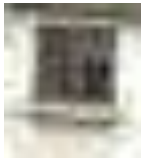
CORRELAÇÂO DIGITAL

Correlação

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

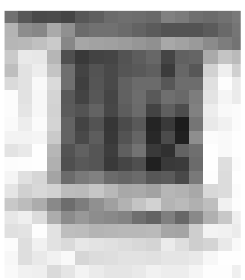
Ex: Onde fica esta janela na imagem?

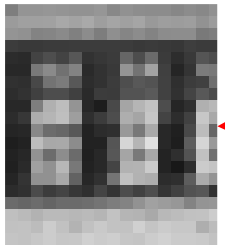


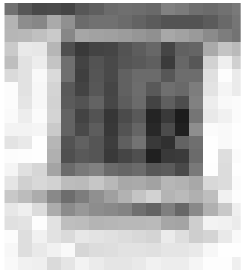
Podemos comparar a região procurada com todas as regiões da segunda imagem e avaliar a semelhança.

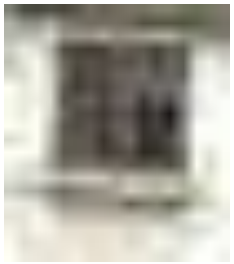
Como medir a similaridade entre as regiões?

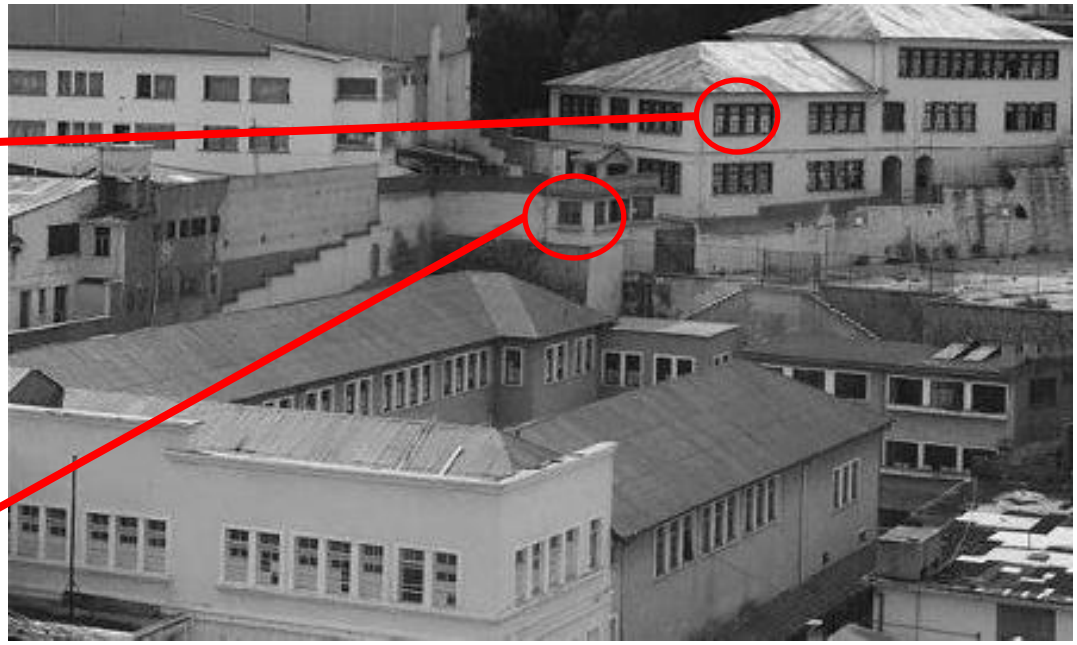
Varrendo a imagem...











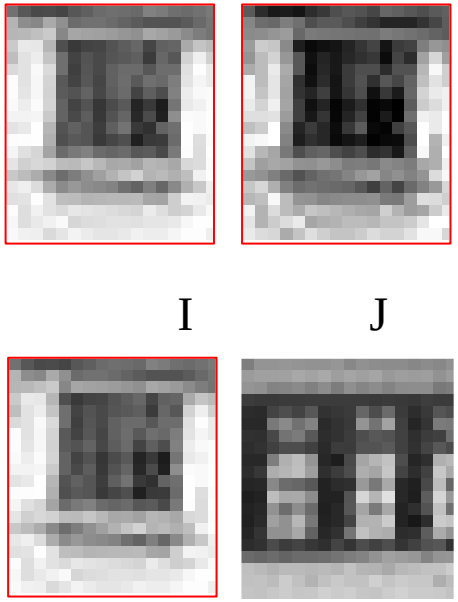
Duas regiões, do mesmo tamanho serão comparadas e devemos decidir quanto se parecem

diferença

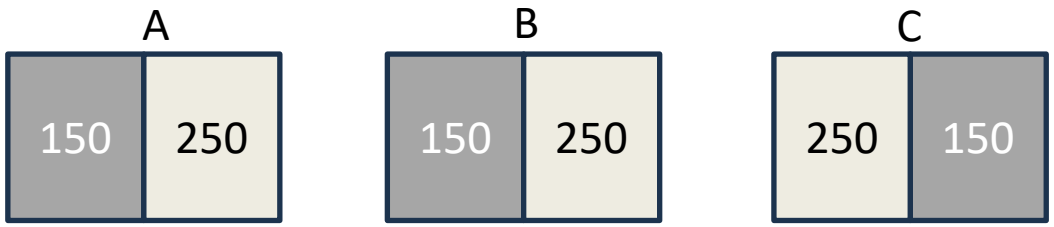
Uma opção é calcular a diferença entre as regiões.

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.



A-B= 0

A-C = (150-250)+(250-150)= -100 + 100 =0

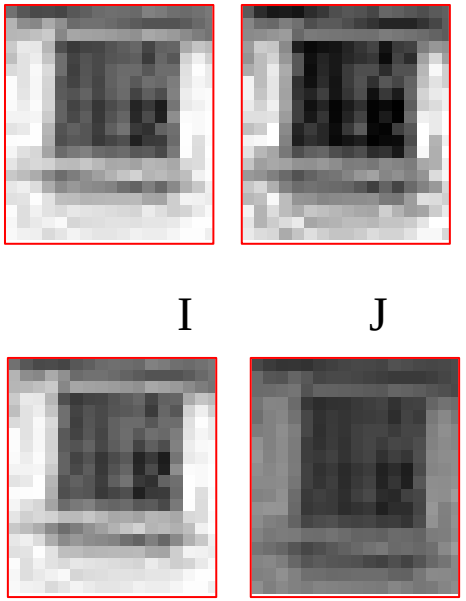
diferença

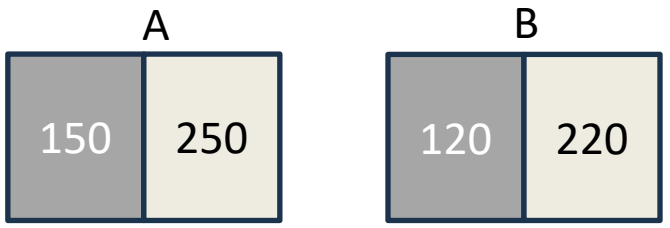
Por isso, podemos usar a soma das diferenças ao quadrado

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

Ou a soma dos valores absolutos. Mas...

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





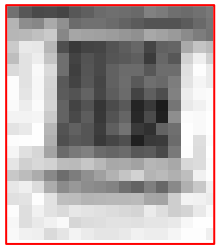
A= B+30

A-B=(150-120)+(250-220)= 30+30=60, diferente de zero

Problema

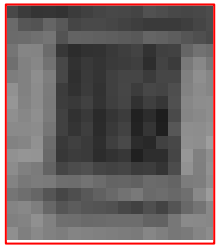
- Dadas duas regiões, do mesmo tamanho, como avaliar se são parecidas.

X



6	8	0	1	0
0	9	12	0	0
2	10	10	1	1
0	11	13	2	0

Y



5	6	0	1	0
0	8	11	0	1
2	10	10	1	1
1	10	10	2	0

x=[6, 8, 0, 1, 0, 0, 9, 12, 0, 0, 2, 10, 10, 1, 1, 0, 11, 13, 2, 0]
y=[5, 6, 0, 1, 0, 0, 8, 11, 0, 1, 2, 10, 10, 1, 1, 1, 10, 10, 2, 0]

As imagens são parecidas se
Dada a posição (linha, coluna)
Ocorrem valores baixos em X e Y
X(i)=baixo, Y(i)=baixo
Ocorrem valores altos em X e Y
X(i)=alto, Y(i)=alto
Os termos “alto” e “baixo” podem ser relativos dentro do contexto de cada imagem

“alto”=“acima da média”
“baixo”=“abaixo da média”

Problema

SE

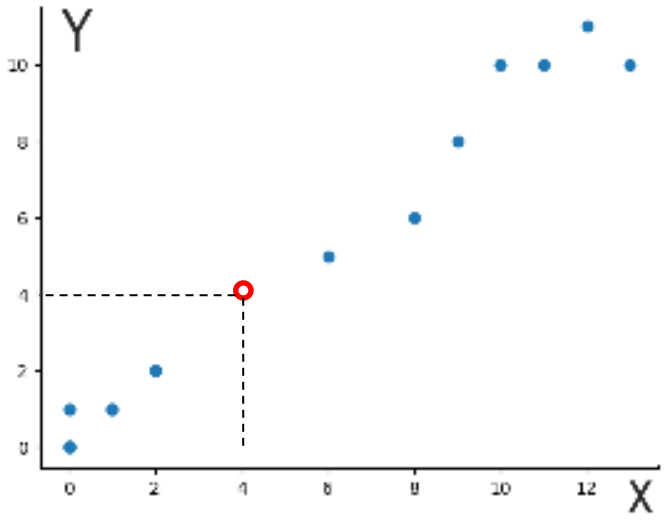
- Ocorre um valor “acima da média” e X e “acima da média” em Y
- Ocorre um valor “abaixo da média” e X e “abaixo da média” em Y
- Existe concordância, similaridade

X

6	8	0	1	0
0	9	12	0	0
2	10	10	1	1
0	11	13	2	0

Y

5	6	0	1	0
0	8	11	0	1
2	10	10	1	1
1	10	10	2	0



Isto pode ser medido com a correlação!

Correlação

Dadas duas variáveis x e y,
os valores da variância e do desvio padrão, são:

$$var_x = \frac{\sum_{i=1}^{n}(x_i - m_x)^2}{n-1}$$

$$dp_x = \sqrt{var_x}$$

A covariância entre as variáveis é.

$$cov(x,y) = \frac{\sum_{i=1}^{n}(x_i - m_x)(y_i - m_y)}{n-1}$$

A matriz variância-Covariância:

$$MVC = \begin{bmatrix} var_x & cov_{xy} \\ cov_{xy} & var_y \end{bmatrix}$$

Correlação

$$corr(x,y) = \frac{cov_{xy}}{dp_x \quad dp_y}$$

- Substituindo:

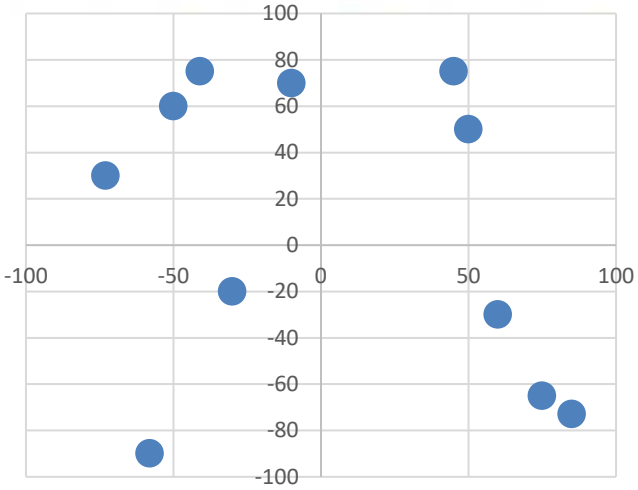
$$corr(x,y) = \frac{\frac{\sum_{i=1}^{n}(x_i - m_x)(y_i - m_y)}{n-1}}{\sqrt{var_x} \quad \sqrt{var_y}}$$

Ou

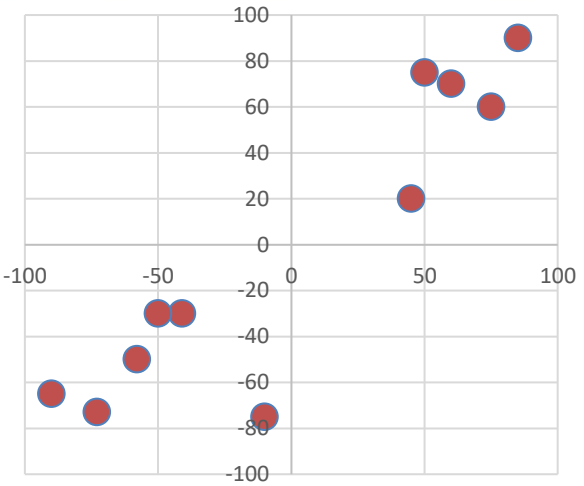
$$corr(x,y) = \frac{\frac{\sum_{i=1}^{n}(x_i - m_x)(y_i - m_y)}{n-1}}{\sqrt{\frac{\sum_{i=1}^{n}(x_i - m_x)^2}{n-1}} \quad \sqrt{\frac{\sum_{i=1}^{n}(y_i - m_y)^2}{n-1}}}$$

$$corr(x,y) = \frac{\sum_{i=1}^{n}(x_i - m_x)(y_i - m_y)}{\sqrt{\sum_{i=1}^{n}(x_i - m_x)^2}\sqrt{\sum_{i=1}^{n}(y_i - m_y)^2}}$$

Compare a correlação



Corr= -0,25



Corr= 0,83

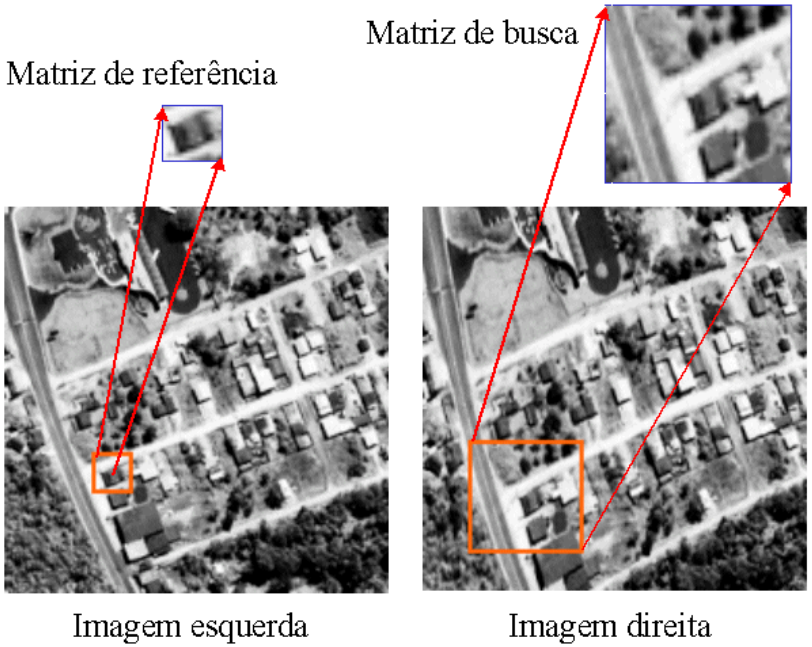
A correlação é um índice que descreve a dependência linear entre duas variáveis e serve como indicador do grau de similaridade entre este par de variáveis.

Correlação

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 ímpar.

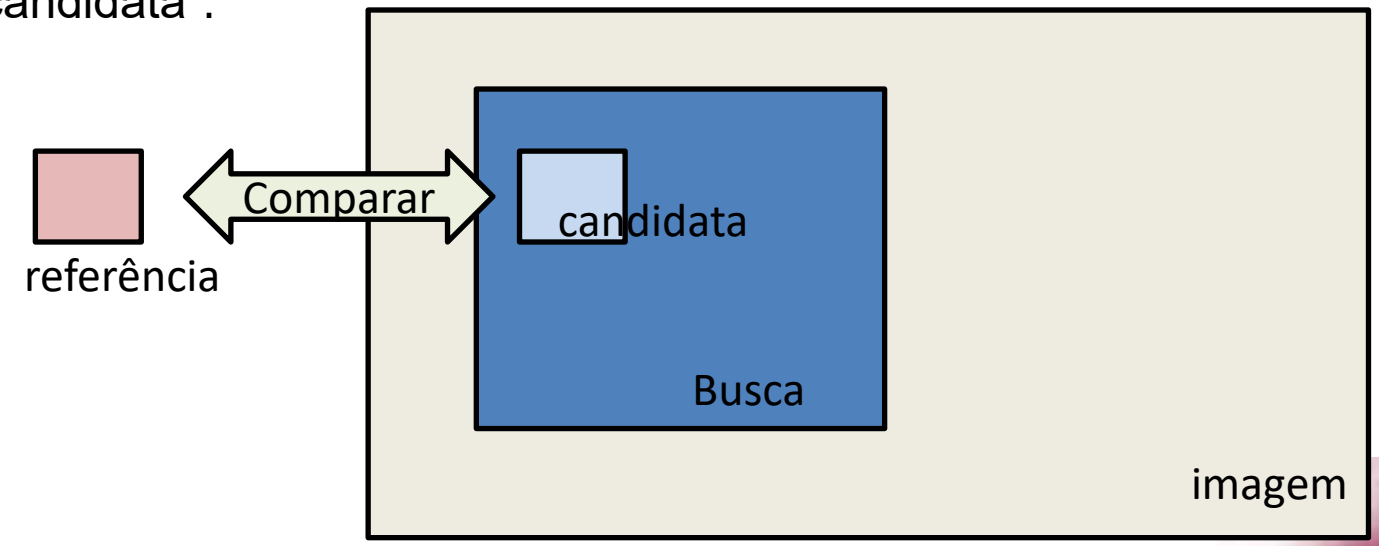
Quando é possível, e para evitar procurar em toda a segunda imagem, uma região onde se espera encontrar este padrão na segunda imagem é definida previamente, a “janela de busca”. A “janela de busca” deve ser maior que a “janela de referência”.



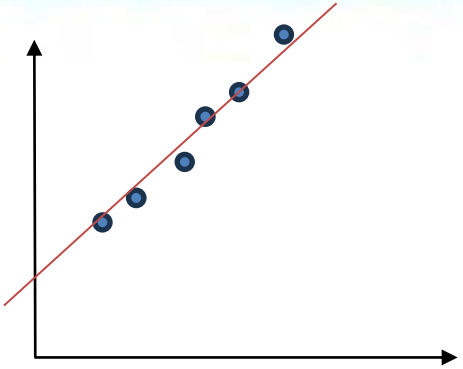
O algoritmo de correlação digital no domínio da imagem (espacial) consiste basicamente em deslocar a “janela de referência” ao longo da “janela de busca” e calcular o valor da correlação entre os níveis valores digitais das duas matrizes para cada posição.

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

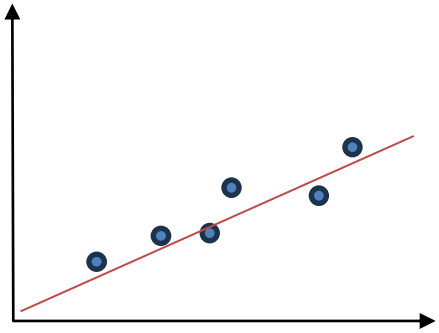
A cada posição (i,j) apenas uma sub-região da “matriz de busca”, de dimensão igual à “matriz de referência” é considerada. Esta região é chamada de “matriz candidata”.



- A correlação absorve diferenças de brilho e contraste



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



vantagem

A correlação absorve o efeito da diferença de brilho

Se z=a*x (c=uma constante multiplicadora) $m_z = a*m_x$

$$corr(z,y) = \frac{\sum_{i=1}^{n}(z_i - m_x)(y_i - m_y)}{\sqrt{\sum_{i=1}^{n}(z_i - m_z)^2} \quad \sqrt{\sum_{i=1}^{n}(y_i - m_y)^2}}$$

$$corr(z,y) = \frac{\sum_{i=1}^{n}(ax_i - am_x))(y_i - m_y)}{\sqrt{\sum_{i=1}^{n}(ax_i - am_x)^2} \quad \sqrt{\sum_{i=1}^{n}(y_i - m_y)^2}}$$

$$corr(z,y) = \frac{a\sum_{i=1}^{n}(x_i - m_x)(y_i - m_y)}{a\sqrt{\sum_{i=1}^{n}(x_i - m_x)^2} \quad \sqrt{\sum_{i=1}^{n}(y_i - m_y)^2}}$$

$$corr(z,y) = corr(x,y)$$

vantagem

A correlação absorve o efeito da diferença de contraste

Se z=x+c (c=uma constante de brilho) $m_z = m_x + c$

$$corr(z,y) = \frac{\sum_{i=1}^{n}(z_i - m_x)(y_i - m_y)}{\sqrt{\sum_{i=1}^{n}(z_i - m_z)^2} \quad \sqrt{\sum_{i=1}^{n}(y_i - m_y)^2}}$$

$$corr(z,y) = \frac{\sum_{i=1}^{n}((x_i+c) - (m_x+c))(y_i - m_y)}{\sqrt{\sum_{i=1}^{n}((x_i+c) - (m_x + c))^2} \quad \sqrt{\sum_{i=1}^{n}(y_i - m_y)^2}}$$

$$corr(z,y) = \frac{\sum_{i=1}^{n}(x_i - m_x)(y_i - m_y)}{\sqrt{\sum_{i=1}^{n}((x_i - m_x)^2} \quad \sqrt{\sum_{i=1}^{n}(y_i - m_y)^2}}$$

$$corr(z,y) = corr(x,y)$$

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

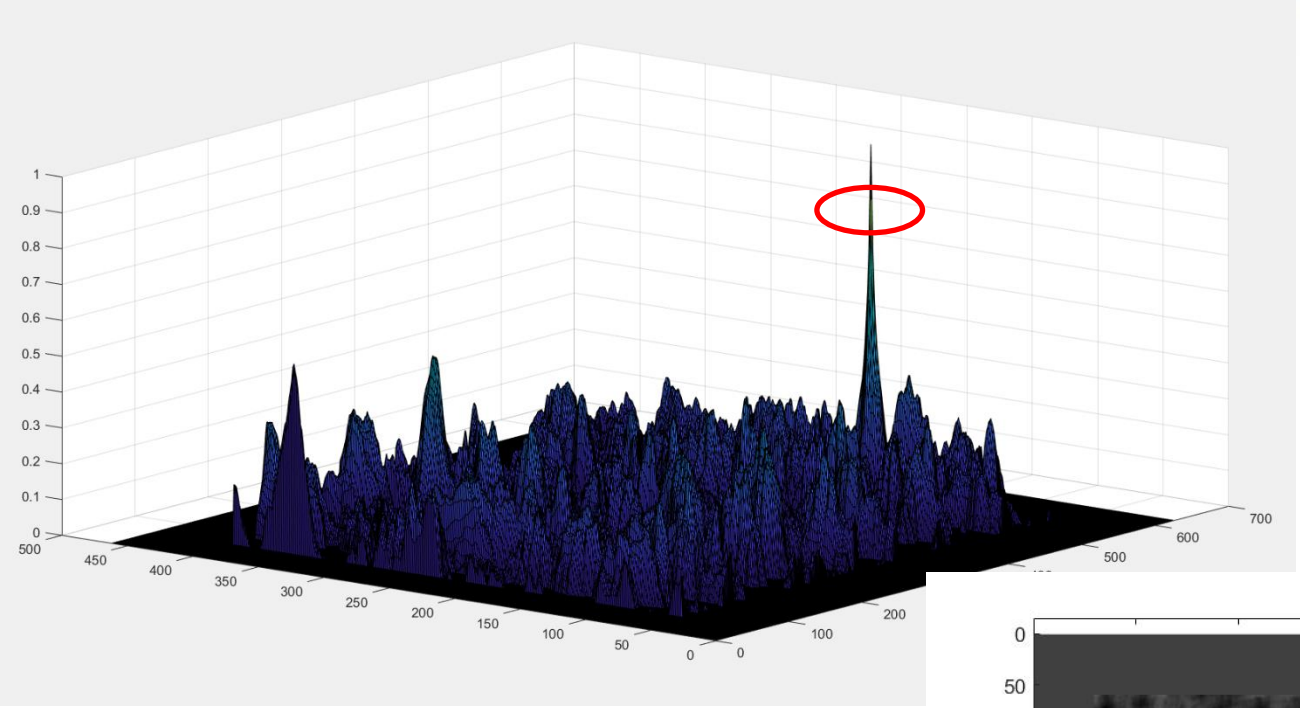
exemplo

imagem

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

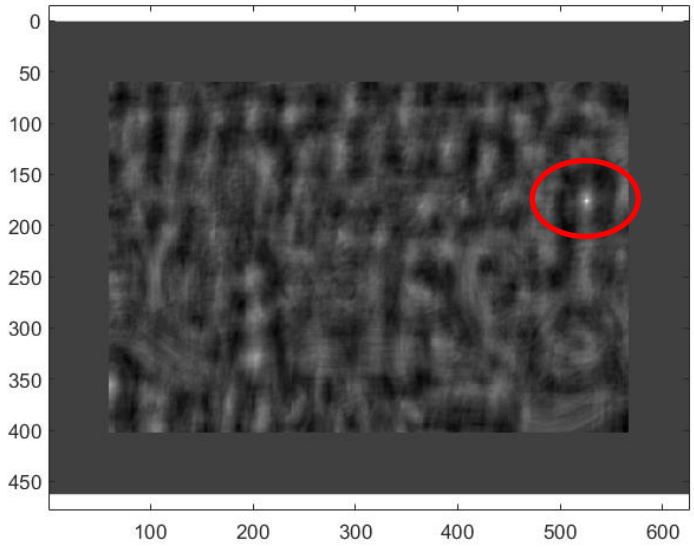
referência

Matriz de correlacoes



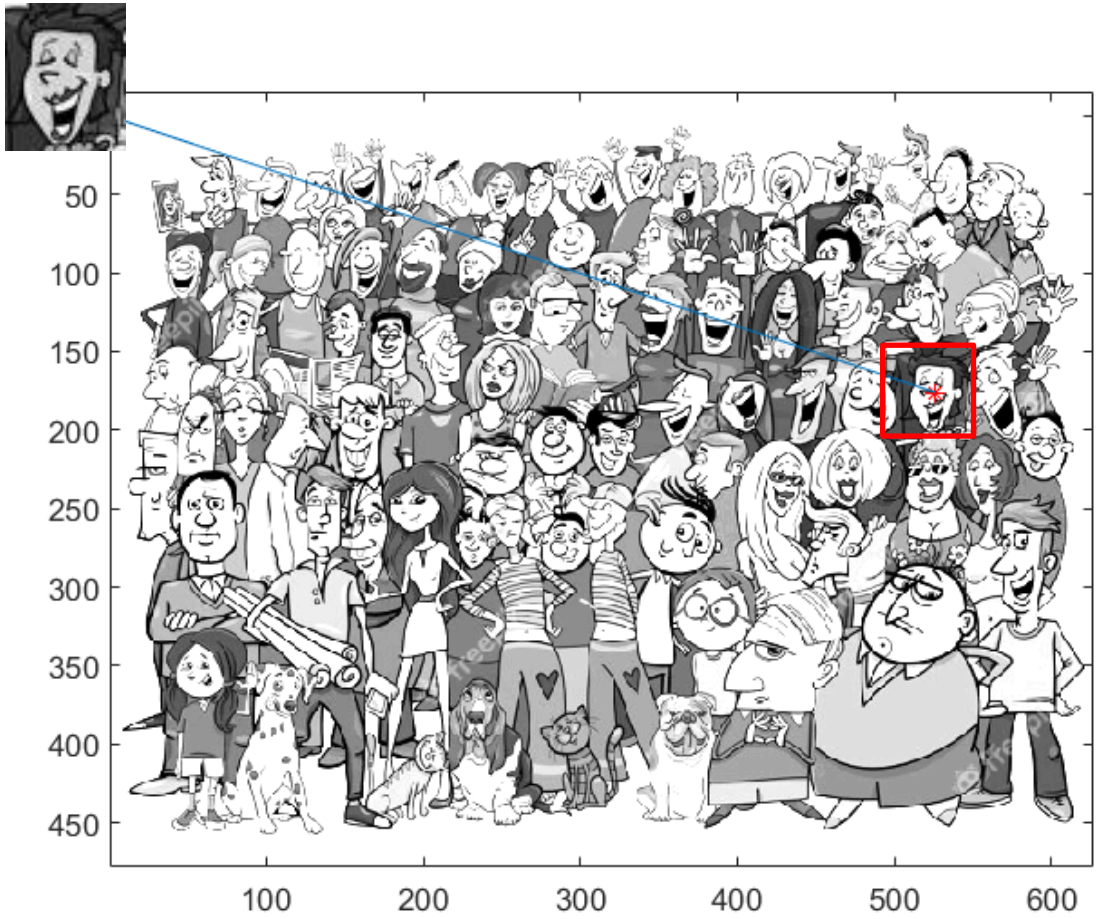
Valores da correlação para cada pixel (3D)

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

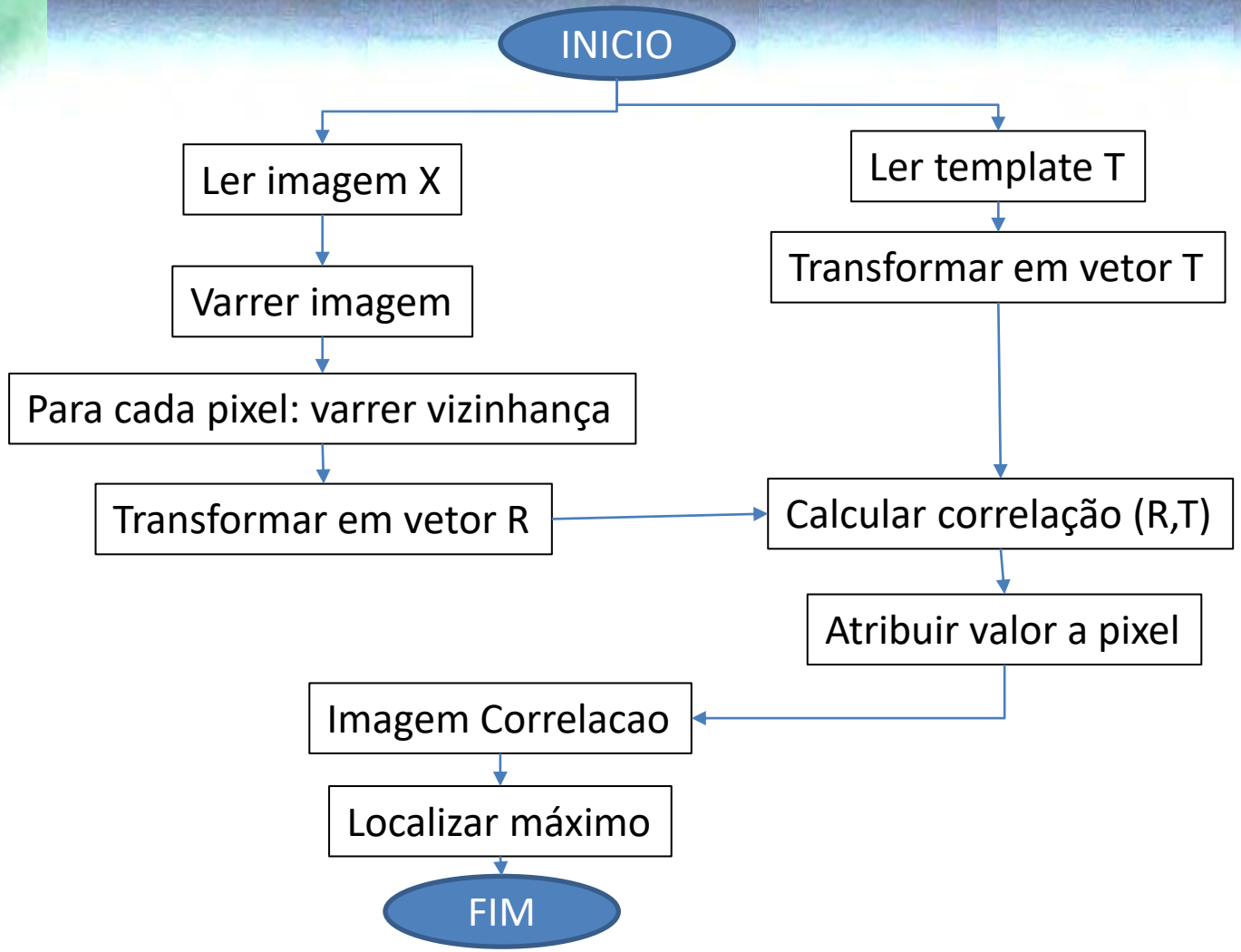


Localizar máximo

- Máximo local de maior valor (absoluto)



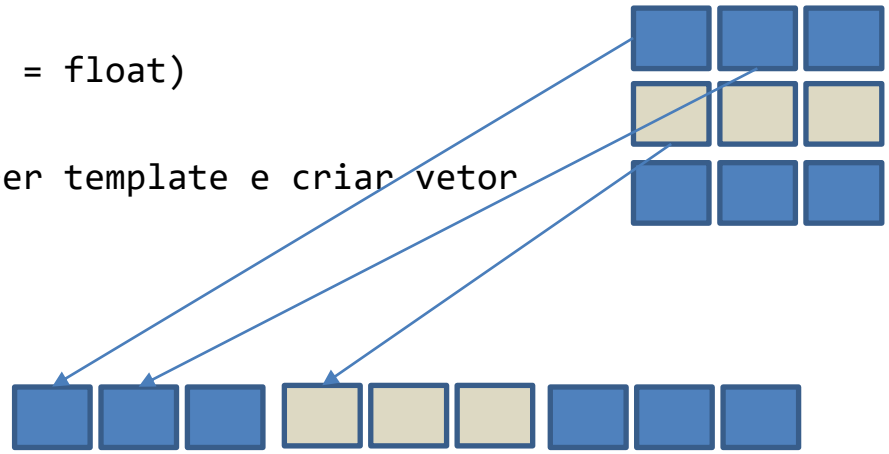
Um programa


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

Encontrar esta região na imagem ao lado.

```
# CORRELACAO
#####################################################################
import numpy as np
import matplotlib.pyplot as plt


## ################ vmatriz to vetor # #################
def mat2vec(M):
    tl,tc = M.shape
    M1=np.zeros(( tl*tc),dtype = float)
    c=0
    for i in range(tl): # varrer template e criar vetor
        for j in range(tc):
        M1[c]=M[i,j]
        c=c+1
    return(M1)
```

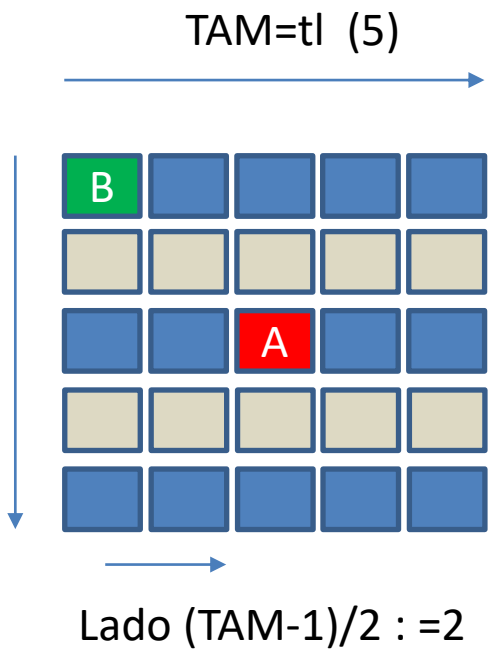


```
X1= plt.imread('recorte.tif’) # ler imagem de entrada e recuperar dimensoes
nl,nc = X1.shape
X=np.array(X1, dtype=float) #transforma em float
Y=np.zeros((nl,nc),dtype = float) # replica para gerar saida
Y=np.uint8(Y)

# Ler template
T1= plt.imread('template.tif')
tl,tc = T1.shape
T=np.array(T1, dtype=float)
print('Image size: ',nl,nc, " Template size: ", tl, tc)
dim=tl                  # tamanho do template, DEVE ser um quadrado!
lado=(dim-1)/2          # numero de vizinhos antes ou depois do central
lado=np.uint8(lado)


TT=mat2vec(T) # transaforma matriz template a vetor
```

```
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) # posicao inicial, canto superior esquero do recorte
        q=(C-lado)
        R=X[p:p+tl, q:q+tc]
    RR=mat2vec(R) # transforma regiao a vetor
    rm=np.corrcoef(RR,TT) # matriz de correlacao
    r=rm[0,1] # correlacao
    if r<0: # nao nos interessam correlacoes negativas
        r=0
    rp=r*255 # transformar em uint 8 para salvar
    v=np.uint8( np.round( rp ) ) # arredonda e muda a uint8
    Y[L,C]=v # salva na posicao do central
plt.imsave('Matriz.png',Y,cmap='gray')
```

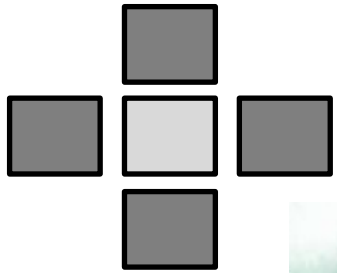


- A=central na posição L,C
- B=pixel no canto sup esq (p,q)
- p=L-lado
- q=C-lado

buscar maximo local que supere limiar

```
Limiar=250  # em uint8 (0-255), na verdade poderia ser em float (0 a 1)
maximo=0  # máximo valor na imagem
for L in range(lado,nl-lado): # varrer em linhas e colunas
    for C in range (lado, nc-lado):
        if Y[L,C]>Limiar: # supera limiar, vejamos se é maximo local
            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: # e se é maior que o anterior
                    maxL=L
                    maxC=C
                    maximo=Y[L,C]

print('localizado em: ', maxL, maxC, '| ',maximo)
```



aplicação