

Sift

Scale Invariant Feature

Transform

Pablo F. Dias

➤ Contextualização

➤ Idéia

➤ Implementação

➤ Considerações

➤ Referências

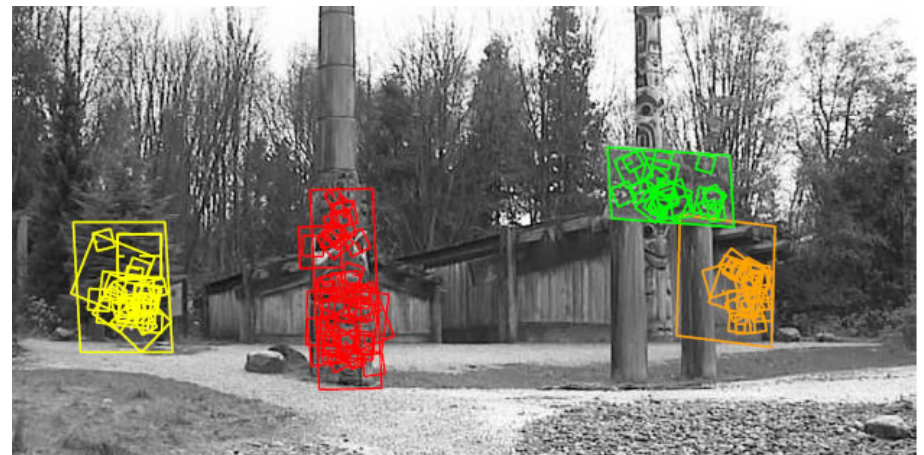
Contextualização

- Encontrar características comuns em diferentes imagens
- Definir um método que atenda a diferentes escalas, rotações, iluminações, ponto de vista
- Fundamental para diversos problemas de processamento de imagens e visão computacional
 - Composição de imagens, reidentificação, rastreamento de trajetória, reconhecimento de gestos.
- Distinctive image features from scale-invariant keypoints. David G. Lowe, International Journal of Computer Vision, 60, 2 (2004), pp. 91-110



Contextualização

- Busca pontos de interesse
- Invariante em relação a:
 - Escala, rotação
- Parcialmente invariante em relação a:
 - Iluminação, ponto de vista, oclusão

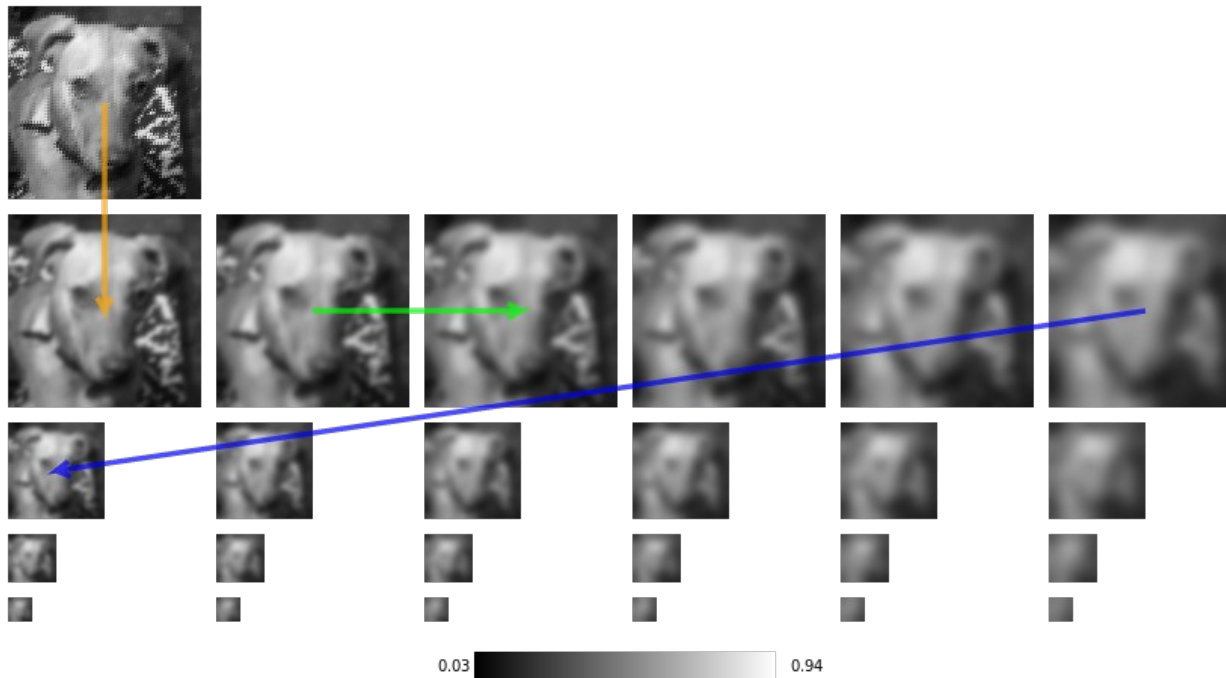


Idéia

- Algoritmo:
 - Gerar um espaço de escala
 - Aproximação usando LoG
 - Encontrar pontos chaves
 - Descartar pontos chaves ruins
 - Definir orientação para os pontos chaves
 - Gerar as características SIFT
- Aplicar as características SIFT para o problema desejado.
- Algoritmo patenteado (SURF também é :P)

Implementação- Gerar Espaço de Escala

- Representação interna da imagem original para garantir invariância a escala. A partir da imagem original gera versões progressivamente desfocadas. Depois reduz o tamanho original pela metade e gera novas imagens desfocadas, e continua repetindo.
- O criador do SIFT sugere 4 oitavas e 5 níveis de desfoque como ideal para o algoritmo
- Primeiramente dobra-se o tamanho utilizando-se interpolação bilinear, depois aplica-se um filtro antialiasing(gaussiano), produzindo 4x mais pontos de interesse.
- Imagens normalizadas



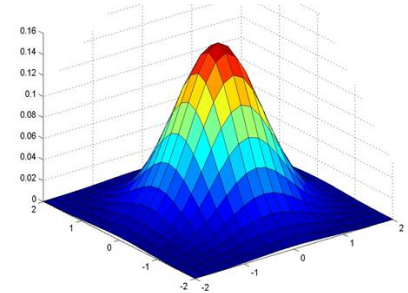
Implementação- Gerar Espaço de Escala

Blurring
Blurring

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y),$$

onde
onde

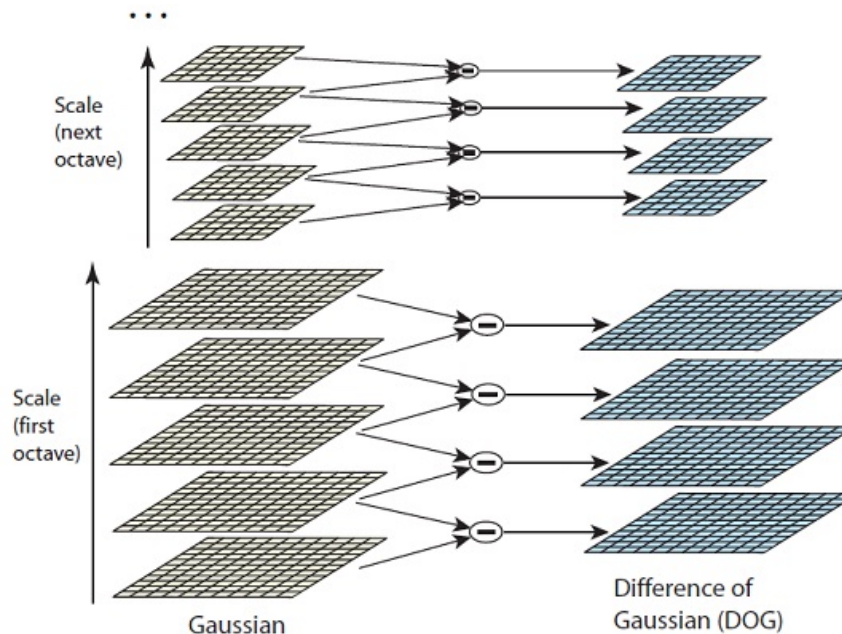
$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



	scale →				
octave	0.707107	1.000000	1.414214	2.000000	2.828427
	1.414214	2.000000	2.828427	4.000000	5.656854
	2.828427	4.000000	5.656854	8.000000	11.313708
	5.656854	8.000000	11.313708	16.000000	22.627417

Implementação- Aproximação por LoG (Laplacian of Gaussian)

- Usar imagens para gerar outro conjunto, as Diferenças de Gaussianos (DoG).
- Essas novas imagens serão usadas para encontrar os pontos de interesse.
- LoG : calcula a derivada de segunda ordem entre uma imagem e ela desfocada. Detecta bordas e pontos de interesse. Desfoque para dessensibilizar a ruídos.
- Operação “cara”. Para ganhar desempenho, utiliza o espaço de escala, calculando a DoG:



Implementação- Aproximação por LoG (Laplacian of Gaussian)

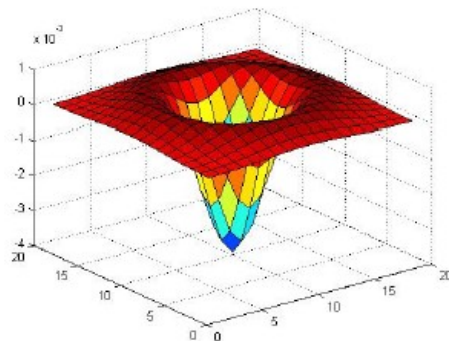
- DoG aproximadamente equivalente a LoG, porém substituindo uma operação cara, por uma subtração de imagens, barata.
- Invariante a escala

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

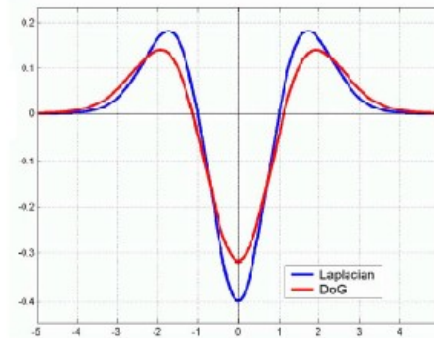
σ representa a escala. O Laplaciano invariante a escala seria representado por $\sigma^2 \nabla^2 G$, mas o DoG já está multiplicado por σ^2

$$G(x, y, k\sigma) - G(x, y, \sigma) \cong (k - 1)\sigma^2 \nabla^2 G$$

- DoG também está multiplicado por uma constant K entre as escalas.



(a)

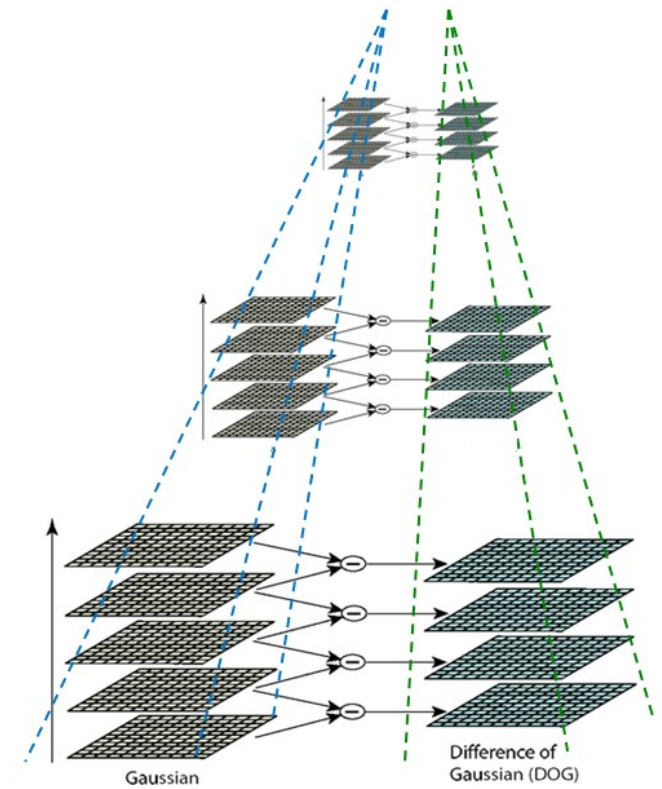
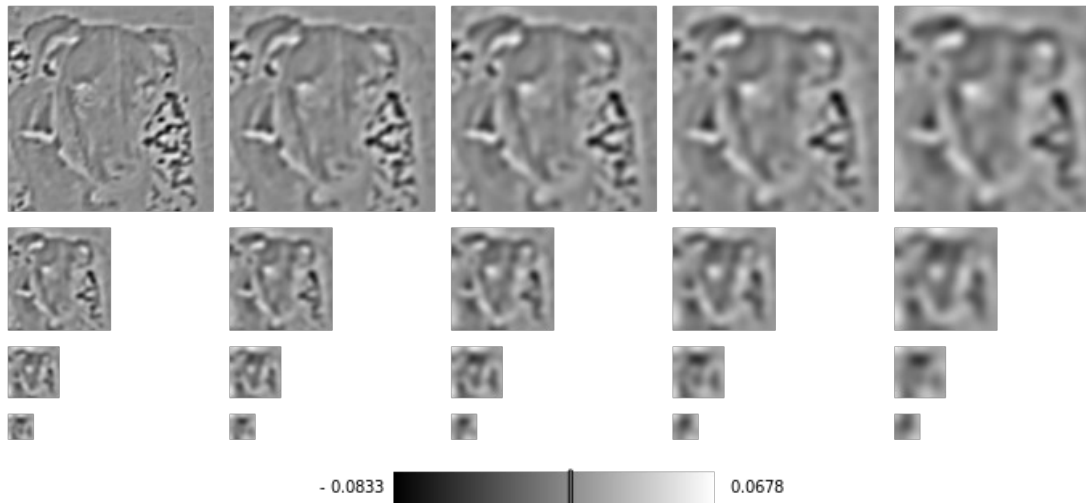


(b)

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) \otimes I(x, y) \\ = L(x, y, k\sigma) - L(x, y, \sigma)$$

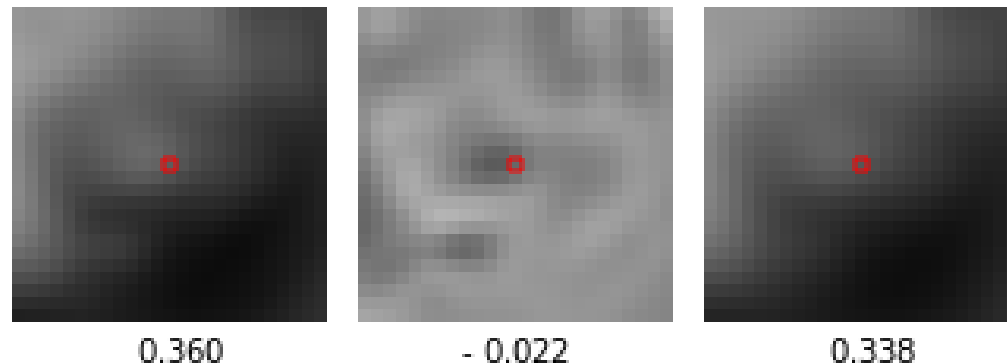
Implementação- Aproximação por LoG (Laplacian of Gaussian)

- Só se procura por localizações de máximos e mínimos nas imagens. Não será necessário checar os valores nessas localizações. A constante não altera as posições.



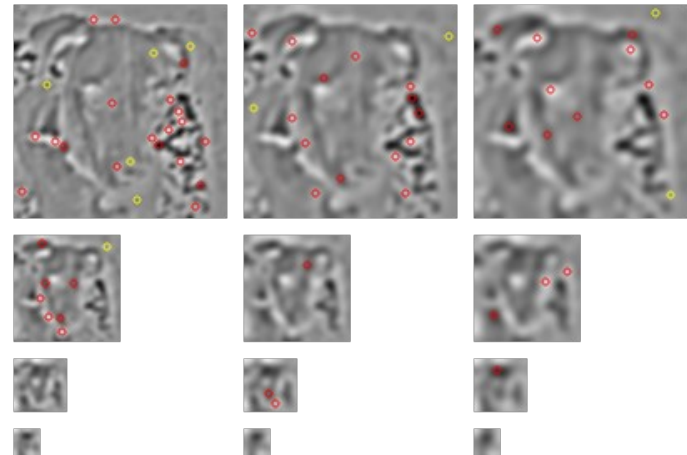
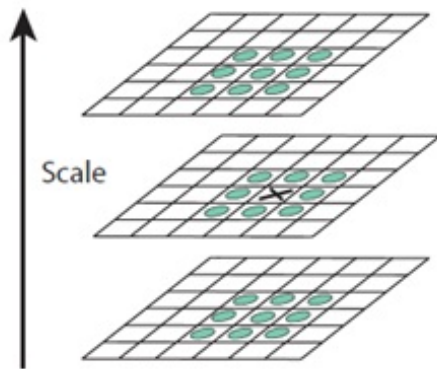
Implementação – Encontrar pontos chaves

- A imagem ao centro é a diferença entre duas imagens, enquanto que as outras são as duas imagens em escala que foram subtraídas.
- Pontos claros na imagem ao centro, representam um aumento no brilho, enquanto que pontos escuros, uma diminuição. Cinzas médios indicam que não houve mudança.



Implementação – Encontrar pontos chaves

- Localizar máximo/mínimo nas imagens DoG. Marcar caso seja o menor/maior dos 26 pontos ao redor. Interromper caso não seja...



- 0.0321	- 0.0367	- 0.0367
- 0.0411	- 0.0479	- 0.0491
- 0.0434	- 0.0478	- 0.0455

- 0.0354	- 0.0388	- 0.0376
- 0.0493	- 0.0526	- 0.0508
- 0.0502	- 0.0511	- 0.0472

- 0.0327	- 0.0338	- 0.0313
- 0.0457	- 0.0460	- 0.0426
- 0.0459	- 0.0443	- 0.0401

0.0078	0.0067	0.0066
0.0099	0.0103	0.0103
0.0064	0.0091	0.0105

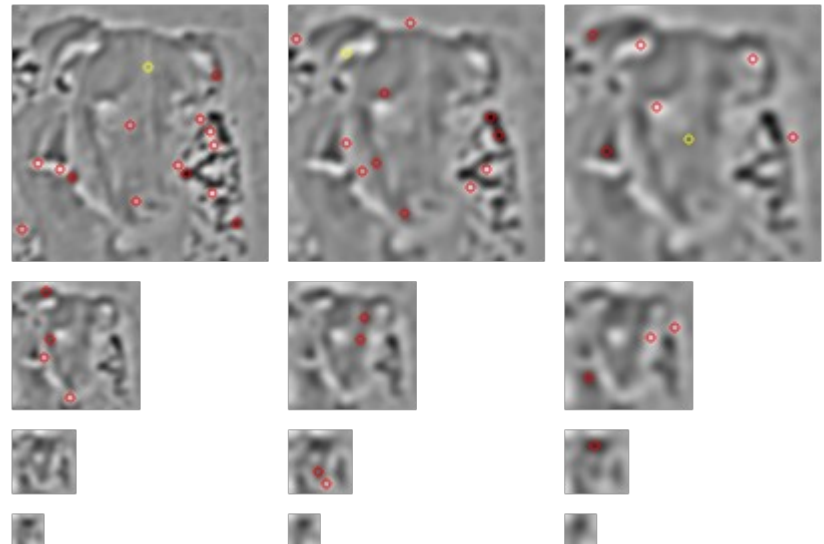
0.0086	0.0081	0.0080
0.0106	0.0108	0.0106
0.0088	0.0100	0.0104

0.0089	0.0090	0.0089
0.0103	0.0107	0.0103
0.0093	0.0101	0.0097

Implementação – Descartar pontos chaves ruins

- Encontrar o máximo e o mínimo dos subpixels
- Aproximar a expansão de Taylor do espaço de escala para definir as coordenadas que até então são discretas. É um processo iterativo e ou se consegue refinar a localização ou se desiste do ponto após algumas tentativas. Os extremos são procurados pela derivadas, igualando-se a zero.
- Descarta-se pontos de interesse próximos a extremos pois são sensíveis a ruídos. Para isso é utilizada a técnica de curvatura principal.
- Dos pontos remanescentes, descarta-se os de baixo contraste.

	discrete	interpolated
x	29.5	29.503
y	29.5	29.755
scale	1	0.639

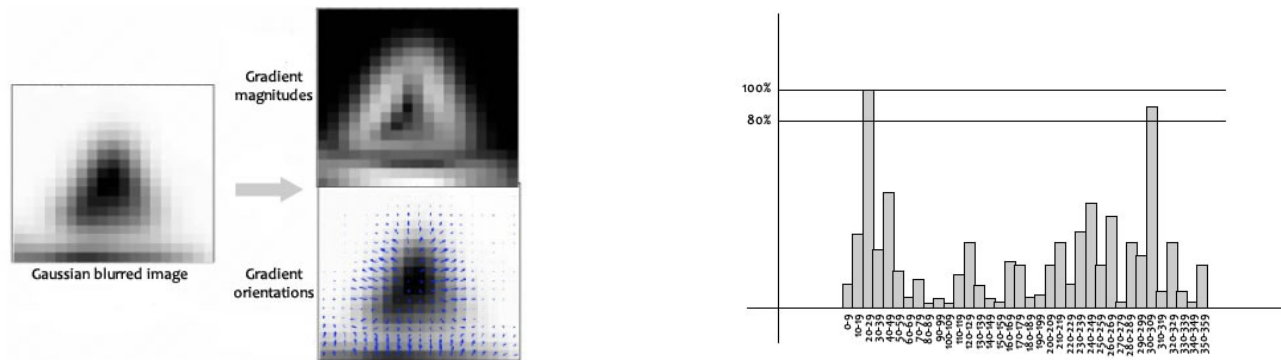


Implementação – Descartar pontos chaves ruins

- Encontrar o máximo e o mínimo dos subpixels
- Aproximar a expansão de Taylor do espaço de escala para definir as coordenadas que até então são discretas. É um processo iterativo e ou se consegue refinar a localização ou se desiste do ponto após algumas tentativas. Os extremos são procurados pela derivadas, igualando-se a zero.
- Descarta-se pontos de interesse próximos a extremos pois são sensíveis a ruídos. Para isso é utilizada a técnica de curvatura principal.
- Dos pontos remanescentes, descarta-se os de baixo contraste, dado pela magnitude do ponto.

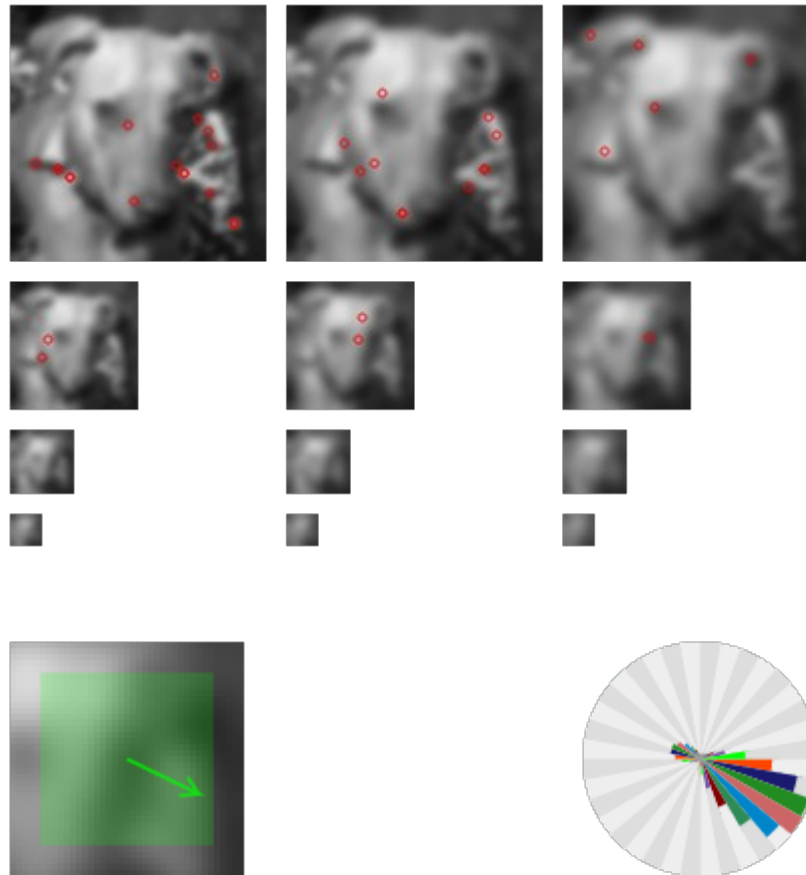
Implementação – Definir orientação para os pontos chaves

- Considerando uma vizinhança em torno do ponto de interesse, conforme a escala, o gradiente de magnitude e direção são calculados nesta região.
- Um histograma de orientação, com 36 bins, cobrindo 360 graus é criado, utilizando o valor proporcional da magnitude do ponto, depois de multiplicado pelo “peso Gaussiano”, para o seu preenchimento.



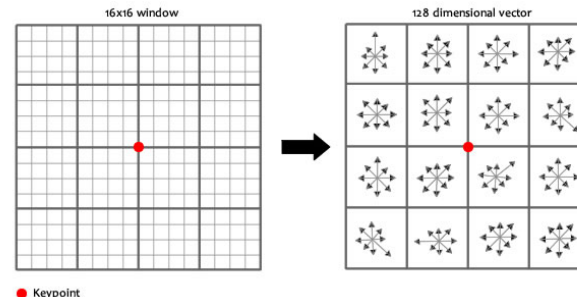
- Qualquer ponto 80% do pico mais alto da região é convertido em um novo ponto chave, que terá a mesma localização e escala do original, mas com nova orientação.
- Pontos sem uma vizinhança suficiente para computar uma orientação são descartados. Pontos chaves sem orientação dominante também são descartados.

Implementação – Definir orientação para os pontos chaves

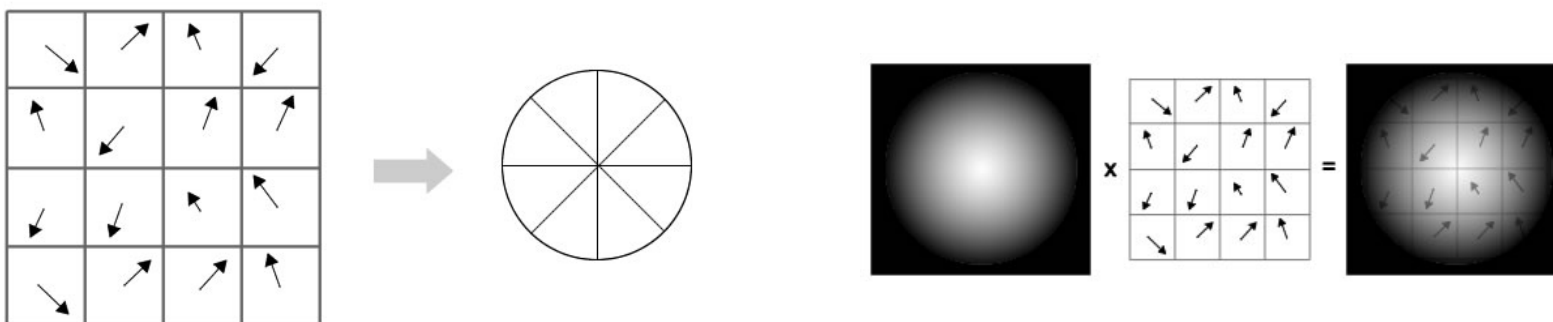


Implementação – Gerando as características SIFT

- Gera-se uma janela 16x16 em torno do ponto chave
- Essa janela é quebrada em 16 janelas 4x4



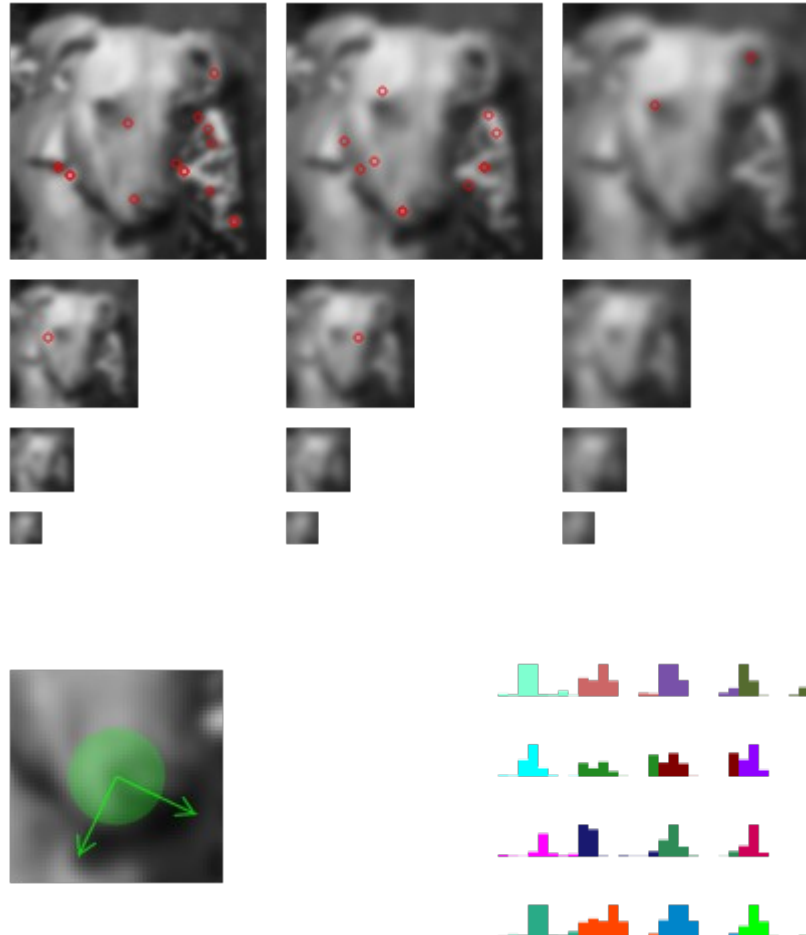
- Em cada janela, o gradiente de magnitude e orientação são calculados e colocados em um histograma de 8 bins, conforme uma função de peso gaussiana.



- Fazendo isso para todas as janelas cria-se 128 números, que são normalizados.
- Para ficar independente de rotação orientação é subtraído da orientação do ponto chave.
- Independência de iluminação é alcançada com aplicando-se um threshold de 0.2 aos números e normalizando-se.

Implementação – Gerando as características SIFT

- Gerando as características
 - Alguns pontos chaves podem ainda ser descartados caso se o filtro não couber na imagem.



Considerações

- Potencialmente define um conjunto grande de descritores, que suportam diferentes iluminações, perspectiva, tamanhos:
 - Pontos-chaves extraídos em diferentes escalas e níveis de desfoque, e todos os cálculos são feitos no espaço de escala. A ideia é fazer com que os descritores fiquem invariantes à escala e pequenas mudanças de perspectiva.
 - Cálculos relativos a uma orientação de referência para tornar os descritores robustos quanto à rotação.
 - A informação dos descritores é armazenada em relação à posição do ponto-chave, e assim é invariante à translação.
 - Muitos pontos-chaves são descartados se considerados instáveis, ou difíceis de localizar com precisão. Os que restarem são mais imunes a ruídos.
 - A normalização dos histogramas ao fim significa que os descritores não irão armazenar as magnitudes dos gradientes, mas somente a relação entre eles. Com isso ficam invariantes às mudanças globais de iluminação.
 - Os valores dos histogramas dos descritores também sofrem uma operação de threshold, diminuindo a influência dos gradientes maiores. Assim diminuem a influência de mudanças locais, não uniformes, de iluminação.

Referências

- <http://weitz.de/sift/>
- <http://aishack.in/tutorials/sift-scale-invariant-feature-transform-introduction/>
- http://cgit.nutn.edu.tw:8080/cgit/PPTDL/WWD_091221064614.PDF
- <http://prog3.com/sbdm/blog/u013467442/article/details/35566981>