

PMR2560 – Visão Computacional

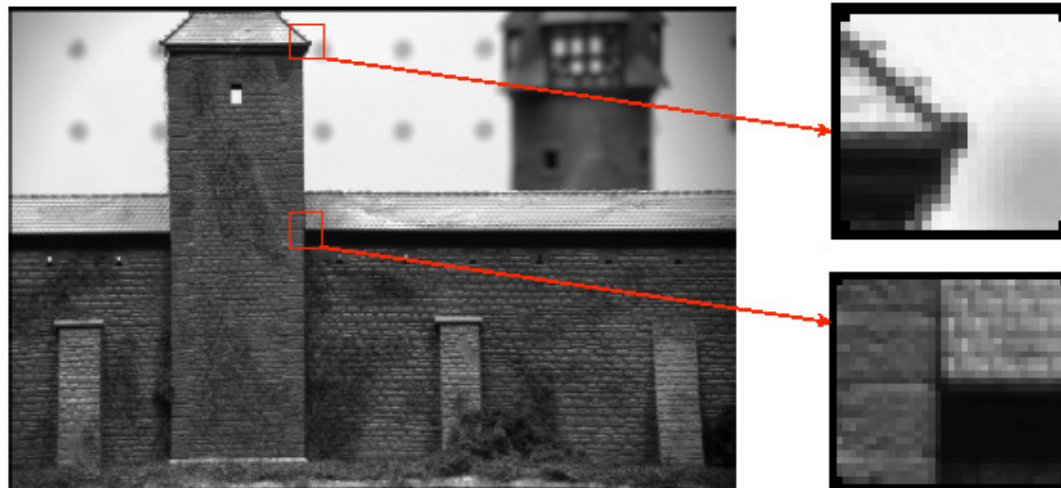
Deteccção de cantos

Prof. Eduardo L. L. Cabral



Detecção de cantos

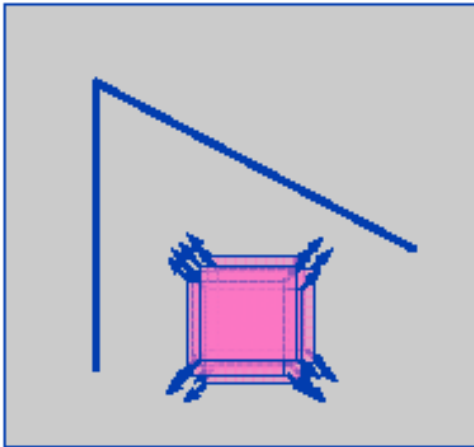
- Cantos correspondem a pontos que existem tanto no mundo real como na imagem.
- O que são cantos?
 - Pontos característicos da imagem;
 - Ponto de intersecção entre duas linhas;
 - Pontos onde o gradiente da imagem tem altas componentes em ambas as direções x e y .



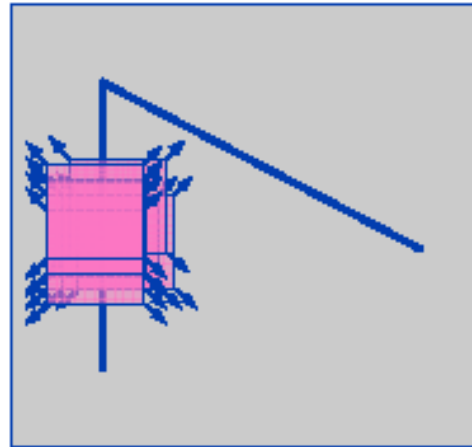
O que são cantos

- Os cantos são as características mais estáveis em relação a diferentes pontos de vista e iluminação.
- Na robótica móvel localizar pontos em imagens consecutivas é uma forma de estimar a localização e orientação do robô.
- Os cantos podem ser detectados por meio de:
 - Intersecção de linhas;
 - **Gradiente da imagem ←**

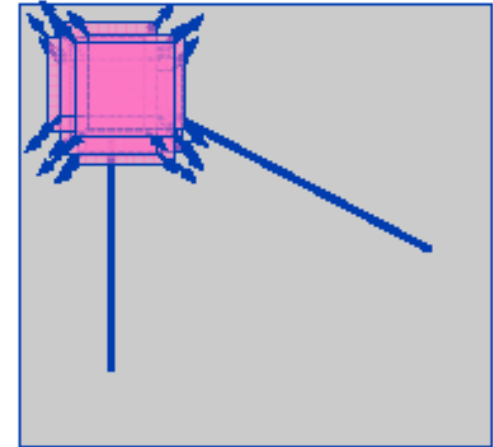
O que são cantos



- Região plana \Rightarrow nenhuma alteração em todas as direções



- Borda \Rightarrow nenhuma alteração ao longo da direção da borda



- Canto \Rightarrow alterações significativas em todas as direções

Detecção de cantos

- **Operador de Moravec:**

1. Dada uma imagem em tons de cinza: $\mathbf{I}(x,y)$;
2. Para cada pixel (i, j) da imagem \mathbf{I} , calcular:

$$\mathbf{M}(i, j) = \frac{1}{8} \sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} |\mathbf{I}(k,l) - \mathbf{I}(i, j)|$$

3. Limiarizar a nova imagem $\mathbf{M}(i,j)$ para isolar os cantos da imagem.

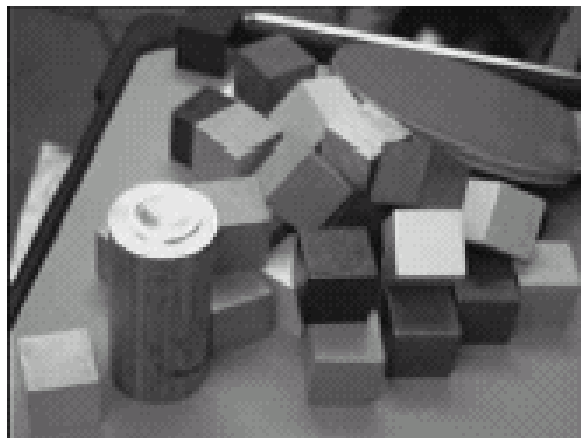
Detecção de cantos

- Características do Operador de Movarec:
 - Tem valor máximo nos pixels com alto contraste;
 - Apresenta implementação muito simples;
 - É computacionalmente eficiente → pode ser utilizado em aplicações de tempo real;
 - Não é muito eficiente na detecção ⇒ existem detectores melhores na teoria e na prática.

Detecção de cantos

- Exemplo:

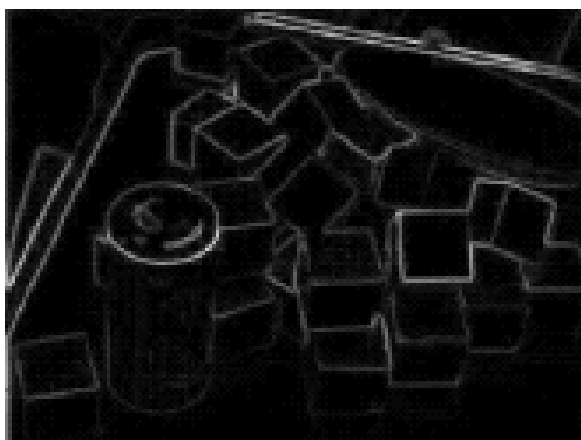
Imagem original



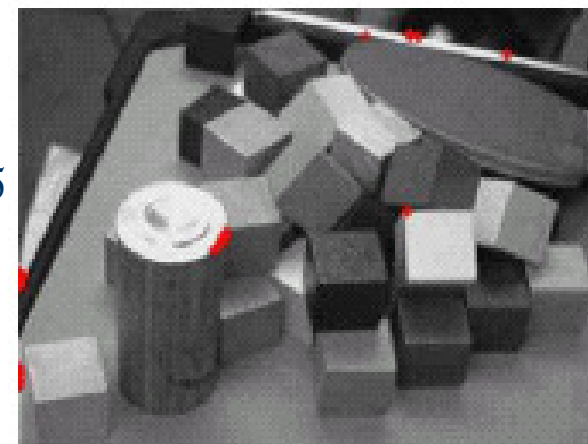
$$M(i,j) > 50$$



Operador de Movarec



$$M(i,j) > 65$$



Detecção de cantos

- **Algorithm de Trucco & Verri:**

1. Calcular o gradiente da imagem para cada pixel nas direções x e y :

$$I_x(i, j) = \frac{\partial I(x, y)}{\partial x}; \quad I_y(i, j) = \frac{\partial I(x, y)}{\partial y} j$$

2. Definir uma vizinhança como sendo uma área de interesse em torno de cada pixel:

10	11	12	12	10
11	11	10	10	10
13	13	50	55	55
13	15	53	55	58
18	19	53	60	61

Vizinhança
3x3 ($m = 1$)

Detecção de cantos

3. Para cada pixel da imagem construir a seguinte matriz usando a vizinhança m definida no passo 2:

$$\mathbf{C}(i, j) = \begin{bmatrix} c_{11}(i, j) & c_{12}(i, j) \\ c_{21}(i, j) & c_{22}(i, j) \end{bmatrix}$$

onde: $c_{11}(i, j) = \sum_{k=i-m}^{i+m} \sum_{l=j-m}^{j+m} I_x^2(k, l); \quad c_{22}(i, j) = \sum_{k=i-m}^{i+m} \sum_{l=j-m}^{j+m} I_y^2(k, l);$

$$c_{12}(i, j) = c_{21}(i, j) = \sum_{k=i-m}^{i+m} \sum_{l=j-m}^{j+m} I_x(k, l)I_y(k, l);$$

– Exemplo:

	10	11	12	12	10
	11	11	10	10	10
$I_x = 13$	13	13	50	55	55
	13	15	53	55	58
	18	19	53	60	61

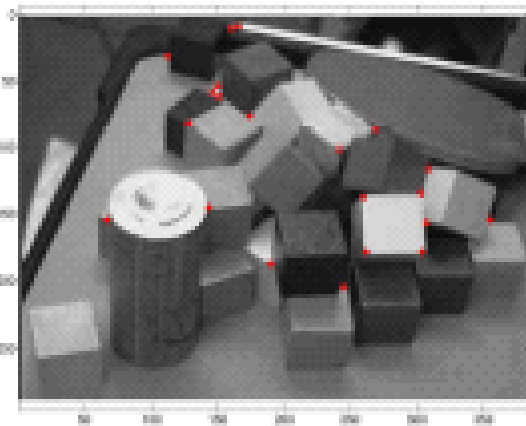
$\Rightarrow c_{11}(3,3) = 11^2 + 10^2 + 10^2 + 13^2 + 50^2 + 55^2 + 15^2 + 53^2 + 55^2$

Detecção de cantos

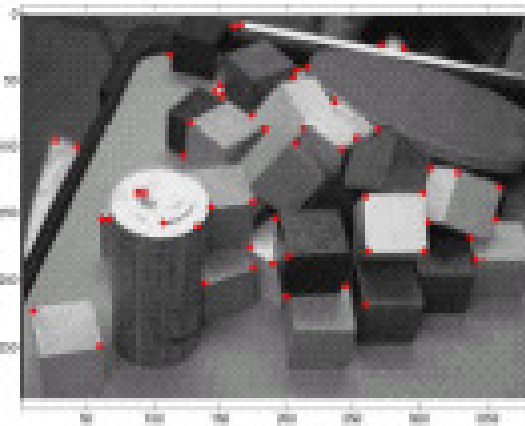
4. Para cada matriz $\mathbf{C}(i,j)$ calcular os dois autovalores λ_1 e λ_2 ;
 5. Construir uma imagem de autovalores $\mathbf{\Lambda}$, onde
$$\mathbf{\Lambda}(i,j) = \min[\lambda(i,j)];$$
 4. Limiarizar a imagem $\mathbf{\Lambda} \Rightarrow$ qualquer ponto acima do limiar é considerado um canto.
- **Problema** \Rightarrow Os cantos obtidos dependem do valor do limiar utilizado!

Detecção de cantos

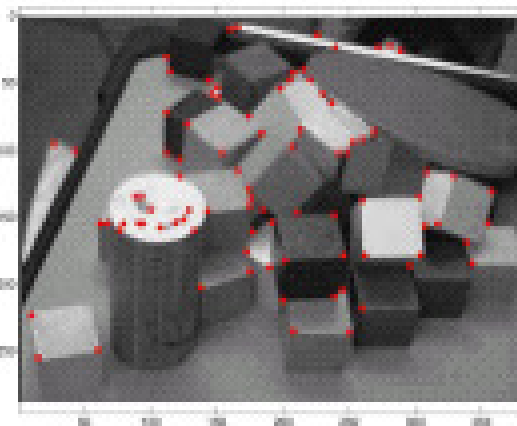
- Exemplo:



Limiar = 25.000



Limiar = 10.000



Limiar = 5.000

Detecção de cantos

- **Algorithm de Harris:**

- No lugar de calcular os autovalores da matriz $\mathbf{C}(i,j)$, para cada pixel calcula-se:

$$R(i, j) = \det[\mathbf{C}(i, j)] - k \text{traço}[\mathbf{C}(i, j)]^2$$

$k =$ constante entre 0,04 e 0,06

- R é uma medida dos autovalores de $\mathbf{C}(i,j)$:

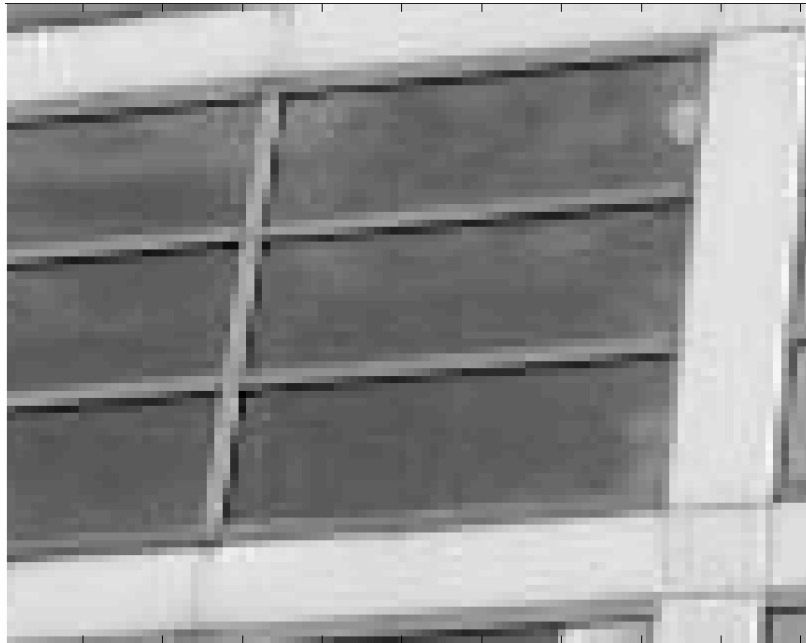
$$\det[\mathbf{C}(i, j)] = \lambda_1 \lambda_2$$

$$\text{traço}[\mathbf{C}(i, j)] = \lambda_1 + \lambda_2$$

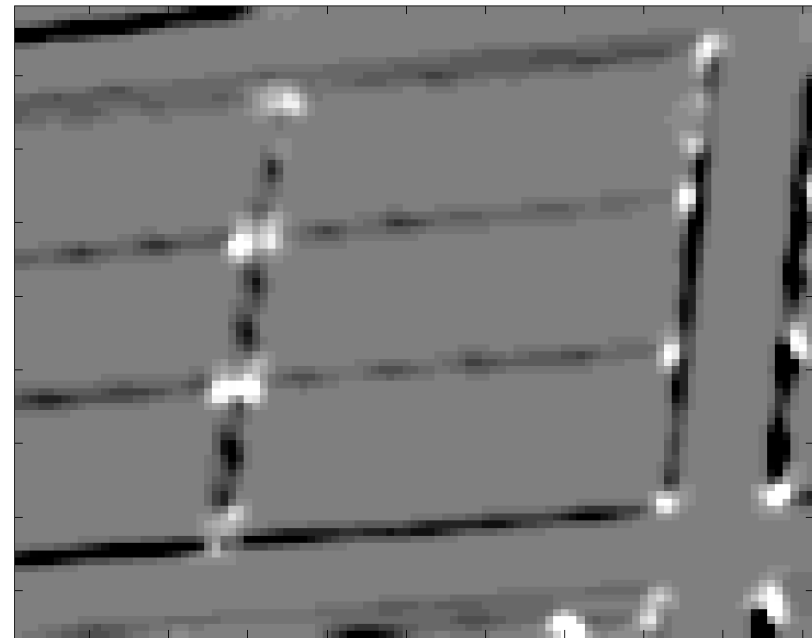
Detecção de cantos

- **Algorithm de Harris:**
 - R depende somente dos autovalores da matriz $\mathbf{C}(i,j)$;
 - R é um número grande para cantos;
 - R é um número negativo grande para bordas;
 - $|R|$ é pequeno para regiões sem grandes variações.

Detecção de cantos

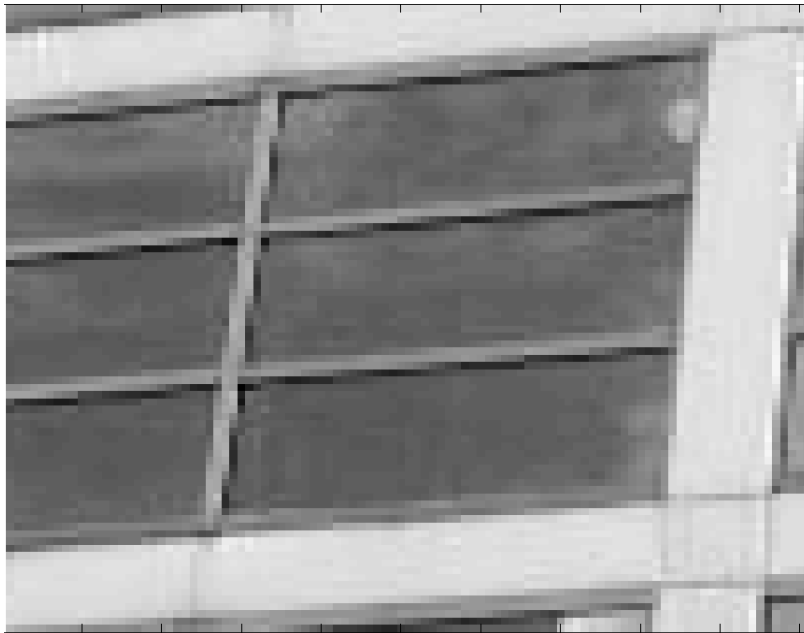


- Imagem original

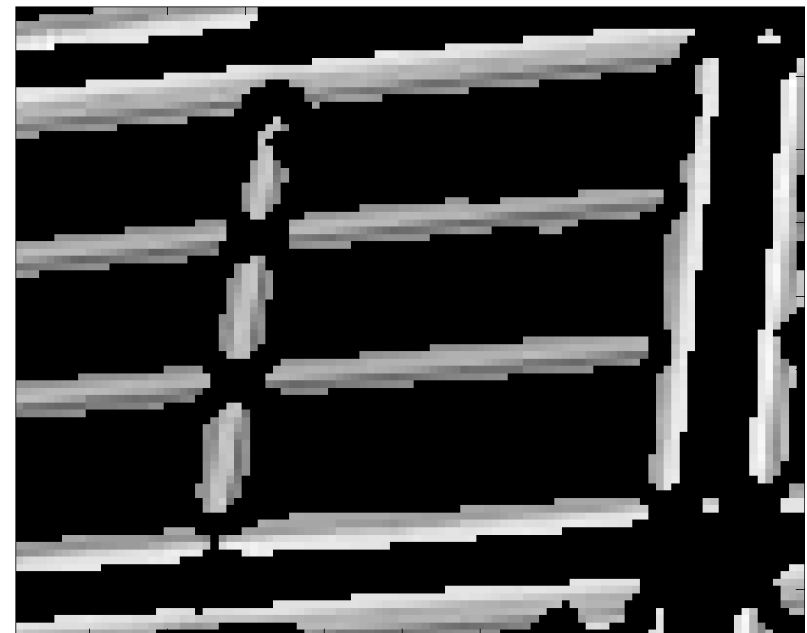


- Imagem de R

Detecção de cantos

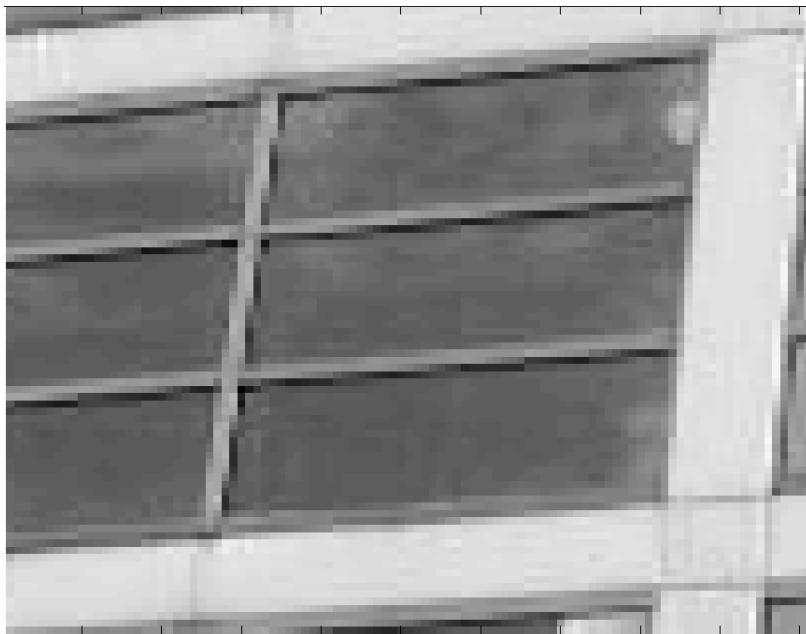


- Imagem original

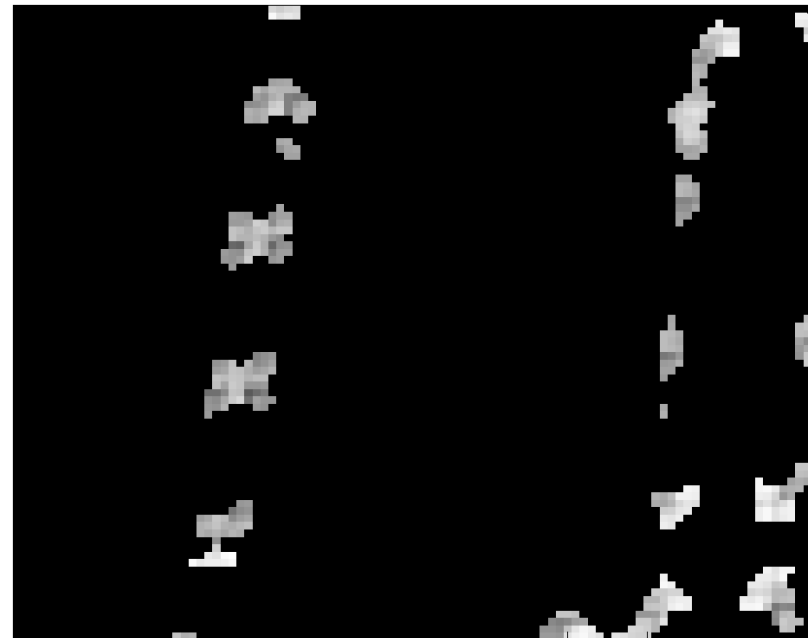


- Limiar: $R < -10000$
(bordas)

Detecção de cantos

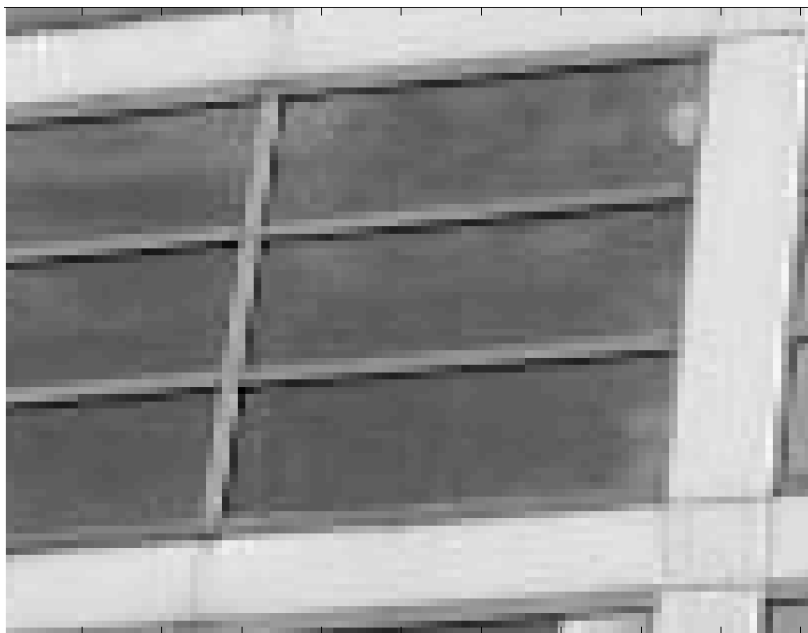


- Imagem original

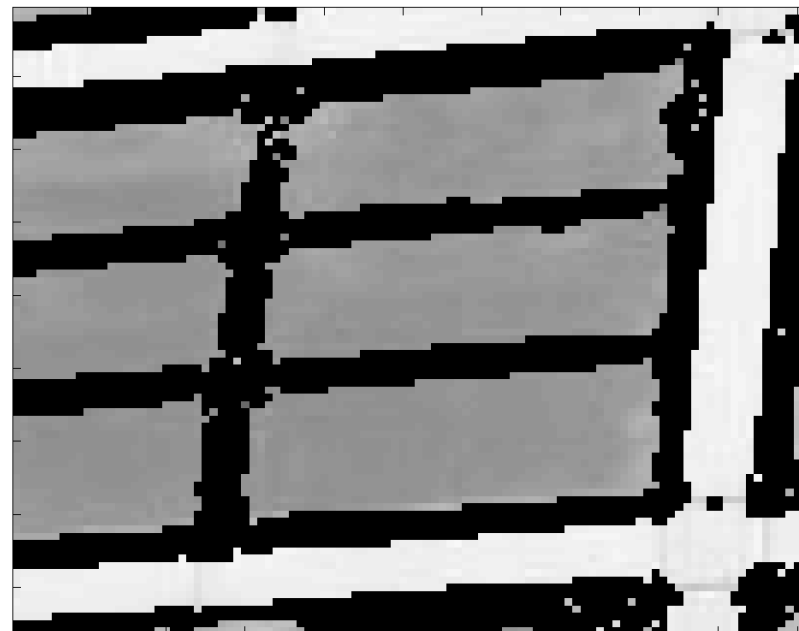


- Limiar: $R > 10000$
(cantos)

Detecção de cantos



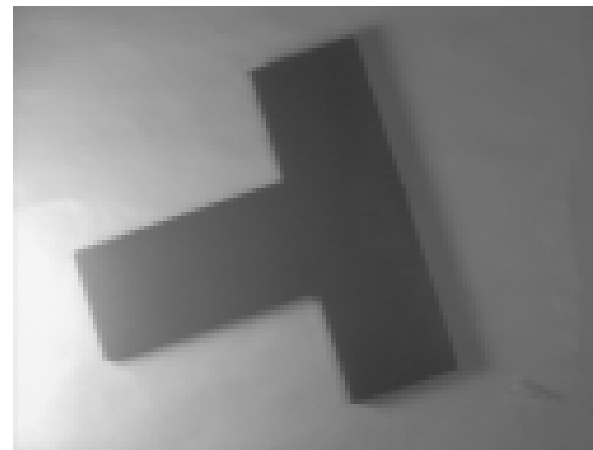
- Imagem original



- Limiar $-10000 < R < 10000$
(nem canto, nem borda)

Exercícios

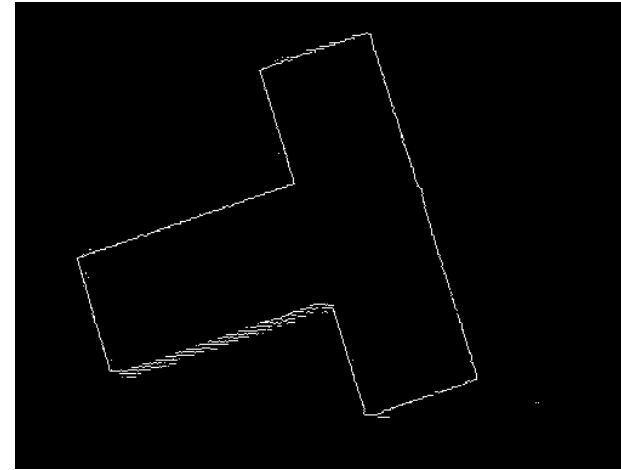
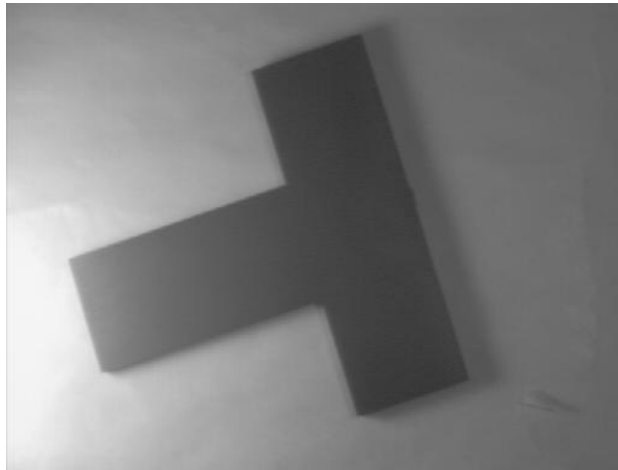
1. Dada a imagem abaixo cujas intensidades luminosas e matriz de bordas estão nas próximas páginas.
 - a) Detecte as linhas presentes na imagem.
 - b) Detecte os cantos calculando a intersecção das linhas.
 - c) Detecte os cantos da imagem usando o operador de Movarec (use a matriz de tons de cinza da imagem dada na próxima página);



Exercícios

80 80 63 80 80 80 63 80 80 80 63 80 80 80 63 80 63 46 63 46 63 46 63 46 63 46 46 63 46 46 46 46
97 80 97 97 97 80 97 97 97 63 80 80 97 63 80 46 80 63 80 46 80 46 80 46 46 29 46 46 63 46 46 46 46
97 80 80 97 97 97 97 97 97 80 80 80 80 80 80 63 80 63 80 63 46 12 0 12 29 29 46 46 46 46 46 46
97 97 97 97 114 97 97 97 114 97 97 80 97 80 97 63 80 80 80 29 29 0 0 0 29 29 46 46 63 46 46 46 46
114 97 97 97 114 97 97 97 114 97 97 80 80 97 63 80 63 29 0 12 0 0 0 12 29 12 29 63 46 46 46 63
131 114 131 97 131 114 114 97 114 97 97 97 80 97 80 80 0 29 0 12 0 0 0 12 29 46 63 46 46 46 63
114 131 114 131 114 131 114 97 114 97 97 97 80 80 80 80 29 12 0 12 0 12 0 0 12 12 29 46 46 63 46 46
131 131 131 114 131 114 131 97 131 97 97 97 97 97 80 97 46 29 0 29 0 12 0 12 0 46 29 46 46 63 46 63
131 131 114 131 131 131 114 131 114 97 97 97 97 97 80 80 12 0 12 0 0 0 12 0 12 29 29 46 46 46 63
148 131 148 131 148 114 131 97 131 97 114 97 114 97 97 97 80 29 0 29 0 12 0 12 0 29 12 46 46 63 46 63
148 148 131 131 131 131 114 131 114 131 114 97 114 97 97 97 80 46 0 12 0 12 0 0 12 29 29 46 63 46 46
165 148 148 148 148 131 148 114 131 114 131 97 131 97 114 97 97 80 0 29 0 29 0 12 0 12 29 46 80 46 80
165 148 148 148 148 148 131 131 131 131 114 131 114 97 97 97 114 97 63 29 12 0 12 0 12 0 29 29 46 46 63
182 148 165 148 165 148 148 148 148 114 131 97 131 114 114 63 46 12 29 0 29 12 12 12 12 0 29 0 12 0 46 12 46 63
165 182 165 182 165 148 165 148 148 131 131 131 97 46 29 29 12 29 12 12 12 12 12 0 12 0 12 0 12 29 46 46 46
189 182 182 165 182 148 182 148 148 97 80 29 46 29 46 12 29 29 29 12 29 0 29 0 29 0 29 0 29 29 46 46 80
182 182 165 182 165 148 131 80 63 46 46 29 29 29 29 29 29 12 29 12 0 12 0 12 0 12 0 12 0 12 29 29 46 63
189 182 182 165 148 80 80 46 80 46 46 46 46 29 46 29 46 12 29 0 29 12 29 0 12 0 29 0 29 12 29 46 63
182 182 182 182 148 80 80 80 63 46 46 46 46 29 29 29 12 29 12 12 12 12 12 0 12 0 12 0 12 29 29 29 46
189 182 189 182 182 80 97 63 80 46 63 46 46 29 46 29 29 29 29 12 29 12 0 12 0 12 0 29 0 29 12 46 29 46
182 189 182 182 182 97 63 80 63 46 46 46 29 46 29 29 29 29 12 29 12 0 12 0 12 0 0 0 12 0 12 29 29
189 182 189 189 189 114 80 63 80 46 63 46 46 29 46 29 29 12 46 46 80 63 29 0 29 0 12 0 12 0 29 29 46
182 182 182 182 182 148 80 80 63 46 46 46 46 29 29 46 63 80 80 80 80 12 0 12 0 12 0 12 0 12 29 29
189 182 189 182 189 165 97 63 80 46 46 29 63 63 97 97 114 97 97 80 97 80 46 0 29 0 29 0 12 0 29 12 29
182 189 182 182 182 182 97 80 63 46 63 97 114 97 114 97 114 97 97 80 80 80 63 0 12 0 0 0 12 0 0 29 29
189 182 189 182 189 165 148 97 131 114 131 114 131 114 131 97 114 97 97 97 80 80 0 12 0 0 0 12 0 0 0 46
182 182 182 182 165 148 165 148 148 131 131 131 114 131 114 97 97 97 80 97 80 80 29 0 0 0 0 0 12 29 46
189 182 189 165 182 148 148 148 148 131 148 114 131 97 131 97 97 97 80 97 80 97 46 0 0 29 12 46 46 46 63
182 182 165 182 165 148 148 148 131 148 114 131 114 97 114 97 97 97 80 80 80 63 80 63 46 46 46 63 46 63
189 182 182 148 182 148 148 148 148 131 131 97 131 97 97 97 97 97 97 63 80 63 80 63 80 46 63 46 63 46 63
165 182 165 182 165 148 148 148 148 131 114 97 114 97 97 97 97 80 80 80 80 63 80 63 63 63 46 63 46 63 46 46
182 148 182 148 165 148 148 131 131 131 131 97 131 97 114 97 97 97 80 80 63 80 63 80 46 80 46 80 46 63 46 63

Exercícios



Exercícios

2. Detecte as linhas e os cantos presentes nas imagens:

