

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE CIÊNCIAS DA TERRA
Departamento de Geomática

Disciplina: PROCESSAMENTO DIGITAL DE IMAGENS II
Código: GA144

CH Total:45 h

CH Semanal 03 h

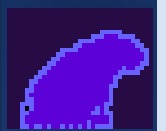
Localização de bordas e pontos

PDI-2
0100
1100
1010
1100
0000
1000

0100
1100
1010
1100
0000
1000

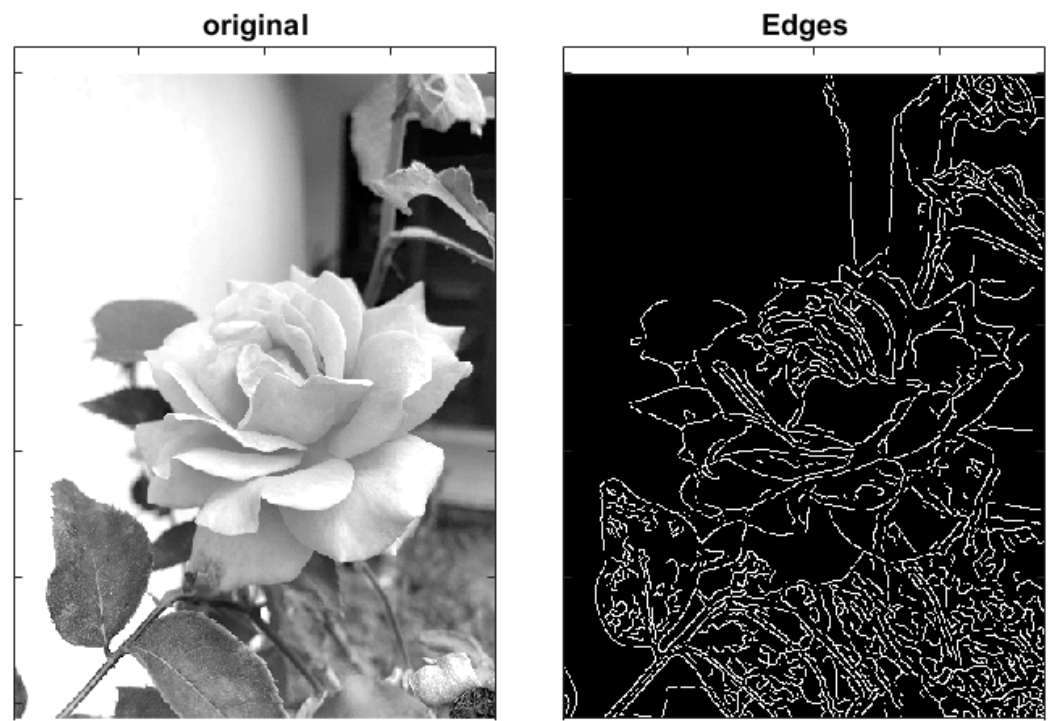


01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

Detectar as bordas de objetos a partir da variação local dos valores digitais da imagem.
Usando o gradiente



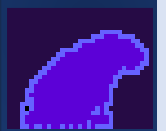
PDI-2
0100
1100
1010
1100
0000
1000

Detector de Canny

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

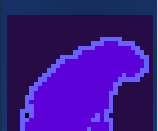
Detección de bordas de Canny

Etapas

- Redução de ruído;
- Cálculo de gradiente;
- Supressão de “não-máximos”
- Dupla limiarização;
- Detecção de bordas por histerese.



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

Redução de ruído

- Com a finalidade de eliminar pequenas perturbações (incluindo ruído) que podem produzir falsas bordas.

Cálculo de gradiente

- O Gradiente descreve a intensidade das bordas e sua direção

Supressão de “não-máximos”

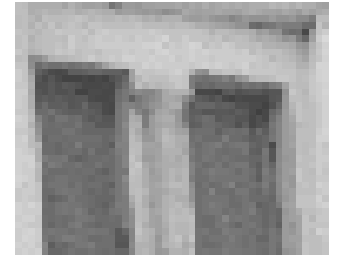
- Trata-se de “AFINAR” as linhas na imagem de gradiente para localizar a verdadeira fronteira

Dupla limiarização

- Busca-se identificar bordas bem definidas e também prováveis bordas menos nítidas.

Detecção de bordas por histerese.

- Trata-se de delinear a borda com base nos elementos detectados

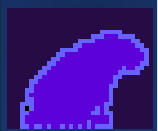


PDI-2
0100
1100
1010
1100
0000
1000

0100
1100
1010
1100
0000
1000



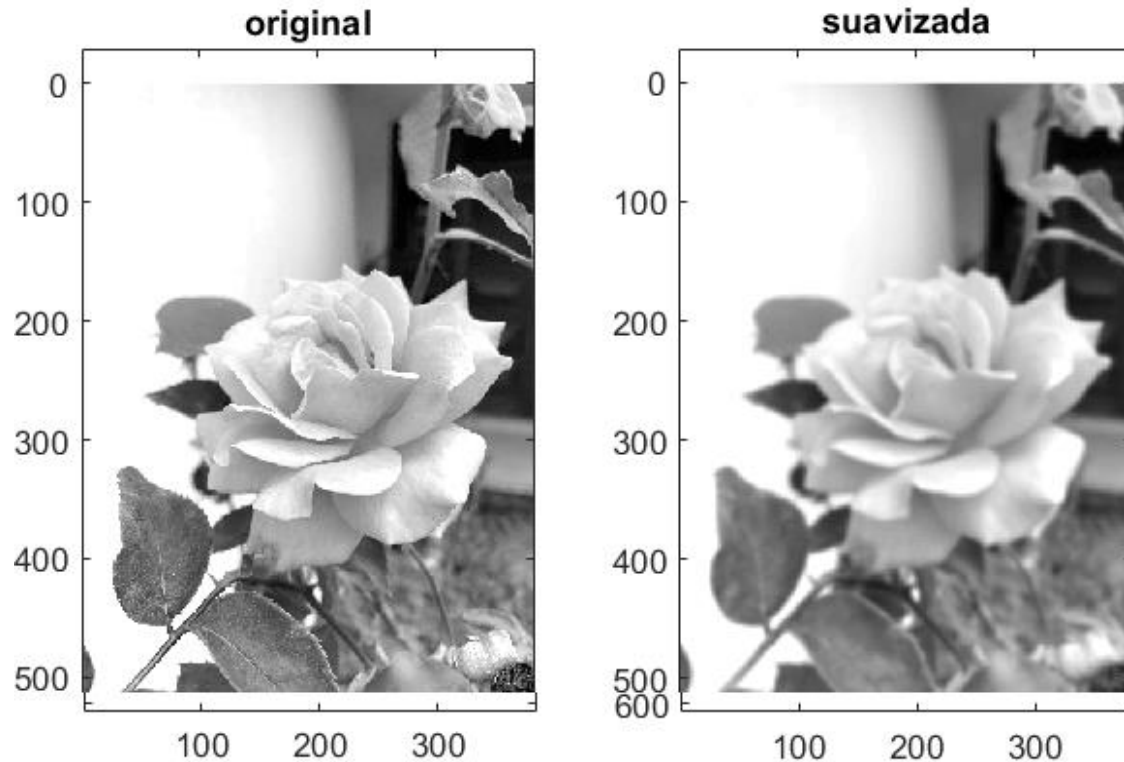
01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

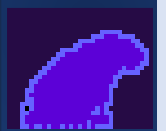
Redução de ruído:

- Como o cálculo do gradiente é altamente sensível à presença de ruído, em uma primeira fase a imagem é suavizada, usando um filtro Gaussiano.





01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

- Cálculo de gradiente:
- Aplica-se o cálculo do Gradiente por Sobel, nas duas direções e se armazenam os valores da intensidade do Gradiente e a sua direção.

$$K_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, K_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

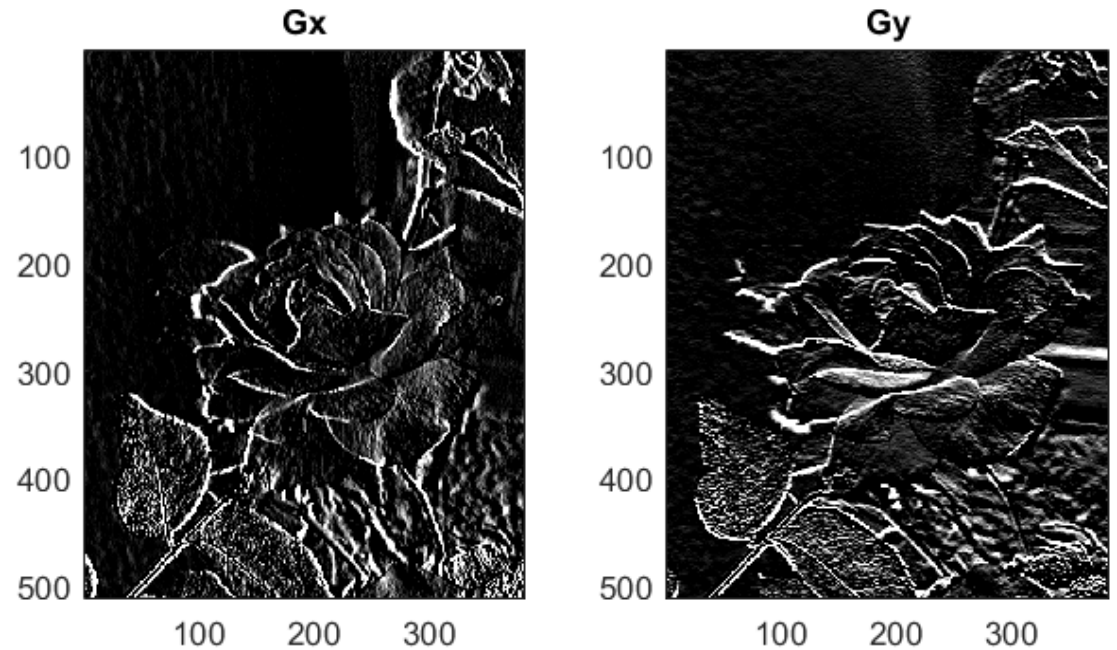
Gradiente em Linhas Gy
Gradiente em Colunas Gx

Intensidade do gradiente:

$$I(x, y) = \sqrt{G_x^2 + G_y^2}$$

Direção do gradiente

$$d(x, y) = \arctan\left(\frac{G_y}{G_x}\right)$$



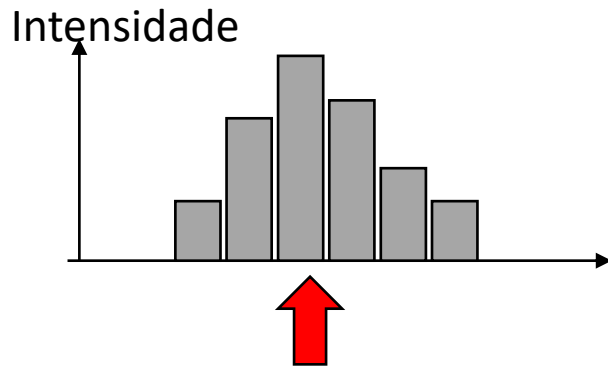
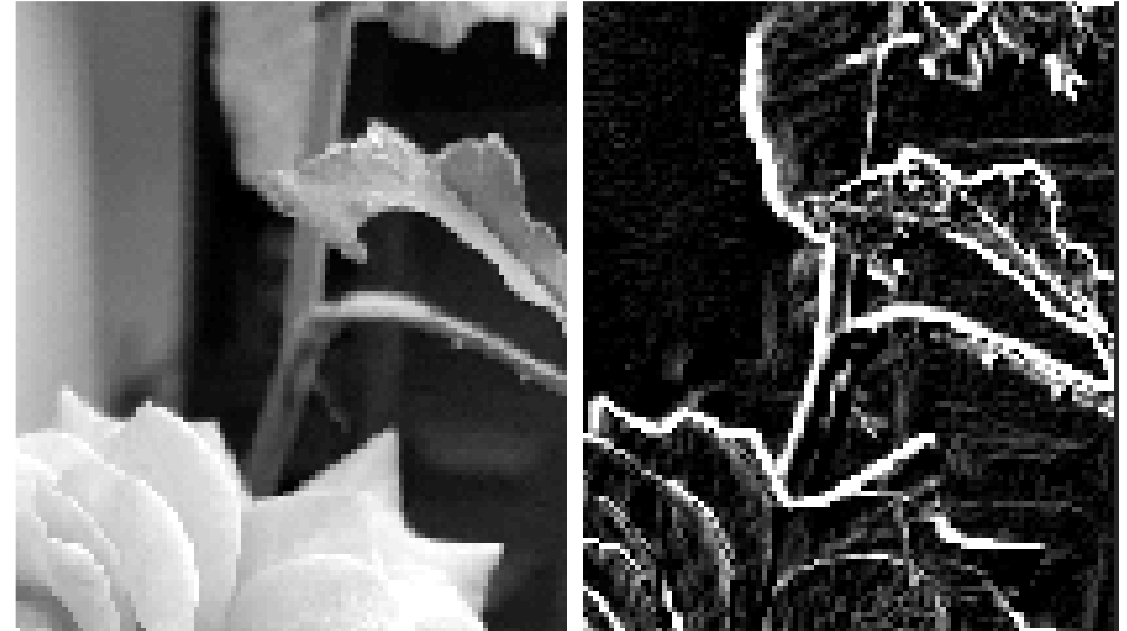
PDI-2
0100
1100
1010
1100
0000
1000

Supressão de “não- máximos”:

0100
1100
1010
1100
0000
1000

O filtro de Sobel origina bordas grossas, devido ao processo de convolução.

O ideal seriam bordas finas. Por isso, nesta etapa se busca detectar os máximos locais da Intensidade do Gradiente e apagar os “não máximos”.



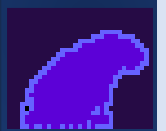
01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



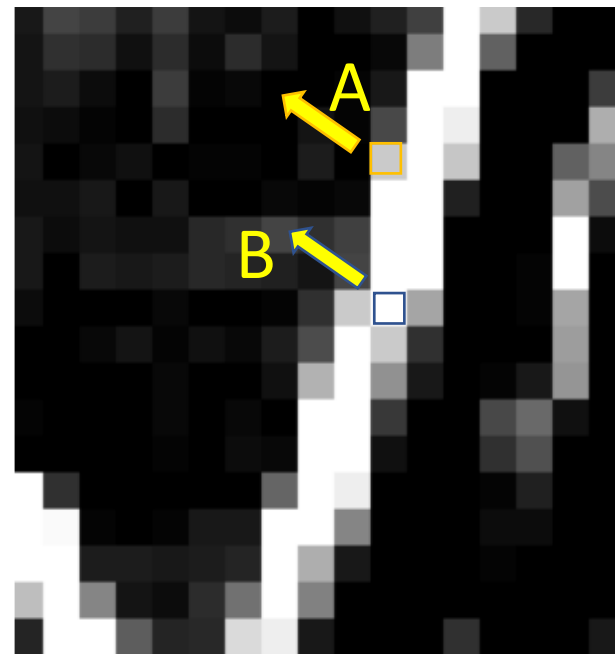
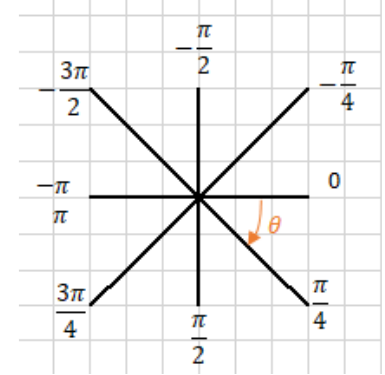
100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

Determina-se a direção do gradiente, o que define uma direção de variação.

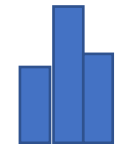
Considera-se apenas 8 direções.

Se analisa a variação do Gradiente ao longo desta direção em torno do pixel de fronteira.

Se o pixel for maior que os seus dois vizinhos ao longo desta direção, ele é preservado, caso contrário, apagado, independentemente do valor da Intensidade.



A



B

PDI-2
0100
1100
1010
1100
0000
1000

Dupla limiarização

0100
1100
1010
1100
0000
1000

Identifica-se as bordas mais intensas, aquelas bem definidas. Mas, para não perder a continuidade de uma fronteira, bordas menos intensas também são detectadas.

Para isto, aplica-se dois limiares para detectar

- a) Bordas intensas ($\text{Grad} > \text{LIM1}$)
- b) Prováveis Bordas ($\text{Grad} < \text{LIM1}$ e $\text{Grad} > \text{Lim2}$)

A ideia é identificar as bordas “fortes” mas não descartar prováveis bordas que podem ser “recuperadas” em uma fase posterior.

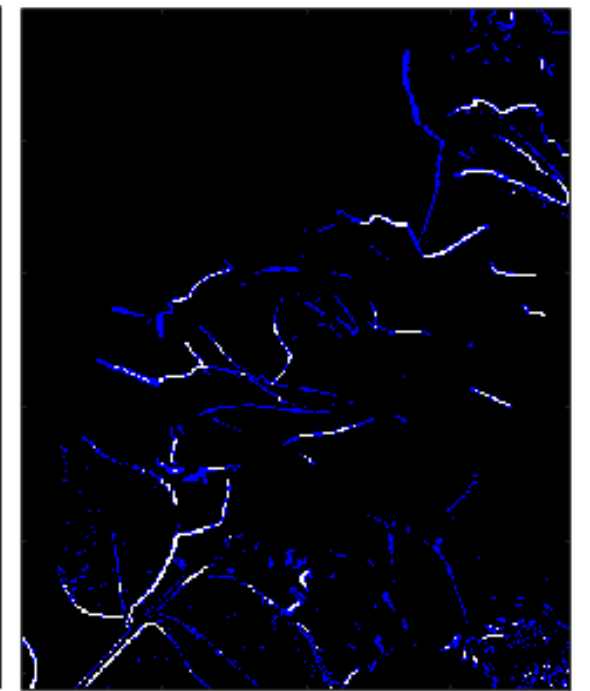
Ao lado:

- Bordas fortes em branco
- Menos intensas em azul.

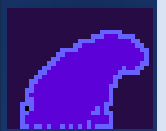
Intensidade



0,7 e 0,3



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

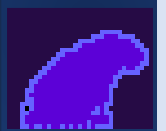
PDI-2
0100
1100
1010
1100
0000
1000

Detecção de bordas por histerese

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000

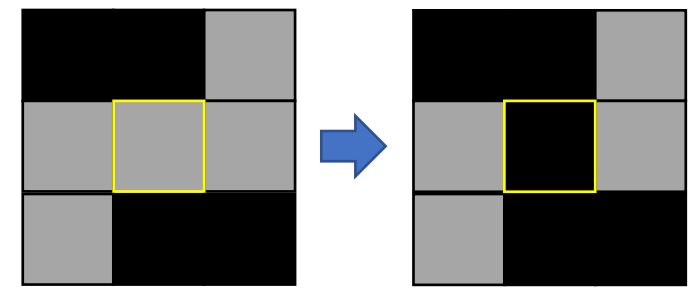


100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

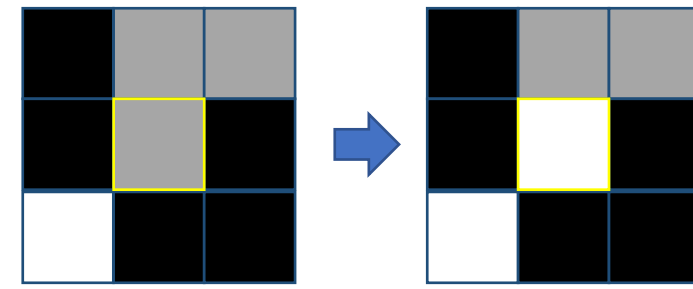
Em uma etapa final, a borda é reconstruída, somando bordas “fracas” ao conjunto de “bordas fortes” caso elas fiquem próximas.

Verifica-se se existem pixels de “bordas fracas” na vizinhança de “bordas fortes”, o que indicaria que eles também são parte de uma fronteira.

- Fundo
- Borda fraca
- Borda forte



descartar



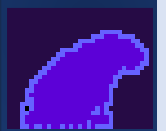
Preservar: Vizinho “forte” presente na vizinhança

PDI-2 0100
1100
1010
1100
0000
1000

0100
1100
1010
1100
0000
1000

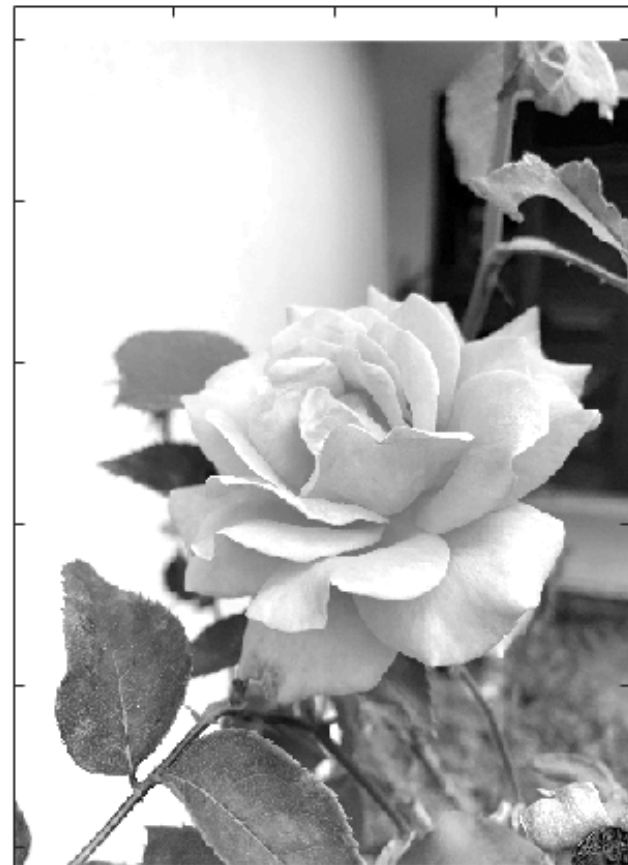


01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

original



Edges

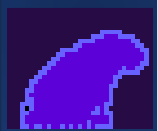


<https://justin-liang.com/tutorials/canny/>
Em Python:
<https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123>

Localização de esquinas



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

Detector de Harris

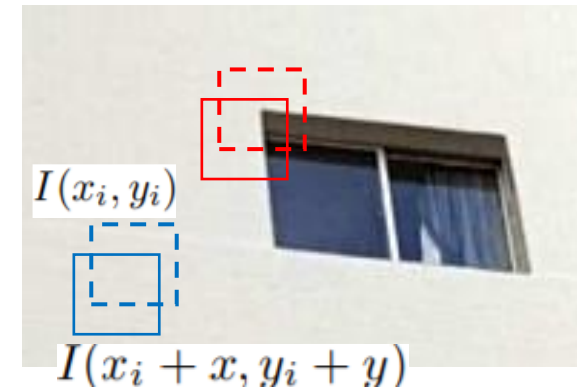
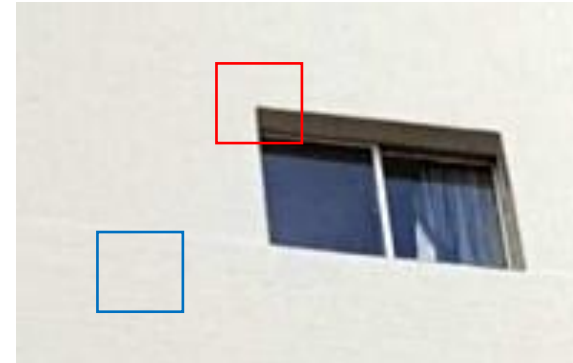
Para detectar uma esquina, Harris propõe analisar a auto-similaridade entre uma região e as regiões em volta.

Por exemplo, analisando a diferença entre a região em questão e regiões próximas, resultantes de um pequeno deslocamento da região (dx,dy).

Comparar: $I(x_i, y_j)$ e $I(x_{i+dx}, y_{j+dy})$

Se a diferença é pequena, trata-se de uma região uniforme.

$$Dif^2 = (I(x_i, y_j) - I(x_{i+dx}, y_{j+dy}))^2$$

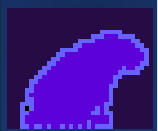


PDI-2
0100
1100
1010
1100
0000
1000

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

Por isso, Harris propõe calcular a soma das diferenças ao quadrado, ponderando cada elemento (w= ponderador)

$$S_w(x, y) = \sum_{x_i y_i} w(x_i, y_i) [I(x_i + x, y_i + y) - I(x_i, y_i)]^2$$

A diferença pode ser aproximada pela série de Taylor como:

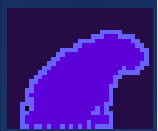
$$\begin{aligned} I(x_i + x, y_i + y) &= I(x_i, y_i) + \frac{\partial I}{\partial x_i} x + \frac{\partial I}{\partial y_i} y \\ &= I(x_i, y_i) + [G_x \quad G_y] \begin{bmatrix} x \\ y \end{bmatrix} \end{aligned}$$

onde G_x e G_y são as componentes do Gradiente nas direções x e y . Para substituir em "S", devemos elevar ao quadrado.

$$S_w(x, y) = [x \quad y] H \begin{bmatrix} x \\ y \end{bmatrix}$$



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

A expressão quadrática pode ser escrita como:

$$S_W(x, y) = [xy]H \begin{bmatrix} x \\ y \end{bmatrix}$$

A matriz H é quadrada 2x2 e simétrica.

$$H(x, y) = \begin{bmatrix} A & B \\ B & C \end{bmatrix}$$

Seus elementos são calculados a partir das componentes do Gradiente nas direções x e y.

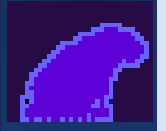
$$A = \sum_{x_i y_i} w(x_i, y_i) I_x^2(x_i, y_i)$$
$$B = \sum_{x_i y_i} w(x_i, y_i) I_{xy}(x_i, y_i)$$
$$C = \sum_{x_i y_i} w(x_i, y_i) I_y^2(x_i, y_i)$$

PDI-2
0100
1100
1010
1100
0000
1000

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

A intenção é determinar os locais onde a diferença é máxima. Para isto se calcula um “score” da diferença, como:

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

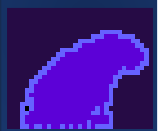
E as componentes desta equação se calculam a partir dos autovalores da matriz H.

$$\det(H) = \lambda_1 \lambda_2$$

$$\text{trace}(H) = \lambda_1 + \lambda_2$$

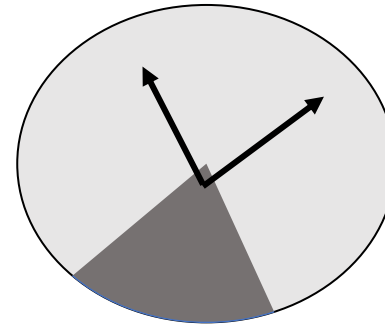
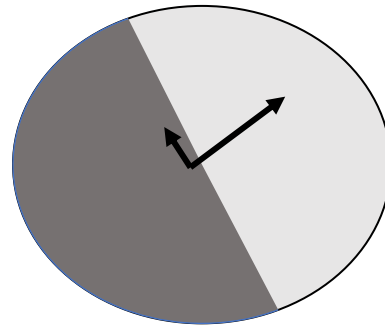
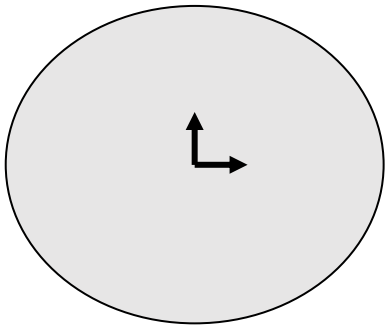


01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

- Se os dois autovalores são pequenos, R é pequeno, logo não se trata de uma borda e sim de uma região uniforme.
- Se o primeiro autovalor for muito grande ($\lambda_1 \gg \lambda_2$) caracteriza-se a existencia de uma direção preponderante, logo trata-se de uma borda (linha)
- Se λ_1 e λ_2 são grandes e $\lambda_1 \sim \lambda_2$, R também é grande, e isto ocorre em quinas.



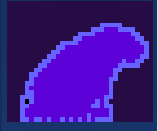
PDI-2
0100
1100
1010
1100
0000
1000

RESUMO

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

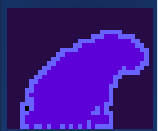
1. Aplicar um filtro de suavização (Gaussiano) para reduzir ruído.
2. Aplicar o operador de Sobel para estimar o gradiente em cada pixel nas direções x e y
3. Com esses valores, calcular o valor do índice "R" para cada pixel da imagem.
4. Aplicar um limiar para detectar os maiores valores que sejam máximos locais.

PDI-2 0100
1100
1010
1100
0000
1000

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

SIFT

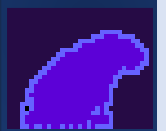
PDI-2
0100
1100
1010
1100
0000
1000

Introdução a SIFT

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

É possível detectar feições (ex: quinas) e uma imagem e identificar a mesma feição em outra.

Mesmo tendo variações de contraste, rotação e escala?



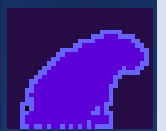
PDI-2
0100
1100
1010
1100
0000
1000

Problema

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

Uso:

Dado um par de imagens (fotografias), detectar pares de pontos homólogos de forma automática.

Porém, podem existir diferenças de contraste, rotação, translação e escala.

Com estes pontos, e usando técnicas de fotogrametria, é possível orientar as fotos e depois calcular coordenadas 3D de outros pontos na imagem.

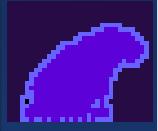


PDI-2
0100
1100
1010
1100
0000
1000

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

SIFT Scale-Invariant Feature Transform

Os conceitos são descritos em:

Distinctive Image Features from Scale-Invariant Keypoints

David G. Lowe

Computer Science Department

University of British Columbia Vancouver, B.C., Canada

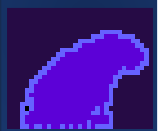
January 5, 2004

Tem uma descrição em:

https://docs.opencv.org/3.2.0/da/df5/tutorial_py_sift_intro.html



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

Detectores de quinas (como Harris), são invariantes à rotação, mas não em relação à escala.

Para contornar este problema Lowe (2004), criou o algoritmo, Scale Invariant Feature Transform (SIFT)

O algoritmo trata de

- a) Detectar pontos significativos (quinas)
- b) Calcular descritores de contraste para caracterizar cada ponto
- c) Encontrar pares de pontos baseado nos descritores.

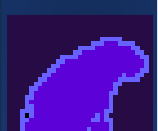
PDI-2
0100
1100
1010
1100
0000
1000

Etapas

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

1. Detectar bordas (aplicar detector de bordas)
2. Selecionar bordas fortes
3. Buscar quinas, não bordas
4. Descrever essas quinas em função de sua vizinhança.
5. Classificar pares de pontos por semelhança.

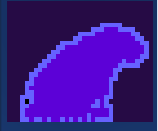
PDI-2
0100
1100
1010
1100
0000
1000

Scale-space Extrema Detection

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



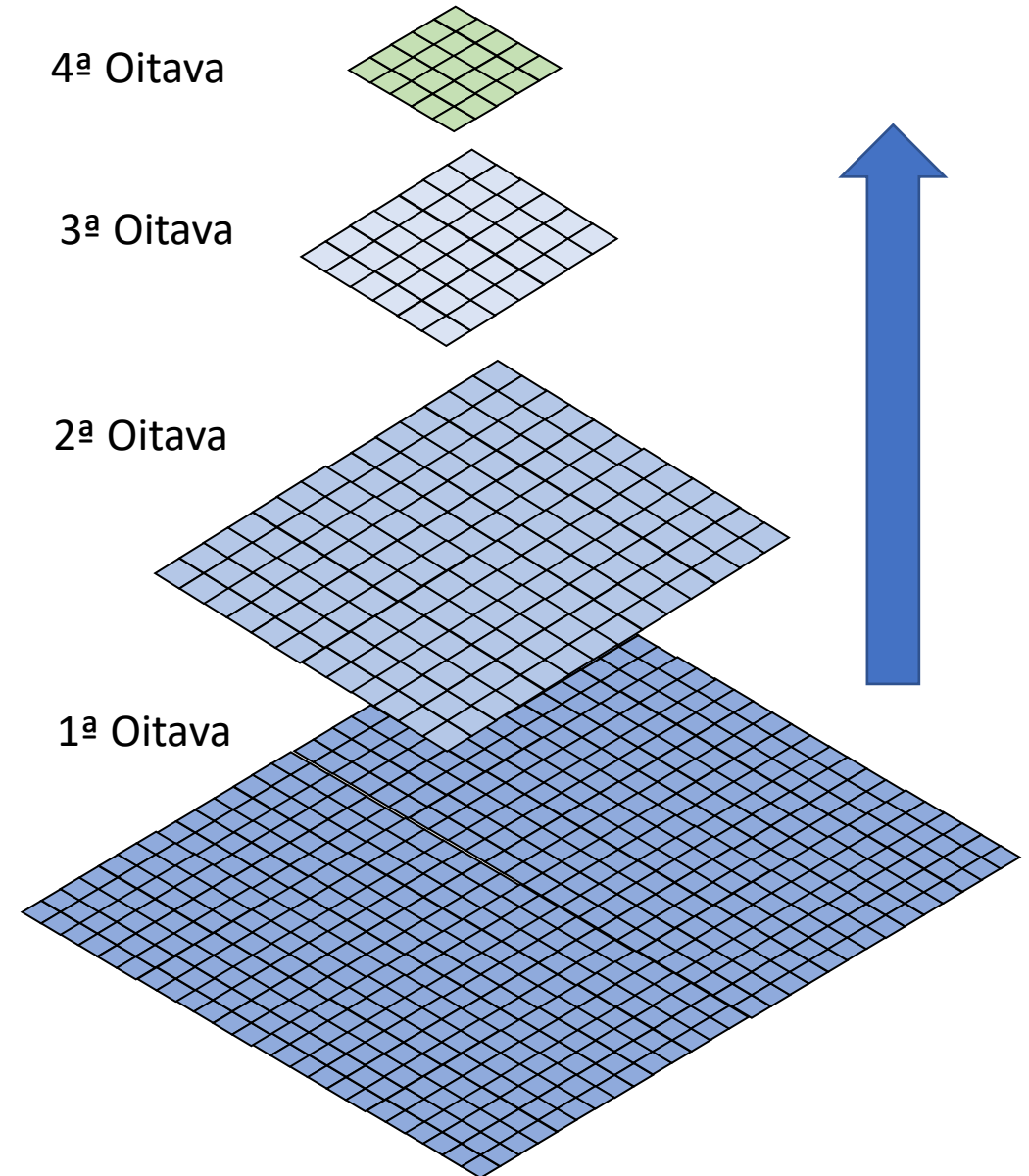
100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

Lowe propõe buscar quinas aplicando a filtragem variando a resolução da imagem

Para lidar com eventual diferença de escala entre imagens, deve-se processar as imagens em diferentes níveis de representação geométrica.

A partir da imagem original podem ser criadas novas imagens menores por reamostragem (resampling)

Lowe recomenda usar 4 representações (oitavas)



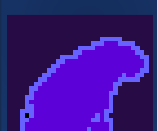
PDI-2
0100
1100
1010
1100
0000
1000

High-pass LoG

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

Deseja-se detectar pontos notáveis. Isto pode ser obtido com um filtro passa altas (Hgh Pass) como o filtro LaPlaciano.

Por questão de eficiência computacional, o filtro Laplaciano do Gaussiano (LoG) é usado.

O LoG consiste em calcular a diferença entre a imagem original e a filtrada para se obter uma aproximação do LaPlaciano.

LoG atua como um detector de detalhes.

Como podem existir detalhes de diferentes tamanhos, Lowe propõe calcular vários LoG a partir da mesma imagem, repetindo a filtragem várias vezes.

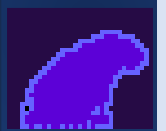
PDI-2
0100
1100
1010
1100
0000
1000

Suavização

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

- Para suavizar a imagem com diferentes intensidades, cada imagem (resampled) é filtrada consecutivamente usando um filtro passa-baixas. Neste caso, usa-se o filtro Gaussiano.
- A filtragem é repetida várias vezes (quatro vezes)

• Original, filtrada, filtrada, filtrada, filtrada,

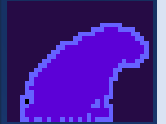


PDI-2
0100
1100
1010
1100
0000
1000

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

Filtragem Gaussiana



1ª Oitava



2ª Oitava



3ª Oitava



4ª Oitava



resampling



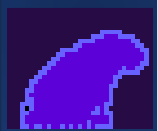
PDI-2
0100
1100
1010
1100
0000
1000

High-pass LoG

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000

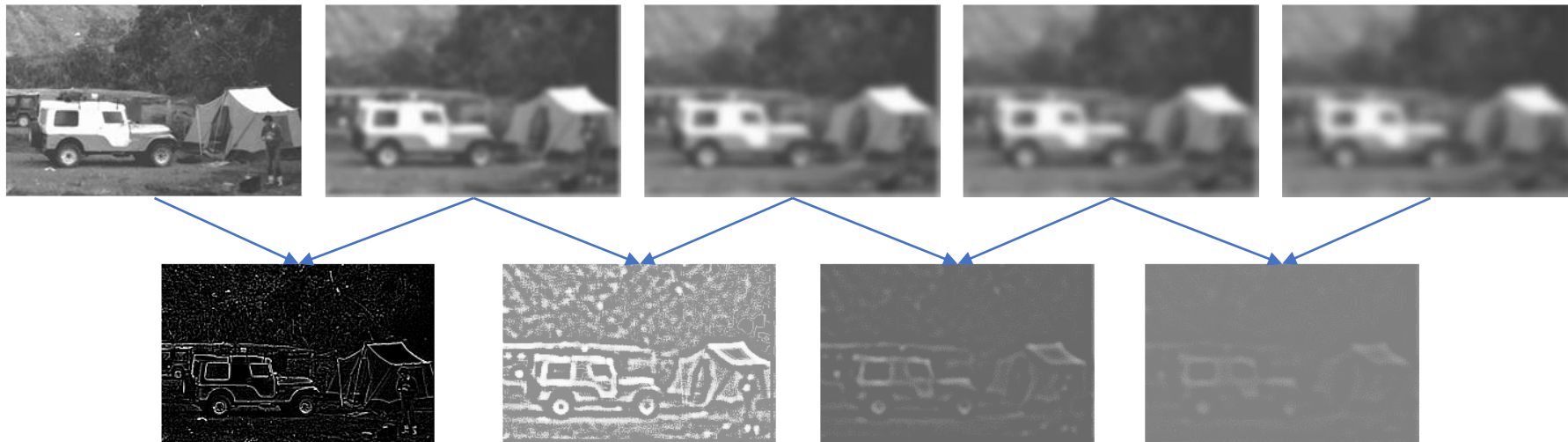


100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

Em cada nível de reamostragem, é aplicado um filtro passa-altas. Por questão de eficiência computacional, o filtro Laplaciano do Gaussiano (LoG)

a) Calcula-se a diferença entre a imagem original e a filtrada para se obter uma aproximação do Laplaciano.

LoG atua como um detector de detalhes.



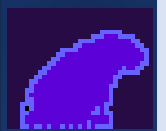
PDI-2
0100
1100
1010
1100
0000
1000

Detecção de pontos significativos

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

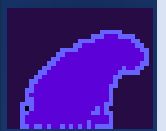
- Localização de Keypoint
- São considerados pontos notáveis os máximos e mínimos das imagens LaPlacianas.

A escolha dos pontos significativos, que podem ser usados como potencial ponto homólogo (keypoint) segue dois passos:

- a) Encontrar os máximos e mínimos locais (detecção)
- b) Remover pontos de baixo contraste (seleção)



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000

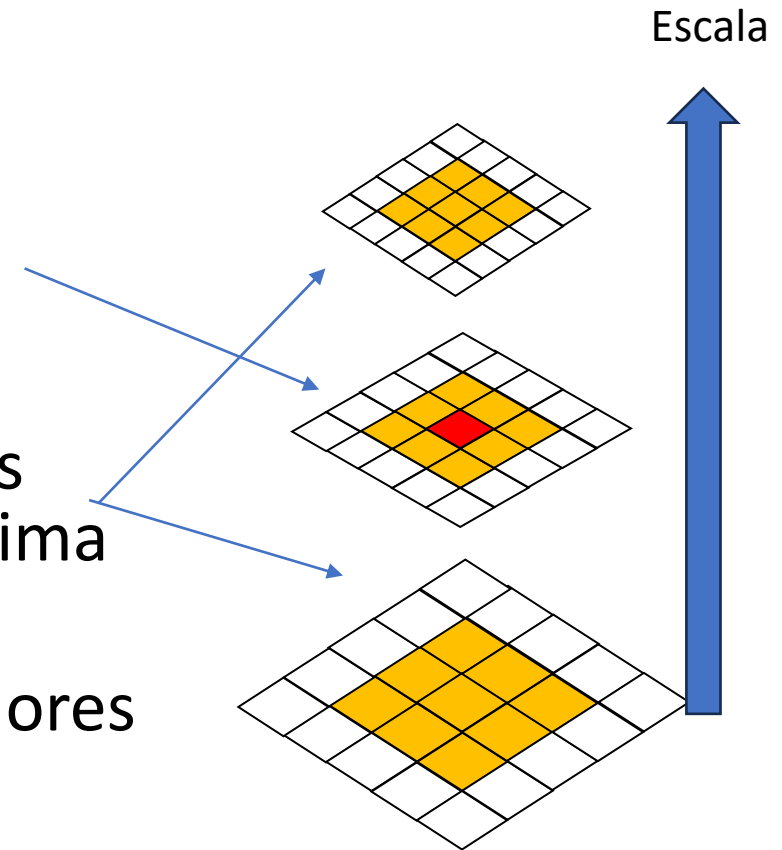


100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

Máximos e mínimos locais

Um pixel é considerado máximo se:

- Seu valor é maior que os valores dos 8 pixels vizinhos
- E também maior que os valores dos pixels na vizinhança imediata em outras resoluções. Os nove pixels da oitava anterior e nove da próxima oitava
- Então o pixel é comparado com outros 26 valores para ser considerado máximo/mínimo local.
- Para isto, se projeta a localização do pixel nas outras oitavas e se define a vizinhança 3x3 em torno desse local.



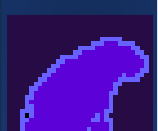
PDI-2
0100
1100
1010
1100
0000
1000

High-pass LoG

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

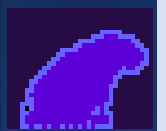
Aplicando-se este princípio a diferentes resoluções, é obtida uma lista de pontos com alto contraste (detalhes) na imagem. Cada ponto é um potencial ponto de interesse (keypoint).

A posição dos pontos detectados é refinada para obter resultados mais acurados

Para isto usa-se uma série de Taylor para interpolar uma superfície com os valores da vizinhança e assim calcular a posição exata do máximo.



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000

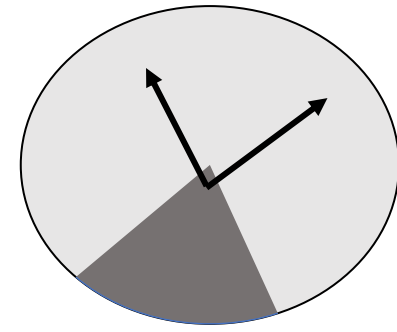
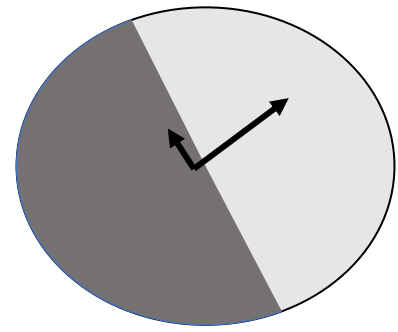


100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

Redução de pontos “fracos” - Se a intensidade do pico ficar abaixo de um limiar, o ponto é descartado, por ser considerado “fraco”

Eliminação de bordas - O DoG detecta pontos e bordas. As bordas precisam ser removidas, pois não fornecem localização precisa. Para isso, usa-se o princípio do detector de Harris, analisando os autovalores. Como nas bordas o primeiro autovalor é muito maior que o segundo, então as bordas podem ser eliminadas.

Assim, elimina-se pontos-chave de baixo contraste e localizados em bordas, restando pontos de interesse fortes.

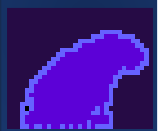


PDI-2
0100
1100
1010
1100
0000
1000

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

- Até aqui, foram localizados os pontos significativos (key points) nas duas imagens.
- Agora devemos procurar os pares homólogos. Para isto:
 1. Calcula-se a orientação da região
 2. Calcula-se descritores espaciais do ponto
 3. Classifica-se pares por semelhança
- ...

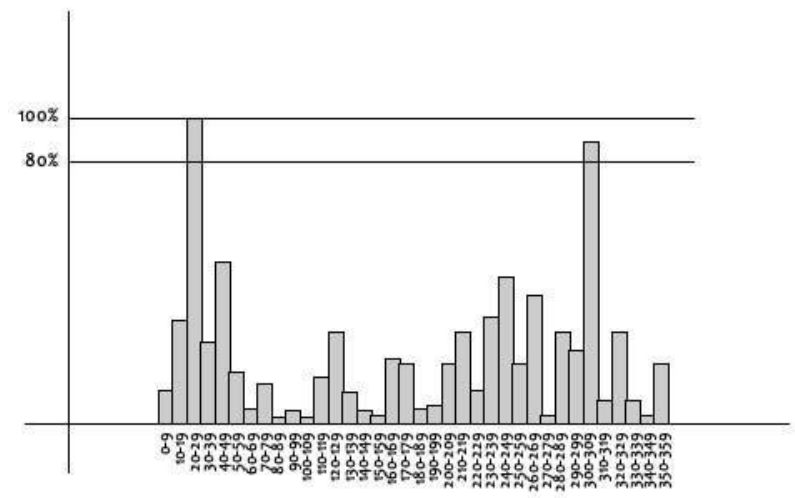
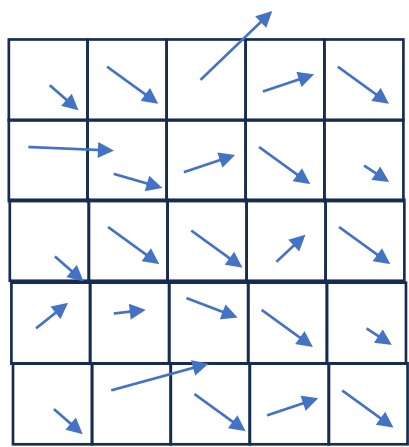
Atribuição de Orientação

Uma vez detectado o keypoint, analisa-se a variação do Gradiente nas suas imediações para estimar a orientação predominante na região.

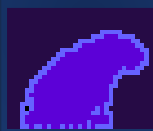
Uma vizinhança em torno do ponto é analisada. Para cada pixel, a intensidade e direção do gradiente local são calculados. Para simplificar, apenas 36 direções são consideradas (*não 360 graus).

Com estes valores, um histograma de intensidades do gradiente por direção é construído. Neste gráfico, os picos mostram a direção predominante (80%).

Se existem vários picos, keypoints são adicionados.



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

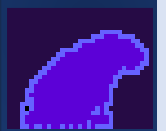
PDI-2
0100
1100
1010
1100
0000
1000

Descritores

0100
1100
1010
1100
0000
1000



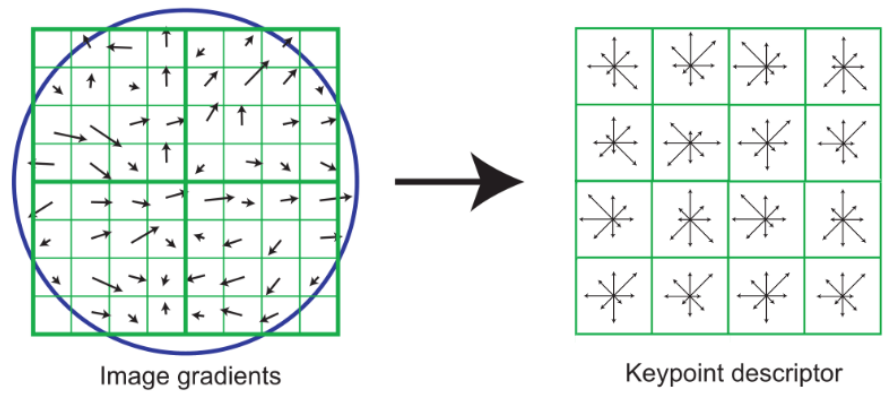
01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

Uma vez encontrada a direção predominante, a região pode ser rotacionada em relação a esta direção, suprimindo assim o efeito da rotação.

Os vetores do gradiente são então rotacionados em relação à direção predominante. A seguir, as direções são simplificadas em 8 valores e todos estes valores armazenados no vetor de características.



$$a = \begin{bmatrix} a1 \\ a2 \\ a3 \\ a4 \\ a5 \\ a6 \\ a7 \\ a8 \end{bmatrix}$$

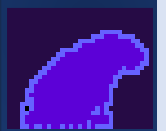
Histograma de 8 direções de uma região

$$V = \begin{bmatrix} a1 \\ a2 \\ \dots \\ b1 \\ b2 \\ \dots \\ c1 \\ \dots \\ c16 \end{bmatrix}$$

Vários histogramas compõem um vetor descriptor

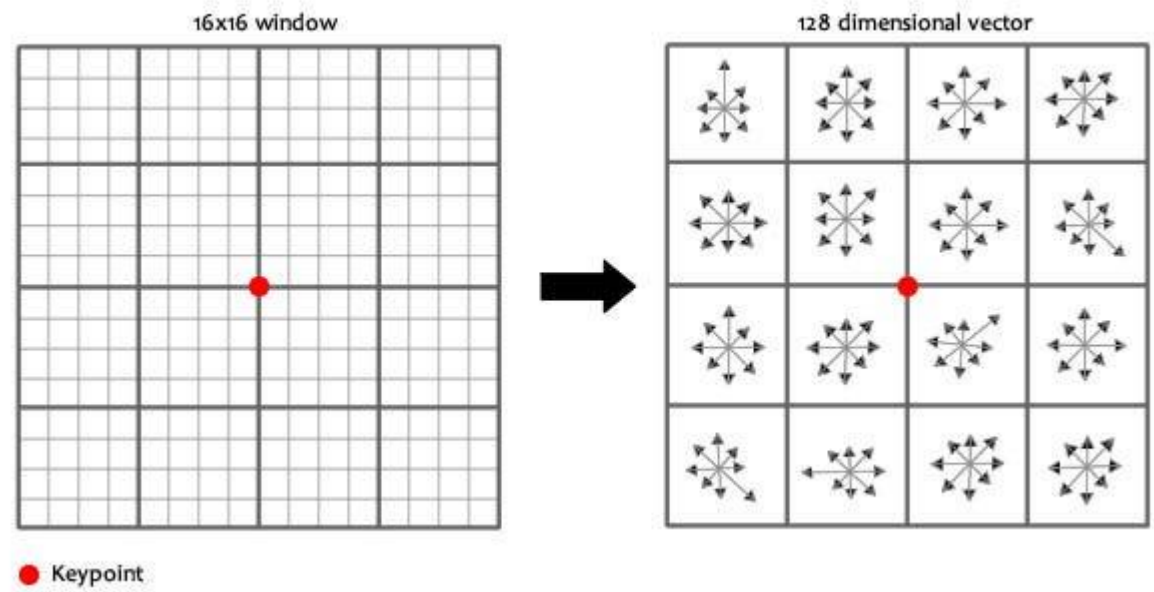


01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

- Para isto, a vizinhança 16x16 em torno do pixel é analisada. 4 blocos 4x4 em torno da posição do keypoint.
- As direções são codificadas usando apenas 8 direções.
- Para cada quadrado 4x4, um histograma de direções é calculado.



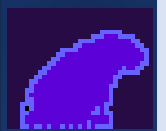
PDI-2
0100
1100
1010
1100
0000
1000

Matching/Correspondência

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000

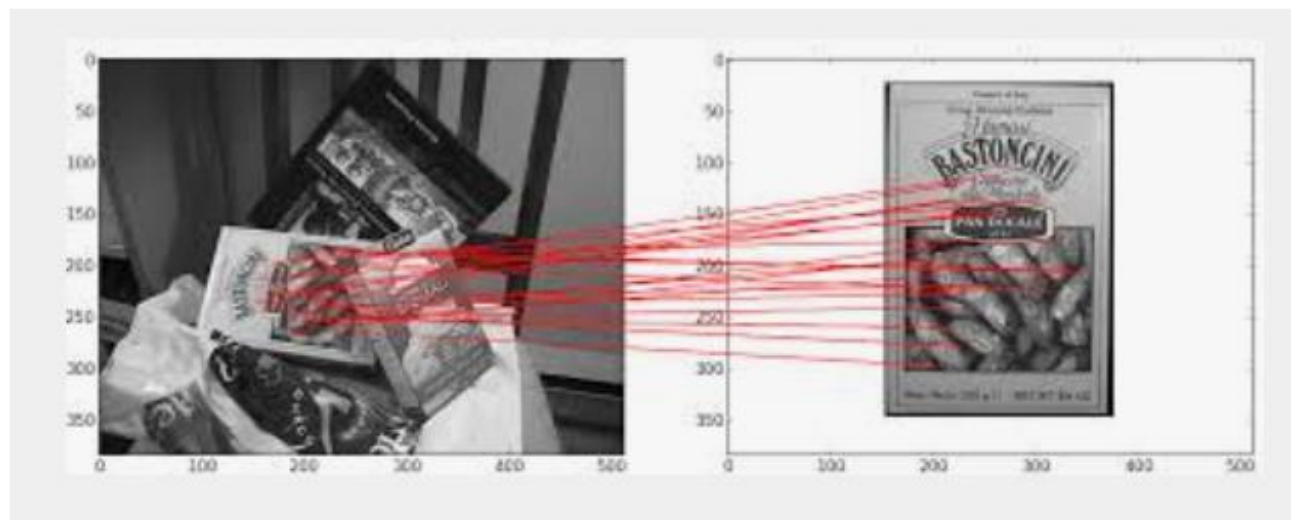


100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

Como cada ponto possui um vetor descritor que não sofre o efeito da escala nem da rotação, então, o seguinte passo é identificar na outra imagem um ponto com o vetor descritor mais parecido.

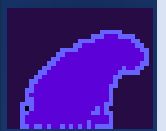
Isto é feito classificando os pares pelo critério de distância mínima Euclidiana.

Em algumas situações, podem ocorrer dois potenciais candidatos com distâncias muito parecidas. Para identificar o melhor par, seleciona-se as situações onde a segunda maior distância é relativamente menor (abaixo de 80% da maior).



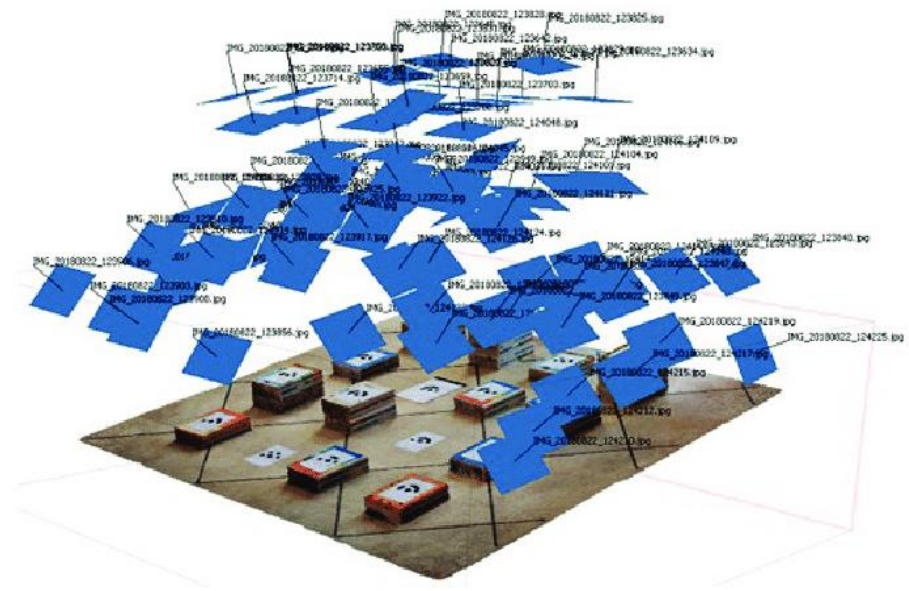


01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



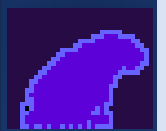
100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

- Uma vez detectados os pares homólogos, calcula-se a orientação exterior de cada imagem por bundle-block adjustment.





01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000

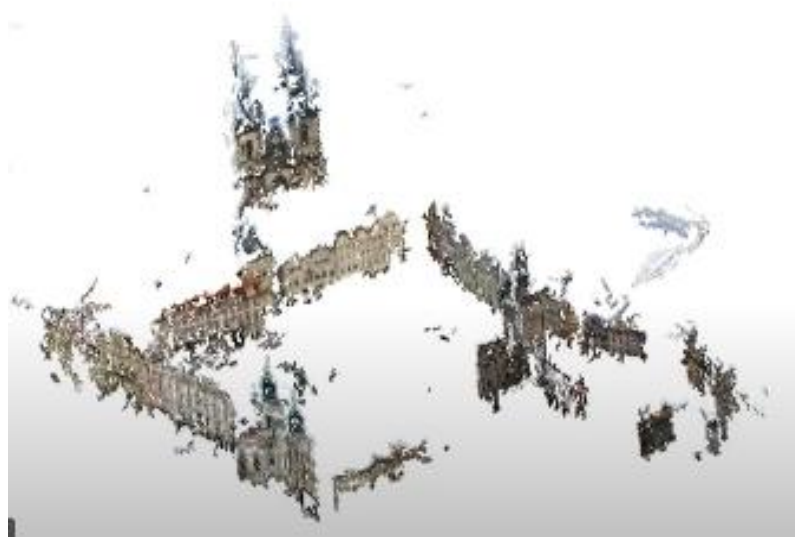


100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000

Com a posição e orientação das imagens pode-se calcular as coordenadas 3D de qualquer outro ponto.

Para isto, novos pares de pontos são detectados e a interseção dos pontos no espaço 3D é calculada (densificação)

- Veja suas aulas de fotogrametria...

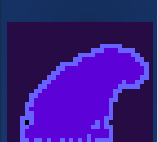


PDI-2 0100
1100
1010
1100
0000
1000

0100
1100
1010
1100
0000
1000



01001000
10102010
21011001
01001110
10010010
01001011
00110001
11100110
10010100
01010100
01000000



100101
100110
001111
001101
001010
001010
100010
000011
100110
100101
000101
01000