



# Filtros e Morfologia

Prof. Dr. Geraldo Braz Junior

# Filtragem

- Operações que visam extrair informações importantes da imagem
  - Características
    - Cantos, bordas, agrupamentos
  - Melhoramento
    - retirada de ruídos, aumento de contraste, ...
- O processo de filtragem forma uma nova imagem a qual é uma combinação dos pixels da imagem original

# Classificação

- Domínio ou espaço em que atuam: da frequência ou espacial
- Tipo de frequência:
  - passa ou elimina baixas frequências
  - passa ou elimina altas frequências
  - passa ou elimina faixas de frequências
- Linearidade: lineares e inversíveis ou não lineares
- Tipos de aplicação:
  - suavização, contraste, globais, adaptativos, janelados....

# Filtros Lineares

$$f(x, y) \rightarrow \boxed{\mathcal{S}} \rightarrow g(x, y)$$

- $\mathcal{S}$  é um filtro linear caso satisfaça a seguinte condição:

$$\mathcal{S}[\alpha f_1 + \beta f_2] = \alpha \mathcal{S}[f_1] + \beta \mathcal{S}[f_2]$$

# Exemplo: Filtro da Média

- Um filtro da média pode ser definido como uma janela 3x3 que se move através de uma vizinhança de pixels, aplicando a seguinte fórmula

$$(f * h)[m, n] = \frac{1}{9} \sum_{k, l} f[k, l] h[m - k, n - l]$$

- \* representa a convolução de duas funções: f e h

Onde

- k varia de n-1 a n+1, l varia de m-1 a m+1
- m e n são as dimensões da imagem f

	h		
1/9	1	1	1
	1	1	1
	1	1	1

# Processo de Filtragem por Média

$F[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$


$$(f * h)[m, n] = \sum_{k, l} f[k, l] h[m - k, n - l]$$

# Processo de Filtragem por Média

$F[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$

	0	10							

$$(f * h)[m, n] = \sum_{k, l} f[k, l] h[m - k, n - l]$$

# Processo de Filtragem por Média

$F[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$

	0	10	20	30	30				

$$(f * h)[m, n] = \sum_{k, l} f[k, l] h[m - k, n - l]$$



# Processo de Filtragem por Média

$F[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

# Convolução Discreta

- Símbolo \*
- Processo basicamente resumido nas etapas:
  - Posicione um filtro  $h[n,m]$  num pixel, sendo que o pixel fica na posição central
  - Multiplique as filtro \* elementos da imagem:  $h[n,m] * f[k,l]$
  - Some todos os elementos da multiplicação
  - Repita o processo para todos os pontos da imagem

# o que aconteceu?

- Gerou uma nova imagem  $g(x,y)$  onde cada pixel é representado pela média do pixel  $(x,y)$  e seus vizinhos diretos
- Objetivo
  - Remover características destacadas
  - Smooth

$$\frac{1}{9} \begin{matrix} & & h[\cdot, \cdot] \\ \begin{matrix} 1 \\ 1 \\ 1 \end{matrix} & \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \end{matrix}$$

# Exemplo: Filtro da Média



# Exemplos: o que acontece?



Original

•0	•0	•0
•0	•1	•0
•0	•0	•0

=

?

# nada



Original

•0	•0	•0
•0	•1	•0
•0	•0	•0

=



Filtered  
(no change)

# Exemplos: o que acontece?



Original

•0	•0	•0
•0	•0	•1
•0	•0	•0

=

?

# Deslocamento 1 pixel para direita



Original

•0	•0	•0
•0	•0	•1
•0	•0	•0

=



Shifted right  
By 1 pixel

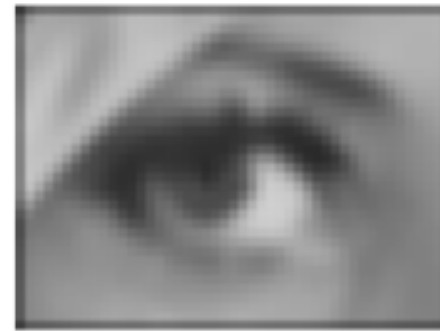


# Exemplo: suavização



Original

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

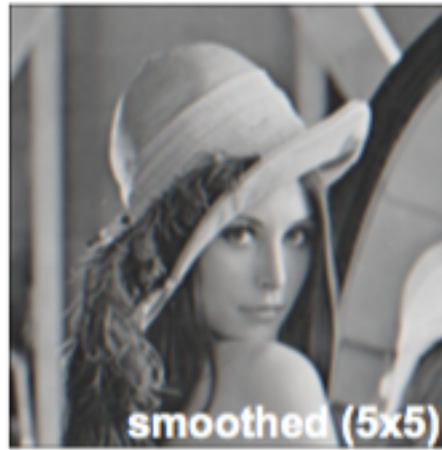


Blur (with a  
box filter)

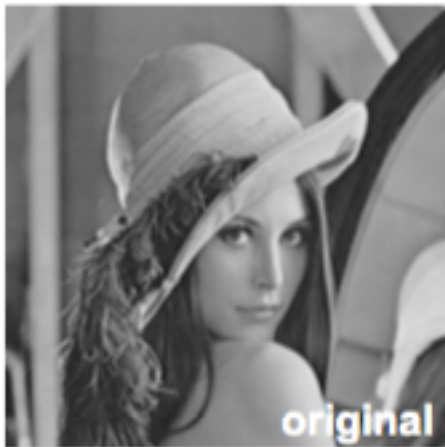
# Exemplo: combinação de filtros?



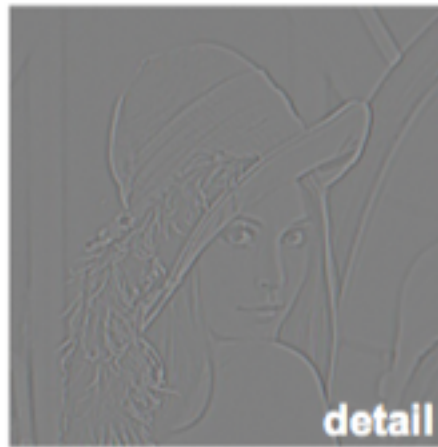
-



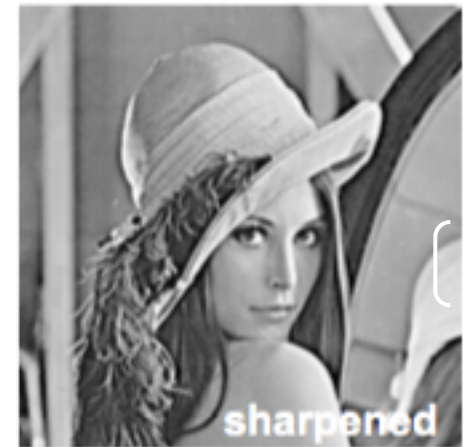
=



+ a



=



# Exemplo: Filtro de Realce

- Aumenta diferença em relação a média local
- Realça alguns contornos



Original

•0	•0	•0
•0	•2	•0
•0	•0	•0

−

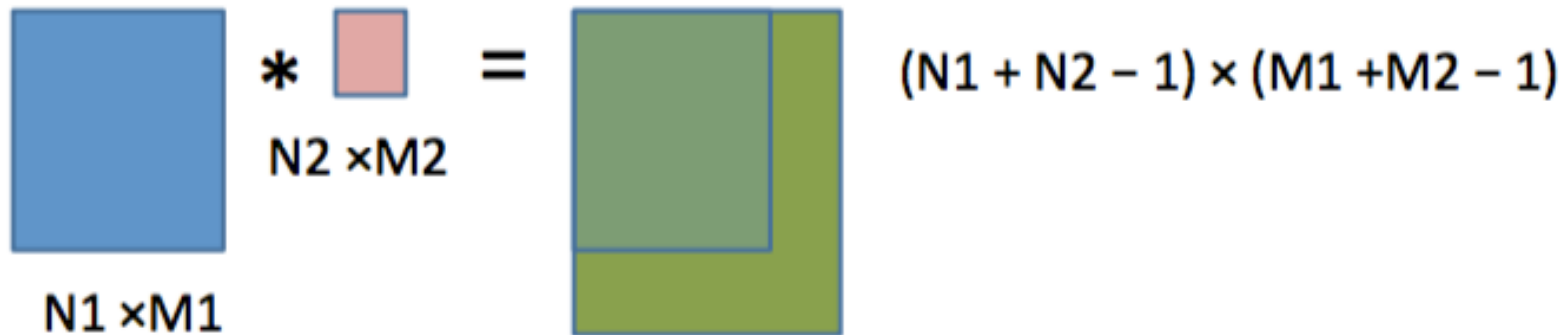
$\frac{1}{9}$

•1	•1	•1
•1	•1	•1
•1	•1	•1

=

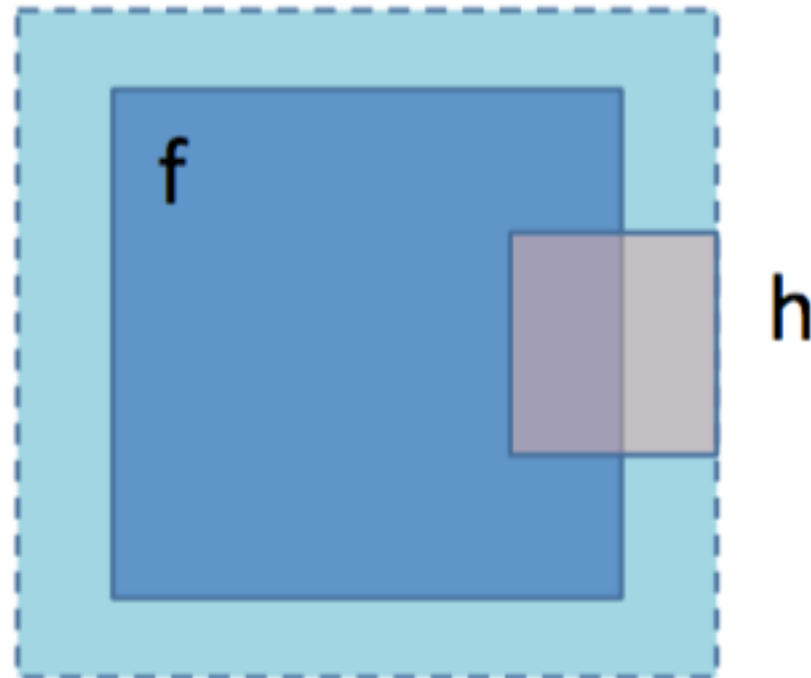


# O que fazer com nas bordas da imagem?



# O que fazer com nas bordas da imagem?

- Somente será computado pixels que tenham representantes completos dentro do filtro
- Estratégias que podem ser adotadas
  - Replicar borda
  - Adicionar zeros
  - Extensão por espelho



# Filtros Passa Baixa

- Principais objetivos:
- Suavizar a imagem pela redução das variações nos de níveis de cinza que dão à aparência de “serrilhado” nos patamares de intensidade.
- Atenuar **altas frequências**
- Minimizar ruídos

# Filtro da Média

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

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

$$Z = \frac{1}{49} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

# Filtros: média

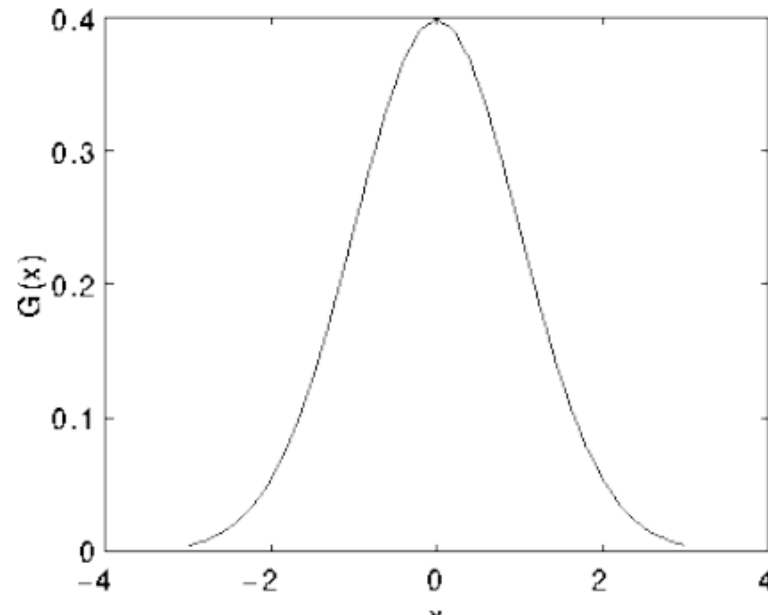
- `blur( original, resultado, Size( 3, 3 ))`
- - o último parâmetro define o tamanho do filtro





# Filtro Gaussiano

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



$$Z = \frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

# Filtros: gaussiano

- GaussianBlur( original, resultado, Size( 3, 3 ),1,1);
  - dois últimos: desvio padrão em x e desvio padrão em y
- Caso queira o filtro em si, use a função: getGaussianKernel



# Filtros não lineares

- Não preservam relação inversa
- Principais:
  - Bilateral
  - Mediana

# Filtro Bilateral

- Redução de ruídos
- Simplificação de estruturas
- Preserva contornos



# Filtros: bilateral



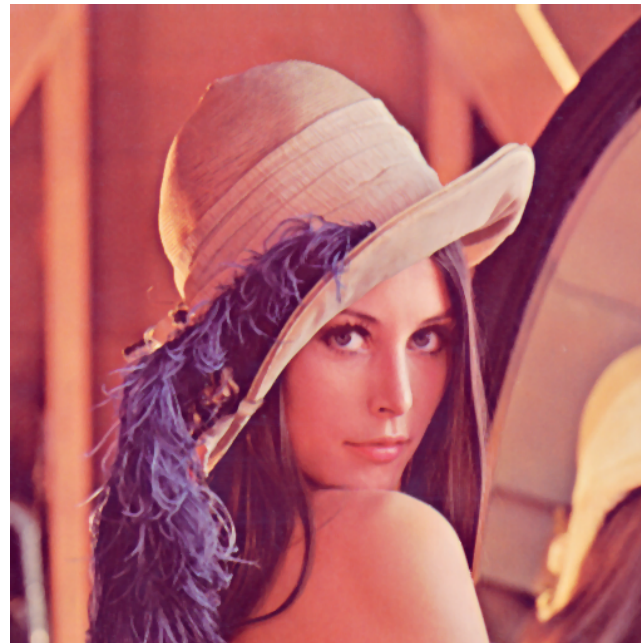
- `bilateralFilter(original, resultado, 15, 15*2, 15/2)`
  - 9 = tamanho do kernel

# Filtro da Mediana

- Melhor para reduzir o ruído
- Ordena-se os vizinhos em ordem crescente.
  - A mediana consiste nos  $m$  elementos que estão no meio do conjunto
- Eficiente contra ruídos localizados

# Filtro: mediana

- `salt(original, 3000);`
- `medianBlur(original, resultado, 3);`
- \*Procure implementação de sal nos exemplos



# Filtro genérico no opencv

- Use a função

```
filter2d(imagem, saida, depth, kernel)
```



# Morfologia

# Morfologia Matemática

- Não Linear
- Objetivo: extrair objetos baseados em alguma geometria pré-definida
  - Elemento Estruturante
- Normalmente sobre imagens binárias
  - representadas com apenas 2 intensidades

# Exemplo de Imagem Binária



Converte em grayscale (`cvtColor(..., CV_BGR2GRAY)`)

**Corte:**  $60 < x < 255$  (`threshold(image, thresholded, 60, 255, cv::THRESH_BINARY)`)

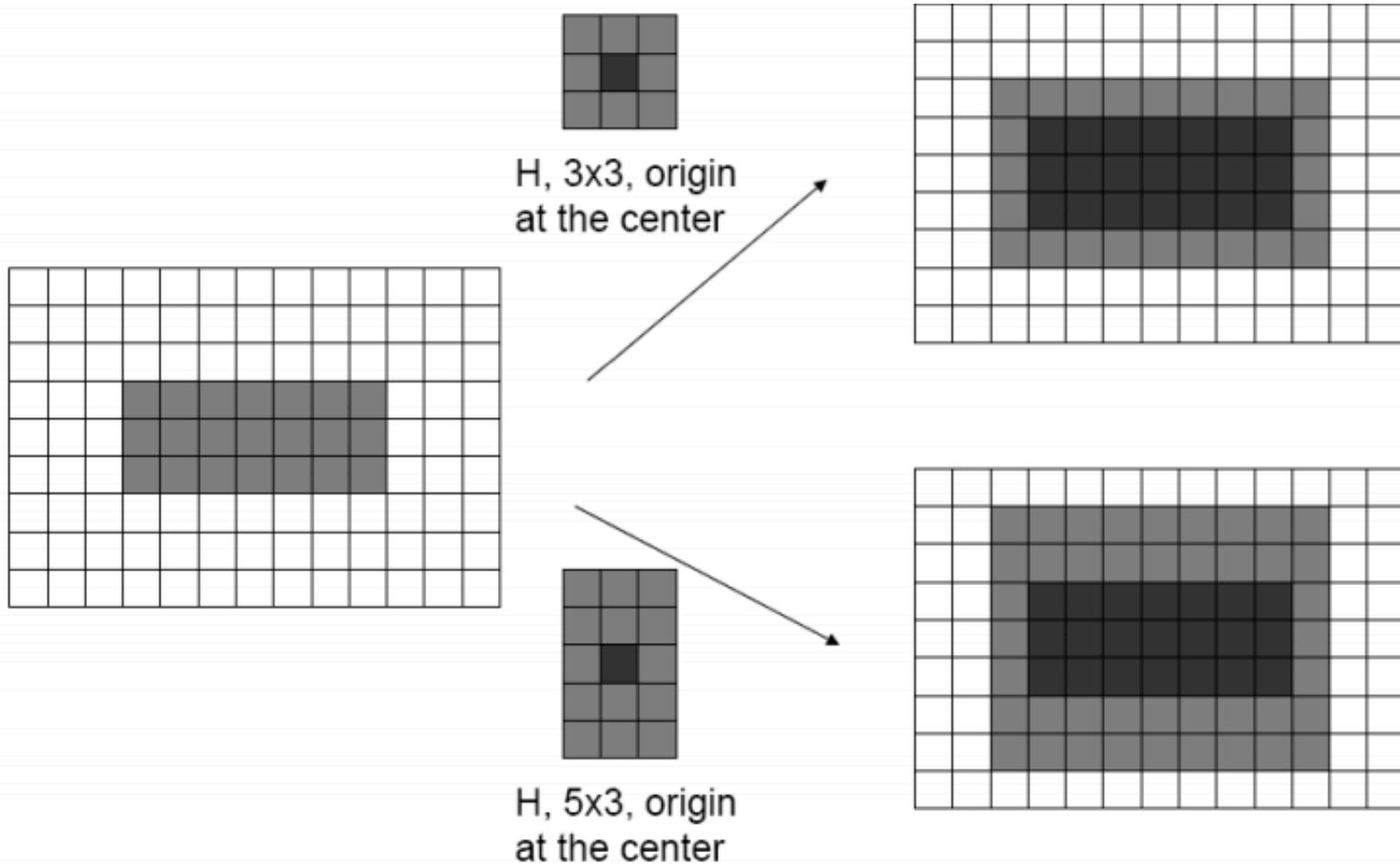
# Operações Básicas

- Dilatação
  - Preenche ou amplifica contornos

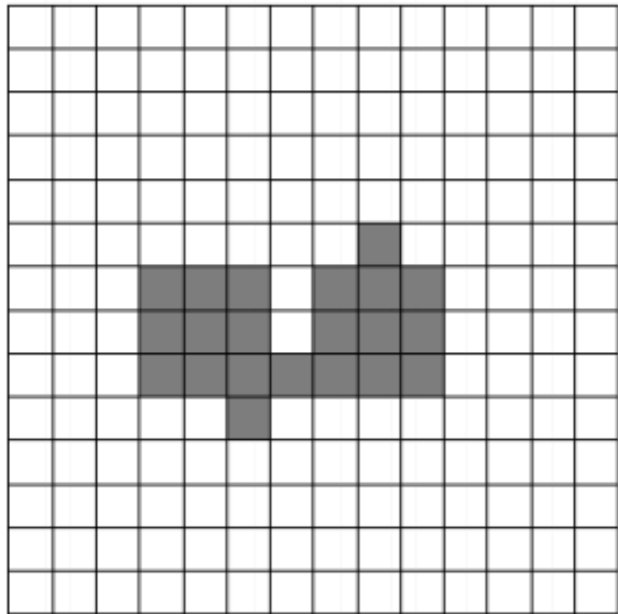
$$A \oplus B = \{z \mid (\hat{B})_x \cap A \neq \emptyset\}$$

- Reflexão (translação) de B para a origem centralizada em um elemento x de A
- B é chamado de **elemento estruturante**

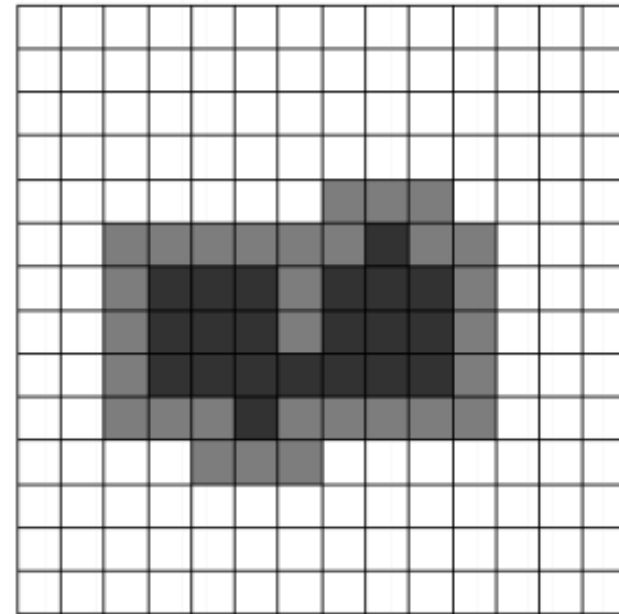
# Exemplos



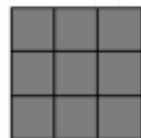
# Exemplos



F



G



H, 3x3, origin at the center

# Exemplos: line

The term watershed  
refers to a ridge that ...

... divides areas  
drained by different  
river systems.

The term watershed  
refers to a ridge that ...



# Exemplo: Rolling Ball





# Erode no Opencv



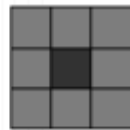
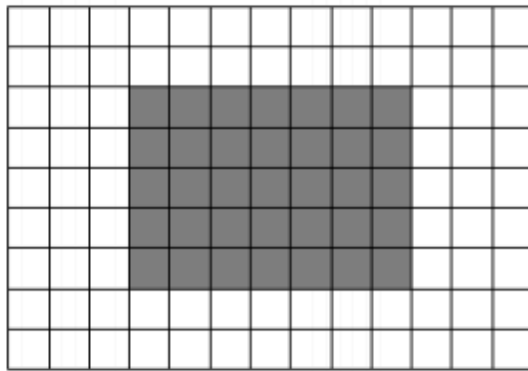
# Operações Básicas

- Erosão
  - Reduz elementos ao elemento estruturante

$$A \ominus B = \{x \mid (B)_x \subseteq A\}$$

- Reflexão (translação) de B para a origem centralizada em um elemento x de A tal que todos os elementos de B estejam em A
- B é chamado de **elemento estruturante**

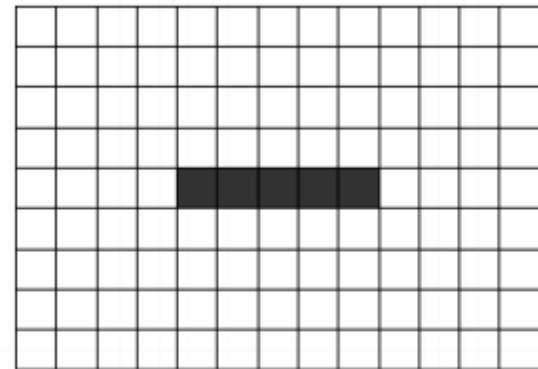
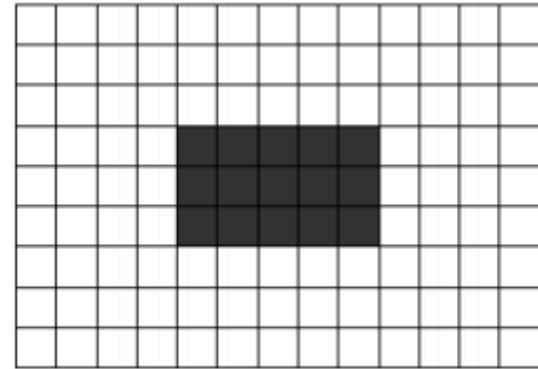
# Exemplos



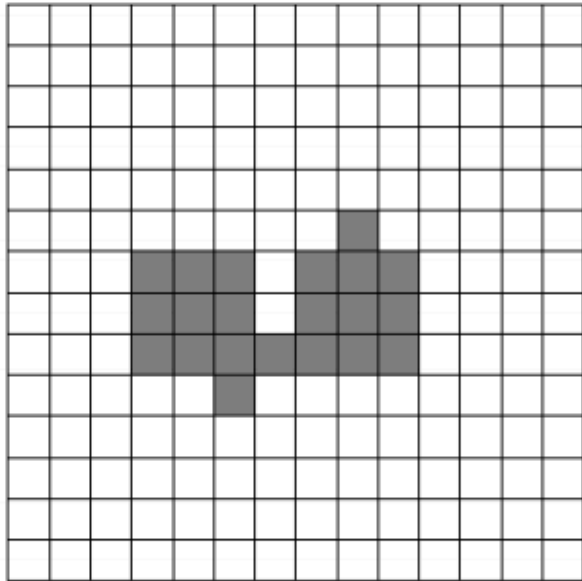
H, 3x3, origin  
at the center



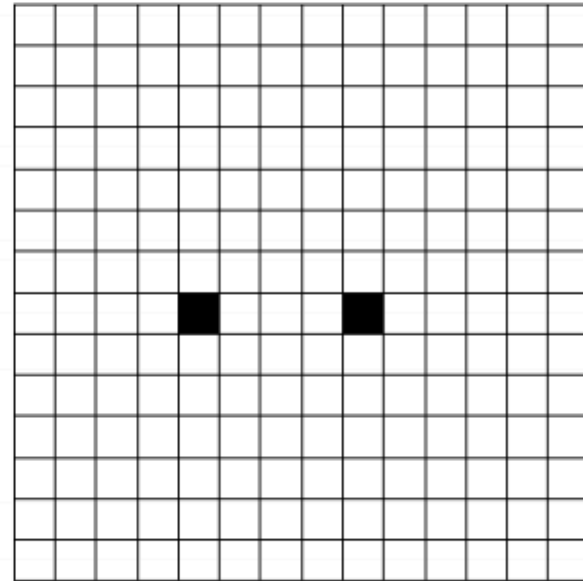
H, 5x3, origin  
at the center



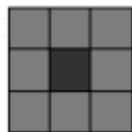
# Exemplos



F

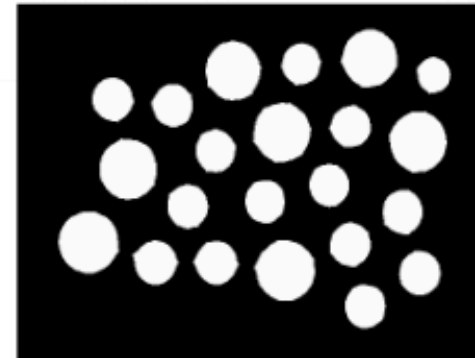
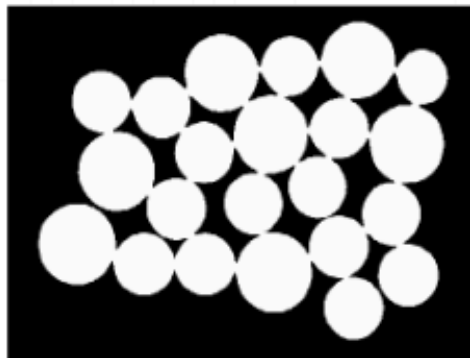


G



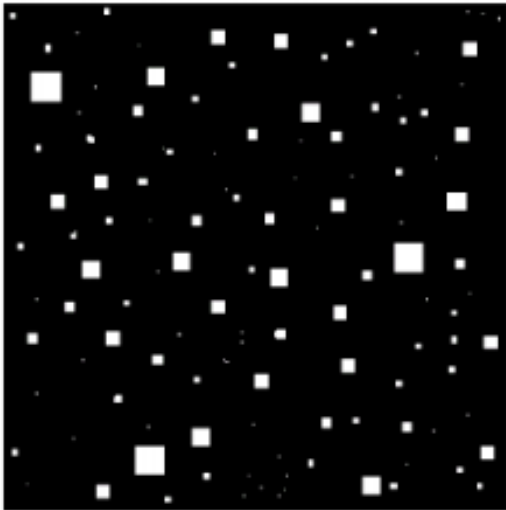
H, 3x3, origin at the center

# Exemplos: separação

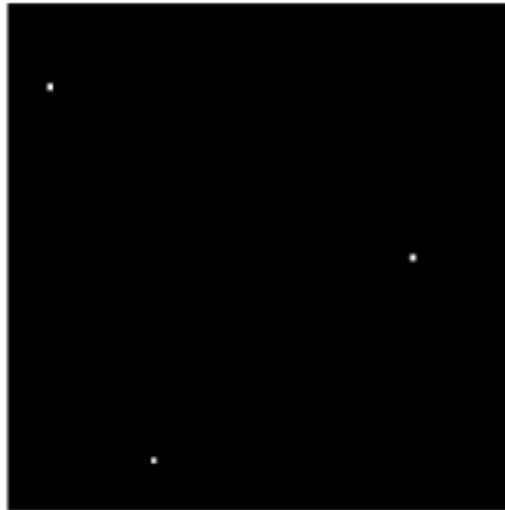


# Exemplos: remoção de componentes

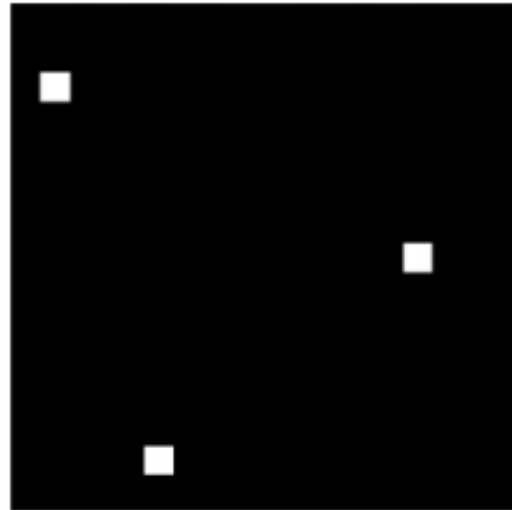
1,3,5,7,9 e 15



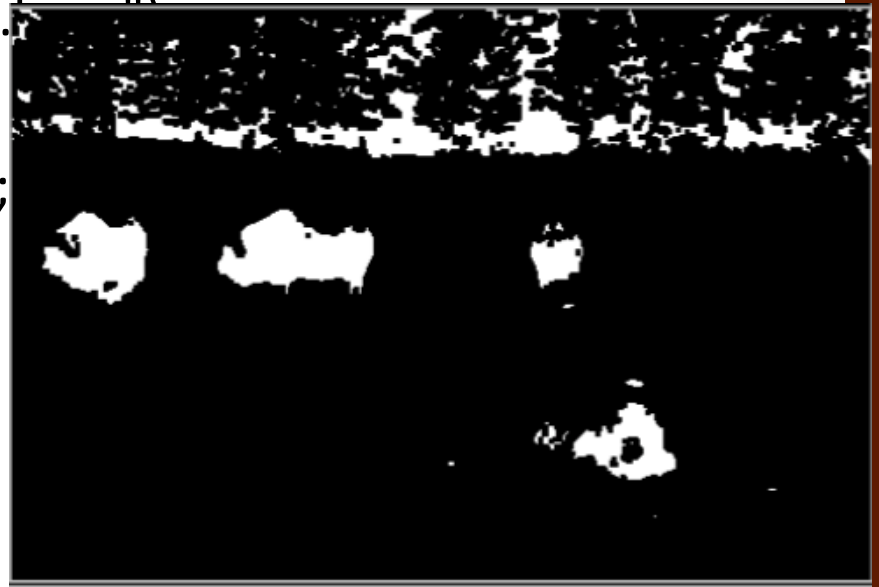
Erosão com 13



Dilatação com 13



# Erode no Opencv



# Obtendo Elementos Estruturantes Diferentes

- Quando se passa uma matriz vazia (Mat()), opencv assume um quadrado 3x3 como elemento estruturante.

Opções para isso:

Mat getStructuringElement(int `shape`, Size ksize)

- `MORPH_RECT` - retangular
- `MORPH_ELLIPSE` - elipse
- `MORPH_CROSS` – em cruz
- `CV_SHAPE_CUSTOM` – especificado pelo usuário (use \* para indicar presença)
- Ou simplesmente criar uma matriz
  - `Mat element(7,7,CV_8U,cv::Scalar(1));`



# Operações mais complexas

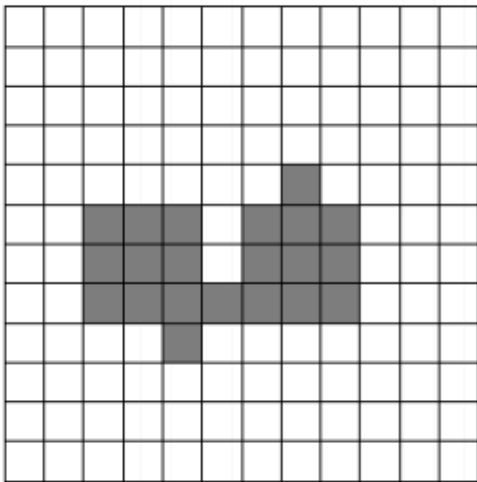
- Utilizam combinações de dilatação e erosão
- Ou
- Expansões na teoria dos conjuntos
- Exemplos:
  - Abertura
  - Fechamento
  - Gradiente
  - TopHat
  - Hit & Miss
  - Fill
  - Skeleton

# Abertura

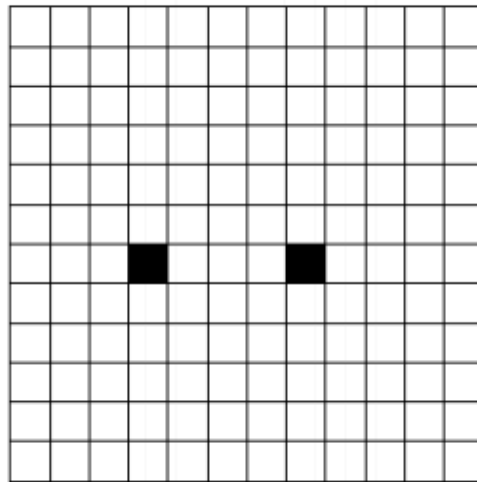
- Suaviza o contorno da imagem ao mesmo tempo que elimina pequenas brechas
- Uma erosão seguida de uma dilatação

$$A \circ B = (A \ominus B) \oplus B$$

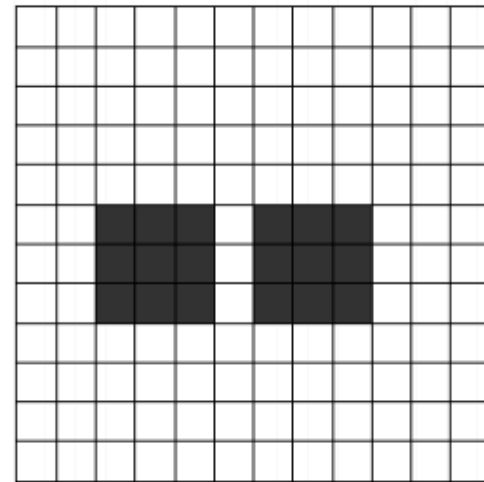
# Exemplo



F



$F \ominus H$



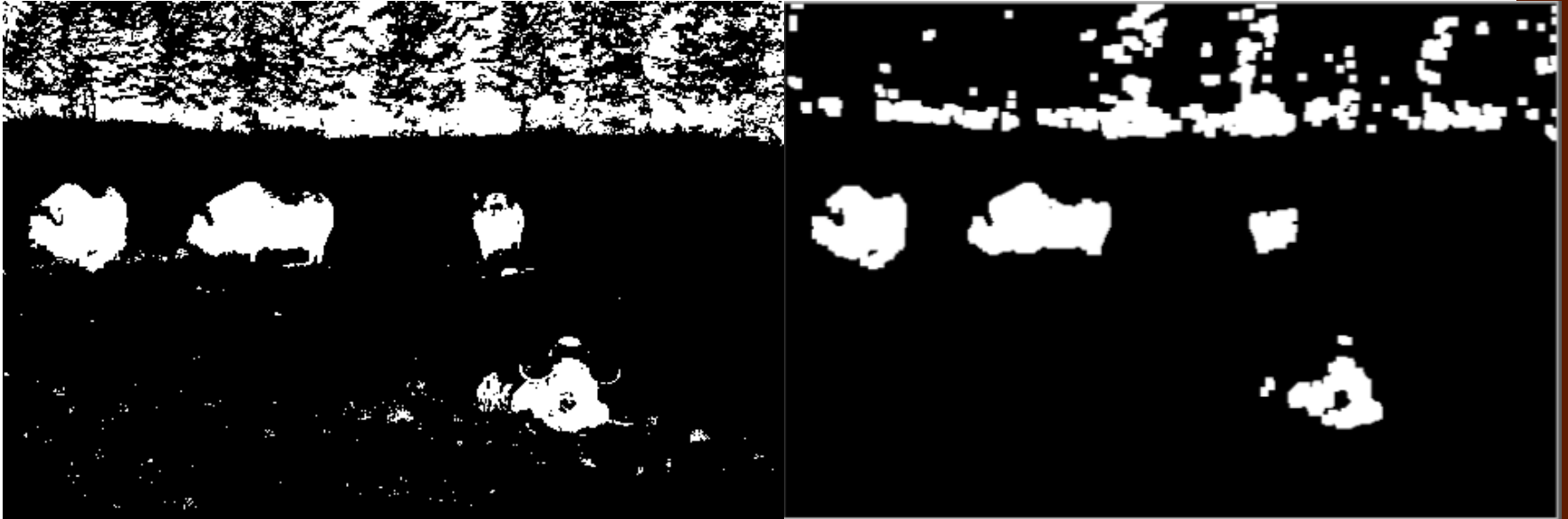
$(F \ominus H) \oplus H$



H, 3x3, origin at the center

# Abertura no Opencv

- Através da função: **morphologyEx**



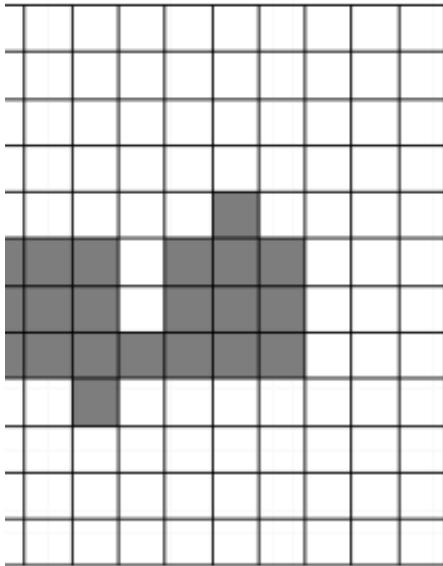
- `Mat element5(5,5,CV_8U,cv::Scalar(1));`
- `Mat opened;`
- `morphologyEx(image,opened,cv::MORPH_OPEN,element5);`

# Fechamento

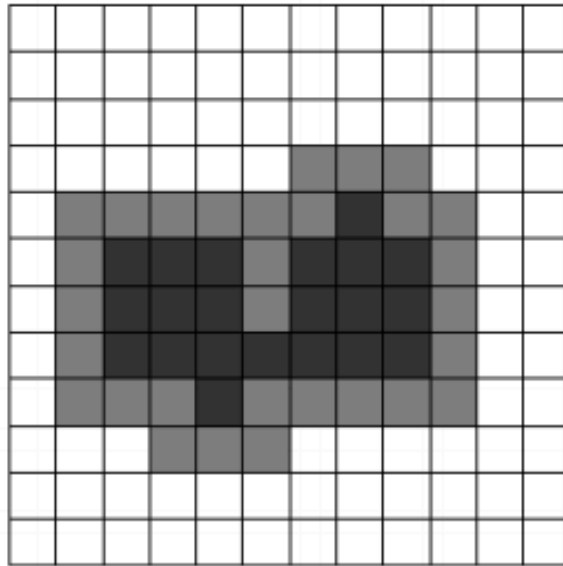
- Suaviza contorno, elimina pequenos buracos, preenche fendas
- Uma dilatação seguida por uma erosão

$$A \bullet B = (A \oplus B) \ominus B$$

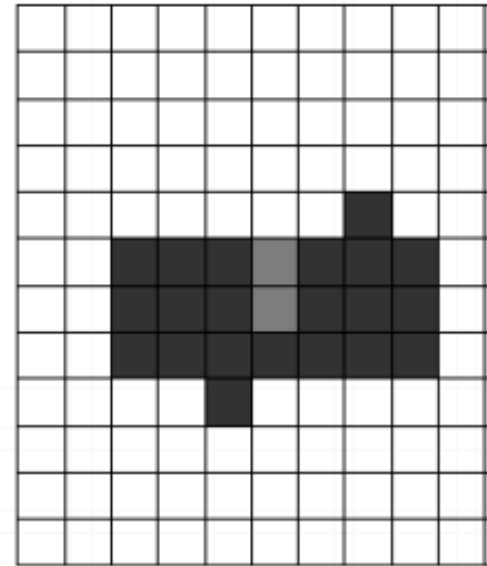
# Exemplo



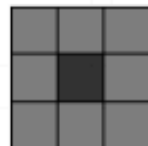
F



$F \oplus H$

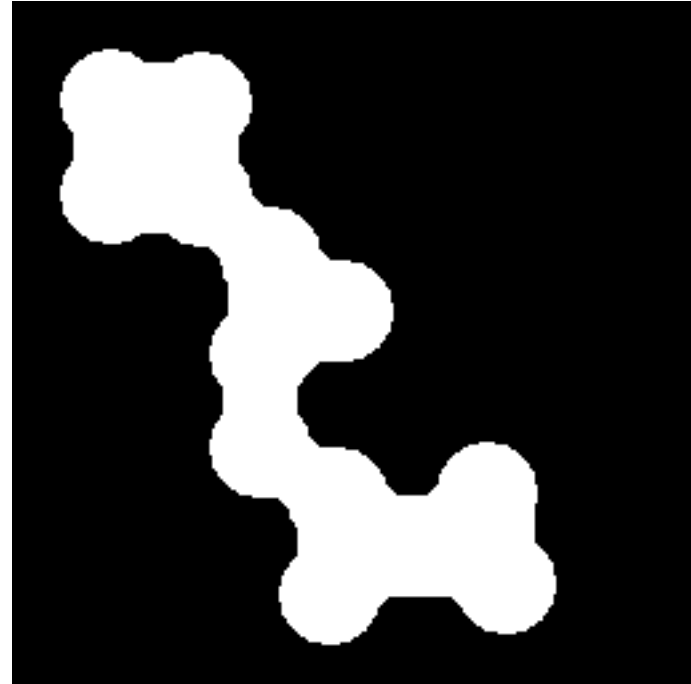
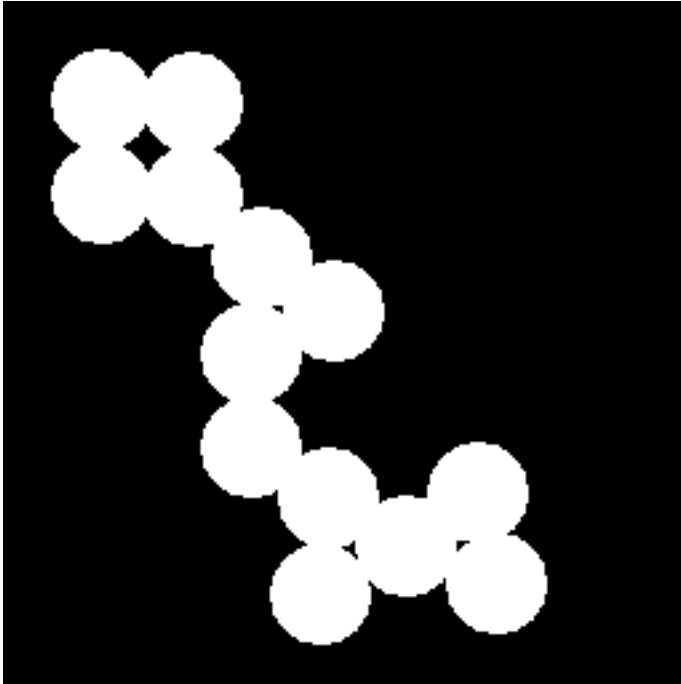


$(F \oplus H) \oplus H$

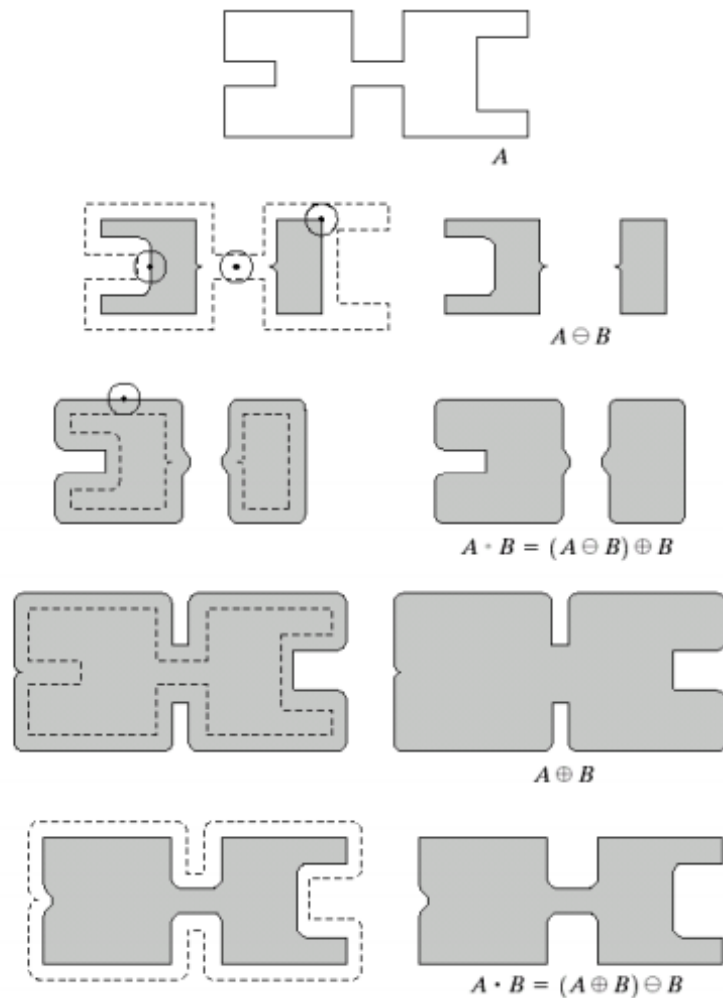


H, 3x3, origin at the center

# Exemplo



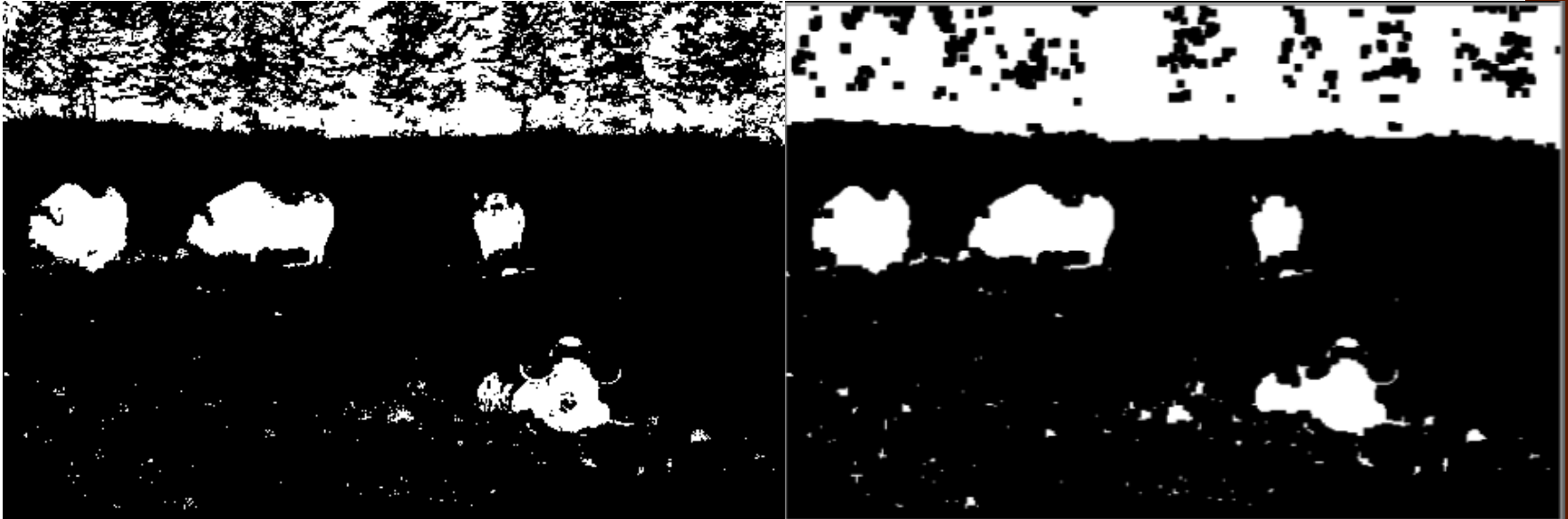
# Abertura e Fechamento





# Fechamento no Opencv

- Através da função: **morphologyEx**



- `Mat element5(5,5,CV_8U,cv::Scalar(1));`
- `Mat closed;`
- `morphologyEx(image,closed,cv::MORPH_CLOSE,element5);`

# Outras opções morphologyEx

- MORPH\_GRADIENT – gradiente morfológico
  - Para obter contornos
  - original – erosao(original, elemento)
- MORPH\_TOPHAT - “top hat”
  - Diferença entre original e abertura
  - original – open(original, elemento)
- MORPH\_BLACKHAT - “black hat”
  - Diferença entre original e fechamento
  - fechamento(original, elemento) - original
- MORPH\_HITMISS - “hit and miss”

# Uma que não é nativa no opencv

- Até então: skeletonização
- É uma representação da componente de sustentação de uma forma (esqueleto)

$$S_n(X) = (X \ominus B) - (X \ominus B) \circ B$$

- repete-se n vezes: ou até que não exista mais pixels na imagem original (caso binário)

# Exemplo

```
while (countNonZero(img) != 0) {  
    erode(img, eroded, element);  
    dilate(eroded, temp, element);  
    subtract(img, temp, temp);  
    bitwise_or(resultado, temp, resultado);  
    eroded.copyTo(img);  
}
```

