



# 1. Свертка сигналов

- Обычно сигнал представляется функцией от времени, то есть аргументом является вещественная переменная  $t$ .
- Пусть на вход некоторого оператора, преобразующего сигнал подается функция  $x(t)$ , а выходе получаем новый сигнал  $y(t)$ . Мы рассматриваем поведение объекта на некотором интервале времени, то есть параметры представляют собой функции от времени на этом интервале.
- Выходной параметр  $y(t)$  зависит от входного параметра, то есть от функции  $x(t)$ , эта зависимость задается оператором преобразования.
- Мы рассматриваем поведение объекта на некотором интервале времени, то есть параметры представляют собой функции от времени на этом интервале.

# 1. Свертка сигналов

- Преобразование будем записывать в виде
  - $y(t) = F[x(t)]$ ,
- $F$  переводит функции  $x(t)$  в функцию  $y(t)$ .
- Преобразование  $F$  называется **оператором**, в общем случае это не функция, а более сложный объект. Например,  $F$  может выражать зависимость в виде решения дифференциального уравнения.
- Одним из видов зависимости функций является свертка

$$y(t) = \int_0^{+\infty} h(\tau)x(t-\tau)d\tau$$

Функция  $h(t)$  называется ядром свертки. Значение сигнала  $x(t)$  в точке  $t$  свертка преобразует в значение нового сигнала  $y(t)$  в той же точке  $t$ .

Свертка широко применяется в теории сигналов, в частности, для моделирования фильтров.

# 1. Свертка сигналов

• В реальной ситуации ядро обычно не равно нулю только на некотором отрезке  $[0, M]$ , поэтому свертка принимает вид

$$y(t) = \int_0^M h(\tau)x(t - \tau)d\tau$$

Если бы нижняя граница интервала интегрирования была бы меньше 0, например, -1, то получалось бы, что функция  $y(t)$  зависит от значения функции  $x(0 - (-1))$  в момент времени от +1, то есть в будущем, что считаем невозможным.



# 1. Свертка сигналов

• Следовательно, при  $t < 0$  ядро  $h(t) = 0$ .

$$y(t) = \int_0^M h(\tau)x(t-\tau)d\tau \quad (1)$$

Для каждого момента времени  $t_0$  функция  $y(t_0)$  зависит от функции  $x(t)$  во все моменты времени от  $t_0 - M$  до  $t_0$ , то есть, от «недалекого прошлого».

$M$  называется интервалом памяти объекта.

Сокращенно соотношение (1) записывается в виде

$$y(t) = h(t)*x(t).$$

# 1. Свертка сигналов

• В общем виде математический оператор свертки действует на интервале  $t \in (-\infty, +\infty)$

$$y(t) = \int_{-\infty}^{+\infty} h(\tau)x(t - \tau)d\tau,$$

В этой записи свертка коммутативна, то есть

$$y(t) = h(t)*x(t) = x(t)*h(t).$$

В запись через интеграл это означает, что

$$y(t) = \int_{-\infty}^{+\infty} h(\tau)x(t - \tau)d\tau = \int_{-\infty}^{+\infty} x(\tau)h(t - \tau)d\tau,$$

Формула легко доказывается заменой переменных

# 1. Свертка сигналов

- Изображение тоже можно рассматривать как двумерный сигнал. Определим оператор свертки для изображения, рассмотрим простейший случай, когда ядро свертки имеет размерность  $3 \times 3$ .
- Ядро свертки (называют также маска), скользит по пикселям изображения, заменяя интенсивность текущего пикселя на новое значение, которое получается как сумма произведений интенсивностей пикселей, входящих в в скользящее окно размера  $3 \times 3$  и чисел, входящих в маску маски.
- Если  $(i, j)$  - центральный пиксель текущего окна с интенсивностью  $f_{i,j}$ , то новое значение интенсивности этого пикселя будет изменено в зависимости от пикселей, входящих в окно преобразования и чисел, входящих в маску.

# 1. Свертка сигналов

• Если маска свертки равна

$$M = \begin{array}{|c|c|c|} \hline m_{-1,-1} & m_{-1,0} & m_{-1,1} \\ \hline m_{0,-1} & m_{0,0} & m_{0,1} \\ \hline m_{1,-1} & m_{1,0} & m_{1,1} \\ \hline \end{array}$$

• то новое значение  $g_{i,j}$  интенсивности пикселя  $(i, j)$  будет результатом воздействия маски на окрестность этого пикселя.

$$g_{i,j} = \sum_{p=-1}^{+1} \sum_{q=-1}^{+1} m_{p,q} \cdot f_{i+p,j+q}$$

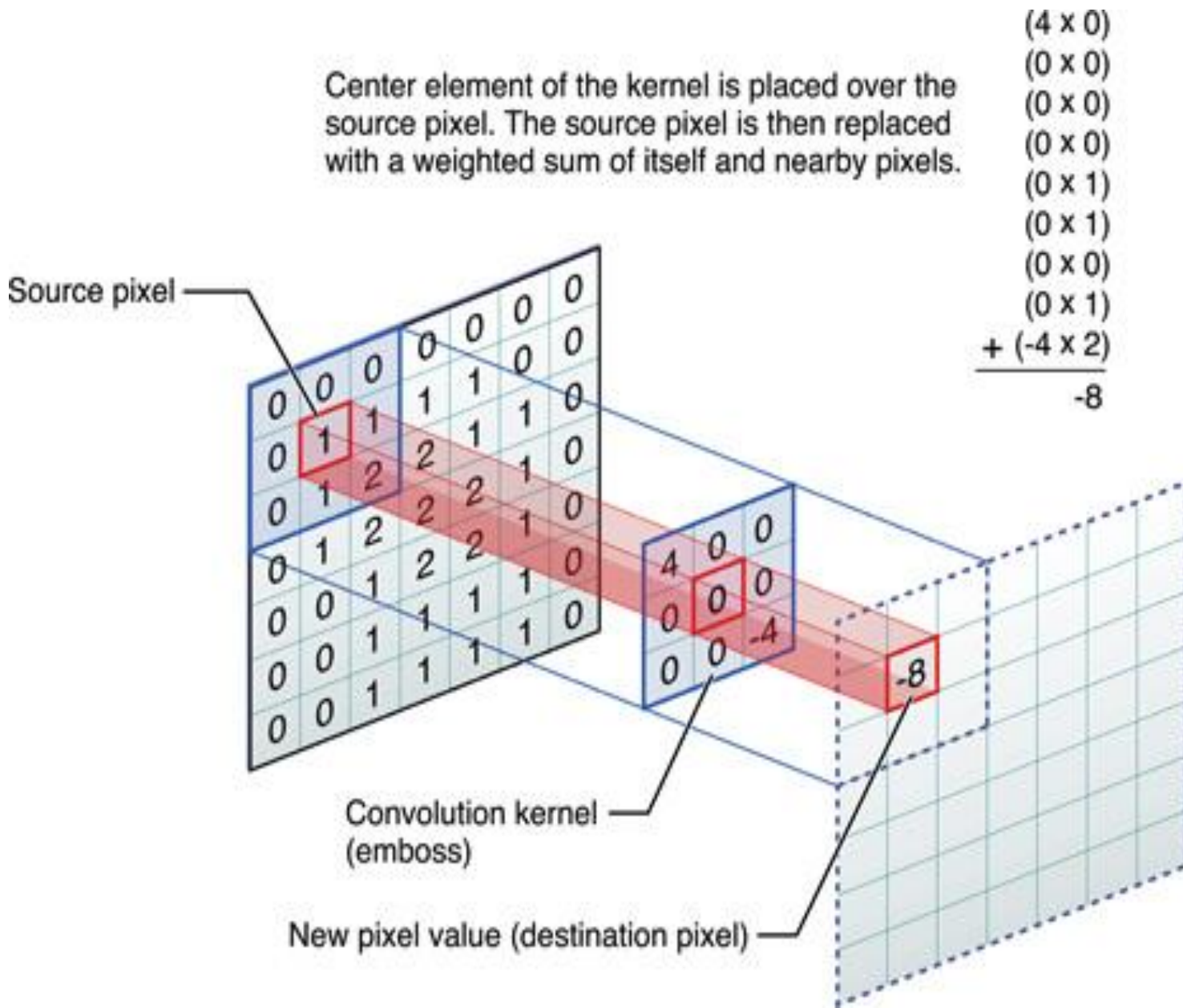
• Индексы  $i, j$  пробегают все значения, соответствующие размеру исходного изображения, за исключением граничных полос (в случае маски  $3 \times 3$  - за исключением первой и последней строки, а также первого и последнего столбца матрицы исходного изображения).



# 1. Свертка сигналов

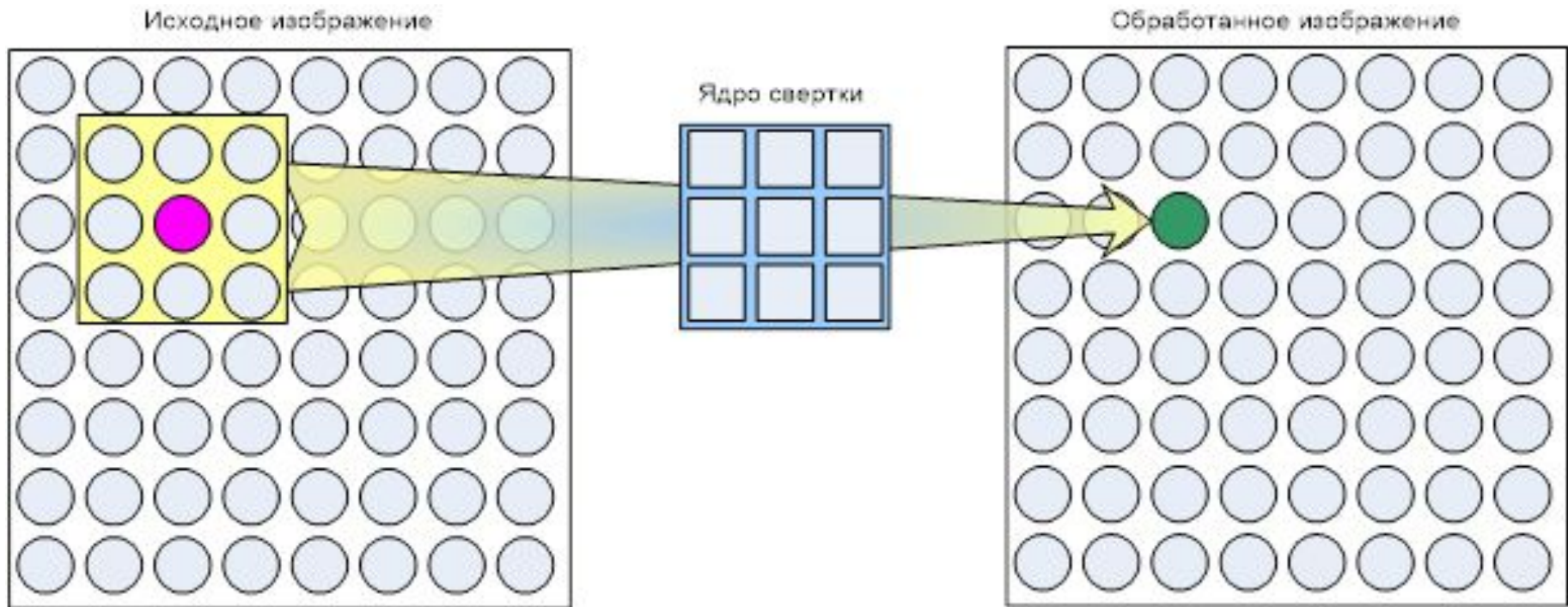
## •Свертка изображения с маской 3x3

Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.



# 1. Свертка сигналов

• Оператор свертки с маской 3x3 пробегает по всем пикселям исходного изображения, кроме граничных, изменяя яркость пикселя (бардовый на Рис) на новую яркость. Изменение яркости выполняется суммой произведений элементов ядра на яркости пикселей изображения, попадающие в действие ядра свертки.



# 1. Свертка сигналов

• Оператор свертки с маской  $3 \times 3$  пробегает по всем пикселям исходного изображения, кроме граничных, изменяя яркость пикселя (бардовый на Рис) на новую яркость. Изменение яркости выполняется суммой произведений элементов ядра на яркости пикселей изображения, попадающие в действие ядра свертки.

# 1. Свертка сигналов

- Если маской 3x3 имеет вид

1/9

1	1	1
1	1	1
1	1	1

- То фильтр сглаживает яркости, присваивая пикселю новую яркость, равную среднему от значений яркости 9 пикселей, с пикселем-объектом преобразования в центре.

- Маска

1/16

1	2	1
2	4	2
1	2	1

- Этот фильтр добавляет к яркости пикселя-объекта взвешенные яркости соседей, с большим вкладом по «кресту» (вес 2), по диагонали с меньшим (вес 1).

# 1. Свертка сигналов

- Горизонтальный фильтр Собела выделяет горизонтальные грани на изображении

-1	-2	-1
0	0	0
1	2	1

- Например,

Исходное изображение

50	50	50	50	50	50
50	50	50	50	50	50
50	50	50	50	50	50
70	70	70	70	70	70
70	70	70	70	70	70
70	70	70	70	70	70

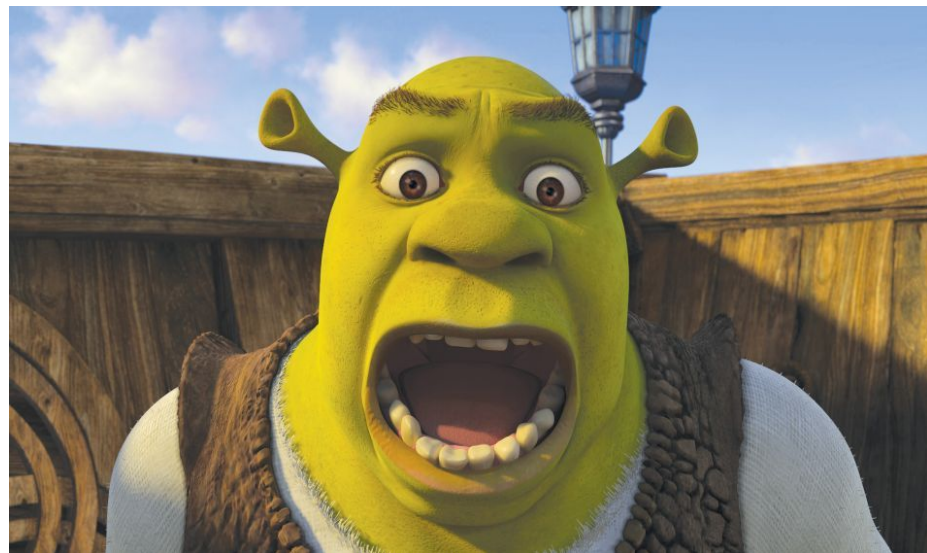
После свертки появилась светлая полоса (яркость 80) .

50	50	50	50	50	50
50	0	0	0	0	50
50	<b>80</b>	<b>80</b>	<b>80</b>	<b>80</b>	50
70	<b>80</b>	<b>80</b>	<b>80</b>	<b>80</b>	70
70	0	0	0	0	70
70	70	70	70	70	70

# 1. Свертка сигналов

- Вертикальный фильтр Собела выделяет вертикальные грани на изображении

-1	0	1
-2	0	2
-1	0	1





# 1. Свертка сигналов

• Горизонтальный фильтр Превитта аналогично фильтру Собеля выделяет горизонтальные грани на изображении

-1	-1	-1
0	0	0
1	1	1

• Вертикальный фильтр Превитта выделяет вертикальные грани на изображении

-1	0	1
-1	0	1
-1	0	1

• Фильтры Превитта проще, но фильтры Собеля снижают влияние соседних пикселей.

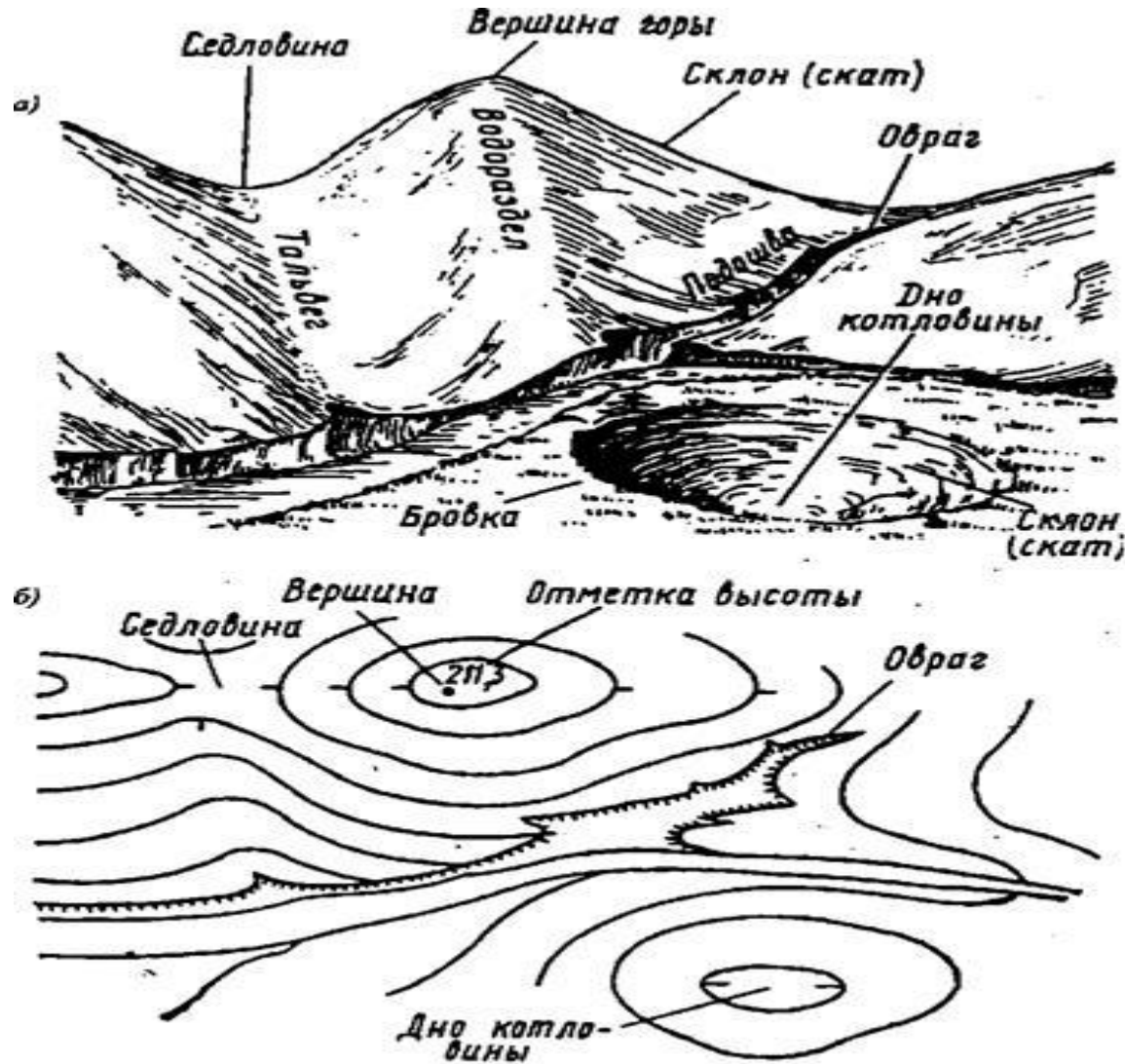
# 1. Свертка сигналов

• Если для каждого пикселя  $(i, j)$  применением вертикального фильтра вычислить соответствующее ему значение результата применения фильтра  $x_{ij}$ ; и применяя горизонтальный фильтр найти  $y_{ij}$ , то вектор  $(x_{ij}, y_{ij})$  будет равен **градиенту поверхности**, которую определяет исходное изображение.

• Градиент  $g_{ij} = (x_{ij}, y_{ij})$  указывает направление самого крутого подъема из точки  $(i, j)$ . Противоположное направление - **антиградиент** указывает направление, по которому потечет вода из точки  $(i, j)$ .

# 1. Свертка сигналов

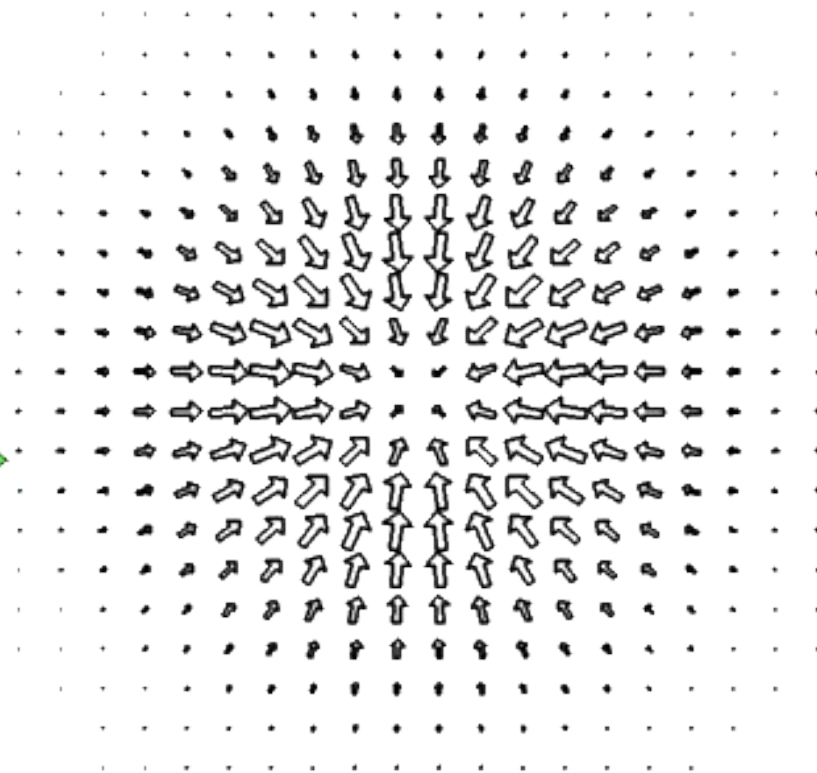
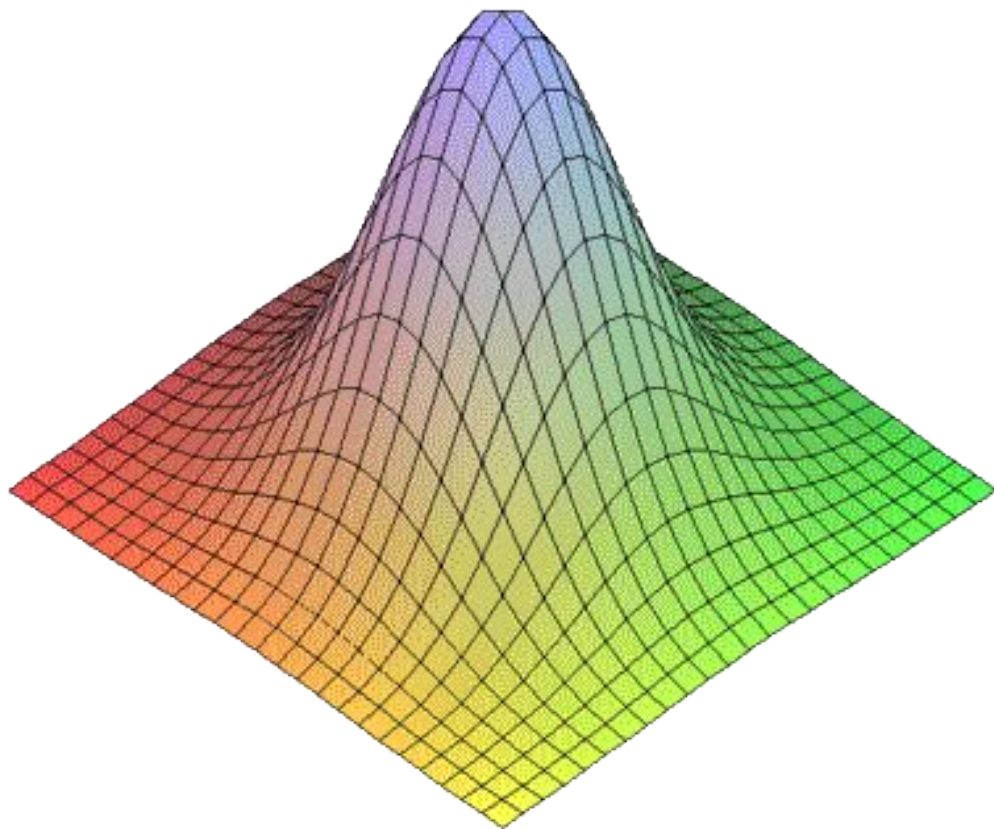
- Градиент и антиградиент на местности



Р и с . 2.7. Элементы рельефа (а) и их изображение на карте (б)

# 1. Свертка сигналов

• Если для каждого пикселя изображения построить градиент (это будет двуслойная матрица, один слой - горизонтальная координата градиента, второй - вертикальная), то будет получено поле градиентов изображения.



# 1. Свертка сигналов

- Точки, в которых **градиент близок к нулю**, соответствуют на поверхности либо вершине, либо яме, либо математической точке перегиба. Во всех трех случаях это будут точки, подозрительные на принадлежность к границе некоторого объекта на изображении.

- 

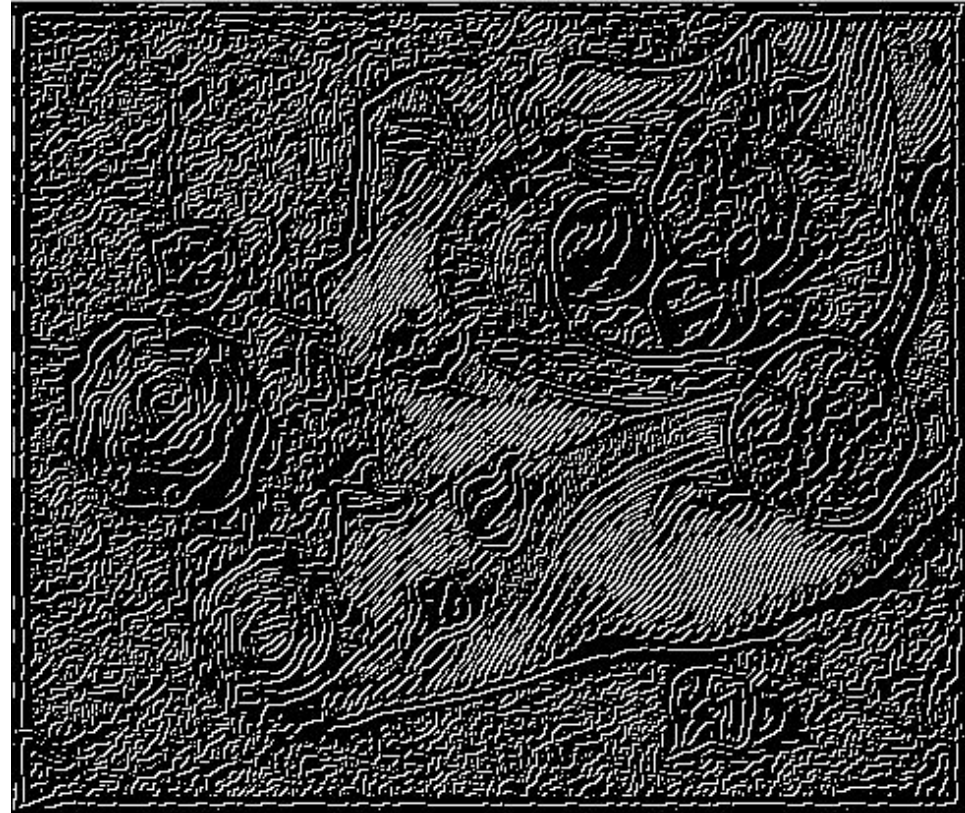
- **Алгоритм Канни (Canny - 1983г)** остается на сегодняшний день **лучшим алгоритмом выделения границ и граней** объектов на изображении. Этот алгоритм использует градиенты, построенные алгоритмом Собеля.

- (это будет двуслойная матрица, один слой - горизонтальная координата градиента, второй - вертикальная), то будет получено поле градиентов изображения.



# 1. Свертка сигналов

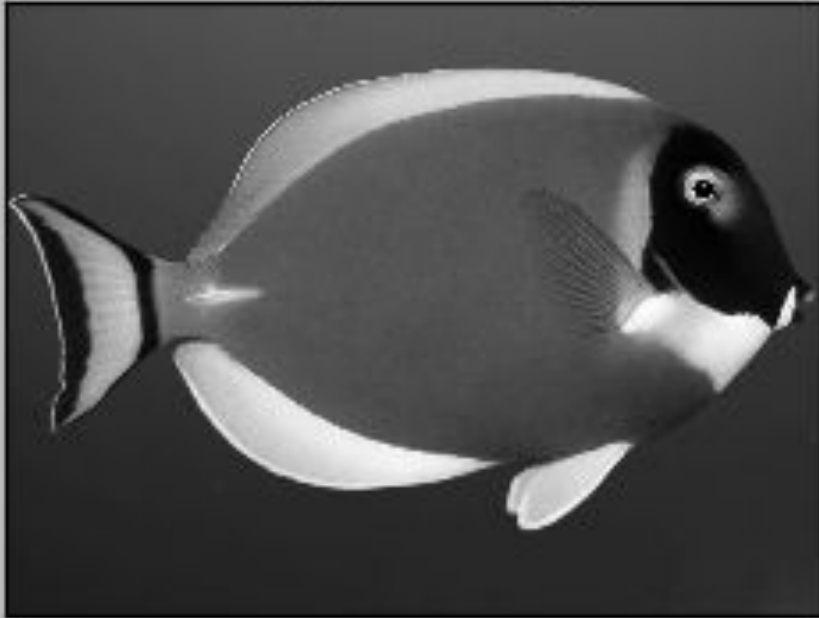
- Построение границ методом Канни



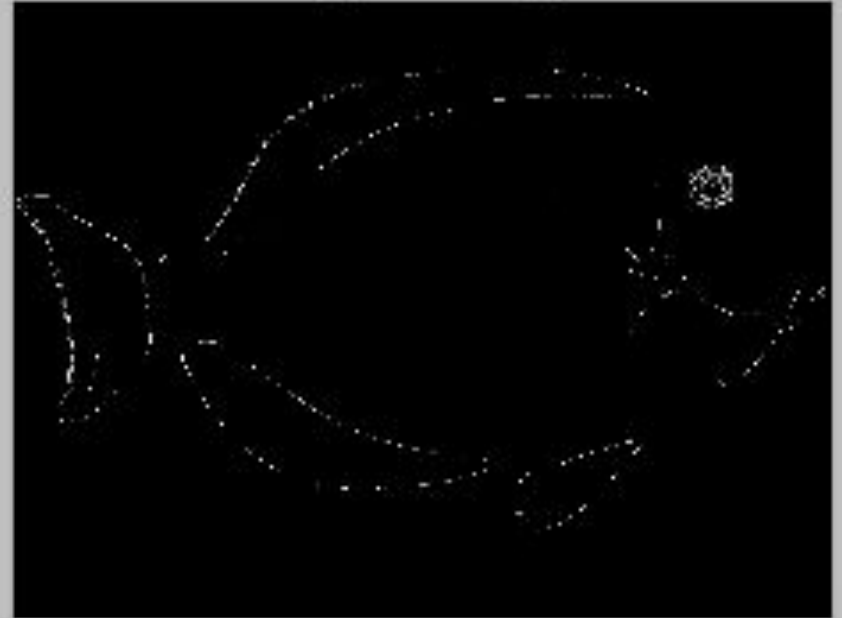


# 1. Свертка сигналов

Original Image



Edge Image



Original Image



Edge Image



# 1. Свертка сигналов

- Фильтры Собеля и Превитта называются дифференцирующими, они, как и производная, дают нулевое значение на плоской поверхности.
- Эти фильтры для дискретных изображений аналогичны производным по  $x$  (вертикальный фильтр) и по  $y$  (горизонтальный).
- Фильтр Робертса аналогичен частным производным  $\frac{\partial f(x, y)}{\partial x}$ ,  $\frac{\partial f(x, y)}{\partial y}$
- Маски фильтров Робертса :

-1	0
0	1

0	-1
1	0

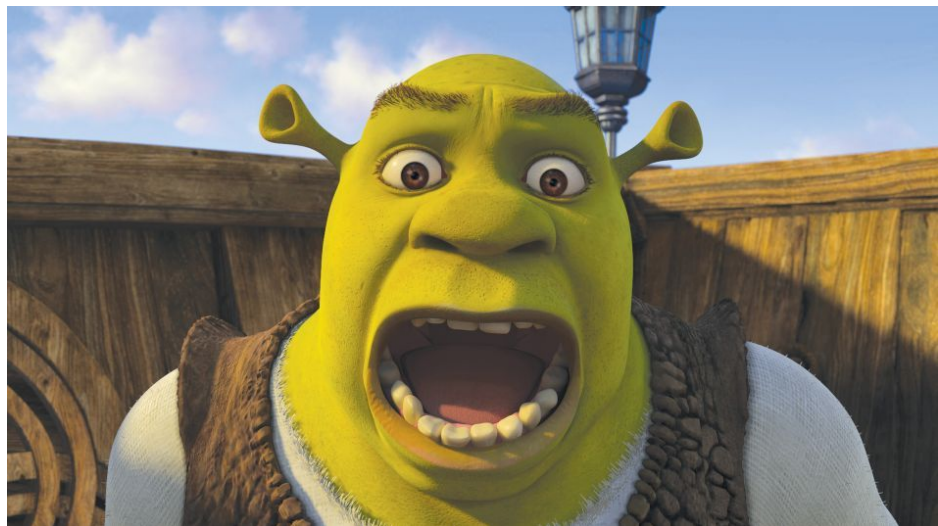
- Фильтр Робертса был предназначен для наклонных граней.

# 1. Свертка сигналов

• Маски фильтров Робертса неудобны для практики, их заменили на симметричные относительно изменяемого пикселя

-1	-1	0
-1	0	1
0	1	1

0	-1	-1
1	0	-1
1	1	0



• Маски Робертса дают тени изображения.

# 1. Свертка сигналов

- Маски Робертса можно изменить по аналогии с масками Собеля, если поставить на ненулевых диагональных элементах в углах числа 2, -2.
- В обработке изображений используется лапласиан яркости пикселя, это сумма вторых частных производных функции  $f(x, y)$

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- Производные второго порядка, то есть и лапласиан, очень чувствительны к шуму.
- В обработке изображений используется лапласиан яркости пикселя, это сумма вторых частных производных функции яркости.

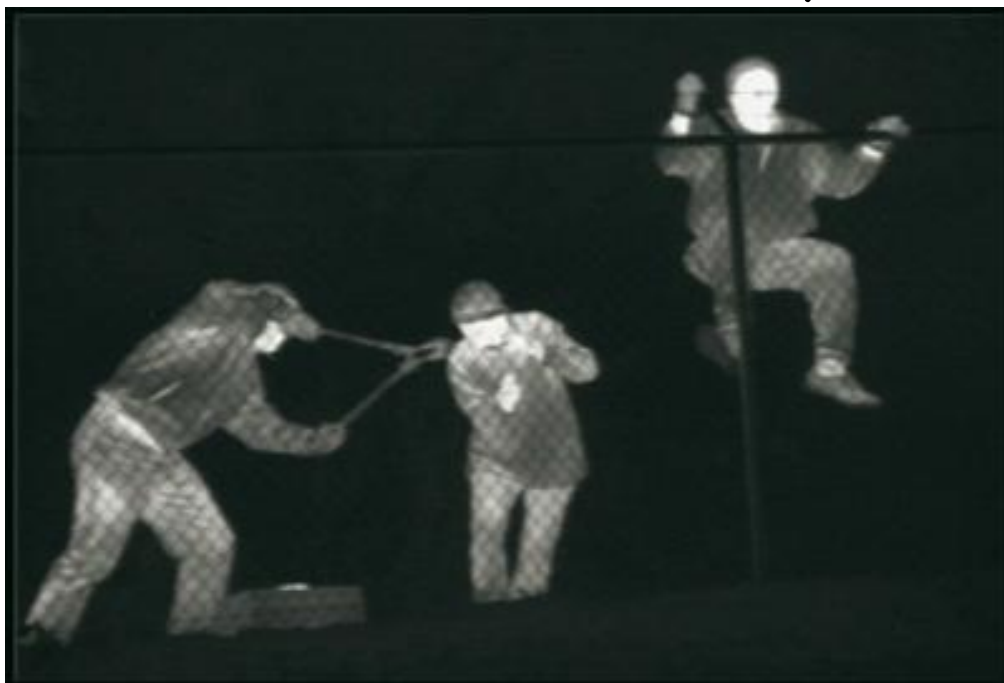
## 2. Сегментация изображений

- **Сегментация** подразделяет изображение на составляющие его области или объекты. Та степень детализации, до которой доводится такое разделение, зависит от решаемой задачи.
- Иначе говоря, сегментация прекращается, когда интересные объекты оказываются на картинке изолированными, то есть выделенными.
- Например, в задаче автоматизированного визуального контроля сборки узлов радиоэлектронной аппаратуры интерес представляет анализ фотографий изготавливаемых изделий с целью выявления определенных дефектов, таких как отсутствие компонентов или разрыв контактных дорожек на плате. Поэтому не имеет смысла проводить сегментацию мельче того уровня детализации<sup>25</sup>, который

## 2. Сегментация изображений

• В этом случае следует выбрать фильтры такого вида, которые будут усиливать сигнал от интересующих объектов и одновременно ослаблять влияние несущественных деталей изображения.

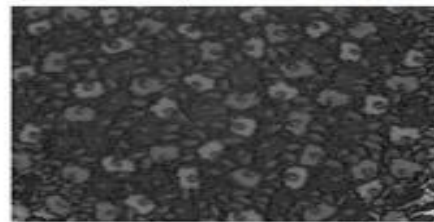
• Хорошим примером такого подхода может служить съемка в инфракрасном диапазоне, используемая для обнаружения объектов с мощным тепловым излучением, например, боевой техни





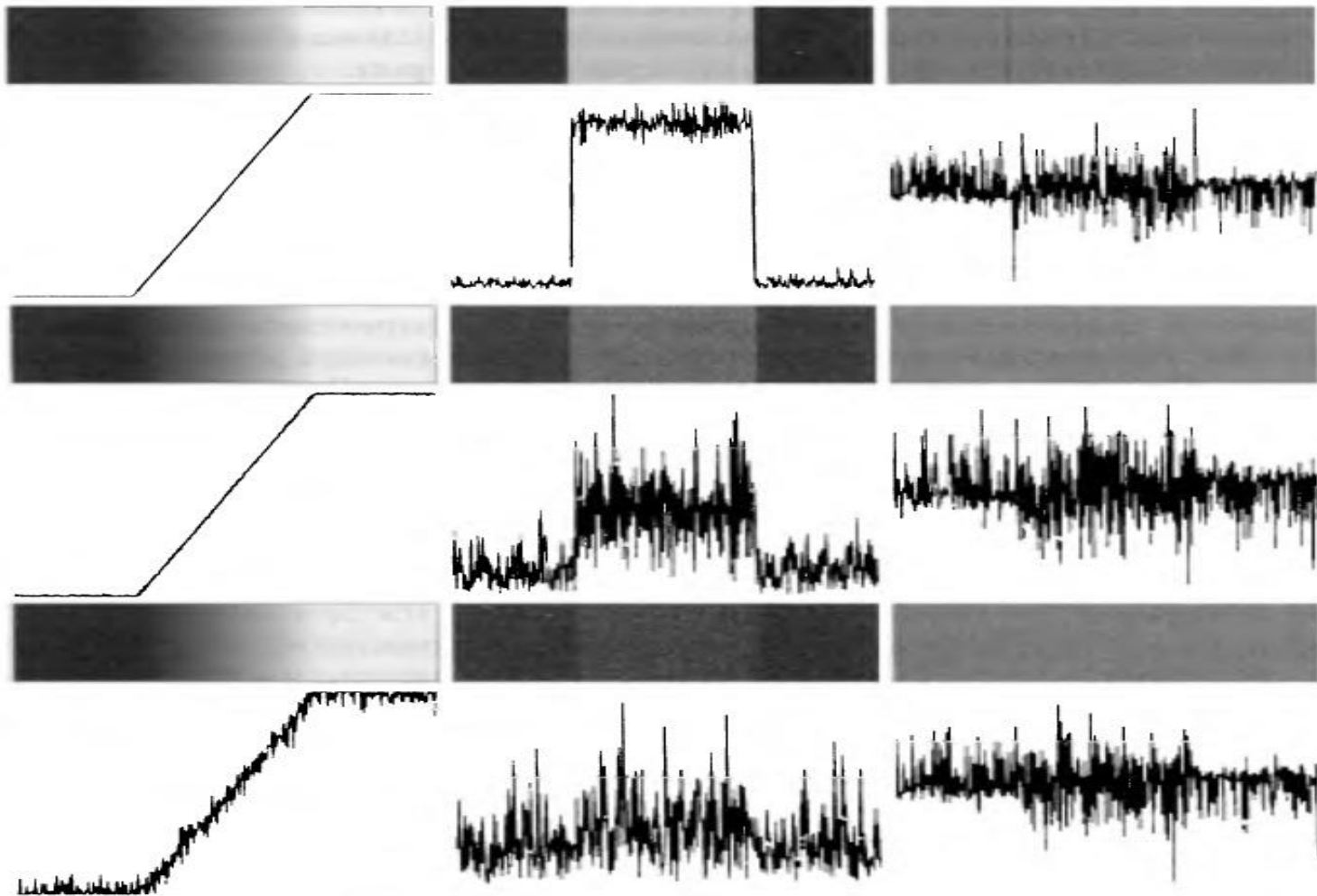
## 2. Сегментация изображений

• Как правило, алгоритмы сегментации изображений основываются на одном из двух базовых свойств сигнала яркости: **разрывности** и **однородности**. В первом случае подход состоит в разбиении изображения на основании **резких изменений** сигнала, таких как перепады яркости на изображении. Вторая категория методов использует разбиение изображения на области, **однородные** в смысле заранее выбранных критериев. Примерами таких методов могут служить **пороговая обработка** и **наращивание областей**,



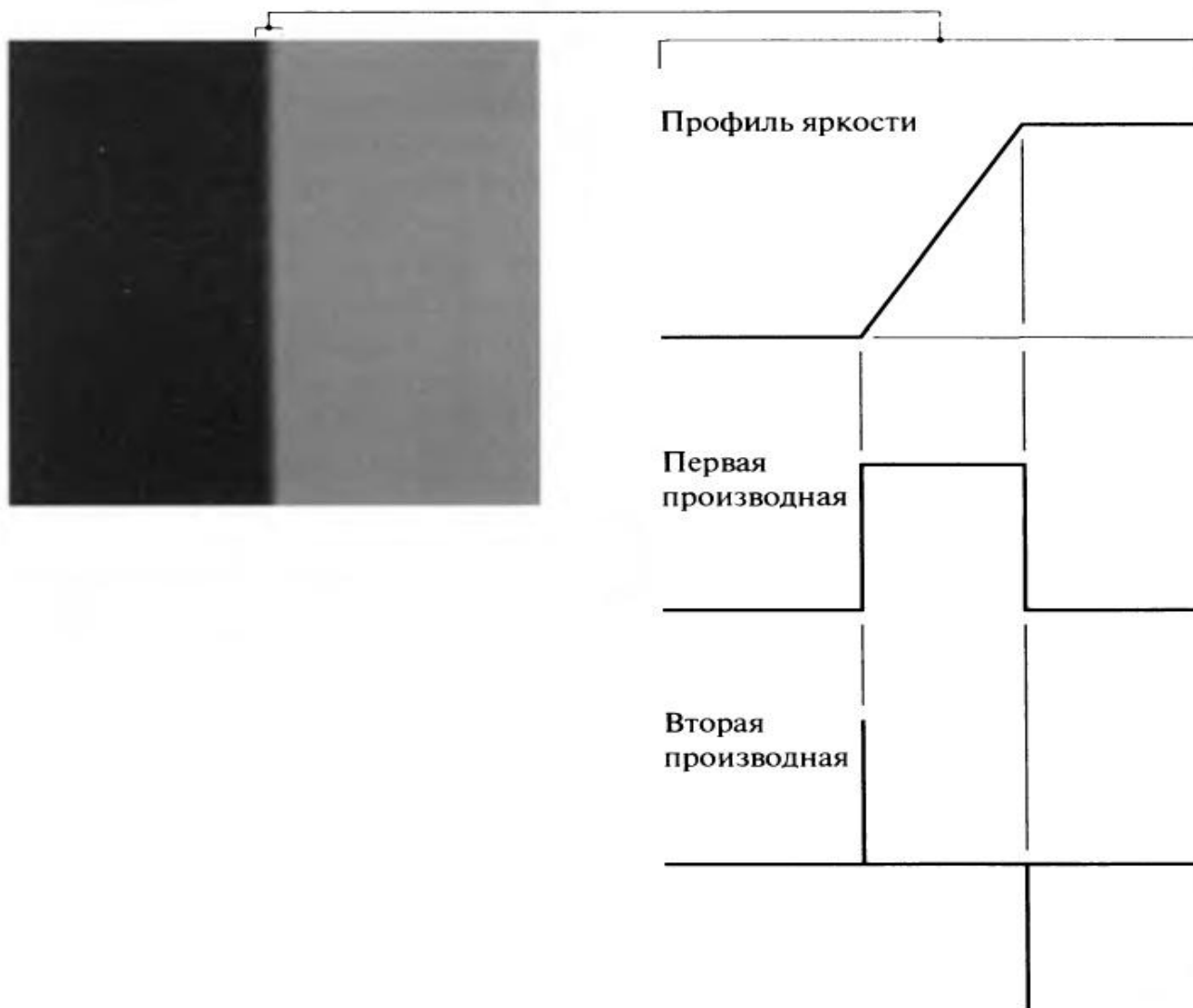
## 2. Сегментация изображений

- Производный 1-го и 2-го порядков для изображения с шумами.
- 1-я строка - шум небольшой, 2-я средний, 3-я сильный.
- 1-й столб исходн изобр 2-й первая произв 3-й вторая пр.



## 2. Сегментация изображений

• Перепады яркости с гладким переходом. Вторая производная изменяет значение от положительного к отрицательному



## 2. Сегментация изображений

• Рисунки показывают, что в зашумленном изображении вторую производную трудно использовать для выделения границ. Но если предварительно удалить шумы размытием рисунка (напр, фильтром Гаусса - второй рисунок). Тогда видно, что **изменение знака второй производной** можно использовать для того, чтобы определить, какой переход выполняется на границе - от темного к светлому или наоборот, от светлого к темному.

• На рисунке видно, что если переход идет от темных пикселей к светлым, то **вторая производная меняет знак с плюса на минус**. И наоборот, при переходе от светлых пикселей к темным вторая производная меняет знак от минуса к плюсу.

## 2. Сегментация изображений

- Лапласиан (сумма квадратов вторых частных производных) реализуется одной из двух матриц

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

- Лапласиан можно применять в сочетании со сглаживанием для обнаружения перехода между светлым и темным фрагментом (в точках нулевого уровня).
- Для этого используется фильтр Гаусса, лапласиан вычисляется одновременно с размытием картинки.
- Матрицы получены дискретизацией мексиканской шляпы.
- Маска такого лапласиана-гауссиана имеет вид

## 2. Сегментация изображений

- Более точная маска лапласиана-гауссиана имеет вид

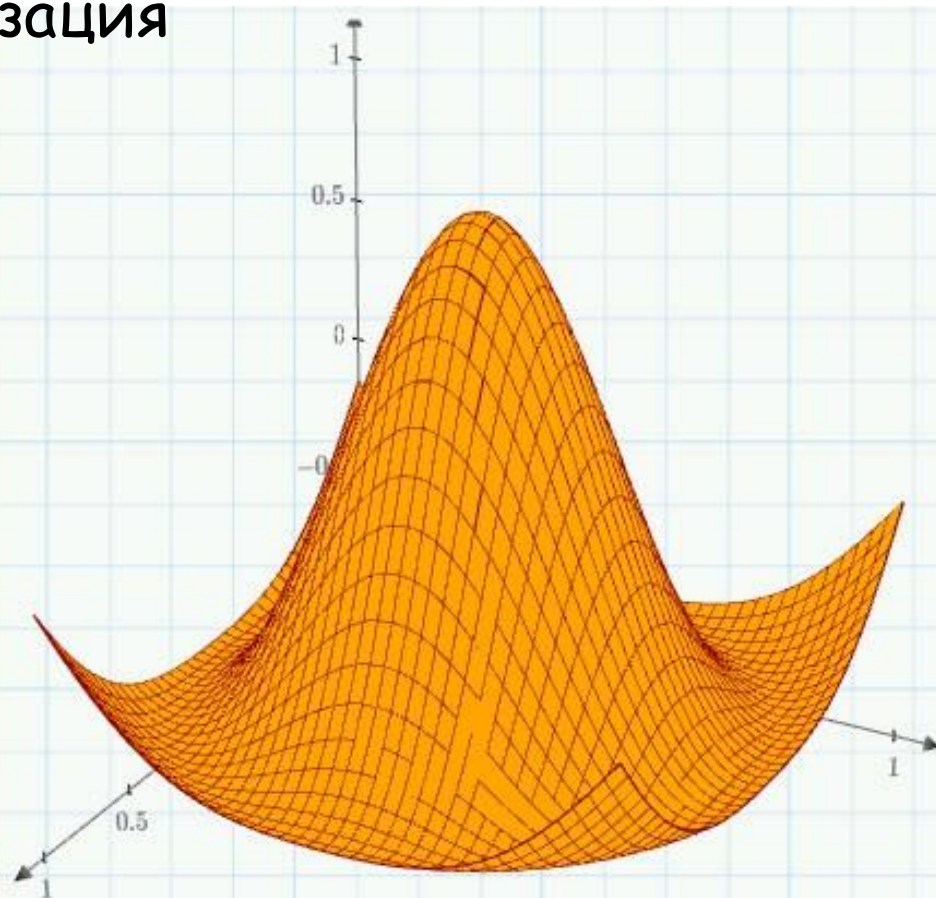
0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

- График непрерывной функции лапласиана-гауссиана имеет характерный вид и называется **мексиканская шляпа**.
- Применение такого фильтра эквивалентно размыванию изображения гауссианом (для уменьшения шума) с последующим вычислением лапласиана.
- Точки, подозрительные на принадлежность к контурам находятся между положительными и отрицательными значениями результата фильтрации



## 2. Сегментация изображений

- «Мексиканская шляпа» - лапласиан-гауссиан и его дискретизация



$Z_{i,j}$

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

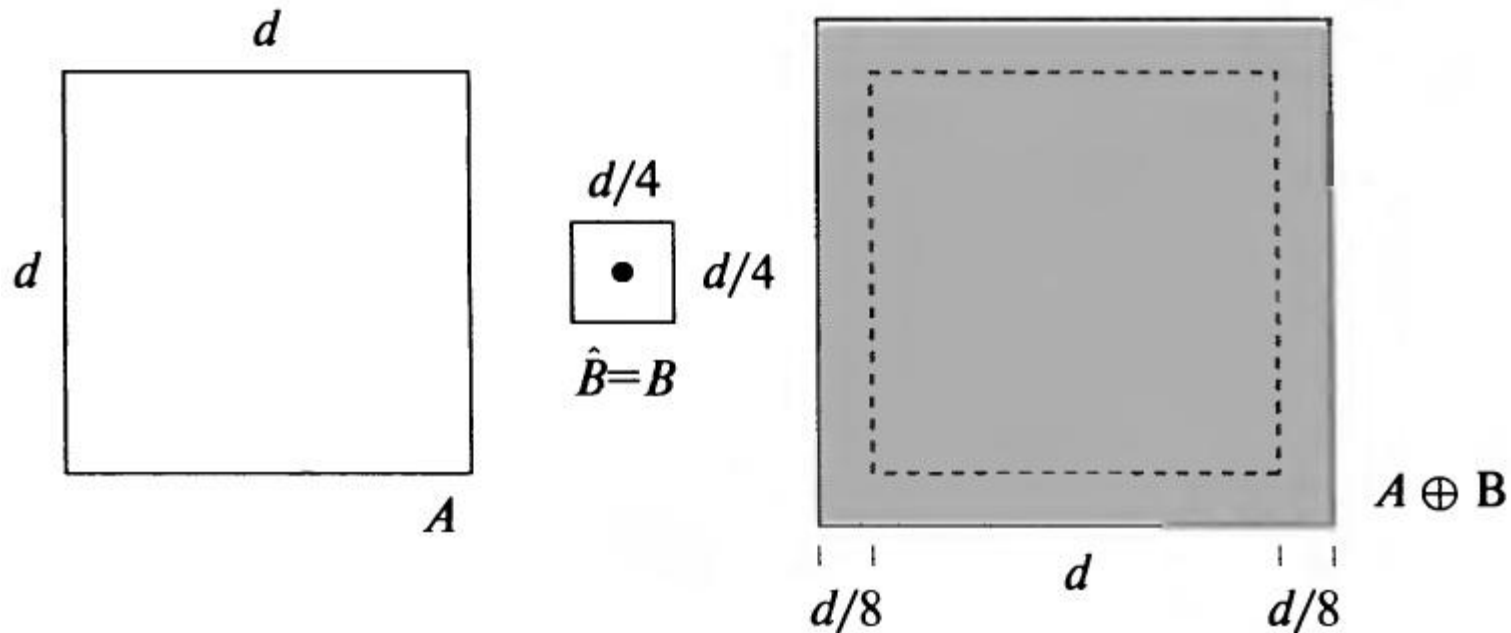
## 2. Сегментация изображений

- Применение лапласиана-гауссиана



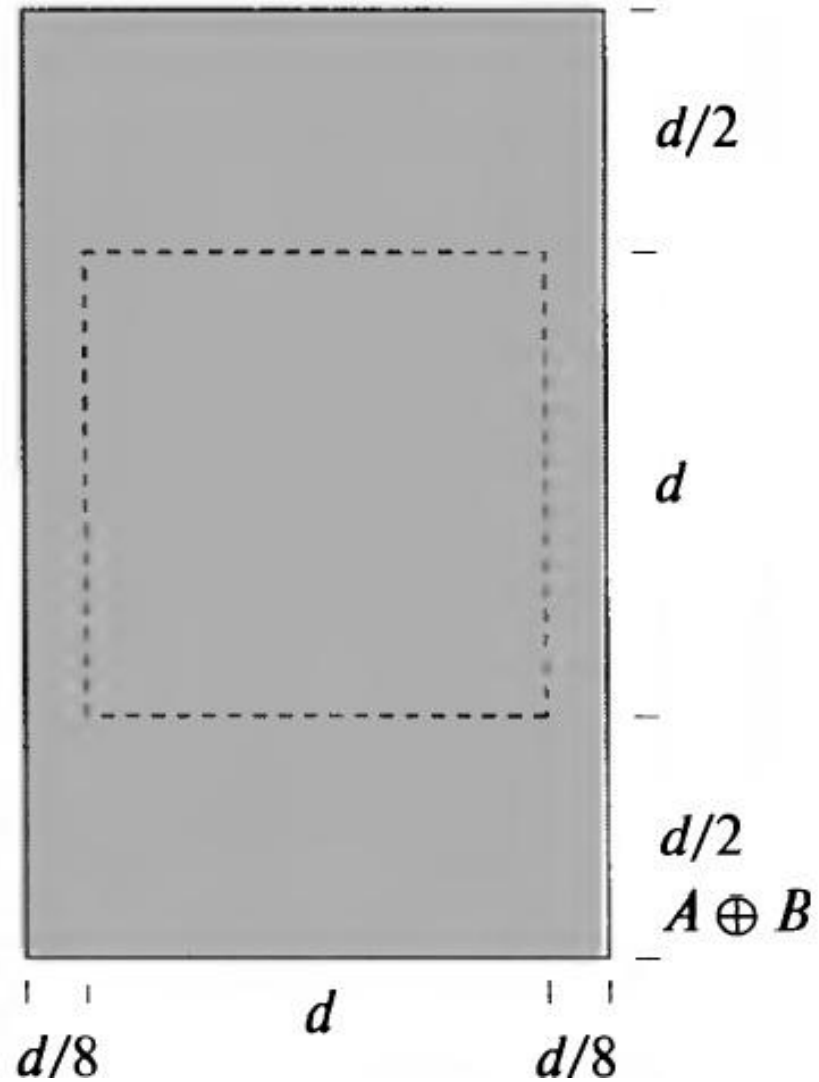
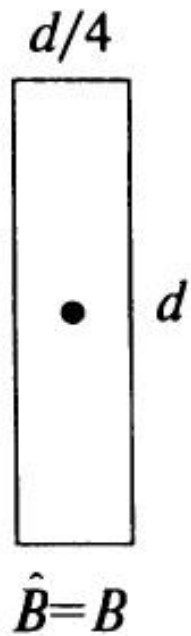
## 2. Сегментация изображений

- **Морфологические операции.** В биологии словом **морфология** называют область, которая изучает форму и строение животных и растений. В обработке изображений **математической морфологией** называют методы для извлечения компонент изображения, полезные для его представления и описания, например, границы, выпуклые оболочки.
- Операция **дилатации** (расширение).
- Пусть  $B$  — множество объектов с центральной симметрией



## 2. Сегментация изображений

- Пример операции дилатации.

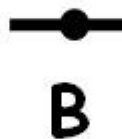


## 2. Сегментация изображений

- Дилатация применяется для удаления разрывов.
- Пример. Замыкание контура  $A$ .



$A$



$B$

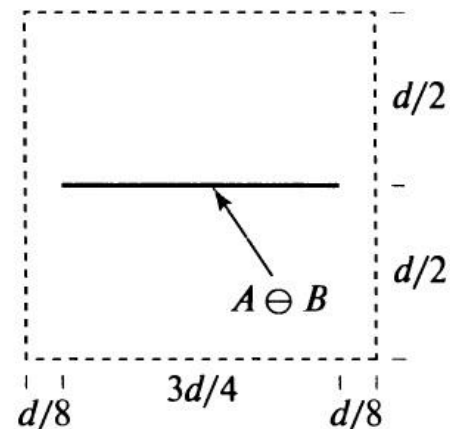
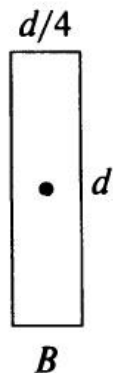
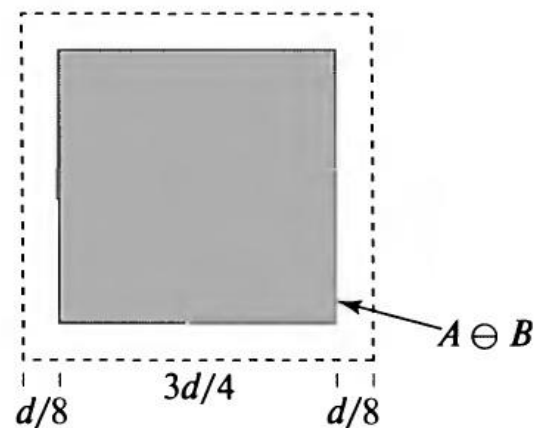
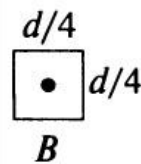
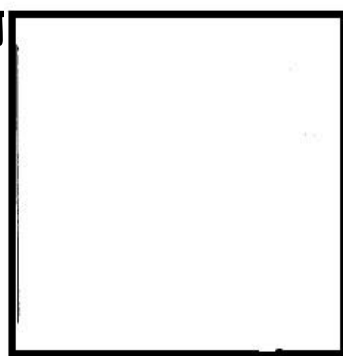


$A^*$

- В MatLab:
- $D = \text{imdilate}(S, se);$

## 2. Сегментация изображений

- Эрозией множества  $A$  по примитиву  $B$  — это множество
- всех таких точек центра  $B$  при сдвиге в которые множество  $B$  целиком содержится в  $A$ .
- Эрозия выделяет внутренность объекта.
- Пример. Эрозия контура

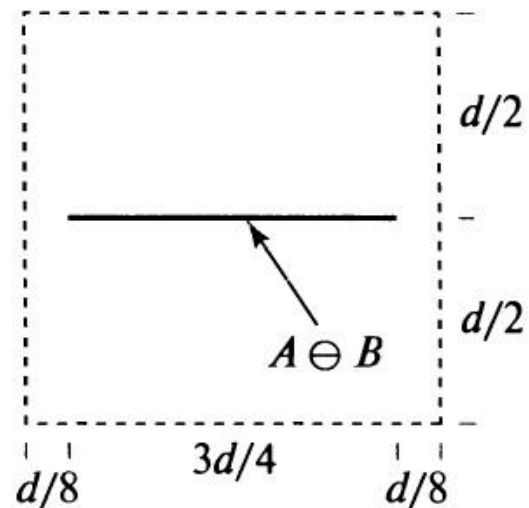
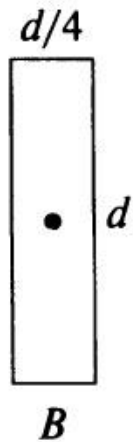
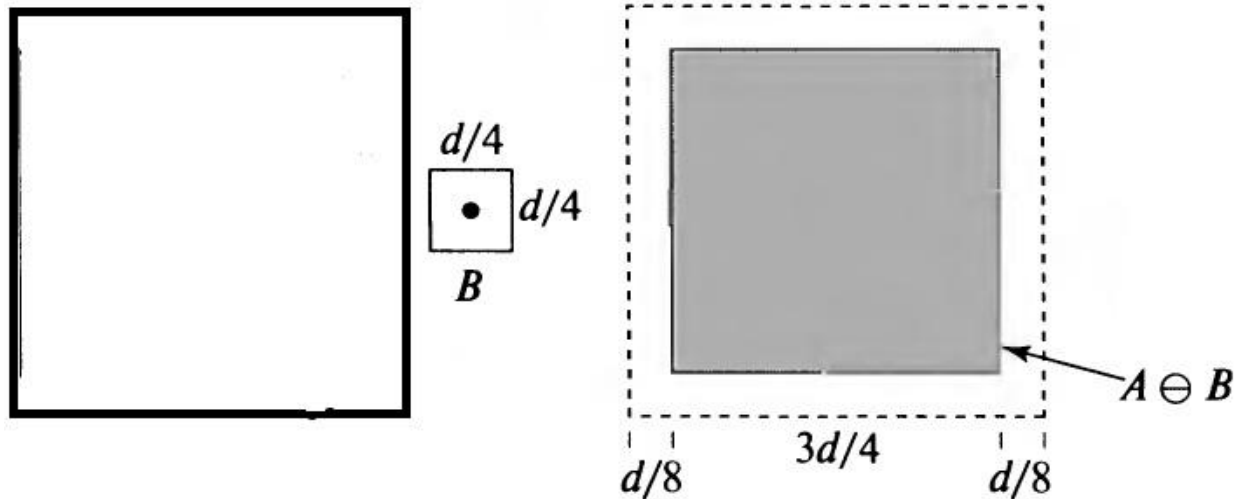




## 2. Сегментация изображений

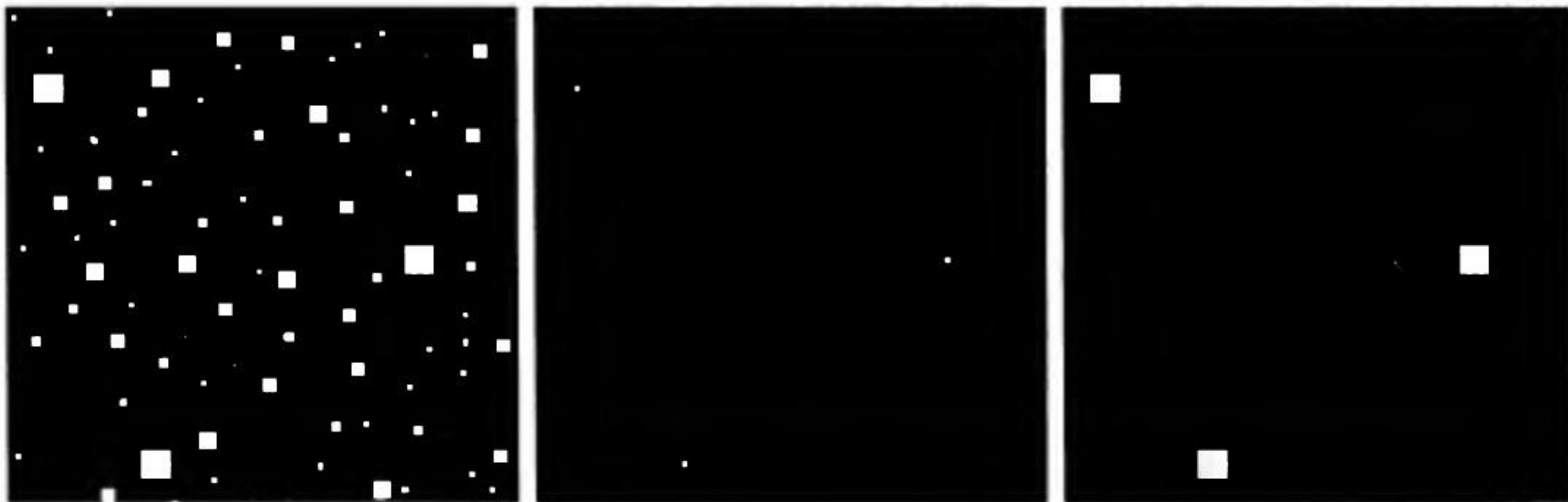
• Эрозия выделяет внутренность объекта.

• Пример. Эрозия контура  $A$ .



## 2. Сегментация изображений

- **Пример.** Удаление мелких деталей. Вначале применяем эрозию с примитивом, чуть меньшим, чем квадраты, которые нужно оставить (Идем по изображению А черного цвета).
- Затем применяет дилатацию и восстанавливаем нужные квадраты (Идем по изображению А белого цвета).



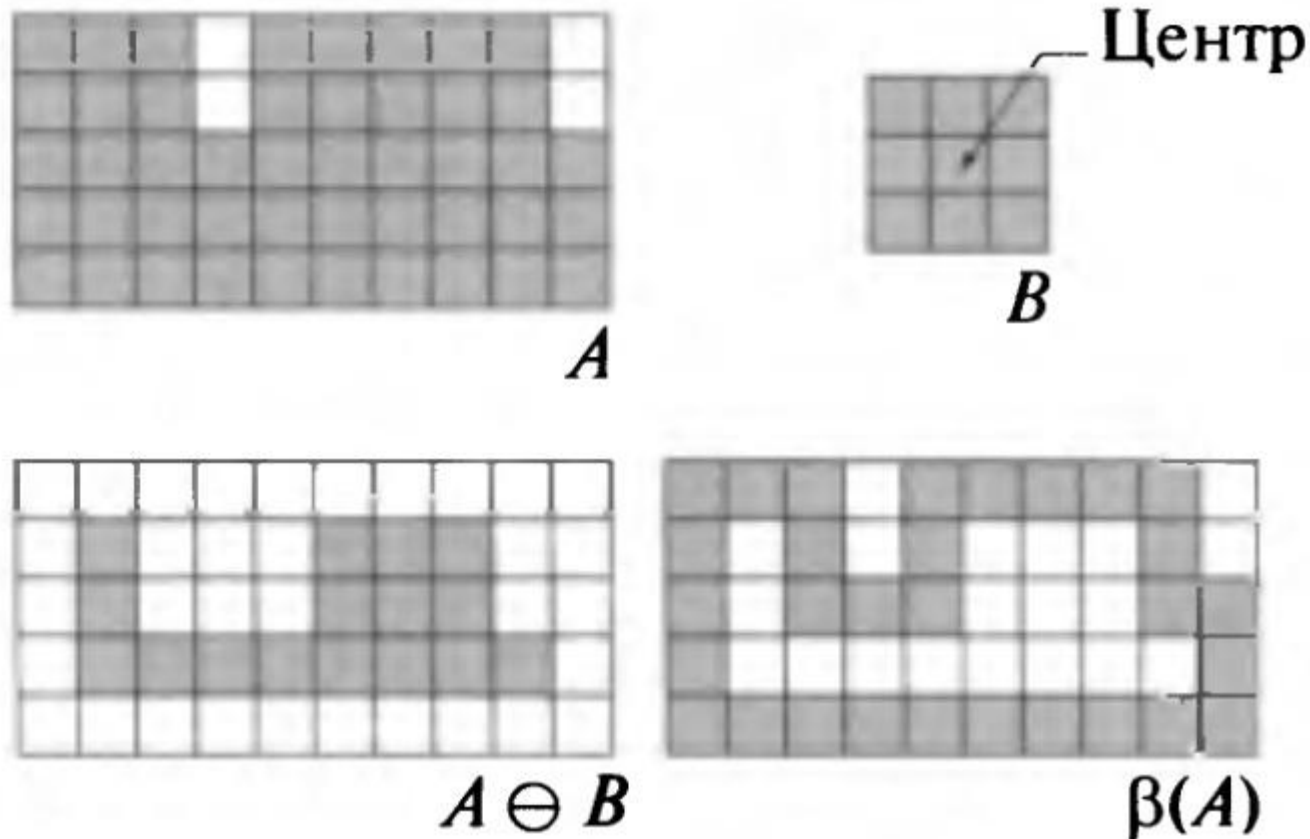
## 2. Сегментация изображений

- Последовательное грамотное применение операций дилатации и эрозии улучшает картинку.



## 2. Сегментация изображений

• Пример. Построение границы объекта морфологическими операциями эрозии и вычитания.



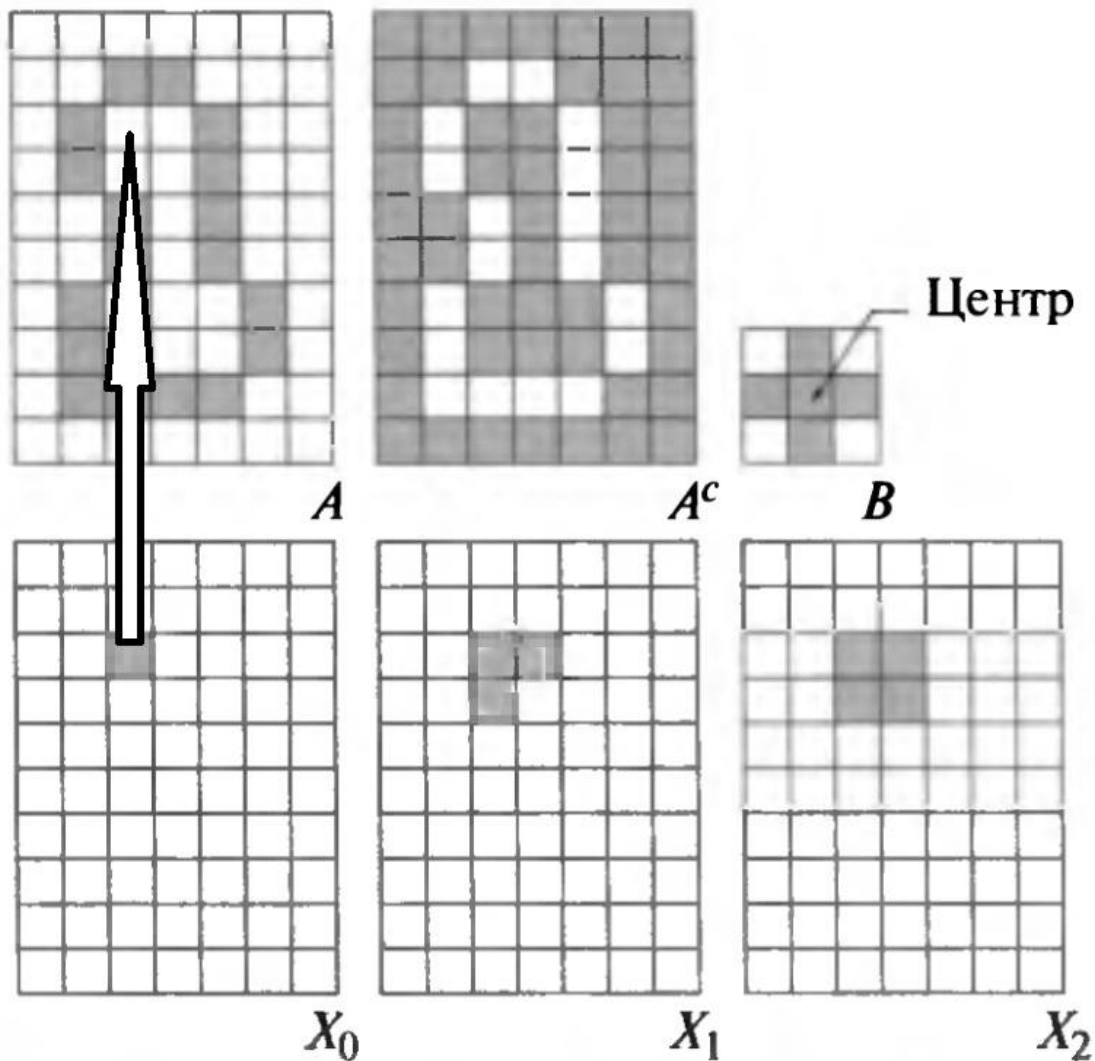
• 1) Применение эрозии; 2) Вычитание из  $A$  результата эрозии.

## 2. Сегментация изображений

- **Пример.** Заполнение области морфологическими операциями.
- Исходное множество  $A$  состоит из граничных точек некоторой области, граница замкнута. Требуется, начиная с некоторой точки внутри этой границы, заполнить единичными значениями всю область внутри  $A$ .
- Предполагаем, что все точки внутри  $A$  имеют значение 0, в результате заполнения им присваивается значение 1.

## 2. Сегментация изображений

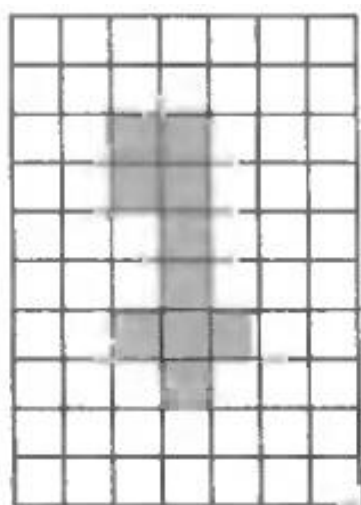
- 1) Применение дилатации. Алгоритм начинает работу с точки  $X_0$ , применяем дилатации с ядром  $B$  с центром в  $X_0$ , берем пересечение результата с дополнением  $A^c$  получаем  $X_1$



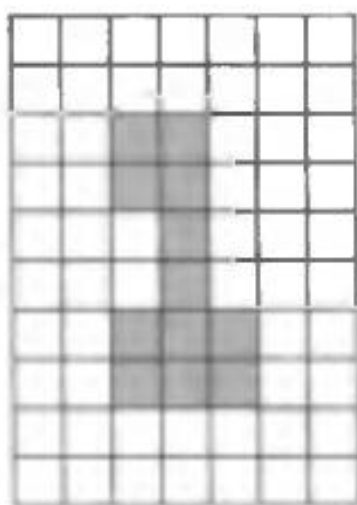


## 2. Сегментация изображений

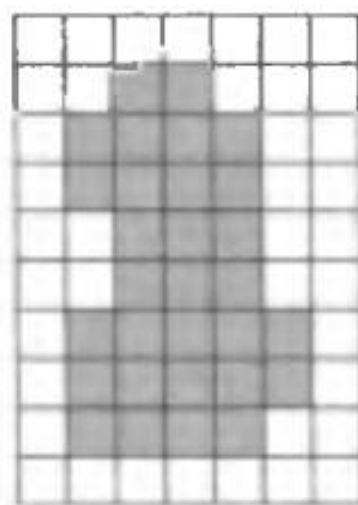
- 2) Нарращиваем область применение дилатации. На каждом шаге берем пересечение результата с дополнением  $A^c$ .  $X_7$  состоит из внутренних точек границы. Можно добавить границу и получить полностью область с границей.



$X_6$



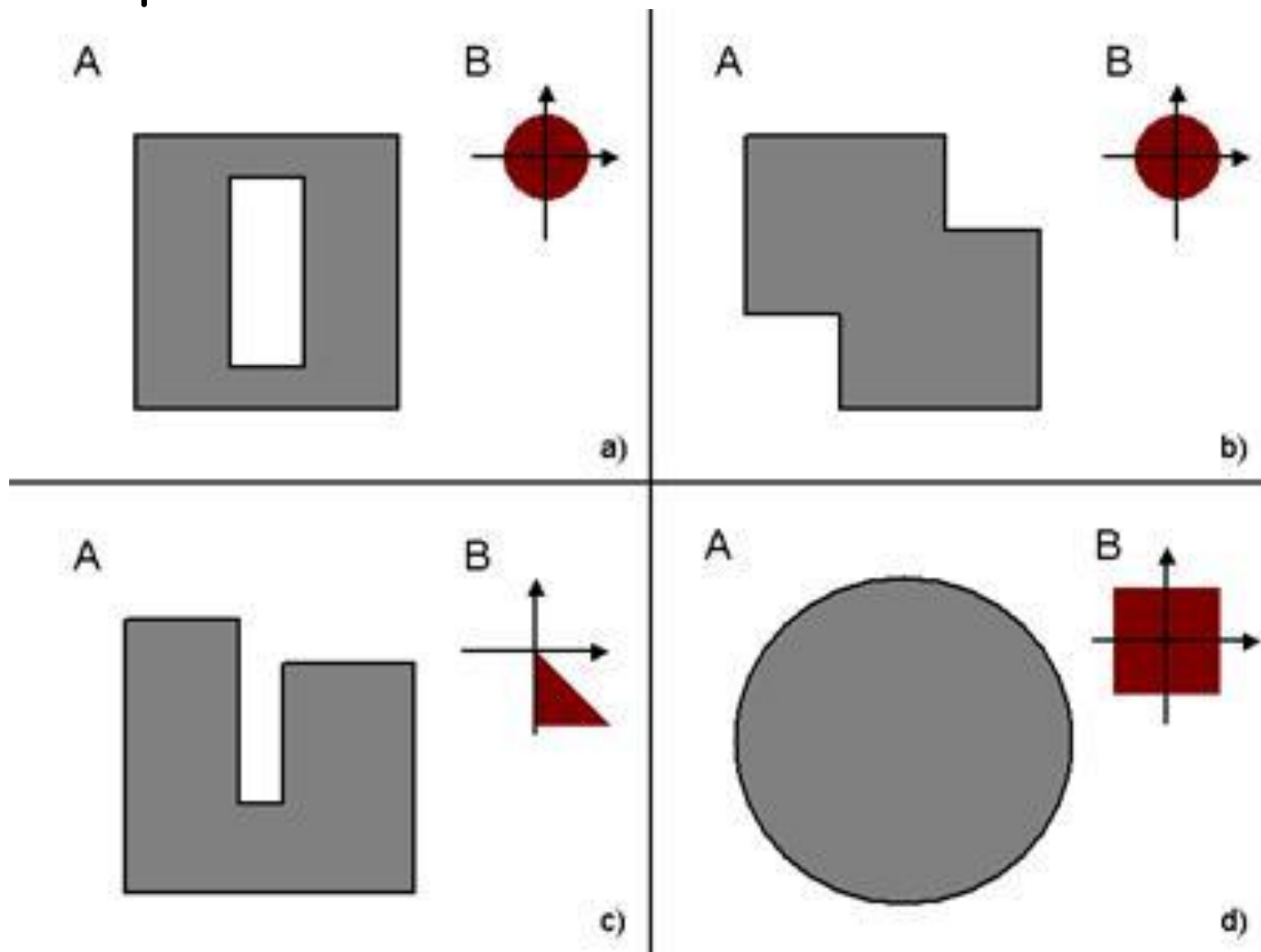
$X_7$



$X_7 \cup A$

## 2. Сегментация изображений

• Морфологическими операциями можно строить выпуклую оболочку фигуры или множества точек, утолщать и утоньшать границы области и т.п.



## 2. Сегментация изображений

- **Преобразование Хафа.**
- Рассмотрим решение задачи: на плоскости дано множество точек. Требуется провести прямые, на которых лежат 3 и более точек.
- Решение, которое первым приходит в голову - провести все прямые через каждую пару точки и проверить каждую прямую, лежит ли на ней третья точка. Такое решение требует достаточно много ресурсов, в том числе и времени.
- Хаф [Hough, 1962] предложил другой подход, который теперь называют **преобразованием Хафа**.
- Изложим идею преобразования.
- Возьмем точку  $(x_i, y_i)$  из заданного множества  $n$  точек и рассмотрим общее уравнение прямой на плоскости в виде  $y = ax + b$ . Очевидно, что через точку  $(x_i, y_i)$  проходит бесконечно много прямых, удовлетворяющих этому

## 2. Сегментация изображений

• Если переписать это уравнение в виде

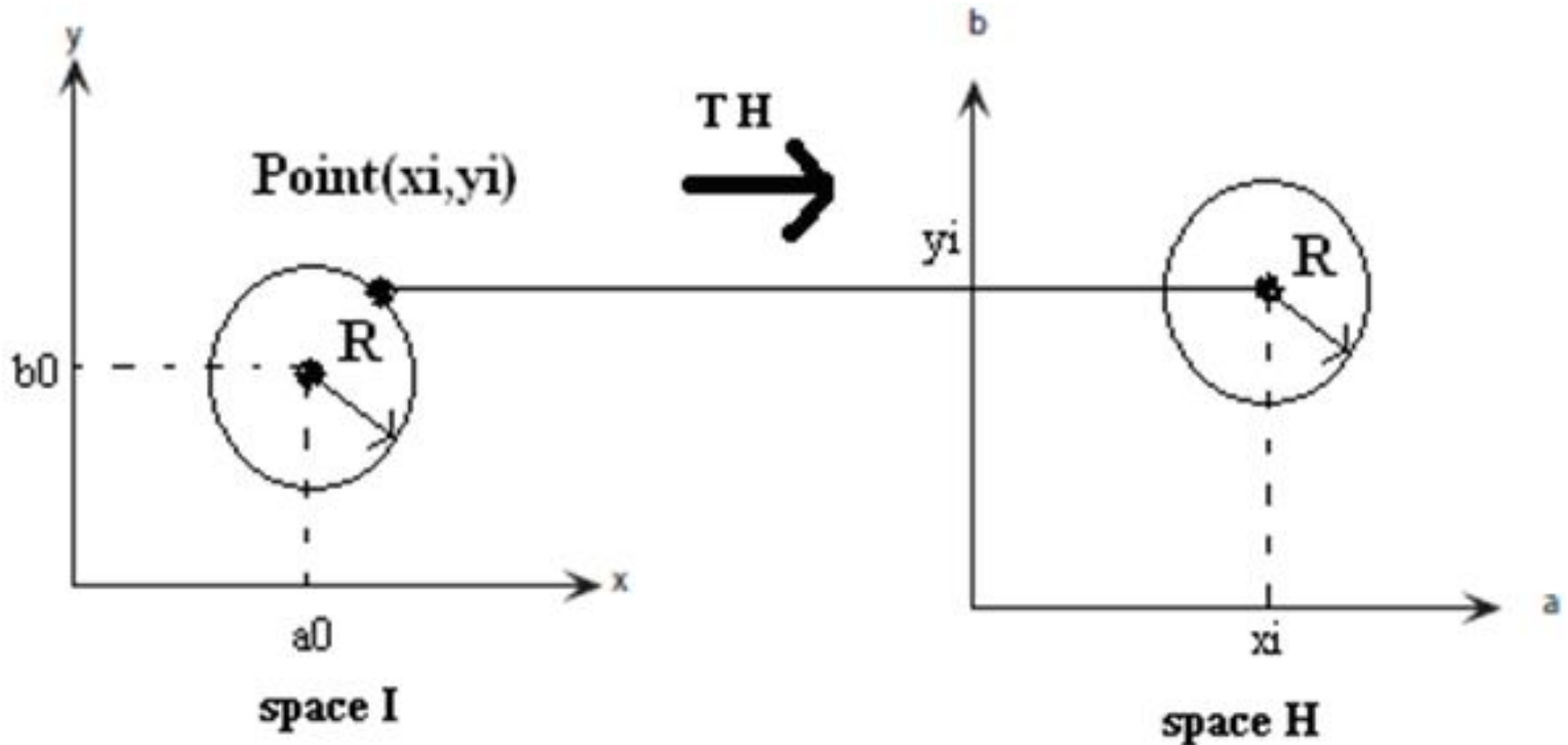
$$-b = -x_i a + y_i$$

и рассмотреть плоскость  $a$   $b$ , называемую пространством параметров, то для заданной пары  $(x_i, y_i)$  получаем уравнение единственной прямой на этой плоскости. Каждая точка  $(a, b)$  соответствует одной прямой, проходящей через точку  $(x_i, y_i)$ .

• Если построить  $n$  прямых для всех точек  $(x_j, y_j)$ , то точка, в которой пересекаются  $k$  таких прямых соответствует прямой на плоскости  $(x, y)$  которая проходит через  $k$  точек.

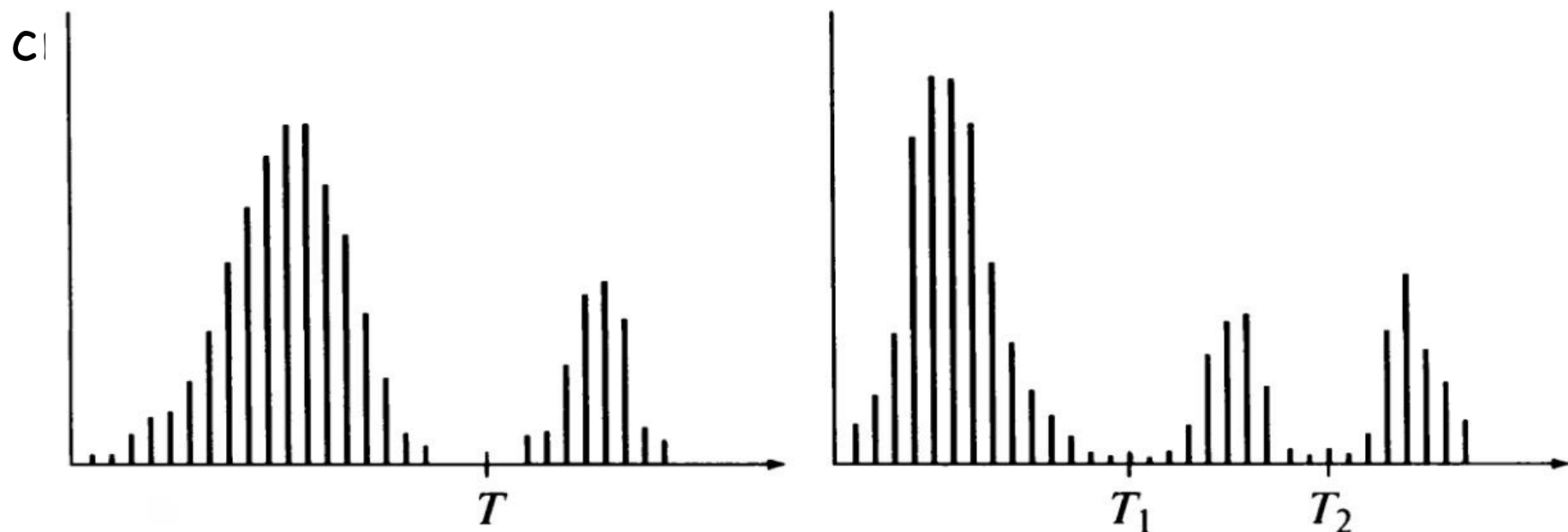
## 2. Сегментация изображений

- В пространстве  $H$  построена окружность. Точки на этой окружности соответствуют центрам тех окружностей в пространстве  $I$ , которые проходят через точку  $(x_i, y_i)$



## 2. Сегментация изображений

- Пороговая обработка.
- Для изображений, на которых объекты интереса и фон рисунка значительно отличаются по яркости, можно применять **пороговую сегментацию**.
- Например, по гистограмме видно, что на рисунке имеется один или два объекта с явно различными яркостями и эти яркости имеют видимое отличие от фона.
- **Пример.** Для рис слева можно применить порог  $T$ , на рис





## 2. Сегментация изображений

- Применение порогов для сложных изображений затруднительно и часто не дает результата. Даже введение **динамических порогов**, зависящих от координат пикселя не улучшает ситуацию.
- Гистограмма рисунка зависит от **освещения объектов**, если это фотография, то возможны **отражения света** от объектов.
- Пороговая сегментация может применяться в тех случаях, где исследователь может **управлять освещением сцены**. Это например, визуальный технический контроль, когда специалист сам устанавливает фотокамеры и приборы освещения.
- Некоторые технические задачи также могут решаться с использованием порогов, например, обработка отпечатков

## 2. Сегментация изображений

- Алгоритм наращивания областей (Region growing).
- Наращивание областей представляет собой процедуру, которая группирует пиксели или подобласти в более крупные области по заданным критериям.
- 
- Основной подход состоит в том, что вначале берется исходный пиксель, играющий роль «затравки», а затем на него и на последующие выбранные пиксели **наращиваются соседи** путем присоединения соседних пикселей, которые по своим свойствам близки к затравке.
- Близость может определяться яркостью или цветом в определенном диапазоне.
- **Выбор затравки** или нескольких затравок (начальных точек роста), может основываться на сути задачи.

## 2. Сегментация изображений

- Правило близости и правило присоединения являются основными в алгоритме. Пусть приращение идет по одному пикселю и по интервалу яркости. Предположим, что граница сегментируемого объекта из-за зашумления содержит пиксель (или пиксели), принадлежащие интервалу приращения.
- В этом случае пиксель (или пиксели) границы будут добавлены в область объекта и далее объект может распространиться далеко за свои реальные границы. Часто это можно избежать, если присоединять не один пиксель, а сразу небольшую область, напр выполнять наращивание по квадратам  $2 \times 2$ .

## 2. Сегментация изображений

- **Пример** наращивания областей. Требуется сегментировать водные бассейны на острове.
- Если применять пороговую сегментацию, то будет выделена вся водная поверхность.
- Для алгоритма наращивания областей в этой задаче самым сложным будет выбор затравки. Если будут выбраны две затравки по соседству, алгоритм легко выделит озера.

