

Sistemas de Visão e Percepção Industrial

Aula Prática nº 7

Morfologia binária

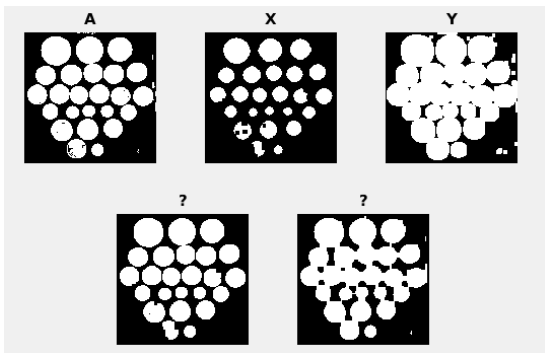
Sumário

- 1 Operações básicas de erosão e dilatação
- 2 Sobre os elementos estruturantes
- 3 Morfologia para deteção de padrões binários
- 4 Operações morfológicas de mais alto nível

Exercício 1)

Binarizar a imagem 'Manycoins.png' a 18% e obter A

- Aplicar uma operação de erosão de 3 vezes sobre A
- Aplicar uma operação de dilatação de 3 vezes sobre A
- As operações a usar são respetivamente
 - `X=bwmorph(A, 'erode',3);`
 - `Y=bwmorph(A, 'dilate',3);`
- Será possível re-obter A a partir de X ou de Y? Como ?



Exercício 2)

Para a imagem 'Manycoins.png' binarizada a 18%:

- Aplicar uma operação de erosão o número máximo de vezes possível, até se preservar pelo menos metade dos pixels brancos originais.
- Usar o ciclo "while" para repetição da operação de erosão.
- Explicar o código sugerido e adaptar para outras frações (como por exemplo preservar 1/3 dos pixels brancos originais)

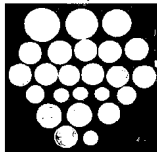
```
clear; close all
A=imread('Manycoins.png');
B=im2bw(A,0.18); BW=B;
tot = sum(sum(B));
part=tot;
n=0;
while (part>tot/2)
    BW=bwmorph(BW, 'erode');
    part=      (BW);
    n=n+1;
end

subplot(1,3,1), imshow(B)
title(sprintf('Original image\n %d pixels',tot));

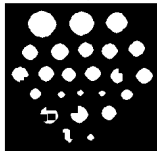
subplot(1,3,2), imshow(BW)
title(sprintf('Eroded %d times\n %d pixels',n,nnz(BW)));

BW=bwmorph(BW, 'erode', n-1);
subplot(1,3,3), imshow(BW)
title(sprintf('Eroded %d times\n %d pixels',n-1,nnz(BW)));
```

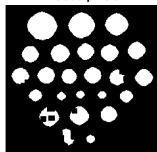
Original image
26705 pixels



Eroded 5 times
10901 pixels



Eroded 4 times
13505 pixels



Exercício 3)

- Binarizar a imagem 'Manycoins.png' a 18%;
- Representar o seu fecho e a sua abertura:
 - 'close', 'open' como argumentos de bwmorph.
 - Determinar qual das duas tem mais pontos em comum com a original binarizada;

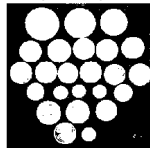
```
close all
A=imread('Manycoins.png');
B=im2bw(A,0.18);

tot=sum(sum(B));
subplot(1,3,1), imshow(B); title('Manycoins.png a 18%');
str=sprintf('Total pixels: %d', tot); xlabel(str);

BW=bwmorph(B, 'close');
comC=sum(      );
subplot(1,3,2), imshow(BW), title('close')
str=sprintf('Tem %d pixels comuns', comC); xlabel(str);

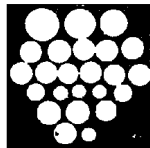
BW=bwmorph(B, 'open');
subplot(1,3,3), imshow(BW), title('open')
comO=sum(      );
str=sprintf('Tem %d pixels comuns', comO); xlabel(str);
```

Manycoins.png a 18%



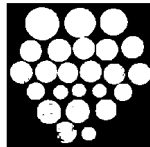
Total pixels: 26705

close



Tem 26666 pixels comuns

open



Tem 26481 pixels comuns

Elementos estruturantes

O elemento estruturante (SE) ...

- ... de uma operação morfológica condiciona o seu alcance e a sua forma de atuação em torno de cada pixel.

Para as operações com `bwmorph()`

- O Matlab assume sempre como elemento estruturante: `ones(3,3)`

Para usar outros SE diferentes de `ones(3,3)` é preciso usar operações especiais em Matlab:

- `imerode()`, `imdilate()`, `imopen()`, `imclose()`

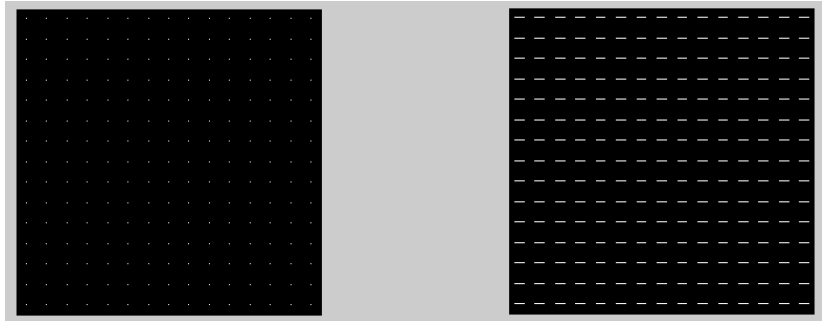
Origem (fulcro) do SE

- Em geral, em Matlab, o SE é assumido com a sua origem sempre no centro geométrico da matriz que o representa

Exercício 4)

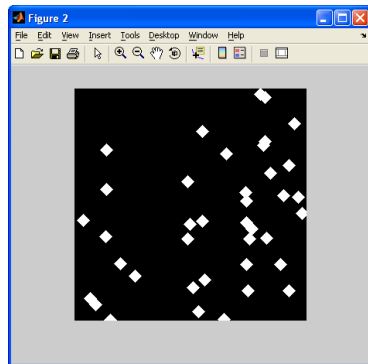
- Gerar uma imagem A com 300x300 e que contenha uma malha de pixels brancos de 20 em 20, começando em (10,10).
- Definir um elemento estruturante (SE1) tal que uma dilatação de A com SE1 gere a imagem ilustrada onde os traços brancos têm 10 pixels de comprimento
 - Código principal: `B=imdilate(A, SE1)`

$$B = (A \oplus SE_1);$$



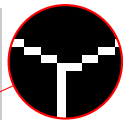
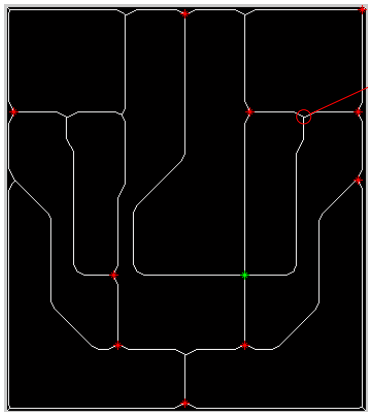
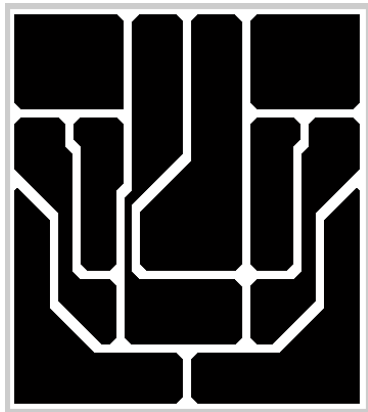
Exercício 5)

- Gerar uma imagem 300x300 com pixels brancos em 0.05 % da região em posições aleatórias (`imnoise` ou `rand(300,300)>0.9995`).
- À custa da função `strel()` do Matlab, gerar um elemento estruturante em forma de diamante (`diamond`) com 8 de dimensão;
- Sintetizar por dilatação (`imdilate`) uma imagem do tipo da ilustrada.



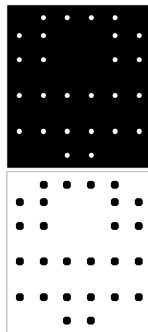
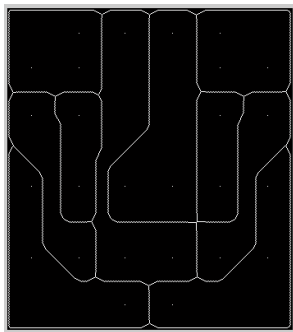
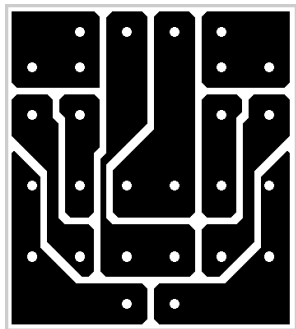
Exercício 6)

- Carregar a imagem 'pcb2.png', binarizá-la e obter o seu esqueleto. (bwmorph() com 'skel')
- Com o operador de hit-and-miss, detetar e assinalar no esqueleto os cruzamentos de pista em cruz ("+") e em "T" (função bwhitmiss())
- Observar os casos "não detetados" e indicar a razão da não deteção!



Exercício 7)

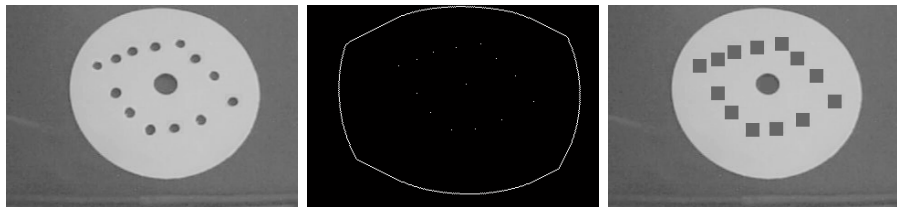
- Repetir a deteção para uma imagem com furos: 'pcb.png'.
- Problema do esqueleto com elementos excedentários!
- Remover os furos antes de calcular o esqueleto:
 - Usar a erosão condicionada só com as duas condições de preservar pontos isolados e "aneis": Função `bwmorph()` com `'shrink'`.
 - Obter os pontos isolados e com base neles, gerar uma máscara para remover os objetos circulares (furos).
 - O resto do procedimento é igual ao exercício anterior.



Exercício 8) Opcional

Carregar a imagem 'lixa10.png' e gerar uma nova imagem similar, mas onde se substituem os furos por furos quadrados com 15 pixels de lado e nível de cinzentos a 40%. Sugestão de procedimento:

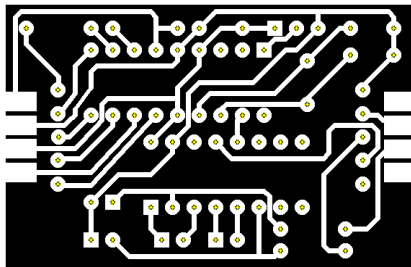
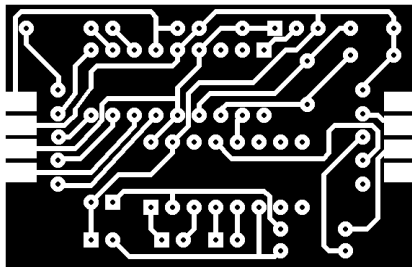
- Binarizar a imagem original;
- localizar os furos com 'shrink' e detecção de pontos isolados;
- Criar uma máscara para os furos quadrados por dilatação;
- Usar a máscara para alterar a imagem original.



- Nota: O furo central não parece substituído porque tem uma dimensão maior do que 15 pixels, mas ele também é contemplado no procedimento.

Exercício 9)

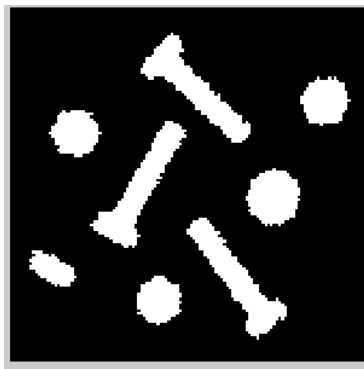
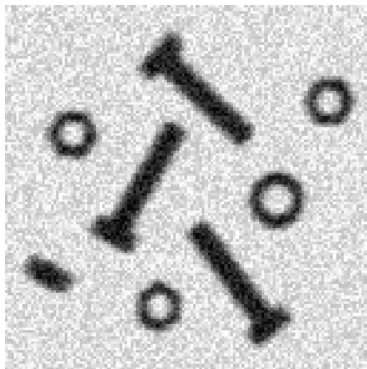
- Contar e assinalar todos os furos da imagem 'pcb_holes.png'. (Solução: 71 furos).
- Sugestão: Notar que os furos são entidades isoladas dos outros objetos.



- Nota: Para confirmar que os pontos foram bem detetados, usando a função 'plot', sobrepor na imagem pontos de uma cor distinta sobre os furos (Cf. figura da direita).

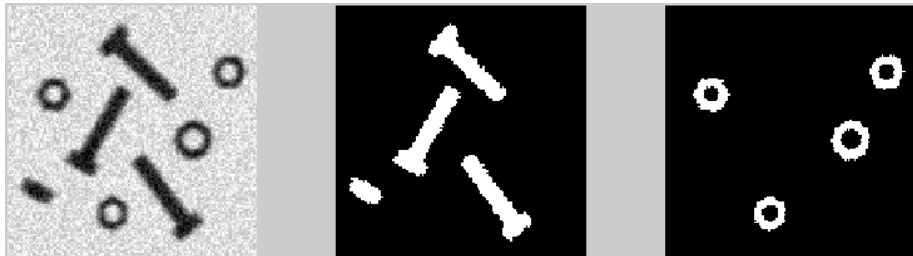
Exercício 10)

- Carregar a imagem 'porcas.png', binarizá-la automaticamente (para obter uma máscara) e encher os "buracos" dessa máscara usando a função `imfill()` do Matlab (Consultar o manual para ver os parâmetros).



Exercício 11)

- A partir da imagem 'porcas.png', criar duas imagens binárias (máscaras): uma com os objetos sem furos e a outra com os objetos com furos.
- Elementos para a solução:
 - Obter o esqueleto reduzido da imagem ("shrink")
 - Separar os pontos isolados do restante.
 - Fazer as duas reconstruções (`imreconstruct()`)



Exercício 12) Opcional

- Na imagem 'HappySad.png' que está a nível de cinzentos (A), usando operações morfológicas, eliminar todas as figuras de caras "tristes" (B). Aqui, "eliminar" significa colocar os respectivos pixels ao valor 1.
- Elementos para a solução:
 - Procurar uma característica distintiva entre os dois tipos de símbolos.
 - Obter a máscara para os símbolos "tristes".
 - Usar a máscara para forçar a 1 os respectivos pixels na imagem original.
 - N.B. As operações morfológicas operam sobre imagens binárias.

