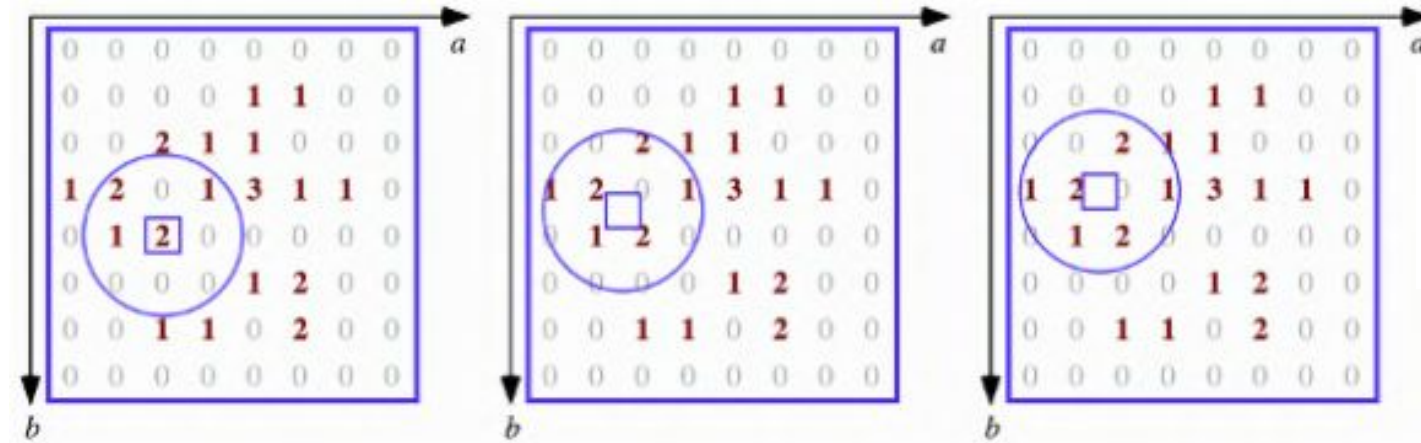


Mean shift



[(5, 2), (6, 2), (3, 3), (3, 3), (4, 3),
 (5, 3), (1, 4), (2, 4), (2, 4), (4, 4),
 (5, 4), (5, 4), (5, 4), (6, 4), (7, 4),
 (2, 5), (3, 5), (3, 5), (5, 6), (6, 6),
 (6, 6), (3, 7), (4, 7), (6, 7), (6, 7)]

$$D = \begin{bmatrix} 5 & 6 & 3 & 3 & 4 & 5 & 1 & 2 & 2 & 4 & 5 & 5 & 5 & 6 & 7 & 2 & 3 & 3 & 5 & 6 & 6 & 3 & 4 & 6 & 6 \\ 2 & 2 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 5 & 5 & 5 & 6 & 6 & 6 & 7 & 7 & 7 & 7 \end{bmatrix}.$$

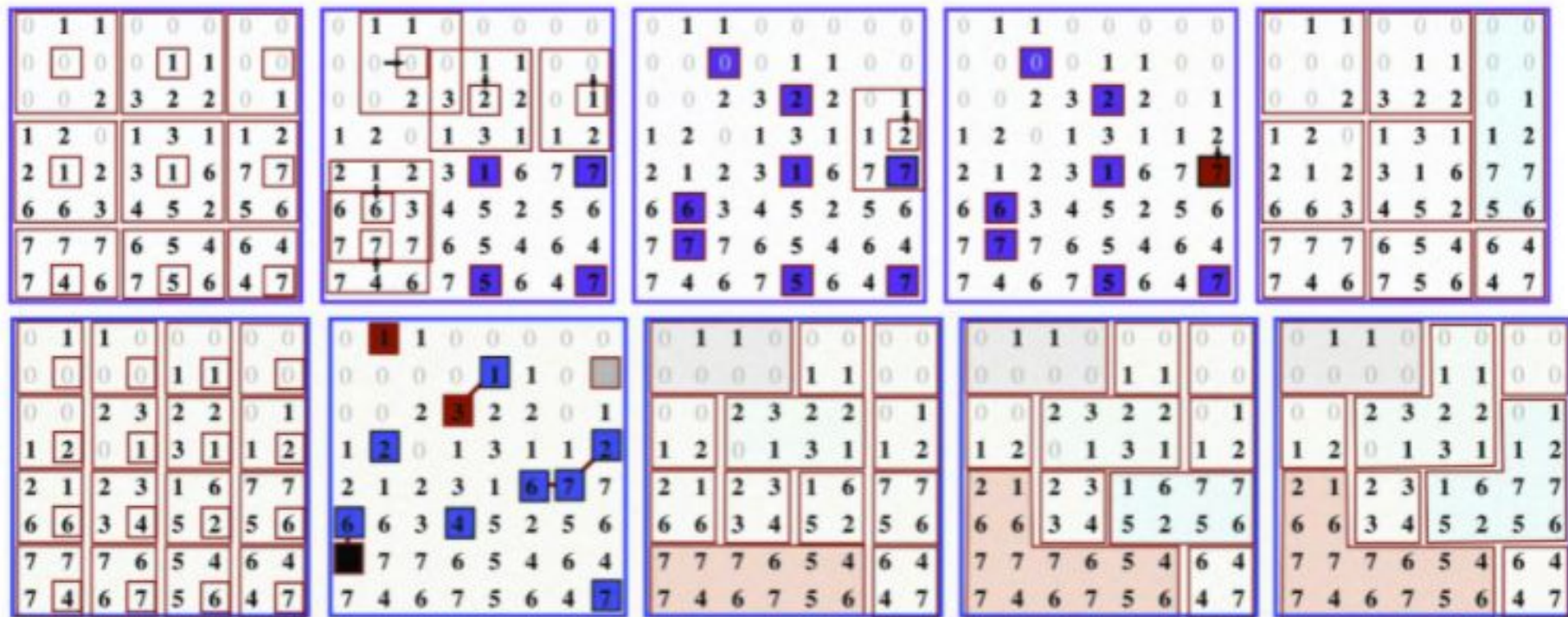
$$M = \begin{bmatrix} 3 & 3 \\ 5 & 5 \end{bmatrix}.$$

$$D - M \Rightarrow E = \begin{bmatrix} 4 & 9 & 0 & 0 & 1 & 4 & 4 & 1 & 1 & 1 & 4 & 4 & 4 & 9 & 1 & 6 & 1 & 0 & 0 & 4 & 9 & 9 & 0 & 1 & 9 & 9 \\ 9 & 9 & 4 & 4 & 4 & 4 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 4 & 4 & 4 & 4 \end{bmatrix}.$$

$$[13 \ 18 \ 4 \ 4 \ 5 \ 8 \ 5 \ 2 \ 2 \ 2 \ 5 \ 5 \ 5 \ 10 \ 17 \ 1 \ 0 \ 0 \ 5 \ 10 \ 10 \ 4 \ 5 \ 13 \ 13].$$

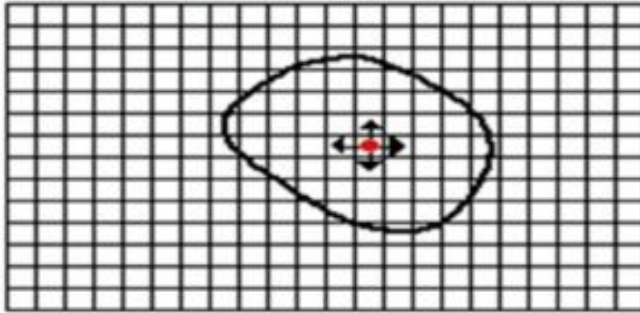
$$S_{<r^2} = \{(2, 4), (2, 4), (4, 4), (2, 5), (3, 5), (3, 5)\} \quad r^2 = 2,56.$$

$$\mathbf{u}_s = (2,67, 4,5)$$

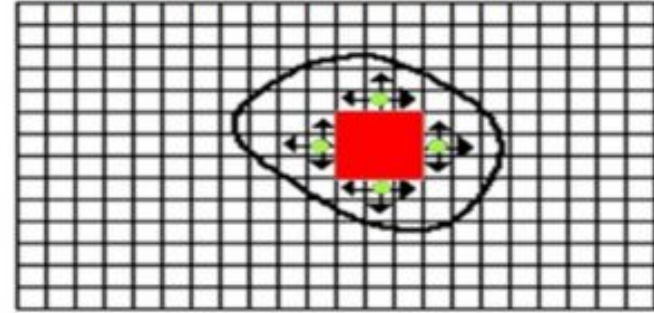


Region growth

Початок росту області



Процес росту після декількох ітерацій

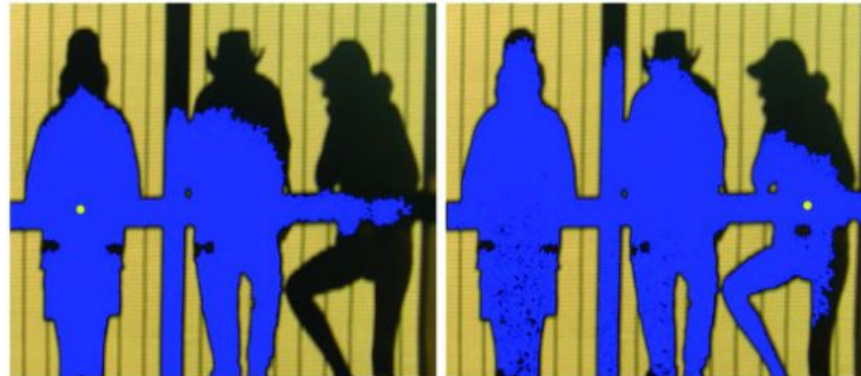
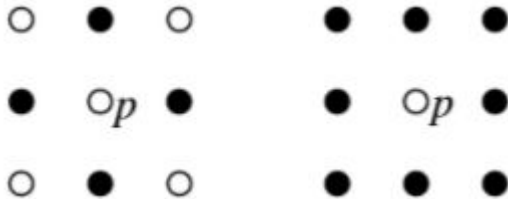


● Вихідний піксель

↑ Напрямок росту

■ Область, що виростає

● Пікселі, що розглядаються





Пікселі в MeanShift відносяться до одного зображення якщо

їх велика ймовірність віднесення до одного й того ж кластера

що відстань від пікселів до центра k-го кластера маленька

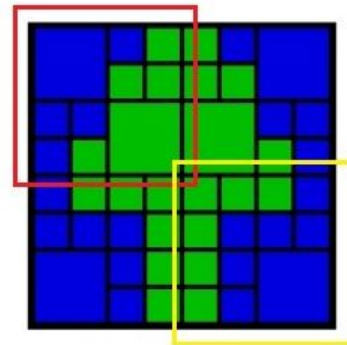
середнє значення обраховане для обох пікселів однакове або схоже

якщо пікселі одного кольору



Split & Merge

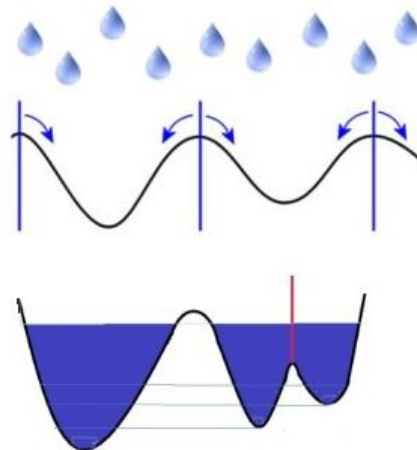
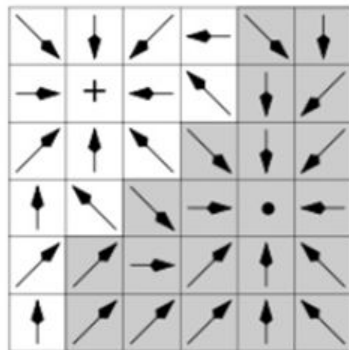
1. Визначаємо критерій, який слід використовувати для однорідності
2. Розділяємо зображення на рівні за розміром області
3. Обчислюємо однорідність для кожної області
Якщо область однорідна, то об'єдуємо її з сусідніми областями
4. Процес повторюють, поки всі області не пройдуть тест на однорідність

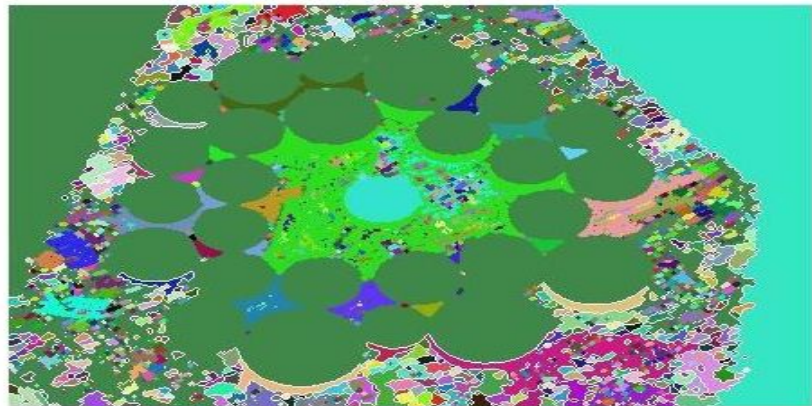


Метод водорозділу (WaterShed)

- Початкові точки вибираються по мінімальних значеннях інтенсивності. При цьому кількість вихідних пікселів (відповідно і областей) визначається наперед. Нарощування областей відбувається паралельно, підняттям рівня інтенсивності до тих пір, поки не встановиться конкуренція по встановленню близькості між сусідніми областями.

16.8	19.2	13.5	20.5	31.2	30.1
18.7	11.9	15.4	18.5	22.1	18.4
20.1	21.9	26.6	20.8	17.3	18.1
25.3	22.8	20.9	19.8	15.1	15.9
30.7	35.5	29.9	18.7	17.6	39.9
34.8	38.6	33.4	32.7	33.5	36.7





Нормалізовані розрізи графа

- Розріз між двома групами A і B визначається як сума всіх зважених ваг,

$$cut(A, B) = \sum_{i \in A, j \in B} w_{ij},$$



де ваги між двома пікселями i і j відповідають їх схожості.

- (Normalized cuts) Кращою мірою сегментації є нормалізований зріз, який визначається як

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)},$$

$$assoc(A, A) = \sum_{i \in A, j \in A} w_{ij}$$

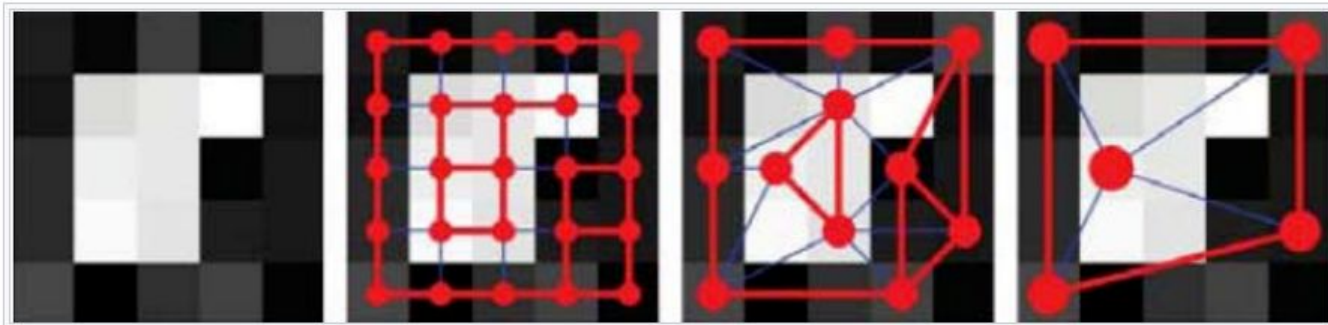


це асоціація (сума
всіх ваг) в кластері

$$assoc(A, V) = assoc(A, A) + cut(A, B)$$



це сума всіх ваг
асоційованих з A



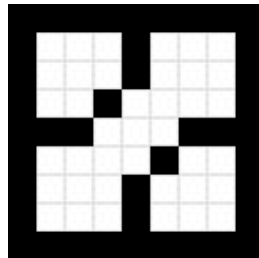
GrabCut

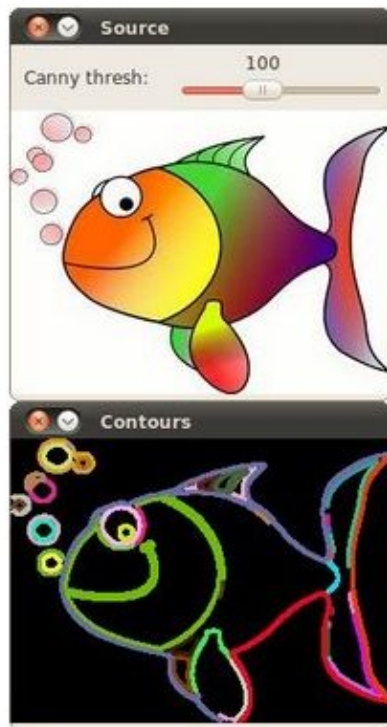
Алгоритм GrabCut :

- Обирає вхідне зображення або (1) **обмежувальною рамкою** , що зазначає розташування об'єкта на зображенні в якості бажаного сегмента або (2) **маски** , яка *апроксимує* сегментацію
- Покроково:
 - **Крок 1.** Оцінка розподілу кольорів переднього та заднього планів за допомогою моделі Гаусової суміші (GMM)
 - **Крок №2:** Побудова випадкового поля Маркова над мітками пікселів (тобто, передній план проти фону)
 - **Крок №3:** Застосування оптимізації вирізання графів для досягнення остаточної сегментації

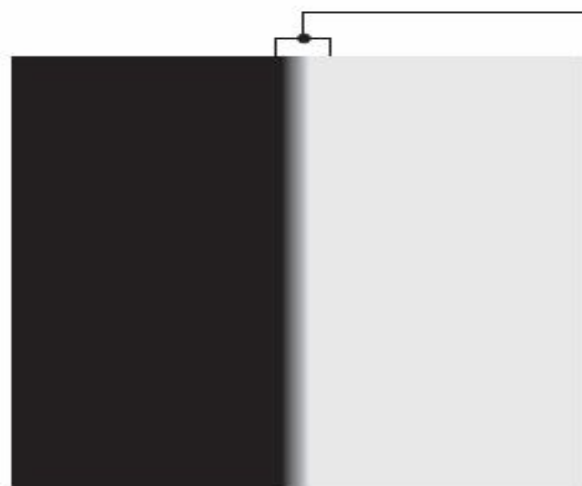
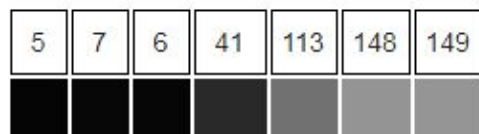
FloodFill

- За допомогою алгоритма затоплення, (FloodFill, Seed Fill) можна виділити однорідні за кольором регіони. Для цього потрібно вибрати початковий піксель і задати інтервал зміни кольору сусідніх пікселів щодо вихідного. Алгоритм буде об'єднувати пікселі в один сегмент (заливаючи їх одним кольором), якщо вони потрапляють в зазначений діапазон. На виході буде сегмент, залитий певним кольором, і його площа в пікселях.





Виділення меж
зображення

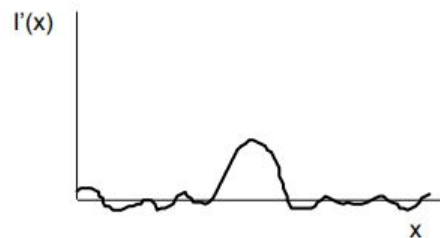
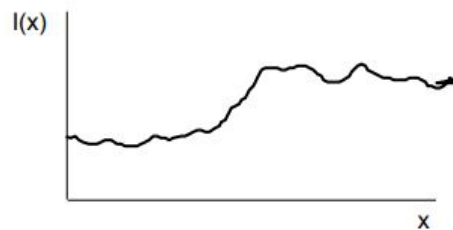


Профіль яскравості
по горизонталі

Перша
похідна

Друга
похідна

Точка перетину
нульового рівня



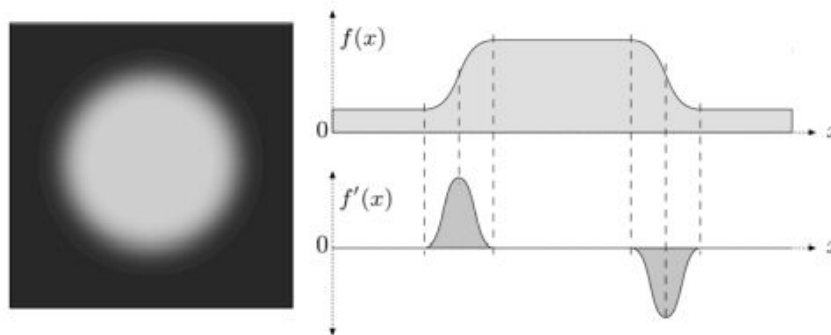
Градiєнт зображення $L(x, y)$ в точці x, y являє собою вектор \mathbf{G} , орієнтований в напрямку максимальної зміни яскравості, модуль якого рівен

$$|\mathbf{G}| = \sqrt{G_x^2 + G_y^2},$$

$$\text{де } G_x = \partial L_c(x, y) / \partial x, \quad G_y = \partial L_c(x, y) / \partial y$$

компоненти цього вектора. Цей вектор з вісью x складає кут $\alpha(x, y)$, який рівний $\alpha(x, y) = \arctg(G_y / G_x)$.

$$f'(x) = \frac{df}{dx}(x)$$



Прості детектори меж

- Оператор Робертса використовує наступні ядра 2 x 2:

$$M_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad M_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

- Оператор Превітта Зображення сканується ядром розміром 3 x 3 пікселя.

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad M_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

- Оператор Собеля Збільшення ваги в ядрі дозволяє дещо зменшити вплив шуму на результат обчислення градієнта

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad M_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Лапласіан

- Стандартні (прості) апроксимації Лапласіан зображення I визначаються з гаступними ядрами.

0	1	0
1	-4	1
0	1	0

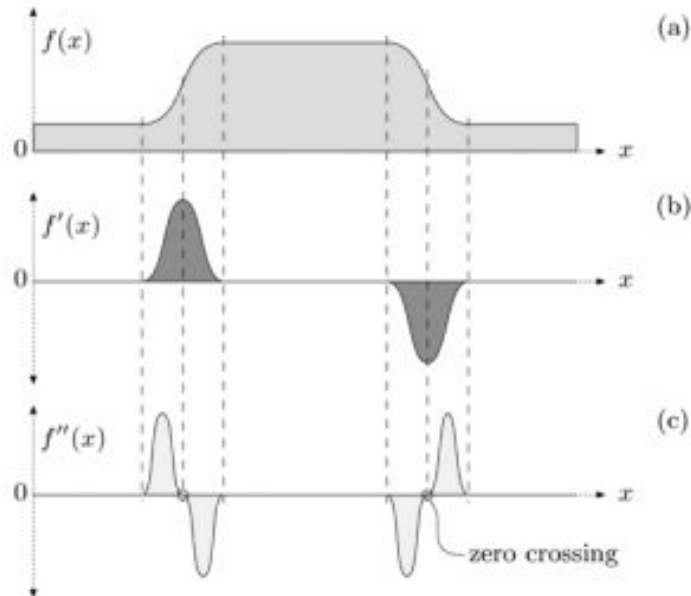
1

1	1	1
1	-8	1
1	1	1

1

-1	2	-1
2	-4	2
-1	2	-1

1



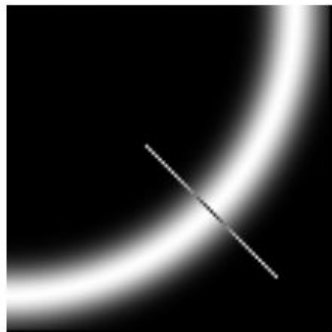
Визначення меж на основі градієнта

- Обчислити похідні зображення (із згладжуванням) за згорткою

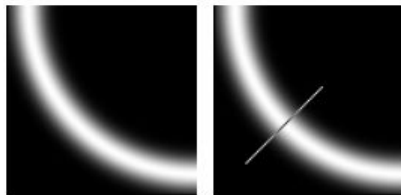
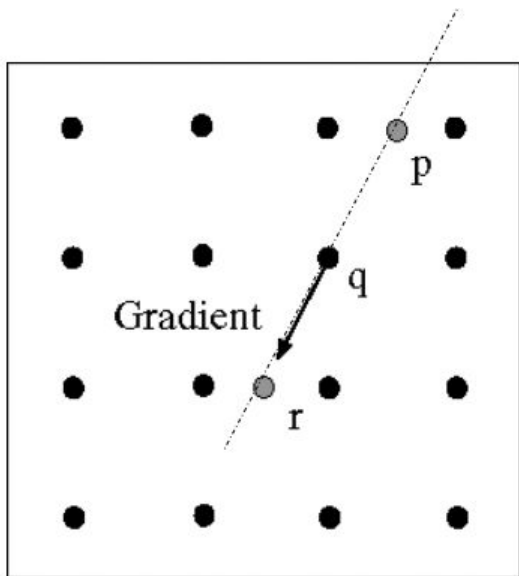
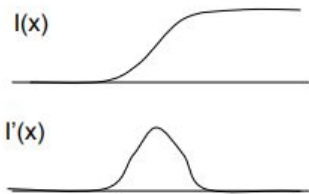
$$D_x(u, v) = m_x * I \quad \text{and} \quad D_y(u, v) = m_y * I$$

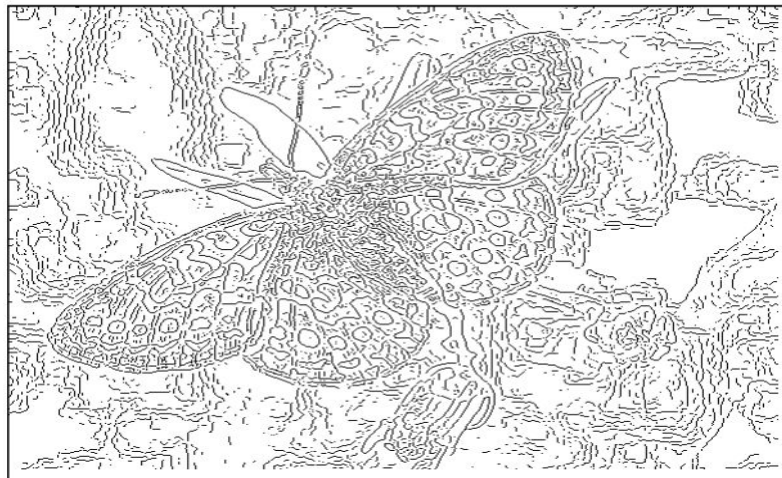
- Обчислити силу межі - величину
градієнта $E(u, v) = \sqrt{(D_x(u, v))^2 + (D_y(u, v))^2}$
- Обчислити орієнтацію межі - напрямок
градієнта

$$\Phi(u, v) = \tan^{-1} \left(\frac{D_y(u, v)}{D_x(u, v)} \right) = \text{ArcTan}(D_x(u, v), D_y(u, v))$$



non-maximal suppression





Метод Кенні

- Метод Кенні переслідує три основні цілі:
 1. **Низька частота помилок.** Повинні виявлятися всі контури з мінімумом помилкових спрацьовувань.
 2. **Хороша локалізація контурних точок.**
 3. **Одиночний відгук на точку контуру.** Для кожної точки істинного контуру детектор повинен виявляти тільки одну точку.



(a) Original



(b) Canny,
 $\sigma = 1$



(c) Canny,
 $\sigma = 3$

Метод Кенні

1. Згладжування зображення за допомогою гаусової фільтрації

$$h(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right),$$

де σ - параметр, що визначає ступінь згладжуваної на присутній в зображенні шум.

2. Знаходження градієнта зміни яскравості в зображенні:

3. Гуртова обробка результатів обчислення градієнтів в кожній точці зображення.

Метод Кенні

- Вибирається кілька дискретних напрямків нормалі до контуру (тобто напрямків вектора градієнта).



- Можна сформулювати наступну схему пригнічення немаксимальних точок для околиці з центром в кожній точці (x, y) зображення $\alpha(x, y)$ і формування зображення $g_N(x, y)$ з витонченими контурами:
 - Знаходимо напрямок dk , найближчий до $\alpha(x, y)$.
 - Якщо значення $G(x, y)$ менше, ніж хоча б у одного з сусідів в напрямку dk , приймаємо $g_N(x, y) = 0$ (пригнічення); в іншому випадку $g_N(x, y) = G(x, y)$. Тут $g_N(x, y)$ - зображення з пригніченими немаксимальними точками.

Метод Кенні

4. Морфологічна обробка результатів, отриманих на попередньому кроці алгоритму.

