CORRELAÇÂO DIGITAL

Revisão de matemática

CORRELAÇÂO

Dadas duas variáveis x e y, podemos calcular suas médias e os valores do desvio padrão, ou a variância, de cada uma delas.

Note que aqui temos a diferença ao quadrado.

$$s^2 = \frac{\sum(x - \bar{x})^2}{n - 1} \qquad s = \sqrt{\frac{\sum(x - \bar{x})^2}{n - 1}}$$

Também podemos calcular a covariância entre as variáveis.

$$\boldsymbol{COV(X,Y) = \frac{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right)}{n-1}}$$

Covariâncias

$$COV(X,Y) = \frac{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right)}{n-1}$$

No cálculo da covariância se calcula o produto das diferenças em relação à média (considerando o sinal!) da combinação de duas variáveis x e y.

O que ocorre quando ...

$\left(X_i - \bar{X}\right) > 0$ e $\left(Y_i - \bar{Y}\right) > 0$

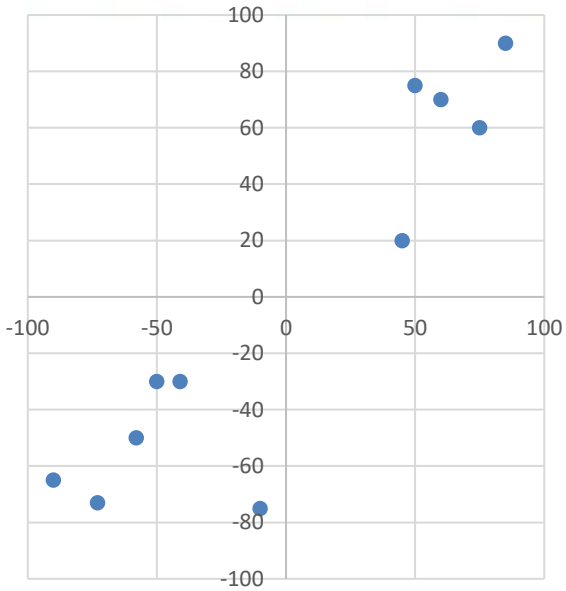
$\left(X_i - \bar{X}\right) < 0$ $\left(Y_i - \bar{Y}\right) < 0$

$\left(X_i - \bar{X}\right) > 0$ e $\left(Y_i - \bar{Y}\right) < 0$

$\left(X_i - \bar{X}\right) < 0$ $\left(Y_i - \bar{Y}\right) > 0$

Mesmo sinal vs. sinais trocados

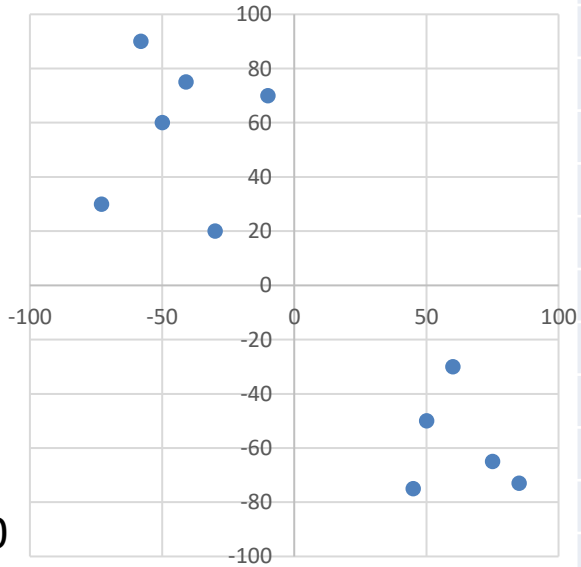
x	y
85	90
50	75
60	70
45	20
75	60
-73	-73
-58	-50
-41	-30
-10	15
-90	-65
-50	-75



Sempre positivo

$(-A)^* (-B)>0$

$A^*B>0$



x	y
85	-73
50	-50
60	-30
45	-75
75	-65
-73	15
-58	90
-41	75
-10	70
-90	20
-50	60

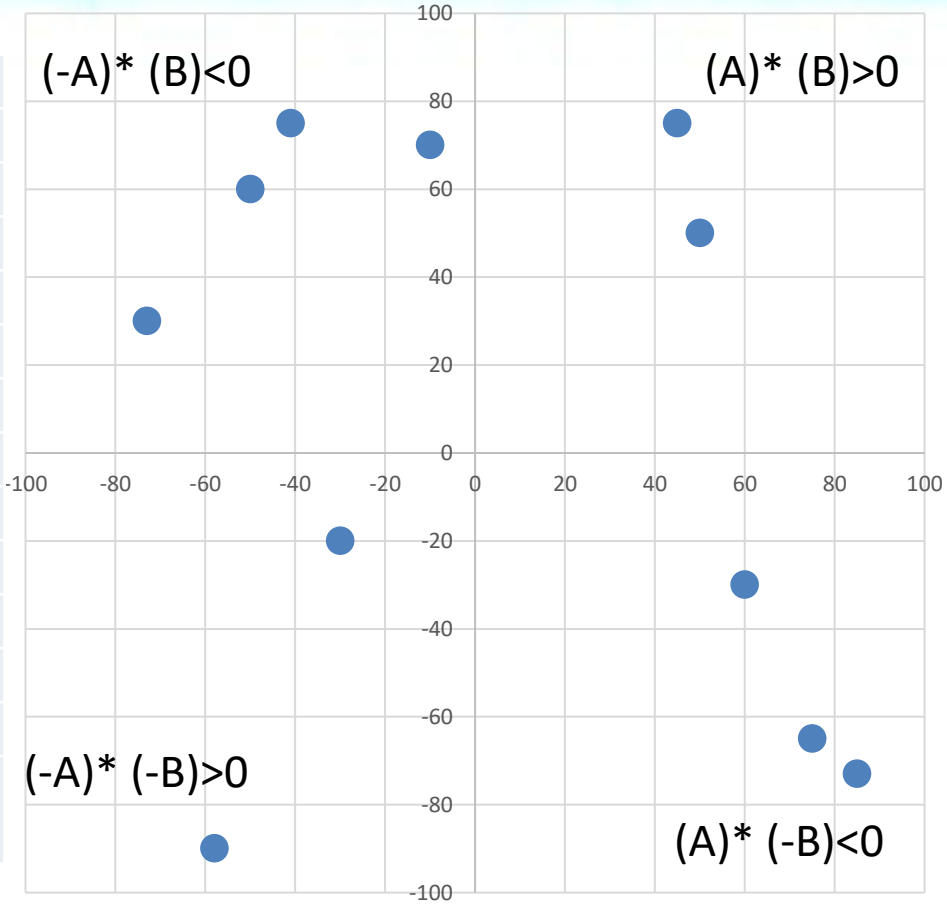
Sempre negativo

$(-A)^* B<0$

$A^*(-B)<0$

E se ocorre de tudo?

x	y	(x-mx)^2	(y=my)^2	(x-mx)*(y-my)
85	-73	6.429	6.472	-6.451,0
50	50	2.041	1.810	1.922,3
60	-30	3.045	1.402	-2.066,8
45	75	1.614	4.562	2.714,1
75	-65	4.925	5.249	-5.085,0
-73	30	6.055	0.508	-1.754,4
-58	-90	3.946	9.497	6.121,9
-41	75	2.099	4.562	-3.094,8
-10	70	0.219	3.911	-0.926,8
-30	-20	1.212	0.753	0.955,9
-50	60	3.005	2.761	-2.880,4
	Soma	34.594	41.493	-10.545
	Soma /n	3.144,9	3.772,1	-0.958,6



$$COV(X,Y) = \frac{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right)}{n-1}$$

- Qual seria a covariância entre X e X?
- ou seja, se X=Y ?

$$s^2 = \frac{\sum(x-\bar{x})^2}{n-1} \qquad s = \sqrt{\frac{\sum(x-\bar{x})^2}{n-1}}$$

Matriz variância -covariância

$$\Sigma = \begin{bmatrix} \mathrm{Var}(X) & \mathrm{Cov}(X,Y) & \mathrm{Cov}(X,Z) \\ \mathrm{Cov}(X,Y) & \mathrm{Var}(Y) & \mathrm{Cov}(Y,Z) \\ \mathrm{Cov}(X,Z) & \mathrm{Cov}(Y,Z) & \mathrm{Var}(Z) \end{bmatrix}$$

O que tem na diagonal principal?

Matriz simétrica e quadrada

Correlação

$$\operatorname{corr}(X,Y) = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y}$$

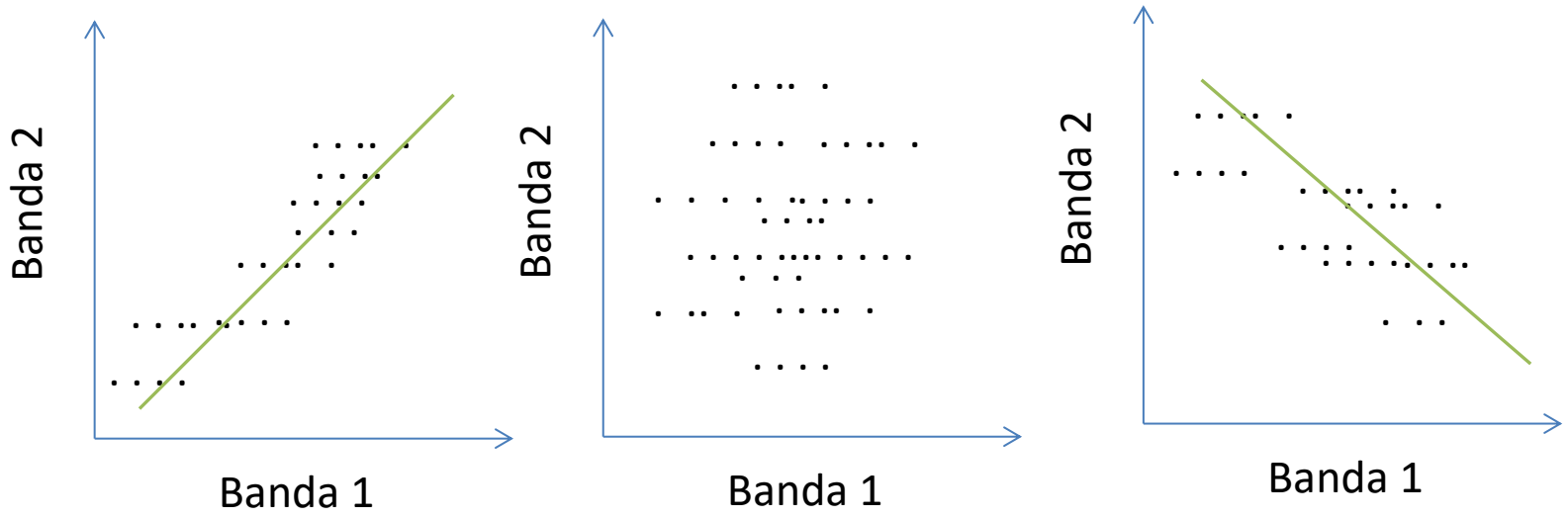
- Qual seria a correlação entre X e X?

- O que ocorre se cov(X,Y)>0?
- O que ocorre se cov(X,Y)=0?
- O que ocorre se cov(X,Y)<0?

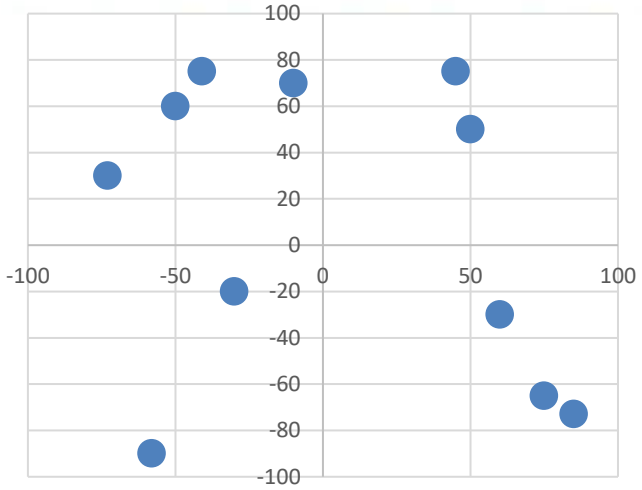
A correlação é um índice que descreve a dependência linear entre duas variáveis e serve como indicador do grau de similaridade entre este par de variáveis.

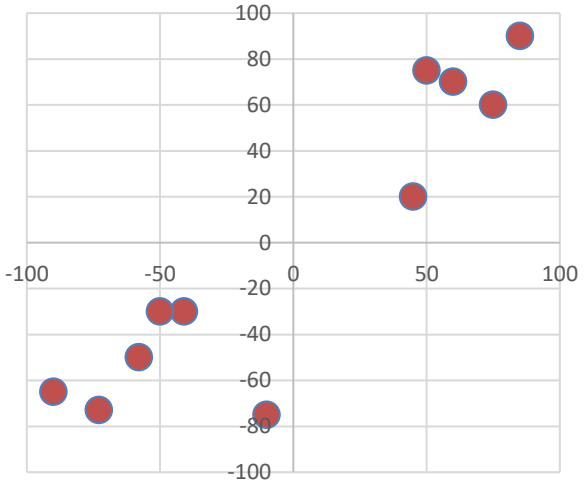
Correlação

Correlação positiva,negativa ou zero?



Compare a correlação





média de (x-mx)(y-my)=-958,64

Corr= -0,25

média de (x-mx)(y-my)=3504,90

Corr= 0,83

Correlação

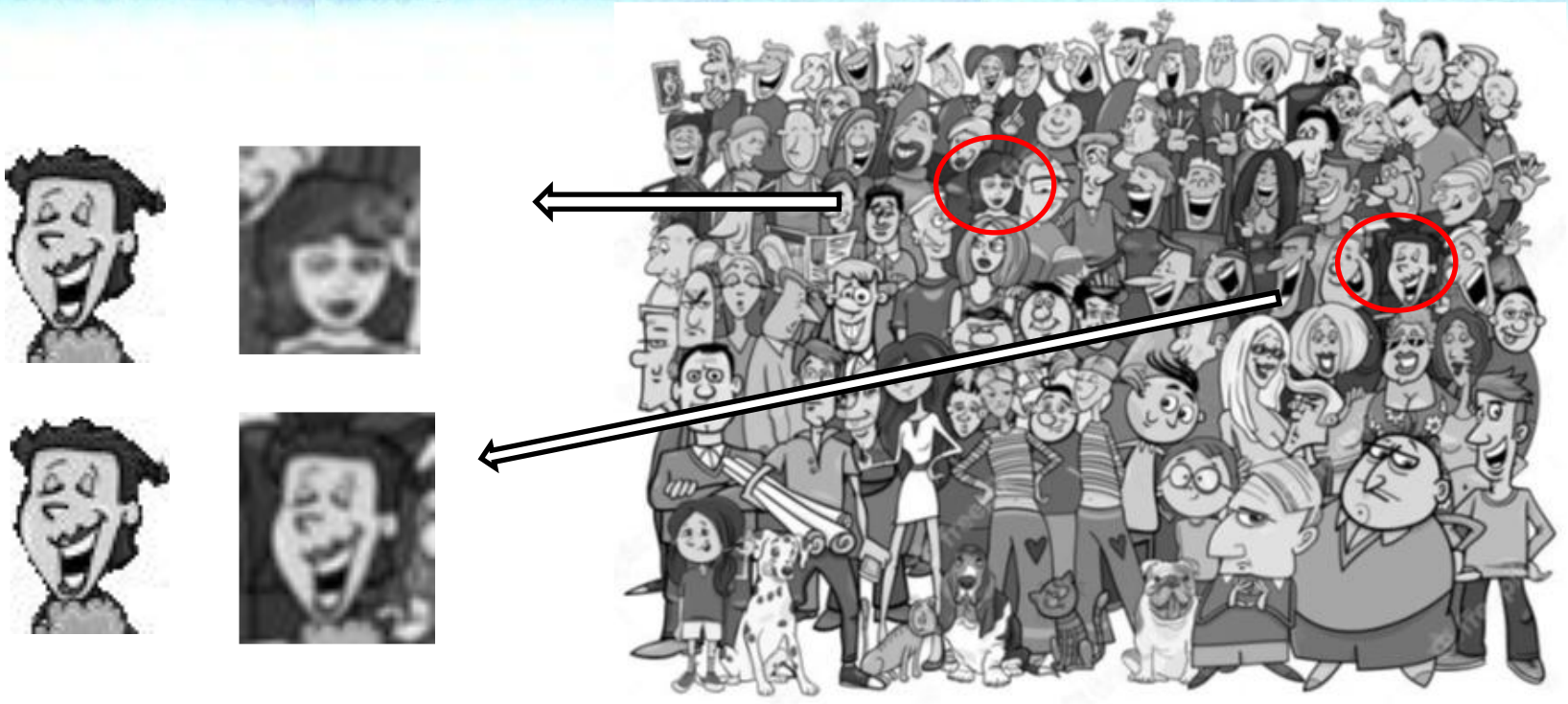
Detectar uma região de uma imagem (pode ser um ponto, uma borda) em outra imagem, por comparação entre as regiões.

Onde está esta pessoa?

Podemos calcular "descritores" da região ou simplesmente comparar diretamente a região procurada com todas a regiões da segunda imagem.

Como medir a similaridade entre as regiões?

Varrendo a imagem...



Duas regiões, do mesmo tamanho serão comparadas e devemos decidir quanto se parecem

diferença

Uma opção é calcular a diferença entre as imagens. Se a diferença for nula, as imagens são idênticas.

$$D(x,y) = \sum_{x_i y_i}(I(x_i, y_i) - J(x_i\ y_i)\ )$$

Porém, a diferença é sensível a ocorrência de valores negativos. Negativos somados a positivos podem gerar um resultado nulo.

I

J

Por isso, podemos usar a soma das diferenças ao quadrado

$$D(x,y) = \sum_{x_i y_i}[J(x_i, y_i) - I(x_i\ y_i)]^2$$

Ou a soma dos valores absolutos. Mas...

Se as imagens forem iguais, porém em diferentes condições de brilho ou contraste? Funcionaria?

correlação

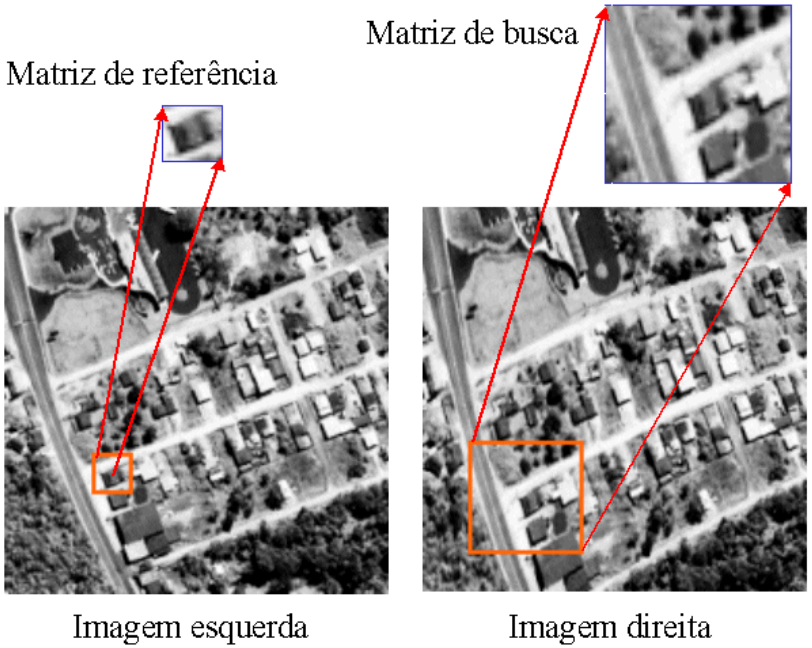
$$\rho = \frac{\sum_{i=1}^{n} x_i - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{\sqrt{\left(\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2\right)\left(\sum_{i=1}^{n} y_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} y_i\right)^2\right)}}$$

Correlação

Este princípio pode ser aplicado no processamento de imagens digitais para procurar determinados padrões em uma imagem ou pontos homólogos em pares estereoscópicos, comparando uma matriz de amostra com as diferentes regiões da a imagem. Os locais onde a matriz e a região da imagem forem similares serão caracterizados por um alto valor da correlação.

É preciso definir, na primeira imagem, a “janela de referência”, ou seja o padrão bidimensional a ser procurado na segunda. Para facilitar a atribuição do resultado da correlação a um pixel costuma-se adotar “janelas de referência” de dimensão impar.

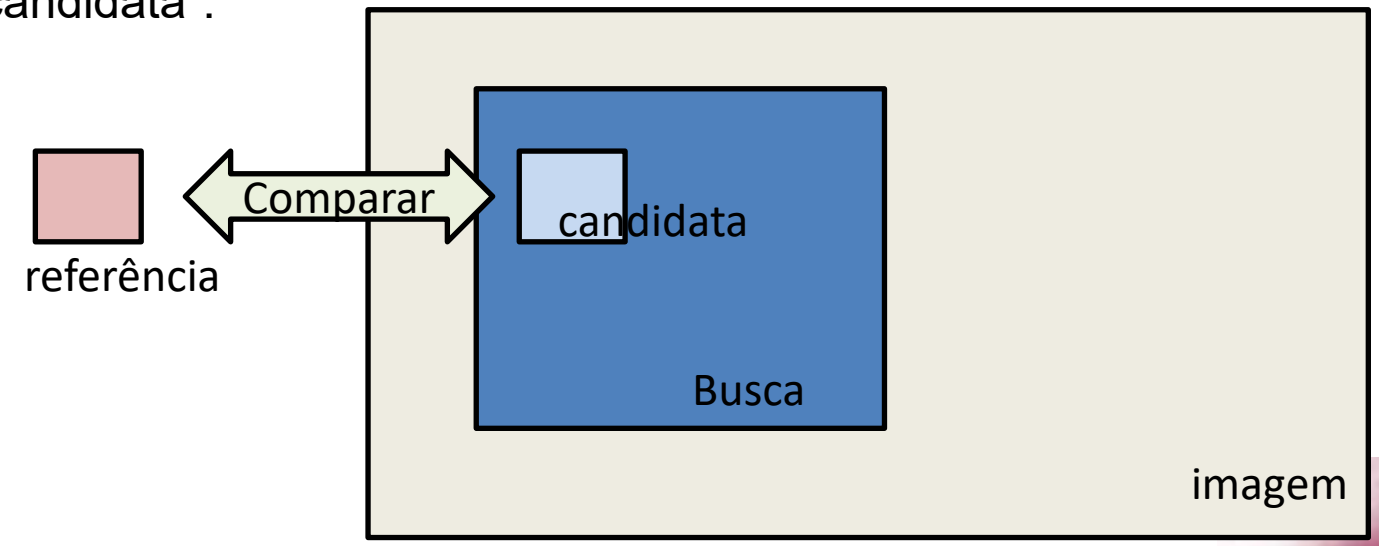
Quando é possível, e para evitar procurar em toda a segunda imagem, uma região onde se espera encontrar este padrão na segunda imagem é definida previamente, a “janela de busca”. A “janela de busca” deve ser maior que a “janela de referência”.



O algoritmo de correlação digital no domínio da imagem (espacial) consiste em deslocar a “janela de referência” ao longo da “janela de busca” e calcular o valor da correlação entre os níveis valores digitais das duas matrizes para cada posição.

A posição do ponto homólogo será caracterizada pelo maior valor da correlação.

A cada posição (i,j) apenas uma sub-região da “matriz de busca”, de dimensão igual à “matriz de referência” é considerada. Esta região é chamada de “matriz candidata”.



Em termos práticos

Ler imagem I (nxm)

Ler recorte J (n1 x m1) (referência)

Transforme J em vetor Y

Para cada ponto viável na imagem I(i,j)

transformar a região (n1xm1) em torno do pixel em vetor

Calcular a correlação entre X e Y

armazenar o valor em uma matriz

Localizar o maior máximo local da matriz de correlações

Problemas: a correlação é sensível a mudanças de rotação e de escala.

vantagem

A correlação absorve o efeito da diferença de brilho

Se q=x+c (c=uma constante de brilho)

Media:

$$COV(X,Y) = \frac{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right)}{n-1}$$

$m_q = m_x + c$

E $q - m_q = (x+c) - m_x + c = x - m_x$

E, ao dividir pelos valores do desvio padrão das variáveis, retira o efeito do contraste, pois normaliza a medida à faixa [-1,1]

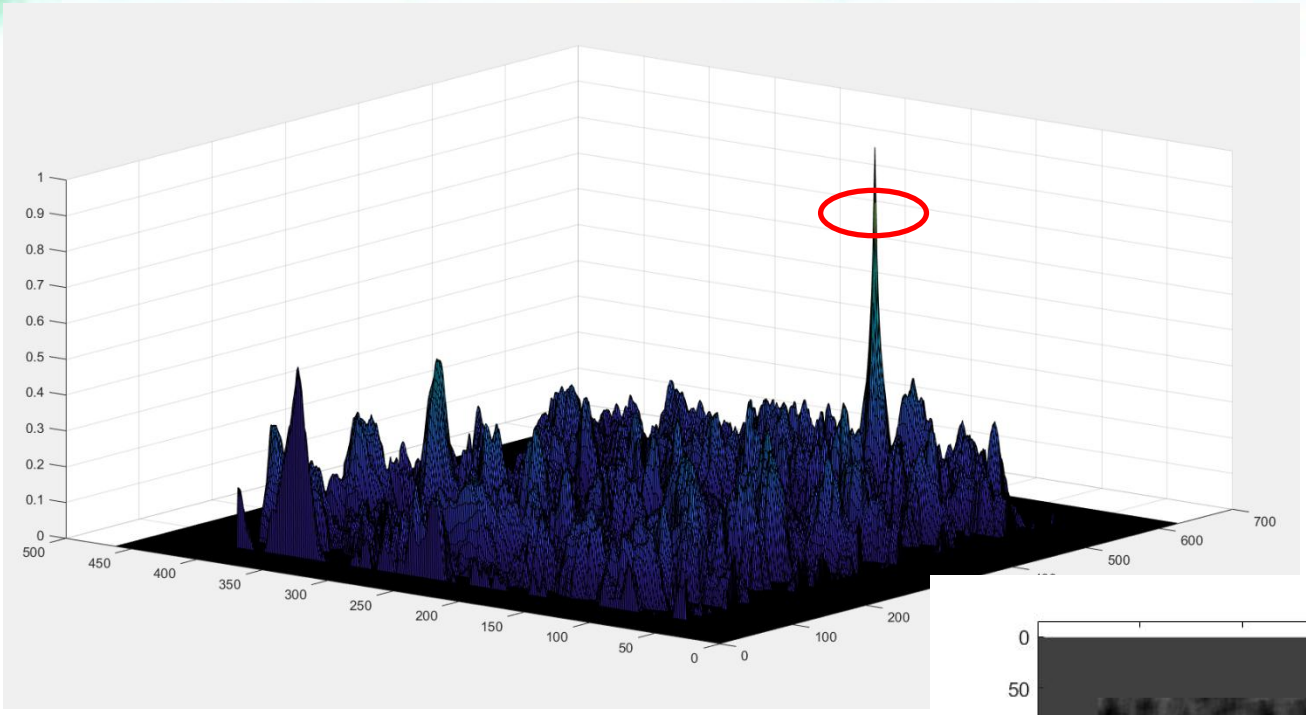
exemplo

imagem

Em que posição da imagem se encontra uma região igual à referência?

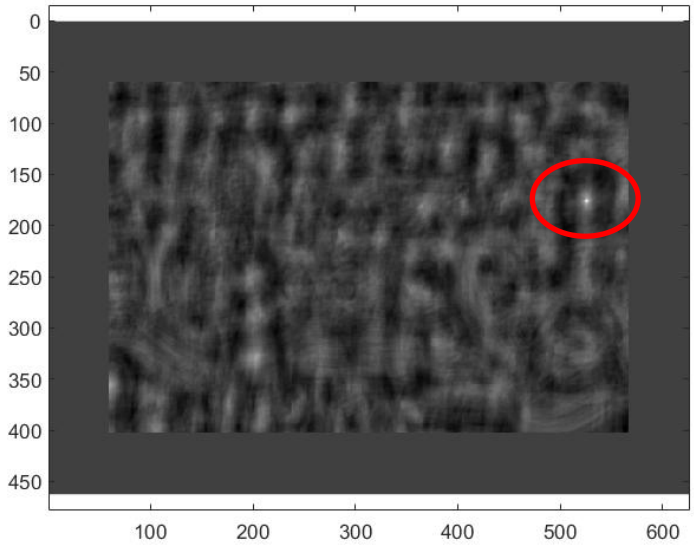
referência

Matriz de correlacoes



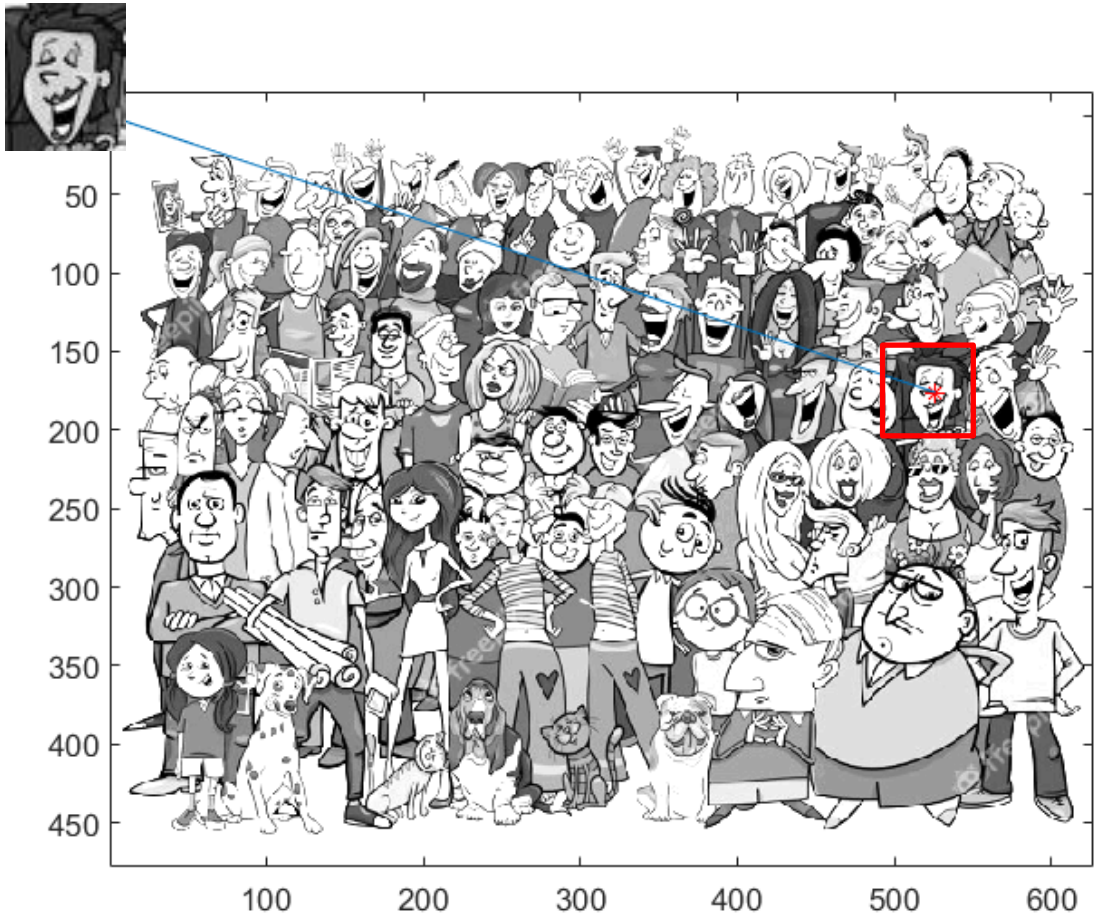
Valores da correlação para cada pixel (3D)

A melhor coincidência se localiza no pixel onde se encontra o máximo

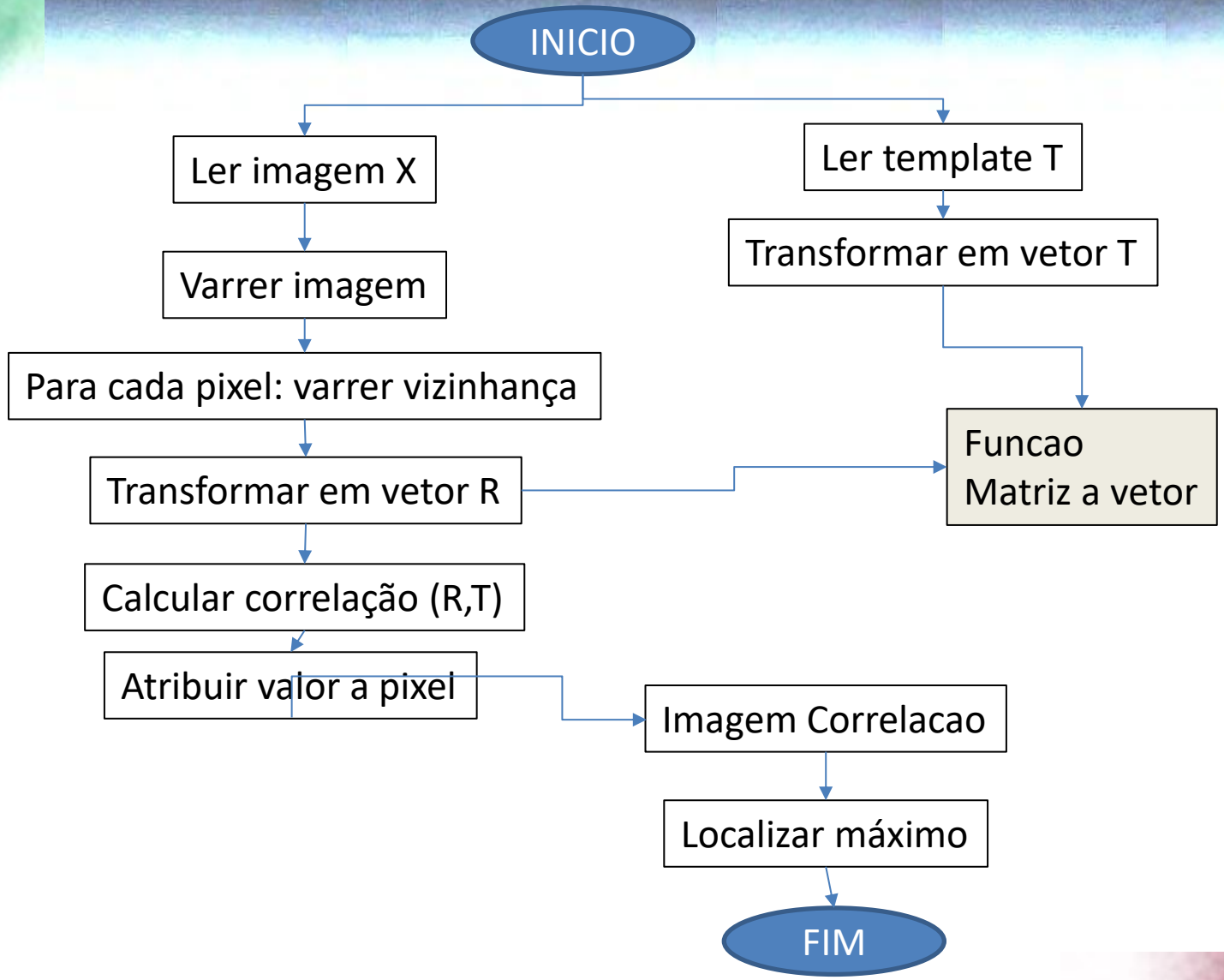


Localizar máximo

- Máximo local de maior valor (absoluto)



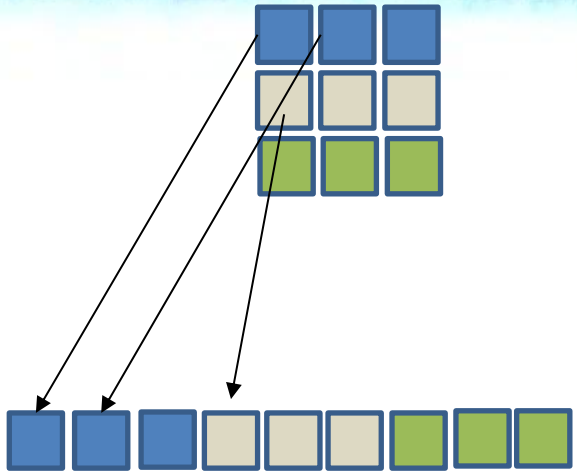
Um programa


INICIO
Ler imagem X
Varrer imagem
Para cada pixel: varrer vizinhança
Transformar em vetor R
Calcular correlação (R,T)
Atribuir valor a pixel
Ler template T
Transformar em vetor T
Funcao
Matriz a vetor
Imagem Correlacao
Localizar máximo
FIM

Encontrar esta região na imagem ao lado.

Algumas ferramentas importantes

RR=np.reshape(R, [1,nl*nc])
Rearranja uma matrix.
Ex: matriz imagem (2D, com nl, nc de tamanho) para vetor (1xnl*nc)

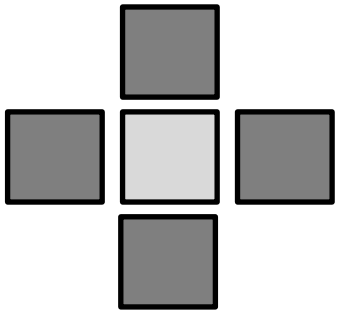


rm=np.corrcoef(X,Y) # calcula matriz de correlacao

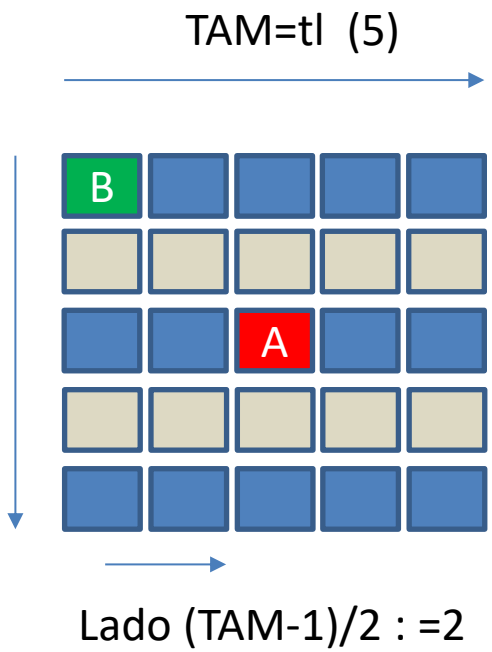
r=rm[0,1] # le elemento fora da diagonal (corr(X,Y))

Detecção de pico:

Se o central for maior que seus 4 vizinhos, é pico



Varredura



- A=central na posição L,C
- B=pixel no canto sup esq (p,q)
- p=L-lado
- q=C-lado

- Programa

básico

```
import matplotlib.pylab as plt
import numpy as np
import cv2
from google.colab.patches import cv2_imshow
from google.colab import drive
drive.mount('/content/drive')
```

Ler dados e verificar tamanhos

```
# Ler a imagem do drive
nome='/content/drive/My Drive/fotos/recorte.tif'
I = cv2.imread(nome) # ler imagem RGB deste endereço
nl,nc, nb= I.shape # Recupera o tamanho da imagem
print('tamanho', nl, nc, 'nro de bandas', nb)
X1 = cv2.cvtColor(I, cv2.COLOR_BGR2GRAY) # transforma imagem   cinza
cv2_imshow(X1)
# Ler o template do drive
nome='/content/drive/My Drive/fotos/template.tif'
I = cv2.imread(nome)
tl,tc, nb= I.shape # Recupera o tamanho do template
print('tamanho', tl, tc, 'nro de bandas', nb)
T1 = cv2.cvtColor(I, cv2.COLOR_BGR2GRAY) # transforma imagem cinza
cv2_imshow(T1)
```

formatação

```
X=np.array(X1, dtype=float) #transforma em float para facilitar multiplicacoes
T=np.array(T1, dtype=float)

Y=np.zeros((nl,nc),dtype = float) # imagem de saída para armazenar a correlacao em cada ponto
Y=np.uint8(Y)

dim=tl # dimensao da janela móvel/template (quadrada).
lado=(dim-1)/2 # numero de vizinhos antes ou depois do central
lado=uint(lado)

TT=np.reshape(T, [1,tl*tc]) # transforma matriz template a vetor
print(TT)
print('dim=', dim, 'lado=', lado)
```

correlação

```
for L in range(lado,nl-lado):        # varrer em linhas lado+1, para vizinhos+central
   for C in range (lado, nc-lado):   # varrer em colunas
      p=(L-lado)          # determina canto superior/esquerdo do recorte
      q=(C-lado)
      R=X[p:p+tl, q:q+tc]                    # copia a regiao de interesse
      RR=np.reshape(R, [1,tl*tc])            # transforma regiao a vetor
      rm=np.corrcoef(RR,TT)                  # matriz de correlação (Regiao/Template)
      r=rm[0,1]           # correlacao
      if r<0:             # descarta correlacoes negativas
         r=0
      rp=r*255            # muda para 8  bits para salvar imagem
      v=np.uint8( np.round( rp ) )   # arredonda e muda a uint8
      Y[L,C]=v                  # salva correlação na posicao do central

cv2_imshow(Y)
```

Busca maior correlação na imagem

```
Limiar=200                  # valor mínimo de correlação aceitável (200/255=0.78)
maximo=0                    # inicializar o maximo
for L in range(lado,nl-lado):           # varrer em linhas
   for C in range (lado, nc-lado):      # varrer em colunas
      if Y[L,C]>Limiar:                 #pixel supera limiar, vejamos se é méximo local
    # se é maximo local, deve ser maior que os quatro vizinhos
        if Y[L,C]>Y[L+1,C] and Y[L,C]>Y[L-1,C] and Y[L,C]>Y[L,C+1] and Y[L,C]>Y[L,C-1]:
           if Y[L,C]>maximo:     # é maior que o anterior máximo registrado
              maxL=L          # coordenadas do pico
              maxC=C
              maximo=Y[L,C]     # valor do máximo atual
print('localizado em: ', maxL, maxC, '| ',maximo)

# visualizar a imagem com o ponto localizado
plt.imshow(Y,cmap='gray')
plt.plot(maxC, maxL, 'r*')
plt.show()
```

aplicação