

Sistemas de Visão e Percepção Industrial

3-Processamento a Baixo Nível

Parte 1 - Conceitos Base e Técnicas de Suavização

1 Relações básicas entre pixels

2 Pré-processamento

3 Filtros

4 Suavização (Smoothing)

- Bibliografia sugerida
 - Gonzalez, Chap. 3
 - Burger, Chap. 5,6
 - Davies, Chap. 3,4

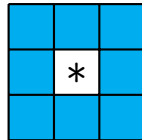
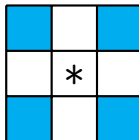
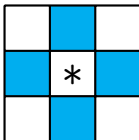
Relações básicas entre pixels

Vizinhanças de pixels

- Para um certo pixel '*' é possível definir vários tipos de vizinhança (conjuntos de pixels que estão em contacto com ele).
- Por vezes, os 8 vizinhos de um pixel têm a seguinte designação:

NW	N	NE
W	*	E
SW	S	SE

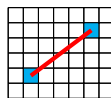
- As vizinhanças mais comuns de um pixel '*' são:
- N_4 (horizontal e vertical)
- N_D (diagonal)
- $N_8 = N_4 \cup N_D$



Medidas de distância entre pixels

- Euclidiana

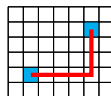
$$D_E(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$$



$$D_E = \sqrt{(4^2 + 3^2)} = 5$$

- D_4 (ou city-block)

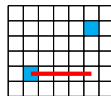
$$D_4(p, q) = |p_x - q_x| + |p_y - q_y|$$



$$D_4 = 4 + 3 = 7$$

- D_8 (ou chessboard)

$$D_8(p, q) = \max(|p_x - q_x|, |p_y - q_y|)$$



$$D_8 = \max(4, 3) = 4$$

Pré-processamento

Pré-processamento de imagem

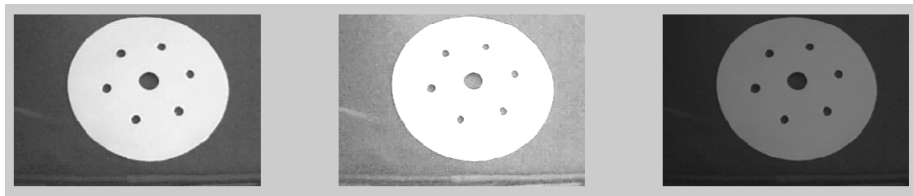
- Abordagens mais usuais
 - Espacial - sobre os pixels da imagem
 - Na frequência - Transformadas de Fourier
- Procedimentos comuns na abordagem espacial
 - Criação de uma nova imagem $g(x, y) = h[f(x, y)]$
 - $h()$: operador de pré-processamento que opera num pixel e/ou na sua vizinhança
 - Uso de máscaras de convolução (ou janelas ou filtros)
 - Matriz de coeficientes apropriados a determinado tipo de tratamento ou detecção de propriedades

Operações em pixels

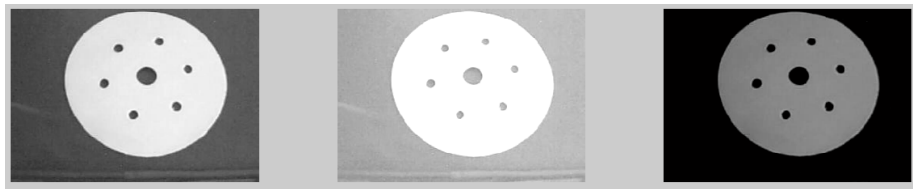
- Operações absolutas – operam em pixels individuais
 - Mudar a intensidade de um pixel
 - $g(x, y) = f(x, y) + a$
 - afeta a luminosidade de forma igual para todos os pixels
 - $g(x, y) = kf(x, y)$
 - afeta a luminosidade em função do valor do pixel – alterando o contraste geral
 - Acautelar a saturação: $g(x, y) \leq valormax$ (100%, 255, ...)
- Operações num pixel com base na vizinhança
 - Filtros de convolução
 - Operação que calcula um novo valor do pixel com base no valor original e nos da sua vizinhança.

Exemplos de operações absolutas

- Fator de escala: 2 e 0.5

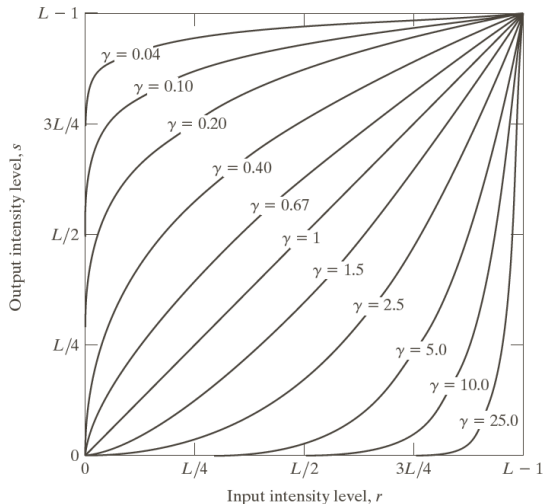


- Adição de constante: +100 e -100 (imagem: 0-255)



Transformação de intensidade não linear

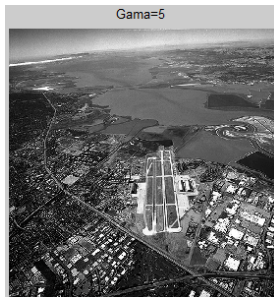
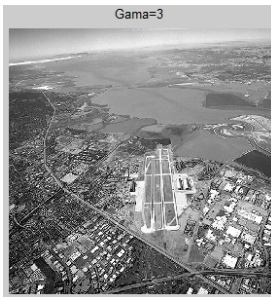
- Correção Gama



- Alteração do valor de um pixel de forma não linear, e em função do seu valor.
- Equação: $s = C \times r^\gamma$
- Para todas as curvas ilustradas tem-se: $C=1$;

Transformação de intensidade não linear (cont.)

- Correção Gama: $s = r^\gamma$



Filtros

Princípio da utilização de um filtro

- O novo pixel é função do pixel original e do filtro:

$$g(x, y) = \sum_{i=1}^N w_i \times p_i$$

- onde p_i pertence à vizinhança do ponto em causa
- Note-se que $g(x, y)$ pode ser o valor direto do novo pixel, ou ser usado de forma indireta e assim permitir calcular o seu novo valor.
- Os coeficientes de um filtro de 3x3:

$$\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix}$$

Formalização da operação com um filtro

- Para todos os efeitos, um filtro é uma matriz bidimensional
- A aplicação do filtro F sobre a imagem A resulta na imagem B :
 - $B = A \star F$, onde o operador \star se designa "correlação cruzada".
 - Se o filtro F for simétrico, esta operação é equivalente à "convolução"
 - Representa-se por: $B = A * F$
 - Como, em imagem, os filtros costumam ser simétricos, usa-se normalmente o termo filtro "convolução" e não de "correlação".
- Em Matlab o código correspondente à correlação/convolução é:
 - `B=filter2(F, A)`

Filtros de "convolução"

- $B = A * F$ [em matlab $B = \text{filter2}(F, A)$]
- Define-se matematicamente do seguinte modo:

$$B(u, v) = \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} A(u-i, v-j) \cdot F(i, j)$$

$$B(u, v) = \sum_{(i,j) \in R_F} A(u-i, v-j) \cdot F(i, j) = \sum_{(i,j) \in R_F} A(u+i, v+j) \cdot F(-i, -j)$$

$$B(u, v) = \sum_{(i,j) \in R_F} A(u+i, v+j) \cdot F^*(i, j)$$

$$F^*(i, j) = F(-i, -j)$$

Que representa a reflexão horizontal e vertical de F , ou seja, uma rotação de 180° da matriz F . Se F for simétrica, então virá $F^* = F$

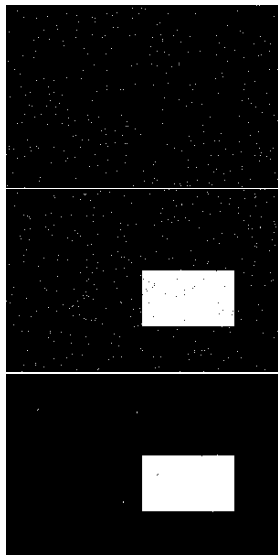
Exemplo de Filtro – para pontos isolados

- Numa imagem **binária** (pixels com valores 0 ou 1), o seguinte filtro pode ser usado para detetar se um pixel é um ponto isolado no meio de outros.

$$F_{PI} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

- Se $g(x,y) = 8$ ou -8 o ponto (x,y) é isolado!
 - Note-se que os valores de $-7, -6, -5, \dots, 6, 7$ correspondem a todas as outras possibilidades que não são pontos isolados.

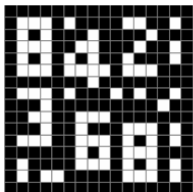
Ilustração do filtro de pontos isolados



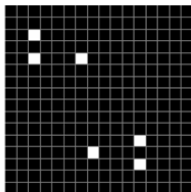
- Imagem com pontos isolados e outros.
- Pontos isolados na imagem detetados com o filtro.
- Imagem sem pontos isolados.
- Obtida a partir das duas anteriores.

Exercício de aplicação de um filtro (exame 2010)

- Indicar um filtro F de 3×3 que possa ser usado para obter a imagem B a partir da imagem A



A



B

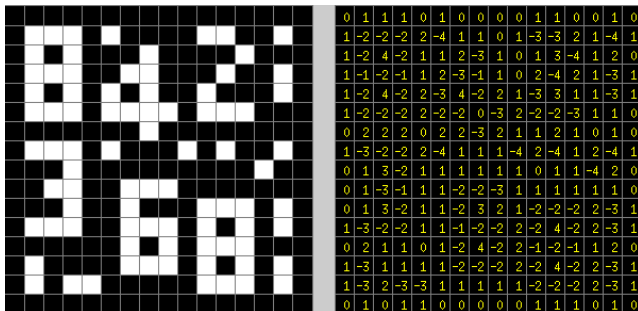
- Sugestão: os pixels brancos de B resultam de todos os pixels da A que têm pixels brancos na vizinhança N4
- O resultado da operação de $A * F$ não dá B, mas B pode-se obter a partir daquele resultado.
- O problema tem múltiplas soluções!

Exercício de aplicação (cont.)

- Optando pelo filtro

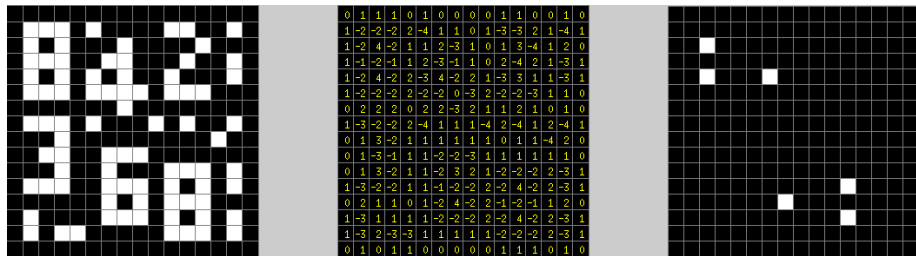
$$F = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- Viria o seguinte resultado:



Exercício de aplicação (concl.)

- Os pontos solução são aqueles onde o processo deu o valor 4.



A

$A2 = \text{filter2}(F, A);$

$B = (A2 == 4);$

Suavização (Smoothing)

Técnicas de Suavização

- As técnicas de suavização dizem em geral respeito à redução de ruído.
- Média de imagens (série de N imagens da mesma cena)

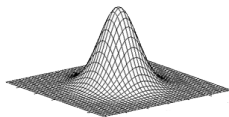
$$g(x, y) = \frac{1}{N} \sum_{i=1}^N f_i(x, y)$$

- Filtro de média
 - Um filtro de 3x3 com todos os pesos de valor 1/9.

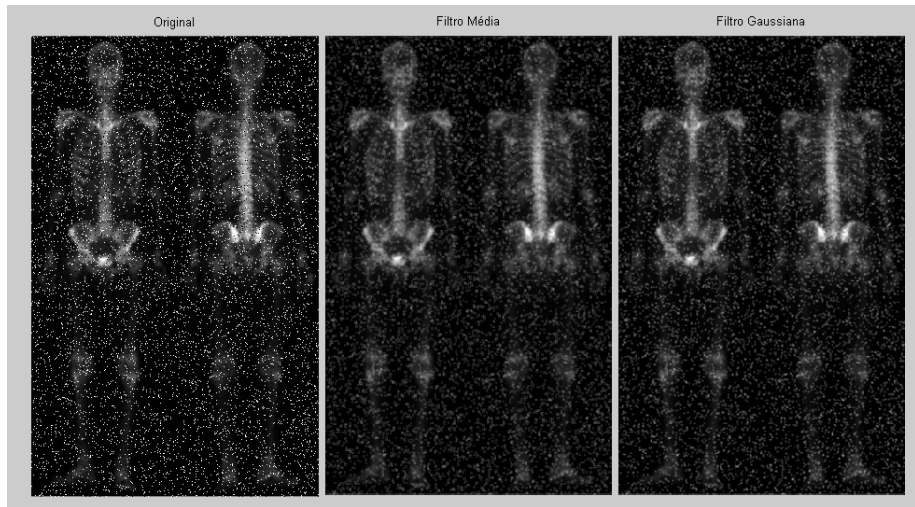
$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

- Filtro 3x3 que aproxima uma Gaussiana

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



Aplicação de filtros de média e gaussiana



Código da operação dos filtros

```
B=imread('BadBoneScan.png');
B=im2double(B);
subplot(1,3,1)
imshow(B); title('Original');
f=ones(3,3)/9;
g=1/16*[1 2 1
        2 4 2
        1 2 1];
C=filter2(f,B);
D=filter2(g,B);
subplot(1,3,2), imshow(C); title('Filtro Media');
subplot(1,3,3), imshow(D); title('Filtro Gaussiana');
```


"Filtro" de Mediana

- Substitui cada pixel pela mediana da distribuição dos níveis de todos os pixels da sua vizinhança (incluindo o próprio). Filtros de 3x3 ou 5x5 são comuns.
- Vantagens em relação à média...? Não cria novos valores!
- Exemplo sobre um pixel com os seus 8 vizinhos ("filtro" de 3x3):

12	10	15
11	12	9
10	11	15

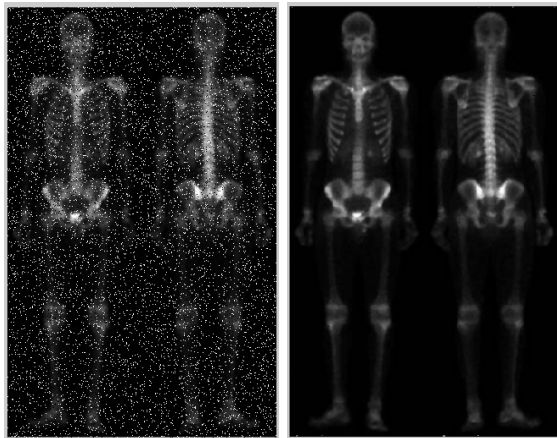
9, 10, 10, 11, 11, 12, 12, 15, 15

12	10	15
11	11	9
10	11	15

- N.B. Embora se use a expressão "Filtro" quando se refere a operação de mediana, não se trata de um verdadeiro filtro de convolução como os anteriores — a operação pertence à categoria dos filtros ditos **não lineares**.

Exemplo de aplicação do “filtro” de Mediana

- Imagem “BadBoneScan.png” que tem muito ruído! Efeito da mediana.



```
B=imread('BadBoneScan.png');  
C=medfilt2(B);  
subplot(1,2,1), imshow(B) subplot(1,2,2), imshow(C)
```