

# РАБОТА С КОНТУРАМИ

Лекция 7

# План

- **Прослеживание контуров**
- Связывание контуров
- Анализ контуров
- Метод активных контуров

# Различные виды изображений: различные способы получения контуров и различные проблемы

- Бинарные
- Полутоновые
- Цветные

# Поиск границ и контуров на бинарном изображении



# Операция оконтуривания объекта

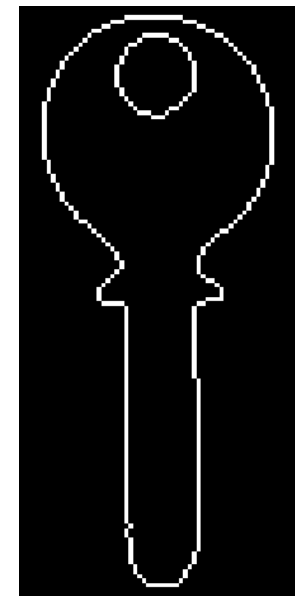
При работе с бинарными изображениями контуры объекта можно получить с помощью операций математической морфологии

Внутреннее оконтуривание

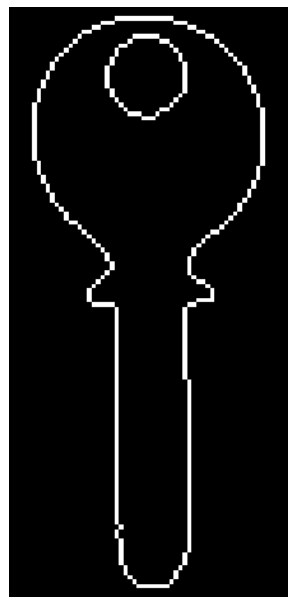
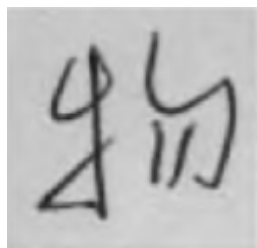
$$C_1 = A - (A (-) B)$$

Внешнее оконтуривание

$$C_0 = (A (+) B) - A$$



# Как отследить контур



# Виды обработки при получении контуров

- **Локальная**
- **Глобальная**

# Классификация пикселей для прослеживания контуров в бинарных изображениях

- Изолированный пиксел
- Вершина (начало или конец сегмента)
- Внутренний пиксел сегмента
- Угловая точка
- Стык

	1	2	3	4	5
1	1	0	0	0	1
2	0	1	0	1	0
3	0	0	1	0	0
4	0	0	1	0	0
5	0	0	1	1	1

Маркированное контурное изображение



# Операции для алгоритма прослеживания сегментов

1. Прослеживание нового сегмента
2. Добавление внутреннего пиксела к сегменту
3. Завершение сегмента
4. Поиск стыка
5. Поиск угла

	1	2	3	4	5
1	1	0	0	0	1
2	0	1	0	1	0
3	0	0	1	0	0
4	0	0	1	0	0
5	0	0	1	1	1

Маркированное контурное изображение

# Результат применения алгоритма для прослеживания контуров к маркированному изображению

	1	2	3	4	5	Segment ID	Length	List
1	1	0	0	0	1			
2	0	1	0	1	0	1	3	(1,1)(2,2)(3,3)
3	0	0	1	0	0	2	3	(1,5)(2,4)(3,3)
4	0	0	1	0	0	3	3	(3,3)(4,3)(5,3)
5	0	0	1	1	1	4	3	(5,3)(5,4)(5,5)

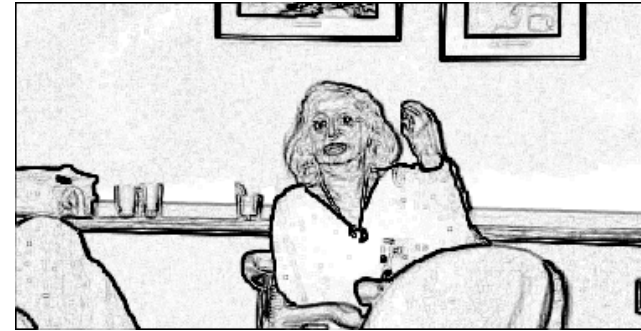
# Виды обработки при получении контуров

- Локальная
- **Глобальная**

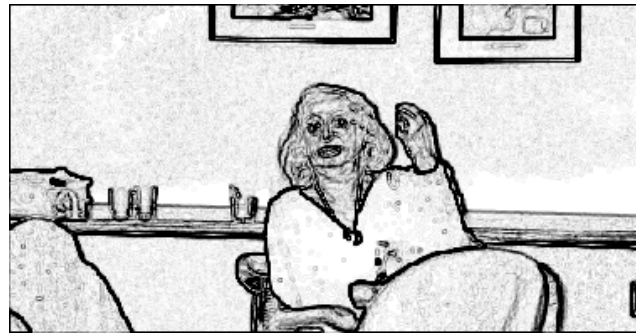
# Выделение границ - карты силы краев



Робертса

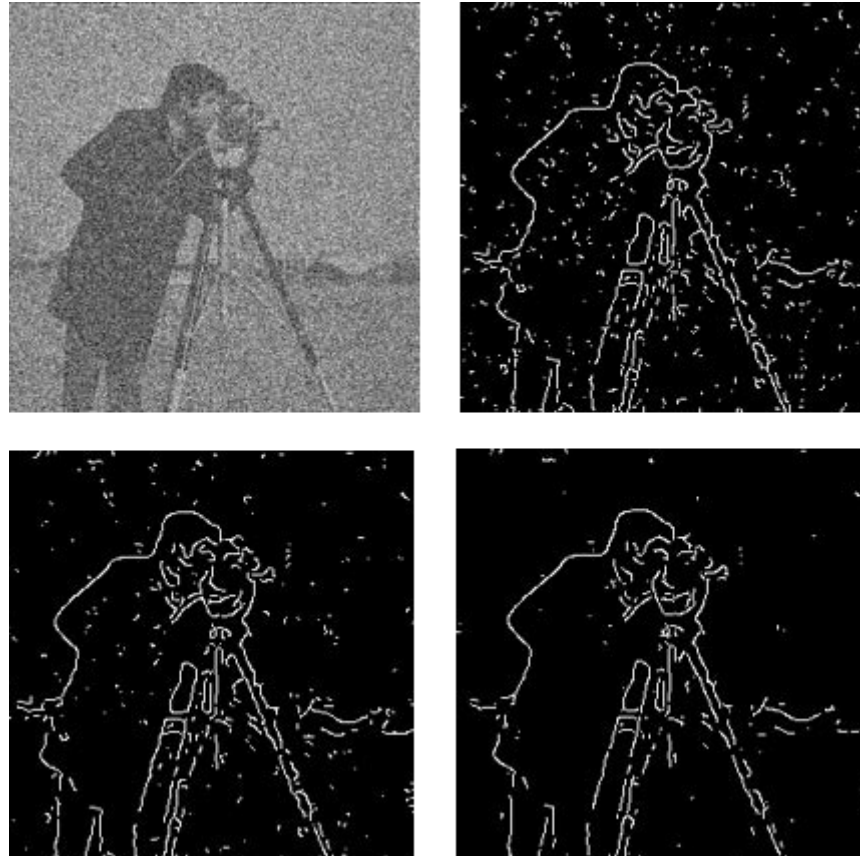


Превитт



Собеля

# Методы, основанные на операторах выделения краёв



# Выделение краёв

- Вычисление градиента – это еще не всё...



Исходное изображение



Карта силы краев

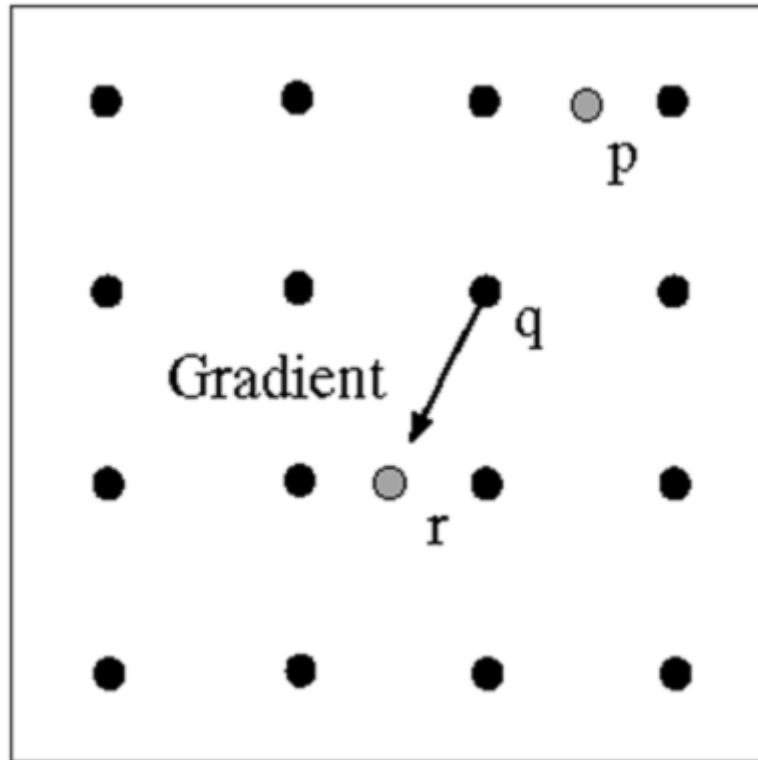
- Чего не хватает?
  - Точности – края «толстые» и размытые
  - Информации о связности

# Детектор Canny

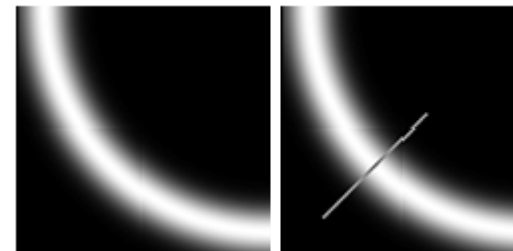
- Свертка изображения с ядром – размывание изображения фильтром Гаусса
- Поиск значения и направления градиента
- Подавление немаксимумов (Non-maximum suppression)
  - выделение локальных максимумов
  - утоньшение полос в несколько пикселей до одного пикселя
- Связывание краев и обрезание по порогу (гистерезис)
  - Определяем пороги: нижний и верхний
  - Верхний порог используем для инициализации кривых
  - Нижний порог используем для продолжения кривых

J.Canny, A Computational Approach To Edge Detection

# Поиск локальных максимумов



Максимум достигается в  $q$ , если значение больше  $p$  и  $r$ .



Source: D. Forsyth

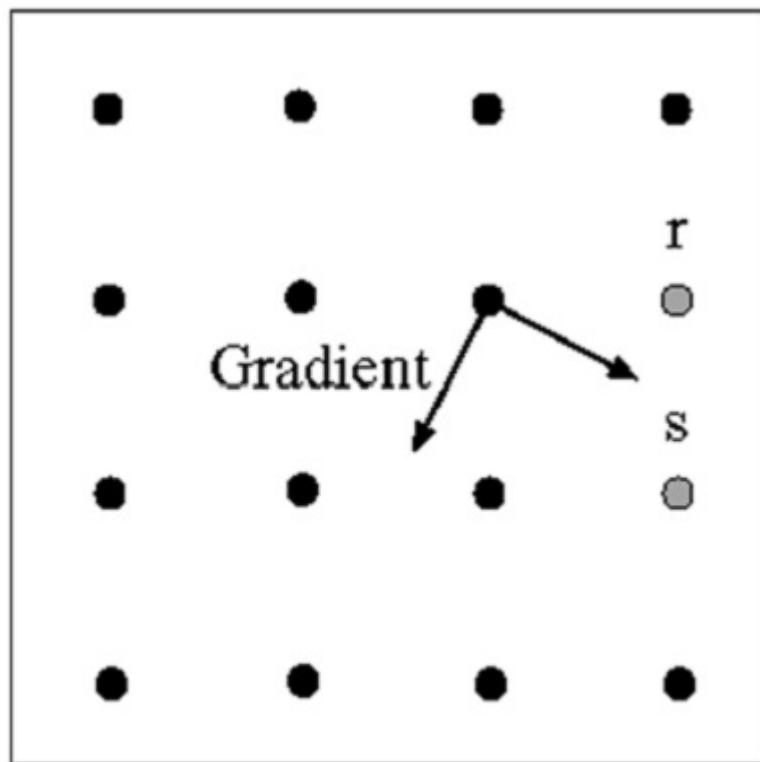


## Отсечение по порогу

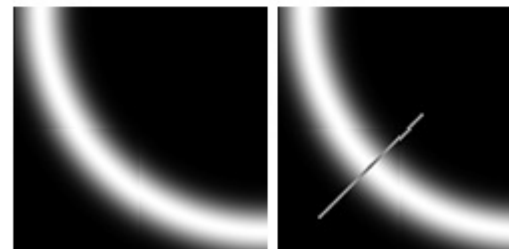
- Проверяем точку, чтобы значение градиента было выше порога
  - Используем **гистерезис**
    - Большой порог для начала построения кривой и низкий порог для продолжения края (связывания)



# Связывание точек



Пусть отмеченная точка – край. Строим касательную к границе (нормаль к направлению градиента) и используем ее для предсказания новой точки (это либо  $s$  либо  $r$ ).



Source: D. Forsyth

# Эффект гистерезиса



Исходное изображение



Высокий порог  
(сильные края)



Низкий порог  
(слабые края)



Порог по гистерезису

# Пример



Утоньшение  
(non-maximum suppression)

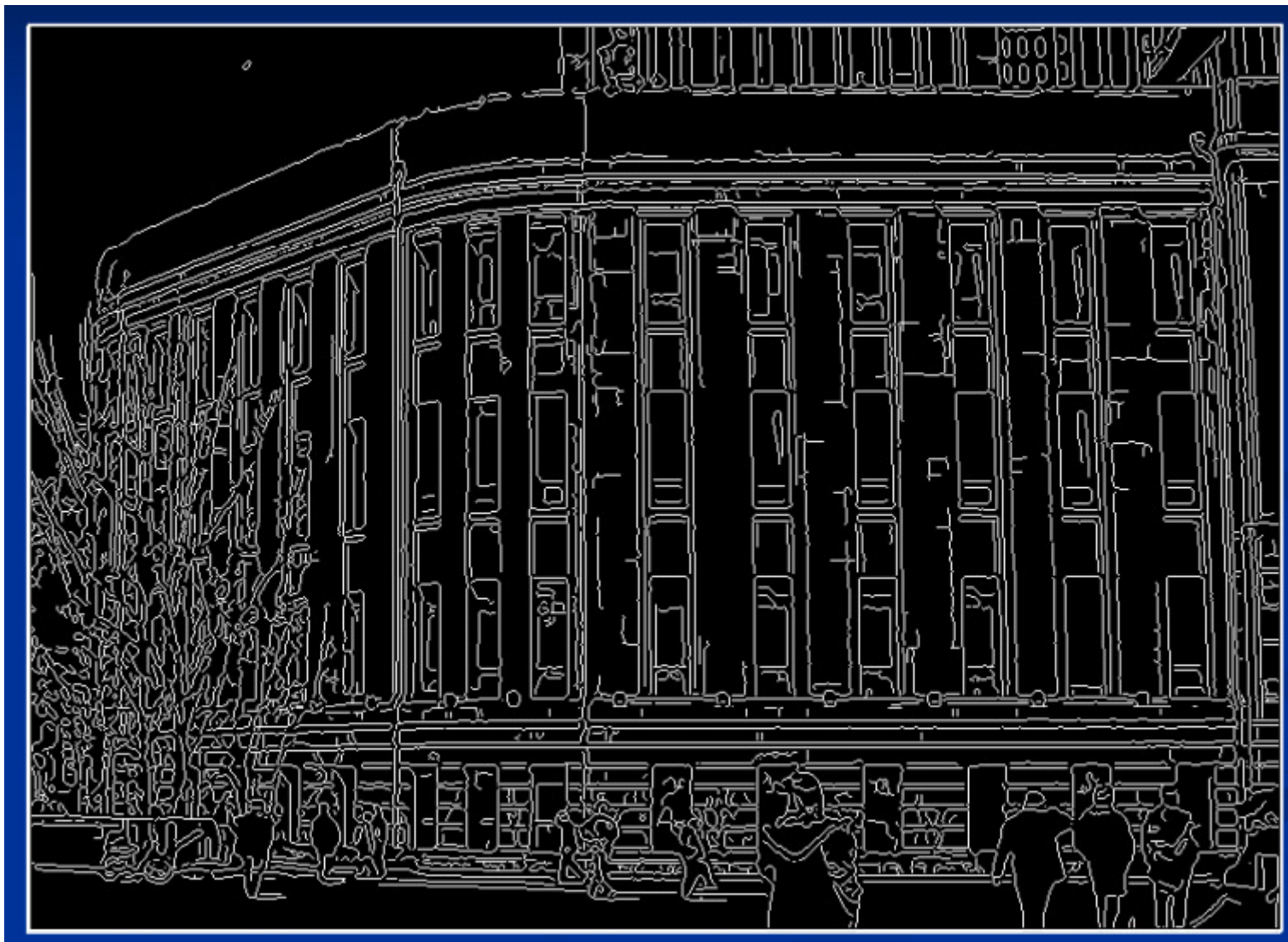


Отсечение по порогу

## Удачный пример применения Sanny



# Canny



## Влияние $\sigma$ (размер ядра размытия)



original

Canny with  $\sigma = 1$

Canny with  $\sigma = 2$

### Выбор $\sigma$ зависит от задачи

- большое  $\sigma$  - поиск крупных границ
- маленькое  $\sigma$  - выделение мелких деталей

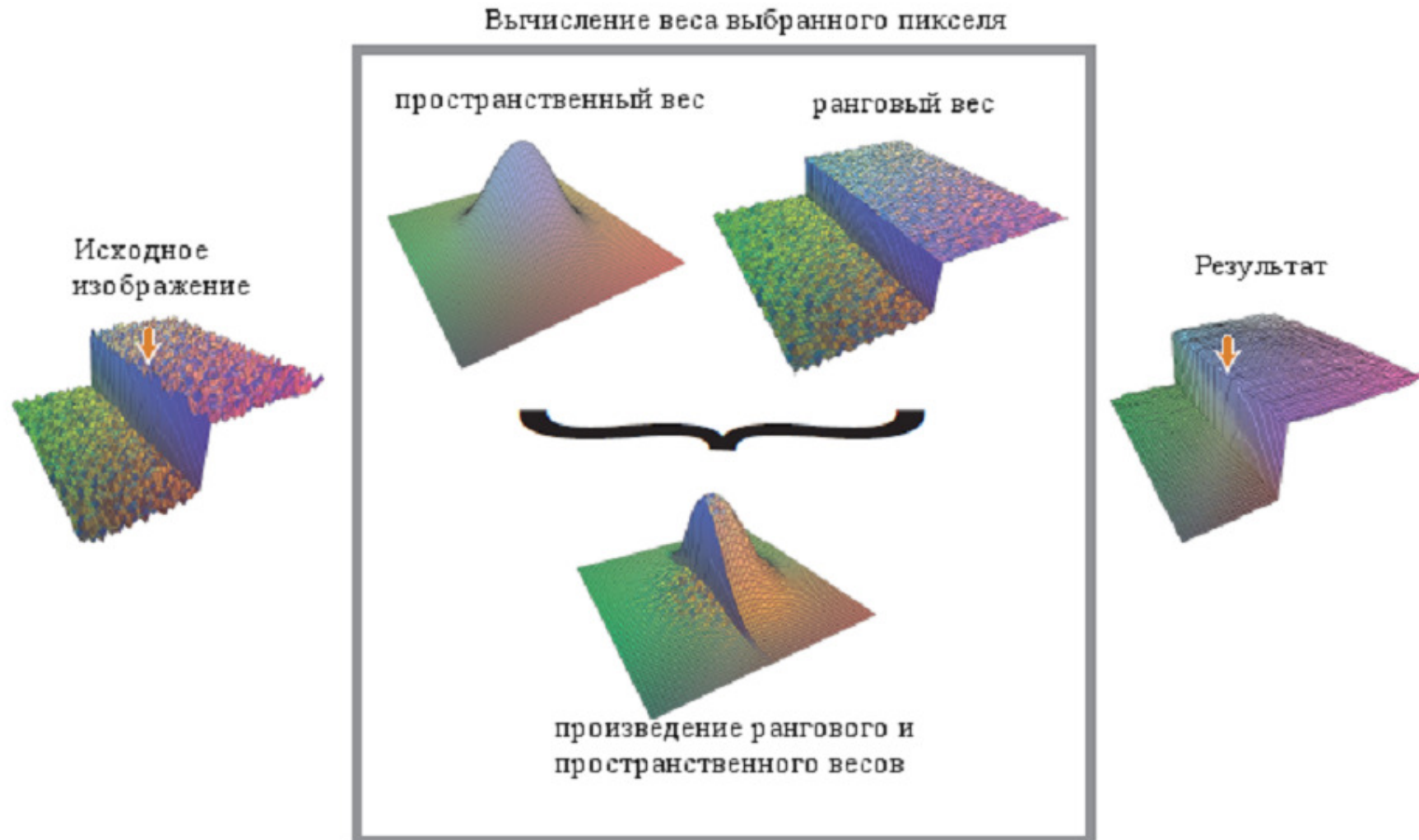
Source: S. Seitz

# Сглаживание билатеральным фильтром с сохранением границ





# Визуализация билатерального фильтра с сохранением границ

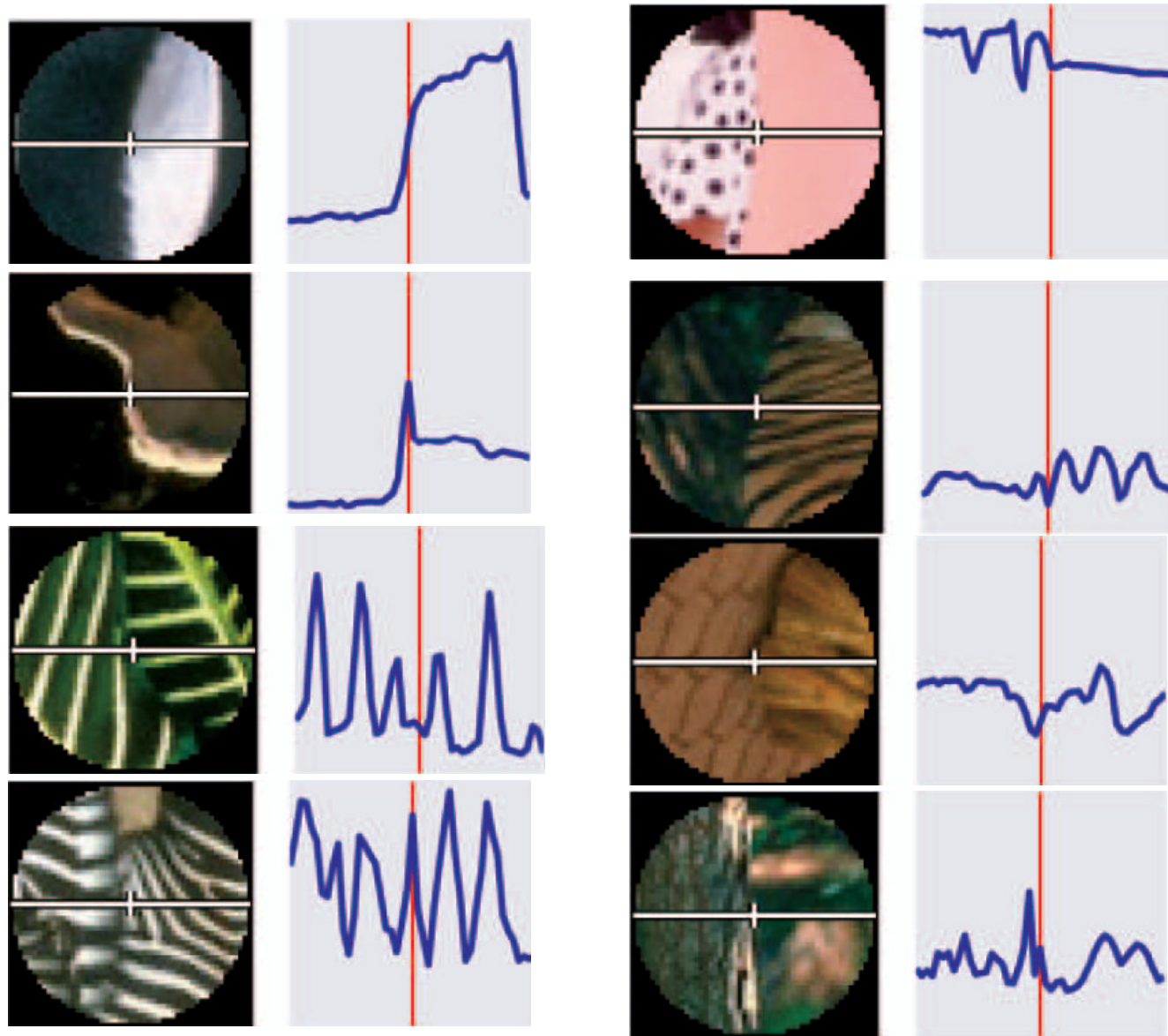


## Билатеральный фильтр с сохранением границ

$$h(a_0) = k^{-1} \sum_{i=0}^{n-1} f(a_i) \times g(a_i) \times r(a_i)$$

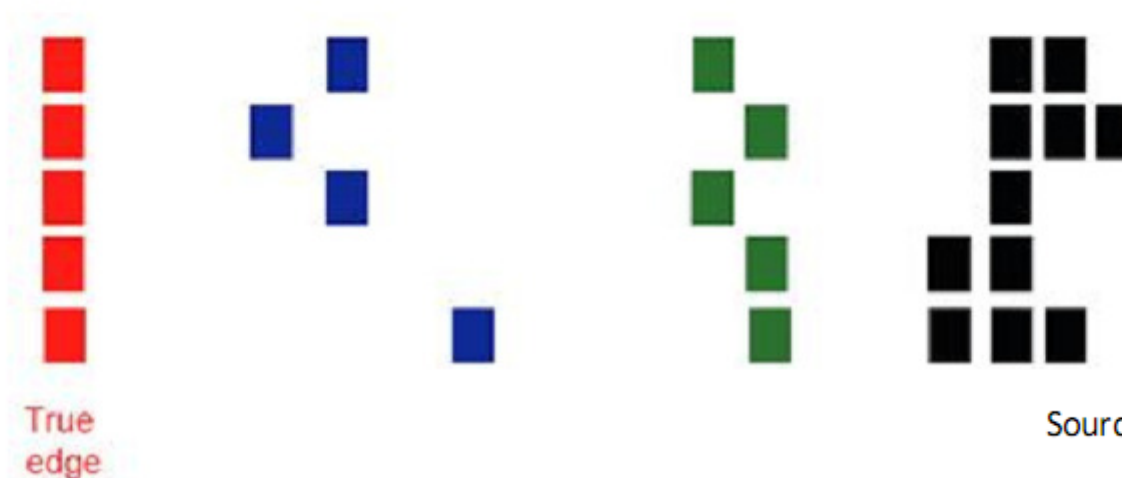
$$r(a_i) = e^{-\frac{(f(a_i) - f(a_0))^2}{2\sigma^2}} \quad g(x, y) = e^{-\frac{x^2 + y^2}{2t^2}}$$

# Ограничения детектора



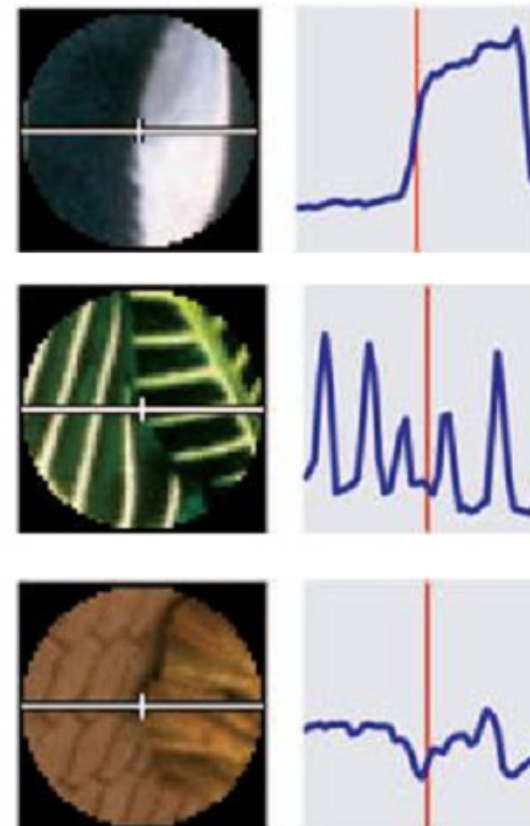
# Критерии качества детектора

- **Good detection:** оптимальный детектор должен минимизировать ошибки 1 и 2го родов (ложные края и пропущенные края)
- **Точная локализация:** найденный край должен быть как можно ближе к истинному краю
- **Единственный отклик:** детектор должен выдавать одну точку для одной точки истинного края, т.е. локальных максимум вокруг края должно быть как можно меньше



# Probability boundary (Pb) -детектор

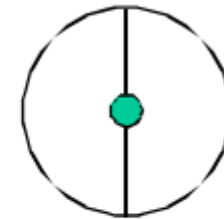
- Недостаточно находить градиент изображения
- Идея – обучить классификатор граница / не граница
- Будем использовать разные признаки:
  - Яркость
  - Цвет
  - Текстура
- «Probability boundary» (Pb) детектор



D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. PAMI 2004.

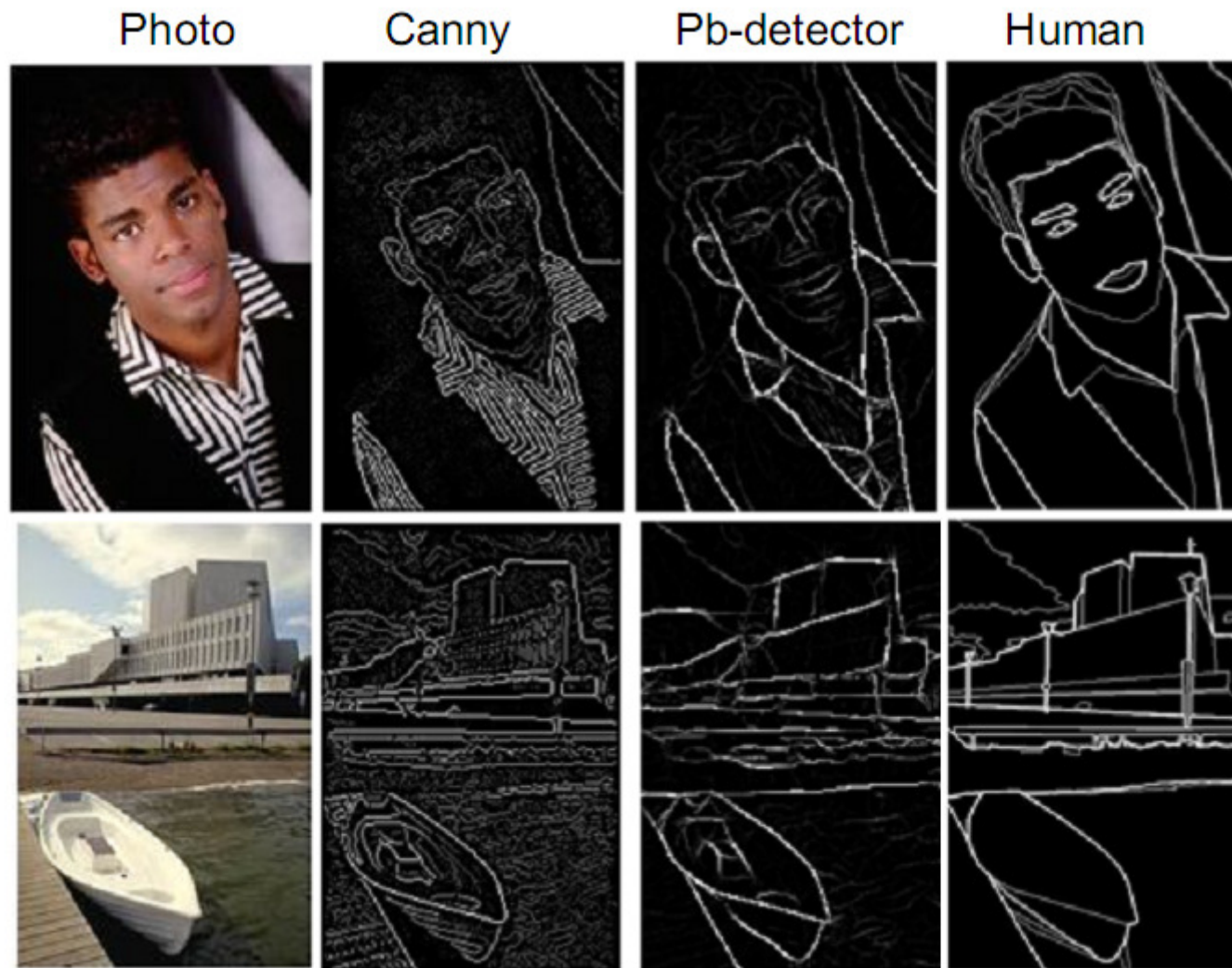
# Схема Рb-детектора

- Берём круг с центром в исследуемом пикселе
- Выбираем ориентацию края
- Считаем градиент между признаками в левой и правой половине круга
- Обучаем классификатор края
- Требуется размеченная коллекция изображений



D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. PAMI 2004.

# Результат Pb-детектора

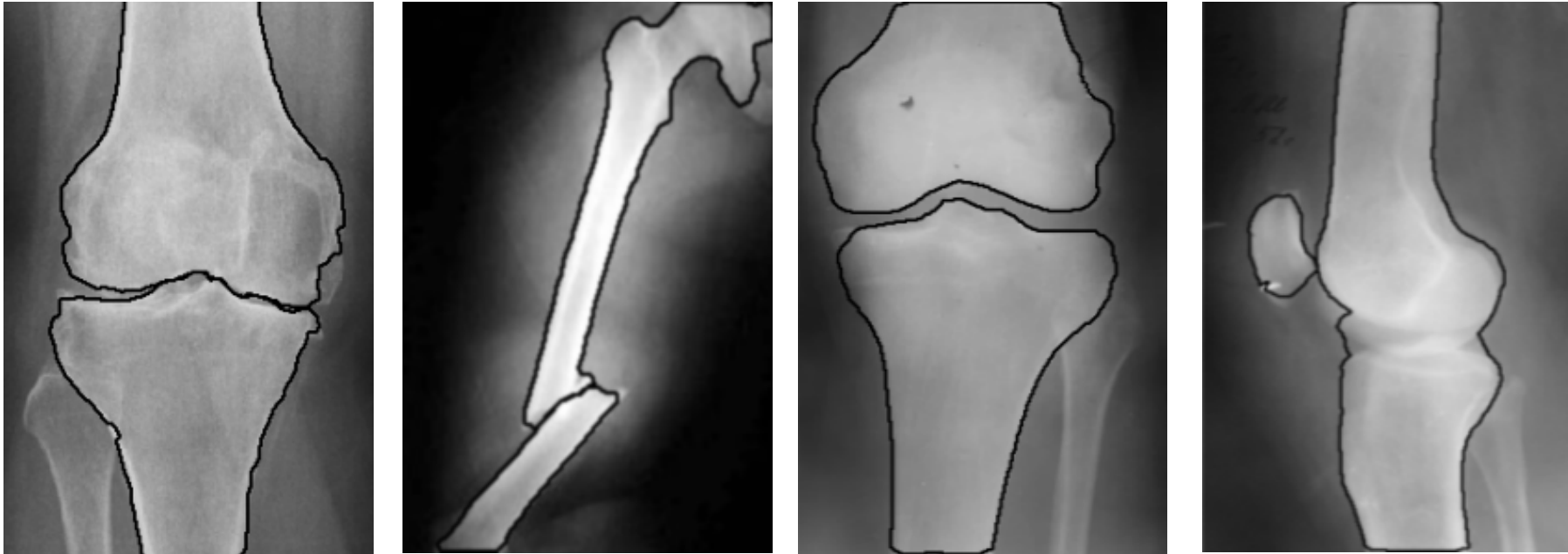


# План

- Прослеживание контуров
- **Связывание контуров**
- Анализ контуров
- Метод активных контуров



# Детектирование контуров на медицинских изображениях



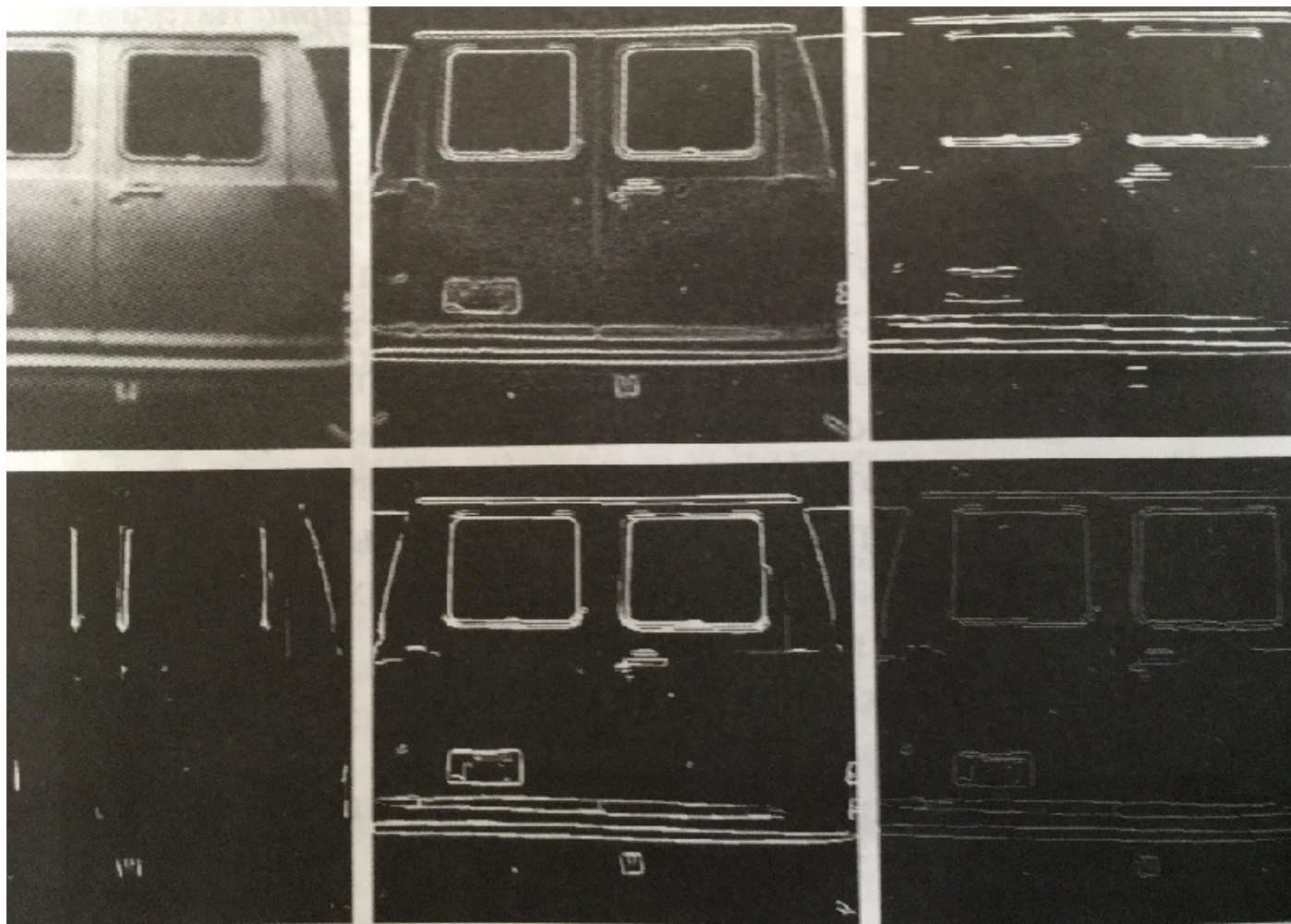
Работа Михайличенко Алексея (2016 магистратура ФИИТ )

# Выделение ногтей и наложение текстуры



Работа Пака Владислава (2016 магистратура ФИИТ )

# Связывание контуров с помощью локальной обработки



534x546

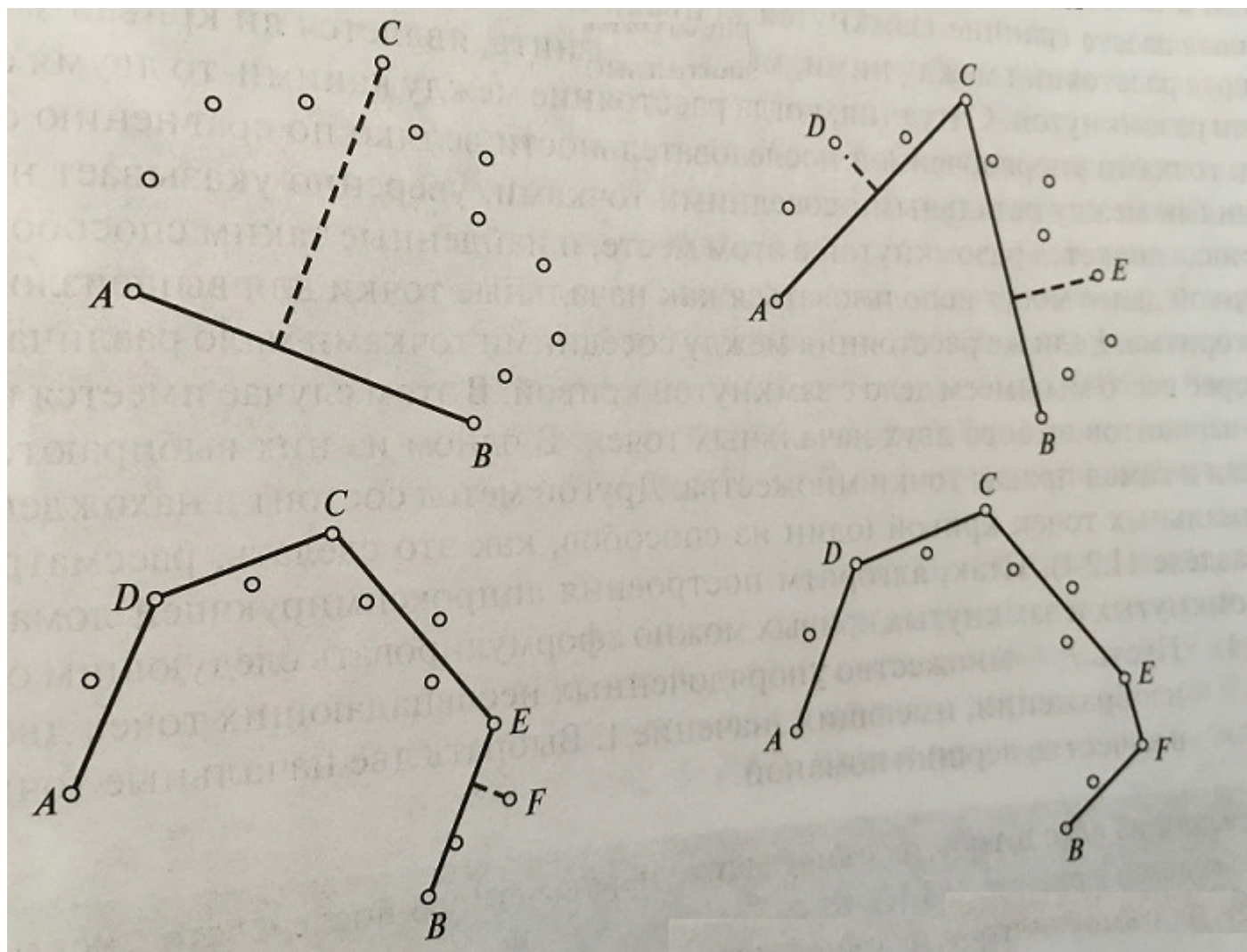
$E=30\%$  от макс  
градиента

$A=45^\circ$

Заполнение  
пропусков в 25  
пикселей

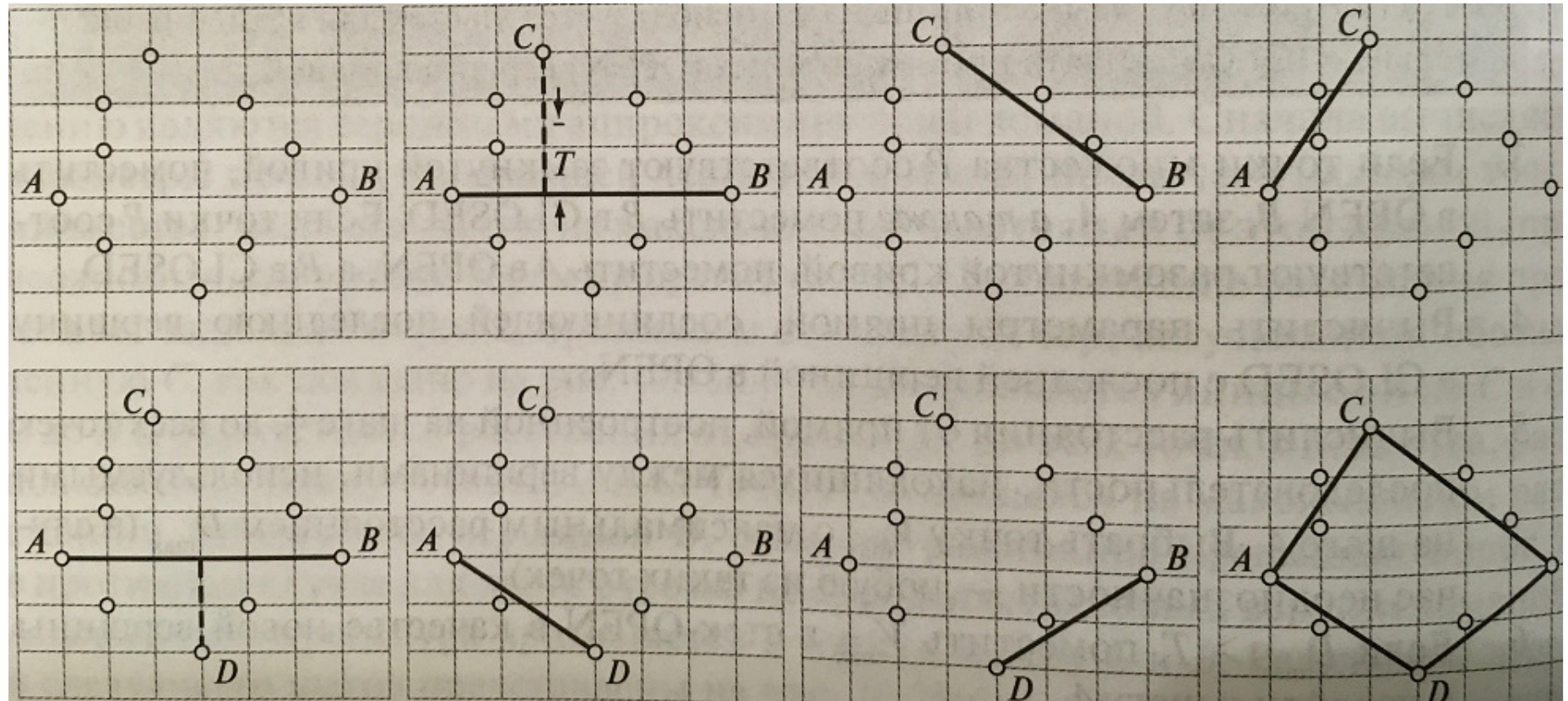
а) исходное изображение; б) модуль градиента; в) горизонтально связанные контурные пиксели; г) вертикально связанные контурные пиксели; д) результат ИЛИ к в) и г); е) результат морфологического утончения

# Обработка в интересующей области

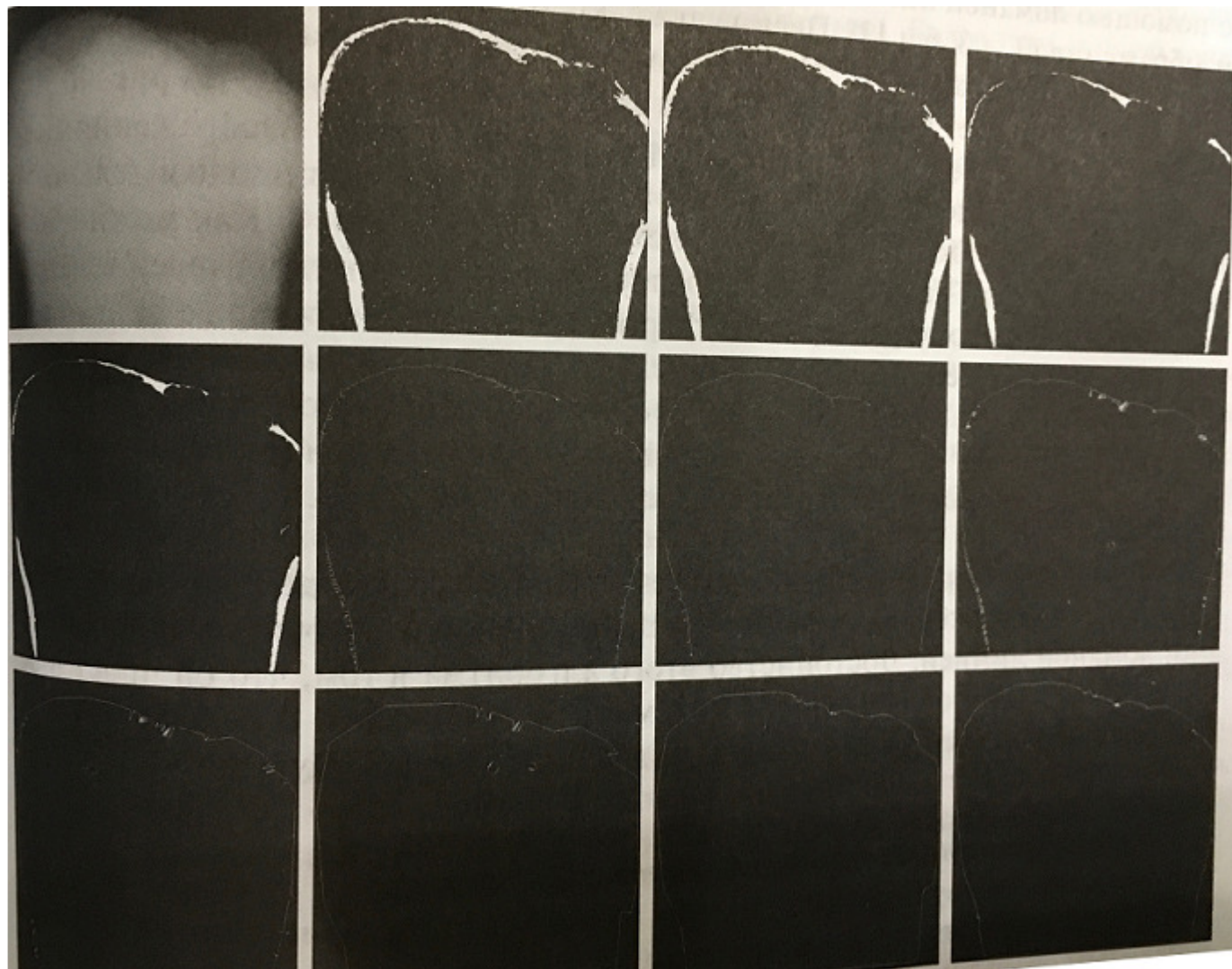


Итерационный алгоритм аппроксимации незамкнутой ломаной

# Итерационный алгоритм аппроксимации замкнутой ломаной



## Пример. Граница зуба



- а) исходное;
- б) градиентное;
- в) результат мажоритарной фильтрации;
- г) морфологическое утончение;
- д) морфологическая очистка;
- е) построение остова;
- ж) усечение отростков;
- з-к) аппроксимация границы;
- л-м) результаты сглаживания

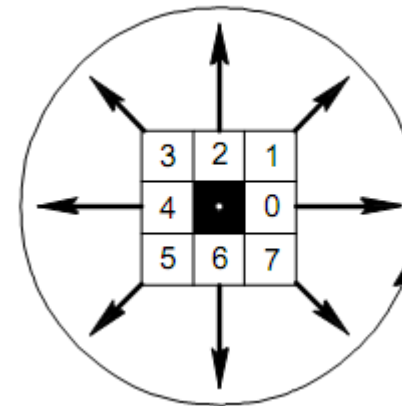
# Локальная обработка полутонового изображения

Критерии сходства

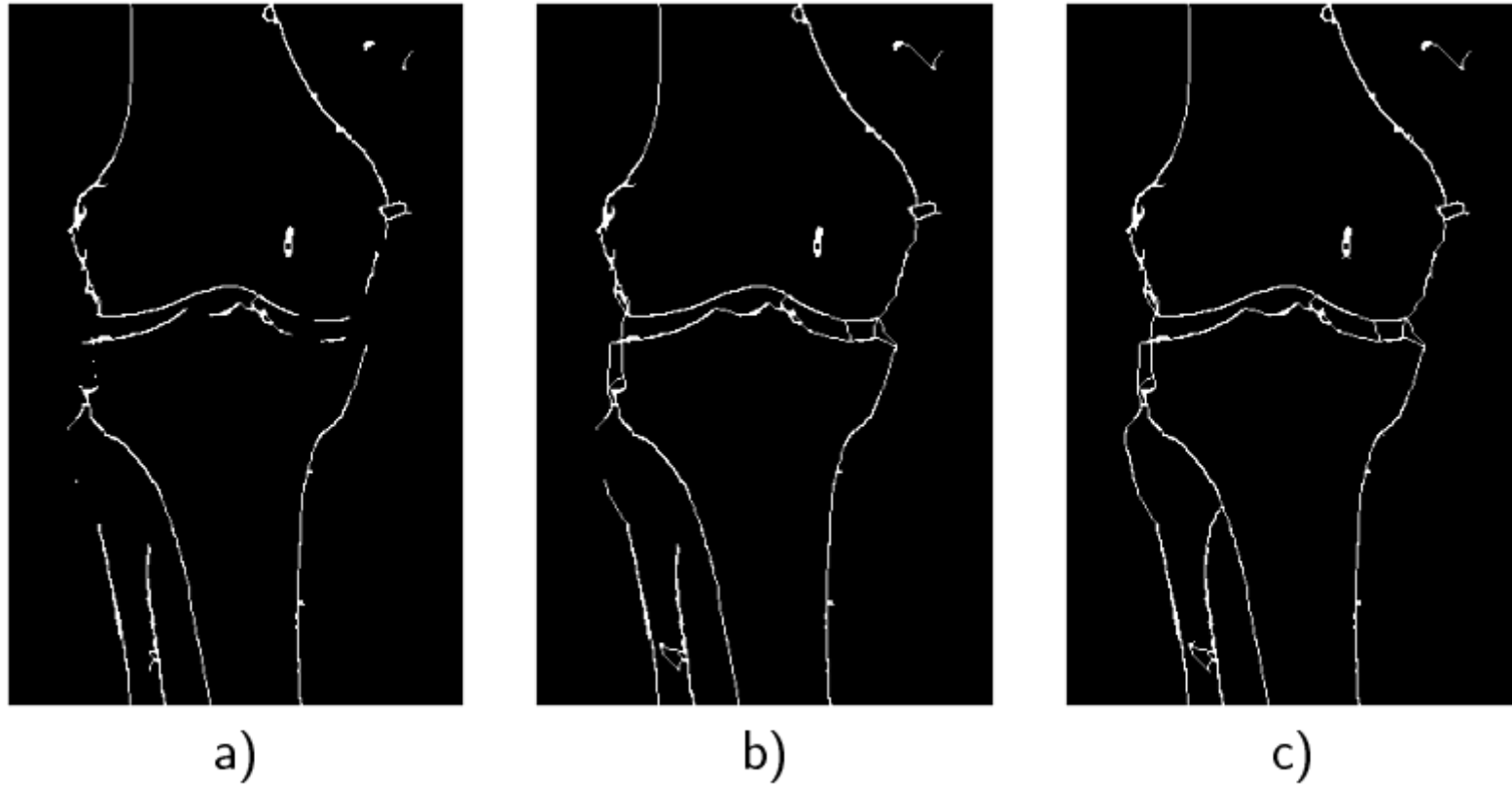
$$|\nabla f(x, y) - \nabla f(x_0, y_0)| \leq E$$

$$|\alpha(x, y) - \alpha(x_0, y_0)| \leq A$$

обычно используют окрестность 3×3



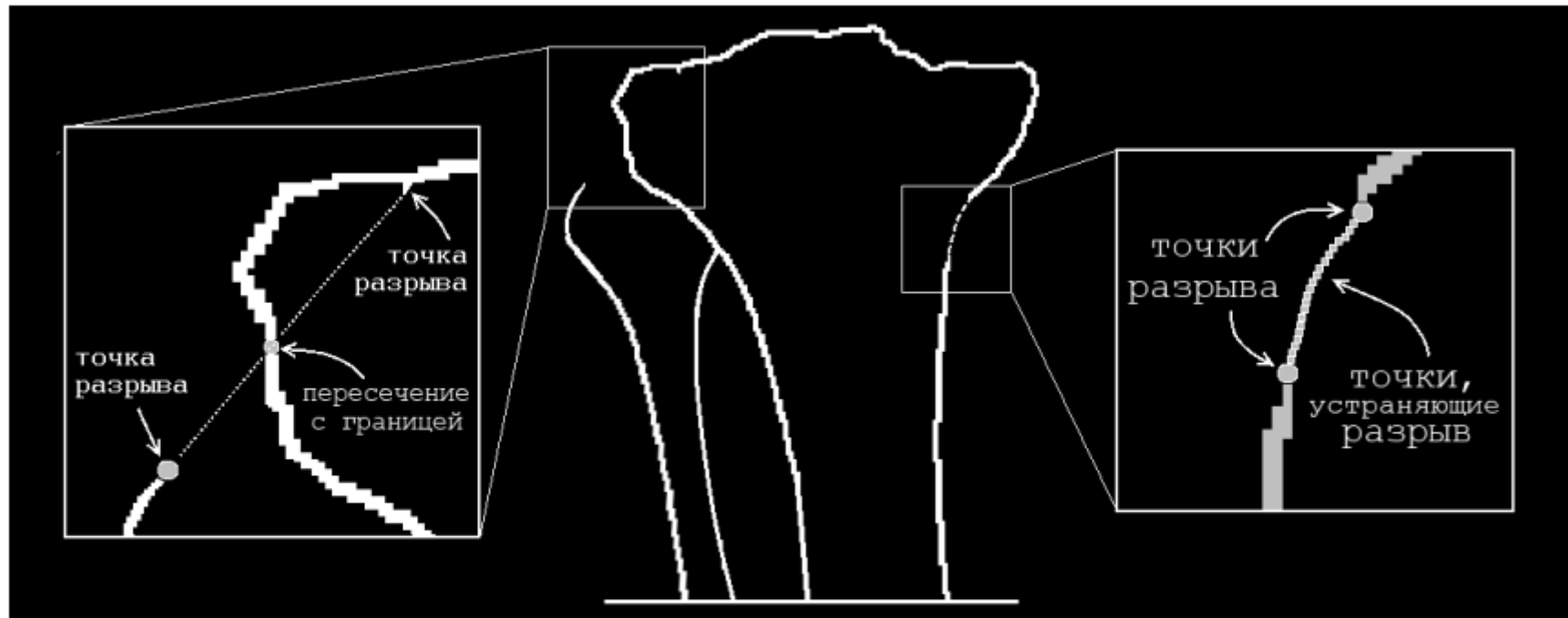
## Пример устранения разрывов



Процесс связывания границ: несколько итераций

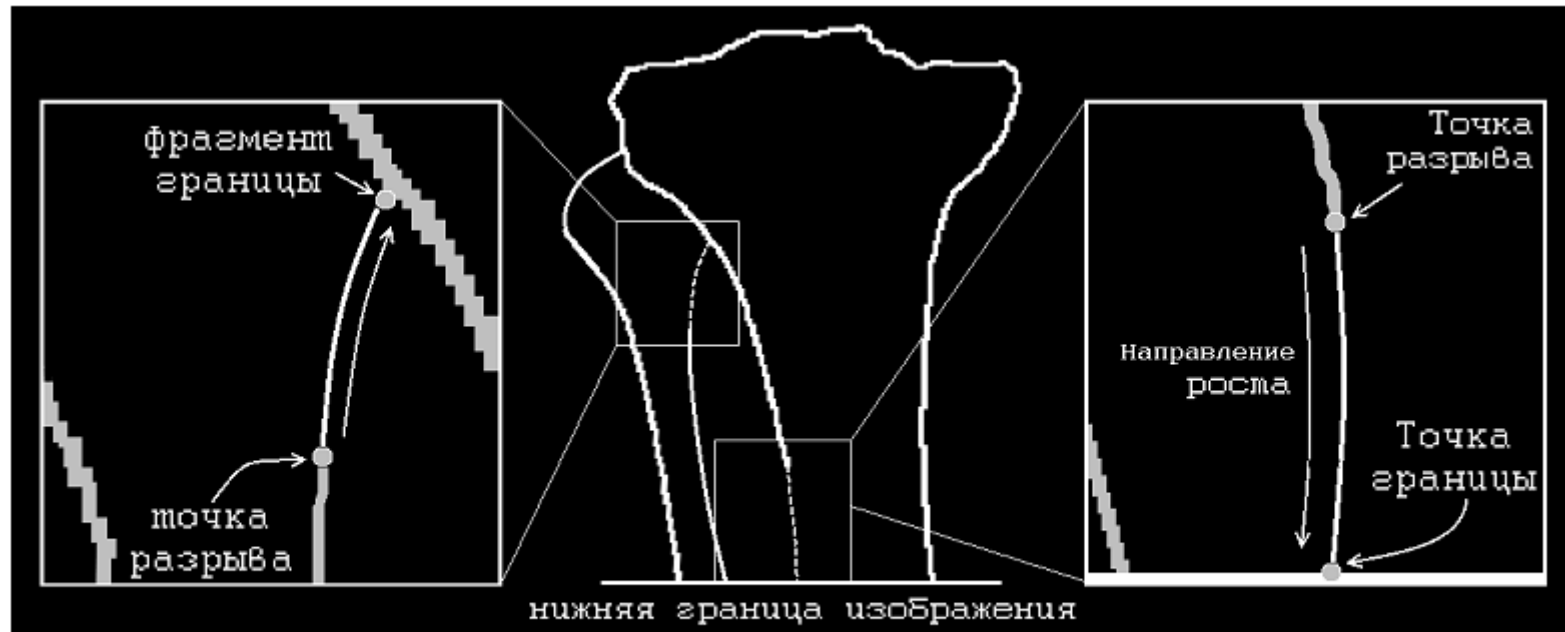


## Пример. Прослеживание контуров



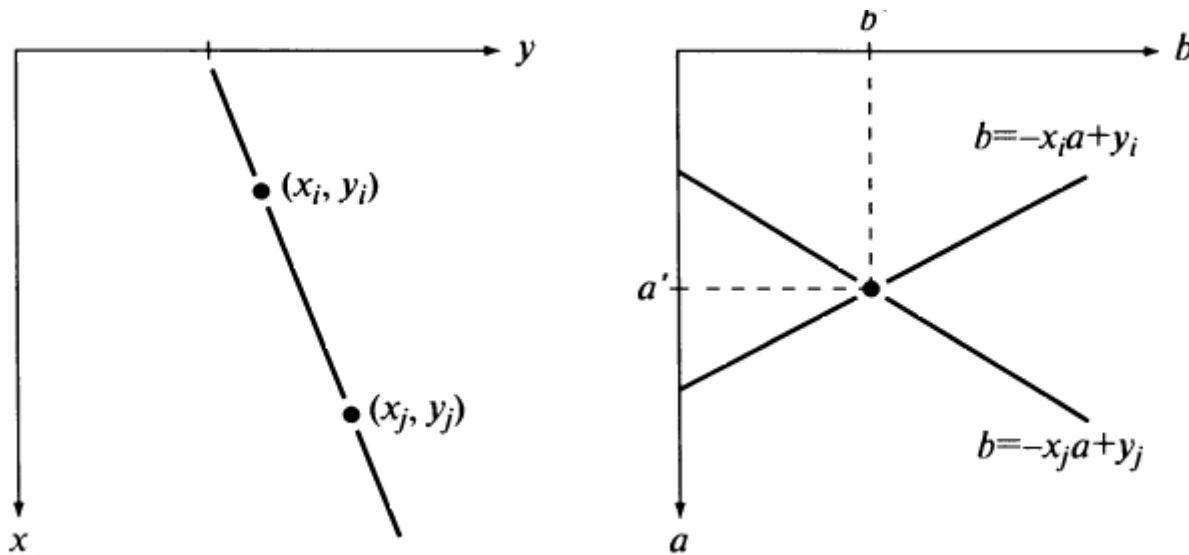
Нарушение условия «не пересекает уже существующие границы на изображении» (слева) и пример устранения разрыва (справа)

# Устранение разрывов



Процесс связывания для одиночной (непарной) точки разрыва

# Глобальный анализ с помощью преобразования Хафа



(а) Плоскость  $xу$ . (б) Пространство параметров.

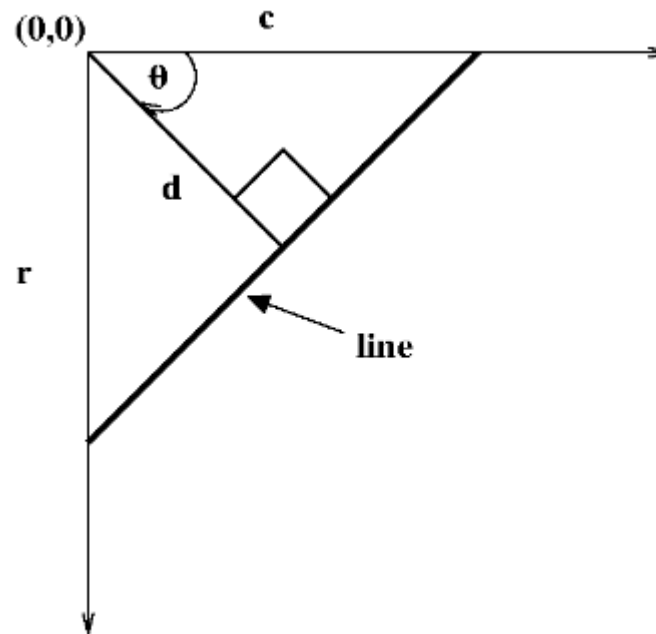
- Хаф [Hough, 1962]

# Обнаружение отрезков прямых

$$d = c \cdot \cos\theta + r \cdot \sin\theta$$

$$d = x \cdot \cos\theta + y \cdot \sin\theta$$

$$F(d, \theta, x, y) = x \cos\theta + y \sin\theta - d$$

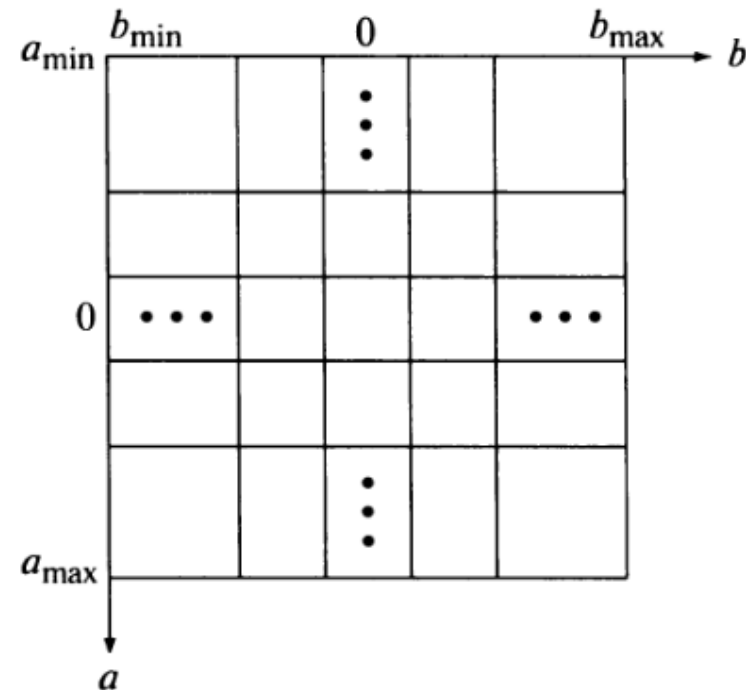


# Основная идея преобразования Хафа

Семейство кривых на плоскости, заданное параметрическим уравнением:  
 $F(a_1, a_2, \dots, a_n, x, y) = 0$ ;

$a_1, a_2, \dots, a_n$  – параметры семейства кривых образуют фазовое пространство

- Аккумуляторный массив
- Список координат всех пикселей, которые внесли вклад в значение соответствующих элементов аккумуляторного массива

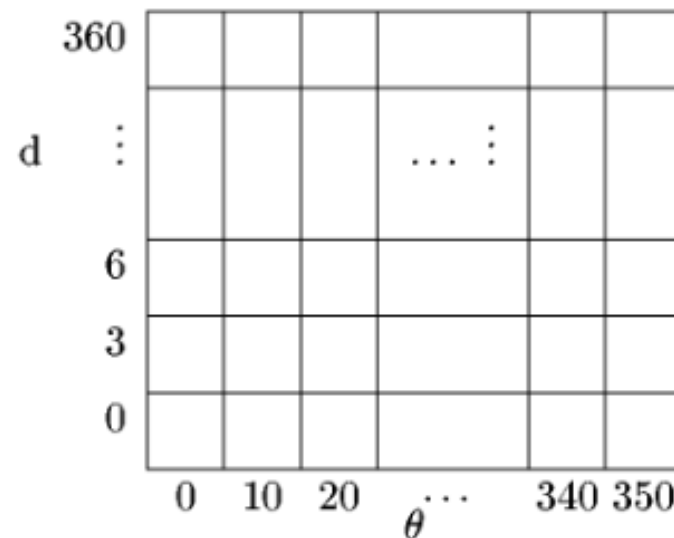


# Разбиение плоскости параметров

$$x \cos\theta + y \sin\theta = d, \text{ где } \theta_i \leq \theta \leq \theta_{i+1}, d_i \leq d \leq d_{i+1}$$

O'Gorman & Clowes

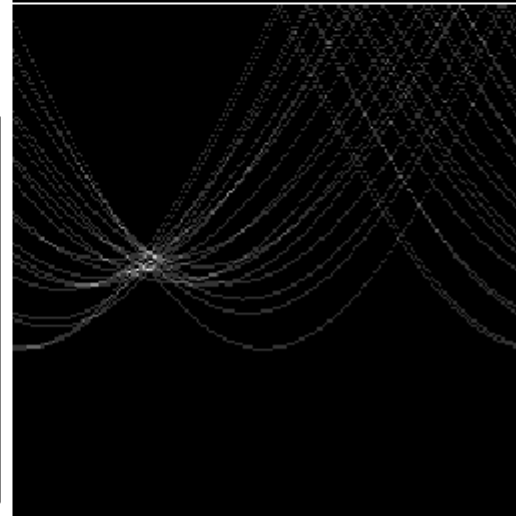
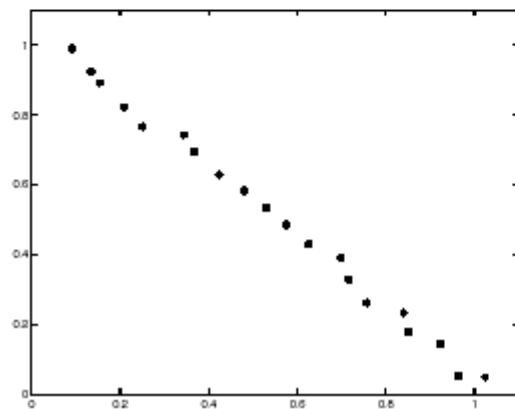
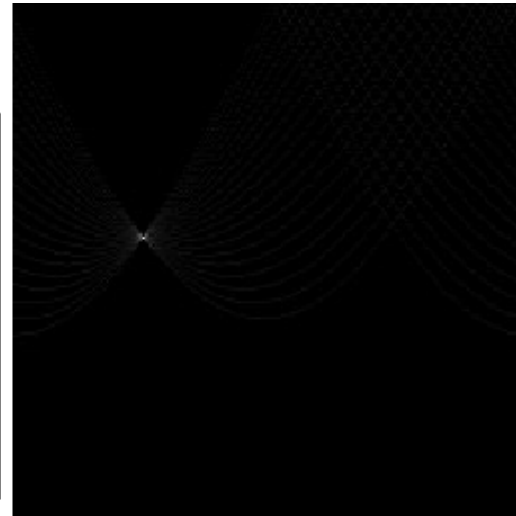
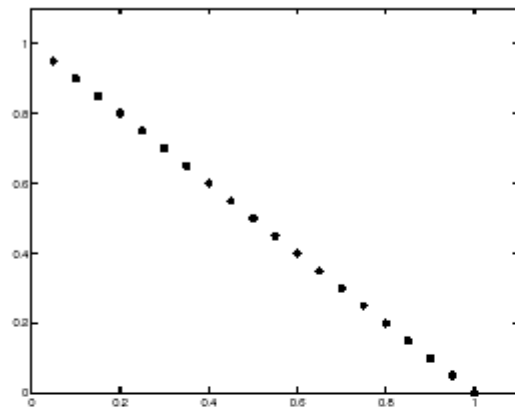
1976



# Алгоритм поиска прямой на изображении

- обнулить счетчики всех ячеек;
- для каждой точки интереса;
- 
- для каждой прямой, проходящей через данную точку;
- 
- увеличить соответствующий счетчик;
- выбрать ячейку с максимальным значением счетчика;
- параметры прямой, проходящей через максимальное число точек принять равным координатам центра выбранной ячейки в фазовом пространстве.

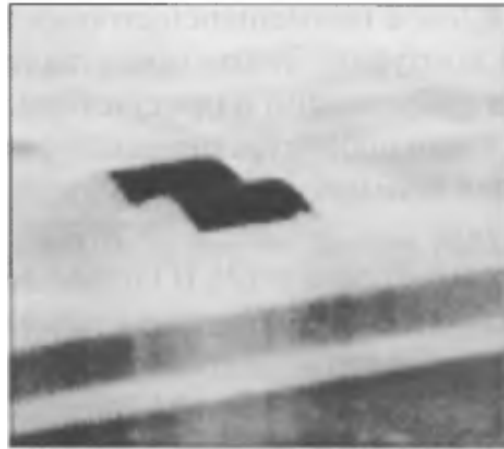
# Пример преобразования Хафа



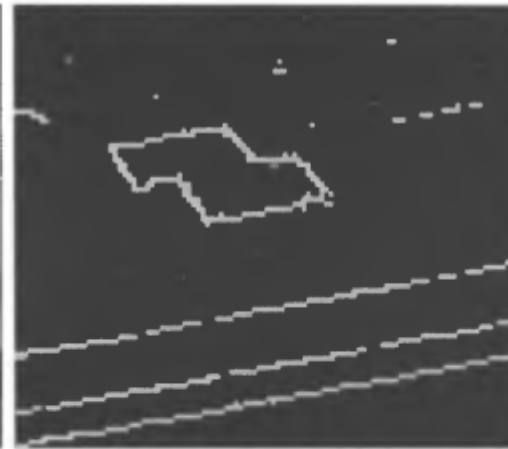


# Пример

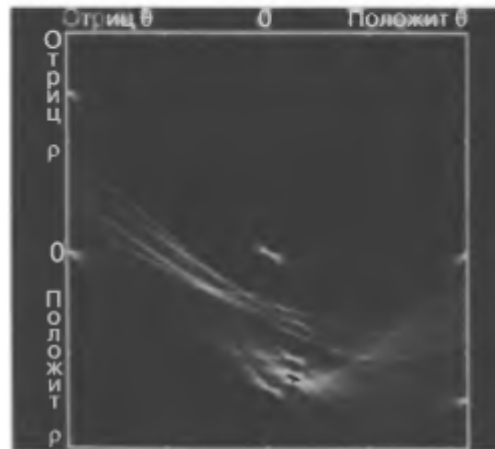
Исходное



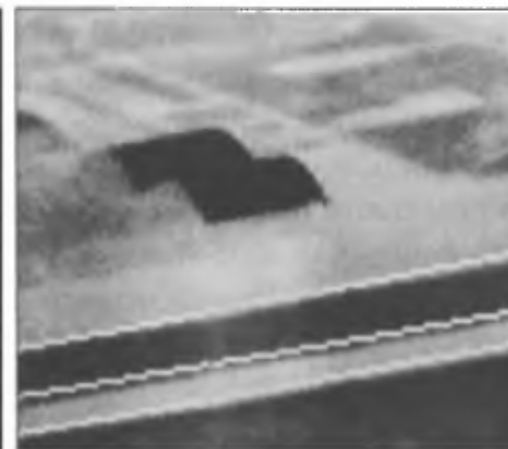
Градиентное



Хафа



Связанные  
пиксели



# Определитель прямых Барнса (1986)

Разработан для обнаружения прямых линий на сложных изображениях естественных сцен

1. Вычисление величины и направление градиента для каждого пиксела изображения
2. Точкам с достаточной большой величиной градиента присваиваются две метки, соответствующие двум различным дискретным значениям направления градиента. Результаты маркировки сохраняются в виде двух маркированных изображений

0 to 44, 45 to 90, 91 to 134, etc.

-22 to 22, 23 to 67, 68 to 112, etc.

# Определитель прямых Барнса (1986)

3. Поиск связных компонент на каждом маркированном изображении и вычисление длины отрезков каждой компоненты:
  1. Каждый пиксел принадлежит двум компонентам, по одной на каждом маркированном изображении
  2. Каждый пиксел используется для голосования за свою длиннейшую компоненту
  3. Счетчик каждой компоненты равен количеству проголосовавших за нее пикселей
  4. Выполняется выбор компонентов (отрезков прямых) с наибольшими значениями счетчиков

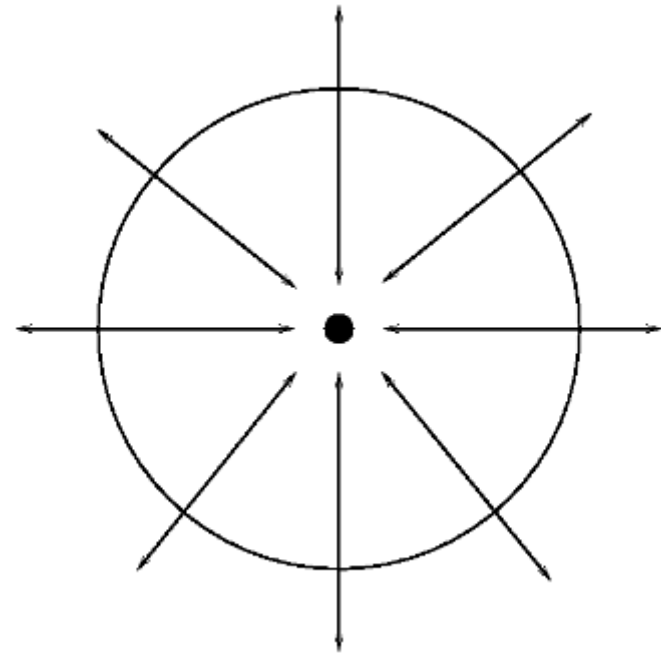
# Чеширский кот не так уж и прямолинеен



## Выделение окружностей на изображении

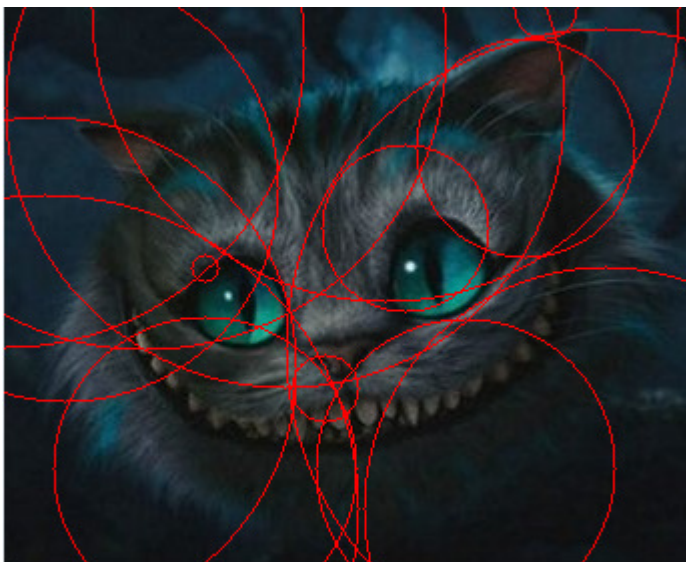
$$(x - a)^2 + (y - b)^2 = R^2$$

$$F(a, b, R, x, y) = (x - a)^2 + (y - b)^2 - R^2$$



[Robyn Owens](http://www.dai.ed.ac.uk/CVonline/LOCAL_COPIES/OWENS/LECT6/node3.html), "Computer Vision IT412, Lecture 6",  
[http://www.dai.ed.ac.uk/CVonline/LOCAL\\_COPIES/OWENS/LECT6/node3.html](http://www.dai.ed.ac.uk/CVonline/LOCAL_COPIES/OWENS/LECT6/node3.html), 1997

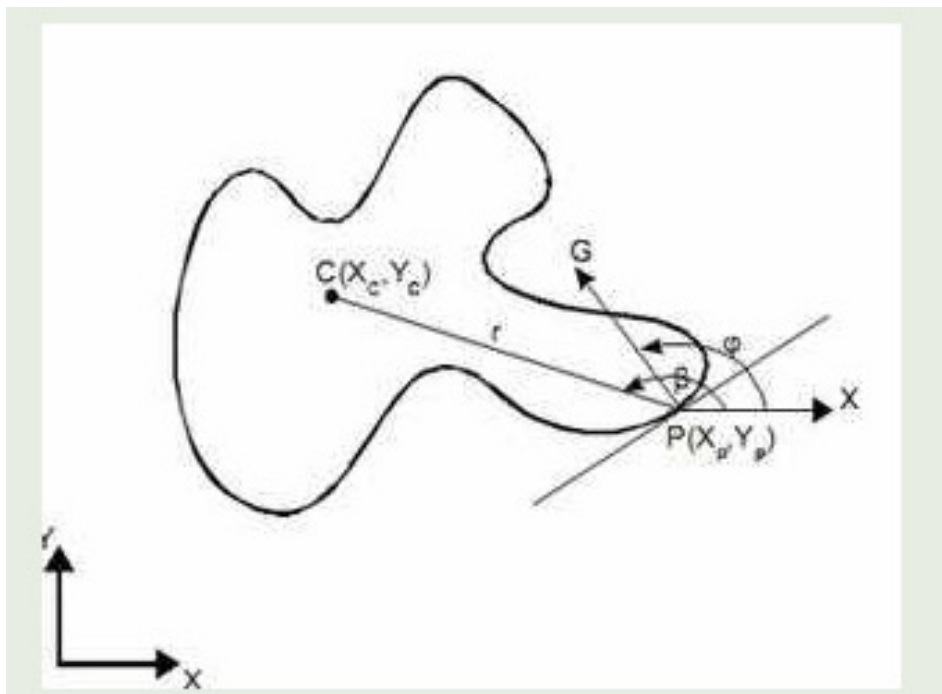
# Чеширский кот любит окружности



# Обобщенное преобразование Хафа (Ballard 1981)

1. Инициализация аккумуляторного массива  $A[a]$  нулевыми значениями
2. Для каждого краевого пиксела определяется такой вектор  $a$ , что  $f(x,a)=0$  и выполняется увеличение значения соответствующего элемента массива  $A[a]= A[a] +1$
3. Локальные максимумы в аккумуляторном массиве  $A$  соответствуют вероятным кривым  $f$  на изображении

# Вычисление параметров граничных точек для занесения их в R-таблицу



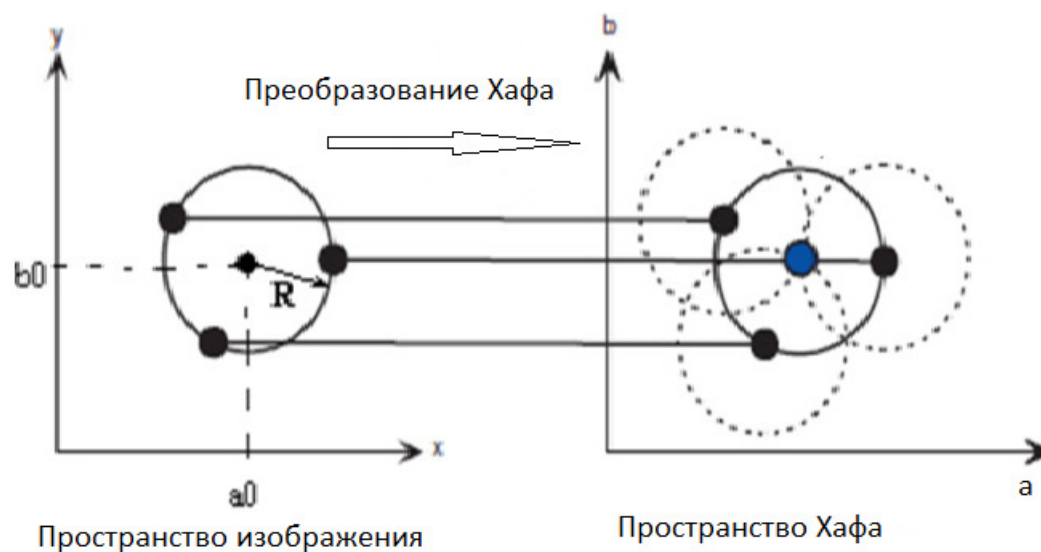
Выбирается произвольная точка отсчета для объекта и для каждого граничного пикселя объекта вычисляется направление градиента  $\varphi$  так же как длина  $r$  и направление  $\beta$  вектора, соединяющего этот граничный пиксель и точку отсчета

$$\begin{cases} x_p = x_c - r \cdot \cos(\beta) \\ y_p = y_c - r \cdot \sin(\beta) \end{cases}$$

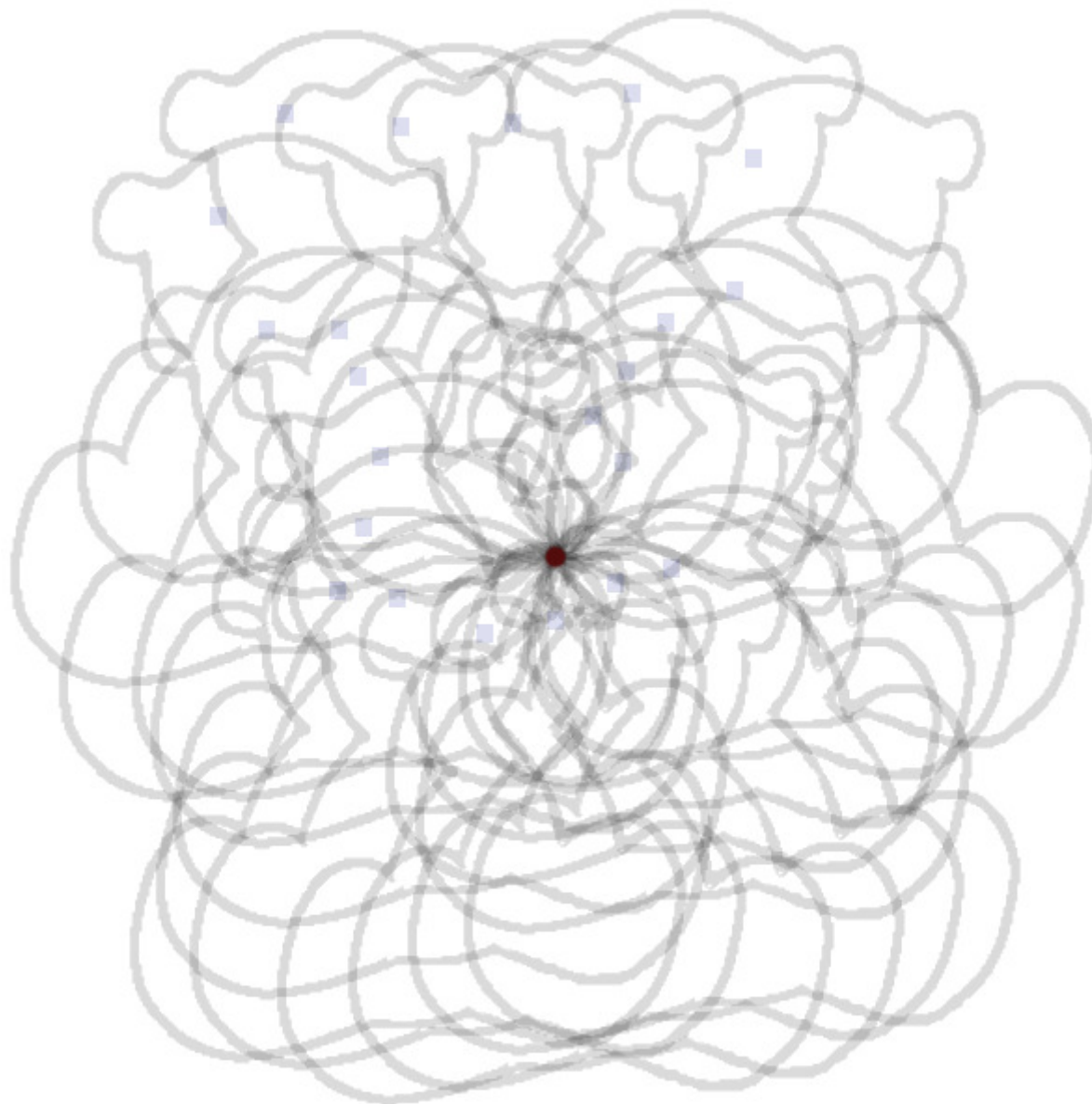
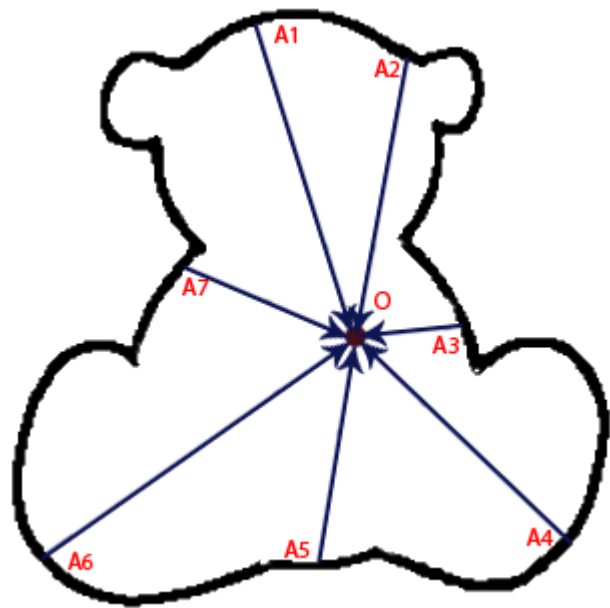


# Модифицированное преобразование Хафа для произвольных объектов методом окружностей

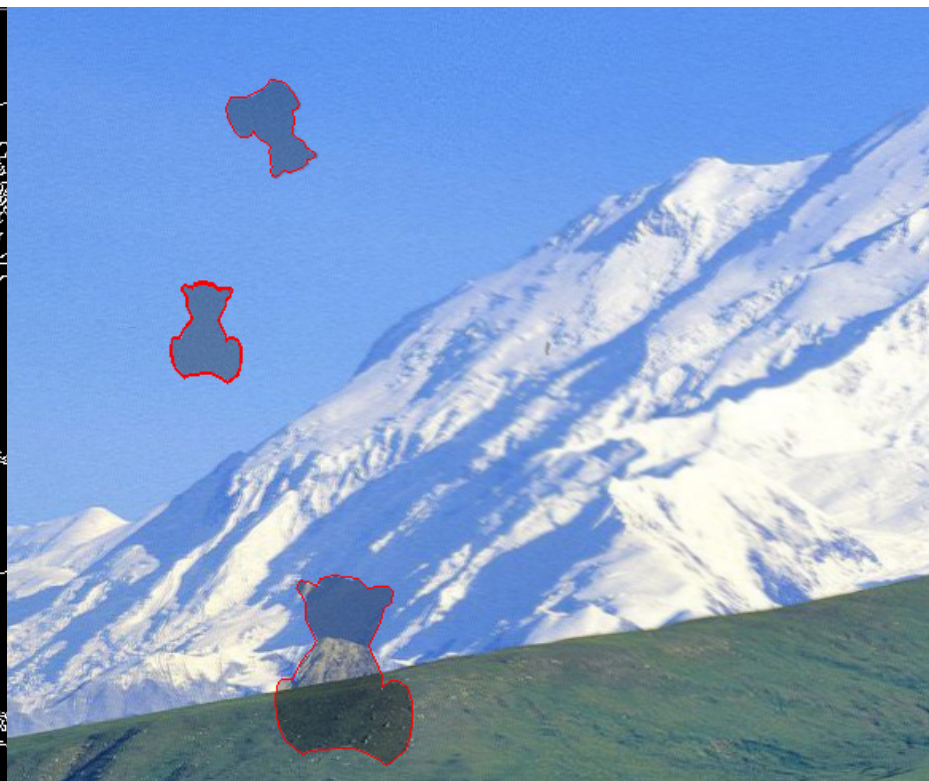
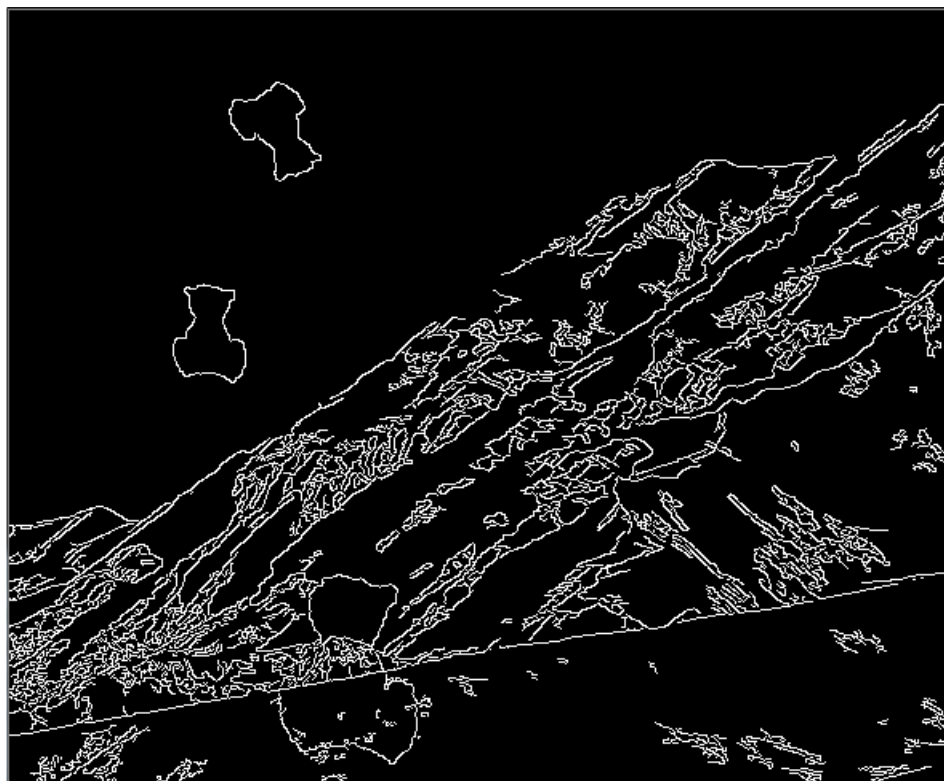
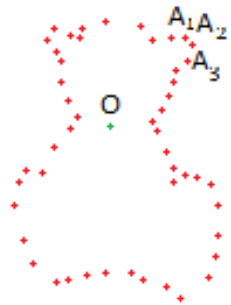
- Если известен радиус окружности, которую требуется детектировать, то число свободных параметров сокращается до двух – координаты центра окружности, то есть пространство Хафа является двухмерным.



# Метод окружностей



# Метод окружностей



# Сравнение с точки зрения быстродействия

Размер изображения	Классическое преобразование Хафа	Метод "окружностей"
200·154	113853 ms	6304 ms
300·231	212492 ms	11026 ms
400·308	315489 ms	16593 ms

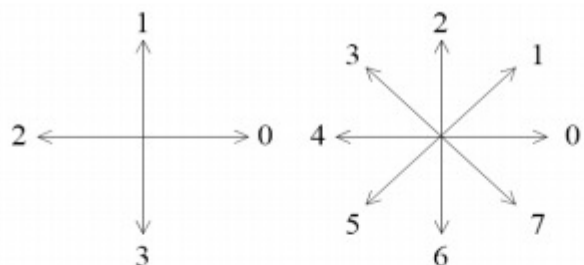
¶

# План

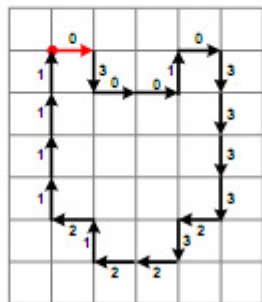
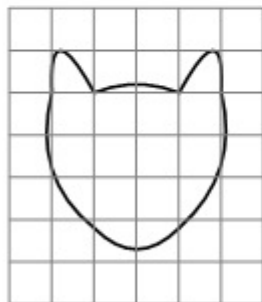
- Прослеживание контуров
- Связывание контуров
- **Анализ контуров**
- Метод активных контуров

# Цепной код Фримана

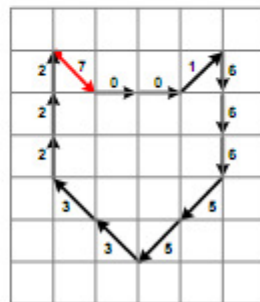
Нумерация направлений для 4-связного и 8-связного цепных кодов:



Пример:



А



Б

А: 03001033332322121111

Б: 70016665533222



Инвариантность к выбору  
начальной точки: минимальный код

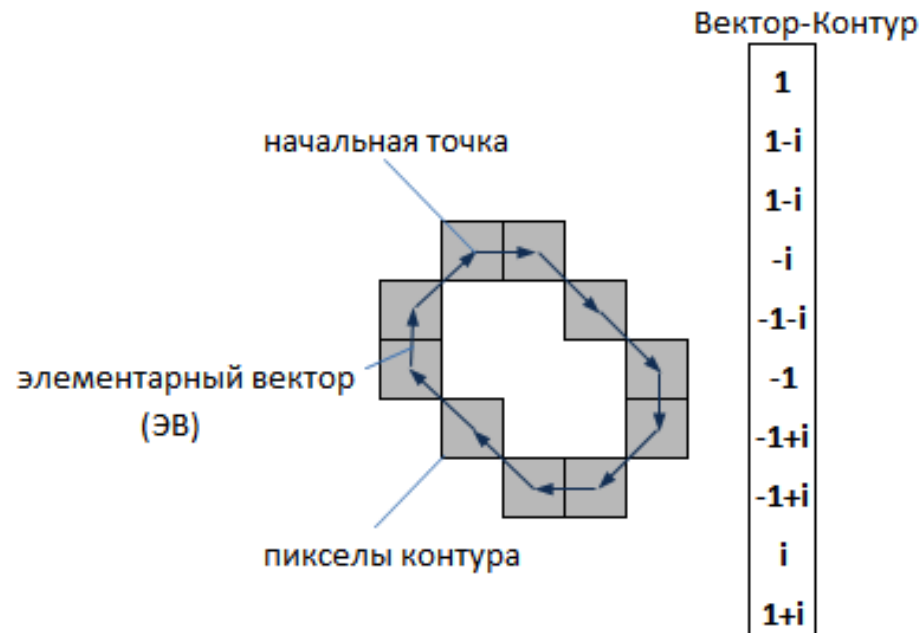
70016665533222 -> 00166655332227



Инвариантность к повороту:  
разности цифр кода

00166655332227 -> 01500706070051

# Кодирование контура



$$\Gamma = (y_0, y_1, \dots, y_{k-1})$$

# Свойства контуров

- Сумма ЭВ замкнутого контура равна нулю.
- Контур-вектор не зависит от параллельного переноса исходного изображения.
- Поворот изображения на определенный угол равносителен повороту каждого ЭВ контура на тот же угол.
- Изменение начальной точки ведет к циклическому сдвигу ВК.
- Изменение масштаба исходного изображения можно рассматривать как умножение каждого ЭВ контура на масштабный коэффициент.



# Скалярное произведение контуров

$$\eta = (\Gamma, N) = \sum_{n=0}^{k-1} (\gamma_n, \nu_n) \quad (1)$$

скалярное произведение элементарных контуров

$$(a + ib, c + id) = (a + ib)(c - id) = ac + bd + i(bc - ad) \quad (2)$$

скалярное произведение векторов

$$((a, b), (c, d)) = ac + bd \quad (3)$$

## Нормированное скалярное произведение (НСП)

$$\eta = \frac{(\Gamma, N)}{|\Gamma||N|} \quad (4)$$

$$|\Gamma| = \left( \sum_{n=0}^{k-1} |\gamma_n|^2 \right)^{\frac{1}{2}} \quad (5)$$

максимально возможное значение модуля НСП = 1

в случае если









$$\Gamma = \mu N \quad (6)$$

При умножении комплексных чисел, их модули(длины) перемножаются, а аргументы(углы) – складываются.

Значит контур  $\mu N$  это тот же контур  $N$ , но повернутый и промасштабированный.

Масштаб и поворот определяется комплексным числом  $\mu$ .

## Свойства НСП

		НСП	$\text{Re}(\text{НСП})=\cos(a)$	$ \text{НСП} $
	x 	<b>1</b>	<b>1</b>	<b>1</b>
	x 	<b>i</b>	<b>0</b>	<b>1</b>
	x 	<b>-1</b>	<b>-1</b>	<b>1</b>
	x 	<b>-i</b>	<b>0</b>	<b>1</b>

# Проблема

$$\Gamma = \mu N \quad (6)$$

равенство (6) достигается только если начальные точки контуров – совпадают

# Взаимокорреляционная функция (ВКФ) двух контуров

$$\tau(m) = (\Gamma, N^{(m)}), \quad m = 0, \dots, k - 1 \quad (7)$$

Где  $N^{(m)}$  — контур, полученный из  $N$  путем циклического сдвига его ЭВ на  $m$  элементов.

ВКФ является периодической, с периодом  $k$

если  $N=(n_1, n_2, n_3, n_4)$ ,  
то  $N(1)=(n_2, n_3, n_4, n_1)$ ,  
 $N(2)=(n_3, n_4, n_1, n_2)$   
...

## Мера похожести двух контуров

$$\tau_{max} = \max \left( \frac{\tau(m)}{|\Gamma||N|} \right), \quad m = 0, \dots, k - 1 \quad (8)$$

Инвариантна

- переносу,
- масштабированию,
- вращению
- и сдвигу начальной точки

## Автокорреляционная функция (АКФ)

$$v(m) = (\Gamma, \Gamma^{(m)}), \quad m = 0, \dots, k - 1 \quad (9)$$

## Некоторые свойства АКФ

1. АКФ не зависит от выбора начальной точки контура
2. Модуль АКФ симметричен относительно центрального отсчета  $k/2$

$$\text{АКФ}(0)=(n_1,n_1)+(n_2,n_2)+(n_3,n_3)+(n_4,n_4)$$

$$\text{АКФ}(1)=(n_1,n_2)+(n_2,n_3)+(n_3,n_4)+(n_4,n_1)$$

$$\text{АКФ}(2)=(n_1,n_3)+(n_2,n_4)+(n_3,n_1)+(n_4,n_2)$$

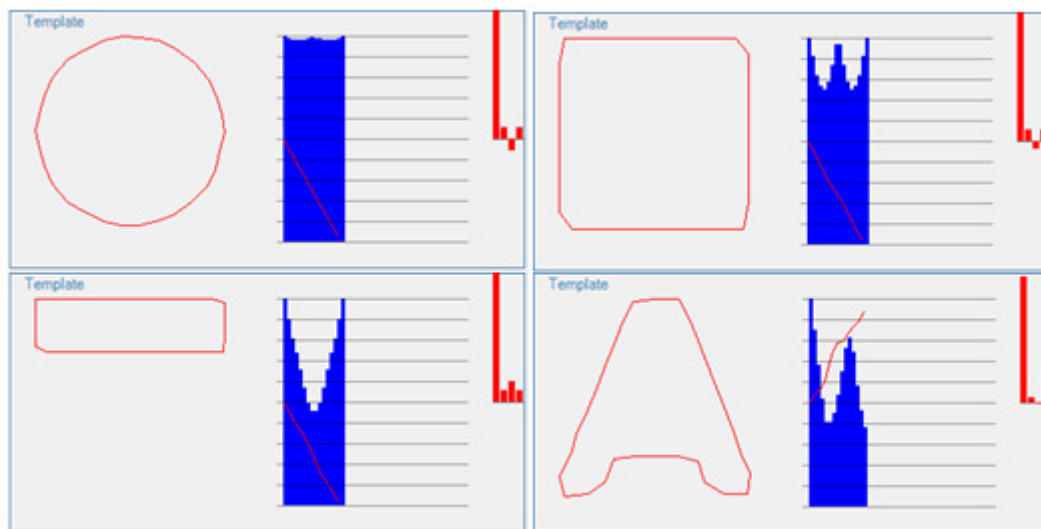
$$\text{АКФ}(3)=(n_1,n_4)+(n_2,n_1)+(n_3,n_2)+(n_4,n_3)$$

$$\text{АКФ}(4)=(n_1,n_1)+(n_2,n_2)+(n_3,n_3)+(n_4,n_4)$$



## Некоторые свойства АКФ

3. Если контур имеет какую-либо симметрию относительно поворота, то аналогичную симметрию имеет его АКФ.
4. АКФ контура в определенном смысле можно считать характеристикой формы контура.



## Некоторые свойства АКФ

5. Нормированная АКФ не зависит от масштаба, положения, вращения и выбора начальной точки контура.

## Эквализация контуров

- Если исходный контур больше необходимого, то перебираем все его ЭВ, и считаем элементы N как сумму всех ЭВ, следующим образом:

```
Complex[] newPoint = new Complex[newCount];
```

```
for (int i = 0; i < oldCount; i++)  
    newPoint[i * newCount / oldCount] += oldPoint[i];
```

## Эквализация контуров

- Если же исходный контур меньше  $k$ , то производим интерполяцию:

```
Complex[] newPoint = new Complex[newCount];  
for (int i = 0; i < newCount; i++) {  
    double index = 1.0 * i * oldCount / newCount;  
    int j = (int)index;  
    double k = index - j;  
    newPoint[i] = oldPoint[j] * (1 - k) + oldPoint[j + 1] * k;  
}
```

# Практическое применение контурного анализа

## Общий алгоритм распознавания

- Предварительная обработка изображения — сглаживание, фильтрация помех, повышение контраста.
- Бинаризация изображения и выделение контуров объектов.
- Начальная фильтрация контуров по периметру, площади, коэффициенту формы, фрактальности и так далее.
- Приведение контуров к единой длине, сглаживание.
- Перебор всех найденных контуров, поиск шаблона, максимально похожего на данный контур.

# Сложность

ВКФ требует вычислений порядка  $O(k^2)$ , где  $k$  – длина контура

Полное время поиска шаблона для *отдельно взятого контура* можно оценить как  $O(k^2t)$ , где  $t$  – число шаблонных контуров

Идентификации *всех контуров изображения*  $O(n^2k^2t)$ , где  $n$  – линейный размер изображения,  $k$  – длина контура,  $t$  – число шаблонов

# Дескриптор контура

Для быстрого поиска шаблонов необходимо ввести некий дескриптор, характеризующий форму контура

В качестве дескриптора, описывающего форму контура можно выбрать АКФ

Если база шаблонов хранит их АКФ, то поиск шаблона для контура, путем сравнения АКФ, составит  **$O(kt)$**

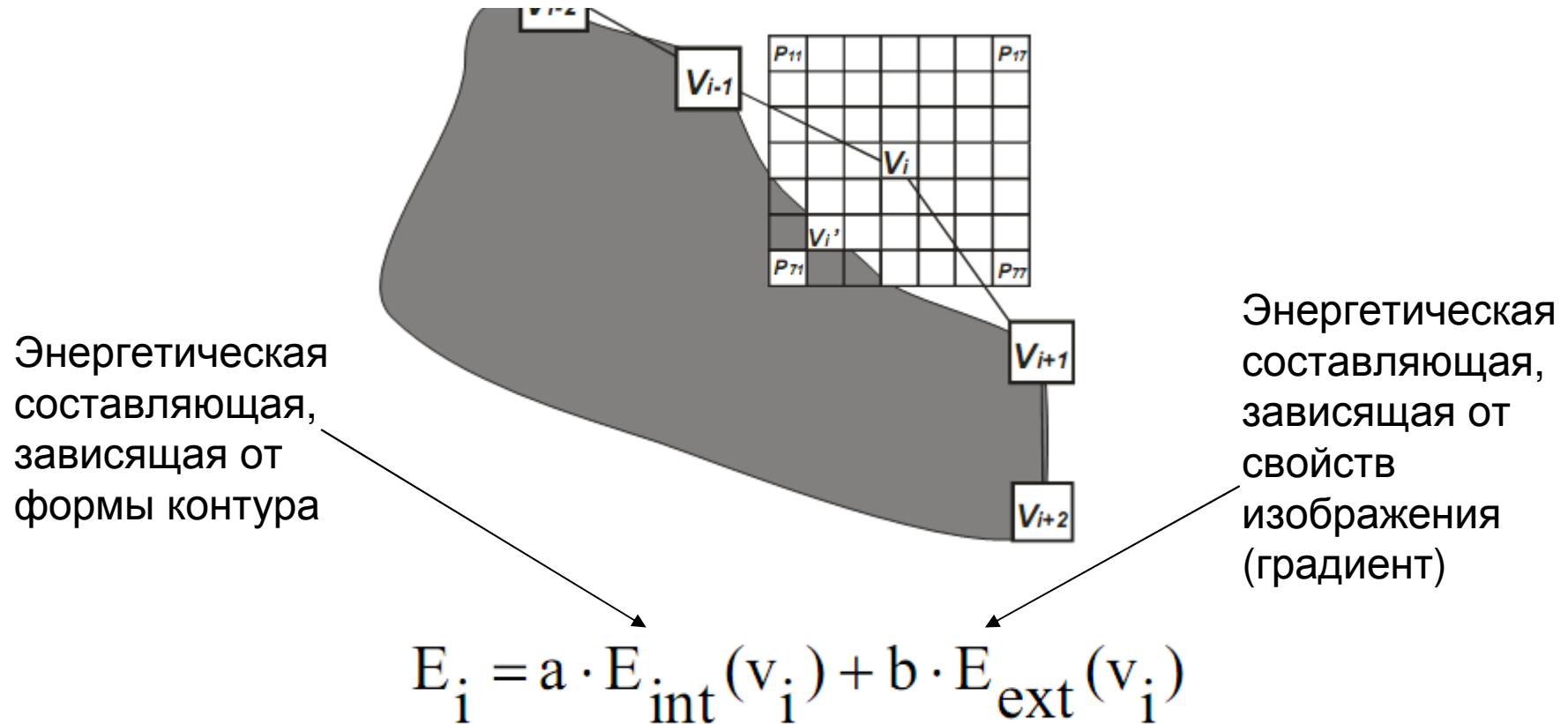
# План

- Прослеживание контуров
- Связывание контуров
- Анализ контуров
- **Метод активных контуров**



# Активный контур

$$V = \{v_1, \dots, v_n\}, \text{ где } v_i = (x_i, y_i), i = \{1, \dots, n\}$$



# Внутренняя энергия

Внутренняя энергия = сглаживающая + распирающая

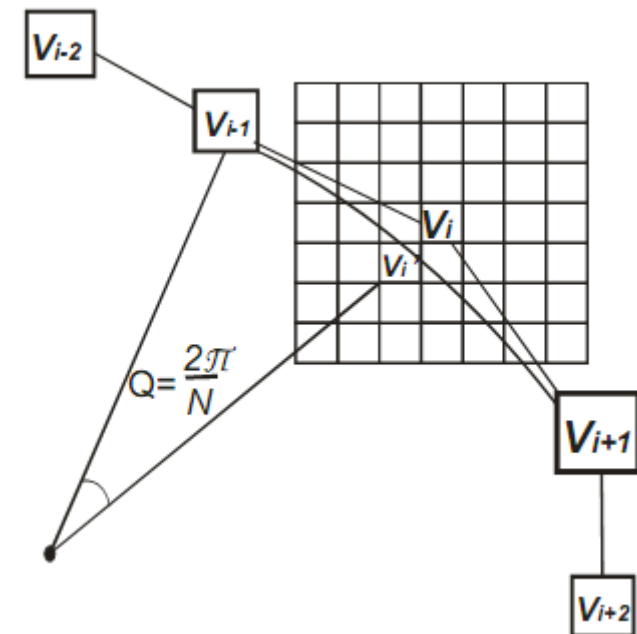
$$a \cdot E_{\text{int}}(v_i) = c \cdot E_{\text{con}}(v_i) + d \cdot E_{\text{bal}}(v_i)$$

## Сглаживающая составляющая

$$e_{jk}(v_i) = \frac{1}{l(V)} \left\| p_{jk}(v_i) - \gamma(v_{i-1} + v_{i+1}) \right\|^2$$

$$\gamma = \frac{1}{2 \cos\left(\frac{2\pi}{n}\right)}$$

$$l(V) = \frac{1}{n} \sum_{i=1}^n \|v_{i+1} - v_i\|^2$$

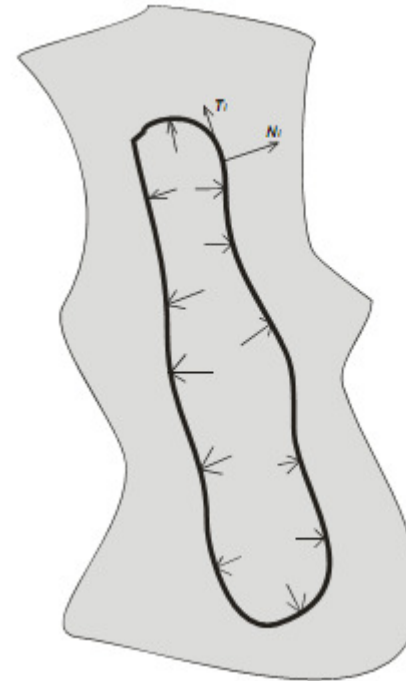


## Распирающая составляющая

$$e_{jk}(v_i) = n_i \cdot (v_i - p_{jk}(v_i))$$

$n_i$  – перпендикуляр к  $t_i$

$$t_i = \frac{v_i - v_{i-1}}{\|v_i - v_{i-1}\|} + \frac{v_{i+1} - v_i}{\|v_{i+1} - v_i\|}$$



# Внешняя энергия

Внешняя энергия = энергия изображения + энергия градиента

$$b \cdot E_{\text{ext}}(v_i) = m \cdot E_{\text{mag}}(v_i) + g \cdot E_{\text{grad}}(v_i)$$

Энергия изображения

$$e_{jk} = I(p_{jk}(v_i))$$

Энергия градиента

$$e_{jk}(v_i) = -\left| \sqrt{I(p_{jk}(v_i))} \right|$$

# Дополнительные энергии

- Энергия схожести по яркости
- Энергия, соответствующая расстоянию от центра масс фигуры