Методы и средства обработки изображений

Яночкин Алексей Леонидович

Ассистент каф. ЭВМ

Сегментация изображений

Лекция 3

Many slides adapted from Fei-Fei Li, Rob Fergus, Antonio Torralba, Jean Ponce and Svetlana Lazebnikб Anton Konushin

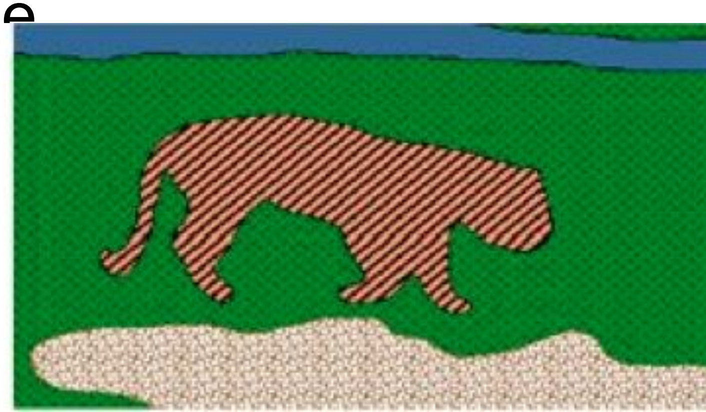
Из чего состоит изображение?

Из «кусков» - отдельных объектов

Сегментация

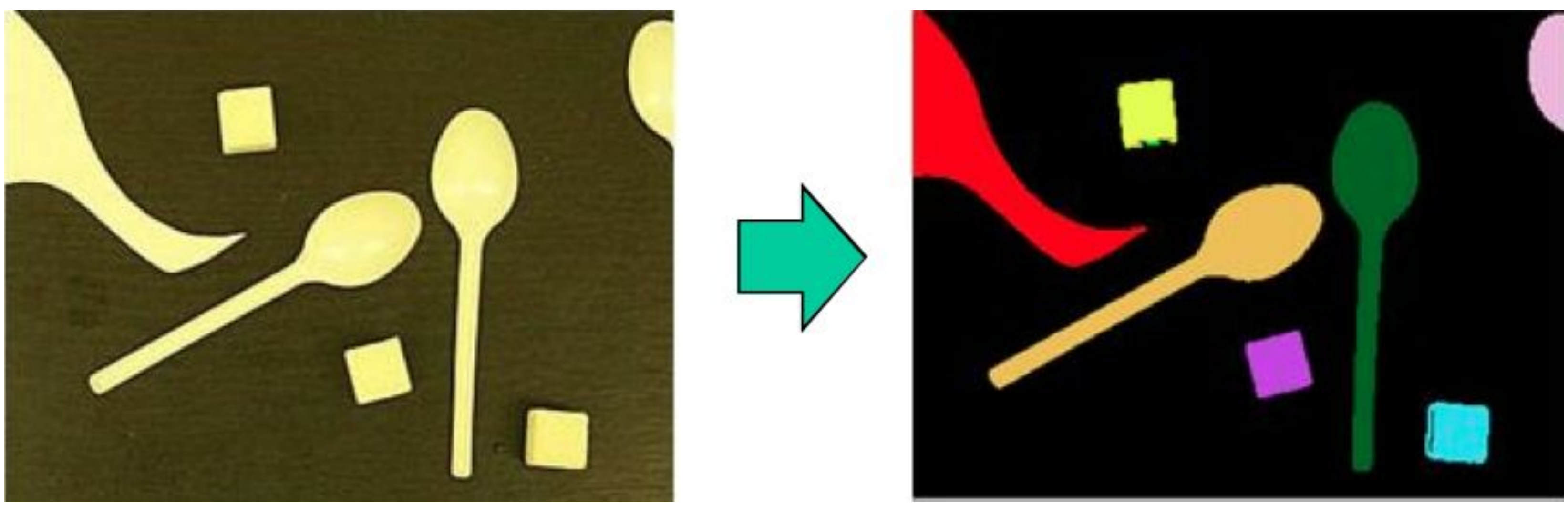
- Сегментация - это способ разделения сцены на «куски», с которыми проще работать
- Тесселяция - разбиение изображения на неперекрывающиеся области, покрывающие все изображение и однородные по некоторым признакам
- Можно и по другому сегментировать изображение
  – Пересекающиеся области
  – Иерархическое представление





Результат сегментации

- Как мы будем записывать результат сегментации?
- Сделаем карту разметки – изображение, в каждом пикселе которого номер сегмента, которому принадлежит этот пиксель
- Визуализировать удобно каждый сегмент своим цветом

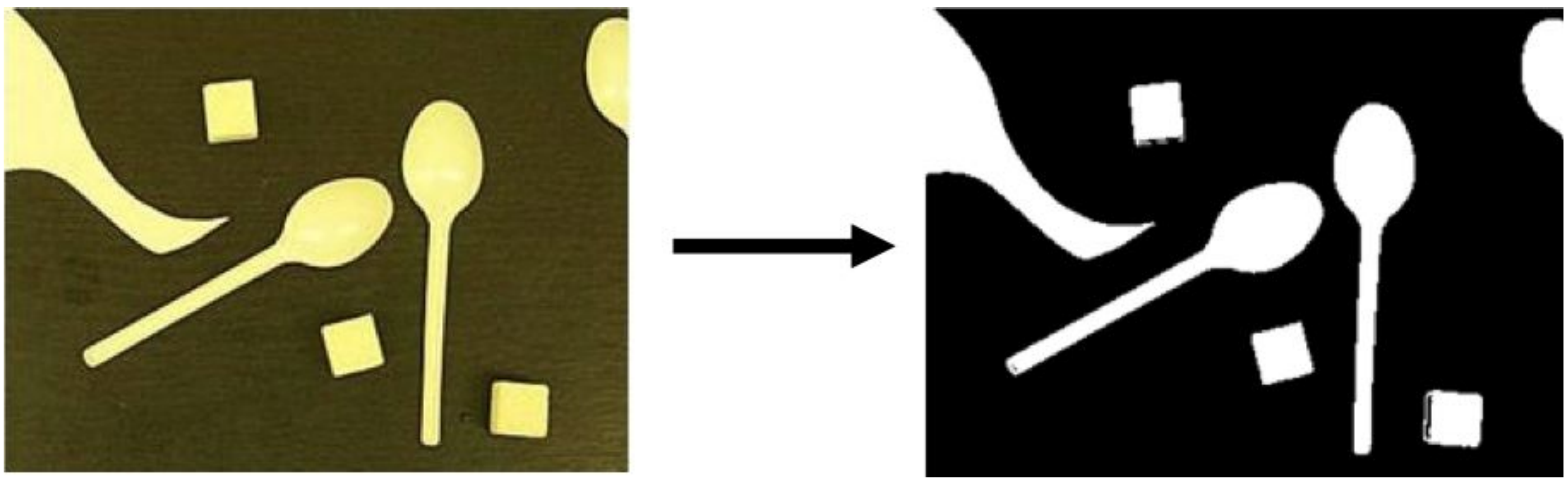


Простейшая сегментация

- Чем отличаются объекты на этом изображении?

- Все объекты яркие, фон тёмный
- Для сегментации такого изображения нам достаточно:
  - пороговая бинаризация
  - обработки шума
  - выделения связанных компонент

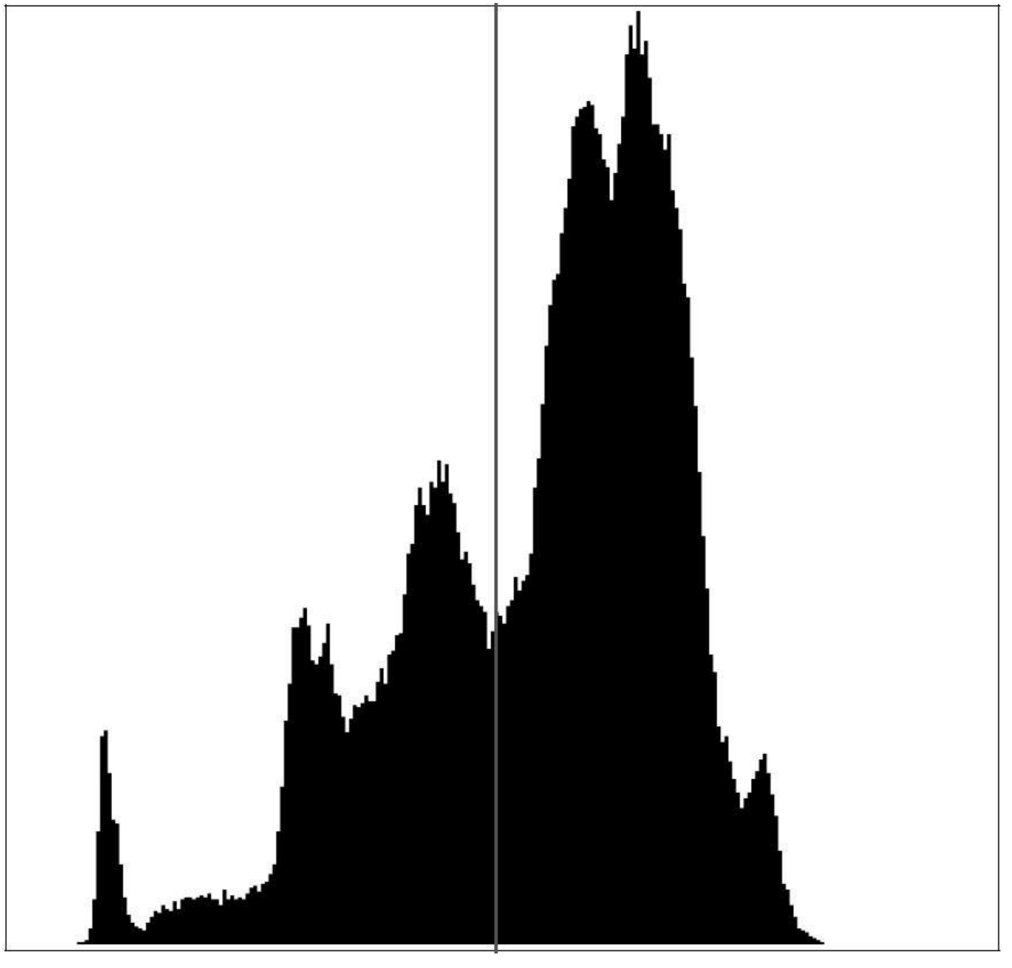
Пороговая бинаризация



Пороговая бинаризация

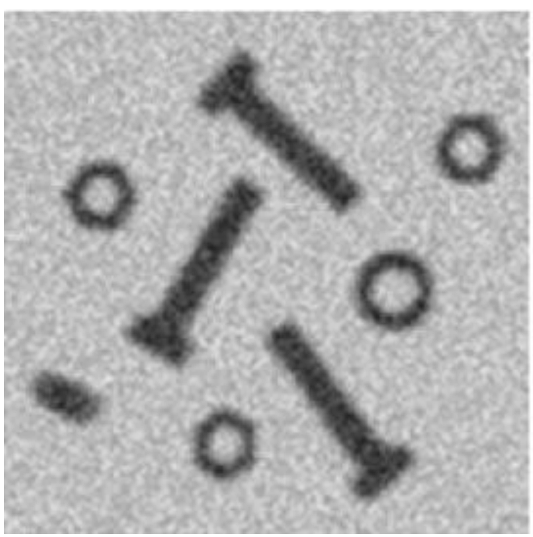
- Пороговая фильтрация (thresholding)
  - Пиксели, которых выше/ниже некоторого порога, заданного «извне», помечаются 1
  - Ниже порога помечаются 0
- Бинарное изображение – пиксели которого могут принимать только значения 0 и 1
- Бинаризация - построение бинарного изображения по полутоновому / цветному

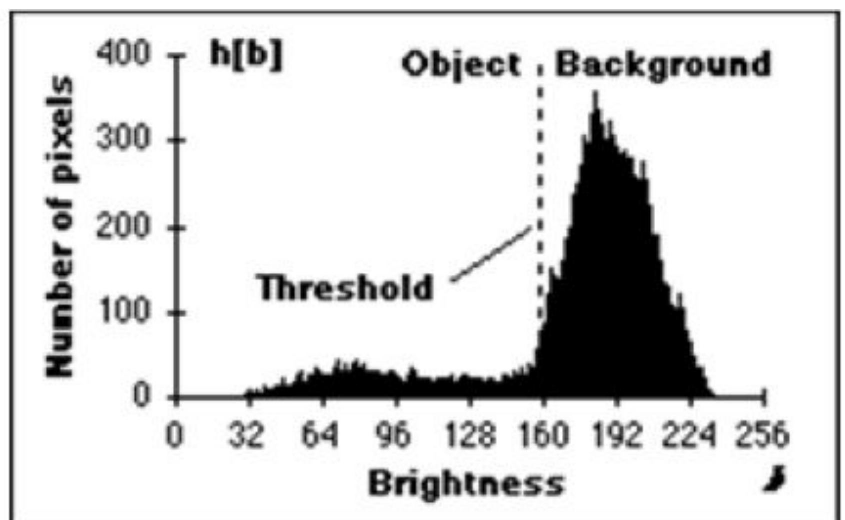
Пороговая бинаризация



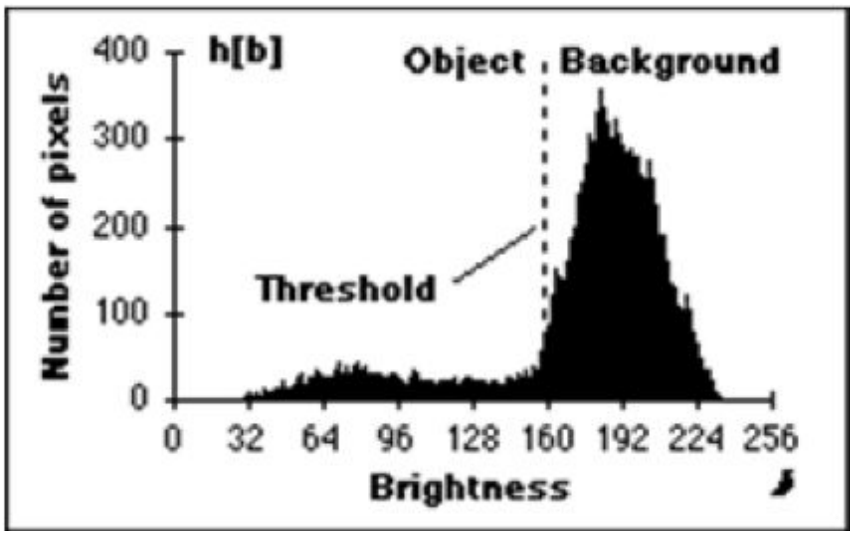
Пороговая фильтрация

- Более интересный способ – определение порога автоматически, по характеристикам изображения
- Анализ гистограммы





Анализ гистограммы

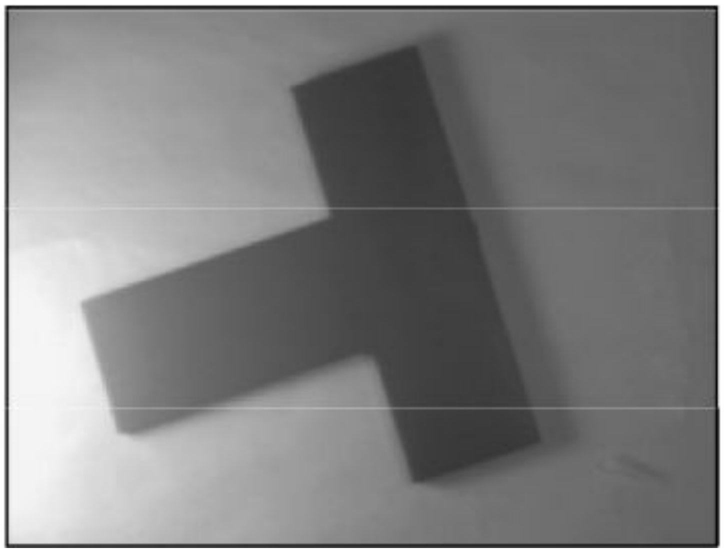


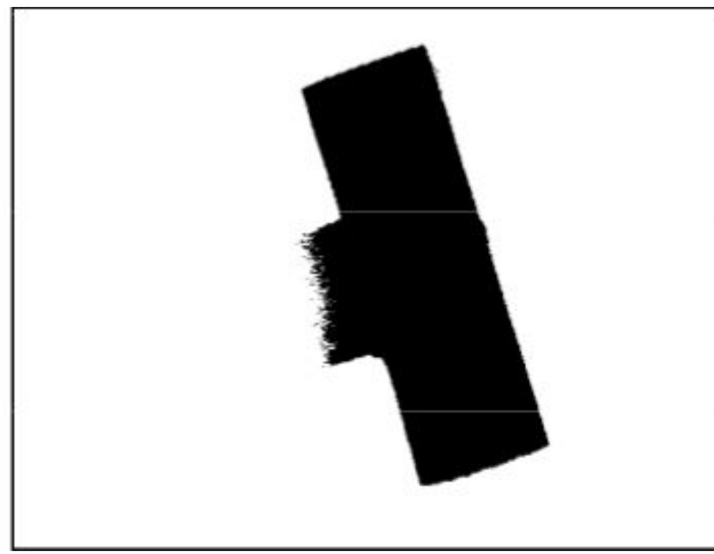
- Анализ симметричного пика гистограммы
- Применяется когда фон изображения дает отчетливый и доминирующий пик гистограммы, симметричный относительно своего центра.

Анализ гистограммы

1. Сгладить гистограмму;
2. Найти ячейку гистограммы h_{max} с максимальным значением;
3. На стороне гистограммы не относящееся к объекту (на примере – справа от пика фона) найти яркость h_p, количество пикселей с яркостью >= h_p равняется *p*% (например 5%) от пикселей яркости которых >= h_{max};
4. Рассчитать порог $T = h_{max} - (h_p - h_{max})$;

Адаптивная бинаризация





Адаптивная бинаризация

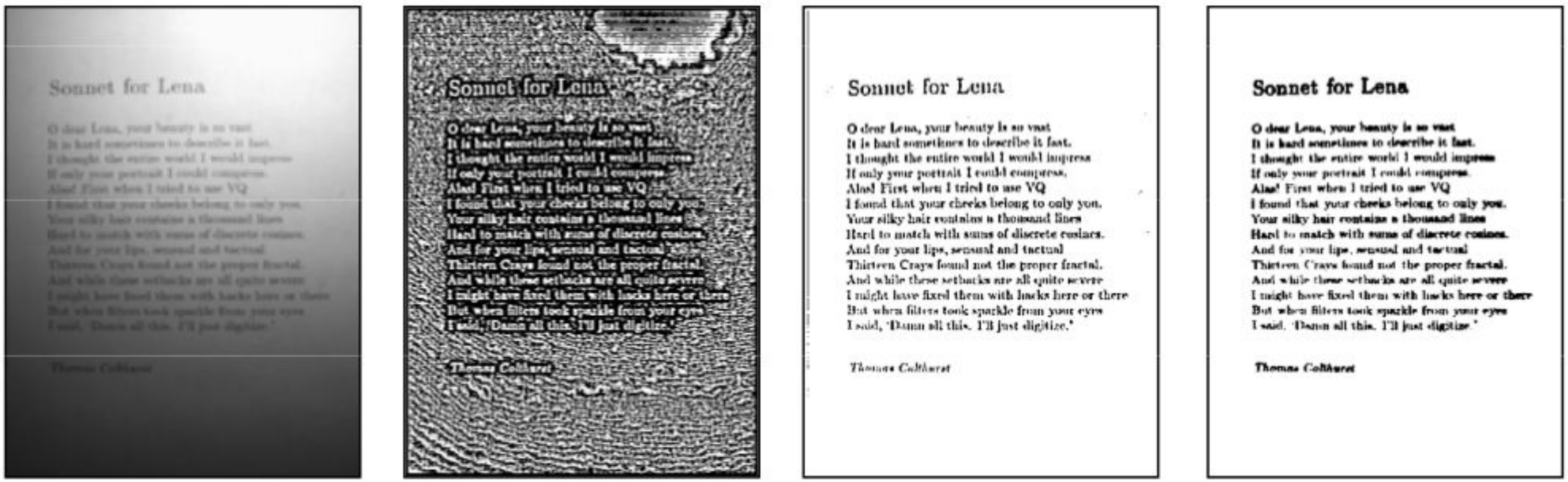
Необходима в случае неравномерной яркости фона/объекта.

- Для каждого пикселя изображения $I(x, y)$:
  1. В окрестности пикселя радиуса r высчитывается индивидуальный для данного пикселя порог T;
  2. Если $I(x, y) > T + C$, результат 1, иначе 0;

Варианты выбора T:

- $T = mean$
- $T = median$
- $T = (min + max) / 2$

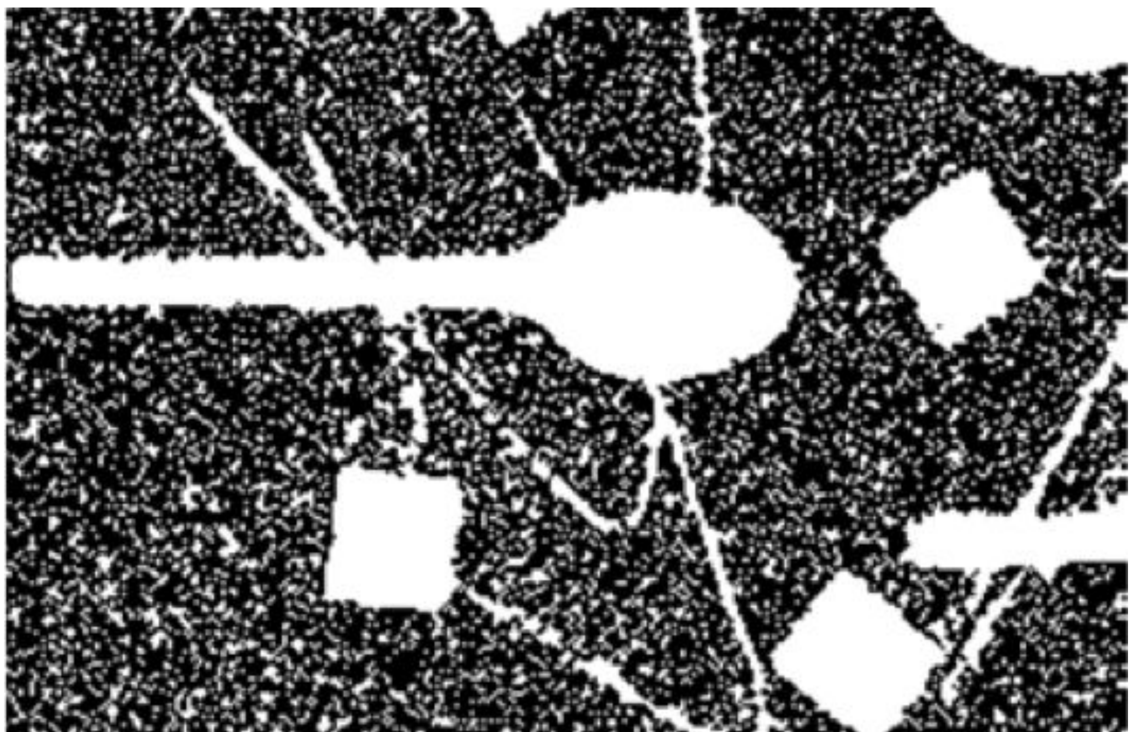
Адаптивная бинаризация



Исходное | r=7, C=0 | r=7, C=7 | r=75, C=10

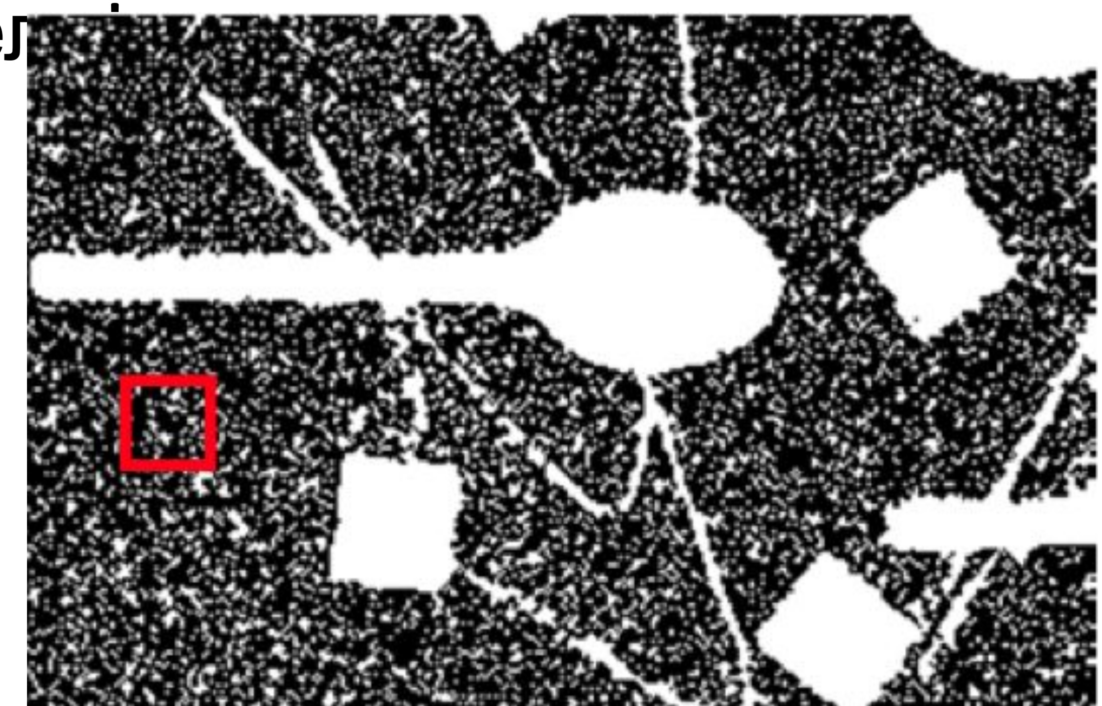
Шум в бинарных изображениях

- Часто возникает из-за невозможности полностью подавить шум в изображениях, недостаточной контрастности объектов и т. д.



Шум в бинарных изображениях

- По одному пикселю невозможно определить – шум или объект?
- Нужно рассматривать окрестность пиксел



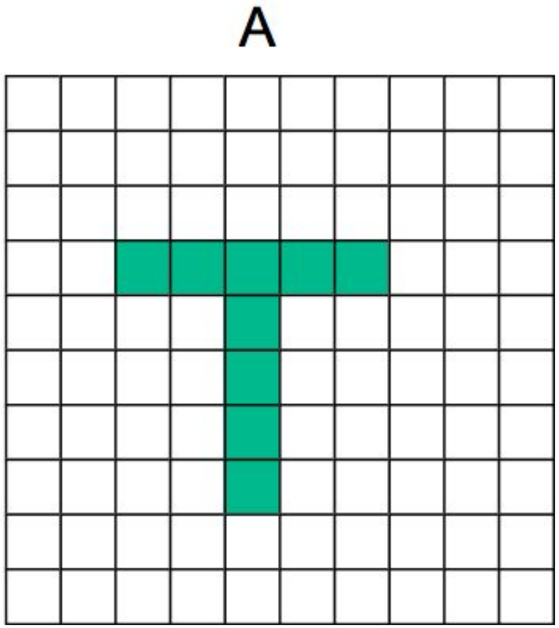
Подавление и устранение шума

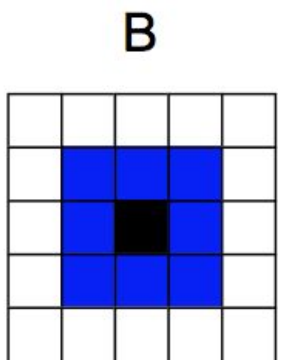
Широко известный способ - устранение шума с помощью операций математической морфологии:

- Сужение (erosion)
- Расширение (dilation)
- Закрытие (closing)
- Раскрытие (opening)

Математическая морфология

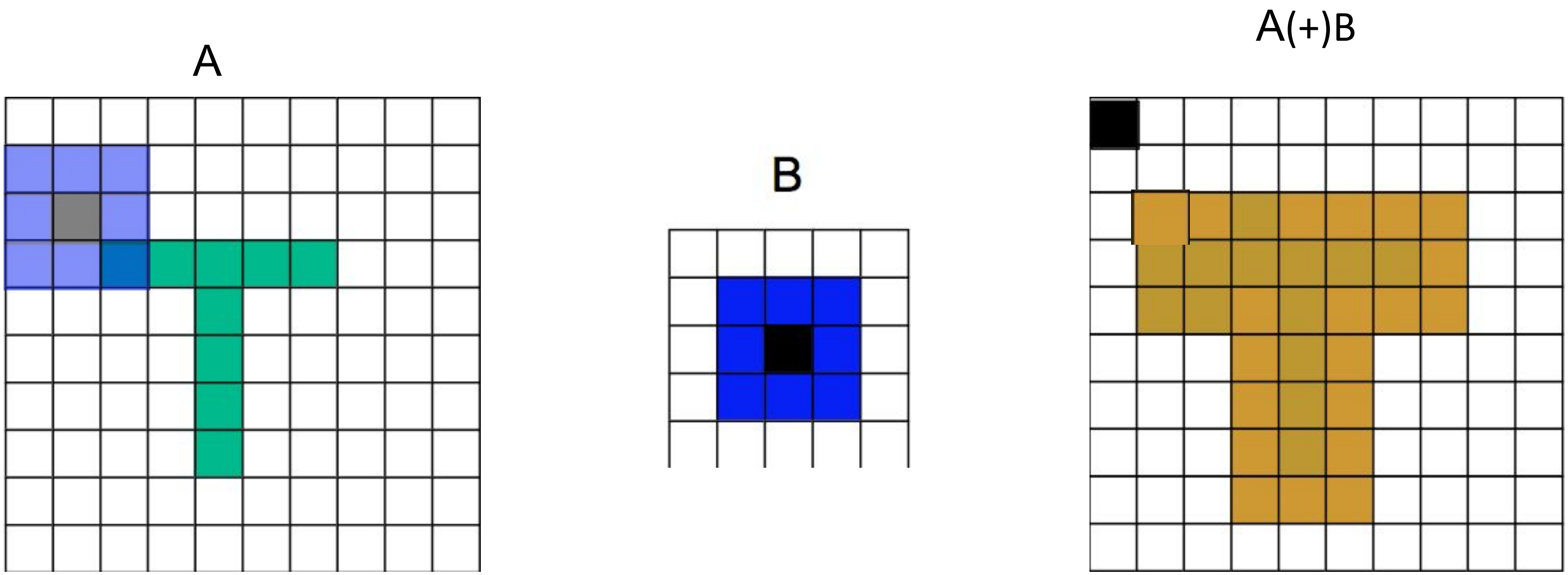
- Множество A обычно является объектом обработки
- Множество B (называемое структурным элементом) – инструмент обработки





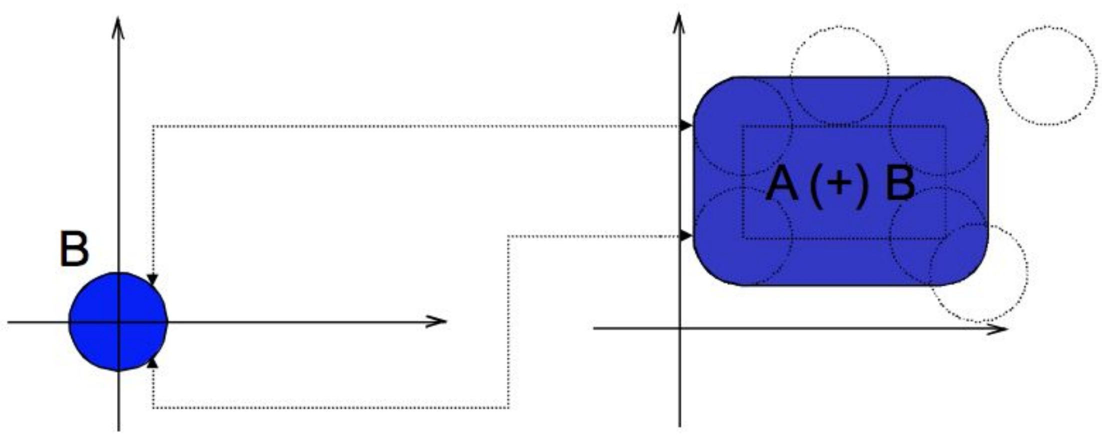
Операция «расширение»

- Операция «расширение» - аналог логического «или»



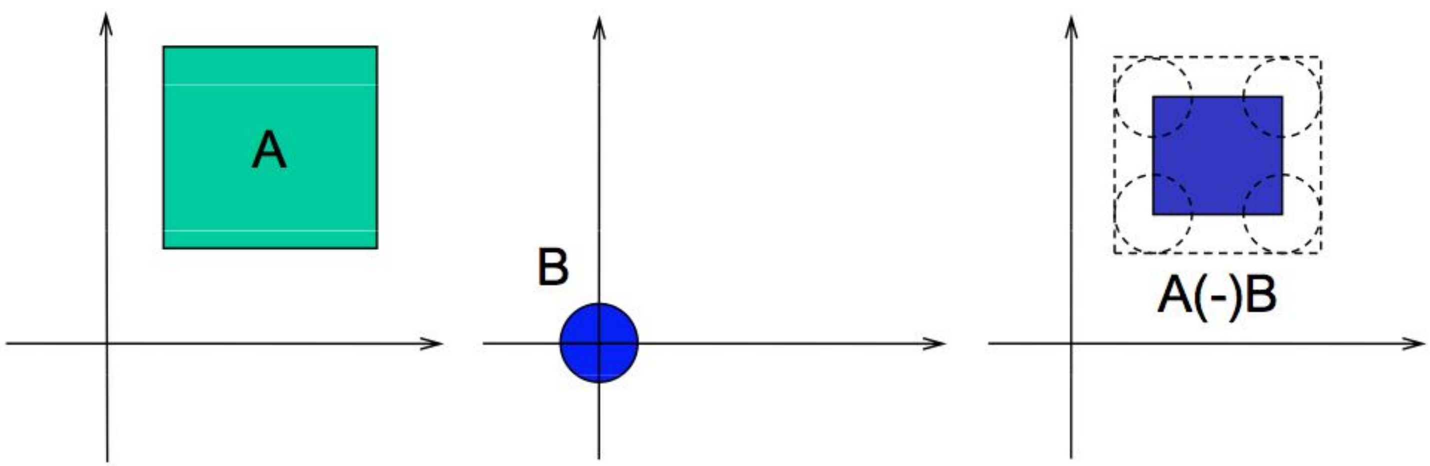
Операция «расширение»

- Расширение (dilation)
- *A (+) B = {t R2: t = a + b, a A, b B}*



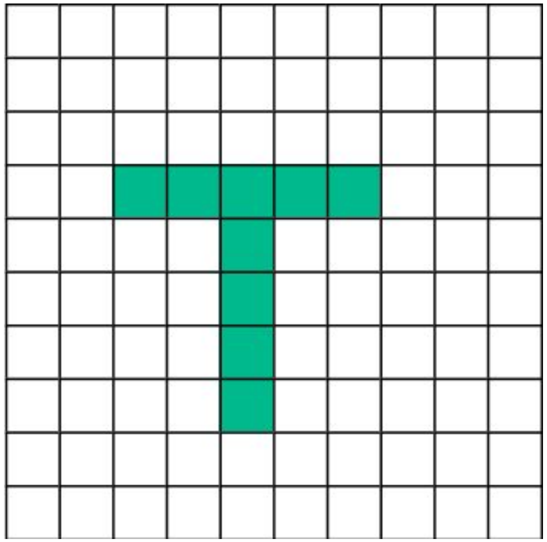
Операция «сужение»

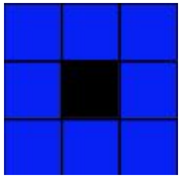
- Сужение (erosion)
- $A\ (\text{-})\ B = (A^C\ (+)\ B)^C$, где A^C -дополнение A

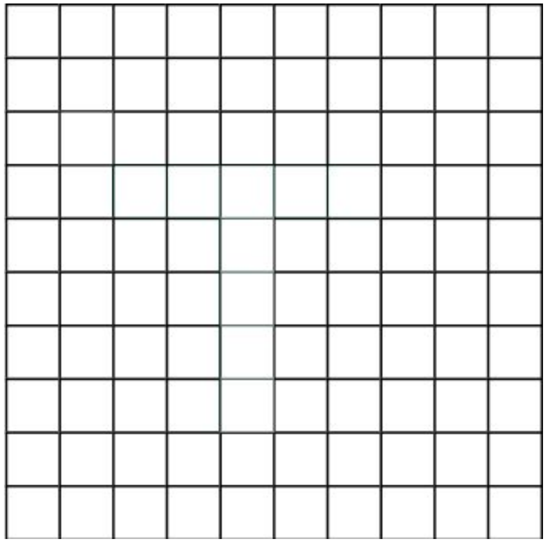


Операция «сужение»

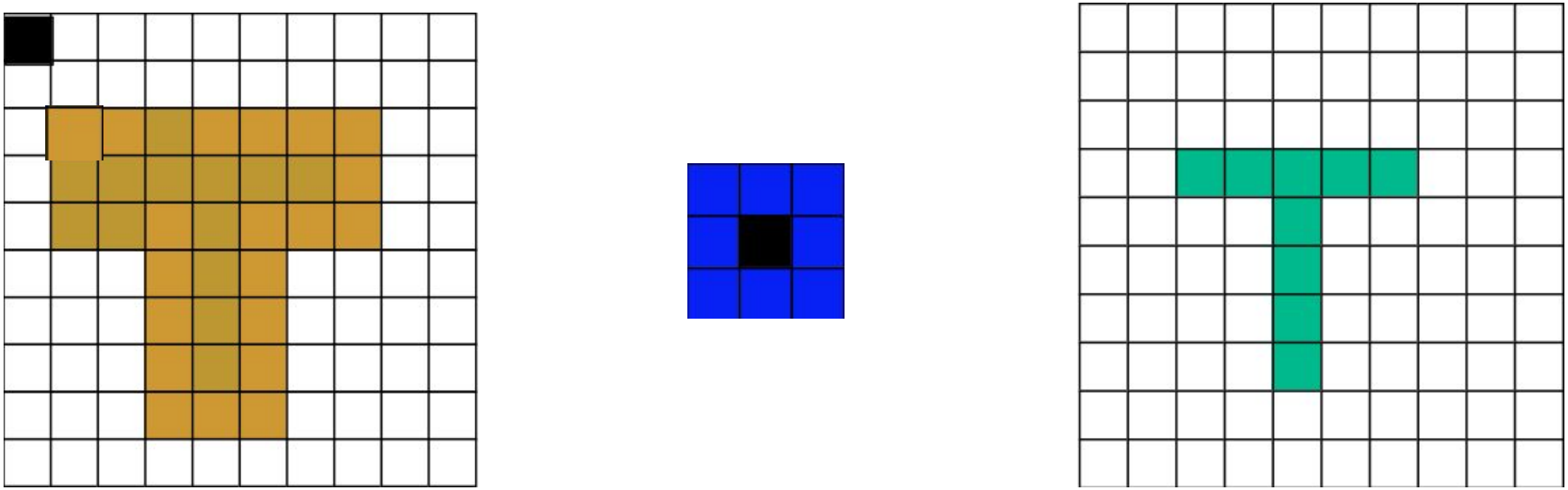
- Что будет?





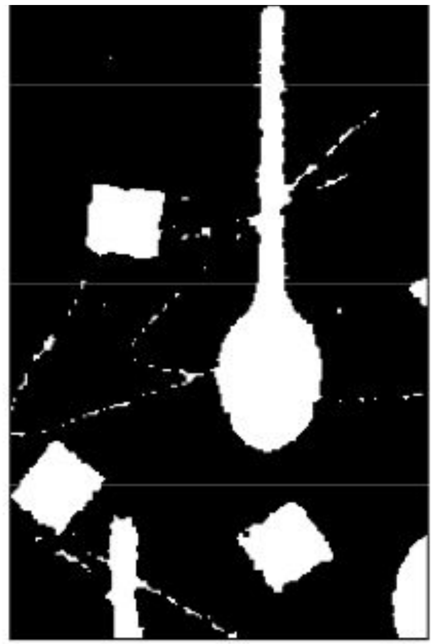


Операция «сужение»

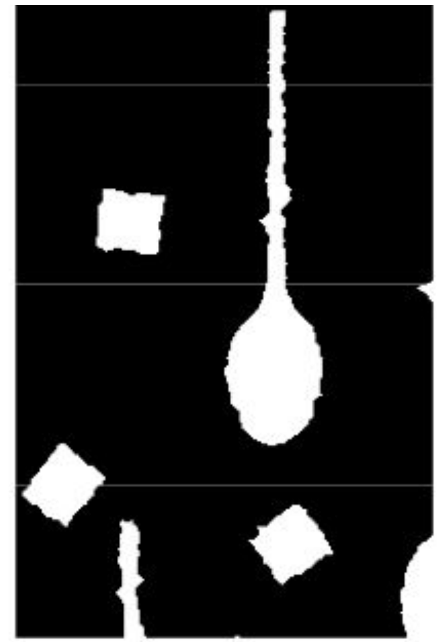


Операция «сужение»

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & [1] & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & [1] & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & [1] & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Метрики

- Евклидово расстояние:

$$Д_E(p,q)=[(x-s)^2+(y-t)^2]^{1/2}$$

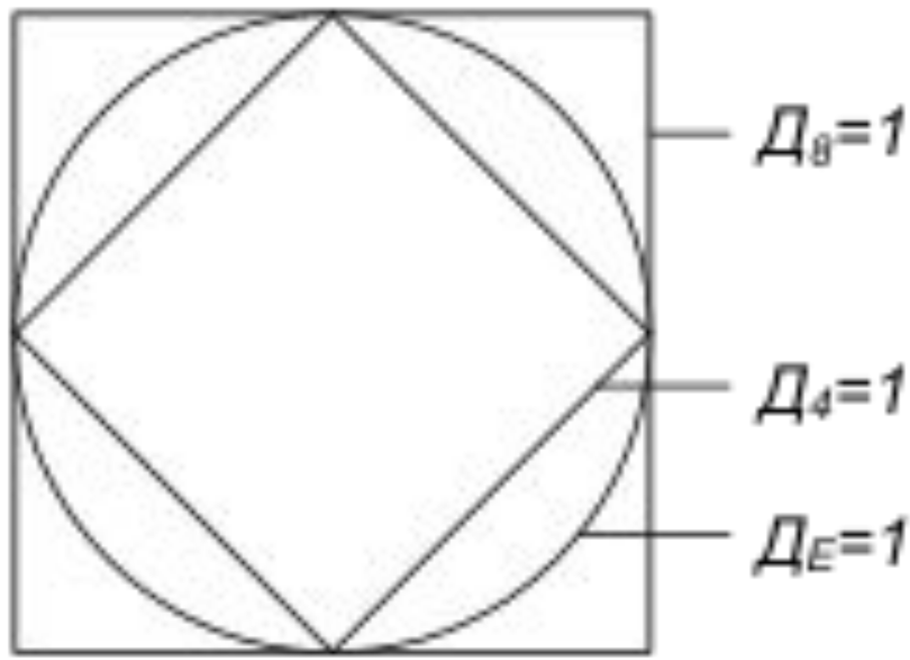
- Модульное расстояние (метрика городских кварталов):

$$Д_4(p,q)=|x-s|+|y-t|$$

- Шахматное расстояние:

$$Д_8(p,q) = \max\{|x-s|,|y-t|\}$$

Метрики



Важное замечание

Результат морфологических операций во многом определяется применяемым структурным элементом. Выбирая различный структурный элемент можно решать разные задачи обработки изображений:

- Шумоподавление
- Выделение границ объекта
- Выделение скелета объекта
- Выделение сломанных зубьев на изображении шестерни

Операция выделения контура объекта

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

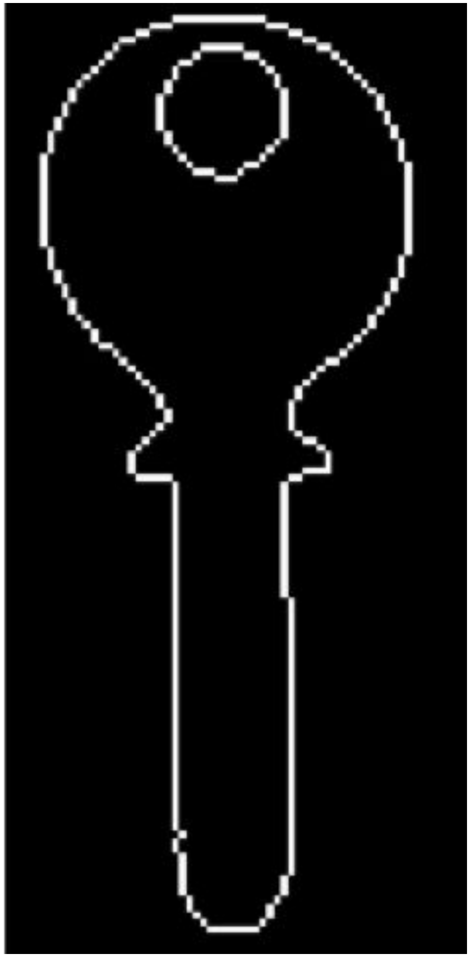
- Внутреннее оконтуривание

$$C_I = A - (A(-)B)$$

- Внешнее оконтуривание

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

Операция выделения контура объекта



Операции раскрытия и закрытия

- Морфологическое раскрытие (opening)

  open(A,B)=(A(-)B)(+)B

- Морфологическое закрытие (closing)

  close(A, B) = (A (+) B) (-) B

Применение открытия

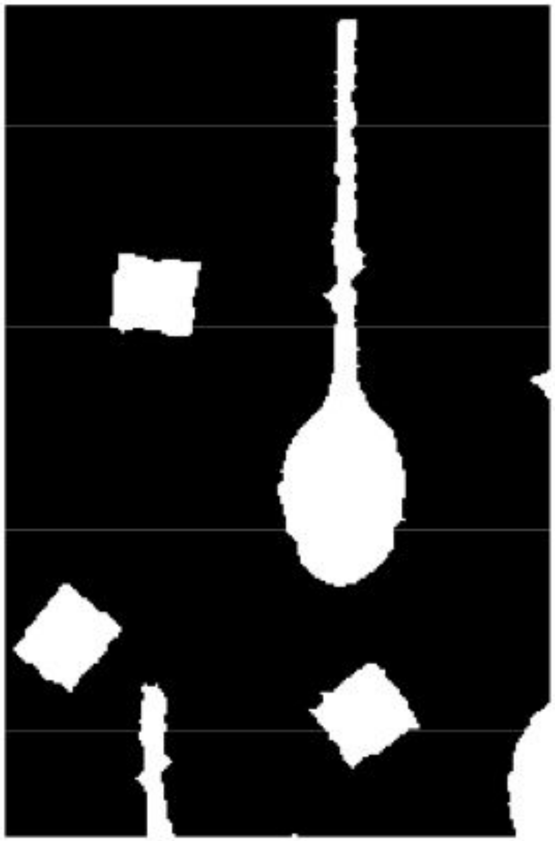
с сильным шумом:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

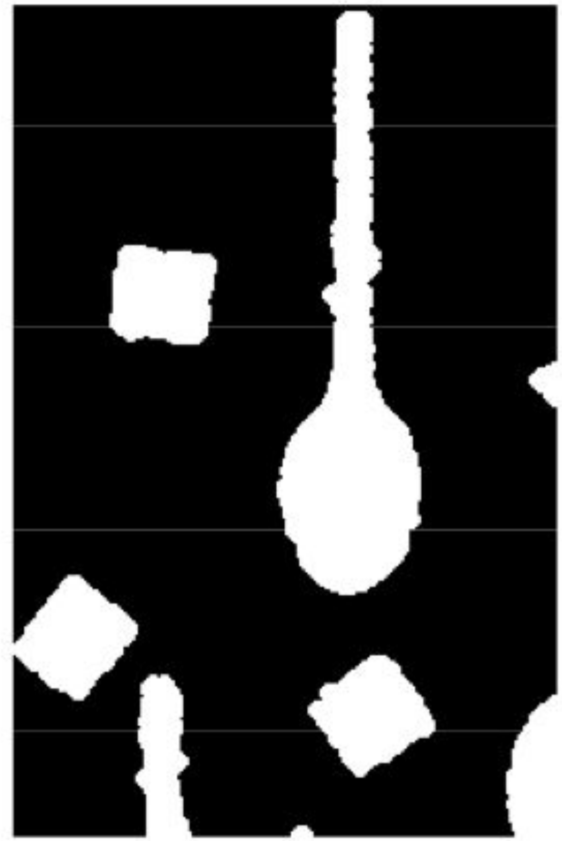
$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Сужение vs Открытие



Сужение



Открытие

Дефекты бинаризации

Применение закрытия

- Применим операцию закрытия к изображению с дефектами объектов:

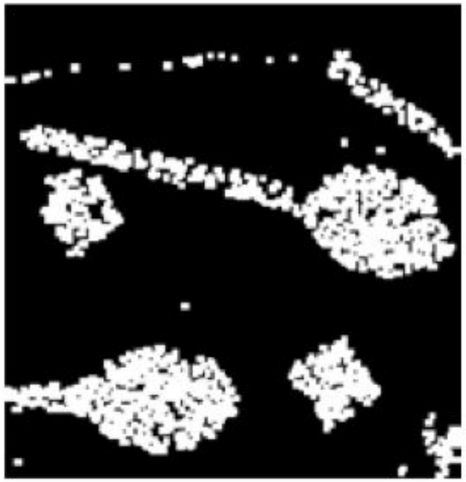
$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Не лучший пример для морфологии

Применение операции «открытия»



$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

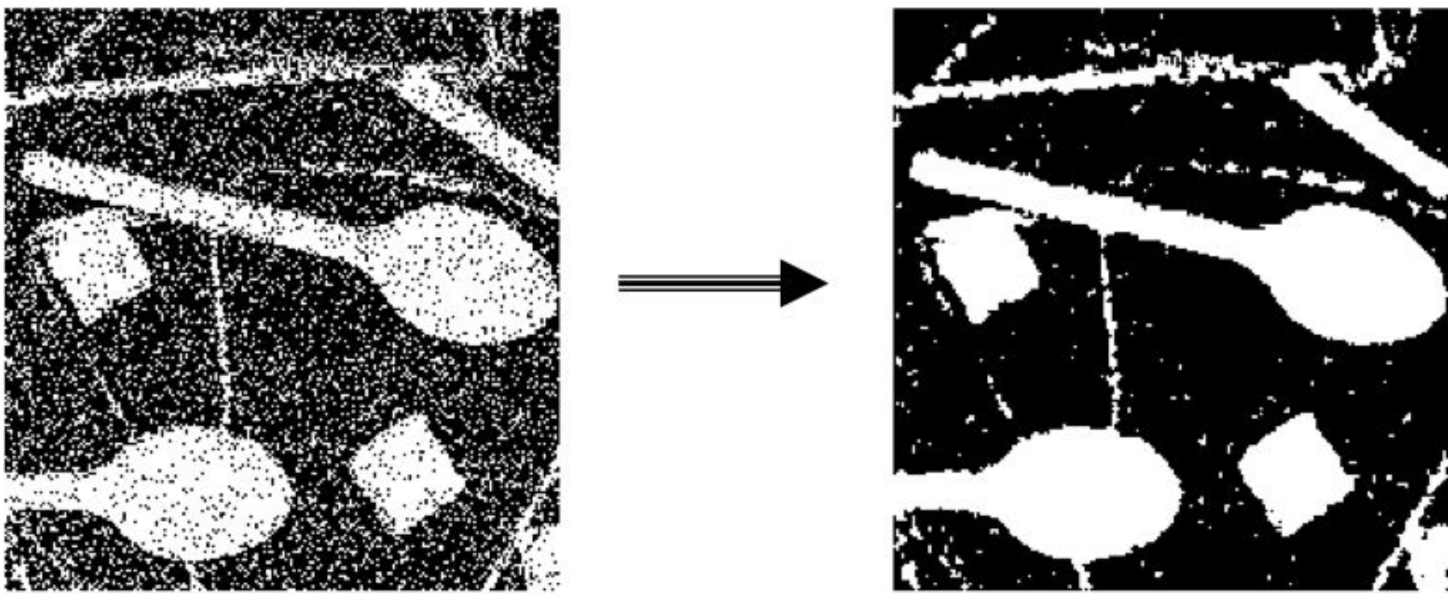
$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

- Часто помогает медианная фильтрация!

Медианный фильтр

- Фильтр с окрестностью 3x3



Теперь можем с помощью морфологии убрать оставшиеся точки, тонкие линии и т.д.

Что дальше?

Получили бинарное изображение

Нужна карта разметки

Выделение связных областей

Определение связной области:

- Множество пикселей, у каждого пикселя которого есть хотя бы один сосед, принадлежащий данному множеству.

- Соседи пикселей:

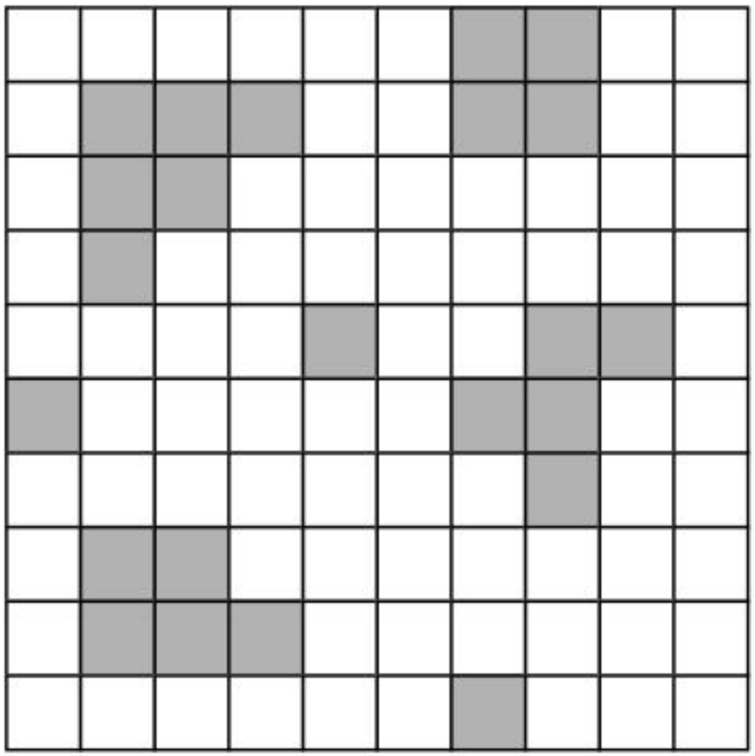
	1	
2	*	3
	4	

4-связность

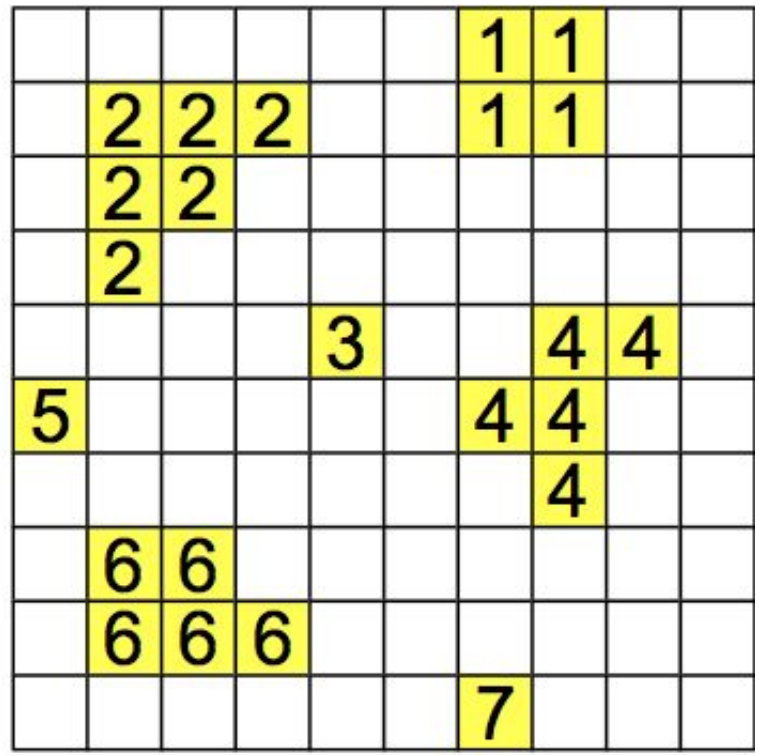
1	2	3
4	*	5
6	7	8

8-связность

Разметка связных областей



Бинарное изображение



Размеченное изображение

Рекурсивный алгоритм

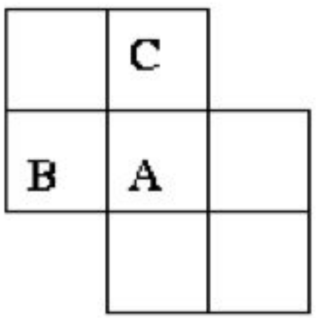
```
void Labeling(BIT* img[], int* labels[])
{
  // labels должна быть обнулена
  L = 1;
  for(y = 0; y < H; y++)
    for(x = 0; x < W; x++)
    {
      Fill(img, labels, x, y, L++);
    }
}
```

Рекурсивный алгоритм

```
void Fill(BIT* img[], int* labels[], int x, int y, int L)
{
   if( (labels[x][y] = = 0) && (img[x][y] = = 1) )
   {
         labels[x][y] = L;
         if( x > 0 )
           Fill(img, labels, x – 1, y, L);
         if( x < W - 1 )
           Fill(img, labels, x + 1, y, L);
         if( y > 0 )
           Fill(img, labels, x, y - 1, L);
         if( y < H - 1 )
           Fill(img, labels, x, y + 1, L);
   }
}
```

Последовательное сканирование

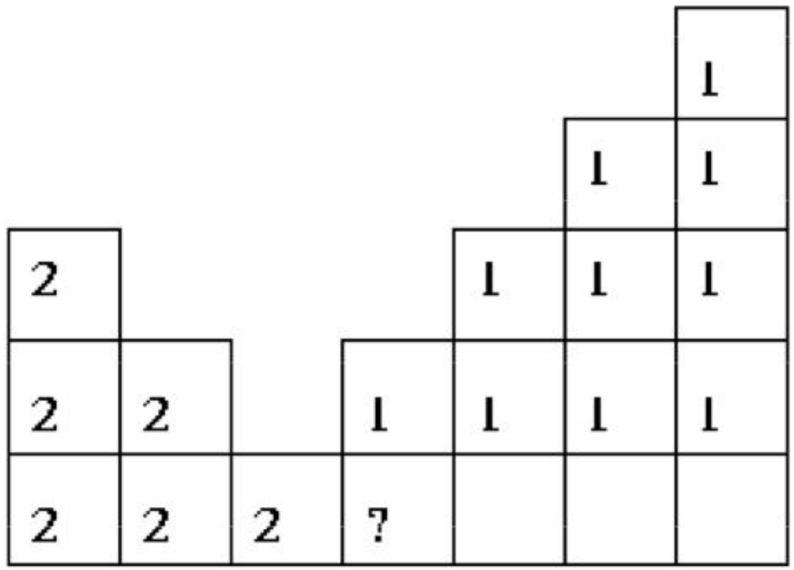
Последовательно, сканируем бинарное изображение сверху вниз, слева направо:




```
if A = O
     do nothing

else if (not B labeled) and (not C labeled)
     increment label numbering and label A

else if B xor C labeled
     copy label to A

else if B and C labeled
     if B label = C label
          copy label to A
     else
          copy either B label or C label to A
          record equivalence of labels
```

Последовательное сканирование

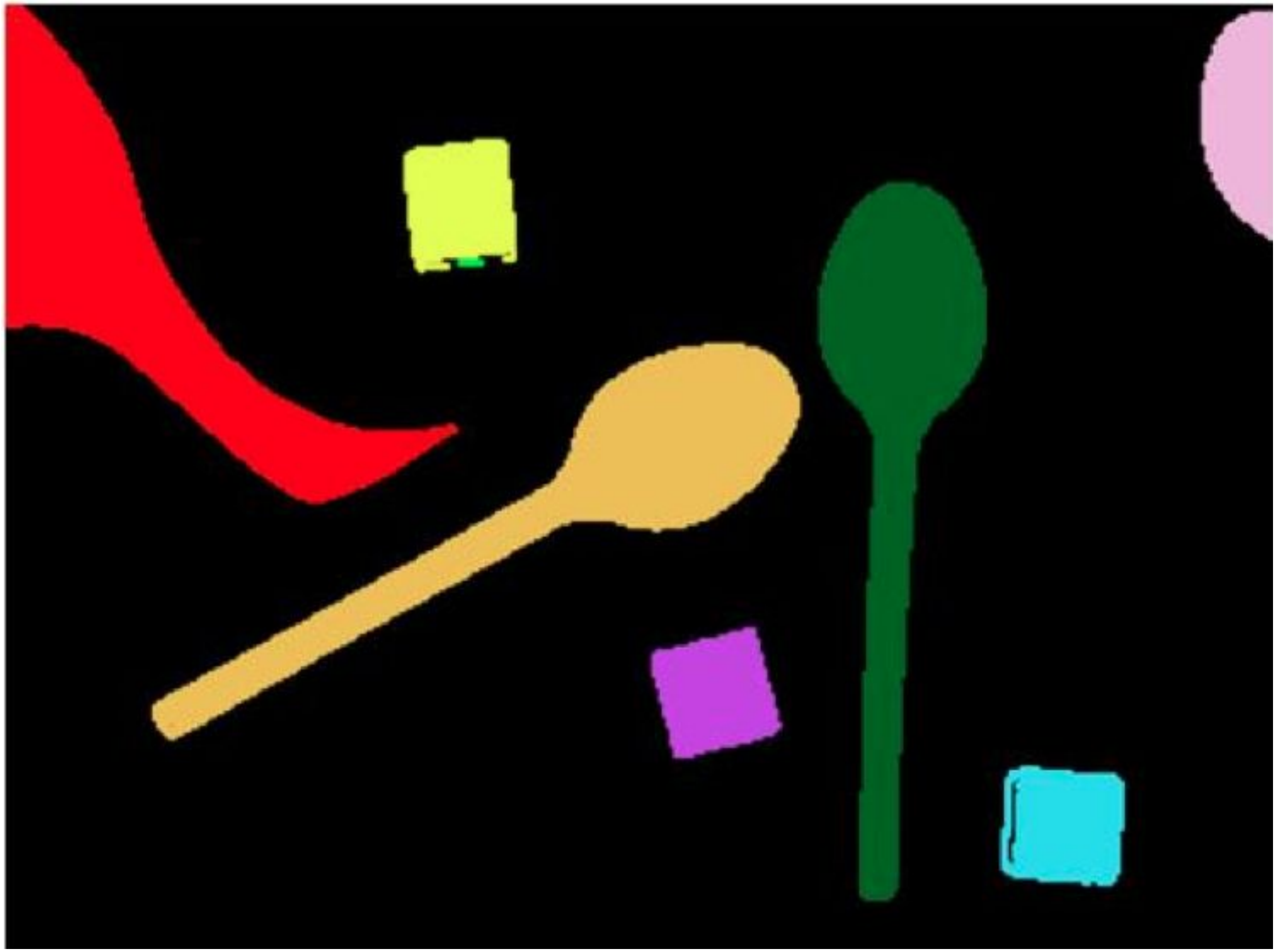
Случай конфликта:



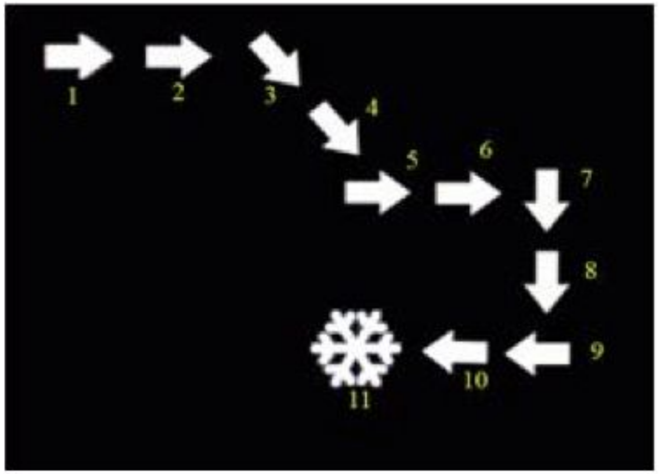
Постобработка - переразметка с учетом эквивалентностей областей

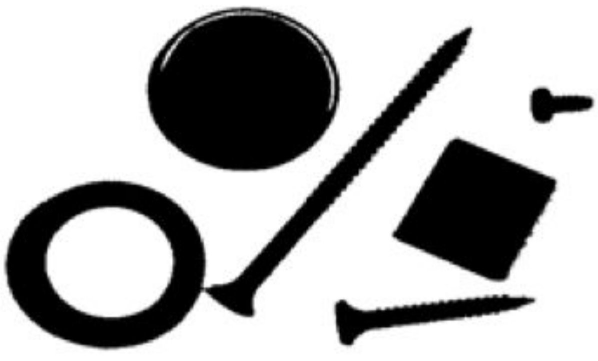
(второй проход в алгоритме)

Выделенные связанные компоненты



Анализ выделенных областей



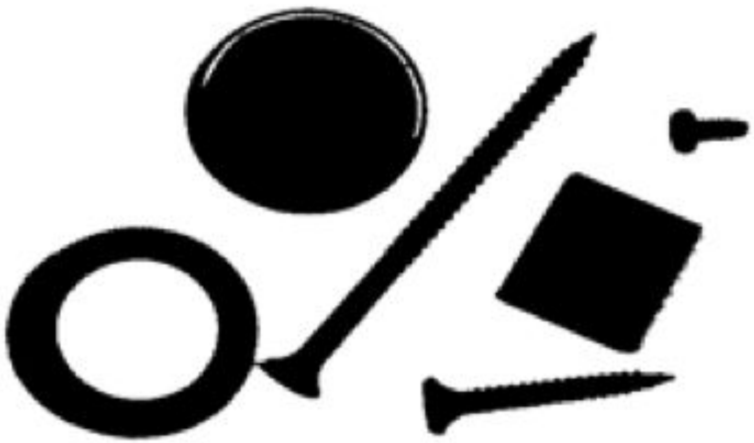


Для анализа требуется вычислить некоторые числовые характеристики (признаки) областей:

- геометрические признаки
- фотометрические признаки

На основе этих характеристик можно классифицировать получаемые области

Геометрические признаки



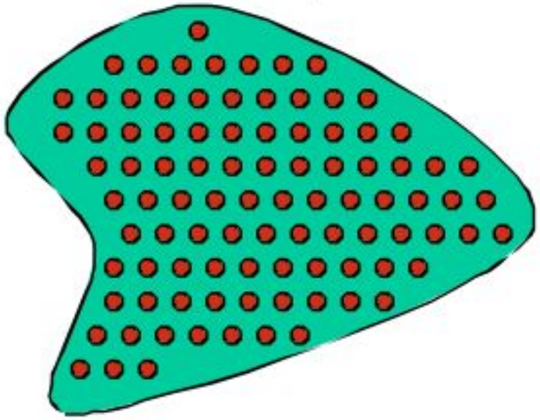
Для каждой области можно подсчитать некий набор простейших числовых характеристик:

- Площадь
- Центр масс
- Периметр
- Компактность
- Ориентацию главной оси инерции
- Удлиненность (эксцентриситет)

Площадь и центр масс

- Площадь – количество пикселей в области;

$$A = \sum_{x=0}^{m} \sum_{y=0}^{n} I(x, y)$$

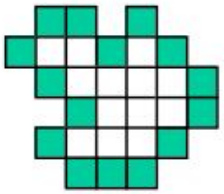


- Центр масс

$$\overline{x} = \frac{\sum_{x=0}^{m} \sum_{y=0}^{n} xI(x, y)}{A}; \overline{y} = \frac{\sum_{x=0}^{m} \sum_{y=0}^{n} yI(x, y)}{A}$$

Периметр и компактность

- Периметр – количество пикселей принадлежащих границе области;



- Компактность – отношение квадрата периметра к площади;

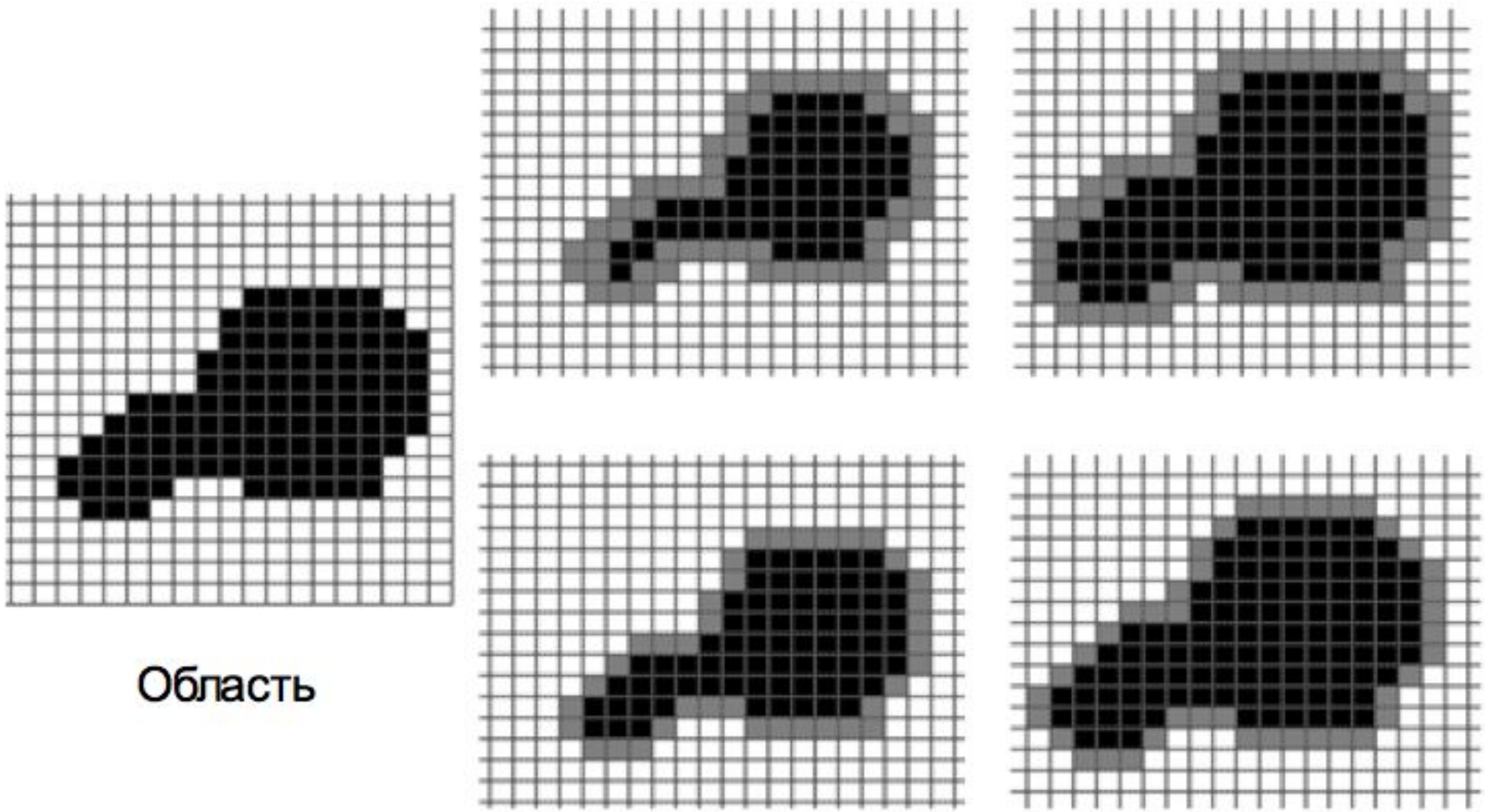
$$C = \frac{P^2}{A}$$

Подсчет периметра области

- Пиксель лежит на границе области, если он сам принадлежит области и хотя бы один из его соседей области не принадлежит. (*внутренняя граница*)
- Пиксель лежит на границе области, если он сам не принадлежит области и хотя бы один из его соседей области принадлежит. (*внешняя граница*)

Периметр зависит также от того 4-х или 8-ми связность используется для определения соседей.

Пример периметров области



Область

Внутренняя граница

Внешняя граница

Инвариантные характеристики

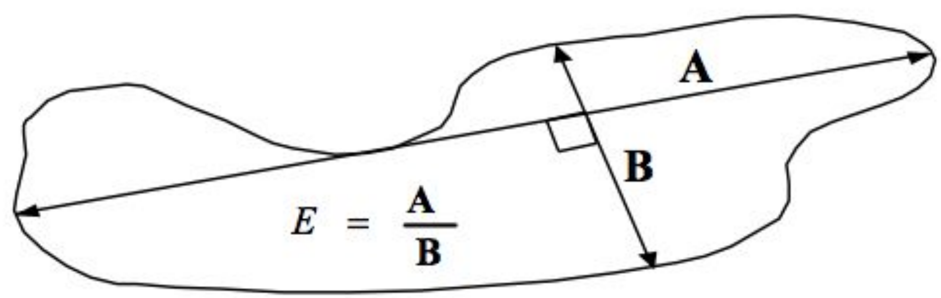
Для распознавания нас интересуют характеристики инвариантные по отношению к масштабированию, переносу, повороту:

- Удлиненность, нецентрированность (эксцентриситет)

$$elongation = \frac{m_{20} + m_{02} + \sqrt{(m_{20} - m_{02})^2 + 4m_{11}^2}}{m_{20} + m_{02} - \sqrt{(m_{20} - m_{02})^2 + 4m_{11}^2}}$$

- Компактность

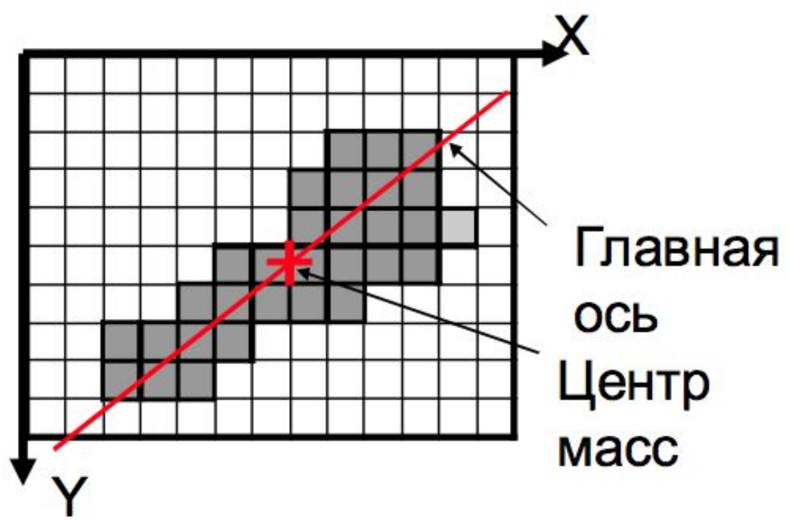
$$C = \frac{P^2}{A}$$



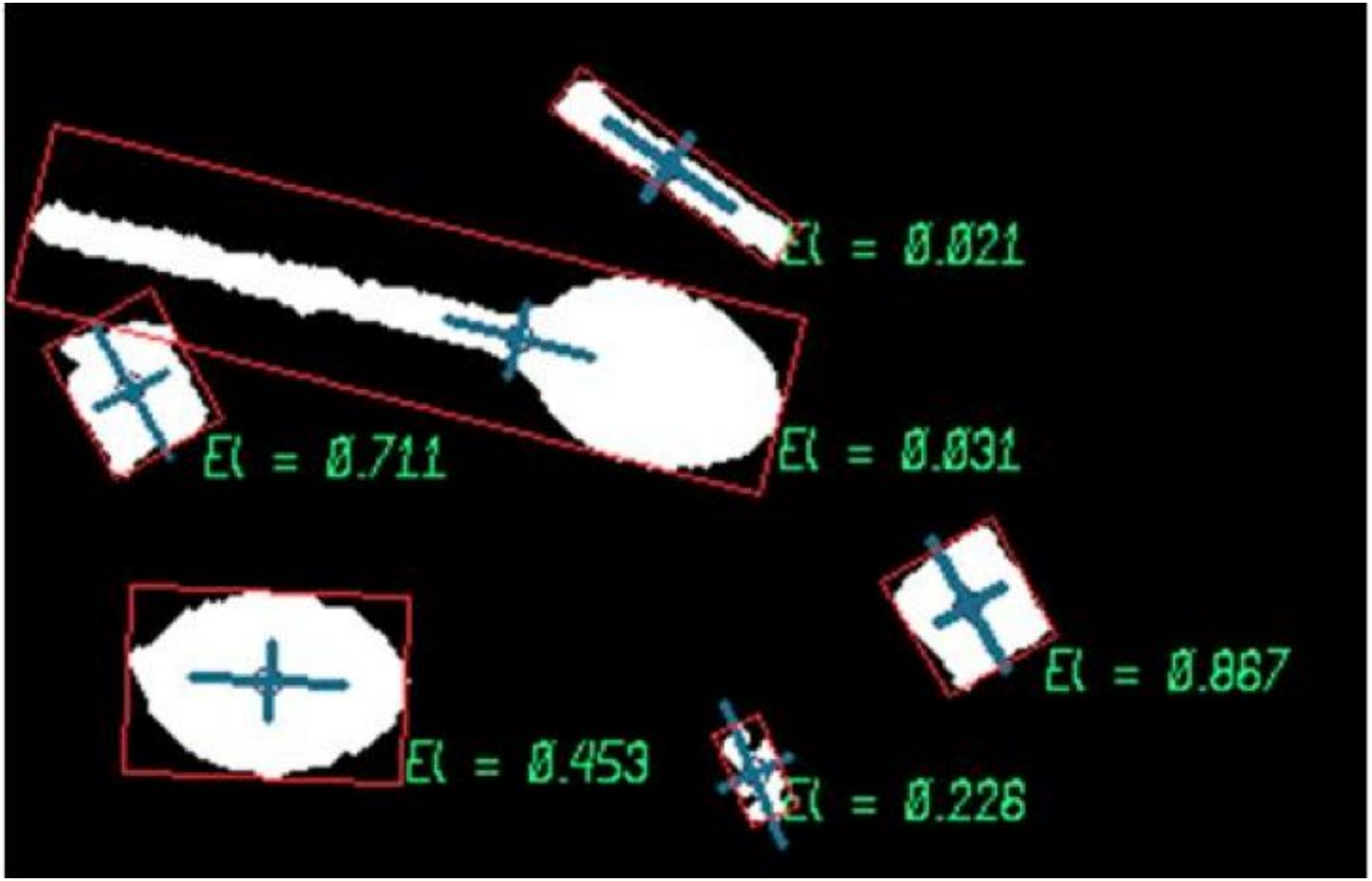
Ориентация главной оси инерции

Не является инвариантной к повороту, но в ряде случаев предоставляет полезную информацию об ориентации объекта:

$$\theta = \frac{1}{2}\arctan\left(\frac{2m_{11}}{m_{20} - m_{02}}\right)$$



Пример



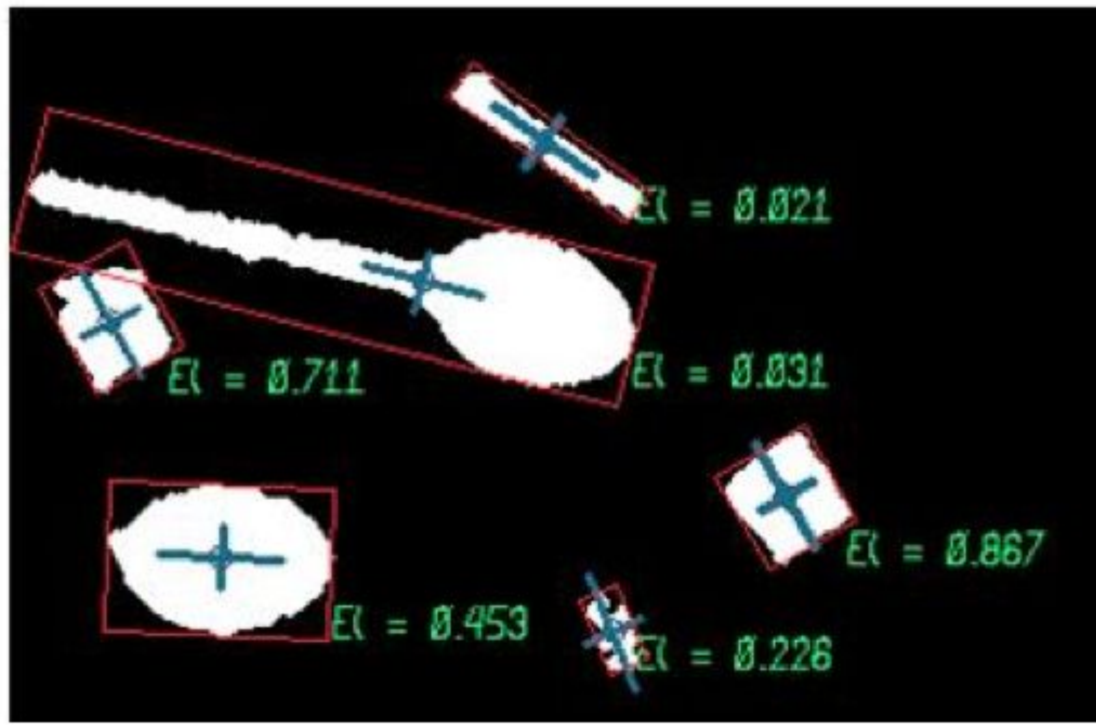
Фотометрические признаки

Для каждой области можно подсчитать некий набор простейших числовых характеристик:

- Средняя яркость
- Средний цвет (если изображение цветное)
- Гистограмма распределения яркостей (или три гистограммы распределения R, G, B)
- Дисперсию (разброс) яркостей или цвета

Разумеется, все это считается по исходному, а не бинарному изображению!

Как анализировать признаки



Как анализировать признаки

Как воспользоваться признаками для классификации?

- Подобрать диапазоны значений для разных классов вручную, экспериментально (может быть весьма трудоемко)
- Подобрать диапазоны значений графически (нужна база для тренировки, трудно, если признаков много)
- Обучить классификатор с помощью машинного обучения

Ручной подбор

- Из общих соображений:
  - Ложки более вытянутые, чем сахарные кусочки
  - Ложки больше чем сахарные кусочки
  - Сахарные кусочки квадратные
  - Области появляющиеся из-за шума обычно небольшие и неквадратные

- Пытаемся сконструировать решающее правило, проверяем экспериментально

- Может быть весьма утомительно

Графический анализ

- Собрать тренировочную базу изображений
  - Где только ложки
  - Где только сахар
  - Где только шум

Как получить такие?

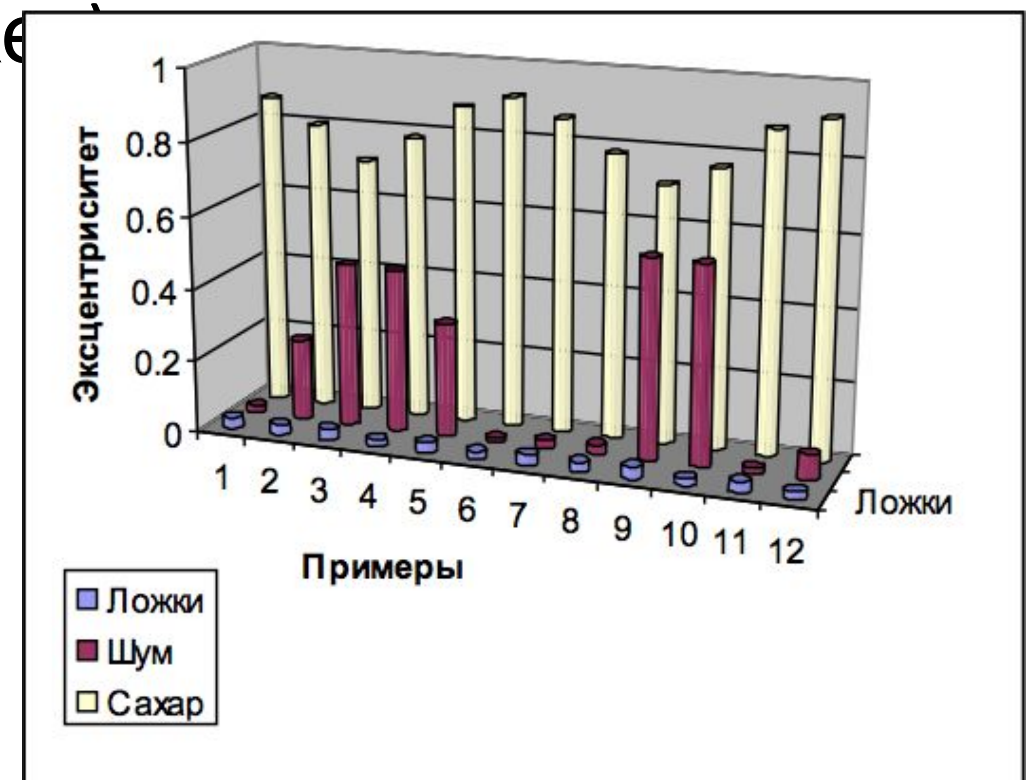
Да просто закрасить все остальное.

- Брать признаки и строить графики

Графический анализ

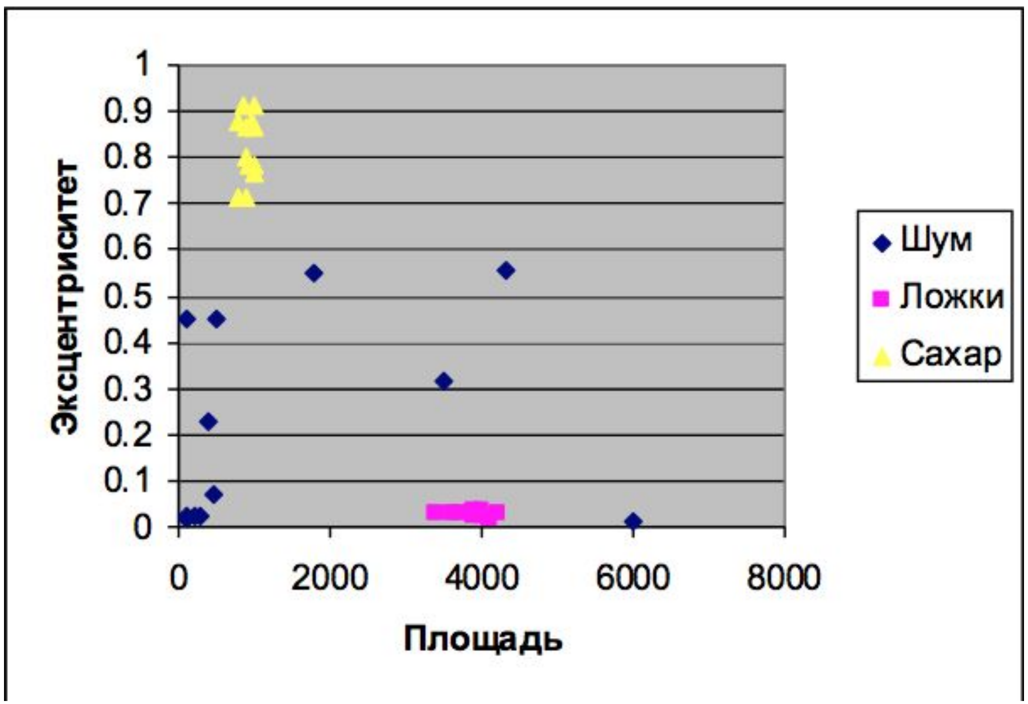
- Диаграмма распределения эксцентриситета

(проблема – не получается отличить шум от ложе



Графический анализ

- График распределения эксцентриситета и площади (гораздо лучше – можем подобрать значения порогов)



Метод k-средних

- Метод k-средних – метод кластеризации данных. Целью задачи кластеризации является разбиение множества объектов на кластеры (классы) на основе некоторой меры сходства объектов.

Метод k-средних

- *Дано:*

Набор векторов , $i = 1,\ldots, p$;

k – число кластеров, на которые нужно разбить набор .

- *Найти:*

k средних векторов m_j, $j = 1,\ldots, k$ (центров кластеров);

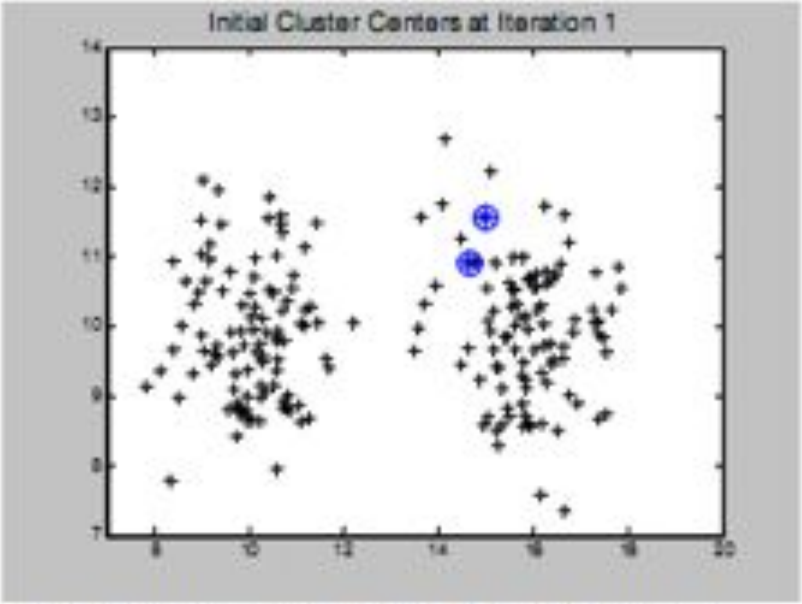
отнести каждый из векторов к одному из k кластеров;

Метод k-средних

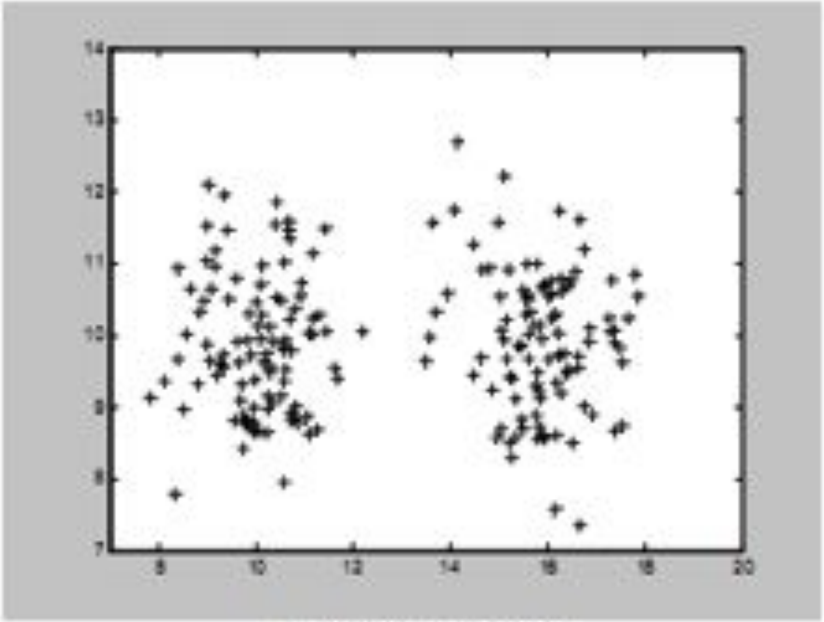
Алгоритм:

1. Случайным образом выбрать k средних m_j $j = 1,\ldots, k$;
2. Для каждого x_i $i = 1,\ldots,p$ подсчитать расстояние до каждого из m_j $j=1,\ldots, k$, отнести (приписать) x_i к кластеру j', расстояние до центра которого $m_{j'}$ минимально;
3. Пересчитать средние m_j $j=1,\ldots, k$ по всем кластерам;
4. Повторять шаги 2, 3, пока кластеры не перестанут изменяться

Метод k-средних

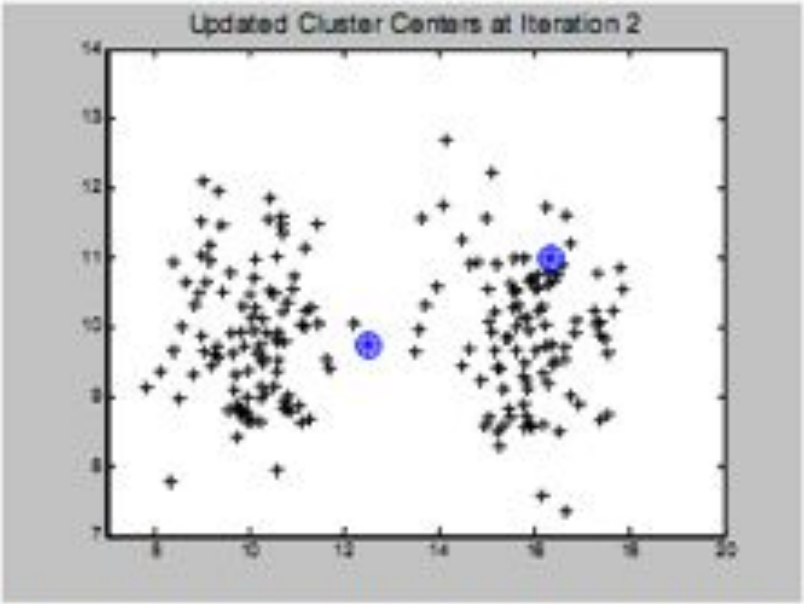


Случайная инициализация центров кластеров (шаг 1)

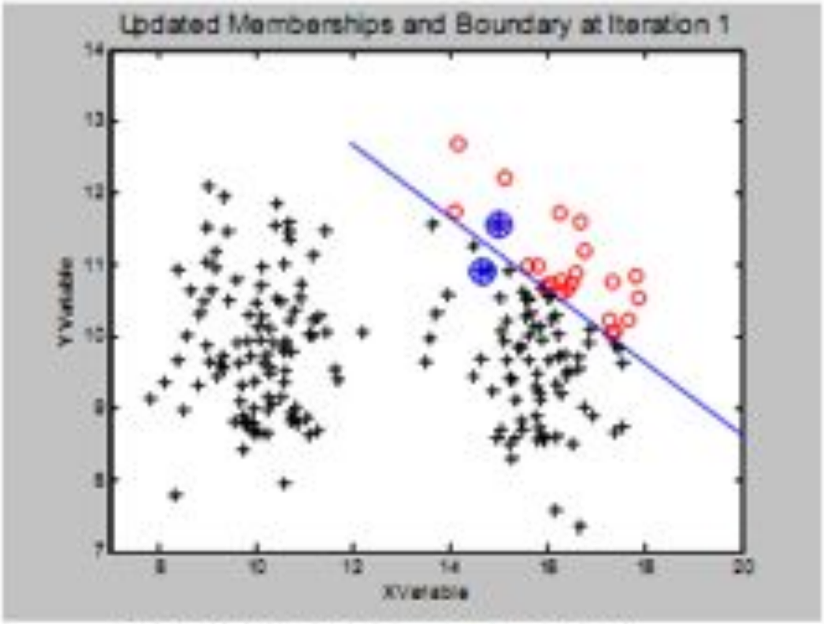


Исходные данные

Метод k-средних

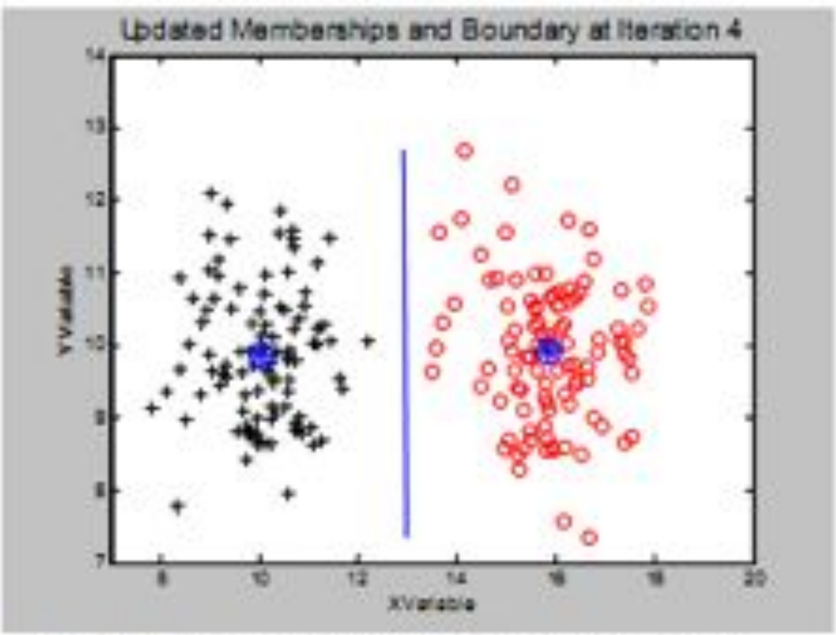


Пересчет центров кластеров после первой итерации (шаг 3)

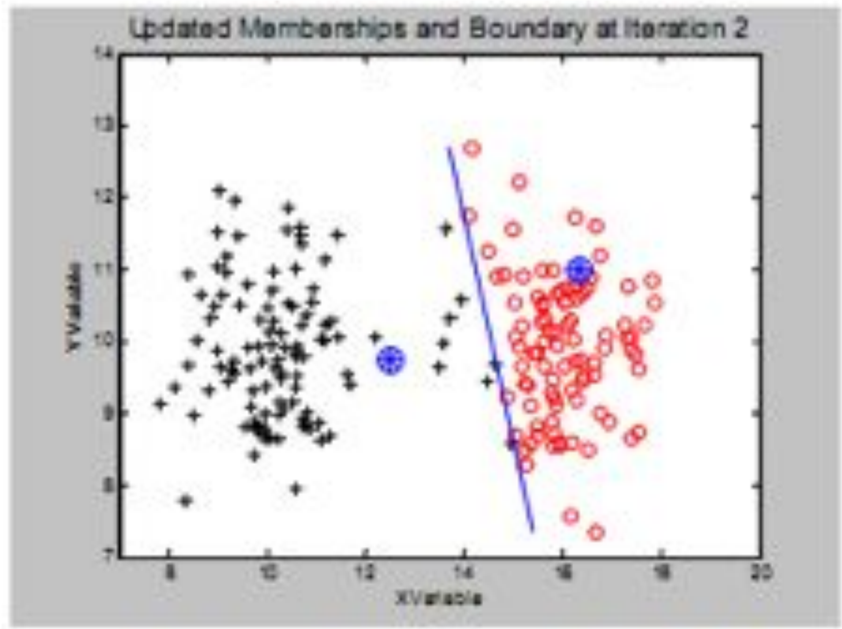


Кластеры после первой итерации (шаг 2)

Метод k-средних



Стабильная конфигурация после четвертой итерации



Кластеры после второй итерации (шаг 2)

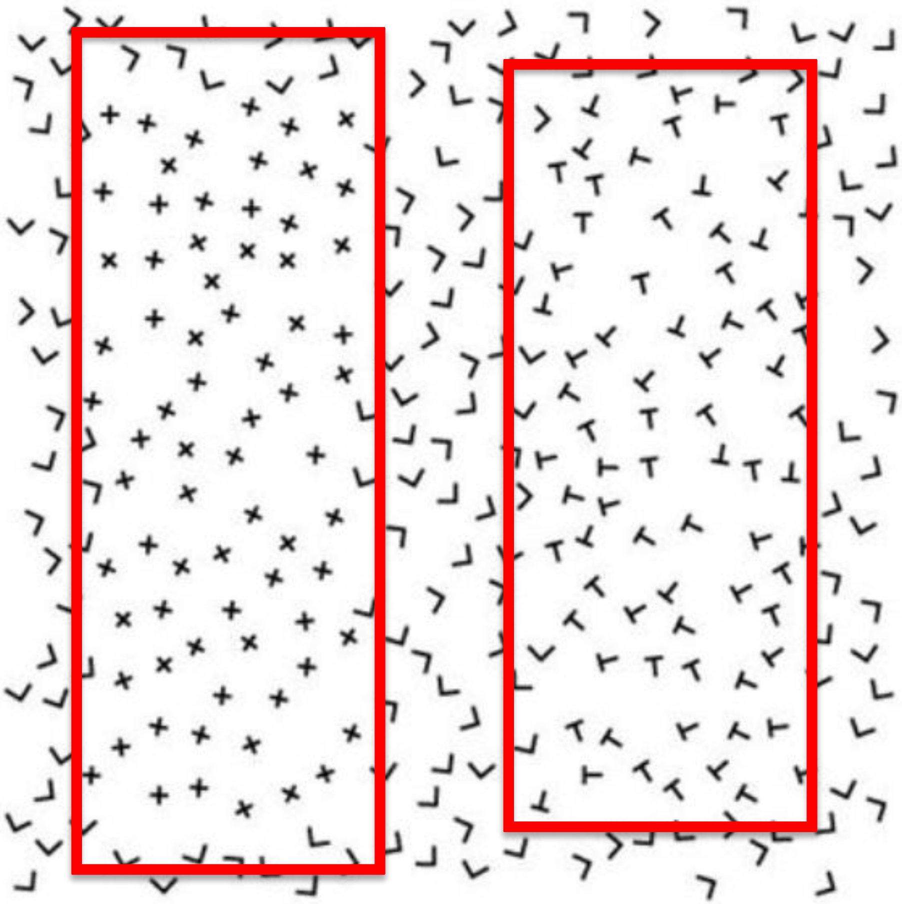
Недостатки

- Не гарантируется достижение глобального минимума суммарного квадратичного отклонения V, а только одного из локальных минимумов.
- Результат зависит от выбора исходных центров кластеров, их оптимальный выбор неизвестен.
- Число кластеров надо знать заранее.

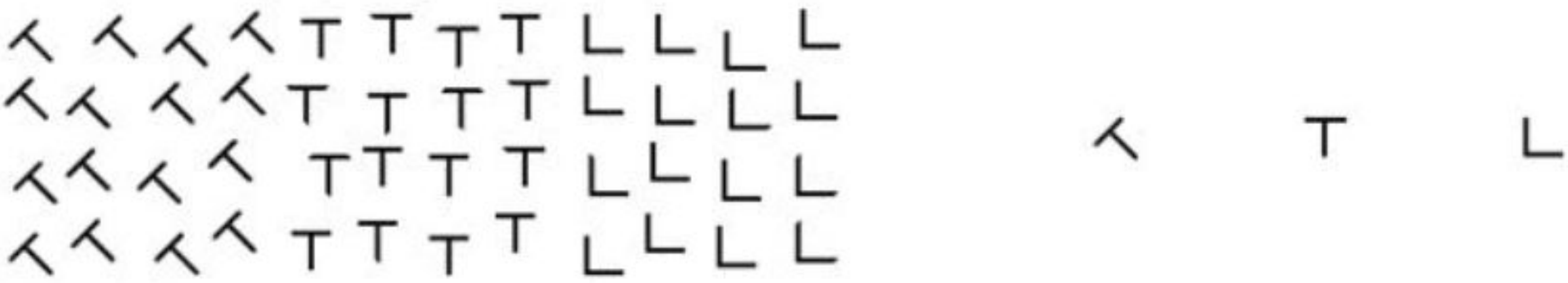
Признаки изображения

- Какие признаки мы можем использовать для сравнения пикселей и регионов?
- Яркость
- Цвет
- ?

Пример



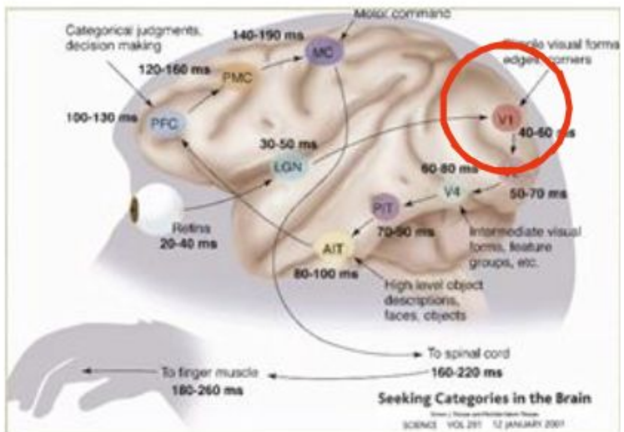
Текстура

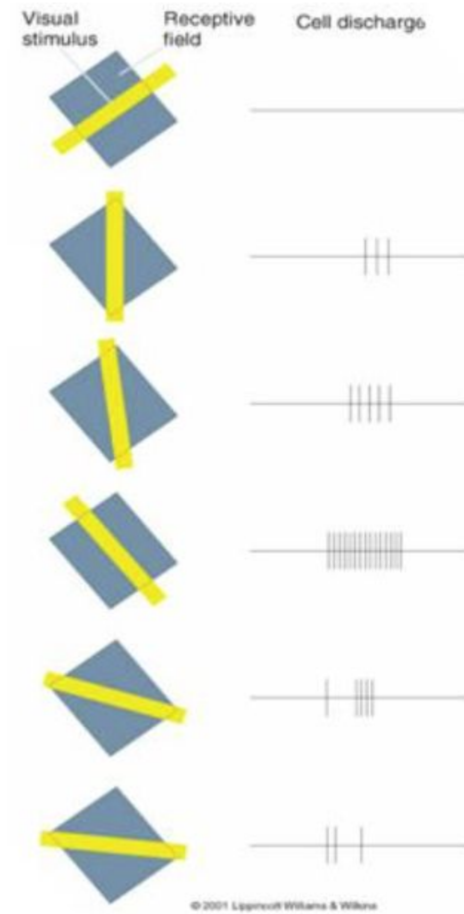


(a) (b)

- Это типичные примеры текстурных шаблонов для исследований психофизиологоического восприятия изображений
- Человек явно использует не только яркость и цвет, но и ориентацию краёв (градиентов изображения), их распределение, для анализа изображений
- Текстура — преимущественная ориентация элементов, составляющих материал (одно из определении)

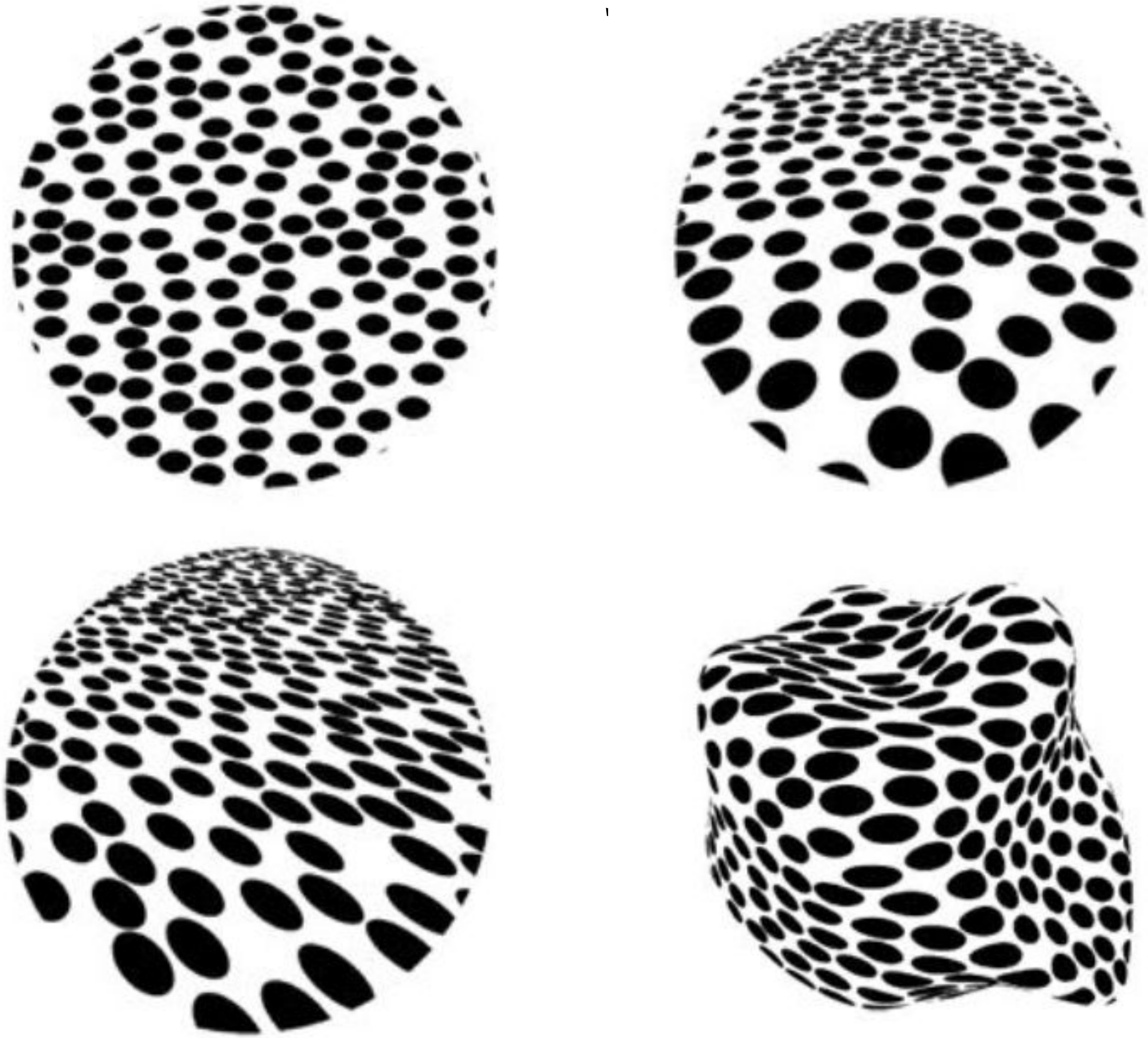
«Простые клетки» V1





- В первичной визуальной коре головного мозга есть клетки, чувствительные к краям определенной ориентации
- Для каждой области есть набор таких клеток, чувствительные к краям разной ориентации

Психологическое свойство текстуры



Форма из текстуры

- Человек интуитивно считает текстуру **изотропной**, т.е. с постоянными свойствами на поверхности объекта
- Shape from texture: Исходя из предположения об изотропности шаблона текстуры, можно определить наклон поверности

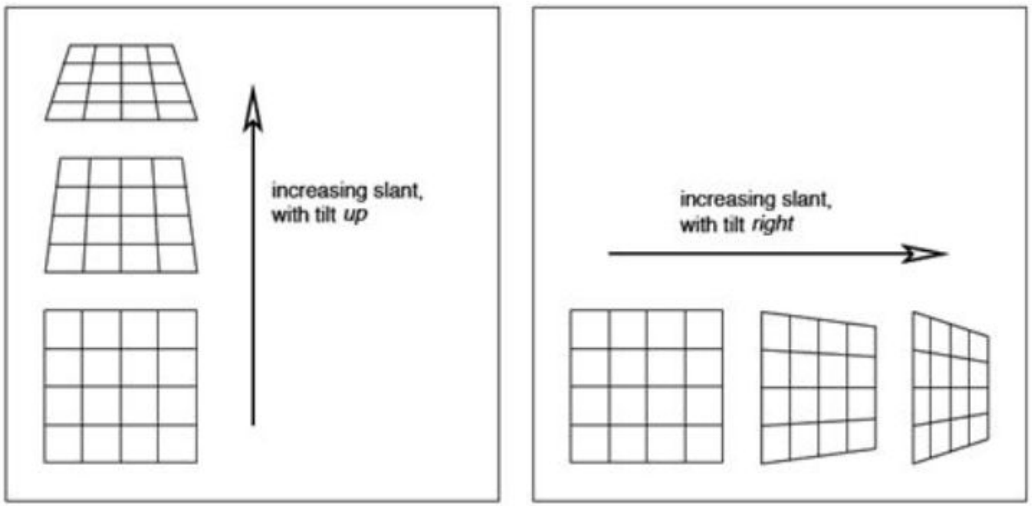
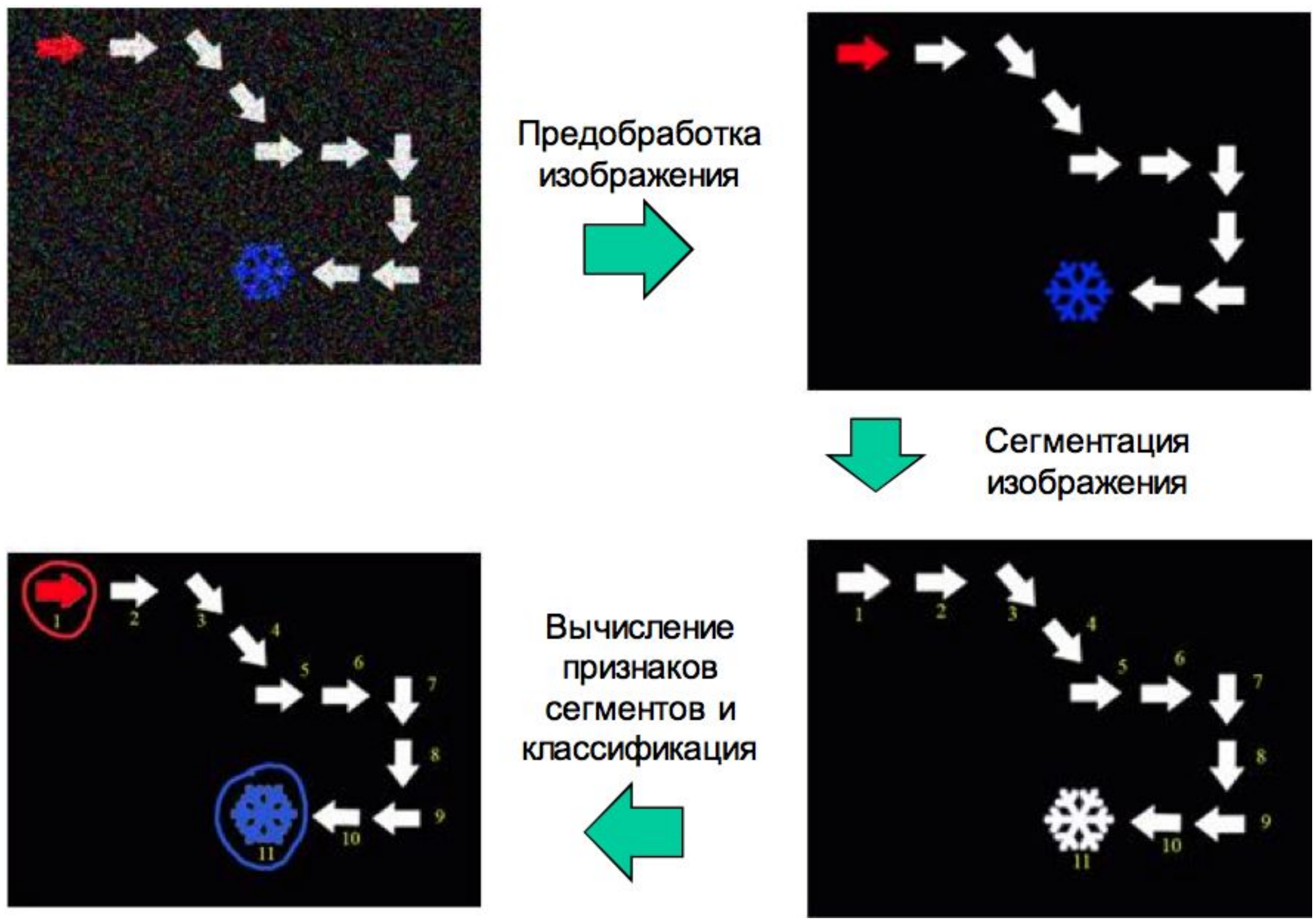
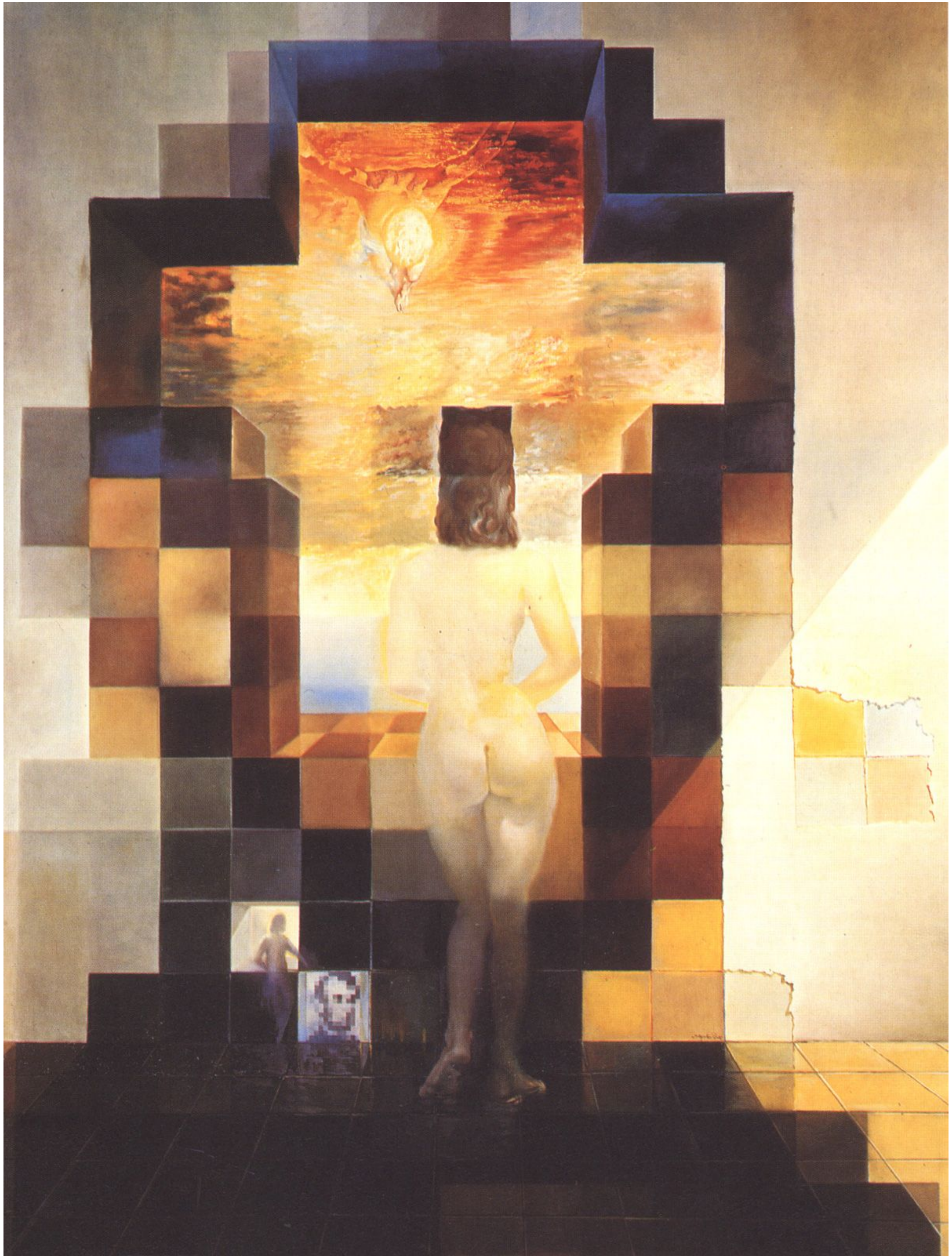


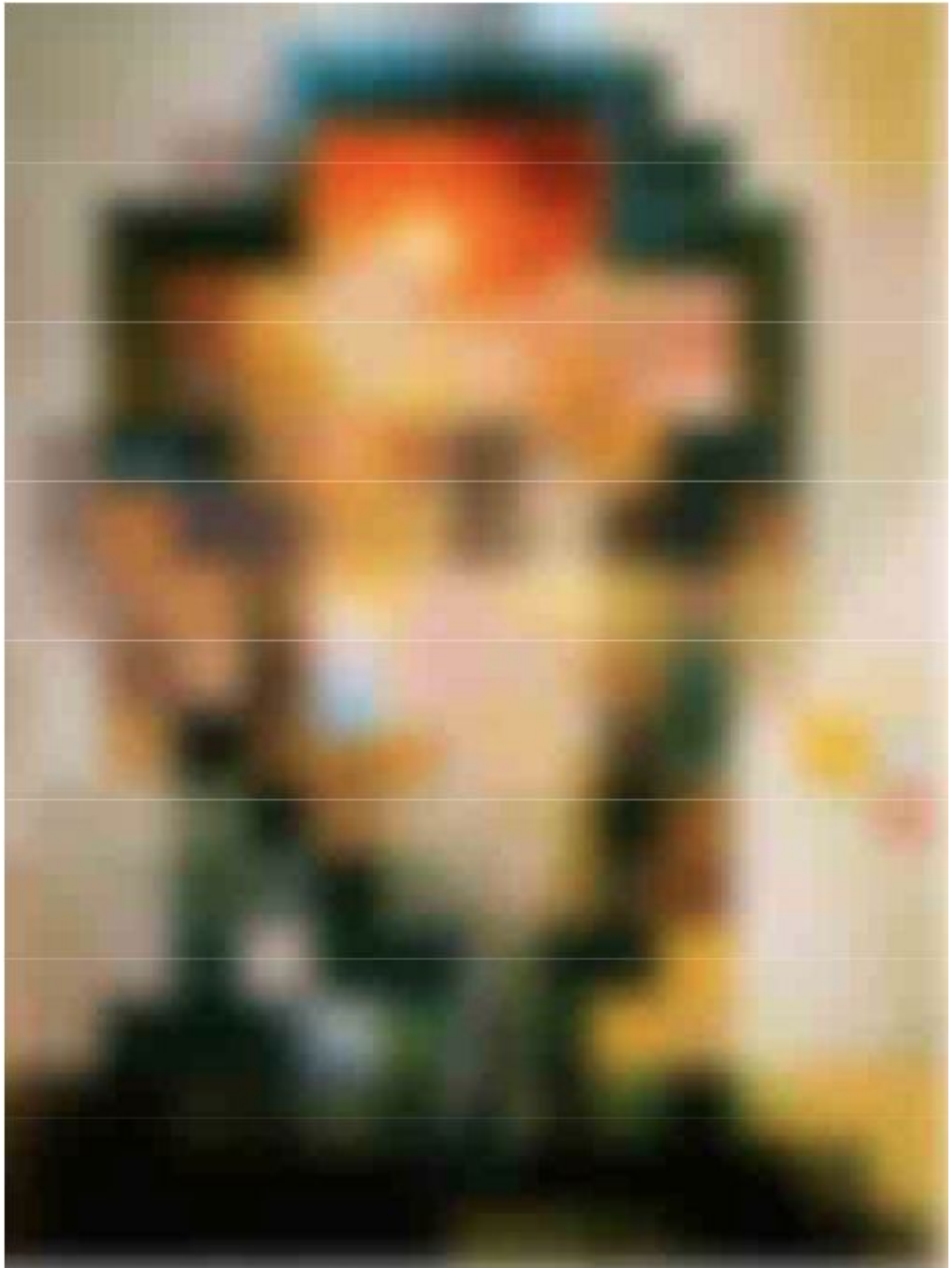
Figure 8.7. Surface orientation is often characterized in terms of *slant* and *tilt*.

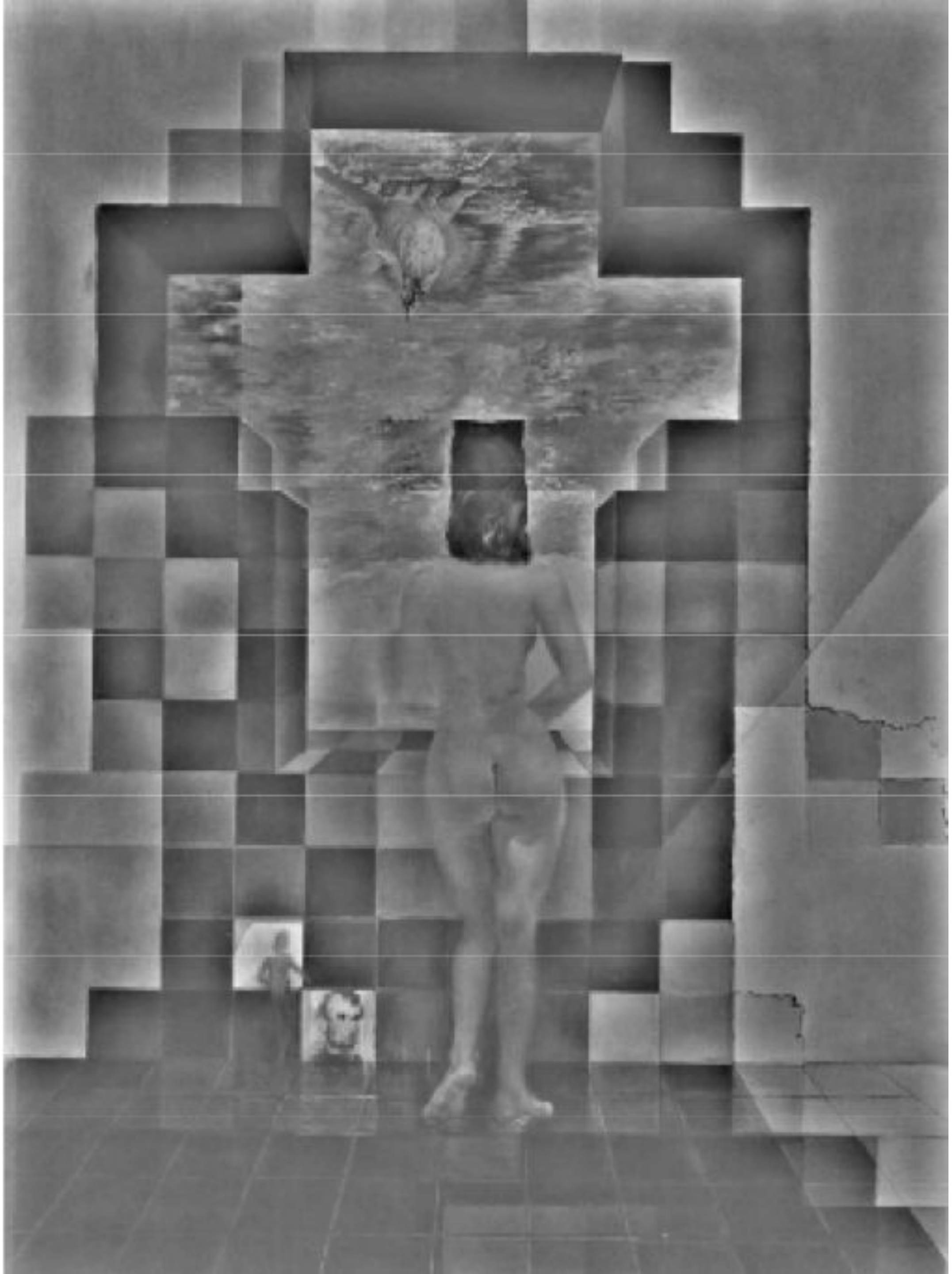
Image source: VPfaCGP Fig 8.7

Схема простого алгоритма









Jean Baptiste Joseph Fourier

- Дикая идея (1807):

Любая периодическая функция может быть представлена как взвешенная сумма синусов и косинусов различной частоты

- Воспринята была не сразу:

Ни Лагранж, ни Лаплас, Пуассон не верили в это

Впервые переведена работа на английский в 1878 году

- Преобразование Фурье

Преобразование Фурье

- Прямое преобразование Фурье:

$$f(x) \longrightarrow \boxed{\text{Fourier Transform}} \longrightarrow F(\omega)$$

Для каждой ω от 0 до бесконечности, ***F(ω)*** содержит амплитуду *A* и фазу ϕ соответствующего синуса

- Для удобной записи используются мнимые числа:

$$F(\omega) = R(\omega) + iI(\omega)$$

$$A = \pm\sqrt{R(\omega)^2 + I(\omega)^2} \qquad \phi = \tan^{-1}\frac{I(\omega)}{R(\omega)}$$

Преобразование Фурье

- Базисные функции образуют N-мерный ортогональный базис в пространстве N-мерных векторов исходных сигналов.
- Весовые коэффициенты вычисляются как скалярное произведение сигнала на базисные функции
- Разложение Фурье обратимо, т.е. по коэффициентам разложения можно точно восстановить исходный дискретный сигнал.

- Обратное преобразование Фурье:

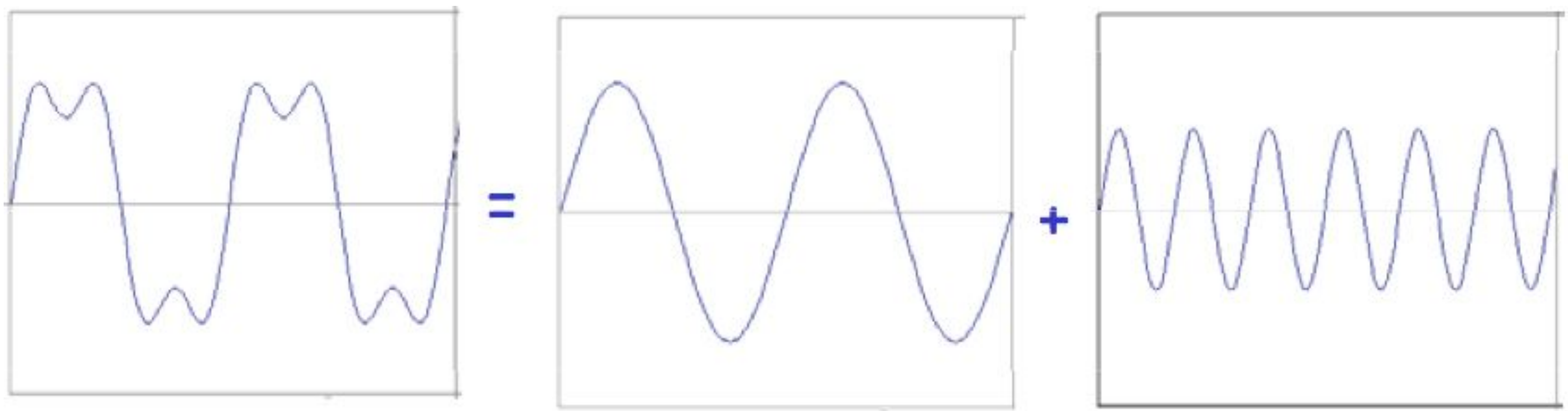
$$\boldsymbol{F(\omega)} \longrightarrow \boxed{\begin{array}{c}\text{Inverse Fourier}\\ \text{Transform}\end{array}} \longrightarrow \boldsymbol{f(x)}$$

Быстрое преобразование Фурье

- Для вычисления всех коэффициентов через скалярное произведение требуется примерно N^2 умножений: очень много при больших длинах сигнала N.
- Быстрое преобразование Фурье (БПФ, FFT) – ускоренный алгоритм вычисления ДПФ
  - Основан на периодичности базисных функций (много одинаковых множителей)
  - Математически точен (ошибки округления даже меньше, т.к. меньше число операций)
  - Число умножений порядка $N \cdot \log_2 N$, намного меньше, чем N2 ► Ограничение: большинство реализаций FFT принимают только массивы длиной $N = 2^m$
- Есть и быстрое обратное преобразование

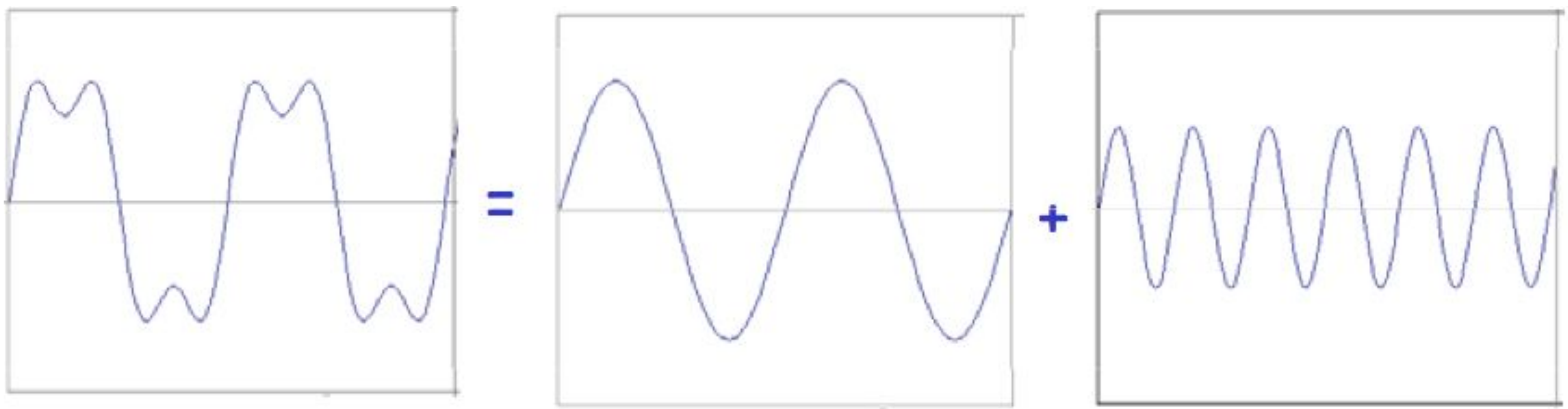
Пример

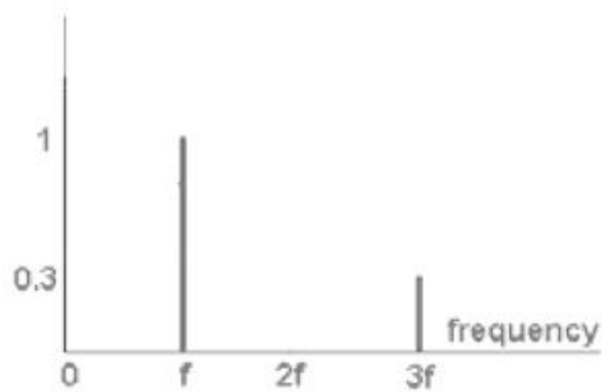
- $g(t) = \sin(2pf\ t) + (1/3)\sin(2p(3f)\ t)$



Пример

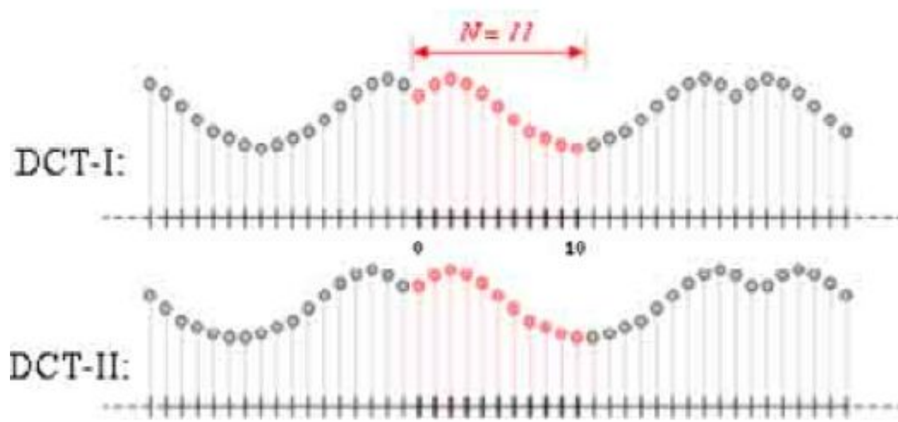
- $g(t) = \sin(2pf\ t) + (1/3)\sin(2p(3f)\ t)$





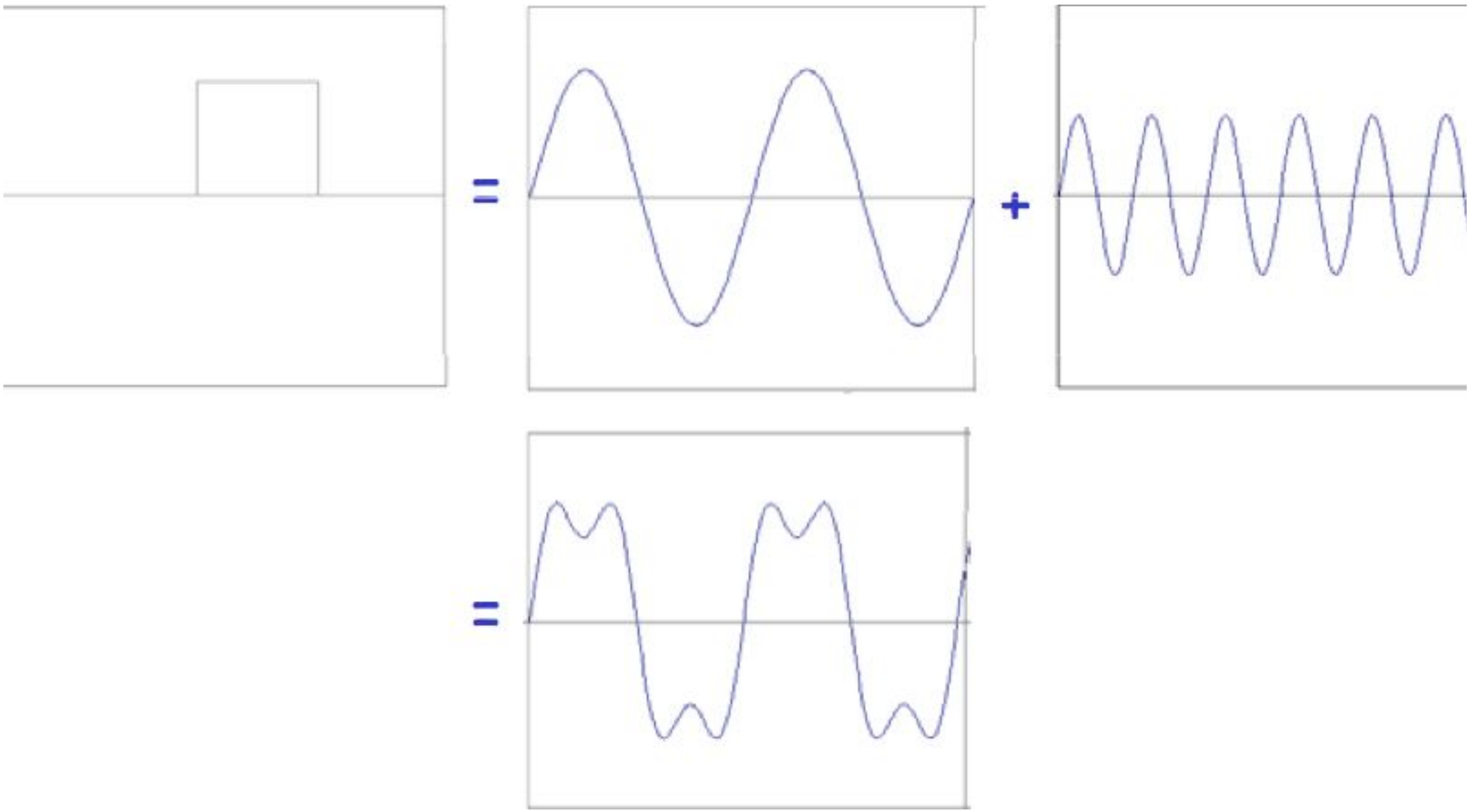
Ограниченный сигнал

- Как быть, если сигнал задан на отрезке?
  - Продлить сигнал за границы отрезка, затем разложить
  - В зависимости от типа разложения, продлять нужно по разному
  - Продление должно быть периодическим
  - Можем использовать только синусы или только косинусы, в зависимости от этого продлевать нужно по-разному
  - Если косинусное преобразование, то продление должно быть чётной функцией

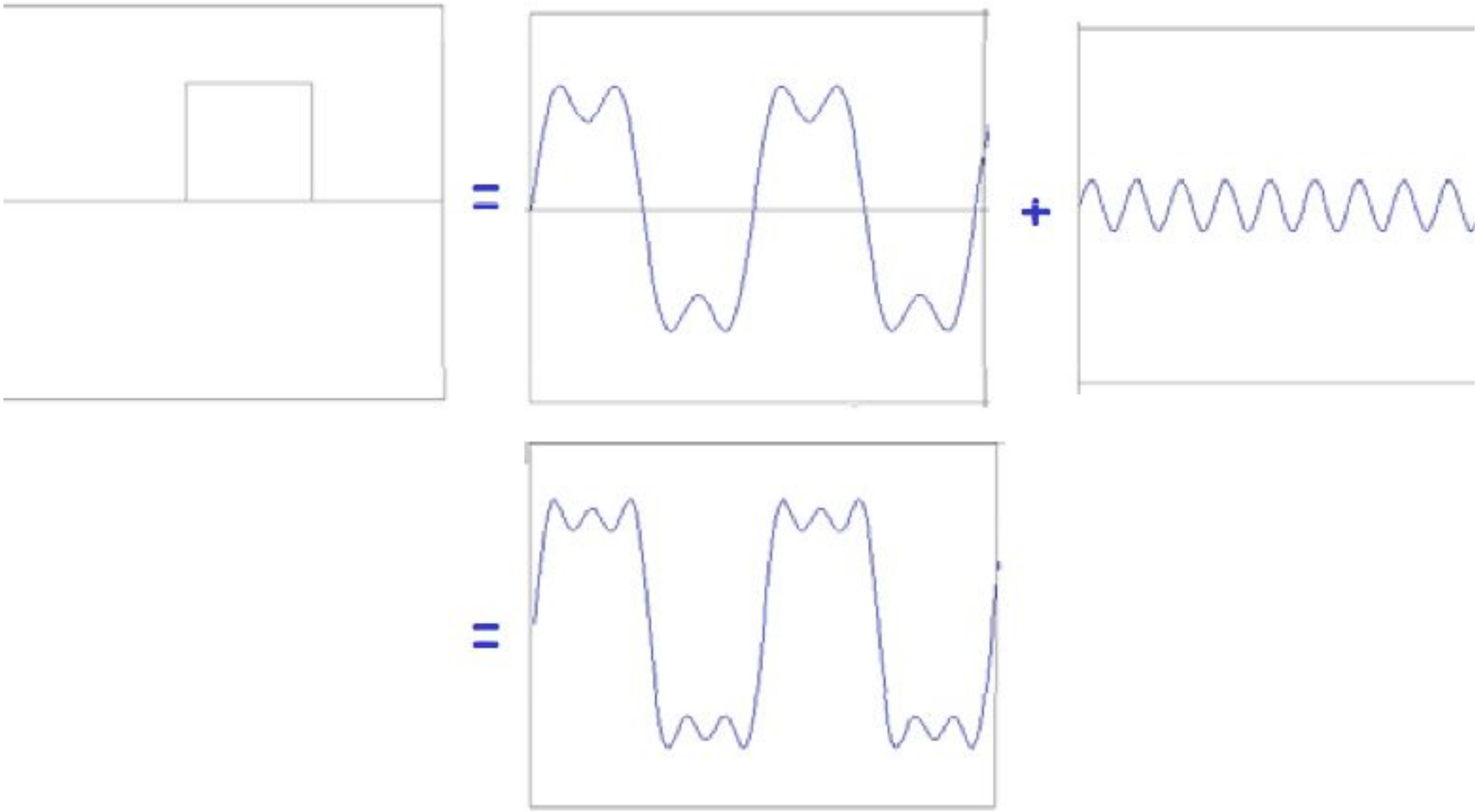


Прямоугольный сигнал

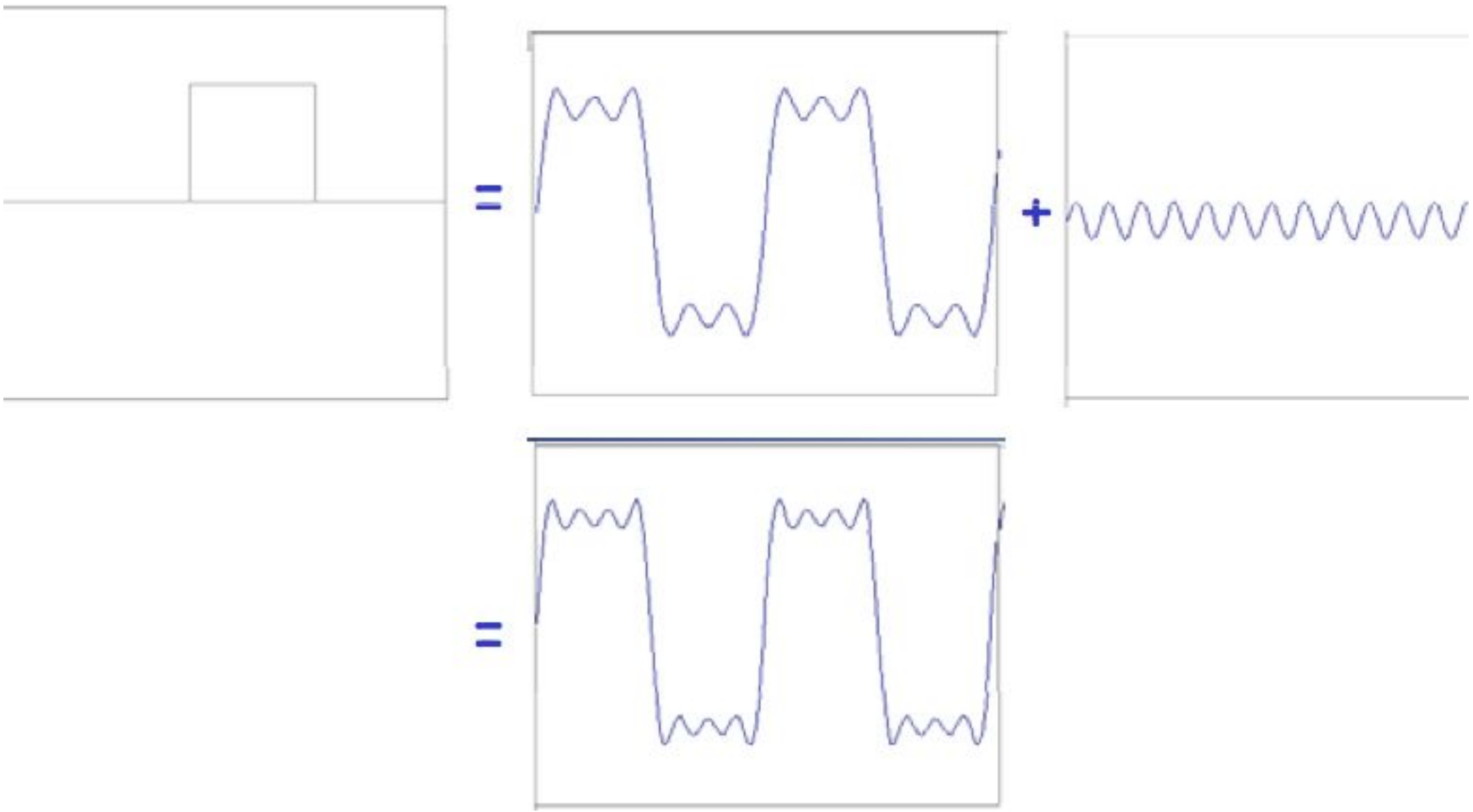
Прямоугольный сигнал



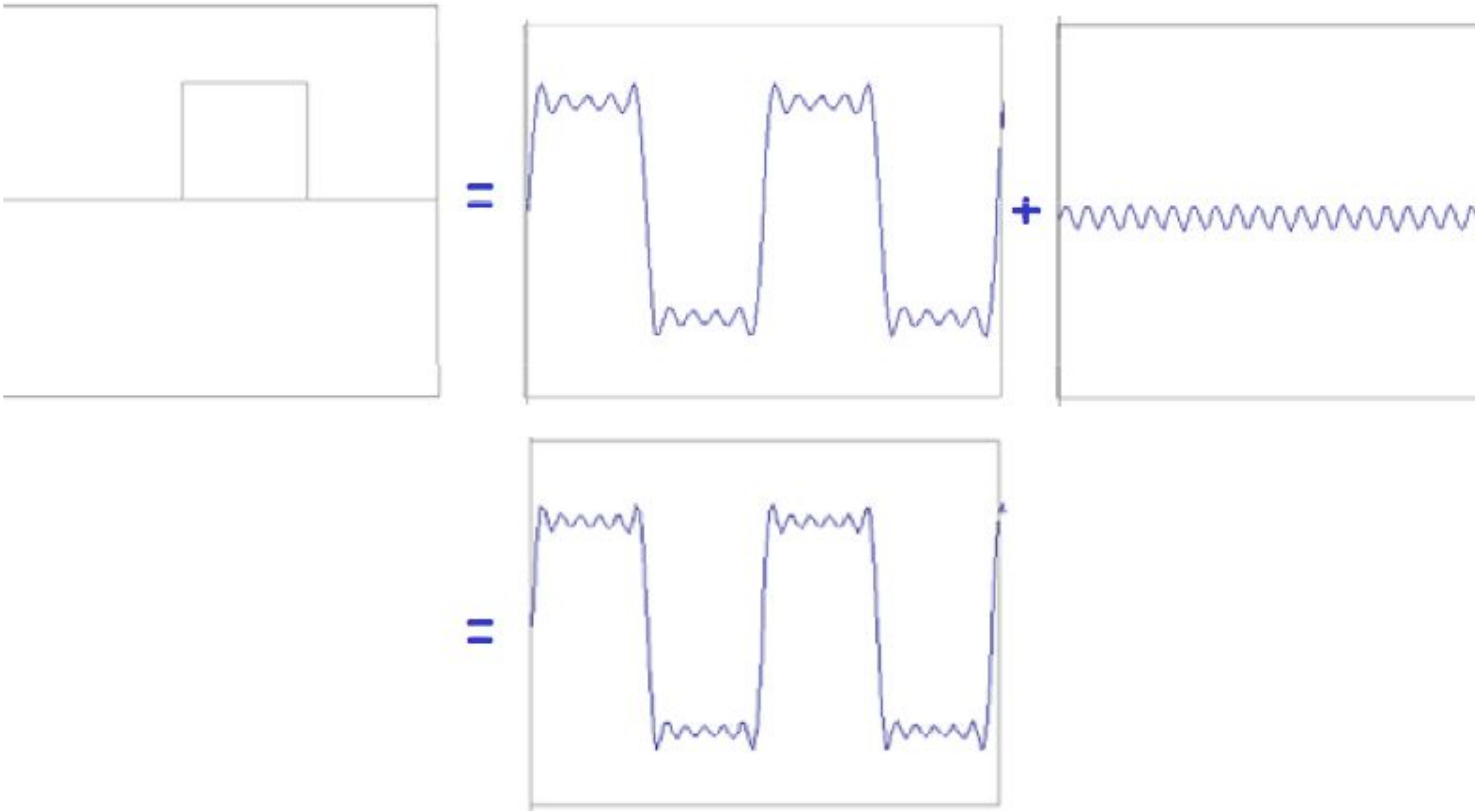
Прямоугольный сигнал



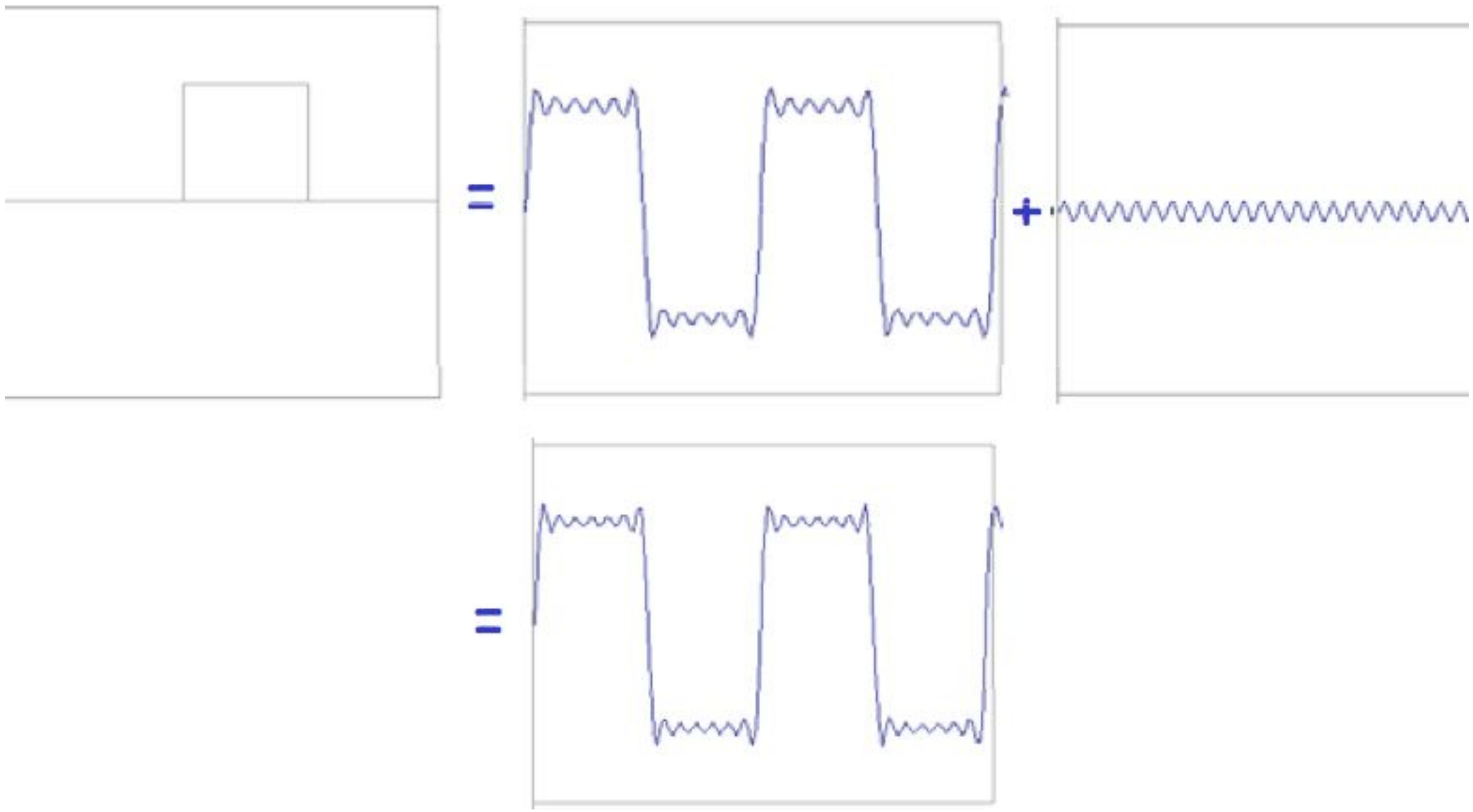
Прямоугольный сигнал



Прямоугольный сигнал

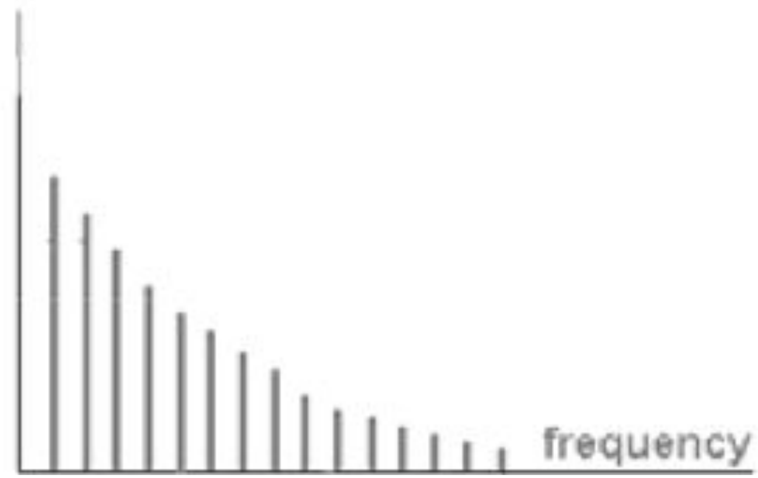


Прямоугольный сигнал

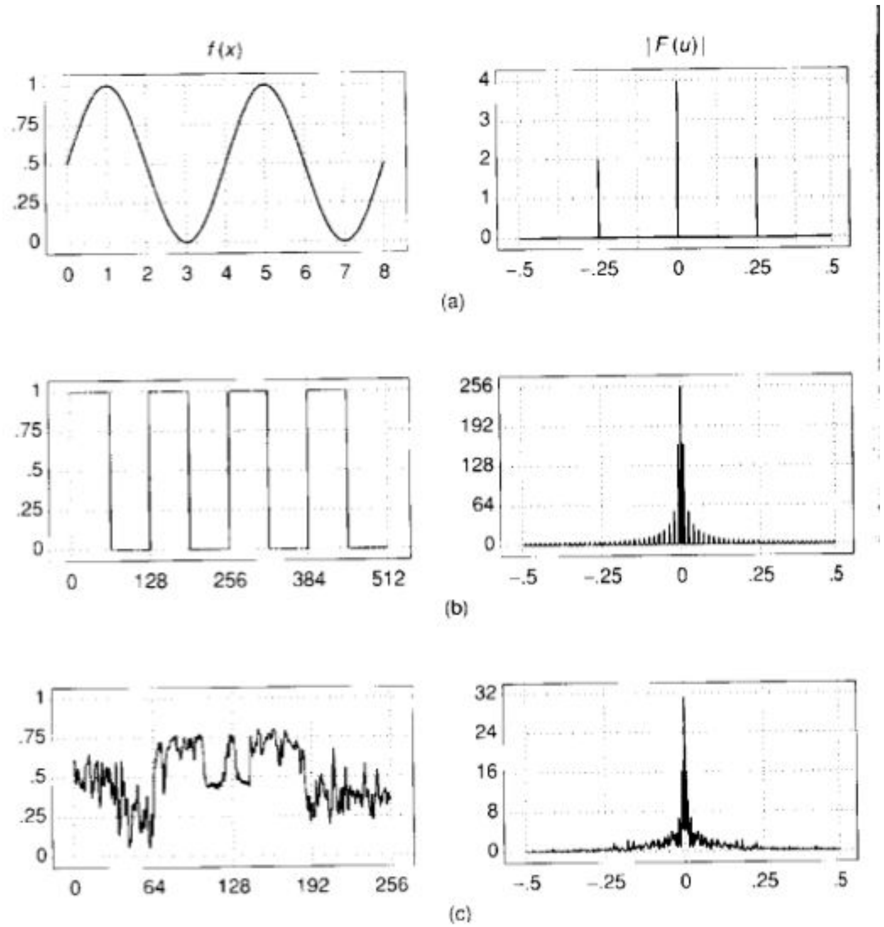


Прямоугольный сигнал

$$= \boxed{A\sum_{k=1}^{\infty}\frac{1}{k}\sin(2\pi kt)}$$

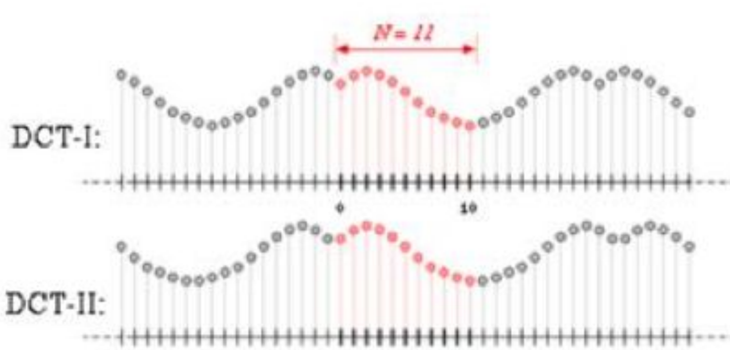


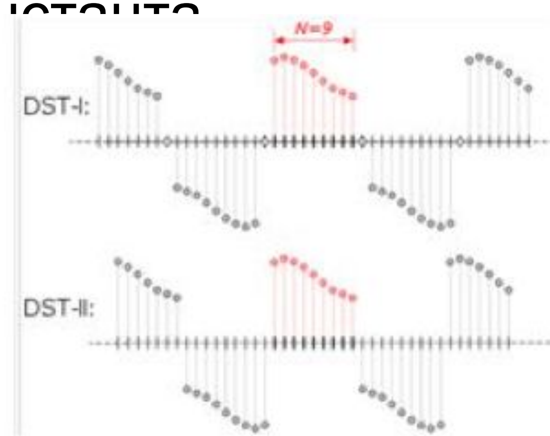
Спектр частот



Свойства

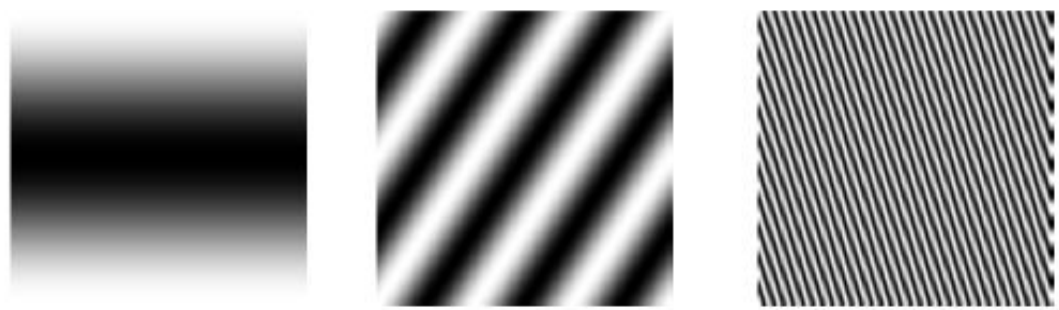
- Разрывы функции приводят к тому, что требуется больше слагаемых для достижения точности
- sin() – нечётная функция, поэтому продление должно быть нечётной функцией
- Поскольку у реального сигнала значение на конце и в начале сигнала обычно разное, то продление почти всегда с разрывом
- Для реальных сигналов разложение через косинусы эффективнее, чем через синусы
- Также в базисе косинусов есть константа





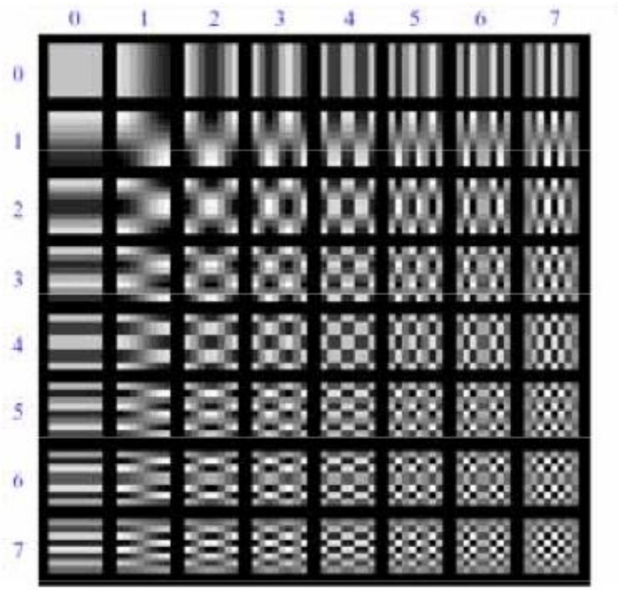
2D преобразование

- Базисные функции имеют вид двумерных синусоид с разными углами наклона и фазами

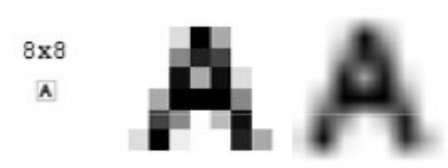


- Вычисление двумерного ДПФ (ДКП, ДСП)
    - Прямой способ – скалярные произведения со всеми базисными функциями. Очень много операций.
    - Быстрый способ – декомпозиция на одномерные ДПФ, затем быстрое преобразование Фурье

Пример



Базис для дискретного косинусного преобразования (ДКП)



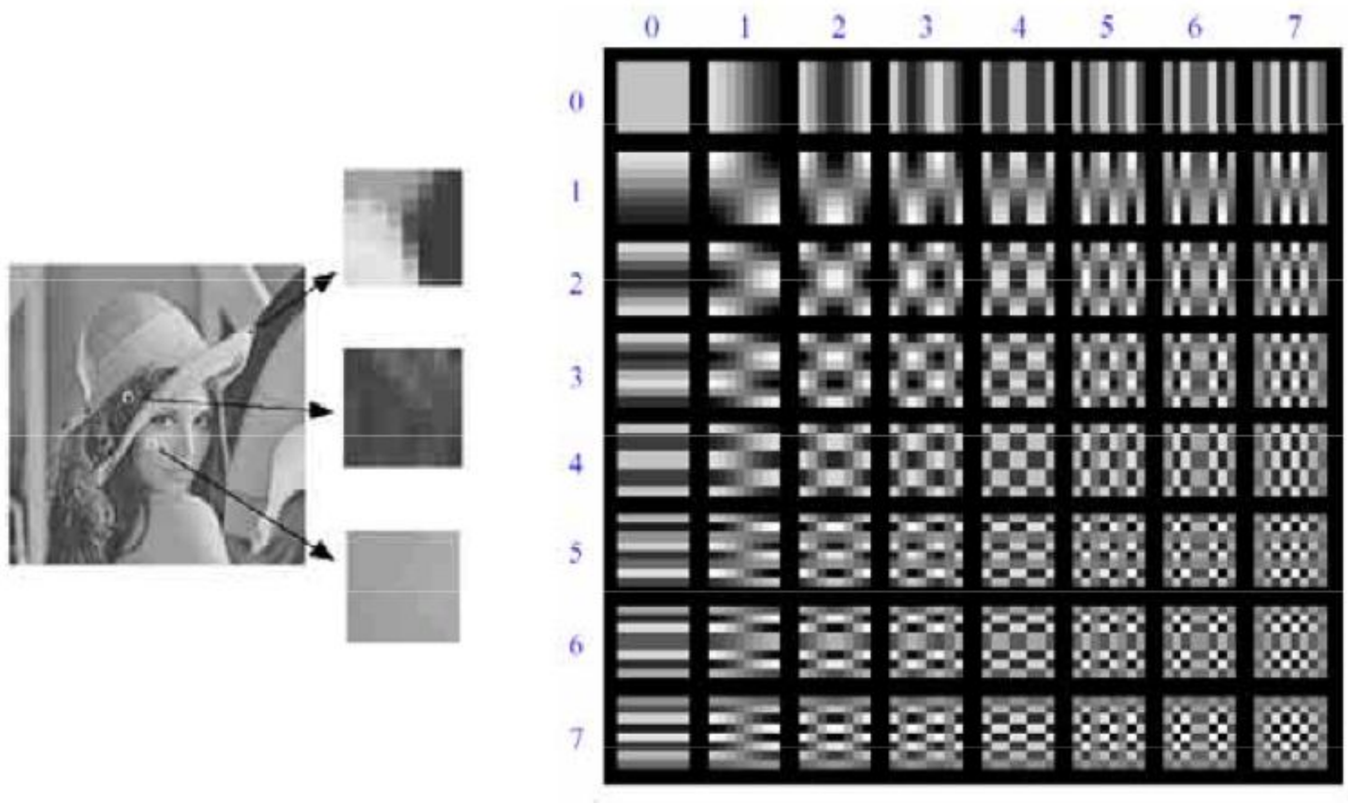
Буква А размером 8x8 пикселей

Реконструкция буквы после ДКП

- Поблочное Дискретное Косинусное Преобразование (ДКП)
- Discrete Cosine Transform (DCT)

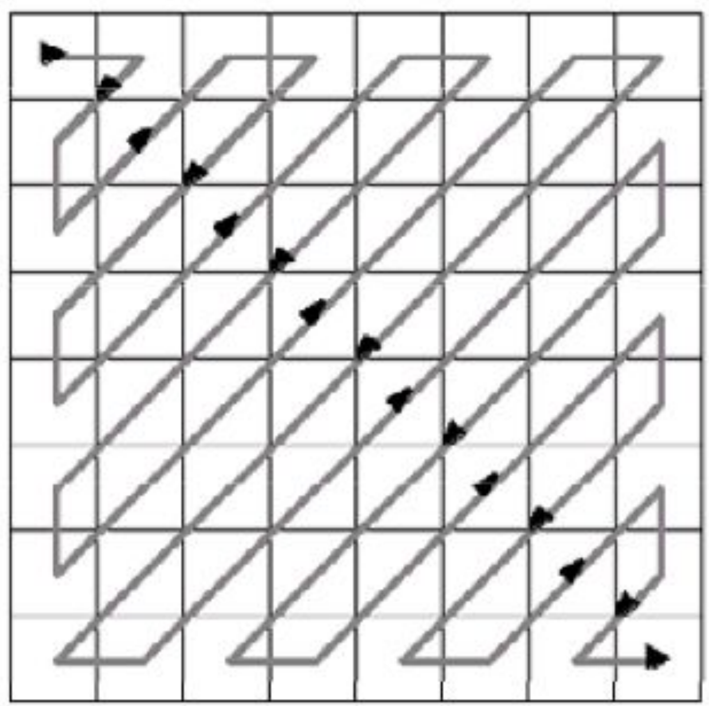
Пример

Сжатие с потерями (JPEG)



- Один из самых ярких примеров применения ДКП

- Первый коэффициент B(0,0) называется DC, средняя интенсивность
- Верхние левые коэффициенты соответствуют низким частотам, верхние – высоким частотам



Сжатие изображения с ДКП

- Следующим шагом является квантование (дискретизация) коэффициентов
- Квантовать мы можем по разному низкие (важные) и высокие (менее важные) частоты
- Именно при квантовании происходит потеря информации
- В декодере проводится обратное преобразование
- Матрица квантования хранится в заголовке файла

3	5	7	9	11	13	15	17
5	7	9	11	13	15	17	19
7	9	11	13	15	17	19	21
9	11	13	15	17	19	21	23
11	13	15	17	19	21	23	25
13	15	17	19	21	23	25	27
15	17	19	21	23	25	27	29
17	19	21	23	25	27	29	31

Пример

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

Блок

$$g = \begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix}$$

Сдвиг среднего

$$G = \begin{bmatrix} -415.38 & -30.19 & -61.20 & 27.24 & 56.13 & -20.10 & -2.39 & 0.46 \\ 4.47 & -21.86 & -60.76 & 10.25 & 13.15 & -7.09 & -8.54 & 4.88 \\ -46.83 & 7.37 & 77.13 & -24.56 & -28.91 & 9.93 & 5.42 & -5.65 \\ -48.53 & 12.07 & 34.10 & -14.76 & -10.24 & 6.30 & 1.83 & 1.95 \\ 12.12 & -6.55 & -13.20 & -3.95 & -1.88 & 1.75 & -2.79 & 3.14 \\ -7.73 & 2.91 & 2.38 & -5.94 & -2.38 & 0.94 & 4.30 & 1.85 \\ -1.03 & 0.18 & 0.42 & -2.42 & -0.88 & -3.02 & 4.12 & -0.66 \\ -0.17 & 0.14 & -1.07 & -4.19 & -1.17 & -0.10 & 0.50 & 1.68 \end{bmatrix}$$

Результат ДКП

Пример

$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \qquad B = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -3 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- Делим G на Q и округляем:
  - *round (G(i,j) / Q(i,j))*
- При этом обнуляются высокие частоты
- Значения Q позволяют менять степень сжатия
- Значения обходятся зигзагом и кодируются без потерь (RLE или арифметическое)

Размер блока JPEG

- Маленький блок
  - Быстрее
  - Больше корреляции между соседними пикселям
- Большой блок
  - Лучше сжатие в плавных регионах
- По стандарту 8x8

Пример сжатия

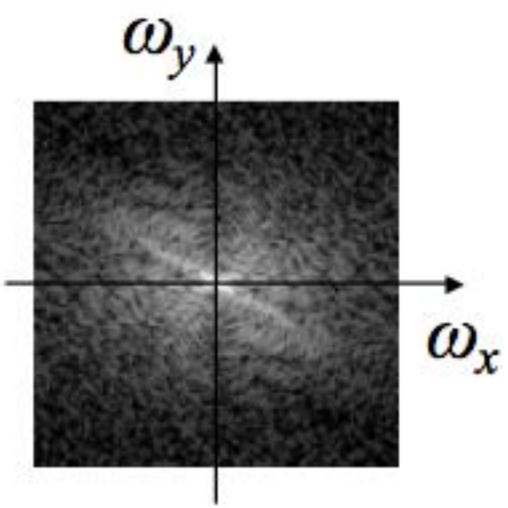
89k

12k

Спектральный анализ для изображений

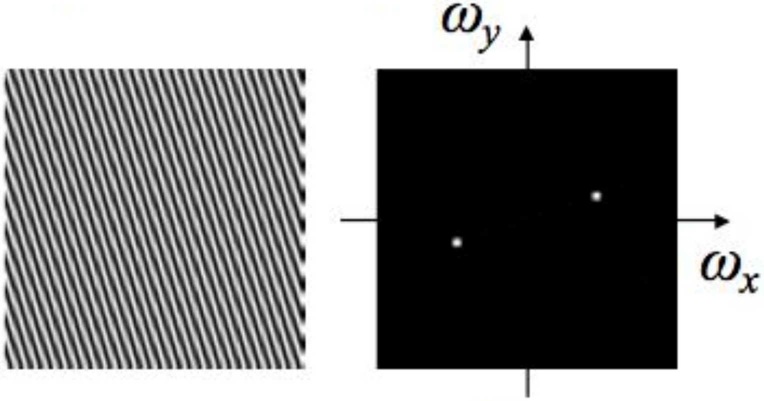
Отображение спектров изображений

- Спектр – это изображение, показывающая зависимость амплитуды от частоты и от направления синусоиды.
- Амплитуды отображаются в виде яркостей.
- Нулевая частота – в центре спектра, низкие частоты вокруг центра, высокие – дальше от центра.
- Спектр обычно продублирован отражением от нулевой частоты.
- В реальных изображениях чаще всего гораздо большие амплитуды имеют низкие частоты (и постоянная составляющая). Поэтому постоянную составляющую иногда удаляют, или применяют логарифмический масштаб отображения амплитуд, чтобы пара самый мощных гармоник не скрыла остальные, менее мощные, но тоже существенные гармоники.

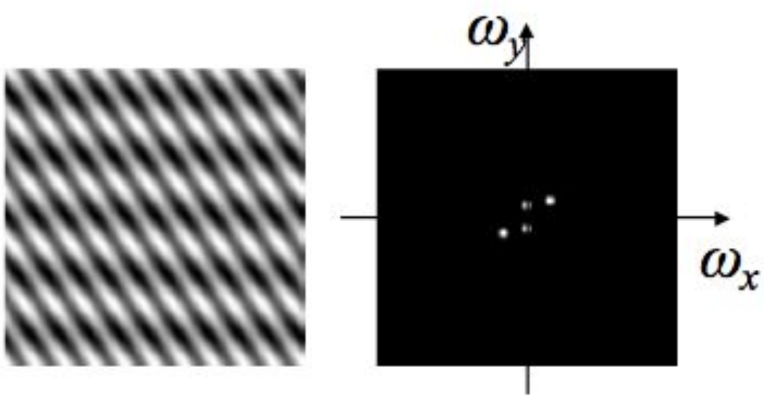


Спектральный анализ

- Примеры изображений и их спектров



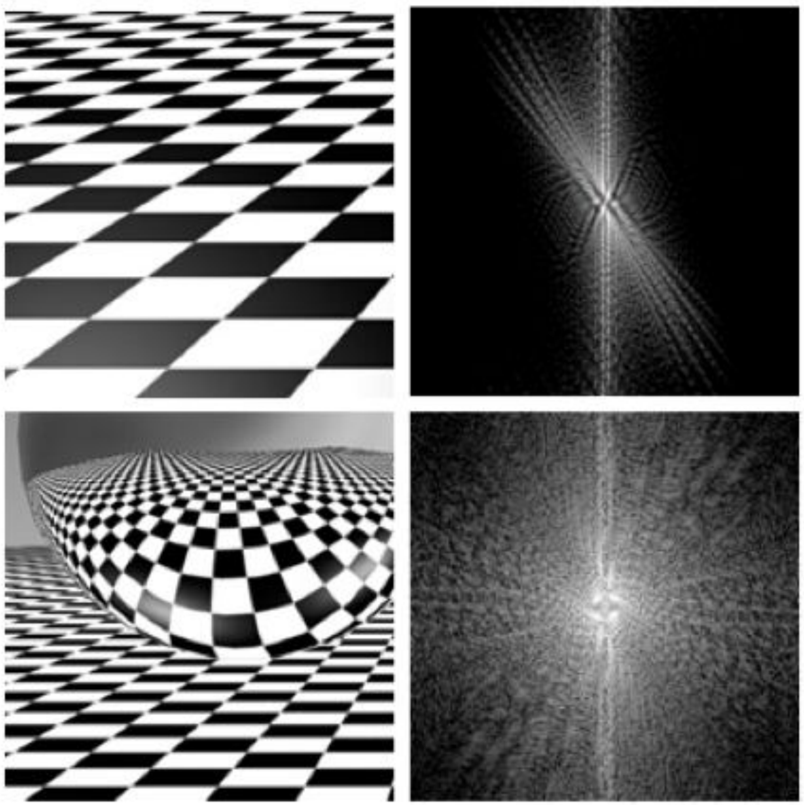
Видно, что спектр одной синусоиды – это точка
(не забываем про симметричное отражение спектра)



Две синусоиды – две точки

Спектральный анализ

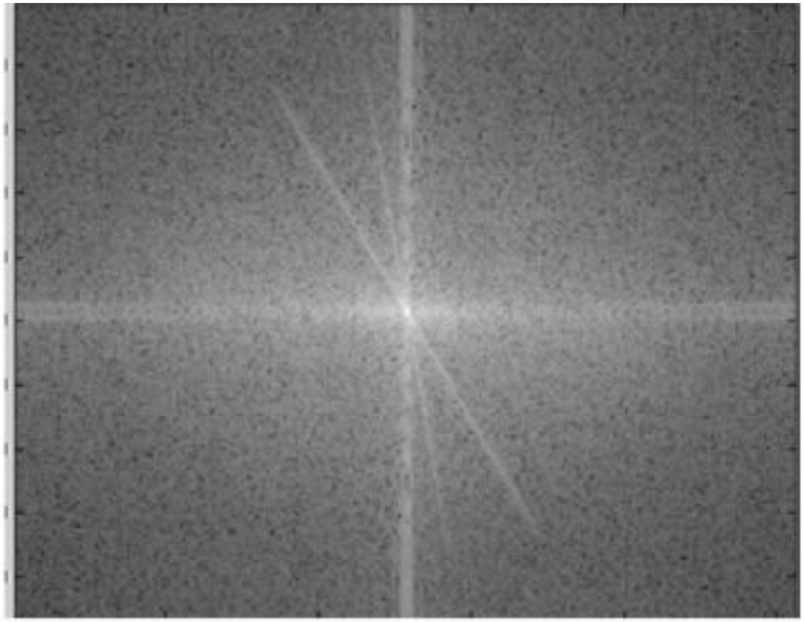
- Примеры изображений и их спектров



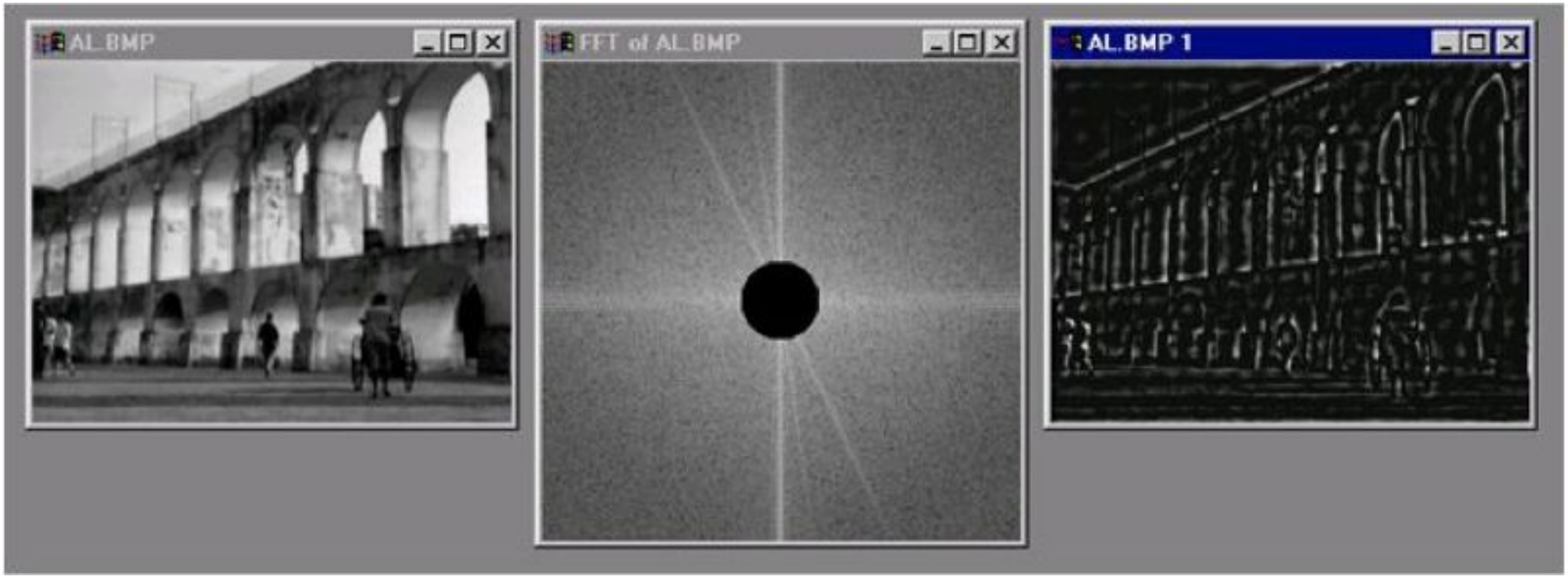
По спектру прослеживаются преобладающие направления в исходной картинке

Много высоких частот в спектре – много мелких деталей в исходном изображении

Искусственная сцена



Края в изображении



Теорема о свёртке

- Преобразование Фурье от свёртки двух функций можно представить как произведение преобразований Фурье каждой из функций

$$\mathrm{F}[g*h] = \mathrm{F}[g]\mathrm{F}[h]$$

- Обратное преобразование Фурье от произведения есть свёртка двух обратных преобразований Фурье

$$\mathrm{F}^{-1}[gh] = \mathrm{F}^{-1}[g]*\mathrm{F}^{-1}[h]$$

- Свёртка в пространстве эквивалентна произведению в частотном диапазоне
- Можно существенно ускорить многие операции свёртки!

Резюме

- Сегментация изображения позволяет работать не со всем изображением в целом, а с отдельными областями
- В отдельных случаях мы можем решить задачу распознавания, анализируя геометрические и фотометрические признаки сегментов
- Сегменты могут быть однородны по яркости, цвету, текстуре и по комбинации этих признаков
- Переход от представления в виде регулярной сетки к частотному представлению позволяет учесть структуру изображения
  - Сжатие изображений по алгоритму JPEG
  - Использование теоремы о свёртке позволяет эффективнее фильтровать изображение
  - Фильтр Гаусса – фильтр низких частот