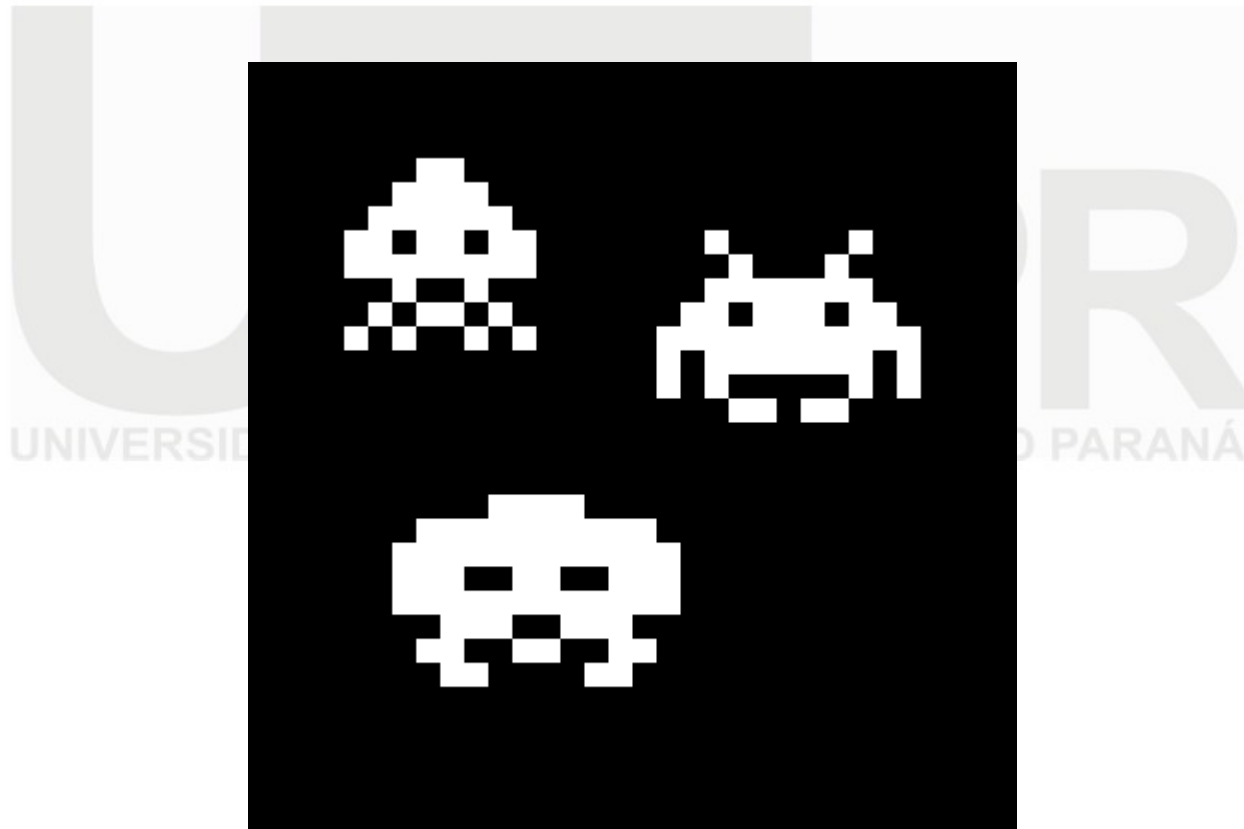


# Processamento Digital de Imagens

Prof. Bogdan Tomoyuki Nassu



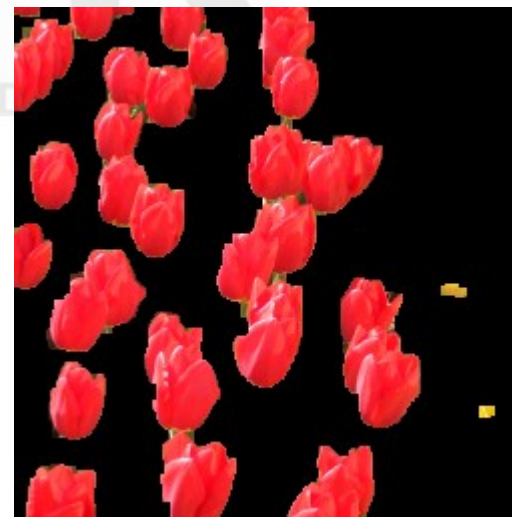
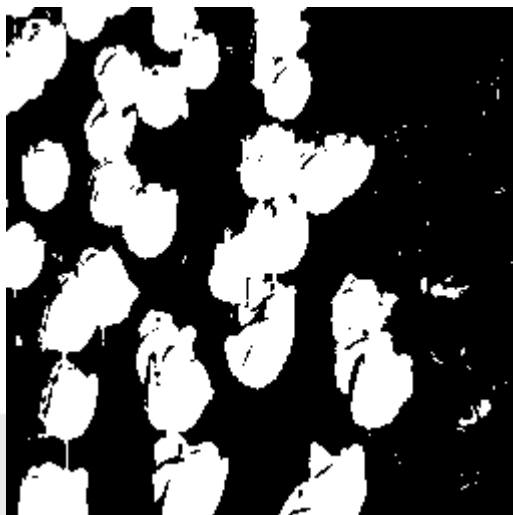
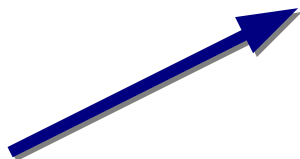
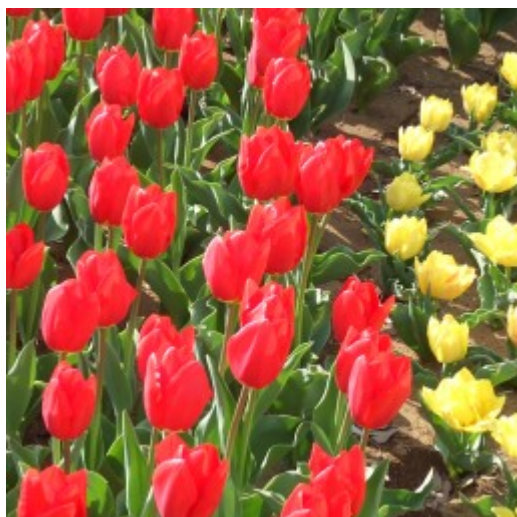
# Hoje

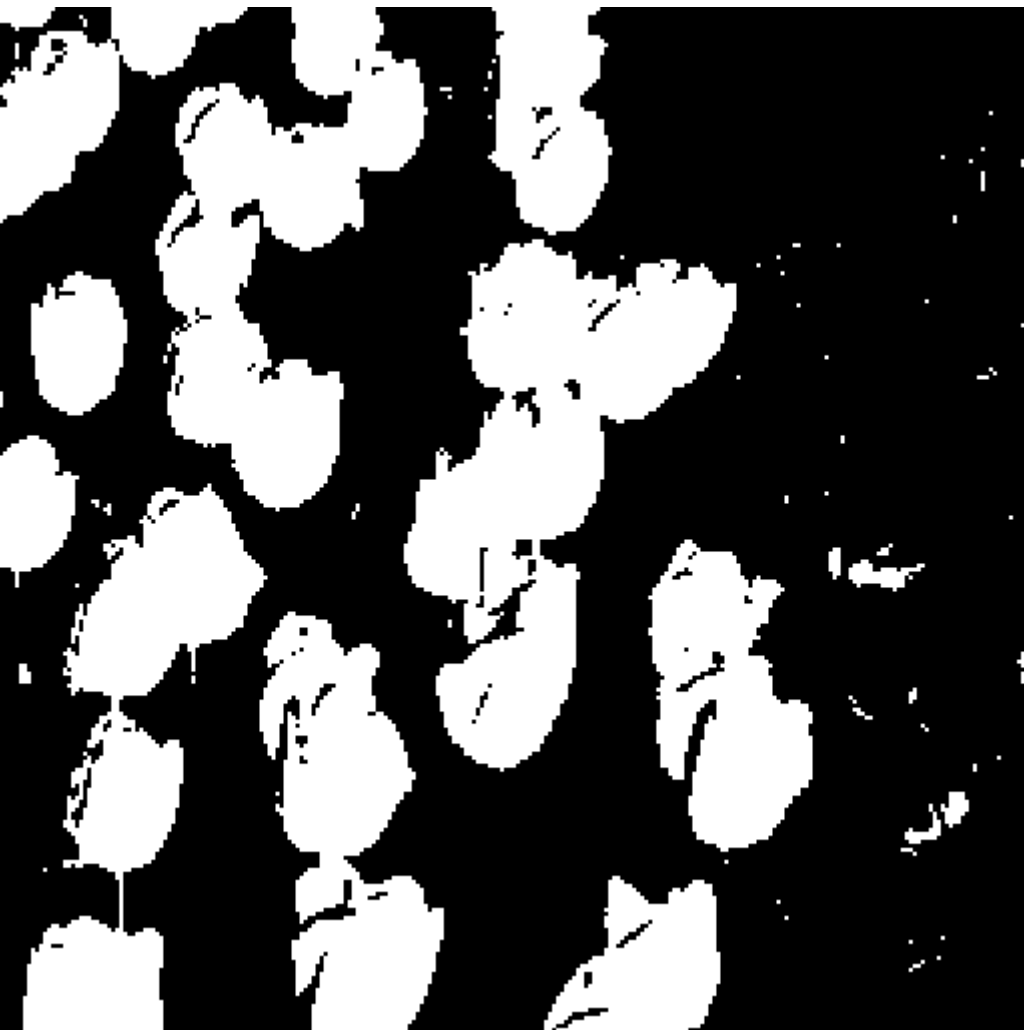
- Morfologia matemática.



# Morfologia? Matemática?

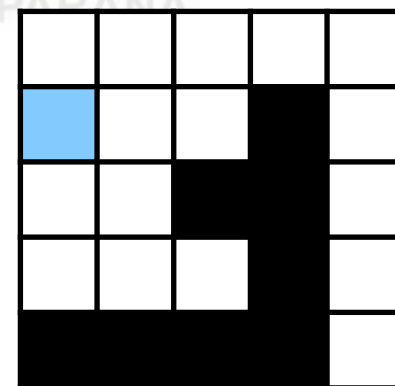
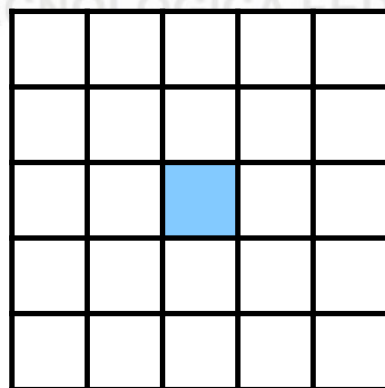
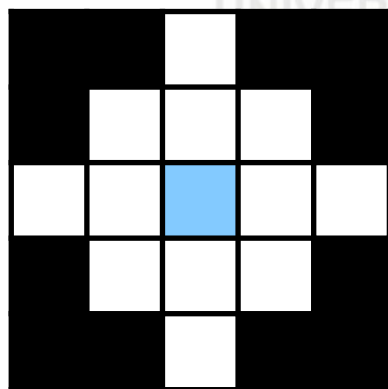
- Teoria e conjunto de técnicas para processamento de imagens com base, principalmente, na teoria dos conjuntos.
  - Conjuntos de pixels.
  - Uso principal: alterar o formato de regiões.
- Formular problemas usando morfologia matemática permite criar soluções baseadas em uma teoria e em técnicas já existentes.
  - Algoritmos podem derivar diretamente de formulações matemáticas.
- Na prática:
  - As técnicas mais fundamentais são bastante usadas.
  - Alguns problemas têm soluções baseadas em morfologia, mas existem algoritmos especializados mais eficientes ou eficazes.
- Nosso foco: parte mais fundamental...
  - Não entraremos em detalhes sobre definições e teoria matemática.





# Operadores morfológicos

- Operadores morfológicos básicos: filtros espaciais não-lineares.
- Todos os operadores usam um *elemento estruturante*.
  - Termo alternativo: *kernel*.
  - Indica quais são os pixels de interesse na janela.
  - É uma imagem binária, associada a um pixel de saída.



# Dilatação

$$I \oplus K$$

- A dilatação “engorda” as regiões de pixels brancos.

para cada vizinhança em torno de  $(x,y)$

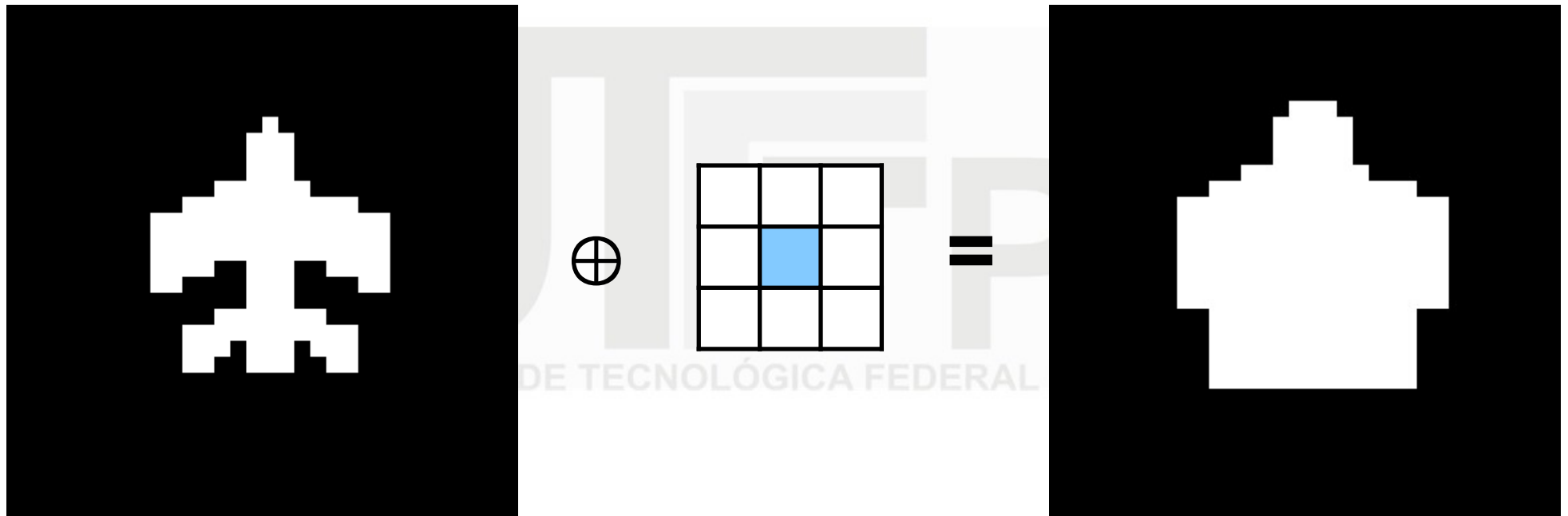
se pelo menos um dos pixels do kernel estiver setado na imagem

saída  $(x,y) \leftarrow 1;$

senão

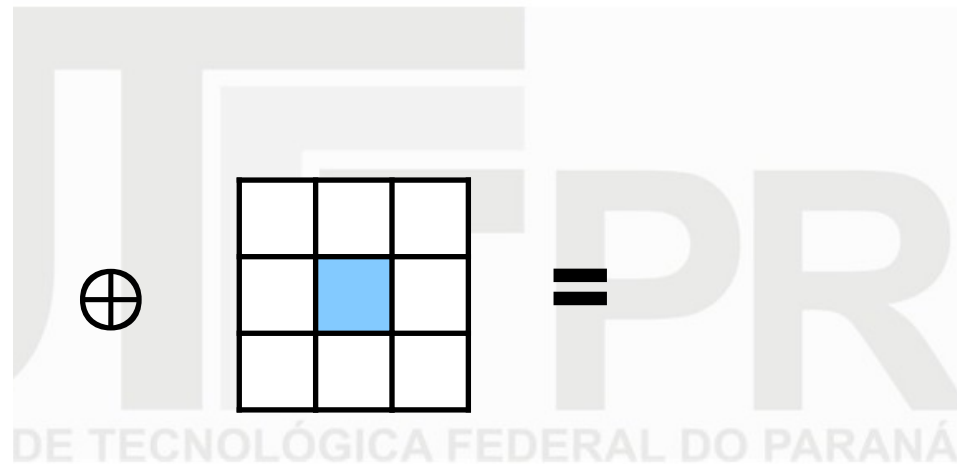
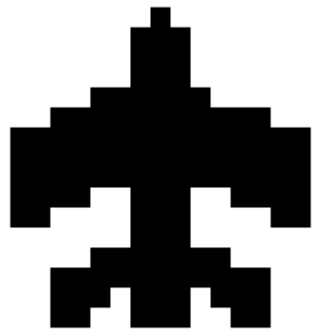
saída  $(x,y) \leftarrow 0;$

# Dilatação

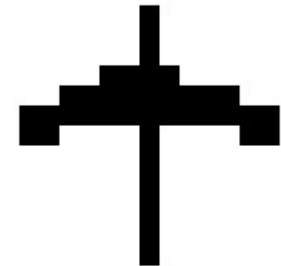
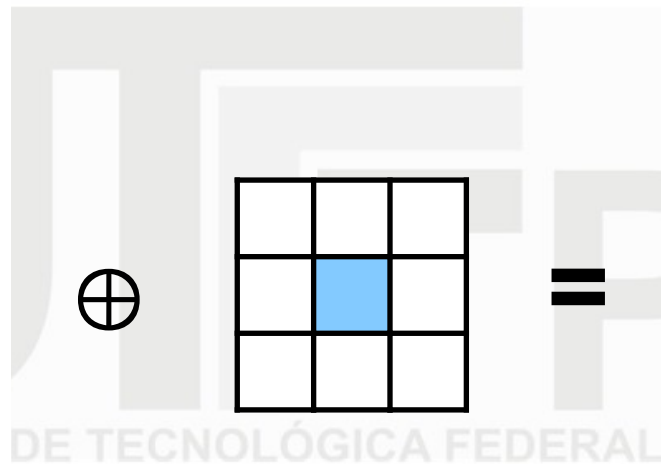
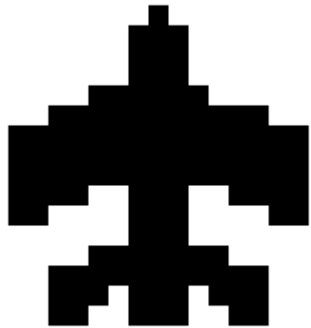




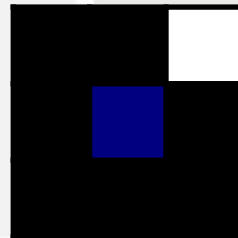
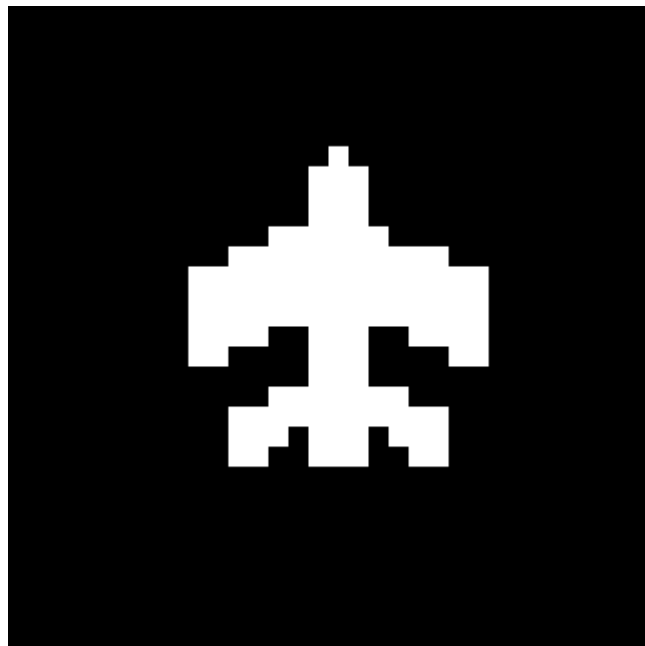
# Dilatação



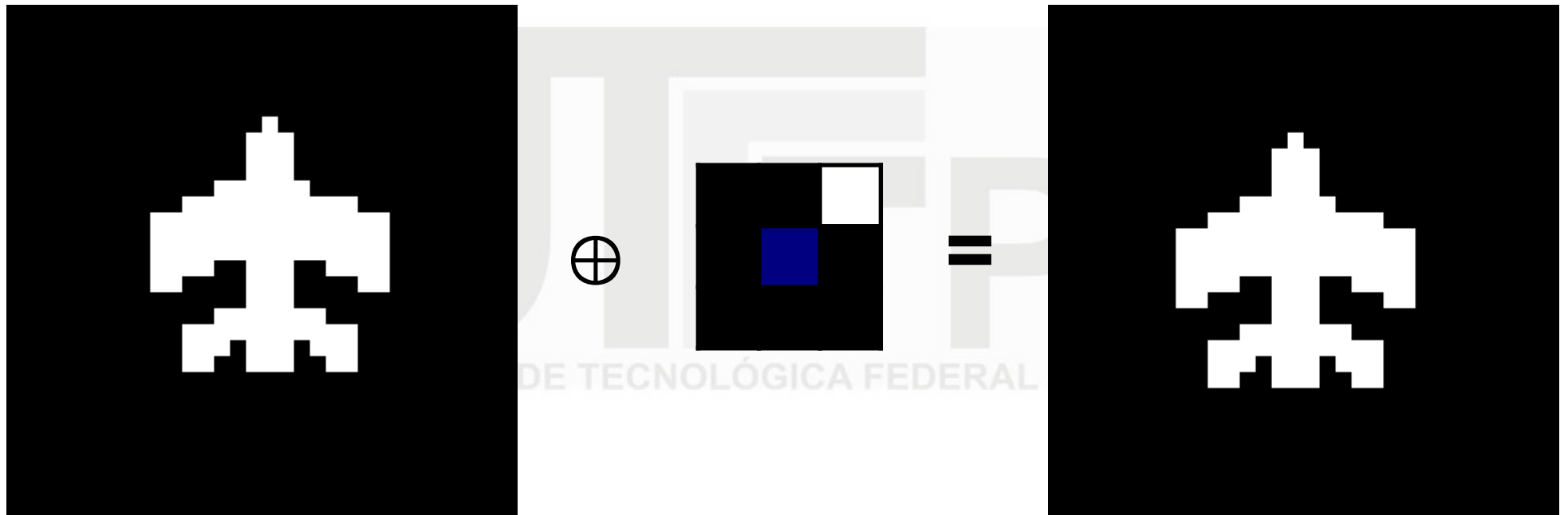
# Dilatação



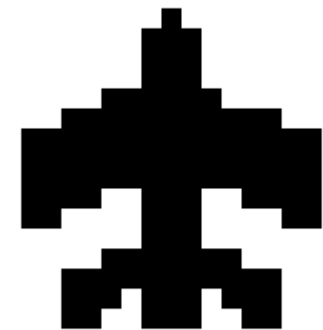
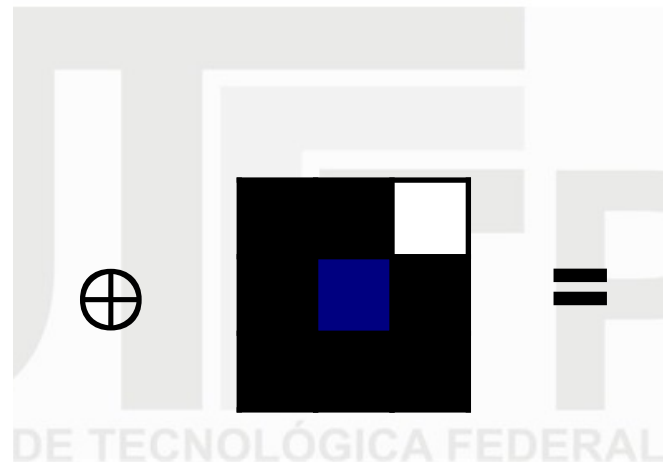
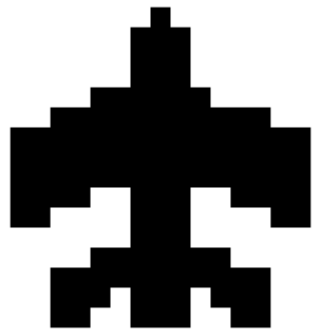
# Dilatação



# Dilatação



# Dilatação



# Erosão

$$I \ominus K$$

- A erosão “afina” as regiões de pixels brancos.

para cada vizinhança em torno de  $(x,y)$

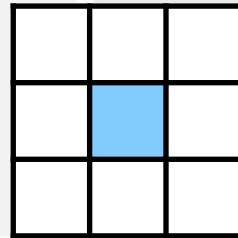
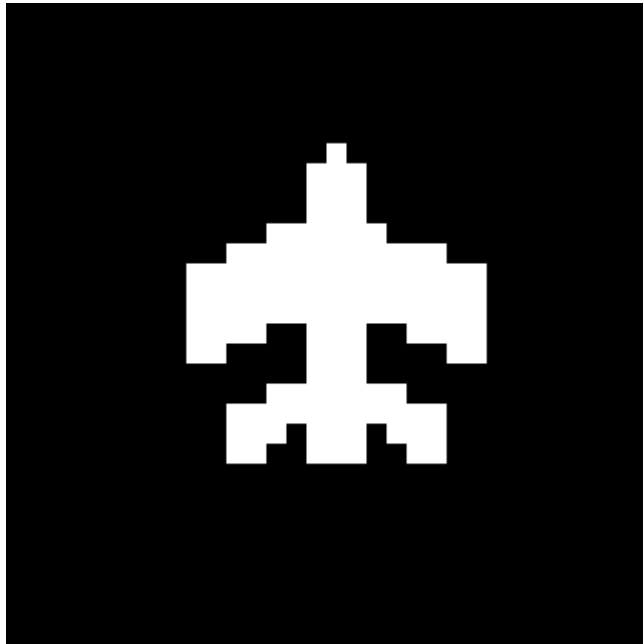
se todos os pixels do kernel estiverem setados na imagem

saída  $(x,y) \leftarrow 1;$

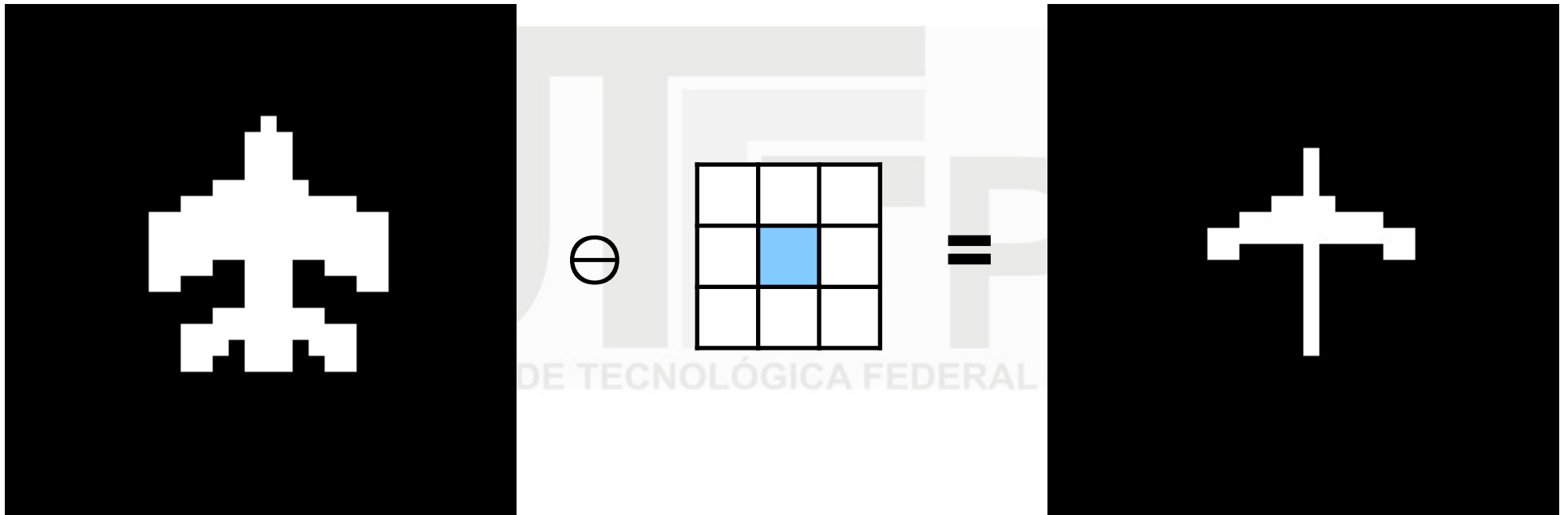
senão

saída  $(x,y) \leftarrow 0;$

# Erosão

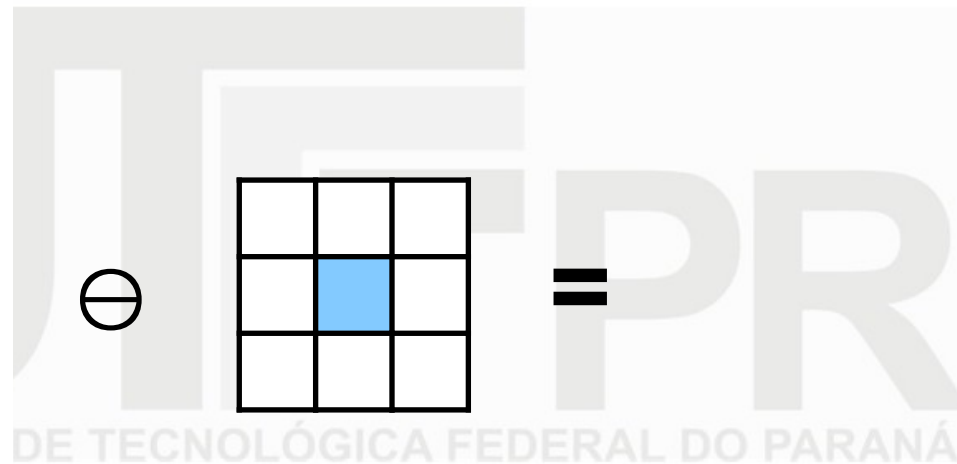
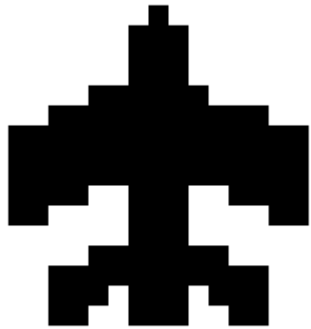


# Erosão

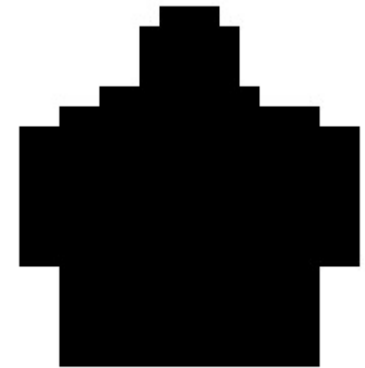
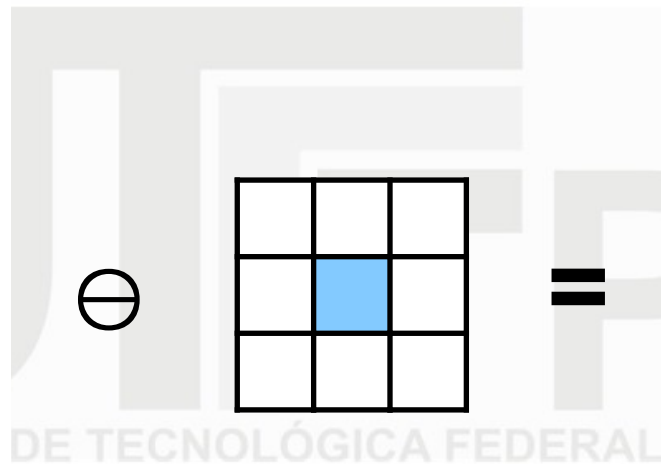
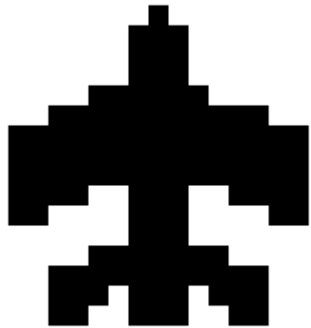




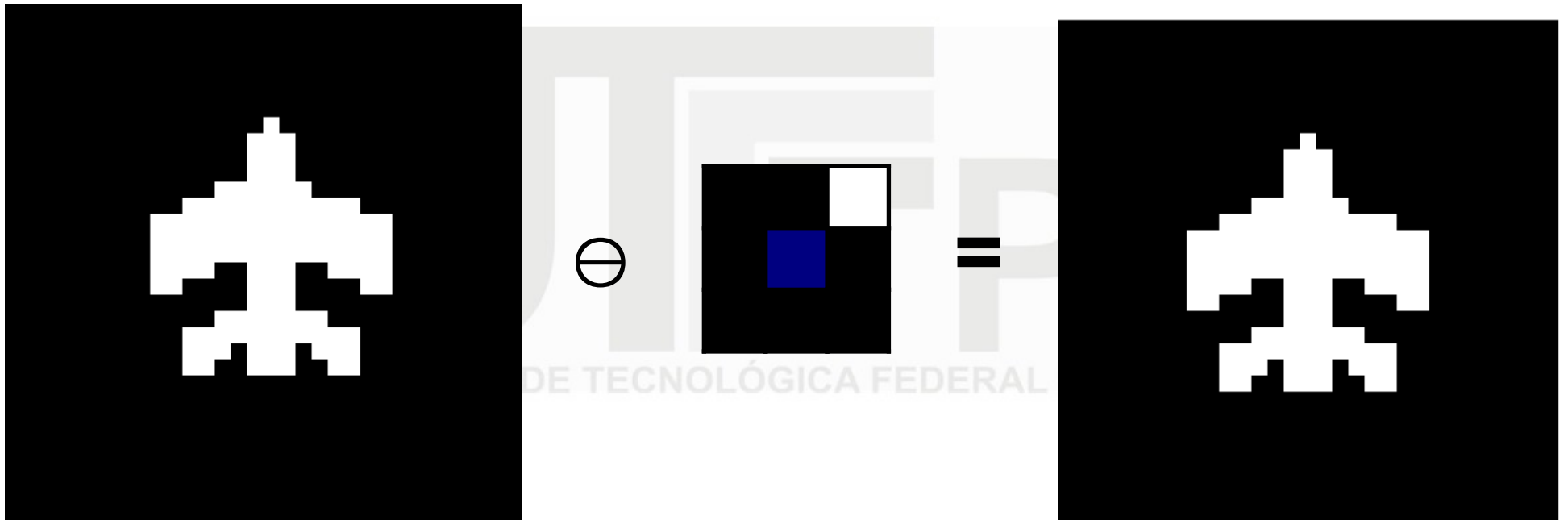
# Erosão



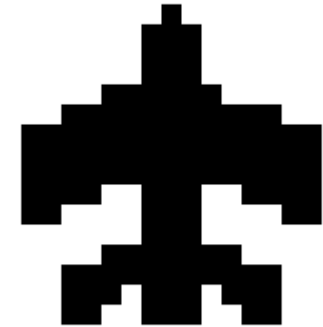
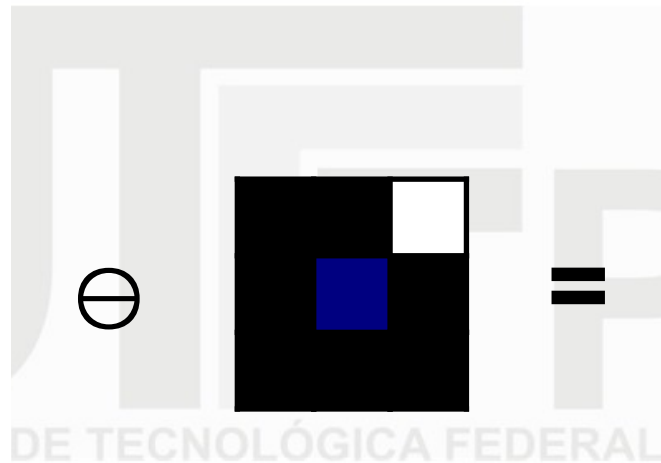
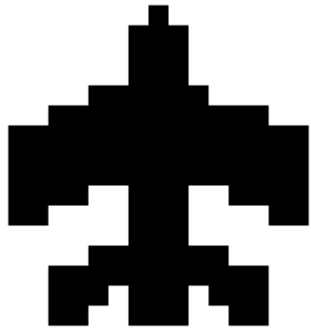
# Erosão



# Erosão



# Erosão

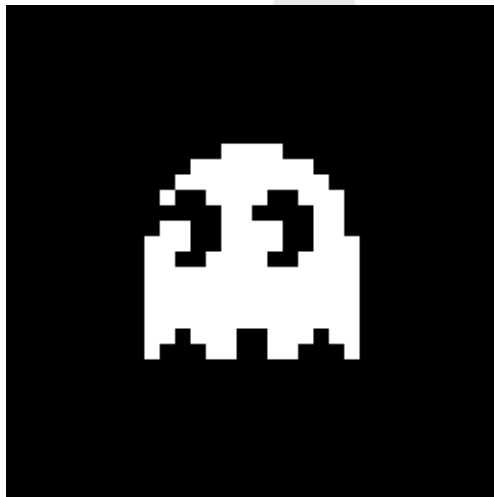


# Dilatação e erosão: notas

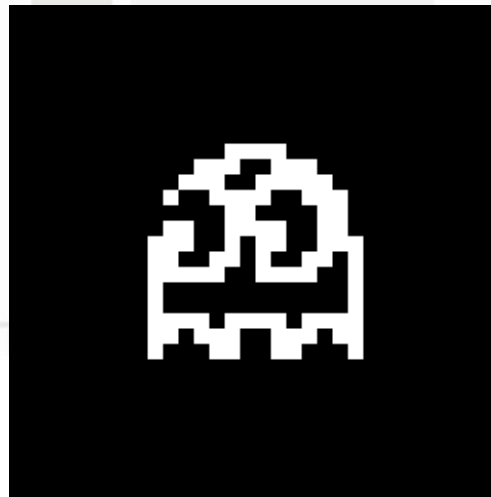
- Todos os outros operadores morfológicos se baseiam na dilatação e na erosão!
- Como dilatação e erosão funcionam como filtros espaciais, temos que lidar com as margens.
  - O que fazer se todos os pixels “setados” no elemento estruturante estiverem fora da imagem?!
- A ideia de que a dilatação “engorda” e a erosão “afina” os blobs só é válida quando o pixel central está “setado” no *kernel*.

# Usando...

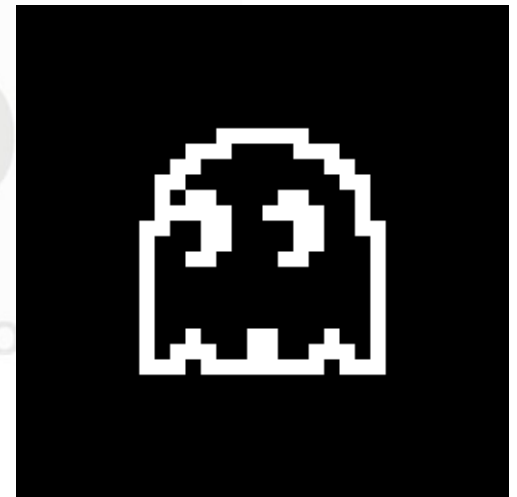
- Como poderíamos usar dilatação e/ou erosão para extrair os contornos de um blob?



Original



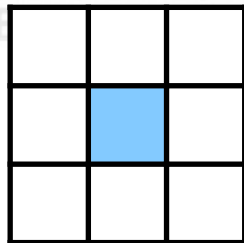
Contorno (interno)



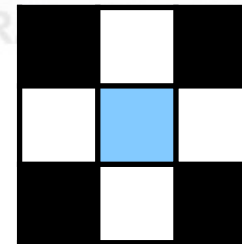
Contorno (externo)

# Contornos

- Como poderíamos usar dilatação e/ou erosão para extrair os contornos de um blob?
- Seja  $I$  uma imagem binária, e  $K$  um elemento estruturante 3x3:
  - Contorno interno:  $I - (I \ominus K)$
  - Contorno externo:  $(I \oplus K) - I$
- Os pixels setados em  $K$  indicam o tipo de vizinhança:

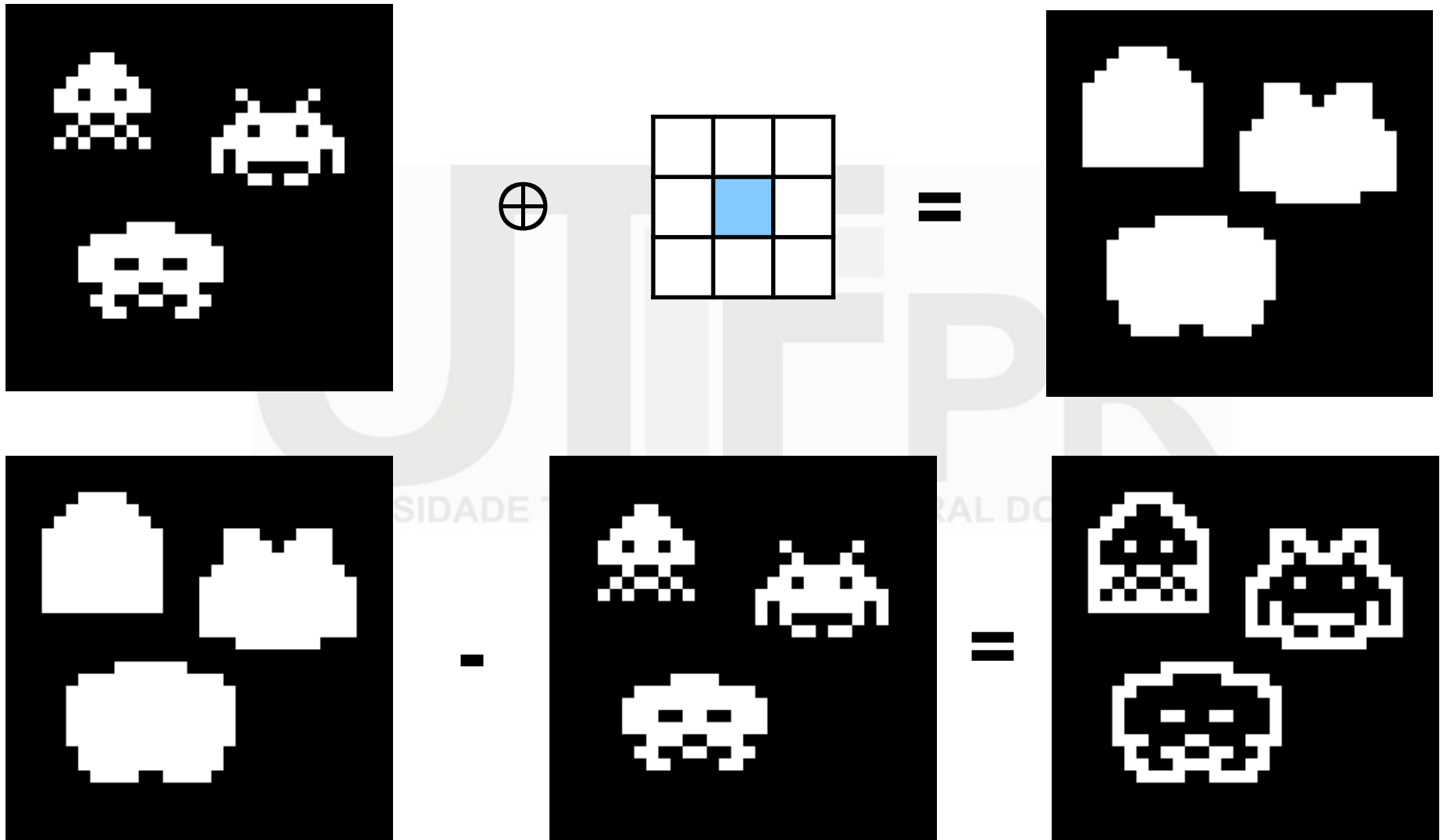


Vizinhança-8



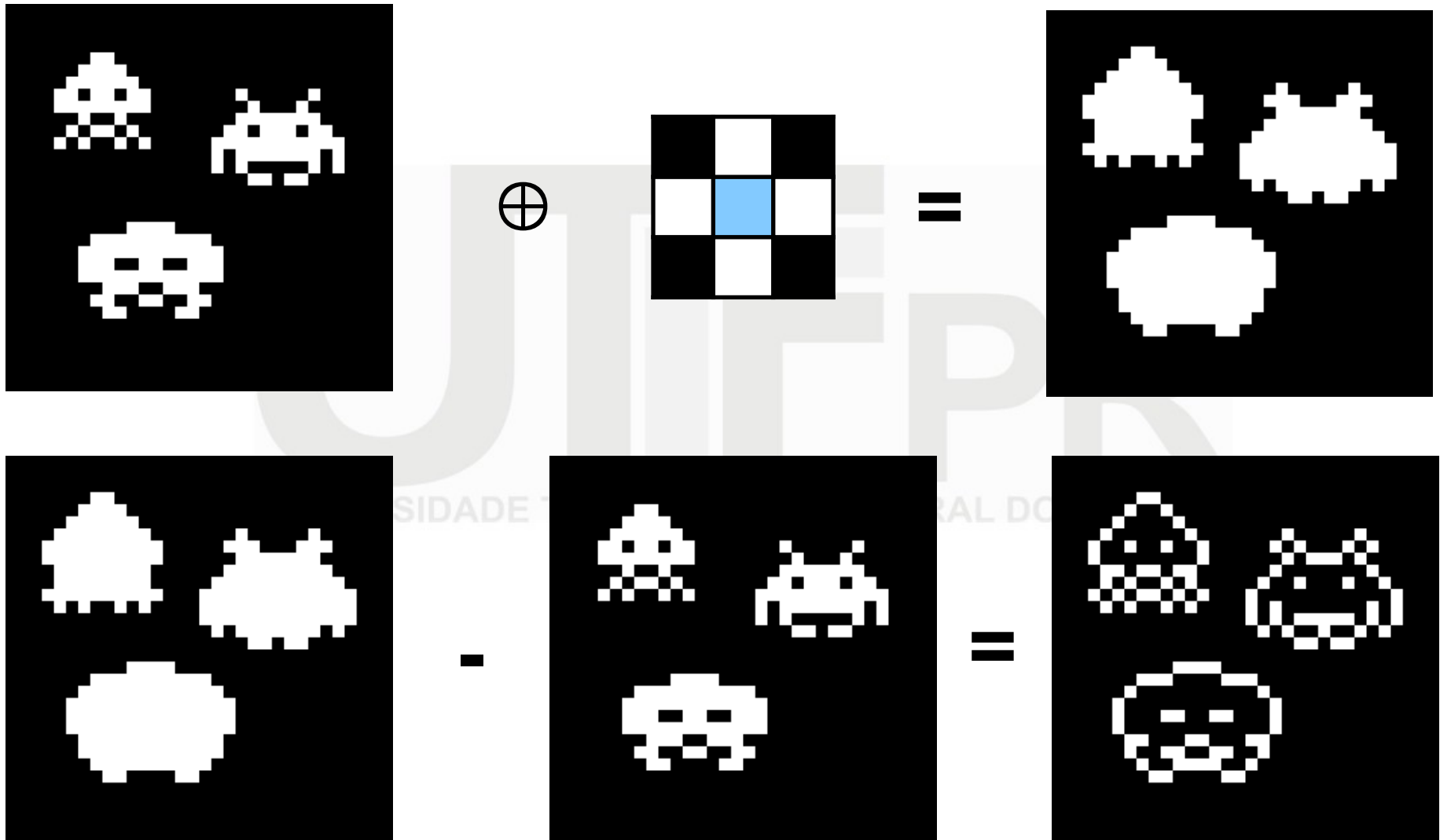
Vizinhança-4

# Exemplo





# Exemplo

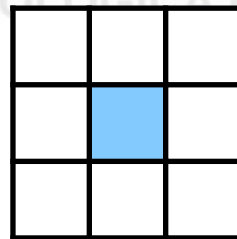


# Abertura

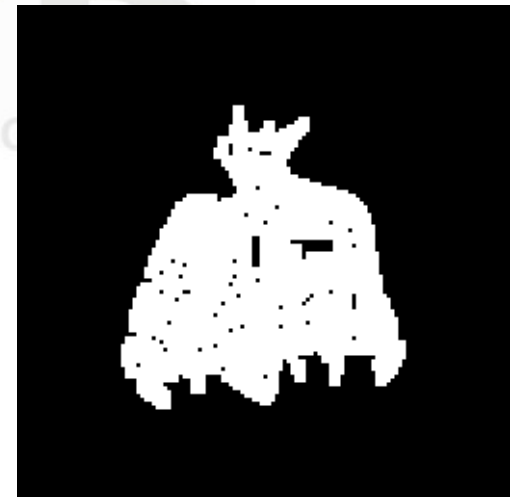
- Abertura: erosão seguida de dilatação.
  - $I \circ K = (I \ominus K) \oplus K$
  - Remove ruídos e protuberâncias.
  - Suaviza o formato do blob.
  - Usar várias vezes em seguida não muda o resultado!
  - $(I \circ K) \circ K = I \circ K$



○



=

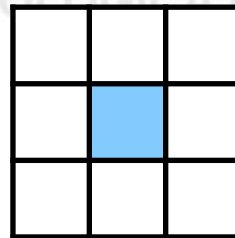


# Fechamento

- Fechamento: dilatação seguida de erosão.
  - $I \bullet K = (I \oplus K) \ominus K$
  - Fecha “buracos”.
  - Suaviza o formato do blob.
  - Usar várias vezes em seguida não muda o resultado!
  - $(I \bullet K) \bullet K = I \bullet K$



•



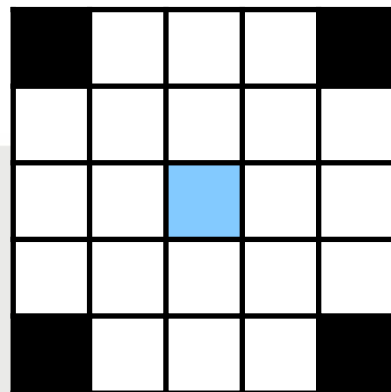
=



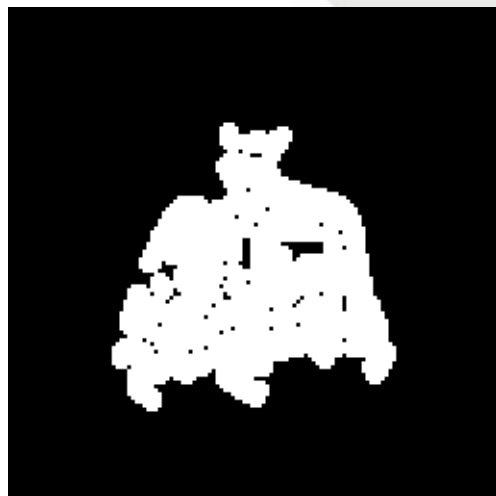
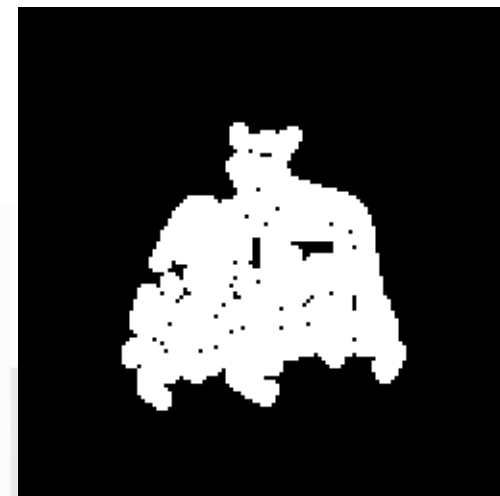
# Mais exemplos...



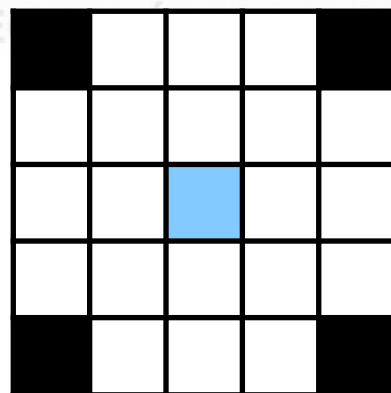
$\circ$



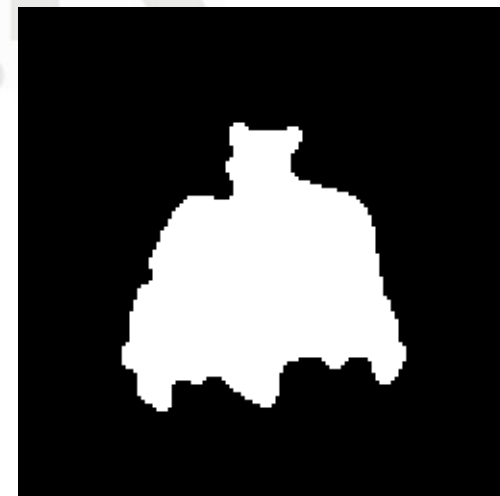
$=$



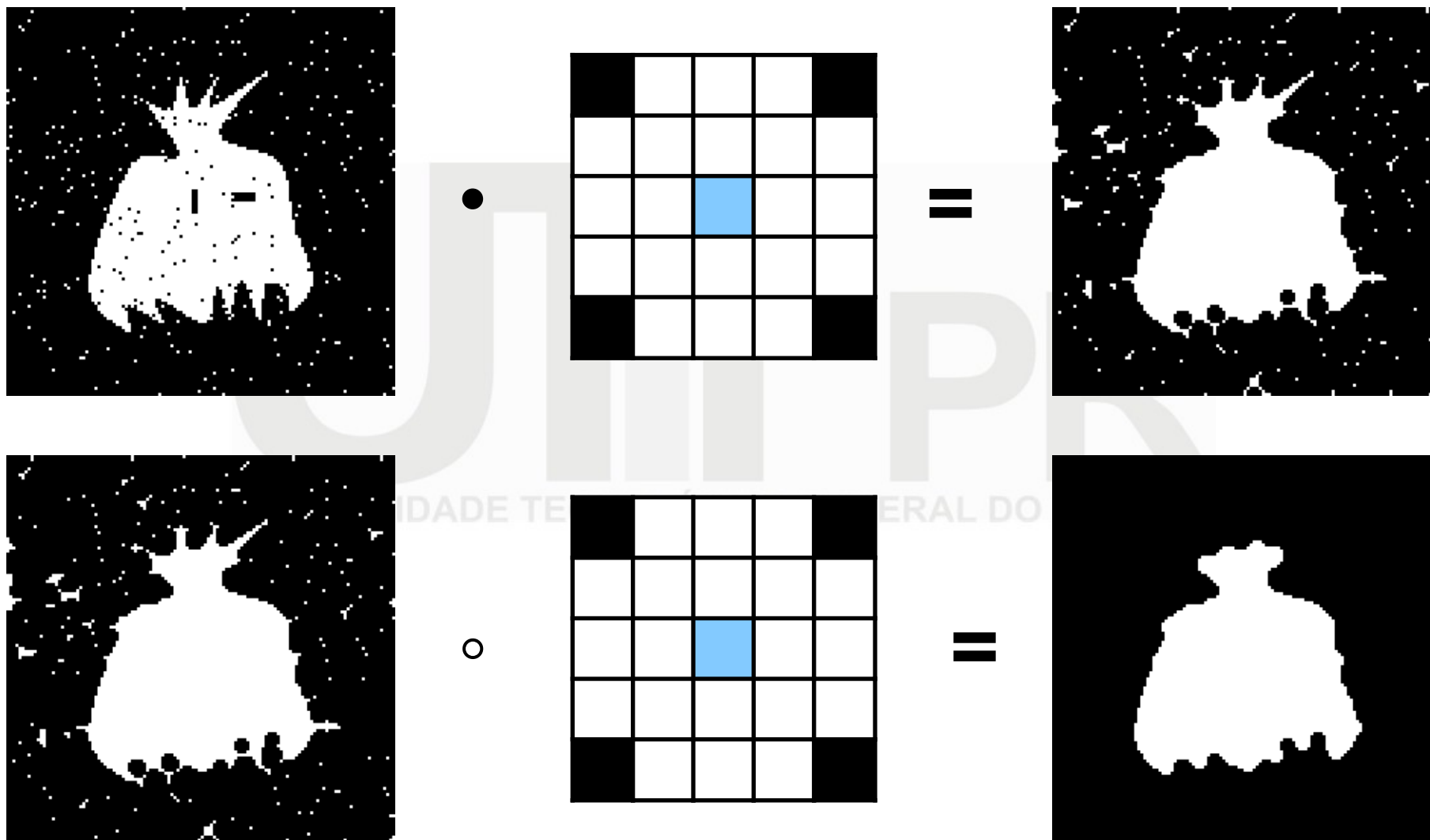
$\bullet$



$=$



# Mais exemplos...



# Morfologia: notas

- Existem outros operadores morfológicos, todos baseados em dilatações e erosões.
- Algumas aplicações que podem usar morfologia:
  - Análise de materiais.
  - (Esta foi a primeira aplicação que usou operações morfológicas!)
  - Reconhecimento de padrões em imagens binárias.
  - Afinamento de caracteres para reconhecimento automático.
  - Processamento de impressões digitais.

# Filtro da mediana binário

- Os operadores morfológicos são muito usados para remoção de ruído em imagens binárias.
- Uma alternativa rápida que pode resolver o mesmo problema em alguns casos é o filtro da mediana binário.
  - Em imagens binárias, a mediana de uma região é igual ao valor que mais aparece na região.
  - Como podemos usar isso para obter a mediana rapidamente?

# Filtro da mediana binário

- Os operadores morfológicos são muito usados para remoção de ruído em imagens binárias.
- Uma alternativa rápida que pode resolver o mesmo problema em alguns casos é o filtro da mediana binário.
  - Em imagens binárias, a mediana de uma região é igual ao valor que mais aparece na região.
  - Podemos simplesmente somar os pixels em cada vizinhança, e verificar se a soma é maior que o tamanho da região dividido por 2.
  - Já vimos um jeito rápido de somar os pixels em uma vizinhança!
    - Qual?



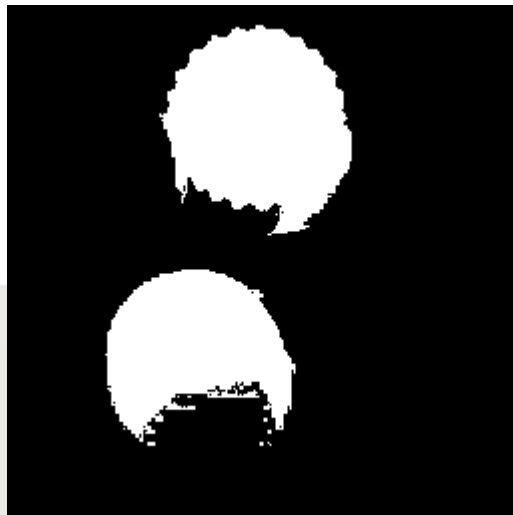
# Filtro da mediana binário

- Os operadores morfológicos são muito usados para remoção de ruído em imagens binárias.
- Uma alternativa rápida que pode resolver o mesmo problema em alguns casos é o filtro da mediana binário.
  - Em imagens binárias, a mediana de uma região é igual ao valor que mais aparece na região.
  - Podemos simplesmente somar os pixels em cada vizinhança, e verificar se a soma é maior que o tamanho da região dividido por 2.
  - Já vimos um jeito rápido de somar os pixels em uma vizinhança!
    - A estratégia baseada em imagens integrais.

# Filtro da mediana binário



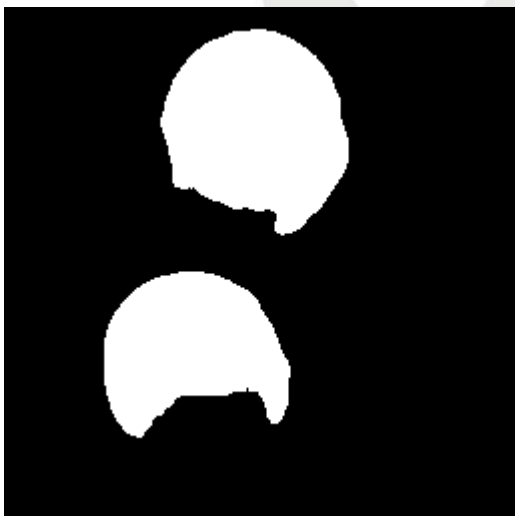
Original



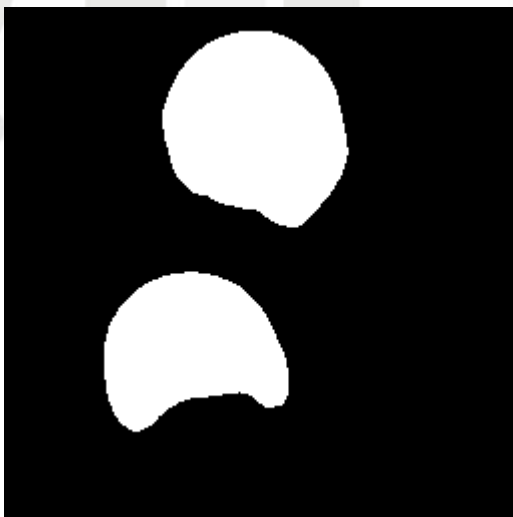
Binarizada



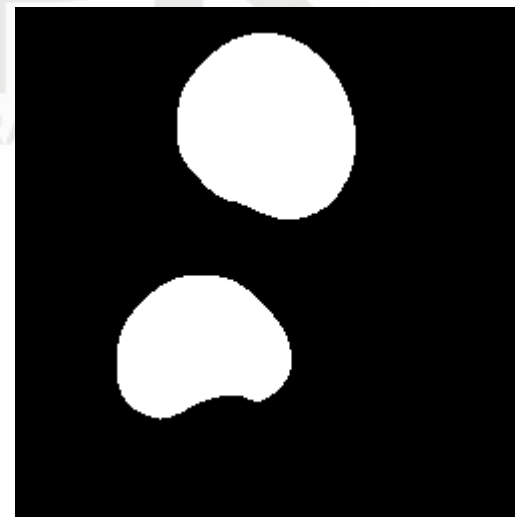
3x3



9x9



21x21

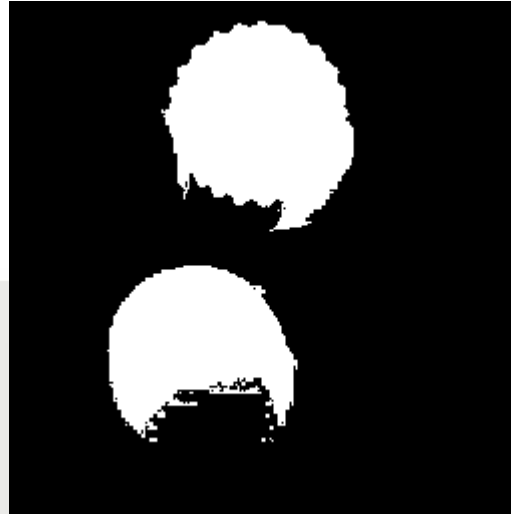


41x41

# Abertura (kernel circular)



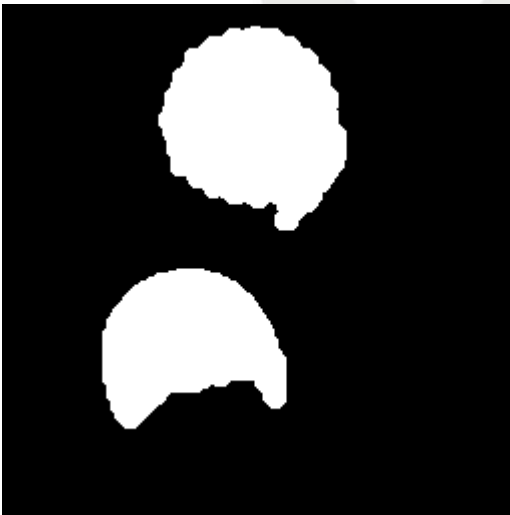
Original



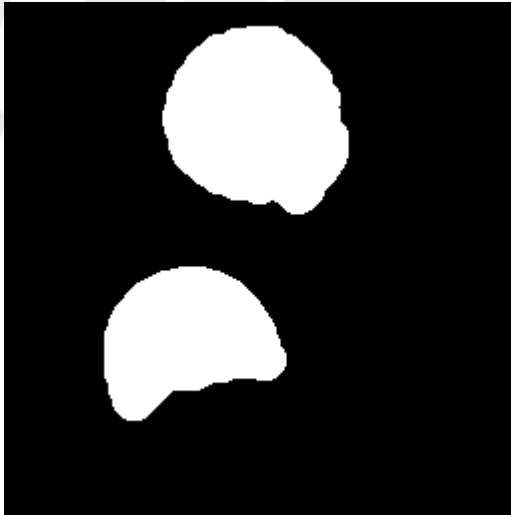
Binarizada



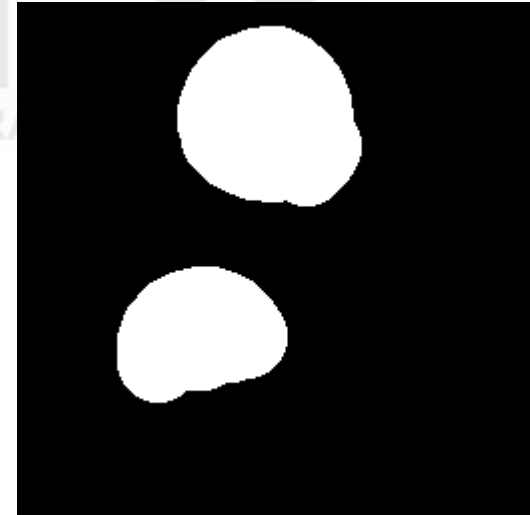
3x3



9x9



21x21



41x41

# Morfologia em escala de cinza

- Usamos até agora morfologia para imagens binárias.
- Os operadores de erosão e dilatação foram estendidos para imagens em escala de cinza.
- Dilatação:
  - ~~Pixel setado se pelo menos um pixel sob o kernel estiver setado.~~
  - Pixel recebe o maior valor sob o kernel.
- Erosão:
  - ~~Pixel setado se todos os pixels sob o kernel estiverem setados.~~
  - Pixel recebe o menor valor sob o kernel.
- As novas definições ainda funcionam para imagens binárias!

# Morfologia em escala de cinza

- Kernel quadrado com pixel de interesse no centro → fica igual aos filtros para mínimos e máximos locais que definimos!
  - Estes filtros são separáveis e fáceis de implementar.

# Imagem original



# Dilatação 21x21



# Erosão 21x21





# Abertura 21x21



# Fechamento 21x21

