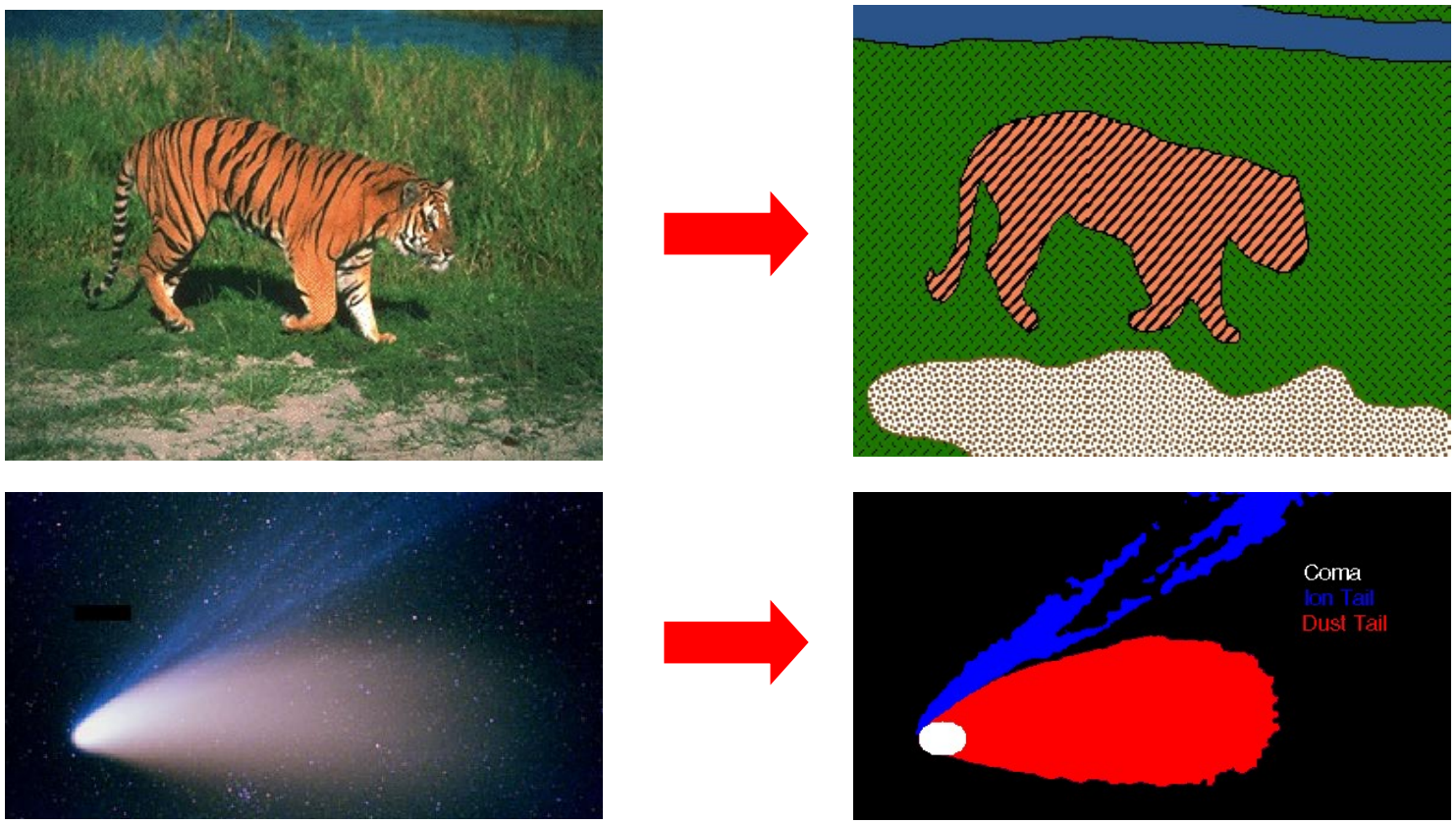
Сегментация изображений

Что такое сегментация изображений?

- Разбиение изображения на множество однородных областей для последующего анализа.



Что такое сегментация изображений?

- Разбиение изображения на множество однородных областей для последующего анализа.
- Формально:
  - Разбиение изображения на набор областей $S = \{S_i\}, i = \overline{1, N}$, однородных по критерию P
    - $I = \bigcup_{i=1..N} S_i$
    - $\forall i, j = \overline{1, N} : i \neq j \; S_i \bigcap S_j = \varnothing$
    - $\forall i = \overline{1, N}, P(S_i) = истина$
    - $\forall i, j = \overline{1, N} : i \neq j \;\; P(S_i \bigcup S_j) = ложь$

Требования к областям сегментации на практике

- Области, являющиеся результатом сегментации, должны быть однородны в смысле некоторого критерия (интенсивности, цвета, текстуры).
- Внутренности областей должны быть просты и не должны содержать большого количества дырок.
- Смежные области должны значимо различаться в смысле выбранного критерия однородности.
- Границы каждой области должны быть гладкими и пространственно точными.

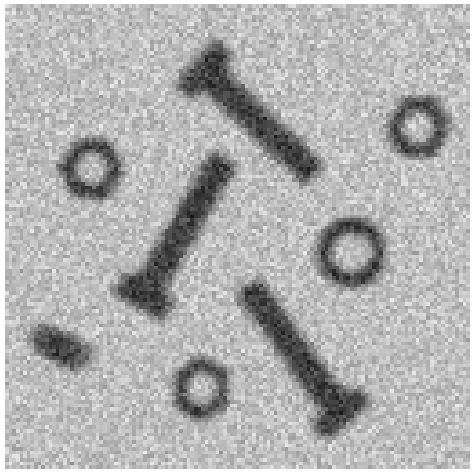
Подходы

- Пороговая сегментация
- Наращивание / декомпозиция областей
- Классические методы кластеризации
- Выделение контуров областей
- Теоретико-графовые методы
- Метод водораздела

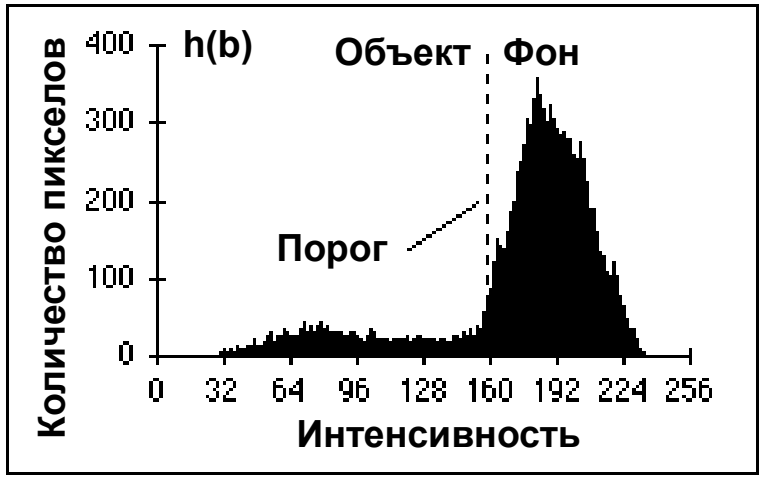
Пороговая сегментация

- Пикселы с интенсивностью ниже порогового значения помечаются, как принадлежащие объекту, остальные пикселы – как принадлежащие фону.
- Здесь и далее рассматриваются изображения, содержащие темные объекты на светлом фоне.

Изображение

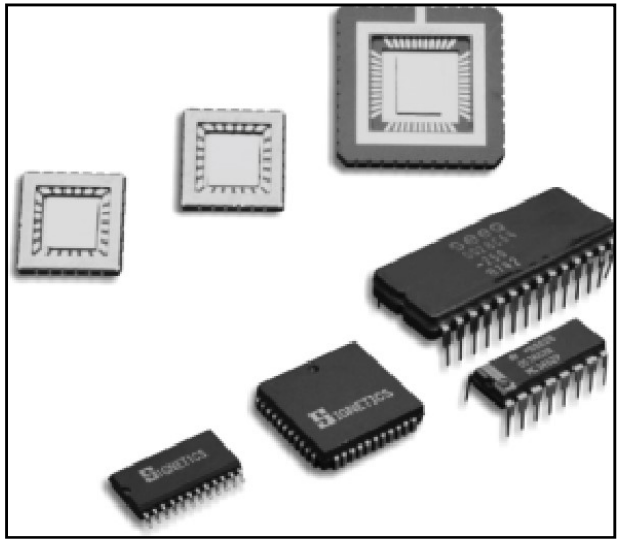


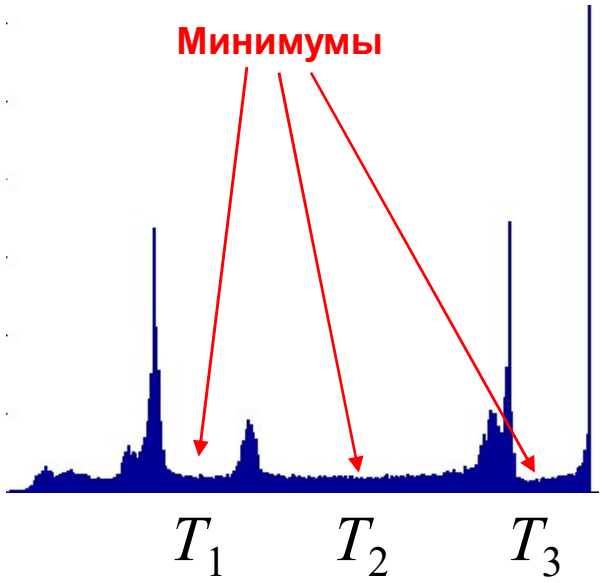
Гистограмма



Пороговая сегментация

Возможно обобщение на случай многопороговой сегментации





$$I(x,y) < T_1, \quad I(x,y) = 1 \text{ (объект 1)}$$
$$T_1 < I(x,y) < T_2, \quad I(x,y) = 2 \text{ (объект 2)}$$
$$T_2 < I(x,y), \quad I(x,y) = 3 \text{ (фон)}$$

Пороговая сегментация

- Основная задача, решаемая при пороговой бинаризации – автоматический выбор значения порога.
- Известно несколько алгоритмов выбора значения порога по гистограмме
  - Алгоритм Изодата (Isodata)
  - Алгоритм симметрии фона
  - Алгоритм треугольника
  - Алгоритм Отсу.

Алгоритм Изодата

- **Авторы**: Ridler, Calvard.
- **Недостаток**: результат зависит от выбора начального значения порога.

1. Начальное значение порога $T_0 = 2^{B\text{-}1}$ разбивает динамический диапазон интенсивностей пополам (B – глубина изображения в битах).
2. Вычисляются средние интенсивности объекта $m_{f,0}$ и фона $m_{b,0}$.
3. Очередное значение порога вычисляется как среднее $m_{f,0}$ и $m_{b,0}$: $T_{i+1} = (m_{f,i} + m_{b,i})/2$.
4. Процесс повторяется до стабилизации значения порога (т.е. пока $T_i \neq T_{i\text{-}1}$).

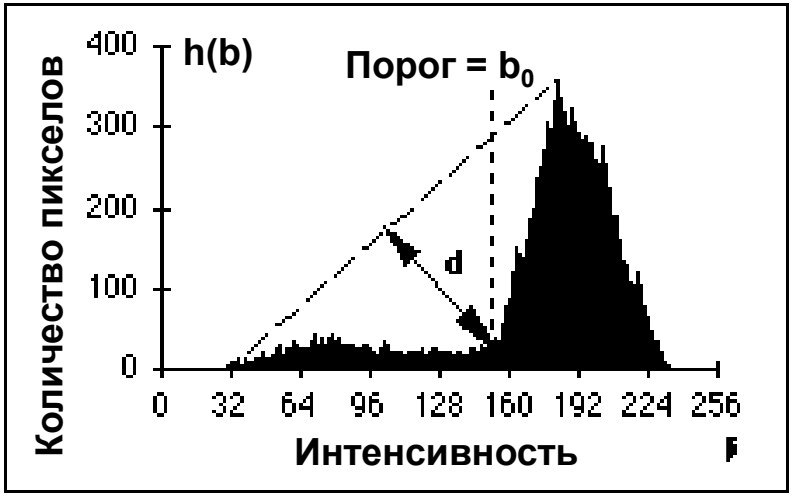
Алгоритм симметрии фона

- Предполагается, что фоновые пикселы дают максимальный пик на гистограмме, и все пикселы фона имеют интенсивности, симметрично сгруппированные вокруг пика.
- **Недостаток**: выдвигаемая гипотеза о гистограмме.

1. Отыскивается $b_{\max} = \arg\max(\mathrm{h}(b))$.
2. Справа от $b_{\max}$ отыскивается точка b_{right}, разбивающая гистограмму так, чтобы справа от нее было сгруппировано $p\%$ всех пикселов (например, 5%, 1%).
3. Значение порога вычисляется как точка, симметричная точке b_{right} относительно $b_{\max}$: $T = b_{\max} - (b_{\text{right}} - b_{\max})$.

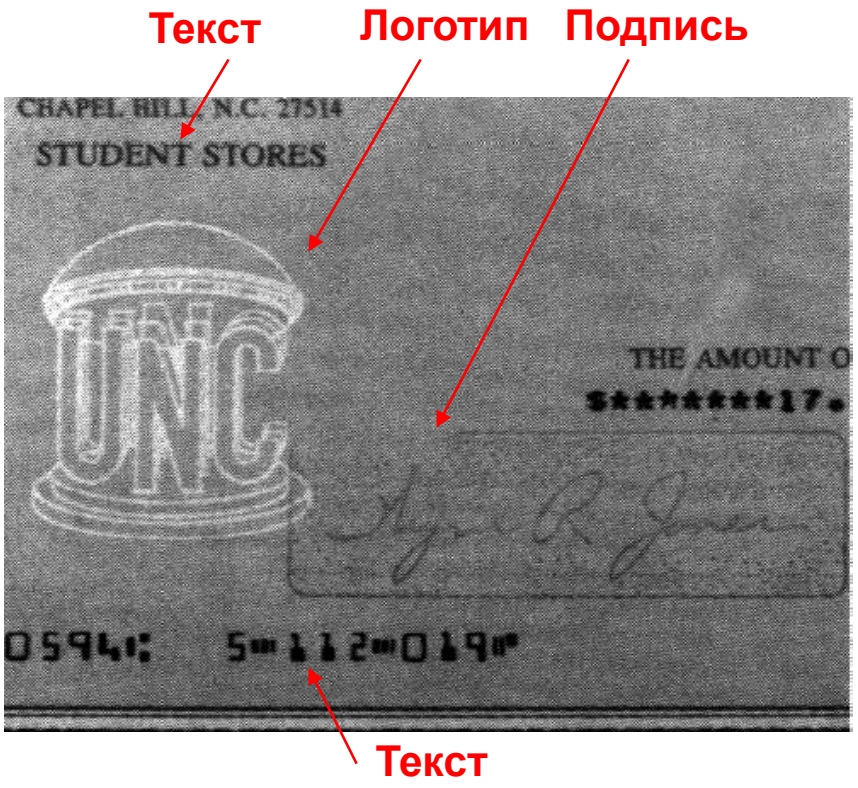
Алгоритм треугольника

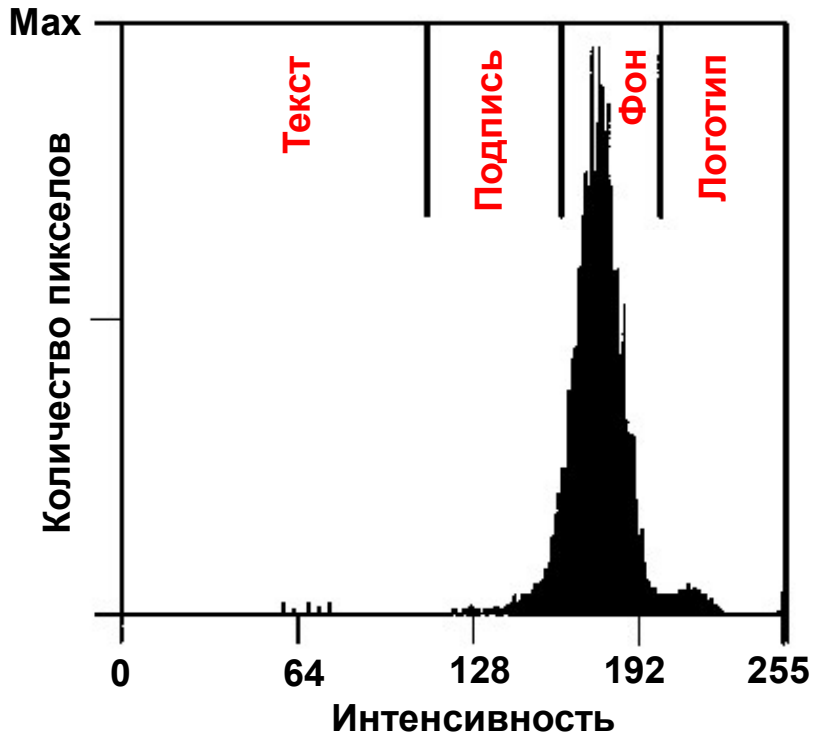
- **Автор**: Zack.
- Особенно эффективен, когда пикселы объекта определяют слабо выраженный пик на гистограмме.
- **Недостаток**: неэффективен для многопиковых гистограмм.



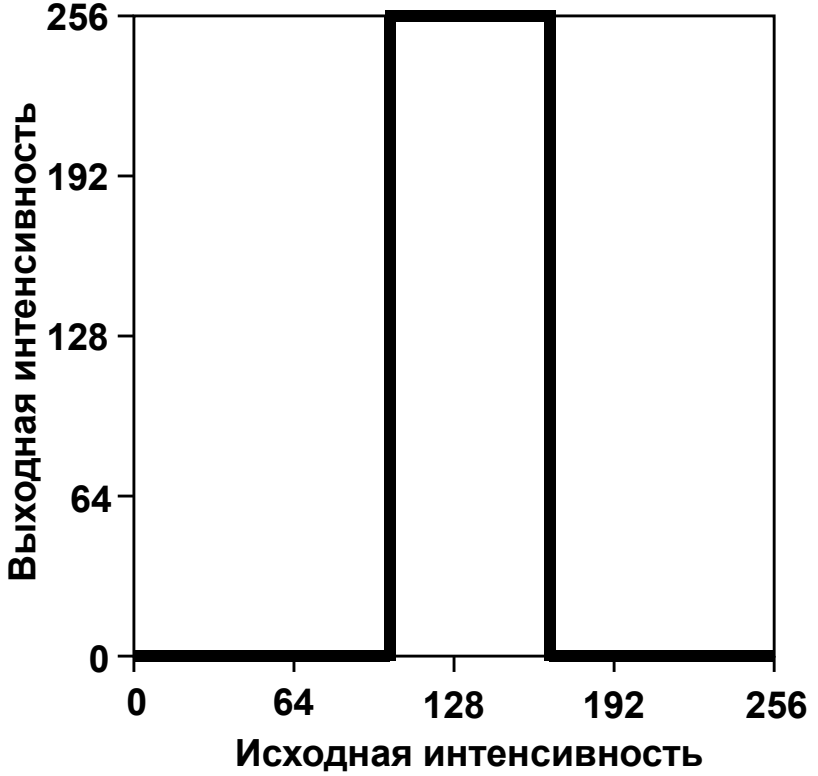
1. Строится прямая, проходящая через пик гистограммы $(b_{\max},\mathrm{h}(b_{\max}))$ и точку $(b_{\min},\mathrm{h}(b))$ – первое слева отличное от нуля значение гистограммы.
2. Для каждого из значений интенсивности $b\in[b_{\min},b_{\max}]$ вычисляется расстояние d от точки $(b,\mathrm{h}(b))$ до прямой.
3. В качестве порогового значения выбирается точка b_0, в которой расстояние d достигает максимума: $T=b_0$.

Пороговая сегментация для многопиковых гистограмм

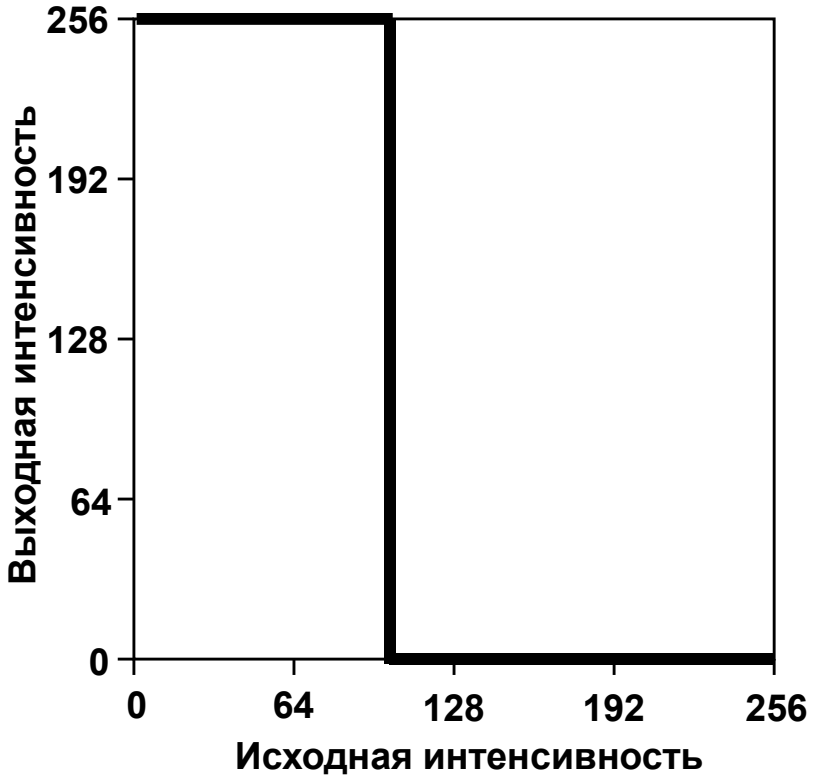




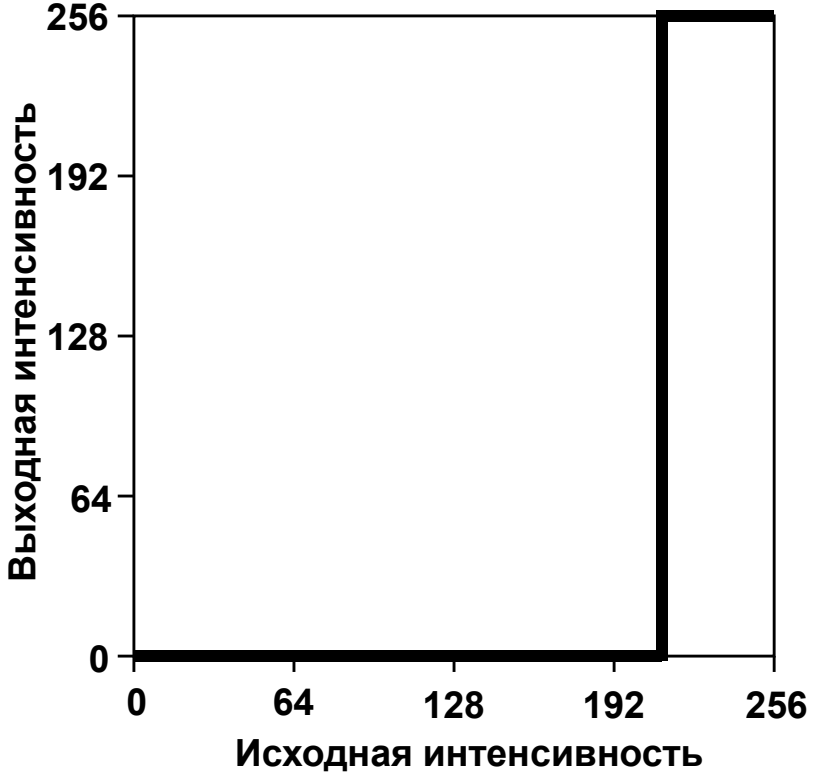
Препарирование изображения и пороговая сегментация



Препарирование изображения и пороговая сегментация



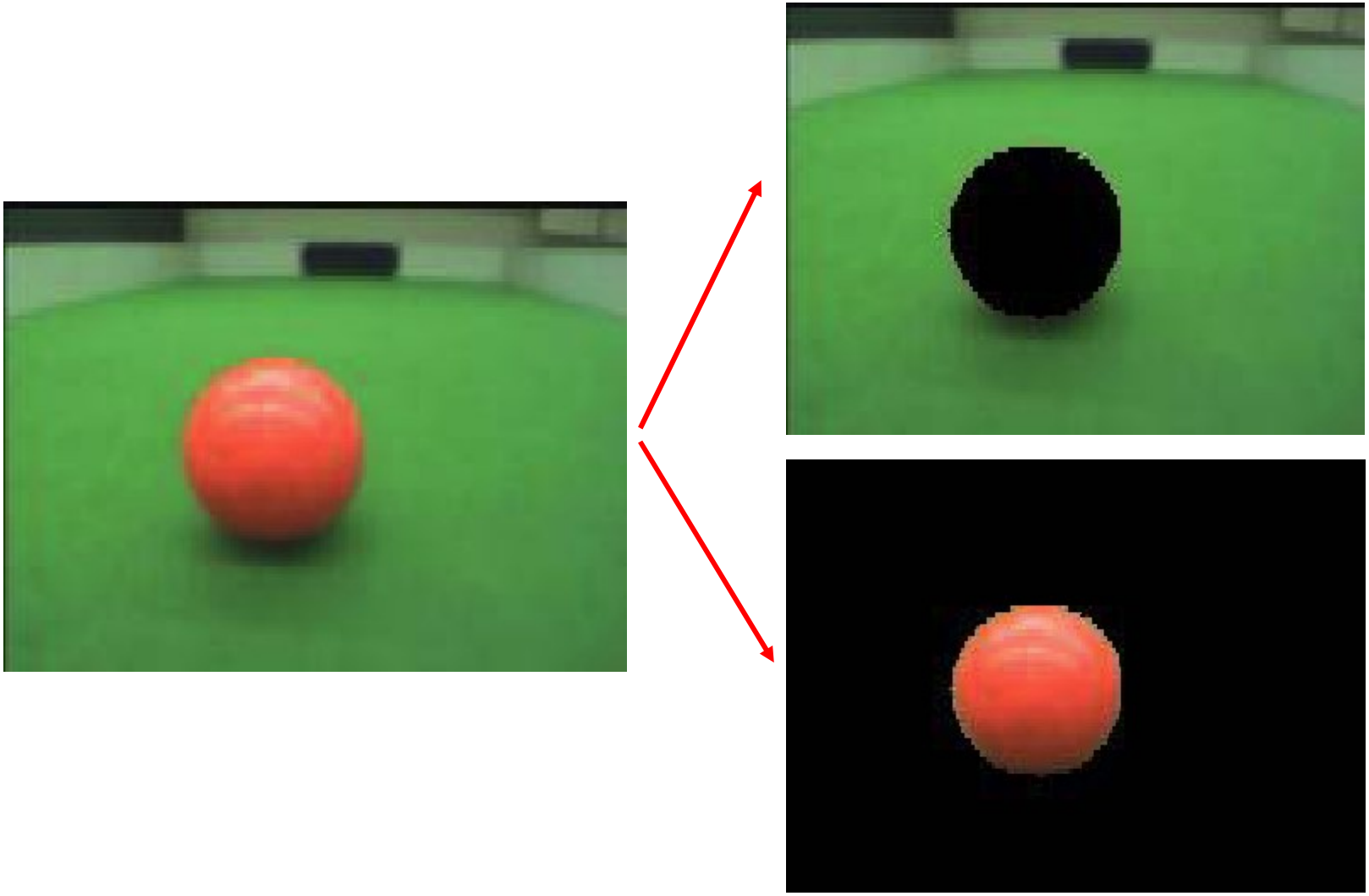
Препарирование изображения и пороговая сегментация



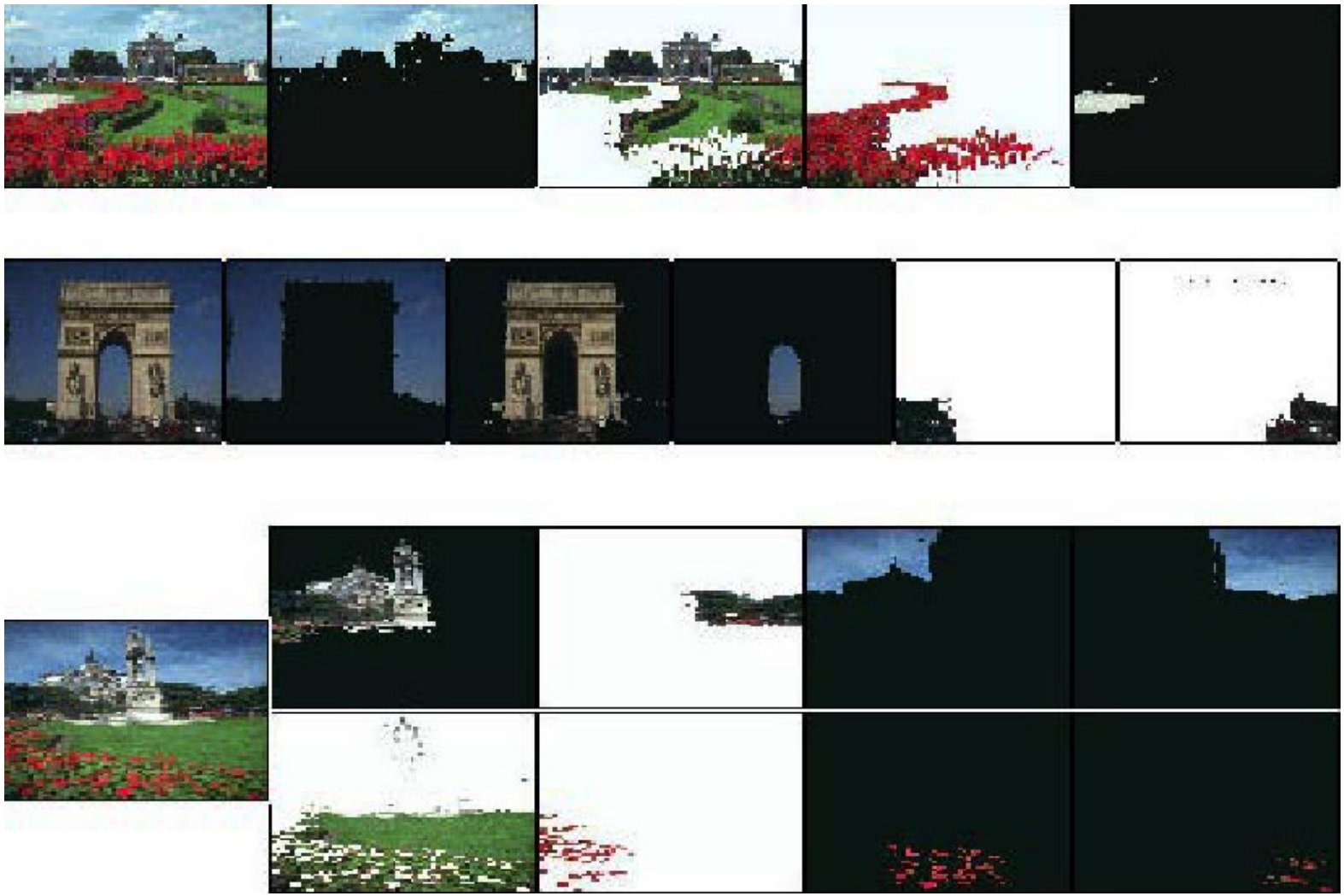
Рекурсивный алгоритм Оландера

- **Авторы**: Ohlander, Price, Reddy (1978)
- **Идея**: Сначала выполнить пороговую сегментацию полного изображения, затем – пороговую сегментацию каждой из полученных областей рекурсивно, до тех пор, пока возможна их сегментация.

Пороговая сегментация цветных изображений



Пороговая сегментация цветных изображений



Пороговая сегментация. Преимущества

- Простота алгоритма сегментации
- Простые алгоритмы выбора порога по гистограмме

Пороговая сегментация. Недостатки

- Трудности в определении количества классов на изображении
- Неоднозначность выбора порога
- Независимость от пространственных свойств изображения.

!

Пороговая сегментация. Некоторые проблемы и их решения

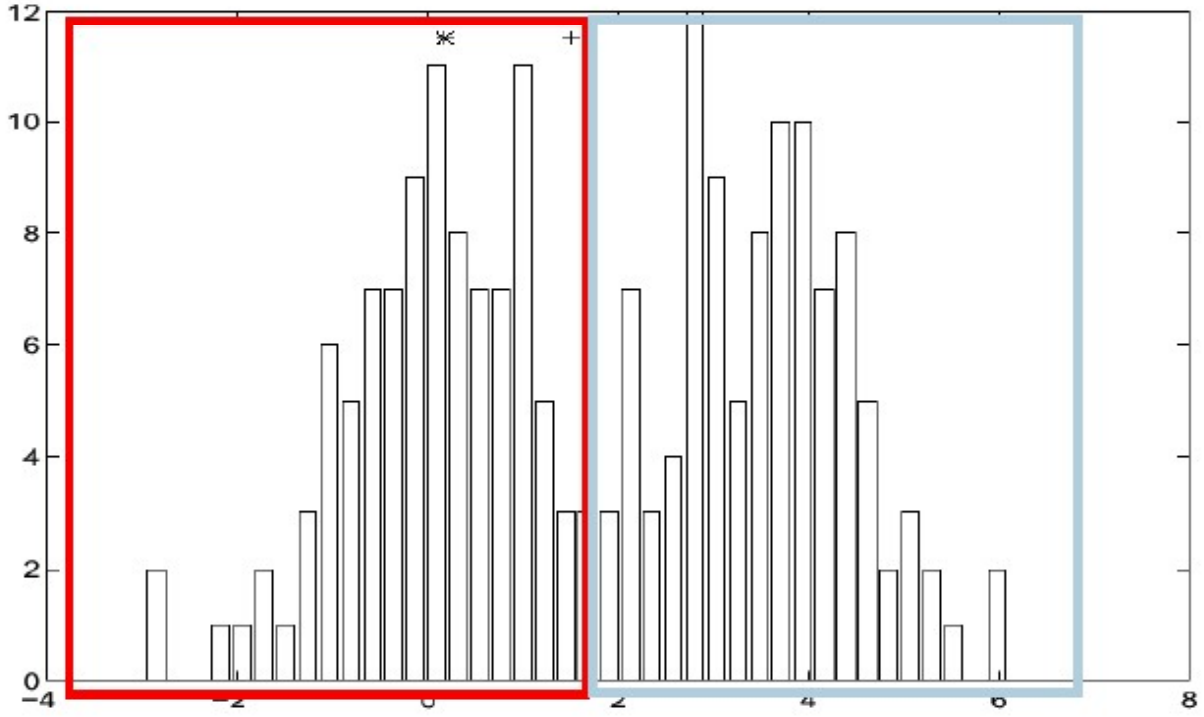
- Зашумленность гистограммы
- Неоднородный фон

Зашумленность гистограмм

Проблема.

Сколько пиков на гистограмме (областей на изображении)?

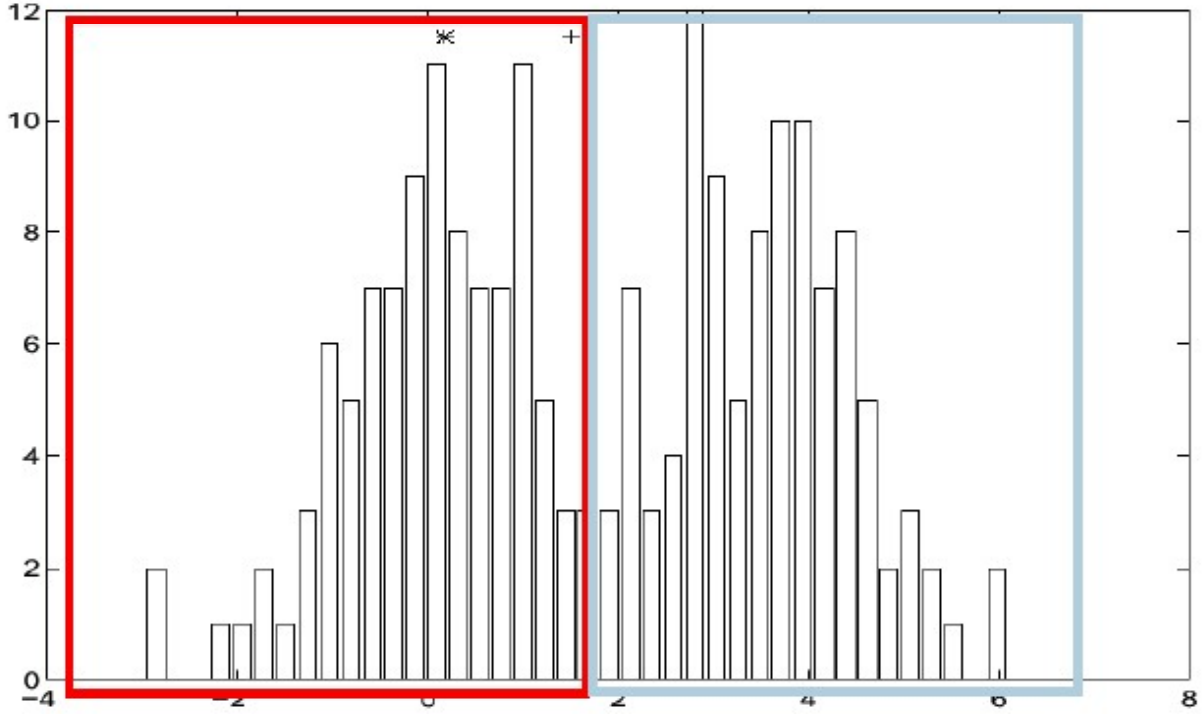
Часто пороговые значения легко увидеть, но тяжело вычислить.



Пороговая сегментация. Недостатки

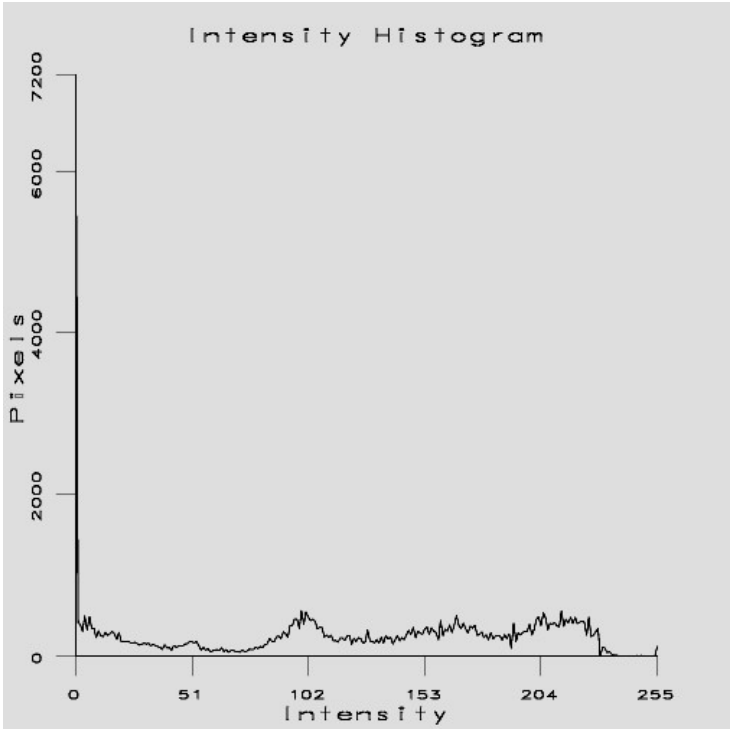
Возможные решения.

1. Используя детекторы контуров, исключить все пикселы, являющиеся граничными, из гистограммы.
2. Сгладить гистограмму.



Зашумленная гистограмма: пример

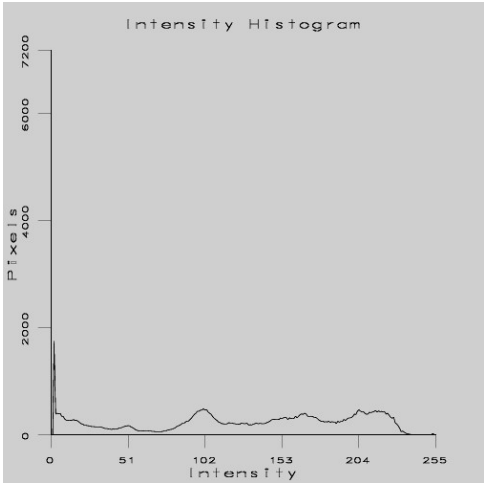
- Много «лишних» локальных максимумов



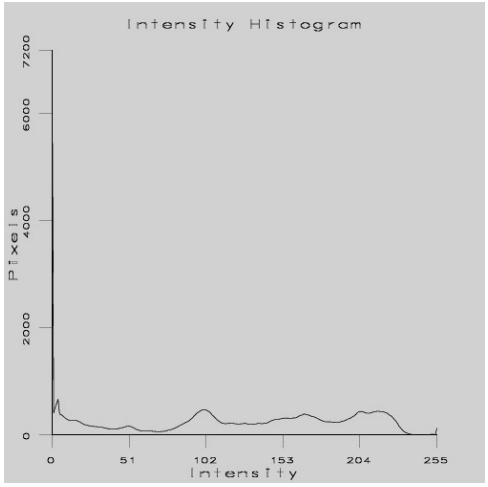
93 пика

Зашумленная гистограмма: пример

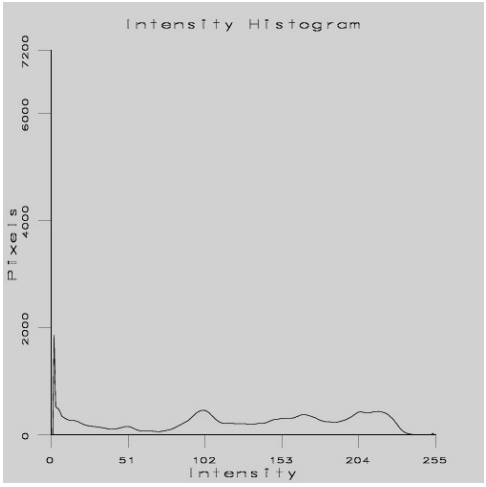
- Сглаживание посредством усреднения соседних значений
  - *Свертка одномерным box-фильтром*



Сглажено 1 раз



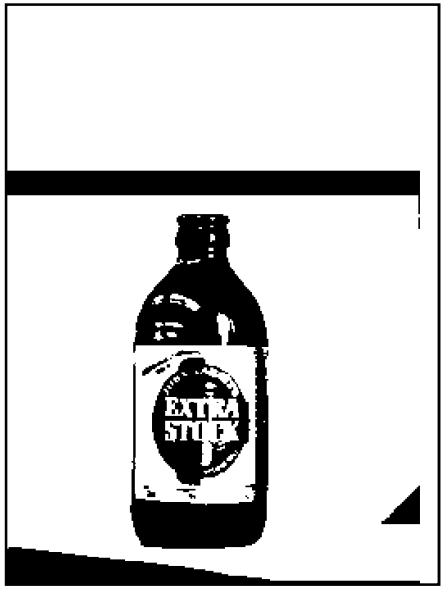
2 раза

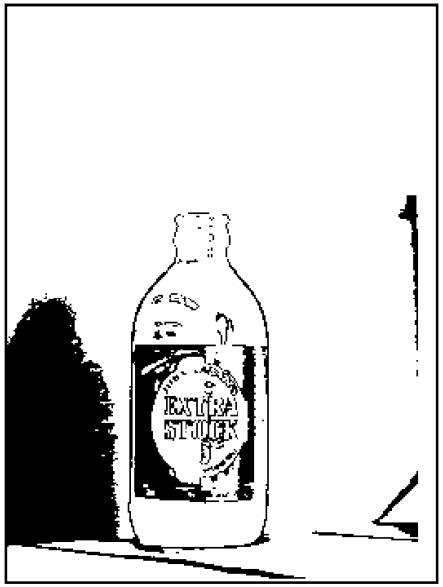


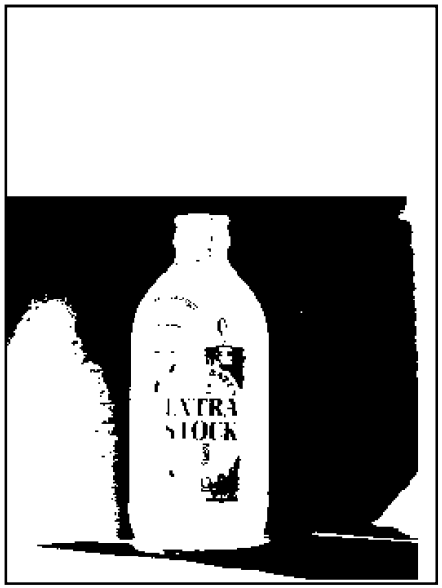
3 раза

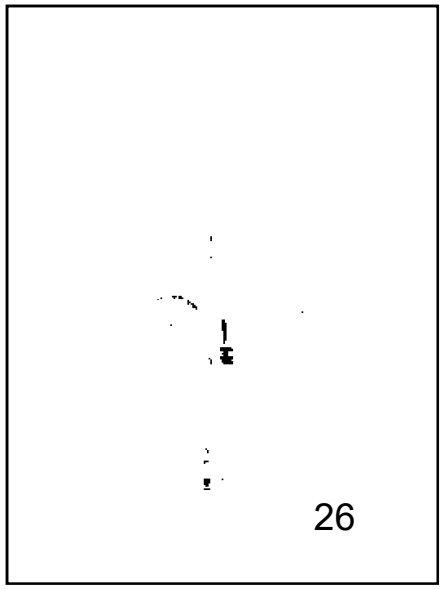
Зашумленная гистограмма: пример

- Области, найденные по 4 пикам





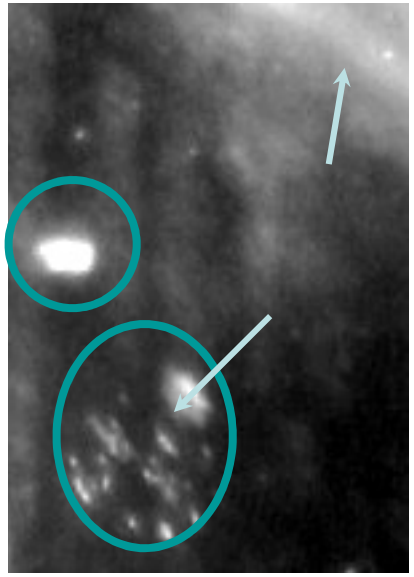


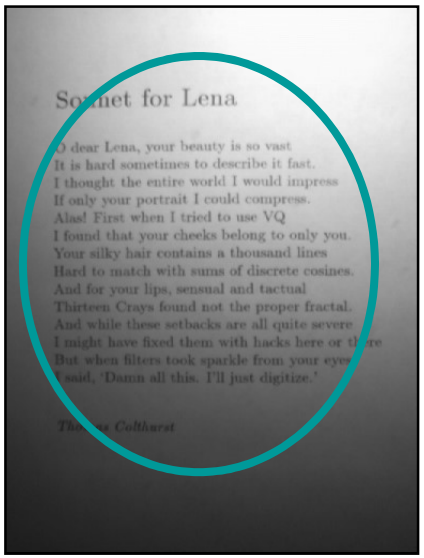


Неоднородность фона и адаптивный порог

Проблема:

- Яркость фона может быть разной в разных частях изображения
- Единый порог не подходит



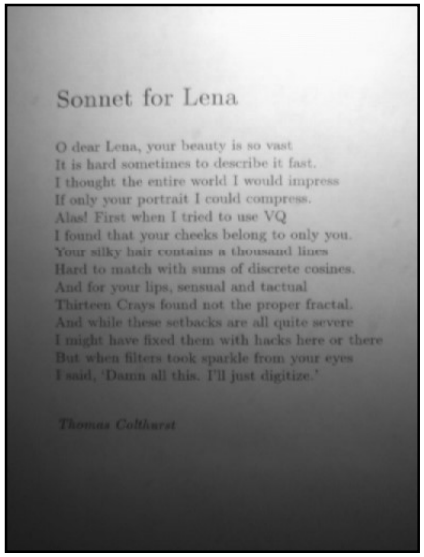


Неоднородность фона и адаптивный порог

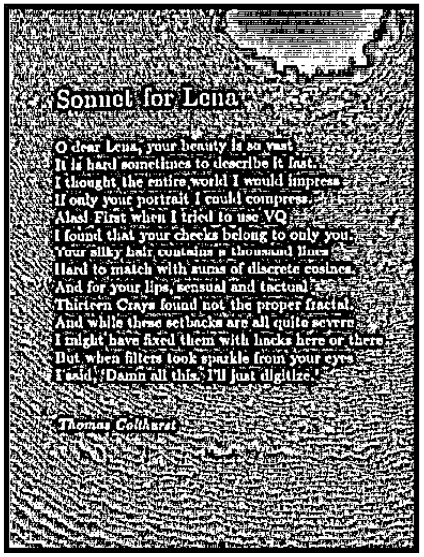
- Для каждого пикселя изображения ***I(x, y)***
  - В окрестности радиуса ***r*** высчитывается индивидуальная для данного пикселя величина ***C***;
  - Если ***I(x, y) - C > T*** , результат 1, иначе 0;

- Варианты выбора ***C*** по окрестности (x, y):
  - ***C*** = *среднее*
  - ***C*** = *медиана*
  - ***C*** = *(min + max) / 2*

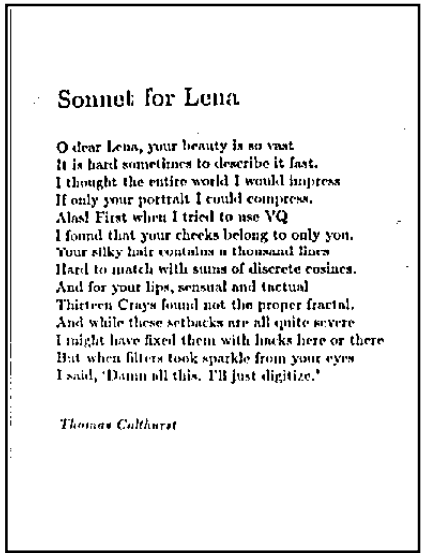
Неоднородность фона и адаптивный порог



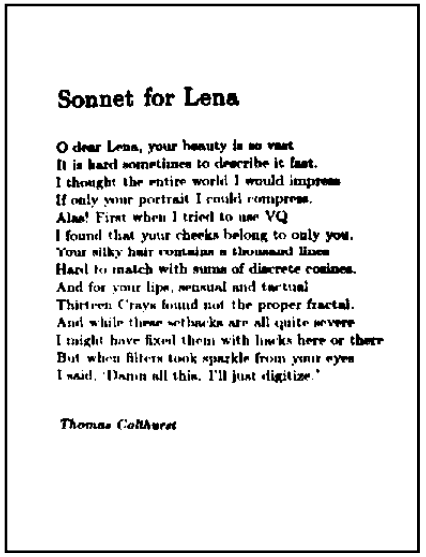
Исходное



r=7, T=0



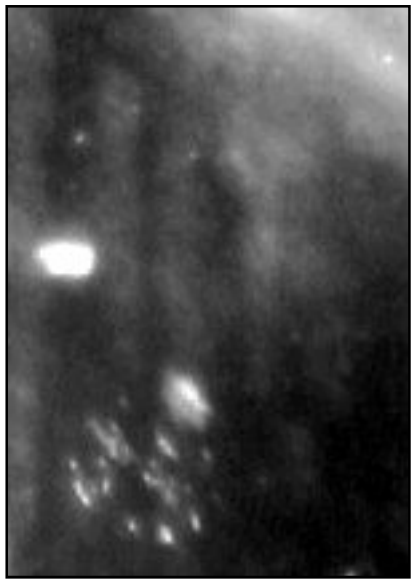
r=7, T=7



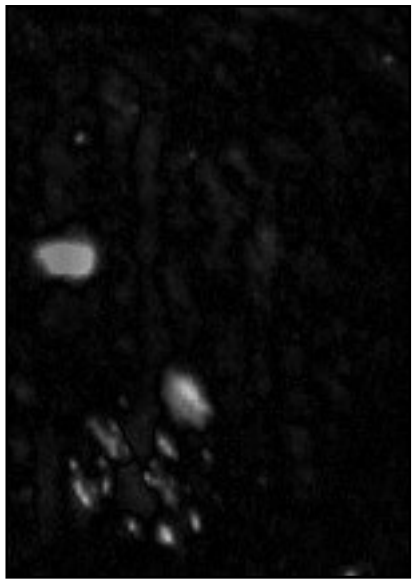
r=75, T=10

Неоднородность фона и адаптивный порог

- Другая формулировка
  - Приближение фона усреднением
  - Вычитание фона: ***I(x, y) – C(x,y) > T***



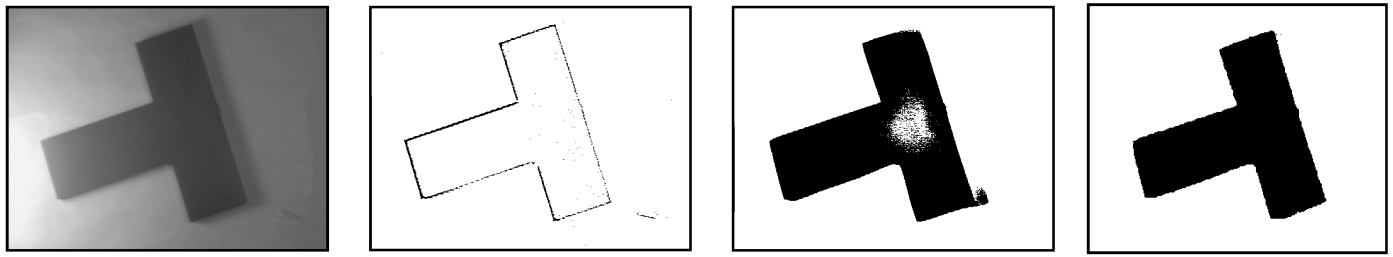
Исходное



I(x,y) - C(x,y), r=18

Неоднородность фона и адаптивный порог

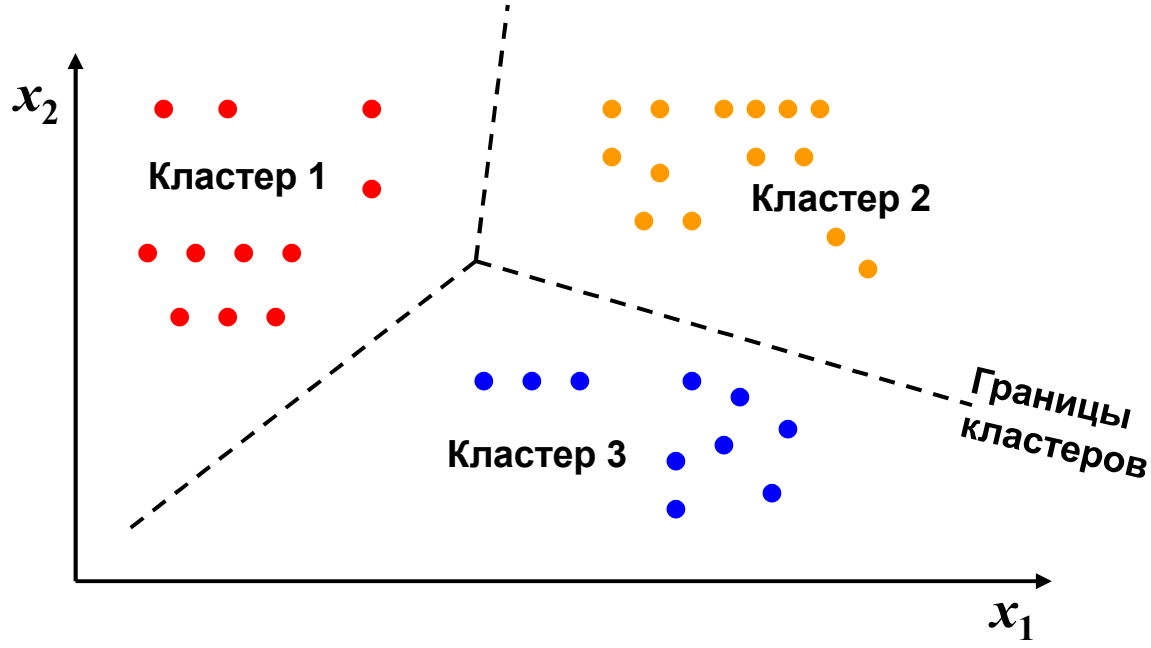
- Адаптивный порог хорошо работает
  - Когда размер искомого объекта заметно меньше размера оцениваемой окрестности
- Адаптивный порог хуже работает
  - Когда объект велик по сравнению с самим изображением



Исходное r=7 r=140 r=300

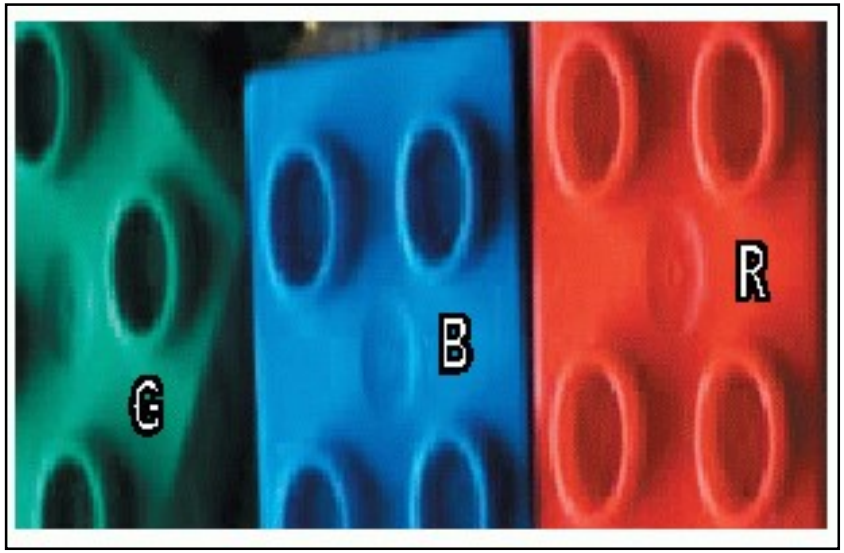
Сегментация = кластеризация

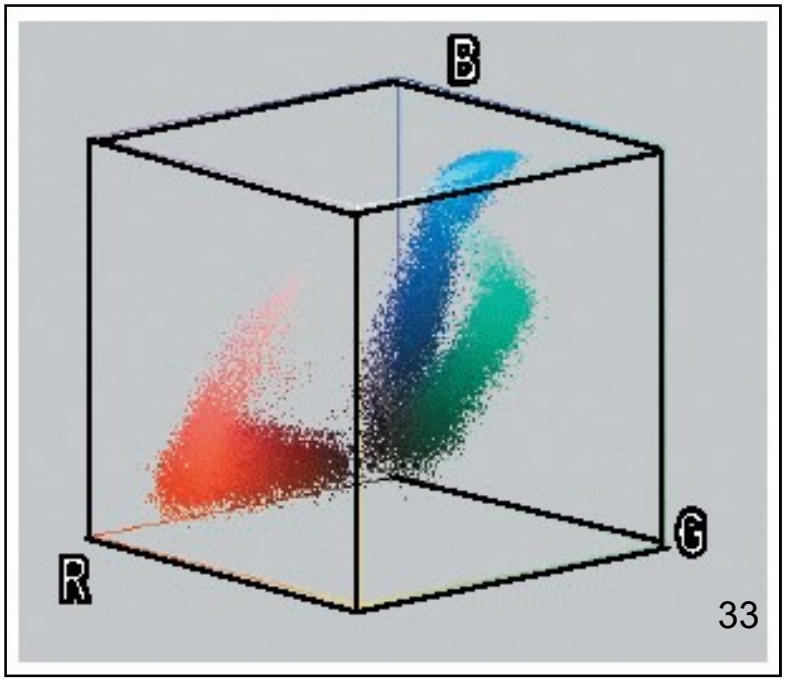
1. Каждый пиксел характеризуется многомерным вектором признаков (интенсивность, цвет, пространственные взаимосвязи и т.п.)
2. В кластеры (области) объединяются векторы (пикселы), со статистически подобными характеристиками.



Сегментация = кластеризация

1. Каждый пиксел характеризуется многомерным вектором признаков (интенсивность, цвет, пространственные взаимосвязи и т.п.)
2. В кластеры (области) объединяются векторы (пикселы), со статистически подобными характеристиками.





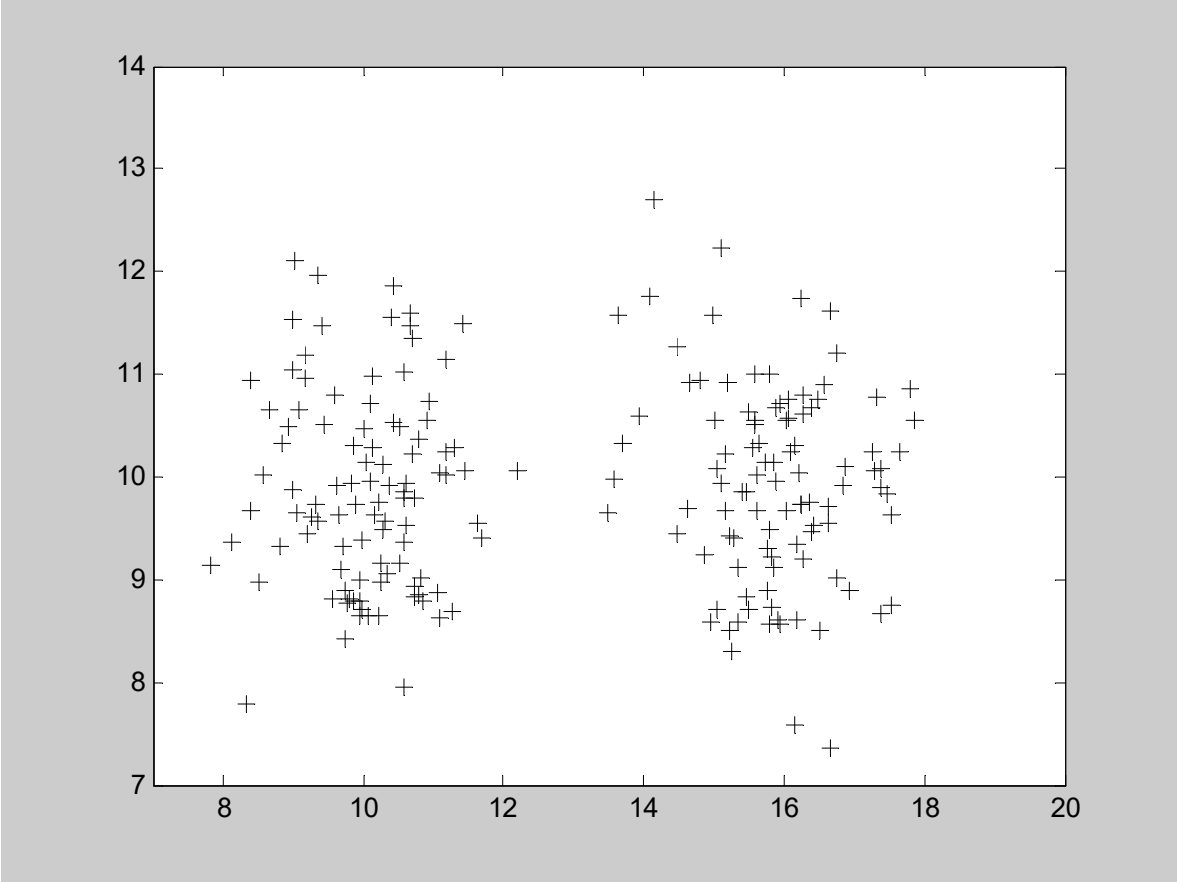
Алгоритм *K* средних

- **Входные данные:** набор n-мерных векторов v_i, $i = 1,\ldots,p$.
- **Выходные данные:** центры кластеров m_j, $j = 1,\ldots,K$ и принадлежность v_i к кластерам

1. Случайным образом выбрать K средних m_j, $j=1,\ldots,K$.
2. Для каждого v_i, $i=1,\ldots,p$ подсчитать расстояние до каждого из m_j, $j=1,\ldots,K$.
3. Отнести (приписать) v_i к кластеру j' такому, что расстояние до m_j, минимально.
4. Пересчитать средние m_j, $j=1,\ldots,K$ по всем кластерам.
5. Повторять шаги 2, 3 пока кластеры не перестанут изменяться.

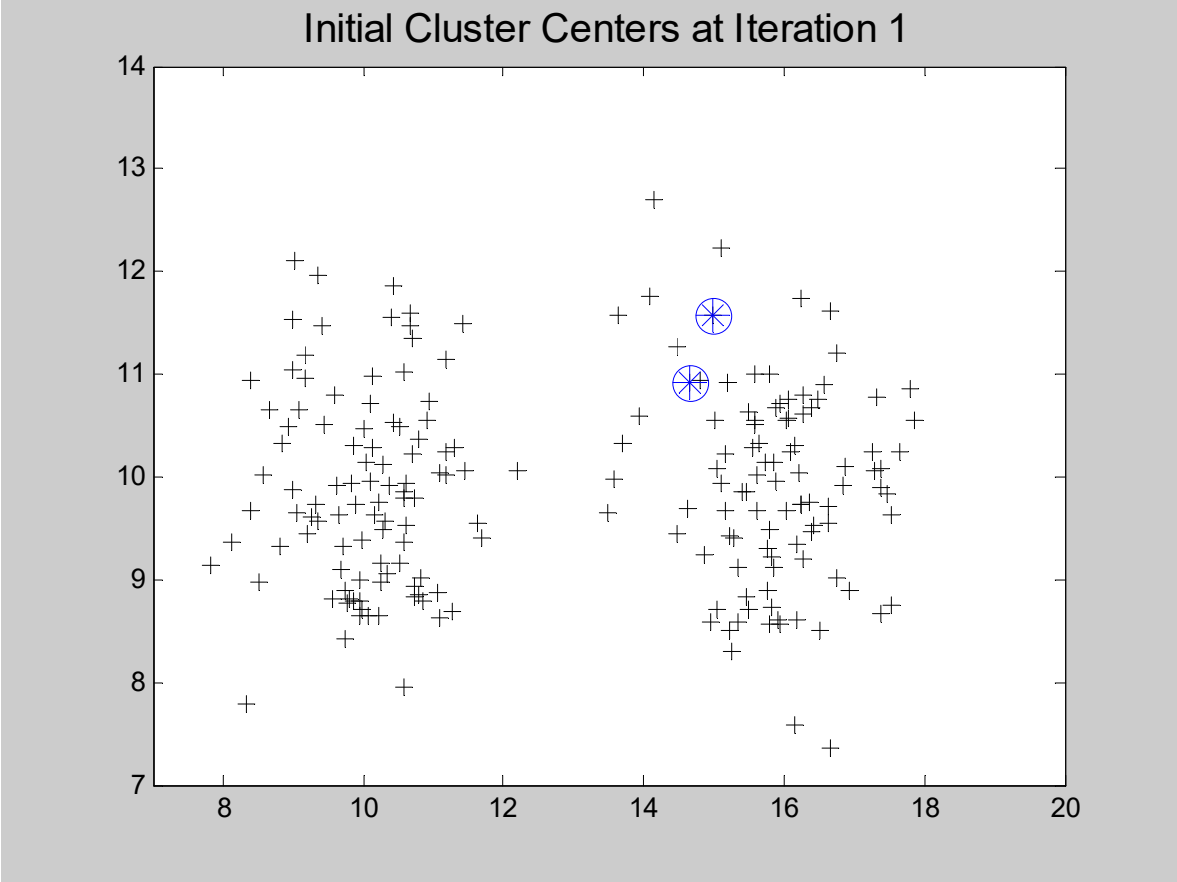
Пример кластеризации в 2D

Исходные данные



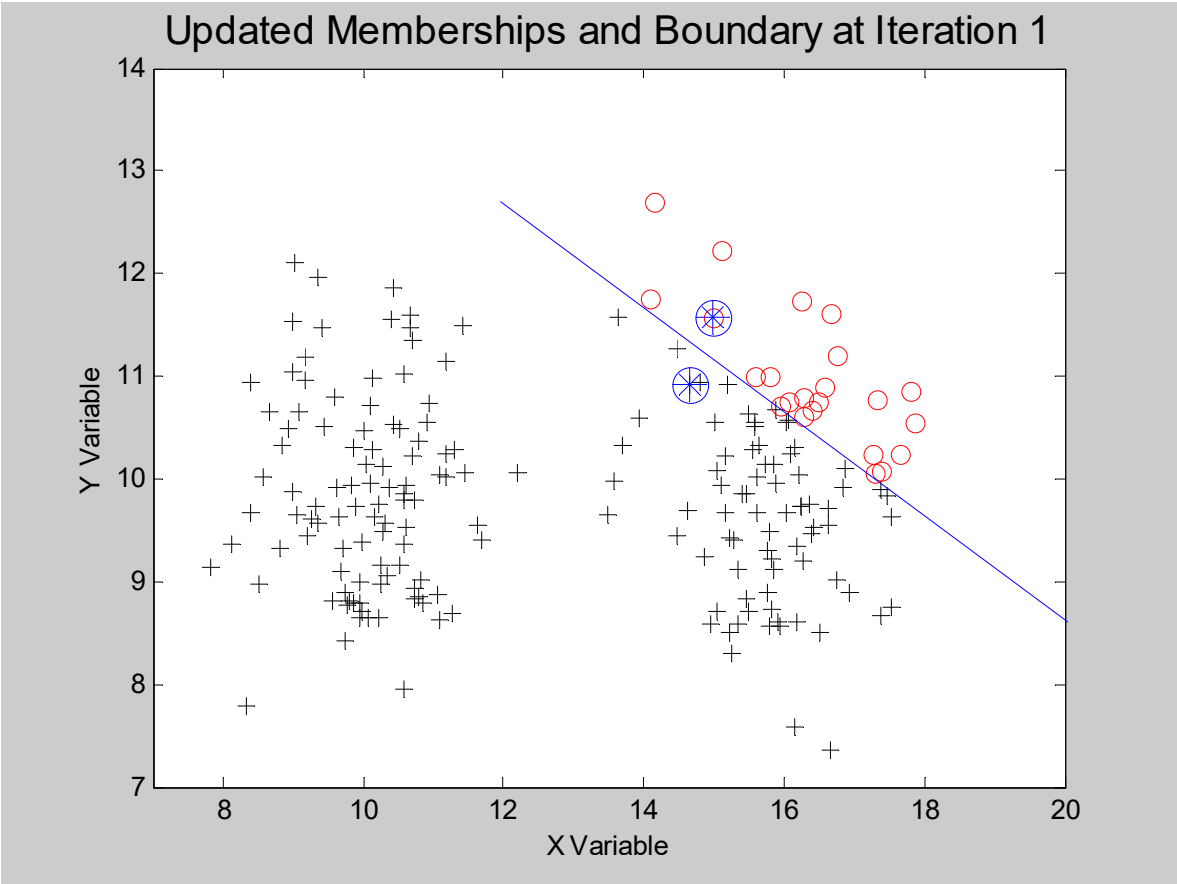
Пример кластеризации в 2D

Случайная инициализация центров кластеров (шаг 1)



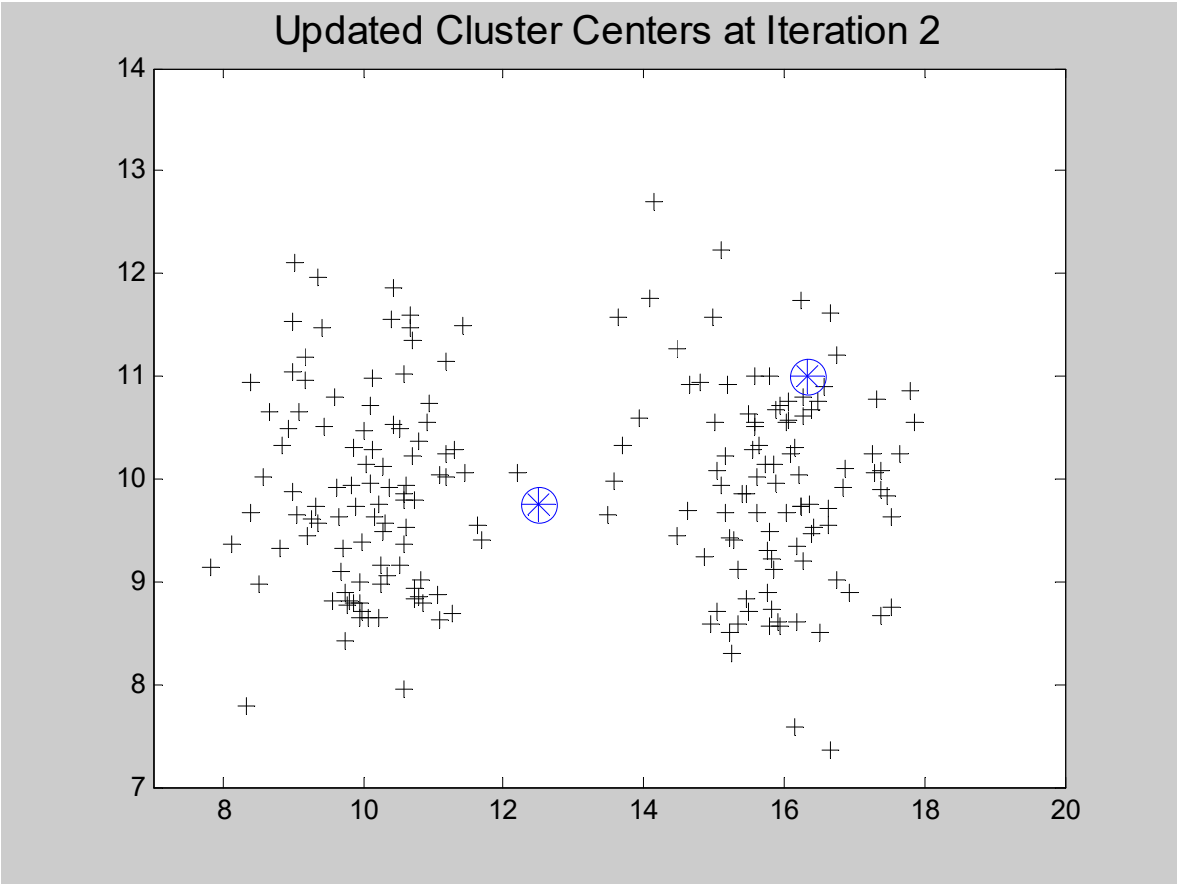
Пример кластеризации в 2D

Кластеры после первой итерации (шаг 2)



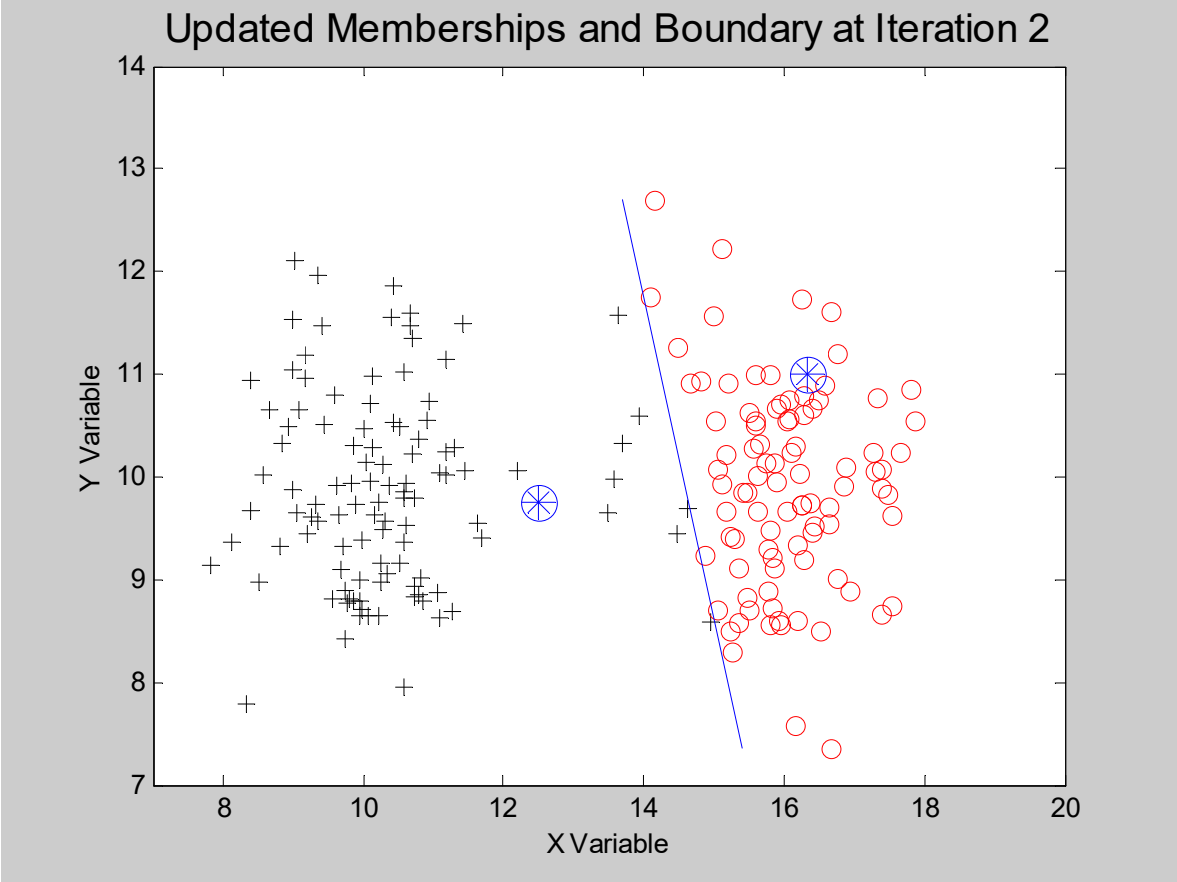
Пример кластеризации в 2D

Пересчет центров кластеров после первой итерации (шаг 3)



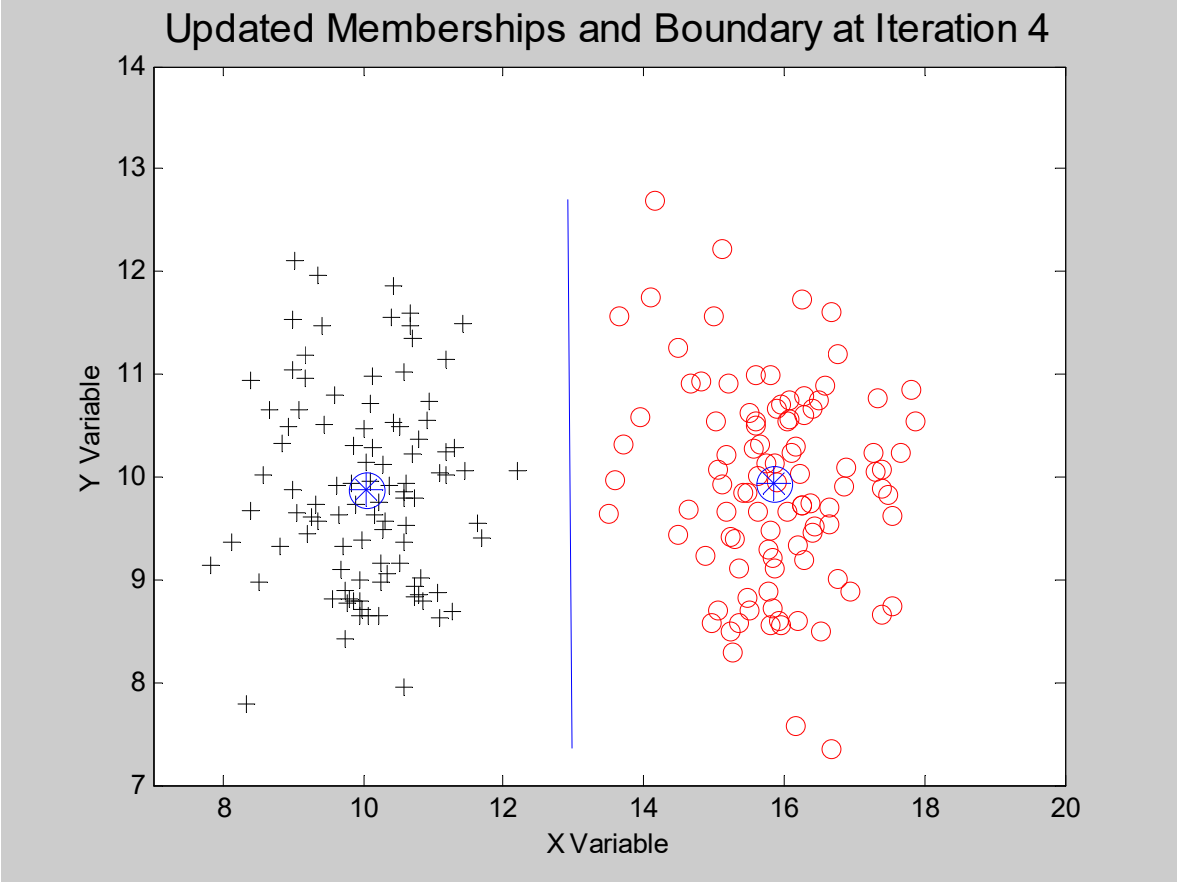
Пример кластеризации в 2D

Кластеры после второй итерации (шаг 2)



Пример кластеризации в 2D

Стабильная конфигурация после четвертой итерации



Алгоритм *K* средних

- Алгоритм гарантированно заканчивается, но не всегда приводит к глобальному оптимуму.

- Шаг 1 может быть заменен на разбиение исходного множества векторов на K кластеров случайным образом и вычисление их средних.

- В качестве условия окончания алгоритма может быть использовано достижение ситуации, когда лишь малый процент всех векторов «меняют» кластеры

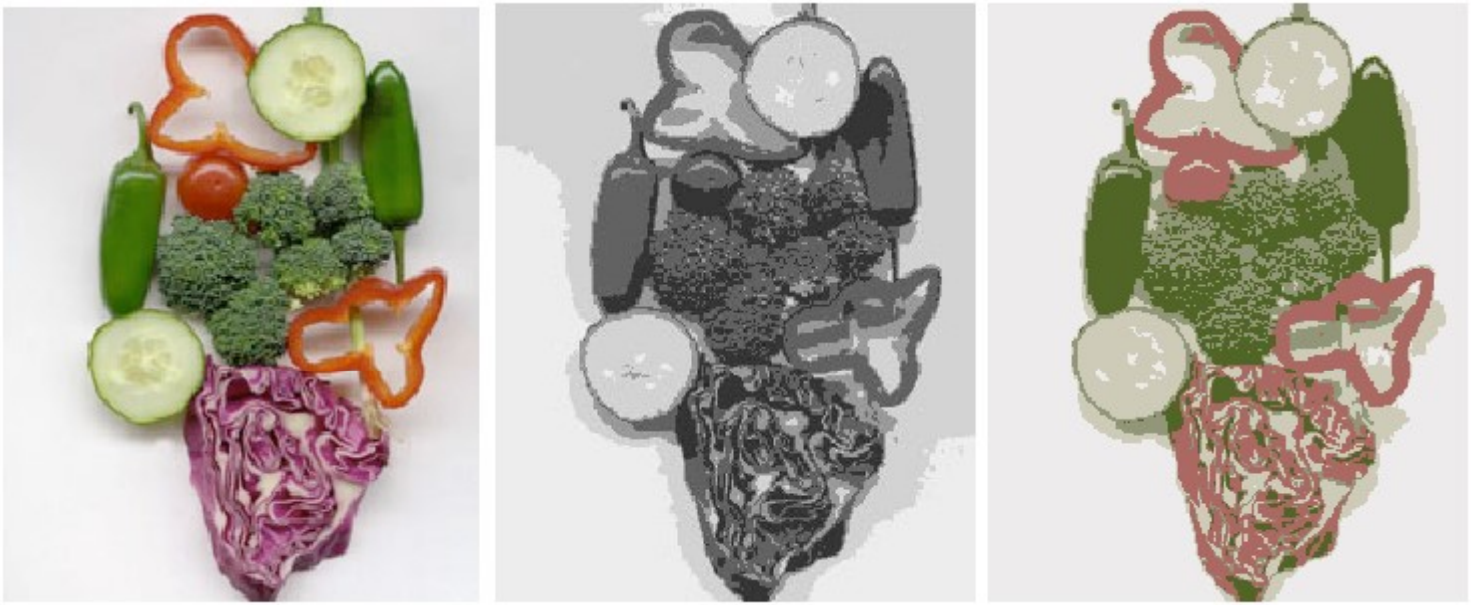
Сегментация методом *K* средних

- Если изображения трехканальное (RGB)
  - $v_i = (R(x, y), G(x,y), B(x, y))$ – работаем в трехмерном пространстве
- Состав признаков можно расширять
  - RGB + вычислимые признаки
- Если изображение одноканальное
  - $v_i = I(x, y)$ – работаем в одномерном пространстве
  - Получается итеративный алгоритм пересчета порога

Пример сегментации методом *K* средних

K=6. Шесть кластеров соответствуют шести главным цветам на исходном изображении: темно-зеленому, светло-зеленому, темно-синему, белому, серому и черному

Пример сегментации методом *K* средних



Сегментация исходного изображения со смесью овощей (слева) дает изображения в центре и справа. На сегментированных изображениях каждый пиксел заменен средним значением соответствующего кластера. Изображение в центре получено только с учетом интенсивности пикселов. Изображение справа получено с привлечением цветовой информации. В каждом из случаев строится 5 кластеров

Пример сегментации методом *K* средних

Результат сегментации при *k*=11. На левом изображении все классы совмещены, значения пикселов соответствуют средним значениям соответствующих кластеров. На остальных изображениях – 4 сегмента по отдельности. Следует заметить, что метод дает не обязательно связные сегменты. Для данного изображения полученные сегменты почти соответствуют объектам, хотя в некоторых случаях один сегмент может представлять несколько объектов (см. перцы). Отсутствие в векторе признаков текстурной информации создает серьезные трудности: срез капусты разбит на много отдельных сегментов

Алгоритм кластеризации Isodata

- *Isodata* – итеративный алгоритм, использующий технику разбиения и слияния
- Предполагается, что имеется K кластеров C_1, C_2, …,C_k со средними значениями m_1, m_2,…,m_k, а Σ_k – ковариационная матрица кластера *k*.

Алгоритм кластеризации Isodata

$$x_i = (v_1, v_2, ..., v_n)$$

$$m_k = (m_{1k}, m_{2k}, ..., m_{nk})$$

$$\Sigma_k = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{12} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \sigma_{1n} & \sigma_{2n} & \cdots & \sigma_{nn} \end{pmatrix}$$

где $\sigma_{ii} = \sigma_i^2$ – вариация i-й компоненты v_i;

$\sigma_{ij} = \rho_{ij}\sigma_i\sigma_j$ – ковариация i-й и j-й компонент;

ρ_{ij} – коэффициент корреляции i-й и j-й компонент;

σ_i, σ_j – стандартные отклонения i-й и j-й компонент

Алгоритм кластеризации Isodata

Формирует изодата-кластеры в множестве n-мерных векторов

1. Приписать x_i к кластеру l такому, что достигается минимум для

$$D_\Sigma = (x_i - m_i)' \Sigma_l^{-1} (x_i - m_i)$$

2. Слить кластеры i и j, если

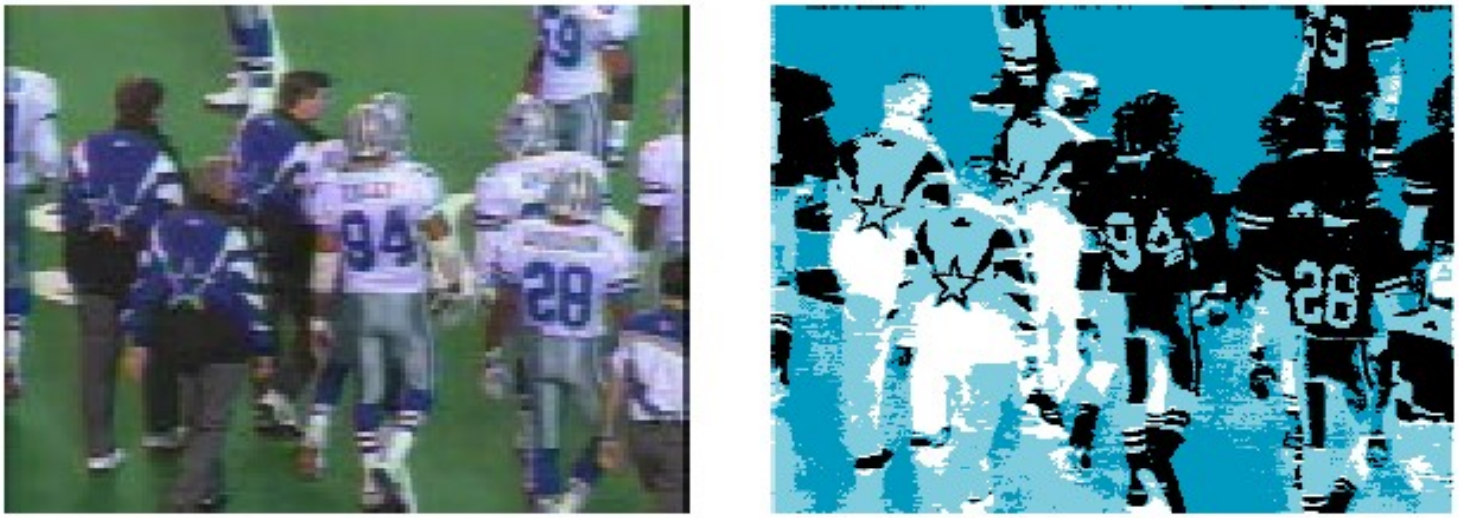
$$\left| m_i - m_j \right| < \tau$$

3. Разбить кластер k, если максимальное собственное значение σ_k больше, чем τ.

4. Остановиться, когда

$$\left| m_i(t) - m_i(t+1) \right| < \varepsilon$$

для каждого i-го кластера или при достижении максимально допустимого количества итераций.

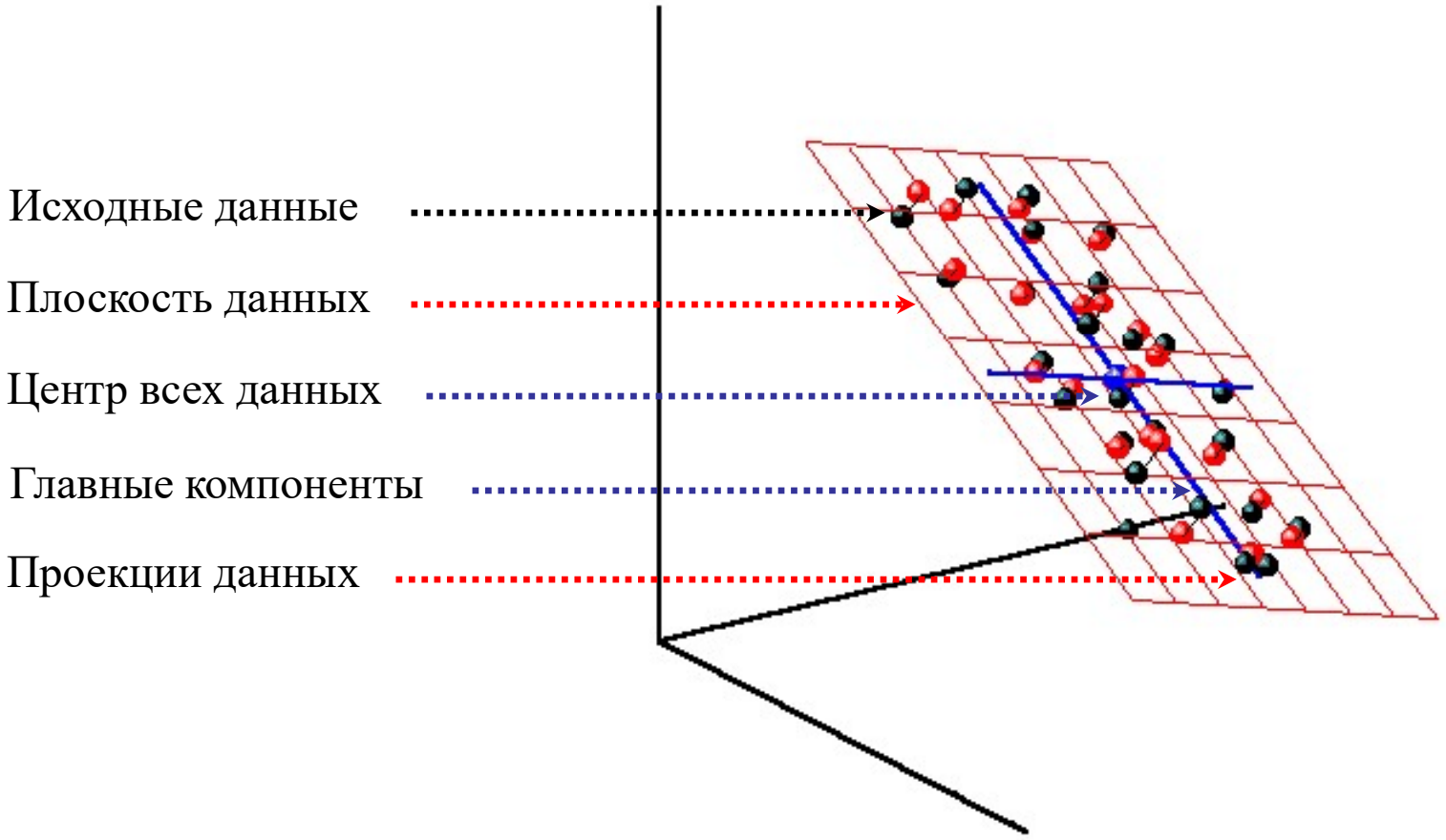


Изодата-кластеры при K=5. Кластеры соответствуют зеленому, темно-синему, белому, серому и черному

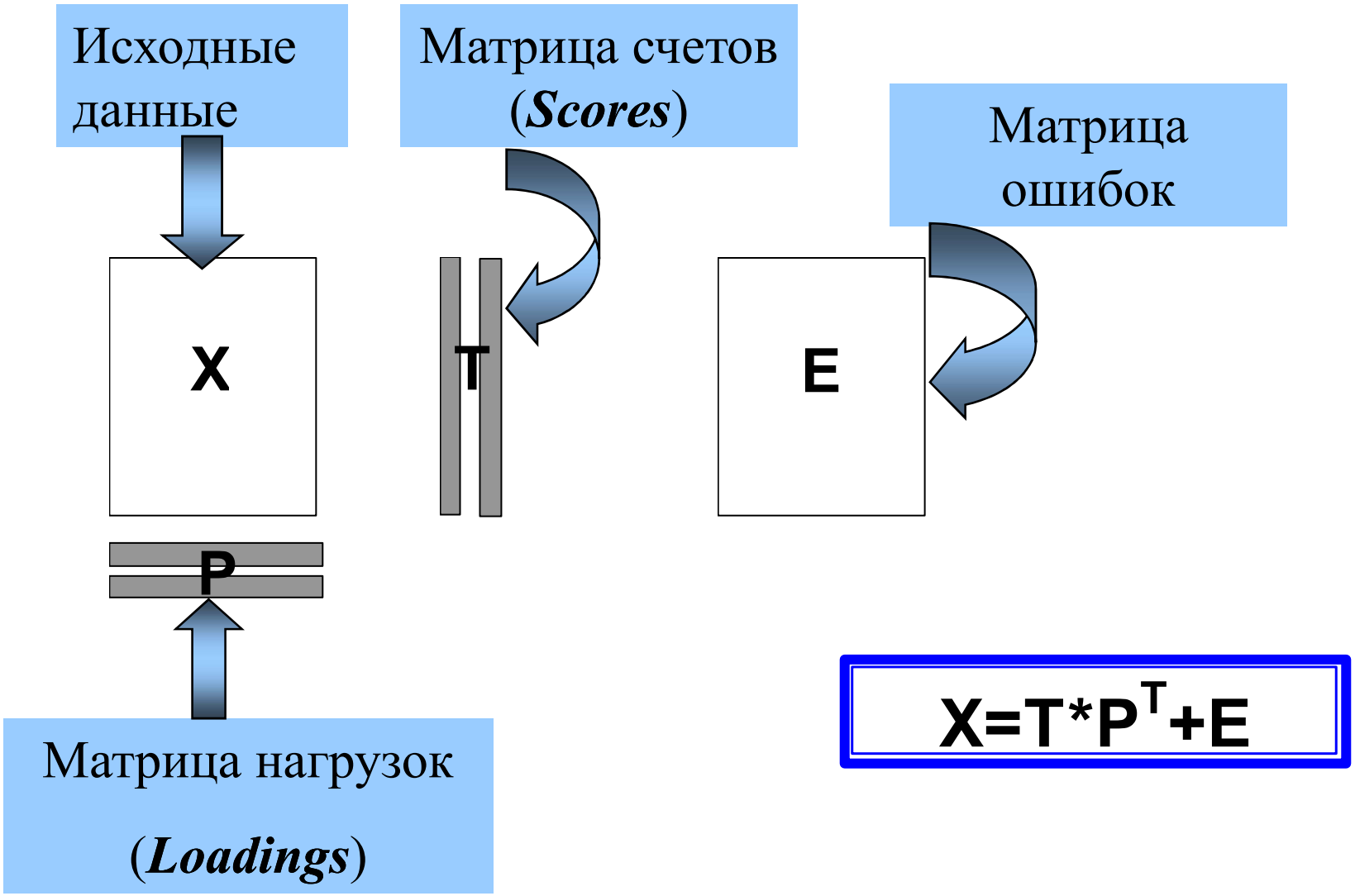
Кластеризация

- Преимущества
  - Высокая скорость
  - Адаптивность
  - Возможность привлечения дополнительной информации
- Недостатки
  - Сложность в подборе адекватных признаков
  - Необходимость предварительного знания количества кластеров

Проекционные методы

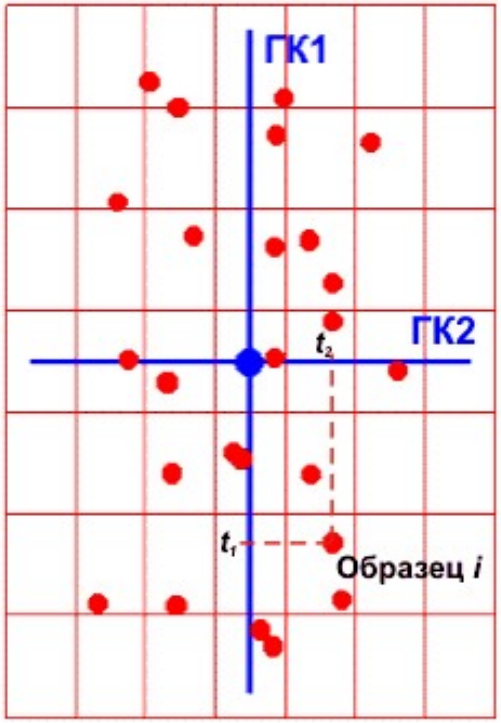


Метод главных компонент



Матрица счетов *T* (scores)

$$X=T*P^{T}+E$$



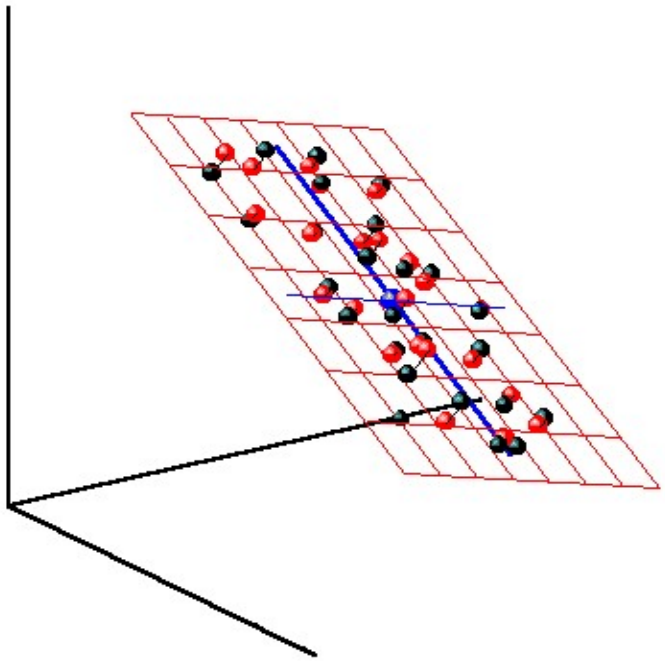
Строка – координаты одного объекта в новой системе координат

Столбец – проекция всех объектов на одну ось главных компонент

t_{11}	t_{12}
t_{21}	t_{22}
.	.
.	.
.	.
…	…
t_{n1}	t_{n2}

Матрица нагрузок *P* (loadings)

$$X=T*P^{T}+E$$

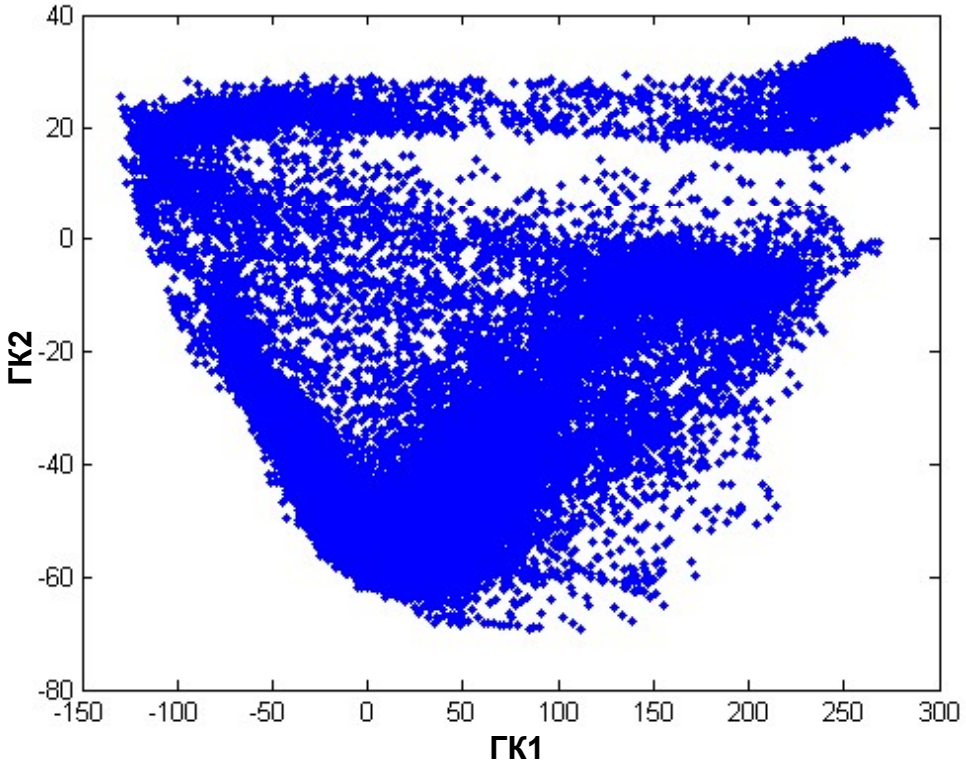


p_{11}	p_{12}		…	p_{1m}
p_{21}	p_{22}		…	p_{2m}

P**^T^*- матрица перехода из пространства ***X в пространство главных компонент

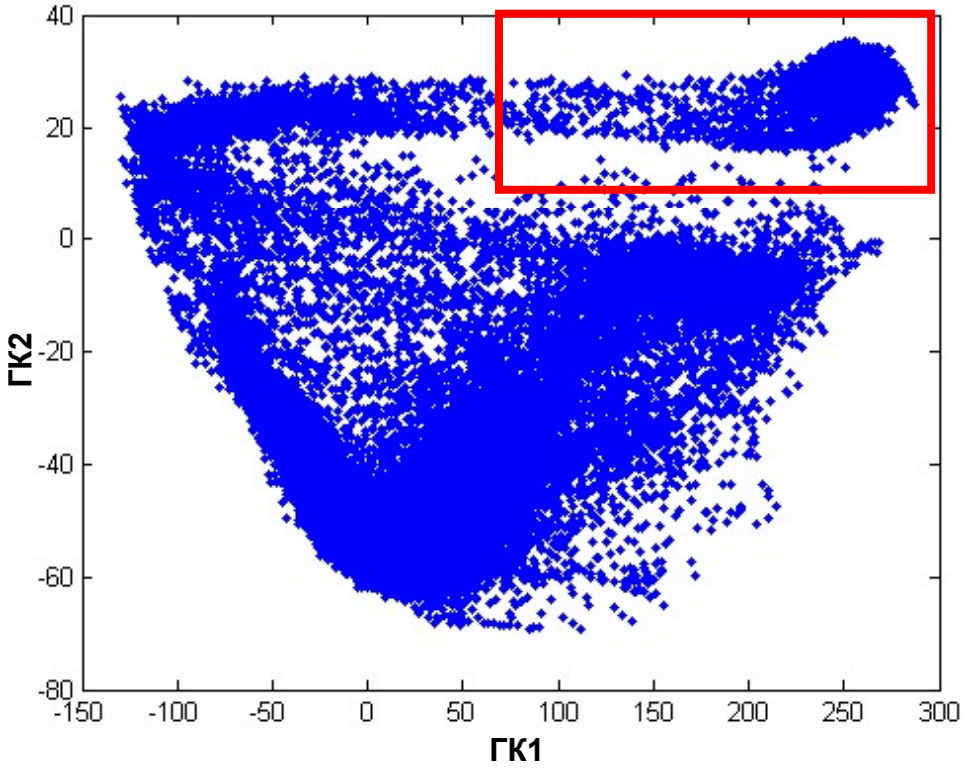
Пример использования МГК

- Сегментация изображения



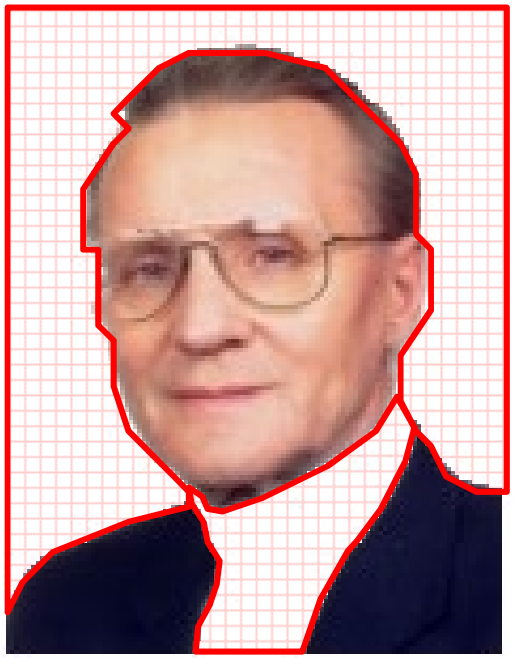
Пример использования МГК

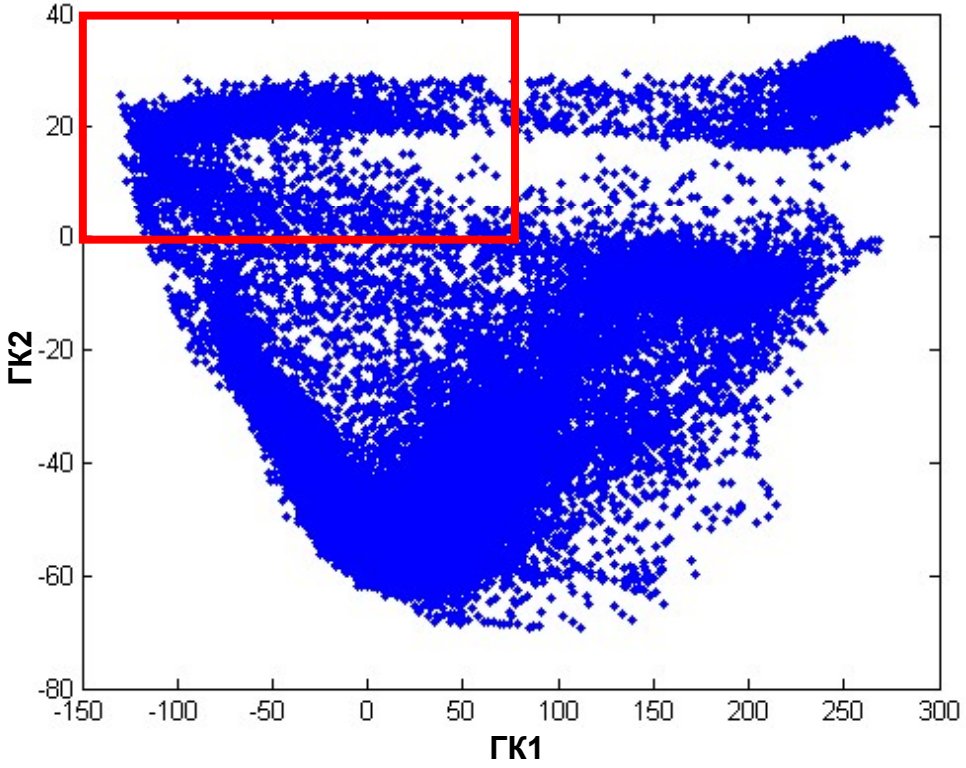
- Сегментация изображения



Пример использования МГК

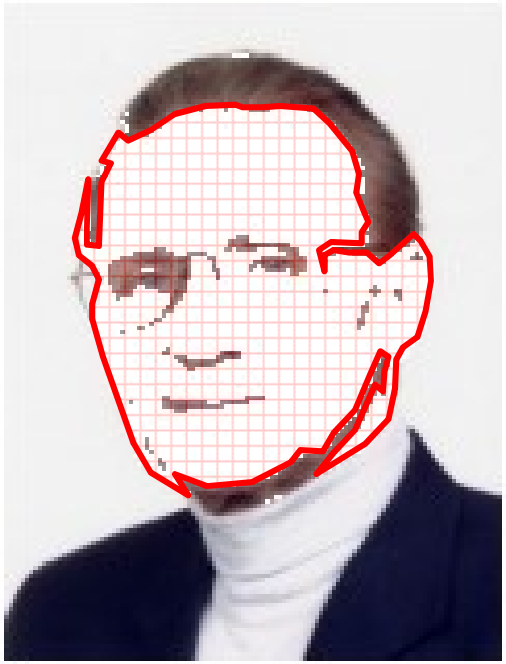
- Сегментация изображения

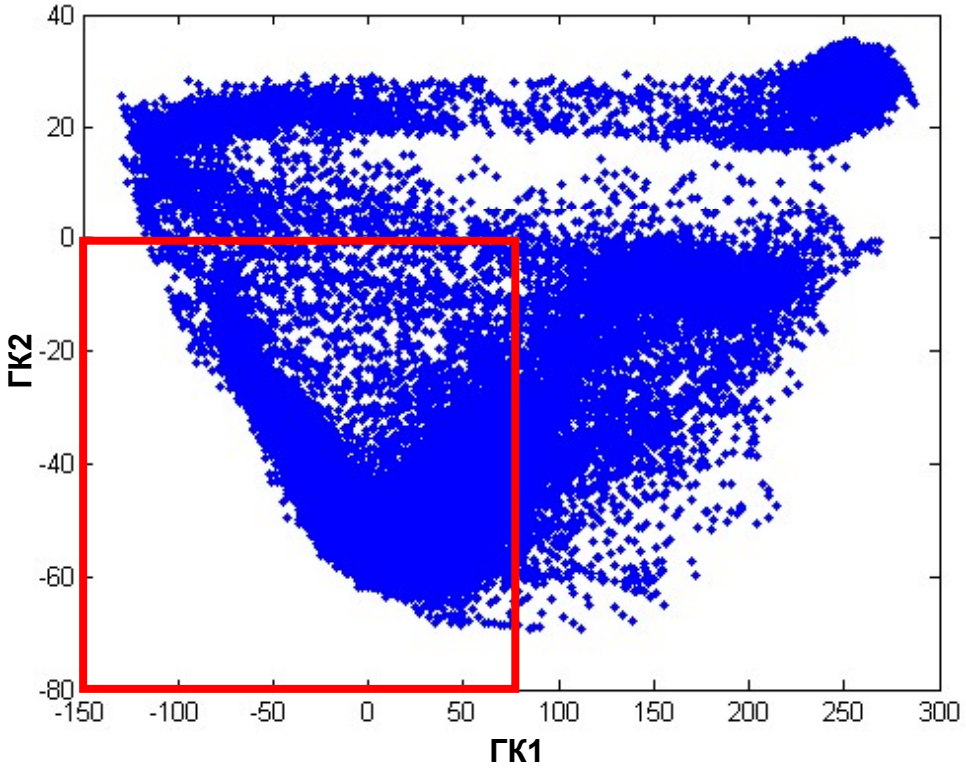




Пример использования МГК

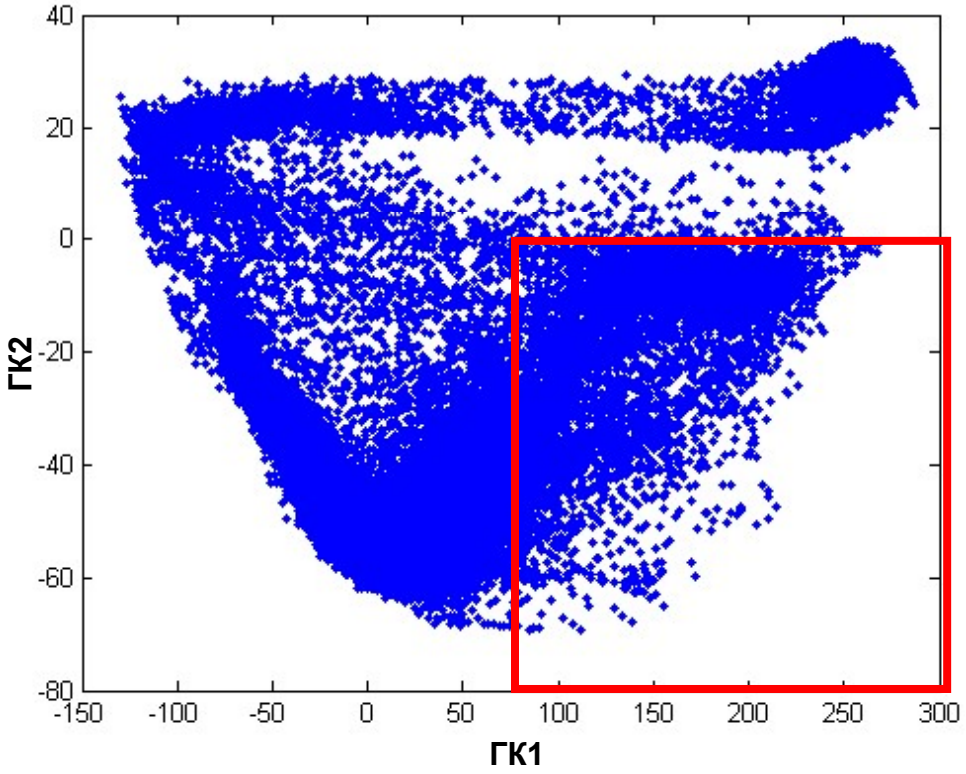
- Сегментация изображения





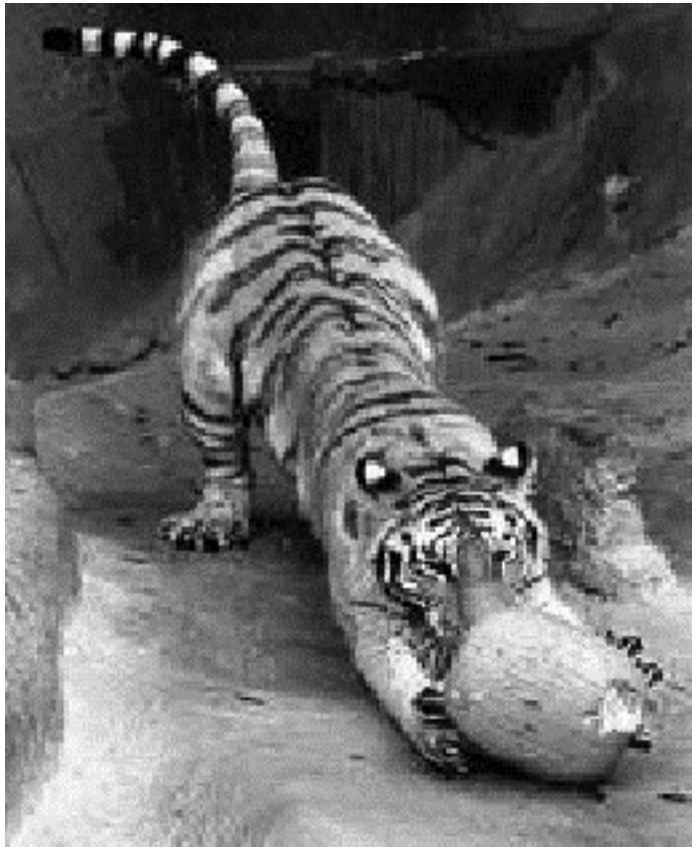
Пример использования МГК

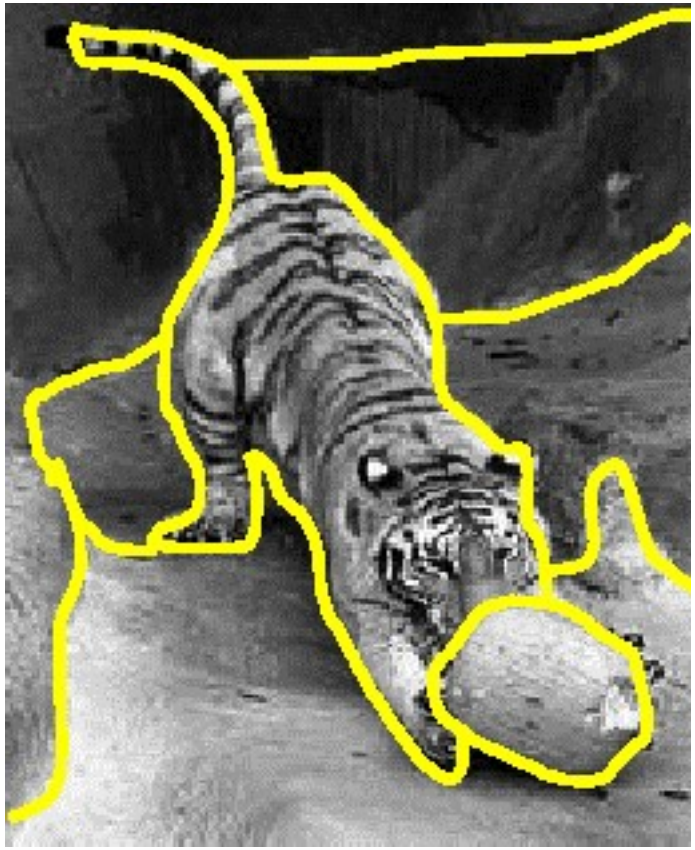
- Сегментация изображения



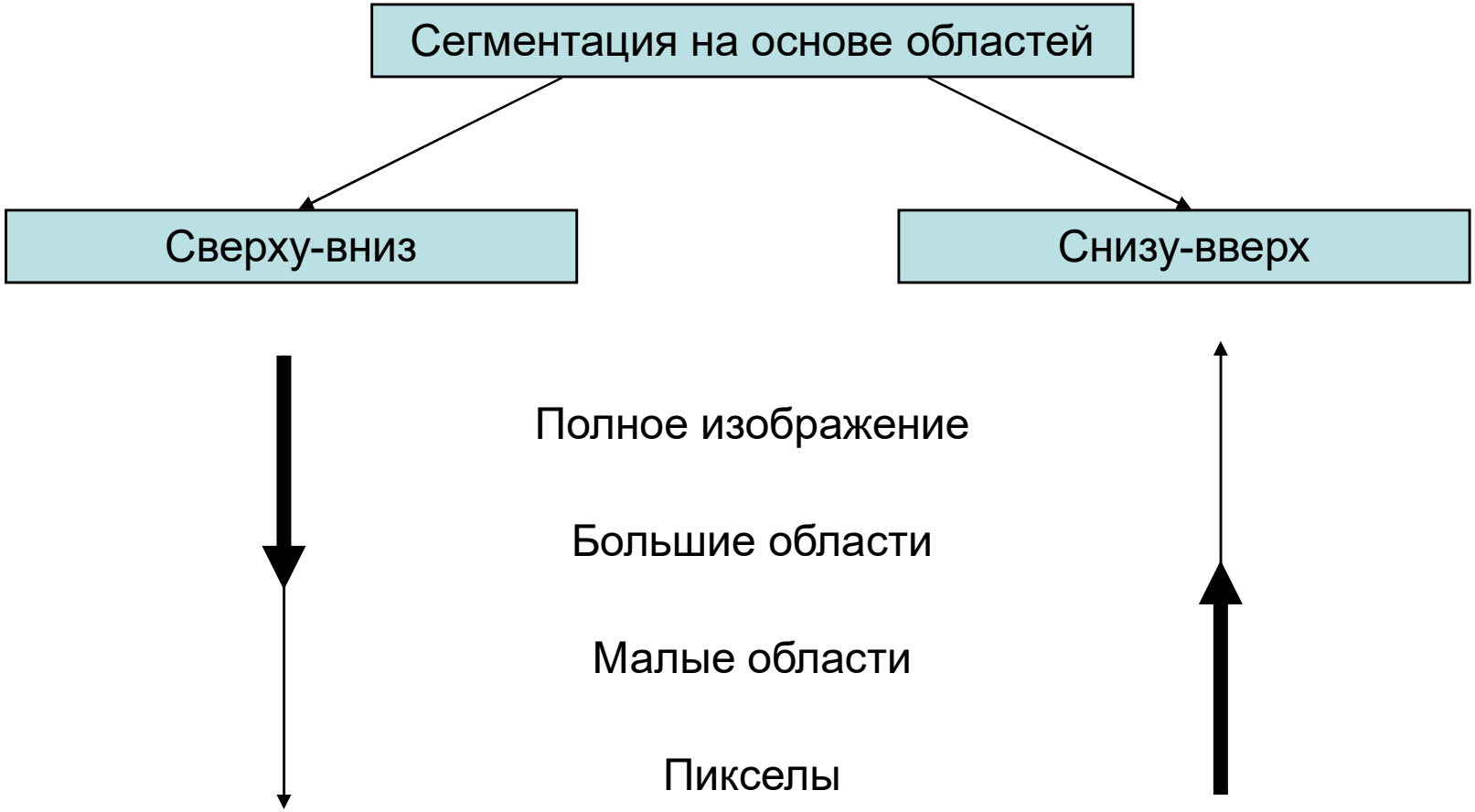
Сегментация наращиванием и/или декомпозицией областей

- Главный принцип: соседние пикселы, имеющие похожие статистические свойства, скорее всего принадлежат одной области.

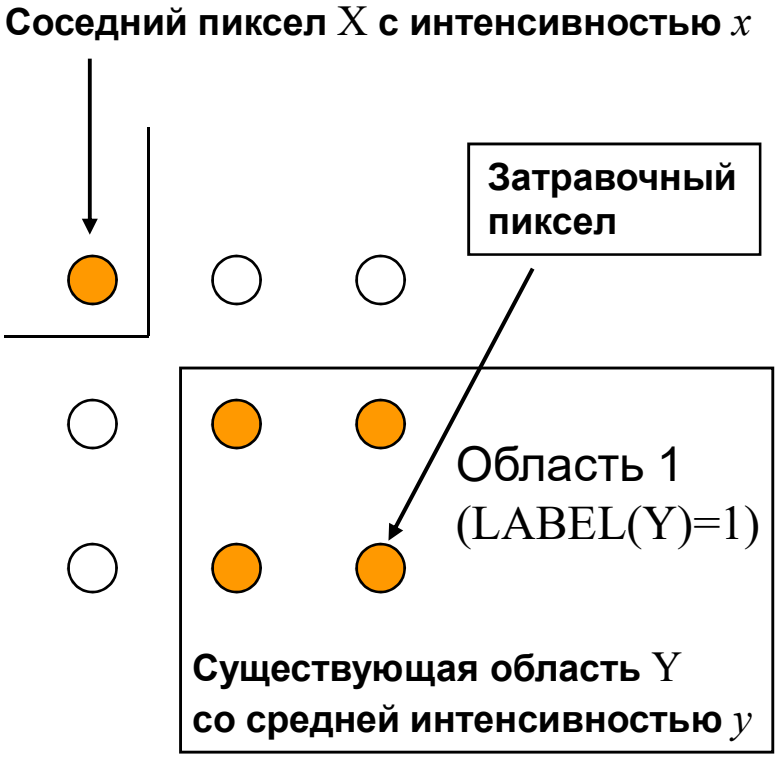




Сегментация на основе областей



Наращивание областей



1. Если $\mathrm{LABEL(X)} = 0$ и $|y - x| < T$, то присоединить пиксел X к области Y: $\mathrm{LABEL(X)} = \mathrm{LABEL(Y)}$.
2. Пересчитать среднее значение интенсивности новой области.

T – пороговое значение

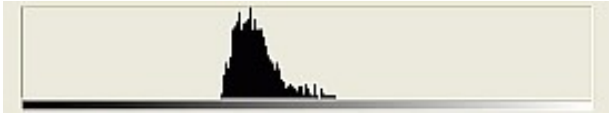
$\mathrm{LABEL(X)}$ – метка области

Простые критерии однородности

- Отклонение от среднего менее порога

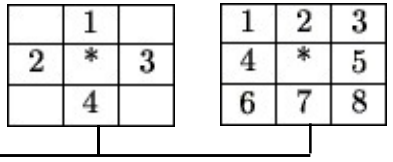
$$\forall p \in S \quad \left| I(p) - \frac{1}{N} \sum_{q \in S} I(q) \right| < T_{avg}$$

- Гистограмма содержит не более одного значительного пика



- Разница между соседями менее порога

$$\forall p \in S, \forall q \in N(p) \quad \left| I(p) - I(q) \right| < T_{diff}$$



Алгоритм наращивания областей Харалика

- Предполагает, что области есть множества связных пикселов с одинаковыми средними и дисперсиями.
- Пусть R – область, содержащая N пикселов со средней интенсивностью y.
- Пусть $\overline{X}$ и S^2 – среднее и дисперсия в области R.
- Вычисляется статистика

$$T = \left[\frac{(N-1)N}{N+1} (y - \overline{X})^2 \Big/ S^2 \right]^{\frac{1}{2}}$$

Алгоритм наращивания областей Харалика

- Если T «мало», то y включается в область R, значение среднего и дисперсии вычисляются снова.

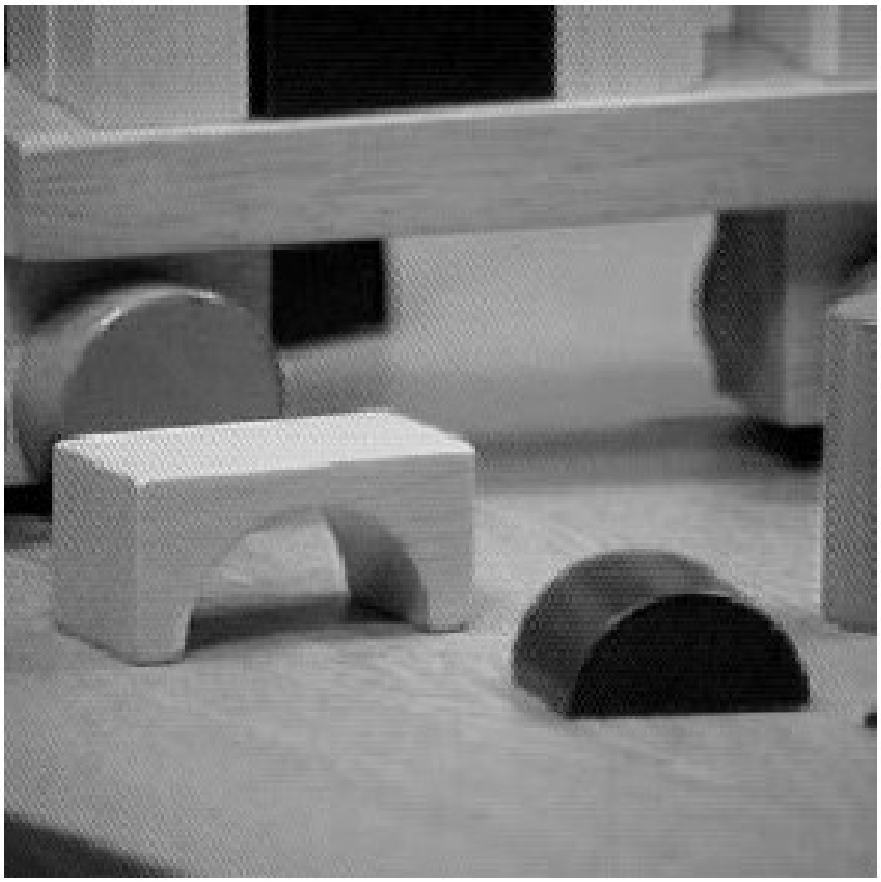
$$\overline{X}_{new} \leftarrow (N\overline{X}_{old} + y)/(N+1)$$

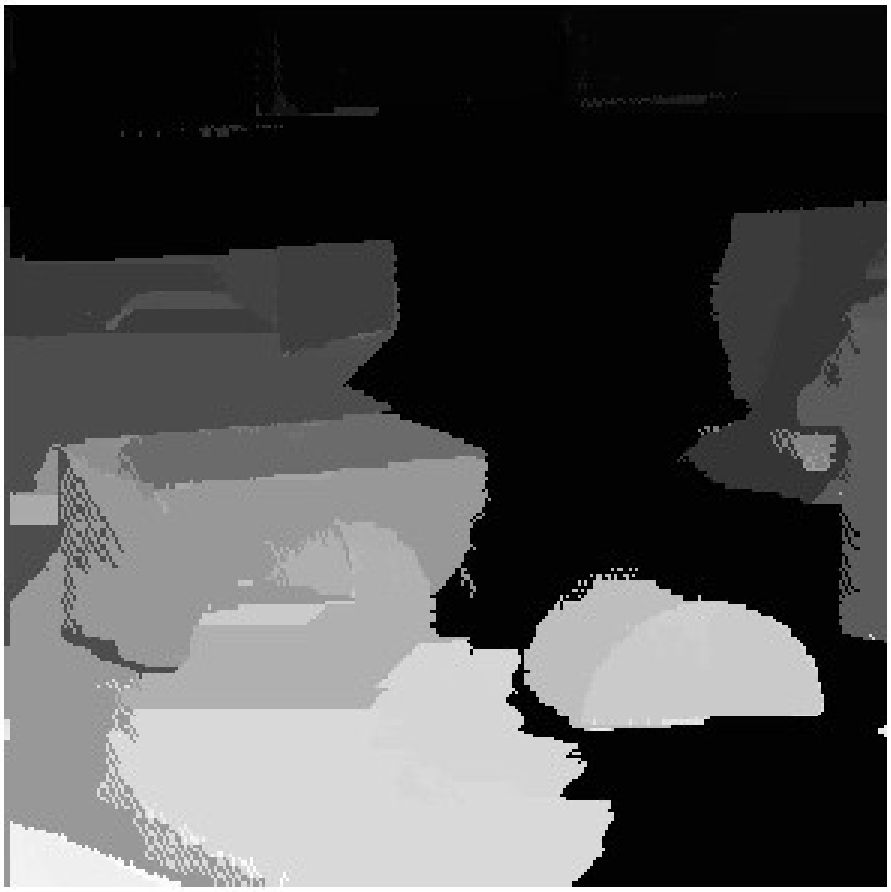
$$S^2_{new} \leftarrow S^2_{old} + (y - \overline{X}_{new})^2 + N(\overline{X}_{new} - \overline{X}_{old})^2.$$

- Если T «велико», то, вероятно, y не принадлежит R.

Алгоритм наращивания областей Харалика

- Чтобы придать точный смысл понятию «велико» можно использовать статистические тесты значимости различия на уровне α.
- Значение α определяет вероятность того, что статистика T с $N-1$ степенями свободы превзойдет $t_{N\text{-}1}(\alpha)$.
- Если наблюдаемое значение T превосходит $t_{N\text{-}1}(\alpha)$, то различия считаются значимыми.

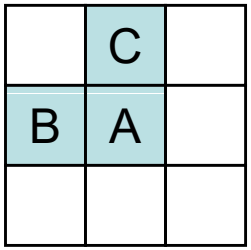




Если пиксел y и область действительно принадлежат к разным генеральным совокупностям, вероятность того, что тест даст неправильный ответ равна α. α -- параметр, задаваемый пользователем. $t_{N\text{-}1}(\alpha)$ дает большие значения для малого количества степеней свободы и наоборот.

Алгоритм наращивания областей

Сканируем изображение сверху вниз, слева направо:



1. if $|I(A) - I_{avg}(B)| > \delta$ and $|I(A) - I_{avg}(C)| > \delta$ -
   создаем новую область, присоединяем к ней пиксел A

2. if $|I(A) - I_{avg}(B)| \le \delta$ xor $|I(A) - I_{avg}(C)| \le \delta$ –
   добавить A к одной из областей

3. if $|I(A) - I_{avg}(B)| \le \delta$ and $|I(A) - I_{avg}(C)| \le \delta$:

   1. $|I_{avg}(B) - I_{avg}(C)| \le \delta$ –
      сливаем области B и C.

   2. $|I_{avg}(B) - I_{avg}(C)| > \delta$–
      добавляем пиксел A к тому классу, отклонение от которого минимально.

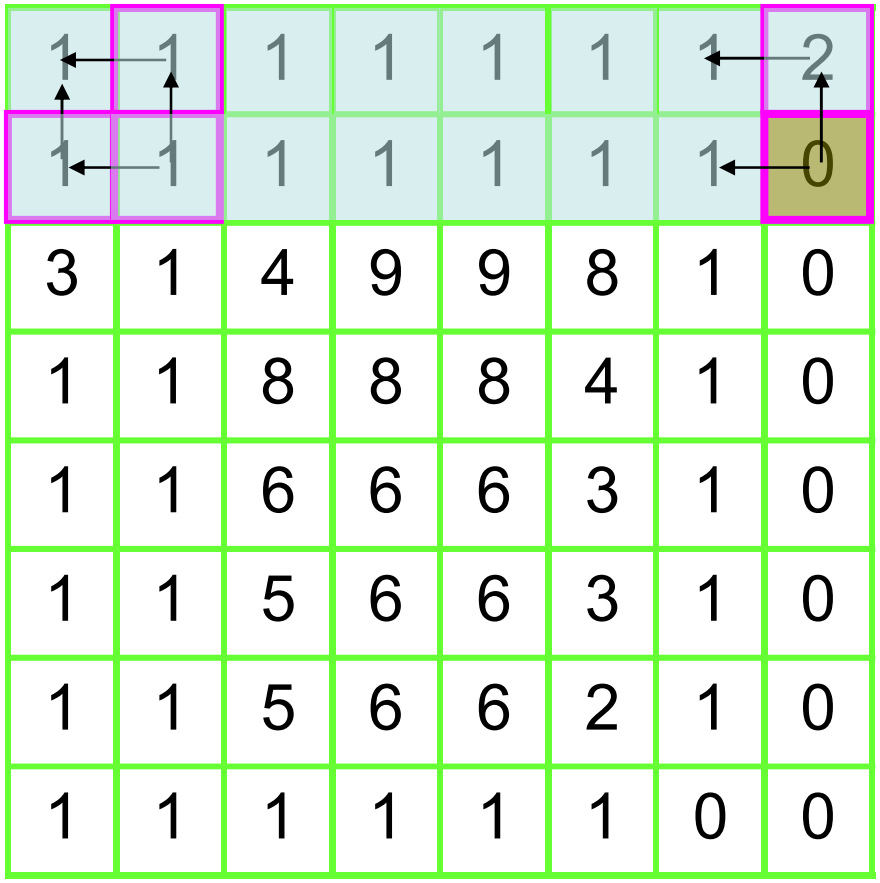
$I(A)$ – яркость пиксела A

$I_{avg}(B)$ – средняя яркость области к которой принадлежит B

Алгоритм наращивания областей

Среднее: 1

Среднее: 1.125

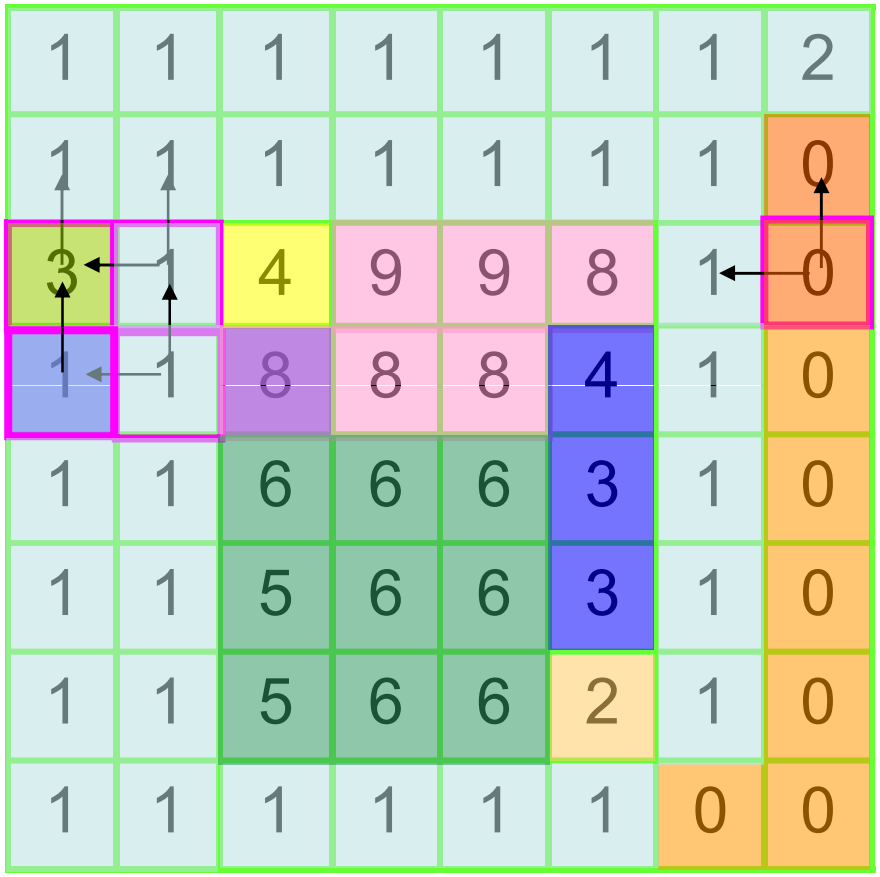


$$\forall p \in S \quad \left| I(p) - \frac{1}{N} \sum_{q \in S} I(q) \right| < \delta$$

Пример δ = 1

Алгоритм наращивания областей

$$\forall p \in S \quad \left| I(p) - \frac{1}{N} \sum_{q \in S} I(q) \right| < \delta$$



$$\forall p \in S \quad \left| I(p) - \frac{1}{N} \sum_{q \in S} I(q) \right| < \delta$$

Пример δ = 1

Слияние областей

Наращивание областей $\Rightarrow$ большое количество мелких областей R_i

Слияние областей

Для каждой из областей R_i вычисляются среднее и стандартное отклонение

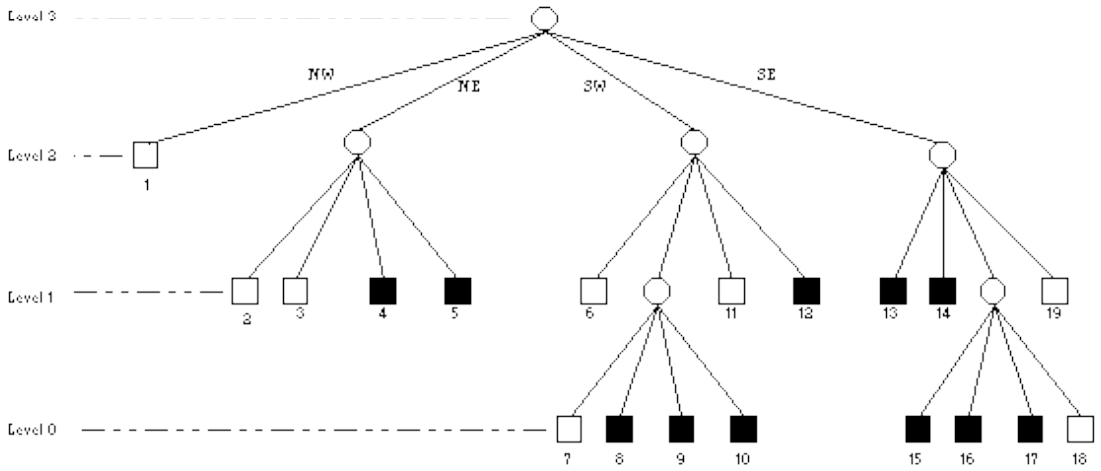
$$m_i = \frac{1}{n} \sum_{(x,y) \in R_i} I(x,y) \qquad s_i = \sqrt{\frac{1}{n} \sum_{(x,y) \in R_i} (I(x,y) - m_i)^2}$$

Если $|m_i - m_j| < ks_i, \ i = 1,2,...$ то области R_i, R_j сливаются.

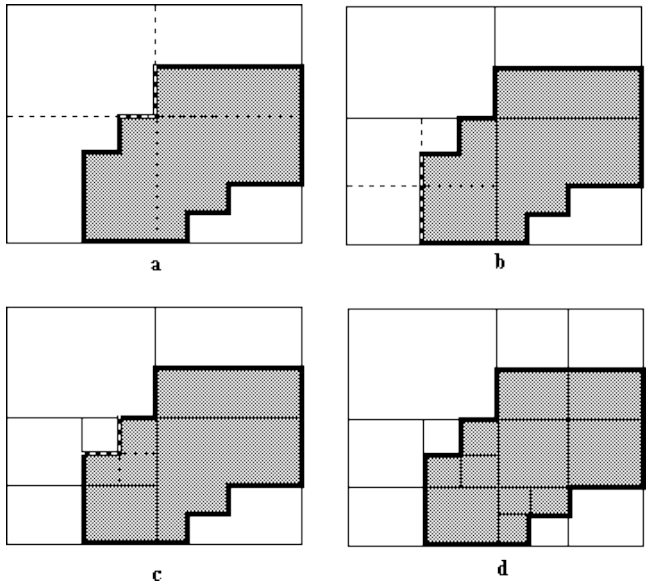
Разбиение областей

1. Если изображение неоднородно, оно разбивается на четыре квадранта.
2. Для каждого из квадрантов алгоритм рекурсивно повторяется.

Квадродерево



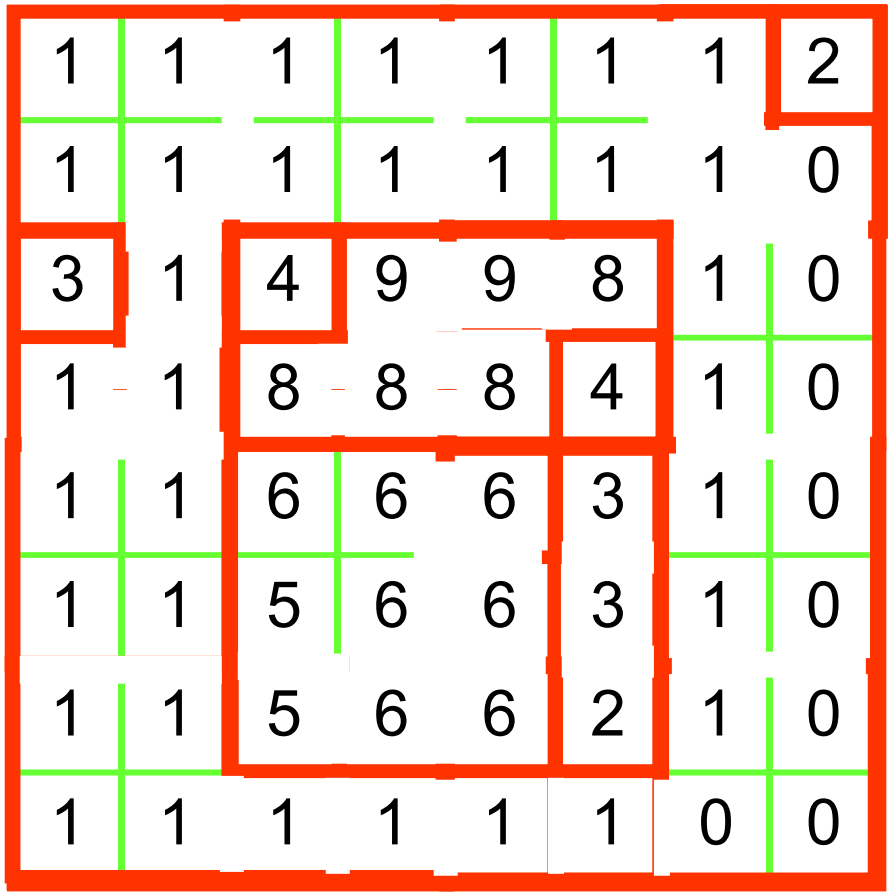
Изображение



Алгоритм разбиения/слияния (split and merge)

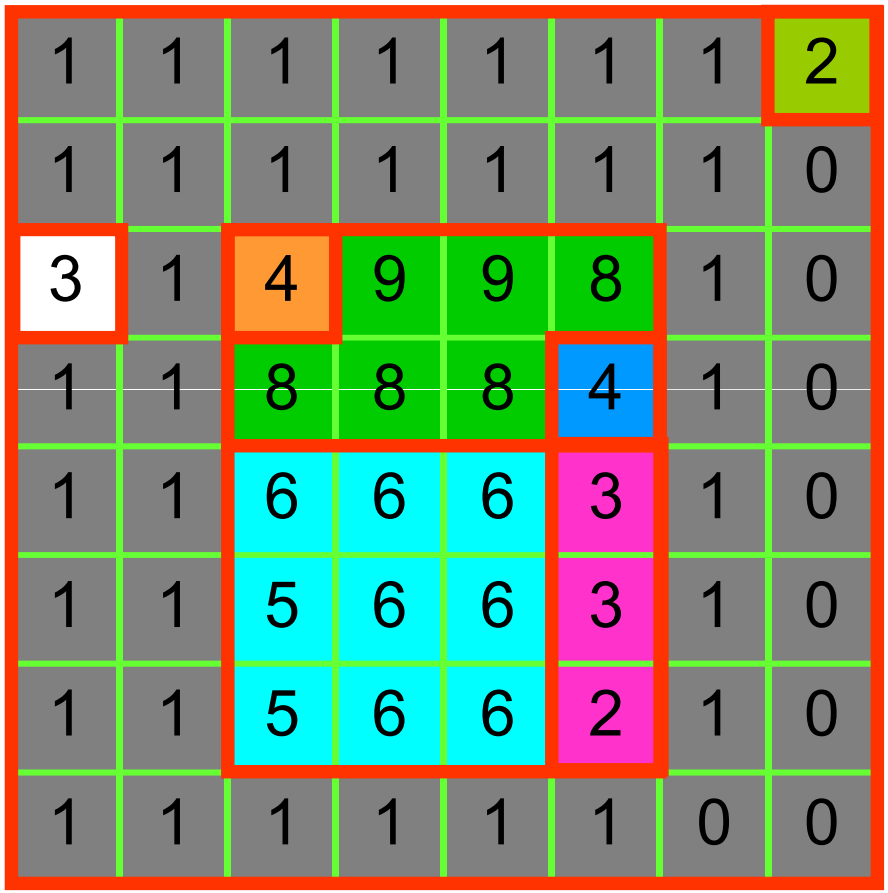
- Идея:
  - Сначала провести разбиение на небольшие однородные области
    - Обычно используется принцип квадродерева
  - Затем слить между собой те из них, которые вместе не нарушат требование однородности
    - Продолжать до тех пор, пока остаются регионы которые можно объединить

Алгоритм разбиения/слияния (split and merge)



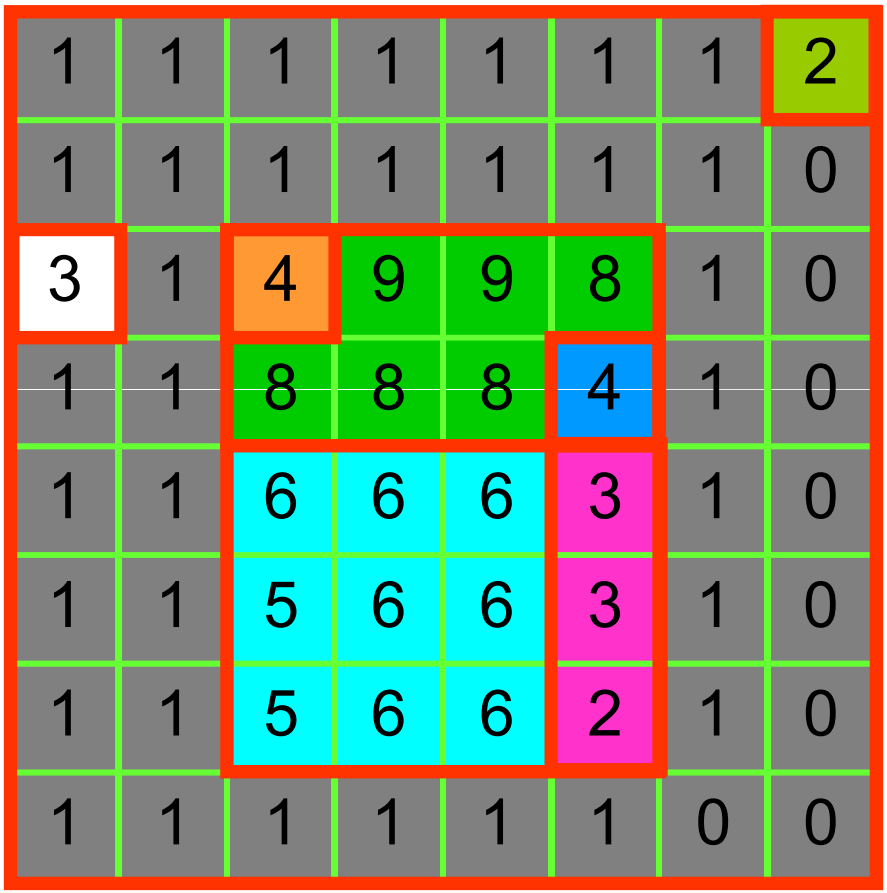
Слияние

Алгоритм разбиения/слияния (split and merge)



Результат

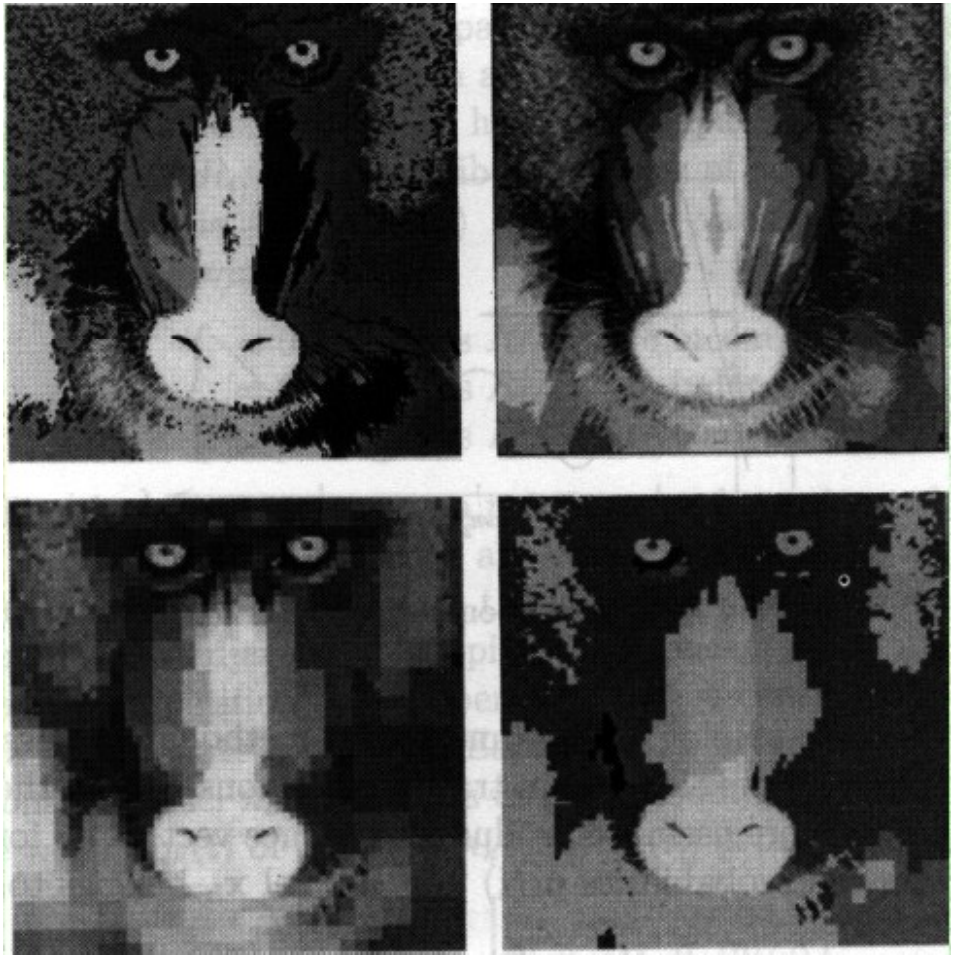
Сравним с разрастанием регионов



Результат

Сегментация, основанная на областях. Примеры

Расширение областей



Слияние областей

Разбиение областей

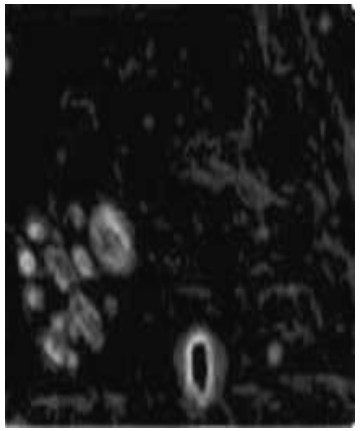
Разбиение и слияние областей

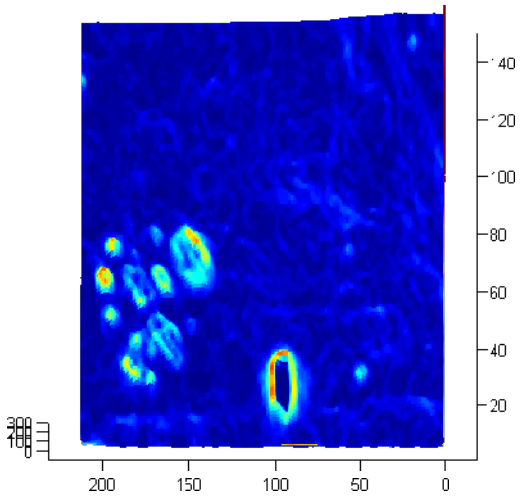
Разбиение и слияние

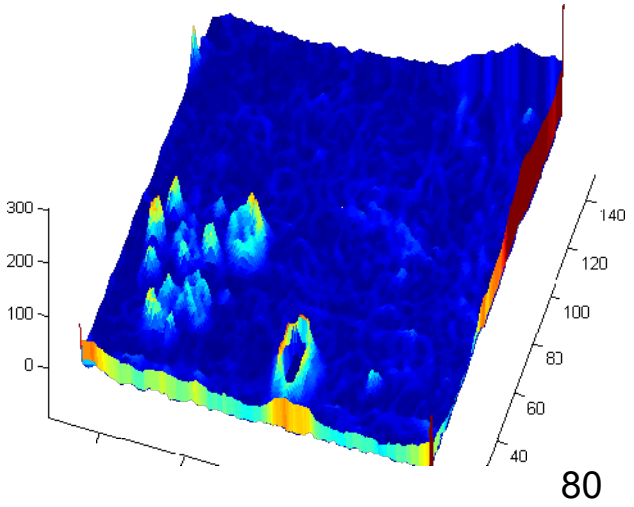
Разбиение и слияние

Алгоритм водораздела (watershed)

- Идея метода:
  - Большие значения градиента соответствуют резким переходам на изображении
  - Рассмотрим абсолютную величину градиента как карту высот ландшафта
  - Там, где резкие границы – получатся «стены»
  - Будем «лить воду» в «ямы» и искать получающиеся «озера»

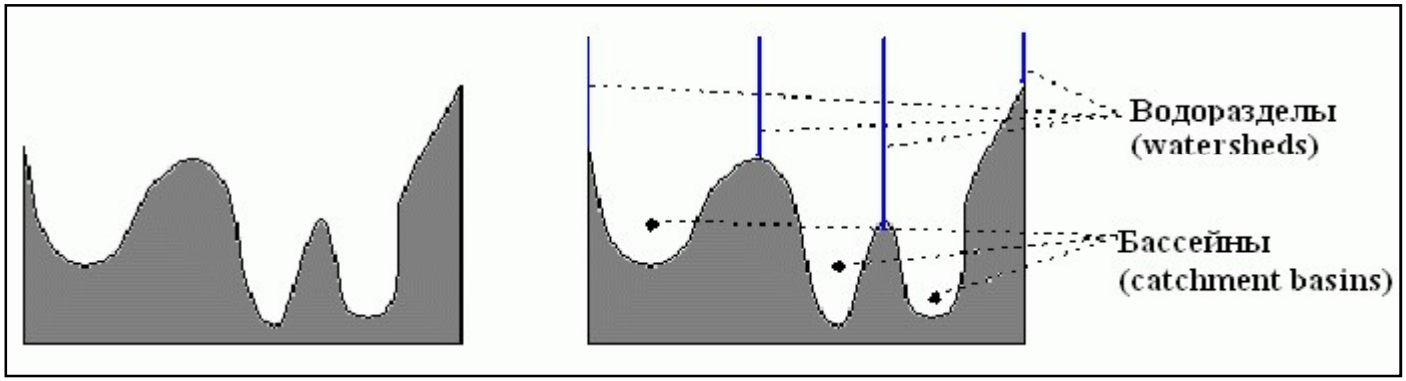






Алгоритм водораздела

Область водораздела, бассейн (*catchment basin*): область в которой поток из любой ее точки «стекает» к одной общей точке

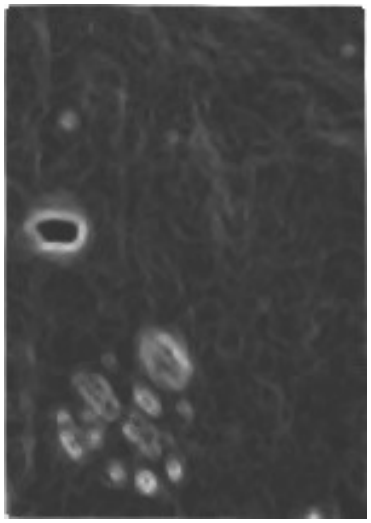


Слева – профиль интенсивностей изображения, справа – локальные минимумы определяют бассейны, локальные максимумы – линии водораздела.

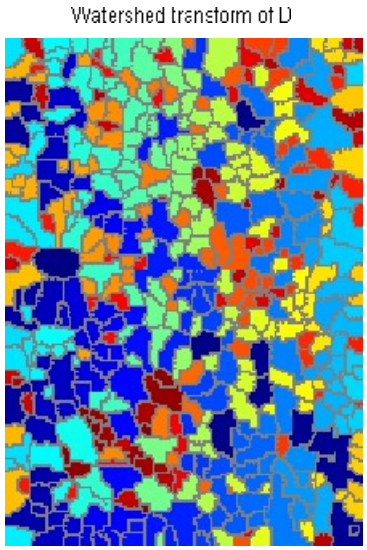
Алгоритм водораздела

– Алгоритм, как и разбиение дает множество небольших регионов

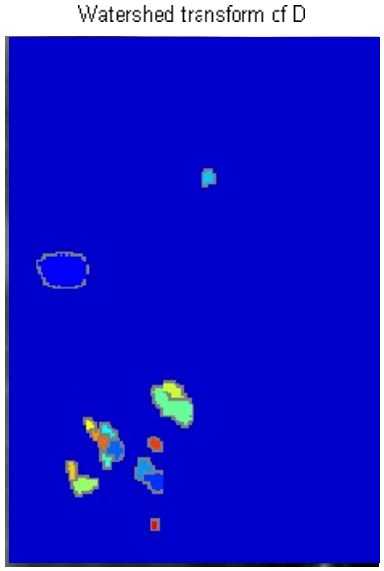
  • Очень чувствителен к шуму – ищет все локальные минимумы



Абсолютная. величина градиента



Результат по данному градиенту



Градиент < 10 обращен в 0

Алгоритм «погружения»

Алгоритм «погружения» (immersion) :

Начнем с самых «глубоких» (темных) пикселей (они определят начальные бассейны)

Для каждой яркости *k*:

Для каждой компоненты пикселей яркости *k*:

Если прилежит только к одному существующему бассейну

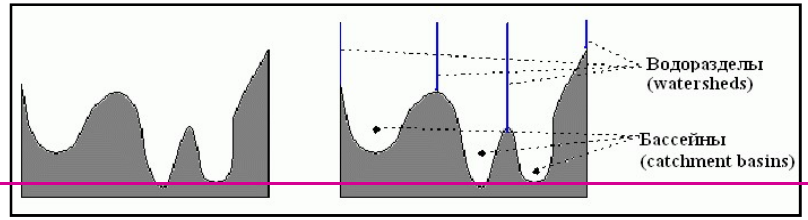
Добавить компоненту к бассейну

Если прилежит более чем к одному существующему бассейну

Пометить как границу (водораздел)

Иначе – создать новый бассейн

Аналог – вода медленно поднимается, пока не погрузятся в нее водоразделы

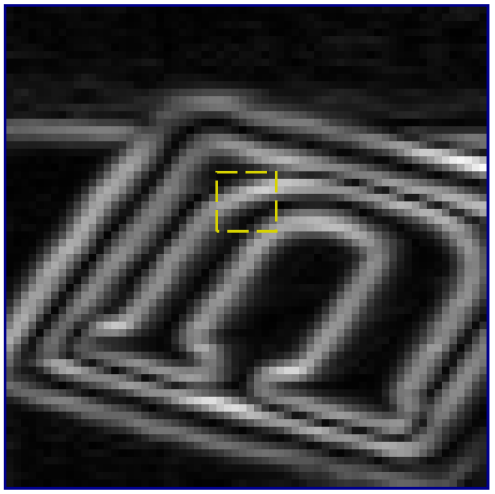


Алгоритм tobogganing

- Идея:
  - Из каждого пикселя «спускаемся» в локальный минимум среди его соседей
  - Спускаемся до тех пор, пока есть куда спускаться
  - Пиксели «спустившиеся» в один минимум – одна область

Как с горы на санках

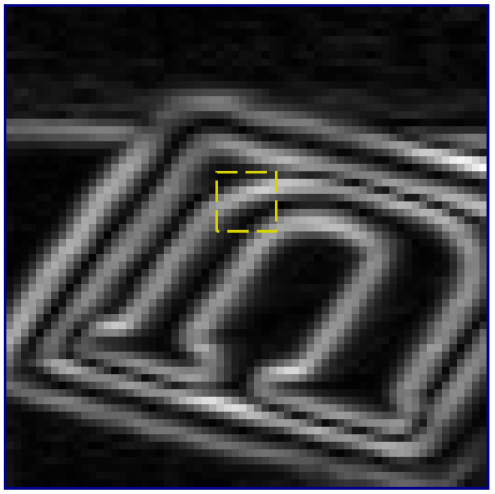
Алгоритм tobogganing



- Из каждого пикселя «спускаемся» в локальный минимум среди его соседей
- Спускаемся до тех пор, пока есть куда спускаться
- Пиксели «спустившиеся» в один минимум – одна область

58	46	50	64	80	88	99	108
80	63	68	106	137	164	185	202
55	113	152	179	202	217	225	227
147	180	199	208	209	202	191	177
192	204	202	190	169	145	122	96
194	186	167	140	109	83	56	63
177	154	124	91	54	41	95	136
159	131	104	81	56	94	142	178

Алгоритм tobogganing



- Из каждого пикселя «спускаемся» в локальный минимум среди его соседей
- Спускаемся до тех пор, пока есть куда спускаться
- Пиксели «спустившиеся» в один минимум – одна область