

# Системы компьютерного зрения

## Лекция 2

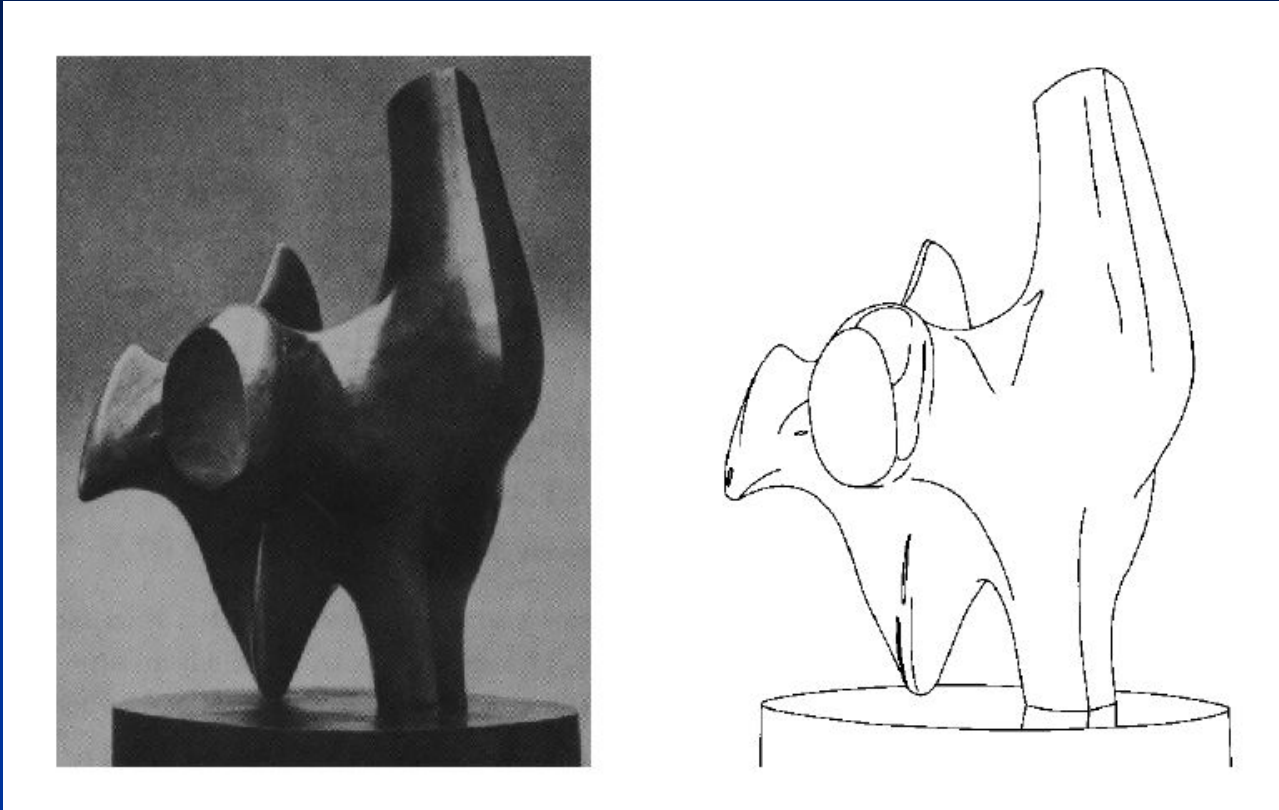
На основе лекций «Введение в компьютерное зрение»  
в МГУ, авторы - В.Вежнев, А.Конущин, В.Вежнев

# Анализ изображений



- Темы этой лекции
  - Выделение краев
    - Градиент изображения
    - Алгоритм Сэнны
  - Работа с контурами
    - Цепные коды
    - Полигональная аппроксимация
    - $\Psi$ -s представление
    - Дескрипторы контуров
  - Как проводить прикладное исследование?

# Выделение краев



Цель – преобразовать изображение в набор кривых для:

- выделения существенных характеристик
- сокращения объема информации для анализа

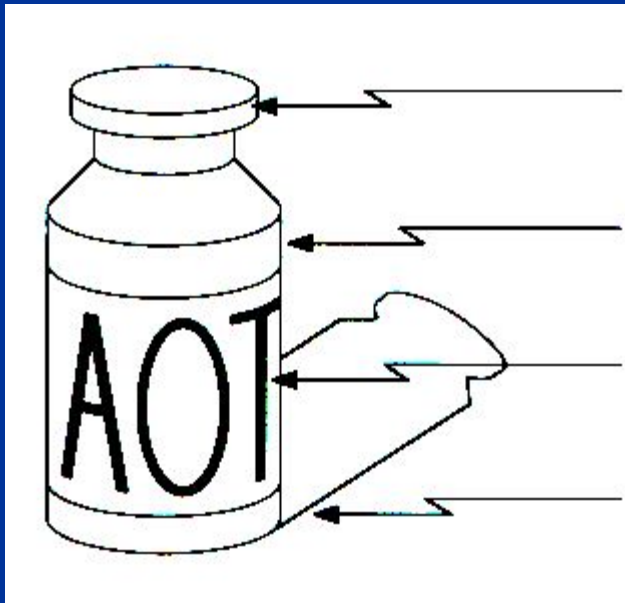
# Выделение краев





# Откуда берутся края?

- Край – резкий переход яркости.
- Различные причины возникновения:



Резкое изменение нормали поверхности

Резкое изменение глубины сцены

Резкое изменение цвета поверхности

Резкое изменение освещенности

# Как найти резкое изменение яркости?



Нас интересуют области резкого изменения яркости – нахождение таких областей можно организовать на основе анализа первой и второй производной изображения.

Рассмотрим одномерный случай...

График функции

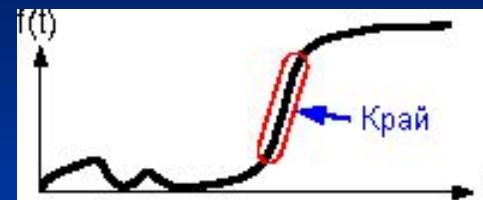


График производной

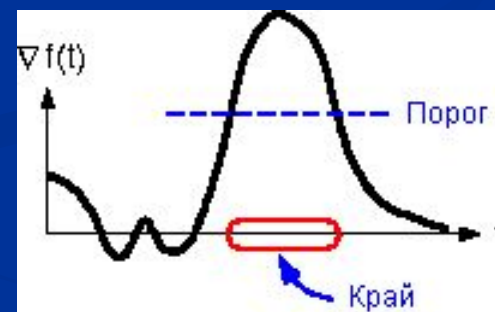
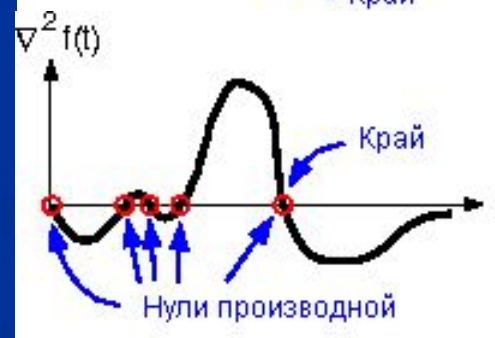


График 2ой производной



# Как найти резкое изменение яркости?



Известно, что наибольшее изменение функции происходит в направлении ее градиента. Величина изменения измеряется абсолютной величиной градиента.

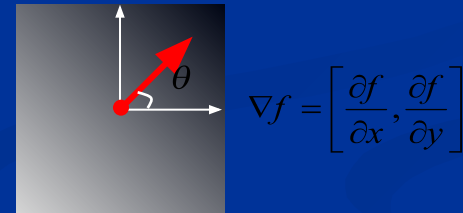
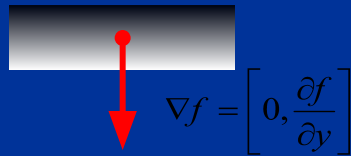
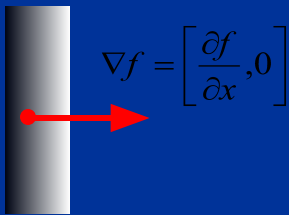
$$\nabla I(x, y) = \left( \frac{\partial I}{\partial x}(x, y), \frac{\partial I}{\partial y}(x, y) \right);$$

$$|\nabla I(x, y)| = \sqrt{\left( \frac{\partial I}{\partial x}(x, y) \right)^2 + \left( \frac{\partial I}{\partial y}(x, y) \right)^2}$$



# Градиент яркости изображения

- Известно, что наибольшее изменение функции происходит в направлении ее градиента
  - Приведем примеры...



$$\theta = \arctan \left( \frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}} \right)$$





# Градиент яркости изображения

- Направление градиента задается:

$$\theta = \arctan\left(\frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}}\right)$$

- «Направление края» задается перпендикулярным градиенту

- «Сила края» задается абсолютной величиной градиента:

$$|\nabla I(x, y)| = \sqrt{\left(\frac{\partial I}{\partial x}(x, y)\right)^2 + \left(\frac{\partial I}{\partial y}(x, y)\right)^2}$$

- Иногда используется приближенное вычисление градиента:

$$|\nabla I(x, y)| \cong \left|\frac{\partial I}{\partial x}(x, y)\right| + \left|\frac{\partial I}{\partial y}(x, y)\right|$$

# Вычисление градиента яркости изображения



Семейство методов основано на приближенном вычислении градиента, анализе его направления и абсолютной величины. Свертка по функциям:

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

Робертса

Превитт

Собеля

Математический смысл – приближенное вычисление производных по направлению

# Карта силы краев



Примеры:



Робертса



Превитт



Собеля

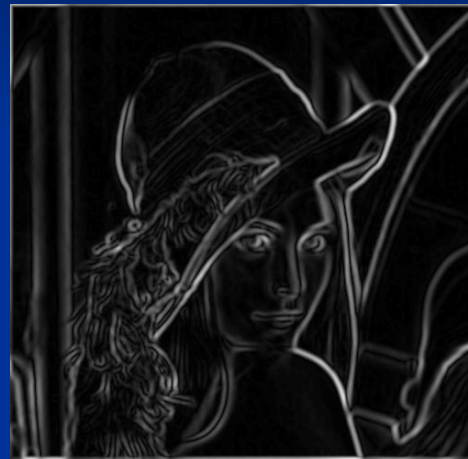
# Выделение краев



- Вычисление градиента – это еще не всё...



Исходное изображение



Карта силы краев

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

# Выделение краев



- Нужно:
  - Убрать слабые края и шум
  - Сделать края тонкими
  - Объединить пиксели краев в связные кривые



# Алгоритм Санны



- Давно придуман, однако до сих пор широко используется
- Шаги:
  - Убрать шум и лишние детали из изображения
  - Рассчитать градиент изображения
  - Сделать края тонкими (edge thinning )
  - Связать края в контура (edge linking)

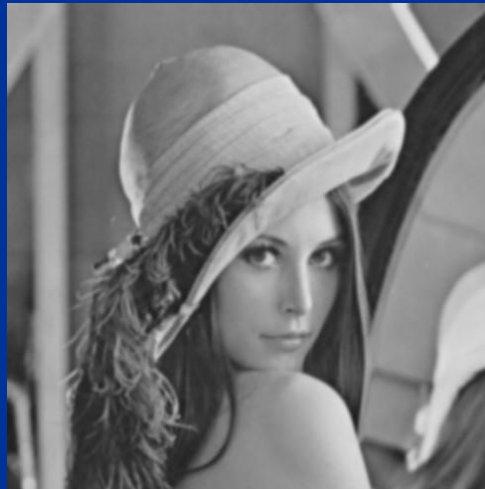
# Начало...



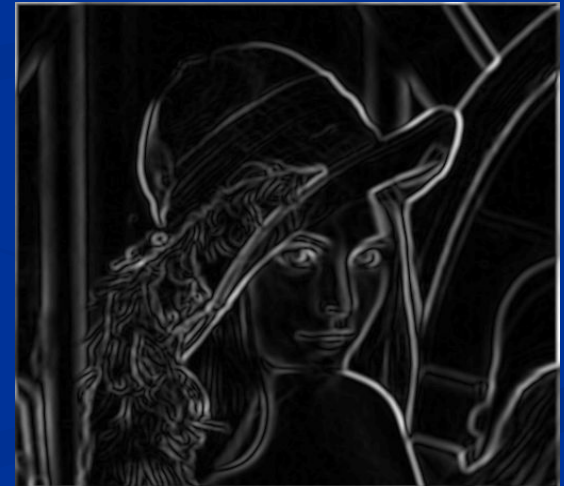
- Размыть изображение с помощью фильтра Гаусса
  - Убрать шум, лишние детали текстуры
- Рассчитать градиент изображения
  - Одним из операторов – например, Собеля



Исходное изображение



Размытое изображение

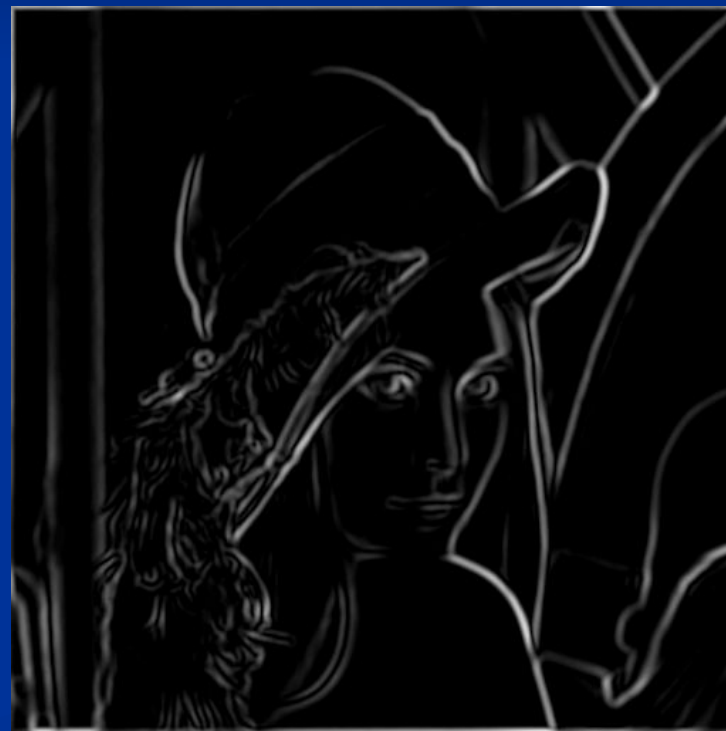
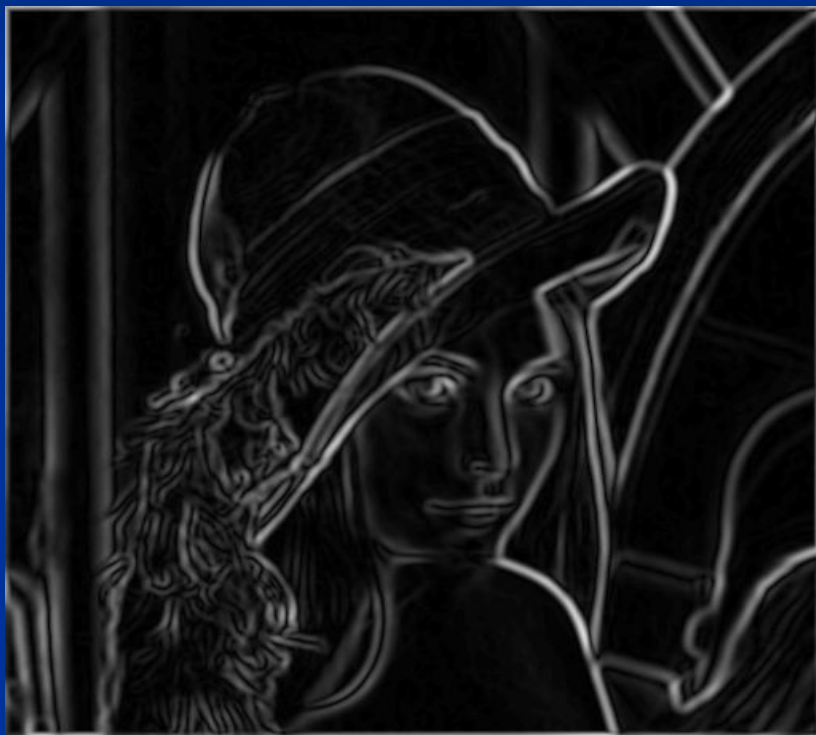


Карта силы краев

# ...середина...



- Все пиксели где сила краев  $< T$  убрать из рассмотрения

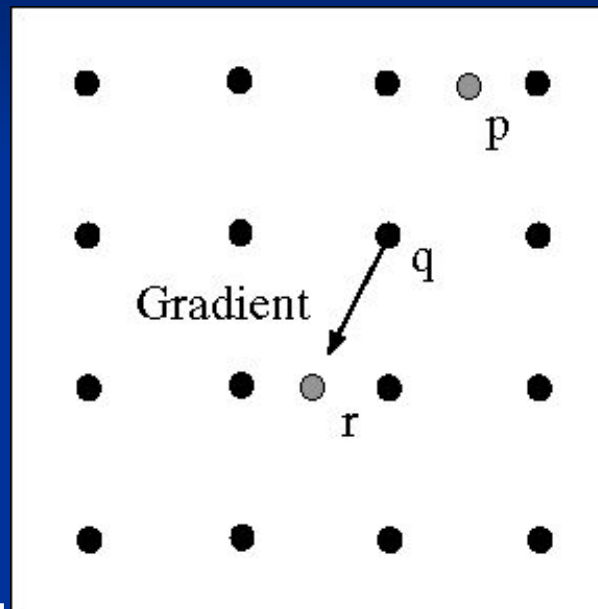




# ...середина...



- Поиск локальных максимумов



- Проверка – является ли пиксель локальным максимумом вдоль направления градиента
  - Приходится интерполировать «нецелые» пиксели  $p$  и  $r$

# ... финал

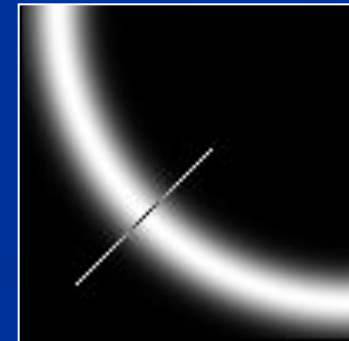
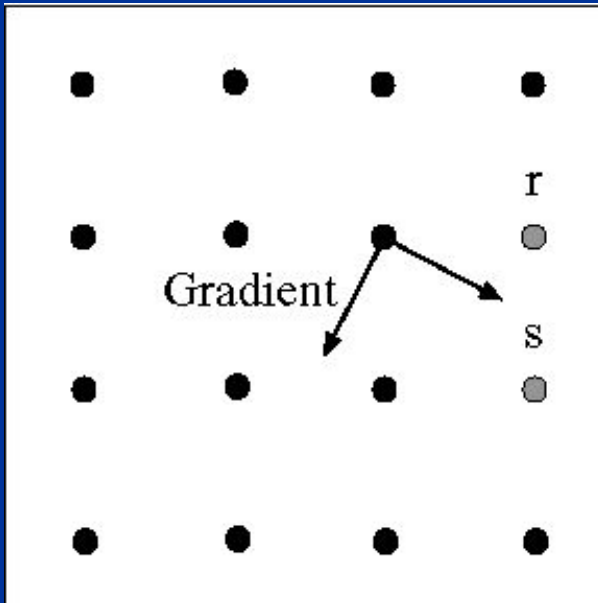


1. Выбираем еще не обработанную точку локального максимума  $p$  в которой сила края  $|\nabla I(p)| > T_1$
2. Прослеживание края выбранного локального максимума  $p$ :
  - a) Предсказание следующей точки края  $q$ ;
  - b) Проверка –  $|\nabla I(q)| > T_2$  ? ( $T_1 > T_2$ )
  - c) Если да –  $p = q$ , переход на начало шага 2;
  - d) Если нет - переход на шаг 1;

# Пояснения



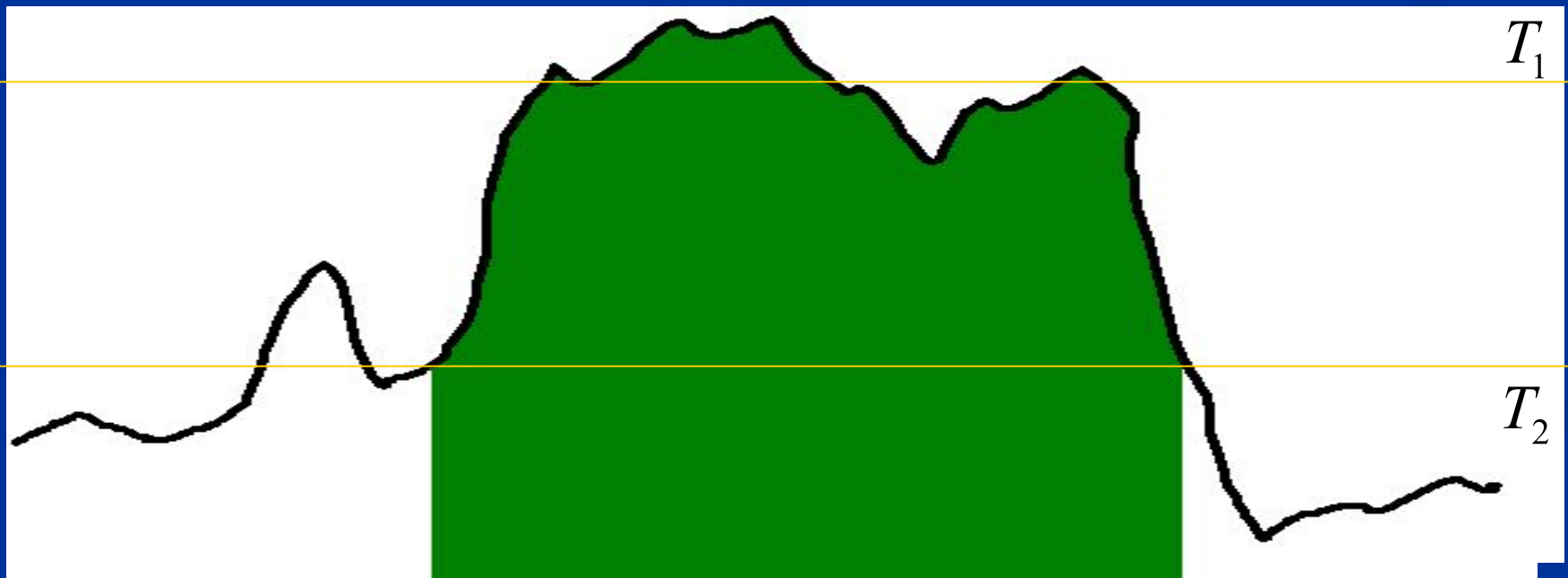
- Как предсказать следующую точку края?
  - От текущей точки шаг в сторону перпендикулярную градиенту
  - В данном случае – в точку  $r$  или  $s$



# Пояснения



1. Для чего используются два порога? ( $T_1 > T_2$ )
  1. Чтобы уменьшить влияние шума для инициализации кривой используем больший порог
  2. Чтобы «не потерять хвост» используем меньший порог при прослеживании

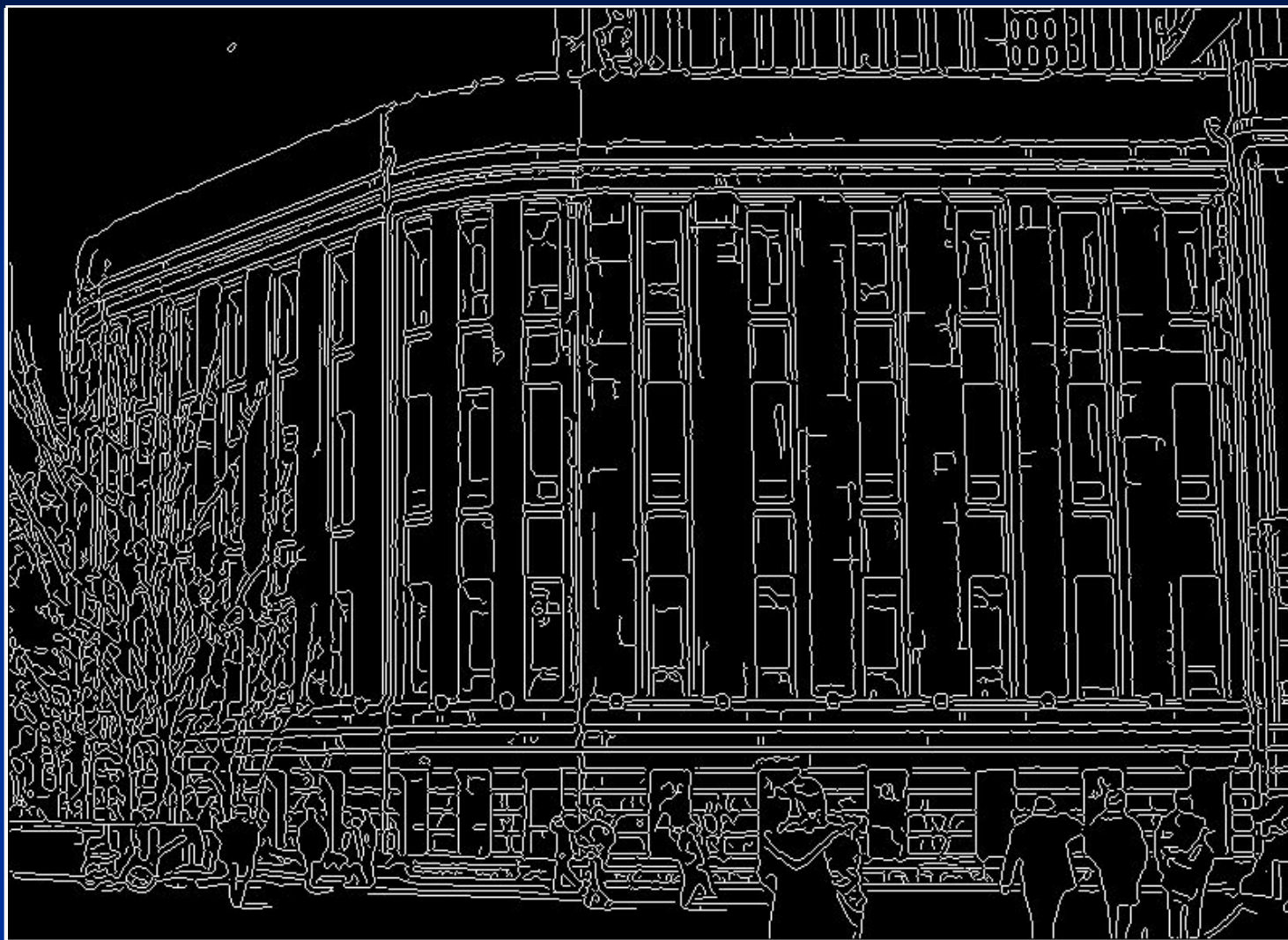


# Алгоритм Canny



- Размыть изображение фильтром Гаусса с некоторым  $\sigma$ 
  - Убрать шум, лишние детали текстуры
- Рассчитать градиент изображения
  - Одним из операторов – например, Собеля
- Все пиксели где сила краев  $< T$  убрать из рассмотрения
- Поиск локальных максимумов
- Прослеживание краев из точек локальных максимумов

# Sanny - результат



# Влияние $\sigma$ (параметр фильтра Гаусса)



исходное



Canny с



Canny с

# Вопрос



- ? Получив контур объекта (связный набор пикселей) – как его дальше анализировать?
  
- ! Вариант - нужно преобразовать контур в некоторое численное представление
  - Один из способов – использование цепных кодов



# Работа с контурами

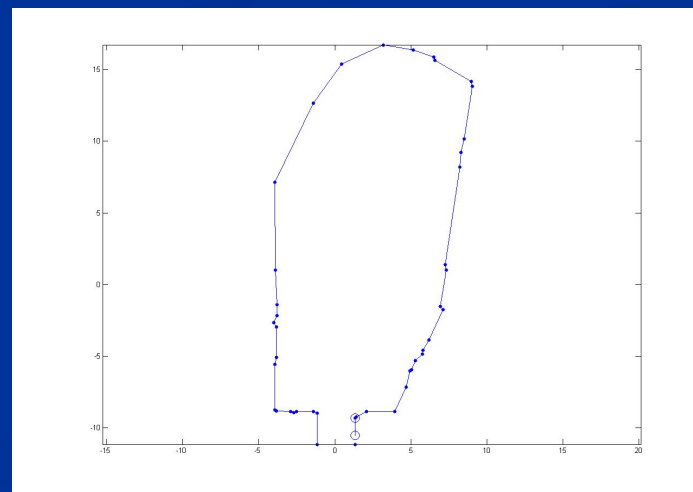
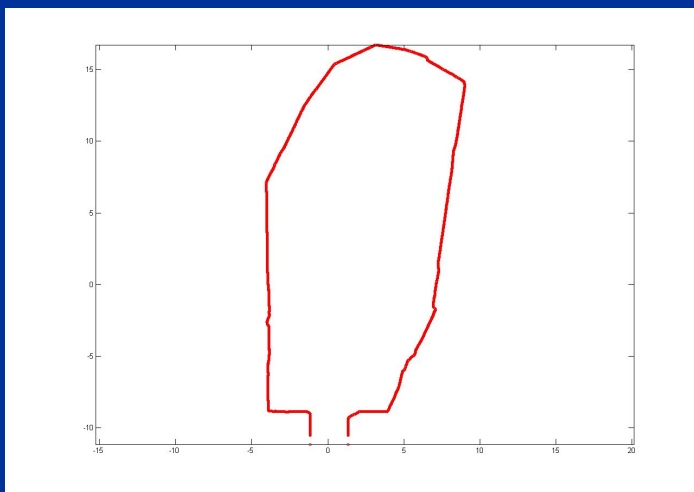


- Полигональная аппроксимация
- Цепные коды
- Дескрипторы контуров

# Полигональная аппроксимация



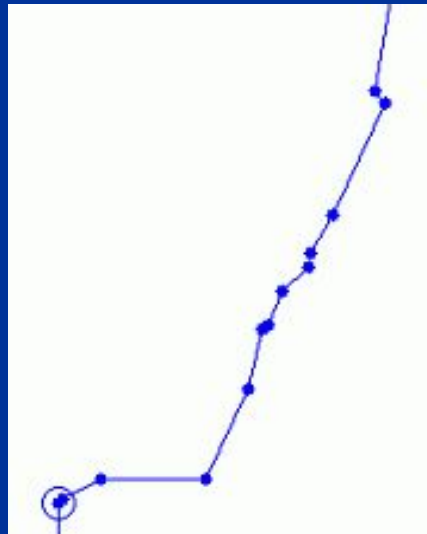
- Постановка:
  - Аппроксимация точечной кривой ломаной линией
- Цель:
  - Сжатие информации
  - Борьба с дискретностью и шумом
  - Облегчение дальнейшего анализа



# Полигональная аппроксимация



- Постановка:
  - Аппроксимация точечной кривой ломаной линией
- Цель:
  - Сжатие информации
  - Борьба с дискретностью и шумом
  - Облегчение дальнейшего анализа



# Алгоритм Дугласа-Пекера (рекурсивное подразбиение)



- Инициализация – начнем с прямой, идущей от начальной точки к конечной
  - Если контур замкнутый – выберем точки максимально удаленные друг от друга



# Алгоритм Дугласа-Пекера (рекурсивное подразбиение)



- Шаг №1 – найти точку контура максимально удаленную от прямой
  - Если расстояние от нее до прямой  $d < \varepsilon$  – разбиение завершено, если нет – к шагу 2



# Алгоритм Дугласа-Пекера (рекурсивное подразбиение)



- Шаг №2 – добавить узел к ломаной линии
  - Затем рекурсивно вызвать Шаг №1 для каждой из половинок ломаной



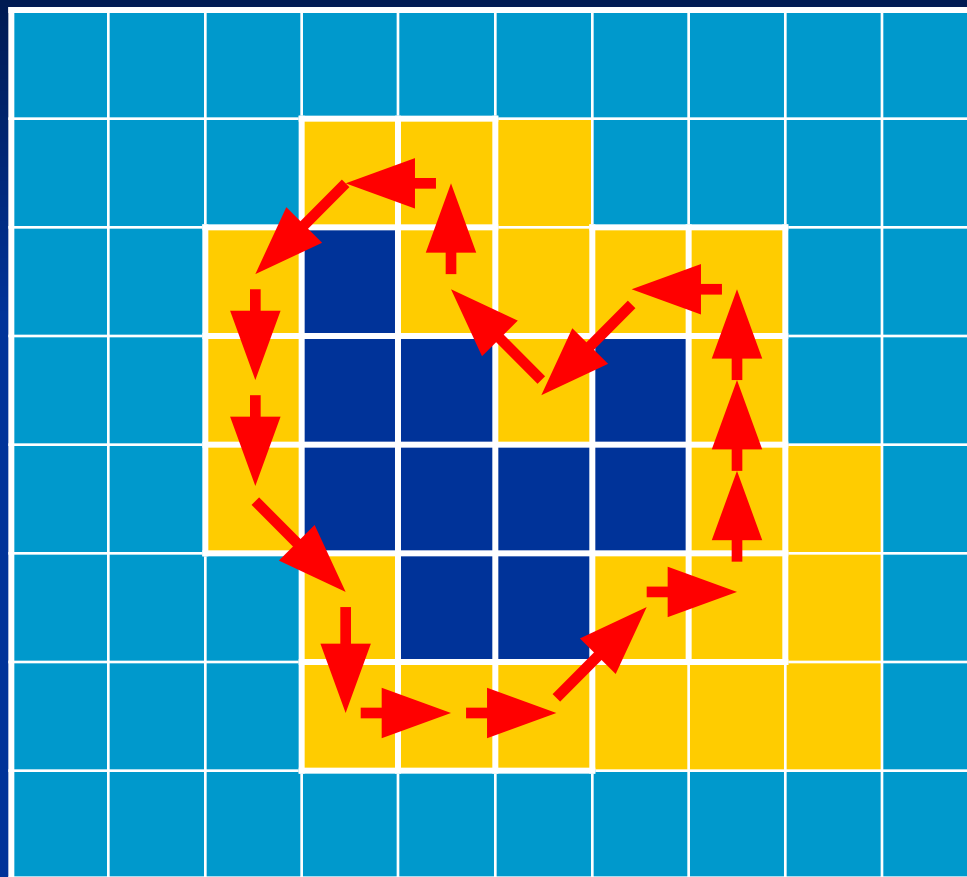
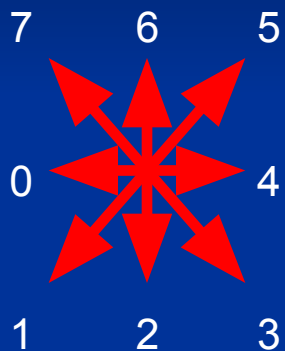
# Алгоритм Дугласа-Пекера (рекурсивное подразбиение)



- Результат:



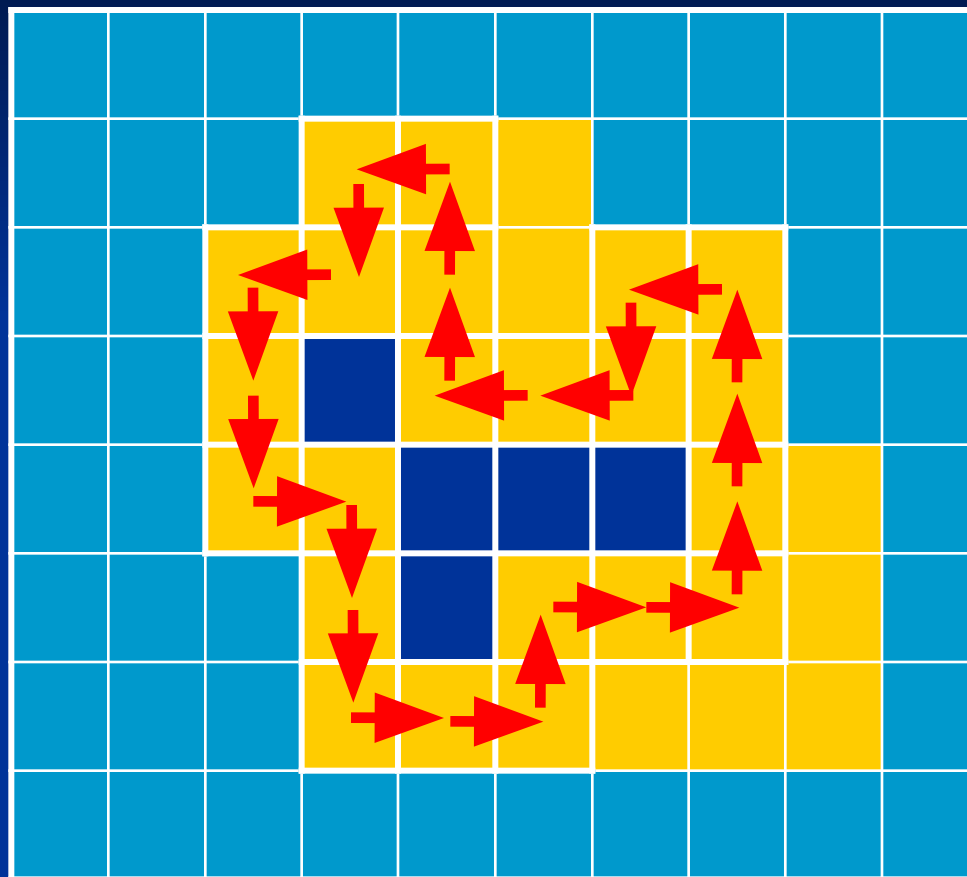
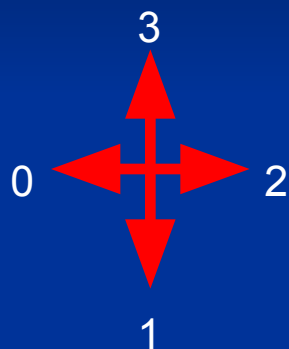
# Цепной код – 8-ми связные контура



- Кодирование контура как последовательности перемещений  
Код: 12232445466601760



# Цепной код – 4-х связные контура



- Кодирование контура как последовательности перемещений  
Код: 1122322333010033010112

# Цепной код - свойства



## ■ Свойства

- Цепной код – представление контура, независимое к его перемещению
- При замене в 8-ми связном кода любого  $n$  на  $(n \bmod 8) + 1$  контур будет **повернут** по часовой стрелке на 45 градусов
- Некоторые особенности контуров, такие как уголки, например могут быть сразу рассчитаны по анализу цепных кодов

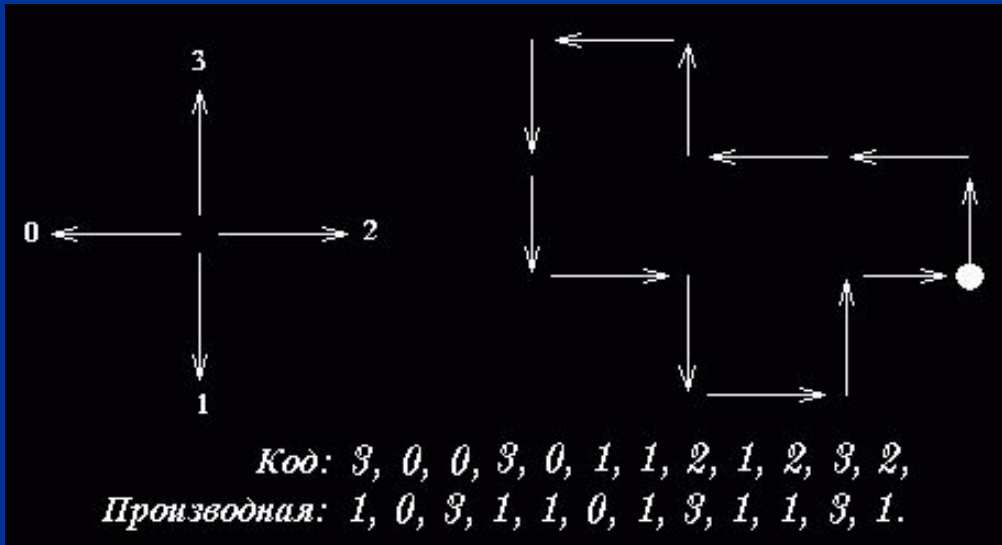
## ■ Сложности

- В цепном коде важна начальная точка – при ее изменении меняется и код
- Небольшие вариации границы (шум) серьезно меняют код. Сравнение двух шумных контуров по цепному коду – сложно
- Цепной код не инвариантен к повороту



# Разностный код

- Это «производная» цепного кода
  - Формула
    - $y_i = (x_{i+1} - x_i) \bmod 8$  – восьмисвязный
    - $y_i = (x_{i+1} - x_i) \bmod 4$  – четырехсвязный



# Разностный код

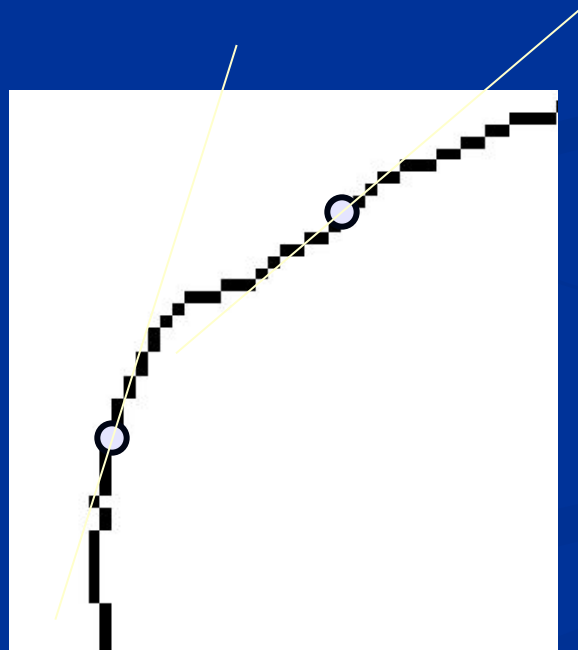


- Свойства:
  - Инвариантен к повороту кратному 45 градусам (восьмисвязный)
- Проблемы:
  - Также чувствителен к шуму
  - Не инвариантен к повороту на произвольный угол

# $\Psi$ -s представление



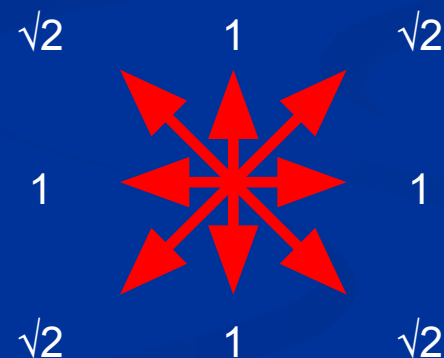
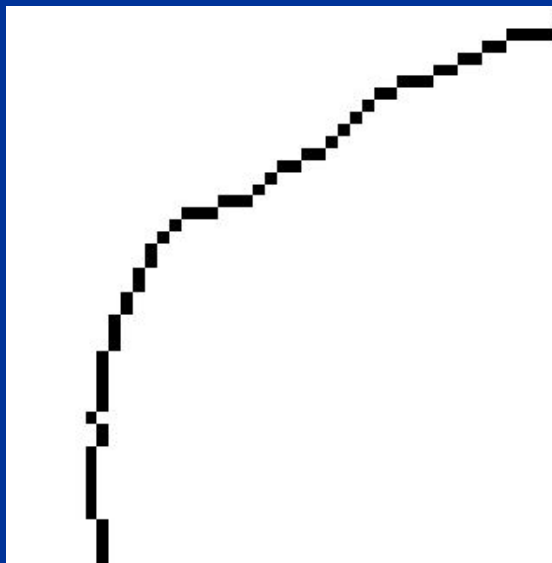
- Аналогично считаем направление контура в каждой точке, но:
  - Направление не ограничиваем точностью в 45 градусов – считаем точную касательную (угол  $\psi$ )



# Ψ-s представление



- Обратите внимание:
  - Результирующая функция  $\psi(s)$  зависит от длины дуги  $s$  (при движении «наискосок» она увеличивается быстрее)

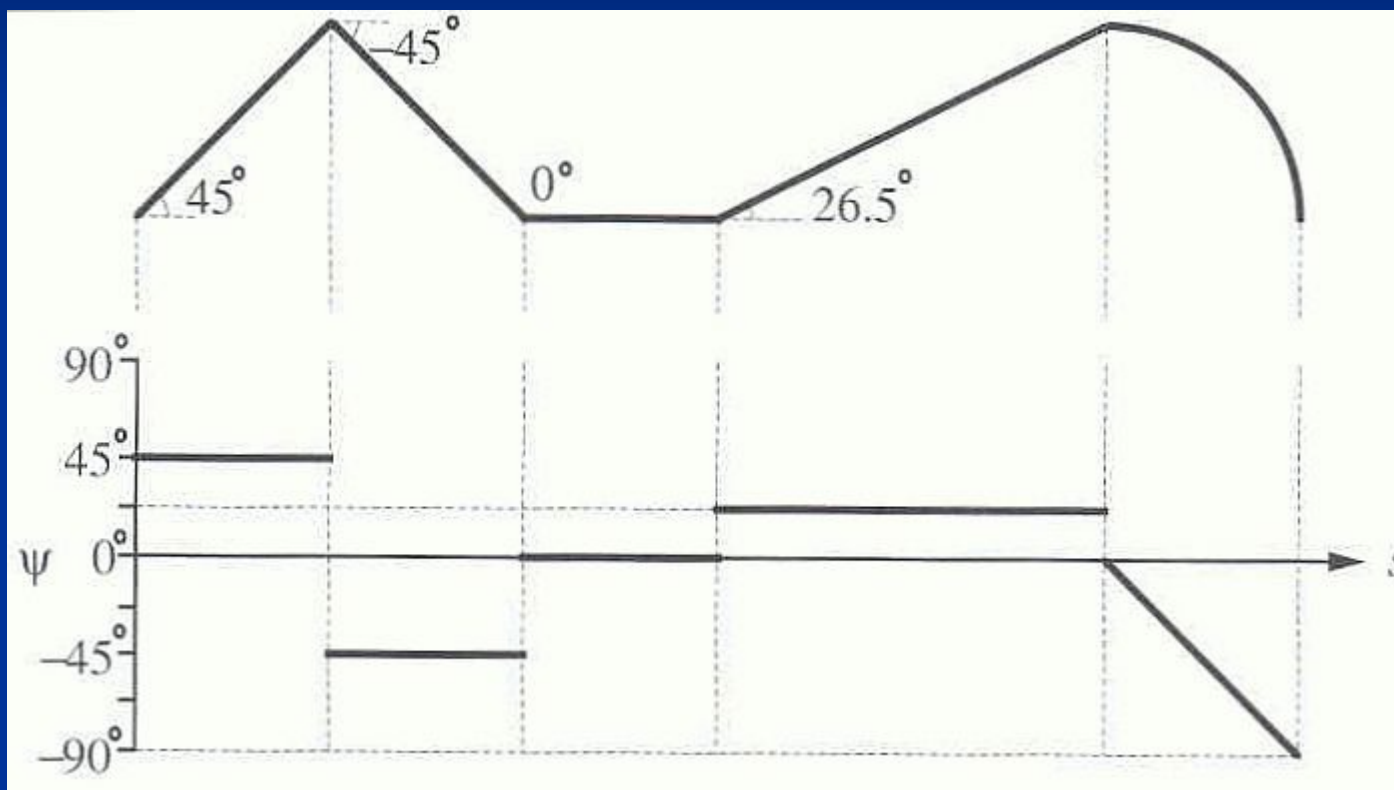


Расстояние между соседними пикселями

# Ψ-s представление



**NB** на данном графике ось абсцисс – не длина дуги, а ее горизонтальная проекция (для наглядности)



# $\Psi$ -s представление



- Свойства:
  - Если нормировать полную длину к 1 – инвариантно к масштабу
  - Несложным преобразованием можно свести к *практически* инвариантной к повороту мере
- Как *быстро* сравнить между собой две функции  $\psi(s)$ ?
  - Использовать функцию плотности наклона (slope density function)
    - Это гистограмма распределения углов наклона касательной контура



# Кривизна



- Кривизна (*curvature*) – производная  $\psi(s)$ 
  - Аналогично разностному цепному коду, но со знаком
$$K(s) = \psi(s) - \psi(s-1)$$
  - Представление, инвариантное к повороту и переносу
- Функция плотности кривизны
  - Гистограмма распределения значений кривизны

# Дескрипторы Фурье



- Контур задается набором точек:

$$(x_i, y_i), i = 1, 2, \dots, L$$

$$(x_c, y_c) = \left( \frac{1}{L} \sum_i x_i, \frac{1}{L} \sum_i y_i \right)$$

$$r_i = \sqrt{[x_i - x_c]^2 + [y_i - y_c]^2}$$

- Кол-во точек нормализуется к некоторому числу  $N$
- Вычисляем коэффициенты дискретного ПФ:

$$A_n = \frac{2}{N} \sum_{i=0}^{N-1} r_i \cos\left(\frac{2\pi \cdot n \cdot i}{N}\right), n = 1, 2, \dots, \frac{N}{2} - 1$$

$$A_n = \frac{1}{N} \sum_{i=0}^{N-1} r_i \cos\left(\frac{2\pi \cdot n \cdot i}{N}\right), n = 0, \frac{N}{2}$$

$$B_n = \frac{2}{N} \sum_{i=0}^{N-1} r_i \sin\left(\frac{2\pi \cdot n \cdot i}{N}\right), n = 0, 2, \dots, \frac{N}{2}$$

# Дескрипторы Фурье



- Вычисляем амплитуды коэффициентов:

$$C_n = \sqrt{A_n^2 + B_n^2}$$

- Дескрипторы:

$$\mathbf{f} = \left[ \frac{|C_1|}{|C_0|}, \frac{|C_2|}{|C_0|}, \dots, \frac{|C_{N/2}|}{|C_0|} \right]$$

- Свойства:

- *Инвариантны к переносу (вычли среднюю точку)*
- *Инвариантны к повороту (берем модули коэффициентов)*
- *Инвариантны к масштабу (нормировали на нулевой коэффициент)*
- *Общая форма – первые по порядку дескрипторы, детали и шум – в конце*
- *Нельзя характеризовать локальные особенности кривой*

# Признаки для распознавания контуров



- ✓ Цепной код (разностный код)
- ✓  $\Psi$ -s представление
  - ✓ Функция плотности наклона (slope density function)
  - ✓ Функция плотности кривизны (curvature density function)
- ✓ Дескрипторы Фурье

# Как вести прикладное исследование?



- Подготовительный этап
  1. Уяснить постановку задачи
  2. Собрать данные для проверки работы методов
  3. Провести обзор литературы
    - Уяснить отличия, сильные/слабые стороны методов, трудоемкость реализации
    - Фиксировать критерии сравнения до начала обзора
  4. Выбрать потенциально подходящие методы

# Как вести прикладное исследование?



- Экспериментальный этап
  1. Составить план экспериментов
    - Какие характеристики и как именно будет проверяться (скорость, точность, устойчивость и т.д.)
  2. Провести проверку методов на собранных данных
  3. Если один из методов работает приемлемо
    - выбираем его
  4. Если ни один метод приемлемо не работает
    - Анализируем проблемы и причины ошибок
    - Делаем предположения – что именно может помочь
    - Создаем свой метод или комбинируем несколько методов, основываясь на сделанных предположениях
    - Переходим на шаг 2 для проверки предположений