

رسالة محمد



مبانی بینایی کامپیوتر

مدرس: محمدرضا محمدی

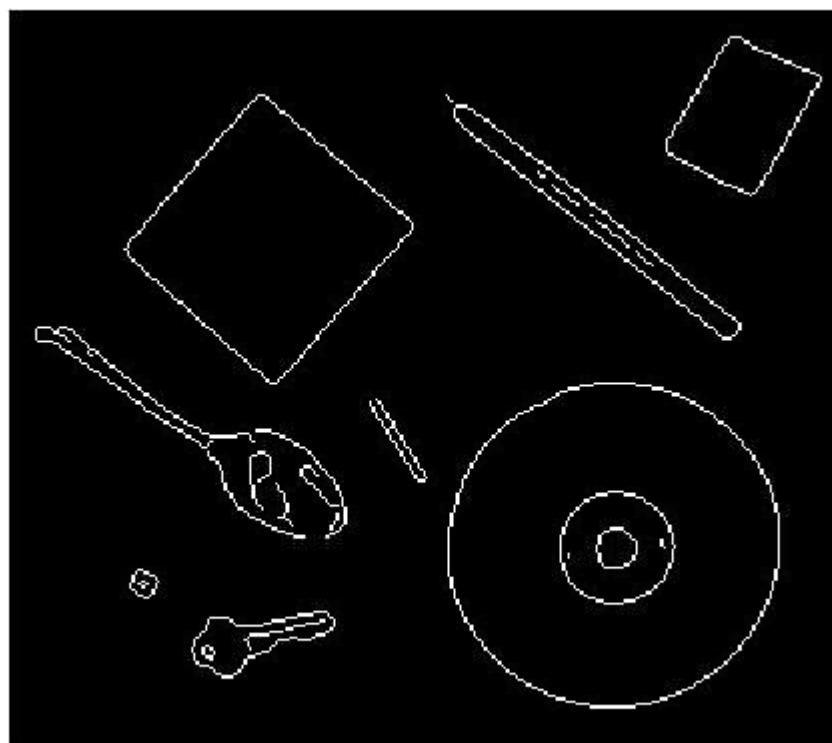
۱۴۰۱

استخراج شكل

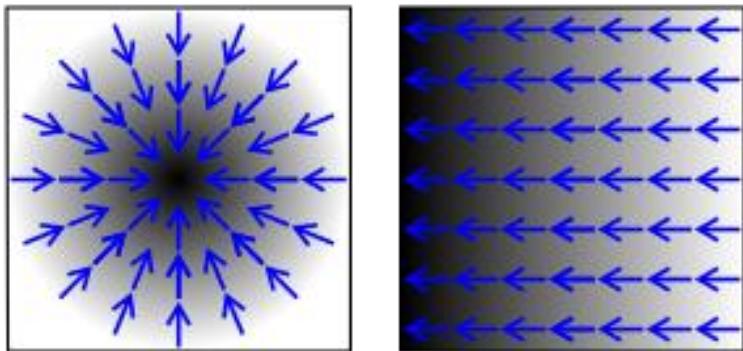
Shape Extraction

تشخیص لبه

- یک لبه، مجموعه‌ای از پیکسل‌هایی به هم پیوسته است که روی مرز دو ناحیه قرار دارند



گرادیان تصویر



$$\nabla f(x, y) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- گرادیان تابع دوبعدی f به صورت زیر تعریف می‌شود:

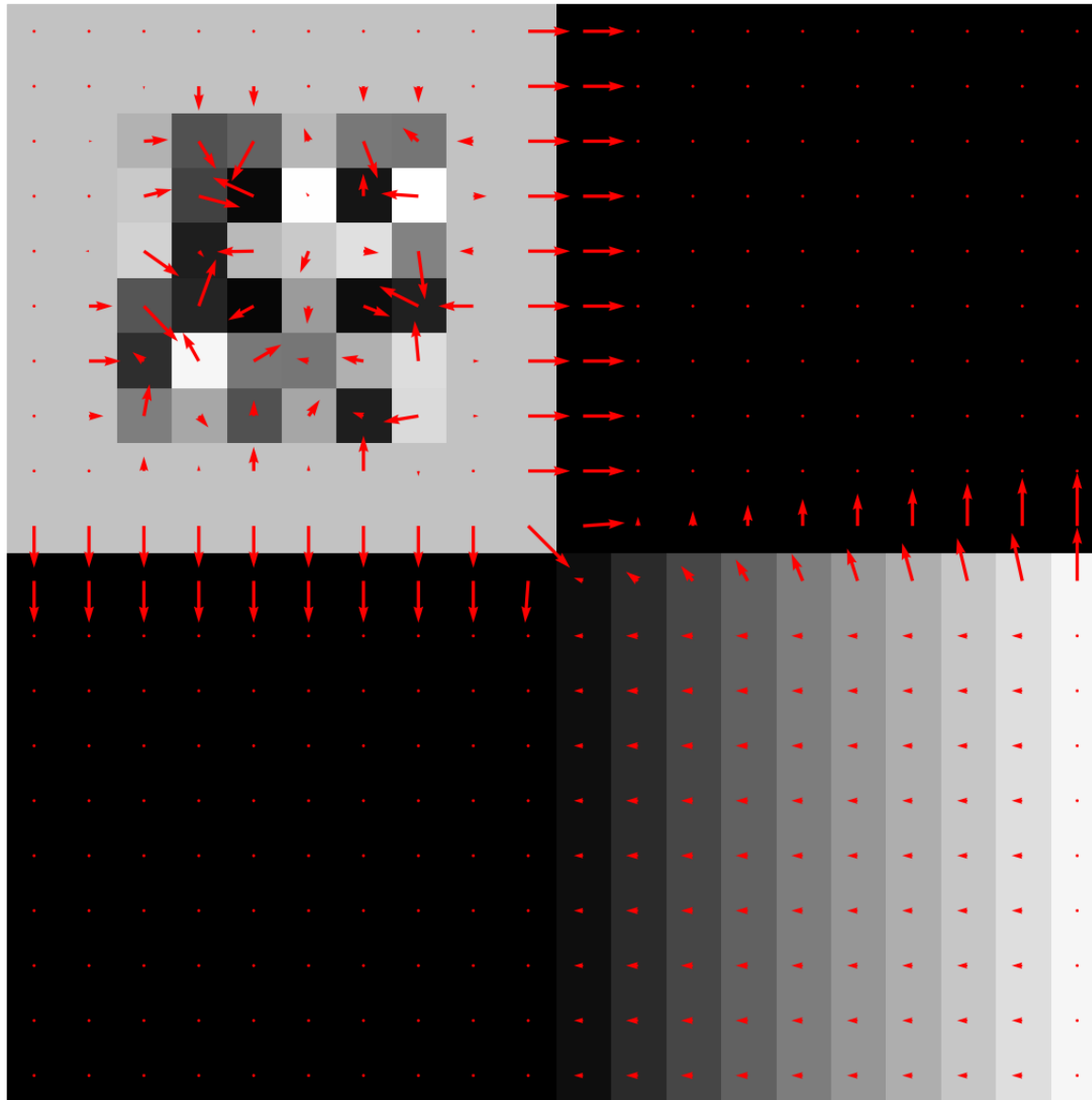
$$M(x, y) = \|\nabla f\| = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2} \approx |g_x| + |g_y|$$

- اندازه گرادیان

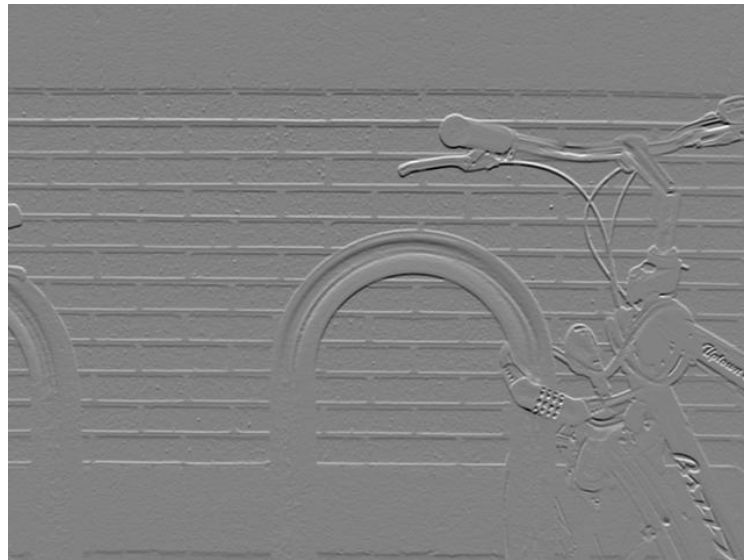
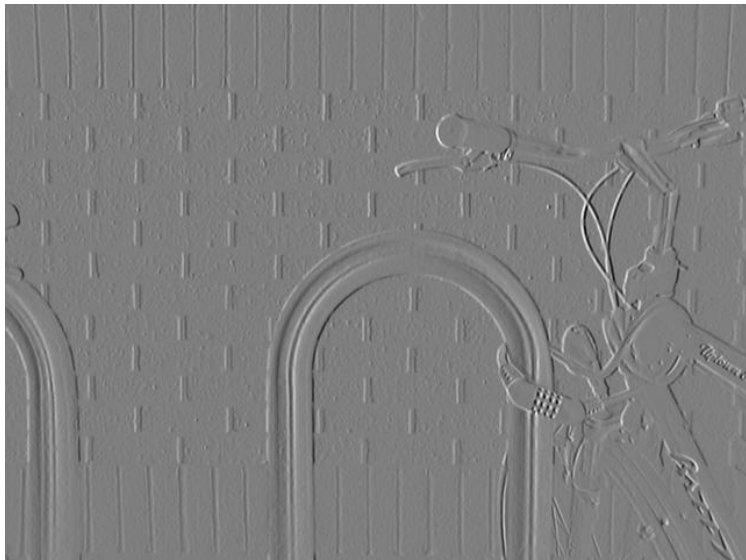
$$\alpha(x, y) = \text{dir}(\nabla f) = \text{atan2}(g_y, g_x)$$

- جهت گرادیان

گرادیان تصویر



گرادیان تصویر



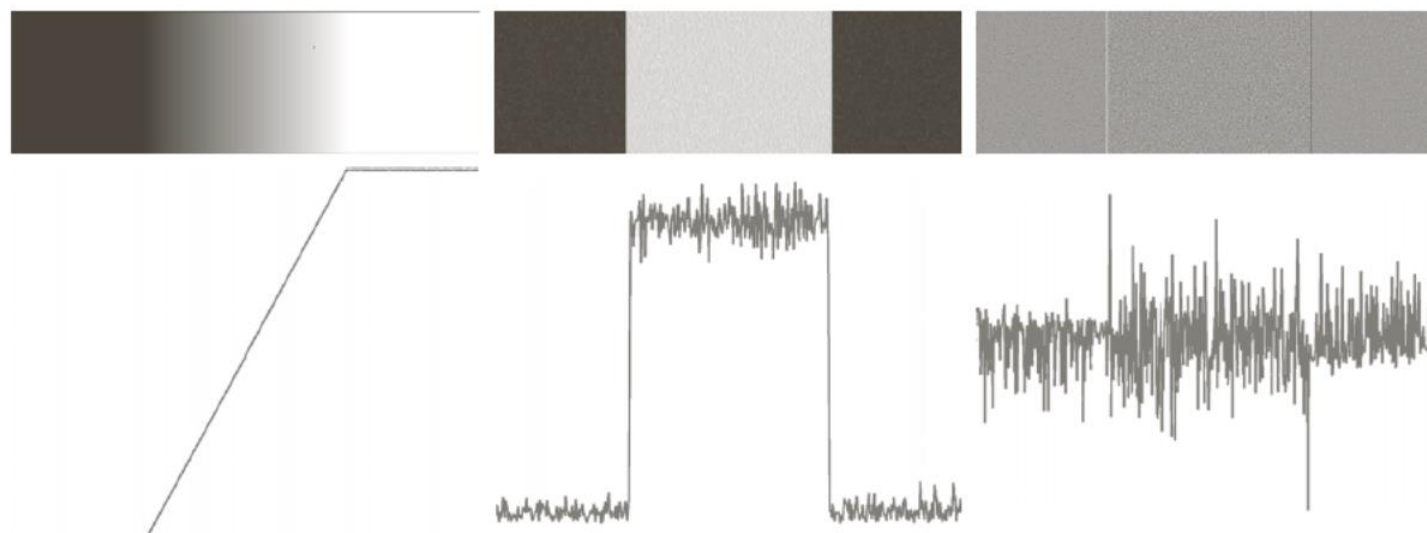
مشتق افقی

- مشتق یک طرفه

-1	+1
----	----

- مشتق دو طرفه

-1	0	+1
----	---	----



مشتق افقی

• عملگر Prewitt

$$\begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix} \begin{bmatrix} -1 & 0 & +1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix}$$

• عملگر Sobel

$$\begin{bmatrix} +1 \\ +2 \\ +1 \end{bmatrix} \begin{bmatrix} -1 & 0 & +1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

مشتق عمودی

• عملگر Prewitt

$$\begin{bmatrix} -1 \\ 0 \\ +1 \end{bmatrix} \begin{bmatrix} +1 & +1 & +1 \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix}$$

• عملگر Sobel

$$\begin{bmatrix} -1 \\ 0 \\ +1 \end{bmatrix} \begin{bmatrix} +1 & +2 & +1 \end{bmatrix} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

لبه‌یاب Sobel

- فیلترهای Sobel برای یافتن لبه‌های افقی و عمودی مناسب هستند

G_y				G_x			
-1	-2	-1		-1	0	+1	
0	0	0		-2	0	+2	
+1	+2	+1		-1	0	+1	

$$\text{mag} = \sqrt{g_x^2 + g_y^2}$$

$$\text{dir} = \text{atan2}(g_y, g_x)$$





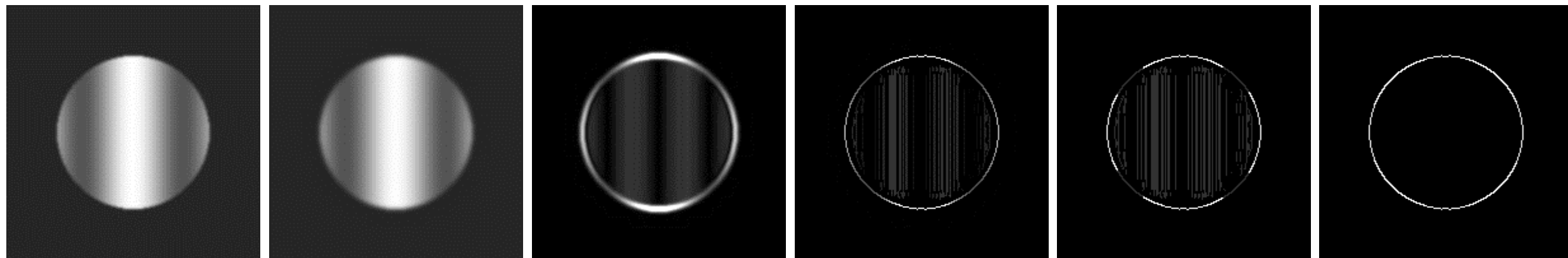
لبه‌یاب Sobel

- آستانه‌گذاری اندازه گرادیان حاصل از عملگر Sobel
- مقادیر بیش از ۰.۳۳ از بزرگترین مقدار با به عنوان لبه در نظر می‌گیریم



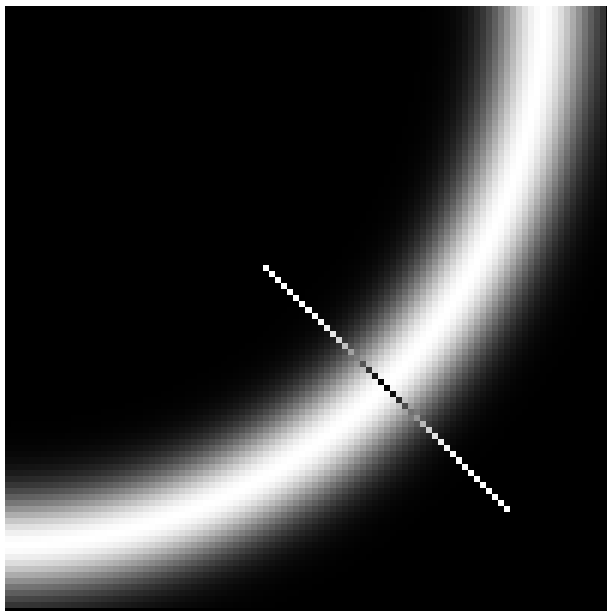
لبه‌یاب Canny

- یکی از پرکاربردترین و موفق‌ترین روش‌های لبه‌یابی است که از ۴ گام اساسی تشکیل می‌شود:
 - هموار کردن تصویر با استفاده از فیلتر گاوسی
 - محاسبه گرادیان
 - حذف مقادیر غیربیشینه
 - آستانه‌گذاری دو مرحله‌ای

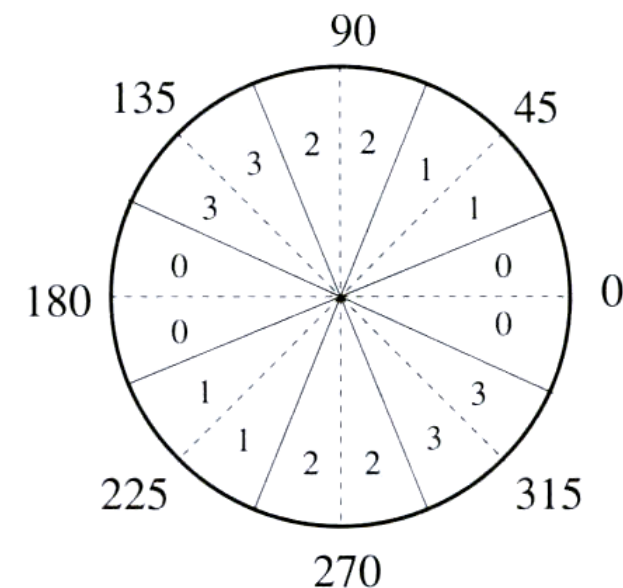


حذف مقادیر غیربیشینه

- هر پیکسل که در راستای گرادیان خود دارای مقدار غیربیشینه باشد حذف می شود
- جهت گرادیان به ۴ گروه تقسیم می شود و همسایگی 3×3 است

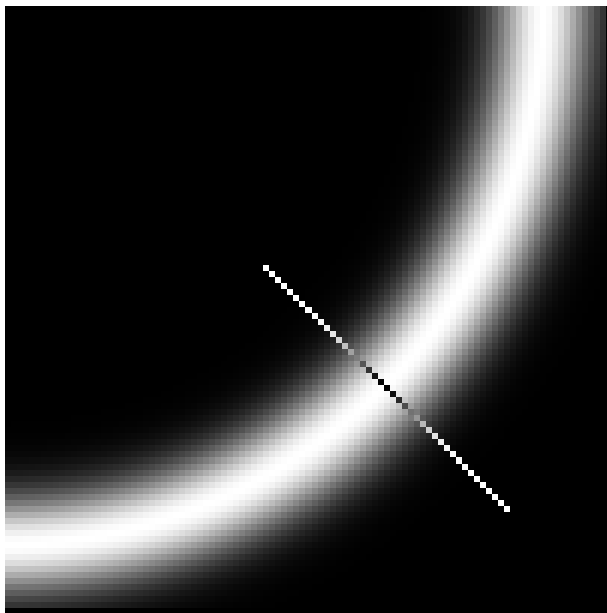


0	0	0	0	1	2	1	3
0	0	0	1	2	1	3	1
0	0	2	1	2	1	1	0
0	1	3	2	1	1	0	0
0	3	2	1	0	0	0	0
2	3	2	0	0	1	0	1
2	3	2	0	1	0	2	0

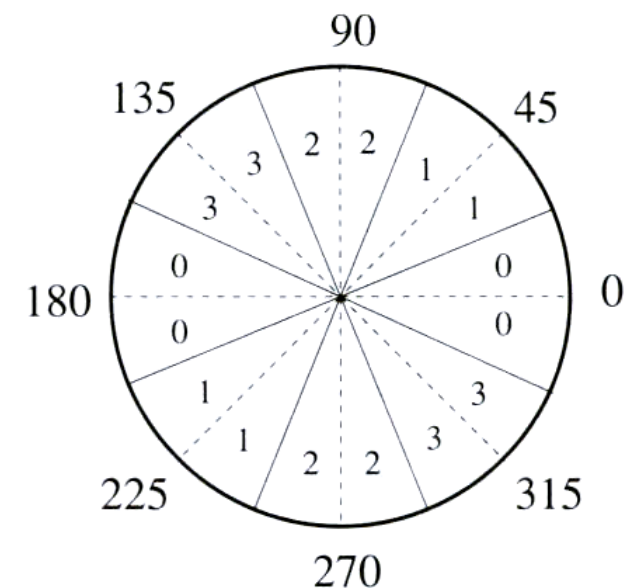


حذف مقادیر غیربیشینه

- هر پیکسل که در راستای گرادیان خود دارای مقدار غیربیشینه باشد حذف می شود
- جهت گرادیان به ۴ گروه تقسیم می شود و همسایگی 3×3 است

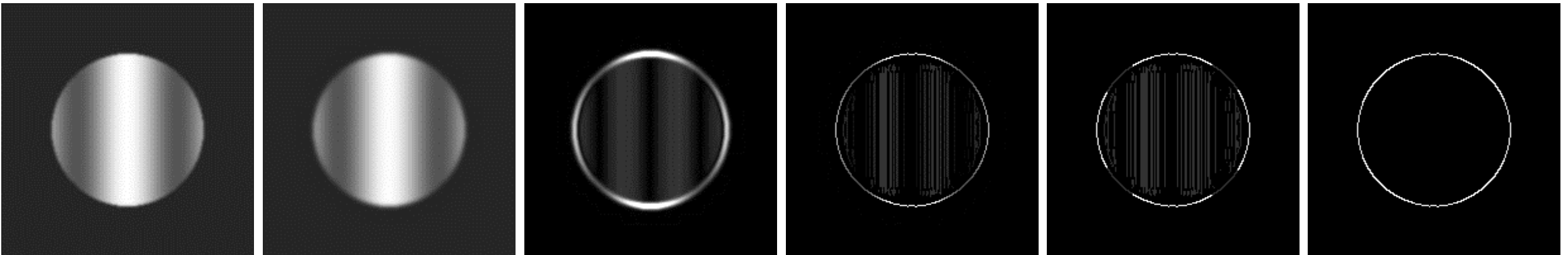


0	0	0	0	0	0	0	3
0	0	0	0	2	1	3	0
0	0	2	1	2	0	0	0
0	0	3	0	0	0	0	0
0	3	2	0	0	0	0	0
0	3	0	0	0	1	0	1
0	3	0	0	1	0	2	0



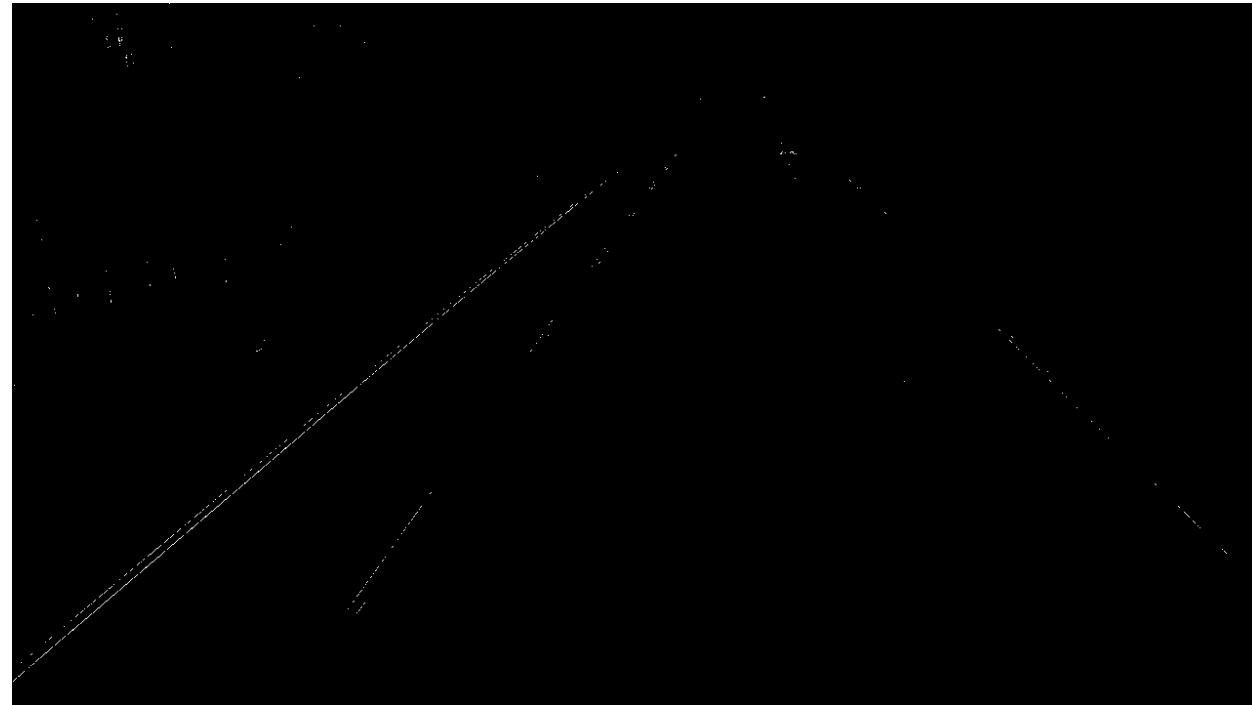
آستانه گذاری دوسطحی

- هر پیکسلی که اندازه گرادیان آن کوچکتر از T_1 باشد به عنوان غیر لبه معرفی می شود
- هر پیکسلی که اندازه گرادیان آن بزرگتر از T_2 باشد به عنوان لبه معرفی می شود
- پیکسل هایی که اندازه گرادیان آنها بین T_1 و T_2 باشد تنها در صورتی به عنوان لبه معرفی می شوند که به یک پیکسل لبه به صورت مستقیم یا از طریق پیکسل هایی که اندازه گرادیان آنها بین T_1 و T_2 است متصل باشند



لبه‌یاب Canny

```
edges = cv2.Canny(img, 800, 800)
```



لبه‌یاب Canny

```
edges = cv2.Canny(img, 100, 800)
```



لبه‌یاب Canny

```
edges = cv2.Canny(img, 100, 200)
```



تشخیص خط

- در حالت ساده فرض می‌کنیم در تصویر تنها ۱ خط وجود داشته باشد

$$y = mx + c$$

- معادله خط

- چطور می‌توان m و c را بدست آورد؟

$$m, c = \arg \min \sum_i (mx_i + c - y_i)^2$$

$$m = \frac{\bar{x}\bar{y} - \overline{xy}}{\bar{x}^2 - \overline{x^2}}$$

$$c = \bar{y} - m\bar{x}$$

