	1	5	20	48	69	104	133	132	91	89	136	189	205	203	197	195	199	170	169	169	
53	74	90	89	96	102	112	118	141	169	184	193	201	205	206	200	161	94	91	104	117	132	144	130	135	142	140	108	63	21	12	10	18	28	55	66	88	121	134	115	106	115	192	204	190	194	190	182	171	159	162		
90	90	70	90	99	96	108	124	132	138	162	184	195	203	204	203	187	120	72	78	94	117	111	122	130	147	144	146	133	95	90	52	57	59	67	89	113	132	133	117	90	121	169	192	195	181	180	180	171	162	153	146	
72	74	94	92	99	102	112	117	120	130	157	179	192	199	205	199	165	96	46	55	90	114	120	132	139	142	135	125	99	94	81	90	92	100	110	111	115	95	91	125	170	167	169	167	170	159	152	143	140				
86	81	73	63	76	94	99	102	131	140	144	146	156	168	169	202	190	141	63	30	61	94	114	116	138	139	151	142	119	124	113	80	52	40	102	100	97	116	117	100	97	156	166	166	164	163	177	167	156	146	143		
103	101	97	70	80	77	70	87	96	96	124	140	148	164	160	170	103	103	170	124	70	78	78	106	130	141	163	140	126	116	123	80	61	40	74	80	84	91	100	90	133	178	161	166	178	176	160	160	147	137	124		
106	101	93	75	76	77	78	78	91	80	117	138	136	140	152	157	163	173	180	182	116	66	70	102	118	124	144	120	127	123	111	70	46	52	79	88	74	79	88														

Many entities can be visualized as image

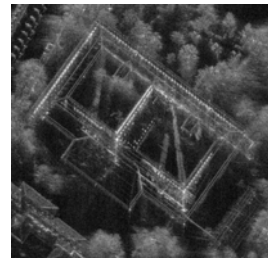
- In principle, any mathematical function of type $g=f(x,y)$ can be coded as image.
 - Height with respect to some reference
 - Chemical concentration
 - Temperature
 - Pressure
- Some examples related to our field:



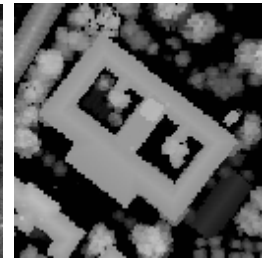
Aerial image



Infrared image (IR)

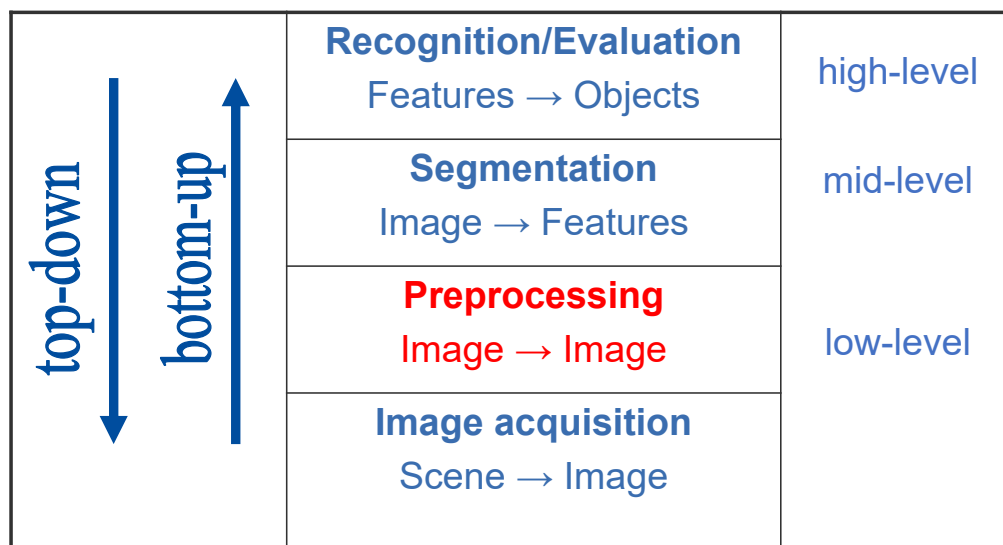


Radar image (SAR)



Laserscan height

Level model of model-based image analysis



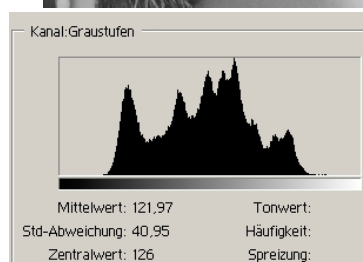
Contents

- Image Acquisition
- Preprocessing
 - Image enhancement
 - Image restoration
- Calculation of derivations

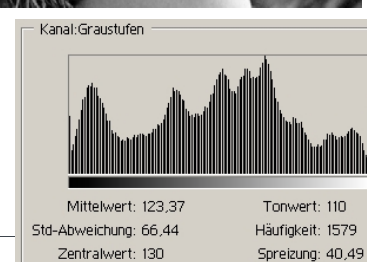
Example for histogram normalisation



original



On both ends 0.5%
of values forced
into saturation



Problem statement of image restoration

- The „true“ image $g(x,y)$ is corrupted by some distortion $n(x,y)$.
- For example, additive zero mean Gaussian noise $\eta(x,y)$:

$$g_R(x,y) = g(x,y) + \eta(x,y)$$

- There are also other kinds of distortions like „Salt-and-Pepper“ noise (randomly distributed white and black noise pixels)



Original



Gaussian noise



„Salt-and-Pepper“

Local filter operation

- Point operations only consider the current pixel value
- Global transformations use the whole image
- *Local filter operations* are a mixture of both:
 - Neighbourhood around current pixel is considered.
 - Neighbourhood is defined via a window (filter mask).
 - Filter mask is continuously moved across the image.
 - In each position value of one pixel (usually the central pixel) is determined.



Different types of local filters



• Linear Filters

- Linear digital filters $h(x,y)$ carry out **convolution** on the image $g(x,y)$.
- The coefficients of filter matrix represent weights, which are multiplied with corresponding pixel values, before adding all products.
- For homogeneous filters the coefficients are constant, independent of image location (shift invariant)

$$h(x,y) = \begin{bmatrix} -1 & -2 & -1 \\ -2 & 12 & -2 \\ -1 & -2 & -1 \end{bmatrix}$$

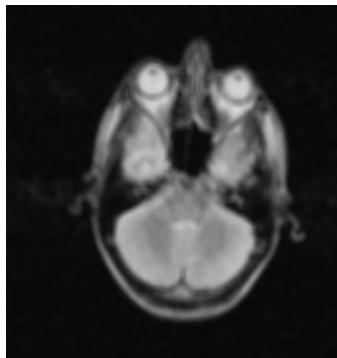
- Non-linear filters, for example.:
 - Rank filter
 - **Median filter**
 - Morphological filter
 - Diffusion filter

Application of linear local filters

- Examples for typical application of some linear local filter
 - Smoothing
 - Enhancement of
 - Salient points
 - Edges
- Can not be achieved by point operations → Context required



Original



Smoothed



Gradient image

Box filter (rectangular filter)

- The most simple smoothing filter is the box or rectangular filter of size 3 x 3.

$$h(x, y) = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \Rightarrow g'(x, y) = g * h = \frac{1}{9} \sum_{i=-1}^1 \sum_{j=-1}^1 g(x+i, y+j)$$

Example

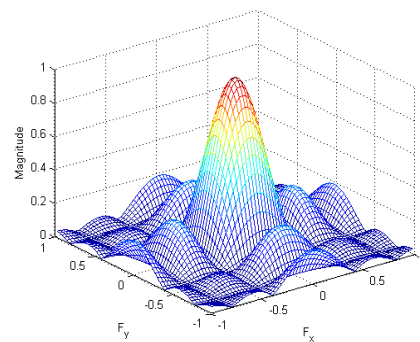
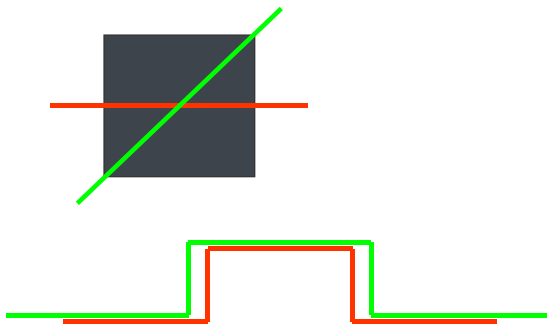
g	..	93	..
:	113	116	104
15	99	101	125
:	0	107	105

Element "101" at position (93,15)

$$g'(93,15) = \frac{1}{9} (113 + 116 + 104 + 99 + 101 + 125 + 0 + 107 + 105) = \underline{\underline{96.7 \approx 97}}$$

Disadvantages of the box filter

- Box filters are no ideal low-pass filters:



Transfer function

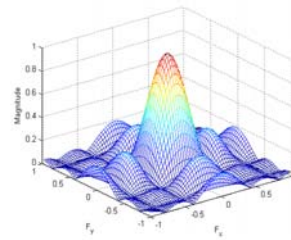
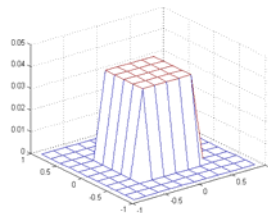
- The box filter is rotation variant
 - Orientation dependency (diagonal longer than edges)
 - Non-uniform suppression of high frequencies
 - Undesired sidelobes may cause artifacts

Comparison of box and Gauss operator

spatial domain

frequency domain

box operator

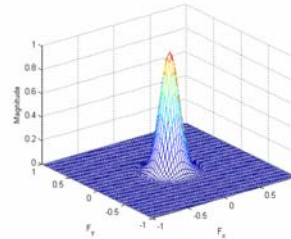
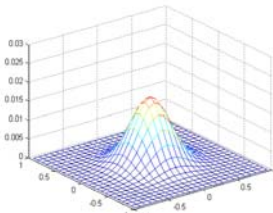


disadvantage: not rotation invariant, does not suppress all high frequencies

advantage: recursive implementation possible

Gauss operator

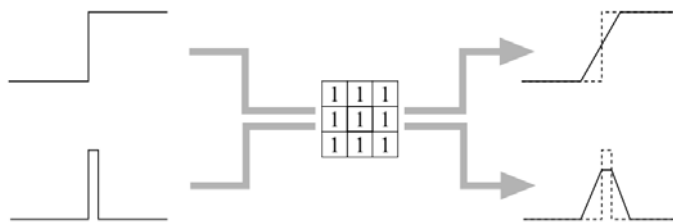
$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



advantage: rotation invariant, suppresses all high frequencies

disadvantage: only approx. recursive implementation -> binomial filter

Disadvantage of linear smoothing filters



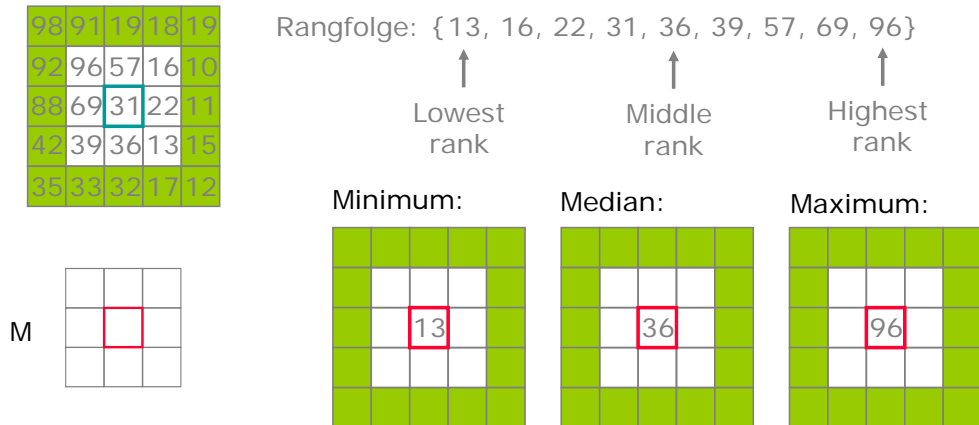
Blur at edges
and lines



Salt-and-Pepper-
Noise not removed

Rank filter

- The grey values inside mask are sorted in ascending manner.
- The choice of the rank (i.e., index) depends on the desired purpose
 - Minimum filter: Lowest rank
 - Maximum filter: Highest rank
 - Median filter: Middle rank



Example of application of median filter

- Removes Salt-and-Pepper noise
- Mostly quite convincing results



Original image

Original image +
Salt-and-Pepper noise

After Median filtering (3x3)

Contents

- Image Acquisition
- Preprocessing
 - Image enhancement
 - Image restoration
- Calculation of derivations

The derivations of an image and their meanings

1. Derivation

$$g_x(x, y) = \frac{\partial g(x, y)}{\partial x}, g_y(x, y) = \frac{\partial g(x, y)}{\partial y}$$

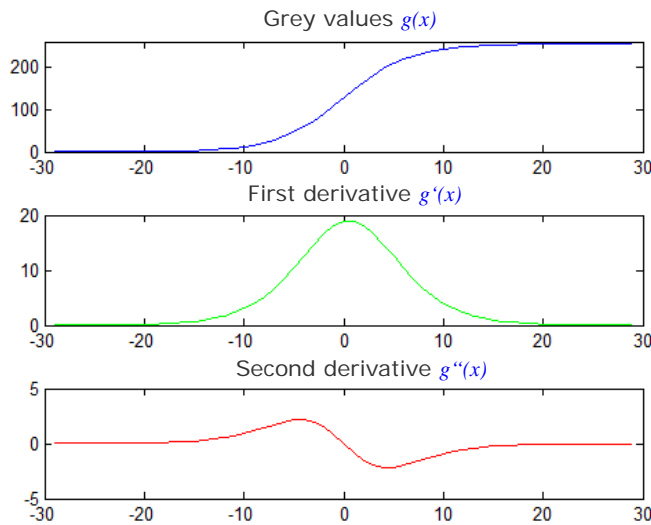
- Digital **images**: Change of grey values → edges, points
- Digital **surface models**: normal vectors, height jumps

2. Derivation

$$g_{xx}(x, y) = \frac{\partial^2 g(x, y)}{\partial x^2}, g_{yy}(x, y) = \frac{\partial^2 g(x, y)}{\partial y^2}, g_{xy}(x, y) = \frac{\partial^2 g(x, y)}{\partial x \partial y}$$

- Digital **images**: At edges the 2. derivation vanishes
- Digital **surface models**: Curvature, torsion

High pass filters: Edge detection



edge = strong straight change of grey values $g(x)$

gradient operator:

1. derivative $g'(x)$

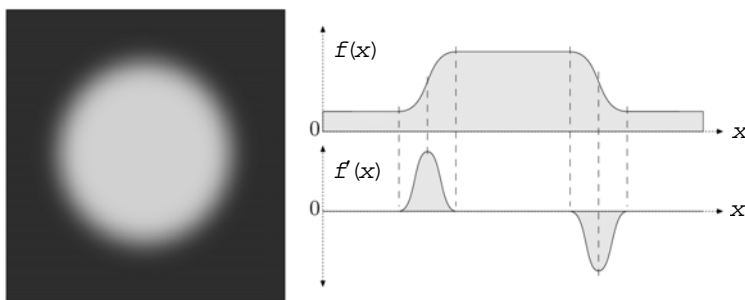
→ Search for maxima

Laplace-Operator:

2. derivative $g''(x)$

→ Search for zero crossings

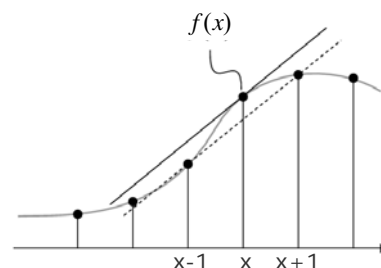
Gradient based edge detection



Discrete approximation of gradient by differencing

- Take left and right value to end up at grid position.
- Σ (filter coeff.) = 0 (i.e., high-pass filter)

$$f'(x) = \frac{\partial f(x)}{\partial x} \approx \frac{f(x+1) - f(x-1)}{2} = 0.5 \cdot (f(x+1) - f(x-1))$$



Prewitt-Operator

Simple realization:

$$f'(x) \approx 0.5 \cdot (f(x+1) - f(x-1)) \rightarrow h_x = 0.5 \cdot \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

$$f'(y) \approx 0.5 \cdot (f(y+1) - f(y-1)) \rightarrow h_y = 0.5 \cdot \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

However, *differencing enhances noise*:

→ noise suppression by low pass filtering (smoothing) in across direction!

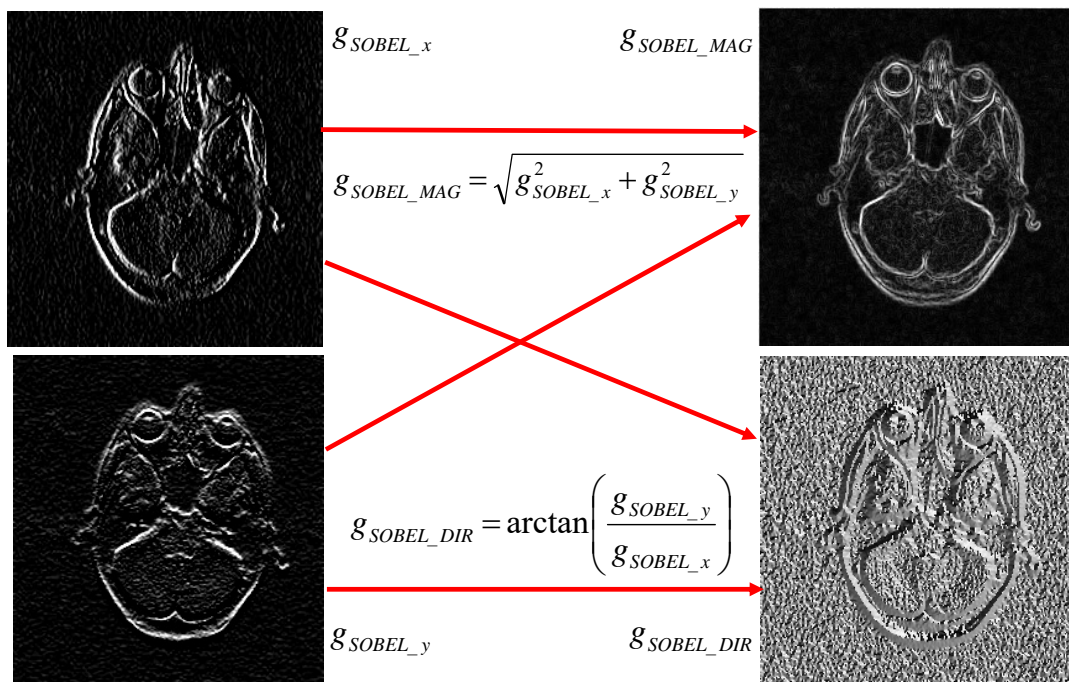
$$h_x = \frac{1}{6} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

gradient smoothing

$$h_y = \frac{1}{6} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

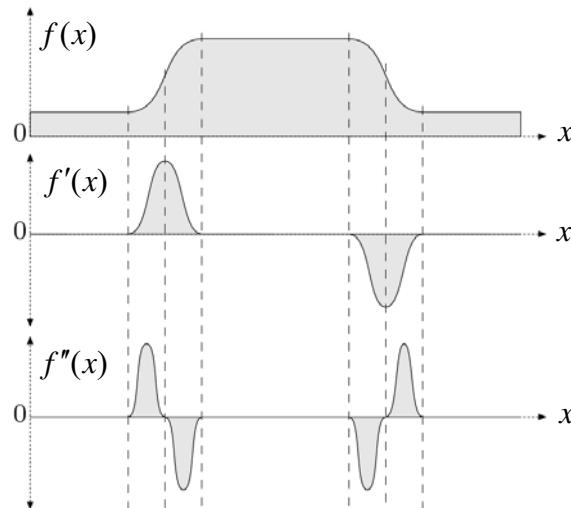
smoothing gradient

Amplitude and direction of gradient



Laplace operator

- Alternative to gradient filtering (1. derivative)
- Search for zero crossings of 2nd derivative.



Laplace operator

Laplace Operator Δ computes sum of 2nd partial derivatives of a continuous function $g(x,y)$ for the variables x and y :

$$\Delta g(x, y) = \frac{\partial^2 g(x, y)}{\partial x^2} + \frac{\partial^2 g(x, y)}{\partial y^2}$$

Discrete realization of 2nd derivative: difference of difference of neighboring pixels per row and column

$$\frac{\partial^2 g(x, y)}{\partial x^2} \approx \frac{g(x+1, y) - g(x, y)}{(x+1) - x} - \frac{g(x, y) - g(x-1, y)}{x - (x-1)}$$

$$\approx g(x+1, y) - 2 \cdot g(x, y) + g(x-1, y)$$

$$\frac{\partial^2 g(x, y)}{\partial y^2} \approx g(x, y+1) - 2 \cdot g(x, y) + g(x, y-1)$$

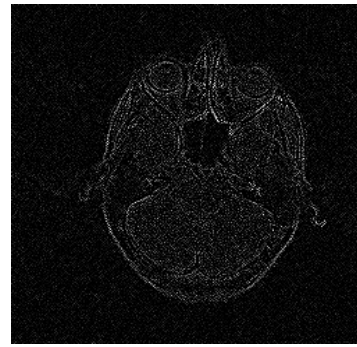
Laplace-Operator for 2D

- It is better to add the two filter masks for each direction in order to yield the result in only a single run:

$$h_{\Delta} = h_{\Delta x} + h_{\Delta y} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



Original



Result

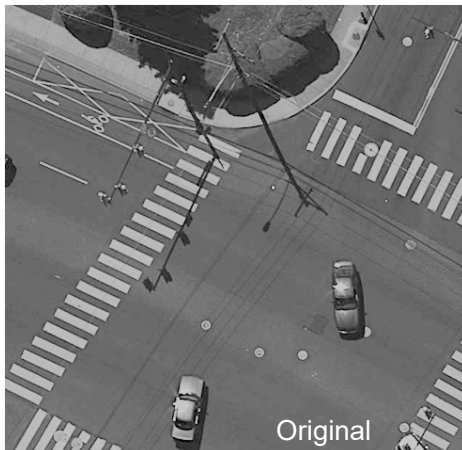
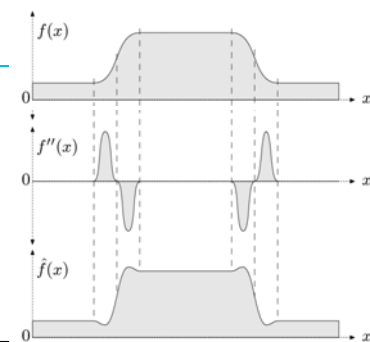
Unfortunately, besides desired object contours also noise is enhanced.

Laplace operator: Image sharpening

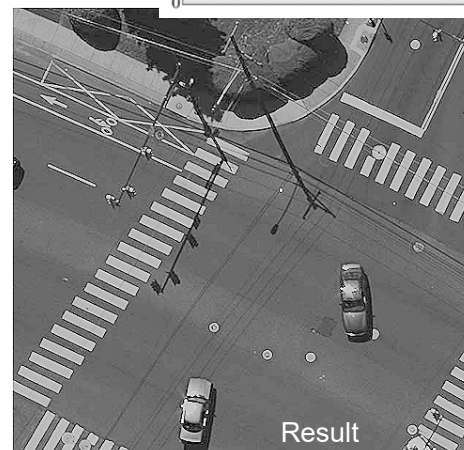
- “Sharper” image by adding high-pass signal:

$$h_{\text{Sharpening}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{4} \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 8 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

“Image” - k · Laplacian = Sharpening



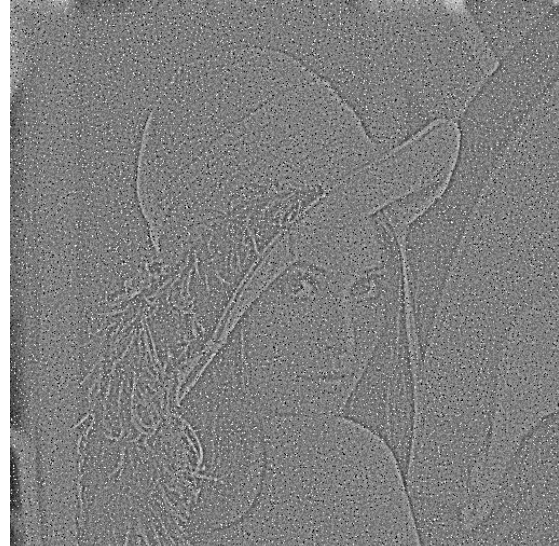
Original



Result

Laplace operator in presence of noise

- The Laplace operator is very noise sensitive because difference operation enhances noise – now we have two of them in a row...



LoG operator (Laplace of Gaussian)

- In order to mitigate noise, it is beneficial to smooth the image ($G * \text{image}$) before subsequent convolution with Laplace Operator ($\Delta(G * \text{image})$).
- Alternatively, - same as for gradient approach - we rather should calculate the 2nd derivation of Gaussian first and use this result for convolution ($(\Delta G) * \text{image}$):

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\text{LoG: } \Delta G = \frac{\partial^2 G(x, y)}{(\partial x)^2} + \frac{\partial^2 G(x, y)}{(\partial y)^2}$$

$$\frac{\partial G(x, y)}{\partial x} = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \cdot \frac{-x}{\sigma^2}$$

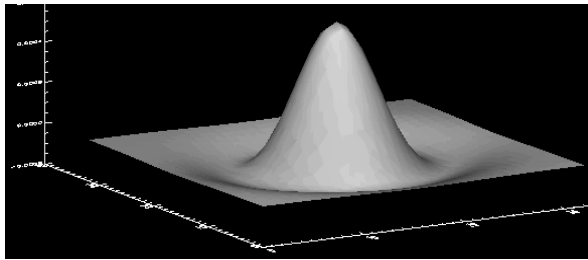
$$\frac{\partial^2 G(x, y)}{(\partial x)^2} = \frac{-1}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} + \frac{1}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \cdot \frac{x^2}{\sigma^2}$$

$$= \frac{1}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \left[\left(\frac{x}{\sigma} \right)^2 - 1 \right]$$

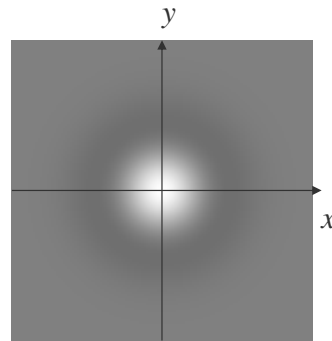
$$\frac{\partial^2 G(x, y)}{(\partial y)^2} = \frac{1}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \left[\left(\frac{y}{\sigma} \right)^2 - 1 \right]$$

LoG-Operator (Laplace of Gaussian)

$$LoG(x, y) = \frac{1}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \left[\frac{x^2+y^2}{\sigma^2} - 2 \right]$$



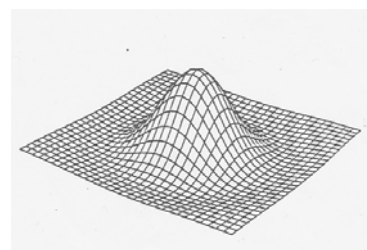
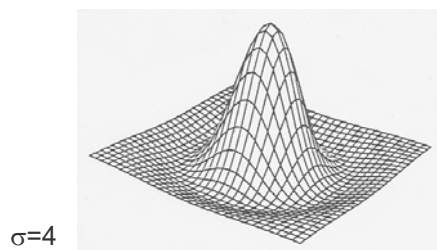
LoG Operator



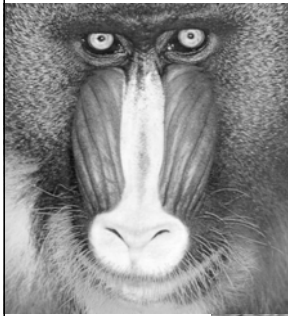
LoG-Operator (Laplace of Gaussian)



- Because of similarity with Sombrero this operator is sometime called Mexican-Hat-Operator.
- The result depends on the choice standard deviation σ .
- An edge is inferred from zero crossings of the 2nd derivation.
- In contrast to gradient methods, which always rely to some threshold, we yield always closed curves



LoG-Operator: Example



LoG, $\sigma = 2.0$

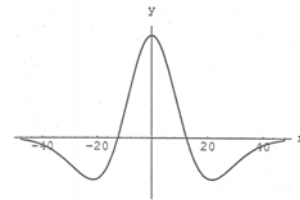
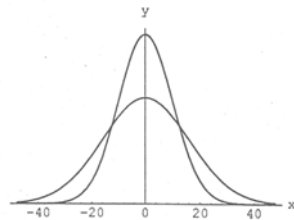


LoG, $\sigma = 4.0$

DoG operator (Difference of Gaussians)

- The very good LoG operator can be quite good approximated by the difference of two low-pass filterings with Gaussians of different σ .
- However, the ration of standard deviations is recommended to be $\sigma_1/\sigma_2=1.6$.
- Similar processing I taking place in visual system of humans.

1d



2d

