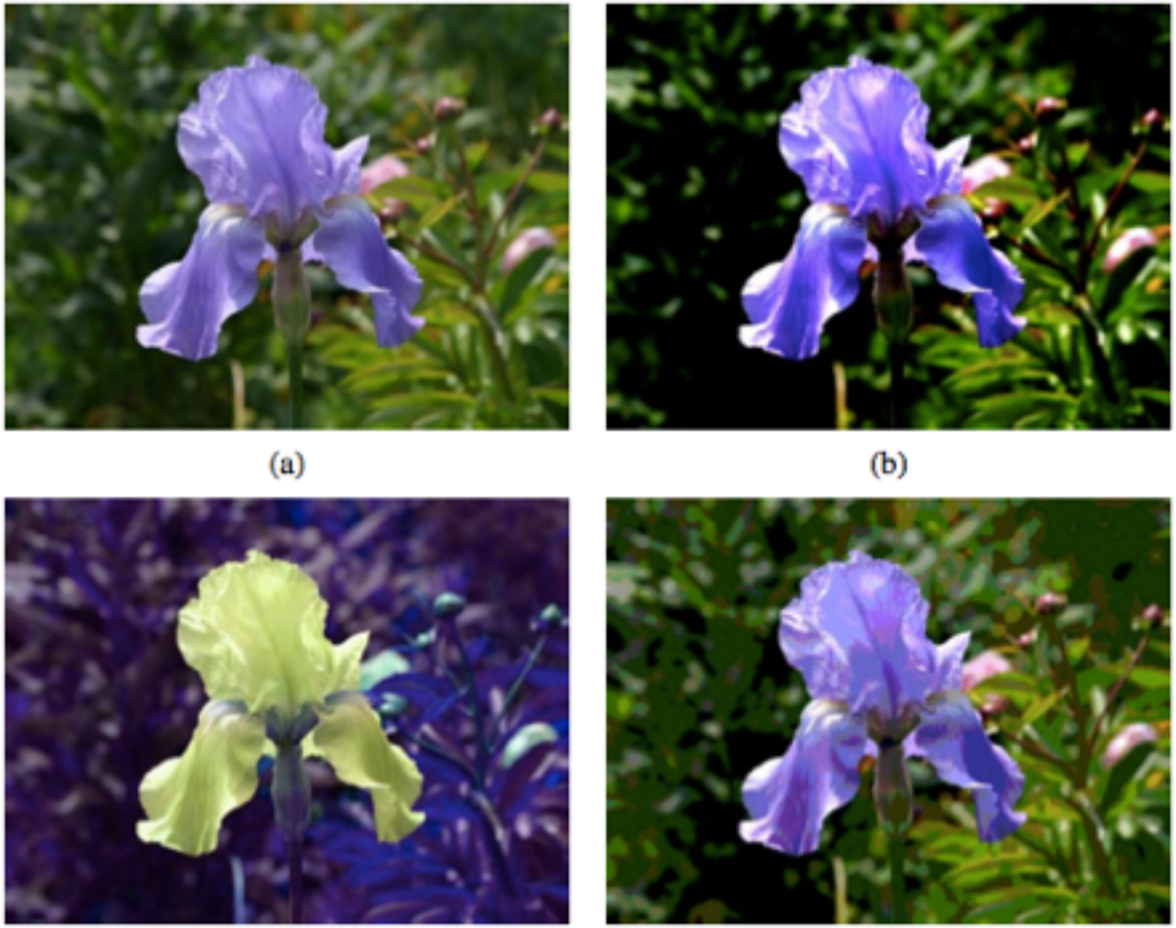
FEUP

Departamento de Engenharia Electrotécnica e de Computadores

Processamento de imagem

- Algumas técnicas de processamento de imagem
  - operadores locais
    - transformação de pixels e de cor
    - equalização de histogramas
  - operadores espaciais
    - Filtragem, Convolução
      - Derivadas da Imagem
      - Algumas funções do Matlab
  - detecção de contornos
    - Operadores de Gradiente
      - Prewitt
      - Sobel
      - Gradiente Gaussiano (Canny)
  - relações básicas entre pixels/valores no espaço bidimensional

Exemplos comuns de processamento de imagem



- aumentar o contraste; modificar tonalidade; quantizar

Porquê processar imagem?

- Filtrar e melhorar imagem
  - a imagem pode necessitar de ser melhorada porque
    - tem ruído
    - tem pouco contraste
    - é necessário evidenciar contornos
    - etc
- Extrair características intrínsecas da imagem
  - para serem posteriormente processadas (por exemplo por técnicas de inteligência artificial) para derivar conhecimento adicional
    - em aplicações de videovigilância para detectar e seguir pessoas
    - na área médica para detectar e contar células cancerígenas
    - na área do entretenimento para caracterizar as preferências do utilizador
    - etc

Operadores pontuais ou locais

- é a forma mais simples de processamento de imagem
  - onde o valor do pixel de saída depende apenas do valor do pixel de entrada
  - exemplos de operadores
    - ajuste de contraste, de brilho, correcção e transformação de cor
  - um operador é uma função que toma como entrada uma ou mais imagens, produzindo uma nova imagem à saída
    - no domínio digital, a imagem de entrada f(x,y) é um conjunto finito de localizações de pixels com um dada intensidade, p(x,y)
      - $g(x,y) = h(f(x,y))$
  - existem operadores de intensidade e de cor

Operadores pontuais de pixel

- $g(x,y) = h(f(x,y))$
  - operadores comuns são a adição e multiplicação
    - $g(p) = a \cdot f(p) + b$
    - os parâmetros *a* e *b* são designados de *ganho* e *bias* e muitas vezes referidos como parâmetros de contraste e de brilho
      - eles próprios podem ser função do espaço (das coordenadas espaciais)
        - $g(p(x,y)) = a(x,y) \cdot f(p(x,y)) + b(x,y)$
    - são operadores lineares

FEUP

Departamento de Engenharia Electrotécnica e de Computadores

Operadores pontuais de pixel - exemplo

- operador *linear blend*
  - recebe 2 imagens f_0 e f_1 à entrada e produz na saída uma nova imagem que combina as 2 entradas
    - $g(p(x,y)) = (1 - \alpha)f_0(p(x,y)) + \alpha f_1(p(x,y))$
    - variando o valor de α de 0 até 1, este operador pode ser usado para realizar um *cross-dissolve* entre duas imagens ou 2 vídeos
    - tal como em produção de filmes, slide-shows
    - pode ser usado para inserir uma marca-de-água
    - em televisão, juntamente com segmentação de imagem é usado para juntar objectos e pessoas captadas numa cena numa outra cena com um fundo diferente
      - tal como as gravações em estúdio com fundo azul (chroma-keying)

Operadores pontuais de pixel - linear blend

- *linear blend* ➛ *Chroma-keying*
  - objecto / pessoa captado em frente a um fundo de cor conhecida
  - segmentação dessa imagem para isolar o objecto / pessoa
  - mistura do objecto / pessoa isolada e da nova imagem
    - operador *linear blending*
      - entradas:
        - imagem com fundo azul e objecto / pessoa (**foreground** image in blue background)
        - imagem com fundo real (**background** image)

- os pixels não azuis da imagem *foreground* vão-se sobrepor aos pixels da imagem *background*

Operadores pontuais de pixel - linear blend (2)

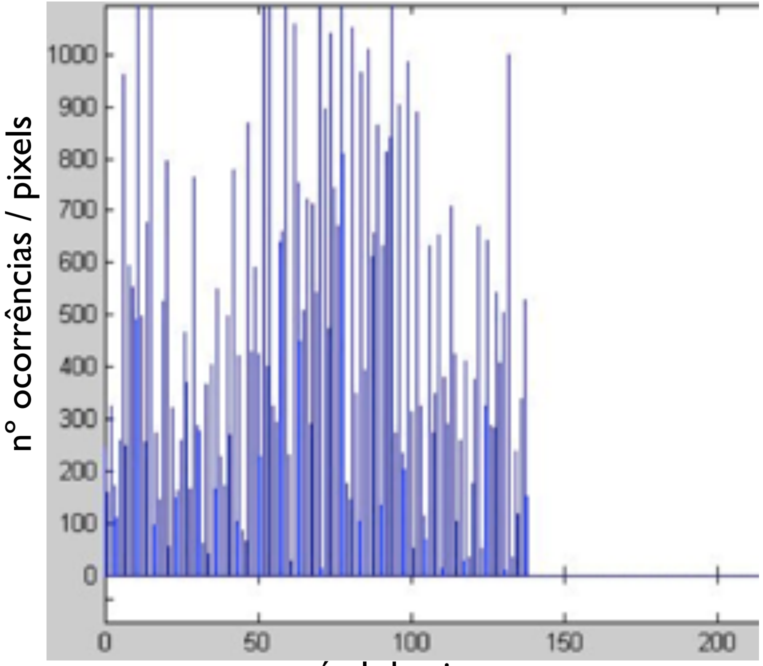
- *linear blending*
  - $g(p(x,y)) = (1 - \alpha)f_0(p(x,y)) + \alpha f_1(p(x,y))$
- o valor de $\alpha \in [0 ; 1]$ vai controlar a interpolação linear entre os pixels da imagens *foreground* e *background* quando se faz a composição das imagens

- o α codifica a informação de cobertura dos pixels
  - $\alpha = 0$: não há cobertura (transparente)
  - $\alpha = 1$: cobertura total (opaco)
  - $0 < \alpha < 1$: cobertura parcial (semi-transparente)

Histogramas

- Um histograma regista a distribuição de níveis de cinzento numa imagem
  - quantas vezes cada nível ocorreu
  - contagem do nº de pixels para cada nível de cinzento

256x256



Histogramas (2)

- Código C:

```
for (i=0;i<m,i++)
    for (j=0;j<n,j++)
      hist[I[i,j]]++;
```

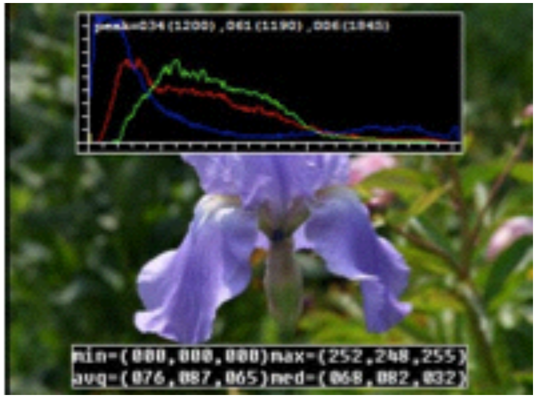
- Operações com histogramas
  - equalização
  - modificação
  - especificação
- MATLAB
  - imhist(image)
  - imhistmatch(image,reference)
  - histeq(image)

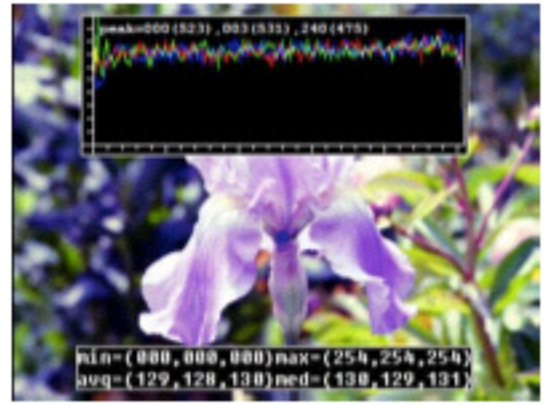
Departamento de Engenharia Electrotécnica e de Computadores

Equalização de histogramas

- Ainda que operadores pontuais permitam controlar o ganho e brilho de uma imagem, como fazer para determinar os valores ideais?
  - uma alternativa seria mapear os valores mais escuros e mais brilhantes em preto e branco
  - ou encontrar o valor médio da imagem e alterá-lo para outro valor médio de cinzento e expandir todos os outros valores de forma a ocuparem toda a gama
- se obtivermos os histogramas da imagem podemos identificar os valores min, med e max de intensidade e as suas probabilidades de ocorrência
- a equalização de histogramas permite ocupar toda a gama dinâmica de intensidades e assim aumentar o contraste
  - operação para que o histograma se torne uniforme
  - re-distribuição uniforme dos níveis de intensidade

Equalização de histogramas (2)

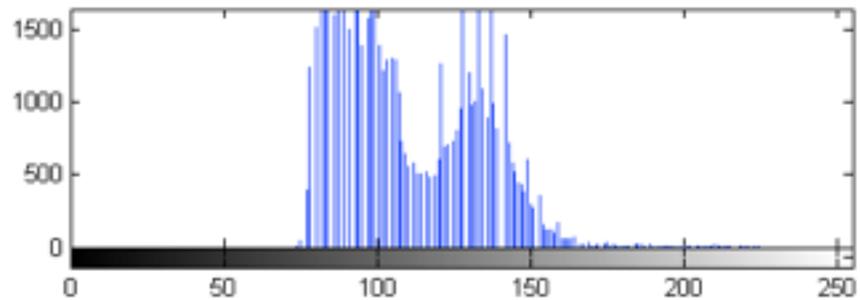


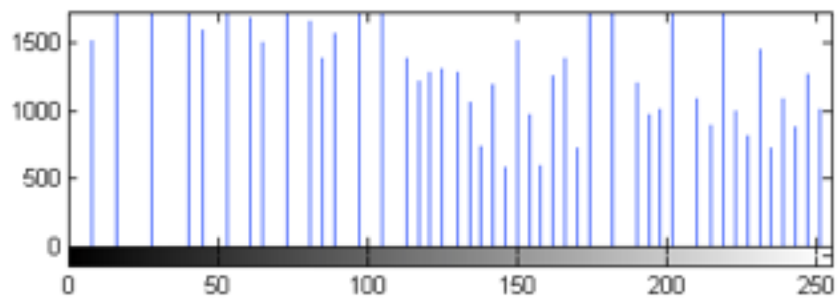


- no Matlab:

  histeq(I)

```
I = imread('image.tif');
J = histeq(I);
subplot(2,2,1);
imshow( I );
subplot(2,2,2);
imhist(I)
subplot(2,2,3);
imshow( J );
subplot(2,2,4);
imhist(J)
```





Operações espaciais

- muitas técnicas de processamento de imagem baseiam-se em operações espaciais realizadas sobre uma vizinhança local de pixels
  - essa operação é realizada como uma convolução da imagem com um filtro de resposta impulsional finita
    - máscara espacial ou kernel
  - ex., média espacial e filtragem passa-baixo
    - cada pixel é substituído pela média pesada dos pixels vizinhos
  - Suavização direccional ("directional smoothing")
    - para proteger os contornos ou fronteiras de objectos quando é feita a suavização da imagem
  - filtro de mediana
    - cada pixel é substituído pelo valor mediano do conjunto dos pixels vizinhos
      - útil para remover linhas ou pontos isolados, sem alterar a resolução espacial
  - realce de contornos e filtro passa-alto

Filtragem - convolução

▸ Filtros espaciais

- Filtrar uma imagem, substituíndo cada pixel original por uma soma pesada de pixels vizinhos
  - ▸ o filtro é definido como uma máscara de convolução
    - designado também de *kernel*
    - geralmente tem um número ímpar de colunas e linhas

100	100	50	50	100
100	100	100	100	100
100	100	100	100	100
100	100	100	100	100

X

0	1	0
0	0	0
0	0	0

=

100	100	50	50	100
100	100	50	100	100
100	100	100	100	100
100	100	100	100	100

- o filtro lê sucessivamente da esquerda para a direita todos os pixels na sua área de acção; multiplica o valor de cada um pelo correspondente valor do kernel e soma o resultado:
(100*0)+(50*1)+(50*0)*(100*0)+(100*0)
+(100*0)+(100*0)+(100*0)+(100*0)+(100*0) = 50
- o pixel inicial com valor 100 passa a assumir o valor 50

Departamento de Engenharia Electrotécnica e de Computadores

FEUP

Filtros com kernels - convolução

- Filtro de média

$$\begin{matrix} 20 & 12 & 14 & 23 \\ 45 & 15 & 19 & 33 \\ 55 & 34 & 81 & 22 \\ 8 & 64 & 49 & 95 \end{matrix} \quad \cdots\cdots \quad \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \rightarrow \begin{matrix} 32.8 & 28.1 & \dots \\ 41.1 & 45.8 & \dots \\ \dots & \dots & \dots \end{matrix}$$

- Filtro passa-alto

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

- Filtro para detecção de contornos

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

FEUP

Departamento de Engenharia Electrotécnica e de Computadores

Filtros com kernels - convolução (3)

- Detecção de contornos com filtro Sobel

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

  - Detecção de contornos com filtro Prewitt

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

- Detecção de contornos com filtro Roberts

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

- Estes filtros ou máscaras, detectam o gradiente da imagem, isto é, a variação de intensidade luminosa
  - se a variação é grande, acima de um determinado valor limiar, determina-se que existe um contorno nesse ponto

FEUP

Departamento de Engenharia Electrotécnica e de Computadores

Filtros com kernels - convolução (4)

- Estes filtros calculam as derivadas do sinal imagem
  - primeira derivada ou gradiente
    - operador de Sobel e de Prewitt
    - o valor da primeira derivada dá indicação do declive ou grau de variação, logo da existência (provável) de um bordo ou fronteira
  - no caso das segundas derivadas essa indicação é dada pela passagem por zero

Derivadas ou gradiente da imagem

- imagem → função bi-dimensional f(x,y)

$$f(x,y)$$

$$\nabla f(x,y) = \begin{bmatrix} \dfrac{\partial f(x,y)}{\partial x} \\ \dfrac{\partial f(x,y)}{\partial y} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$$

$$|\nabla f(x,y)| = \sqrt{f_x^2 + f_y^2}$$

$$\theta = \tan^{-1} \frac{f_x}{f_y}$$

Derivadas ou gradiente da imagem (2)

- exemplo de duas máscaras derivativas aplicadas a uma zona de uma imagem

$$Hx = 1/3\begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} \quad Hy = 1/3\begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & +1 \end{bmatrix} \quad I = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

$$Gx = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad Gy = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

detectados contornos / linhas verticais (variações na horizontal)

não há contornos / linhas horizontais (na vertical não há variações)

Detector de contornos de Canny

- é conhecido como sendo um dos mais eficientes detectores de contornos
  - funciona em vário passos
    - filtra a imagem para eliminar ruído, por ex., com filtros gaussianos 5x5

$$\frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \qquad \frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

    - calcula o gradiente da imagem (tal como Sobel) aplicando máscaras nas duas direções (x e y):

$$Hx = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \qquad Hy = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

Detector de contornos de Canny (2)

- obtém a amplitude e direcção do gradiente com as expressões:

$$G = \sqrt{(G_x^2 + G_y^2)} \qquad \theta = \text{arctang}\,(G_y + G_x)$$

- elimina pixels de forma a que os contornos sejam linhas finas
- ciclo de *histeresis* com 2 limiares de detecção (*thresholds*):
  - se pixel acima de th_{max} ou abaixo de th_{min}, é considerado contorno
  - se entre os dois valores, só é contorno se estiver "ligado" a um pixel com valor acima de th_{max}

Departamento de Engenharia Electrotécnica e de Computadores

FEUP

Alguns conceitos sobre a imagem e processamento digital

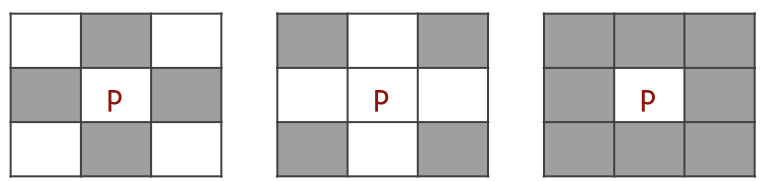
- Uma imagem digital f(x,y) corresponde a um sinal visual que foi *discretizado* no espaço (amostragem) e em amplitude (quantização)
    - vista como uma matriz bidimensional de valores numéricos que correspondem aos níveis de cinza da imagem em cada ponto (pixel)
- podem ser estabelecidas relações básicas entre pixels/valores no espaço bidimensional
    - vizinhança
    - adjacência
    - conectividade / ligação
    - percurso
    - regiões e fronteiras

Alguns conceitos sobre a imagem e processamento digital (2)

- conceito de vizinhanças
  - um pixel p, de coordenadas (x,y), tem 4 vizinhos horizontais e verticais, cujas coordenadas são (x+1, y), (x-1, y), (x, y+1) e (x, y-1)
    - esses pixels formam a chamada "*vizinhança-4*" de p, N4(p)
  - os quatro vizinhos diagonais de p são os pixels de coordenadas (x-1, y-1), (x-1, y+1), (x+1, y-1) e (x+1, y+1)
    - “*vizinhança-d*”, Nd(p)
  - A "*vizinhança-8*" de p é definida como N8(p) = N4(p) ∪ Nd(p)
- <u>um vizinho do pixel p é um pixel que está a uma unidade de distância de p</u>

Conceitos sobre processamento de imagem digital - vizinhança

- o conceito de vizinhança, permite estabelecer critérios para determinar de uma forma automática se dois pixels são vizinhos
  - os três critérios que normalmente são utilizados são
    - *vizinhança-4*, *vizinhança-d* e *vizinhança-8*



- a cinza estão assinalados os vizinhos do pixel P de acordo com os 3 critérios

Conceitos sobre processamento de imagem digital - conectividade

- Conectividade de pixels
  - conceito importante utilizado em conjunto com o conceito de vizinhança
    - para permitir identificar de forma automática os limites de objetos e componentes de regiões numa imagem
    - ao conceito e critérios de *vizinhança/proximidade*, junta o conceito e critérios de *semelhança de valor*
  - dois pixels dizem-se "ligados" se forem vizinhos de acordo com um determinado critério de vizinhança e se os seus níveis de cinzento satisfazem um determinado critério de semelhança
    - por exemplo, numa imagem binária, onde os pixels podem assumir os valores 0 e 1 (preto e branco), dois pixels podem ser "vizinhos-4" mas apenas serão "ligados-4" se tiverem o mesmo valor

Conceitos sobre processamento de imagem digital - conectividade

- Seja V o conjunto de valores de tons de cinza utilizados para se definir a conectividade
  - "conectividade-4": dois pixels p e q com valores de tom de cinza contidos em V, são "ligados-4" se q ∈ N4(p)
  - "conectividade-8": dois pixels p e q com valores de tom de cinza contidos em V, são "ligados-8" se q ∈ N8(p)
  - "conectividade-m" ("conectividade mista"): dois pixels p e q com valores de tom de cinza contidos em V, são "ligados-m" se:

    (i) q ∈ N4(p) ∧ p e q têm valores contidos em V (i. e., são ligados-4)

    (ii) q ∈ Nd(p) ∧ N4(p) ∩ N4(q) = ⦰ (tendo em conta a condição dos vizinhos-4 comuns de p e q terem que ter valores contidos em V)

FEUP
Departamento de Engenharia Electrotécnica e de Computadores

Conectividade mista

- conectividade mista
  - permite resolver o problema da existência de percursos múltiplos quando se usa a conectividade-8

1	1	0
0	1	0
1	0	0

usando conectividade-4

1	1	0
0	1	0
1	0	0

conectividade-8

1	1	0
0	1	0
1	0	0

conectividade-m

<u>usando conectividade-m:</u>

p	q	0
0	r	0
s	0	0

q é "ligado-m" a p porque $q \in N4(p) \wedge v_p$ e $v_q \in V$

r **não** é "ligado-m" a p porque $q \in Nd(p) \wedge N4(p) \cap N4(q) \neq \varnothing$

r é "ligado-m" a q porque $r \in N4(q) \wedge v_q$ e $v_r \in V$

s é "ligado-m" a r porque $s \in Nd(q) \wedge N4(r) \cap N4(s) = \varnothing$

Conceitos sobre a imagem digital - adjacência e percurso

- adjacência
  - Um pixel p é adjacente a um pixel q se p e q forem ligados
  - dois conjuntos, S1 e S2, de pixels adjacentes de uma imagem são adjacentes entre si se pelo menos um pixel em S1 é adjacente a um pixel em S2
- percurso entre dois pixels *p* e *q*
  - conjunto de pixels entre *p* e *q* que são adjacentes entre si
    - um percurso (path) do pixel p de coordenadas (x,y) até ao pixel q de coordenadas (s,t) consiste numa sequência de pixels distintos de coordenadas $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$, onde:
      - $(x_0, y_0) = (x,y)$
      - $(x_n, y_n) = (s,t)$
      - (x_i, y_i) é adjacente a (x_{i-1}, y_{i-1}) com $1 \leq i \leq n$
      - n é o comprimento do percurso

Percurso e conjuntos ligados

- dentro de um conjunto de pixels *S* de uma imagem, dois pixels *p* and *q* dizem-se ligados em *S* se
  - existir um percurso entre *p* e *q*
  - os pixels desse percurso pertencerem todos a *S*
- para qualquer pixel de *S*, o sub-conjunto de pixels ao qual está ligado designa-se de "*componente ligado*" de *S*
  - se *S* tiver apenas um componente, então *S* é designado de "*conjunto ligado*" ou "*região*"

regiões e fronteiras

- então, uma região de uma imagem é um conjunto ligado de pixels
  - duas regiões são designadas de adjacentes se a sua união formar um conjunto ligado
    - e por isso uma nova região!
  - duas regiões, S1 e S2, são adjacentes entre si se pelos menos um pixel em S1 é adjacente a um pixel em S2
- se não são adjacentes, duas regiões são designadas de disjuntas
- para encontrar a fronteira de uma região S1 podemos atribuir o valor oposto/simétrico aos pixels da sua região disjunta S2
  - todos os pixels de S1 que forem adjacentes ao oposto/simétrico de S2, constituem a sua fronteira ou contorno

Alguns conceitos sobre a imagem digital - distância

- Dados os pixels p, q e z, de coordenadas (x,y), (s,t) e (u,v), define-se a função distância D, cujas propriedades são:

  $D(p,q) \geq 0$ nota: $D(p,q) = 0$ se e só se $p = q$

  $D(p,q) = D(q,p)$

  $D(p,z) \leq D(p,q) + D(q,z)$

- Distância Euclidiana
  - $D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$
  - pixels com distância euclidiana em relação ao pixel (x,y) menor ou igual a a r, correspondem aos pontos contidos num círculo de raio r centrado em (x,y)

Departamento de Engenharia Electrotécnica e de Computadores

Alguns conceitos sobre a imagem digital - distância

- Distância D_4 (*city-block*)
  - $D_4(p,q) = |x - s| + |y - t|$
  - pixels com uma distância D_4 em relação ao pixel (x,y) menor ou igual a um valor r formam um losango centrado em (x,y)
  - pixels com $D_4 = 1$ são os 4-vizinhos de (x,y)
- Distância D_8 (tabuleiro de xadrez)
  - $D_8(p,q) = \max(|x - s|, |y - t|)$
  - pixels com distância D_8 em relação a (x,y) menor ou igual a um valor r formam um quadrado centrado em (x,y)
  - pixels com $D_8 = 1$ são os 8-vizinhos de (x,y)

Referências

- Sites de onde foram recolhidas algumas imagens exemplo:
- http://www.cs.princeton.edu/courses/archive/spr11/cos426/notes/cos426_s11_lecture04_compositing.pdf
- outras fontes de informação
  - http://mesh.brown.edu/engn1610/szeliski/03-imageprocessing.pdf
  - http://www.mathworks.com/help/hdlcoder/examples/image-enhancement-by-histogram-equalization.html
  - https://www.mathworks.com/help/images/functionlist.html