



Uma não tão breve introdução ao básico de PI

Visão Computacional

Programa de Pós-Graduação em Ciência da Computação – UFMA

Prof. Geraldo Braz Junior

O que fazer com essa aula?

- Utilize essa aula para conhecer métodos básicos que serão usados em visão
- Busque leituras recomendadas no site da disciplina
 - Depois que ler esses textos, busque por outros

Passos Clássicos em uma metodologia de Visão

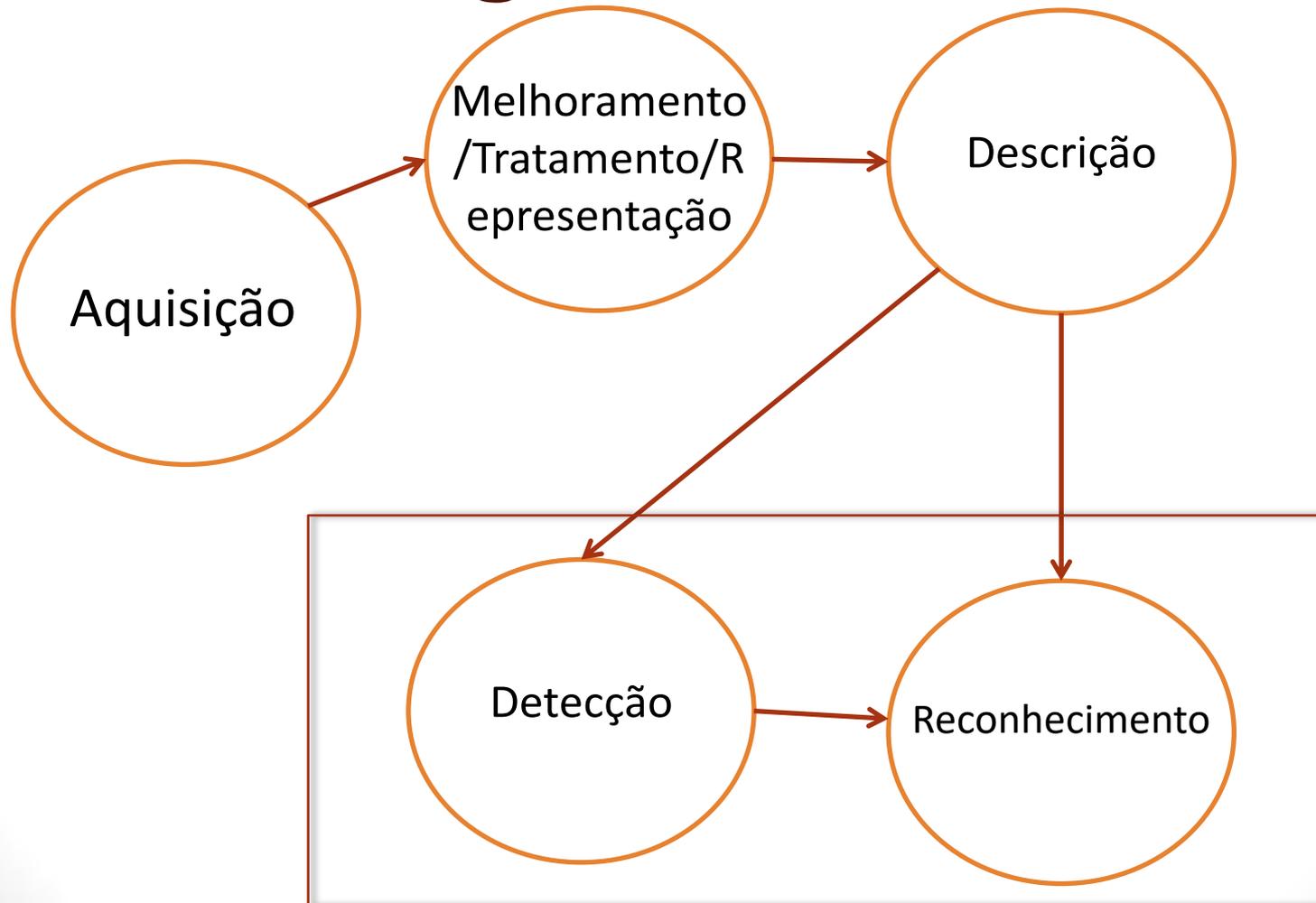


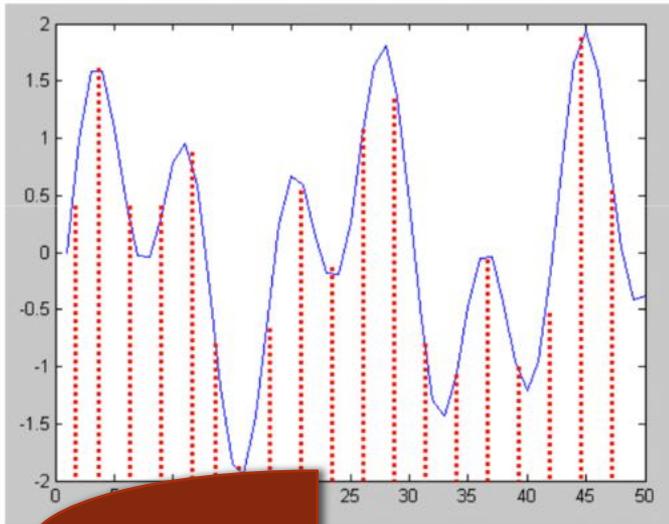
Imagem como função

- Teoricamente, imagens são funções de \mathbb{R}^2 a \mathbb{R}^M do brilho refletido de uma cena

$f(x,y)$ fornece o valor de intensidade numa posição (x,y)

- O processo envolve as etapas de:
 - Amostragem
 - Quantização

Amostragem e Quantização



Domínio da Frequência

* O processo inverso pode ser obtido com a Transformada de Fourier

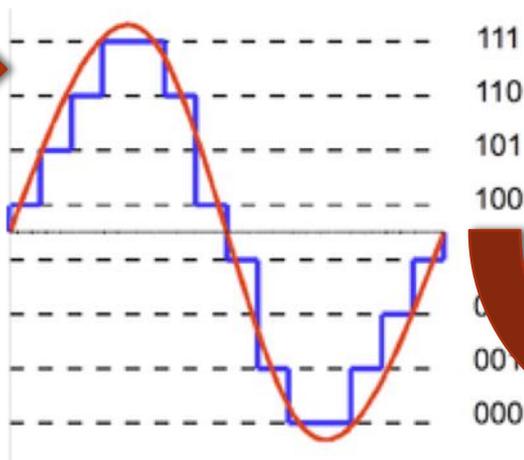
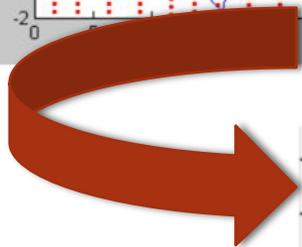
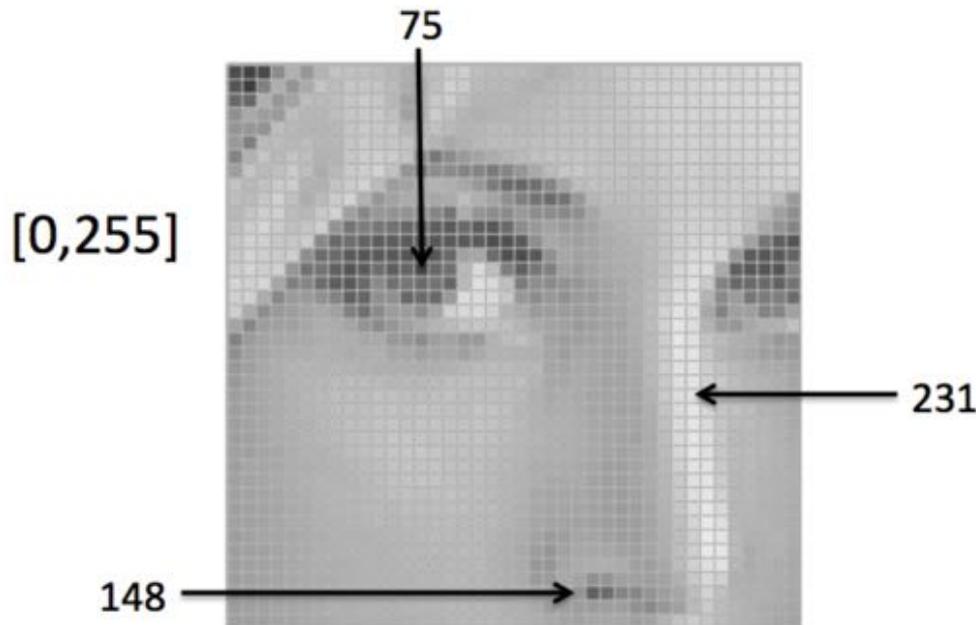
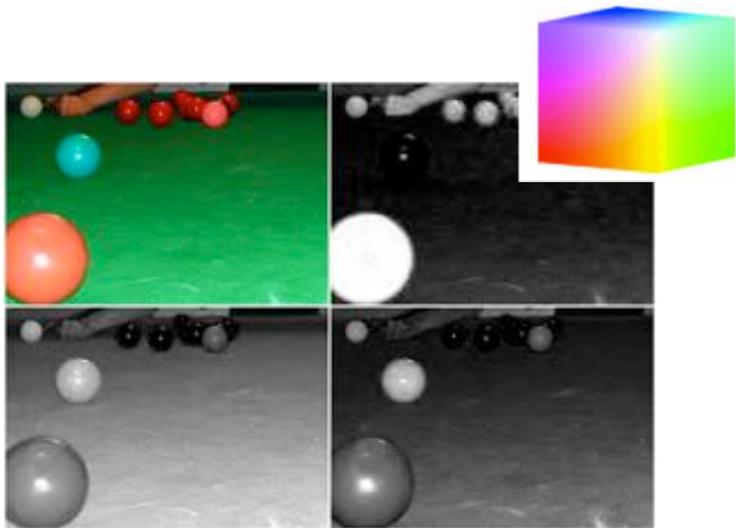


Imagem Discreta Digital

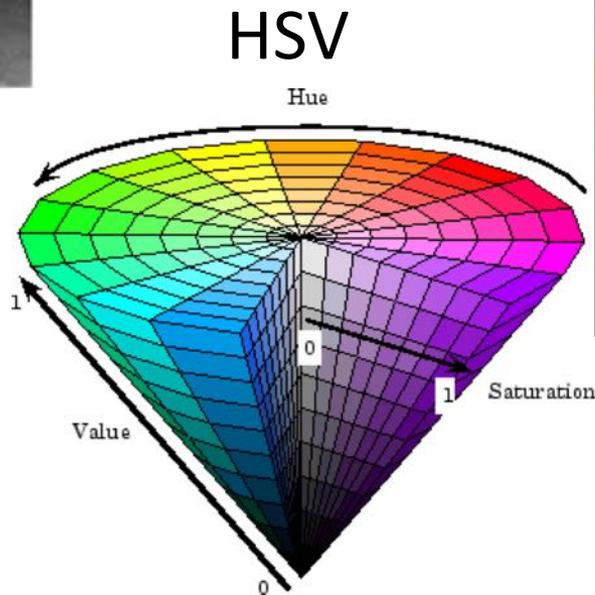
- Uma imagem contém uma quantidade discreta de elementos chamados **pixels**
- Cada pixel possui um valor
 - intensidade no caso de tons de cinza



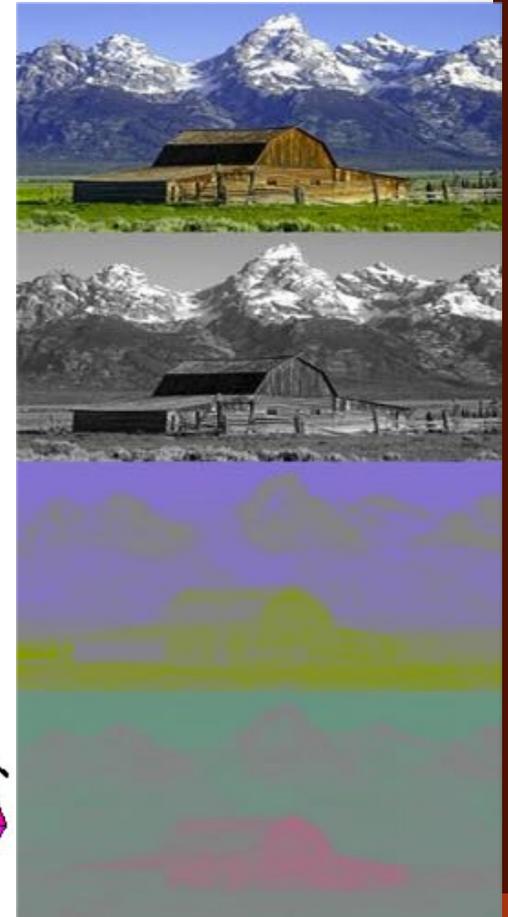
Esquemas diferentes de quantização



RGB



HSV



YCbCr

Nem sempre...

- nem sempre a aquisição é perfeita (pense numa aplicação sem ruídos ...)
- Sujeita a:
 - Qualidade do equipamento de captura
 - Efeitos de iluminação idesejados
 - Ruídos
- Numa situação real a técnica tem que ser o máximo robusta a este tipo de caso

Melhoramento/Tratamento

- Operações que visam extrair informações importantes da imagem
 - Características
 - Cantos, bordas, agrupamentos
 - Realce
 - retirada de ruídos, aumento de contraste, ...
- O processo de filtragem forma uma nova imagem a qual é uma combinação dos pixels da imagem original

Filtros Lineares

$$f(x, y) \rightarrow \boxed{\mathcal{S}} \rightarrow g(x, y)$$

- \mathcal{S} é um filtro linear caso satisfaça a seguinte condição:

$$\mathcal{S}[\alpha f_1 + \beta f_2] = \alpha \mathcal{S}[f_1] + \beta \mathcal{S}[f_2]$$

Exemplo: Filtro da Média

- Um filtro da média pode ser definido como uma janela 3x3 que se move através de uma vizinhança de pixels, aplicando a seguinte fórmula

$$(f * h)[m, n] = \frac{1}{9} \sum_{k, l} f[k, l] h[m - k, n - l]$$

- * representa a convolução de duas funções: f e h

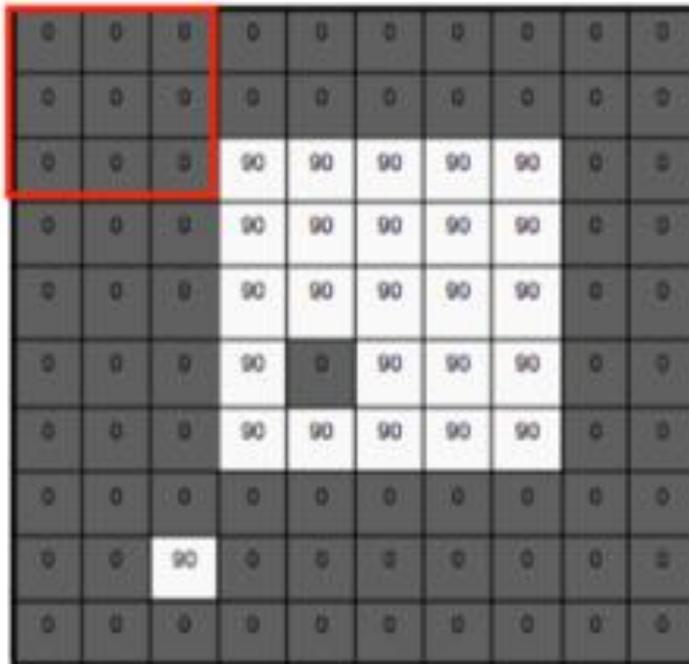
Onde

- k varia de n-1 a n+1, l varia de m-1 a m+1
- m e n são as dimensões da imagem f

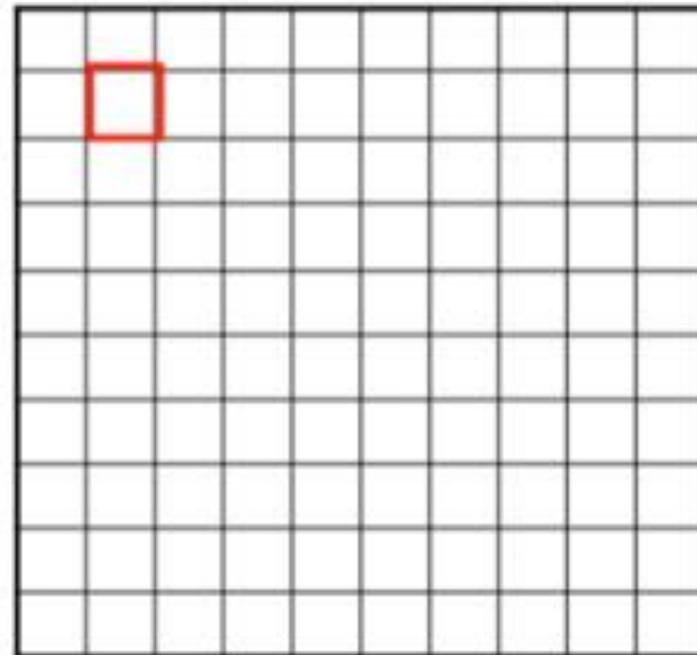
	h		
$\frac{1}{9}$	1	1	1
	1	1	1
	1	1	1

Processo de Filtragem por Média

$F[x, y]$



$G[x, y]$



$$(f * h)[m, n] = \sum_{k, l} f[k, l] h[m - k, n - l]$$

Processo de Filtragem por Média

$F[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$

0	10								

$$(f * h)[m, n] = \sum_{k, l} f[k, l] h[m - k, n - l]$$

Processo de Filtragem por Média

$F[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$

	0	10	20	30	30				

$$(f * h)[m, n] = \sum_{k, l} f[k, l] h[m - k, n - l]$$

Processo de Filtragem por Média

$F[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$

		0	10	20	30	30	30	20	10
		0	20	40	60	60	60	40	20
		0	30	60	90	90	90	60	30
		0	30	50	80	80	90	60	30
		0	30	50	80	80	90	60	30
		0	20	30	50	50	60	40	20
		10	20	30	30	30	30	20	10
		10	10	10	0	0	0	0	0

Correlação vs Convolução

- Correlação 2d

$$h[m,n] = \sum_{k,l} f[k,l] I[m+k,n+l]$$

`h=filter2(f,I);` or `h=imfilter(I,f);`

- Convolução 2d

$$h[m,n] = \sum_{k,l} f[k,l] I[m-k,n-l]$$

`h=conv2(f,I);` or `h=imfilter(I,f,'conv');`

`conv2(I,f)` é igual a `filter2(rot90(f,2),I)`

Correlação e convolução são idênticos quando os filtros são simétricos

Exemplo: Filtro da Média



Exemplo: combinação de filtros?



-



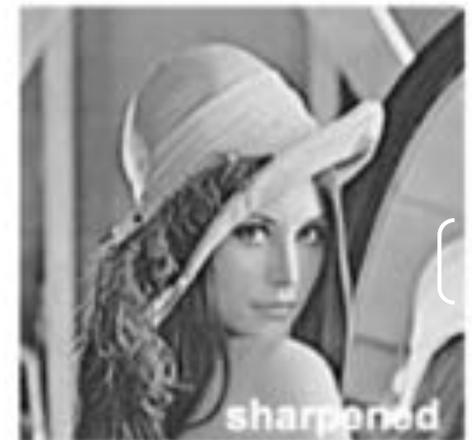
=



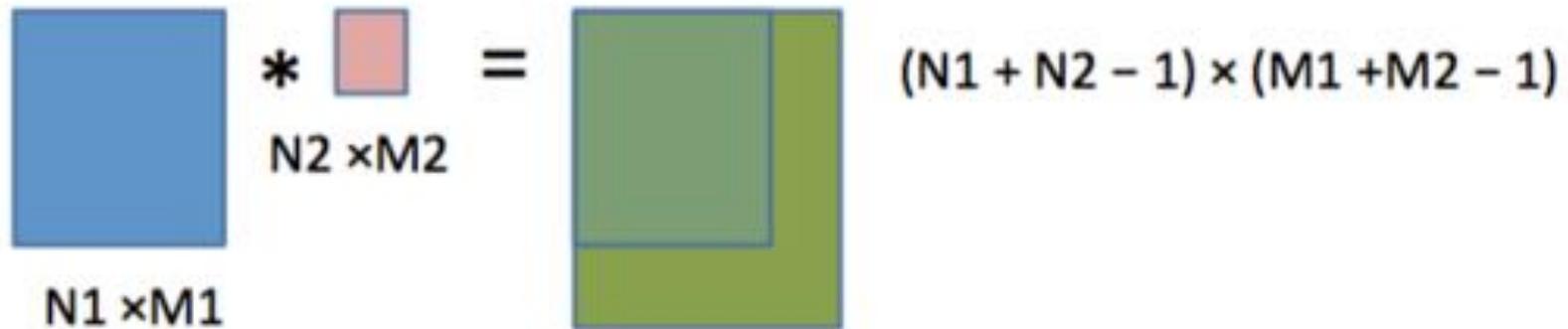
+ a



=

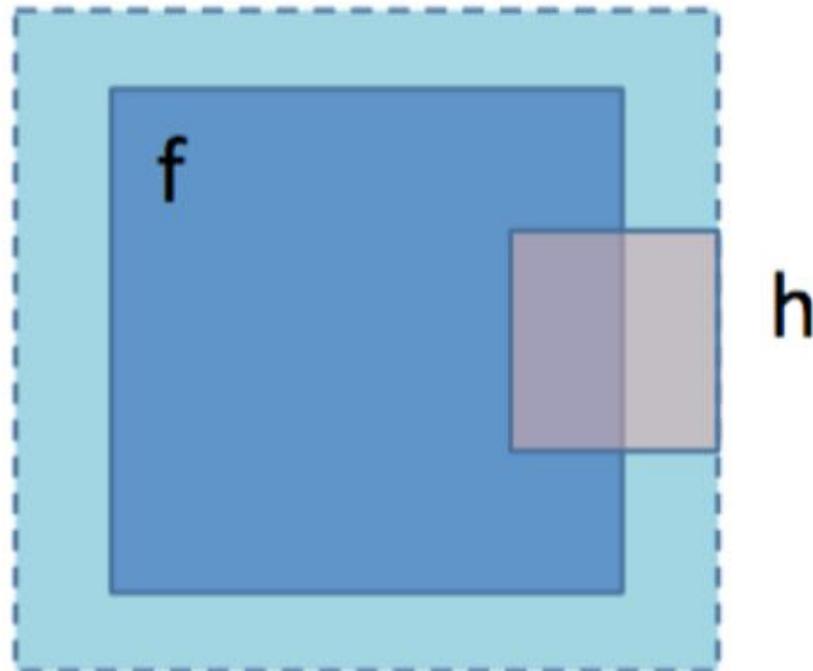


O que fazer com nas bordas da imagem?



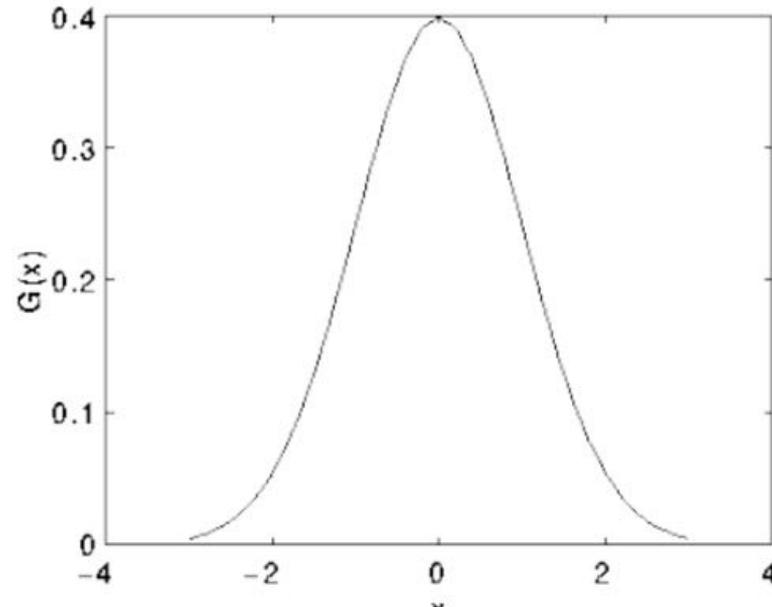
O que fazer com nas bordas da imagem?

- Somente será computado pixels que tenham representantes completos dentro do filtro
- Estratégias que podem ser adotadas
 - Replicar borda
 - Adicionar zeros
 - Extensão por espelho



Filtro Gaussiano

$$G(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$



$$Z = \frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

Filtros não lineares

- Não preservam relação inversa
- Principais:
 - Bilateral
 - Mediana

Filtro Bilateral

- Redução de ruídos
- Simplificação de estruturas
- Preserva contornos

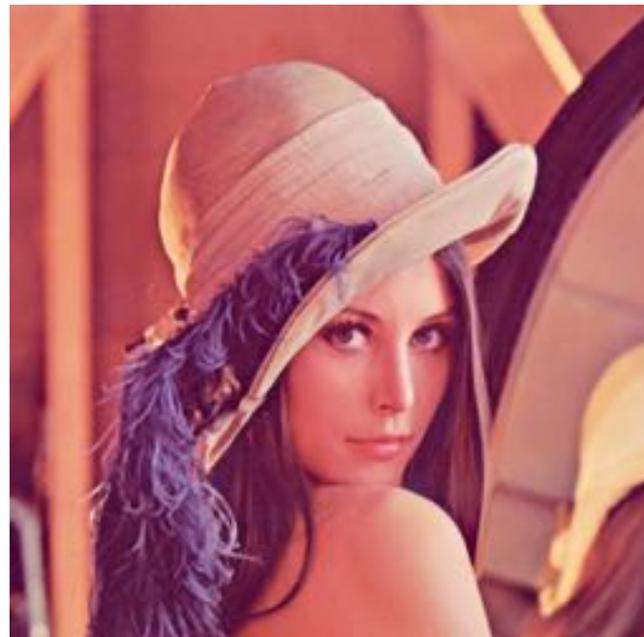


Filtro da Mediana

- Melhor para reduzir o ruído
- Ordena-se os vizinhos em ordem crescente.
 - A mediana consiste nos m elementos que estão no meio do conjunto
- Eficiente contra ruídos localizados

Filtro: mediana

- `salt(original, 3000);`
- `medianBlur(original, resultado, 3);`
- *Procure implementação de sal nos exemplos



Morfologia Matemática

- Não Linear
- Objetivo: extrair objetos baseados em alguma geometria pré-definida
 - Elemento Estruturante
- Normalmente sobre imagens binárias
 - representadas com apenas 2 intensidades

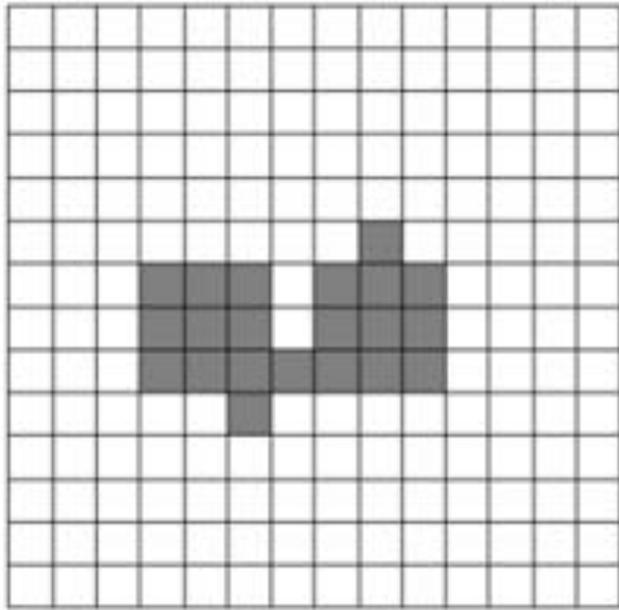
Operações Básicas

- Dilatação
 - Preenche ou amplifica contornos

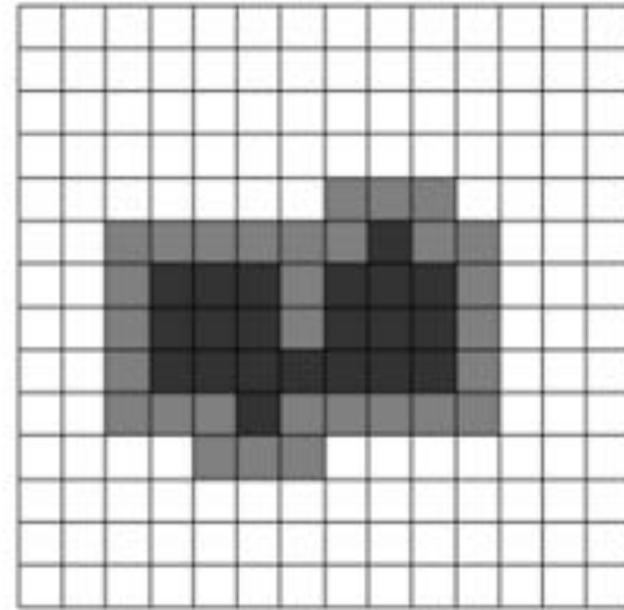
$$A \oplus B = \{z \mid (\hat{B})_x \cap A \neq \emptyset\}$$

- Reflexão (translação) de B para a origem centralizada em um elemento x de A
- B é chamado de **elemento estruturante**

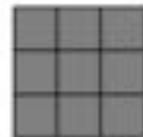
Exemplos



F



G



H, 3x3, origin at the center

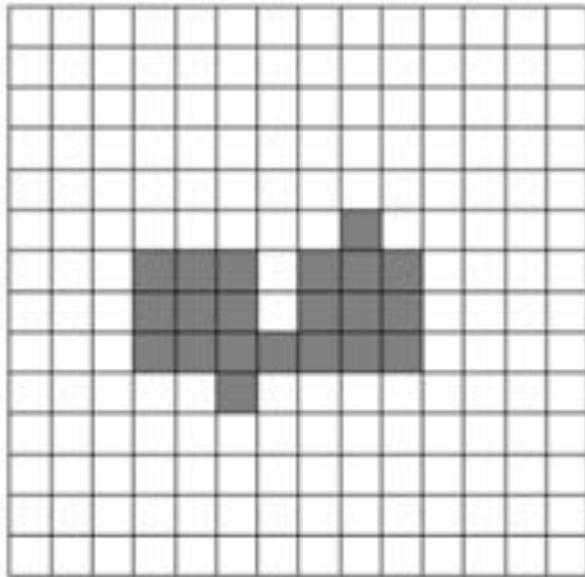
Operações Básicas

- Erosão
 - Reduz elementos ao elemento estruturante

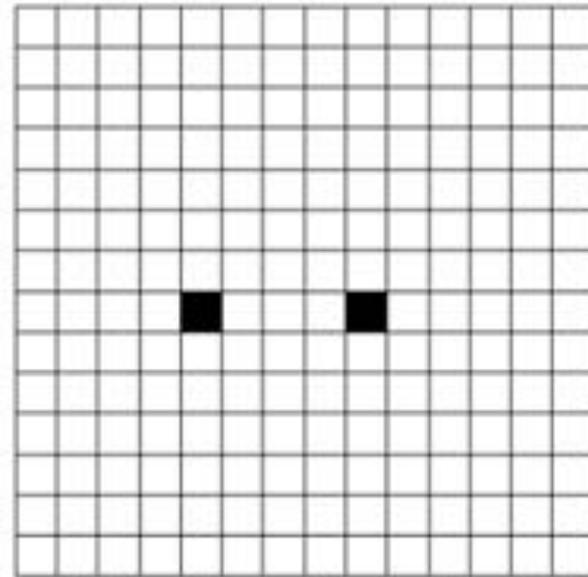
$$A \ominus B = \{x \mid (B)_x \subseteq A\}$$

- Reflexão (translação) de B para a origem centralizada em um elemento x de A tal que todos os elementos de B estejam em A
- B é chamado de **elemento estruturante**

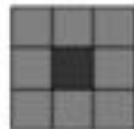
Exemplos



F

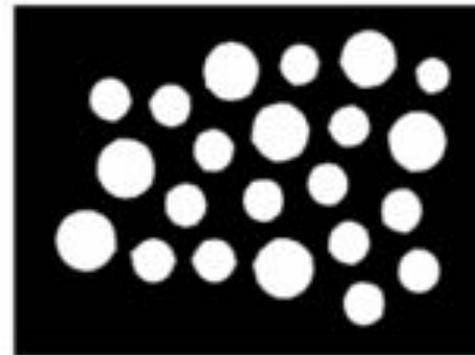
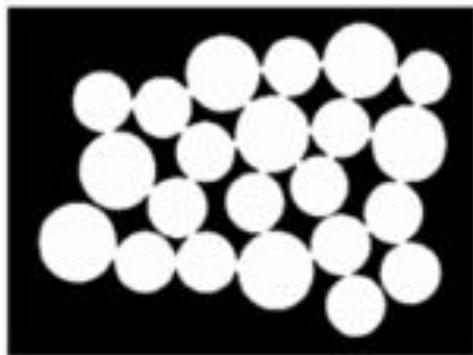


G



H, 3x3, origin at the center

Exemplos: separação



Operações mais complexas

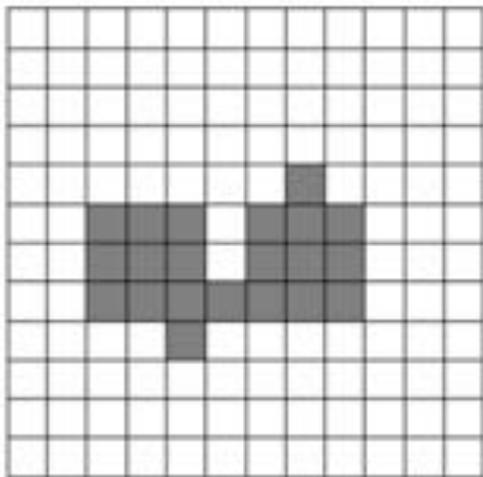
- Utilizam combinações de dilatação e erosão
- Ou
- Expansões na teoria dos conjuntos
- Exemplos:
 - Abertura
 - Fechamento
 - Gradiente
 - TopHat
 - Hit & Miss
 - Fill
 - Skeleton

Abertura

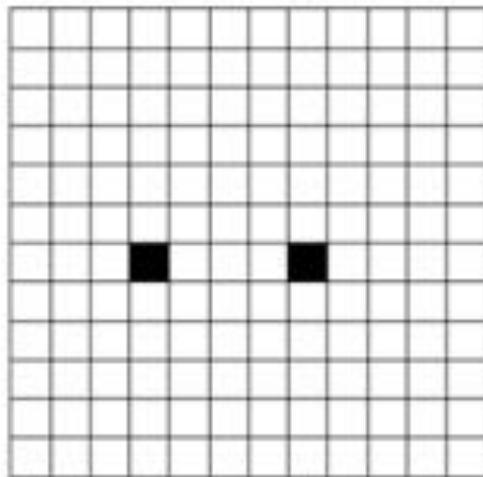
- Suaviza o contorno da imagem ao mesmo tempo que elimina pequenas brechas
- Uma erosão seguida de uma dilatação

$$A \circ B = (A \ominus B) \oplus B$$

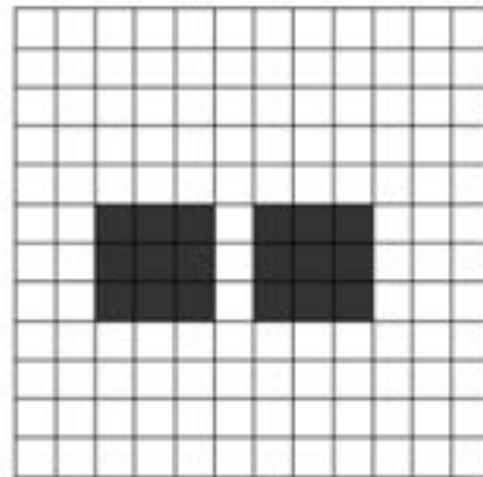
Exemplo



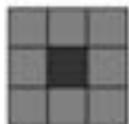
F



$F \ominus H$



$(F \ominus H) \oplus H$



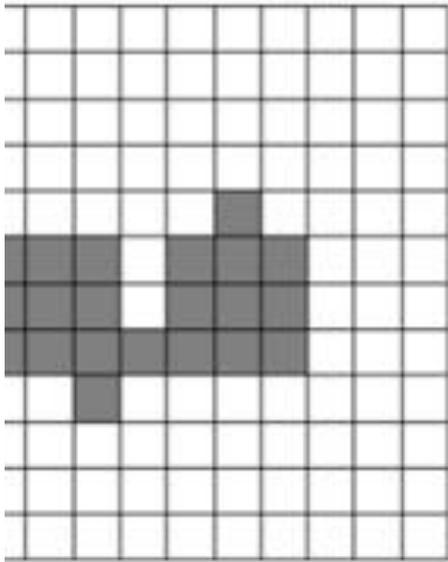
H, 3x3, origin at the center

Fechamento

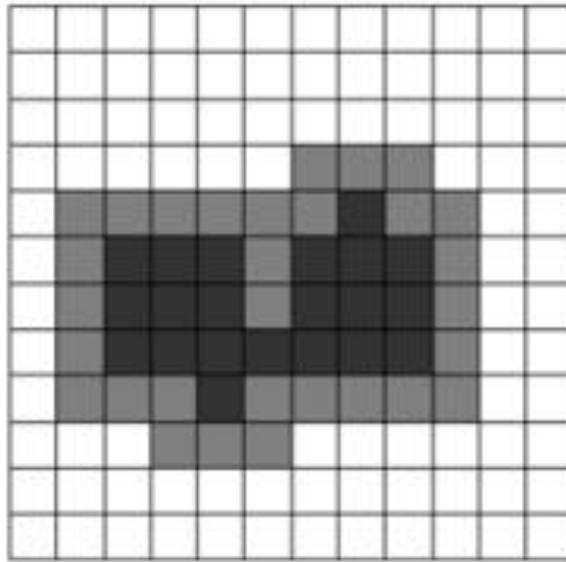
- Suaviza contorno, elimina pequenos buracos, preenche fendas
- Uma dilatação seguida por uma erosão

$$A \bullet B = (A \oplus B) \ominus B$$

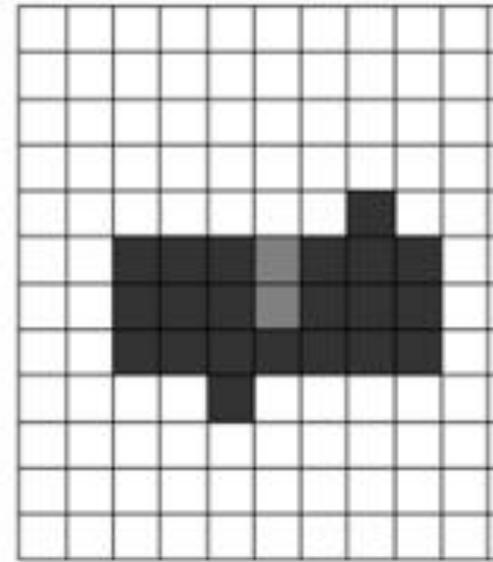
Exemplo



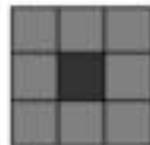
F



$F \oplus H$

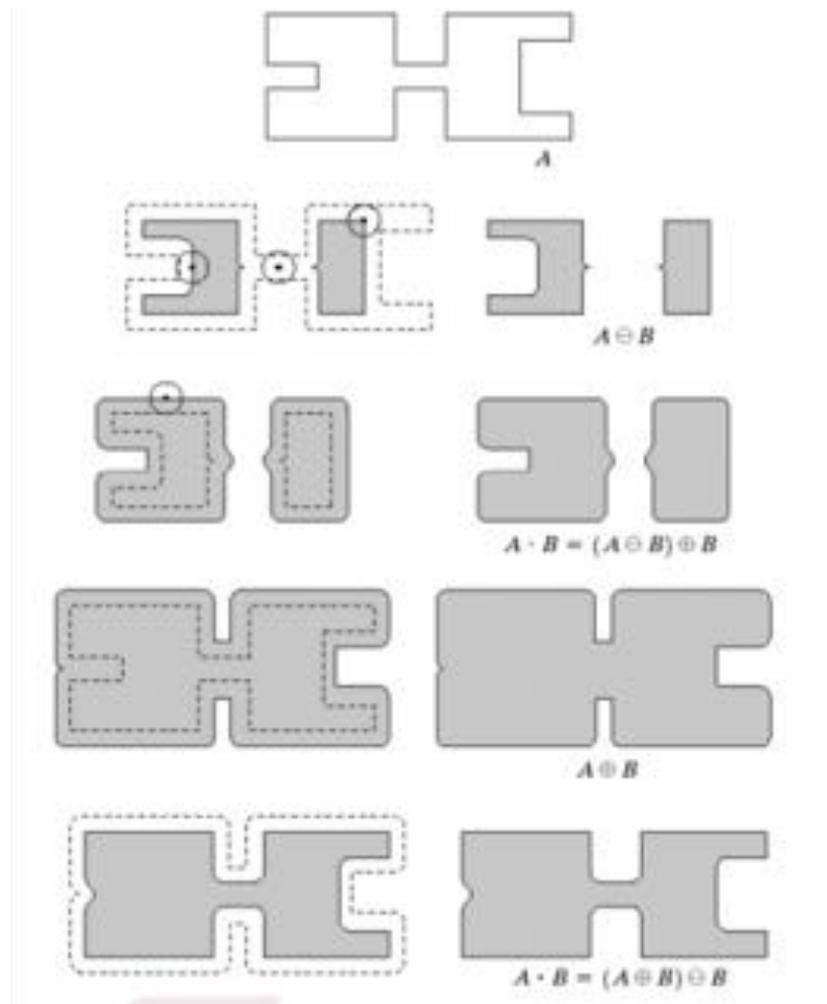


$(F \oplus H) \oplus H$



H, 3x3, origin at the center

Abertura e Fechamento



Uma que não é nativa no opencv

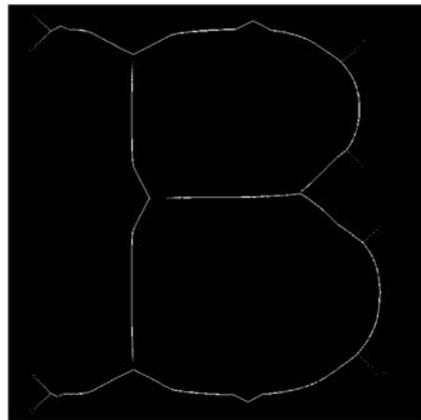
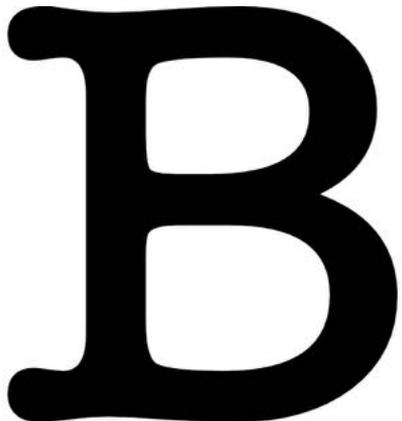
- Até então: skeletonização
- É uma representação da componente de sustentação de uma forma (esqueleto)

$$S_n(X) = (X \ominus B) - (X \ominus B) \circ B$$

- repete-se n vezes: ou até que não exista mais pixels na imagem original (caso binário)

Exemplo

```
while (countNonZero(img) != 0) {  
    erode(img, eroded, element);  
    dilate(eroded, temp, element);  
    subtract(img, temp, temp);  
    bitwise_or(resultado, temp, resultado);  
    eroded.copyTo(img);  
}
```



O processo de melhoramento

- Também pode ser usado para extrair informação da imagem ou realçadas de maneira a destacar essas informações.
- Uma categoria específica trata do destaque de **altas frequências** na imagem para destacar **bordas** como a finalidade de gerar **contornos**

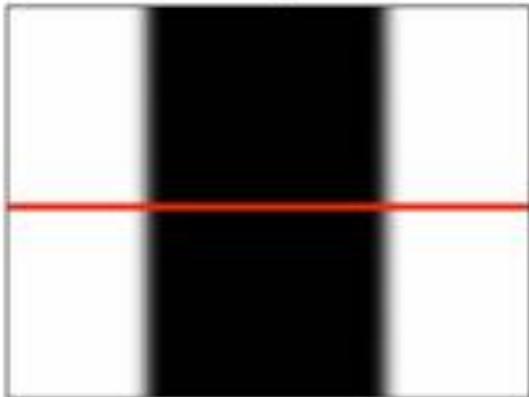
Porque contornos são importantes?



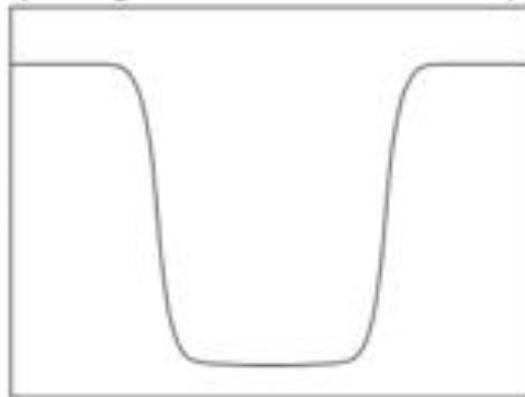
O que são bordas?

- Local na imagem onde aconteceu rápida mudança de tonalidades

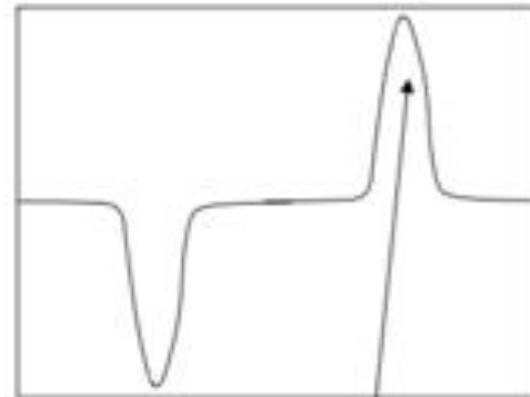
image



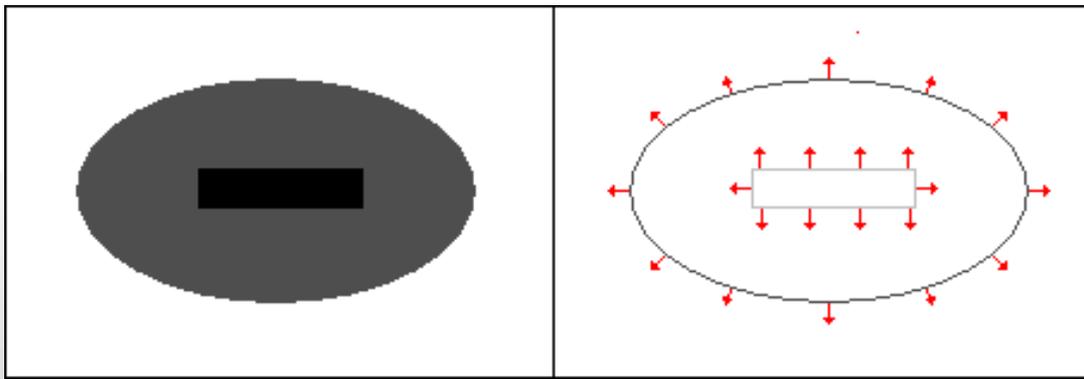
intensity function
(along horizontal scanline)



first derivative



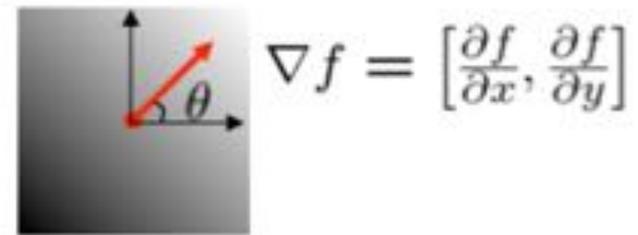
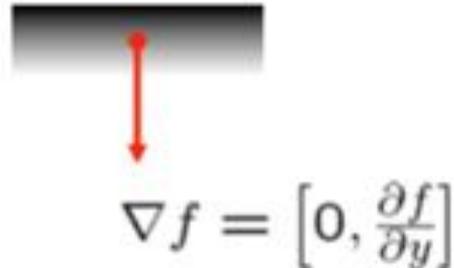
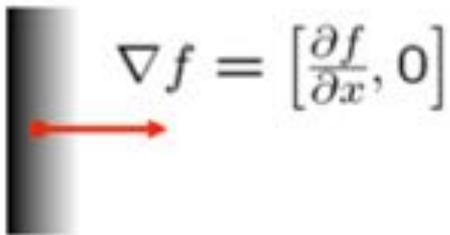
edges correspond to
extrema of derivative



Gradiente de uma imagem

- O gradiente de uma imagem é a derivada no ponto

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$



Gradiente de uma imagem

- A força (magnitude) de um gradiente é dado por:

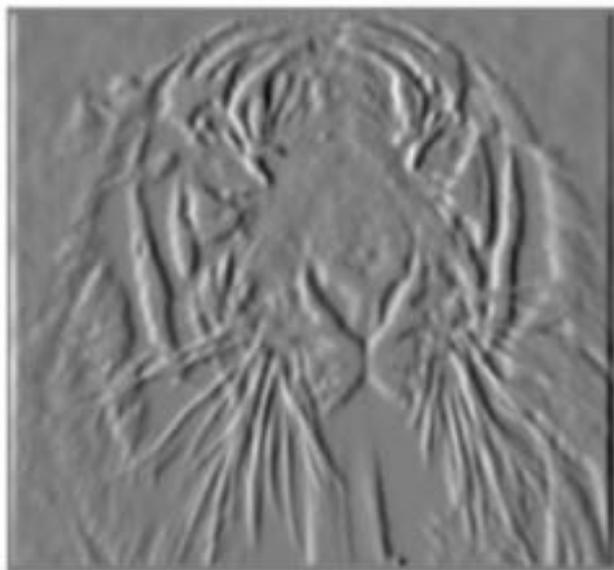
$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

- Representa a força direcional num ponto (x,y)
- A orientação do gradiente é dada por:

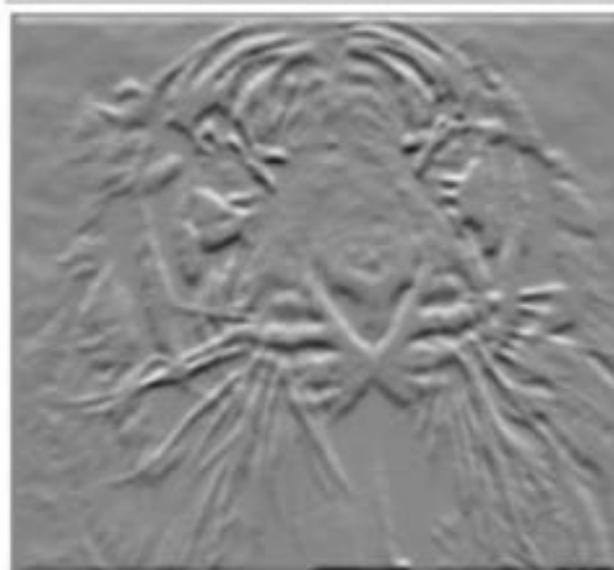
$$\theta = \tan^{-1} \left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)$$

- Representa a orientação da borda no ponto (x, y)

Gradiente de uma imagem



X



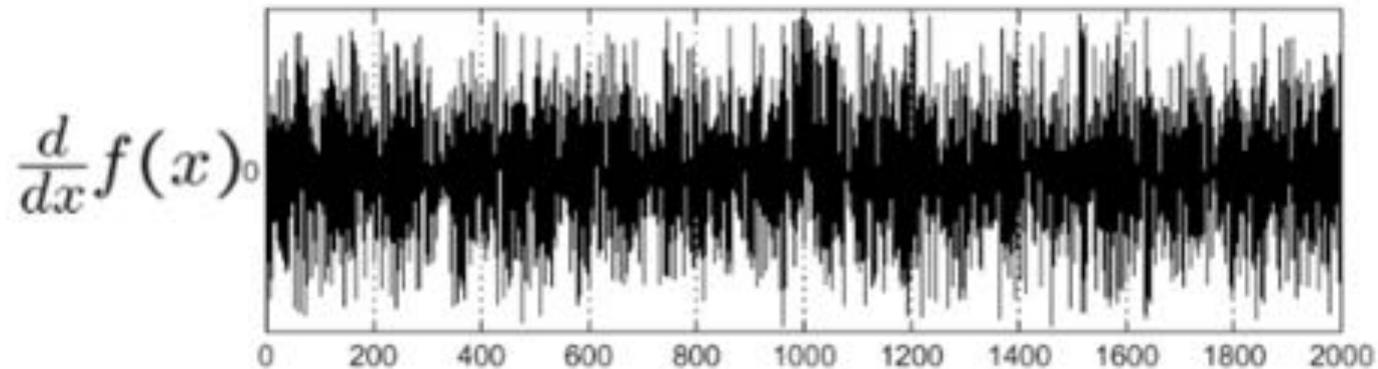
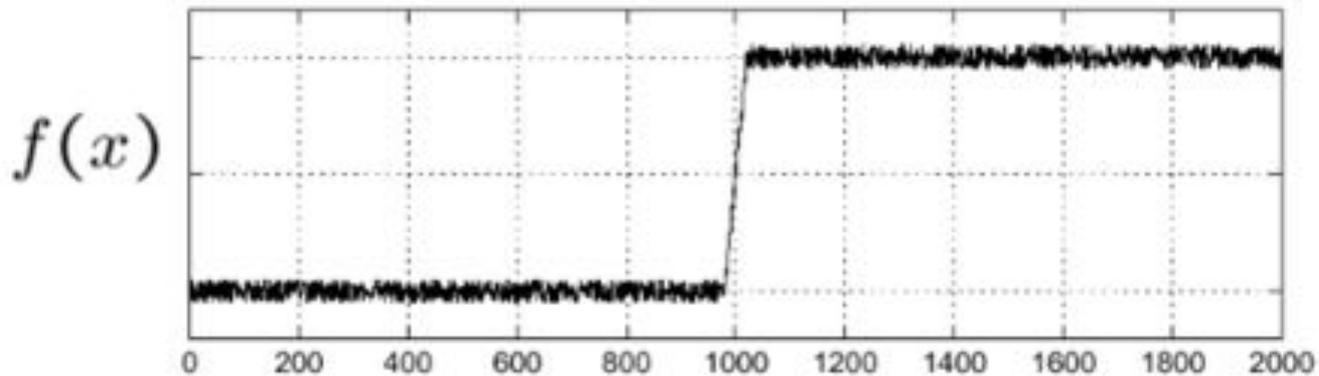
Y

Magnitude do gradiente



Efeitos causados por ruídos

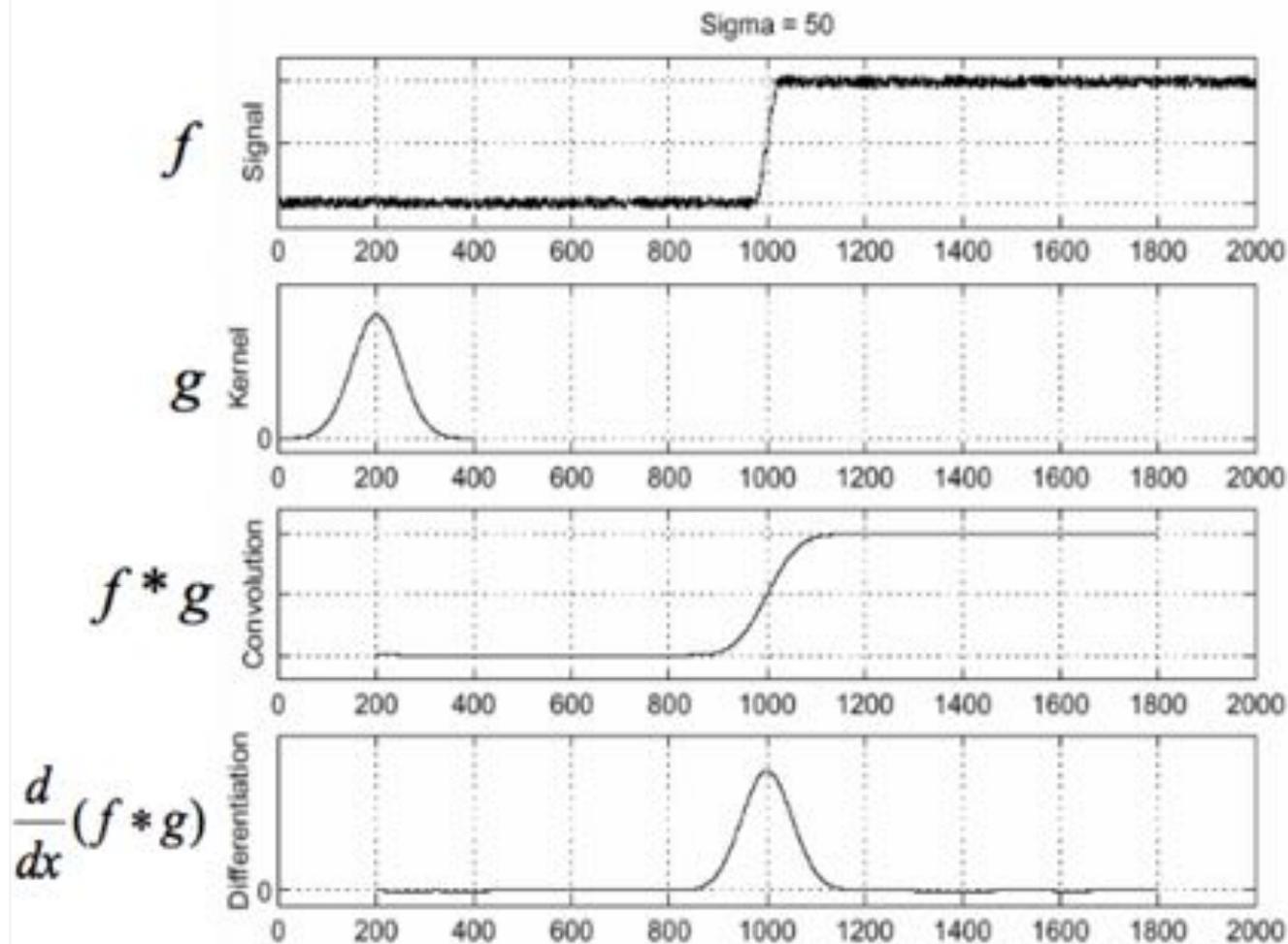
- Onde está o contorno?



Efeito dos ruídos: o que fazer?

- O ruído faz com que bordas não sejam corretamente obtidas:
 - Pixels muito próximos possuem comportamentos aleatório
 - Mudanças súbitas de cor
- O que pode ser feito?
 - Remover ruídos com filtros passa baixa

Filtro Primeiro: Gaussiano



Opções de Detectores

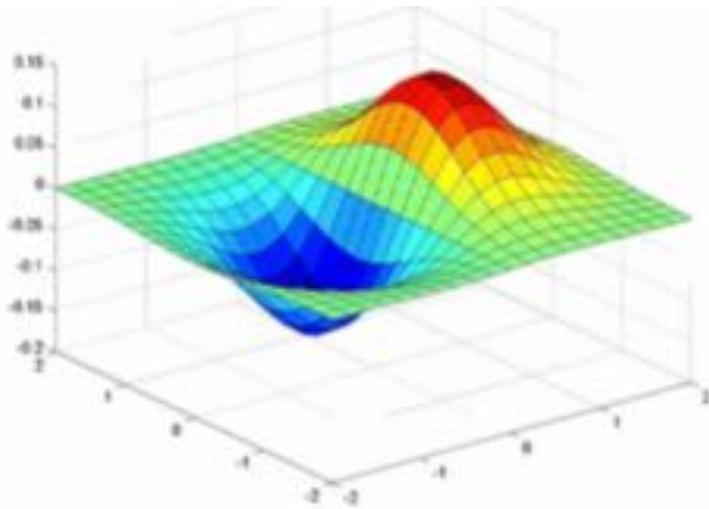
- a) Roberts
- b) Prewitt
- c) Sobel
- d) Canny
- e) Laplaciano

Canny

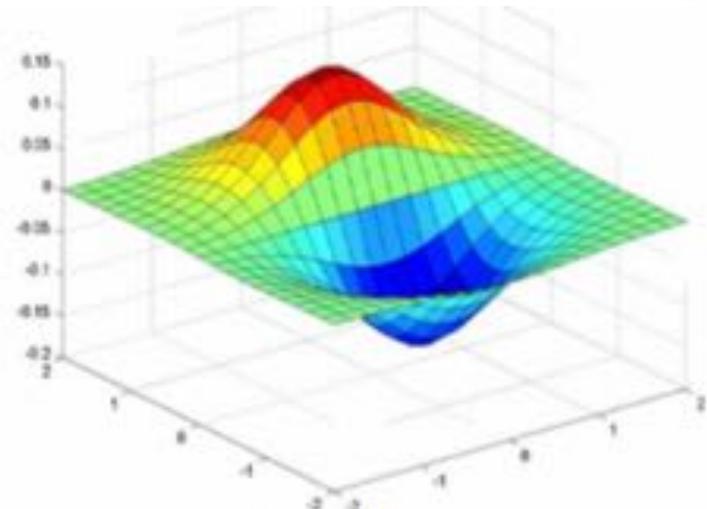
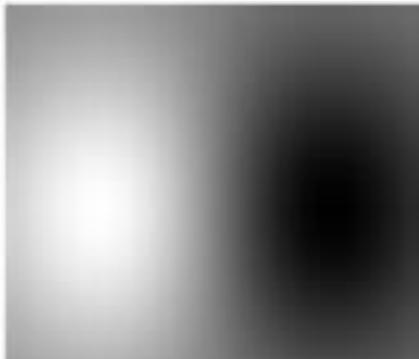
- Baseado em sucessivas Gaussianas (ou usando gaussianas como complementar)
- Canny demonstra que a primeira derivada da gaussiana aproxima com alta precisão a **detecção** de bordas, com as propriedades de **localização** e **atomicidade**
- Primeira derivada de uma gaussiana, porque?

$$\frac{d}{dx}(f * g) = f * \frac{d}{dx}g$$

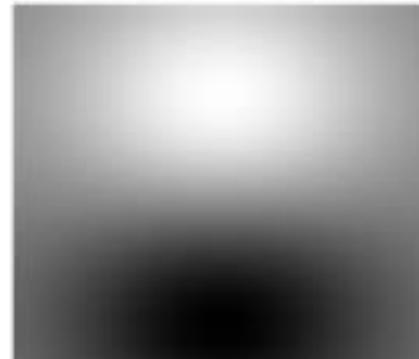
Primeira Derivada de uma Gaussiana



x-direction



y-direction



Passo a Passo do Canny

1. Filtre x , y com as derivadas do Gaussiano
2. Obtenha magnitude e orientação
3. Aplique a regra **Non-Maximum-Suppression**
4. **Threshold** e **hysteresis**
 - Defina duas faixas: inferior e superior
 - Use o superior para iniciar os contornos e o inferior para garantir continuidade

Gradiente por gaussiana



X-Derivative of Gaussian



Y-Derivative of Gaussian



Gradient Magnitude

Orientações



Non-Maximum-Suppression



Antes



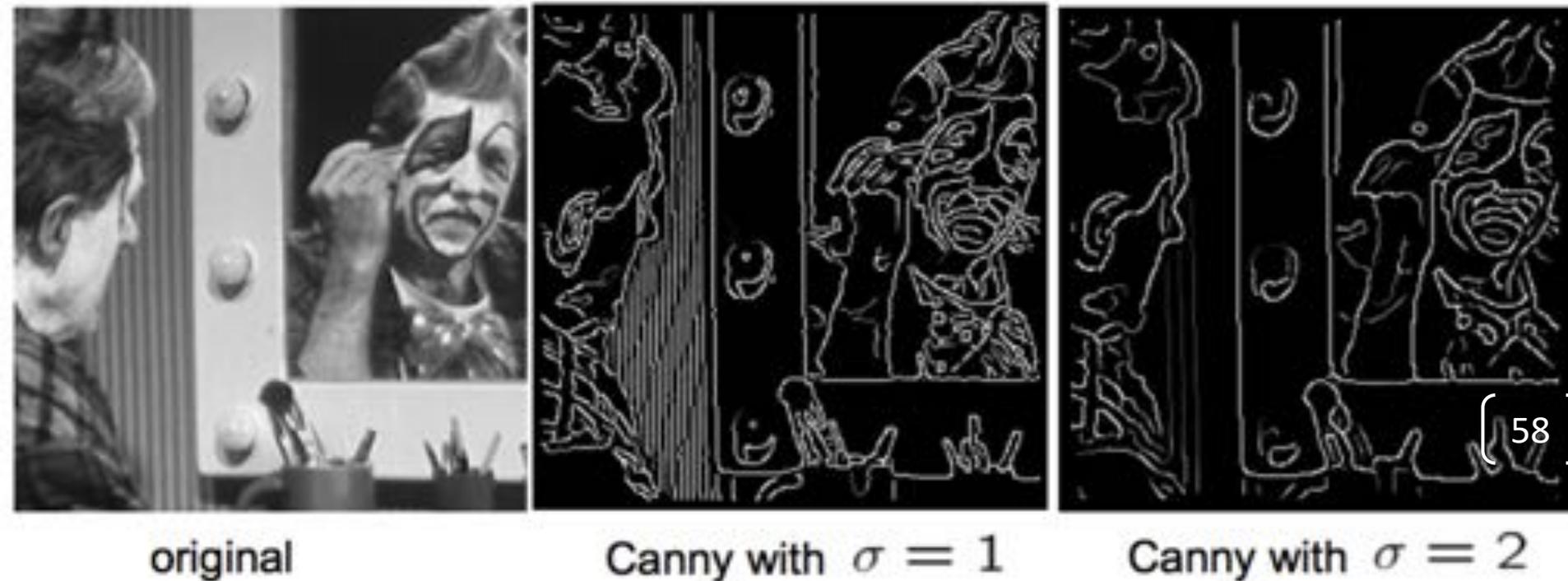
Depois

Resultado do Canny



Efeitos do Kernel Gaussiano

- Quanto maior o desvio, menor será a quantidade de detalhes capturada



ok, temos informação?

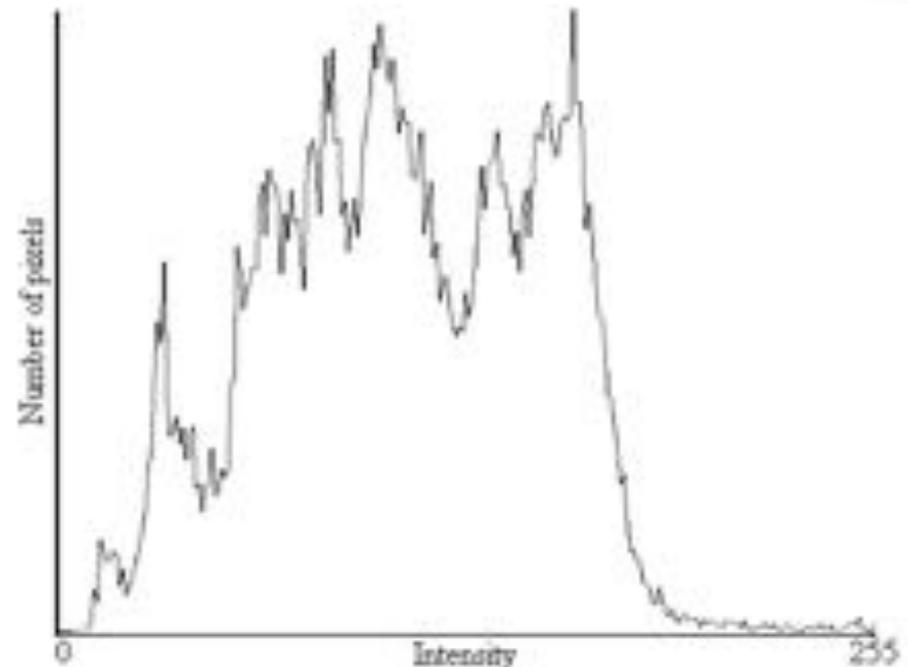
- A partir de imagens realçadas (para obter ou não contorno)
- É possível descrever o que aparece na imagem
- Mas qual método escolher?

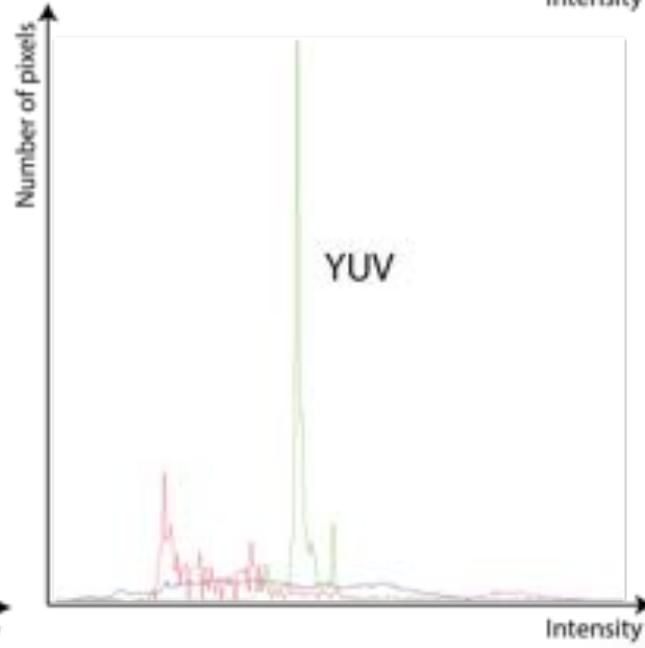
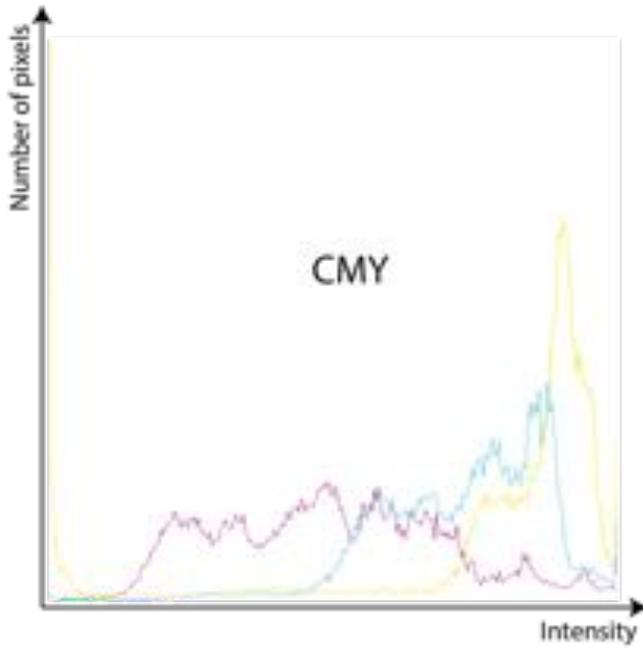
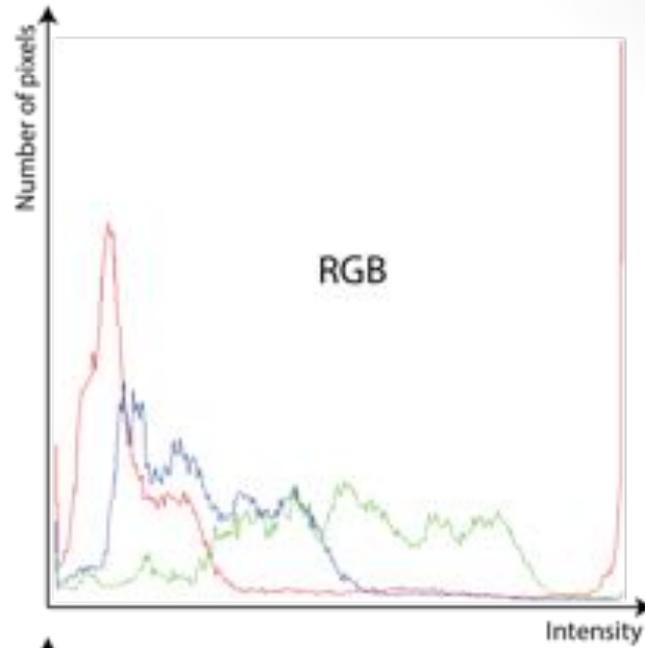
Qual característica você deseja?

- Depende totalmente da aplicação desejada:
 - Textura
 - Forma
 - Estrutura
 - Estatística
- Dentre outras

Quantificação de Cores – Histograma Básico

- O histograma de uma imagem em tons de cinza é uma **função $H(k)$** que produz o número de ocorrências de cada nível de cinza de uma imagem (função de distribuição de probabilidades)





Somente cor, que tal Histograma?

- Informação global da distribuição de tonalidades
 - Probabilidades
- Independente de localização e orientação
- Pode ser usada em problemas de classificação?
 - cuidado, imagens diferentes podem possuir o mesmo histograma

Processo Histograma 1D

Para todo valor de Z : inicialize $H(z) = 0$

onde z varia de 0 a $L-1$, sendo L a quantidade de tons diferentes possíveis para a imagem

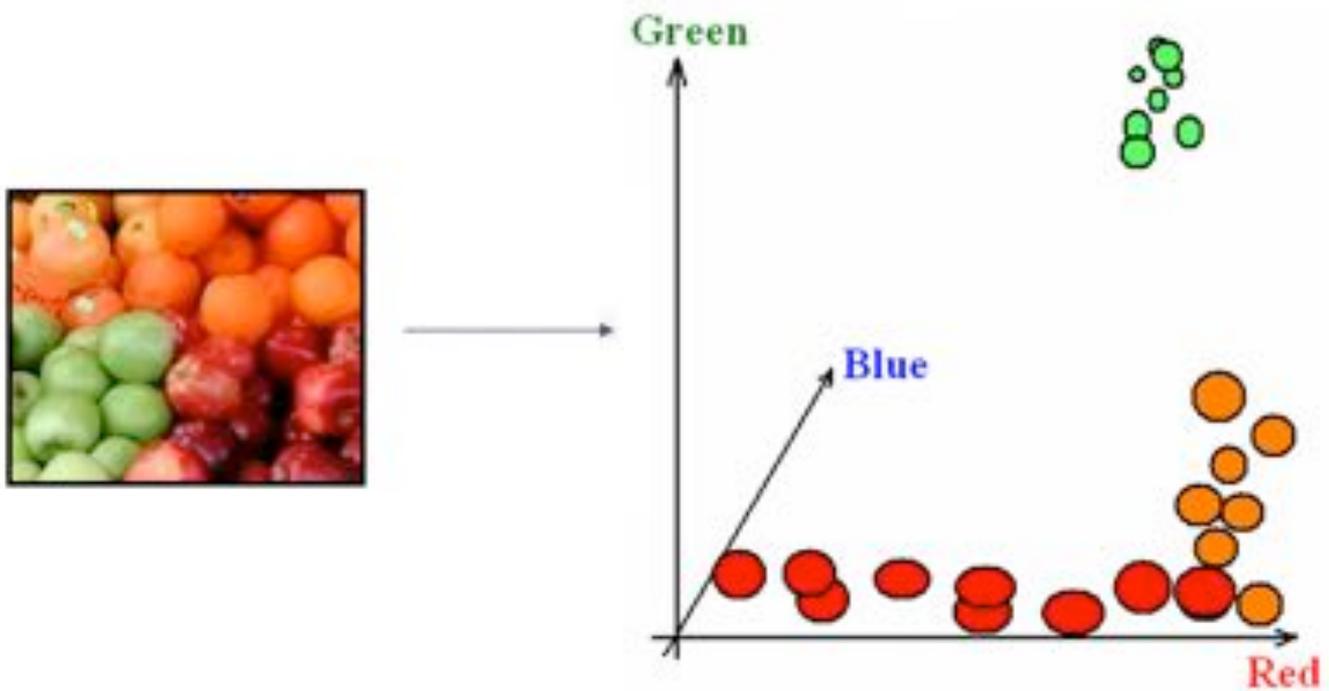
z também é chamado de bin

Para cada pixel da imagem

- $H(\text{pixel}(\text{img}(l,j))) ++;$

E se considerarmos o histograma 3D?

- E utilizar todos os canais ao mesmo tempo?
- O que podemos supor é que o poder de discriminação seja melhor
- **O que fazer com o número de bins do histograma?**

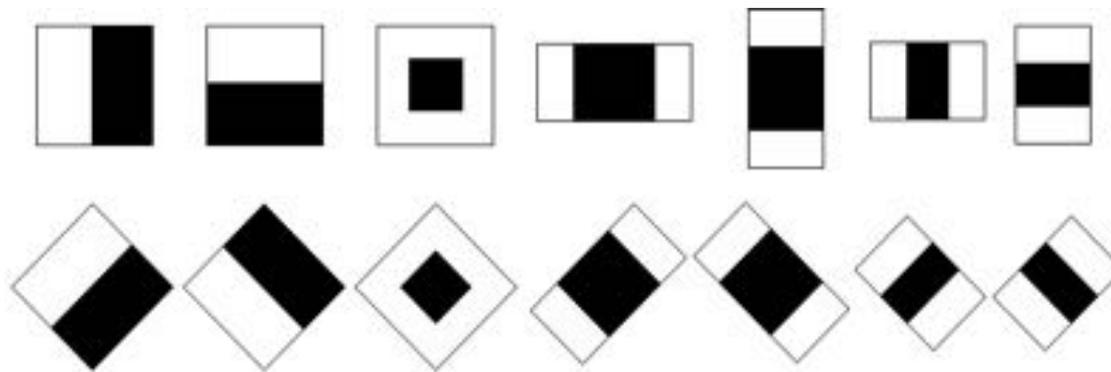


Que medidas tirar de um histograma de cores?

- Além de poder utilizar diretamente os bins, você pode retirar:
 - Variância
 - Contraste
 - Entropia
 - Distribuição relativa de cores
- Mas não é textura

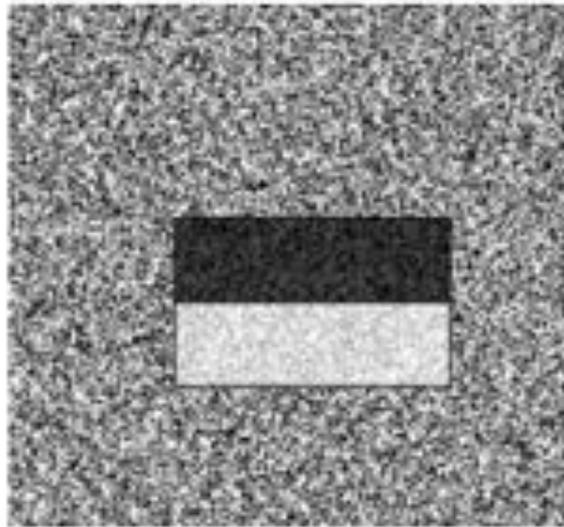
Um pouco mais de distribuição: Haar Features

- Diferença da soma de “brancos” com “pretos”
- O filtro deve ser posicionado na imagem, numa escala específica
- E depois obtido a métrica



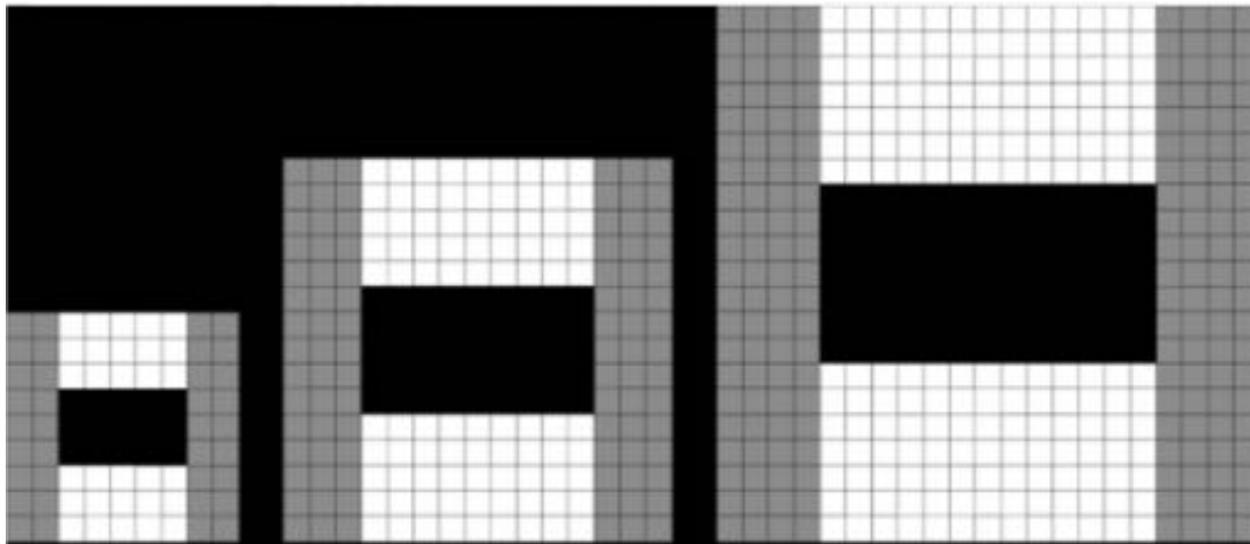
Haar Features

- Melhores respostas em imagens com definição de formas, textura, ...



O tamanho do filtro Haar pode mudar

- Os filtros são organizados em octaves
- Começam com dimensão 9x9 (equivalente a $\sigma=1.2$)
- Incrementos são na ordem de 6



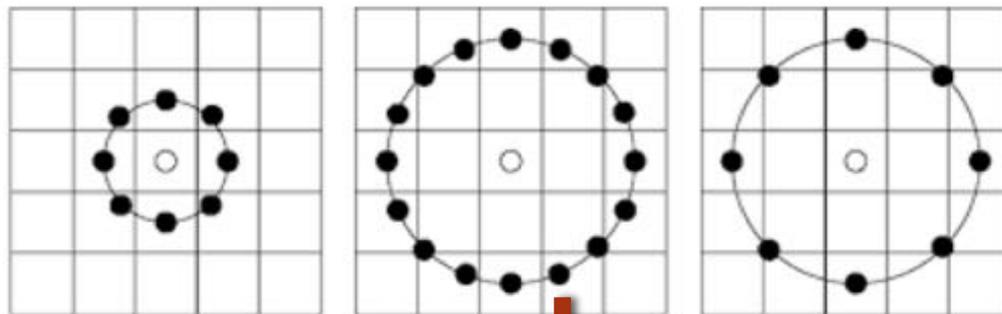
Haar Vector: apenas uma alternativa

- Pode ser de várias maneiras, como todas as contribuições ou:
- Essa região é dividida em 4x4
- Para cada região, Haar Feature na horizontal e vertical é calculada
- As respostas são somadas

$$[\sum dx \quad \sum dy \quad \sum |dx| \quad \sum |dy|]$$

- Totalizando 64 características : $4 \times 4 = 16 * 4 = 64$
- Mas existem várias outras maneiras: alguém pensou em Histograma?

Um pouco de textura: Local Binary Patterns

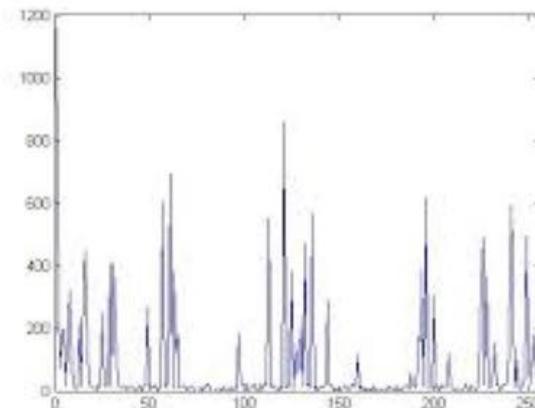


1	2	2
9	5	6
5	3	1

Threshold

0	0	0
1		1
1	0	0

Binary: 00010011
Decimal: 19



Equacionando

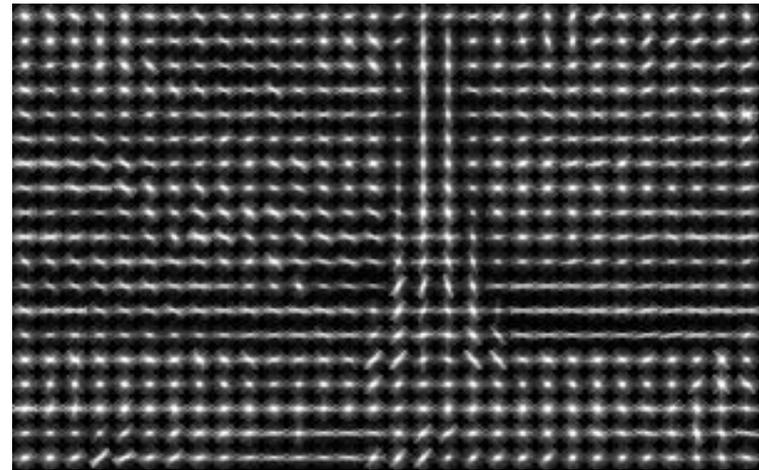
$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(i_p - i_c) 2^p$$

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Local Binary Pattern (LBP)

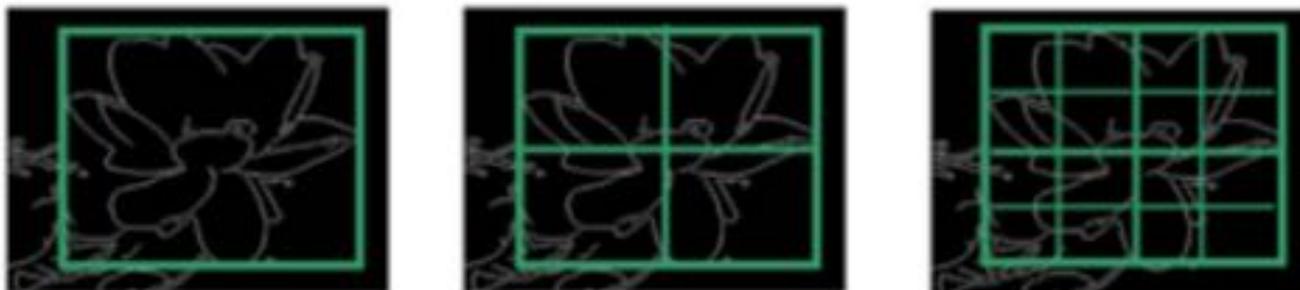
- Prós:
 - Computacionalmente eficiente e robusto a variação monotônica de iluminação.
 - Simples.
- Limitações:
 - Desempenho ruim na presença de ruídos aleatórios
 - Considera apenas o sinal da diferença entre dois valores de cinza, descartando a magnitude.

E se fossem características de Bordas? Histogram Of Gradients



Calculano HOG

- Divida a imagem em blocos de tamanhos iguais
- Cada bloco deve ser dividido em células de igual tamanho



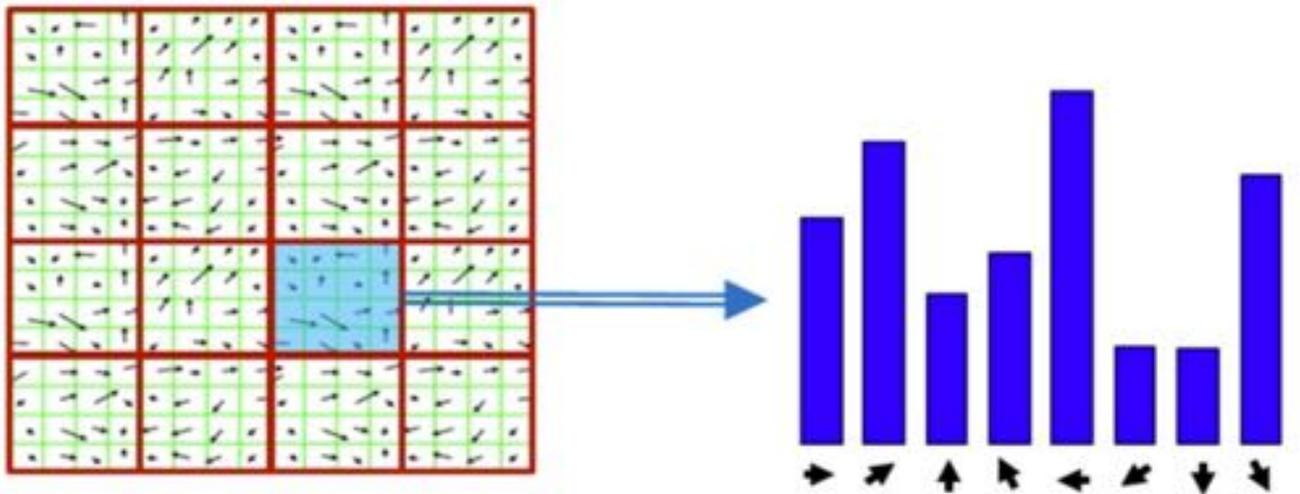
- Qual o tamanho do bloco correto?
- Qual e quantas células são corretas?
- Quanto de sobreposição?
- Qual o shape? R ou C?

Calculano HOG

- Para cada célula, obtenha o histograma de orientações:

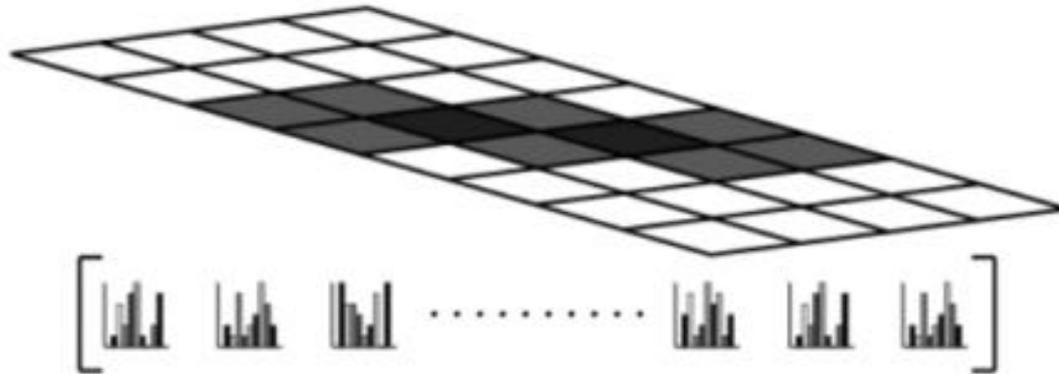
- Obter magnitude:
$$S = \sqrt{S_x^2 + S_y^2}$$

- Obter orientação:
$$\theta = \arctan\left(\frac{S_y}{S_x}\right)$$



Calculano HOG

- Normalize os histogramas e junte todos para gerar um vetor de características



Input image

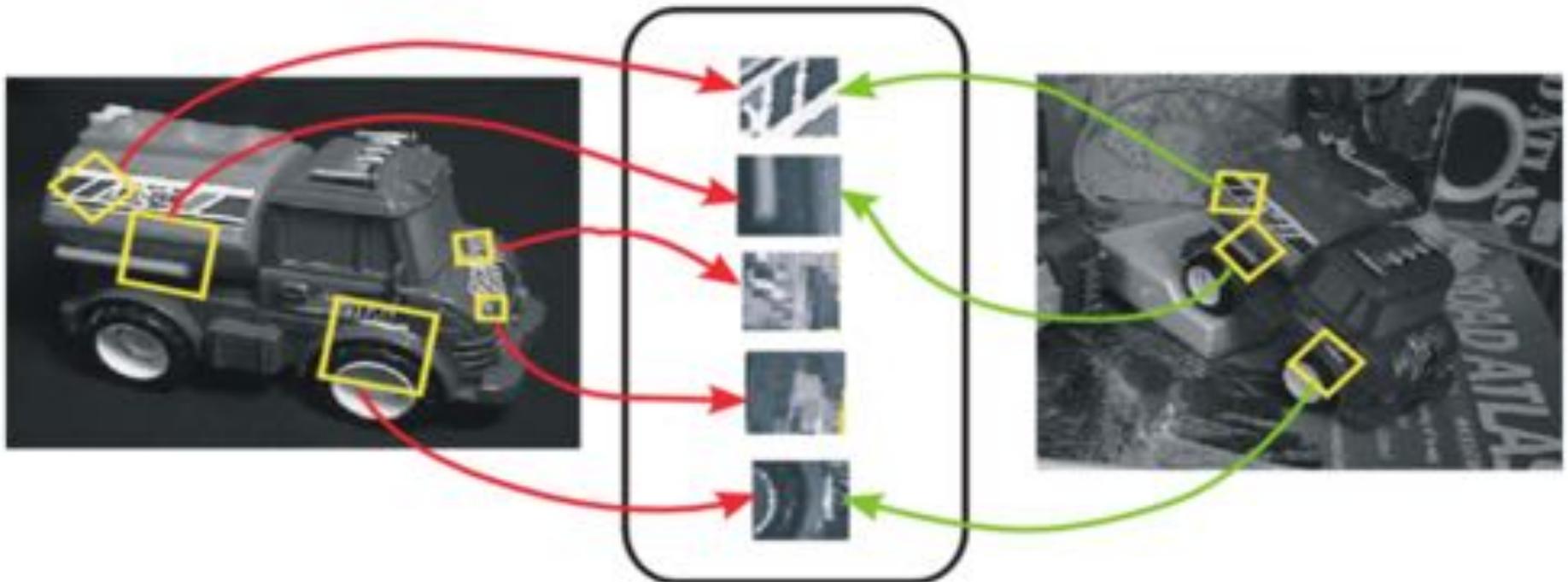


Histogram of Oriented Gradients



Onde usar as características?

- Onde está o caminhão de brinquedo?
- **Matching**



Pergunta:

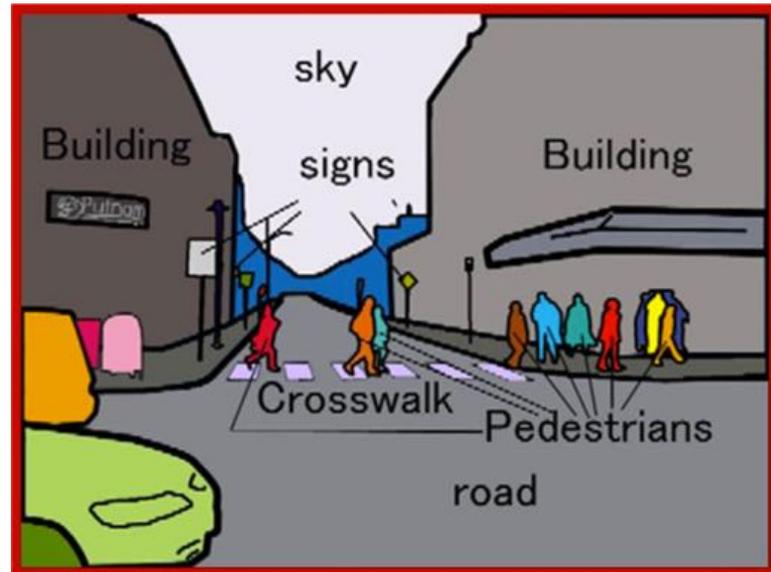
- Essa imagem possui carro(s)?
- **Detecção de Objetos**

Vários



Pergunta:

- O que essa cena apresenta?
- **Segmentação** (em alguns casos, semântica) – localização precisa

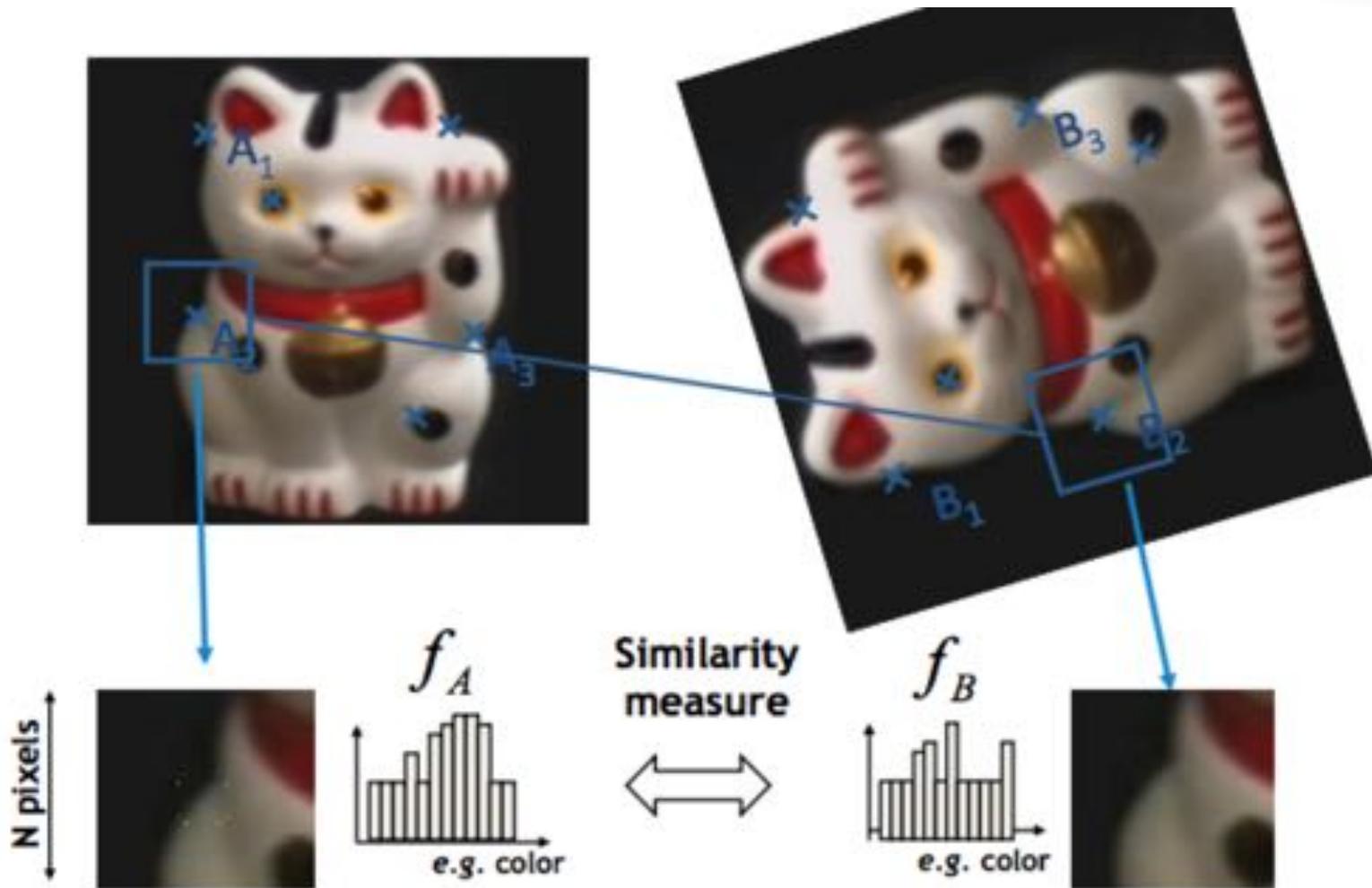


Matching com Histogramas



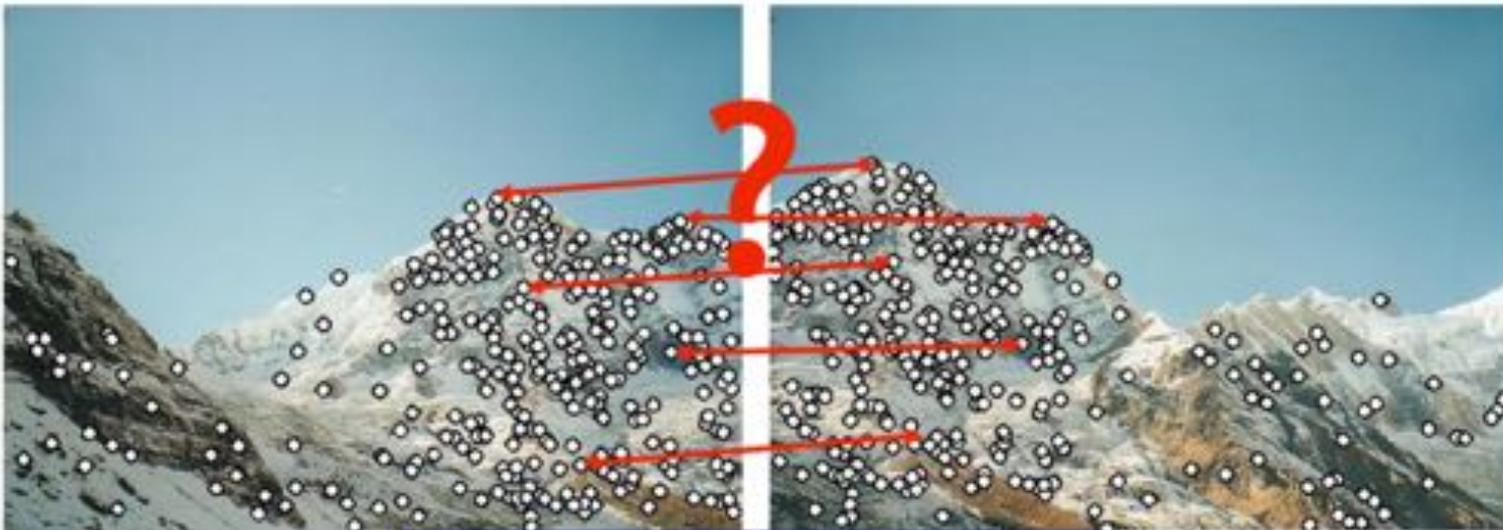
```
calcHist( &imageROI, 1, channel_numbers, Mat(), histogram,  
image.channels(), number_bins, channel_ranges );  
normalize( histogram, histogram, 1.0);  
calcBackProject(&image,  
1,channel_numbers,histogram,result,channel_ranges,255.0);
```

Matching com abordagem Local



Keypoints?

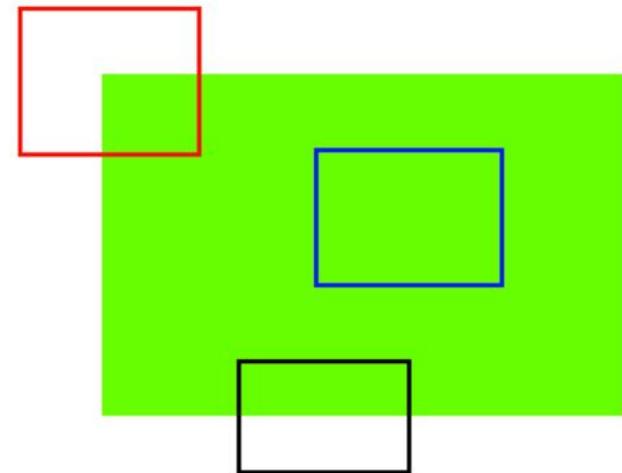
- Pontos Chave são **Importantes!**
- Independente da imagem, o algoritmo extrator deve extrair os keypoints sempre da mesma maneira
 - Consistência
 - Repetição



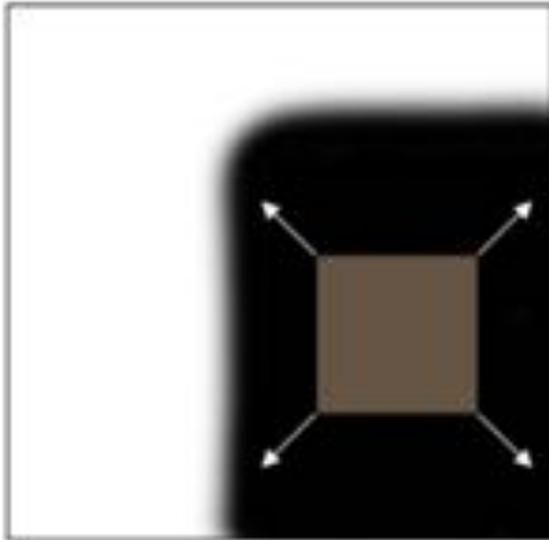
Características para o método

- Repetição (independência de imagem)
- Invariante: translação, escala e rotação
- Robusto
- Eficiente
- Discriminativo
- Quantitativo
- Local

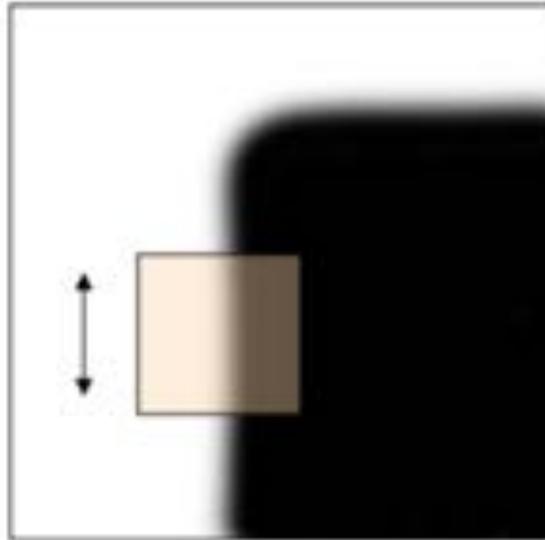
O que são bons candidatos a keypoints?



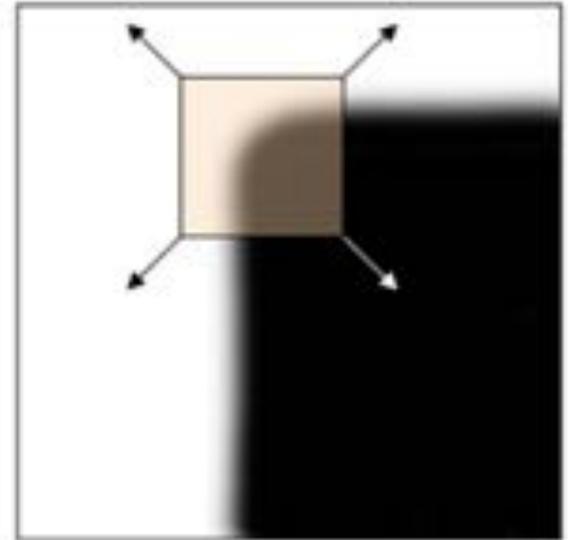
Distinção



Interior: sem variações em todas as direções



borda: variações na direção da borda



canto: mudanças bruscas em todas as direções

Harris Detector

- Ideia da janela
- Maximizar

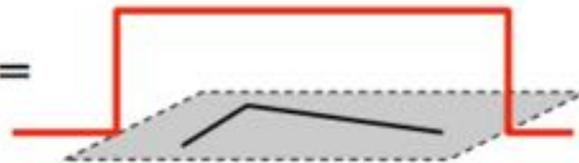
$$E(u, v) = \sum_{x, y} w(x, y) [I(x + u, y + v) - I(x, y)]^2$$

Window function

Shifted intensity

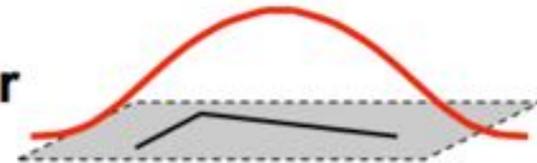
Intensity

Window function $w(x, y) =$



1 in window, 0 outside

or



Gaussian

Harris Detector

- Pode ser aproximada por:

$$E(u, v) \approx [u \ v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

- onde M significa

$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

- I_x e I_y são os gradientes em x e y

- Essa é uma matriz de covariância

Harris Detector

$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y]$$

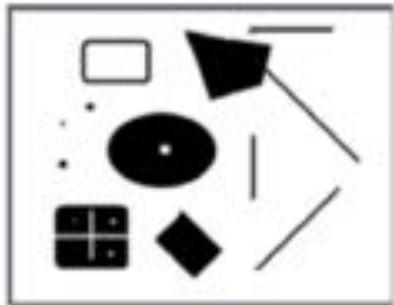


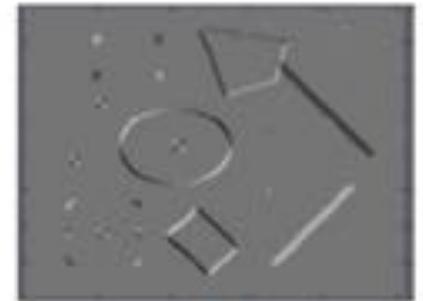
Image I



I_x



I_y



$I_x I_y$

Pode-se usar Sobel para calcular o gradiente

Harris Detector

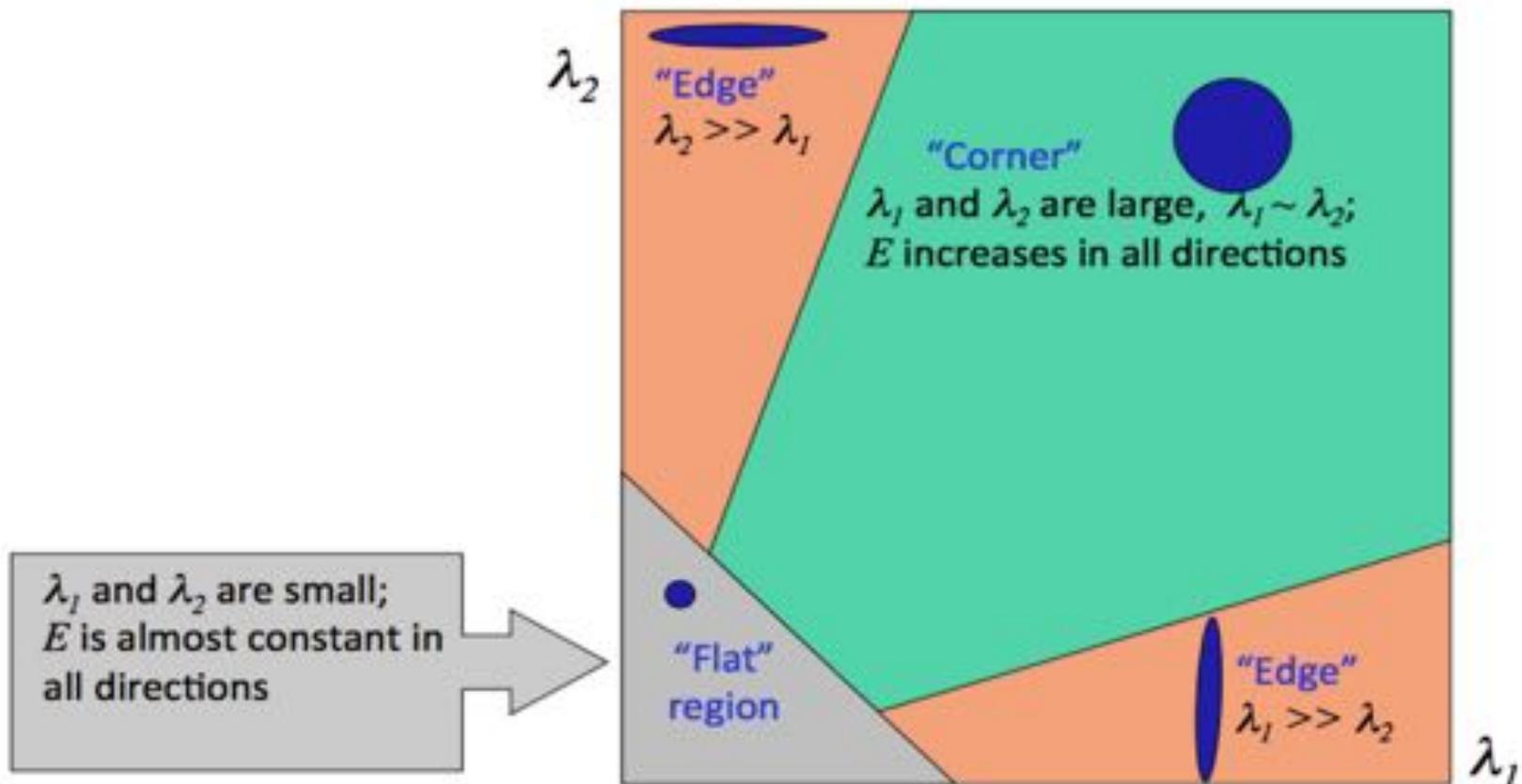
- A equação anterior gera um score de cada janela testada.
- Para saber se a janela possui um canto:

$$R = \det(M) - k(\text{trace}(M))^2$$

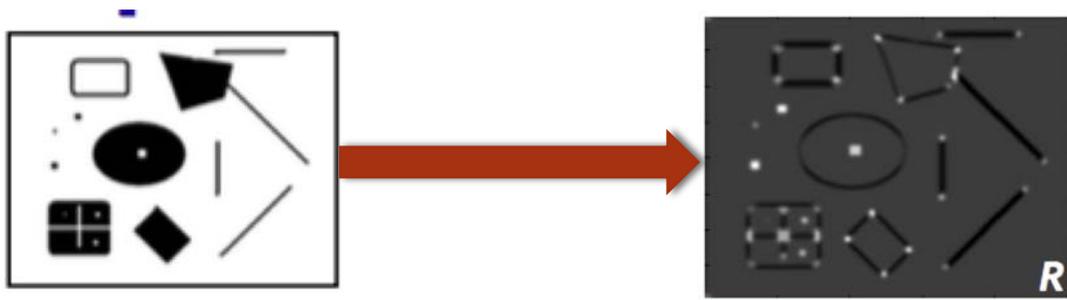
- onde:
 - $\det(M) = \lambda_1\lambda_2$
 - $\text{trace}(M) = \lambda_1 + \lambda_2$
 - k é uma constante que varia entre 0.04 e 0.06
 - λ_1 e λ_2 são os autovalores de M

Harris Detector

- Os valores dos autovalores decidem se a determinada janela possui ou não um canto:
 - quando R é grande (que acontece quando λ_1 e λ_2 são grandes) então a janela possui um canto

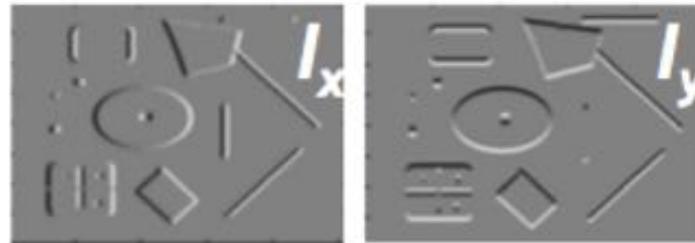


Passo a Passo



- Threshold do máximo
- ou
- non-maximum suppression

1. Image derivatives



2. Square of derivatives



3. Gaussian filter $g(\sigma_l)$

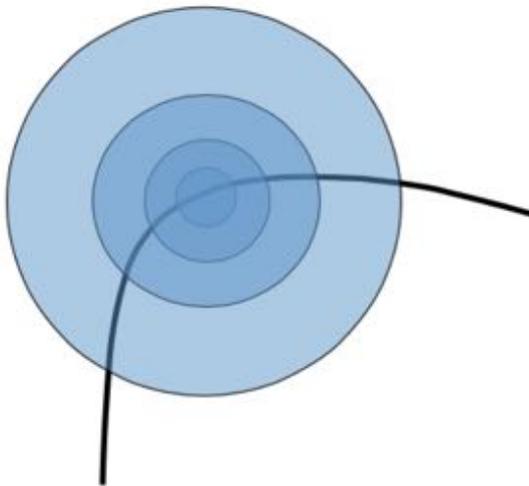
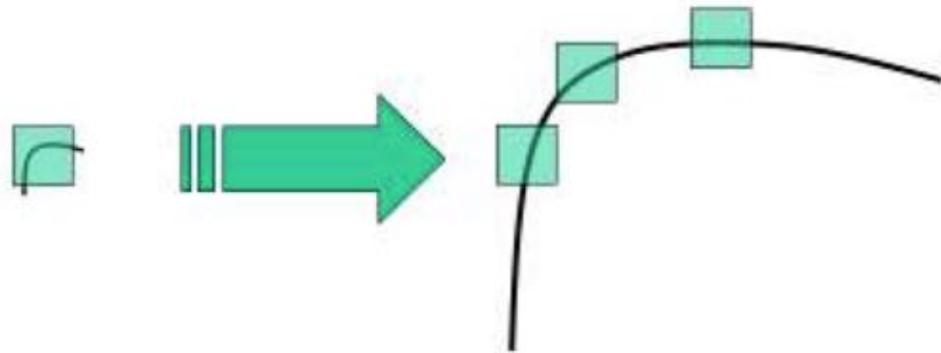


Harris



Problema

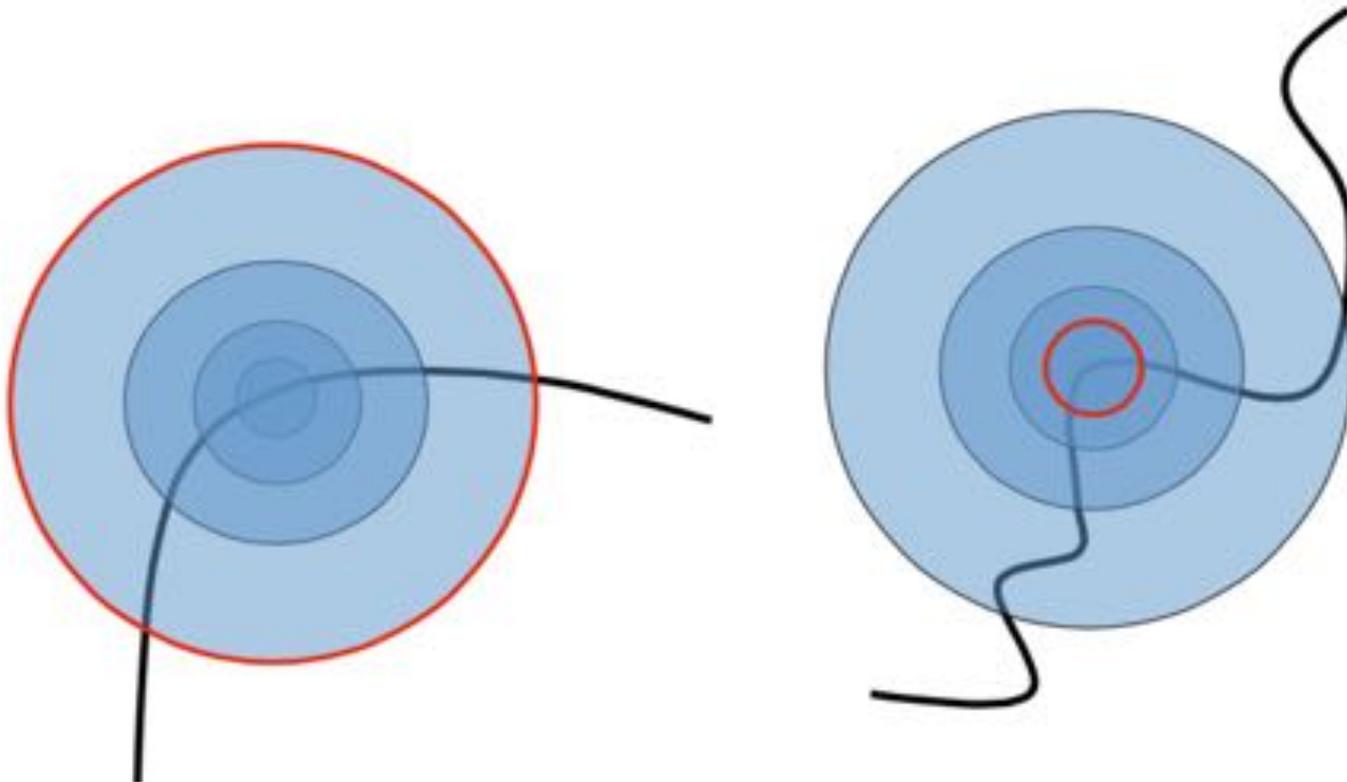
- Variação de escala. Um **canto** nem sempre é canto numa escala diferente



Quando a escala diminui, a janela pode identificar que o canto agora é apenas borda

Problema

- Como achar a janela (ou círculo neste caso) do tamanho correto, independente da imagem?



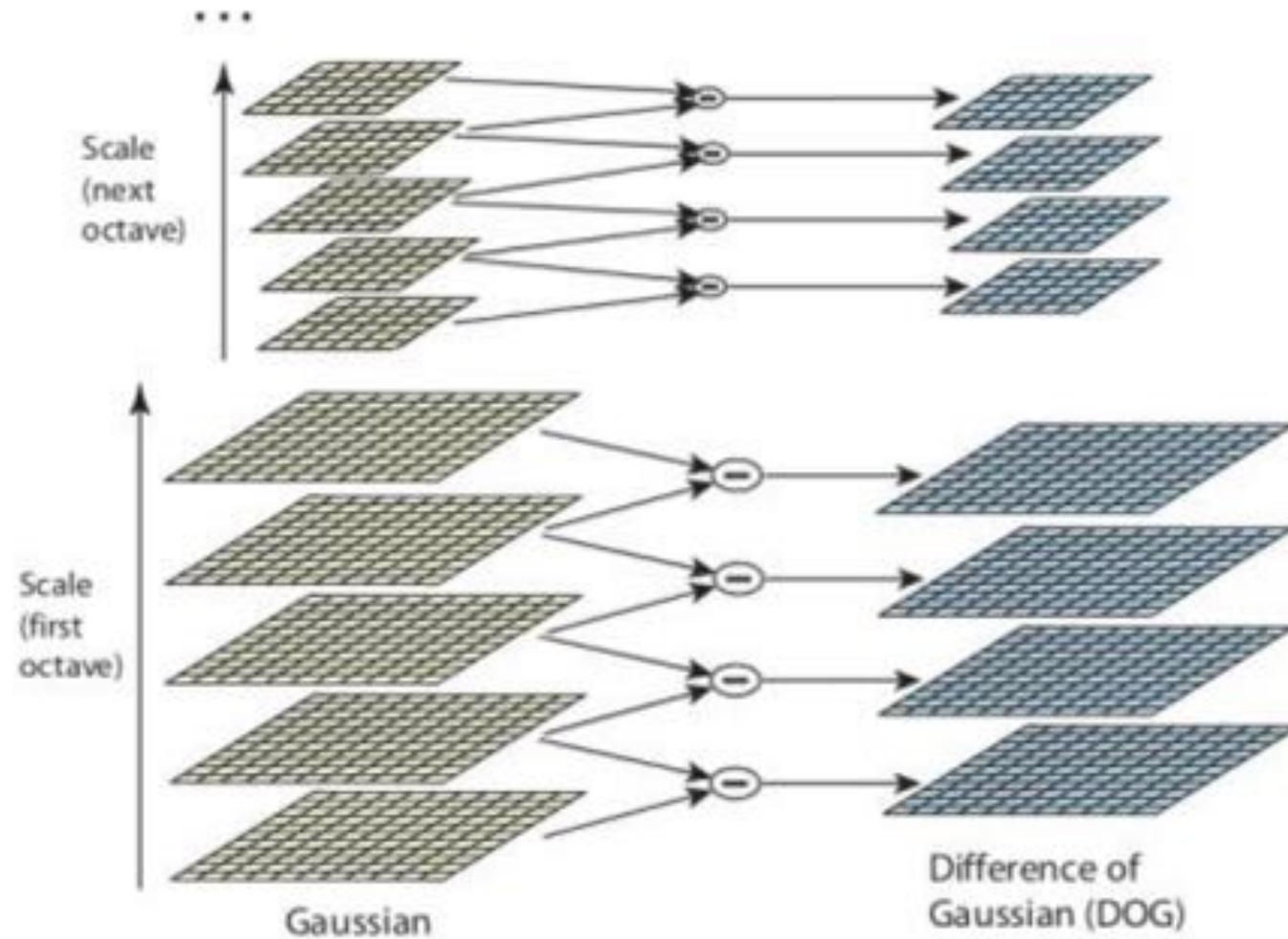
SIFT Princípio

- Scale-Space Filtering
 - Aplica **Difference of Gaussians (DoG)** -> aproximação de LoG e mais rápido
- Efeito: borramento da imagem
 - Quanto maior o borramento, existe a simulação de baixa escala
- A diferença é calculada sobre imagens que passaram por gaussianas σ e $k\sigma$

Espaço de Escala - exemplo

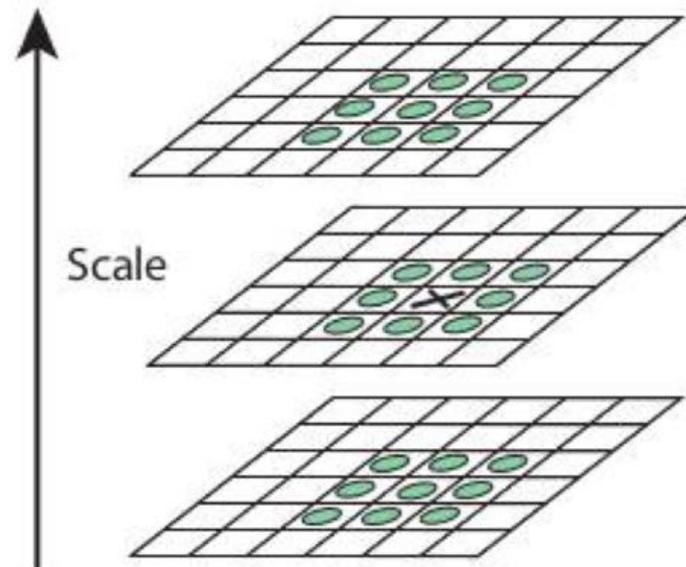


DoG



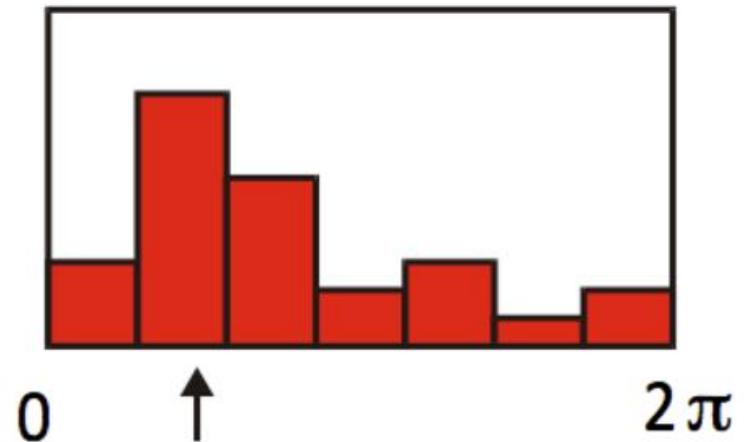
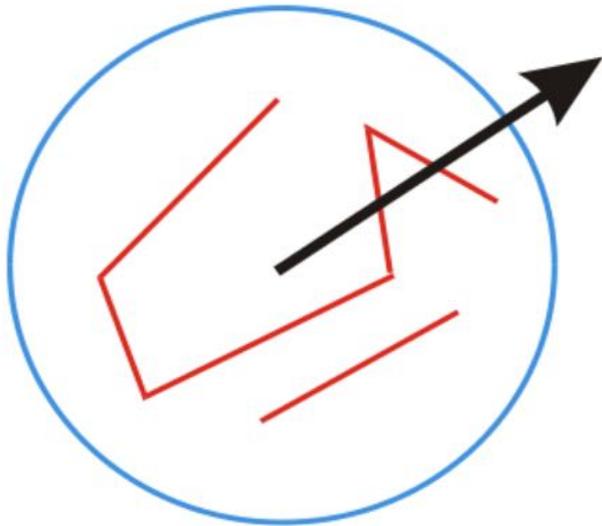
Local Extrema

- Com a DoG construída, procura-se pelos pontos de extrema
- Esse ponto é maior que os seus 27 vizinhos de diferentes escalas
- Representa um **possível keypoint**

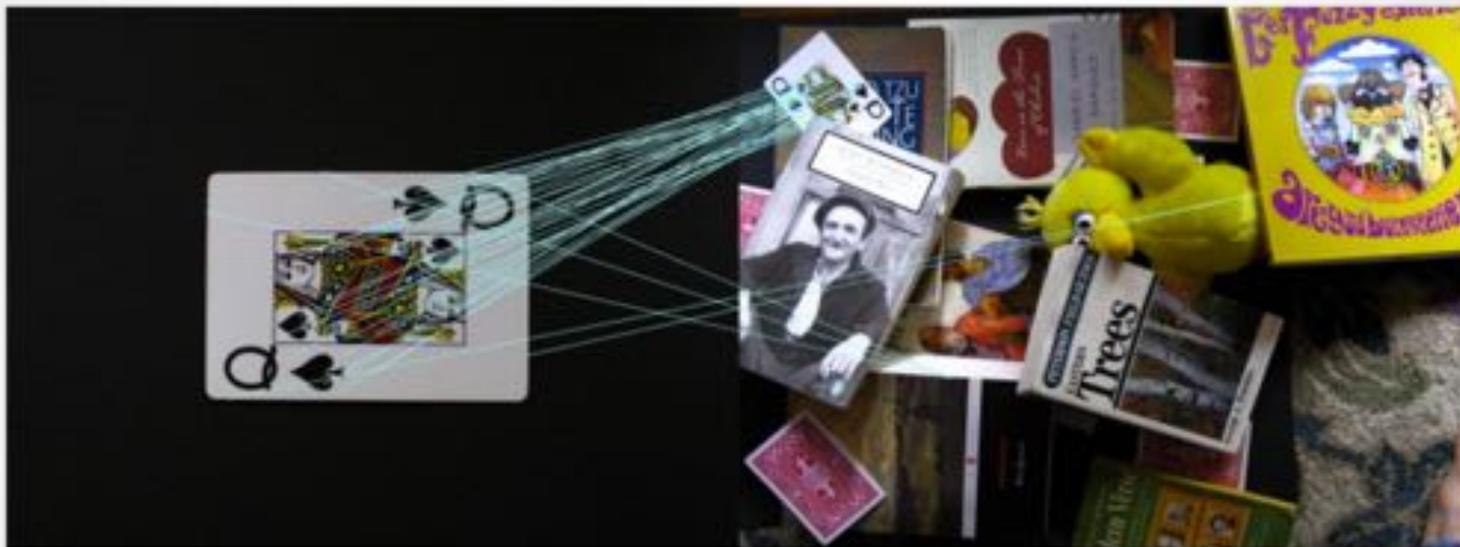
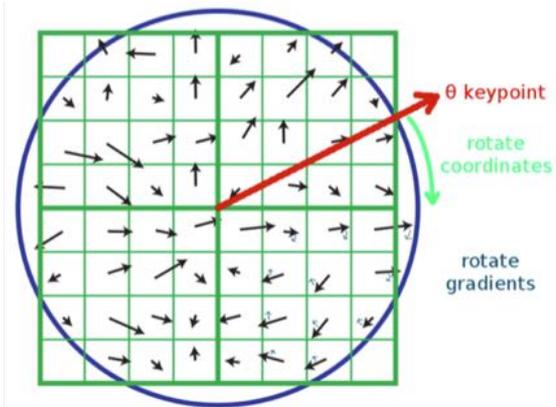
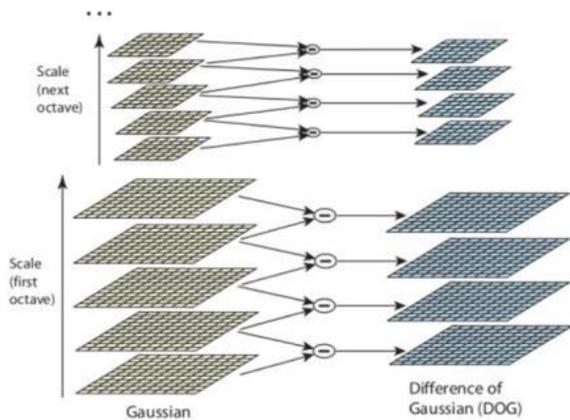


Invariante a Rotação

- A orientação do **possível keypoint** é obtida através da análise da orientação dos contornos próximos
- Orientação do keypoint = pico de orientação no histograma direcional ponderado pela magnitude



SIFT (DoG + HOG)

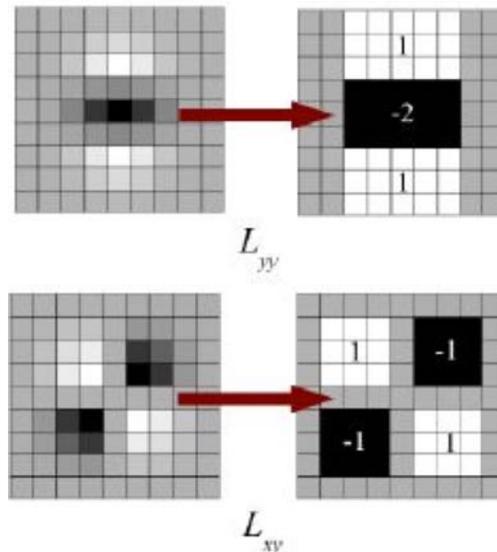


SIFT e placas de transito



Surf: A ideia

- SIFT relativamente lento
- No SIFT:
 - Aproxima-se Laplaciano de Gaussiana com DoG
- No SURF
 - A aproximação é feita com BoxFilter



Bay, H., Tuytelaars, T. and Van Gool, L, published another paper, “SURF: Speeded Up Robust Features”

Obtendo os pontos de interesse

- Baseado em Integral Images

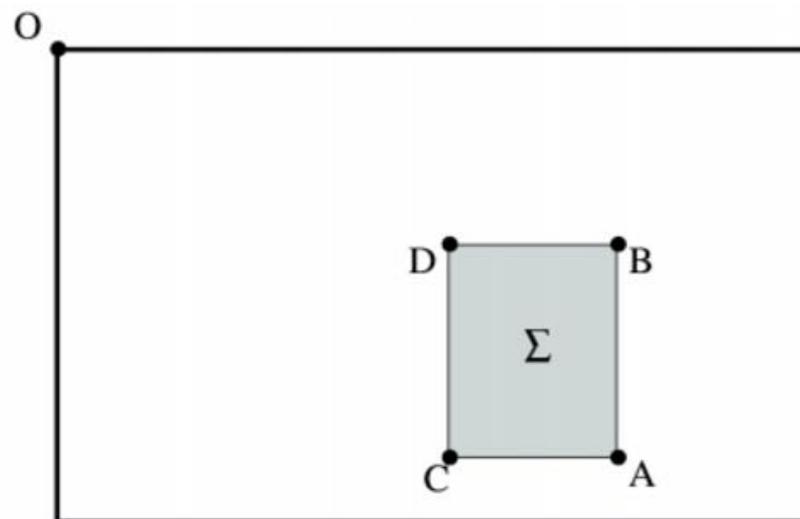
$$I_{\Sigma}(\mathbf{x}) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \quad \mathbf{x} = (x, y)^{\top}$$

1	2	2	4	1
3	4	1	5	2
2	3	3	2	4
4	1	5	4	6
6	3	2	1	3

input image

0	0	0	0	0	0
0	1	3	5	9	10
0	4	10	13	22	25
0	6	15	21	32	39
0	10	20	31	46	59
0	16	29	42	58	74

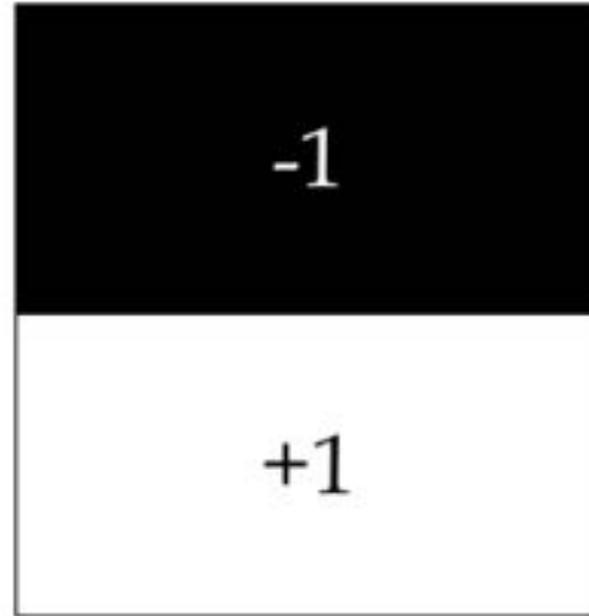
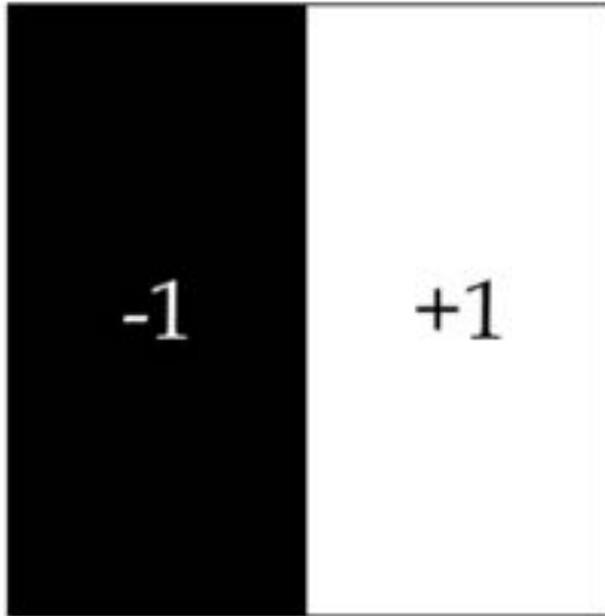
integral image



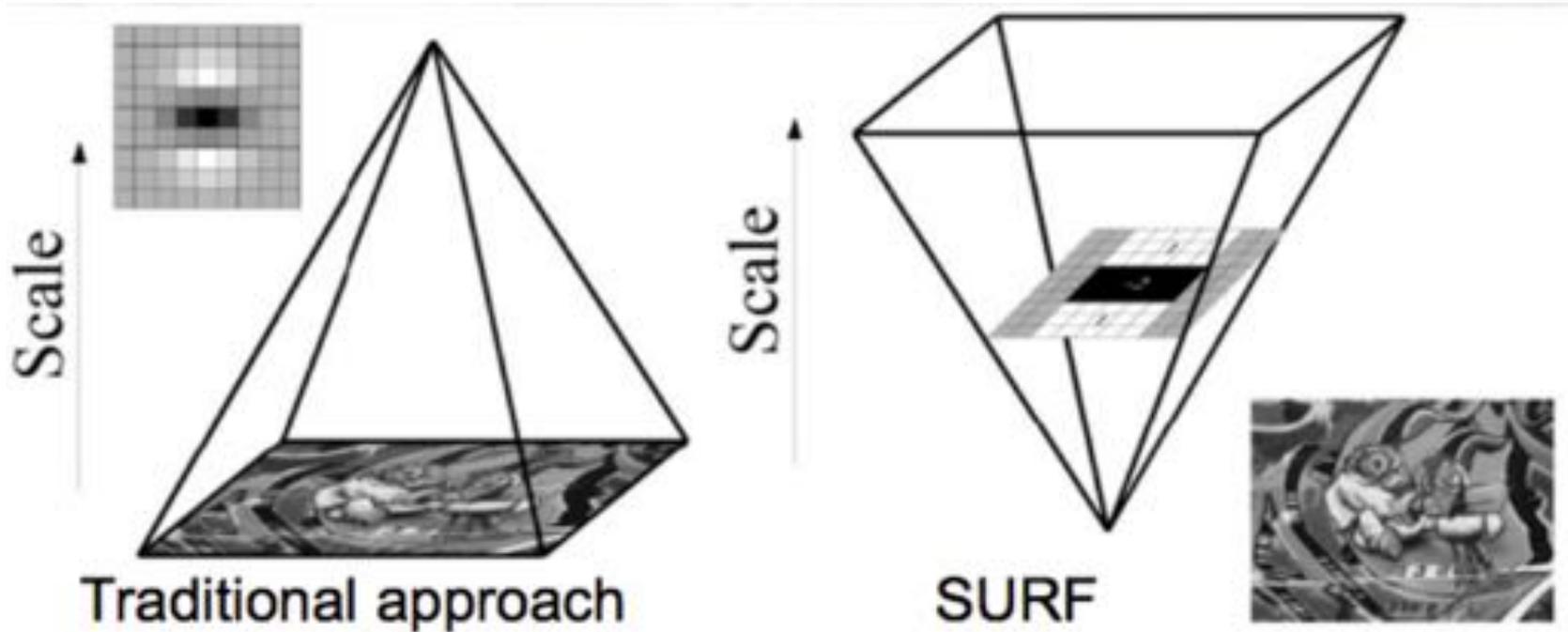
$$\Sigma = A - B - C + D$$

Descritores

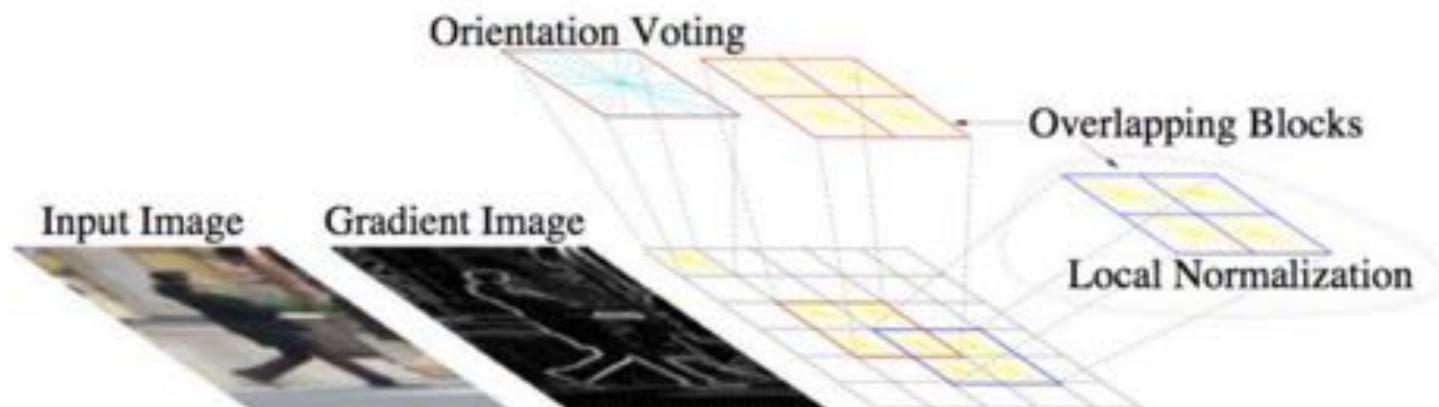
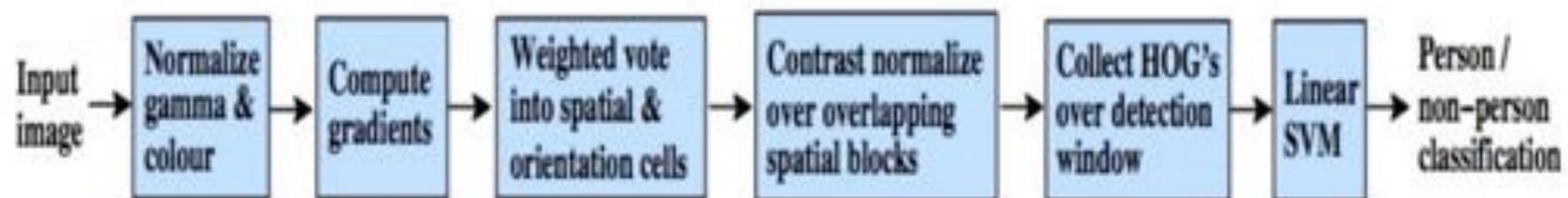
- Contêm basicamente a resposta a primeira ordem de Haar Waveltes
 - Calculados em x e y (acha gradientes)
 - Novamente sobre as imagens integral

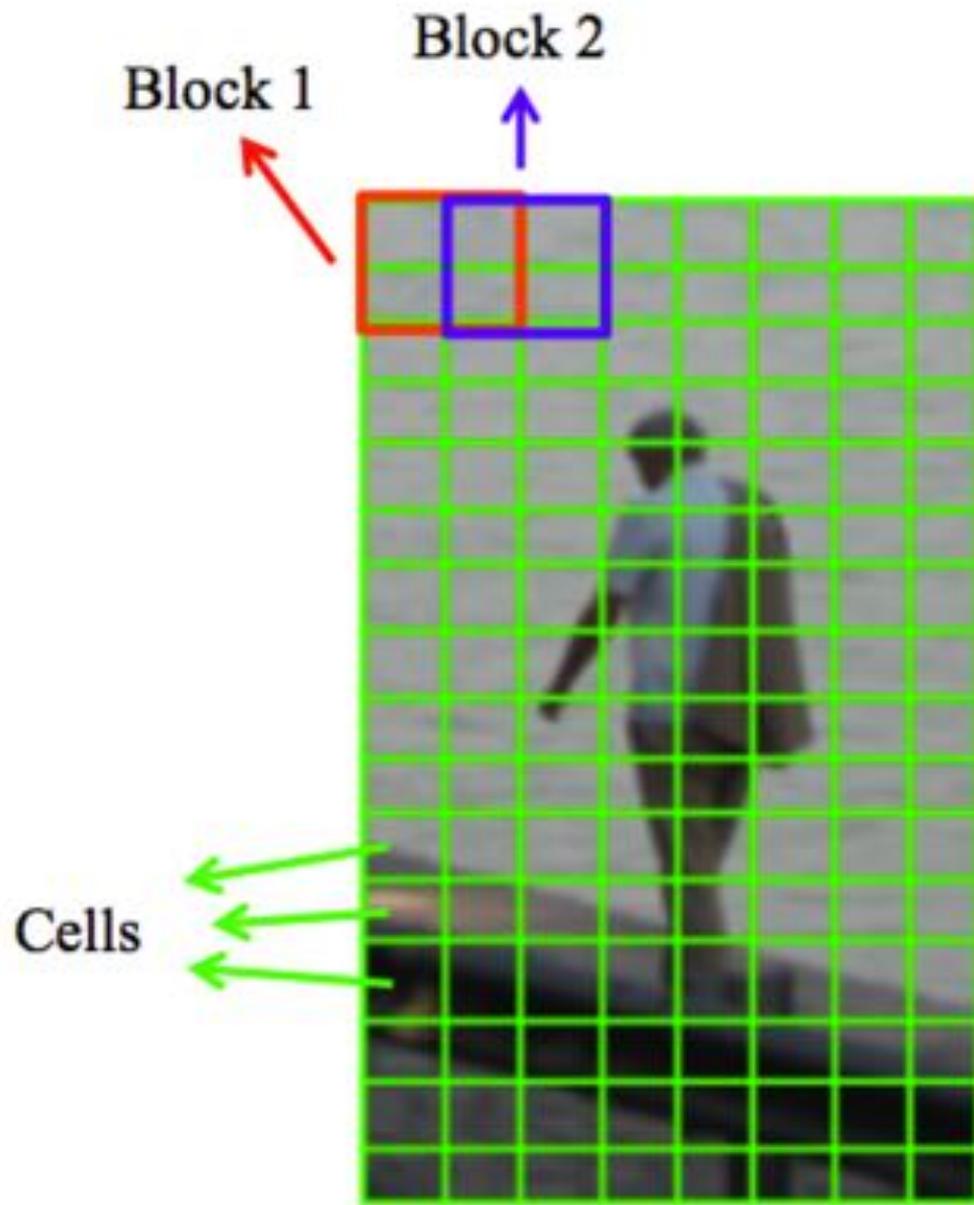


SURF (BoxFilter + Haar Features)



Detecção de Pessoas com HOG





Colha Resultados

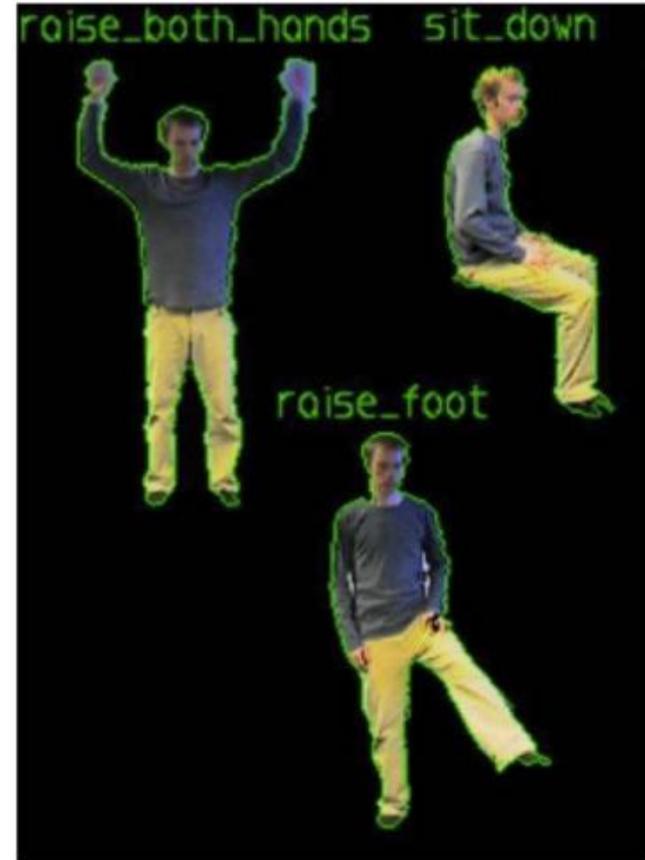
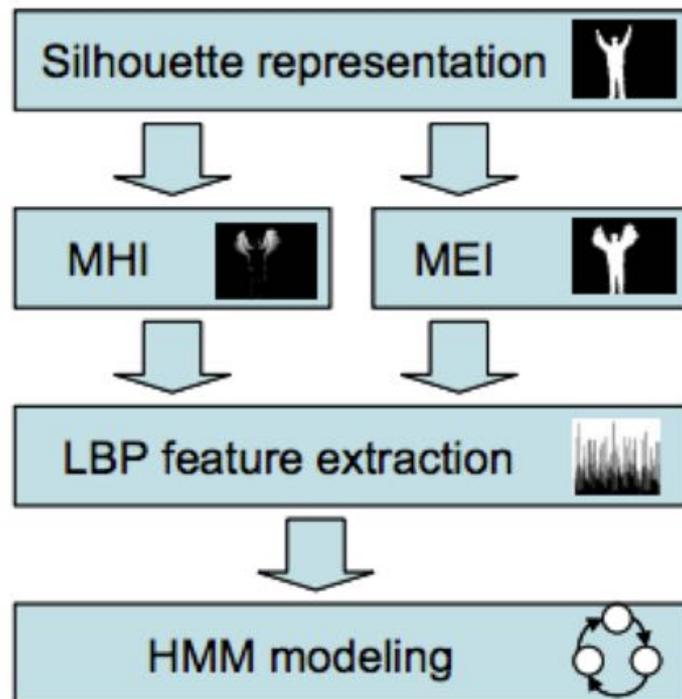


Porque não em outras classes?



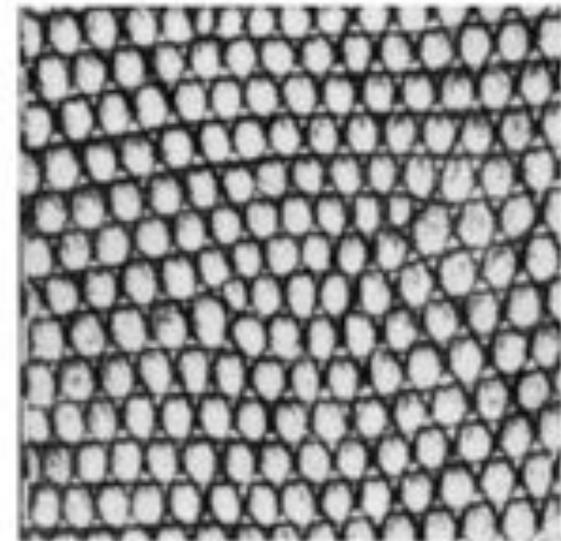
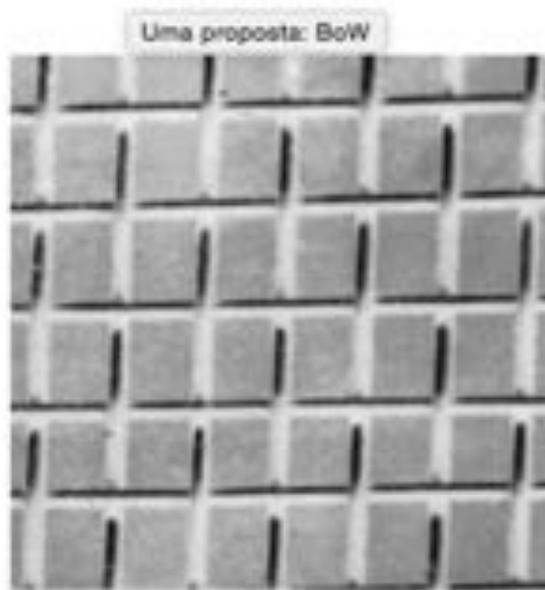
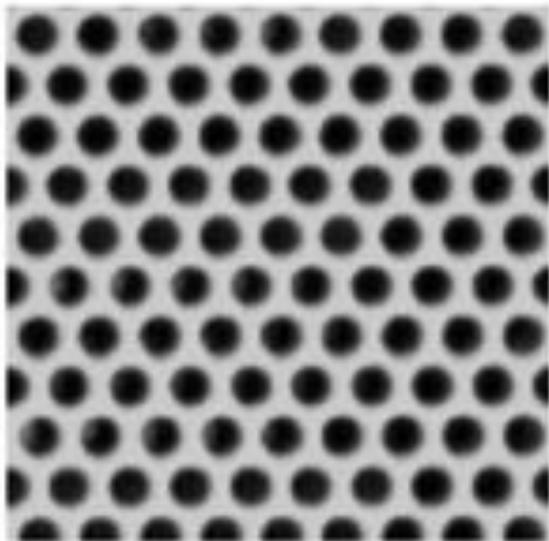
LBP para detecção de atividades

- Kellokumpu V, Zhao G & Pietikäinen M (2009) Recognition of human actions using texture. Machine Vision and Applications (available online).

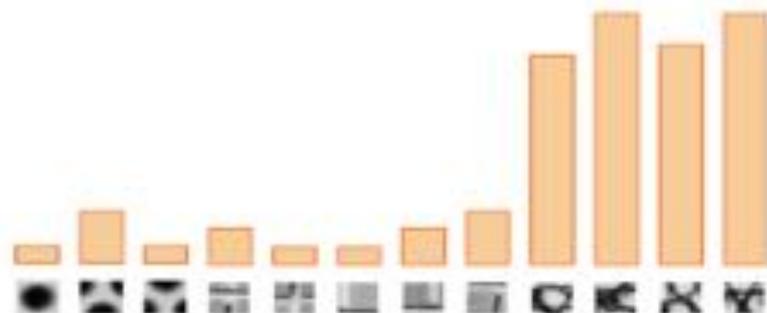
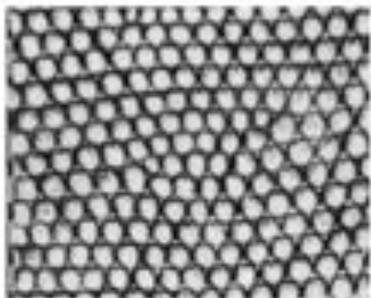
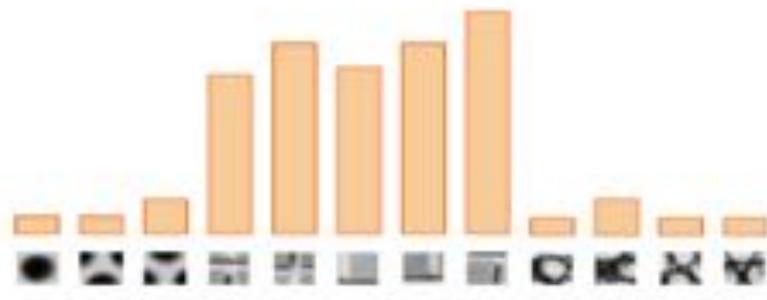
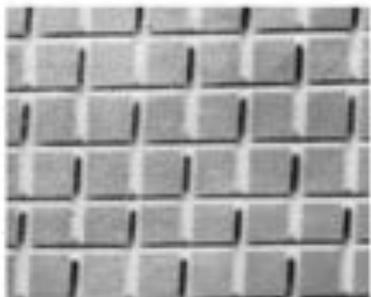
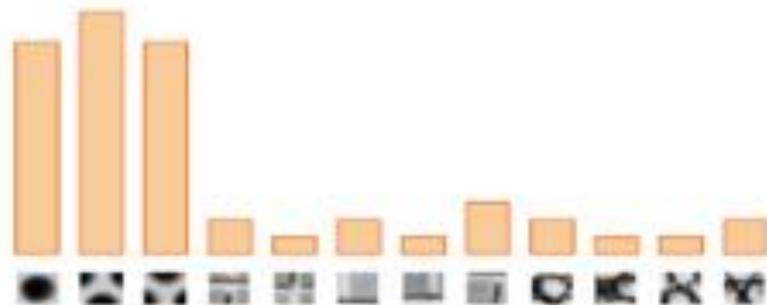
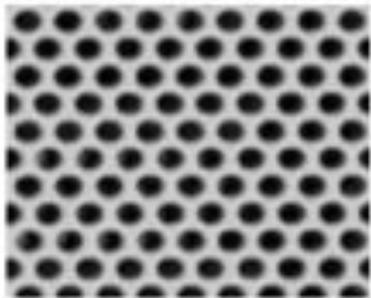


E quando as features se repetem, mas não se sabe a quantidade?

- Reconhecimento de Textura
 - repetição ao longo da imagem de textons



Contagem de Textons



BoW



Bag Of Words - BOW

- Representação da **frequência** acumulada de **palavras** em documentos diferentes a partir de um **dicionário** pré-criado

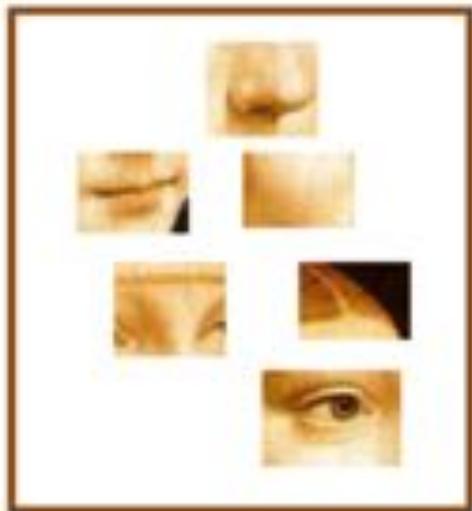
2007-01-23: State of the Union Address

George W. Bush (2001-)

abandon accountable affordable afghanistan africa aided ally anbar armed army **baghdad** bless **challenges** chamber chaos
choices civilians coalition commanders **commitment** confident confront congressman constitution corps debates deduction
deficit deliver **democratic** deploy dikembe diplomacy disruptions earmarks **economy** einstein **elections** eliminates
expand **extremists** falling faithful families **freedom** fuel **funding** god haven ideology immigration impose
insurgents iran **iraq** islam julie lebanon love madam marine math medicare moderation neighborhoods **nuclear** offensive
palestinian payroll province pursuing **qaeda** radical regimes resolve retreat rieman sacrifices science sectarian senate
september **shia** stays strength students succeed sunni **tax** territories **terrorists** threats uphold victory
violence violent **war** washington weapons wesley

Para imagem: Bag of Features

- Inicialmente, extraia features (partes)



Para imagem: Bag of Features

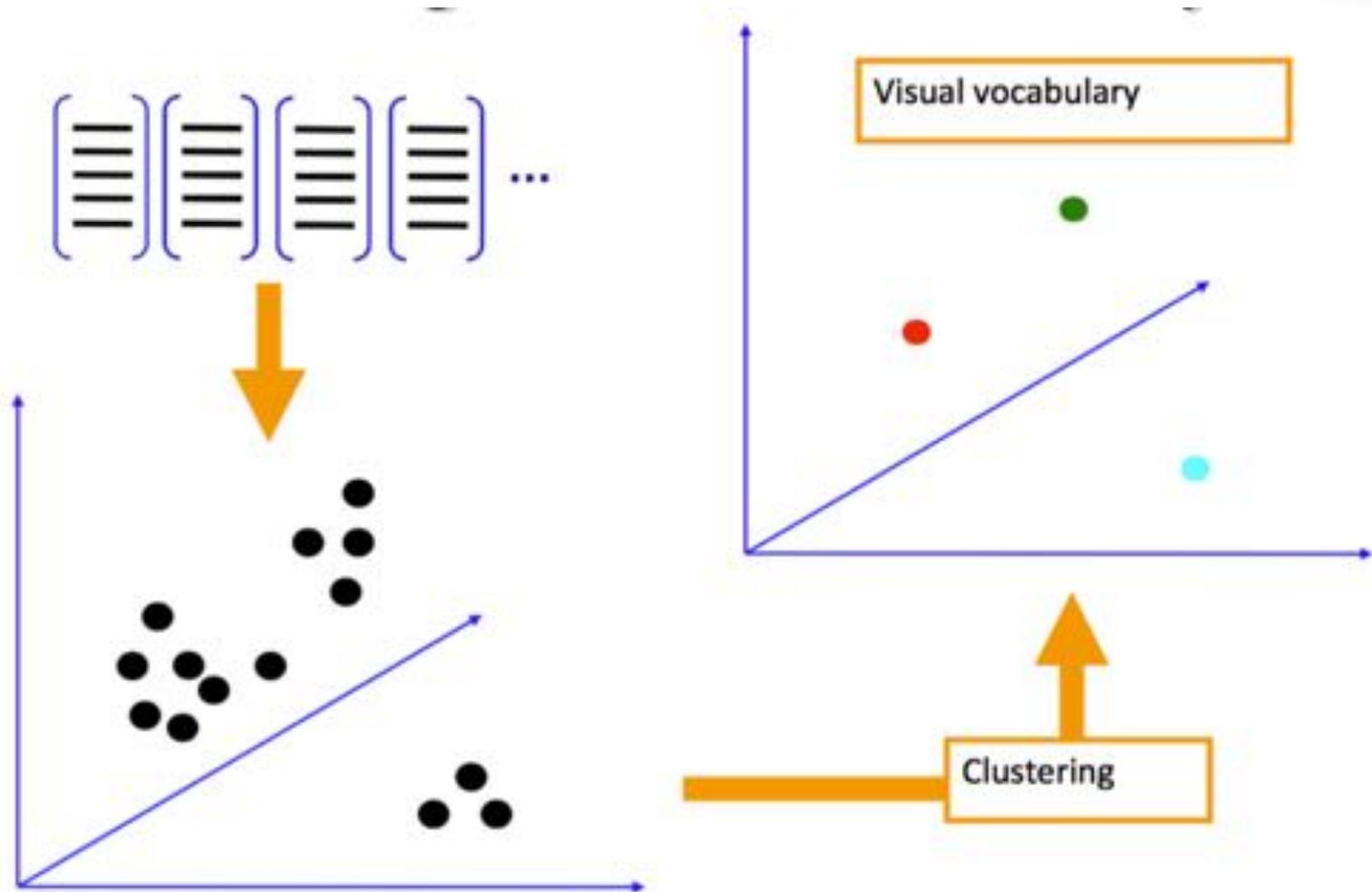
- Aprenda um **vocabulário** visual de **partes**



Para imagem: Bag of Features

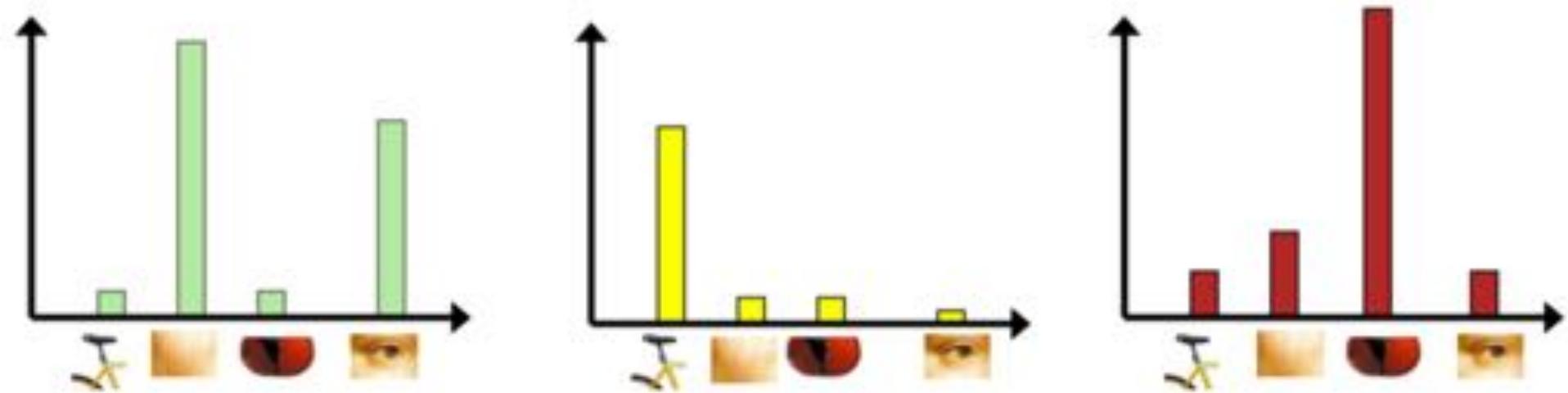
- Se utiliza alguma técnica não supervisionada de aprendizado para fazer o vocabulário
- A mais comum, **kmeans**
 - Classifica o grupo de amostras em K grupos por proximidade
 - **K neste caso é o tamanho do vocabulário!**

Para imagem: Bag of Features



Para imagem: Bag of Features

- Quantize as partes em relação ao **vocabulário**
 - Representar as imagens na forma de frequências (histograma) de partes do vocabulário



Para imagem: Bag of Features

1. Inicialmente, extraia features (**partes – SIFT, SURF ...**)
2. Aprenda um **vocabulário** visual de **partes**
3. Quantize as partes em relação ao **vocabulário**
 - Representar as imagens na forma de frequências (histograma) de partes do vocabulário
4. Parta para o aprendizado de máquina usando a representação do BOW

Etapas

