ЛЕКЦИЯ 4. АЛГОРИТМЫ ВЫДЕЛЕНИЯ КОНТУРОВ

Демидов Д.В.

Обработка аудиовизуальной информации.
Бакалавры, 6 семестр. Магистры, 9 семестр

План лекции

- Алгоритмы выделения контуров.
- Операторы.
- Методы прослеживания и описания контуров.
- Цепной код Фримена. Модифицированный код Фримена.
- Сегментация контурных линий.
- Методы обработки и распознавания контурных изображений. Метод концевых точек.

Задача выделения контуров

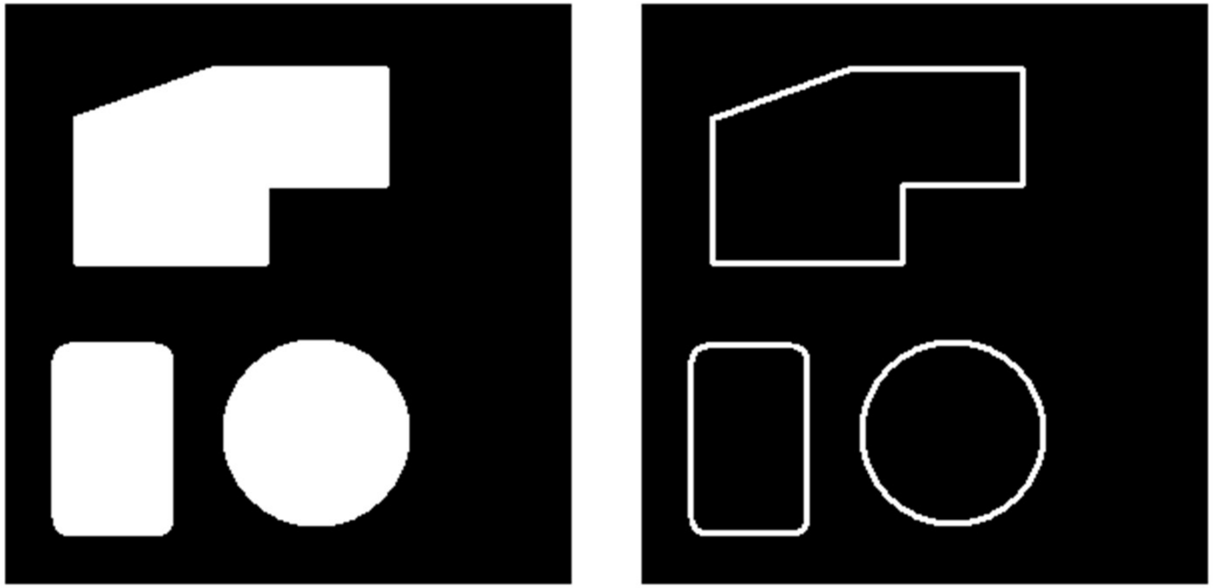
Производные функций, градиент изображения

Операторы Робертса, Прюитт, Собеля, Шарра

Операторы Кэнни, Ротуэлла, Айверсона

Морфологическое выделение контуров

- Выполняется эрозия изображения (сжатие объекта)
- Строится разностное изображение



https://ru.bmstu.wiki/%D0%A4%D0%B0%D0%B9%D0%BB:8_17.png

Схема оконной фильтрации

- Коэффициенты маски с относительными значениями координат

$$
\begin{matrix}
w_{-1,-1} & w_{0,-1} & w_{1,-1} \\
w_{-1,0} & w_{0,0} & w_{1,0} \\
w_{-1,1} & w_{0,1} & w_{1,1}
\end{matrix}
$$

- Элементы изображения в окне того же размера

$$
\begin{matrix}
f(x-1,y-1) & f(x,y-1) & f(x+1,y-1) \\
f(x-1,y) & f(x,y) & f(x+1,y) \\
f(x-1,y+1) & f(x,y+1) & f(x+1,y+1)
\end{matrix}
$$

- Отклик фильтра в точке ***(x,y)*** как сумма произведений коэффициента на значение соответствующего пикселя:

$$
R = w_{-1,-1}f(x-1,y-1) + w_{-1,0}f(x-1,y) + w_{-1,1}f(x-1,y+1) + \ldots + w_{0,0}f(x,y) + \ldots + w_{1,1}f(x+1,y+1)
$$

Производные одномерной функции

- Первая производная функции яркости определяется как разность значений соседних элементов

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

- Вторая производная определяется как разность соседних значений первой производной

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

Производные двумерной функции

- Вычисление первой производной цифрового изображения основано на различных дискретных приближениях двумерного градиента.
- По определению, градиент изображения ***f(x,y)*** в точке ***(x,y)*** — это вектор частных производных:

$$\nabla f = \left(G_x, G_y\right) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$$

- Направление вектора градиента совпадает с направлением максимальной скорости изменения функции ***f*** в точке ***(x,y)***

Градиент функции

- **Модуль** вектора градиента равен значению максимальной скорости изменения функции ***f*** в точке ***(x,y)***, причем максимум достигается в направлении вектора ∇f.

$$\nabla f = \left|\nabla f\right| = \sqrt{G_x^2 + G_y^2} = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

- **Направление** вектора градиента в точке ***(x,y)*** перпендикулярно направлению контура в этой точке.
- Угол ***α(x,y)*** между направлением вектора ∇f в точке ***(x,y)*** и осью ***x***:

$$\alpha(x, y) = arctg\left(\frac{G_y}{G_x}\right)$$

- Формулы расчёта частных производных для каждой точки в разных методах определяются по-разному.

Вычисление градиента функции яркости

- Градиент в точке (x,y): $$\nabla f = \sqrt{G_x^2 + G_y^2}$$

  или приближенно $$\nabla f \approx |G_x| + |G_y|$$

- Нормализованный градиент в точке (x, y) в диапазоне [0..255]:

$$\nabla f_{norm} = \nabla f \frac{255}{\max\limits_{(x,y)} \nabla f}$$

- Решающее правило с порогом *T* для получения контурного изображения:

$$\mathrm{f}(x, y) = \begin{cases} 255, & if\ \nabla f_{norm} > T \\ 0, & otherwise \end{cases}$$

Оператор Робертса

- Рассмотрим окно 3x3 для каждого элемента:

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- Частные производные в центральной точке z5 в матричном виде:

$$G_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * A \qquad G_y = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} * A$$

- Частных производные в скалярном виде:

$$G_x = z_9 - z_5$$

$$G_y = z_8 - z_6$$

Оператор Прюитт

- Маски оператора Прюитт:

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

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

- Частные производные в матричном виде:

$$G_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} * A \qquad G_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} * A$$

- Частные производные в скалярном виде:

$$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

- https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_%D0%9F%D1%80%D1%8E%D0%B8%D1%82%D1%82

Оператор Прюитт 5x5

- Маски оператора:

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

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

Оператор Собеля

- Маски оператора Собеля:

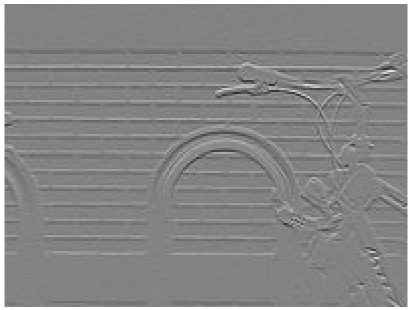
$$G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * A \qquad G_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * A$$

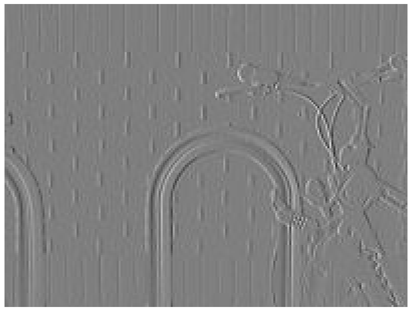
- Частные производные по x и по y:

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$


Оригинал


Gx


Gy


Нормализованный градиент

Оператор Шарра (Scharr)

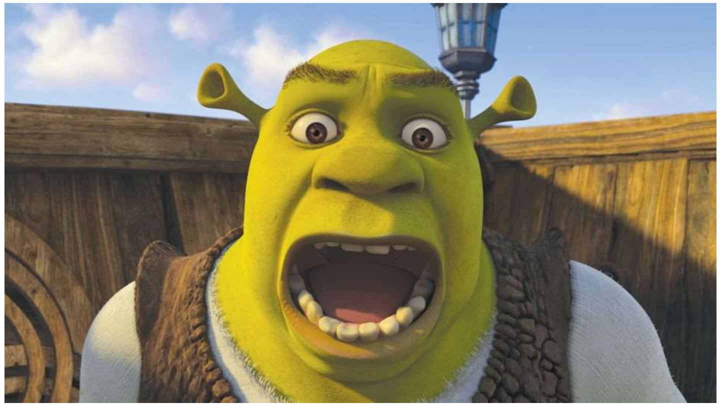
- Аналог оператор Собеля с другим ядром:

$$G_x = \begin{bmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{bmatrix} * A \qquad G_y = \begin{bmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix} * A$$

$$G_x = \left(3z_1 + 10z_4 + 3z_7\right) - \left(3z_3 + 10z_6 + 3z_9\right)$$

$$G_y = \left(3z_1 + 10z_2 + 3z_3\right) - \left(3z_7 + 10z_8 + 3z_9\right)$$

Пример 1. Шрек



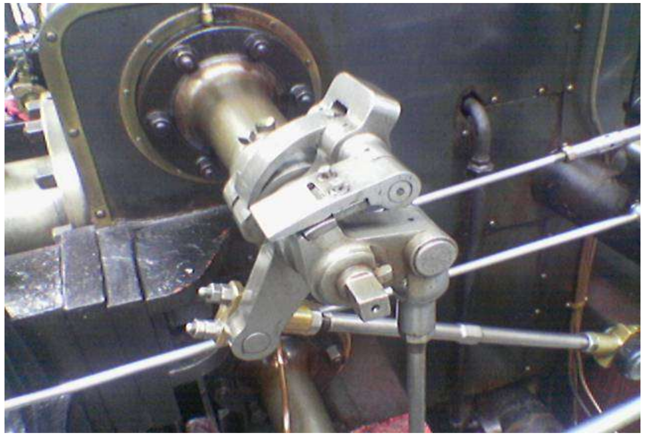
Оригинал

Прюитт

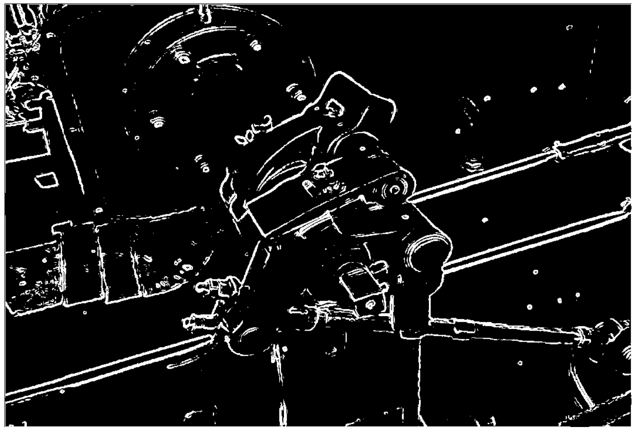
Робертс

Собель

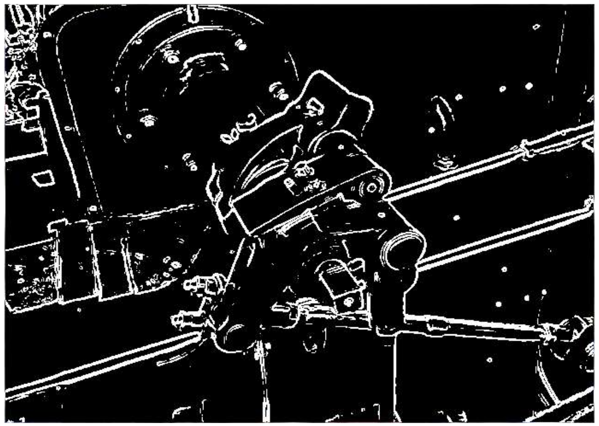
Пример 2. Паровая машина



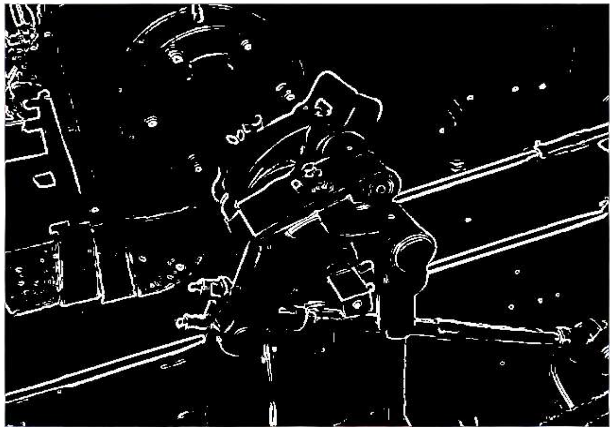
Оригинал



Робертс



Собель

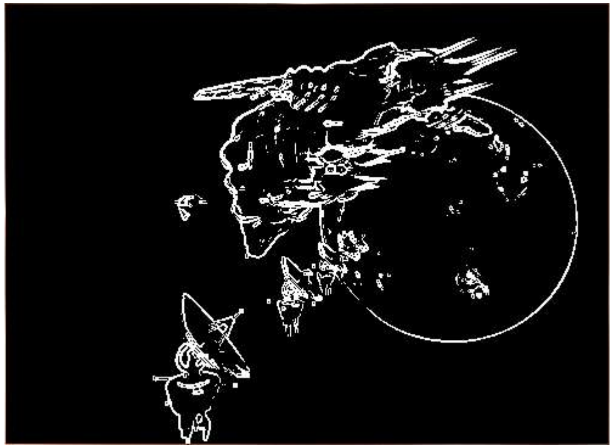


Прюитт

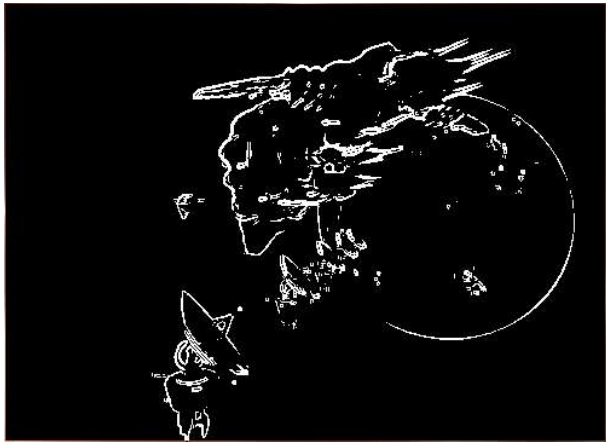
Пример 3. Звездолёт

Оригинал

Робертс



Собель



Прюитт

Пример 4. Lenna 3x3

Оригинал

Прюитт 3x3

Шарр 3x3

Собель

Пример 5. Lenna 5x5

Оригинал

Прюитт 5x5

Шарр 3x3

Собель

Оператор Кэнни (Canny)

- Известен как оптимальный детектор края, оператор обнаружения границ, детектор границ.
- Критерии оптимальности:
  - Низкий уровень ошибок
  - Высокая локализованность краевых точек
  - Одна граница – одно обнаружение

Adobe Acrobat Document

Основные этапы алгоритма

- **Шаг 0.** Перед применением детектора обычно преобразуют изображение в оттенки серого, чтобы уменьшить вычислительные затраты.
- **Шаг 1.** Сглаживание изображения, чтобы устранить шум.
- **Шаг 2.** Поиск градиентов изображения, чтобы подсветить области с высокими пространственными производными.
- **Шаг 3.** Проход по этим областям с подавлением всех пикселей, которые не в максимуме (немаксимальное подавление).
- **Шаг 4.** Удаление слабых границ.
- **Шаг 5.** Подавление краёв, не связанных с сильными границами.

Шаг 1. Сглаживание

- Для размытия используется фильтр, хорошо приближенный к первой производной гауссианы. σ= 1.4:

$$B = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * A$$

- Чем больше ширина маски Гаусса, тем меньше чувствительность детектора к шуму и тем больше локализация ошибки в обнаружении краев.

Шаг 2. Поиск градиентов

- Для вычисления градиента используется оператор Собеля.
- Границы отмечаются там, где градиент изображения приобретает максимальное значение.
- Они могут иметь различное направление, поэтому алгоритм Кэнни использует четыре фильтра для обнаружения горизонтальных, вертикальных и диагональных ребер в размытом изображении.

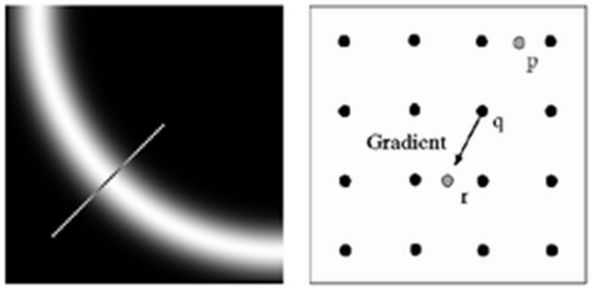
$$G = \sqrt{G_x^2 + G_y^2}$$

$$\theta = arctg\left(\frac{G_y}{G_x}\right)$$

- Угол направления вектора градиента округляется до ближайшего из четырёх значений: 0, 45, 90, 135.

Шаг 3. Подавление немаксимумов

- Только локальные максимумы отмечаются как границы.
- Не максимальные точки, лежащие рядом с границей, удаляются.
- Используется информация о направлении границы для того, чтобы удалять точки именно рядом с границей и не разрывать саму границу вблизи локальных максимумов градиента.



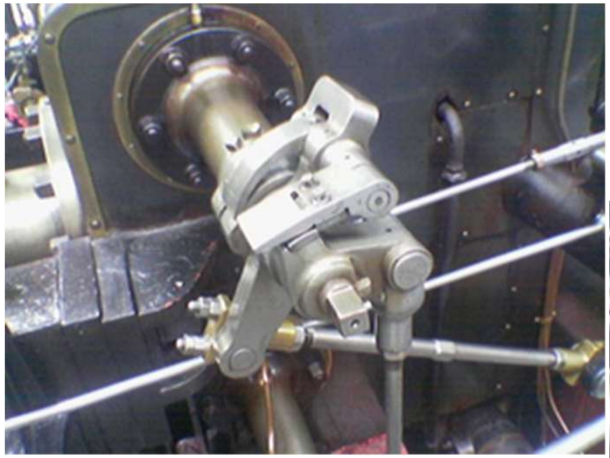
Шаг 4. Двойная пороговая фильтрация

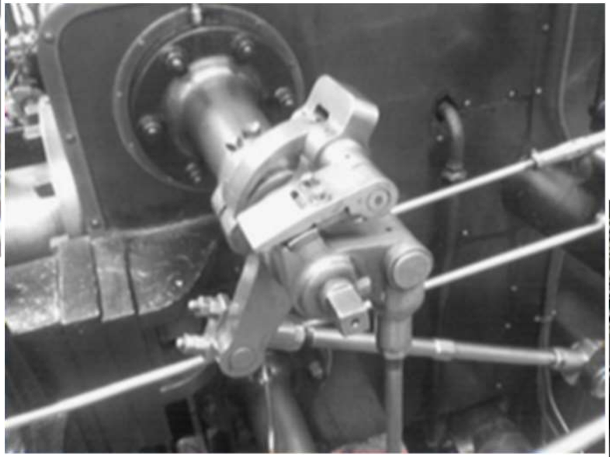
- Потенциальные границы определяются двумя порогами. Решающее правило:
  - Если $G < T1$, то она устанавливается в ноль (делается не краевой).
  - Если $G > T2$, она делается краевой.
  - Если $T1 < G < T2$, то она устанавливается в ноль, в том случае если нет пути от этого пикселя к пикселю с градиентом выше T2.
- Таким образом, слабые границы удаляются.
- Гистерезис используется, чтобы отследить оставшиеся пиксели, которые не были подавлены.

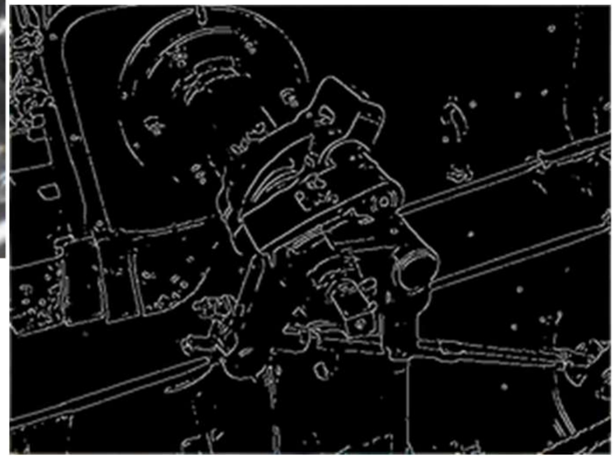
Шаг 5. Трассировка области неоднозначности

- Итоговые границы определяются путём подавления всех краёв, не связанных с определенными (сильными) границами.
- После того как известны направления краев, применяем немаксимальное подавление. Оно используется для отслеживания вдоль края в направлении края и подавлении любых значений пикселя (устанавливая их равным 0), которые не считаются краем. Это даст тонкую линию в результирующем изображении.

Пример 2 для оператора Кэнни







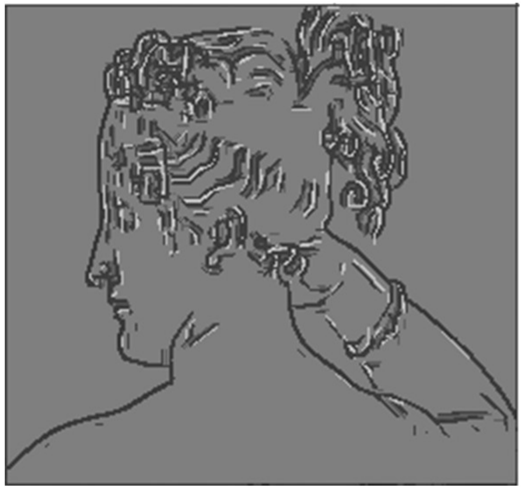
Оригинал → Фильтр Гаусса → Оператор Кэнни

Оператор Ротуэлла (Rothwell)

- Похож на оператор Кэнни
- Разница:
  - алгоритм Ротуэлла использует истончение краёв вместо подавления немаксимумов;
  - динамическое определение порога используется вместо гистерезиса.

Оператор Айверсона (Iverson)

- Основным преимуществом является значительное уменьшение количества ошибочно положительных откликов (распознавания несуществующих границ)
- Позволяет четко разделять между собой три вида границ:
  - Края (*step-edges*).
  - Светлые линии (*positive contrast lines*).
  - Темные линии (*negative contrast lines*).



Что почитать

- Оператор Кэнни:
  - http://www.limsi.fr/Individu/vezien/PAPIERS_ACS/canny1986.pdf
  - http://suraj.lums.edu.pk/~cs436a02/CannyImplementation.htm
  - https://en.wikipedia.org/wiki/Edge_detection
  - Комментарий к алгоритму выделения контуров Канни https://habrahabr.ru/post/114766/
  - https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_%D0%9A%D1%8D%D0%BD%D0%BD%D0%B8
  - Билл Грин, Алгоритм выделения контуров CANNY http://masters.donntu.org/2010/fknt/chudovskaja/library/article4.htm
- Оператор Ротуэлла https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_%D0%A0%D0%BE%D1%82%D1%83%D1%8D%D0%BB%D0%BB%D0%B0
- Iverson, Zucker - Logical/Linear Operators for Image Curves http://www.ai.sri.com/~leei/pami95/paper.html

Прослеживание контуров

Алгоритм Жука

Алгоритм на основе градиентов

Поиск особых точек

Код Фримена

Дескрипторы Фурье

Прослеживание контуров

- Результатом выполнения процедуры прослеживания является **дискретное представление контуров, при котором каждый контур определяется множеством точек, из которых он состоит**.
- Для упрощения вычисления направления края весь диапазон возможных значений 0,…,360° разбивается на 8 направлений (секторов). Каждое направление отличается от соседнего на 45°. При этом поиск точек, принадлежащих одному контуру, следует проводить среди точек соседних секторов, имеющих расхождения значений градиентов меньше заданного порога.

Прослеживание контуров (II)

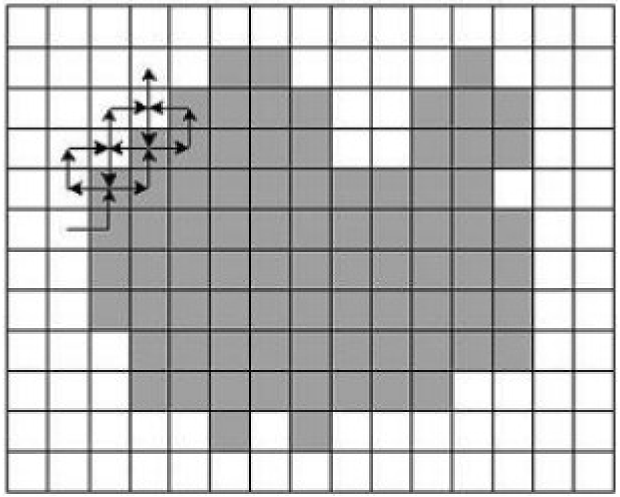
- Алгоритмы прослеживания контуров (edge following) можно разделить на несколько групп:
  - методы, использующие информацию о значении и направлении градиента в каждой точке;
  - методы, использующие динамическое программирование для решения задачи прослеживания контура;
  - методы поиска оптимального пути в графе. Каждая краевая точка представляется вершиной графа.

Алгоритм прослеживания контуров

- На бинарном изображении, содержащем одну фигуру, отыскивается точка, принадлежащая этой фигуре.
- Далее осуществляется проход по контуру фигуры с помощью двух правил:
  - оказался внутри фигуры - движение влево по отношению к предыдущему шагу;
  - оказался снаружи фигуры - движение вправо по отношению к предыдущему шагу.
- В случае перехода из внутренней области фигуры наружу необходимо проверить пиксел, расположенный слева по отношению к предыдущему шагу.
- Прослеживаемые пикселы помечаются.

Схема работы алгоритма

- "Жук" начинает движение с *белой* области по направлению к *черной*. Как только он попадает на *черный* элемент, он поворачивает *налево* и переходит к следующему элементу.
  - **Если** этот элемент *белый,*
  - **то** жук поворачивает *направо,*
  - **иначе** жук поворачивает *налево*.



- Процедура повторяется до тех пор, пока жук не вернется в исходную точку. Координаты точек перехода с черного на белое и с белого на черное и описывают границу объекта.

Сканирующие алгоритмы

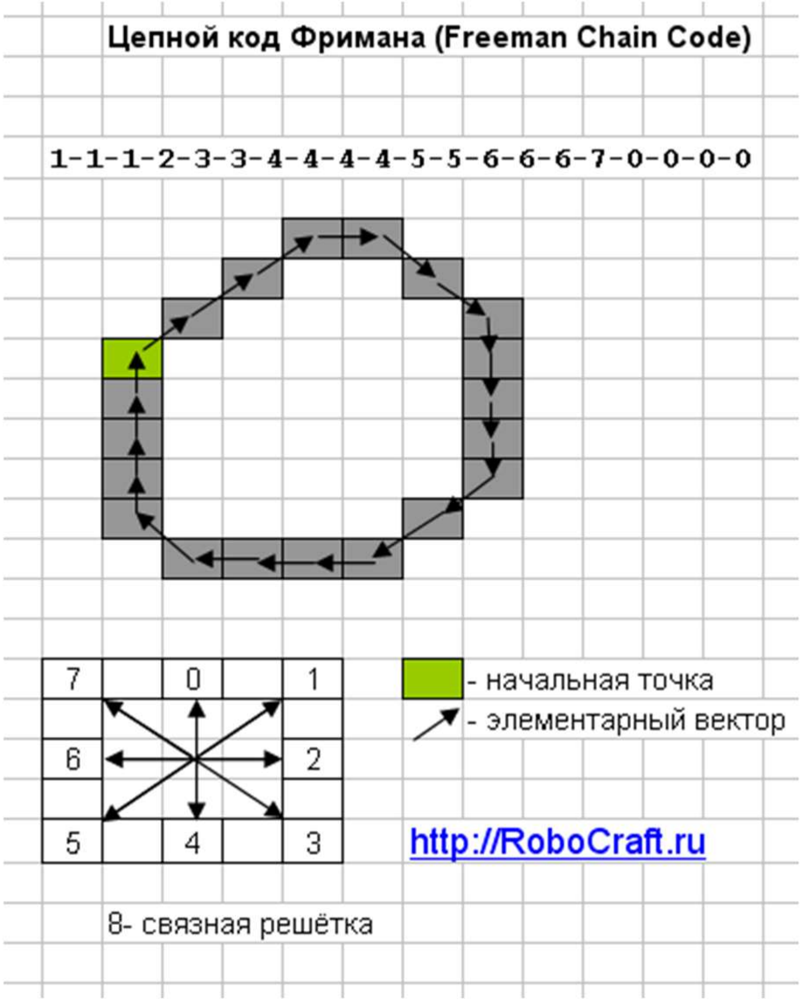
Сегментация контуров методом концевых точек TODO

- На бинарном изображении нанесена ломаная линия.
- Координаты каждой из точек хранятся в массиве данных. Из массива данных выбираются первая и последняя точки и проверяется, удовлетворяют ли точки, находящиеся между ними, критерию соответствия прямой линии.
  - Если да, то процесс сегментации закончен.
  - Если нет, то вместо последней точки берется предпоследняя и процедура повторяется до тех пор, пока не будет обнаружен участок прямой линии. Это будет первый сегмент.
- Далее в качестве первой точки принимается последняя точка первого сегмента и процедура повторяется до тех пор, пока не будет сегментирована вся ломаная линия.
- Отдельные сегменты помечаются.

Цепной код Фримена

- Цепные коды применяются для представления границы в виде последовательности отрезков прямых линий определённой длины и направления.
- В основе этого представления лежит 4- или 8- связная решётка.
- Длина каждого отрезка определяется разрешением решётки, а направления задаются выбранным кодом.
- Для представления всех направлений в 4-связной решётке достаточно 2-х бит, а для 8-связной решётки цепного кода требуется 3 бита.
- Оригинальная статья: http://ieeexplore.ieee.org/document/5219197/
- http://robocraft.ru/blog/computervision/640.html

Пример цепного кода

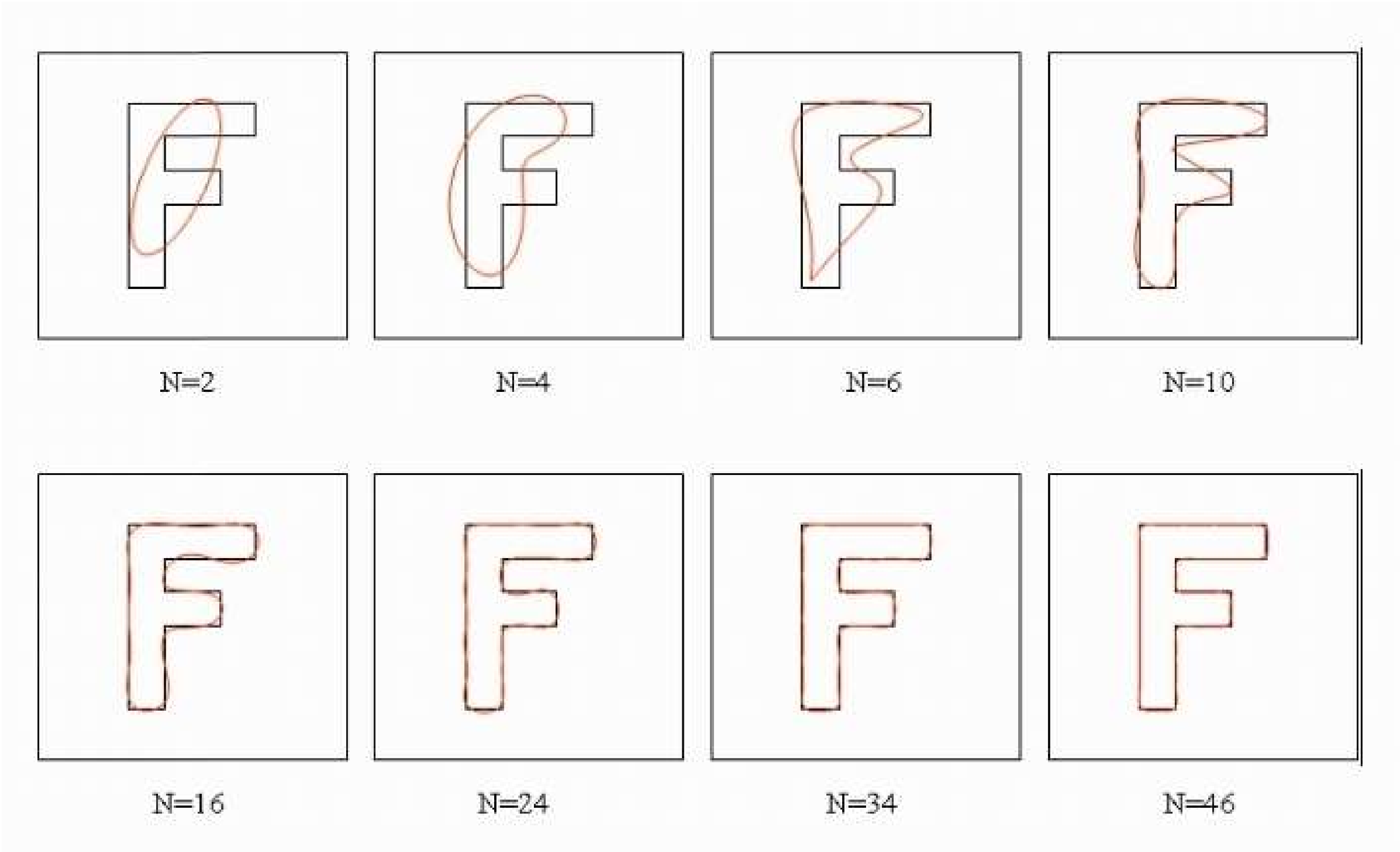


- Достоинства:
  - компактность
- Недостатки:
  - зависимость от начальной точки кодирования.
  - не обладает свойством инвариантности к вращению.
  - неустойчивость к зашумлению. Локальные изменения контура могут привести к различным результатам кодирования.

Дескрипторы Фурье

- ДПФ (*Discrete Fourier Transform, DFT*)
  - прямое
  - обратное (InvDFT)
- Быстрое преобразование Фурье (FFT) позволяет за O(n*log(n)) рассчитывать DFT и InvDFT.
- Кодирование контура дескрипторами
  - Преобразование координат контура в комплексные числа и ДПФ этих чисел
  - Выбор числа дескрипторов (N) и отбрасывание высокочастотной части спектра
- ДПФ также применяется при сжатии в MP3, JPEG

Описание контура разным числом дескрипторов Фурье



Прослеживание контуров на основе градиентов

- Предполагается, что точки, принадлежащие одному контуру, должны иметь близкие значения модуля и направления вектора градиента.
- Рассматривается окрестность точки ⟨i,j⟩ размером M×M (обычно используют окрестность 3×3), и в каждой точке ⟨k,l⟩ окрестности проверяются следующие условия:
  - $|G_{i,j} - G_{k,l}| \le \Delta G$,
  - $|\alpha_{i,j} - \alpha_{k,l}| \le \Delta\alpha$,

где ⟨i,j⟩ - центральная точка окрестности; G - модуль градиента; α - направление градиента в точке; ΔG - предельное значение расхождения модулей градиента в точках ⟨i,j⟩ и ⟨k,l⟩; Δα - предельное значение расхождения направлений векторов градиента в точках ⟨i,j⟩ и ⟨k,l⟩.

- Если в точке ⟨k,l⟩ выполняются описанные выше условия, то считается, что пара точек принадлежит одному контуру.

Поиск точек ветвления

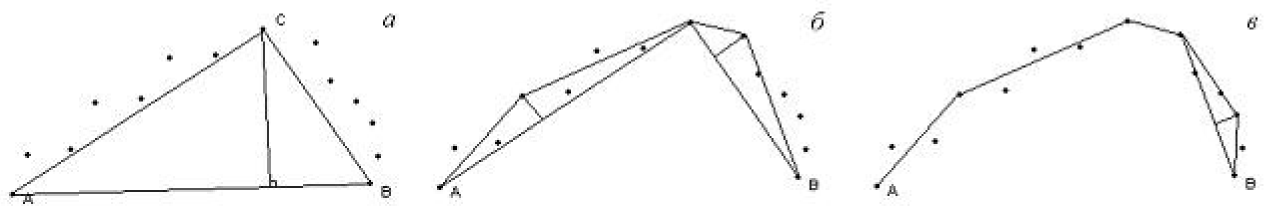
- Далее выделяются точки ветвления (точки соединения кривых). Наличие точек ветвления свидетельствует о сложной геометрической структуре объекта, существенно затрудняет формальное описание и сам процесс распознавания объектов.
- Выделение точек ветвления позволяет значительно упростить структуру объекта путем разбиения контура на множество кривых.

Анализ особых точек контура

- В качестве характерных признаков можно использовать число и положения особых точек контура:
  - точки максимального перегиба,
  - локальные экстремумы функции кривизны,
  - концевые точки,
  - точки ветвления.
- Наиболее простым и быстрым (но не лучшим) способом является поиск точек максимального перегиба при помощи итеративного алгоритма подбора концевых точек.

Подбор концевых точек

- На первом этапе работы алгоритма концевые точки контура A и B соединяются прямой линией. Для всех оставшихся точек вычисляются расстояния до прямой AB.
- Точка, имеющая наибольшее отклонение от прямой AB, берется в качестве дополнительного узла. При этом кривая заменяется двумя отрезками AC и CB (б).
- Процедура продолжается до тех пор, пока максимальное значение отклонения точек меньше заданного порога. Точность аппроксимации прямыми линиями определяется величиной порога.



Функция кривизны $K(x,y)$

- Одним из важнейших параметров, характеризующих контур, является его *кривизна*.
- Кривизна обладает свойствами инвариантности к сдвигу, повороту и вычисляется по формуле

$$K(x,y)=\frac{f'_x f''_y - f'_y f''_x}{\sqrt{\left(f'^2_x + f'^2_y\right)^3}}$$

- где f′x,f′y - первые производные по x и y соответственно; f′′x,f′′y - вторые производные по x и y;

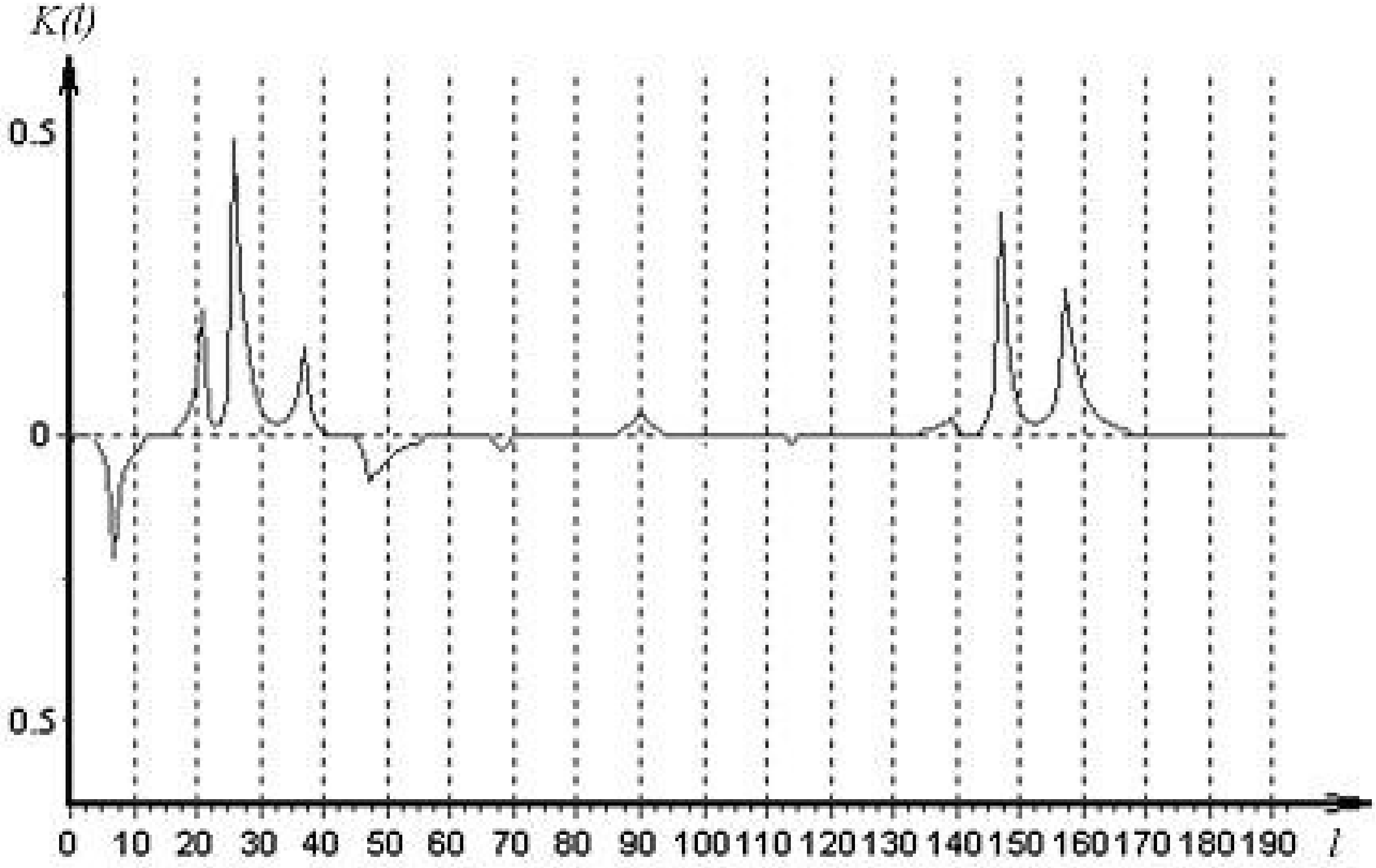
Функция кривизны *K(l)*

- Контур можно представить в виде одномерной функции какого-либо атрибута от длины дуги. Длину дуги дискретного контура в точке P(j)=⟨xj,yj⟩ можно аппроксимировать следующим образом:

$$l = \sum_{i=1}^{j-1} \sqrt{\left(x_i - x_{i+1}\right)^2 + \left(y_i - y_{i+1}\right)^2}$$

- Естественное представление кривой подразумевает отсутствие на контурах точек соединений и разветвлений, в противном случае контур не может быть представлен в виде одномерной функции. Данное ограничение требует введения дополнительных процедур обработки и анализа полученного контурного препарата:
  - поиск на контурах точек ветвления;
  - разделения сложных структур на составляющие.

Пример функции кривизны $K(l)$

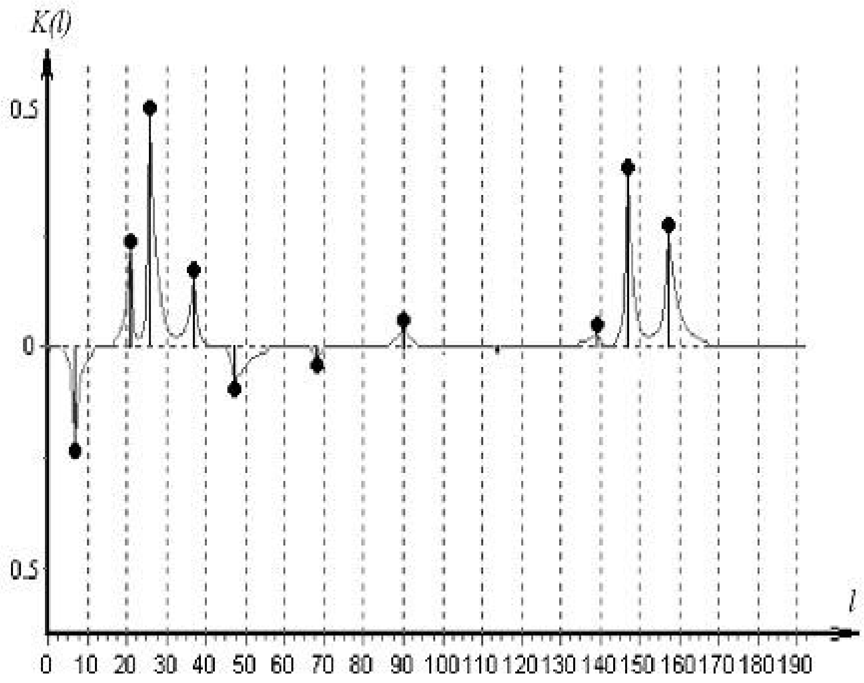


Поиск локальных экстремумов функции кривизны

- Выполнить кусочно-полиномиальную аппроксимацию контура;
- Построить функцию кривизны;
- Найти все локальные экстремумы кривизны.

Функция перегиба

- Экстремумы соответствуют максимальным перегибам контуров



Что почитать

- Р. Гонсалес, Р. Вудс Цифровая обработка изображений — М: Техносфера, 2005 – 1007с
- Анисимов Б.В. Распознавание и цифровая обработка изображений – М.: Высш. школа, 1983 – 295с
- Дуда Р., Харт П. Распознавание образов и анализ сцен. Перевод с английского Г. Г. Вайештейна и А. М. Васьковского, под редакцией В. Л. Стефанюка, Издательство «МИР», Москва 1976. - 509 с.
- Введение в контурный анализ и его приложения к обработке изображений и сигналов / Под ред. Я.А. Фурмана. М., 2002.
- Выделение и описание контуров http://wiki.technicalvision.ru/index.php/%D0%92%D1%8B%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B8_%D0%BE%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D0%B5_%D0%BA%D0%BE%D0%BD%D1%82%D1%83%D1%80%D0%BE%D0%B2