

Цифрова обработка на изображения

Локални операции с изображения

доц. Милена Лазарова, кат. КС, ФКСУ

Видове операции в ЦОИ

- Точкови операции
- **Локални операции**
- Глобални операции

Локални операции

- Изходната стойност за всеки пиксел зависи от входните стойности на съседни пиксели в малка локална околност
- I и J са изображения, за които $J = T [I]$
- $T [\cdot]$ е трансформация от следния вид

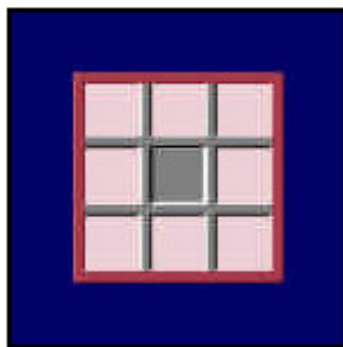
$$J(r, c) = T[I](r, c) = f(\{I(u, v) \mid u \in \{r - s, \dots, r, \dots, r + s\}, v \in \{c - d, \dots, c, \dots, c + d\}\})$$

- Стойността на пиксел с координати (r, c) в трансформираното изображение J е функция от стойностите на оригиналното изображение I в малка правоъгълна околност с център пиксела с координати (r, c) и размери $(2s+1) \times (2d+1)$

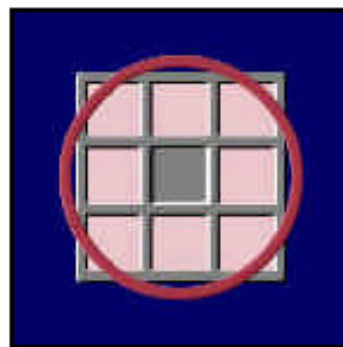
Плъзгащ се прозорец

- Операциите в локални околности се наричат още **операции с плъзгащ се прозорец**
 - пикселите от изображението I , използвани в трансформацията T , са различни за всеки отделен пиксел, но размерите и формата на локалната околност са еднакви

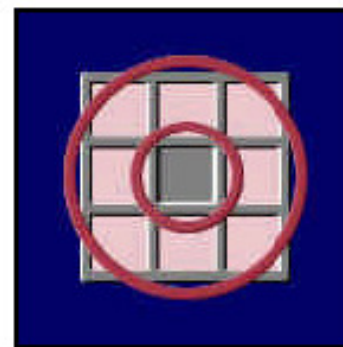
Плъзгащ се прозорец (Moving Window)



SQUARE



CIRCLE

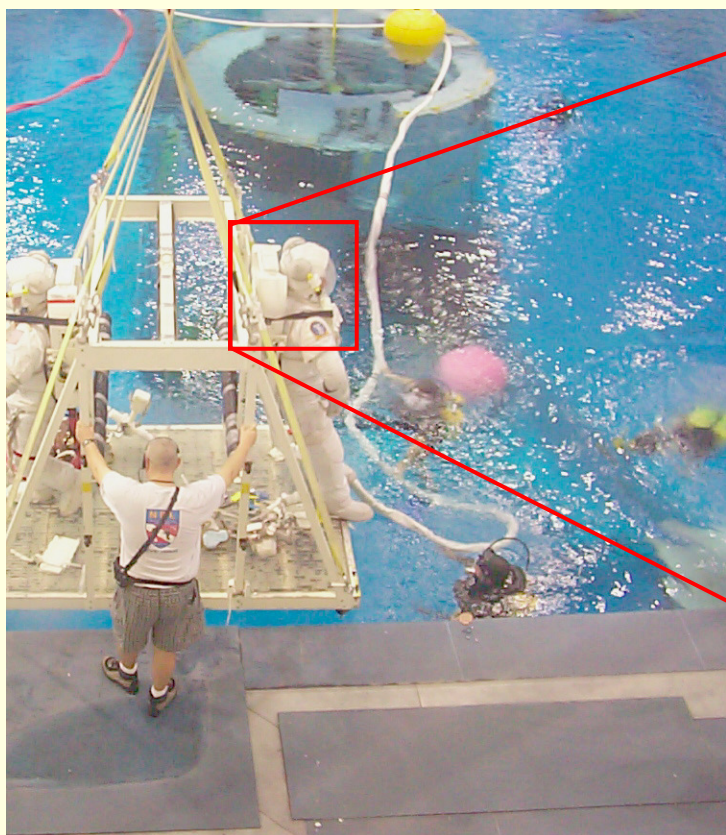


DONUT

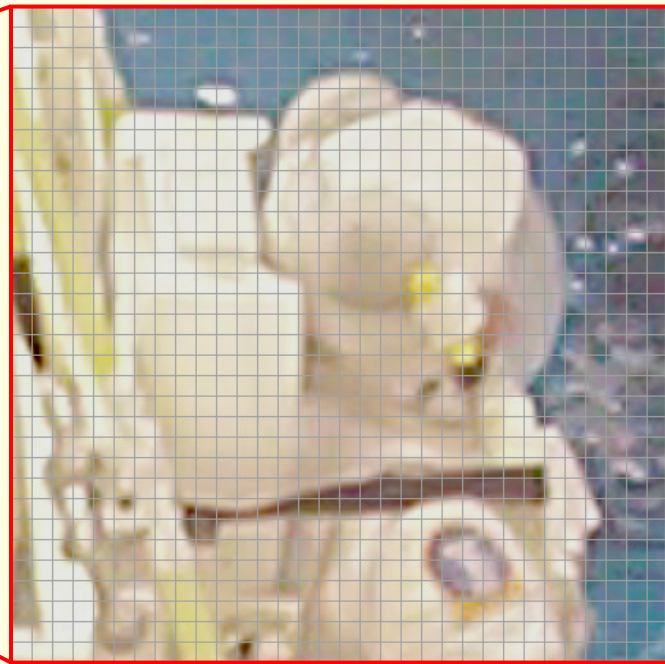
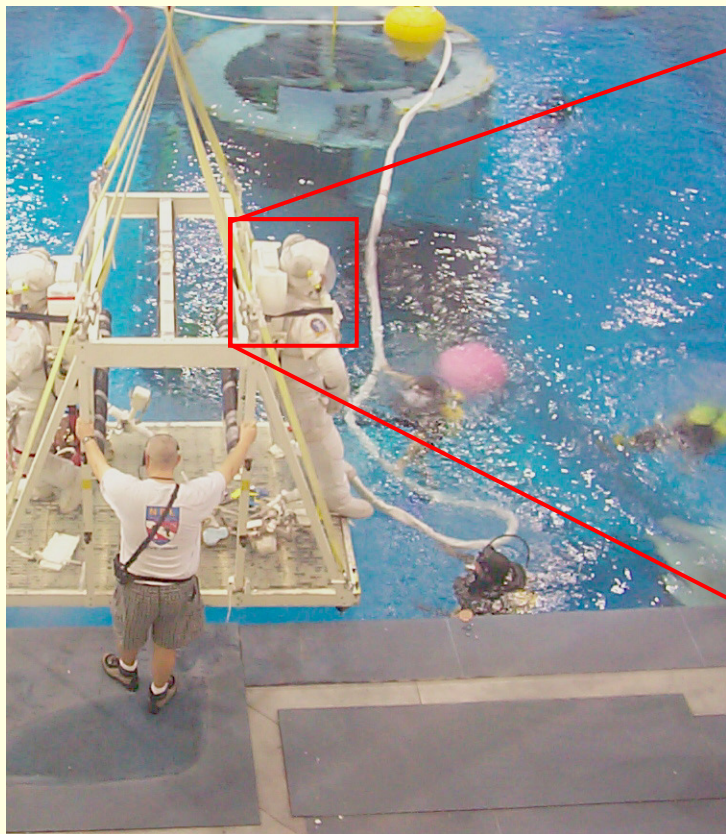


WEDGE

Плъзгащ се прозорец

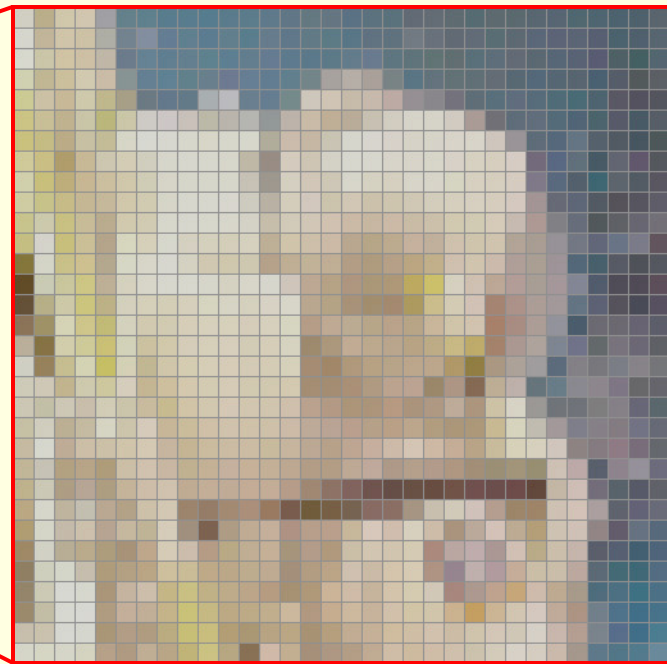
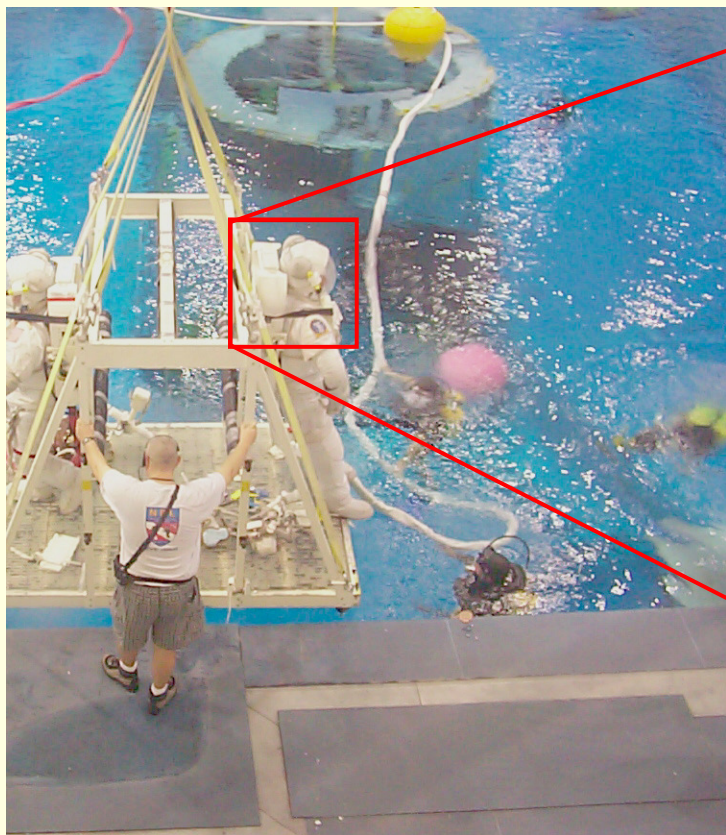


Плъзгащ се прозорец



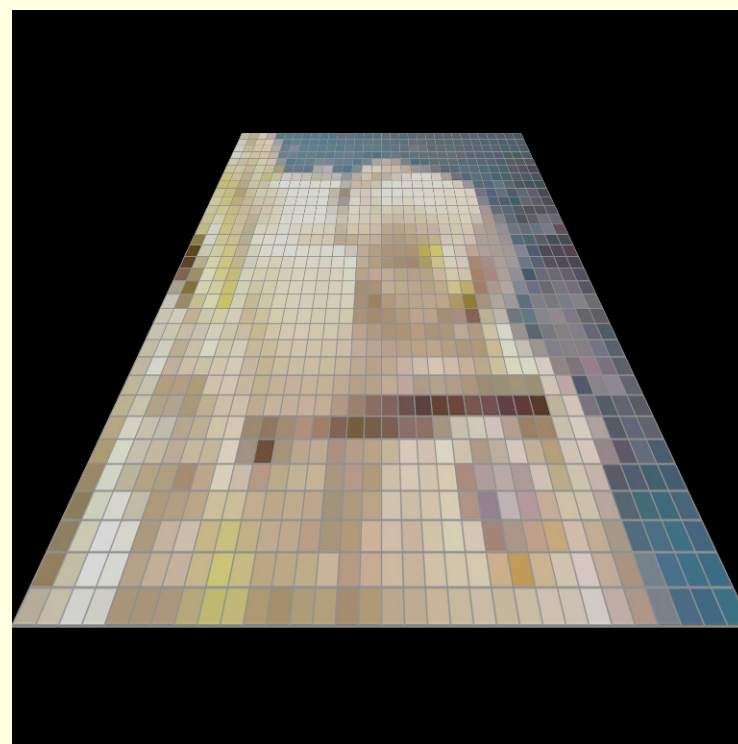
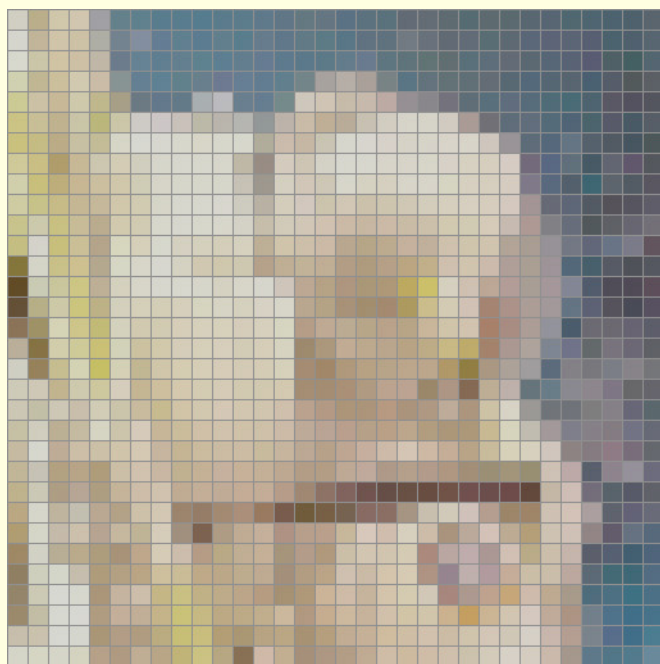
пространствено дискретизиране
(*sampling*)

Плъзгащ се прозорец

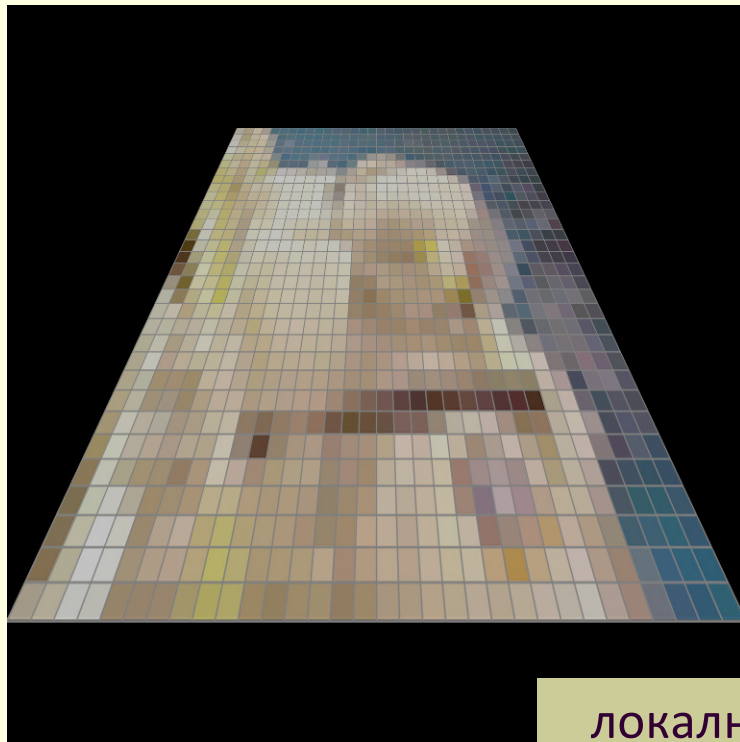


квантоване по нива
(*quantization*)

Плъзгащ се прозорец

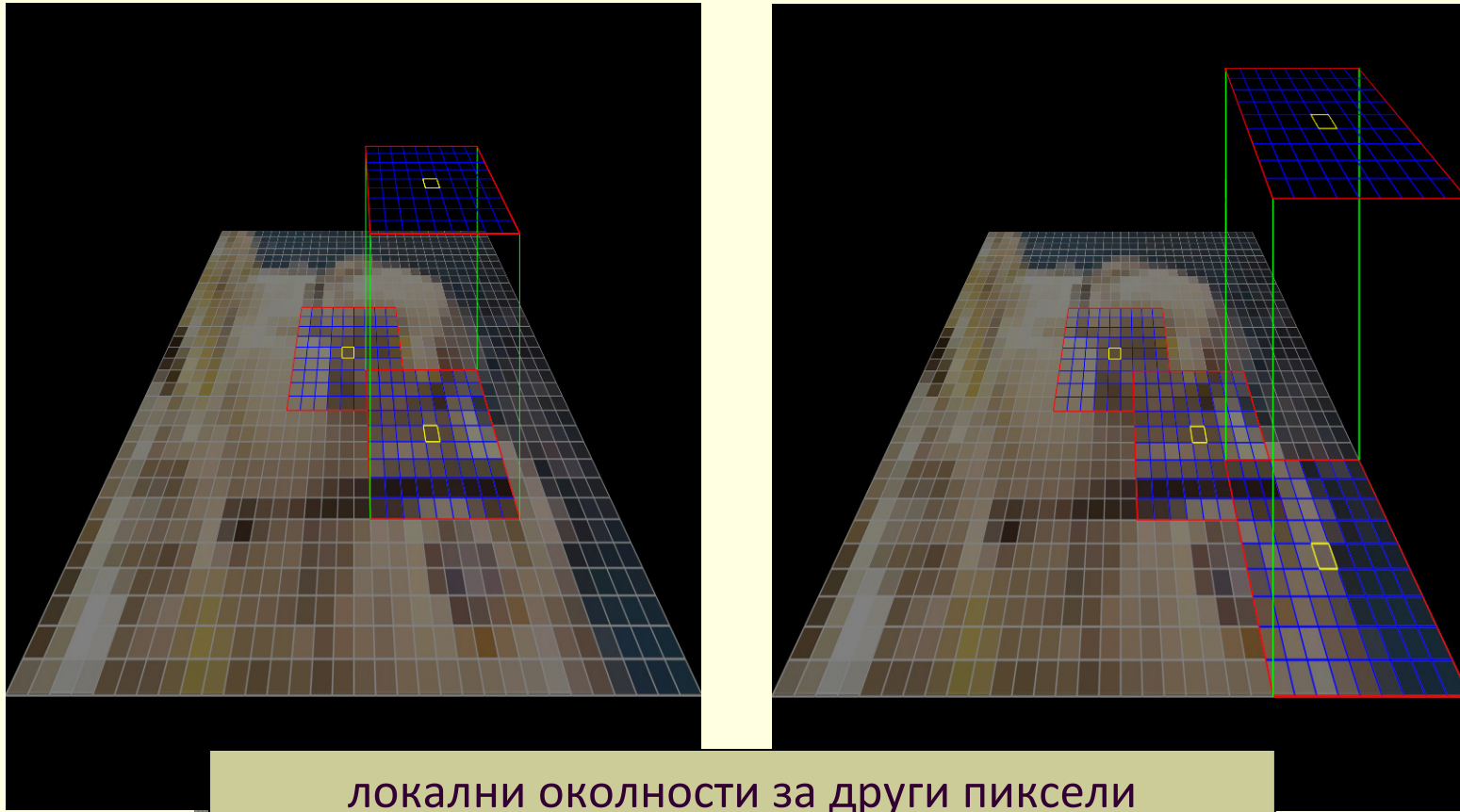


Плъзгащ се прозорец



локална околност за
централния пиксел

Плъзгащ се прозорец



Локални операции

- В зависимост от трансформацията T се променя влиянието на локалната операция върху оригиналното изображение
- Два вида операции в локална околност
 - **линейни**
 - стойността на пиксела в новото изображение се получава като линейна комбинация на стойностите в локалната му околност и зададени теглови коефициенти
 - **нелинейни**
 - стойността на пиксела в новото изображение се получава като нелинейно преобразуване на стойностите в локалната му околност
 - не се използват теглови коефициенти

Линейни операции

- Използва се операцията **КОНВОЛЮЦИЯ**
 - прилага се *линейна трансформация* за изображението I
- В *непрекъснато пространство* конволюцията се дефинира чрез конволюционен интеграл

$$J(r, c) = [I * h](r, c) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(r - \rho, c - \kappa) h(\rho, \kappa) d\rho d\kappa,$$

- В *дискретния случай* за цифрови изображения конволюцията се реализира чрез сумиране в дискретно пространство

$$J(r, c) = [I * h](r, c) = \sum_{\rho=-\infty}^{\infty} \sum_{\kappa=-\infty}^{\infty} I(r - \rho, c - \kappa) h(\rho, \kappa)$$

Линейни операции

- Конволюционна маска $h(\rho, \kappa)$
 - теглова **функция** в непрекъснатия случай
 - теглова **матрица** в дискретния случай
- Стойността на пиксел с координати (r, c) в изходното изображение се получава като **претеглена сума** на стойностите на пикселите, попадащи в плъзгащия се прозорец във входното изображение
 - всеки пиксел в локалната околност се умножава по съответната теглова стойност в конволюционната матрица
 - сумата на тези произведения е новата стойност на централния пиксел в резултантното изображение

Линейни операции

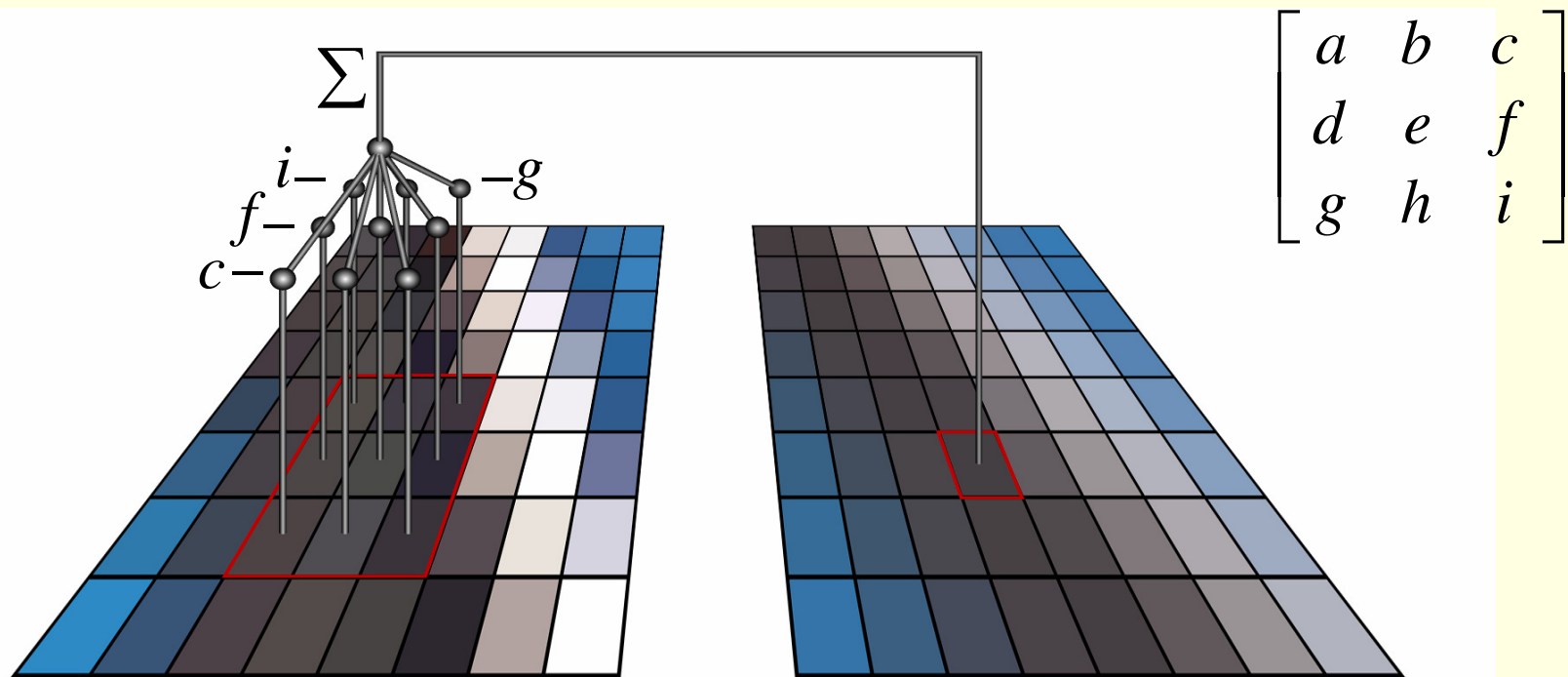
		$l(r-1,c-1)$	$l(r,c-1)$	$l(r+1,c-1)$					
		$l(r-1,c)$	$l(r,c)$	$l(r+1,c)$					
		$l(r-1,c+1)$	$l(r,c+1)$	$l(r+1,c+1)$					

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

$$J(r, c) =$$

$$\begin{aligned} & l(r-1,c-1) \times a + l(r,c-1) \times b + l(r+1,c-1) \times c + \\ & l(r-1,c) \times d + l(r,c) \times e + l(r+1,c) \times f + \\ & l(r-1,c+1) \times g + l(r,c+1) \times h + l(r+1,c+1) \times i \end{aligned}$$

Конволюция



Линейни операции

- В зависимост от стойностите на теглата в конволюционната матрица се постига различен ефект върху обработваното изображение
 - **изглаждащи филтри**
 - *soften, average, blur*
 - **изострящи филтри**
 - *sharpen*
 - **филтри за определяне на контури**
 - *edge detection*

Изглаждащи филтри

- Използват се за “замъгляване” на изображението и редуциране на шум
- Базират се на осредняване на стойностите на елементите в локалния прозорец
 - **филтър по средна стойност** (*average, mean, box filter*)
 - матрица H
 - **претеглено осредняване** (*blur*)
 - матрица H_1, H_2, H_3

$$H = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, H_1 = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, H_2 = \frac{1}{14} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 2 & 2 \\ 1 & 2 & 1 \end{bmatrix}, H_3 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Изглаждащи филтри

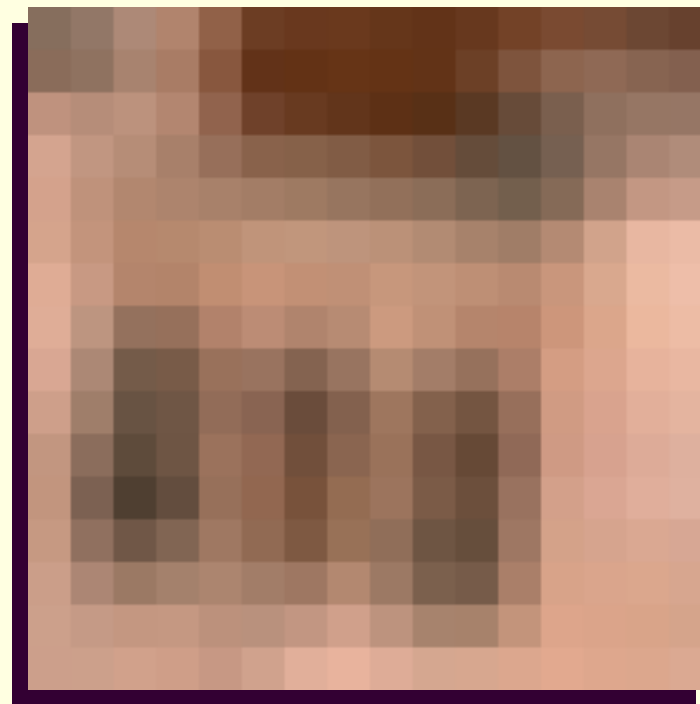
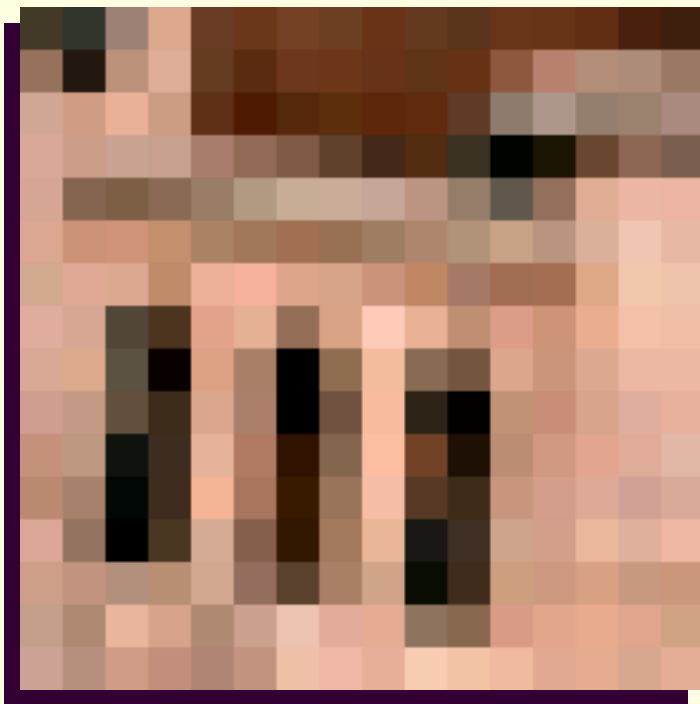
$$H_{rect} = \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$H_{circ} = \frac{1}{21} \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

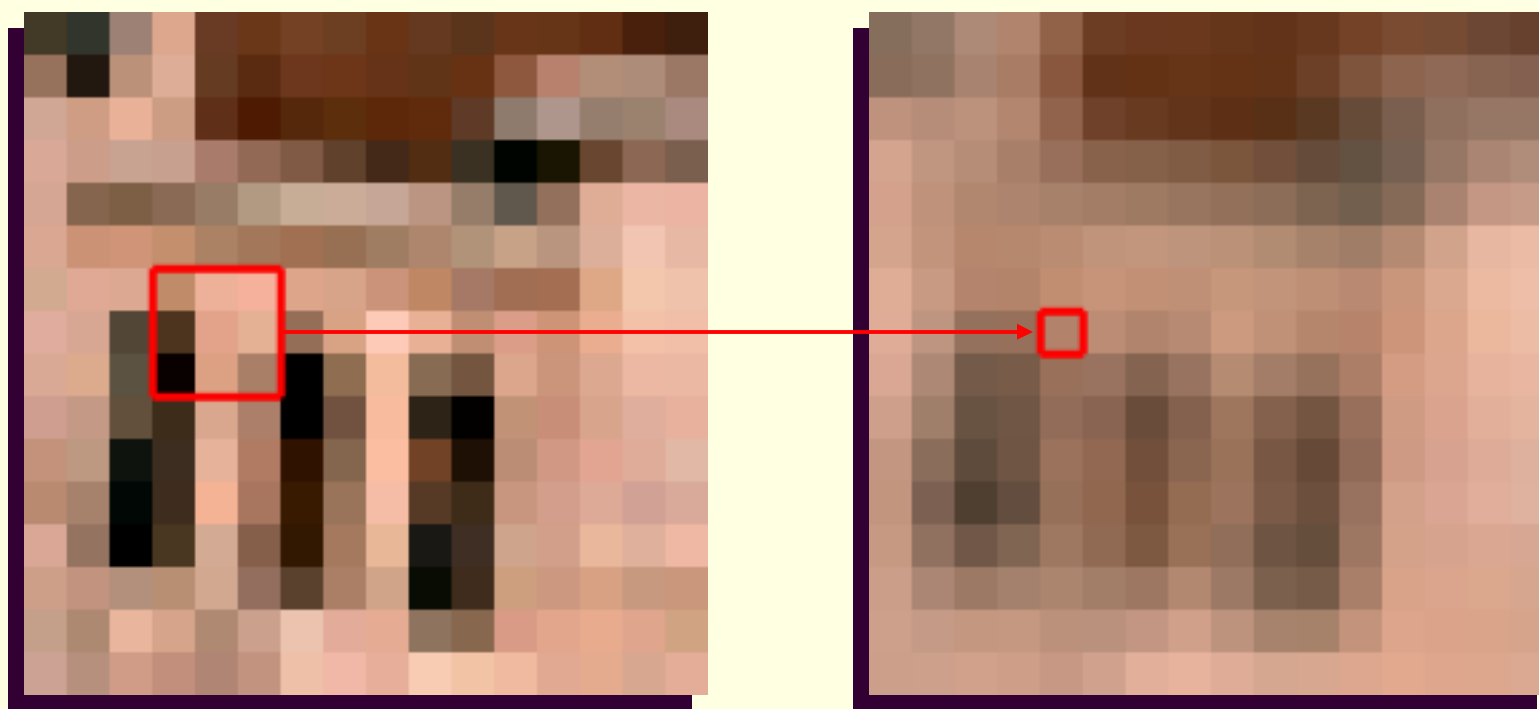
$$H_{pyr} = \frac{1}{81} \begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 3 & 6 & 3 & 2 \\ 3 & 6 & 9 & 6 & 3 \\ 2 & 3 & 6 & 3 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix}$$

$$H_{cone} = \frac{1}{25} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 2 & 2 & 1 \\ 1 & 2 & 5 & 2 & 1 \\ 0 & 2 & 2 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

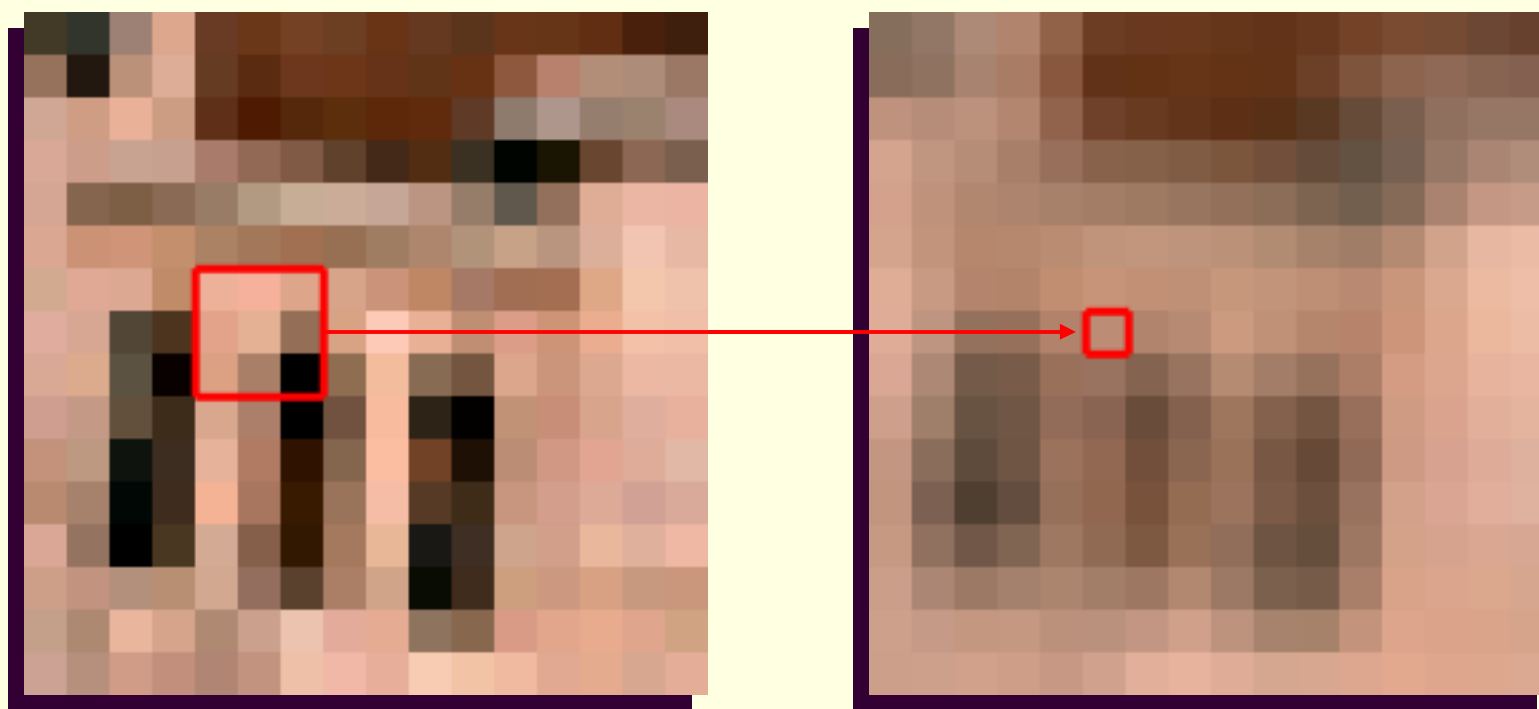
Изглаждащи филтри



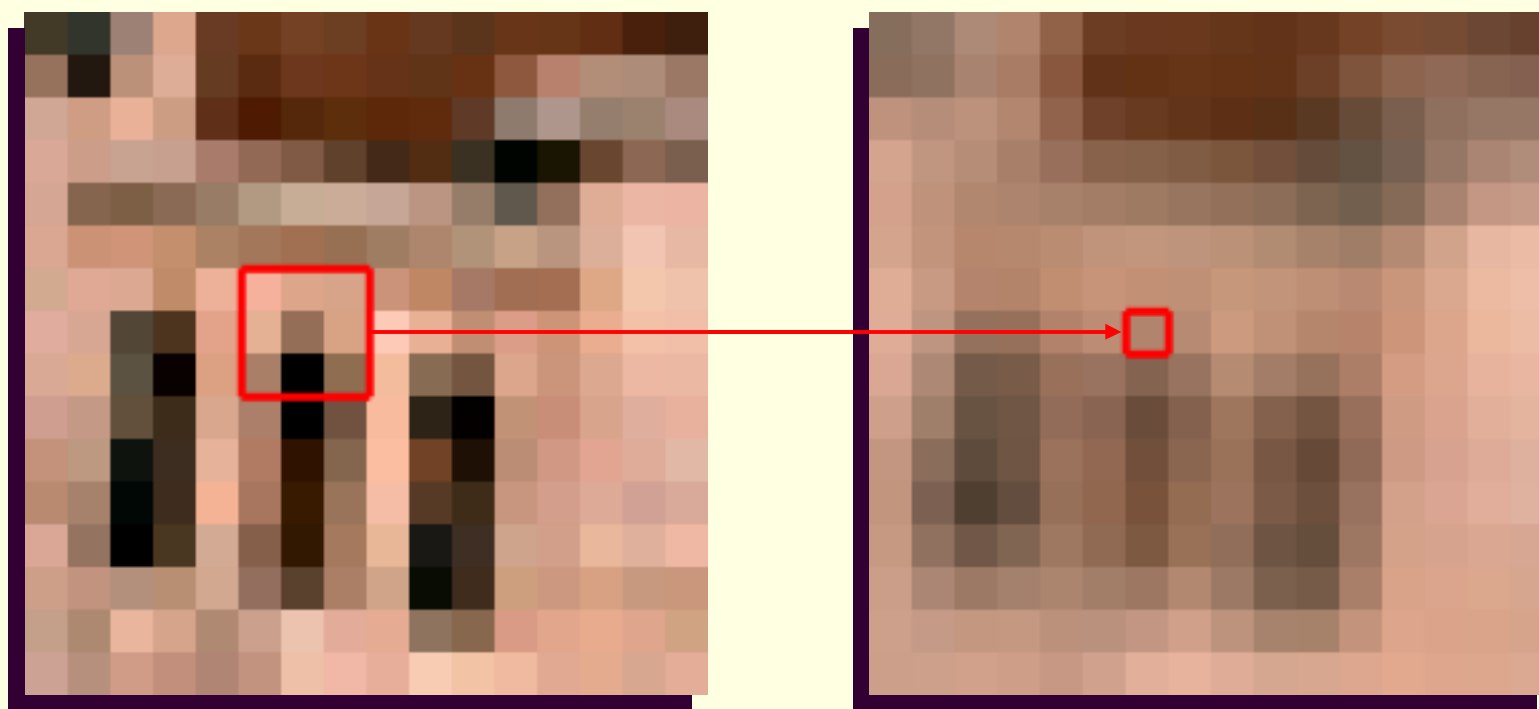
Изглаждащи филтри



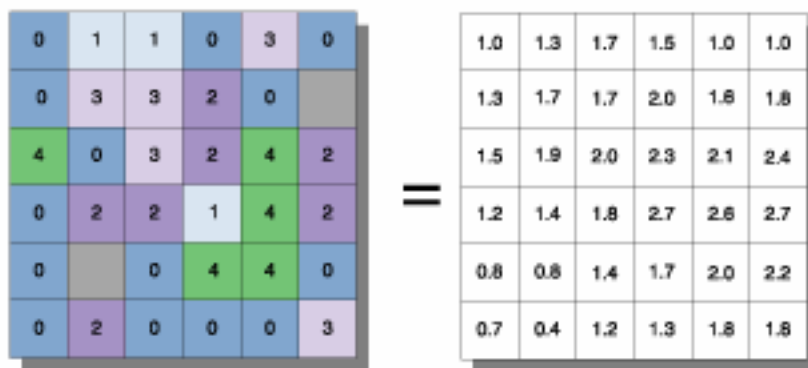
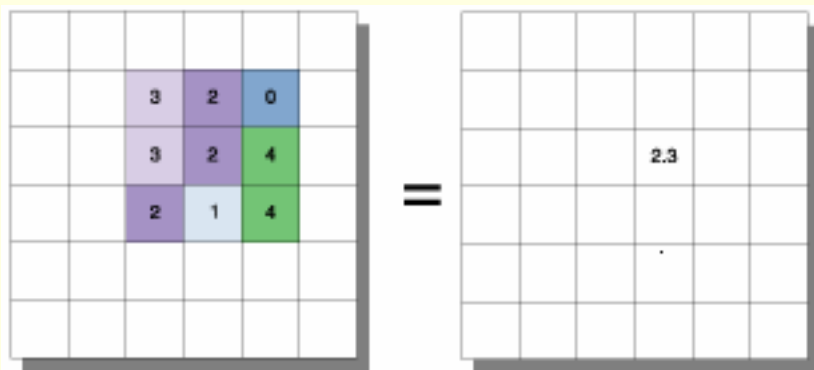
Изглаждащи филтри



Изглаждащи филтри



Изглаждащи филтри



Конволюция – свойства

- **Коммутативность**

$$a \otimes b = b \otimes a$$

- **Ассоциативность**

$$a \otimes (b \otimes d) = (a \otimes b) \otimes d = a \otimes b \otimes d$$

- **Дистрибутивность**

$$a \otimes (b + d) = (a \otimes b) + (a \otimes d)$$

Изглаждащи филтри

■ Разделимост

- прилагането на 2D филтър по средна стойност е еквивалентно на последователно прилагане на **два ортогонално ориентирани 1D филтри**

$$H_{unif\ 3\times 3} = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} = \begin{bmatrix} 0 & 1/3 & 0 \\ 0 & 1/3 & 0 \\ 0 & 1/3 & 0 \end{bmatrix} \circ \begin{bmatrix} 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 \\ 0 & 0 & 0 \end{bmatrix}$$

Изглаждащи филтри

- Стойности върху рамката на изображението
 - **padding** (увеличаване размерите на изображението)
 - нулеви стойности
 - стойностите по рамката
 - циклично повторение

8	3	4	5
7	6	4	5
4	5	7	8
6	5	5	6

0	0	0	0	0	0
0	8	3	4	5	0
0	7	6	4	5	0
0	4	5	7	8	0
0	6	5	5	6	0
0	0	0	0	0	0

8	8	3	4	5	5
8	8	3	4	5	5
7	7	6	4	5	5
4	4	5	7	8	8
6	6	5	5	6	6
6	6	5	5	6	6

6	6	5	5	6	6
5	8	3	4	5	8
5	7	6	4	5	7
8	4	5	7	8	4
6	6	5	5	6	6
5	8	3	4	5	8

Изглаждащи филтри

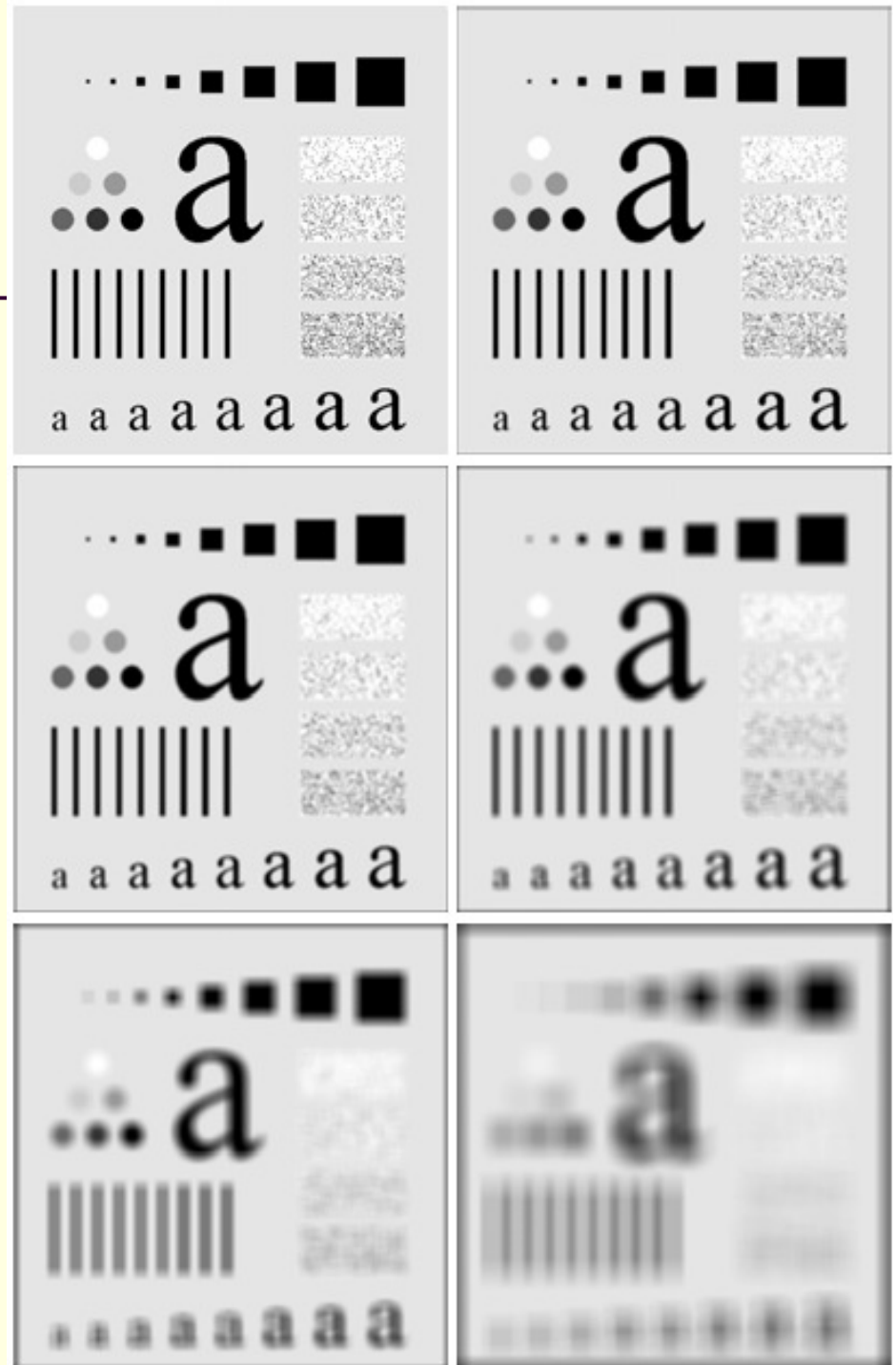
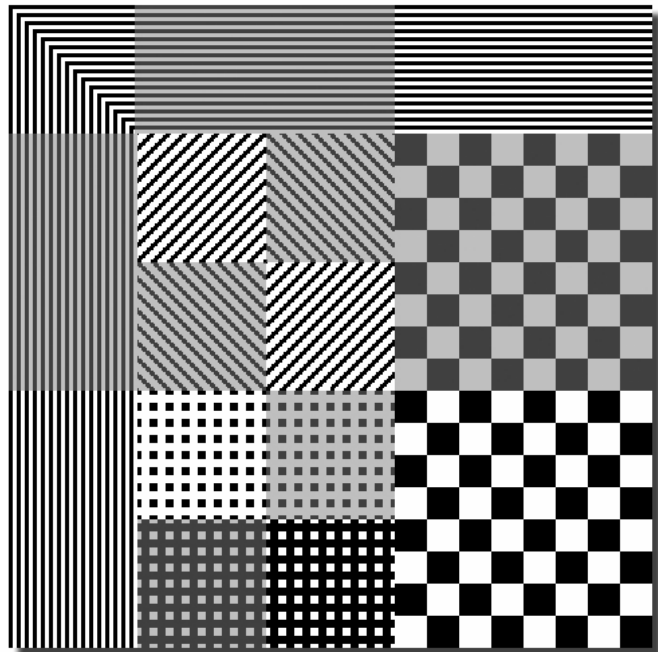


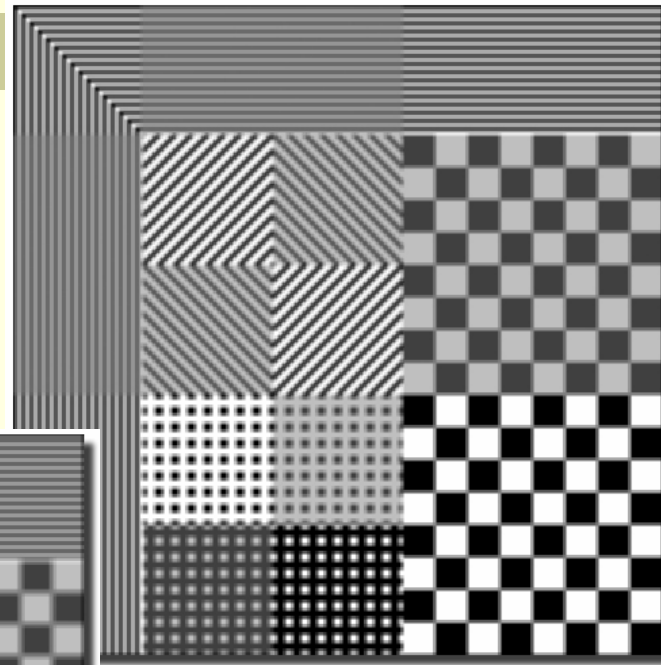
FIGURE 3.35 (a) Original image, of size 500×500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $n = 3, 5, 9, 15,$ and $35,$ respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their gray levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50×120 pixels.

Изглаждащи филтри

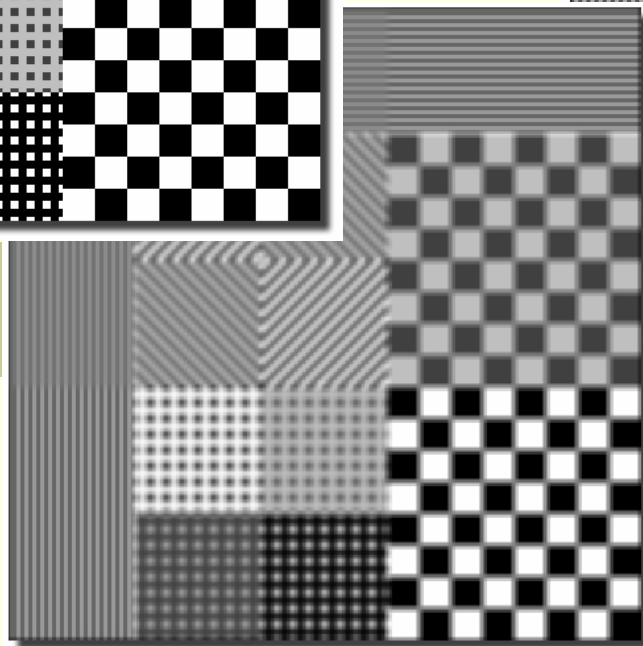


оригинално
изображение

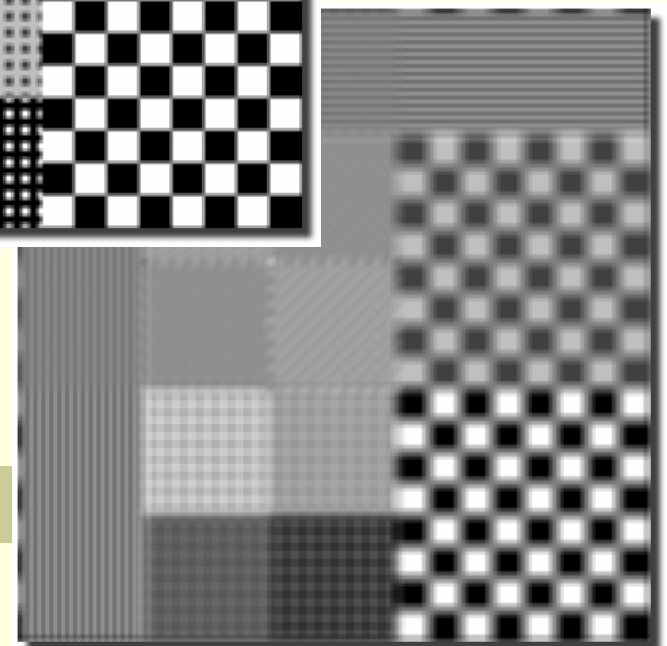
3x3



5x5



9x9



Изглаждащи филтри



оригинално
изображение

3x3



5x5

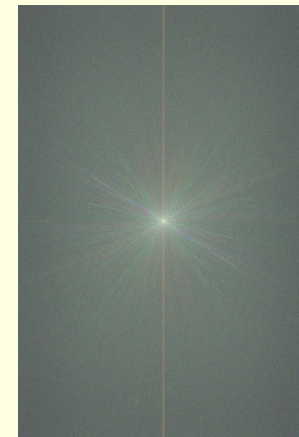
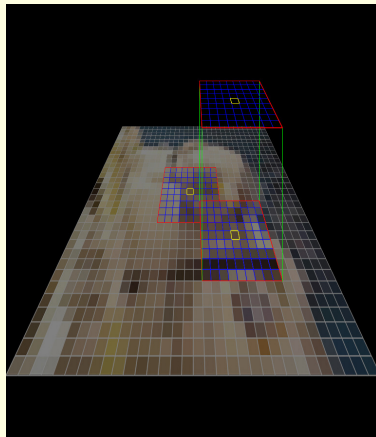


9x9



Конволюция

- Три начина да се приложи конволюция
 - *плъзгащ се прозорец*
 - *изместване, умножение, сумиране*
 - *Фурие трансформация*



Изместване, умножение, сумиране

- Изображението се **копира** по веднъж за всеки елемент в конволюционната матрица
- Всяко копие се **премества** съгласно позицията на съответния елемент на матрицата
- Всяко копие се **умножава** със стойността на елемента от конволюционната матрица
- **Сумират** се съответните елементи от полученото множество изображения след изместването и умножението

Изместване, умножение, сумиране

■ **Конволюция с импулсно изображение**

- импулсно изображение – съдържа единствен пиксел със стойност единица, всички останали са нули

■ **импулс с позиция (ρ, χ)**

$$\delta(r - \rho, c - \chi) = \begin{cases} 1, & \text{ако } r = \rho \text{ и } c = \chi \\ 0, & \text{в противен случай} \end{cases}$$

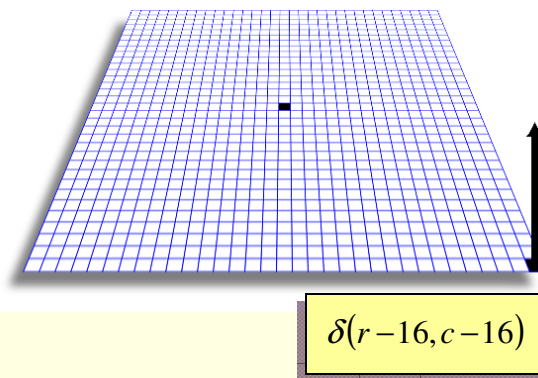
- конволюция на изображението I с импулс

$$[I * \delta(r - \rho, c - \chi)](r, c) = I(r - \rho, c - \chi)$$

- **изместване на изображението с ρ пиксела надясно и с χ пиксела надолу**

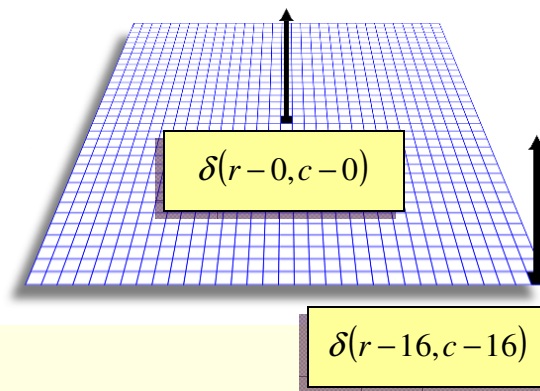
Изместване, умножение, сумиране

- Конволюция с импулс $\delta(r-16, c-16)$



Изместване, умножение, сумиране

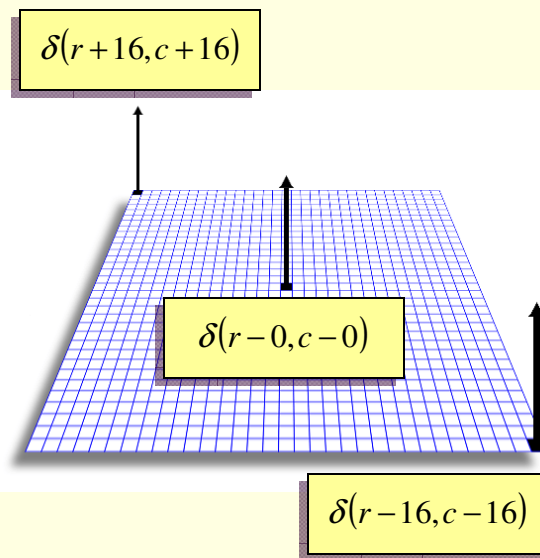
- Конволюция с два импулса $\delta(r-16, c-16)$ и $\delta(r-0, c-0)$



сумата на двете изместени
изображения умножена с 1/2

Изместване, умножение, сумиране

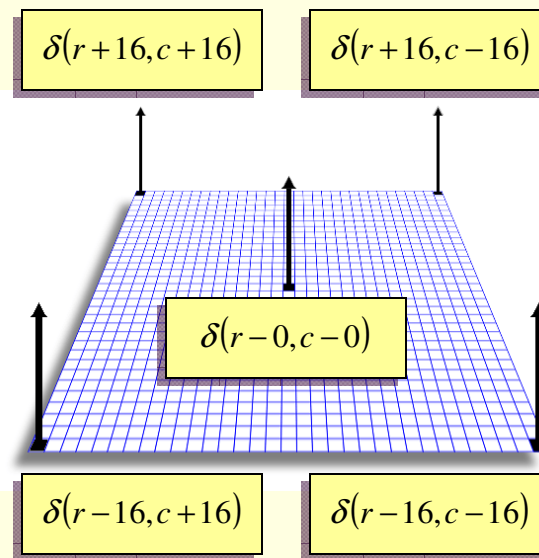
■ Конволюция с три импулса



сумата на трите изместени
изображения умножена с 1/3

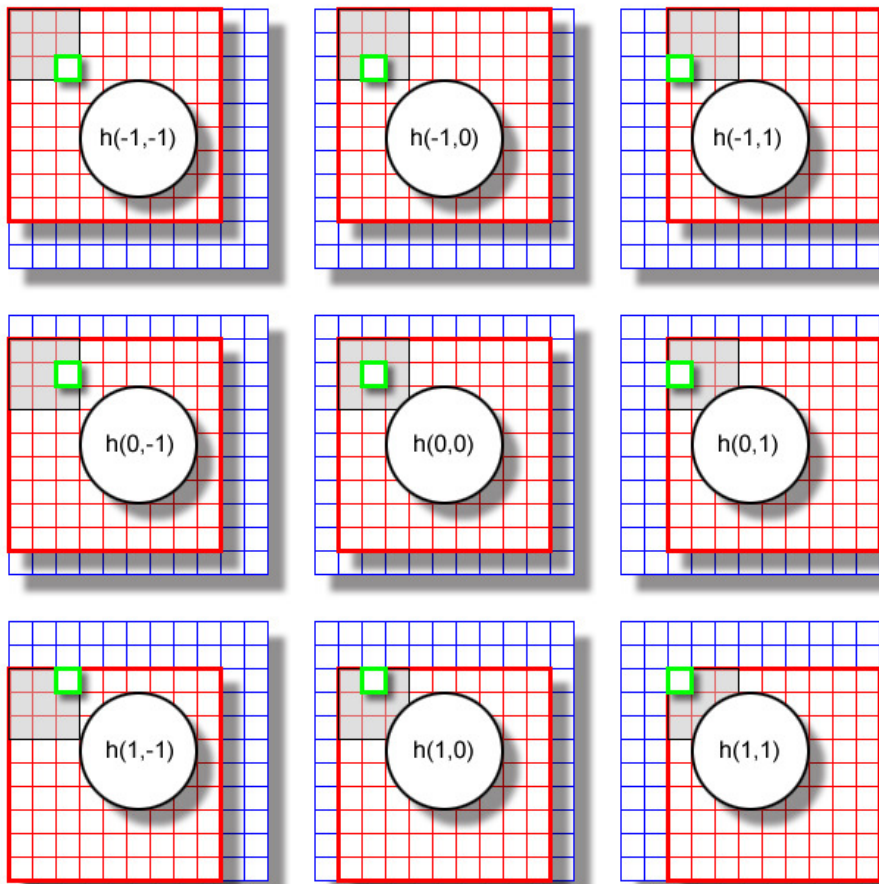
Изместване, умножение, сумиране

■ Конволюция с пет импулса



сумата на петте изместени
изображения умножена с 1/5

Изместване, умножение, сумиране



оригинално изображение, I
разширено изображение, P
локална околност

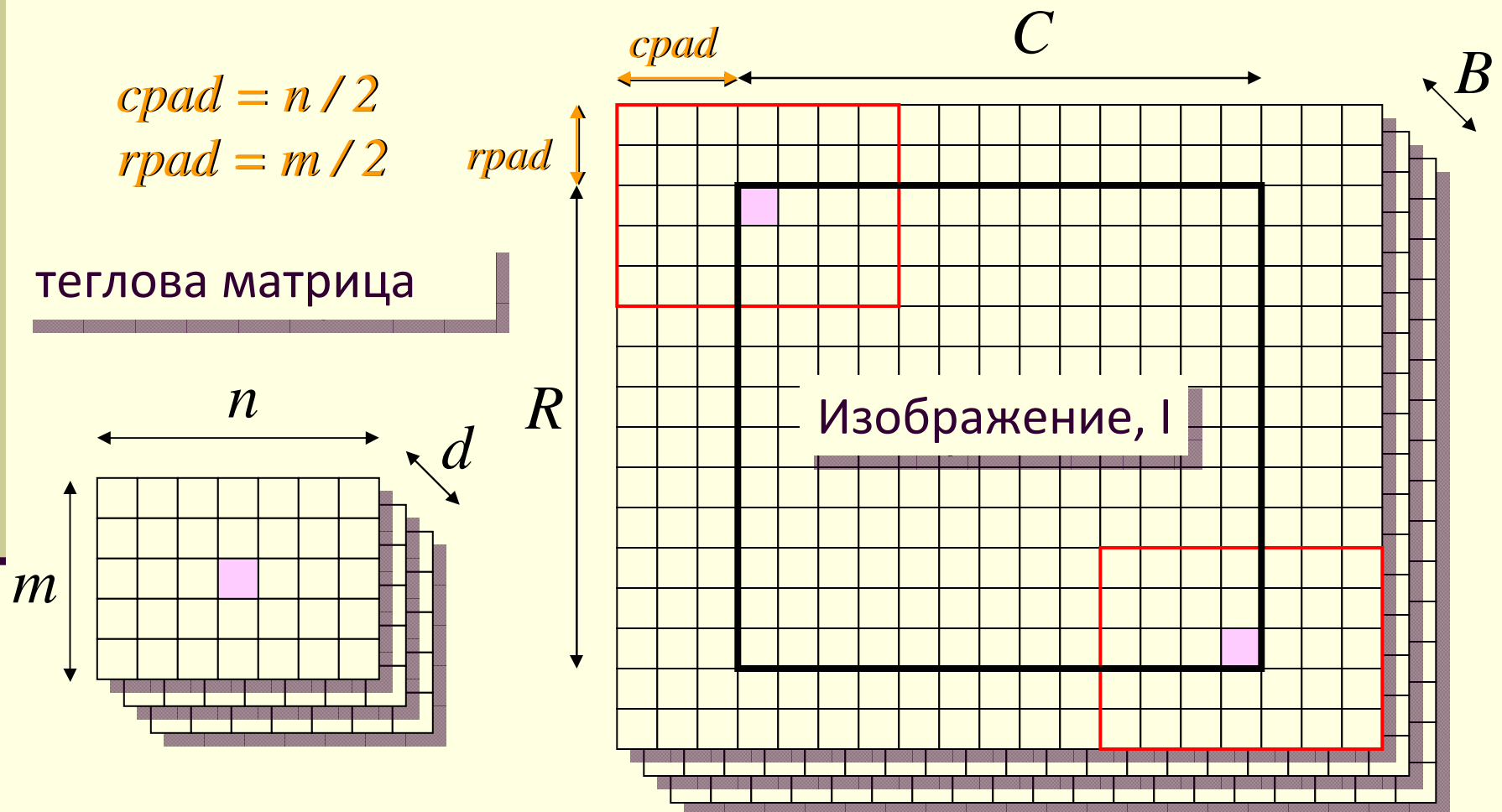
$h(-1,-1)$	$h(-1,0)$	$h(-1,1)$
$h(0,-1)$	$h(0,0)$	$h(0,1)$
$h(1,-1)$	$h(1,0)$	$h(1,1)$

теглова матрица

пиксели, които се сумират

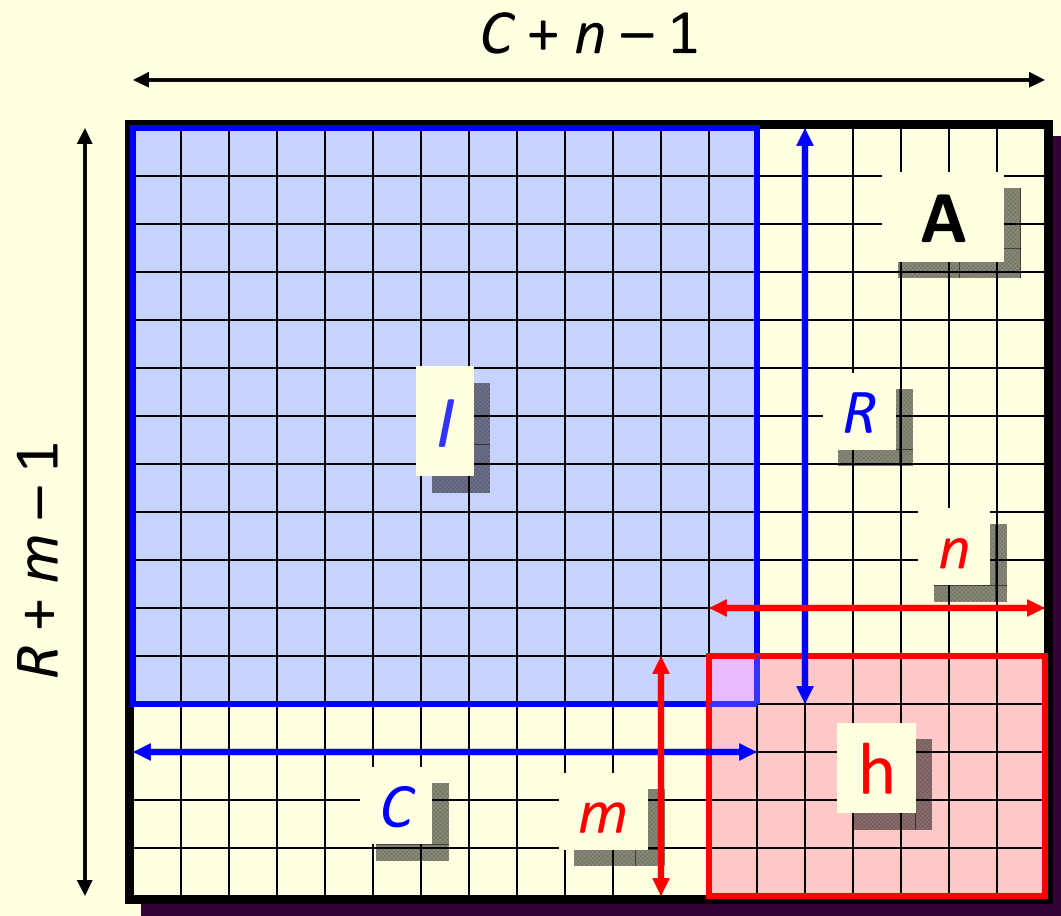
теглова стойност ○

Изместване, умножение, сумиране



Изместване, умножение, сумиране

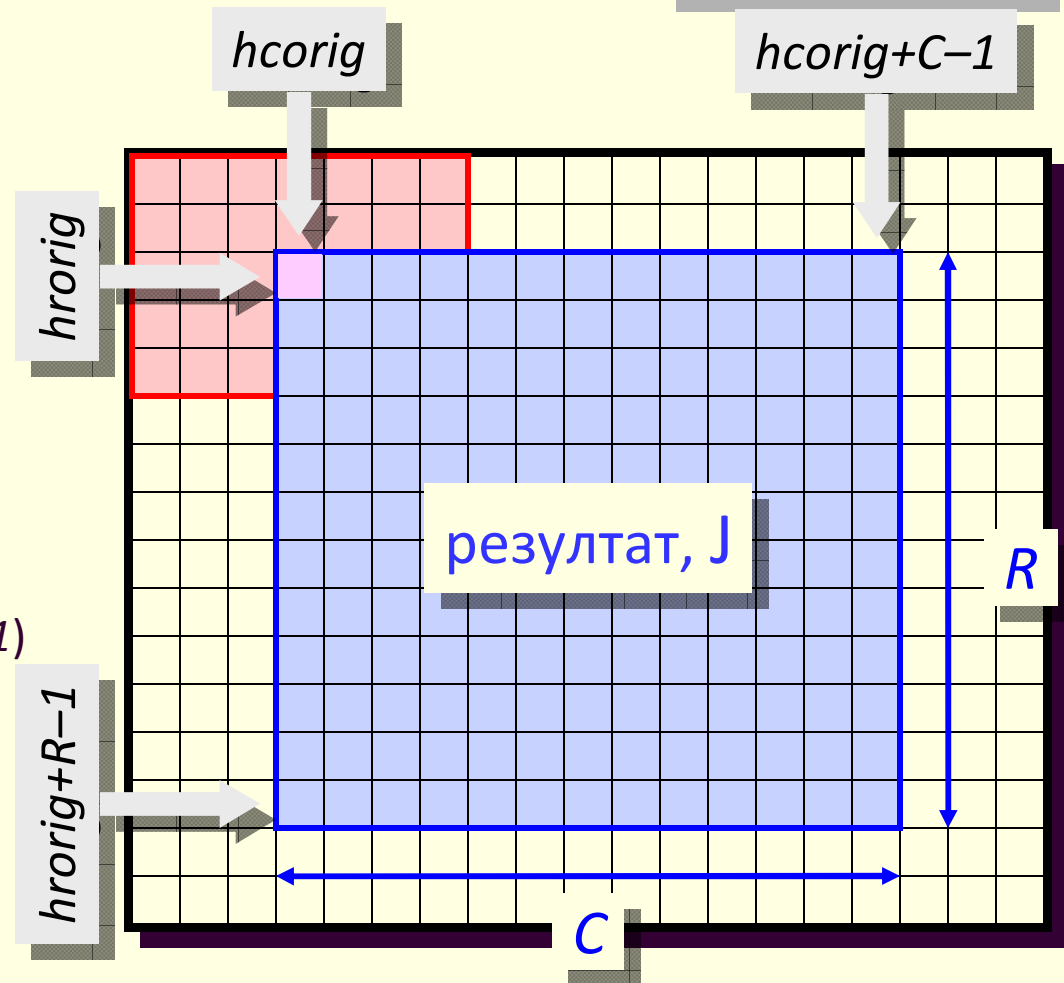
- **Акумулаторно изображение A**
 - с размери $(R+m-1) \times (C+n-1)$
- **Оригинално изображение I**
 - с размери $R \times C$



Изместване, умножение, сумиране

■ Резултантното изображение J

- копира се от акумулатора
- горен ляв ъгъл ($horig, horig$)
- долен десен ъгъл ($horig+R-1, horig+C-1$)



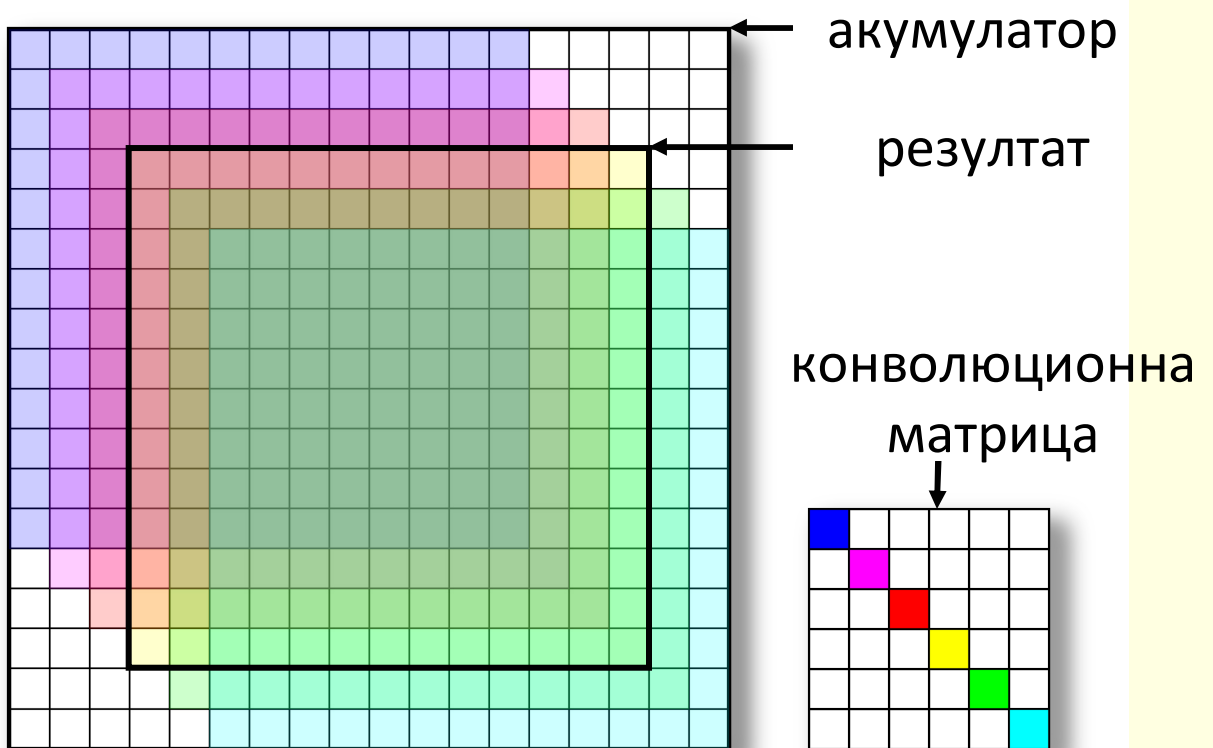
Изместване, умножение, сумиране

- **Оригинално изображение**

- с размери 13×13

- **Теглова матрица 6×6**

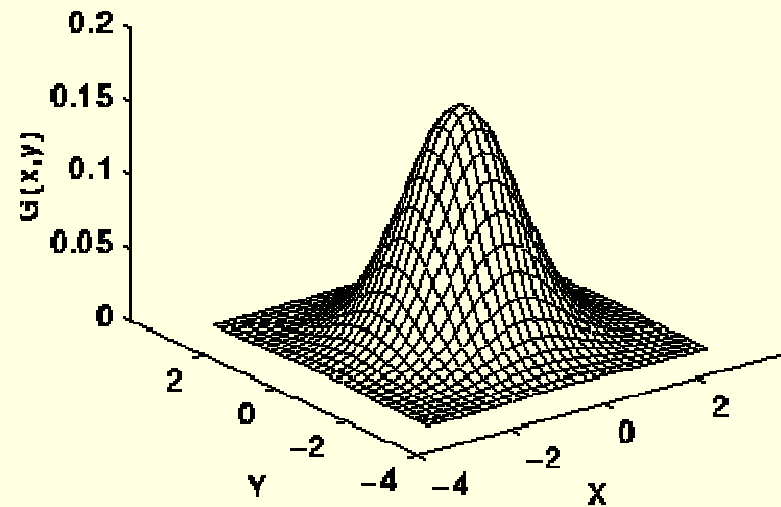
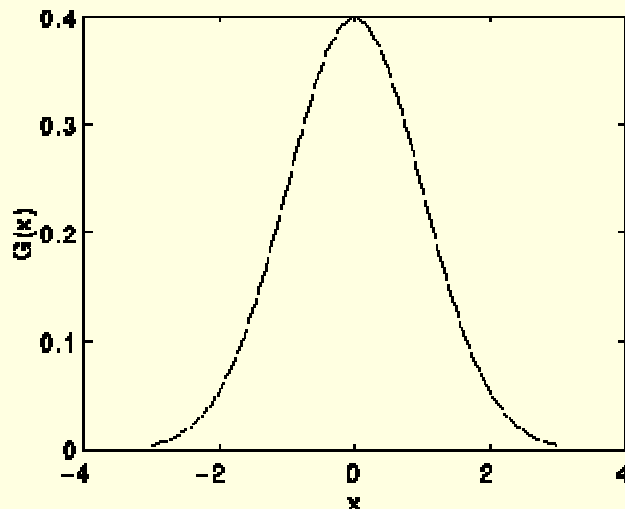
- ненулеви стойности по диагонала



Гаусови филтри

- Специален клас **линейни изглаждащи** филтри
 - теглата в изглаждащата матрица се получават от Гаусова функция

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



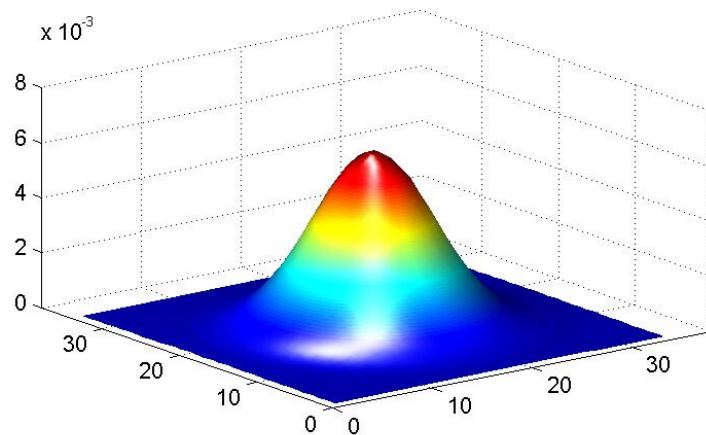
Гаусови филтри

- Степента на изглаждане на зависи от **ширината** на Гаусовия филтър
 - определя се от параметъра σ (**дисперсия**)
- Зависимостта е **право пропорционална**
 - с увеличаване на стойността на σ се увеличава ширината на филтъра
 - размера на разглежданата локална апертура
 - ширината на филтъра е свързана степента на изглаждане и чувствителността спрямо дребни детайли в изображението

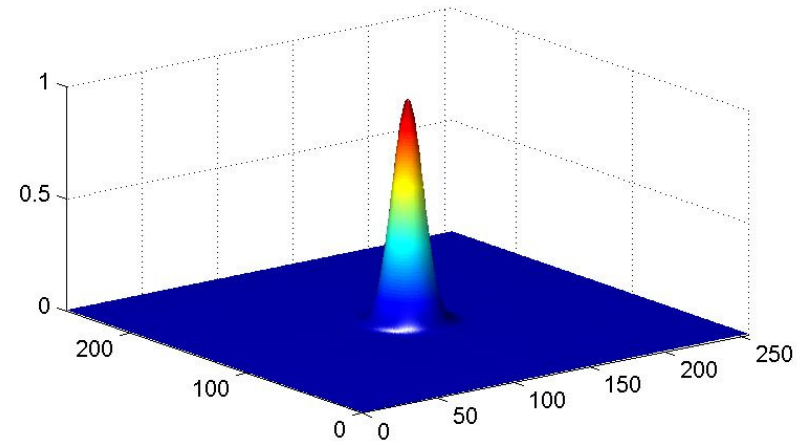
Гаусови филтри

- Степен на изглаждане може да се настройва с промяна на параметъра σ
 - зависимост между размера на локалната апертура n и стойността на параметъра σ

$$n = 3\sqrt{2}\sigma$$



$$n = 2\sqrt{2}\sigma$$



Изострящи филтри

- Използват се за “подсилване” на визуалното представяне на детайлите в изображението
- Базират се на определяне на контрастни изменения в стойностите на съседни пиксели и увеличаване на тези контрастни разлики
 - **определяне на контрастни изменения**
 - *edge detection*
 - **изострящи филтри**
 - *sharpen, enhancement*

Изострящи филтри

- Реализират се чрез сумиране на оригиналното изображение с изображение, в което е приложен филтър за определяне на контрастни изменения

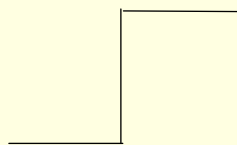
- Базират се на **пространствено диференциране**
 - разлика на обработвания пиксел и съседна стойност

диференциране в дискретната област \equiv *крайна разлика*

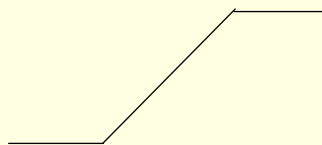
- **Edge**
 - дъга, контур, ръб, граница
 - **съществено локално изменение** в стойностите на интензитета, формирано на границата между съседни пиксели

Видове контури

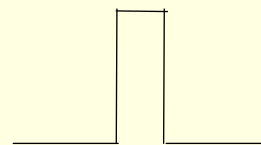
- В зависимост от характера на локалното изменение на интензитета
 - **Стъпаловидни (step)**
 - идеална стъпаловидна промяна в стойностите на интензитета в два съседни пиксела
 - **Линейни (line)**
 - две последователни скокообразни изменения в стойностите на интензитета в малка локална област
 - **Наклонени (ramp)**
 - стъпаловидно изменение на стойностите на интензитета в малка локална околност, а не в два съседни пиксела
 - **Покривообразни (roof)**
 - линейна промяна в стойностите на интензитета в малка околност



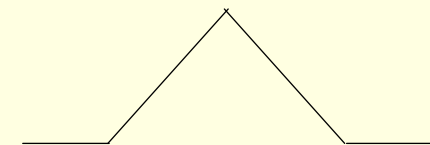
стъпаловиден



наклонен



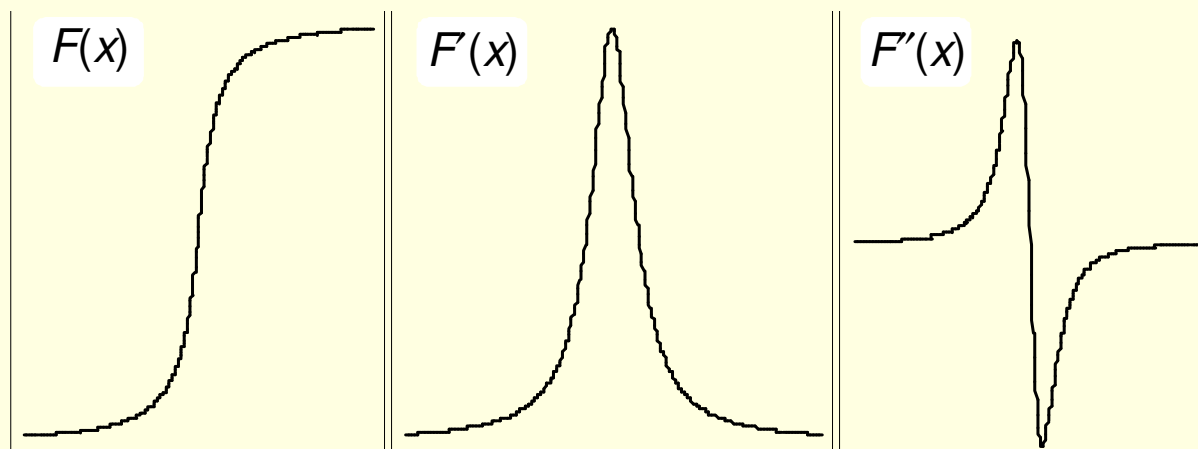
линеен



покривообразен

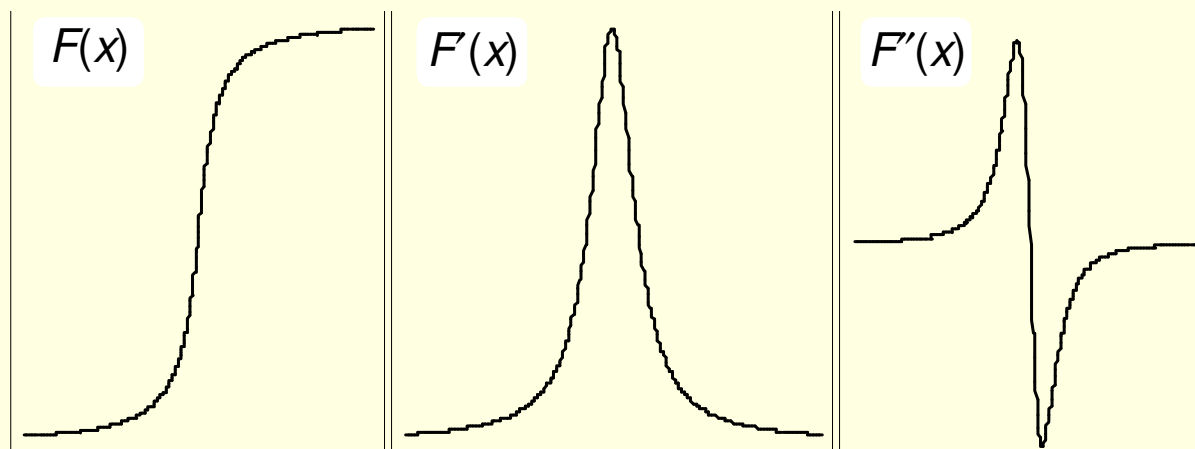
Изострящи филтри

- Откриват скокообразни изменения в стойностите на интензитета чрез **оценка на градиента**
 - изменението на интензитета е свързано с
 - локален пик в първата производна на функцията на интензитета
 - пресичане на нулевото ниво за втората производна на функцията на интензитета



Изострящи филтри

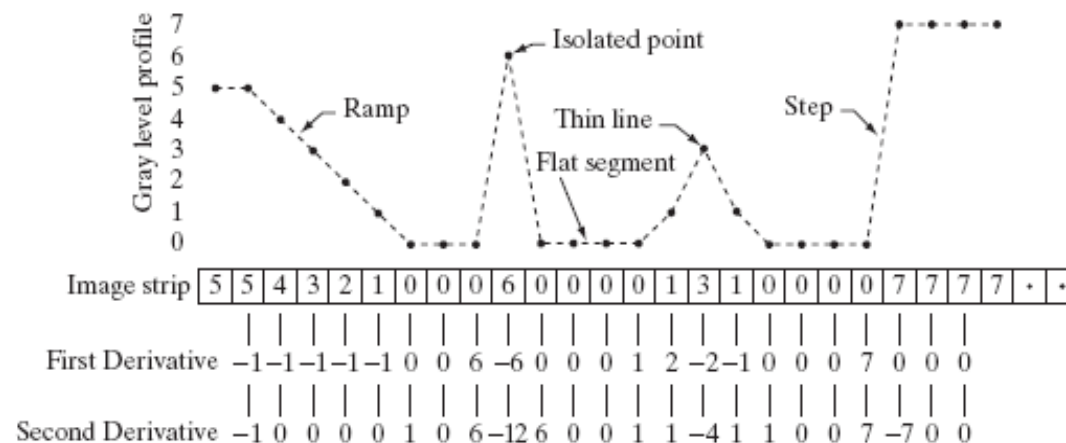
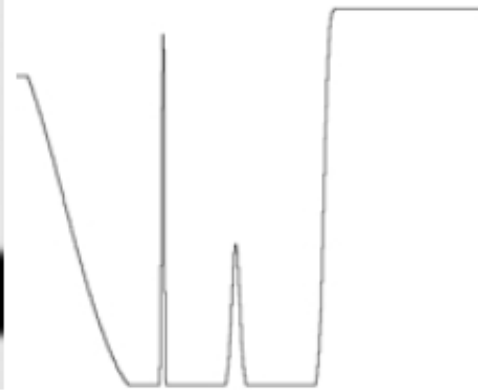
- Откриването на контурните пиксели се свежда до
 - апроксимиране на първата или втората производна на функцията на интензитета
 - определяне съответно на локален максимум или пресичане на нулата



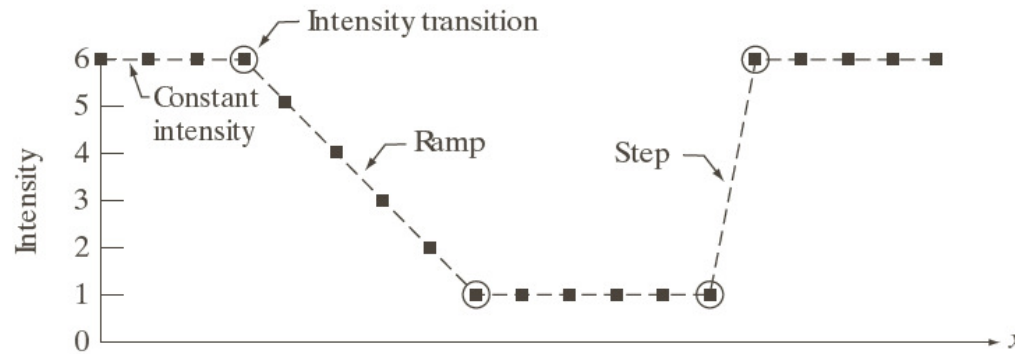
Изостражи филтри

a b
c

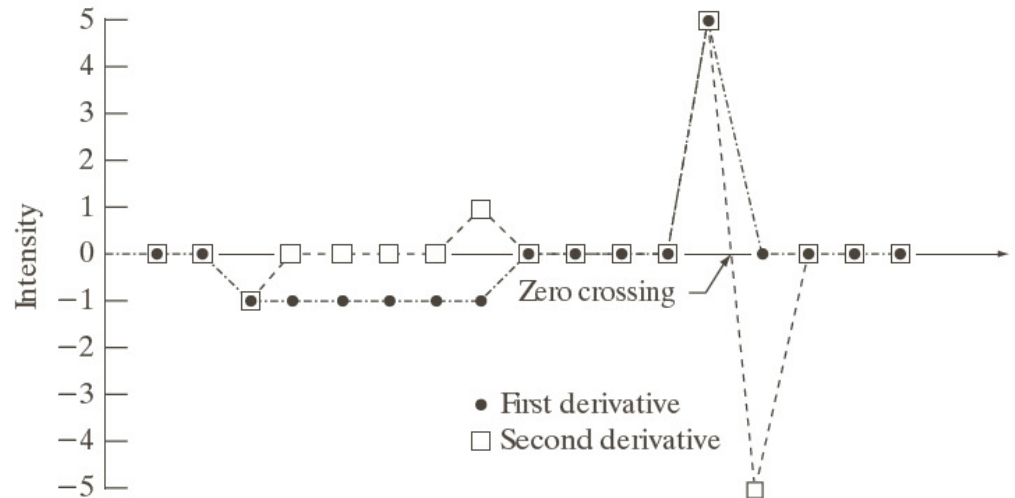
FIGURE 3.38
 (a) A simple image. (b) 1-D horizontal gray-level profile along the center of the image and including the isolated noise point.
 (c) Simplified profile (the points are joined by dashed lines to simplify interpretation).



Изостражи филтри



Scan line	6	6	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6
1st derivative	0	0	-1	-1	-1	-1	-1	0	0	0	0	0	0	5	0	0	0	0	0
2nd derivative	0	0	-1	0	0	0	0	0	1	0	0	0	0	5	-5	0	0	0	0

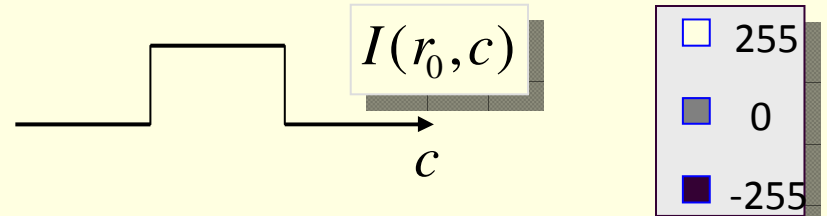
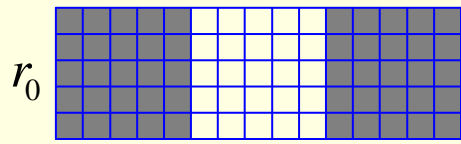


a
b
c

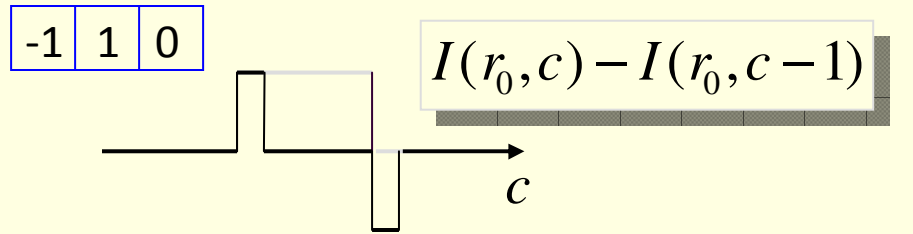
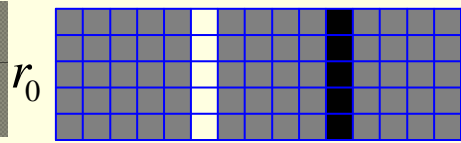
FIGURE 3.36 Illustration of the first and second derivatives of a 1-D digital function representing a section of a horizontal intensity profile from an image. In (a) and (c) data points are joined by dashed lines as a visualization aid.

Изострящи филтри

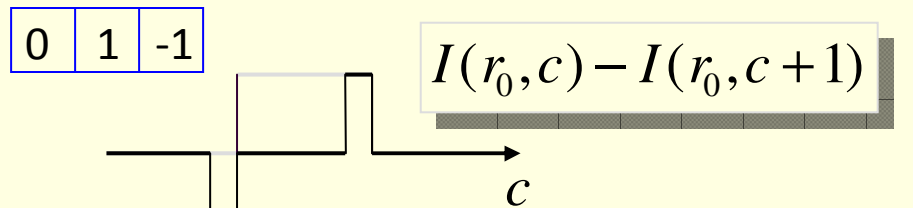
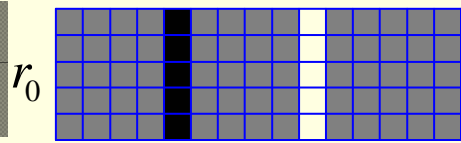
Изображение



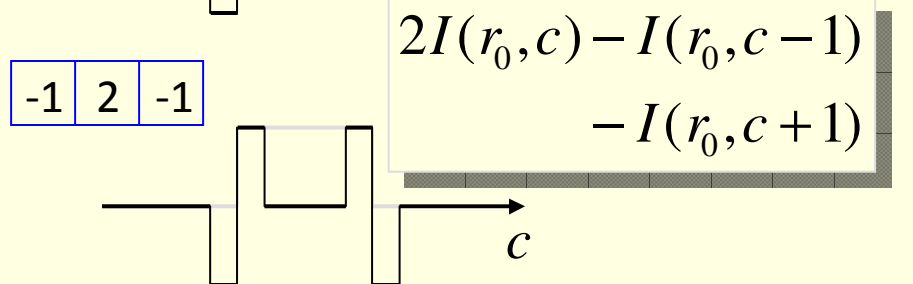
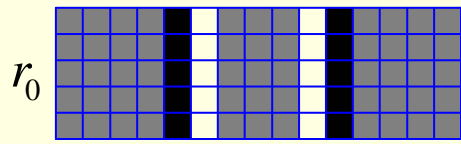
Крайна разлика
"backward"



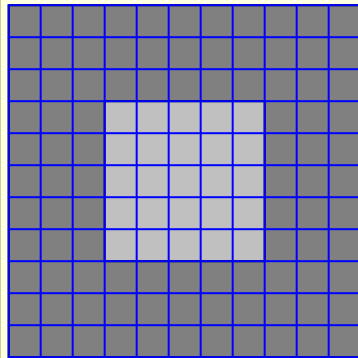
Крайна разлика
"forward"



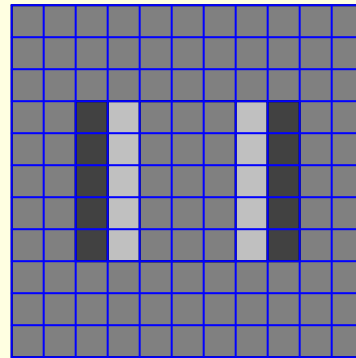
Сума на
разликите



Изостражи филтри

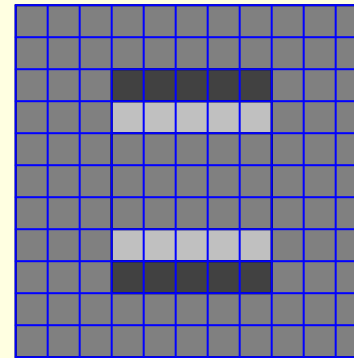


$I(r,c)$



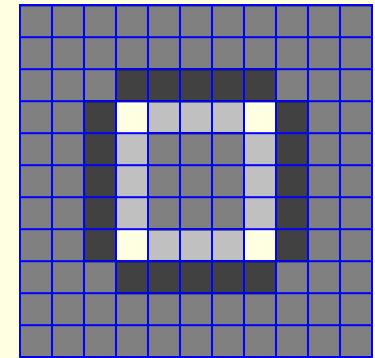
$$2I(r,c) - I(r,c-1) - I(r,c+1)$$

-1	2	-1



$$2I(r,c) - I(r-1,c) - I(r+1,c)$$

	-1	
	2	
	-1	



$$4I(r,c) - I(r-1,c) - I(r+1,c) - I(r,c-1) - I(r,c+1)$$

	-1	
-1	4	-1
	-1	

□	510
□	255
■	0
■	-255

Диференциални оператори

■ Оператор на Робъртс

$$G[f(x, y)] = |f[i, j] - f[i + 1, j + 1]| + |f[i + 1, j] - f[i, j + 1]|$$

или

$$G[f[i, j]] = |G_x| + |G_y|$$

където

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

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

Дифференциални оператори

■ Оператори на Собел и Превит

■ оператор на Собел: $c = 2$

■ оператор на Превит: $c = 1$

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -c & 0 & c \\ -1 & 0 & 1 \end{bmatrix}, G_y = \begin{bmatrix} 1 & c & 1 \\ 0 & 0 & 0 \\ -1 & -c & -1 \end{bmatrix}$$

$$[\mathbf{h}_x] = \frac{1}{3} \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \cdot [1 \ 0 \ -1]$$

$$[\mathbf{h}_y] = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \cdot [1 \ 1 \ 1]$$

$$[\mathbf{h}_x] = \frac{1}{4} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \cdot [1 \ 0 \ -1]$$

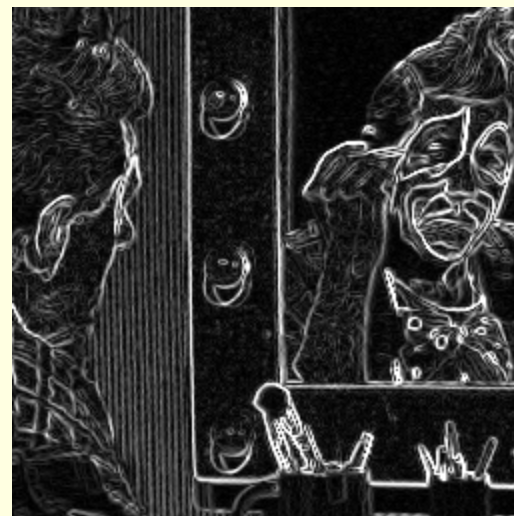
$$[\mathbf{h}_y] = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \cdot [1 \ 2 \ 1]$$

Диференциални оператори

- *Оператор на Робъртс*



оригинално изображение



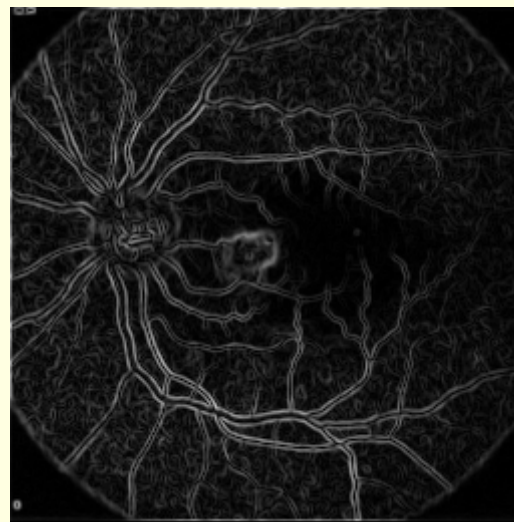
градиентно изображение

Диференциални оператори

- *Оператор на Собел*



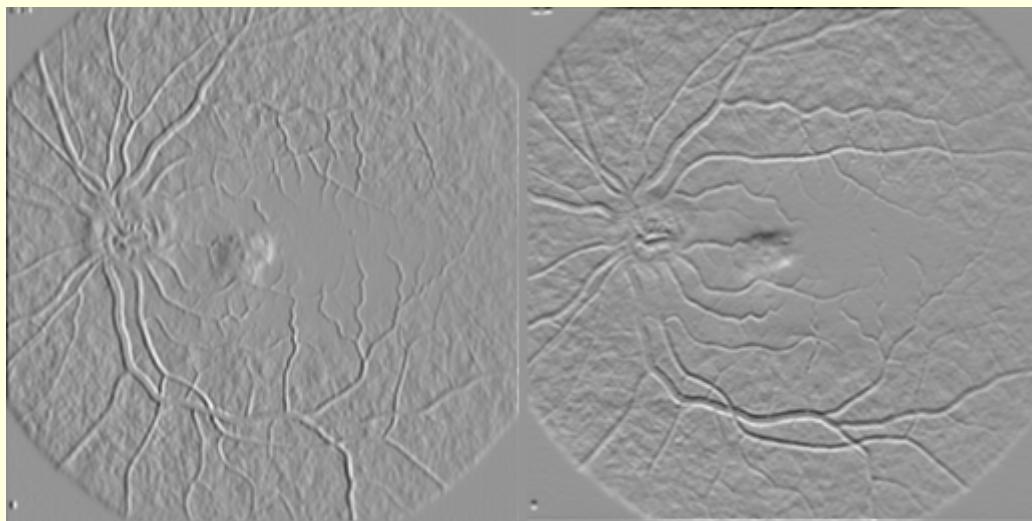
оригинално изображение



градиентно изображение

Диференциални оператори

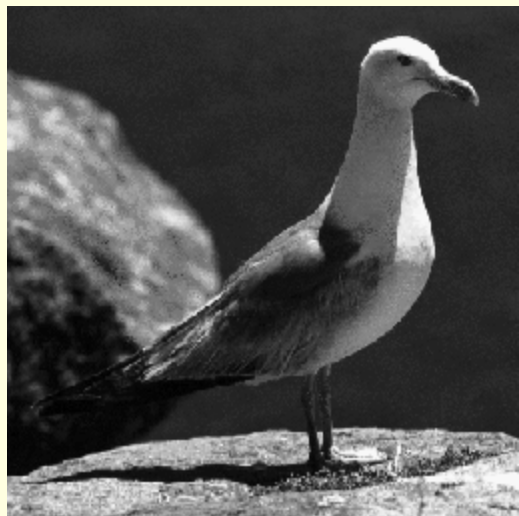
- *Оператор на Собел*



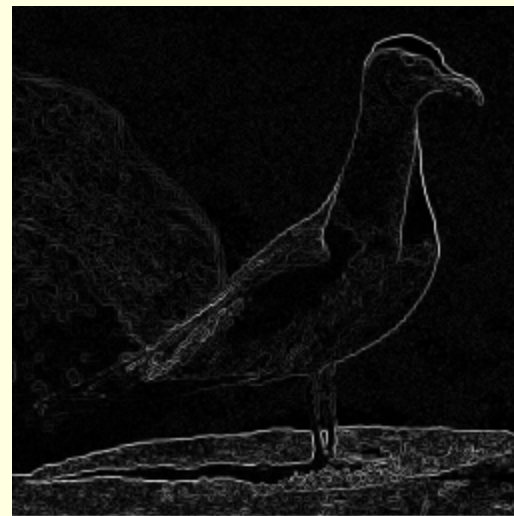
G_x
хоризонтална
компонента на
градиента

G_y
вертикална
компонента на
градиента

Диференциални оператори



Робъртс



Собел



Превит



Диференциални оператори

■ Лапласиан оператор

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

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial G_x}{\partial x} = \frac{\partial(f[i, j+1] - f[i, j])}{\partial x} = \frac{\partial f[i, j+1]}{\partial x} - \frac{\partial f[i, j]}{\partial x} =$$

$$= (f[i, j+2] - f[i, j+1]) - (f[i, j+1] - f[i, j]) = f[i, j+2] - 2f[i, j+1] + f[i, j]$$

$$\frac{\partial^2 f}{\partial x^2} = f[i, j+1] - 2f[i, j] + f[i, j-1]$$

$$\frac{\partial^2 f}{\partial y^2} = f[i+1, j] - 2f[i, j] + f[i-1, j]$$

Диференциални оператори

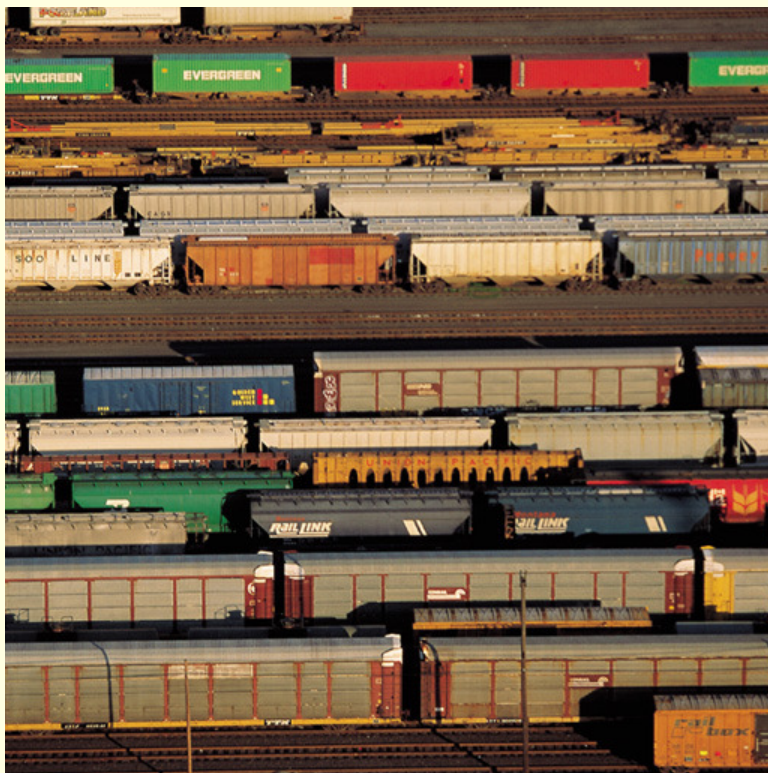
■ Лапласиан оператор

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

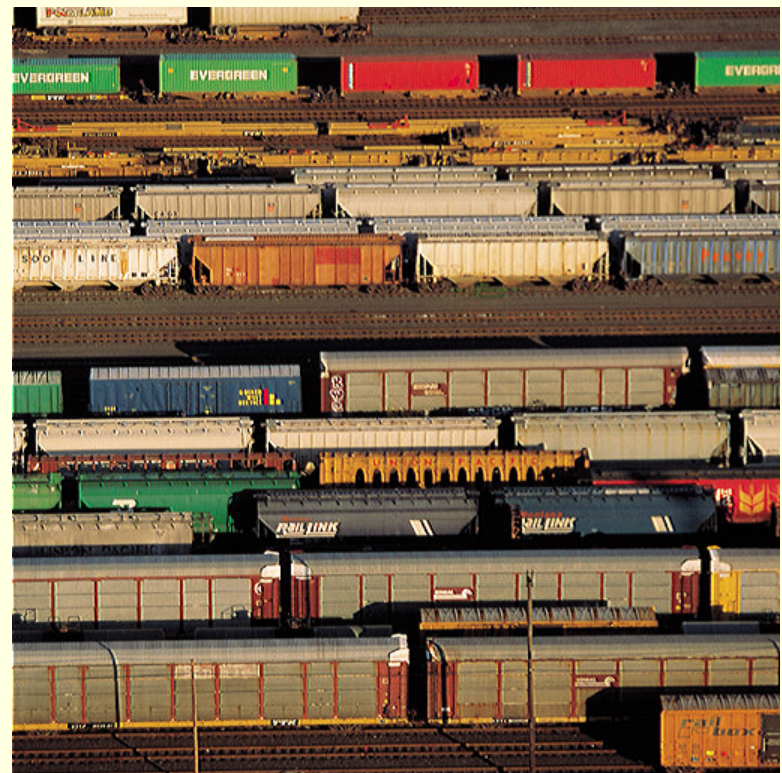
$$\nabla^2_{(4)} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \equiv \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} + [1 \quad -2 \quad 1]$$

$$\nabla^2_{(8)} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Изострящи филтри



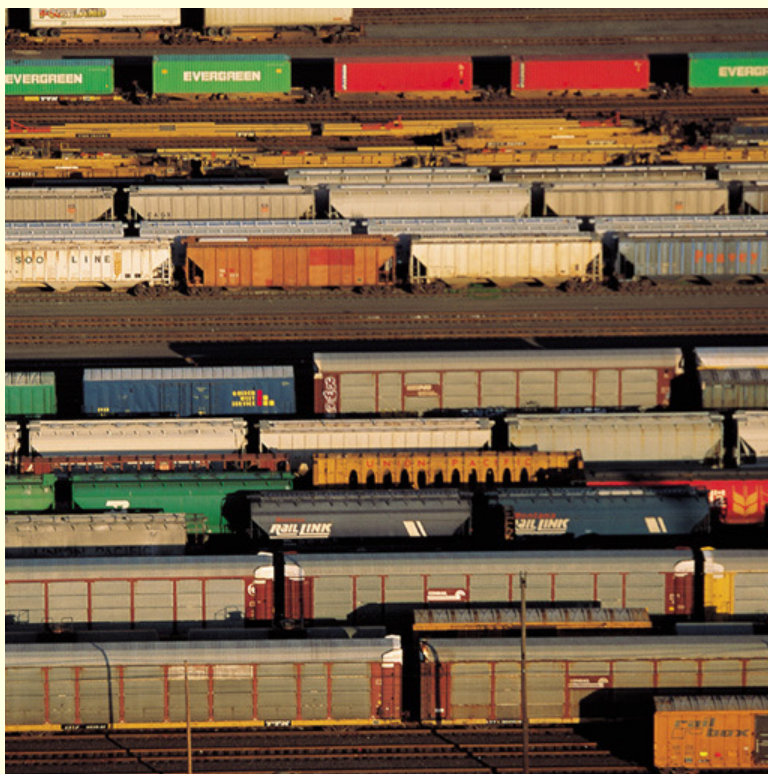
изображение I



лява разлика

$$I + (I * h) = I + (I * [-1 \ 1])$$

Изострящи филтри



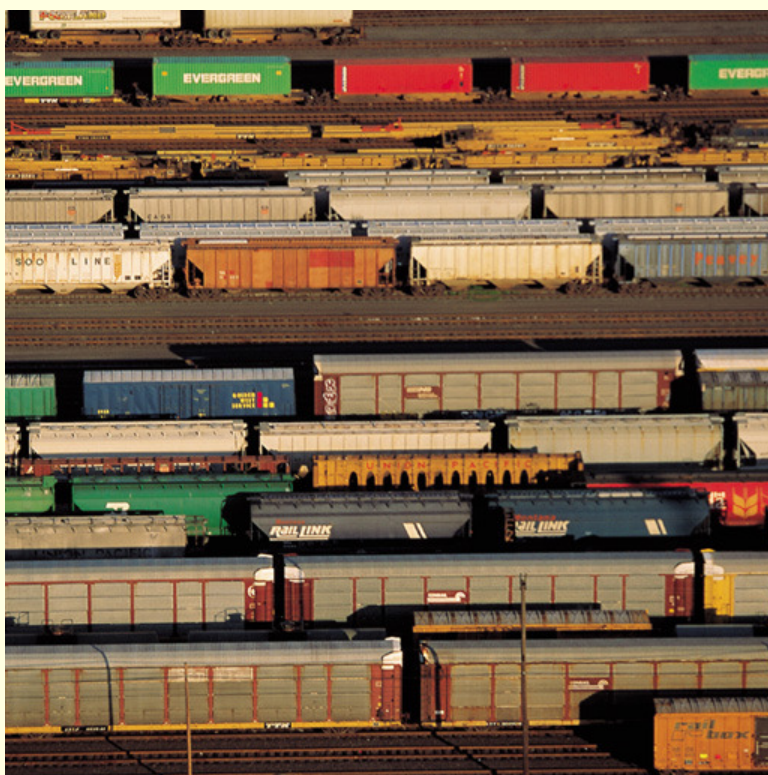
изображение I



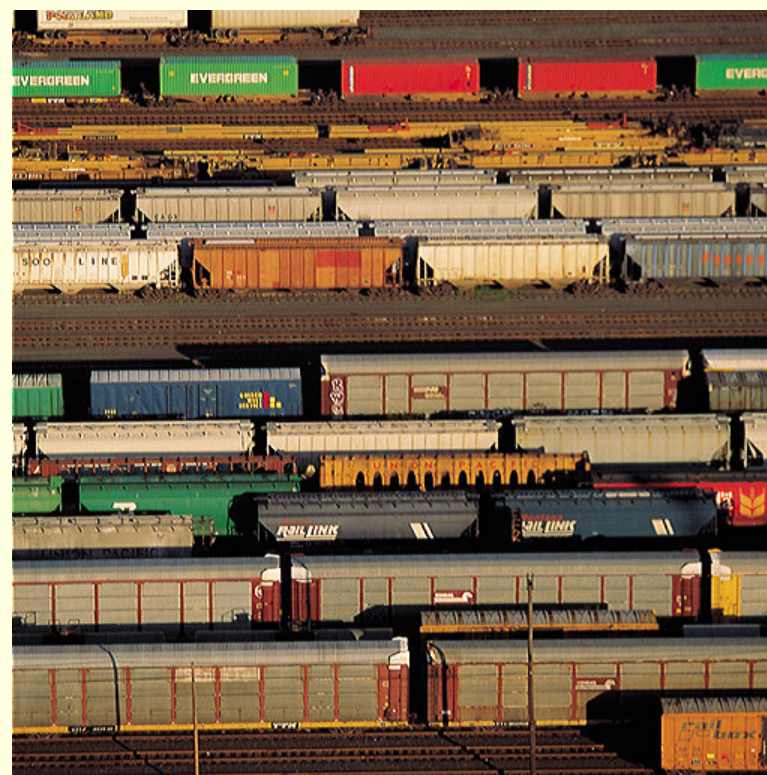
дясна разлика

$$I+(I*h) = I+(I*[1 - 1])$$

Изострящи филтри



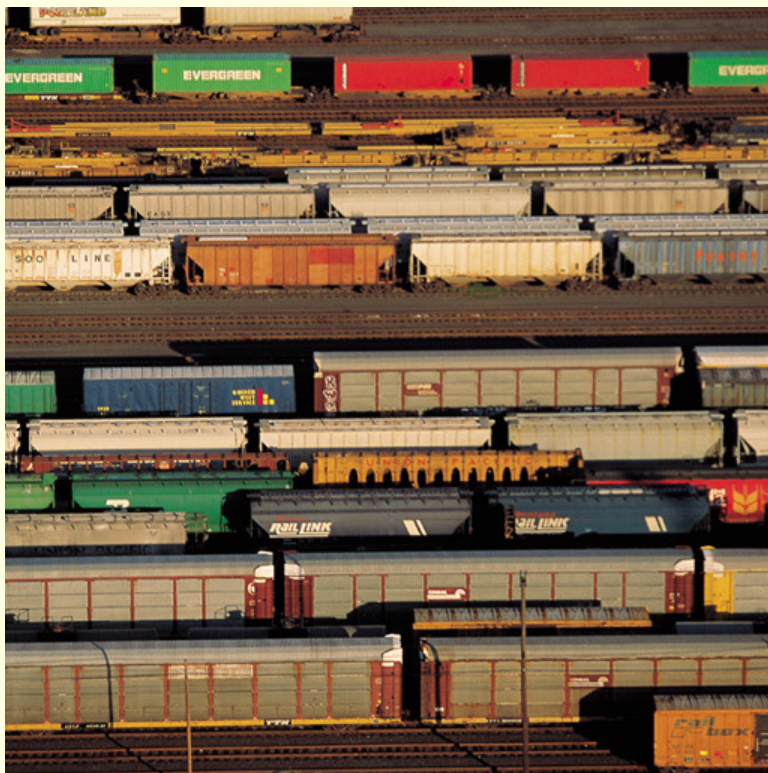
изображение I



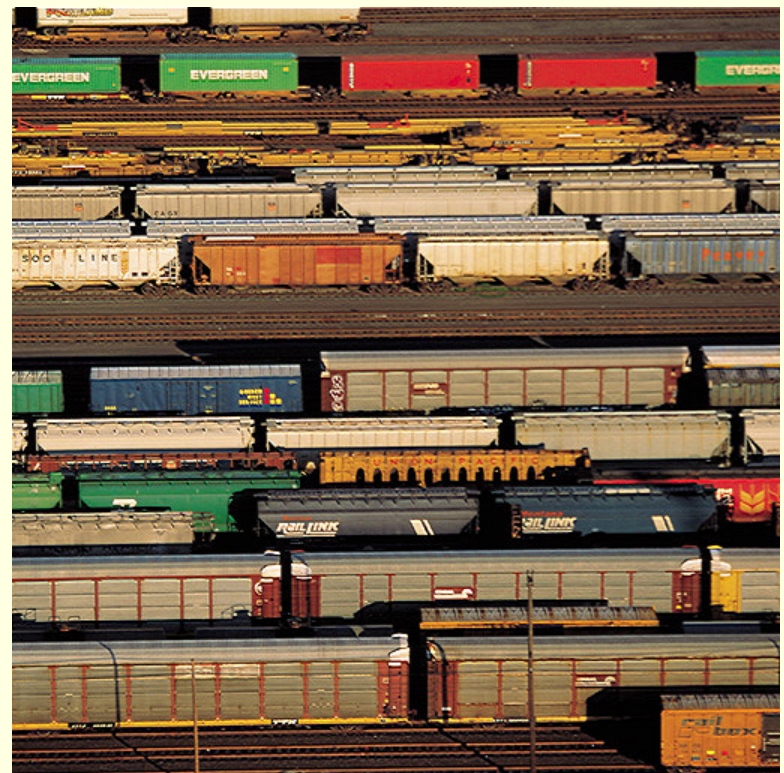
вертикални контури

$$I+(I*h) = I+(I*[-1 \ 2 \ -1])$$

Изостражи филтри



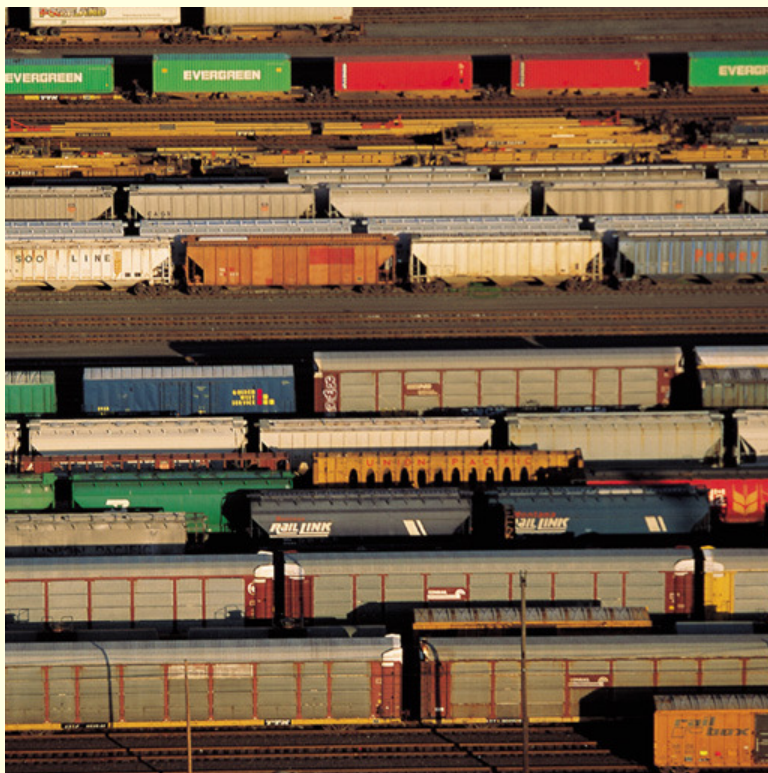
изображение I



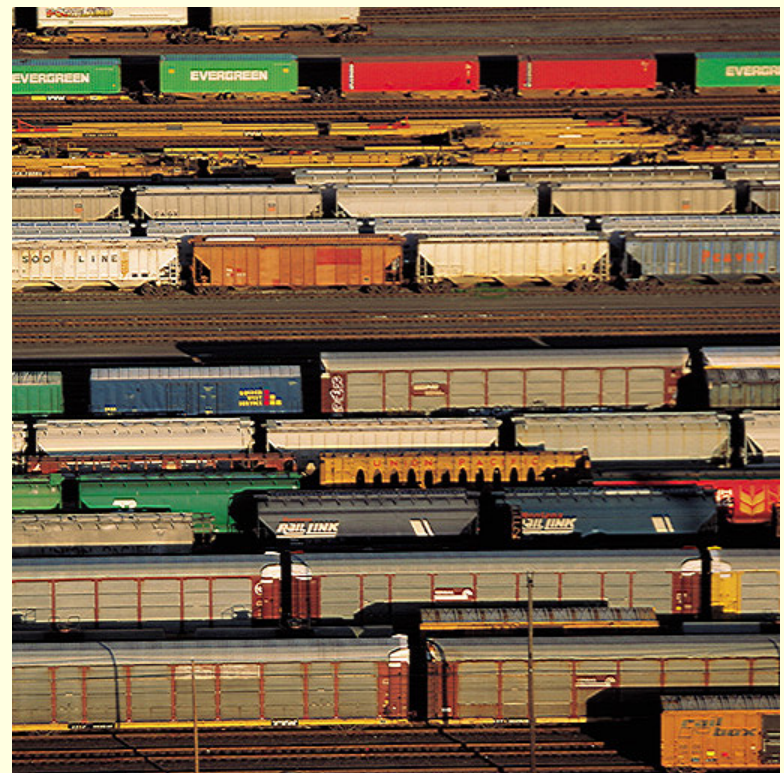
долна разлика

$$I + (I * h) = I + (I * \begin{bmatrix} -1 \\ 1 \end{bmatrix})$$

Изострящи филтри



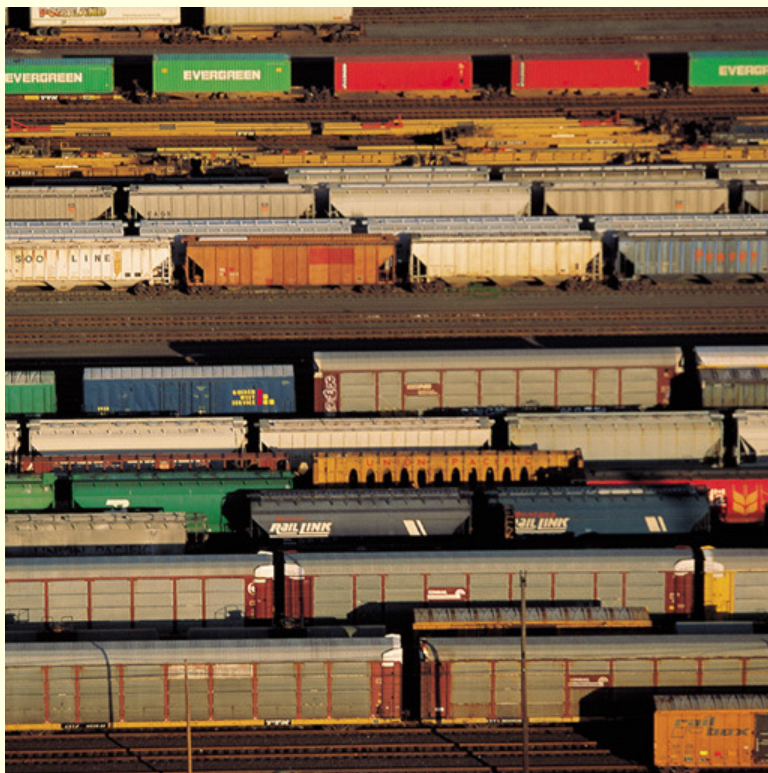
изображение I



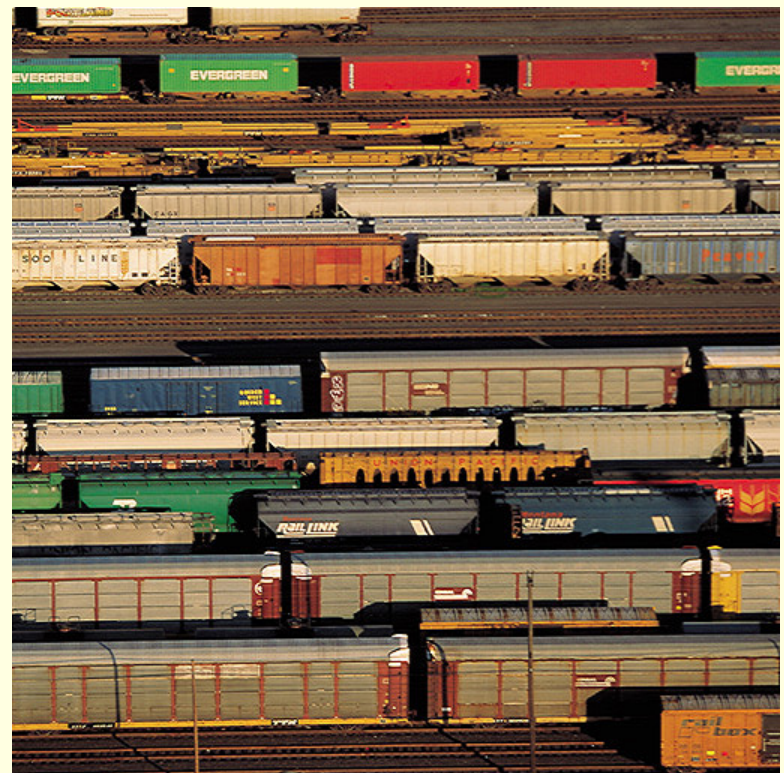
горна разлика

$$I + (I * h) = I + (I * \begin{bmatrix} -1 \\ 1 \end{bmatrix})$$

Изострящи филтри



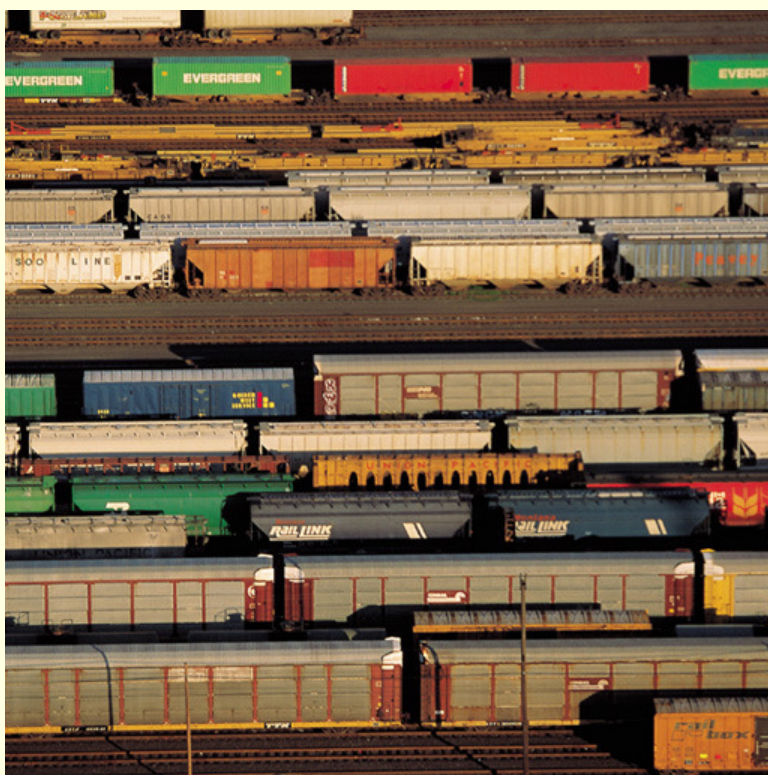
изображение I



горизонтални контури

$$I+(I*h) = I+(I* \begin{bmatrix} -1 \\ 2 \\ -1 \end{bmatrix})$$

Изострящи филтри



изображение /



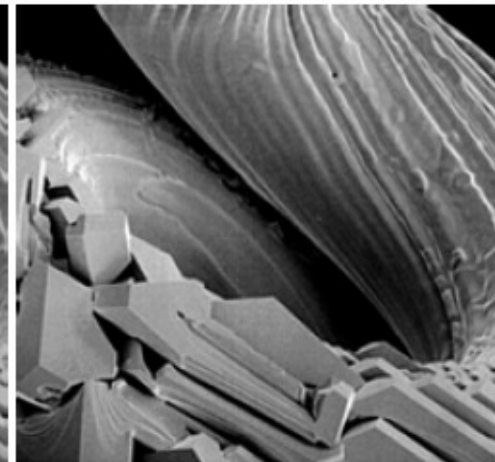
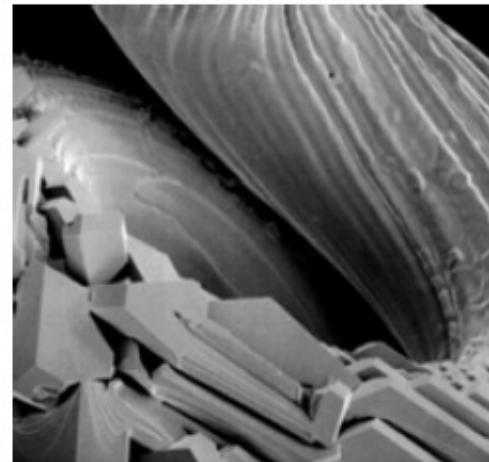
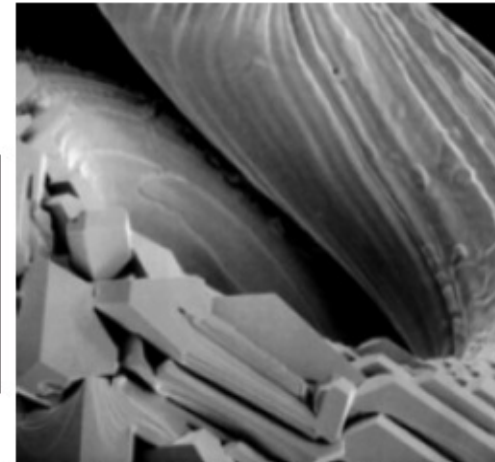
изострено изображение

Изострящи филтри

- *Изострящ филтър с Лапласиан оператор*

0	-1	0
-1	5	-1
0	-1	0

-1	-1	-1
-1	9	-1
-1	-1	-1



a b c
d e

FIGURE 3.41 (a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

Изострящи филтри

a b
c d

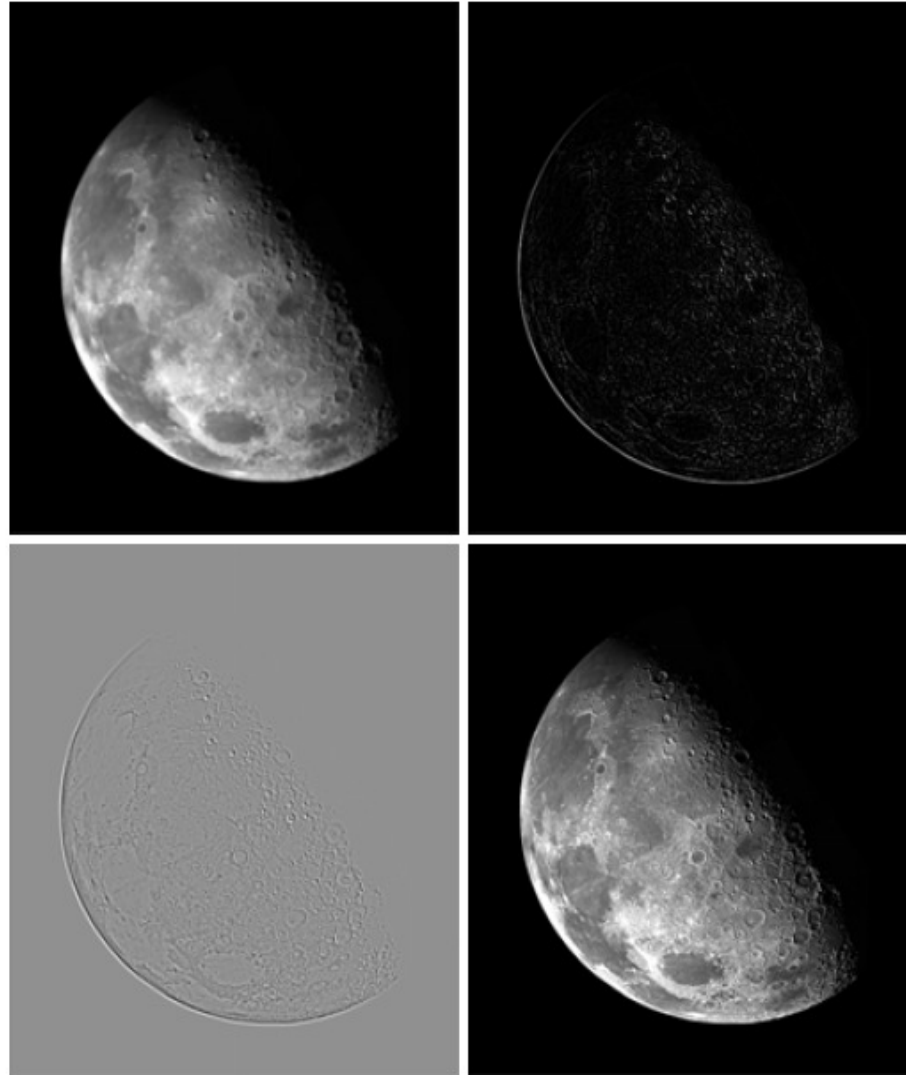
FIGURE 3.40

(a) Image of the North Pole of the moon.

(b) Laplacian-filtered image.

(c) Laplacian image scaled for display purposes.

(d) Image enhanced by using Eq. (3.7-5). (Original image courtesy of NASA.)



Изострящи филтри

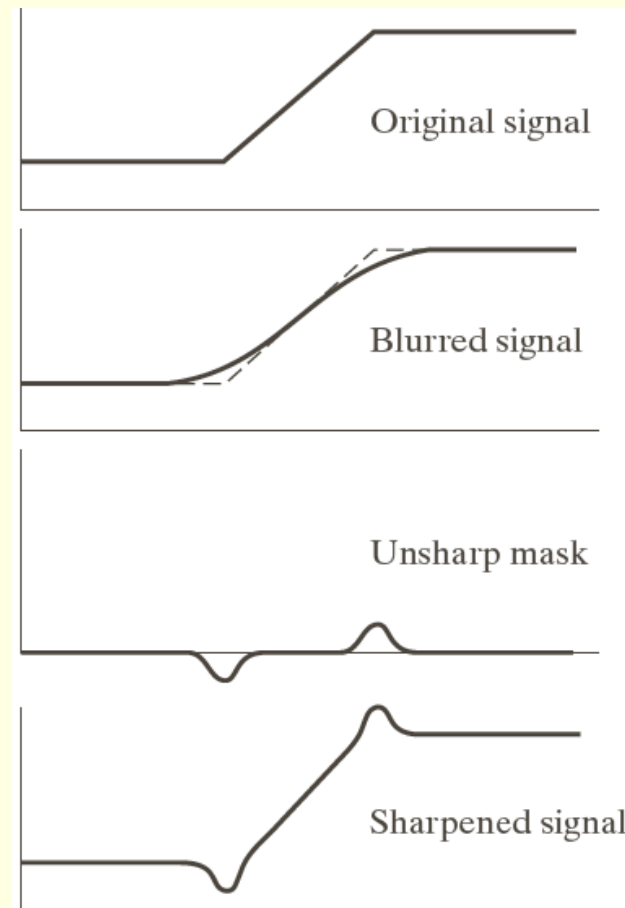
- **Unsharp masking**

- разлика между
 - оригиналното изображениеи изгладеното изображение

$$f_{sharp}(x, y) = A.f(x, y) - \bar{f}(x, y)$$

$f(x, y)$

$\bar{f}(x, y)$



a
b
c
d

FIGURE 3.39 1-D illustration of the mechanics of unsharp masking. (a) Original signal. (b) Blurred signal with original shown dashed for reference. (c) Unsharp mask. (d) Sharpened signal, obtained by adding (c) to (a).

Изострящи филтри

■ High-boost filtering

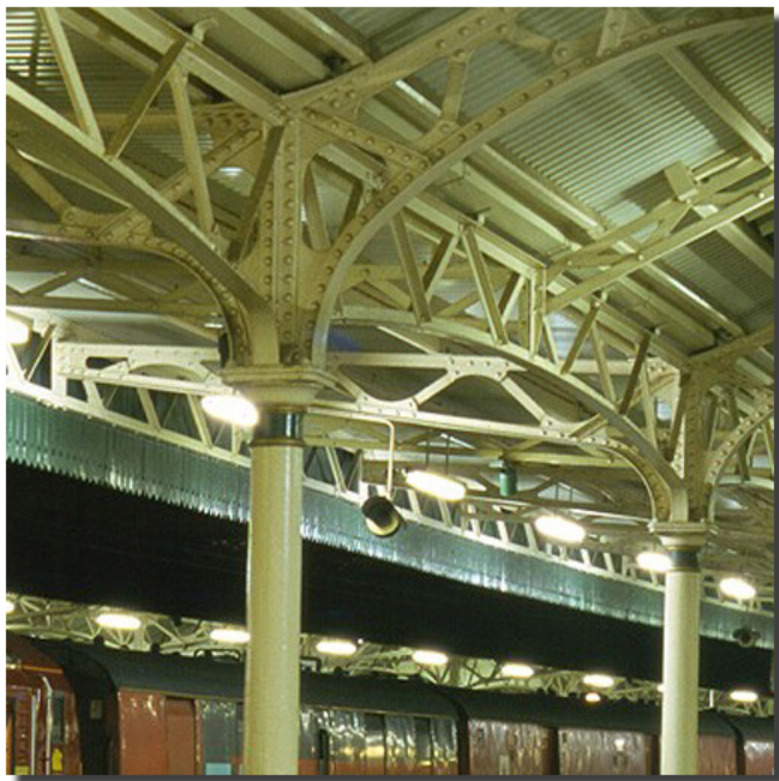
$$f_{HB}(x, y) = f(x, y) - \bar{f}(x, y) \quad A \geq 1$$

$$f_{HB}(x, y) = (A - 1)f(x, y) + f(x, y) - \bar{f}(x, y)$$

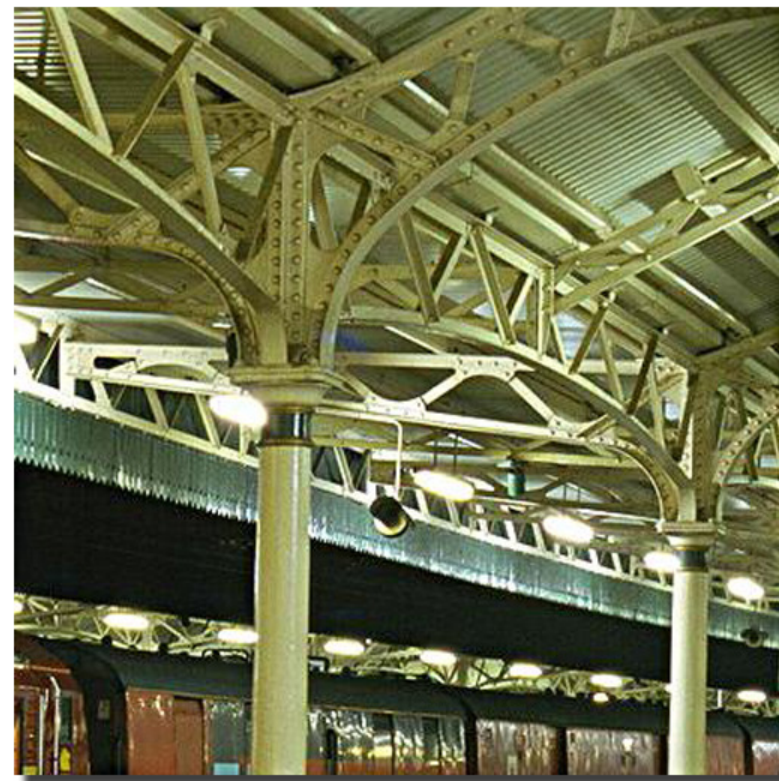
$$f_{HB}(x, y) = (A - 1)f(x, y) + f_{sharp}(x, y)$$

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & A+4 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & A+8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Изострящи филтри



изображение /



unsharp mask

Изострящи филтри

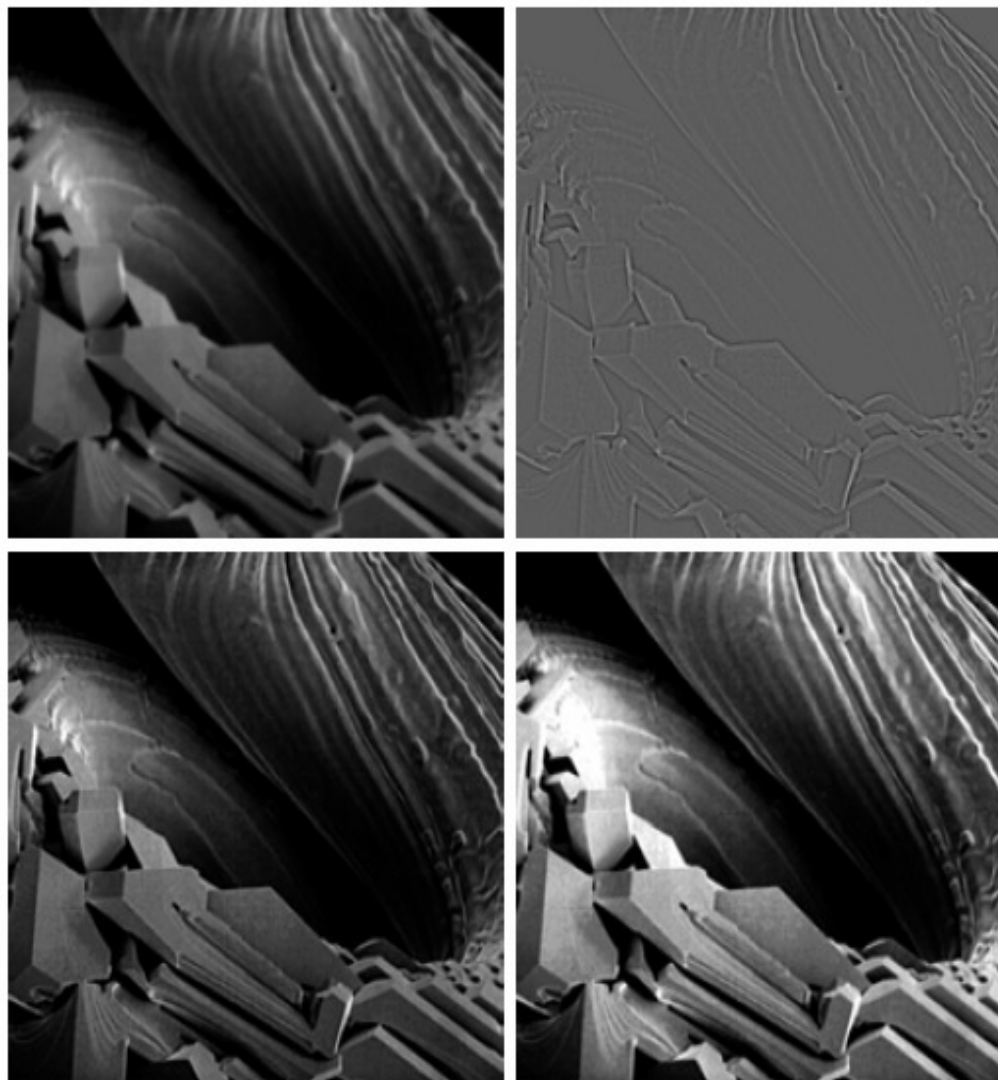
a b
c d

FIGURE 3.43

(a) Same as Fig. 3.41(c), but darker.

(b) Laplacian of (a) computed with the mask in Fig. 3.42(b) using $A = 0$.

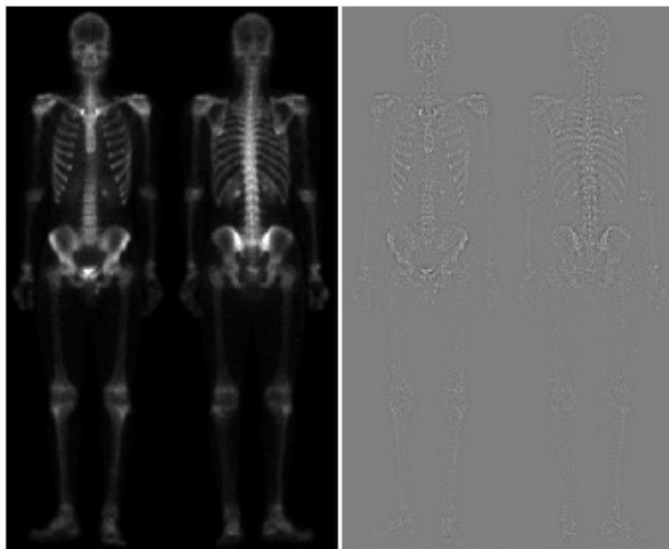
(c) Laplacian enhanced image using the mask in Fig. 3.42(b) with $A = 1$. (d) Same as (c), but using $A = 1.7$.



Изостраци филтри

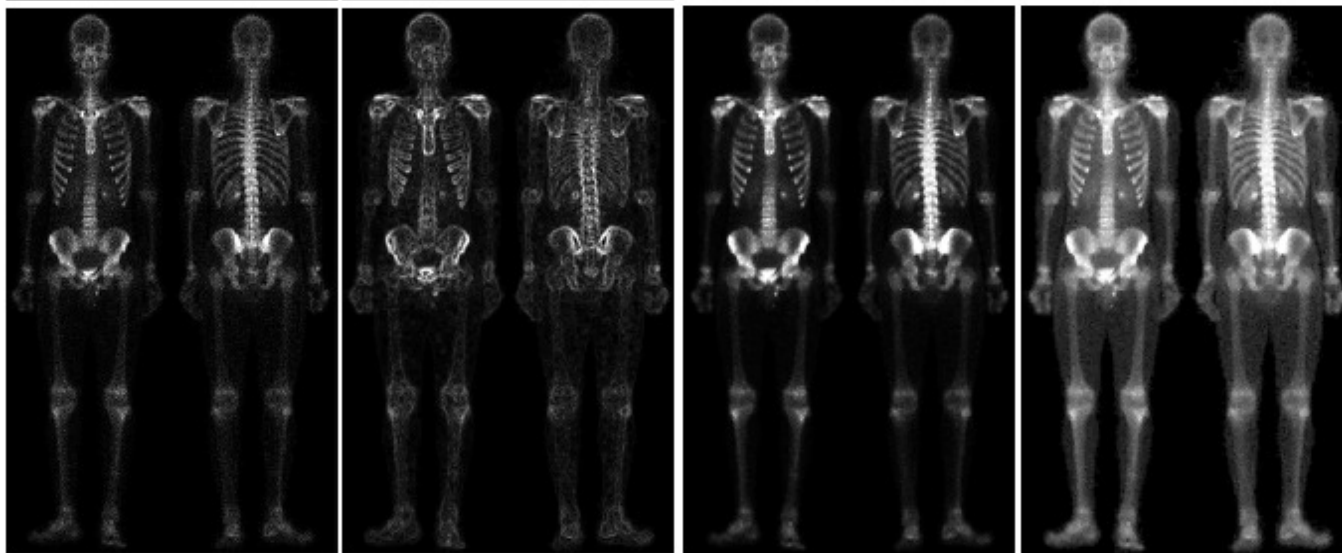
a b
c d

FIGURE 3.46
(a) Image of whole body bone scan.
(b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b). (d) Sobel of (a).



e f
g h

FIGURE 3.46
(Continued)
(e) Sobel image smoothed with a 5×5 averaging filter. (f) Mask image formed by the product of (c) and (e). (g) Sharpened image obtained by the sum of (a) and (f). (h) Final result obtained by applying a power-law transformation to (g). Compare (g) and (h) with (a). (Original image courtesy of G.E. Medical Systems.)



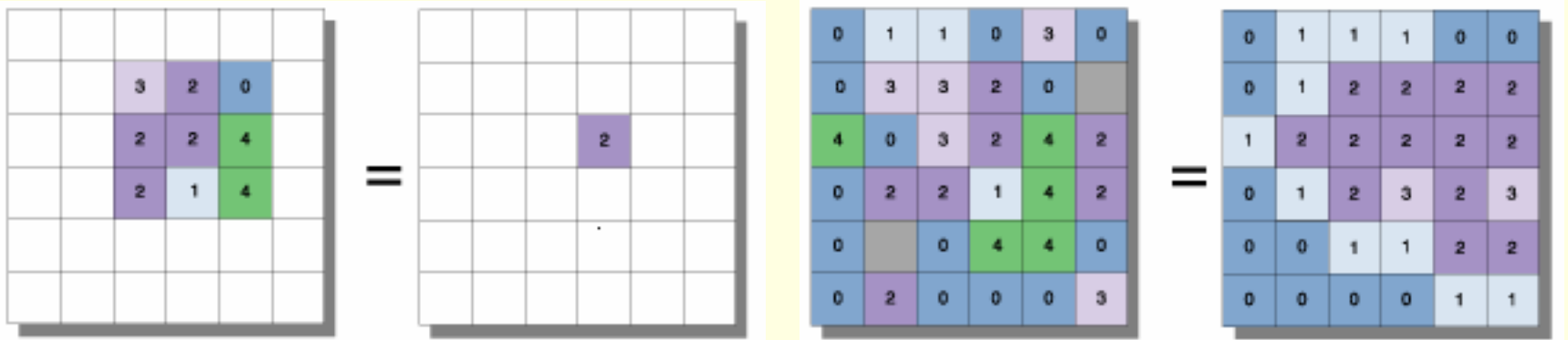
Нелинейни операции

- Базират се на **нелинейна трансформация** на стойностите в малка локална околност
 - ***филтри базирани на подреждане***
 - *rank-ordered*
 - *ordered-statistics*
 - стойността на централния пиксел се заменя със стойност, получена в резултат на подреждане или статистически определена за стойностите в локалната околност
 - *медиана*
 - *минимална стойност*
 - *максимална стойност*
 - *мода*

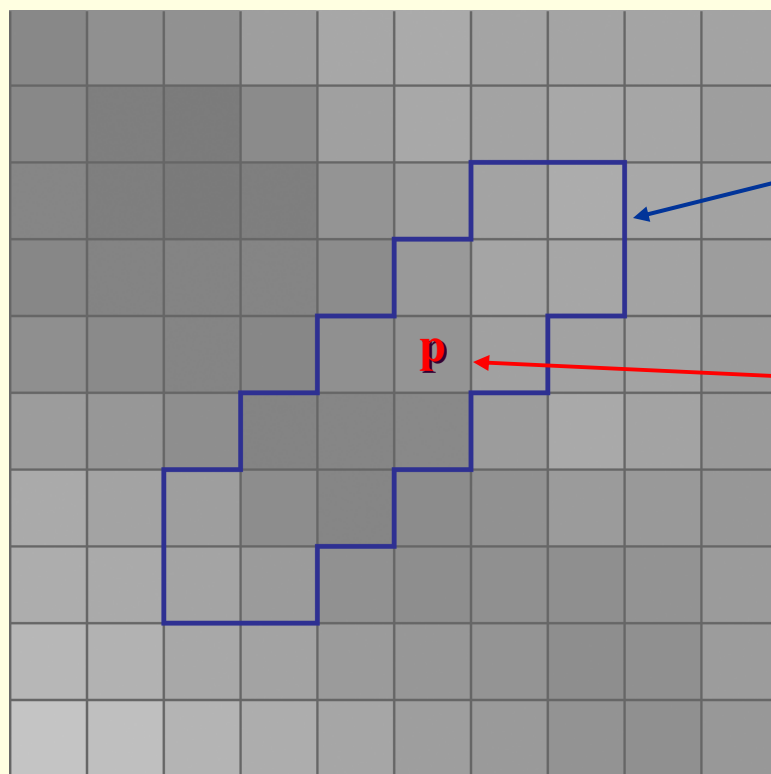
Медианен филтър

- За всеки пиксел се определя **медианата** за стойностите в локална околност

$$f_{med}(i, j) = MED[f(i+k, j+l)], \quad k, l = -\frac{n-1}{2}, \dots, \frac{m-1}{2}$$



Медианен филтър



сортирани
стойности в
локалната
околност на **p**

медиана
филтрирана
стойност за
пиксел на
позиция **p** в
резултантното
изображение

- 131
- 133
- 133
- 136
- 140
- 143
- 147
- 152
- 154
- 157
- 160
- 162
- 163
- 164
- 165
- 171

Медианен филтър

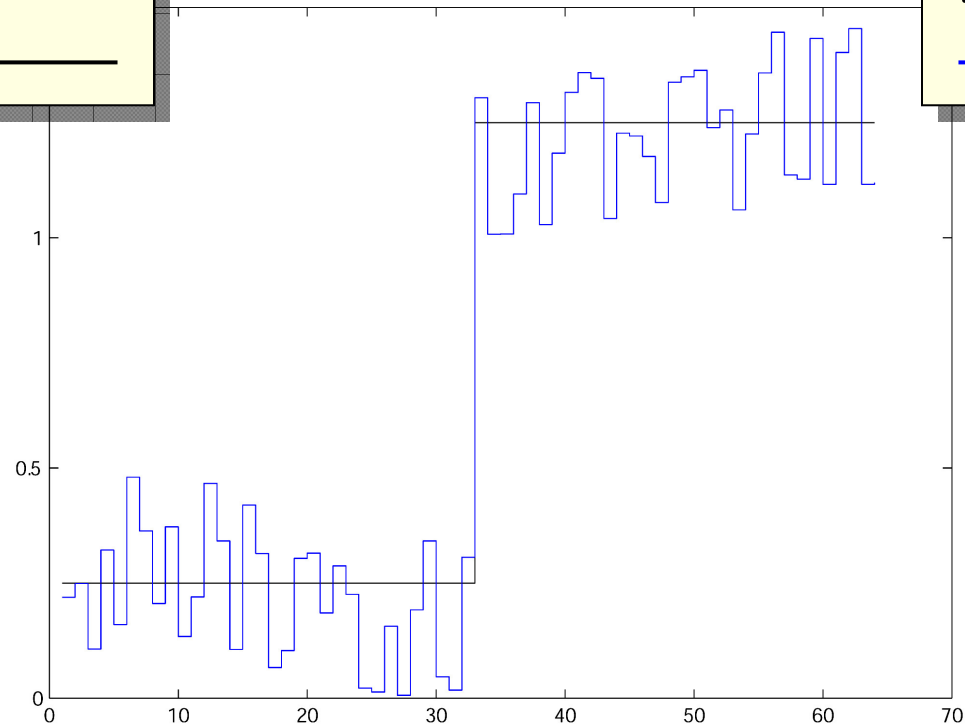
$$H(n-325)+0.25$$

$$= \begin{cases} 0.25 & \text{for } n \leq 32 \\ 1.25 & \text{for } n \geq 33 \end{cases}$$

$$H(n-325)+u(n)$$

where

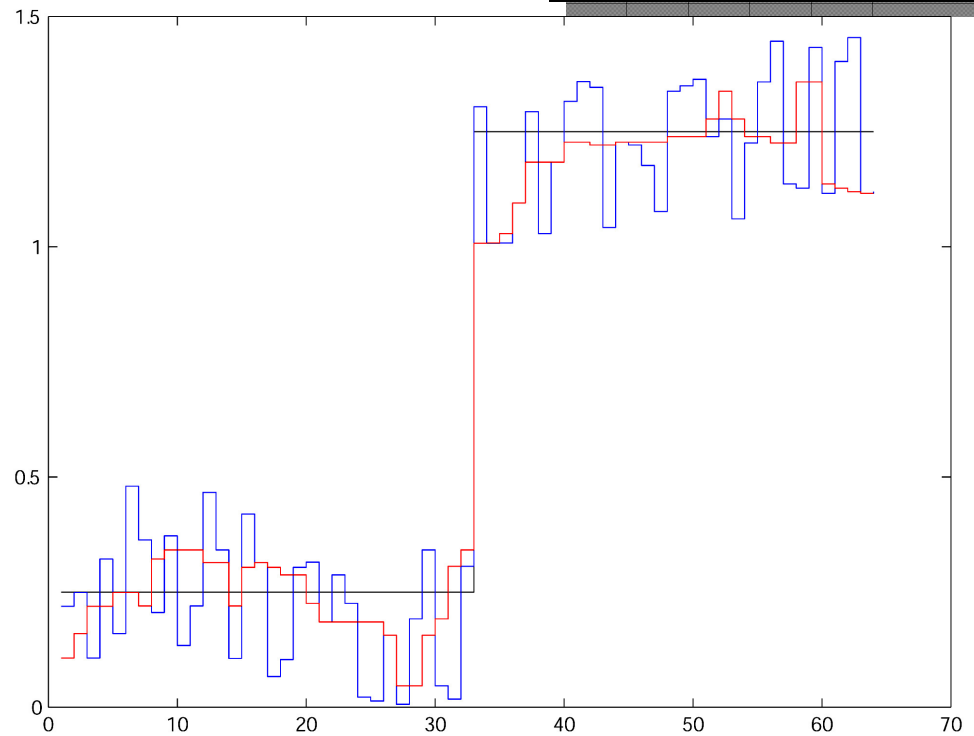
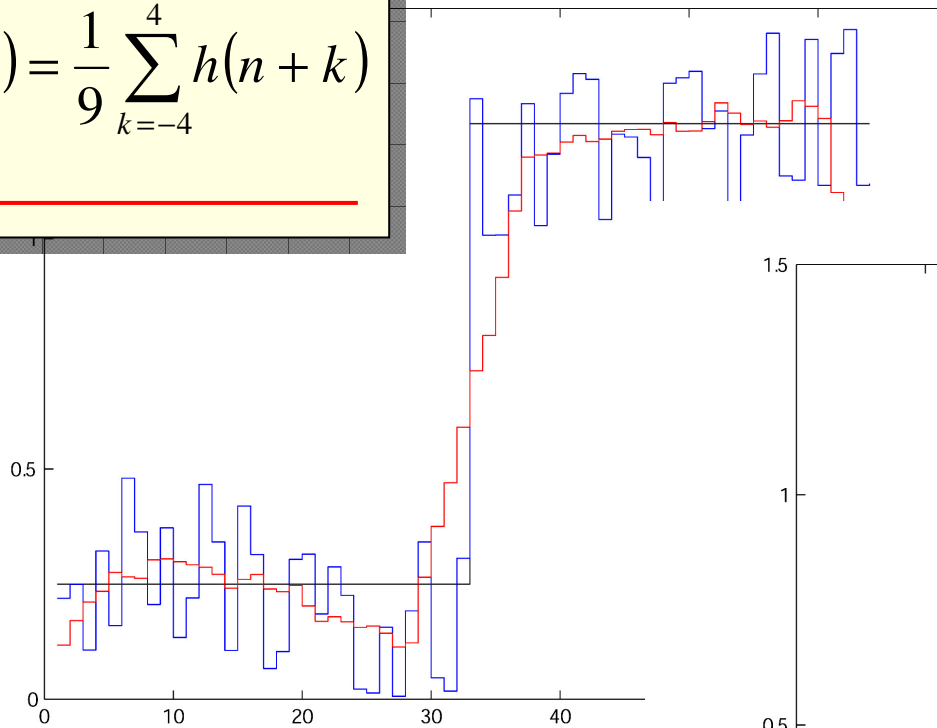
$$u(n) = \text{unif}(-0.25, 0.25)$$



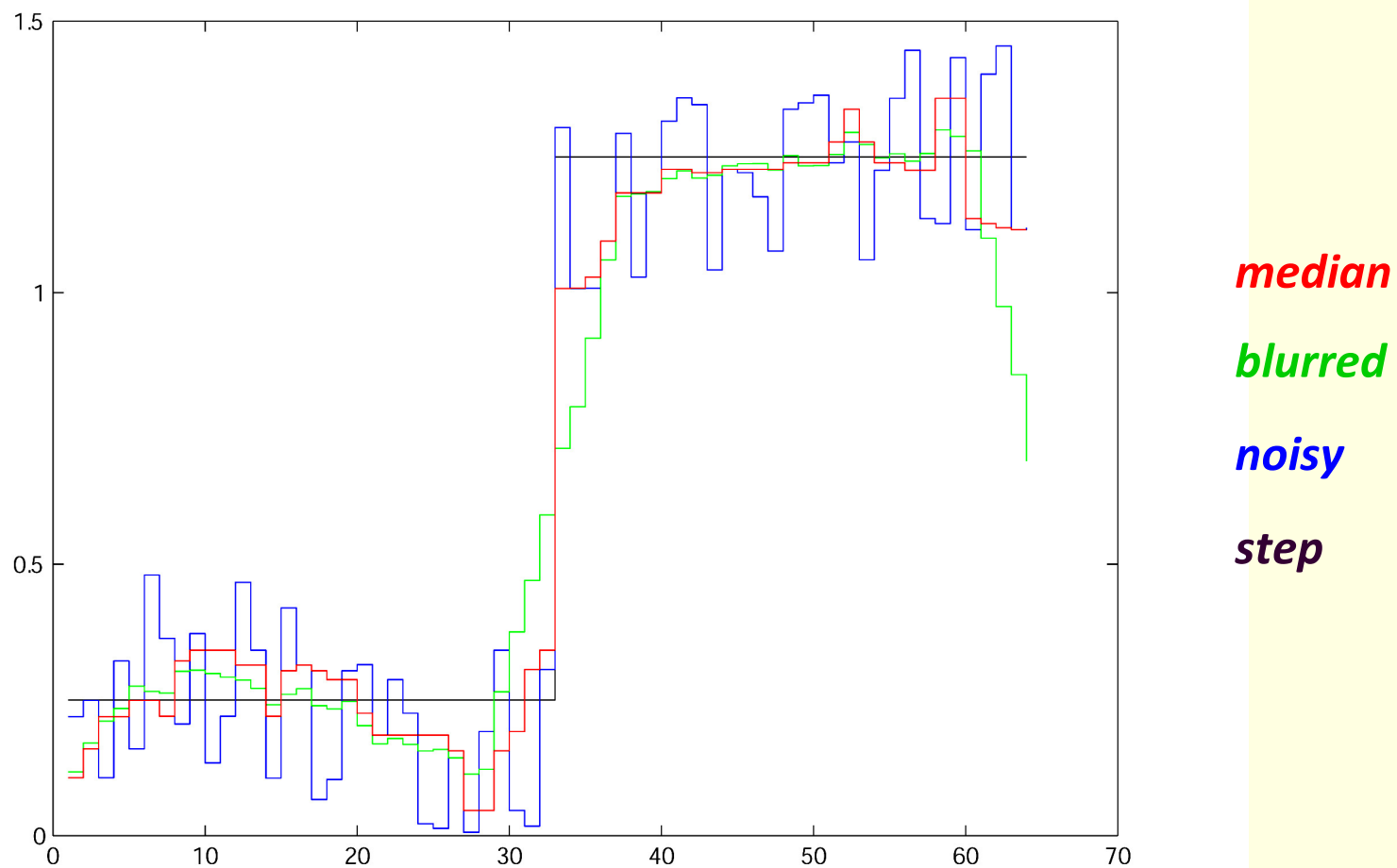
Медианен филтър

$$h(n) = \frac{1}{9} \sum_{k=-4}^4 h(n+k)$$

$$h(n) = \text{med}\{h(n+k)\}_{k=-4}^4$$

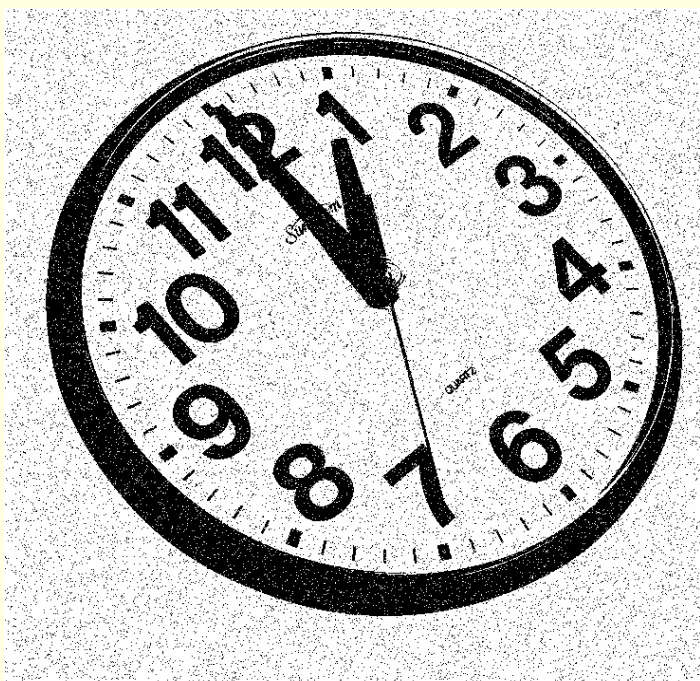


Медианен филтър

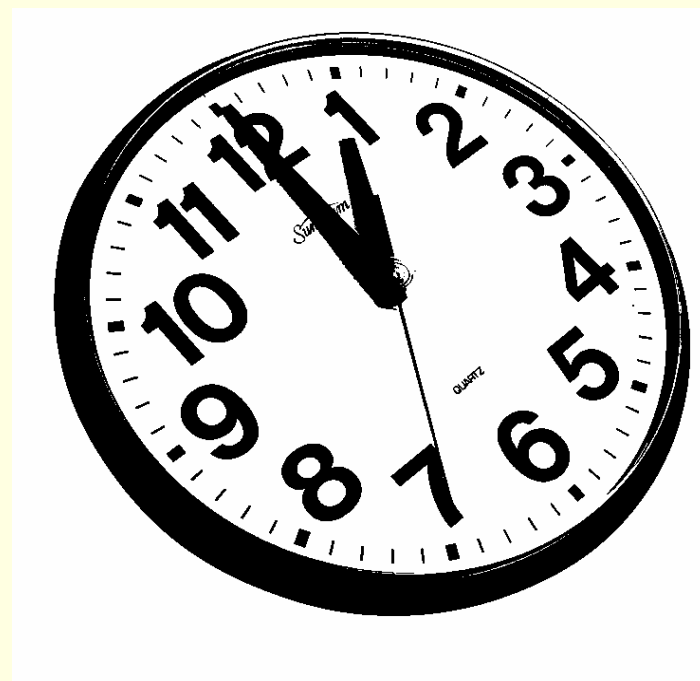


Медианният филтър запазва контурите по-добре от филтъра по средна стойност⁸⁴

Медианен филтър

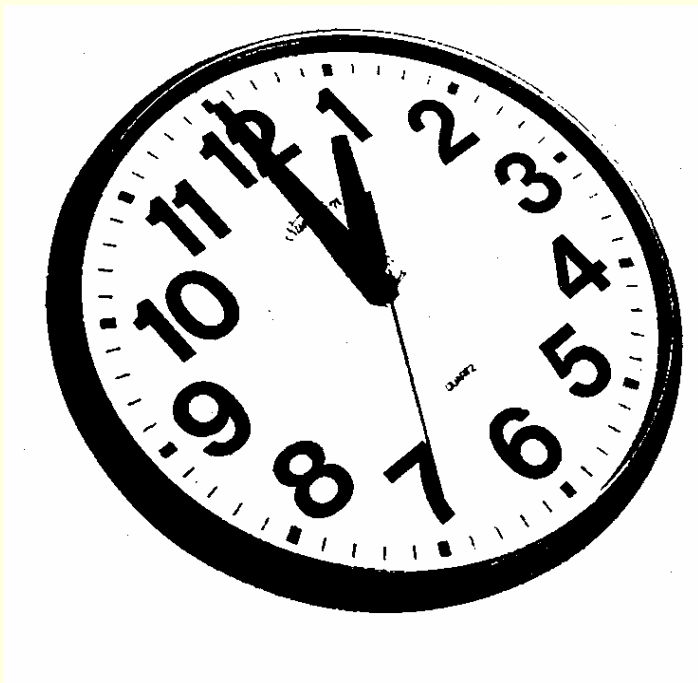


изображение с шум

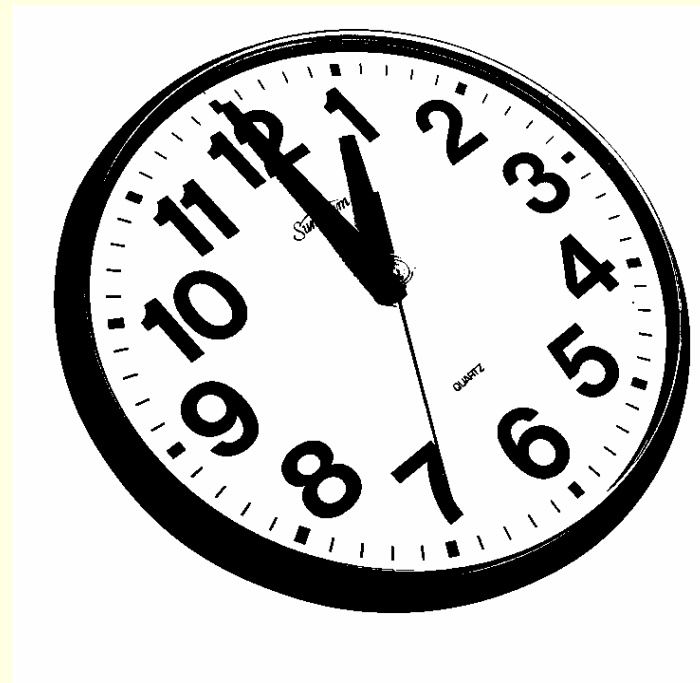


оригинално изображение

Медианен филтър

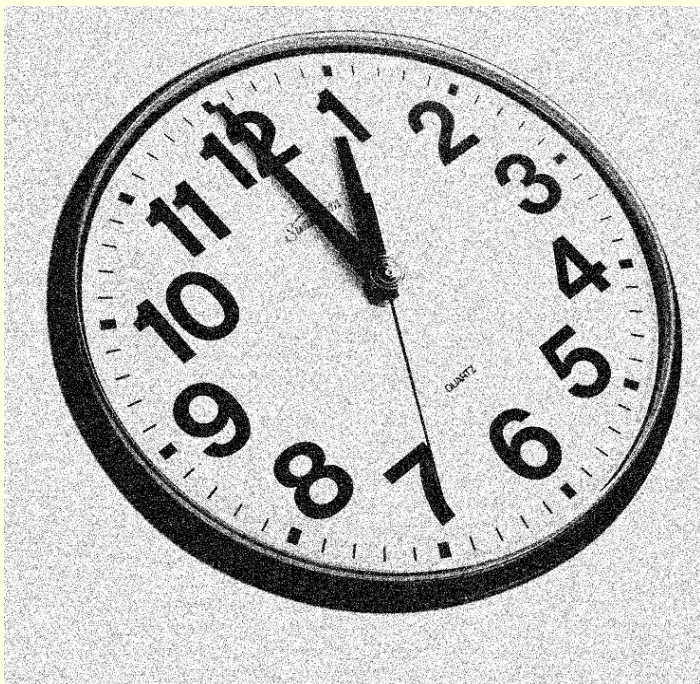


медианен филтър за
зашуменото изображение



оригинално
изображение

Медианен филтър

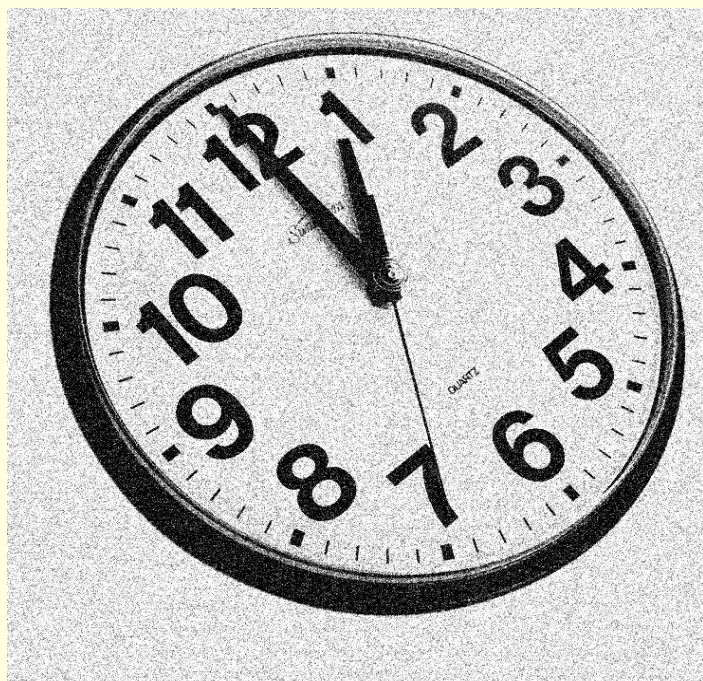


изображение с шум

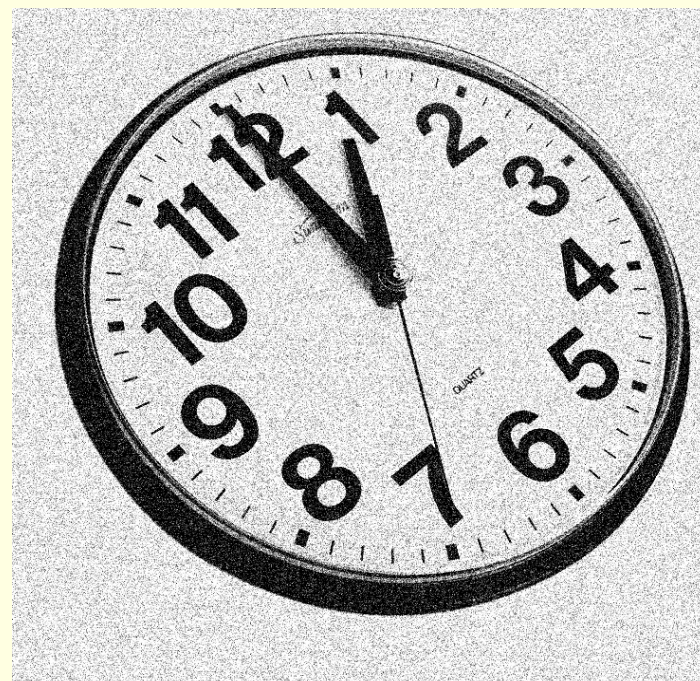


оригинално изображение

Медианен филтър

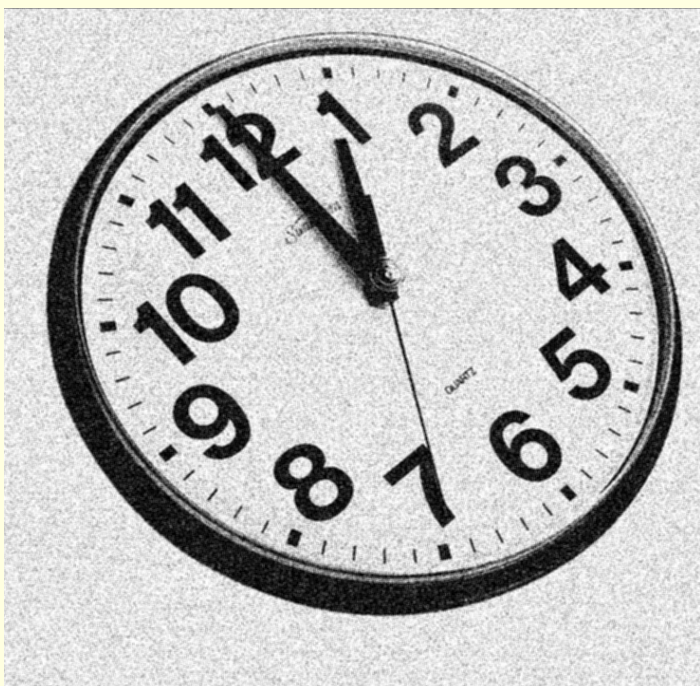


изображение с шум

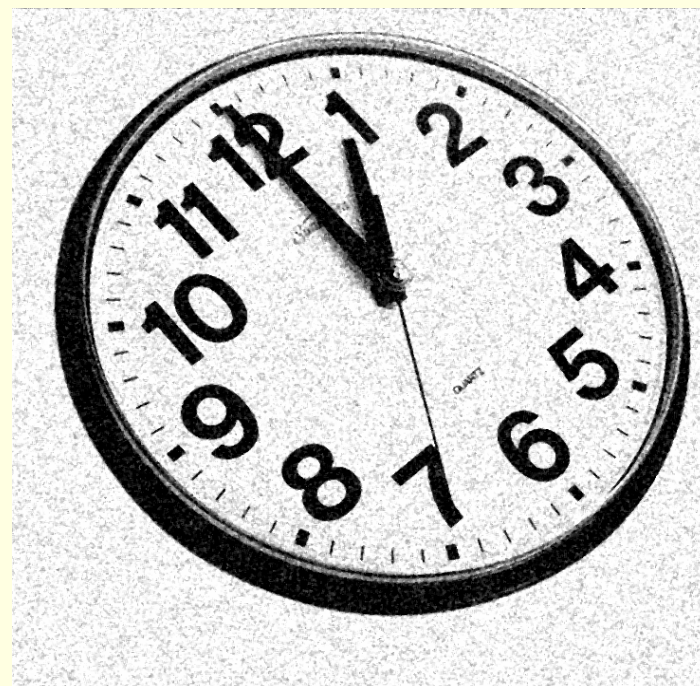


изображение с шум

Медианен филтър

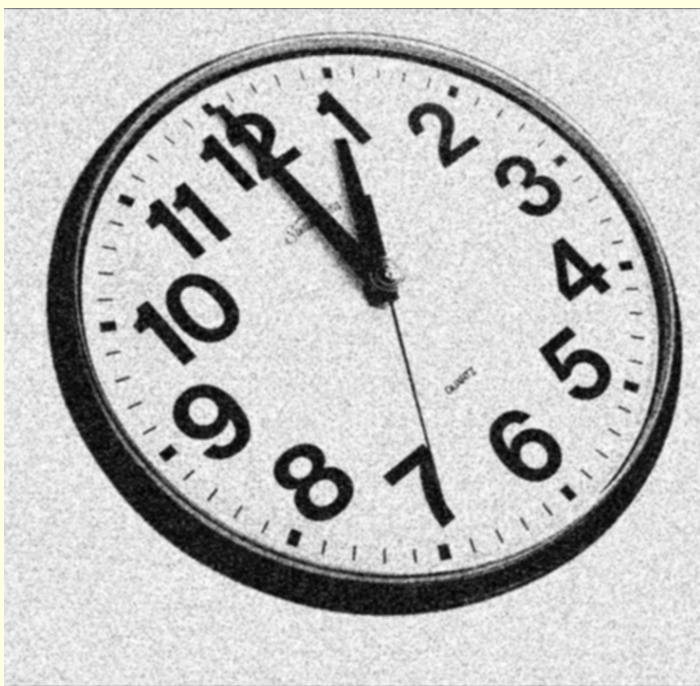


3x3 blur – 1 итерация

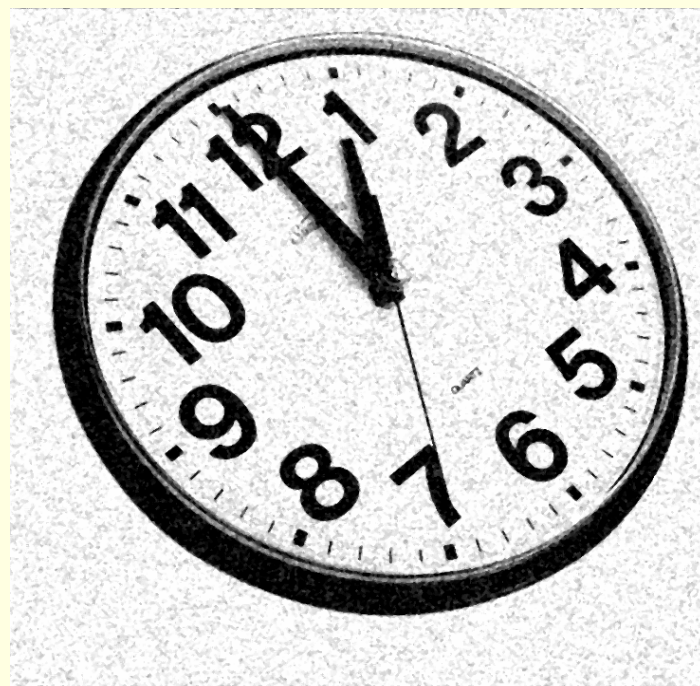


3x3 median – 1 итерация

Медианен филтър

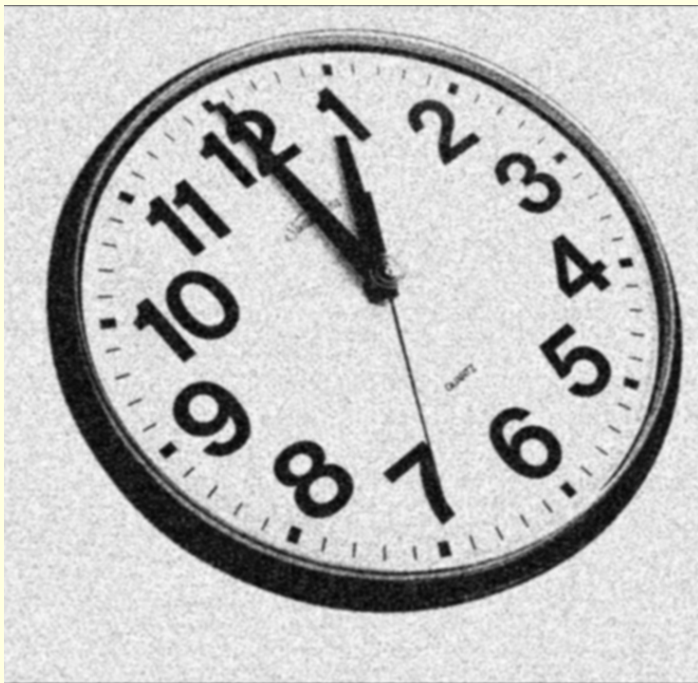


3x3 blur – 2 итерация

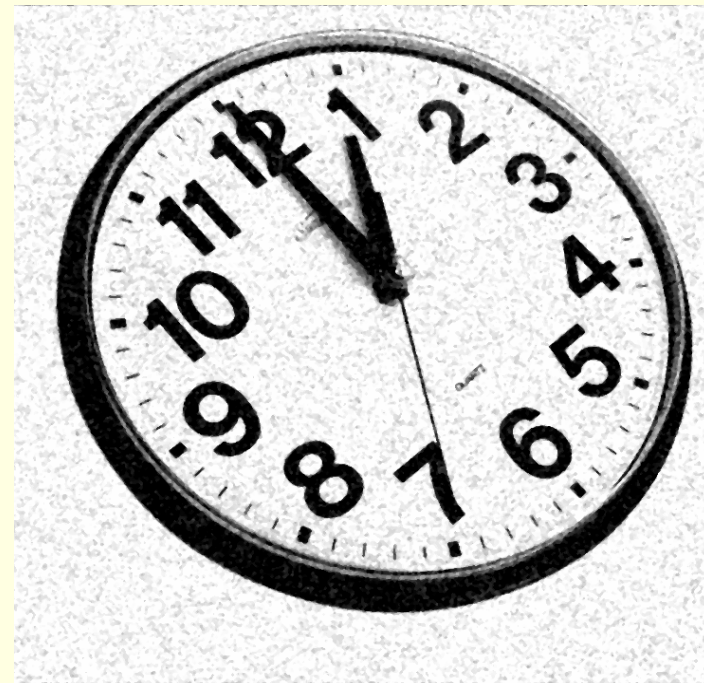


3x3 median – 2 итерация

Медианен филтър

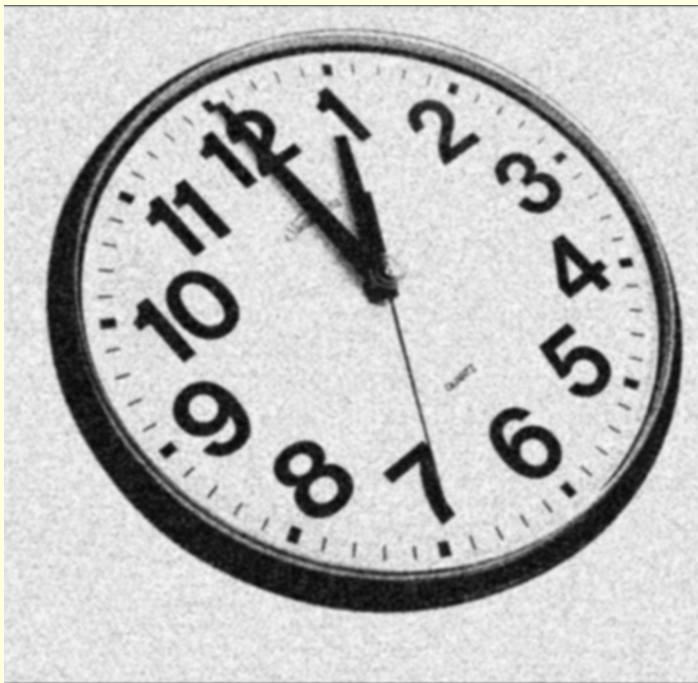


3x3 blur – 3 итерация

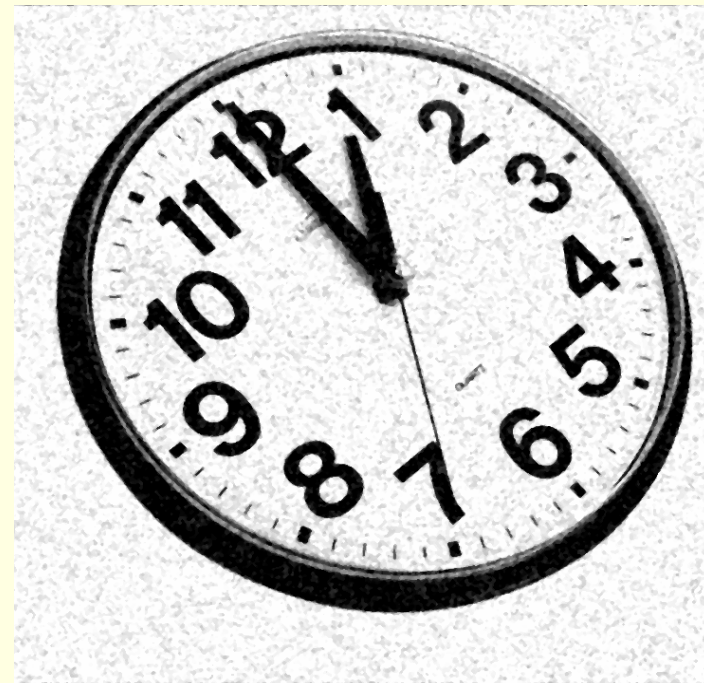


3x3 median – 3 итерация

Медианен филтър

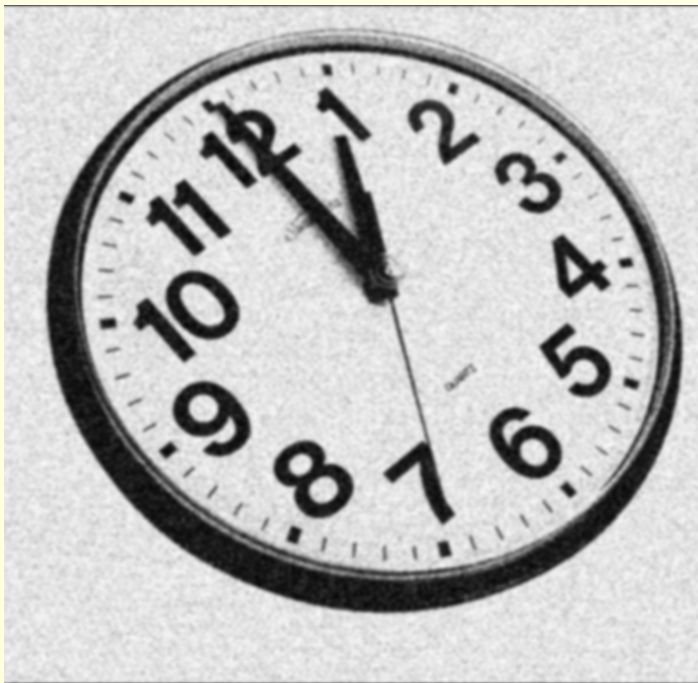


3x3 blur – 4 итерация

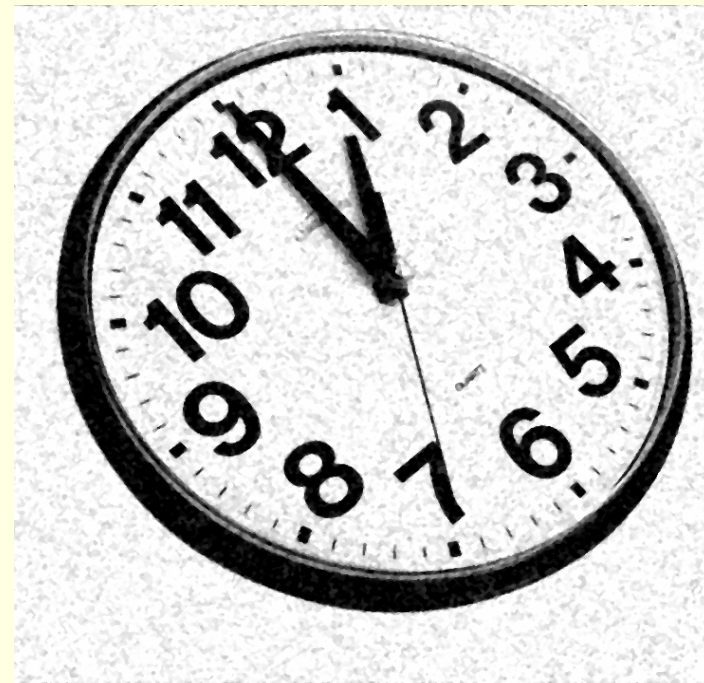


3x3 median – 4 итерация

Медианен филтър



3x3 blur – 5 итерация

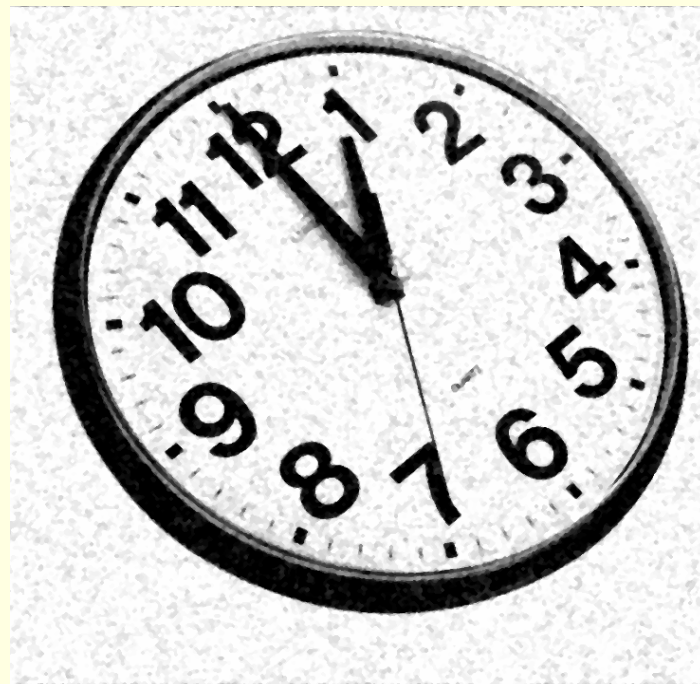


3x3 median – 5 итерация

Медианен филтър

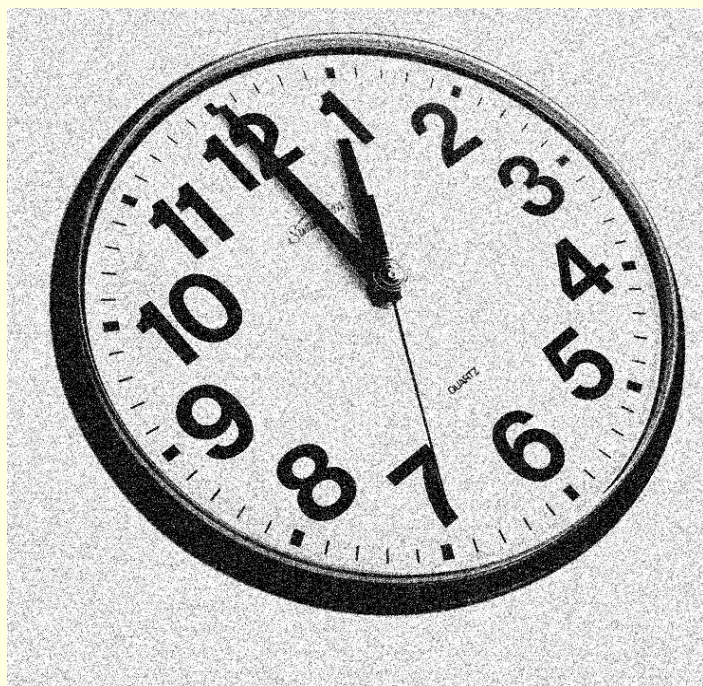


3x3 blur – 10 итерация

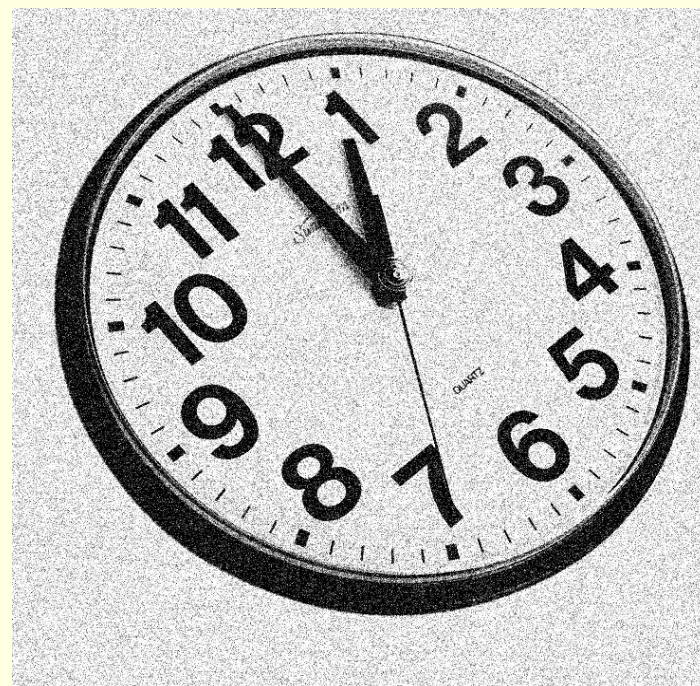


3x3 median – 10 итерация

Медианен филтър



изображение с шум

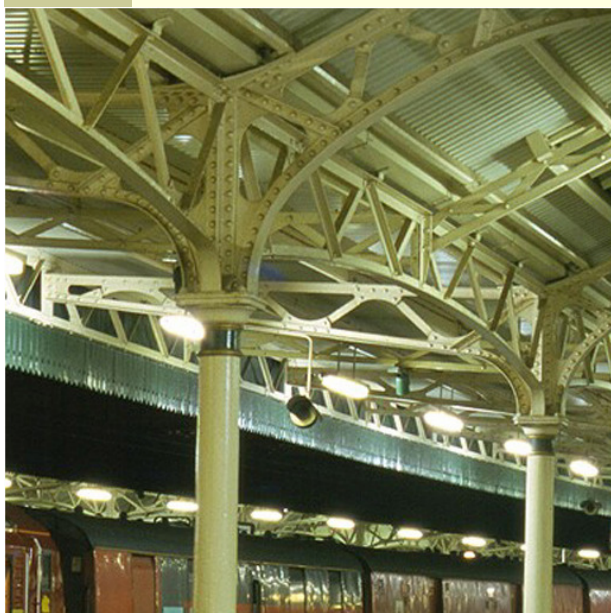


изображение с шум

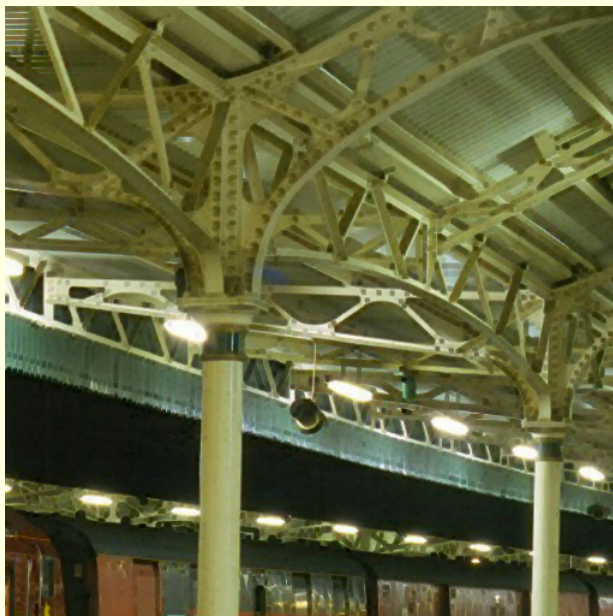
MAX и MIN филтър

- За всеки пиксел се определя **минималната** или **максималната** стойност в локална околност

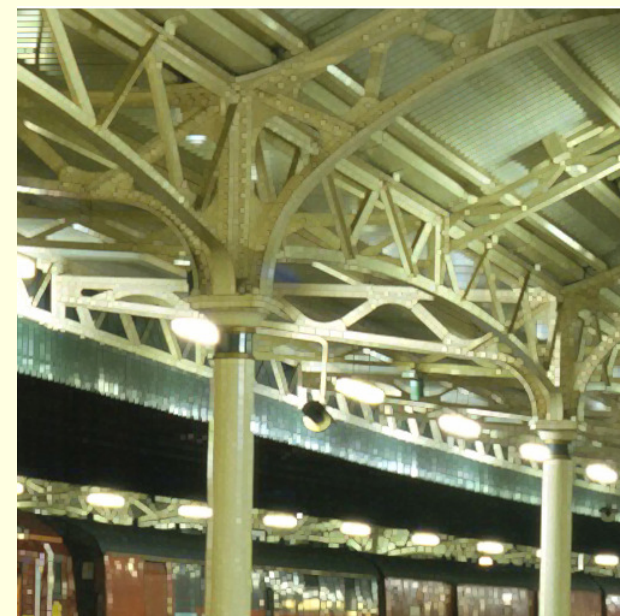
$$f_{\min}(i, j) = \text{MIN}[f(i+k, j+l)] \quad f_{\max}(i, j) = \text{MAX}[f(i+k, j+l)]$$



оригинално изображение



min филтър



max филтър

Мажоритарен филтър

- За всеки пиксел се определя **модата**
 - **най-често срещаната стойност** в локалната околност



КРАЙ

Следваща тема:

*Дискретни трансформации в
честотната област*