VISÃO COMPUTACIONAL

Extração de Caracterísicas Locais

FILIPE ROLIM CORDEIRO (FILIPE.ROLIM@GMAIL.COM)

Reconhecimento de instância de Objeto

Schmid and Mohr 1997

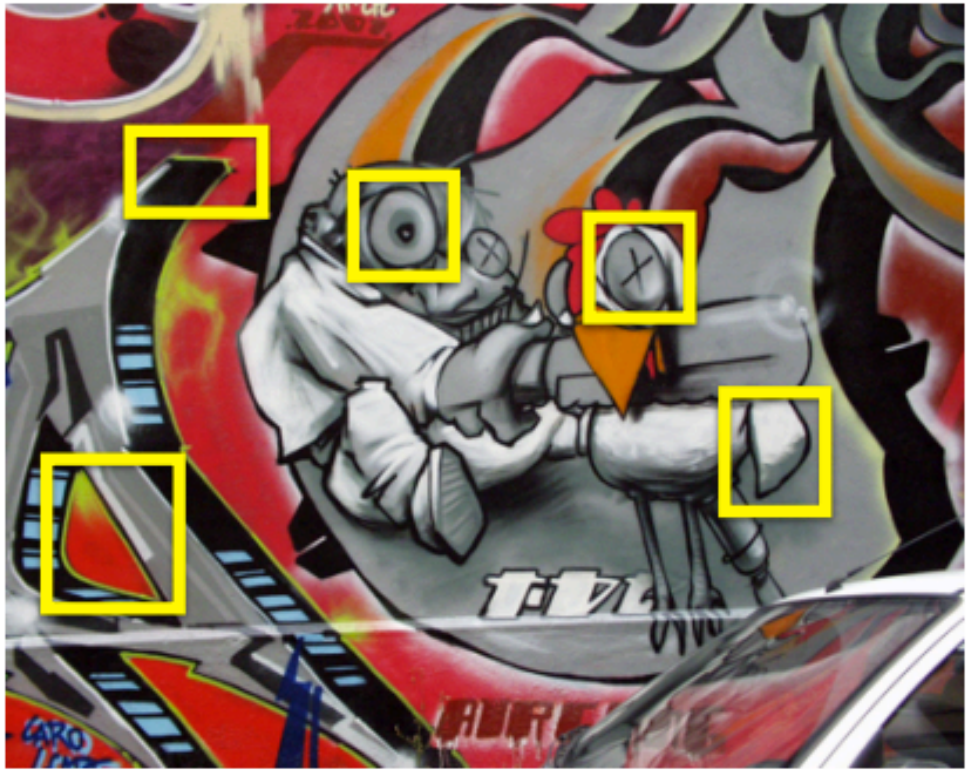
Sivic and Zisserman, 2003

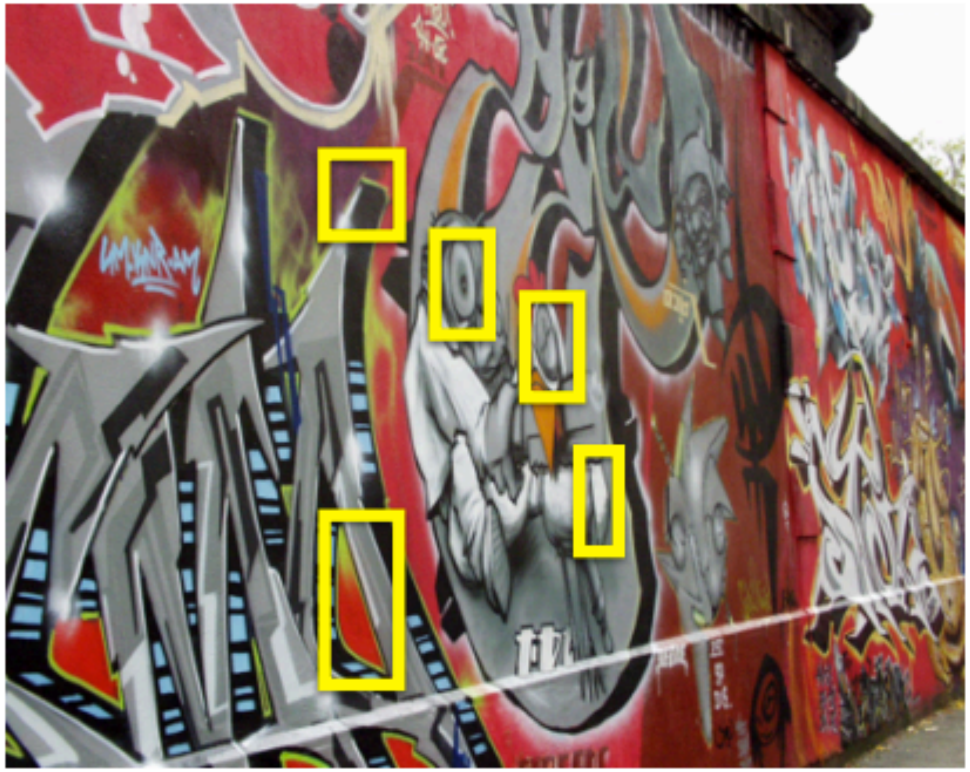
Rothganger et al. 2003

Lowe 2002

Mosaico de Imagens

▷Se olharmos essa imagens, como relacionamos?

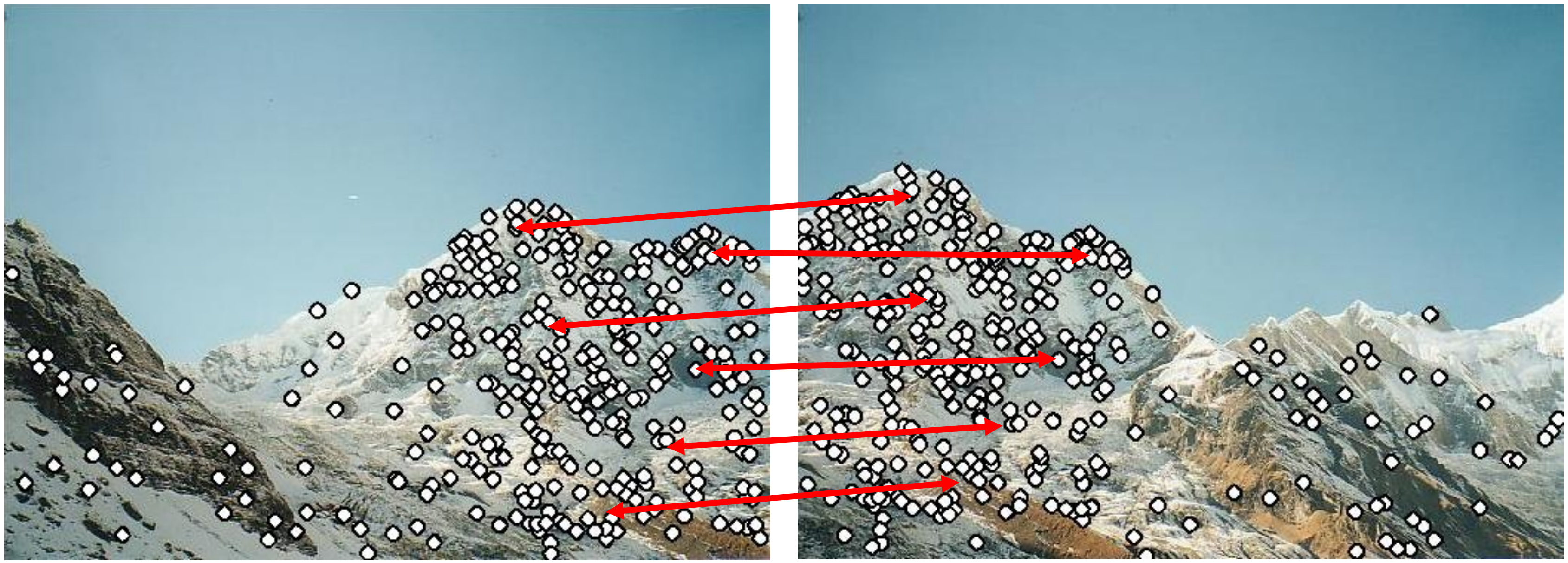




Como combinar as duas imagens?

Como combinar as duas imagens?

- Passo 1: extrair características
- Passo 2: casar características



Como combinar as duas imagens?

- Passo 1: extrair caraterísticas
- Passo 2: casar características
- Passo 3: alinhar imagens

Aplicações

- ▷Alinhamento de imagem
- ▷Reconstrução 3D
- ▷Reconhecimento de objetos
- ▷Rastreamento
- ▷Navegação robótica

Casamento de Imagens

Casamento de Imagens

by Diva Sian

by swashford

Caso mais difícil

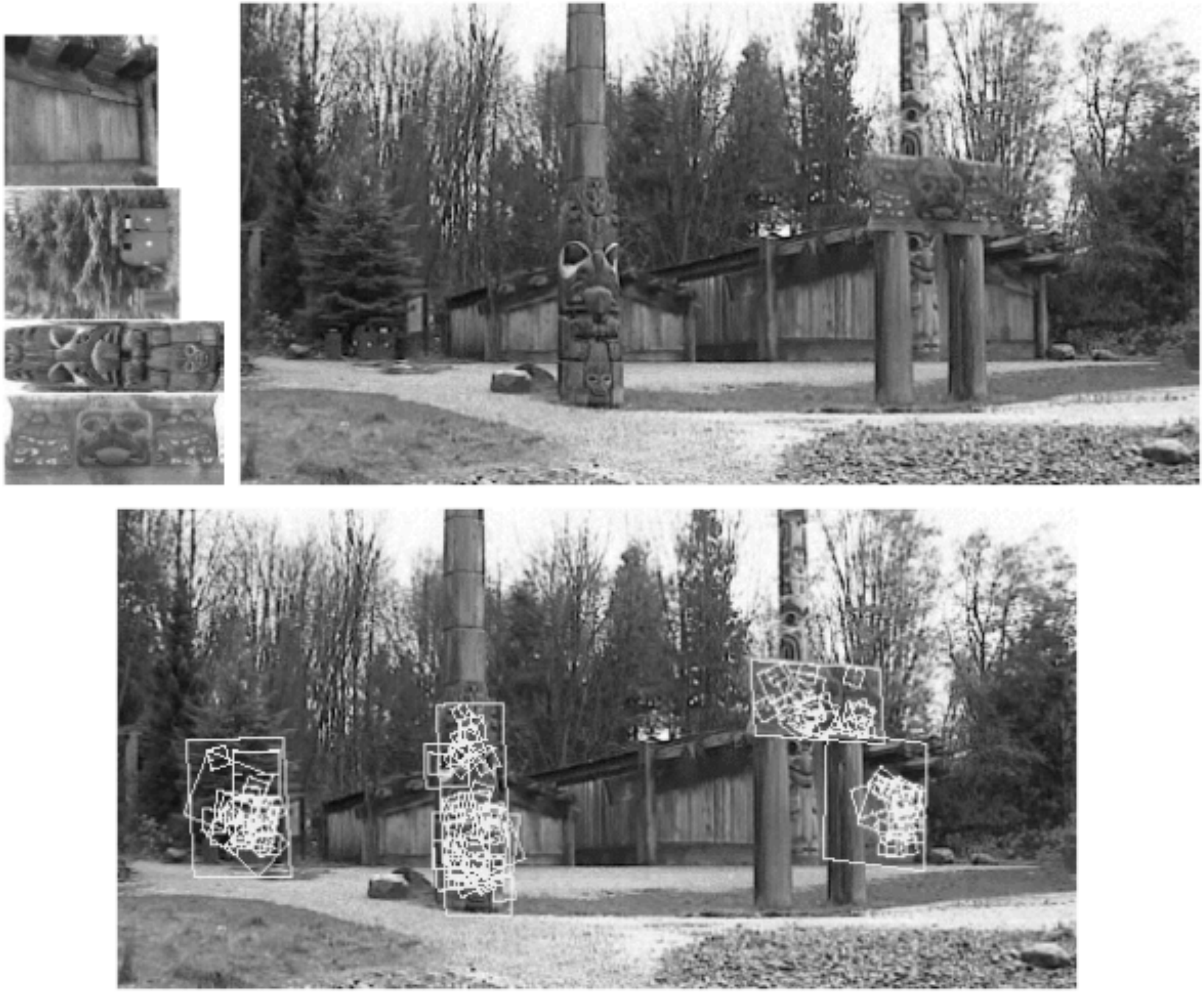
by Diva Sian

by scgbt

Reconhecimento de instância

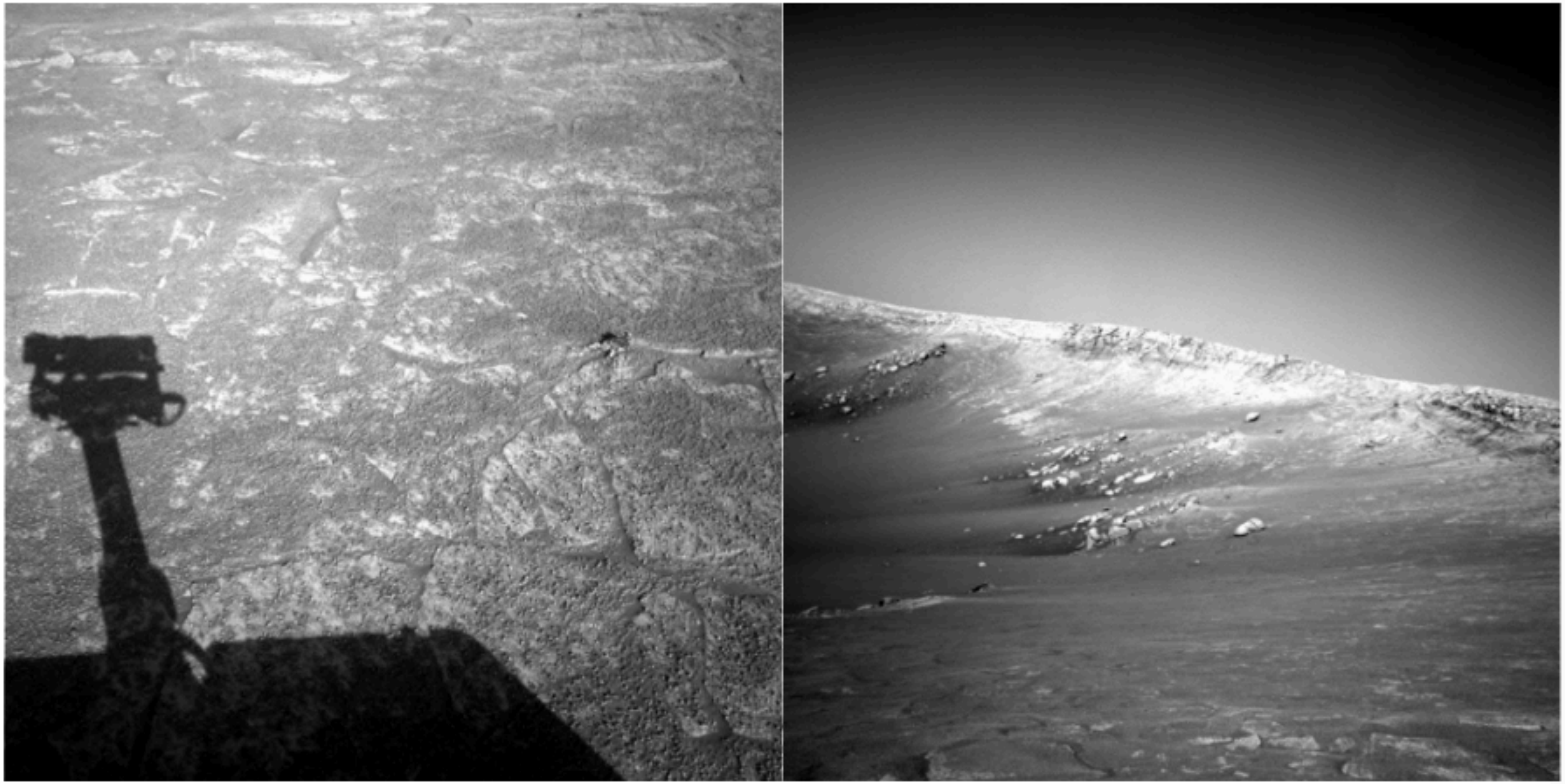
Reconhecimento sobre oclusão

Reconhecimento de Localização



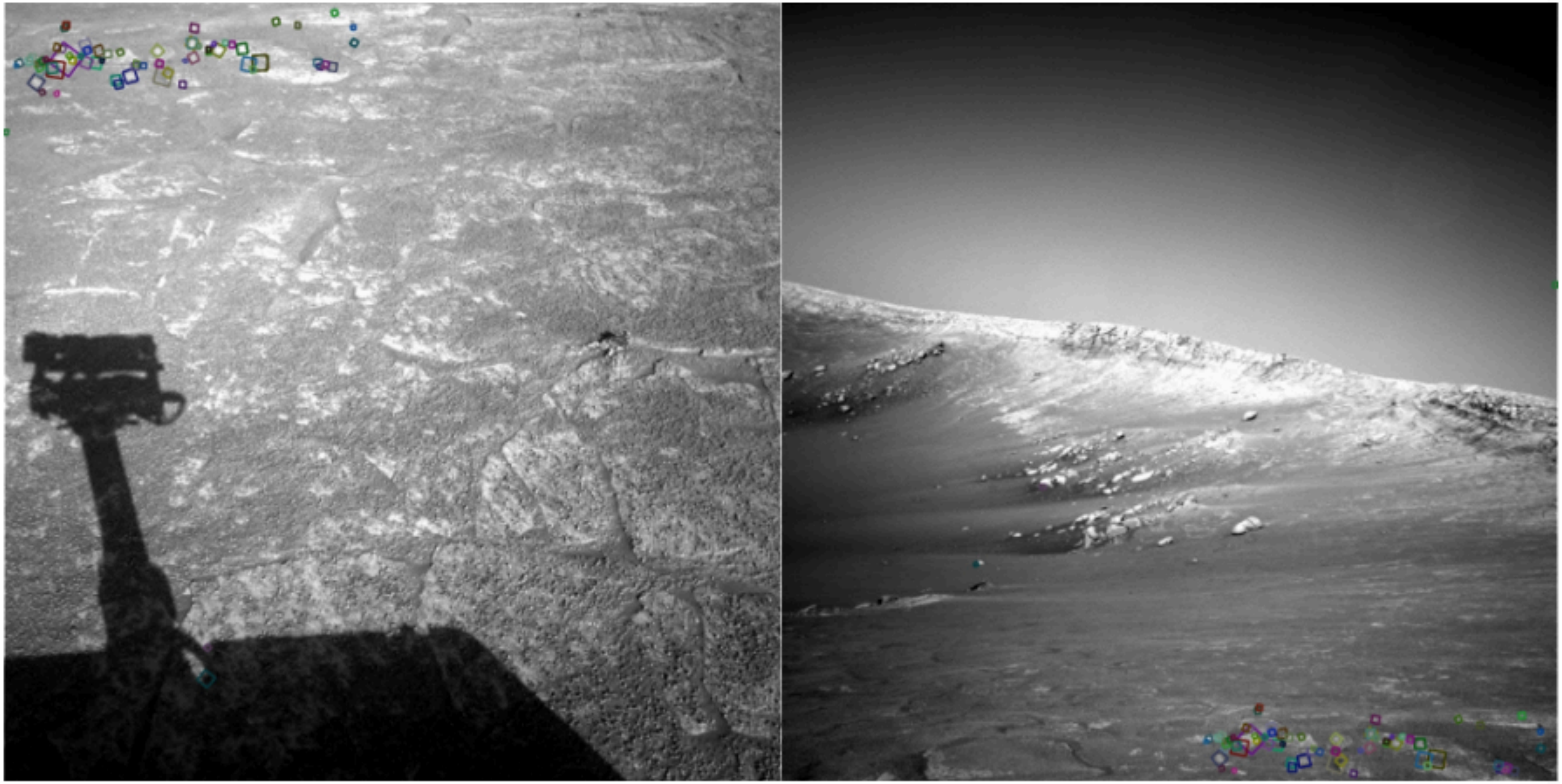
Casamento de imagens

Como encontrar os pontos correspondentes?



NASA Mars Rover images

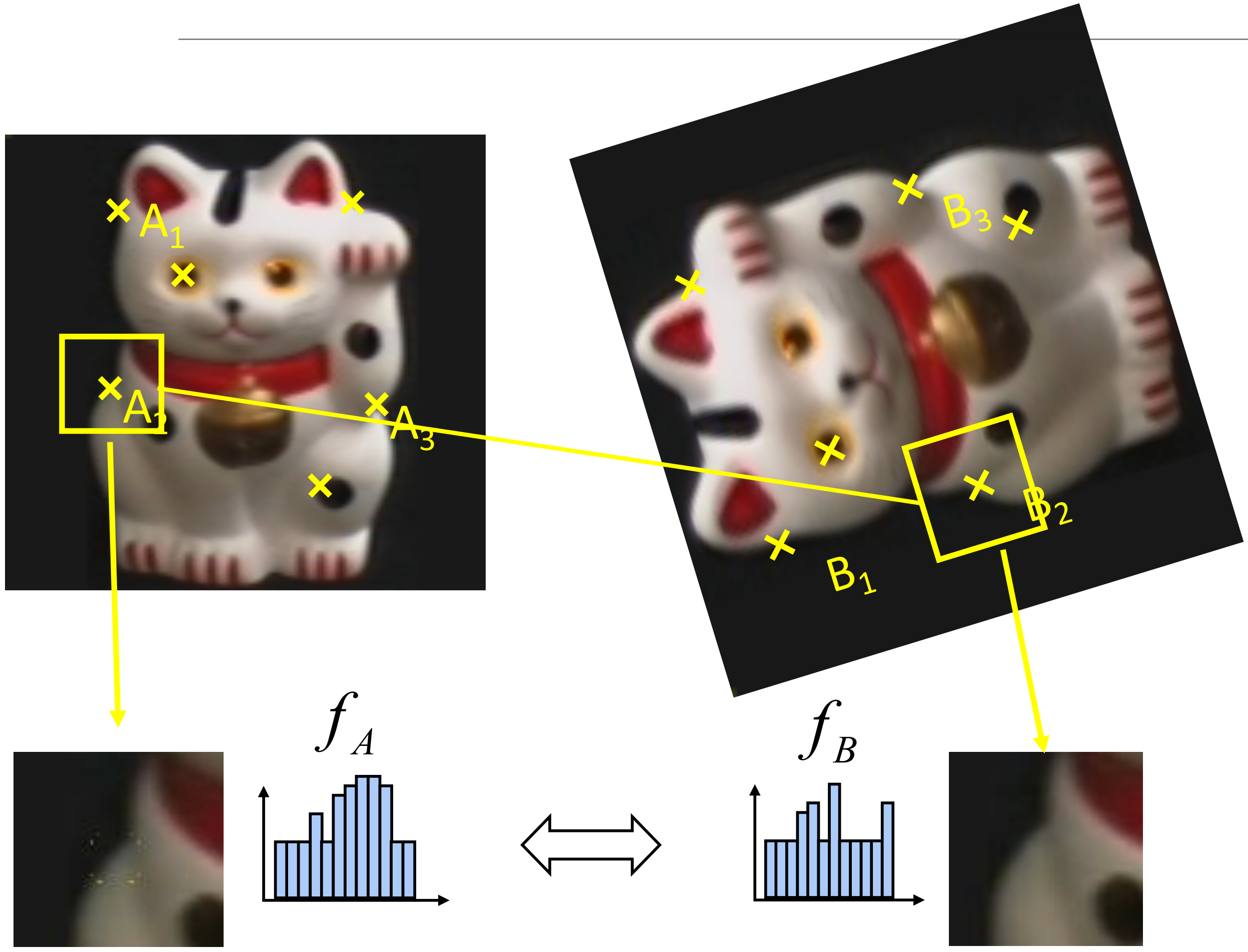
Onde estão os pontos correspondentes?



Que tipo de características você está tentando encontrar?

Explique seu raciocínio

Overview



$$d(f_A, f_B) < T$$

1. Detecção:
características distintas

2. Descritor:

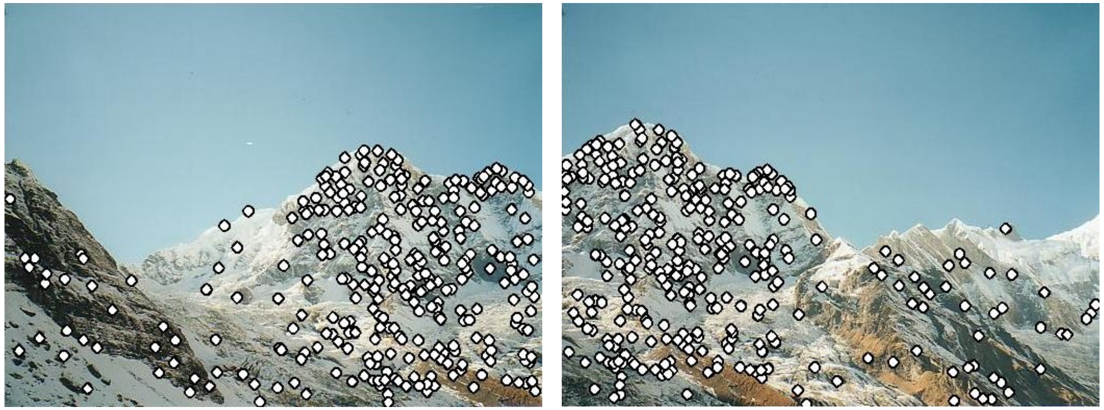
(a) Define região em volta de cada feature

(b) Extrai e normaliza conteúdo da região

(c) Computa descritor local da região normalizada

3. Relaciona descritores locais

Características de Bons Descritores

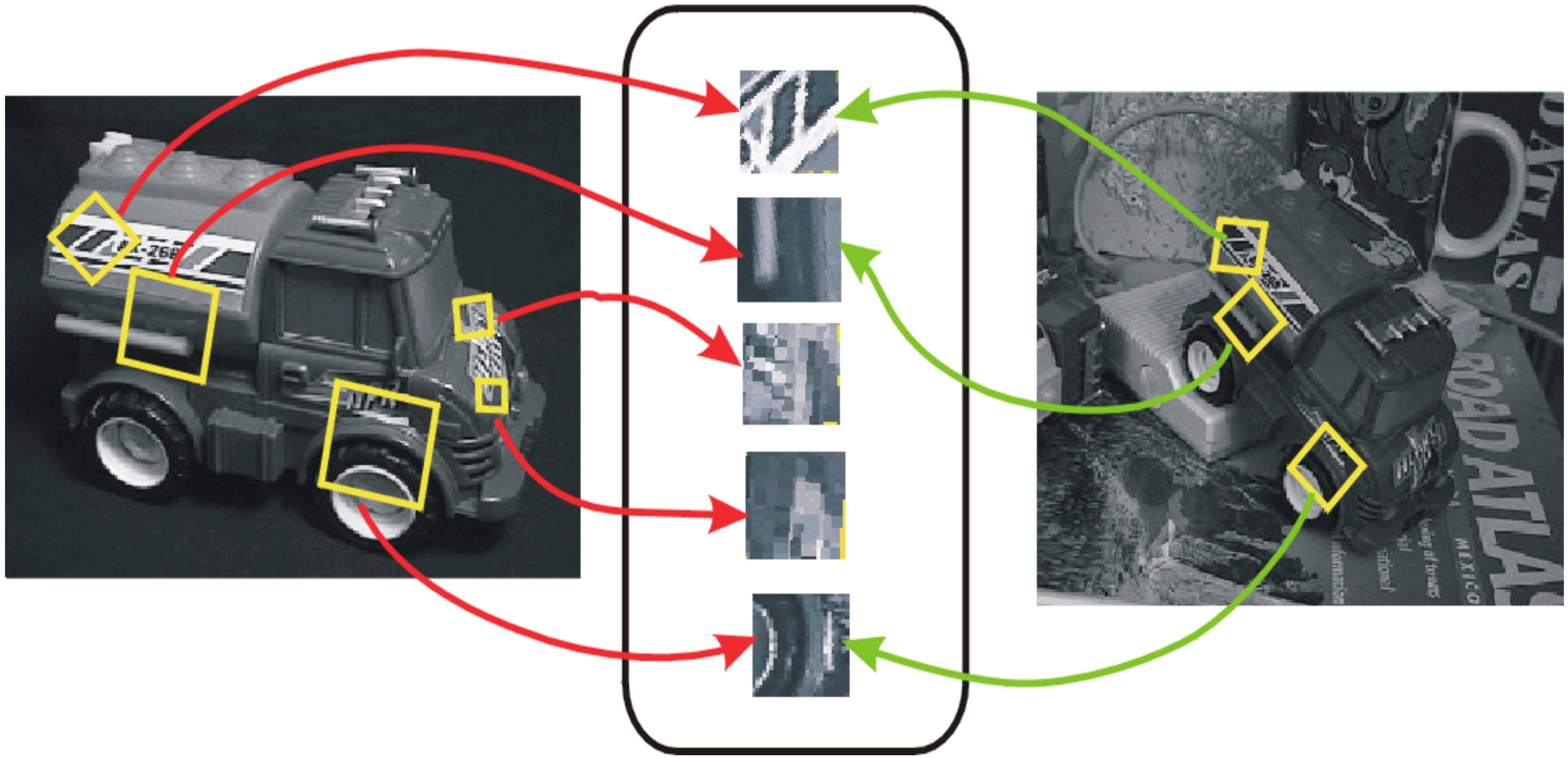


- Repetibilidade
  - A mesma característica pode ser encontrada em diversas imagens, independente de transformações geométricas ou fotométricas
- Saliência
  - Cada característica é única
- Compactação e eficiência
  - Muito menos características do que pixels
- Localidade
  - A característica ocupa uma região pequena da imagemç robusta a oclusão e clutter

Características Locais invariantes

Encontrar características que são invariantes a transformaçoes:

- ❑Invariância geométrica: translação, rotação, escala
- ❑Invariância fotométrica: brilho, exposição, …



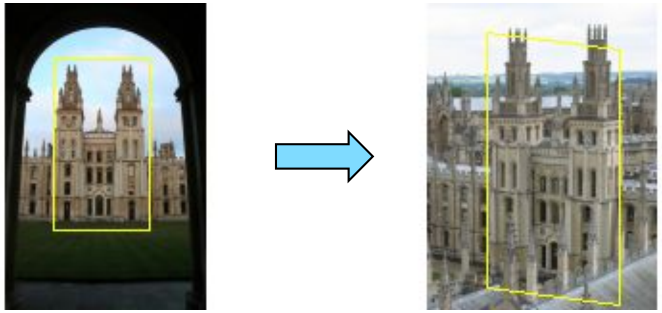
Descritor de características

Wide baseline stereo

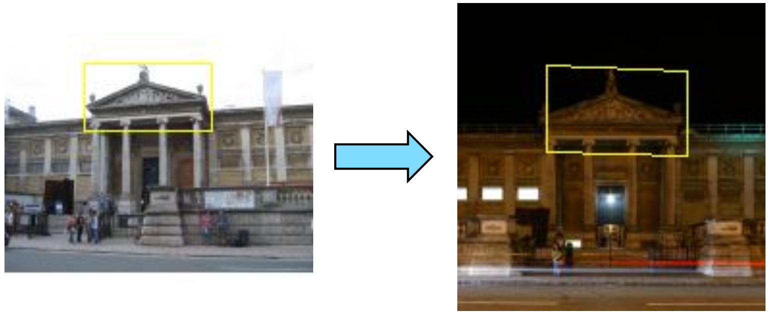
[Image from T. Tuytelaars ECCV 2006 tutorial]

Recognition of specific objects, scenes

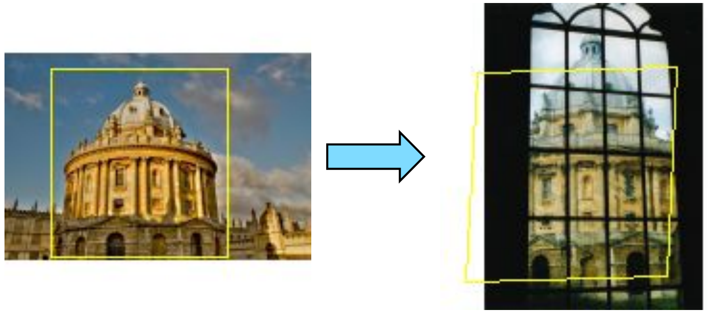
Scale



Viewpoint

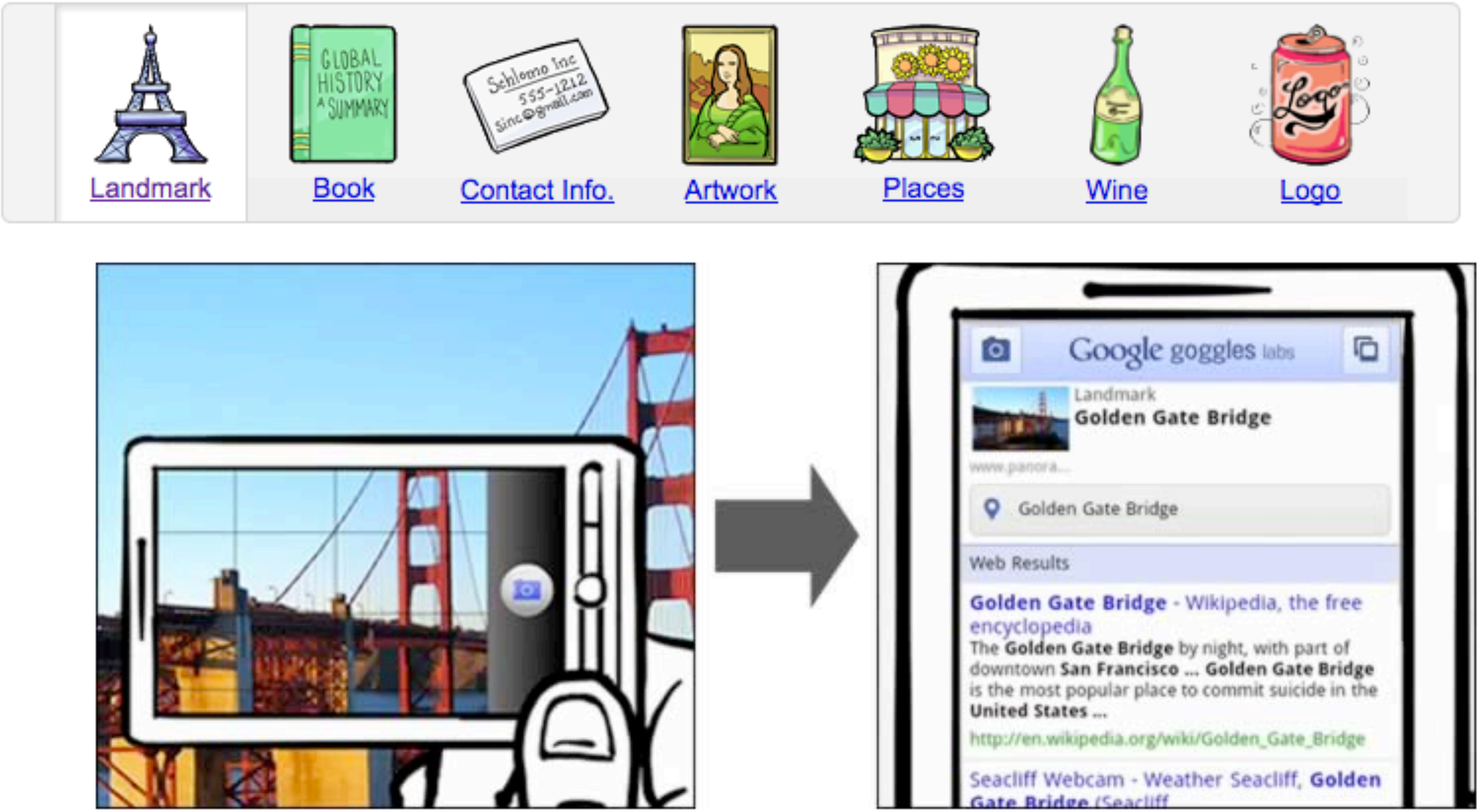


Lighting

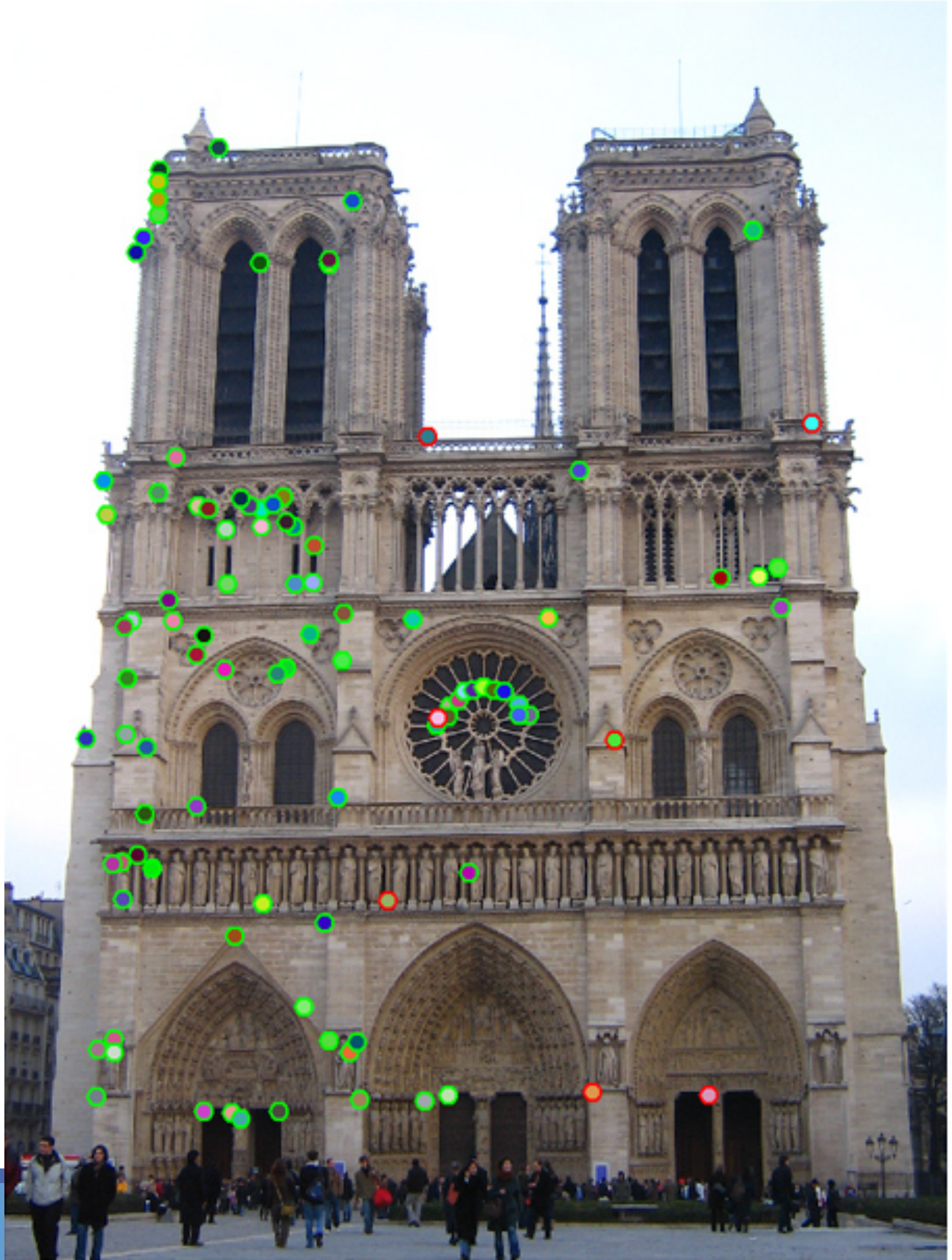


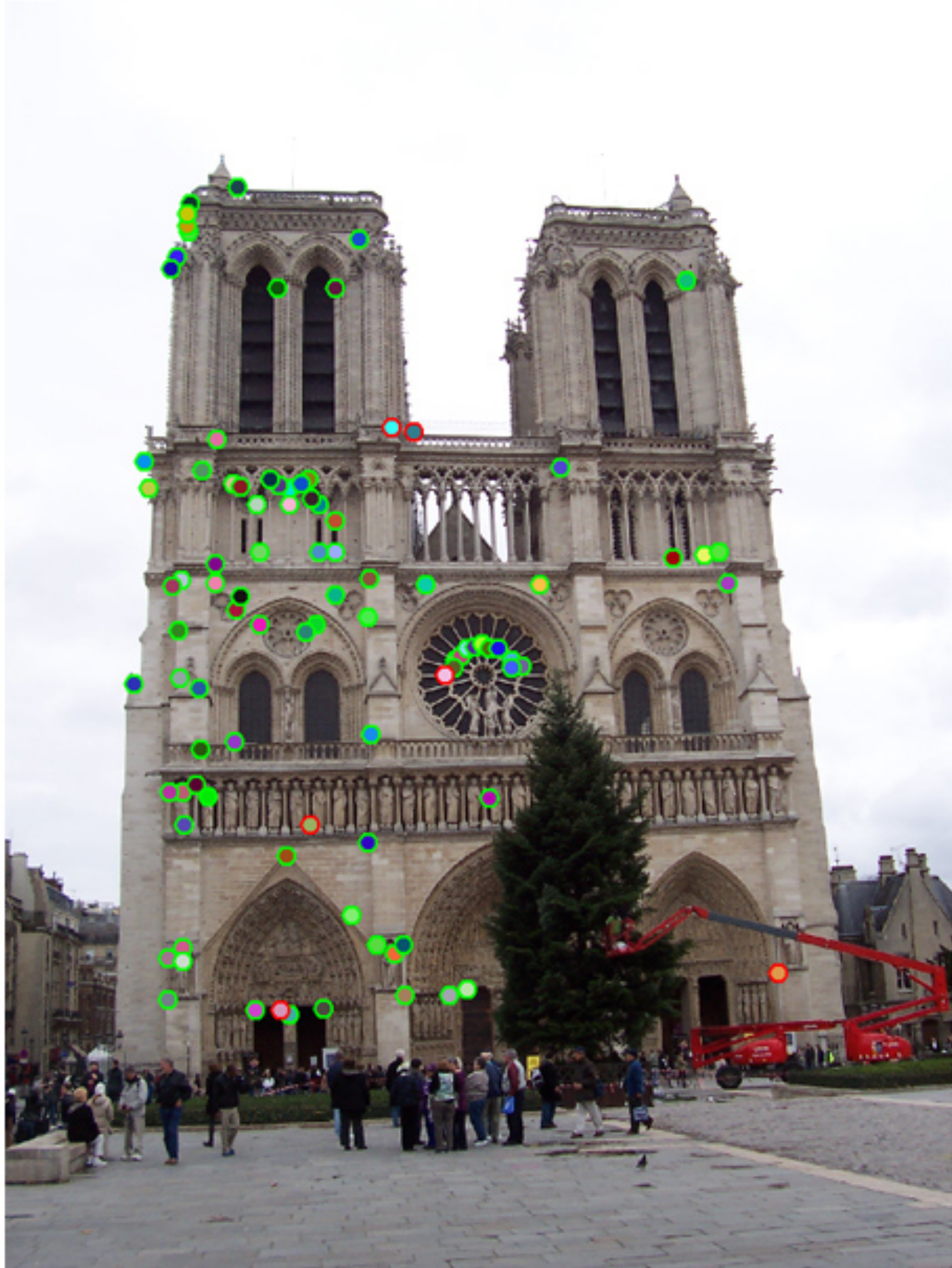
Occlusion

Google Goggles



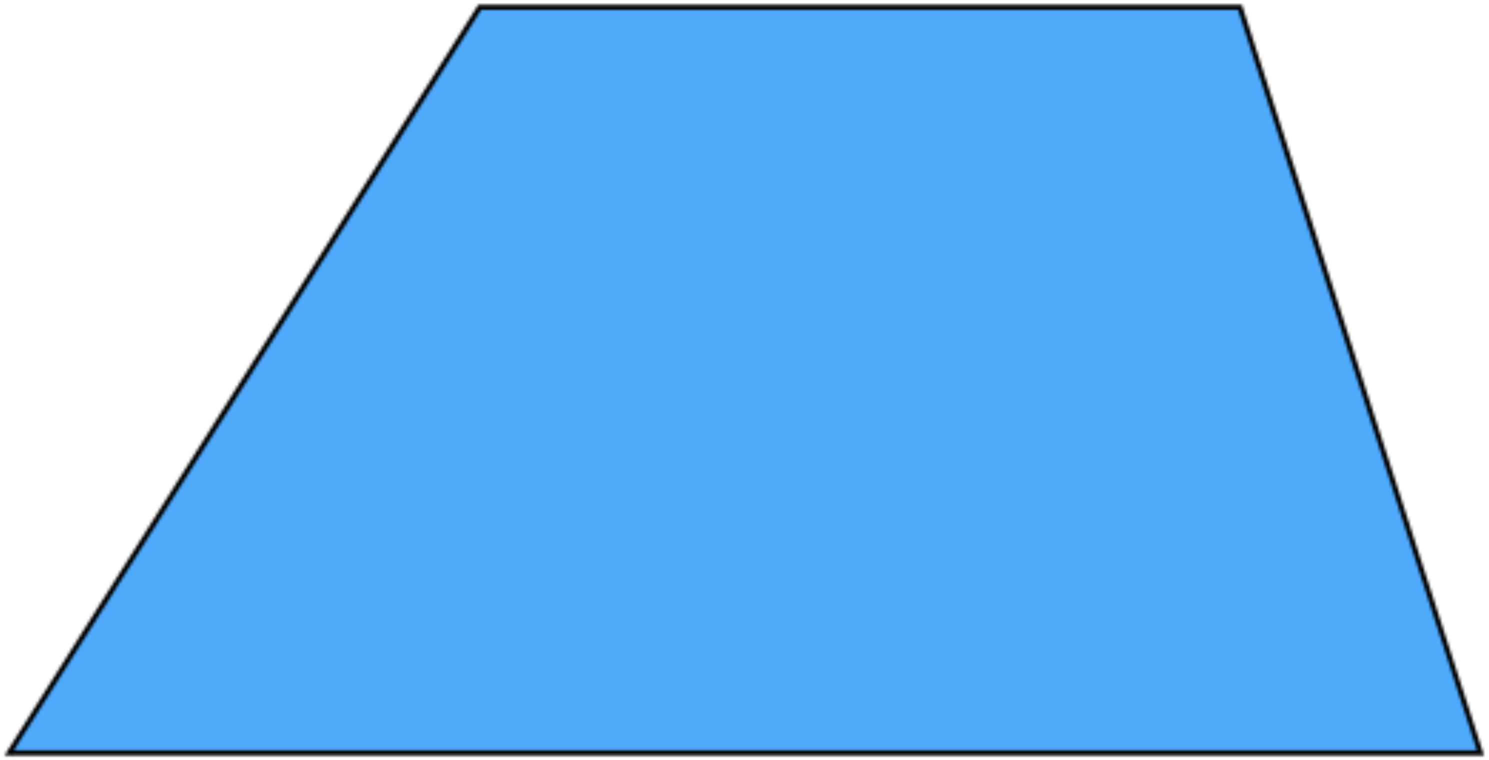
Quais características se relacionam?





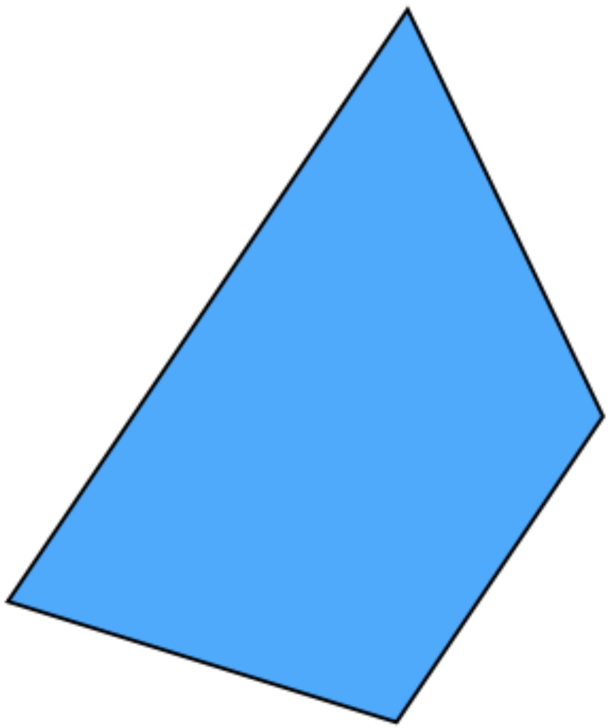
Detecção de Pontos Chave

- Escolha um ponto da imagem e encontre denovo na próxima imagem.



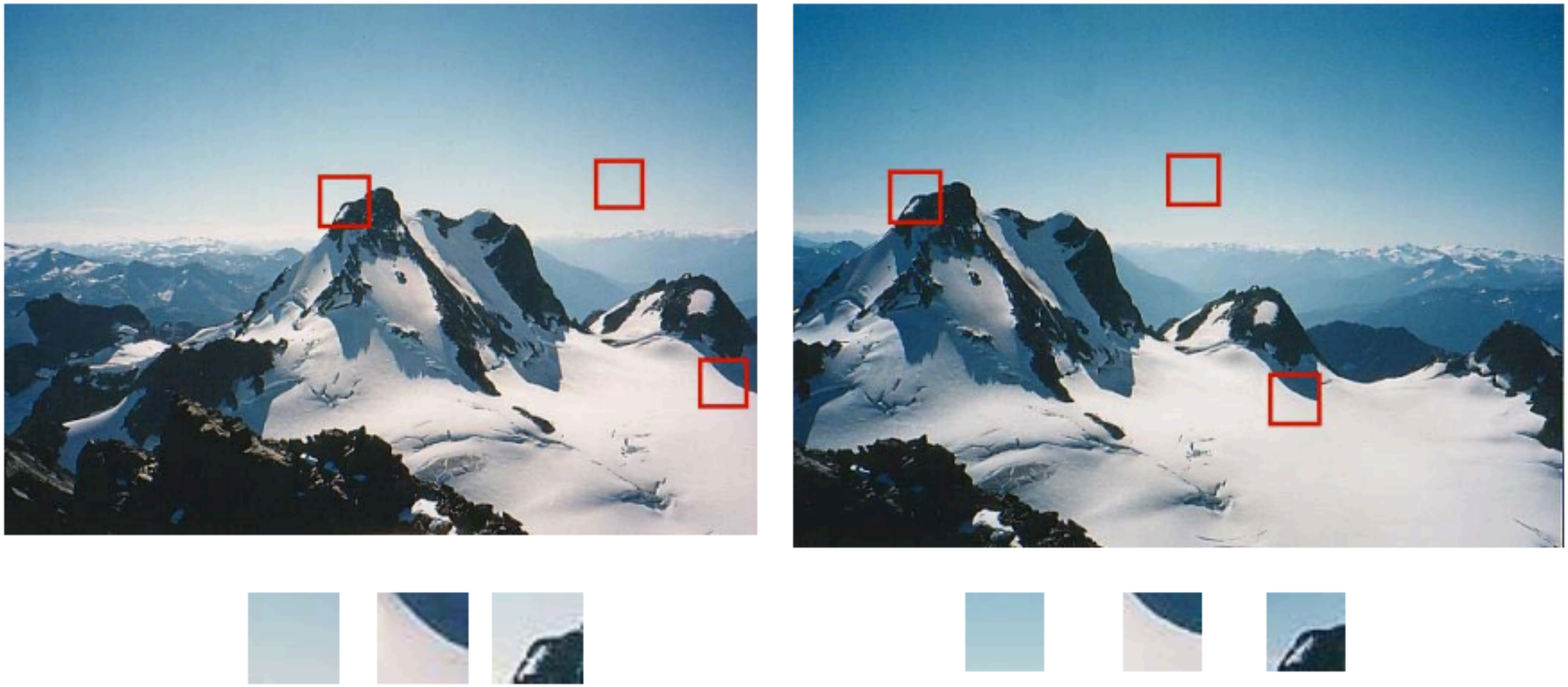
Que tipo de característica você selecionaria?

- Escolha um ponto da imagem e encontre denovo na próxima imagem.



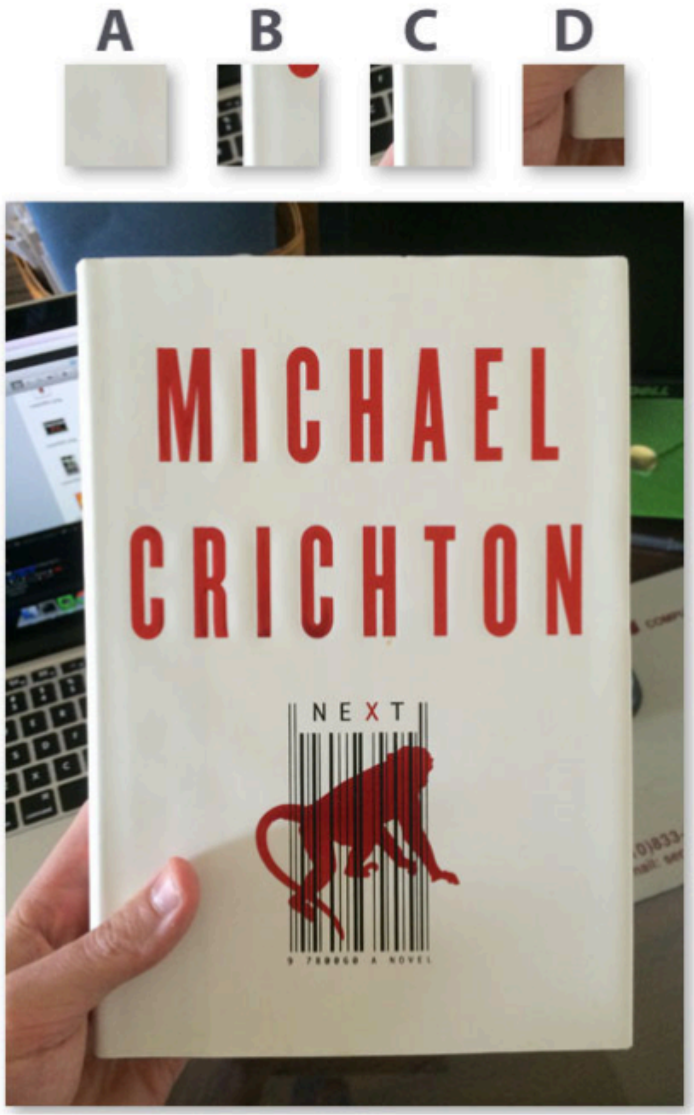
Que tipo de característica você selecionaria?

Quais características usar?

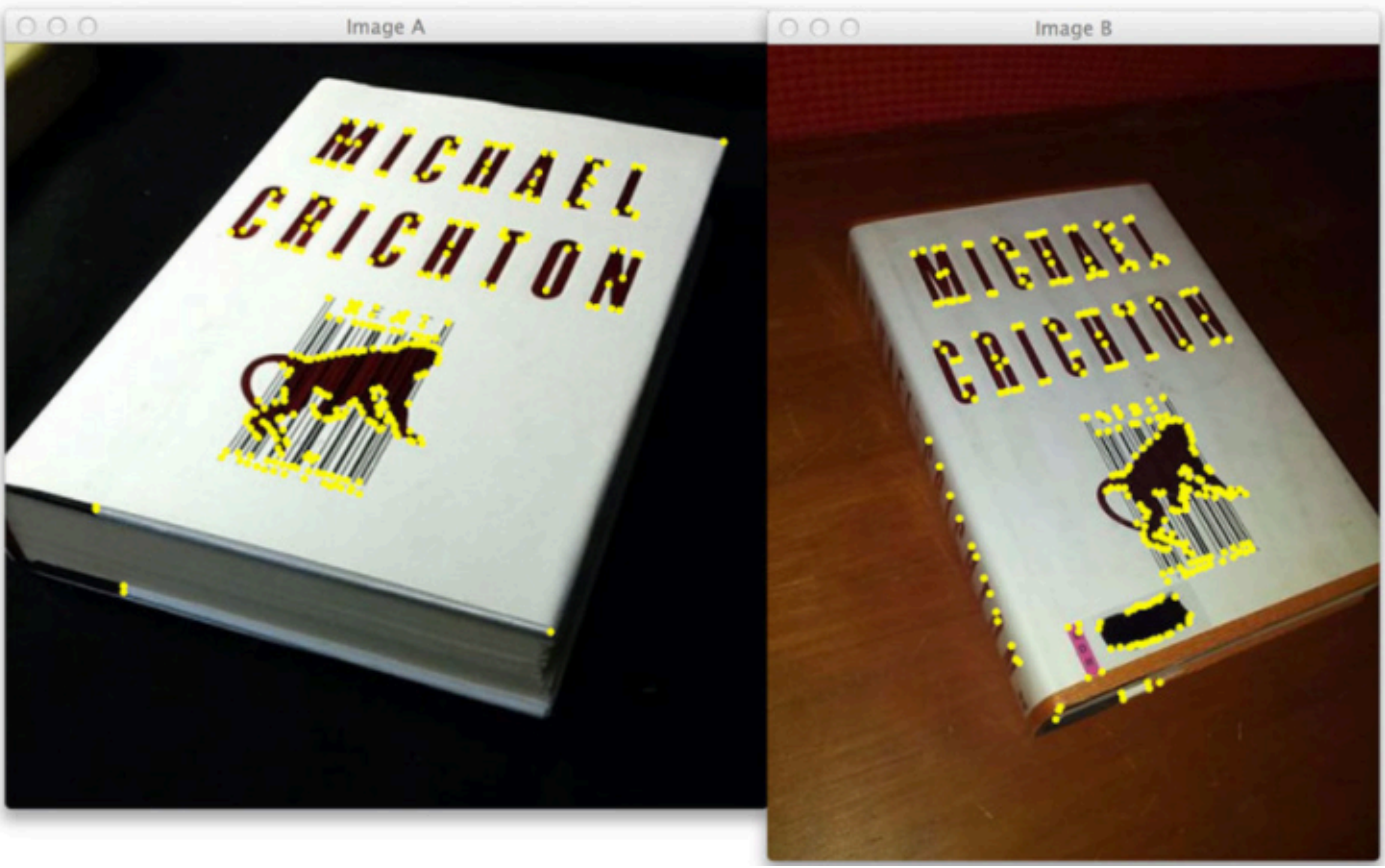


Usar características com gradientes em pelo menos duas orientações diferentes. Cantos?

▷ Quais desses patches são mais fáceis de encontrar na imagem?

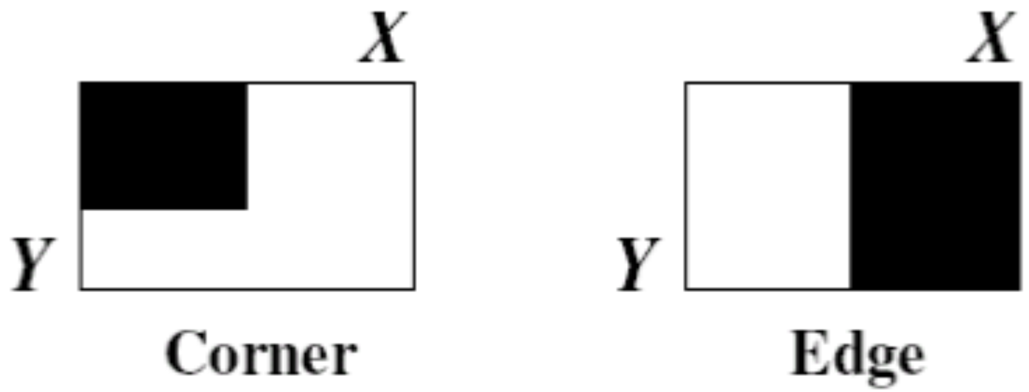


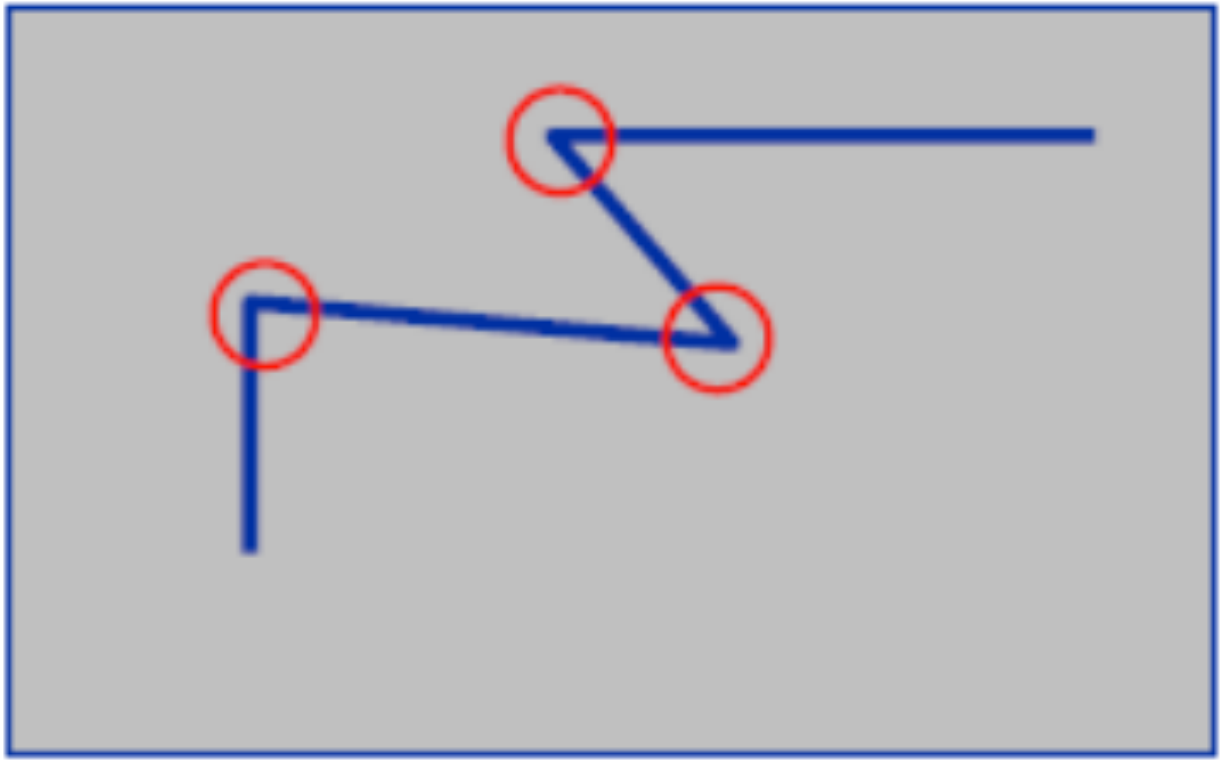
Repetibilidade

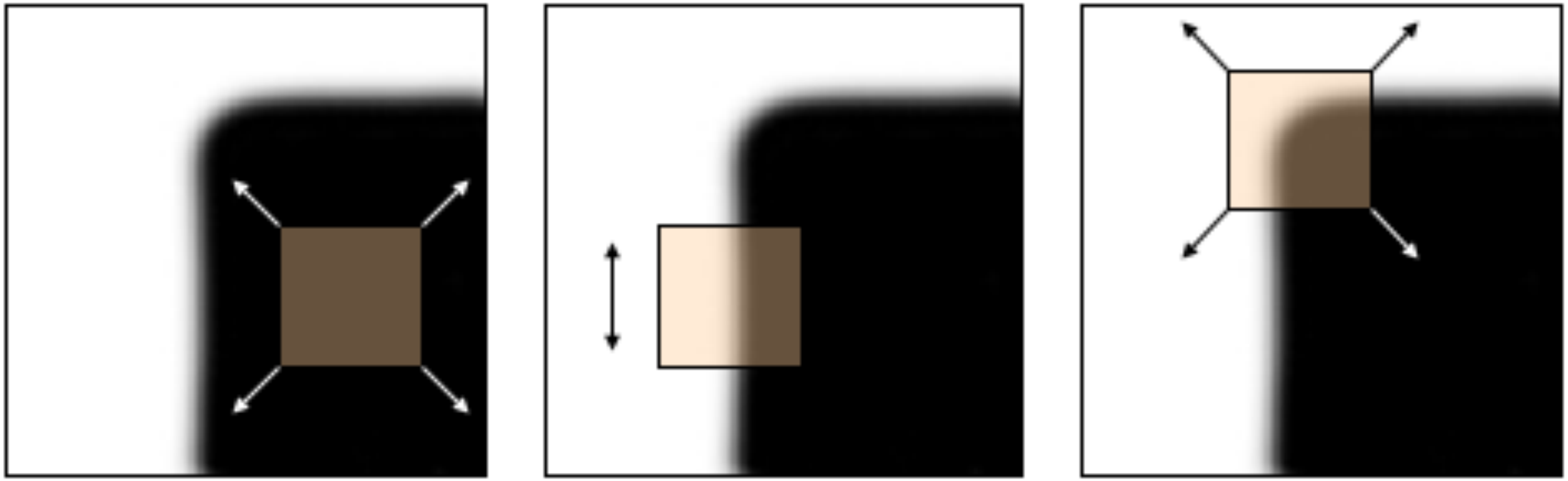


Cantos x Bordas

- **Corners** são localizações onde variações de intensidades de f(x,y) são altas em ambos X e Y
- **Edges** são localizações onde variação de f(x,y) em certa direção é alta, enquanto a variação ortogonal é baixa.







"flat" region:

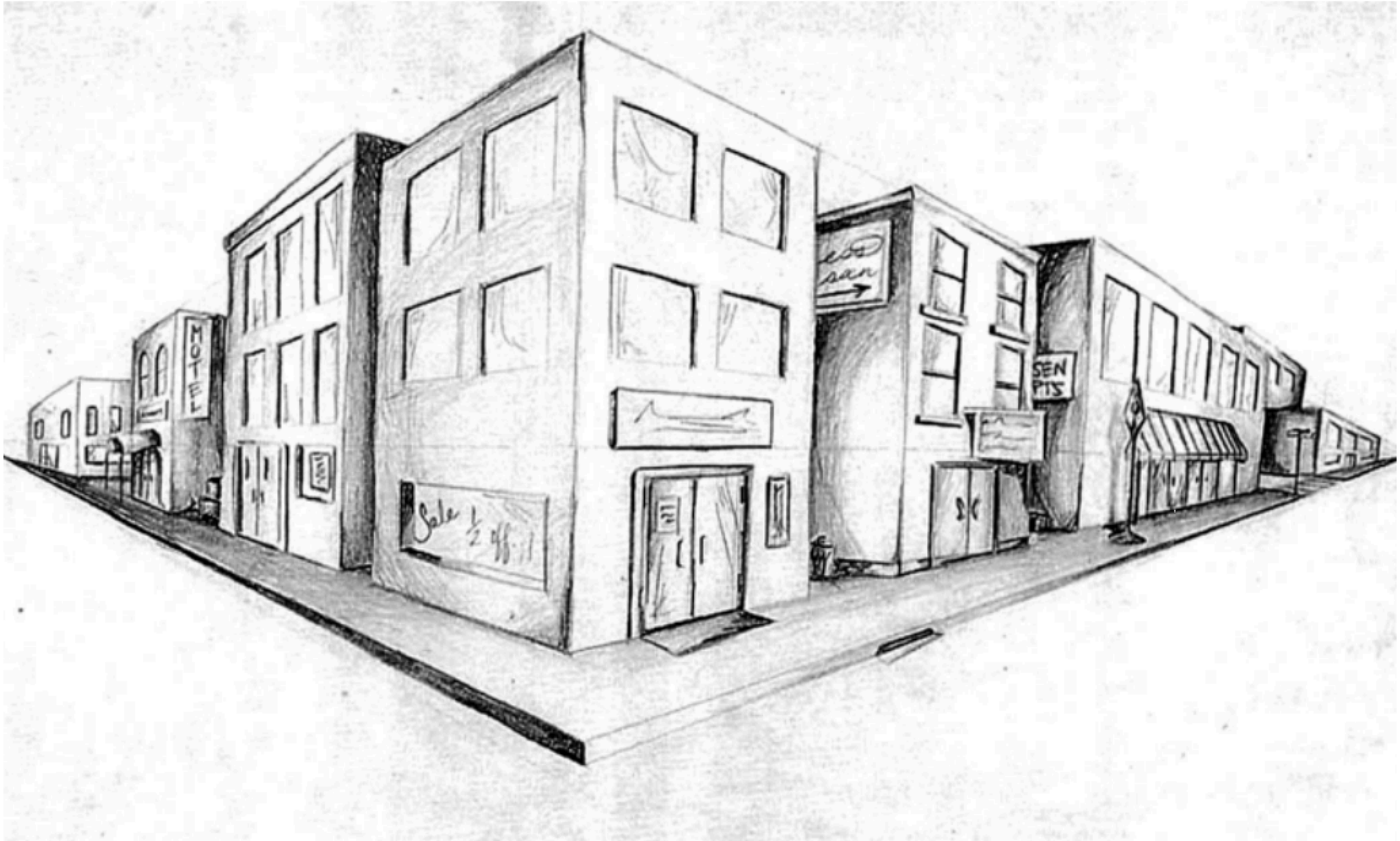
Nenhuma mudança em todas as direções

"edge":

Nenhuma mudança na direção da borda

"corner":

Mudança significativa em mais de uma direção



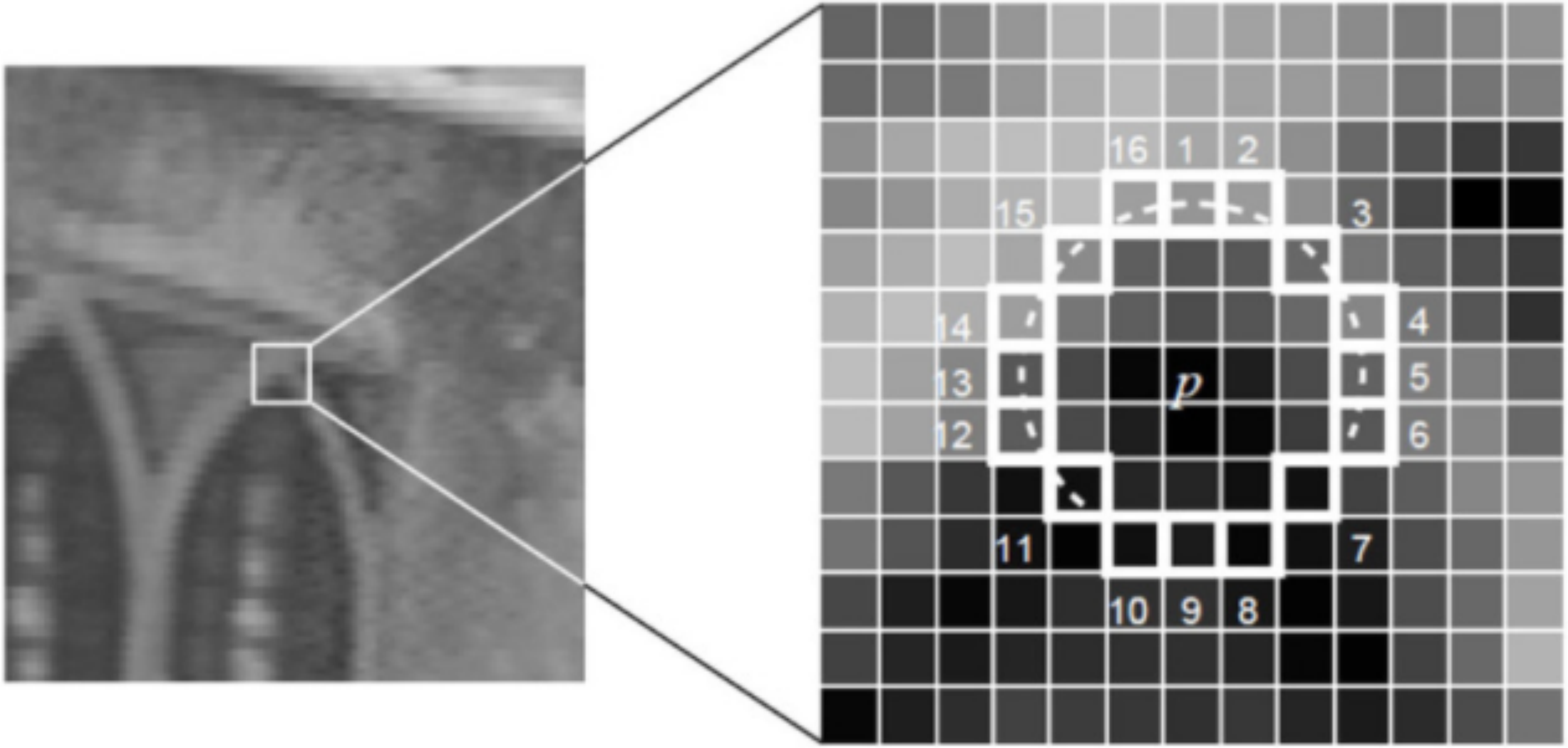
Detecting Corners

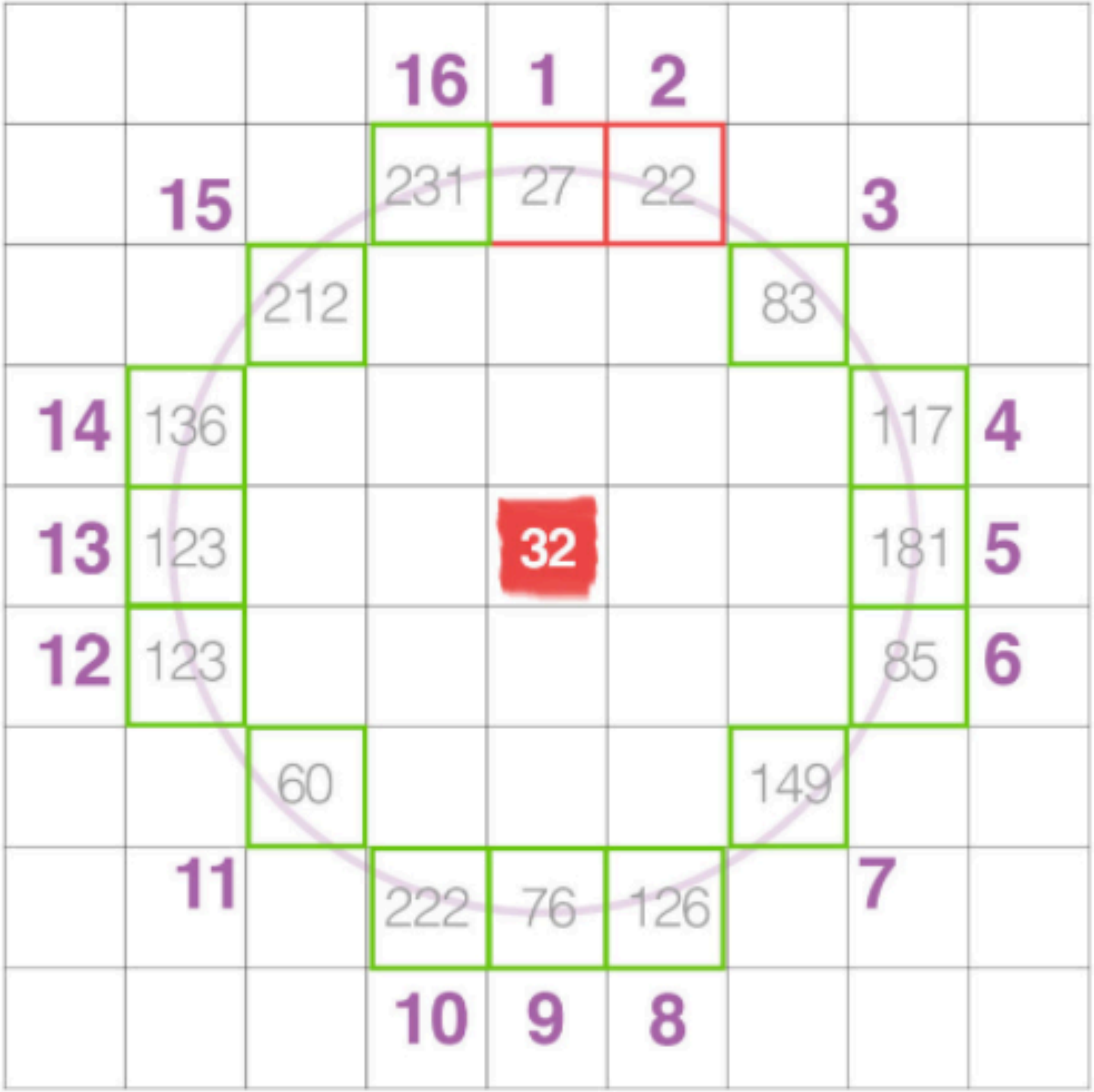
FAST

- Rosten and Drummond (2005)
  - Versões mais novas: 2006, 2010
- Detecta cantos

Fast

- Raio = 3
- Círculo de 16 pixels
- Qtd de pixels mais claros/escuros que o centro?

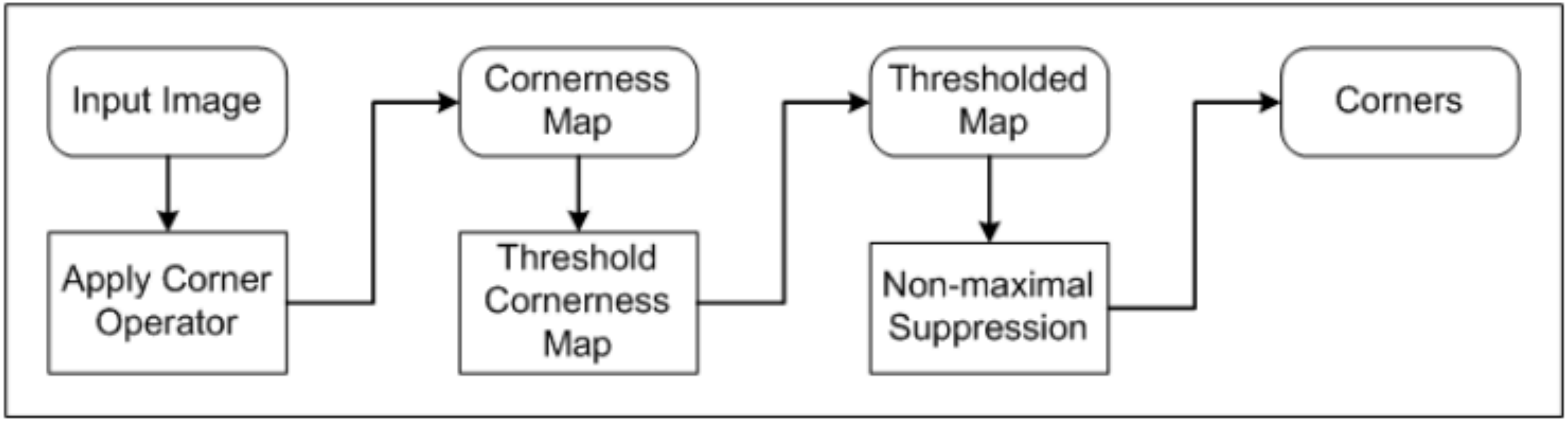


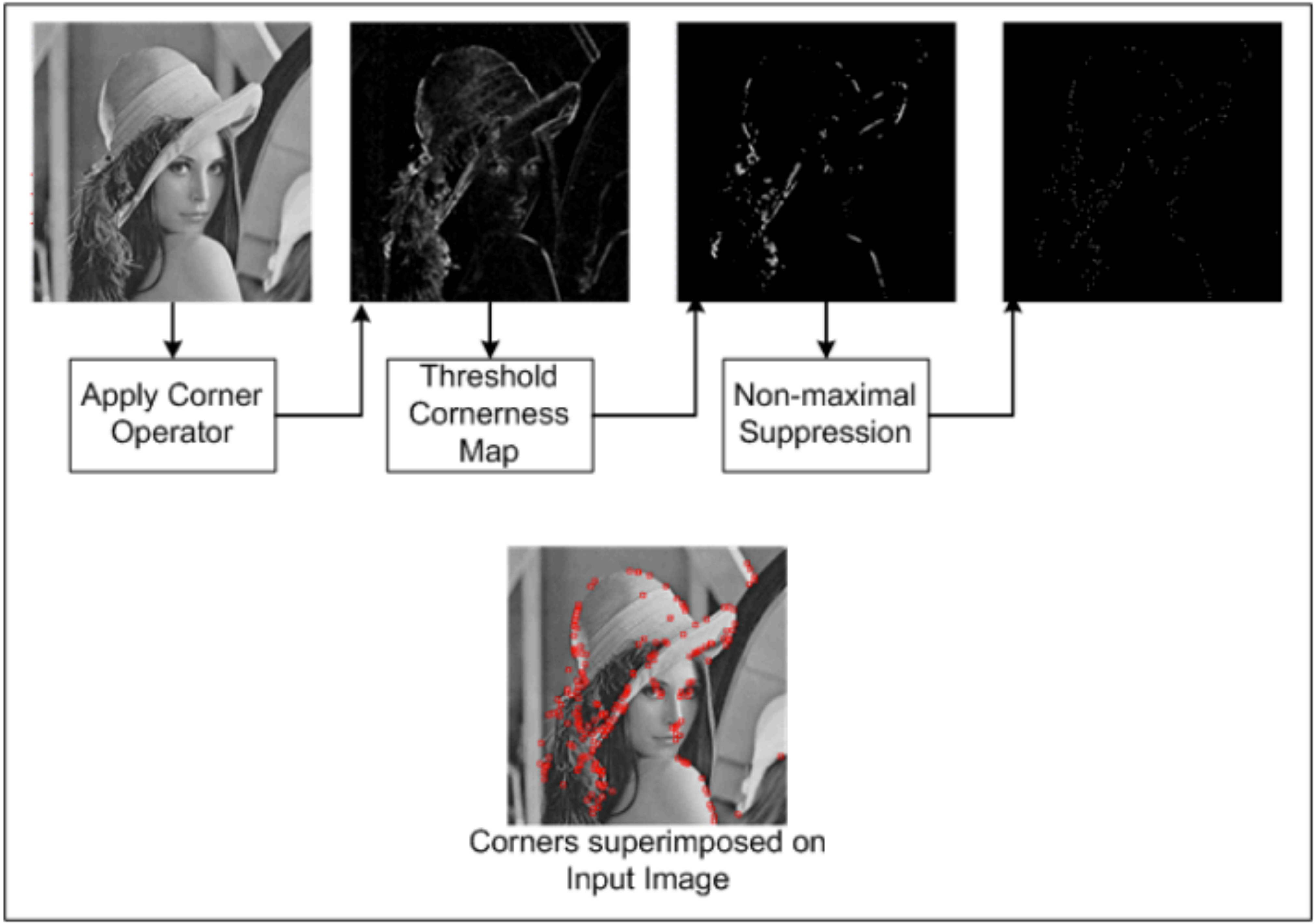

16
1
2
15
231
27
22
3
212
83
14
136
117
4
13
123
32
181
5
12
123
85
6
60
149
11
222
76
126
7
10
9
8

Images
MICHAEL
CRICHTON
NEXT

```python
import numpy as np
import cv2

image = cv2.imread('obama.jpeg')
gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)

detector = cv2.FastFeatureDetector_create()
kps = detector.detect(gray,None)

img2 = cv2.drawKeypoints(image, kps, None, color=(255,0,0))

cv2.imshow("imagem", np.hstack([image, img2]))
cv2.waitKey(0)
```

Harris


Input Image
Apply Corner Operator
Cornerness Map
Threshold Cornerness Map
Thresholded Map
Non-maximal Suppression
Corners


Apply Corner Operator
Threshold Cornerness Map
Non-maximal Suppression
Corners superimposed on Input Image

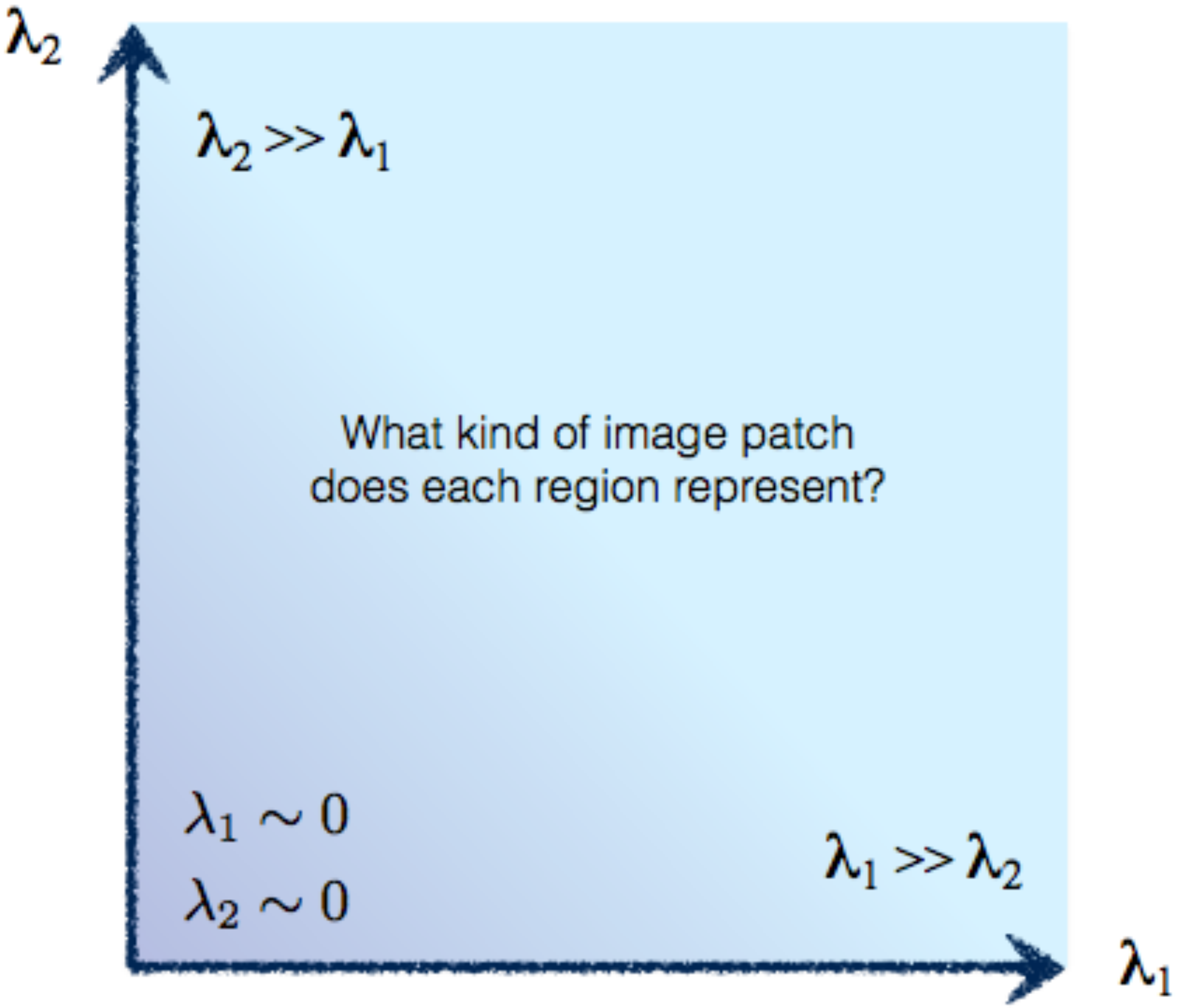
Cantos de Harris

- Matriz de covariância dos gradientes de vizinhança

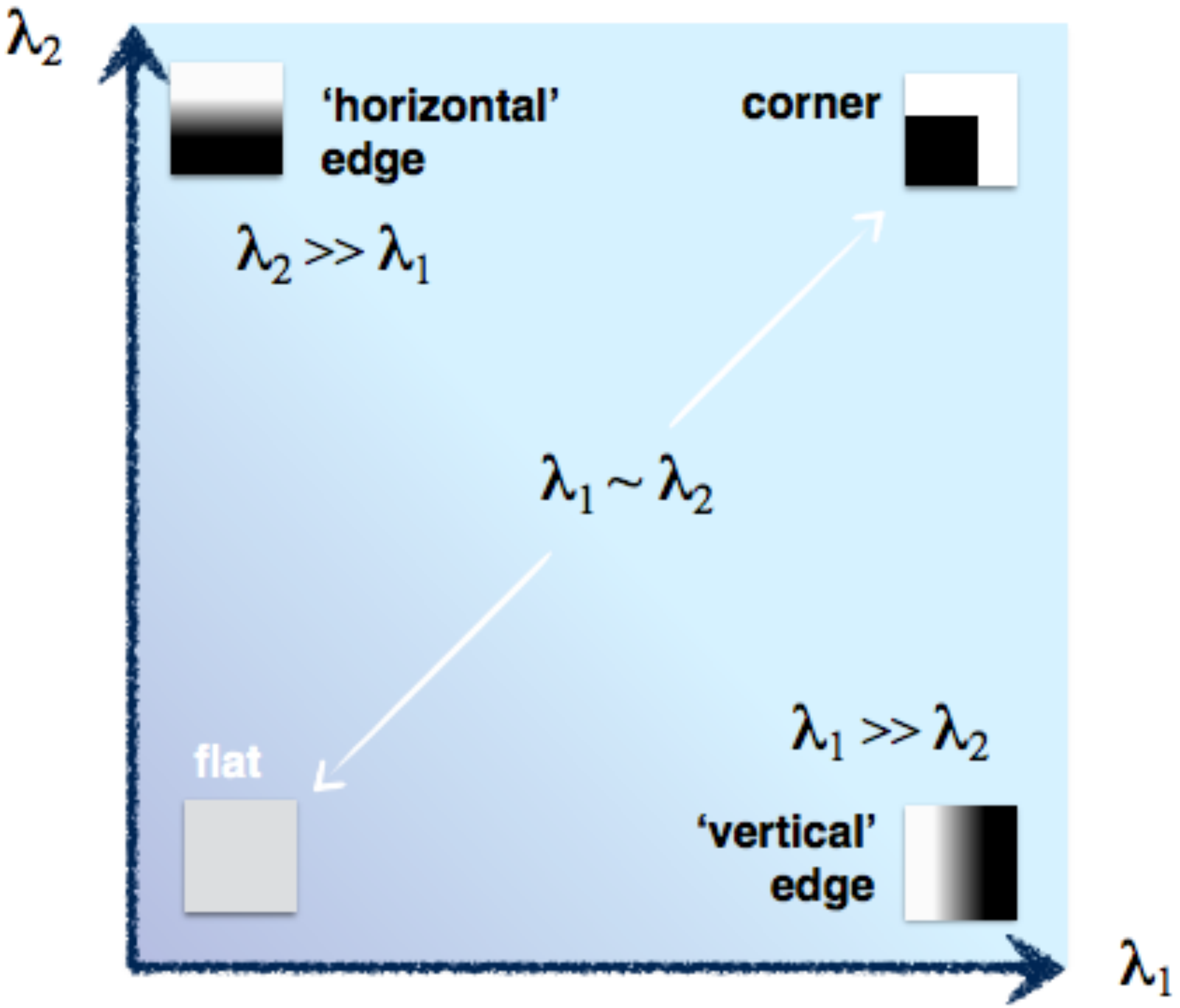
$$M(p) = \begin{bmatrix} \sum_{N(p)} {G_x}^2 & \sum_{N(p)} G_x G_y \\ \sum_{N(p)} G_x G_y & \sum_{N(p)} {G_y}^2 \end{bmatrix}$$

- Autovalores (alfa) e (beta) de M

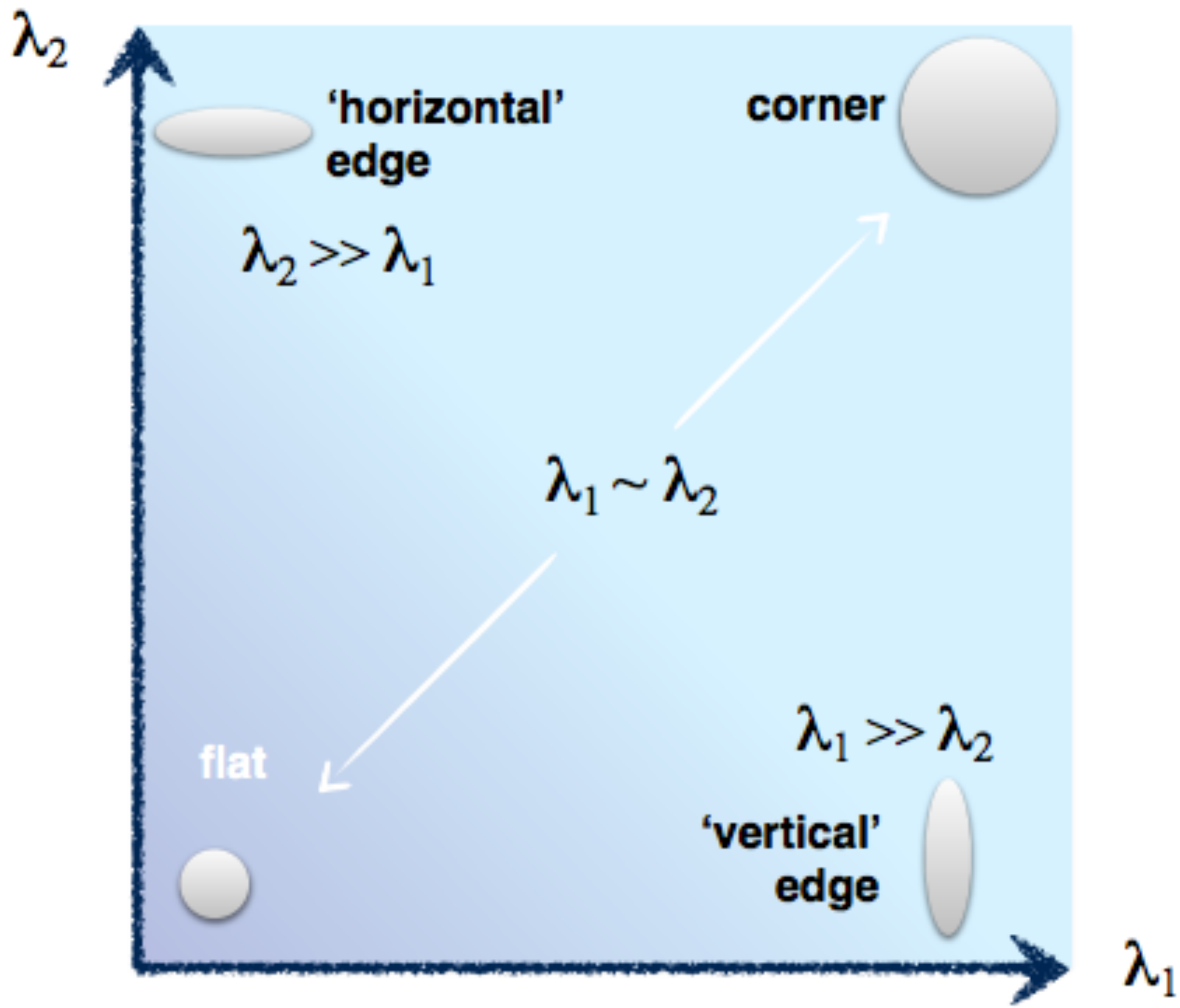
Interpretando autovalores



Interpretando autovalores



Interpretando autovalores



Cantos de Harris

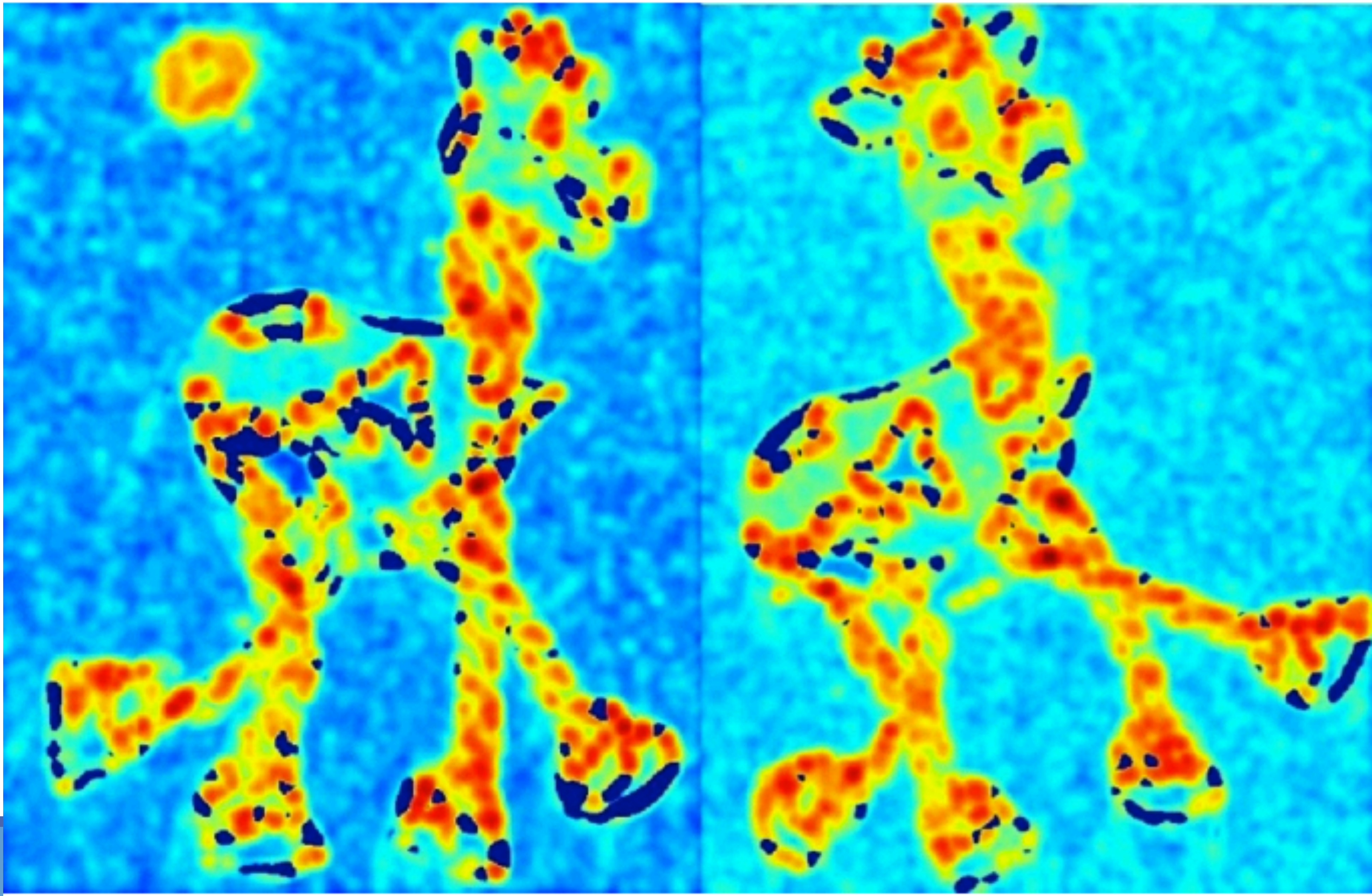
- Cantos: α e β grandes
- $\mathrm{Det}\big(M(p)\big) = \alpha \cdot \beta$
- $\mathrm{Tr}\big(M(p)\big) = \alpha + \beta$

- Função de resposta:

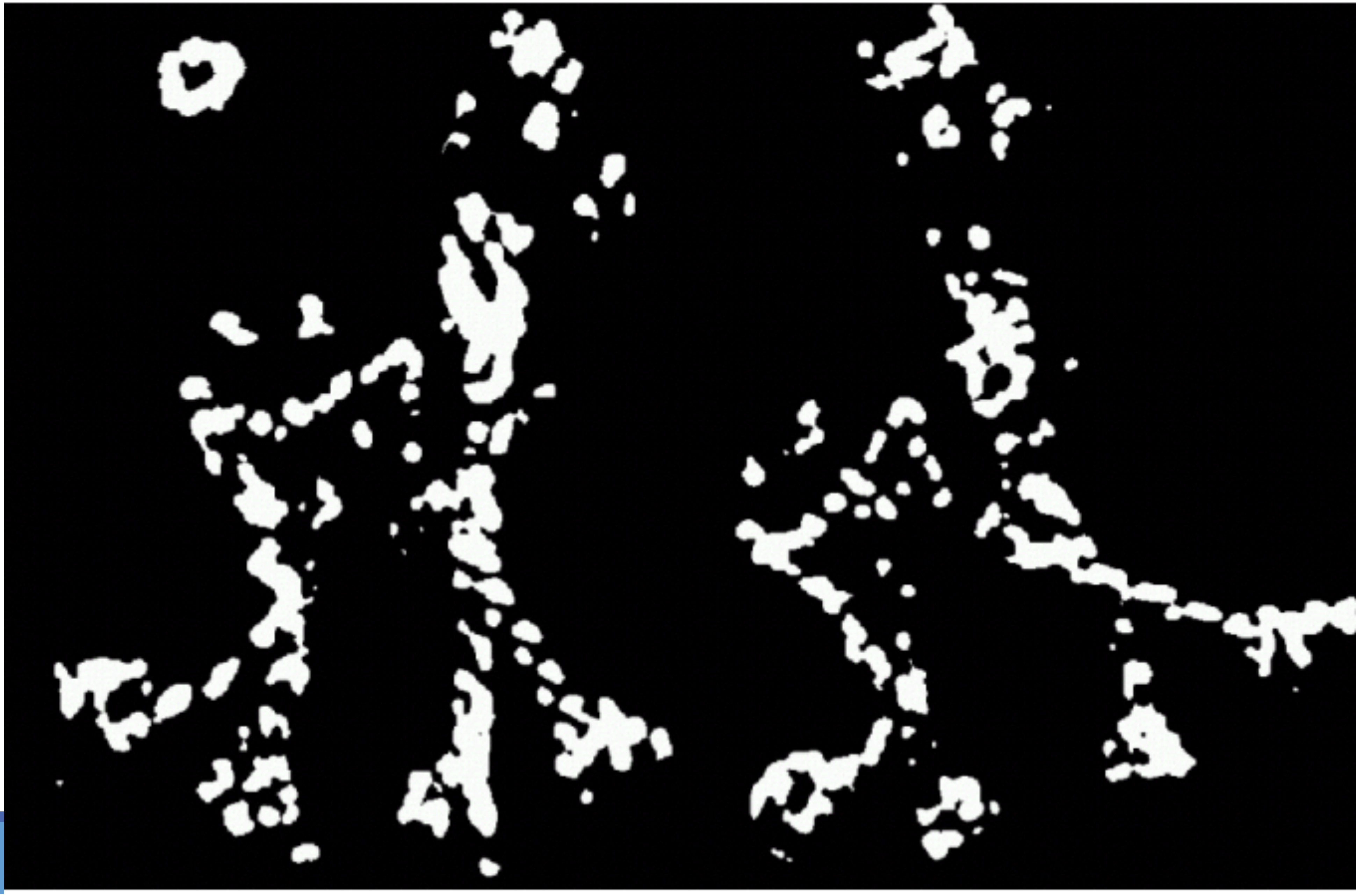
$$R(p) = \mathrm{Det}\big(M(p)\big) - k \cdot \mathrm{Tr}\big(M(p)\big)^2$$

- k: parâmetro ajustável
  - Valores empíricos: 0.04– 0.15

Corner response

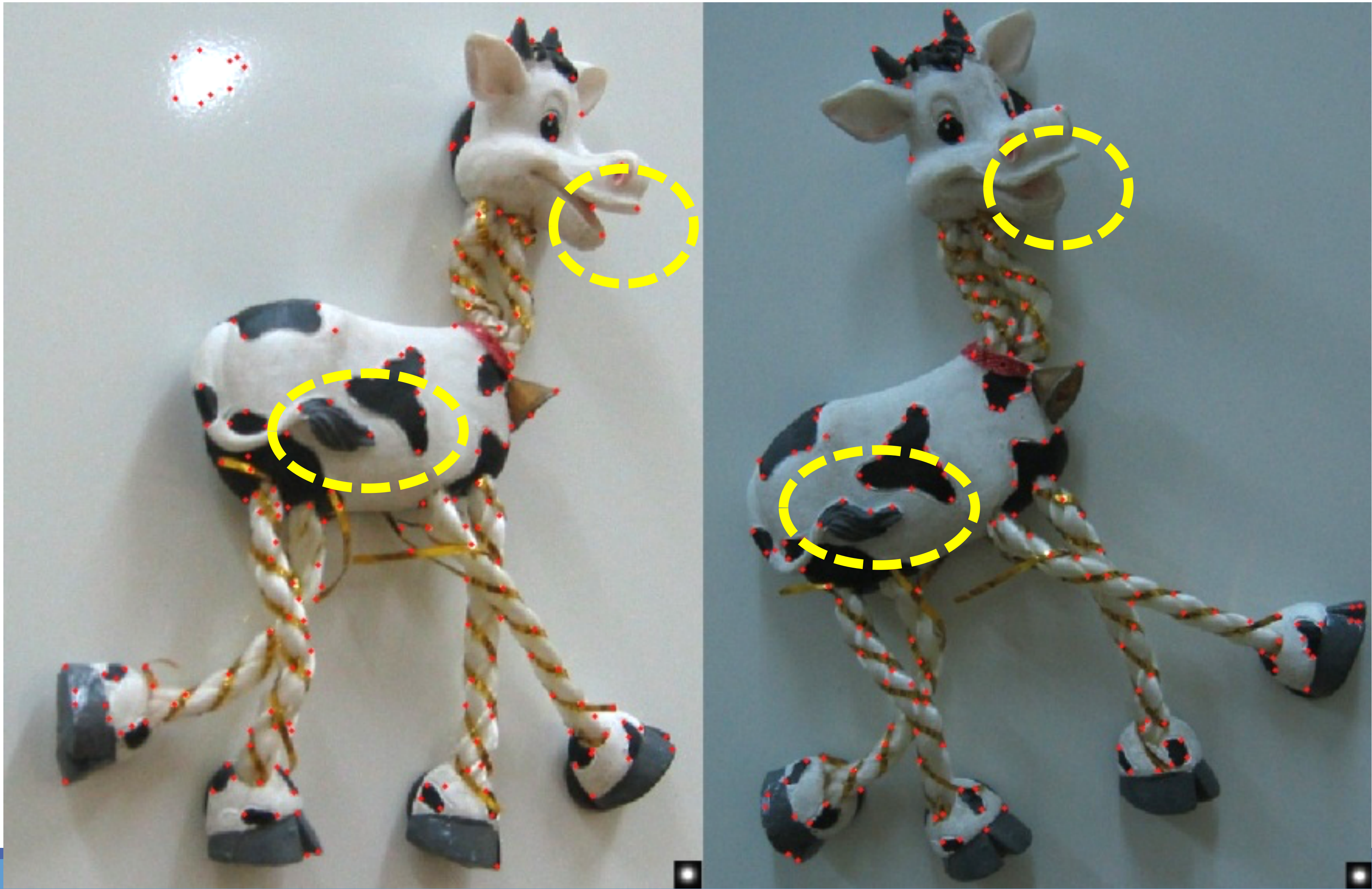


Thresholded corner response



Non-maximal supression

Harris Detector: Steps



Harris

Propriedades do detector de cantos

Invariante à rotação?

Invariante à escala?

Propriedades do detector de cantos de Harris

Invariante à rotação? 

Invariante à escala?

Invariante à rotação? 

Invariante à escala? 

Borda! 

Canto! 

GFTT (Good Features to Track)

- Variação no detector de Harris

$$R = min(\lambda_1, \lambda_2)$$

- Ao invés de

$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2$$

- Cantos mais estáveis
- Mais rápido
- Pouca diferença visual

GFTT

MSER

- Maximally Stable Extremal Regions
- Objetivo é identificar ”blobs” com as características:
- (1) possuam componentes conexos
- (2) intensidade dos pixels uniforme
- (3) contraste com o fundo

MSER

- ▷Computa vários limiares e observa as formas que não mudam entre os limiares

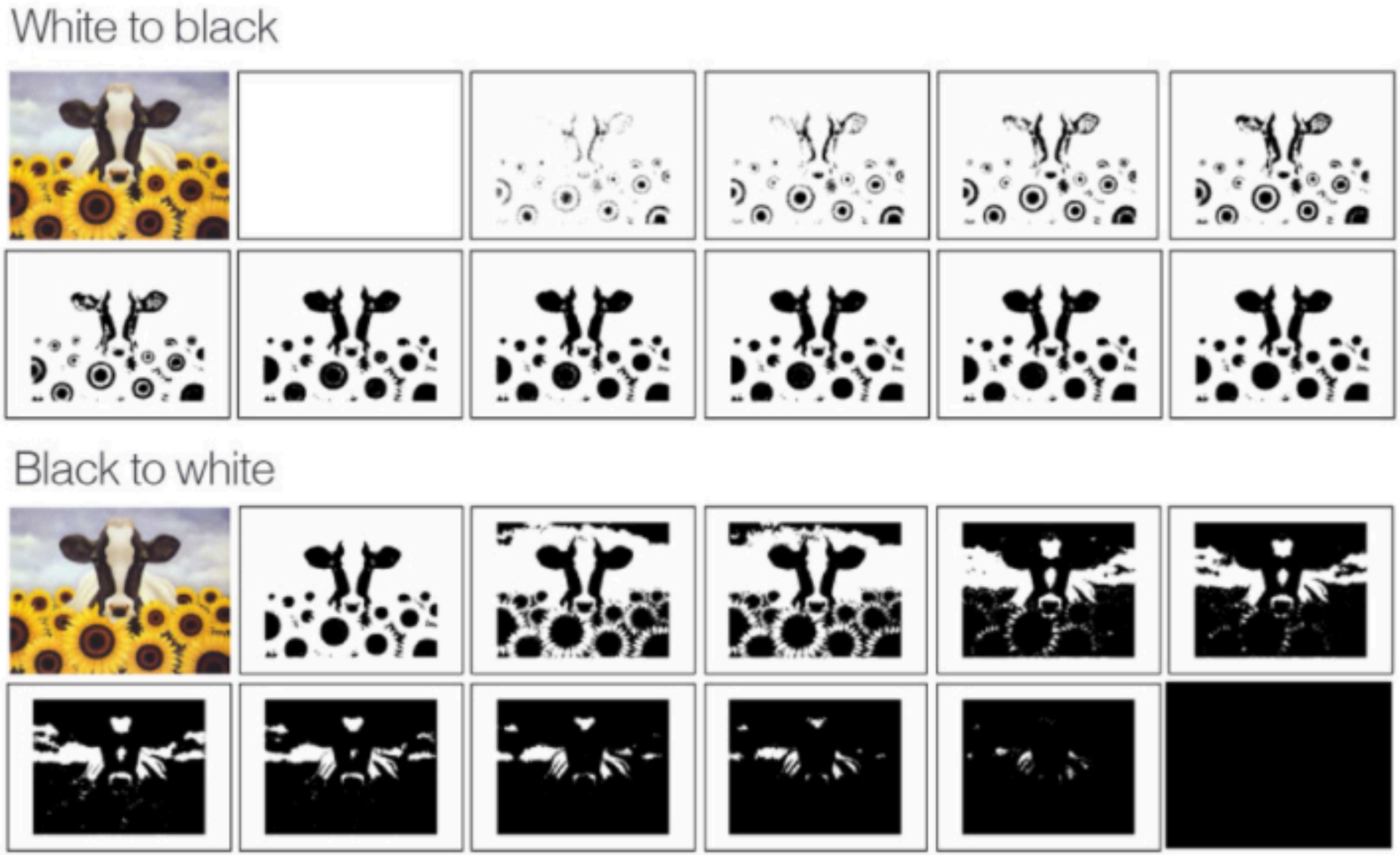
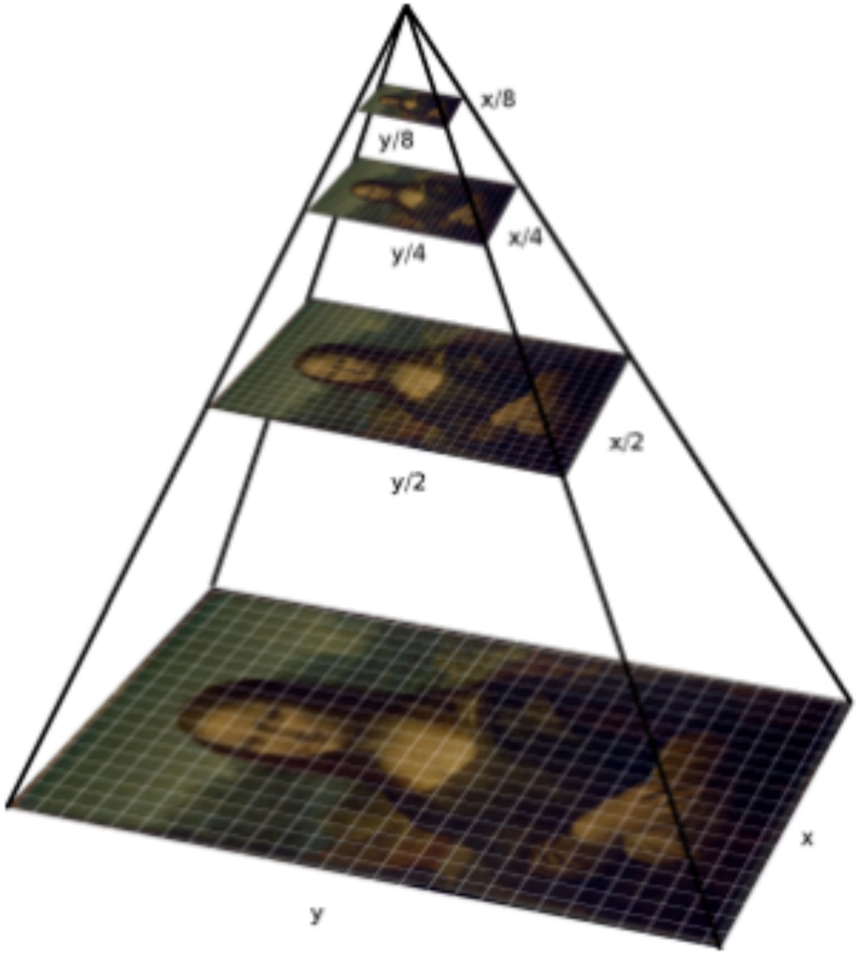


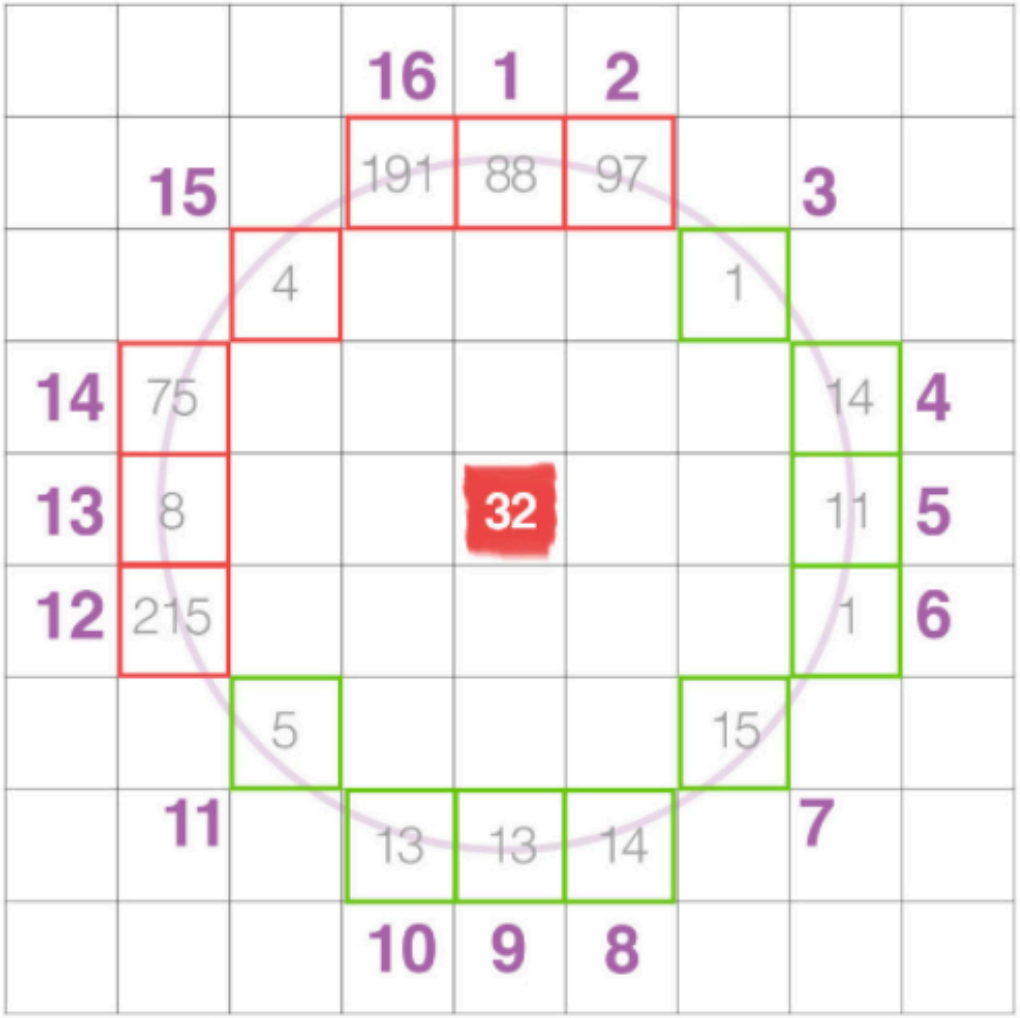
FIGURE 1: EXAMPLE OF DETECTING MSER REGIONS IN AN IMAGE. NOTICE THAT EACH OF THESE INDIVIDUAL REGIONS HAVE: (1) SIMILAR COLOR DISTRIBUTIONS, (2) ARE SURROUNDED BY CONTRASTING COLOR, AND (3), SINCE THEY HAVE SIMILAR COLOR DISTRIBUTIONS, ARE "SIMILAR" AND "CONNECTED" TO THE SEMANTIC CONCEPT OF THE REGION (I.E. THE WINDOWS OR DOOR).


Images
MICHAEL
CRICHTON
NEXT
MICHAEL
CRICHTON
NEXT

BRISK

▷FAST + Multiscale






Images
MICHAEL
CRICHTON
NEXT

```python
import numpy as np
import cv2

image = cv2.imread('obama.jpeg')
gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)

#BRISK
brisk = cv2.BRISK_create()
(kps, descs) = brisk.detectAndCompute(gray, None)
print("# kps: {} - size: {}".format(len(kps),descs.shape) )

img2 = cv2.drawKeypoints(image, kps, None, color=(255,0,0))
cv2.imshow("imagem", np.hstack([image, img2]))
cv2.waitKey(0)
```

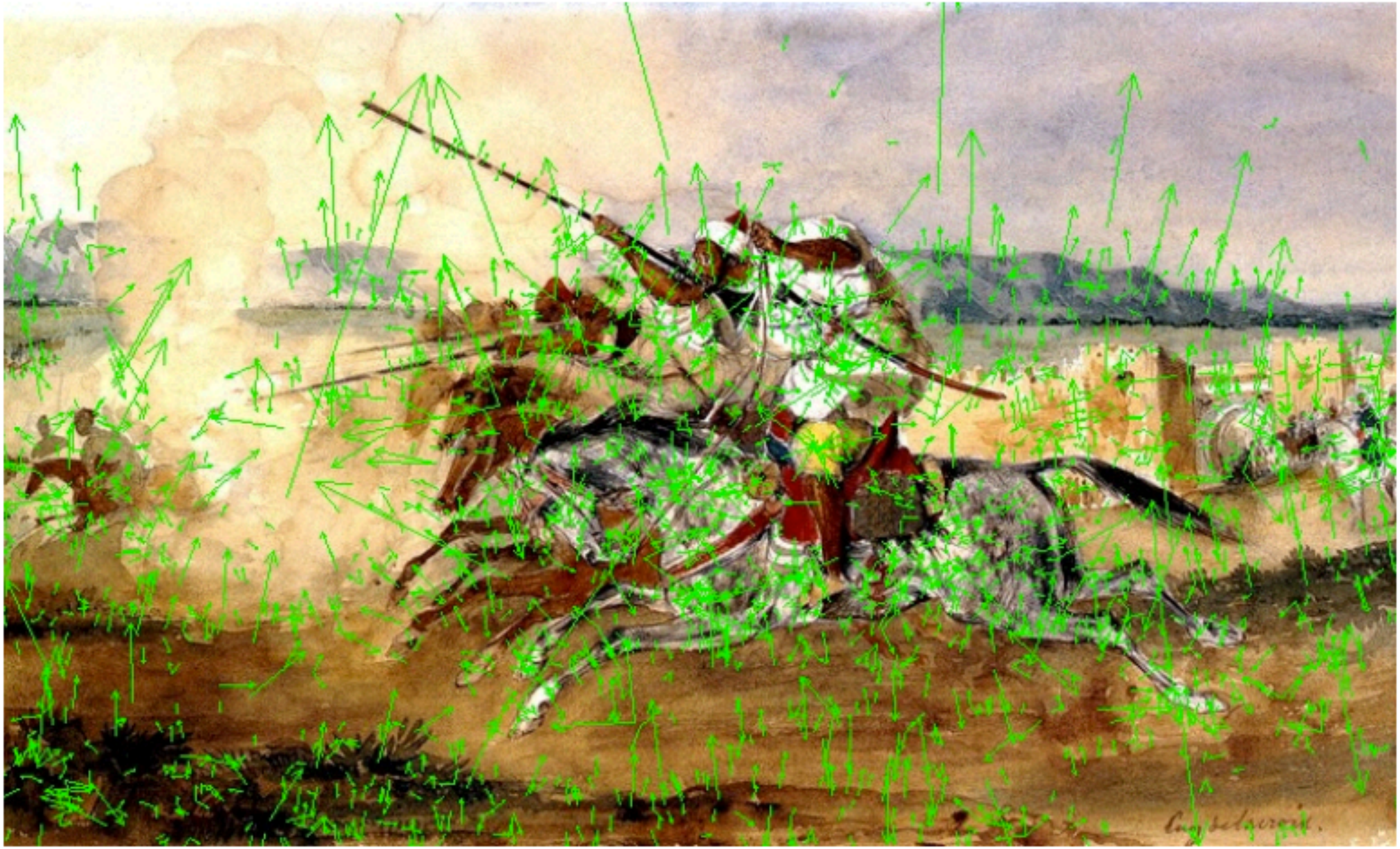
ORB (Oriented Fast and Rotated Brisk)

- Rublee et al. (2011)

- BRISK + Rotation Invariant
  - Rotaciona patch para o eixo dominante


Images
MICHAEL
CRICHTON
NEXT

```python
import numpy as np
import cv2

image = cv2.imread('obama.jpeg')
gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)

# Initiate ORB detector
orb = cv2.ORB_create()
# find the keypoints with ORB
kp = orb.detect(gray,None)
# compute the descriptors with ORB
kp, des = orb.compute(gray, kp)
# draw only keypoints location,not size and orientation
img2 = cv2.drawKeypoints(image, kp, None, color=(0,255,0), flags=0)
cv2.imshow("imagem", np.hstack([image, img2]))
cv2.waitKey(0)
```

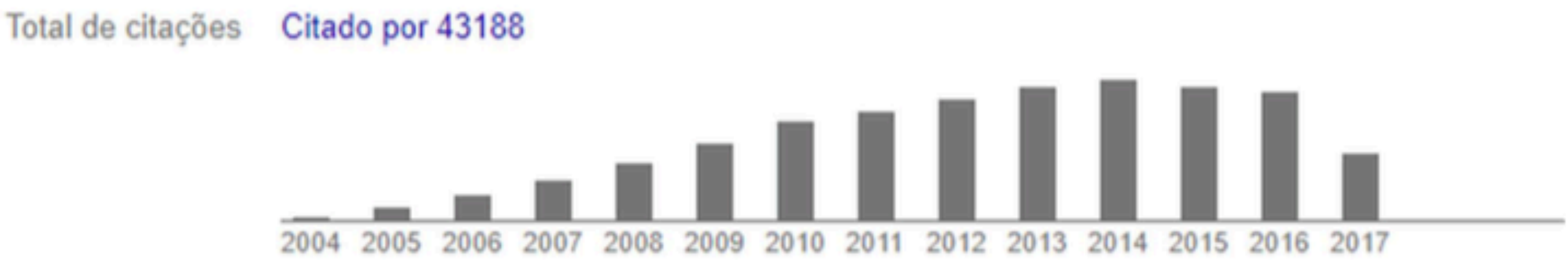


SIFT

SIFT

"Scale Invariant Feature Transform" [5]

International Journal of Computer Vision, 2004

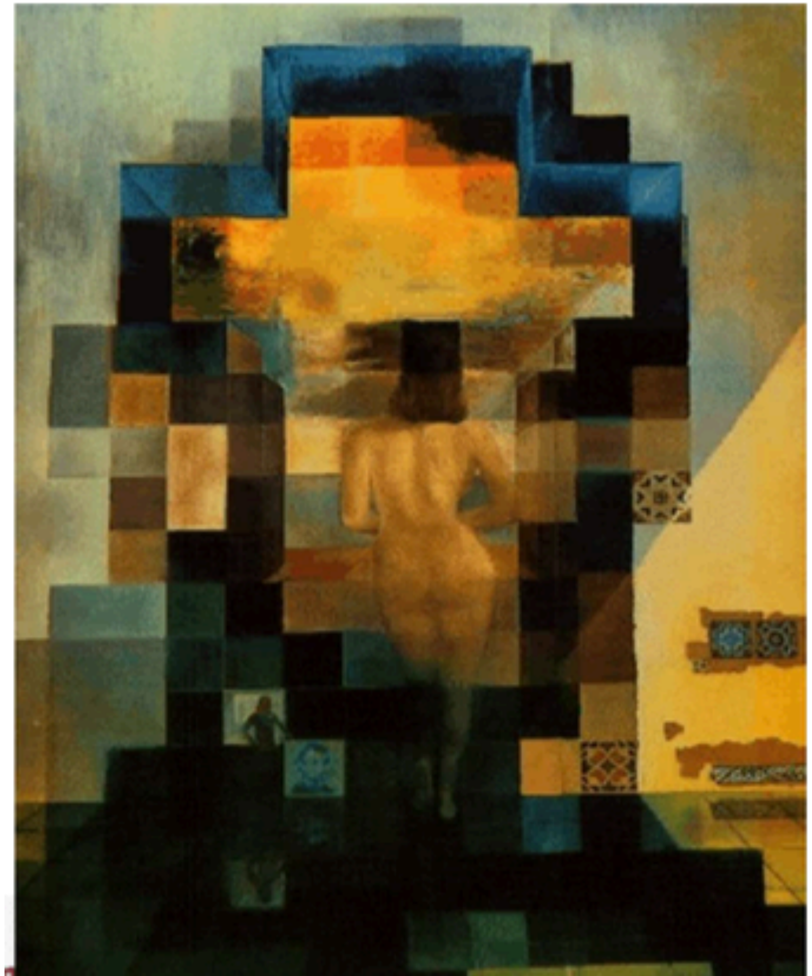


SIFT

- Scale Invariant Feature Transform
  - IJCV, 2004
- Passos
  - Construção do espaço-escala (scale-space)
  - Localização de keypoints
  - Atribuição de orientação
  - Descrição da região de vizinhança de cada keypoint

SIFT

▷ Conceito Scale-Space



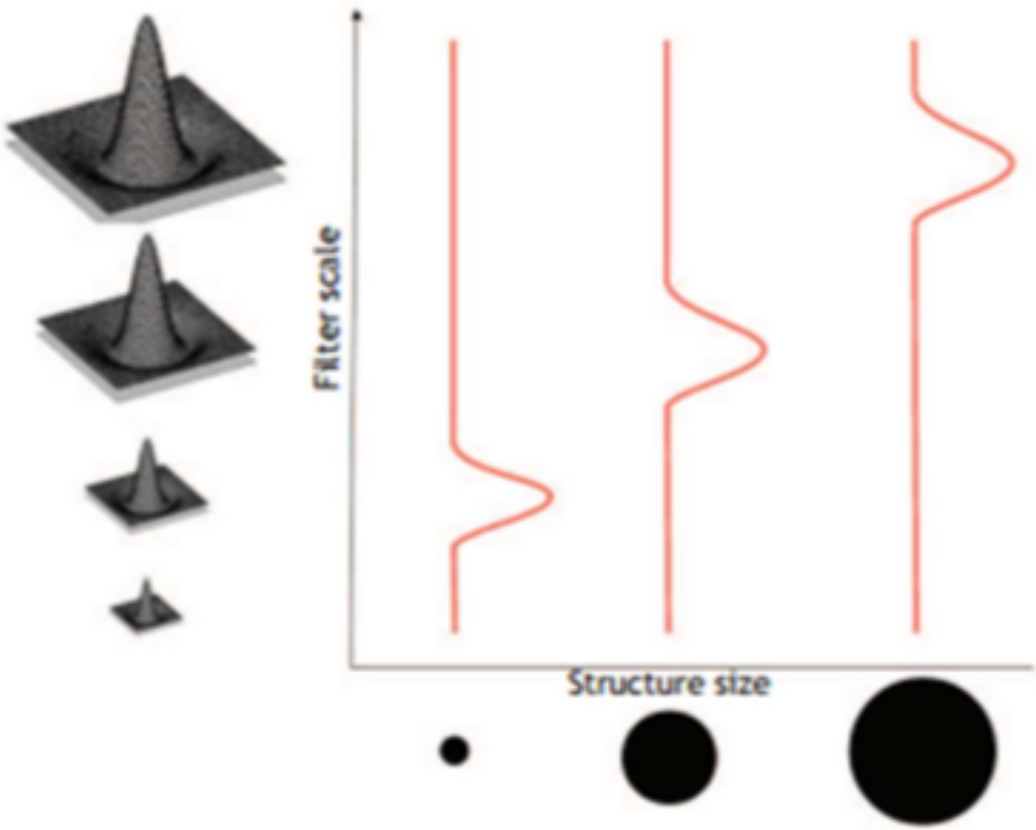
- Percepção é alterada em função da escala analisada;
- Não se sabe *a priori* que escala olhar

SIFT

- Solução: considerar múltiplas escalas
  - Sistema visual humano já faz isso...

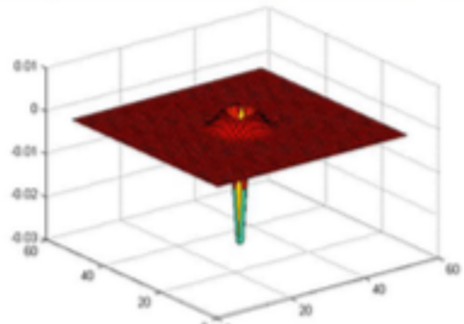
SIFT

- Scale-space – Estruturas (círculos) de diferentes tamanhos são detectadas em diferentes escalas (filtros de diferentes tamanhos)



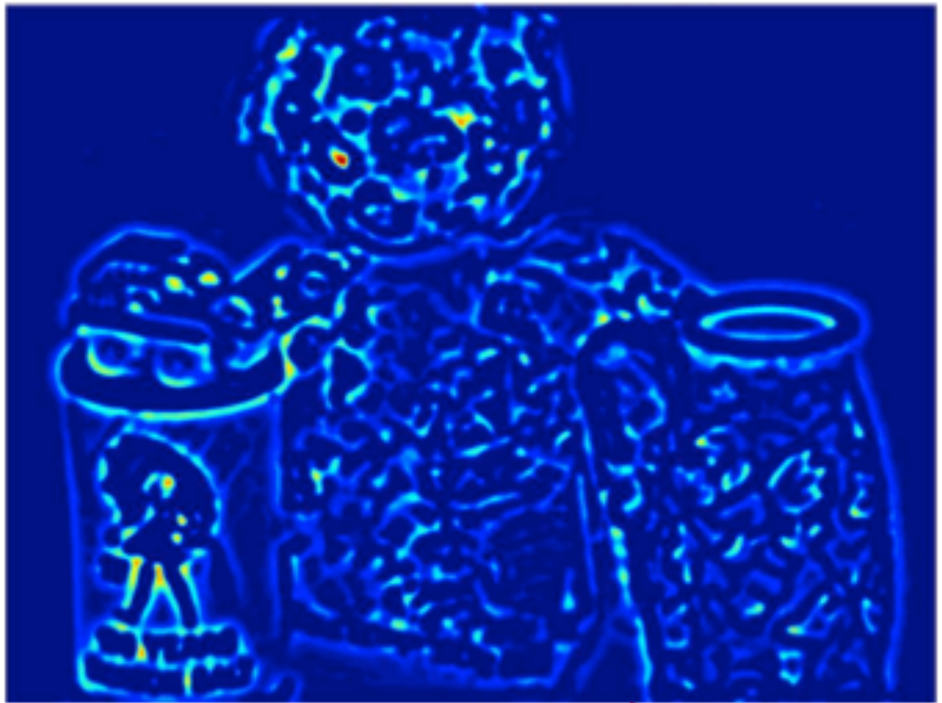
SIFT

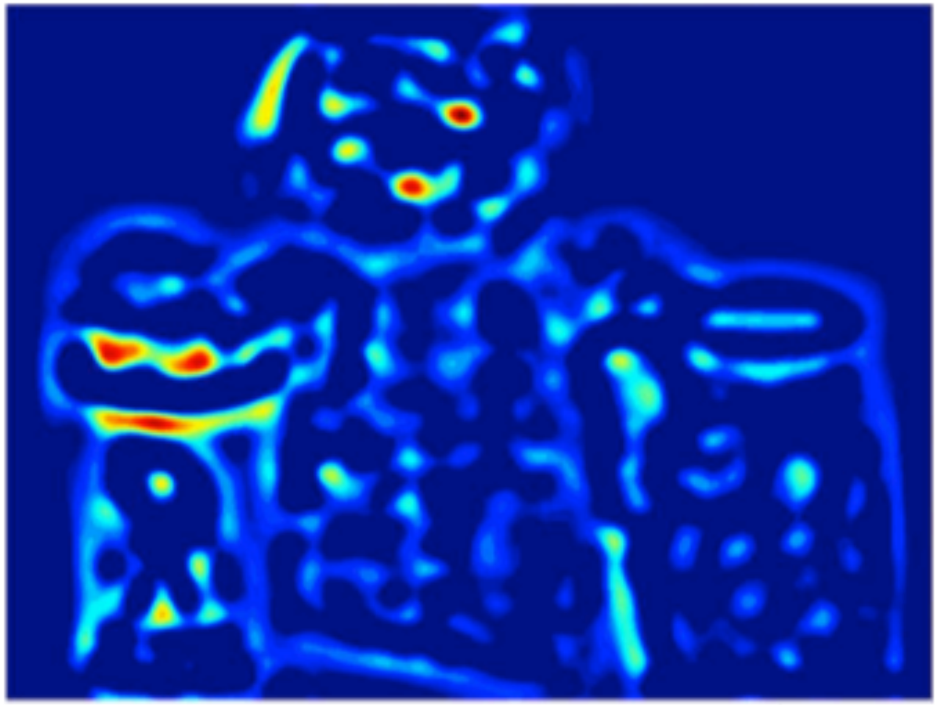
- Scale-space – Estruturas (círculos) de diferentes tamanhos são detectadas em diferentes escalas (filtros de diferentes tamanhos)



SIFT

▷Scale-space – Estruturas (círculos) de diferentes tamanhos são detectadas em diferentes escalas;





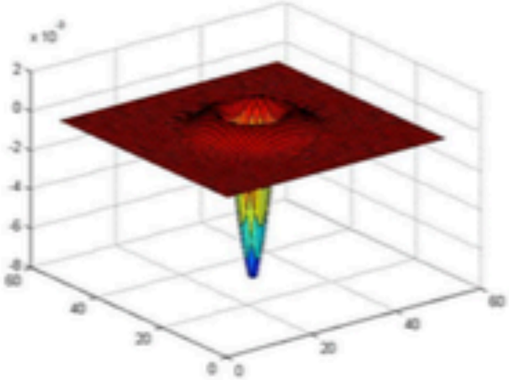
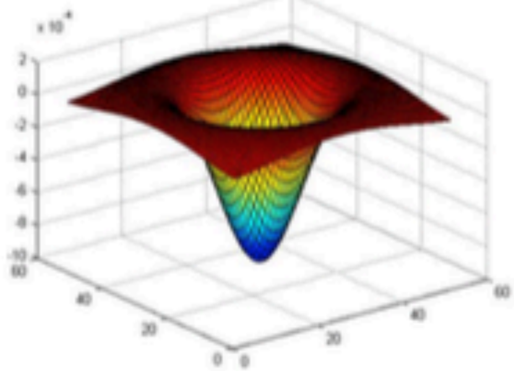


*Imagem original com tamanho reduzido pela metade



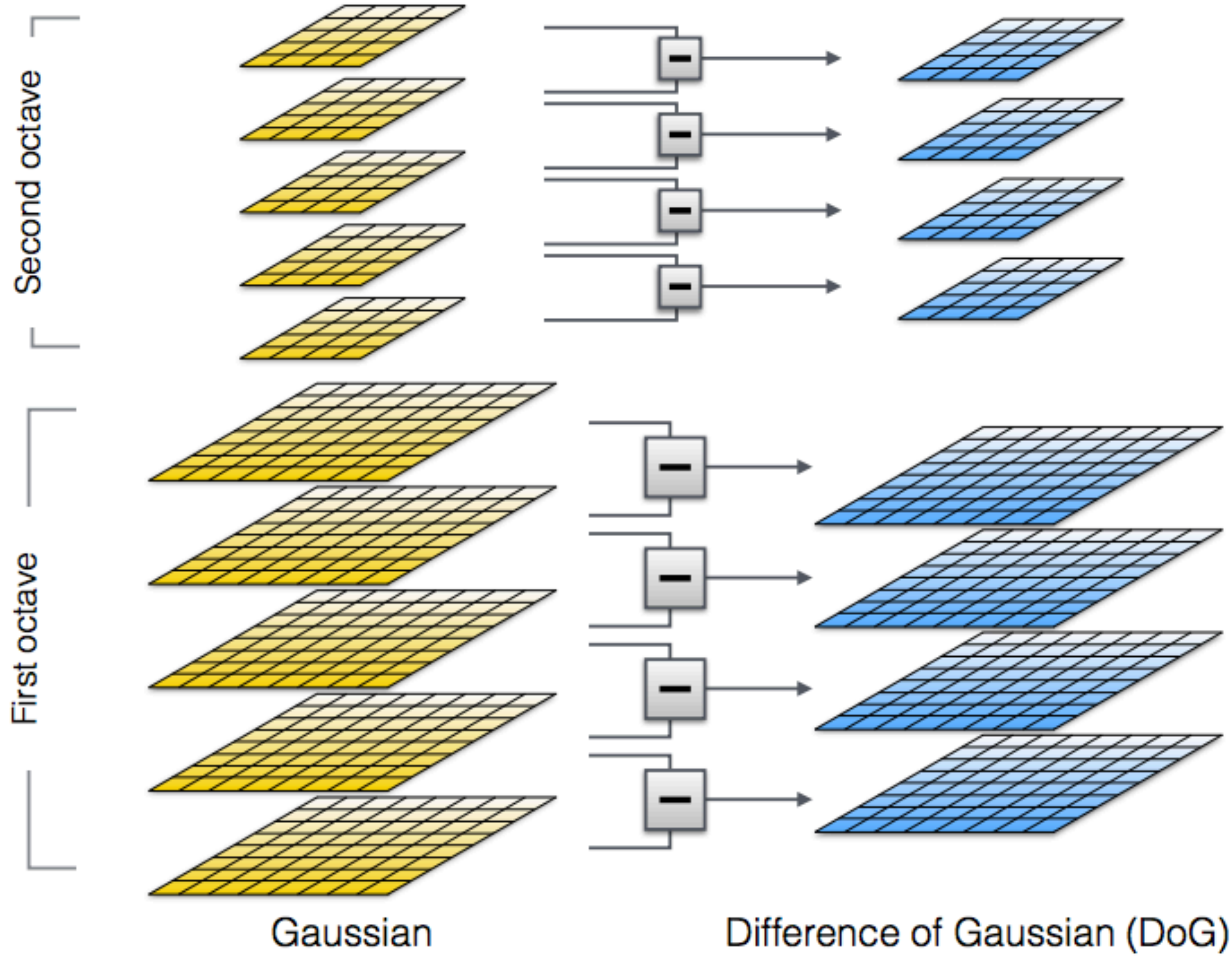
SIFT

- Exemplo detecção de keypoints

Tamanho do círculo: escala do keypoint

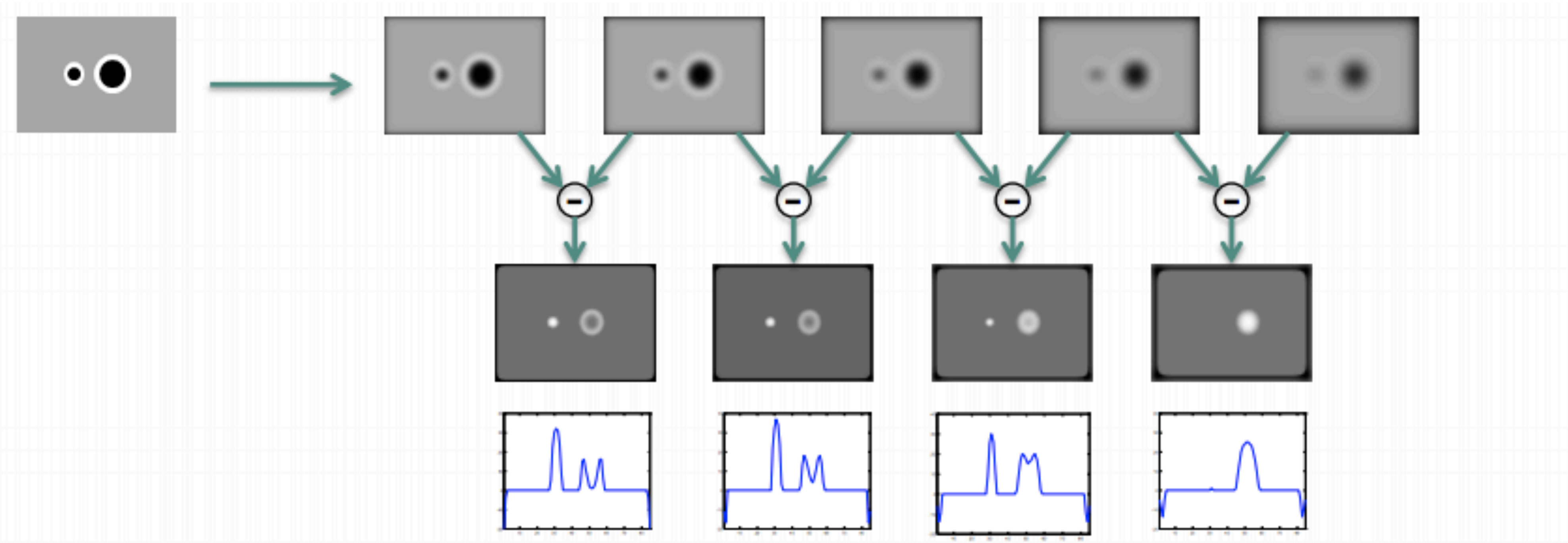
Orientação raio: direção do keypoint

1. Multi-scale extrema detection

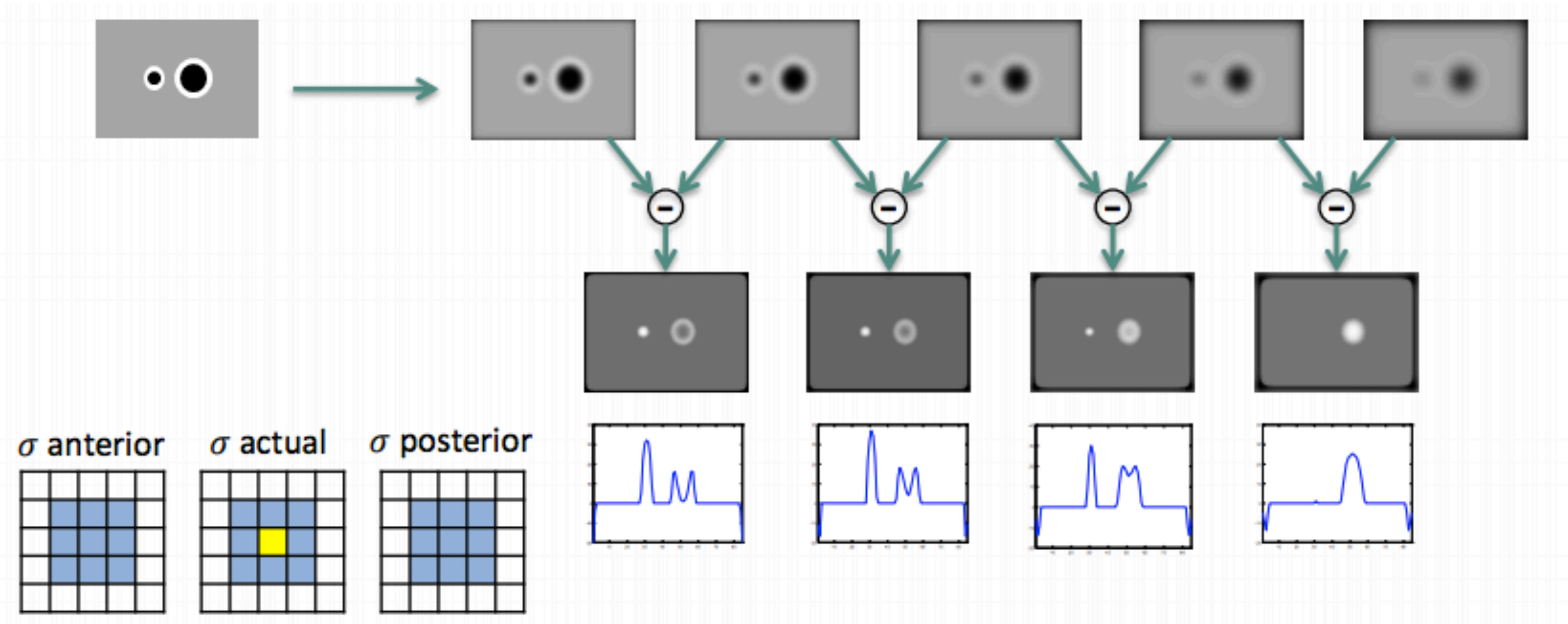


Diferença de Gaussianas

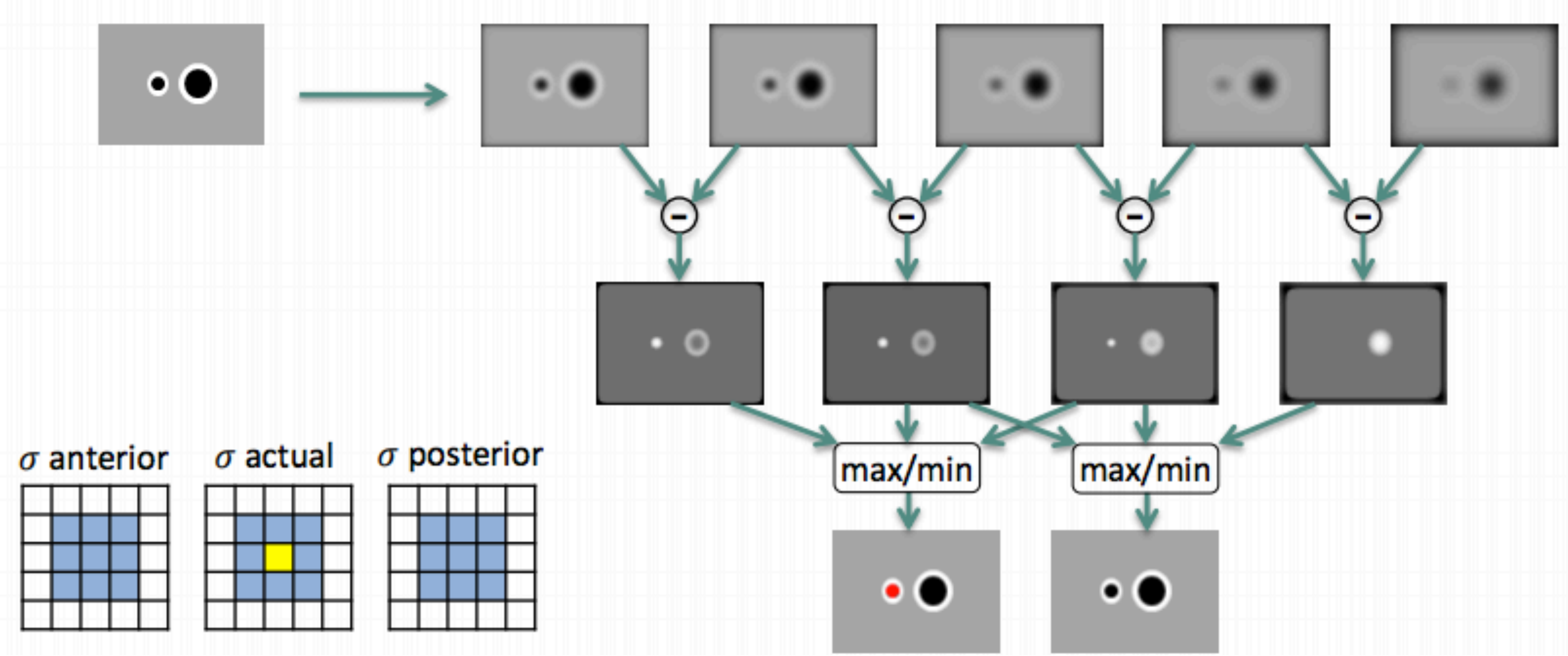
▷O desvio padrão do filtro gaussiano determina a resolução (escala) que se detectam as mudanças de intensidade



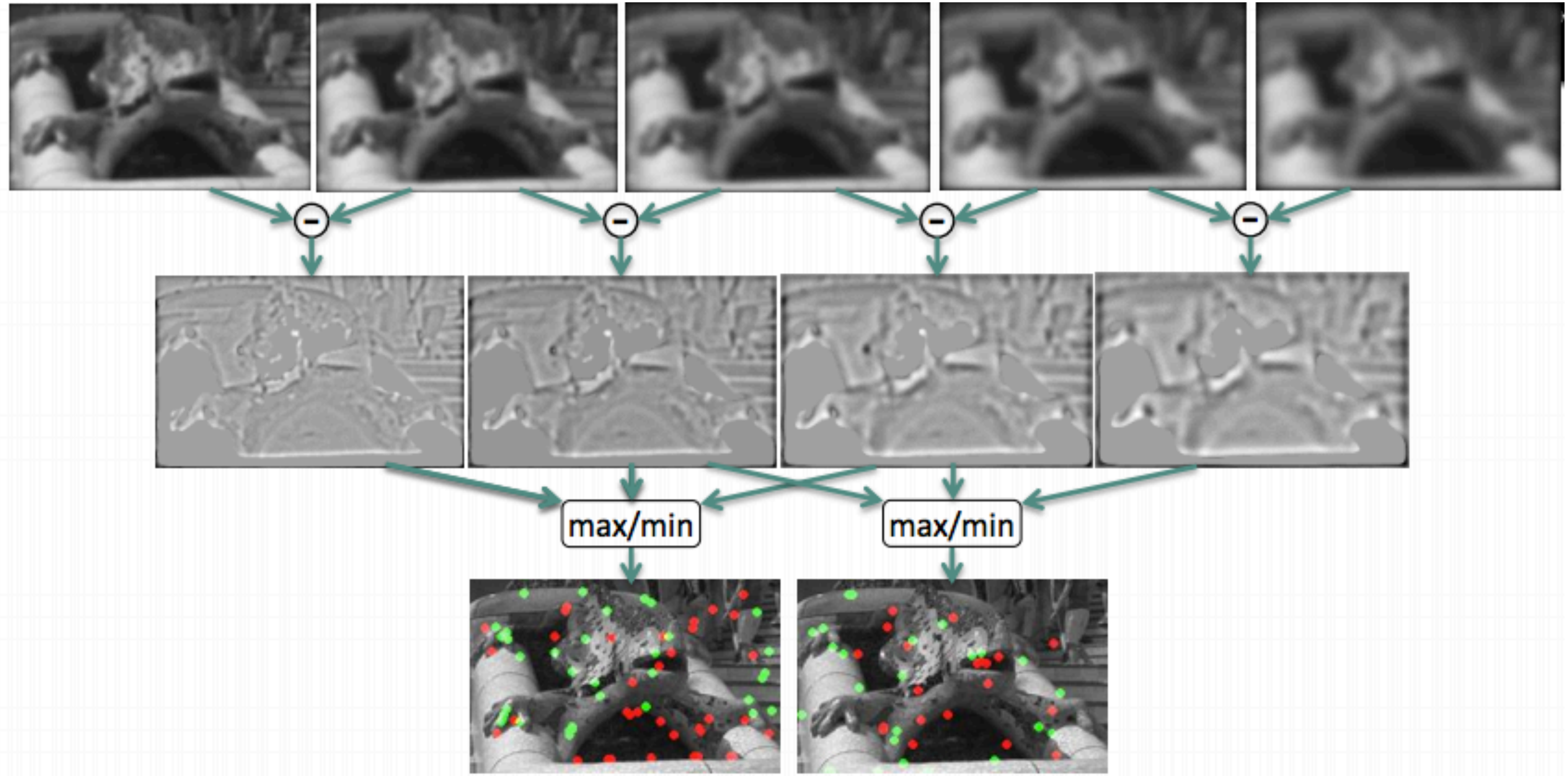
▷Máximos/mínimos locais do DoG em relação ao anterior, atual e posterior



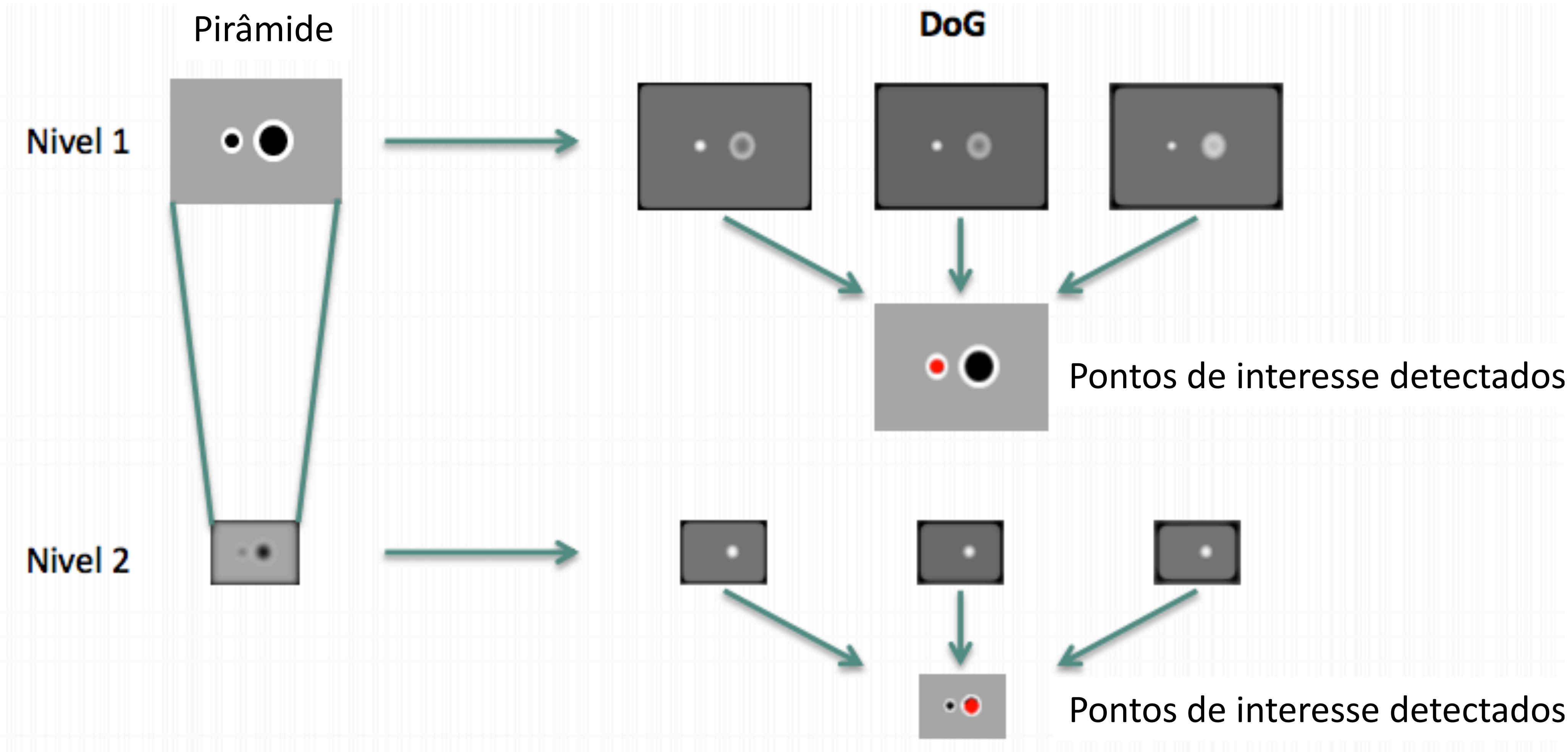
▷ Detecta regiões de maior resposta

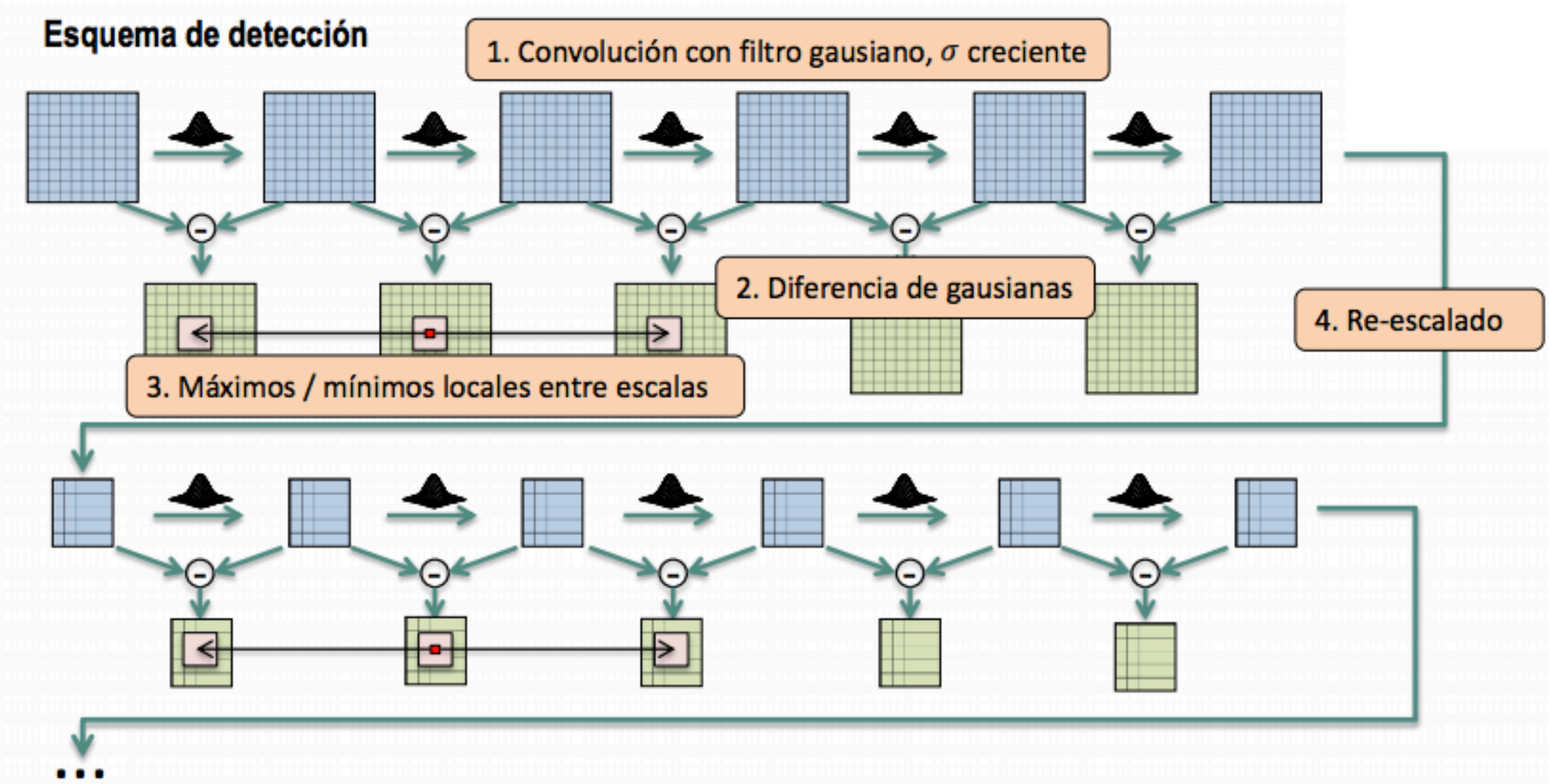


Exemplo



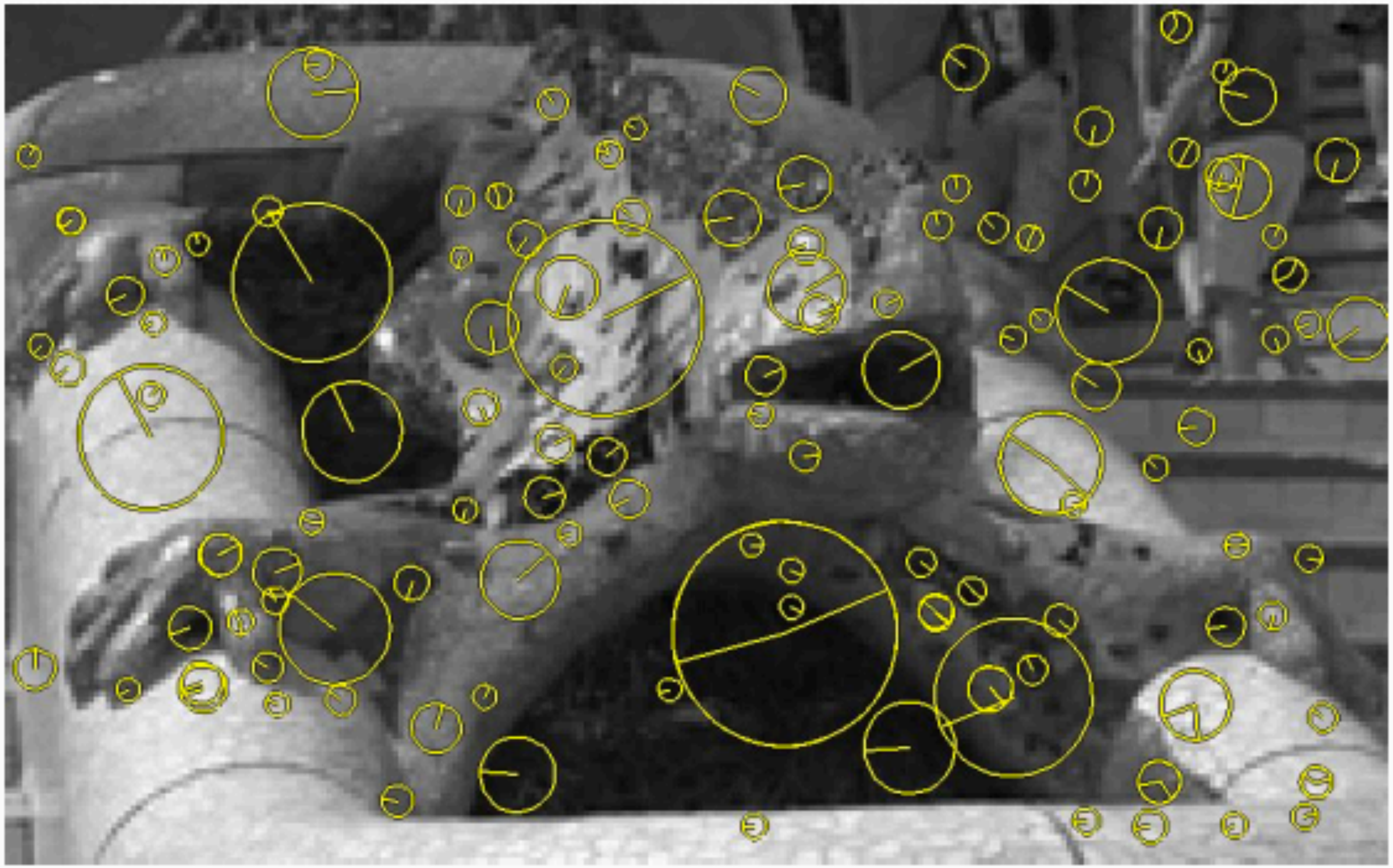
Pirâmides de imagens: exemplo



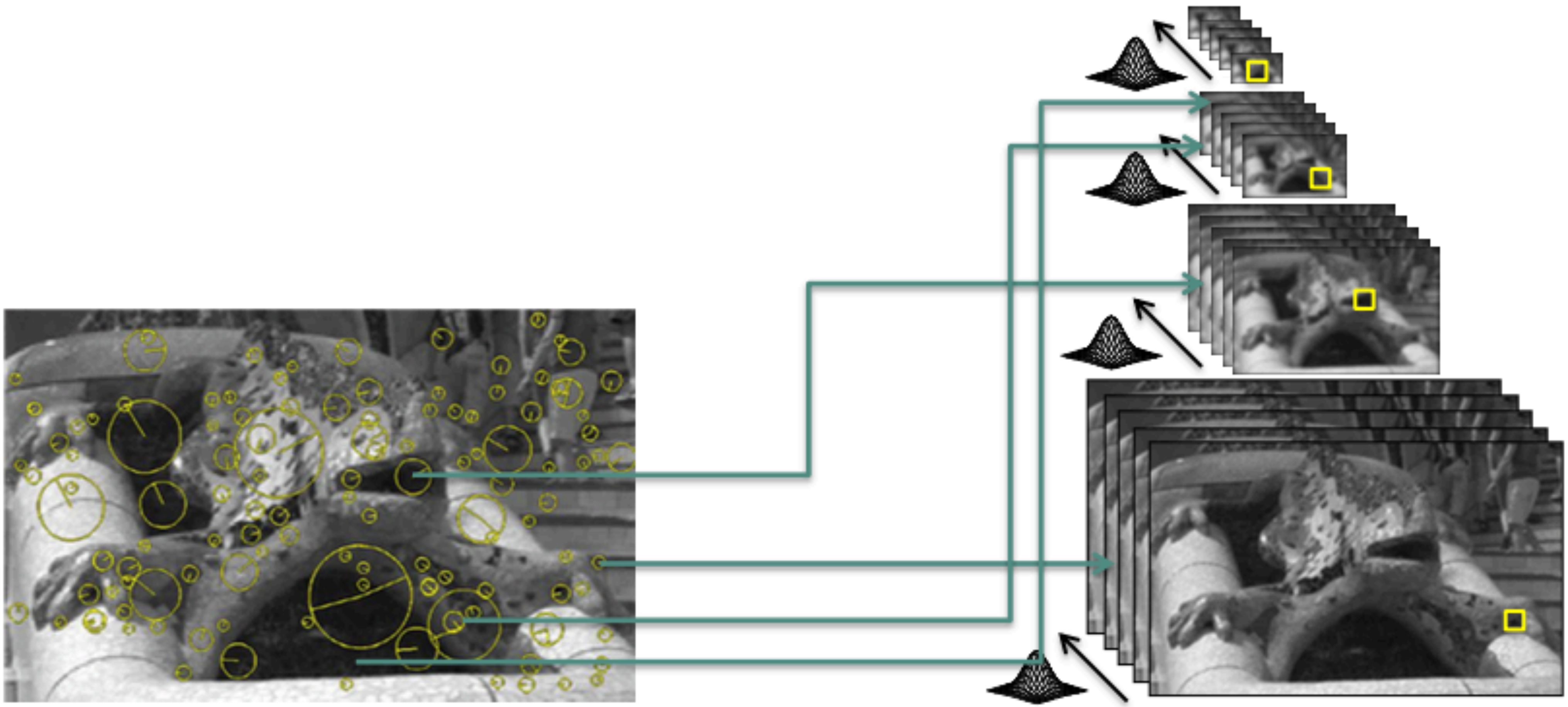


Esquema de detección
1. Convolución con filtro gausiano, σ creciente
2. Diferencia de gausianas
3. Máximos / mínimos locales entre escalas
4. Re-escalado
...

Exemplo de detecção



- Região de tamanho fixo na imagem correspondente na pirâmide
- Níveis baixos: regiões pequenas na imagem original
- Niveis altos: regiões grandes na imagem original



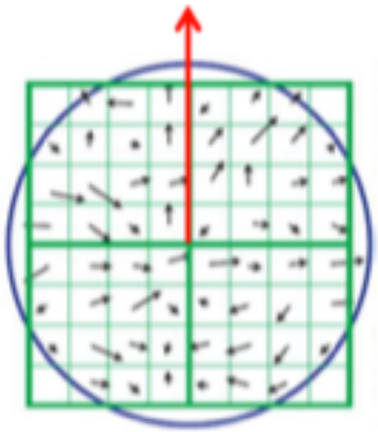
Etapa de descrição

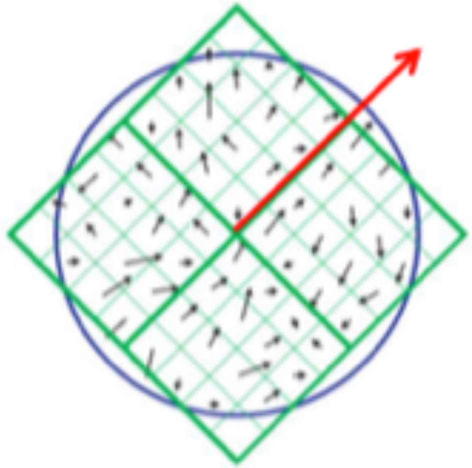
- Para cada keypoint encontrado na etapa de detecção
  - Um vetor será construído para descrever a vizinhança do ponto
    - Descritor

Descritor

▷ Uso de gradientes
- Menor sensibilidade à variações de iluminação

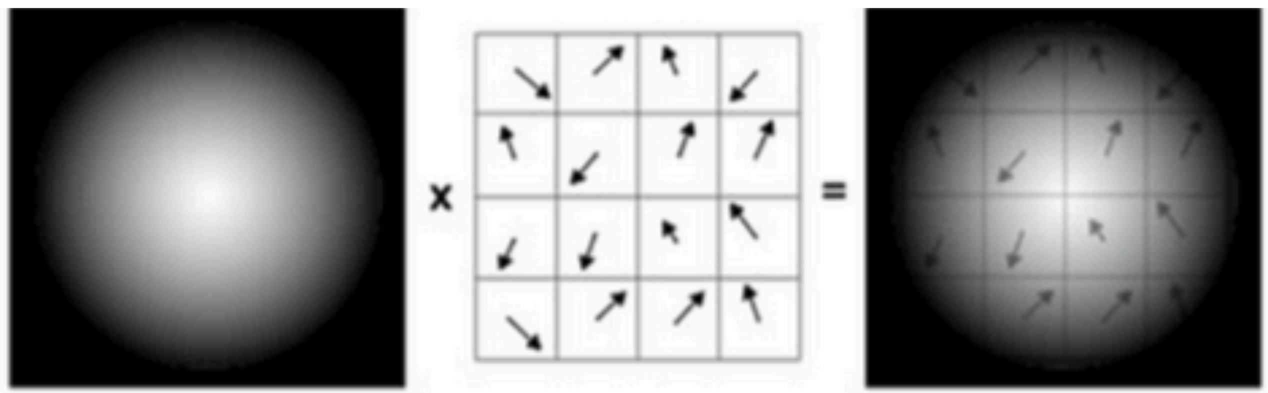
▷ Magnitude e orientação do gradiente ao redor da vizinhança de cada keypoint são computados
- Cálculo realizado em função da orientação detectada
  - Invariância à rotação



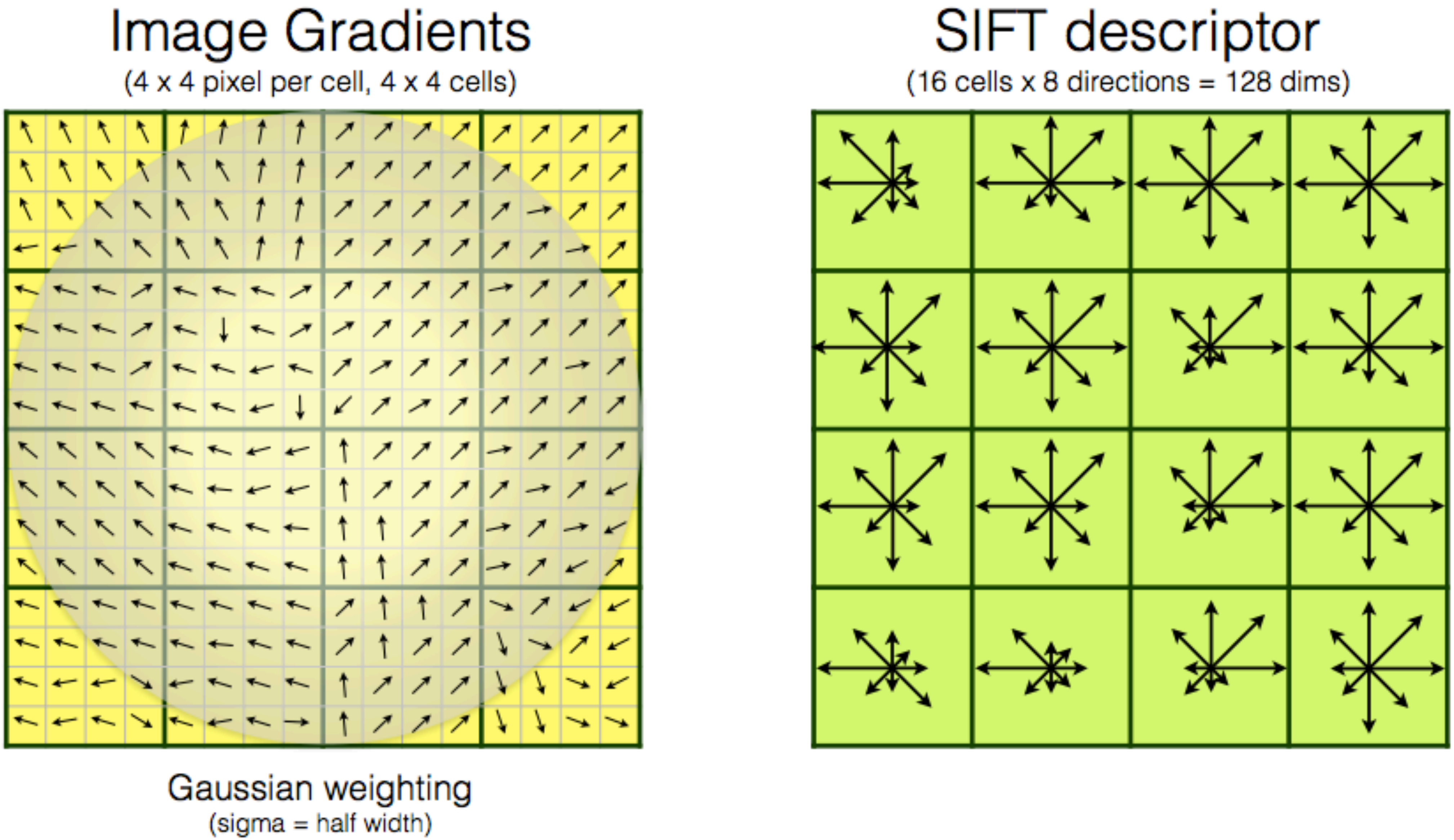


Descritor

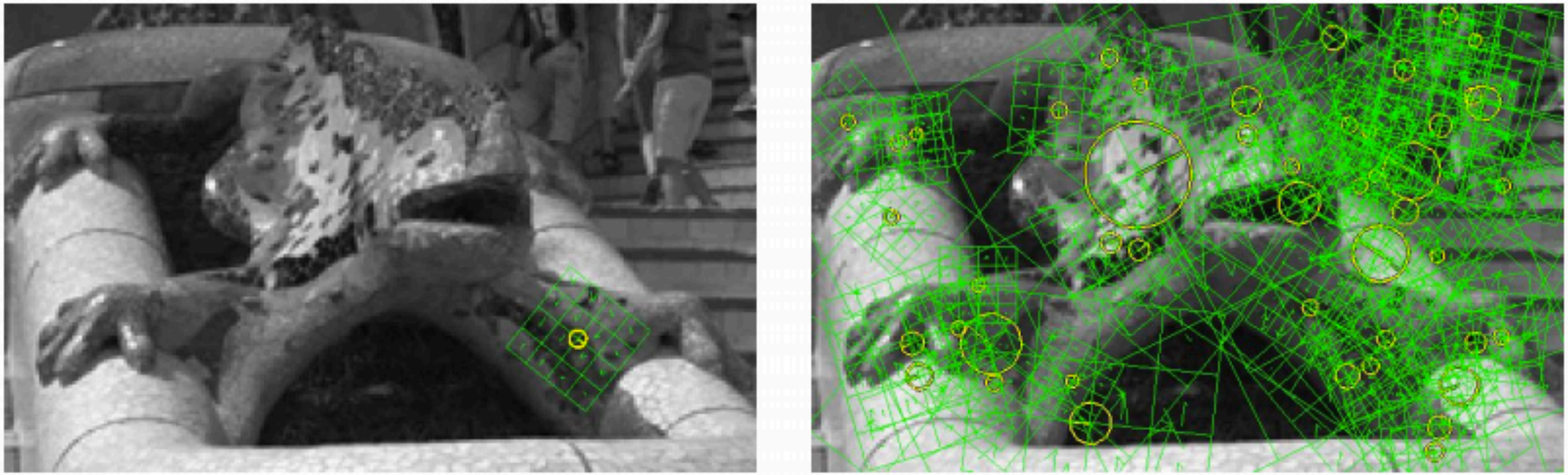
- Histograma de orientações é construído em uma vizinhança de cada keypoint
  - Para cada keypoint, um descritor é formado
  - Valores são pesados em função (i) da magnitude do graiente e de uma (ii) Gaussiana centrada no ponto



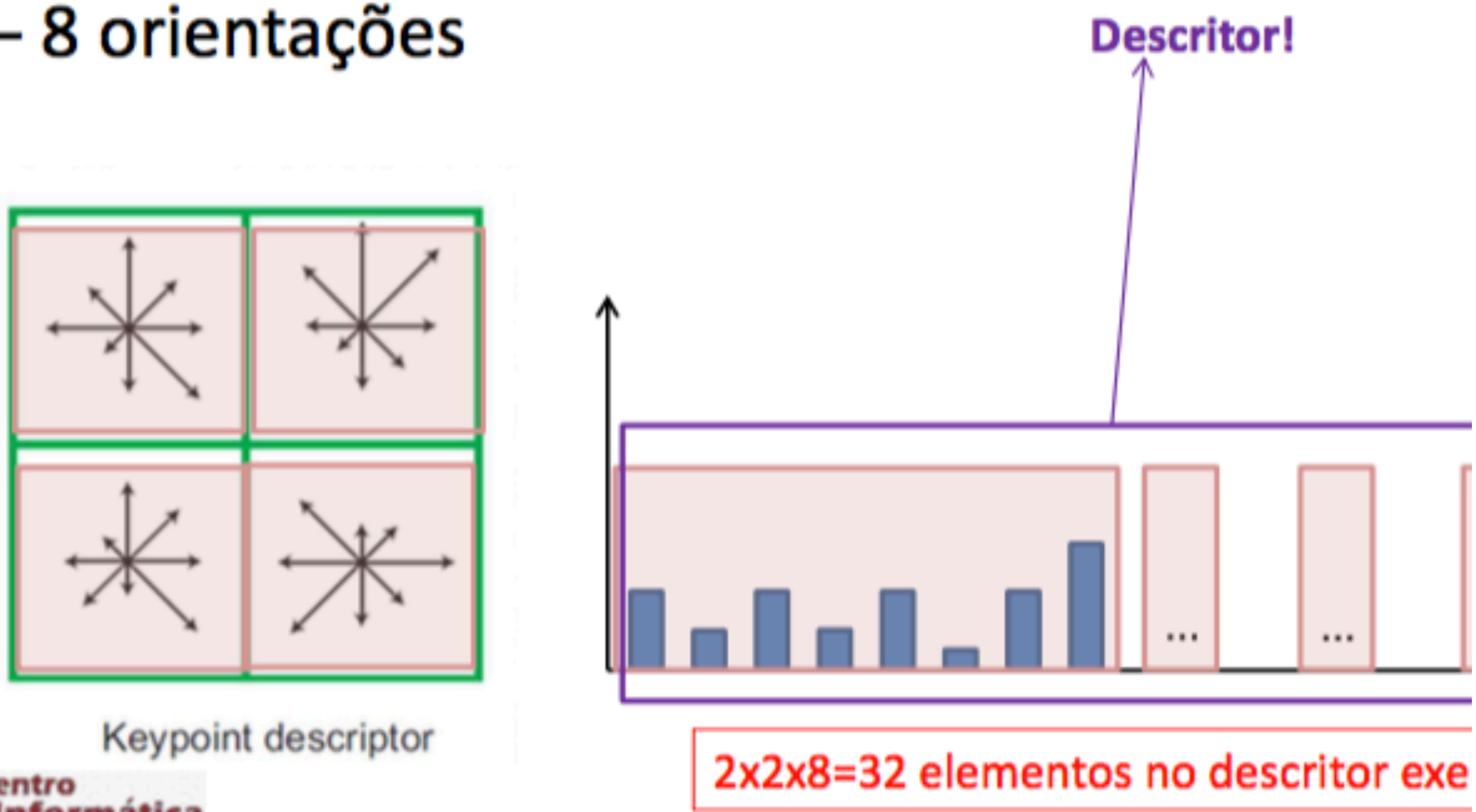
4. Descritor de Pontos Chave



Exemplo do Descritor



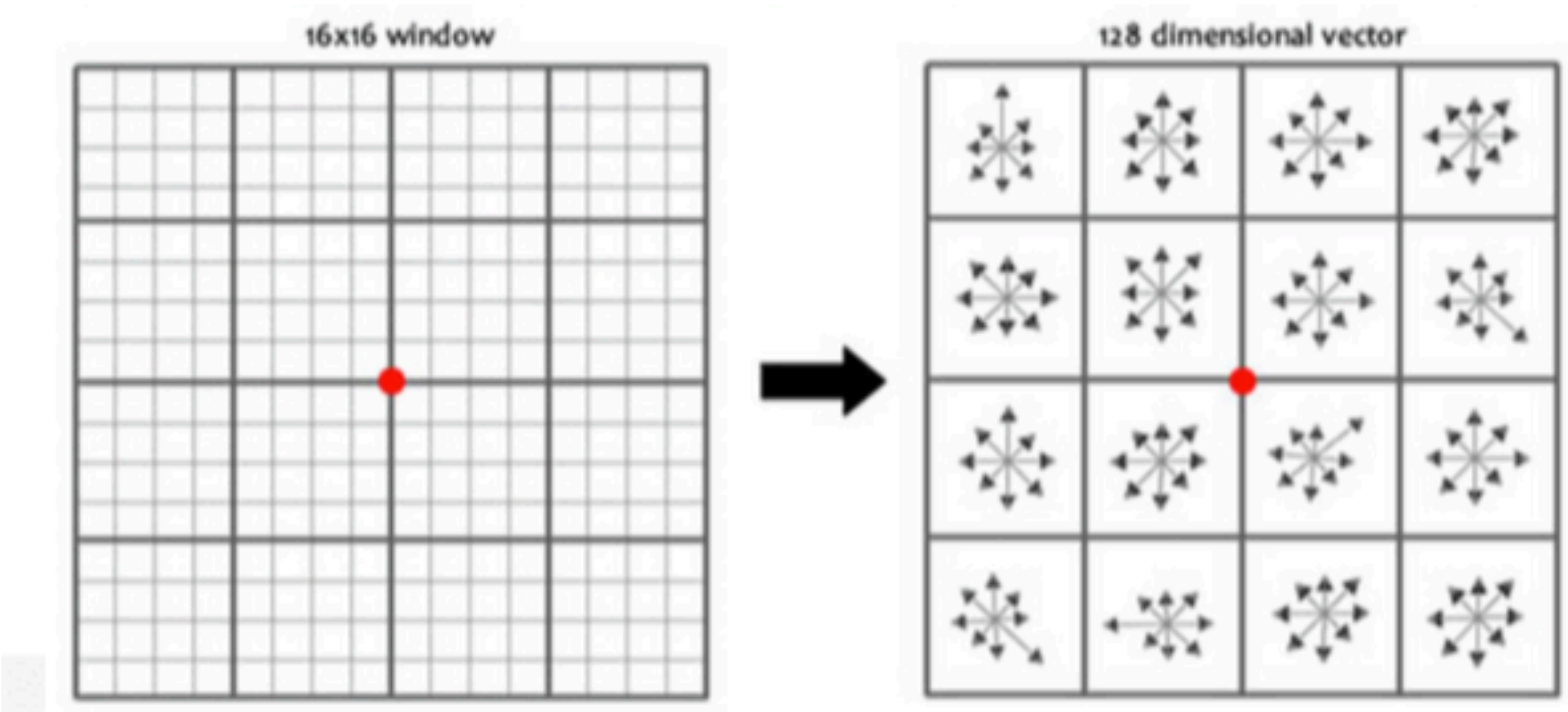
- Descritor (Exemplo)
  - Região 2x2
  - 8 orientações



Centro de Informática

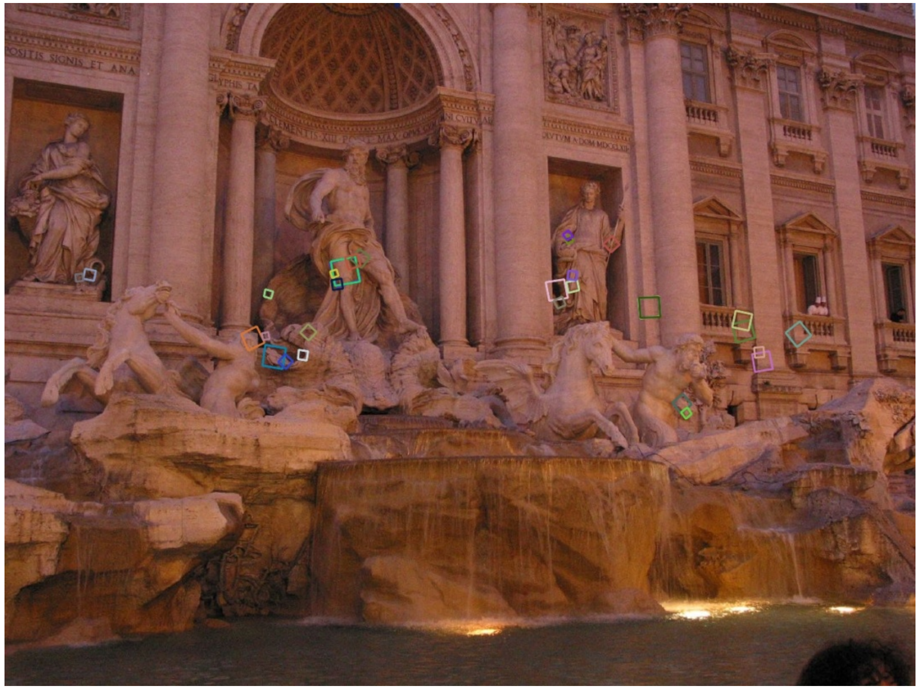
Descritor (sugerido no artigo)

- Região 4x4; 8 orientações
  - Tamanho do vetor descritor: 4x4x8=128

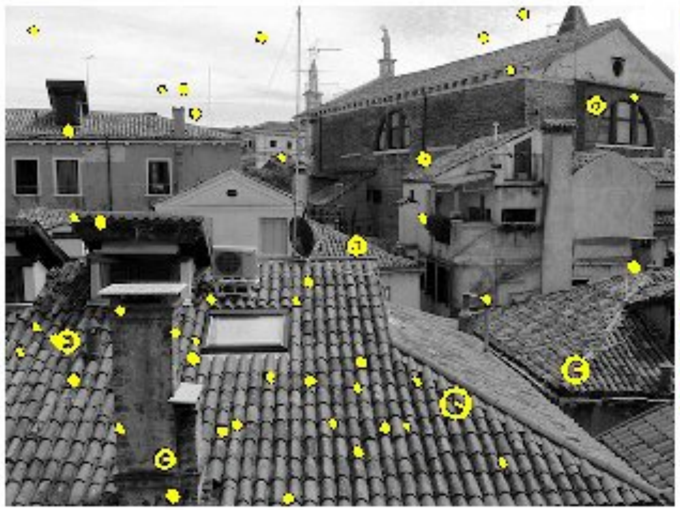


Propriedades do SIFT

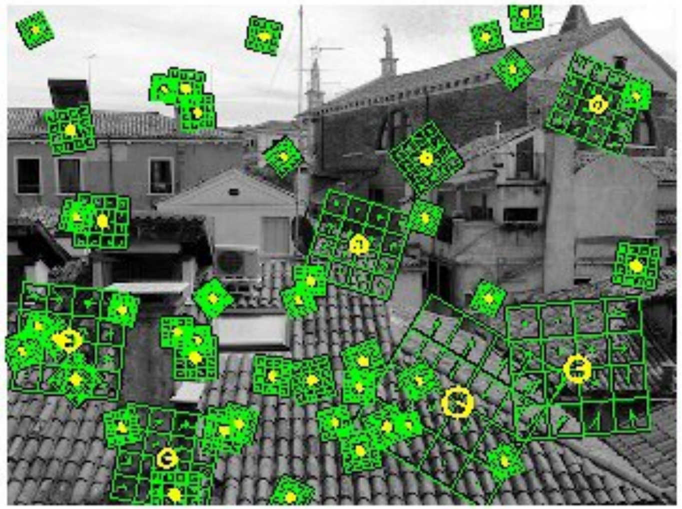
- Pode lidar com rotação de plano de até cerca de 60 graus
- Robusto a variação de iluminação
- Pode rodar em tempo real



SIFT



Interest points and their scales and orientations (random subset of 50)



SIFT descriptors

Code

```
sift = cv2.xfeatures2d.SIFT_create()
(kps, descs) = sift.detectAndCompute(gray, None)
```

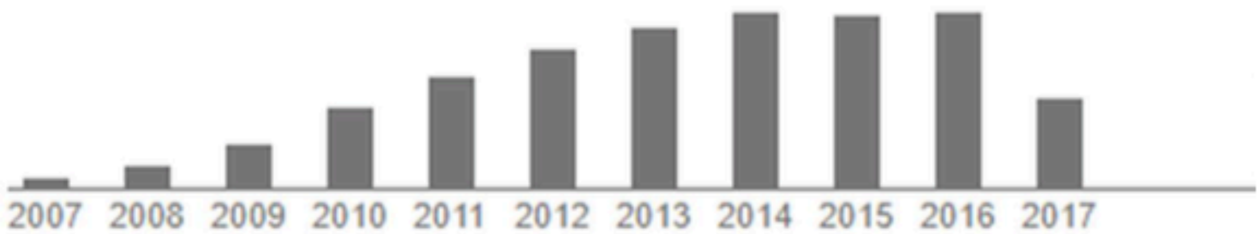
SURF

SURF

“Speeded-Up Robust Features” [3]

Computer Vision and Image Understanding, 2004

Total de citações Citado por 18614



SURF

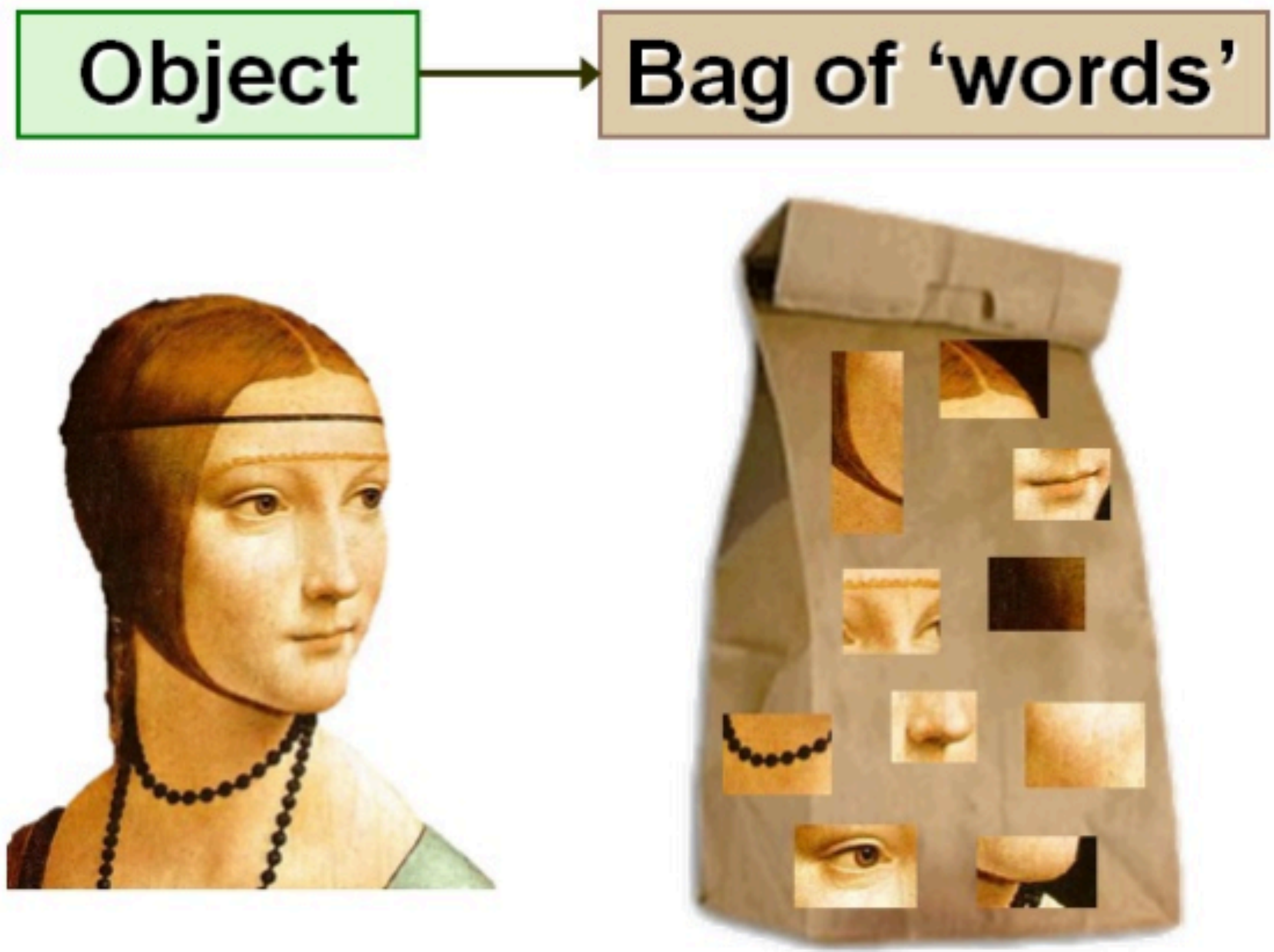
- Baseado em conceitos similares ao SIFT
- Porém com uma grande vantagem: maior velocidade!
  - Aproximação da DoG
  - Uso de imagens integrais [9] reduzem o tempo de processamento

Descritores Binários

- Mais rápidos
- Menos memória

- BRIEF
- ORB
- BRISK
- FREAK
- AKAZE

Bag of Words



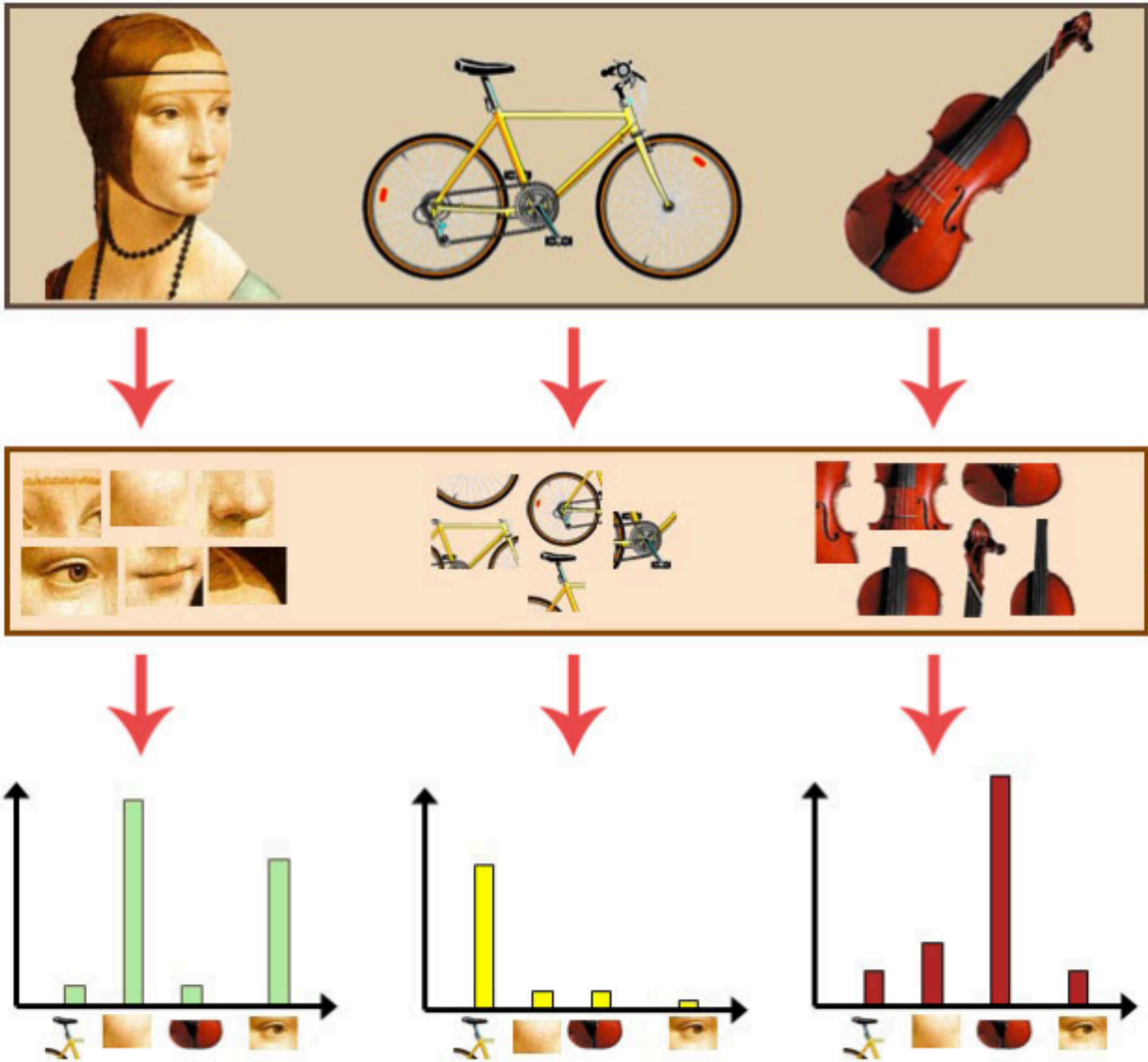
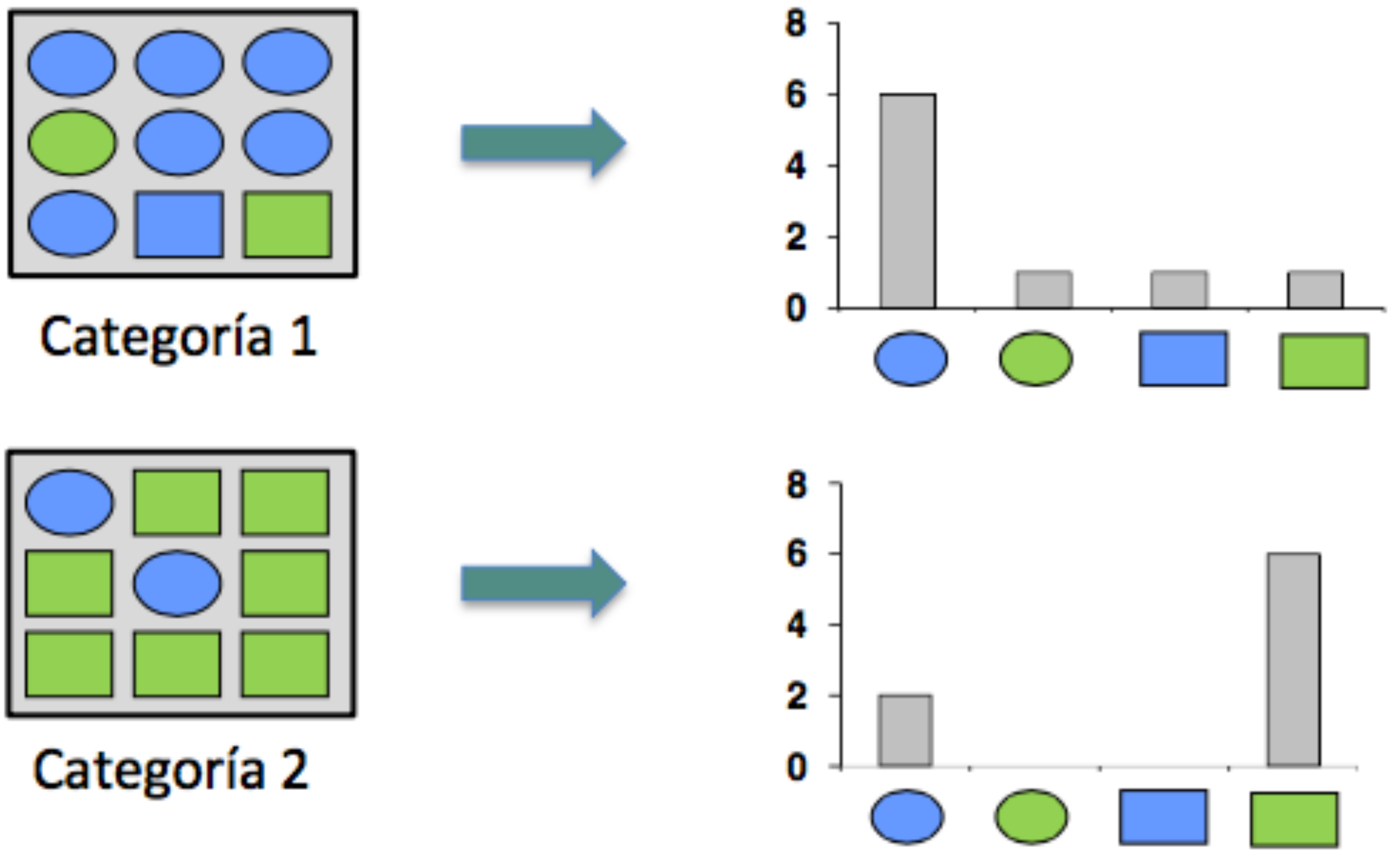


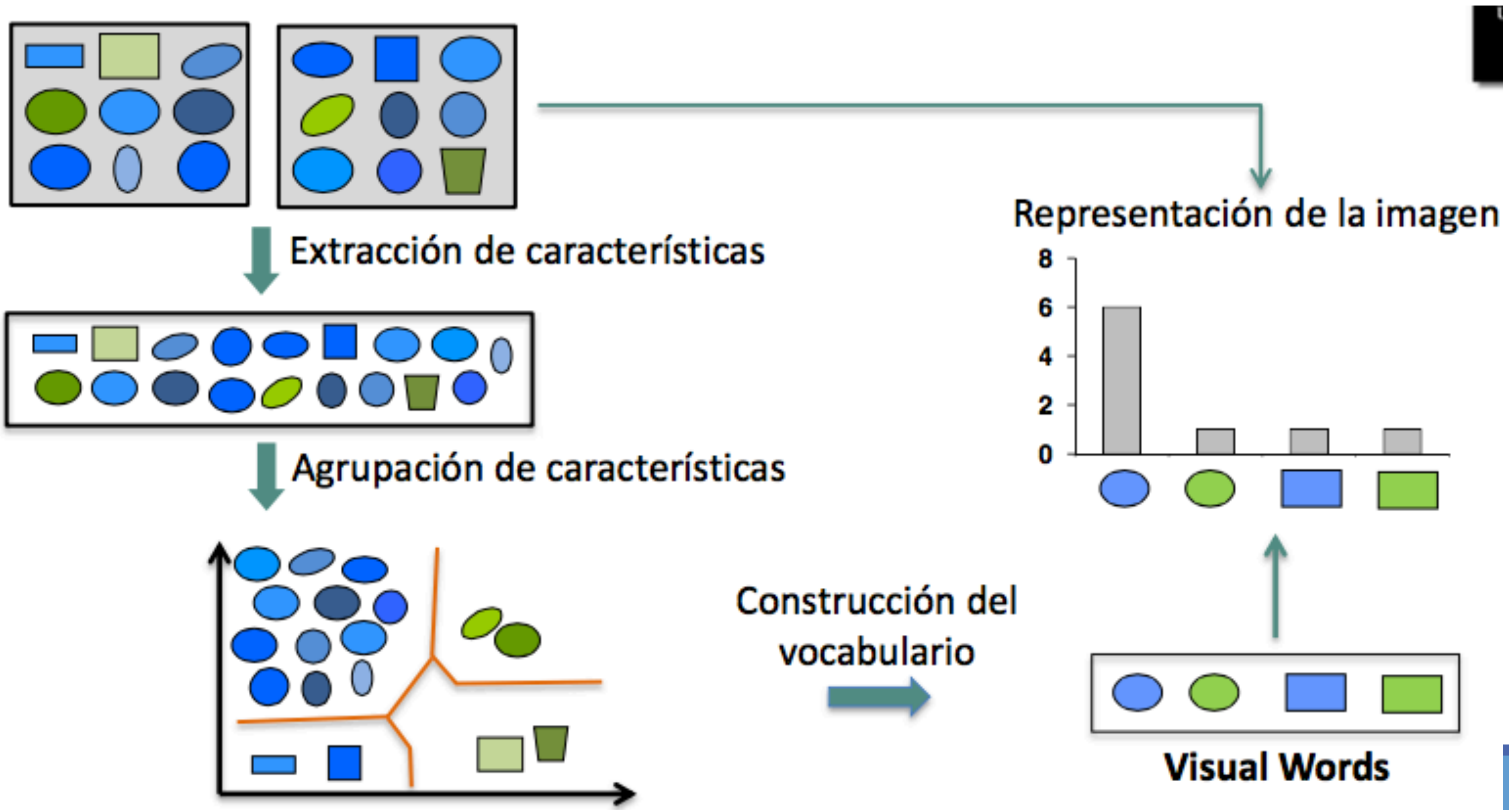
FIGURE 4: TAKING THREE INPUT IMAGES *(TOP)*, EXTRACTING IMAGE PATCHES FROM EACH OF THEM *(MIDDLE)*, AND THEN COUNTING THE NUMBER OF TIMES EACH VISUAL WORD APPEARS IN THE RESPECTIVE IMAGES *(BOTTOM)*.

- Identificar características comuns e relevantes: palavras visuais
- Representação: histograma de palavras visuais

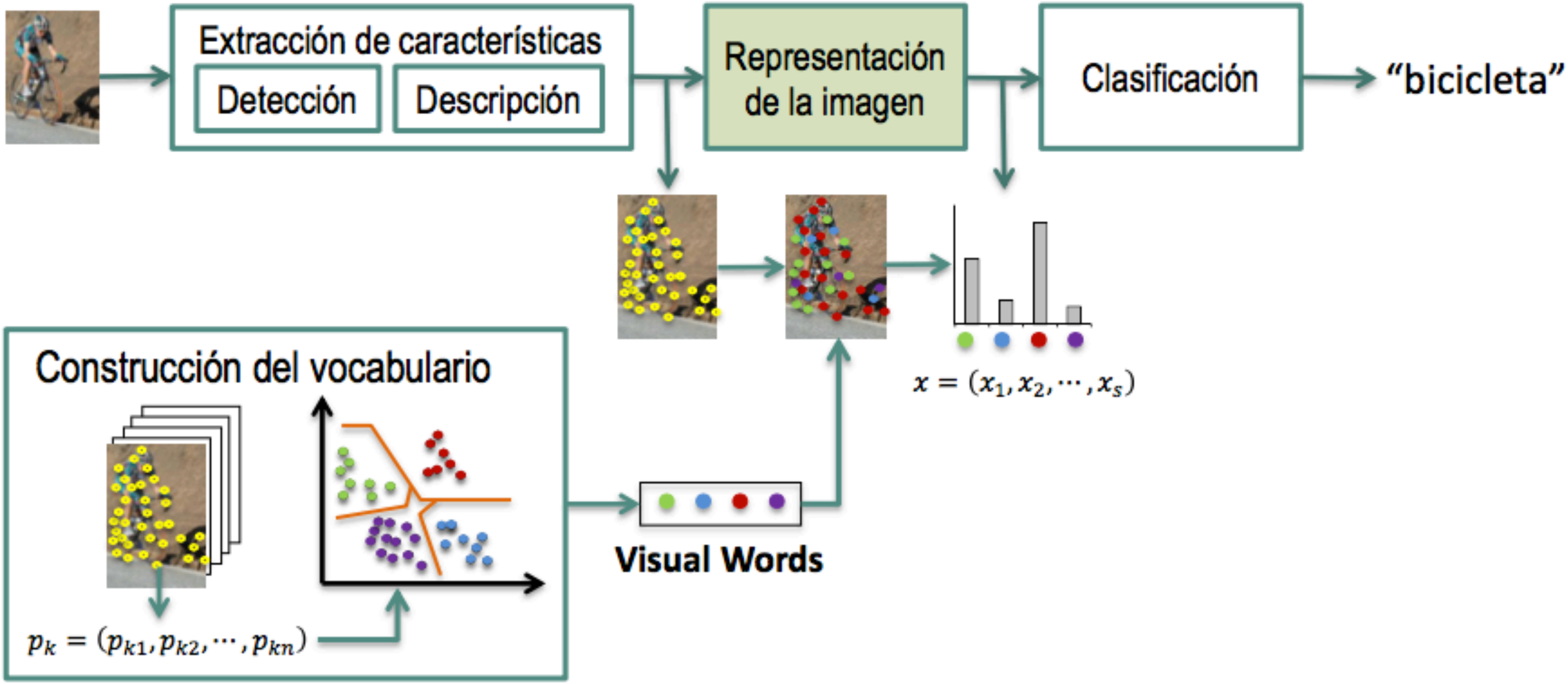


Bag of Visual Words

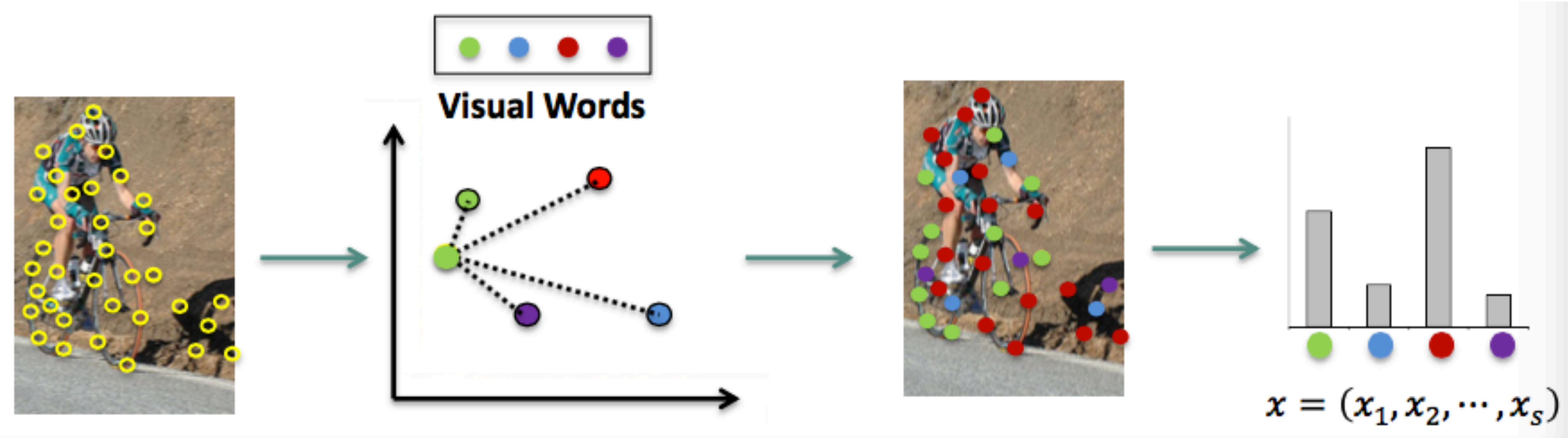
- Construção do vocabulário de palavras visuais



BoVW- Classificação

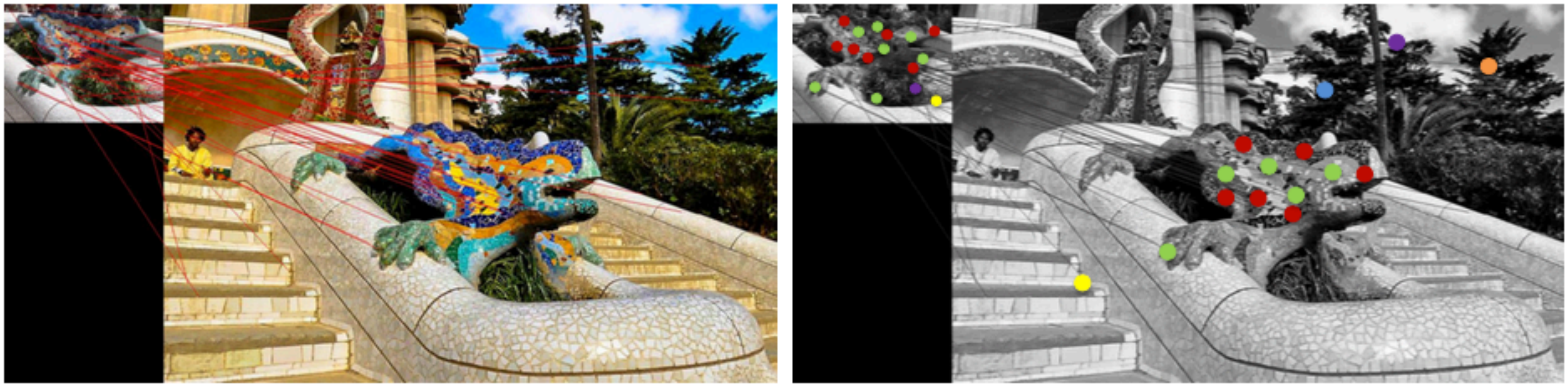


Construção do histograma



Correspondência de Imagens

▷Melhor correspondência entre características parecidas que correspondem à mesma palavra visual



▷Pior correspondência entre características diferentes que não correspondem à mesma palavra visual