

Predspracovanie obrazu

Gonzales, Woods: Digital Image Processing

kapitola: Image restoration

Sonka, Hlavac, Boyle: Image Processing, Analysis and Machine vision,

kapitola: Image pre-processing

Szeliski: Computer Vision: Algorithms and Applications

kapitola: Image processing

Predspracovanie obrazu

- operácie s obrazmi na najnižšej úrovni abstrakcie
- aj vstup aj výstup sú ikonické dáta

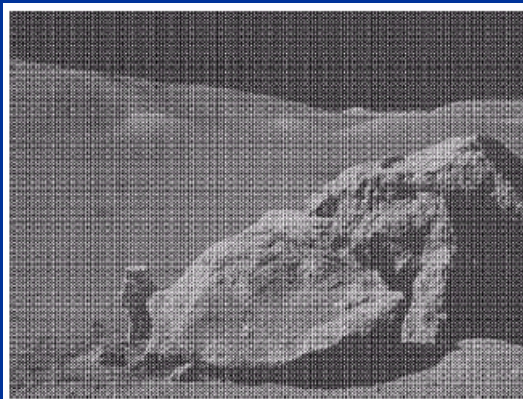
- **Cieľom** predspracovania je zlepšenie kvality obrazových dát, ktorá potlačí neželané deformácie a skreslenia alebo zlepší niektoré črty obrazu, ktoré sú dôležité z hľadiska ďalšieho spracovania.
- Niekedy vylepšovanie (enhancement) a rekonštrukcia (restoration) obrazu

Existujú 4 základné typy metód predspracovania:

- Jasové transformácie – transformácia závisí od vlastností pixla samotného
- Geometrické transformácie
- **Lokálne predspracovanie** – transformácie závisí od samotného pixla a jeho okolia
- Rekonštrukcia obrazu – niekedy nepatrí, rieši chyby pri snímaní

Lokálne predspracovanie

- Metódy lokálneho predspracovania používajú malé okolie obrazového bodu (obyčajne štvorcové)
- zo vstupného obrazu vytvoríme novú jasovú hodnotu obrazového bodu vo výstupnom obraze.
- Také operácie sa nazývajú **filtrácie**.



- Predspracovanie môže sledovať dva ciele: *vyhladzovanie* a *detekciu (zvýraznenie) hrán*.
 - vyhladzovanie – rozostčuje hrany, lebo potláča vyššie frekvencie
 - detekcia hrán – zvýrazňuje vyššie frekvencie, zvýrazňuje aj šum
 - to je protichodné, ale niektoré techniky robia oboje
- Iná klasifikácie metód lokálneho predspracovania rozlišuje lineárne a nelineárne transformácie.

Noise and Images

The sources of noise in digital images arise during image acquisition (digitization), transmission and processing

- Imaging sensors can be affected by ambient conditions
- Interference can be added to an image during transmission



Noise Model

We can consider a noisy image to be modelled as follows:

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

$f(x, y)$ is the original image pixel,

$\eta(x, y)$ is the noise term and

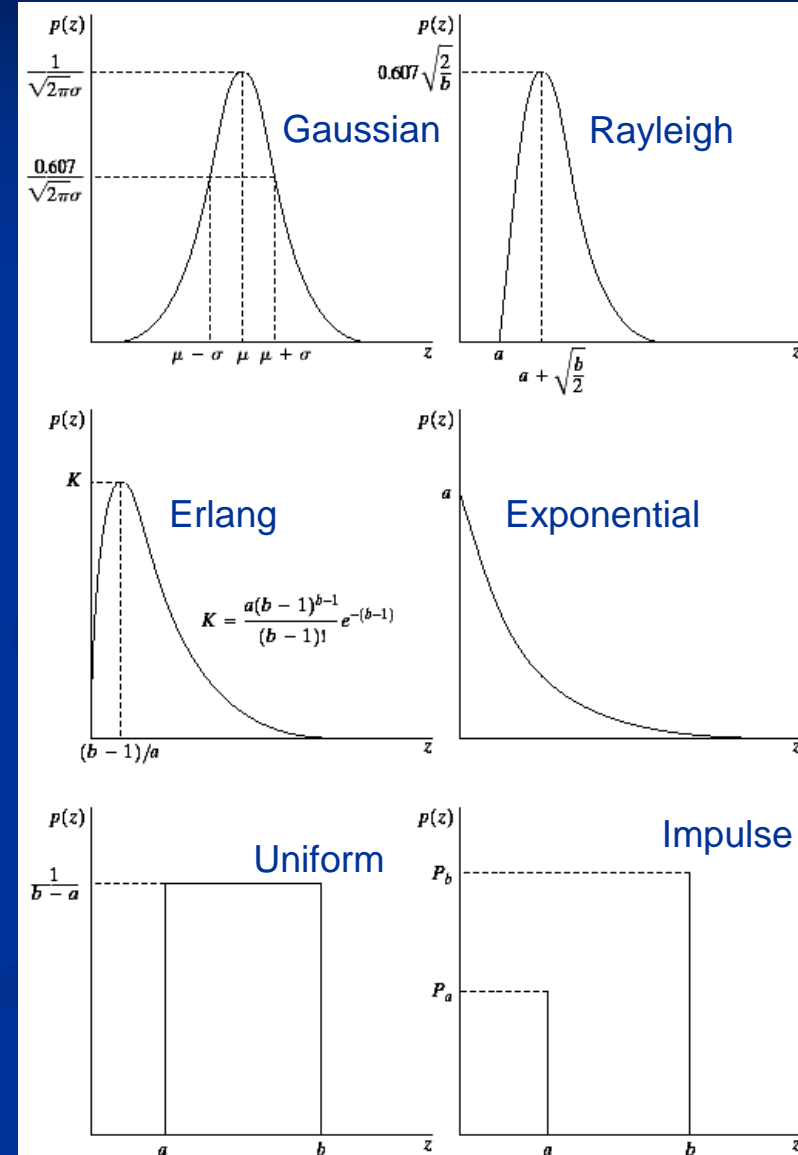
$g(x, y)$ is the resulting noisy pixel

If we can estimate the model it will help us to figure out how to restore the image

Noise Models

There are many different models for the image noise term $\eta(x, y)$:

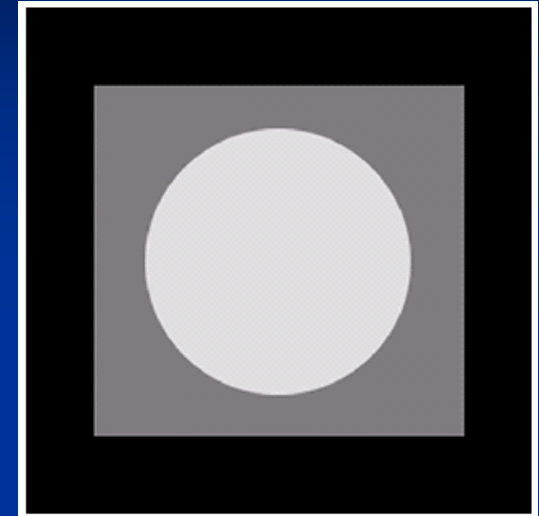
- Gaussian – poor illum, high temp.
 - Most common model
- Rayleigh - range imaging
- Erlang (Gamma) – laser imaging
- Exponential - laser imaging
- Uniform
- Impulse - digitalization
 - *Salt and pepper* noise



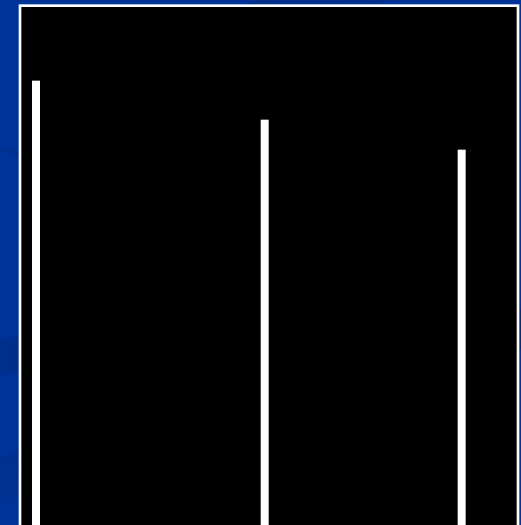
Noise Example

The test pattern to the right is ideal for demonstrating the addition of noise

The following slides will show the result of adding noise based on various models to this image

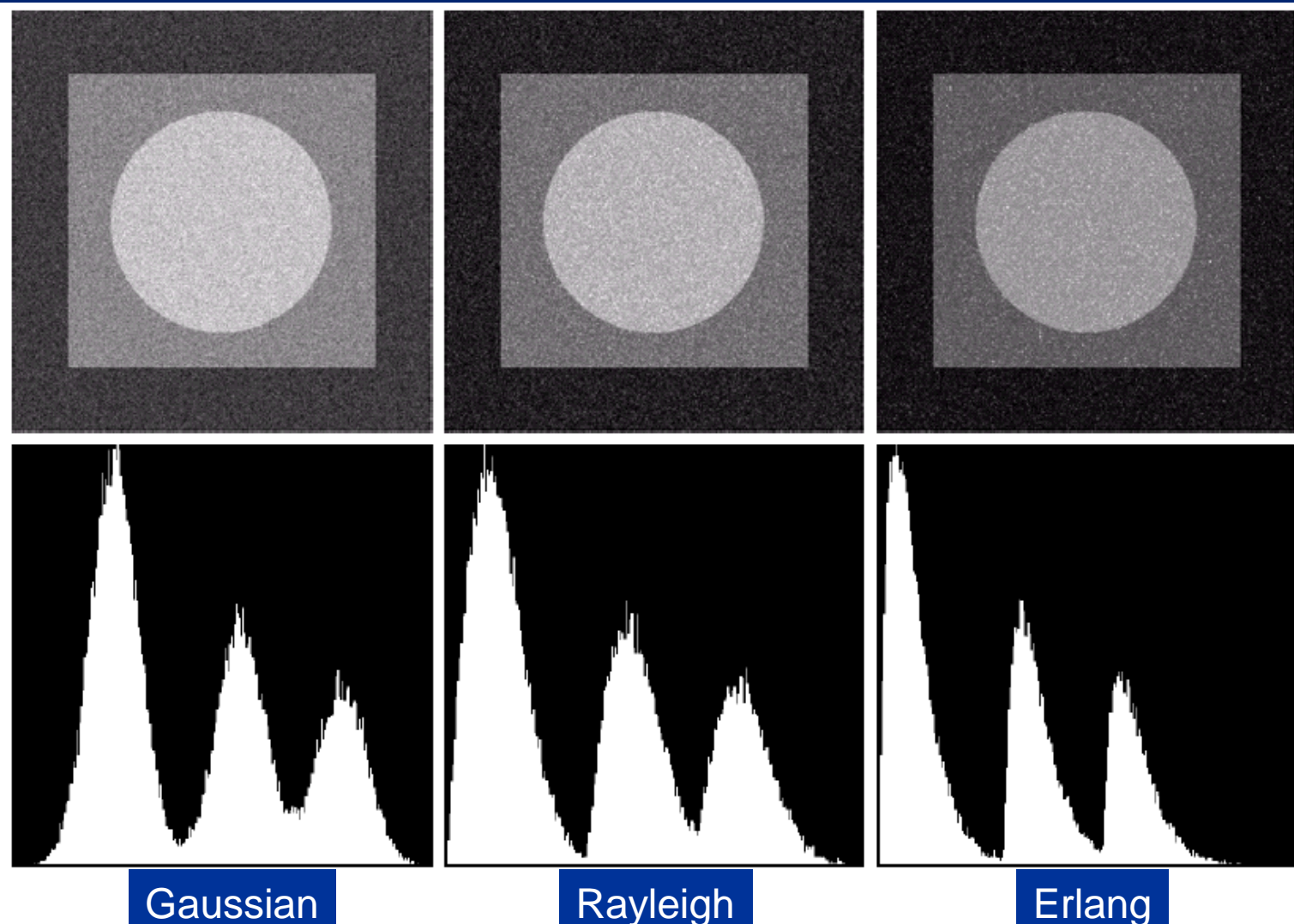


Image

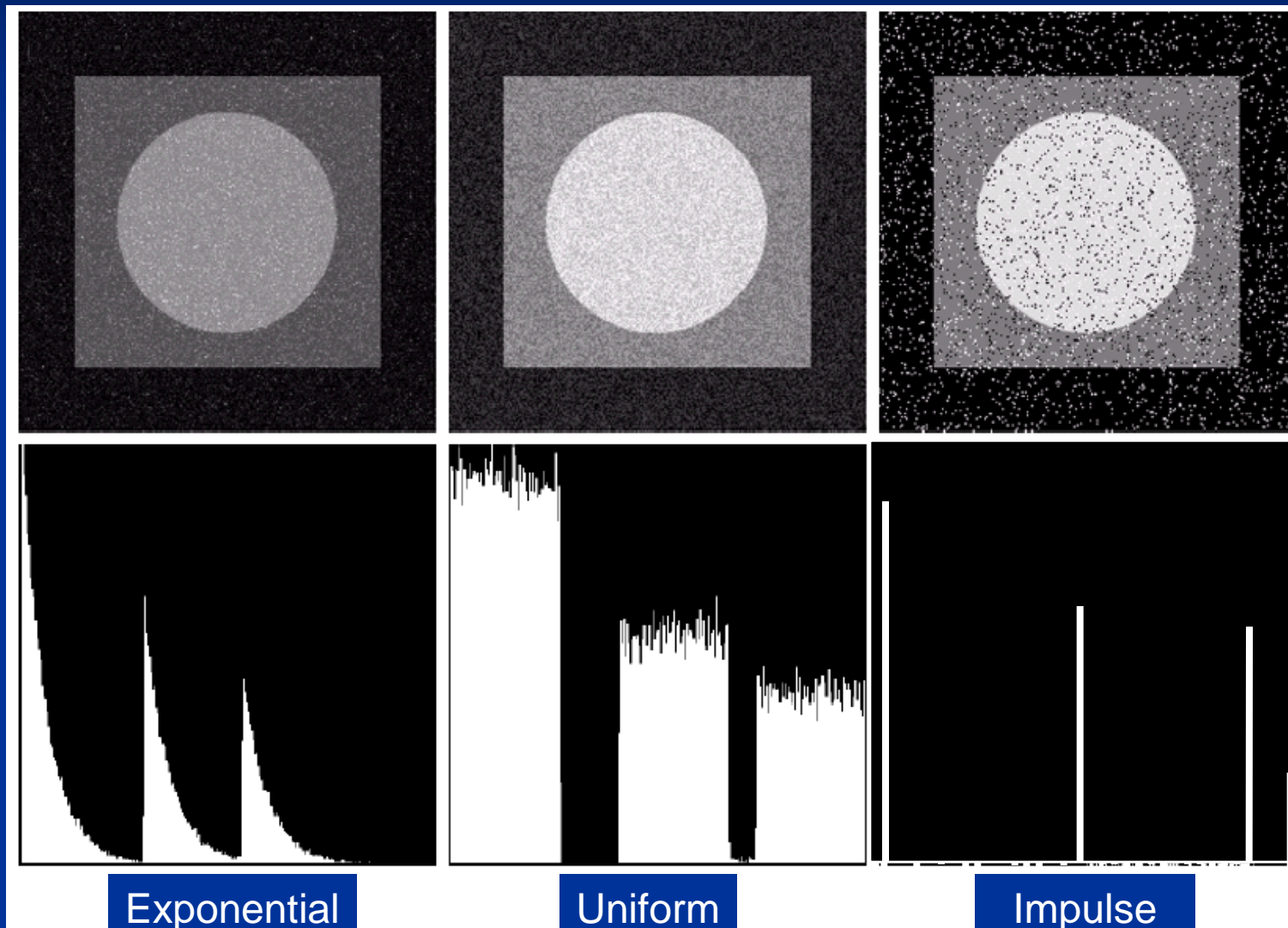


Histogram

Noise Example (cont...)



Noise Example (cont...)



Filtrovanie obrazu

- Lineárne filtre
 - Výsledná hodnota pixla je vážený súčet okolitých pixlov (mean, ...)
- Nelineárne filtre
 - Median filtering
 - Bilaterálny filtering
 - ...
- Morfológia – na nasledujúcej prednáške

Vyhladzovanie

- založené na priemerovaní a jeho modifikáciách, alebo na usporiadaných filtroch (ako medián)

$$f(i, j) = \sum_{(m, n) \in \mathcal{O}} h(i - m, j - n) g(m, n)$$

- Ekvivalentne diskkrétnej konvolúcii s kernelom h

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$h = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$h = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$



(a)



(b)



(c)



(d)

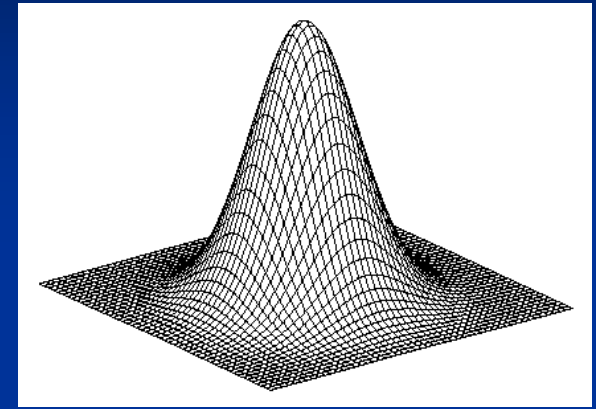
Figure 5.10: Noise with Gaussian distribution and averaging filters. (a) Original image. (b) Superimposed noise (random Gaussian noise characterized by zero mean and standard deviation equal to one-half of the gray-level standard deviation of the original image). (c) 3×3 averaging. (d) 7×7 averaging.

Gaussov filter

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

- Vyhladenie obrazu
- Vhodný pri odstraňovaní Gaussovho šumu

Gaussian



Filtering to Remove Noise

We can use spatial filters of different kinds to remove different kinds of noise

The **arithmetic mean** filter is a very simple one and is calculated as follows:

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t)$$

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

This is implemented as the simple smoothing filter

Blurs the image to remove noise

Other Means

There are different kinds of mean filters all of which exhibit slightly different behaviour:

- Geometric Mean
- Harmonic Mean
- Contraharmonic Mean

Other Means (cont...)

There are other variants on the mean which can give different performance

Geometric Mean:

$$\hat{f}(x, y) = \left[\prod_{(s,t) \in S_{xy}} g(s, t) \right]^{\frac{1}{mn}}$$

Achieves similar smoothing to the arithmetic mean, but tends to lose less image detail

Other Means (cont...)

Harmonic Mean:

$$\hat{f}(x, y) = \frac{mn}{\sum_{(s,t) \in S_{xy}} \frac{1}{g(s,t)}}$$

Works well for **salt noise**, but fails for pepper noise

Also does well for other kinds of noise such as **Gaussian noise**

Other Means (cont...)

Contraharmonic Mean:

$$\hat{f}(x, y) = \frac{\sum_{(s,t) \in \mathcal{S}_{xy}} g(s, t)^{Q+1}}{\sum_{(s,t) \in \mathcal{S}_{xy}} g(s, t)^Q}$$

Q is the *order* of the filter and adjusting its value changes the filter's behaviour

Positive values of Q eliminate pepper noise

Negative values of Q eliminate salt noise

Noise Removal Examples

Original Image

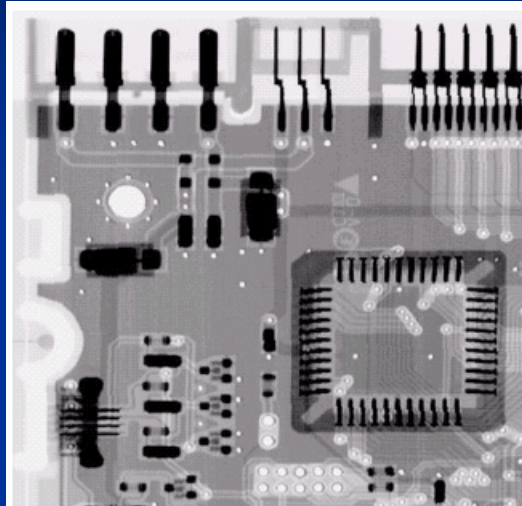
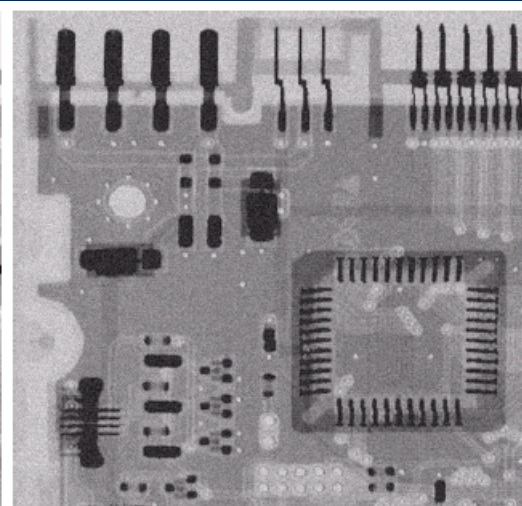
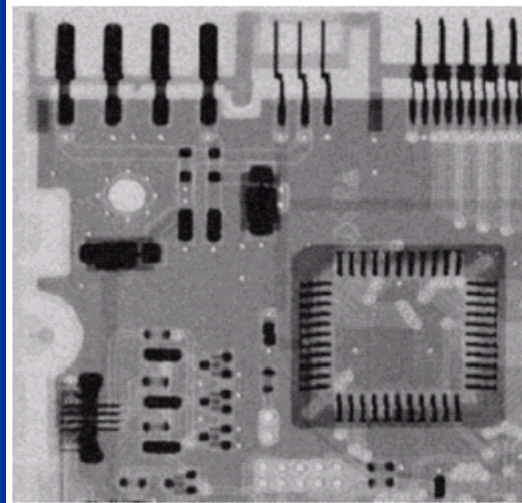


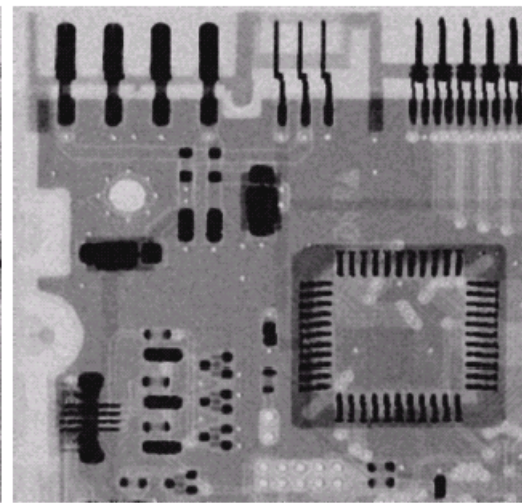
Image Corrupted By Gaussian Noise



After A 3*3 Arithmetic Mean Filter

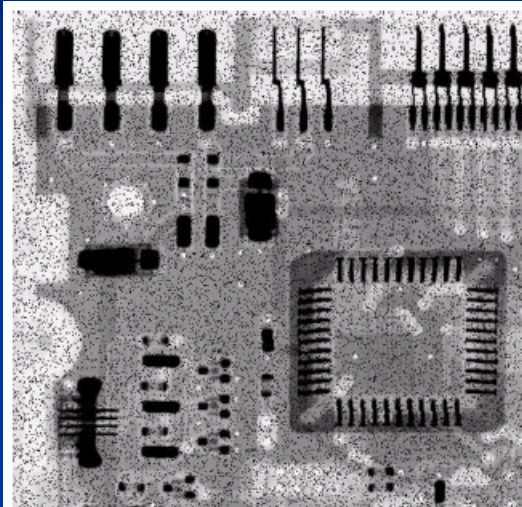


After A 3*3 Geometric Mean Filter

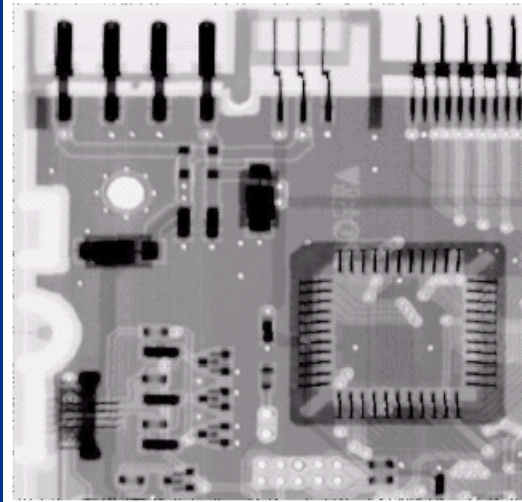


Noise Removal Examples (cont...)

Image
Corrupted
By Pepper
Noise



Result of
Filtering Above
With 3*3
Contraharmonic
 $Q=1.5$



Noise Removal Examples (cont...)

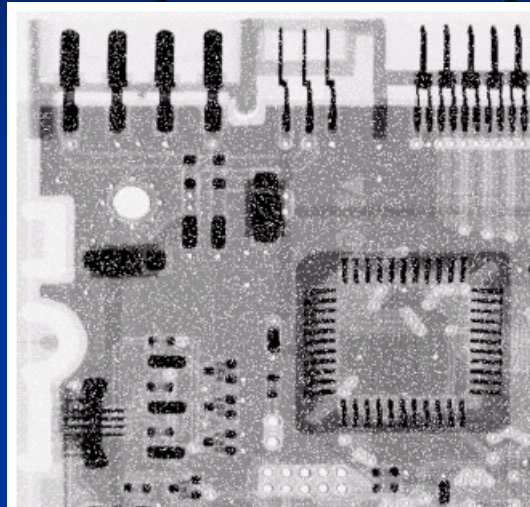
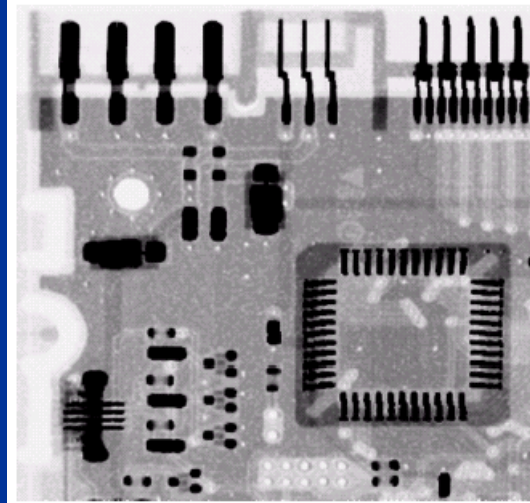


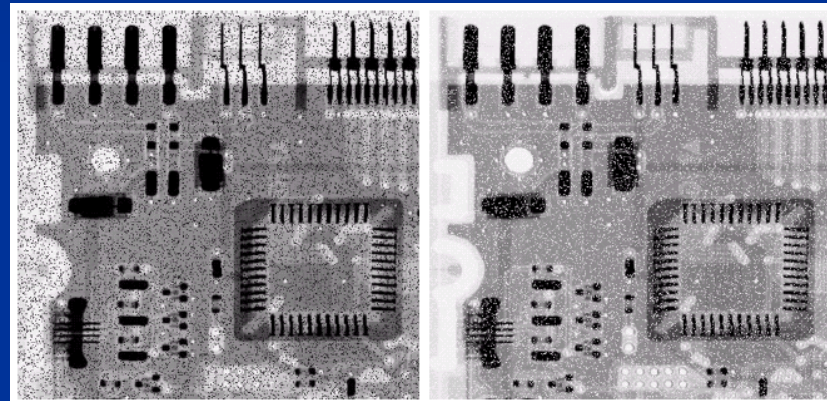
Image
Corrupted
By Salt
Noise



Result of
Filtering Above
With 3*3
Contraharmonic
 $Q=-1.5$

Contraharmonic Filter

Choosing the wrong value for Q when using the contraharmonic filter can have drastic results

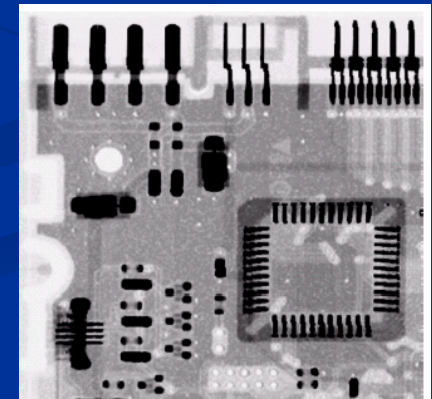
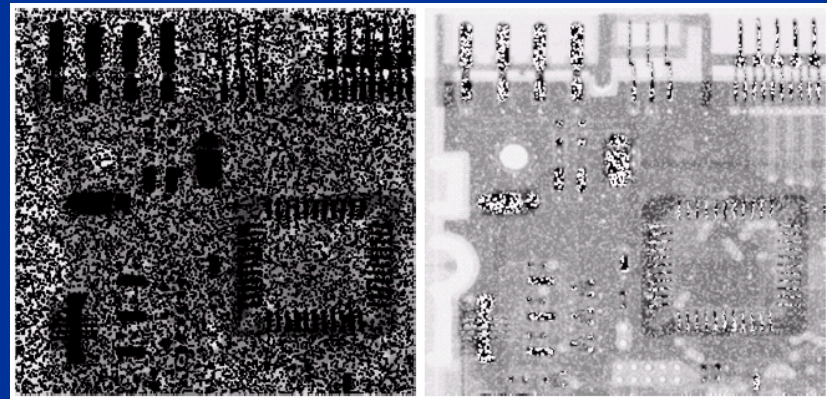
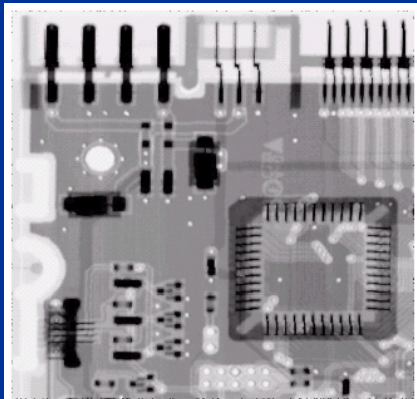


$Q=1.5$

$Q=-1.5$

$Q=1.5$

$Q=-1.5$



Nelineárne filtre

Filtre založené na usporiadaní pixlov

- Median filter
- Max and min filter
- Midpoint filter
- Alpha trimmed mean filter

Nelineárne filtre

Median Filter:

$$\hat{f}(x, y) = \underset{(s,t) \in S_{xy}}{\text{median}}\{g(s, t)\}$$

- Excellent at noise removal, **without the smoothing effects** that can occur with other smoothing filters
- Particularly good when **salt and pepper** noise is present

Nelineárne filtre

Alpha-Trimmed Mean Filter:

$$\hat{f}(x, y) = \frac{1}{mn - d} \sum_{(s,t) \in S_{xy}} g_r(s, t)$$

We can delete the $d/2$ lowest and $d/2$ highest grey levels

So $g_r(s, t)$ represents the remaining $mn - d$ pixels

Nelineárne filtre

- Median filtering
 - Pri filtrovaní snímacieho šumu (shot noise)
- Alfa trimmed mean
 - Odstránim najmenšiu a najväčšiu hodnotu

1	2	1	2	4
2	1	3	5	8
1	3	7	6	9
3	4	8	6	7
4	5	7	8	9

(a) median = 4

1	2	1	2	4
2	1	3	5	8
1	3	7	6	9
3	4	8	6	7
4	5	7	8	9

(b) α -mean = 4.6

Nelineárne filtre

Max Filter:

$$\hat{f}(x, y) = \max_{(s,t) \in S_{xy}} \{g(s, t)\}$$

Min Filter:

$$\hat{f}(x, y) = \min_{(s,t) \in S_{xy}} \{g(s, t)\}$$

Max filter is good for pepper noise and
min is good for salt noise

Nelineárne filtre

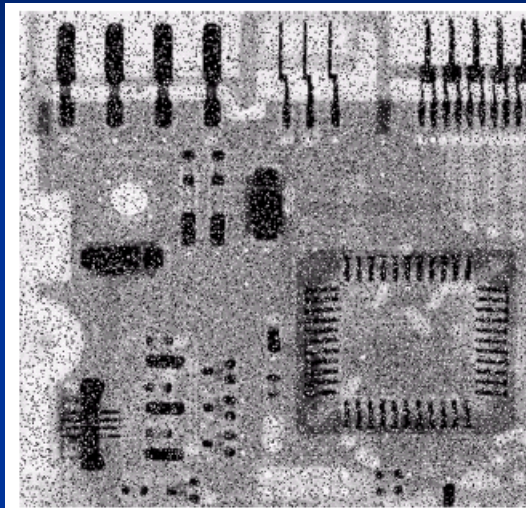
Midpoint Filter:

$$\hat{f}(x, y) = \frac{1}{2} \left[\max_{(s,t) \in \mathcal{S}_{xy}} \{g(s,t)\} + \min_{(s,t) \in \mathcal{S}_{xy}} \{g(s,t)\} \right]$$

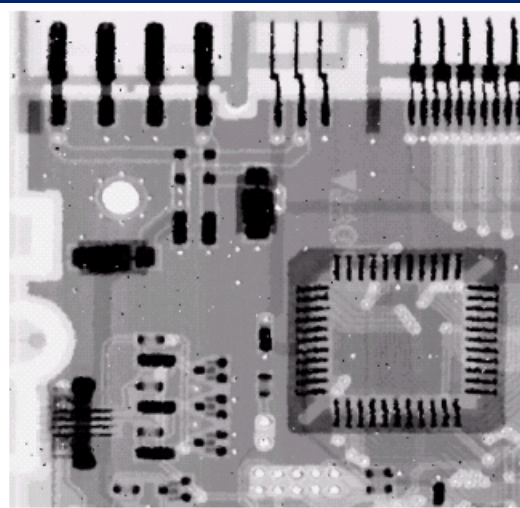
Good for random Gaussian and uniform noise

Noise Removal Examples

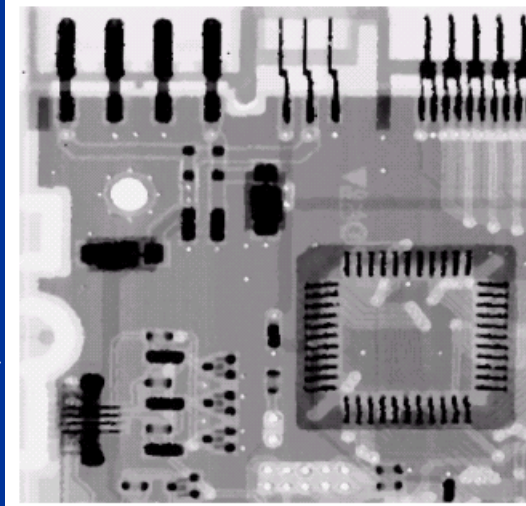
Image
Corrupted
By Salt And
Pepper Noise



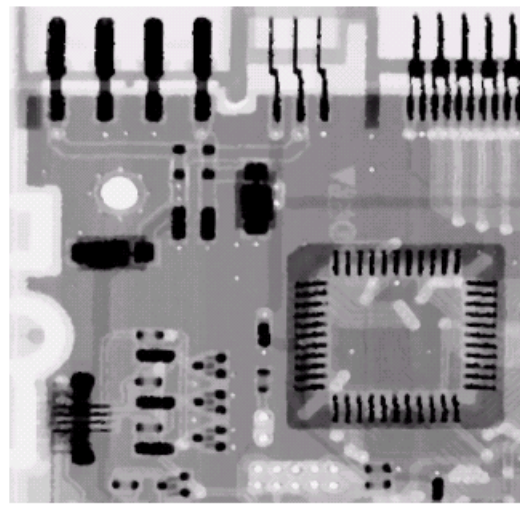
Result of 1
Pass With A
3*3 Median
Filter



Result of 2
Passes With
A 3*3 Median
Filter



Result of 3
Passes With
A 3*3 Median
Filter



Noise Removal Examples (cont...)

Image
Corrupted
By Pepper
Noise

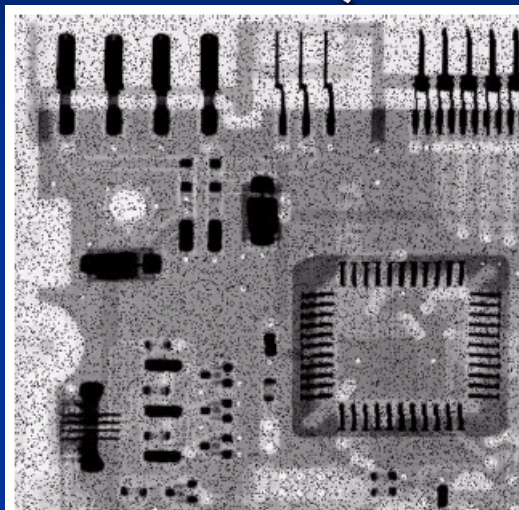
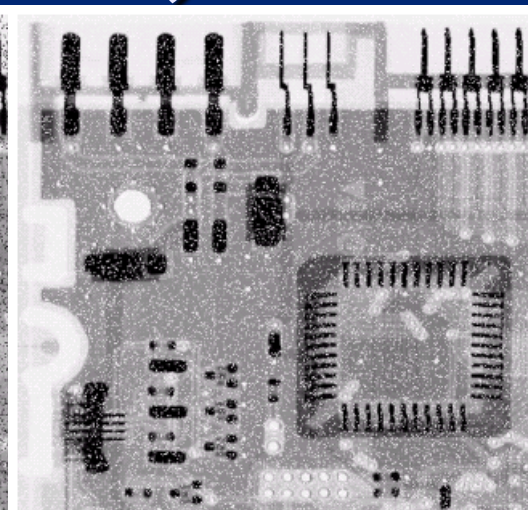
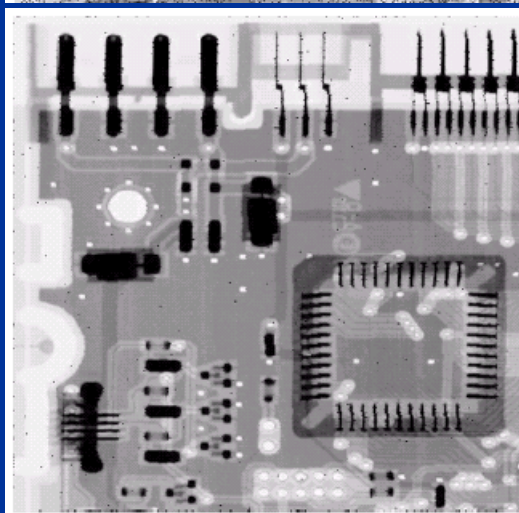


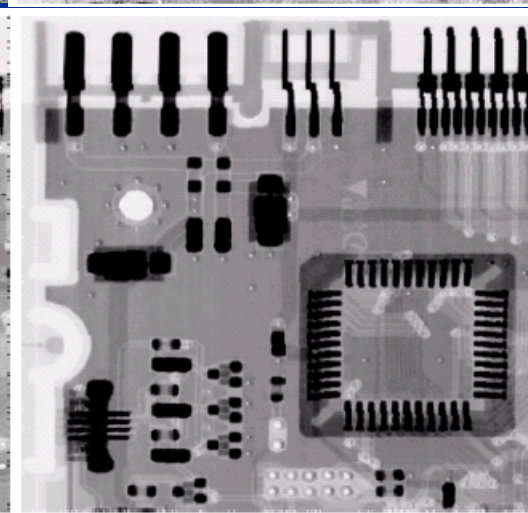
Image
Corrupted
By Salt
Noise



Result Of
Filtering
Above
With A 3*3
Max Filter



Result Of
Filtering
Above
With A 3*3
Min Filter



Noise Removal Examples

(cont...)

Image
Corrupted
By Uniform
Noise

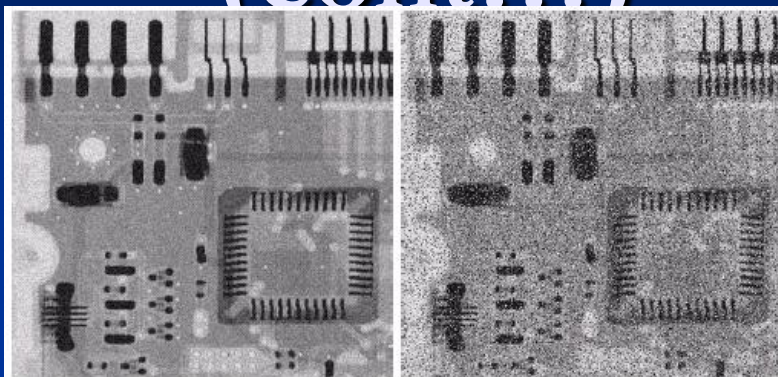
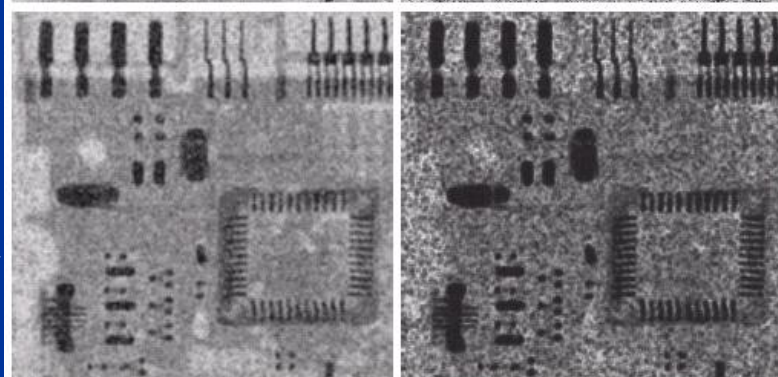


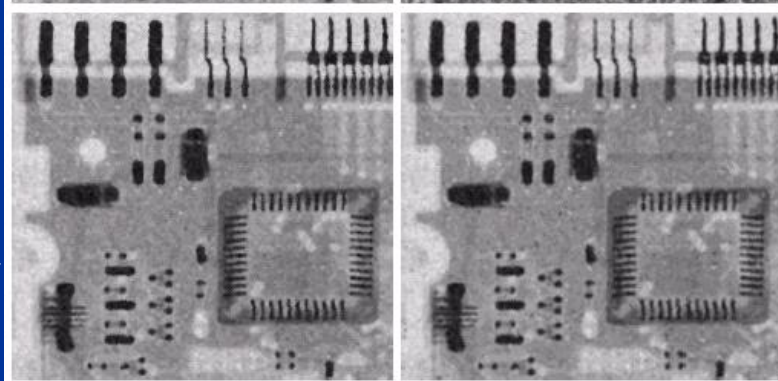
Image Further
Corrupted
By Salt and
Pepper Noise

Filtered By
 5×5 Arithmetic
Mean Filter



Filtered By
 5×5 Geometric
Mean Filter

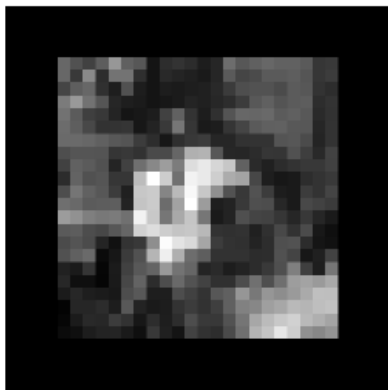
Filtered By
 5×5 Median
Filter



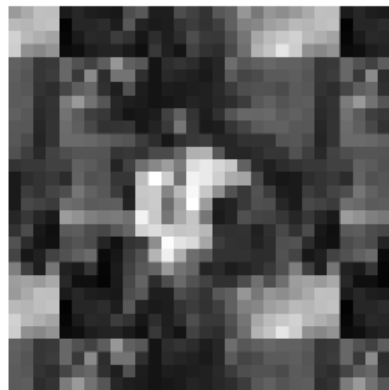
Filtered By
 5×5 Alpha-Trimmed
Mean Filter

Riešenie okrajov

- Nula
- Konštanta – pevne daná farba okrajov
- Clamp - Opakovanie okrajových pixelov
- Wrap – Opakovanie cyklicky
- Zrkadlovo



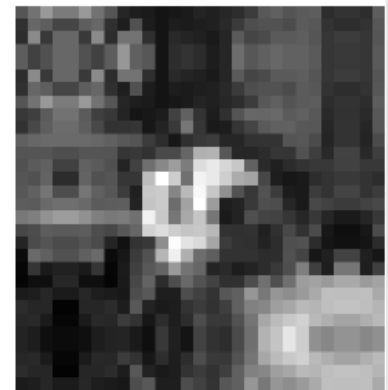
zero



wrap



clamp



mirror

Separabilné filtre

- Konvolúcia maskou $k \times k$ stojí k^2 operácií pre každý pixel
- Niektoré masky môžu byť rozložené na 1D masky – horizontálne a vertikálne

$$K = v h^T$$

- cena sa zníži na $2k$ operácií pre každý pixel

Separabilné filtre

$$\frac{1}{K^2}$$

1	1	...	1
1	1	...	1
⋮	⋮	1	⋮
1	1	...	1

$$\frac{1}{16}$$

1	2	1
2	4	2
1	2	1

$$\frac{1}{256}$$

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

$$\frac{1}{8}$$

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

$$\frac{1}{4}$$

1	-2	1
-2	4	-2
1	-2	1

$$\frac{1}{K}$$

1	1	...	1
---	---	-----	---

$$\frac{1}{4}$$

1	2	1
---	---	---

$$\frac{1}{16}$$

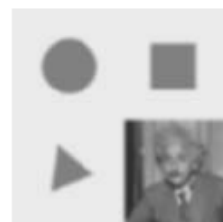
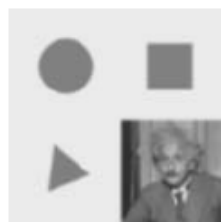
1	4	6	4	1
---	---	---	---	---

$$\frac{1}{2}$$

-1	0	1
----	---	---

$$\frac{1}{2}$$

1	-2	1
---	----	---



(a) box, $K = 5$

(b) bilinear

(c) "Gaussian"

(d) Sobel

(e) corner

Ako zistíme či je kernel separabilný?

- Použitím SVD – singulárny rozklad matice

$$K = \sum_i \sigma_i \mathbf{u}_i \mathbf{v}_i^T$$

- Ak iba 1. singulárna hodnota je nenulová kernel je separabilný

- $\sqrt{\sigma_0} \mathbf{u}_0$ vertikálny kernel

- $\sqrt{\sigma_0} \mathbf{v}_0^T$ horizontálny kernel

Bilateral filtering

- Hodnota výsledného pixla závisí od váženej kombinácie okolitých pixlov

$$g(i, j) = \frac{\sum_{k,l} f(k, l)w(i, j, k, l)}{\sum_{k,l} w(i, j, k, l)}$$

- váhový koeficient w závisí od *domain kernela* (Gaussian)

$$d(i, j, k, l) = \exp\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_d^2}\right)$$

- a na dátach závislom *range kernely*

$$r(i, j, k, l) = \exp\left(-\frac{\|f(i, j) - f(k, l)\|^2}{2\sigma_r^2}\right)$$

- Prenásobením dostaneme na dátach závislú bilaterálnu váženú funkciu

$$w(i, j, k, l) = \exp\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_d^2} - \frac{\|f(i, j) - f(k, l)\|^2}{2\sigma_r^2}\right)$$

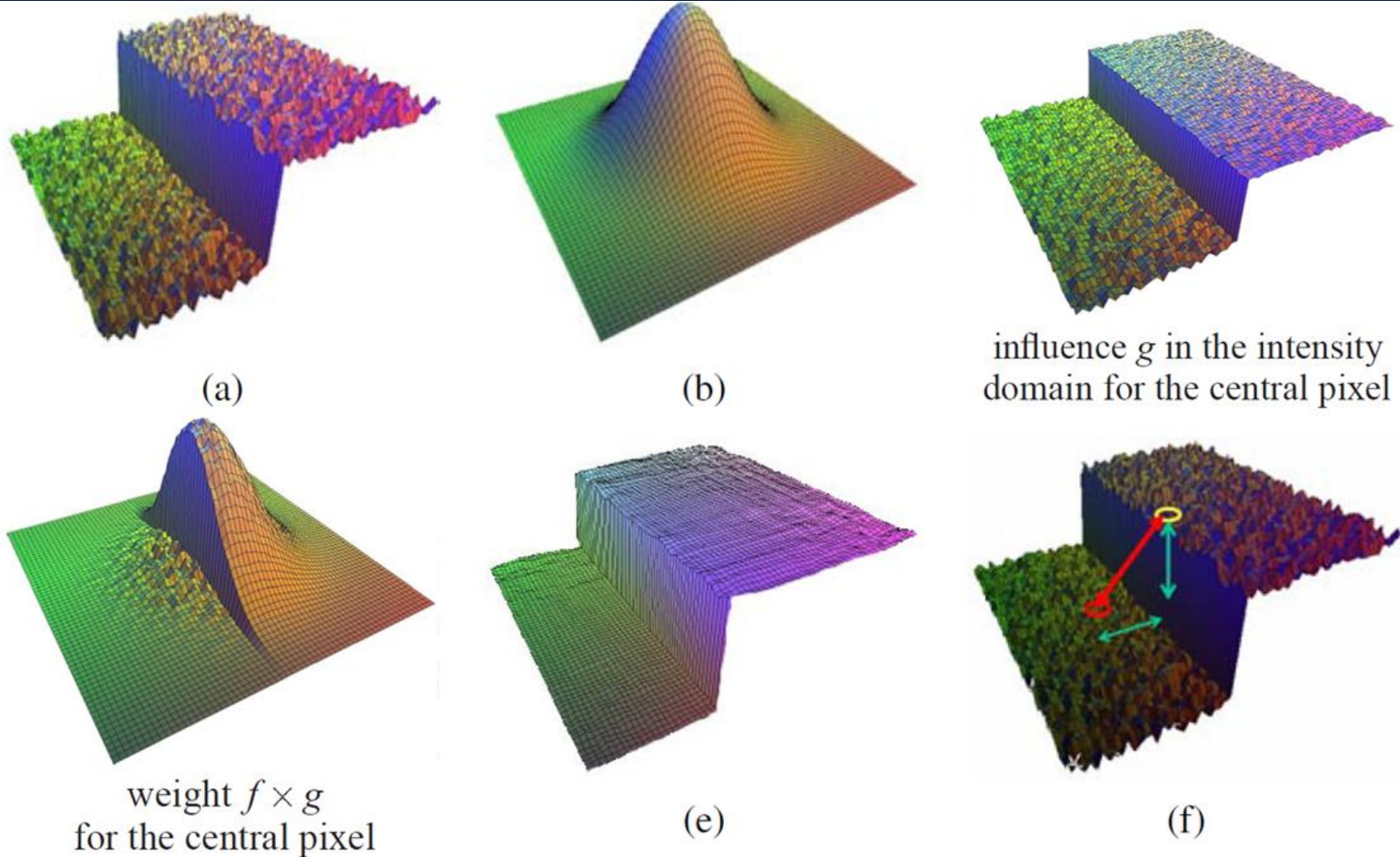
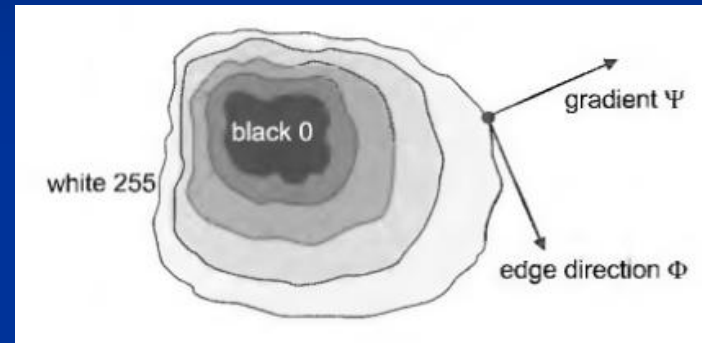


Figure 3.20 Bilateral filtering (Durand and Dorsey 2002) © 2002 ACM: (a) noisy step edge input; (b) domain filter (Gaussian); (c) range filter (similarity to center pixel value); (d) bilateral filter; (e) filtered step edge output; (f) 3D distance between pixels.

Detekciu hrán

- robíme cez gradientné operátory
- Veľkosť gradientu a smer



- Ak sa intenzity v danom okolí bodu príliš nelíšia - pravdepodobne tam nie je hrana
- Ak sa líšia - bod môže patriť hrane

Tri druhy gradientných operátorov

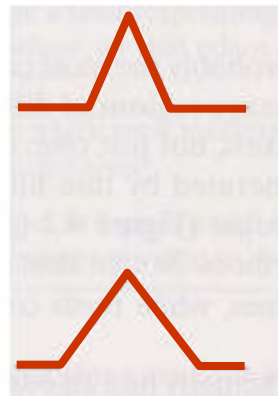
1. aproximujúce **prvú deriváciu** obrazovej funkcie, ktoré sú založené na **diferenciách** – obvykle majú viacero masiek, okrem Laplaciánu, orientácia sa určuje ako najlepšia zhoda viacerých vzorov
2. založené na **zero-crossing** druhej derivácie obrazovej funkcie (Marr-Hildrethovej operátor, Cannyho operátor)
3. založené na predstave obrazovej funkcie mapovanej parametrickým modelom hrán

Typy hrán

- skutočné hrany
- šum



schod
rampa

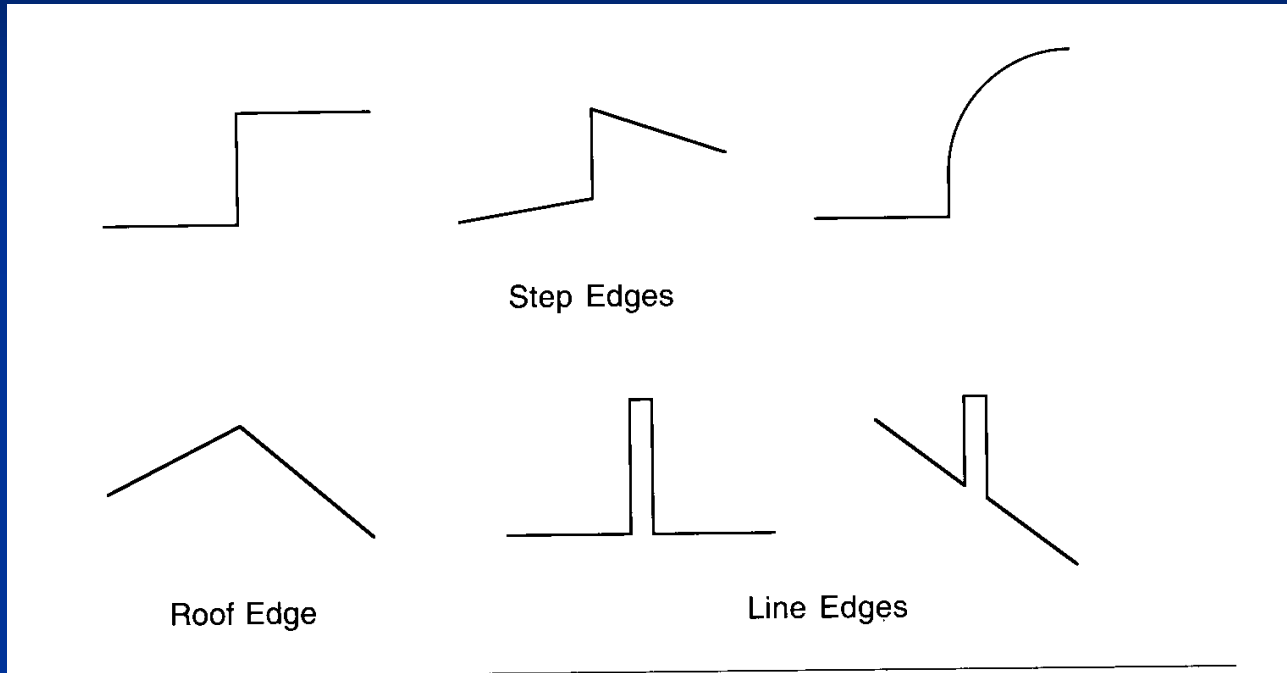


strecha



čiara
hrebeň

Typy hrán



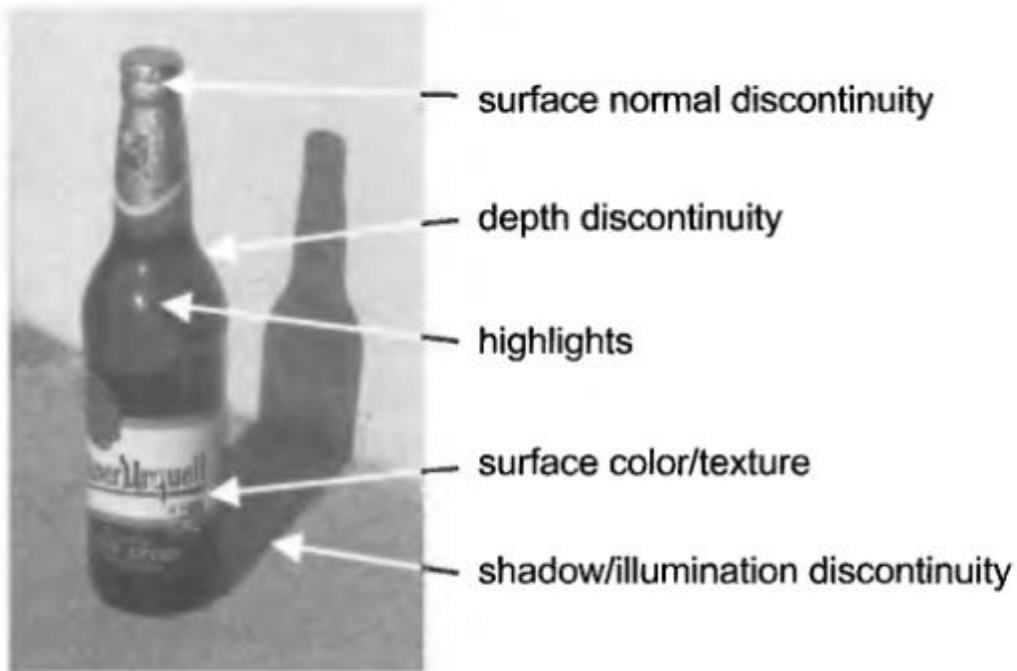


Figure 5.16: Origin of edges, i.e., physical phenomena in the image formation process which lead to edges in images.



Figure 5.17: Detected edge elements.

Metódy hľadania hrán

- konvolučné masky
- diskretná aproximácia diferenciálnych operátorov (miera zmeny intenzity)
- Informácia o:
 - existencia ✓
 - orientácia ?

Diferencovanie 2D

$$\frac{\partial f(x, y)}{\partial x} \approx \frac{f(x_{n+1}, y_m) - f(x_n, y_m)}{\Delta x}$$

$$\frac{\partial f(x, y)}{\partial y} \approx \frac{f(x_n, y_{m+1}) - f(x_n, y_m)}{\Delta \bar{y}}$$

$$\frac{\partial I(x, y)}{\partial x} = \frac{I(x+1, y) - I(x-1, y)}{2}$$

$$\frac{\partial I(x, y)}{\partial y} = \frac{I(x, y+1) - I(x, y-1)}{2}$$

$$\begin{bmatrix} 1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$\frac{\partial I}{\partial x} = I * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

$$\frac{\partial I}{\partial y} = I * \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

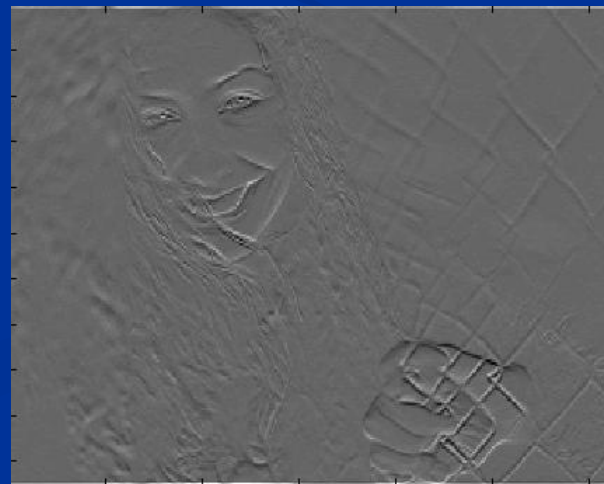
Diferencovanie



I



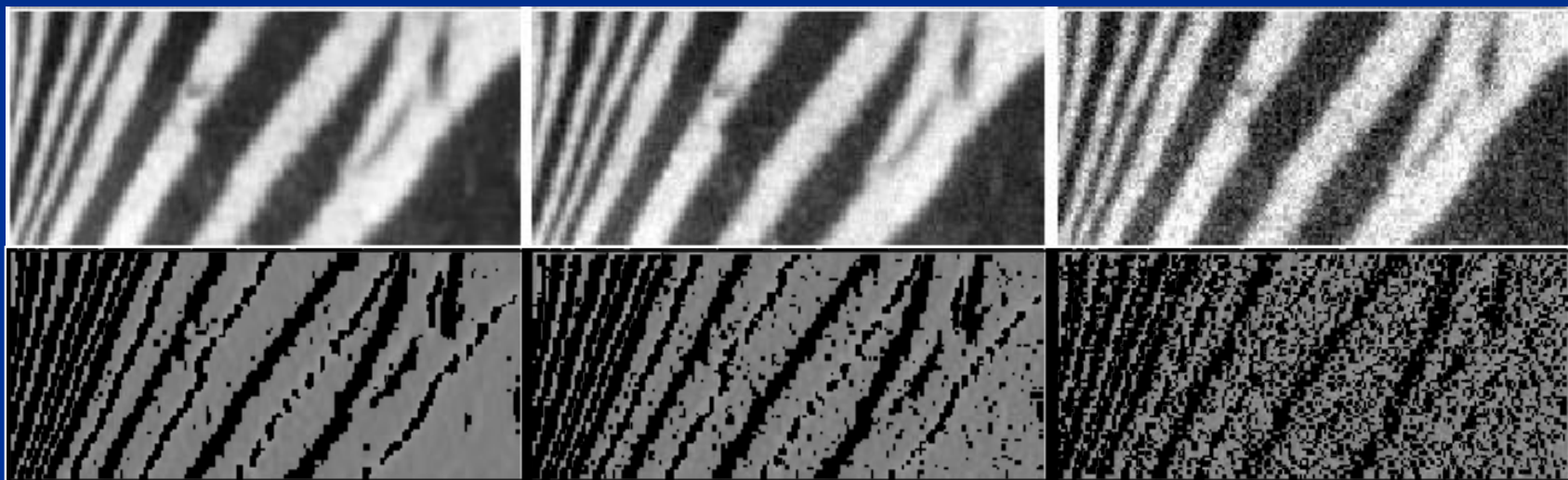
$$I_x = I * \begin{bmatrix} 1 & -1 \end{bmatrix}$$



$$I_y = I * \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Ktorý obrázok je I_x ?

Diferencovanie a šum



Vyhladenie

prah 20



prah 50

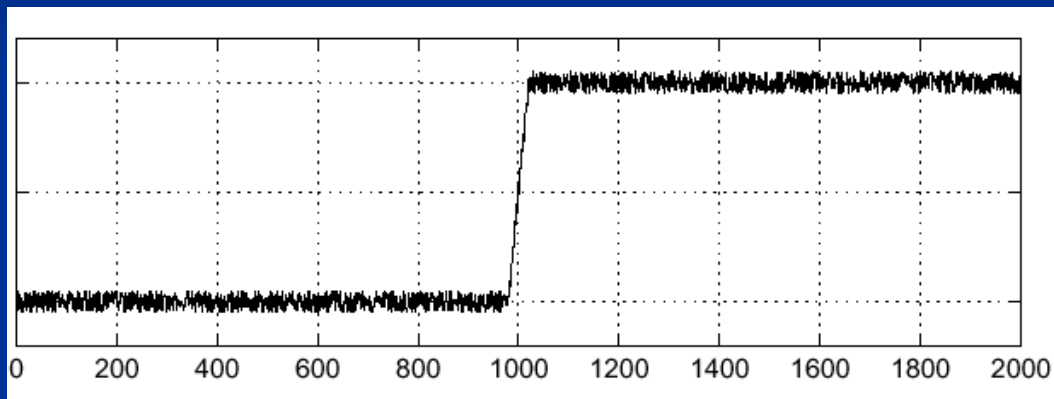


originál

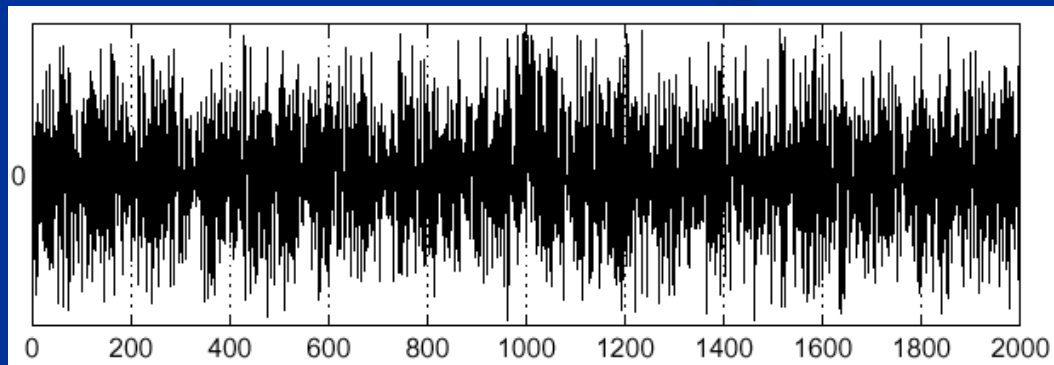
Gaussovské vyhladenie

Následky šumu

$f(x)$

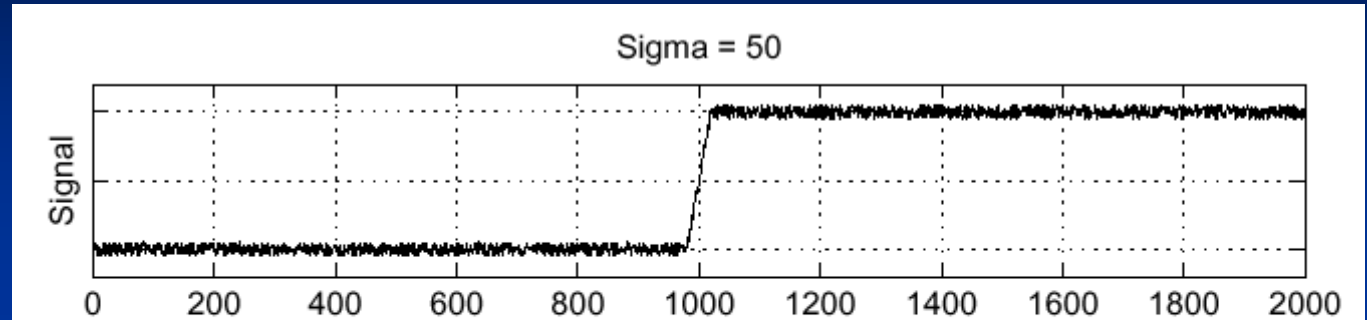


$\frac{d}{dx} f(x)$



Vyhladenie

f



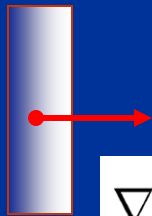
h

$h \star f$

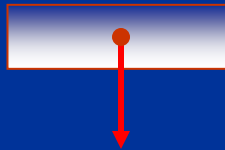
$\frac{\partial}{\partial x}(h \star f)$

Gradient

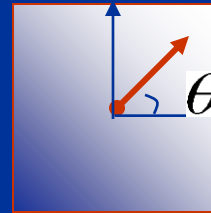
- Gradient: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$
- Smer – najväčšia zmena intenzity



$$\nabla f = \left[\frac{\partial f}{\partial x}, 0 \right]$$



$$\nabla f = \left[0, \frac{\partial f}{\partial y} \right]$$



$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

Smer gradientu:

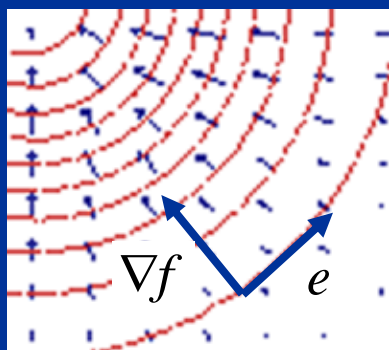
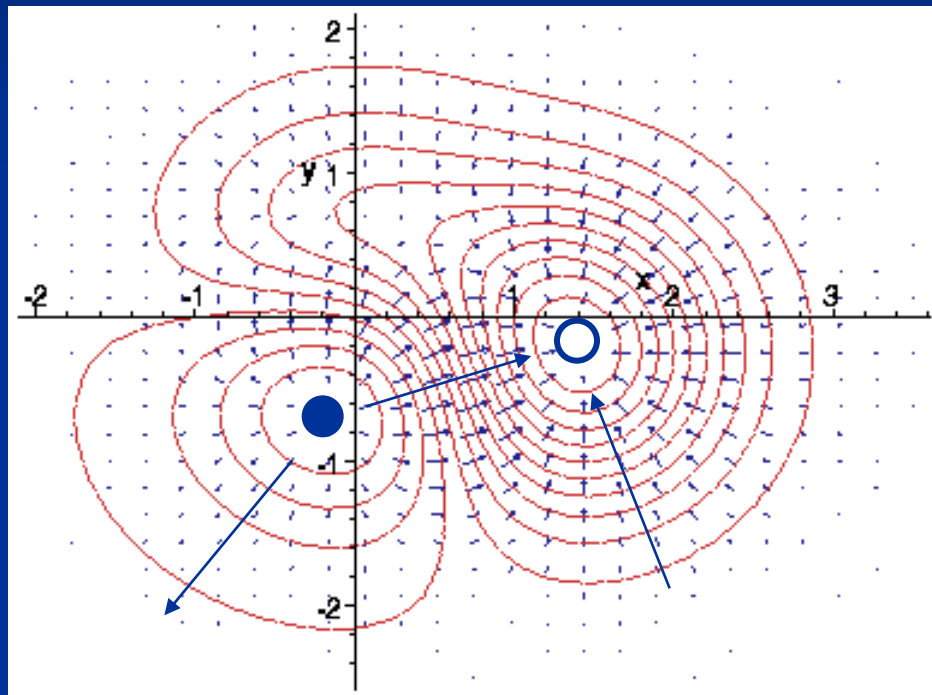
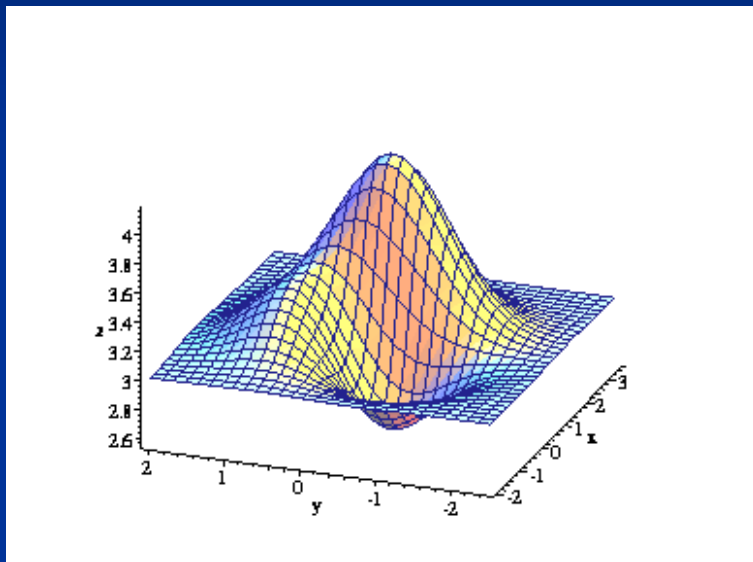
$$\theta = \tan^{-1} \left(\frac{\partial f / \partial y}{\partial f / \partial x} \right)$$

Veľkosť gradientu:

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

$$\|\nabla f\| \approx |G_x| + |G_y|$$

Gradient / hrany



Sila (dôležitosť) hrany = veľkosť gradientu
Smer hrany = smer gradientu $- 90^\circ$

Diferenčné gradientné operátory

- možno vyjadriť pomocou konvolučných masiek; dostaneme Robertsov, Prewittov, Sobelov, Robinsonov a Kirschov operátor.
- Hlavnou nevýhodou konvolučných hranových detektorov je ich závislosť na škálovaní a citlivosť na šum.
- Väčšinou existuje málo rozumných dôvodov pre výber konkrétnej veľkosti okolia operátora.

Gradient



Roberts

- Najjednoduchšie masky

-1	0
0	1

Len body hrán

Nie orientácia

Vhodné pre binárne obrazy

Nevýhody:

- Veľká citlivosť na šum

- Nepresná lokalizácia

- Málo bodov na aproximáciu gradientu

0	-1
1	0

Sobel

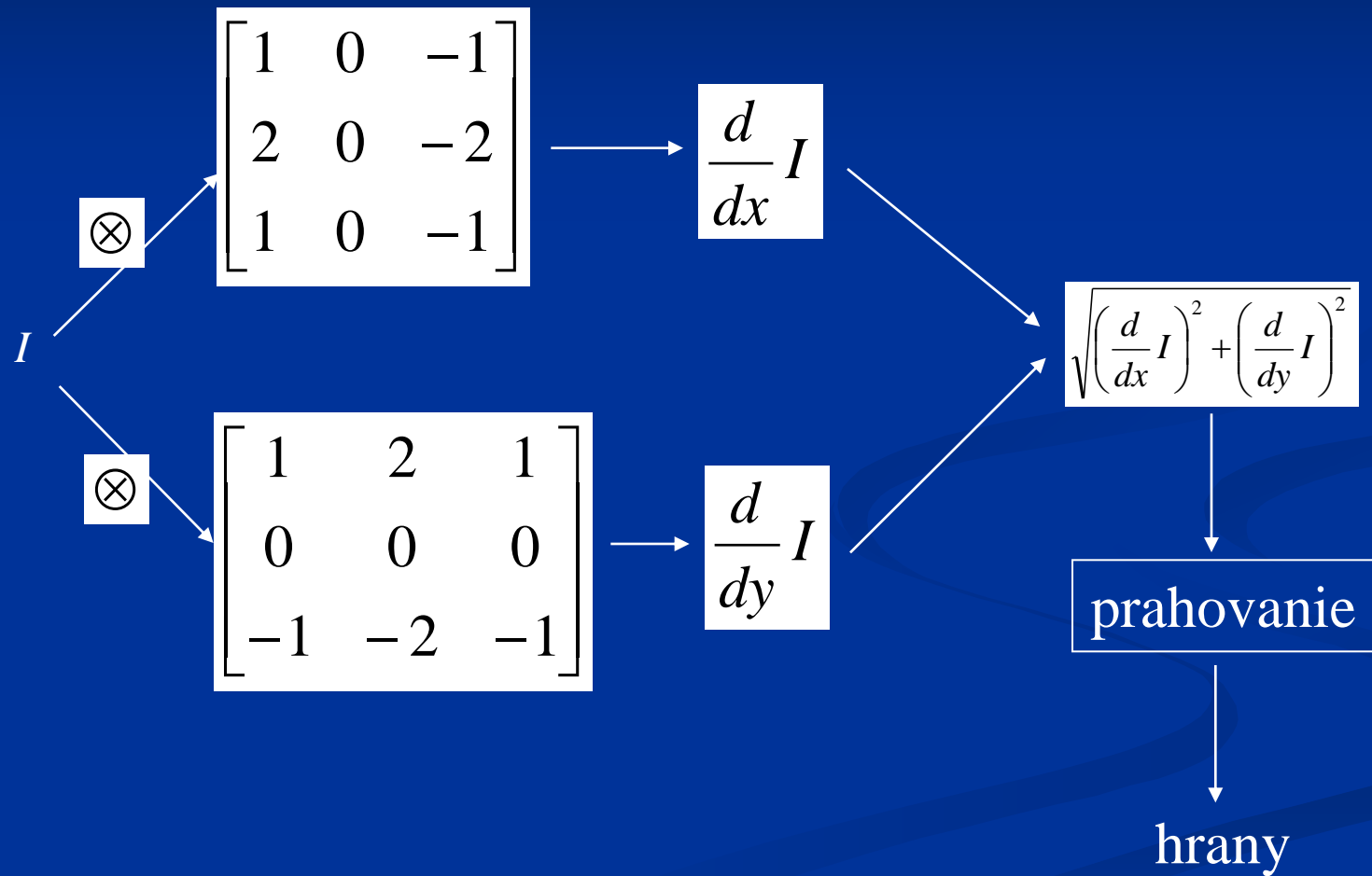
- Hľadá horizontálne a vertikálne hrany
- Konvolučné masky:

$$y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

- Dlhší výpočtový čas v porovnaní s Robertsom
- Menej citlivý na šum

Sobel



Sobel

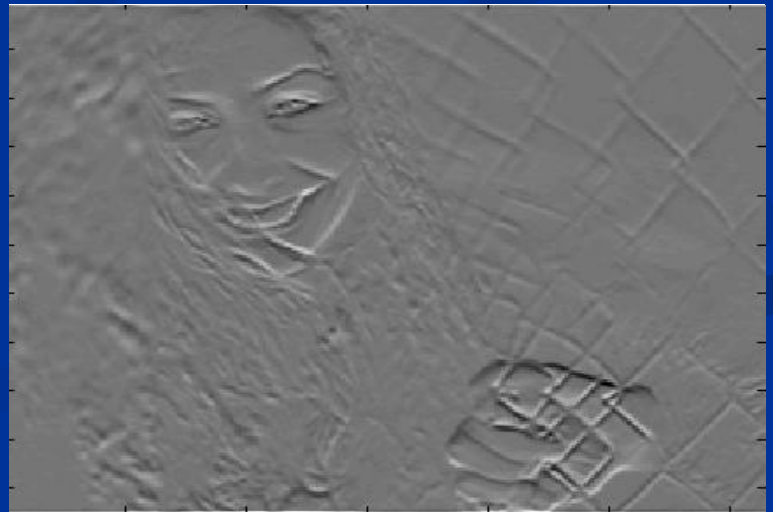
$$\frac{d}{dx} I$$



I



$$\frac{d}{dy} I$$



Sobel

$$E = \sqrt{\left(\frac{d}{dx} I\right)^2 + \left(\frac{d}{dy} I\right)^2}$$

I



$E \geq \text{Threshold} = 100$



Prewitt

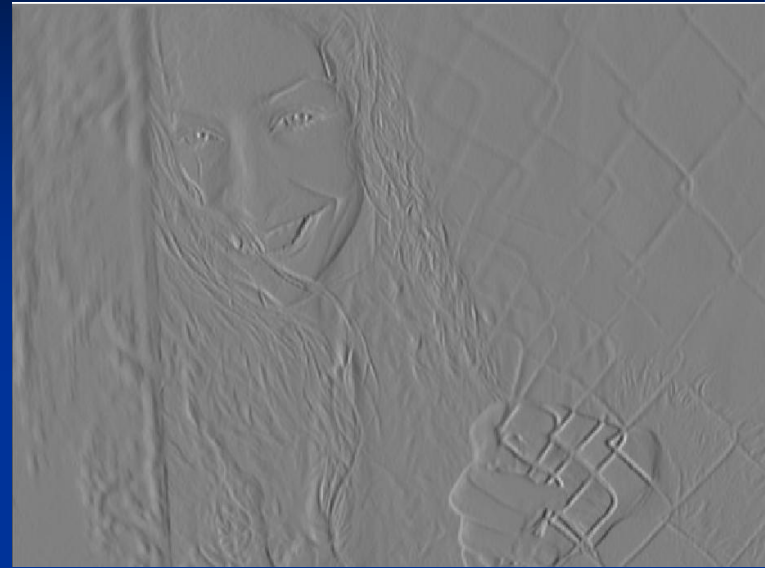
- Podobne ako Sobel
- Masky:

$$y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Prewitt

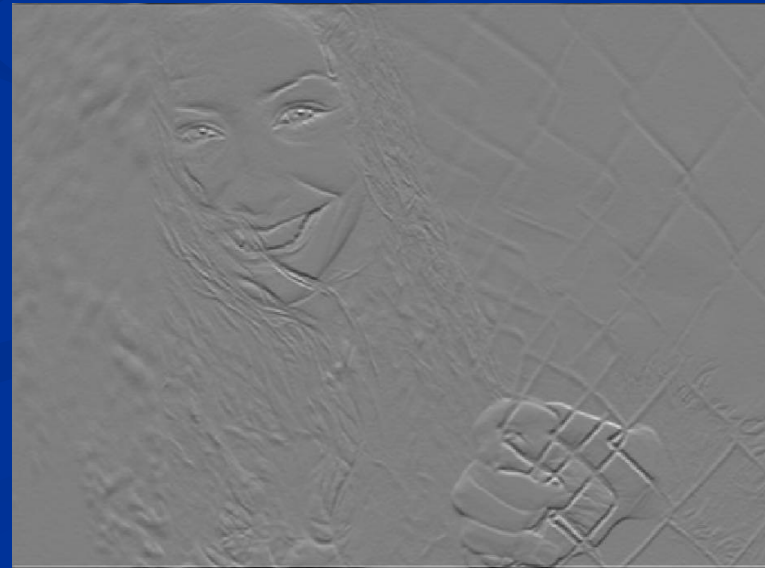
$$\frac{d}{dx} I$$



I



$$\frac{d}{dy} I$$



Prewitt

$$E = \sqrt{\left(\frac{d}{dx} I\right)^2 + \left(\frac{d}{dy} I\right)^2}$$

I



$E \geq 100$



Kirsch - kompas operátor

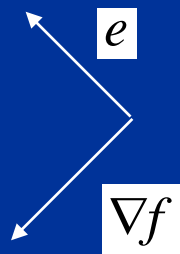
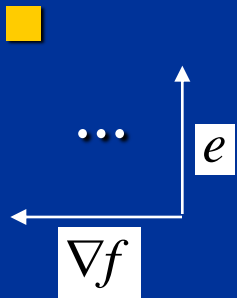
- Rotujúca maska
- Smery: 0° , 45° , 90° , 135° , ...
- Sila hrany – maximum cez jednotlivé masky
- Smer hrany – maska dávajúca maximum

$$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$$

$$\begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$$

$$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

$$\begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$



Robinson

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}$$

Robinson Kirsch Prewitt Sobel

Gradient
direction

East
 H_1

$$\begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Northeast
 H_2

$$\begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$$

$$\begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}$$

North
 H_3

$$\begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Northwest
 H_4

$$\begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

West
 H_5

$$\begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Southwest
 H_6

$$\begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$$

South
 H_7

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Southeast
 H_8

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}$$

Zisťovanie prechodu nulou (Zero-crossing) druhej derivácie

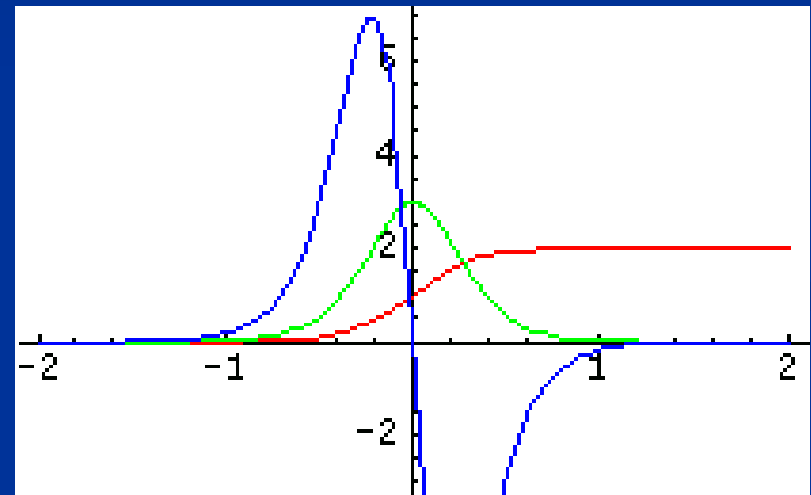
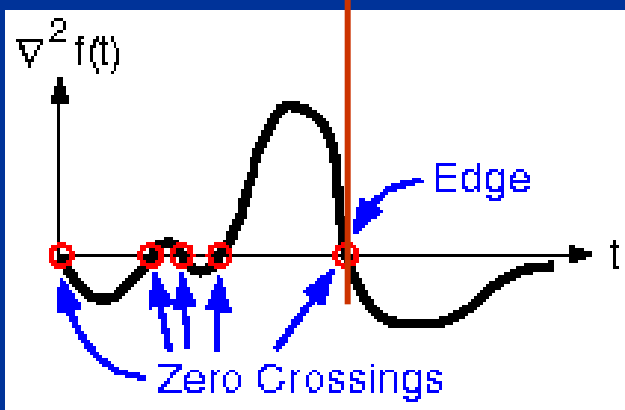
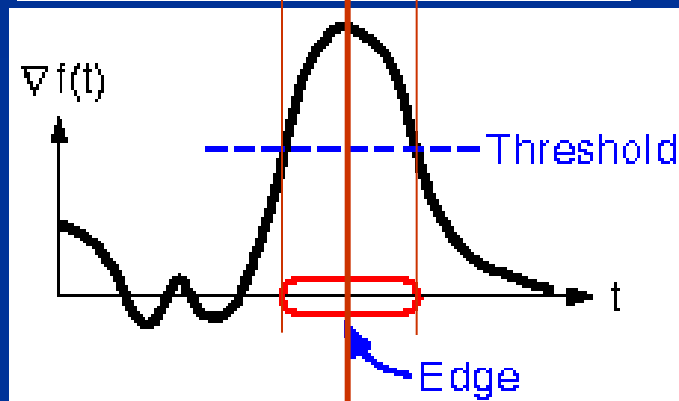
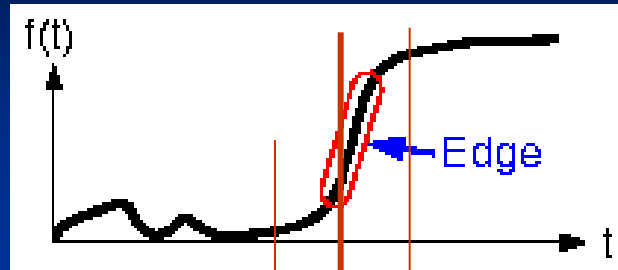
- je robustnejšia metóda ako gradientné operátory malej veľkosti – počíta sa ako Laplacián z Gaussiánov (LoG) alebo ako diferenciacia Gaussiánov (DoG).
- prečo druhá derivácia ?
 - lebo prechod nulou sa zisťuje ľahšie ako extrém pri prvej derivácii
- ako môžeme počítať druhú deriváciu robustne ?
 - najprv obraz vyhladiť (na vyhladenie sa použije Gaussovský filter) a potom urobiť druhú deriváciu

vzorce – Gaussián alebo Gaussovský filter $G(x,y)$ je daný ako

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

- kde x a y sú obrazové súradnice a σ je štandardná odchýlka súvisiaceho rozdelenia pravdepodobnosti.
- Parameter σ je jediným parametrom filtra a je úmerný veľkosti okolia, v ktorom filter pôsobí.
- rozmer konvolučných masiek narastá s rastúcim σ (väčšie σ znamená, že len výraznejšie zmeny budú zachytené)
- Obrazové body, ktoré sú vzdialenejšie od centra filtra majú menší vplyv, tie, čo sú vzdialené viac ako 3σ majú zanedbateľný vplyv.

Druhá derivácia



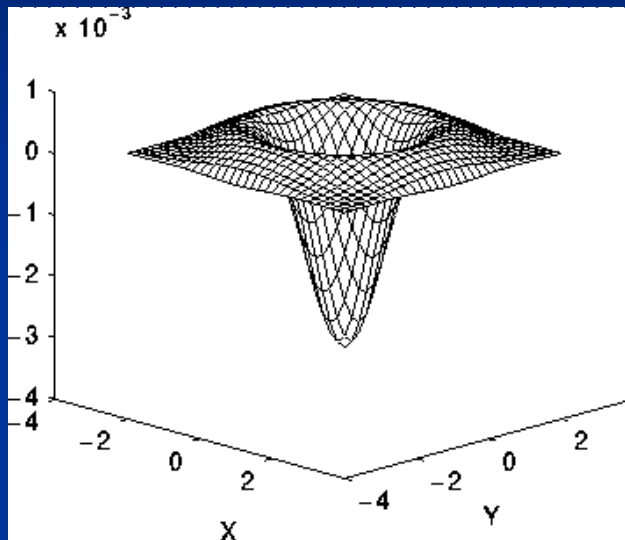
Laplacián

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Konvolúcia [1, -2, 1]

- je izotropný operátor, na všetky zmeny reaguje rovnako

Laplacián



Nevýhody:

Veľmi citlivý na šum

Produkujú dvojité hrany

Neurčuje smer hrany

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

Laplacián



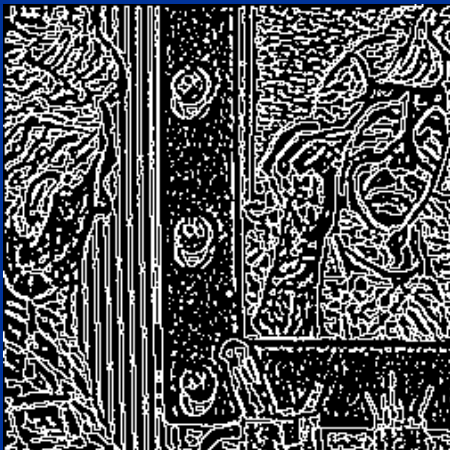
3x3



5x5



7x7



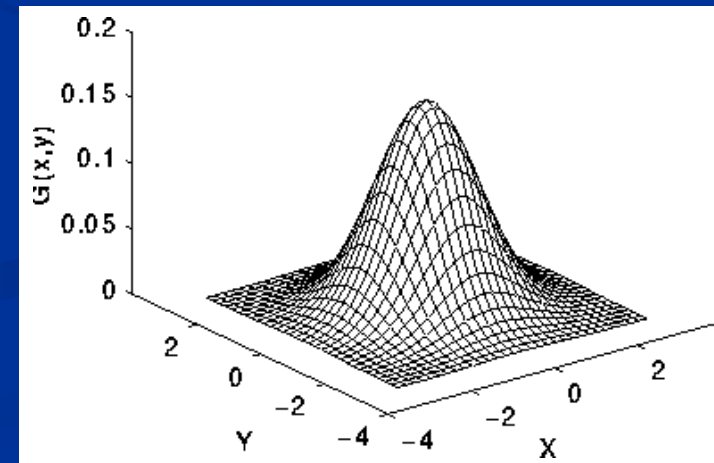
Laplacián Gaussiánu

- Marr – Hildreth operátor, LoG operátor
- Vyhladenie pomocou 2D Gaussiánu
- Následná aplikácia Laplaciánu

$$S = G_{\sigma} \otimes I$$

$$G_{\sigma} = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\nabla^2 S = \frac{\partial^2}{\partial x^2} S + \frac{\partial^2}{\partial y^2} S$$

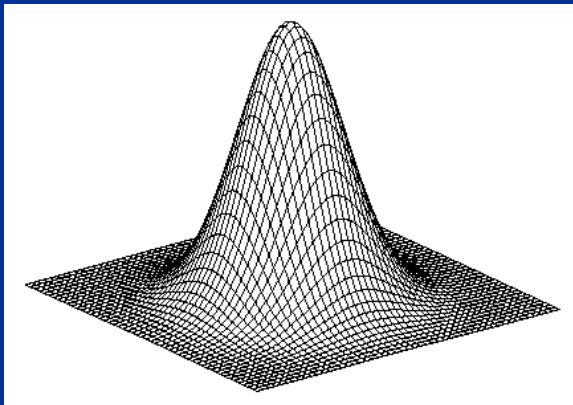


Laplacián Gaussiánu

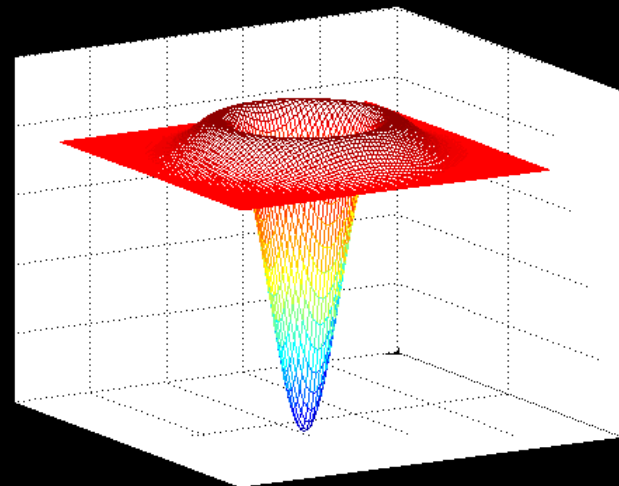
$$E = (I * G) * L = I * (G * L)$$

$$\nabla^2 S = \nabla^2 (G_\sigma * I) = \nabla^2 G_\sigma * I$$

Gaussian



Laplacian of Gaussian



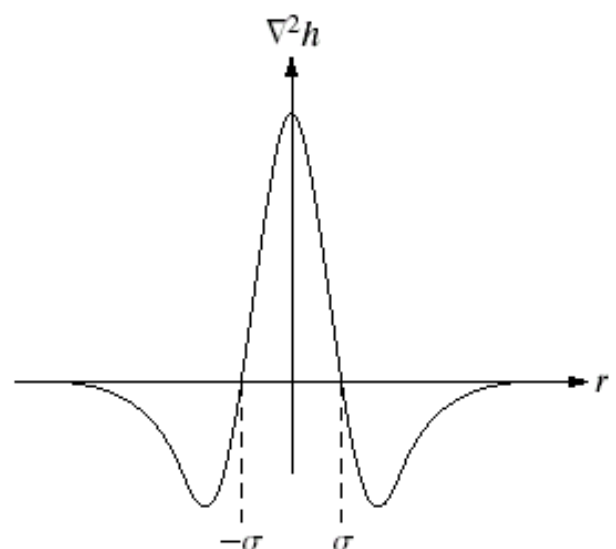
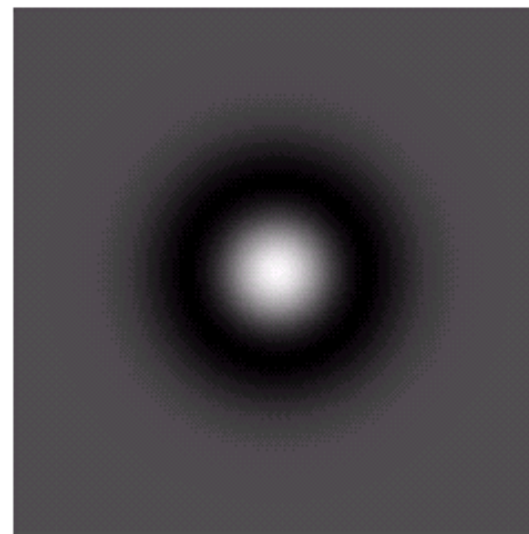
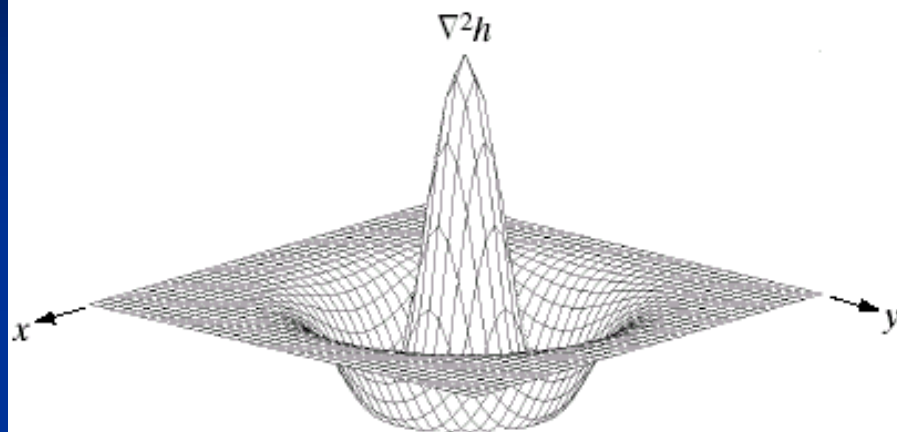
$$\nabla^2 [G(x, y, \sigma) * f(x, y)].$$

$$[\nabla^2 G(x, y, \sigma)] * f(x, y)$$

- deriváciu Gaussovho filtra v hranatej zátvorke si môžeme predvypočítat' analyticky, pretože nezávisí na obrazovej funkcii.
- substitúciou $r^2 = x^2 + y^2$ prevedieme Gaussián na 1D prípad, vypočítame 2. deriváciu a po spätnom využití substitúcie dostaneme konvolučnú masku

Marr-Hildrethovej operátora

$$h(x, y) = c \left(\frac{x^2 + y^2 - \sigma^2}{\sigma^4} \right) e^{-\frac{x^2 + y^2}{2\sigma^2}}$$



0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

a b
c d

FIGURE 10.14
Laplacian of a Gaussian (LoG).
(a) 3-D plot.
(b) Image (black is negative, gray is the zero plane, and white is positive).
(c) Cross section showing zero crossings.
(d) 5×5 mask approximation to the shape of (a).

príklad diskkrétnej aproximácie na 17x17

0	0	0	0	0	0	-1	-1	-1	-1	-1	0	0	0	0	0	0
0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0
0	0	-1	-1	-1	-2	-3	-3	-3	-3	-3	-2	-1	-1	-1	0	0
0	0	-1	-1	-2	-3	-3	-3	-3	-3	-3	-3	-2	-1	-1	0	0
0	-1	-1	-2	-3	-3	-3	-2	-3	-2	-3	-3	-3	-2	-1	-1	0
0	-1	-2	-3	-3	-3	0	2	4	2	0	-3	-3	-3	-2	-1	0
-1	-1	-3	-3	-3	0	4	10	12	10	4	0	-3	-3	-3	-1	-1
-1	-1	-3	-3	-2	2	10	18	21	18	10	2	-2	-3	-3	-1	-1
-1	-1	-3	-3	-3	4	12	21	24	21	12	4	-3	-3	-3	-1	-1
-1	-1	-3	-3	-2	2	10	18	21	18	10	2	-2	-3	-3	-1	-1
-1	-1	-3	-3	-3	0	4	10	12	10	4	0	-3	-3	-3	-1	-1
0	-1	-2	-3	-3	-3	0	2	4	2	0	-3	-3	-3	-2	-1	0
0	-1	-1	-2	-3	-3	-3	-2	-3	-2	-3	-3	-3	-2	-1	-1	0
0	0	-1	-1	-2	-3	-3	-3	-3	-3	-3	-3	-2	-1	-1	0	0
0	0	-1	-1	-1	-2	-3	-3	-3	-3	-3	-2	-1	-1	-1	0	0
0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0
0	0	0	0	0	0	-1	-1	-1	-1	-1	0	0	0	0	0	0

Rozdiel Gaussiánov

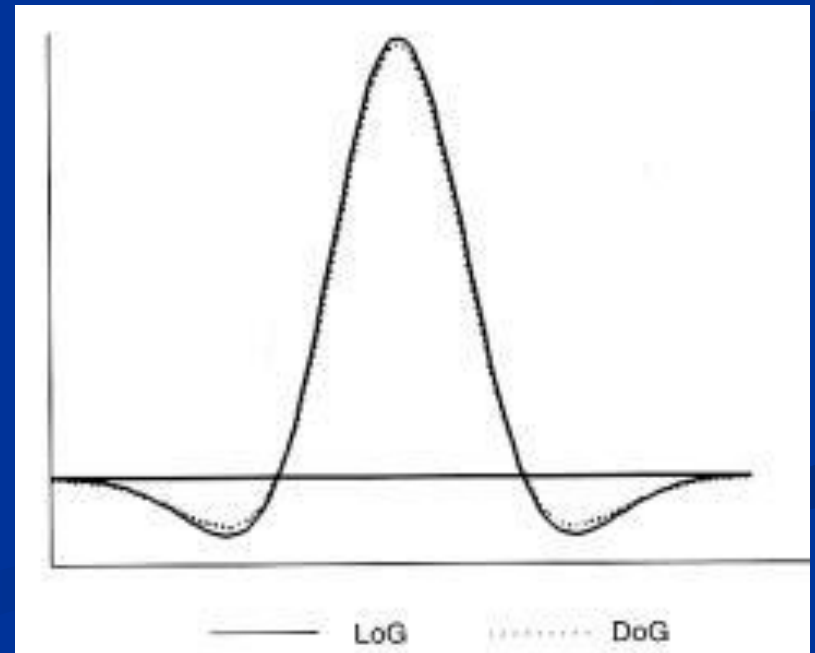
Aproximácia LoG operátora – DoG

Rozdiel Gaussiánov s dostatočne rozdielnymi σ

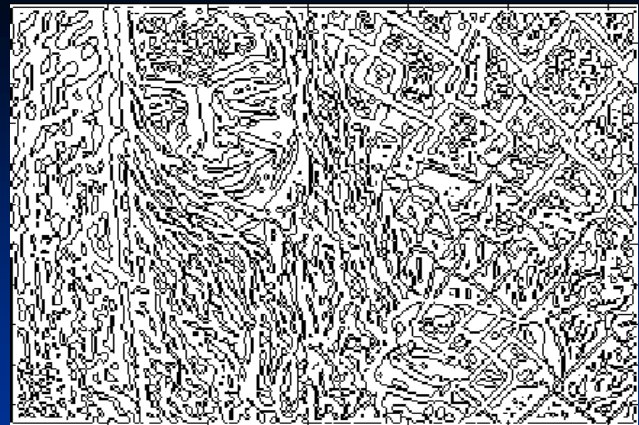
$$\sigma_1 / \sigma_2 = 1.6$$

LoG ($\sigma = 12.35$)

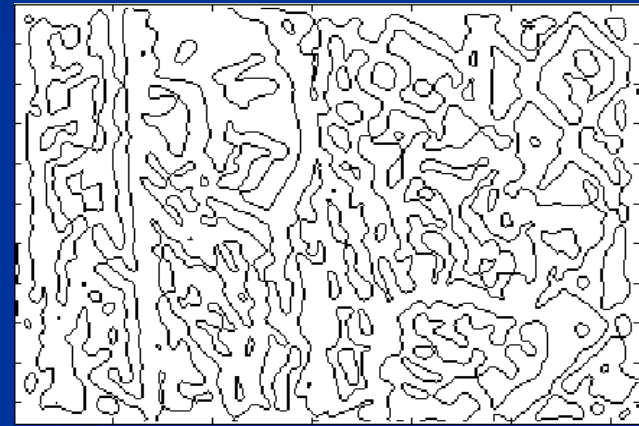
DoG ($\sigma_1 = 10, \sigma_2 = 16$)



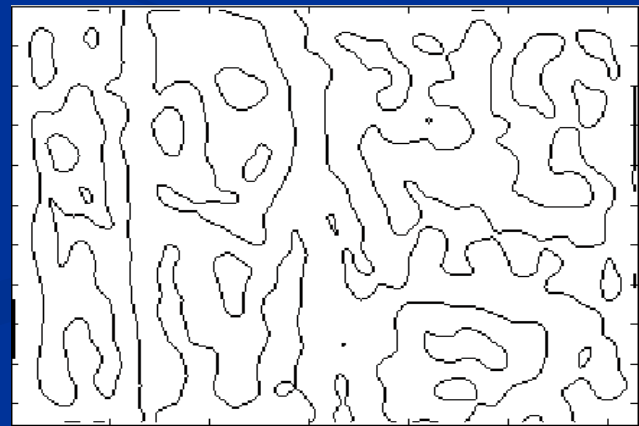
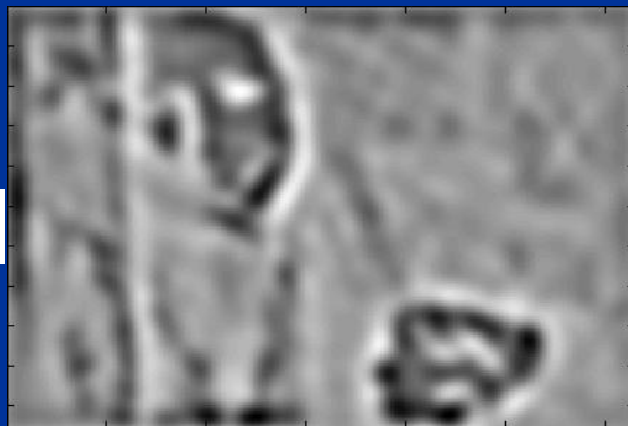
$\sigma = 1$



$\sigma = 3$



$\sigma = 6$



Cannyho hranový operátor

- je optimálny pre ostré hrany zašumené bielym šumom.
- Kritérium optimality je založené na 3 požiadavkách:
 1. detekovať každú hranu, ale vynechať zdanlivé hrany,
 2. minimálna lokalizačná chyba medzi skutočnou a detekovanou hranou
 3. na potlačení viacnásobných odoziev na jednu hranu.

Canny

- 1) Vyhladenie Gaussiánom
- 2) Gradientný operátor
 - Veľkosť gradientu
 - Smer gradientu
- 3) Výber maxím v danom smere
- 4) Prahovanie dvoma prahmi

Original



Canny



Canny

- Vyhladenie Gaussiánom

$$S = G_\sigma * I$$

- Gradientný operátor (Sobel)

$$\nabla S = \left[\frac{\partial}{\partial x} S \quad \frac{\partial}{\partial y} S \right]^T = [S_x \quad S_y]^T$$

$$G_\sigma = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

- Veľkosť gradientu

$$|\nabla S| = \sqrt{S_x^2 + S_y^2}$$

- Smer gradientu

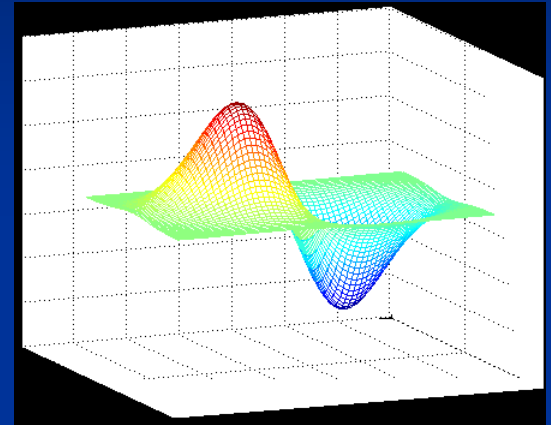
$$\theta = \tan^{-1} \frac{S_y}{S_x}$$

Canny

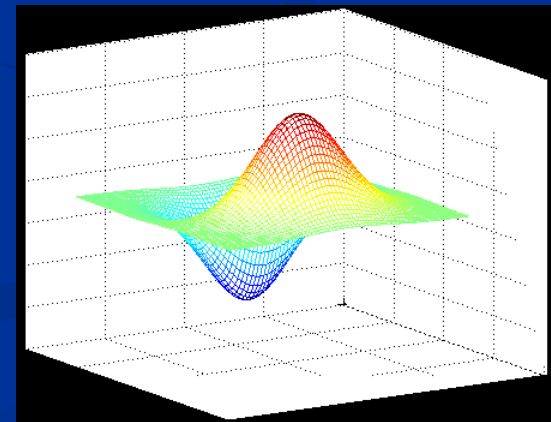
$$\nabla S = \nabla(G_\sigma * I) = \nabla G_\sigma * I$$

$$\nabla G_\sigma = \left[\begin{array}{cc} \frac{\partial G_\sigma}{\partial x} & \frac{\partial G_\sigma}{\partial y} \end{array} \right]^T$$

$$\nabla S = \left[\begin{array}{cc} \frac{\partial G_\sigma * I}{\partial x} & \frac{\partial G_\sigma * I}{\partial y} \end{array} \right]^T$$

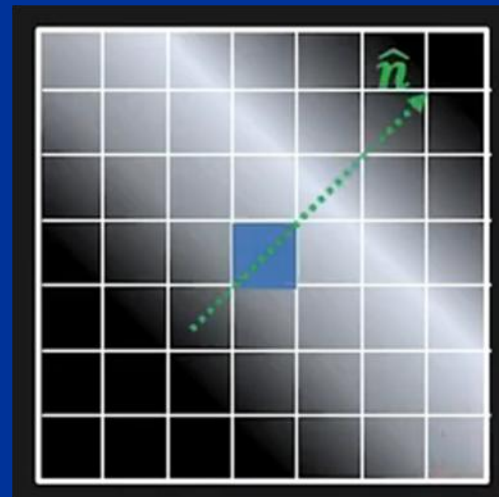


σ_x

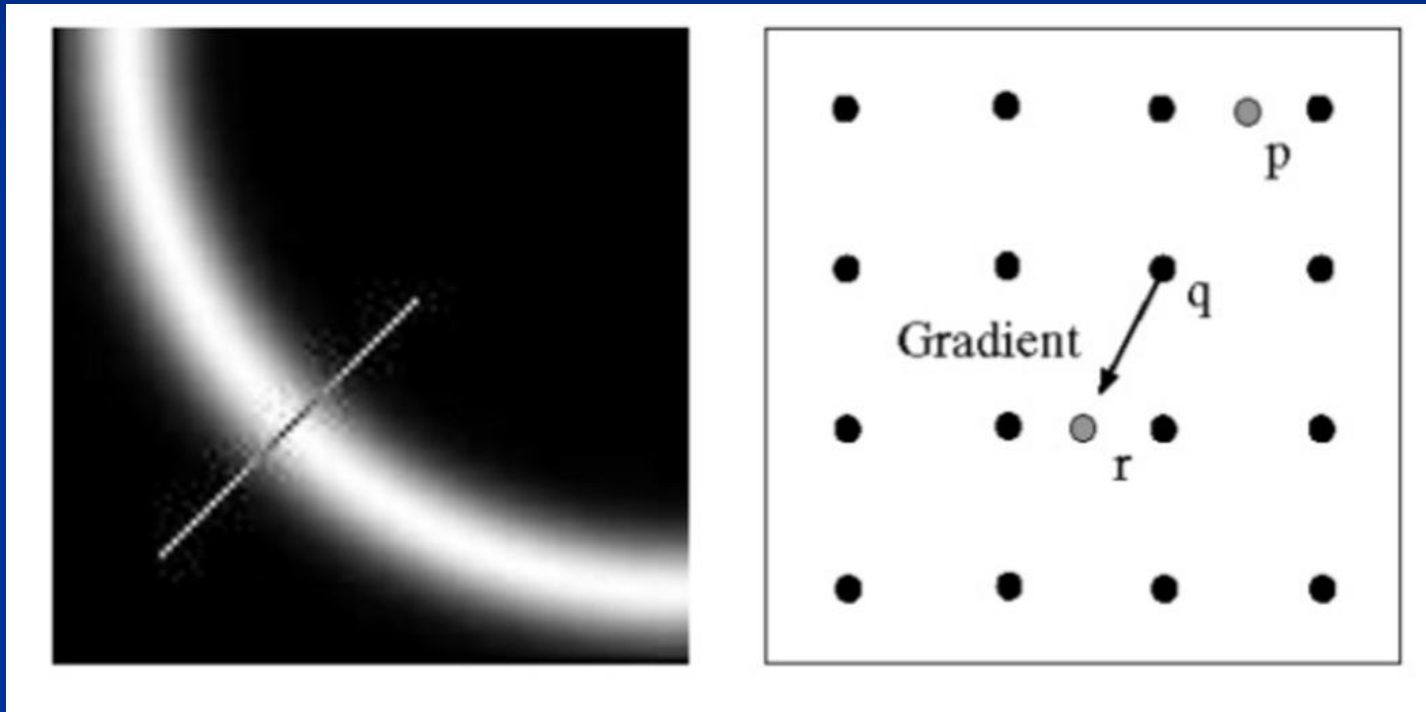


σ_y

Výber maxím (non-maximum suppression)

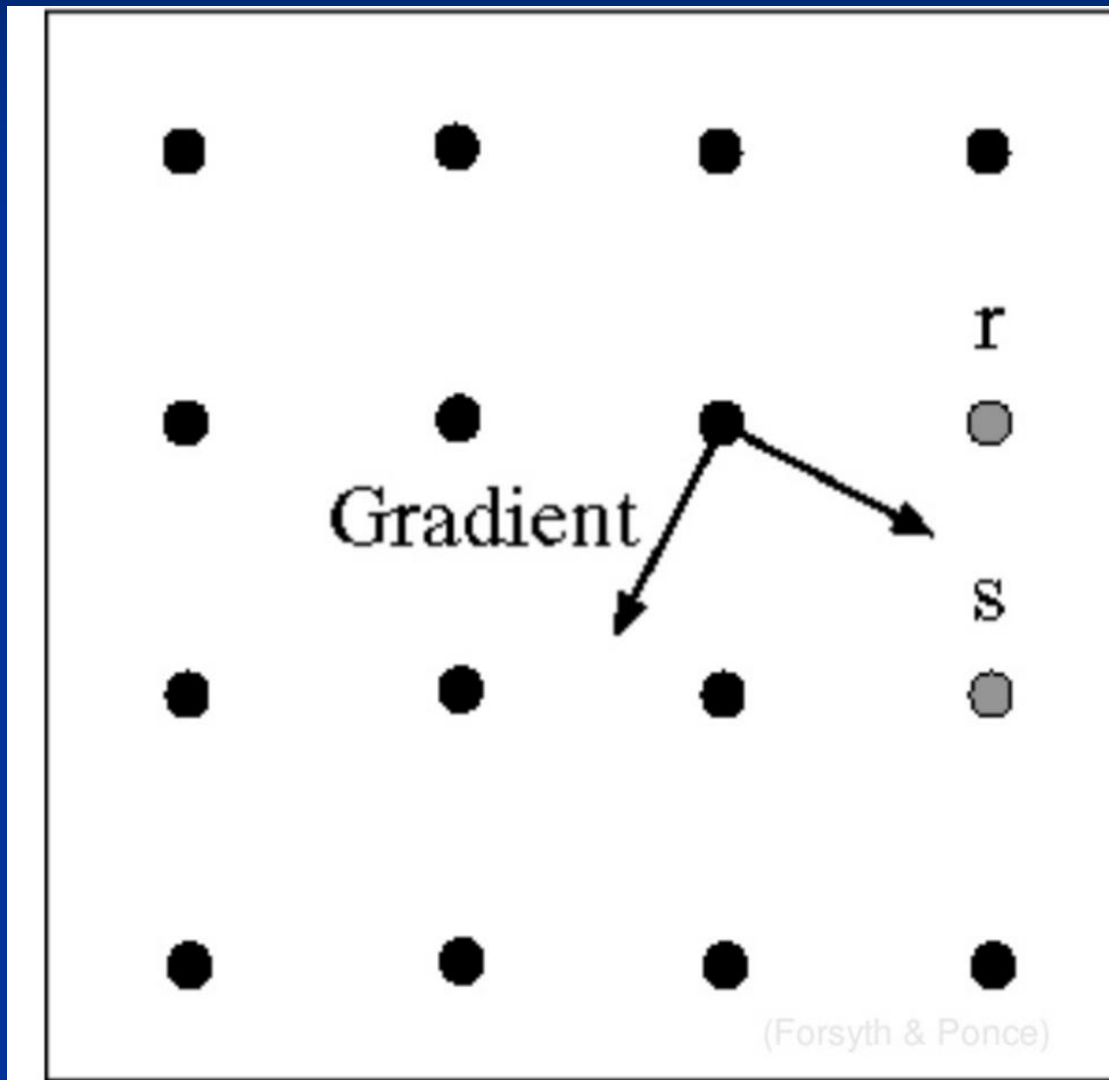


Výber maxím (non-maximum suppression)



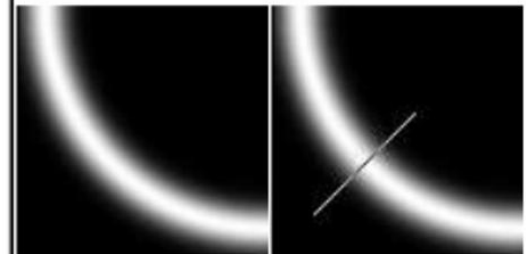
- Zisti či pixel je lokálne maximum pozdĺž smeru gradientu
 - Treba skontrolovať interpolované body p a r

Výber maxím (non-maximum suppression)



Predicting
the next
edge point

Assume the marked point is an edge point. Then we construct the tangent to the edge curve (which is normal to the gradient at that point) and use this to predict the next points (here either r or s).



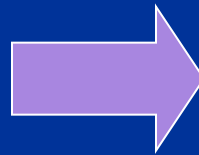
- Nasleduje **hysterézne** prahovanie - prahovanie odoziev hrany s dvoma prahmi
 - silná odozva – vracia hranu,
 - nízka odozva spojená s blízkosťou vysokej odozvy môže vrátiť hranu)
- syntézu príznakov - zistí sa také sigma Gaussiánu, ktoré dáva odozvu väčšiu ako prahovú, a potom sa zväčšuje sigma a agreguje sa získaná informácia
- Na rozdiel od Marr-Hildrethovej operátora, ktorý je založený na Laplaciáne a teda nemá orientáciu, tu sa dá robiť aj orientácia.

Canny



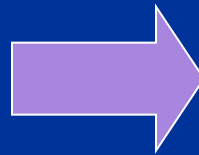
Canny príklady

- Gauss 5x5, T1=255, T2=1



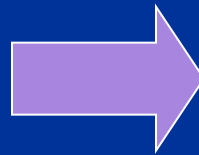
Canny príklady

- Gauss 5x5, T1=255, T2=220



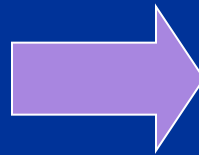
Canny príklady

- Gauss 5x5, T1=128, T2=1



Canny príklady

- Gauss 9x9, T1=128, T2=1



- Canny je výpočtovo náročnejší ako Sobel, Prewitt, Roberts ale dosahuje vo všeobecnosti lepšie výsledky.

Parametrické modely hrán

- diskretnú funkciu intenzity obrazu možno považovať za vzorkovanú a zašumenú aproximáciu určitej spojitej alebo po častiach spojitej funkcie intenzity obrazu.
- Keďže jej priebeh nie je známy, odhaduje sa zo známej diskretnej funkcie a jej vlastností.
- Používajú sa po častiach spojité funkcie, ktoré sa nazývajú **fazety** (alebo plôšky) a model sa nazýva **fazetový** (plôškový) **model**.

- Najjednoduchší plochý **fazetový model**, používa konštantné funkcie a okolie pixla je reprezentované funkciami konštantnej intenzity.
- lineárne modely, kvadratické a bikubické modely
- Príklad: bikubický fazetový model

$$g(i, j) = c_1 + c_2x + c_3y + c_4x^2 + c_5xy + c_6y^2 + c_7x^3 + c_8x^2y + c_9xy^2 + c_{10}y^3$$

- c_i sa odhadujú z okolia obrazového bodu, pomocou metódy najmenších štvorcov, alebo priamo z okolia rozmeru 5 x 5.

- Keď už máme fazetové parametre pre každý obrazový bod hrany určíme pomocou extrému 1.derivácie alebo ako prechod nulou druhej derivácie lokálne spojitej fazetovej funkcie.
- Určenie hrán je potom presnejšie ako pri diferencných hranových operátoroch, ale platíme za to náročnosťou výpočtu.