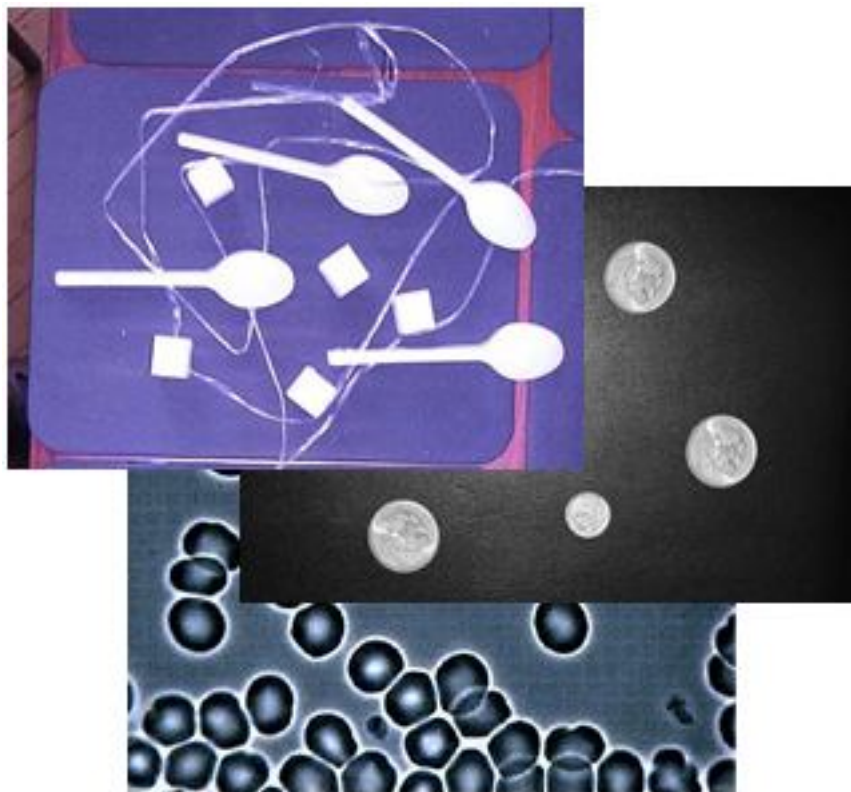
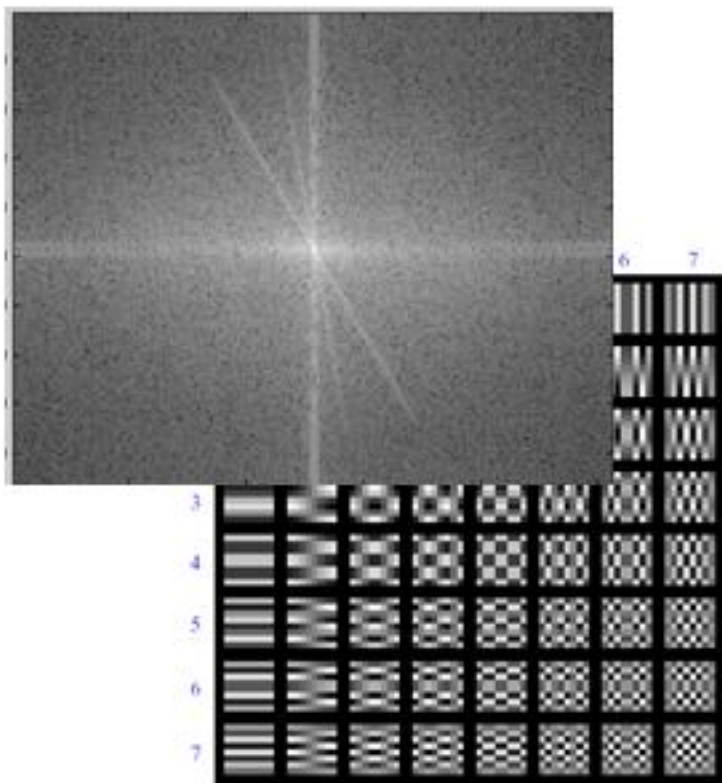




Обработка изображений, ч. 2

Яндекс



АНТОН КОНУШИН

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



План лекции

- Частотная фильтрация изображений
 - DCT-разложение, JPEG
 - Теорема о свёртке
- Сегментация изображений
 - Простейшие методы сегментации
 - Текстура
 - Использование сегментации для анализа изображений



План лекции

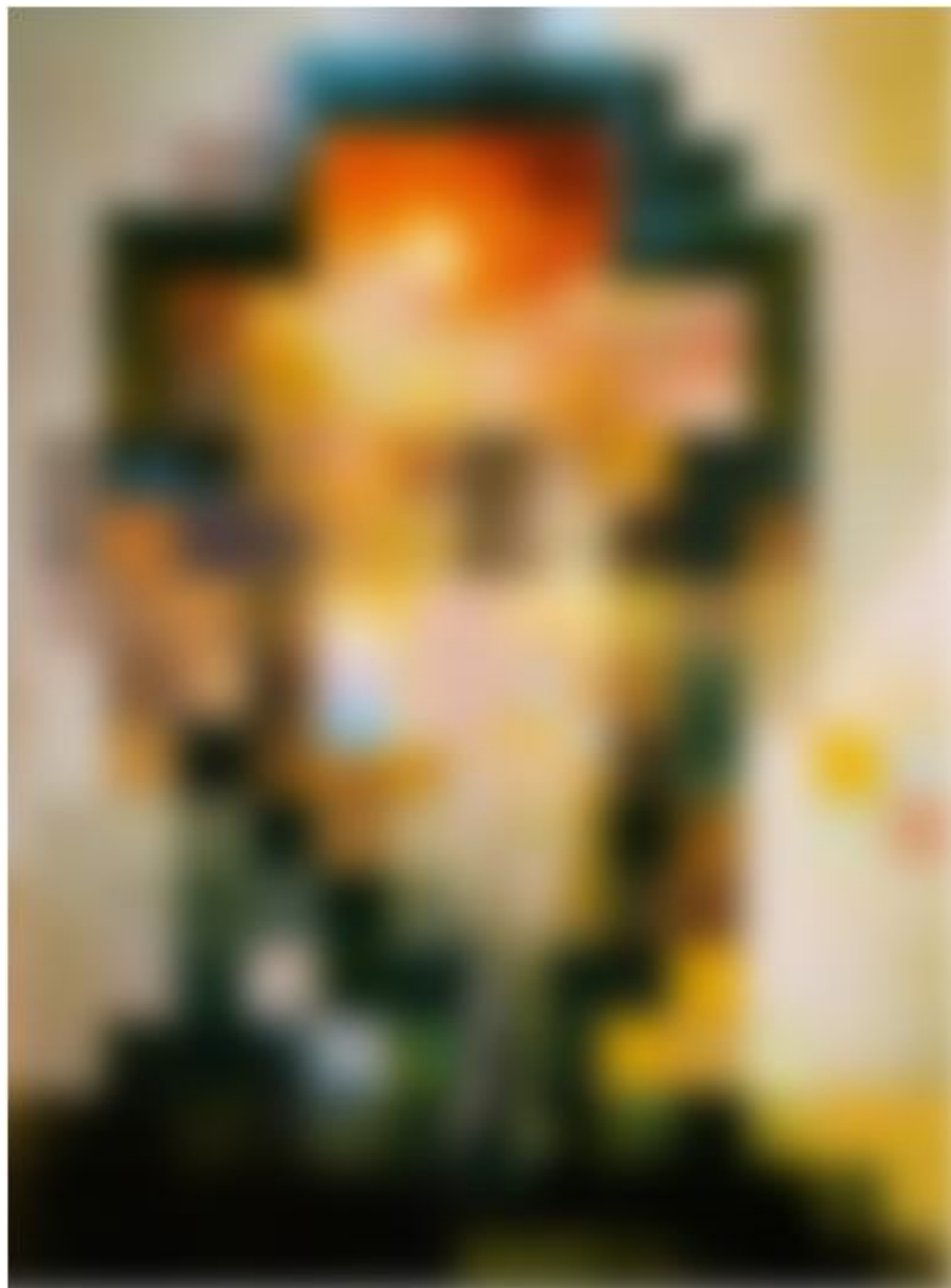
- Частотная фильтрация изображений
 - DCT-разложение, JPEG
 - Теорема о свёртке
- Сегментация изображений
 - Простейшие методы сегментации
 - Текстура
 - Использование сегментации для анализа изображений

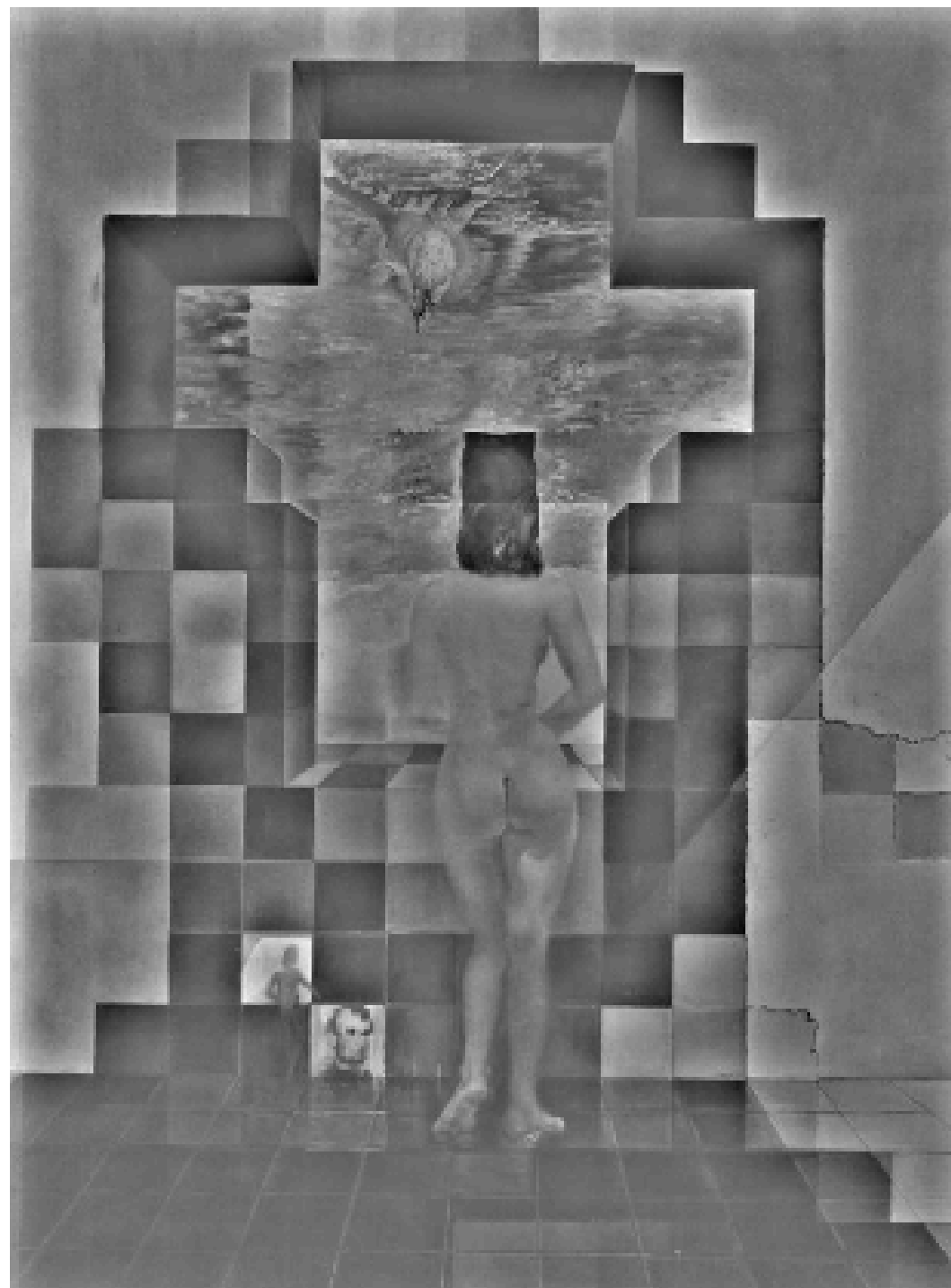


Salvador Dali

*"Gala Contemplating the Mediterranean Sea,
which at 30 meters becomes the portrait
of Abraham Lincoln", 1976*









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





Ряд Фурье

Идея: любая периодическая функция $f(x)$ с периодом T может быть представлена в виде ряда:

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{+\infty} A_k \cos\left(2\pi \frac{k}{T}x + \theta_k\right)$$

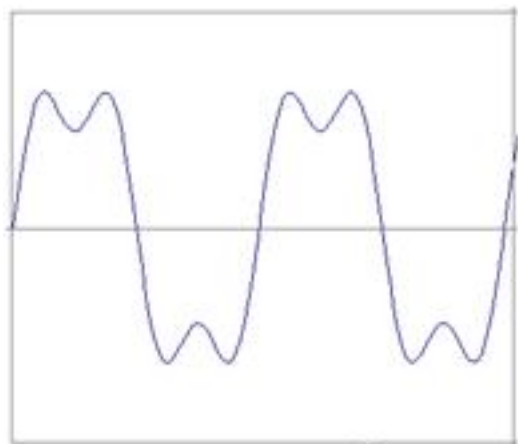
где $\cos(k)$ – k -ая гармоника, A_k – амплитуда k -ой гармоника, θ_k – фаза k -ой гармоника



Пример

Яндекс

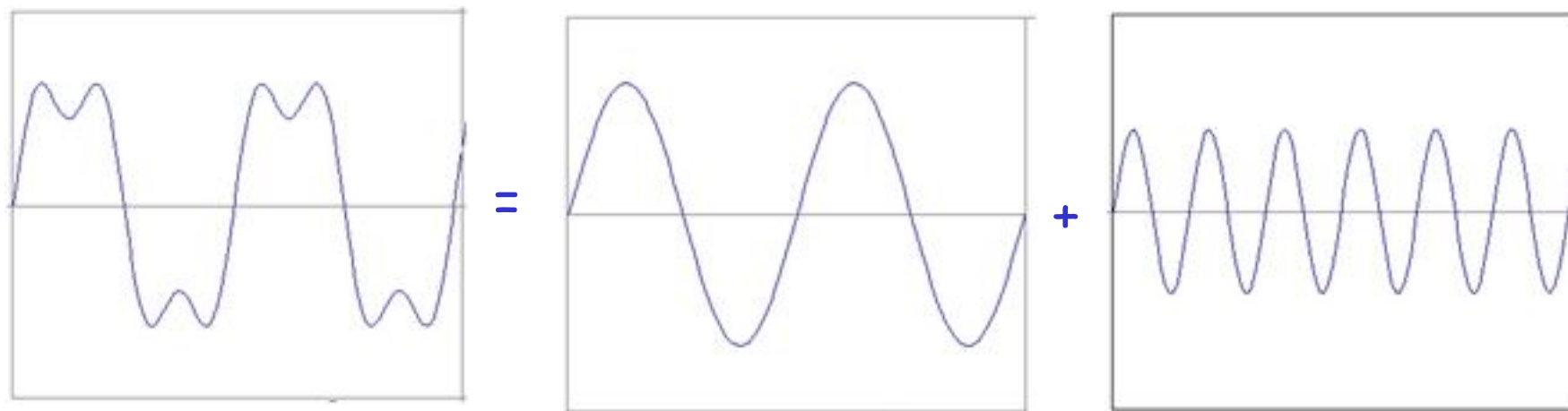
$$g(t) = \sin(2\pi f t) + (1/3)\sin(2\pi(3f) t)$$





Пример

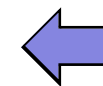
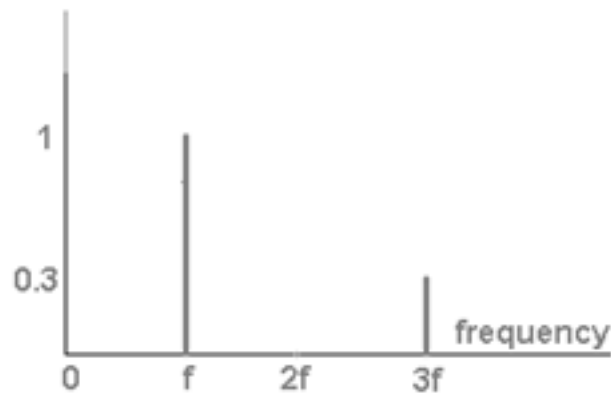
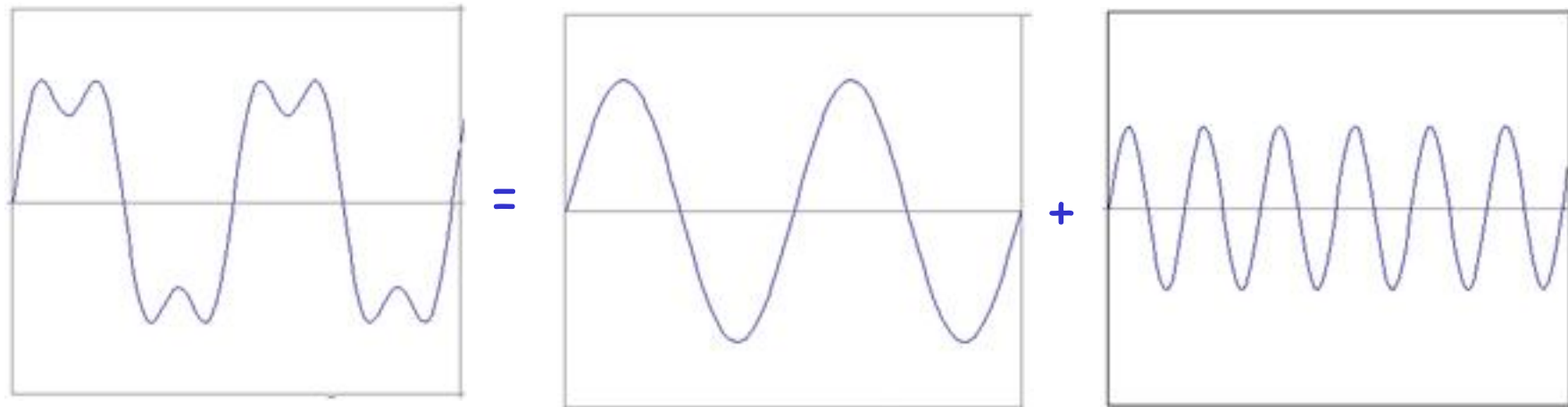
$$g(t) = \sin(2\pi f t) + (1/3)\sin(2\pi(3f) t)$$





Частотный спектр

$$g(t) = \sin(2\pi f t) + (1/3)\sin(2\pi(3f) t)$$

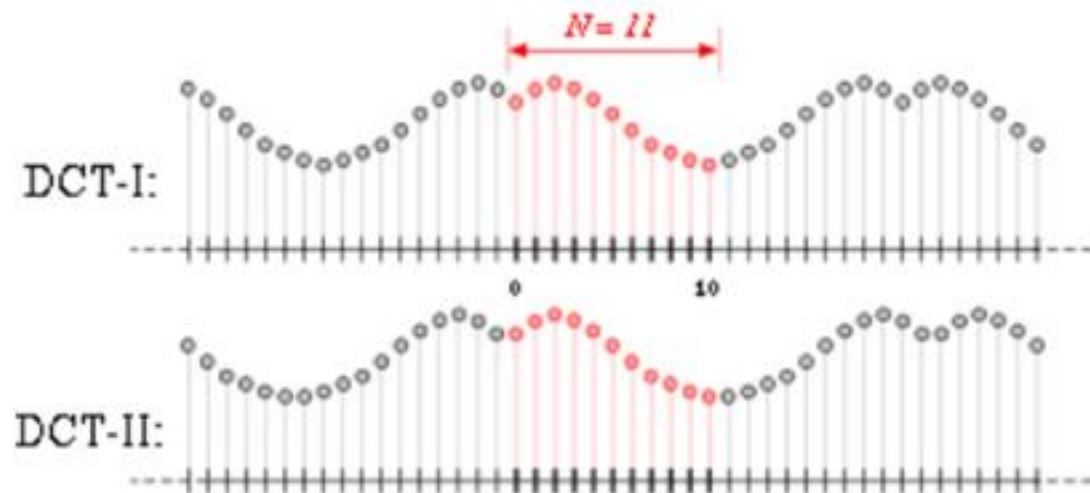


Спектр – график зависимости амплитуды от частоты и фазы.



Ограниченный сигнал

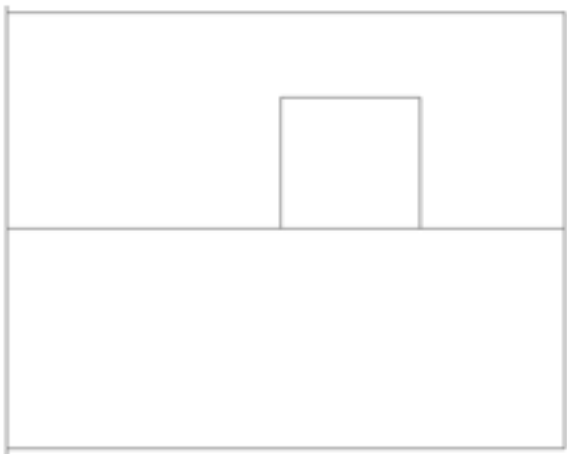
- Как быть, если сигнал задан на отрезке?
 - Продлить сигнал за границы отрезка, затем разложить
 - Продление должно быть периодическим
 - Можем использовать только синусы или только косинусы, в зависимости от этого продлевать нужно по-разному
 - Если косинусное преобразование, то продление должно быть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





Прямоугольный сигнал

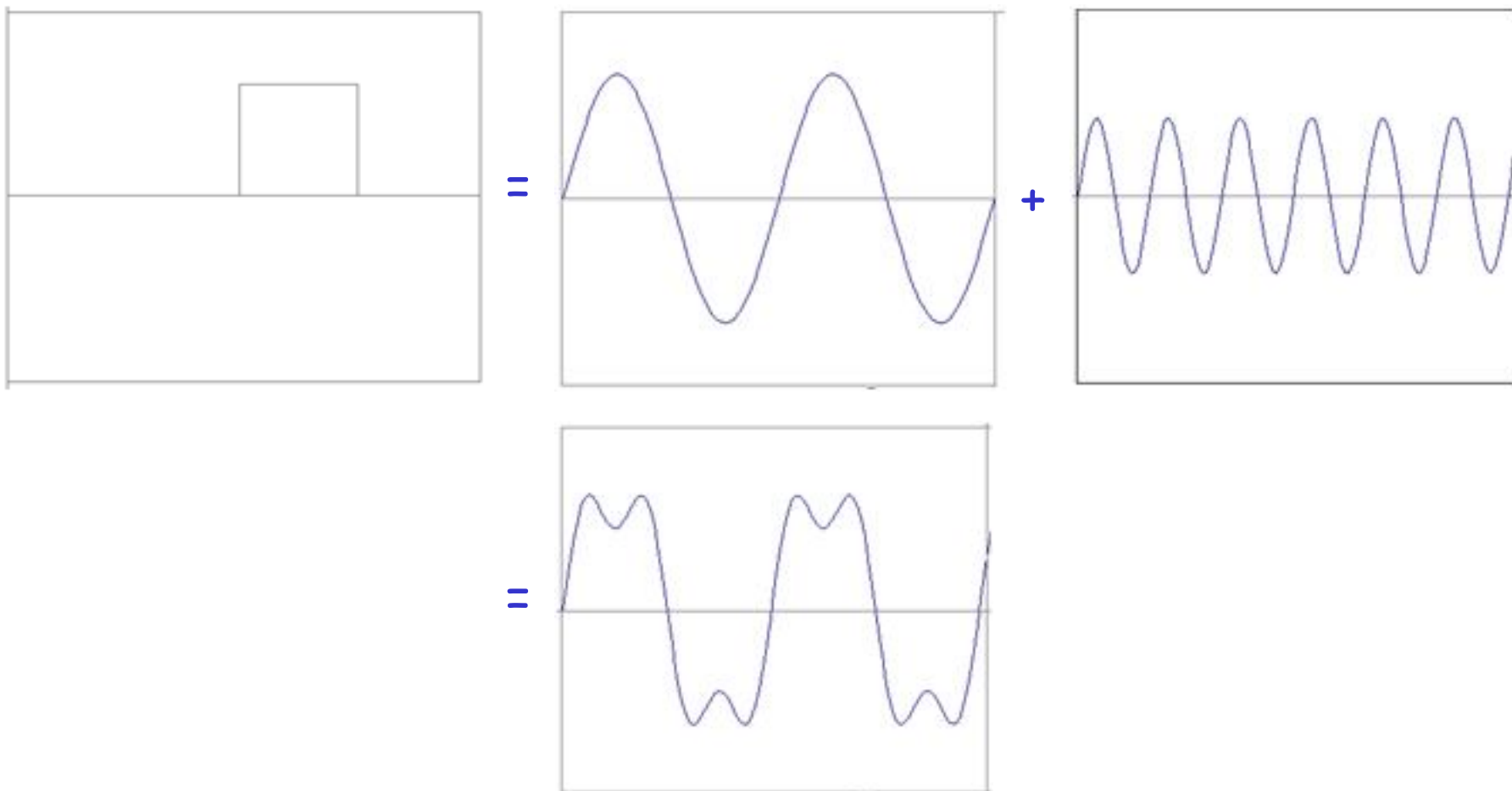
Яндекс





Прямоугольный сигнал

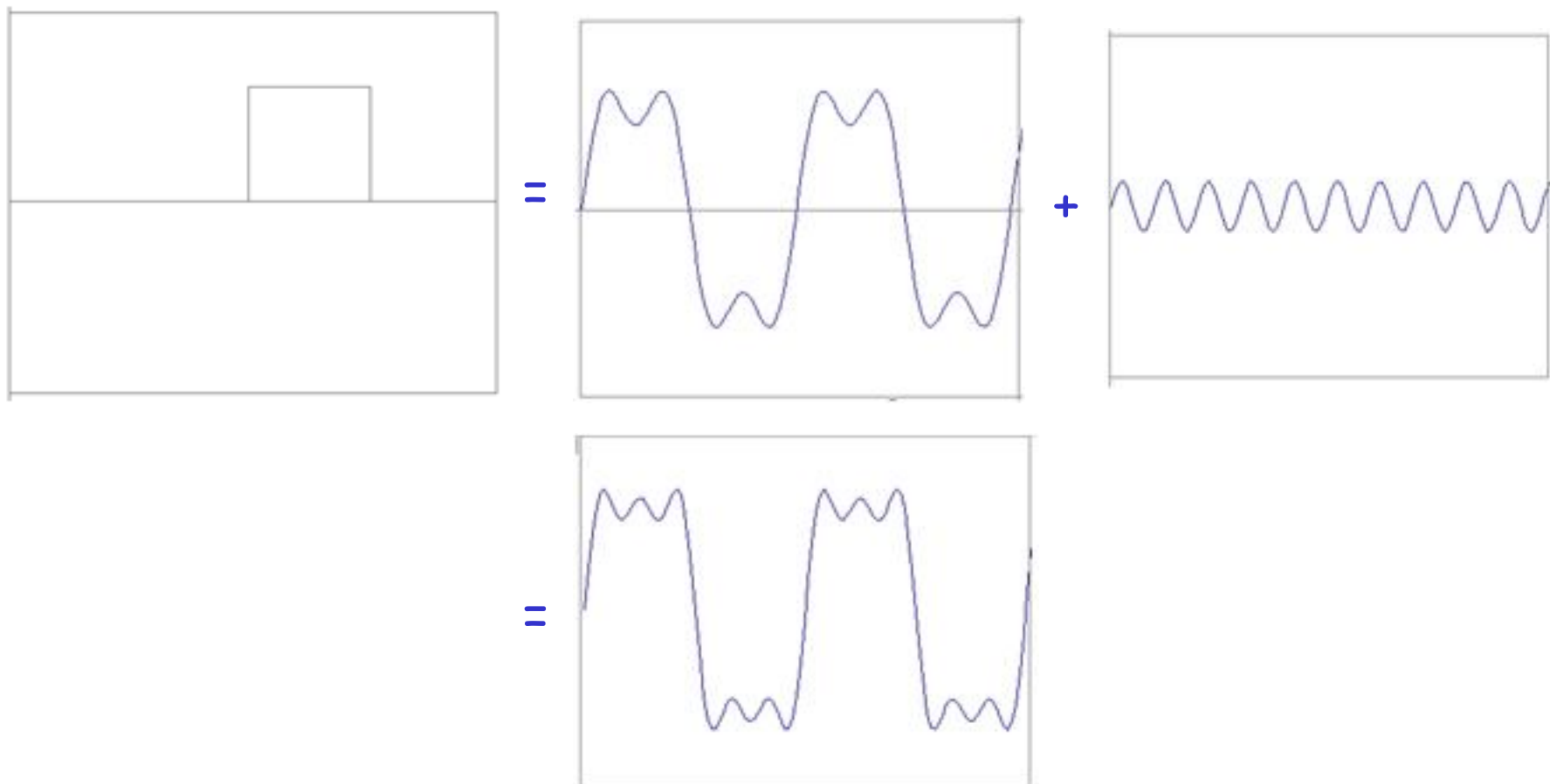
Яндекс





Прямоугольный сигнал

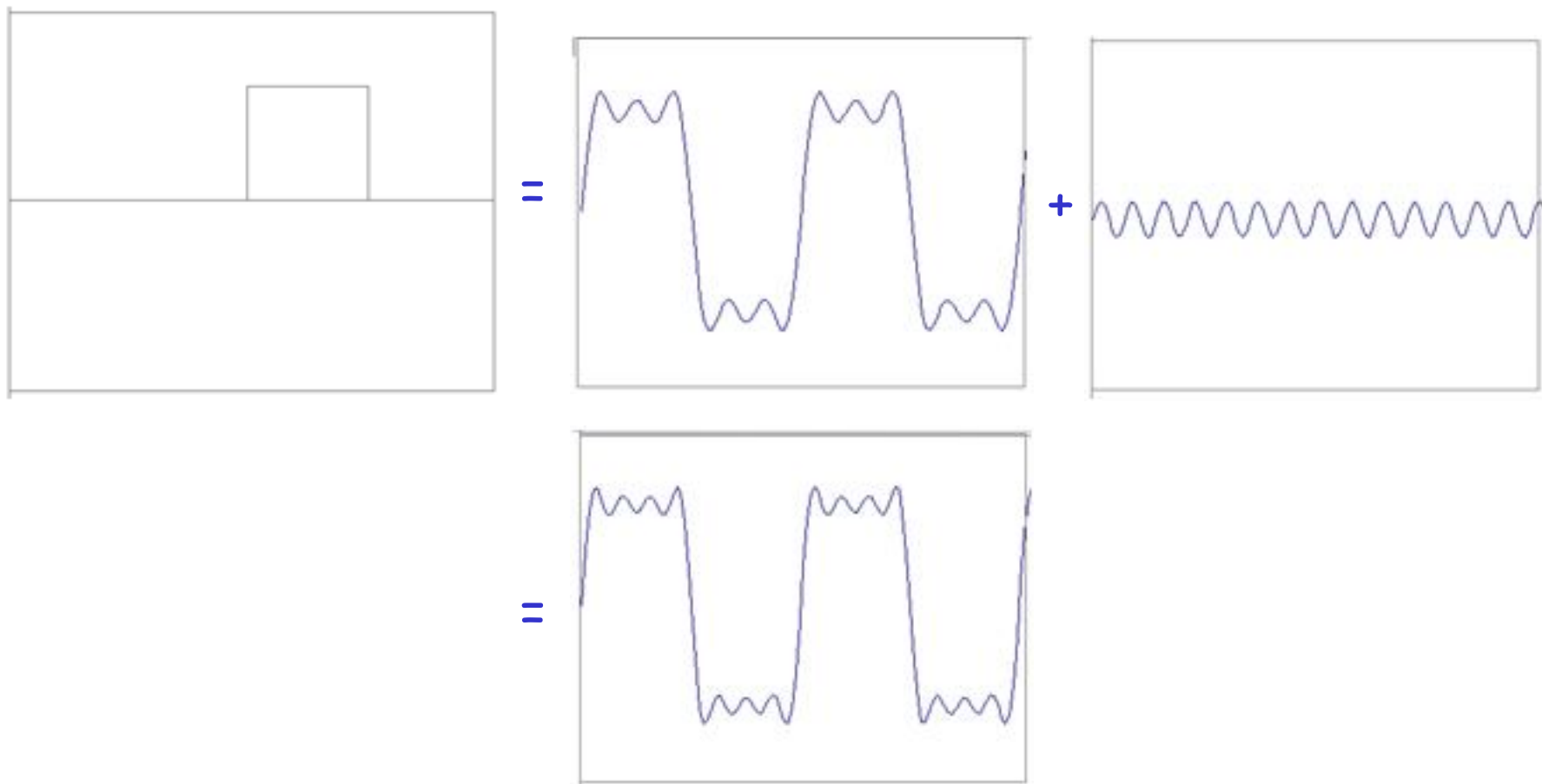
Яндекс





Прямоугольный сигнал

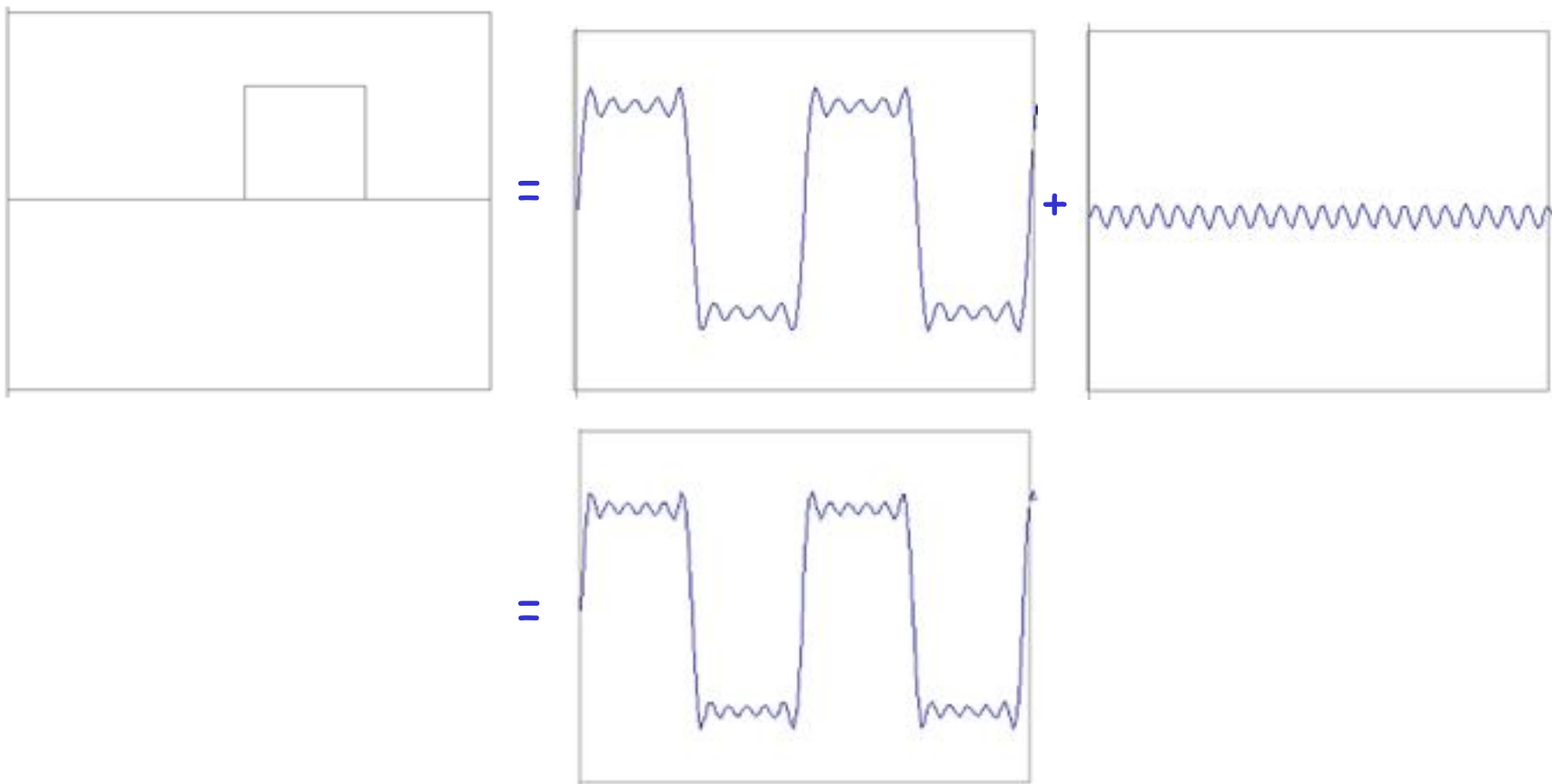
Яндекс





Прямоугольный сигнал

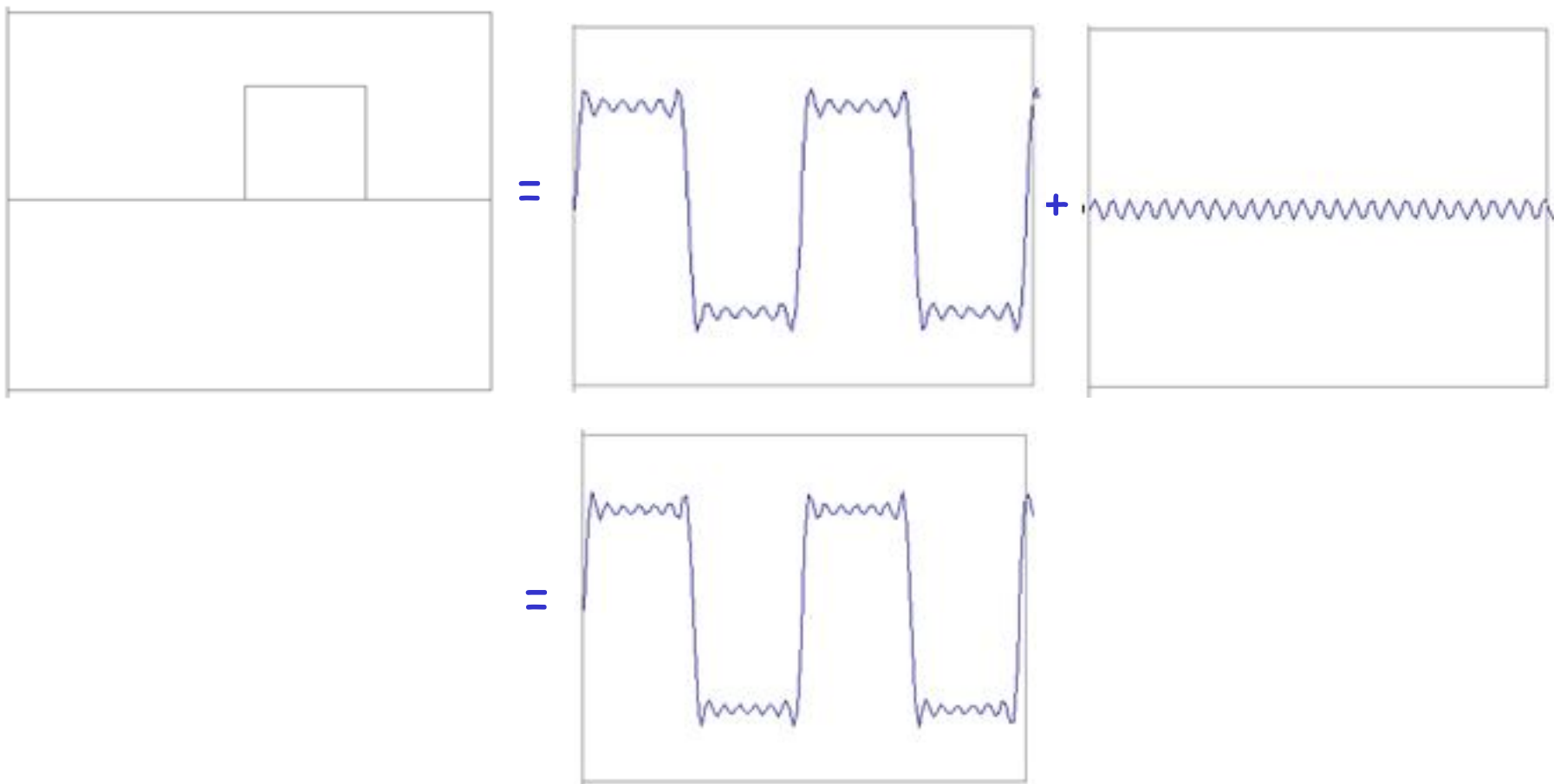
Яндекс





Прямоугольный сигнал

Яндекс



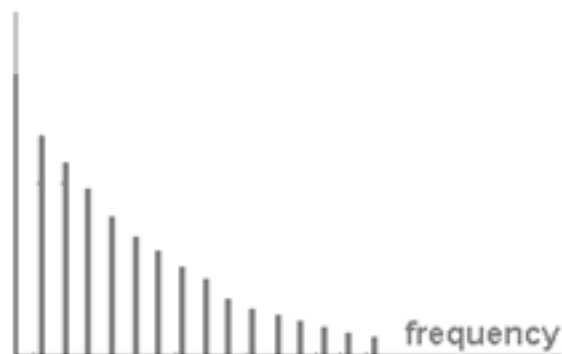


Прямоугольный сигнал



=

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





Ringing

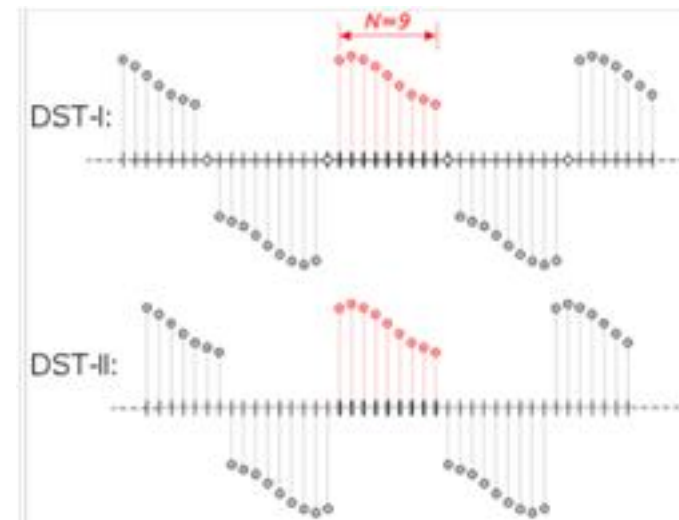
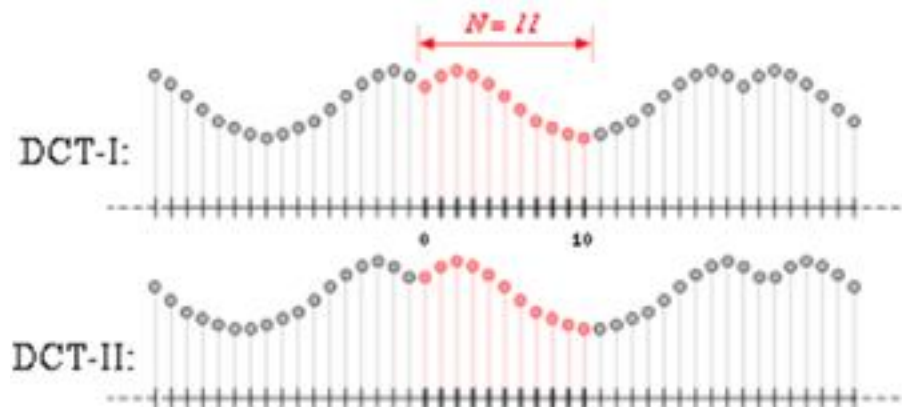


- [Gibb's Phenomenon](#)
- Частичная сумма ряда Фурье отличается от сигнала на резких границах
- Постоянно с этим борются



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





Дискретное преобразование Фурье

Яндекс

- Для дискретных сигналов длиной N можно ввести прямое дискретное преобразование Фурье:



Для каждой ω от 0 до $N-1$, $F(\omega)$ содержит амплитуду A и фазу ϕ соответствующего синуса или косинуса

- Для удобной записи используются мнимые числа:

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

$$A = \pm \sqrt{R(\omega)^2 + I(\omega)^2}$$

$$\phi = \tan^{-1} \frac{I(\omega)}{R(\omega)}$$



Преобразование Фурье

- Разложение Фурье обратимо, т.е. по коэффициентам разложения можно точно восстановить исходный дискретный сигнал.
- Обратное преобразование Фурье:





Вычисление ДПФ

- ДПФ является линейным преобразованием
- Базисные функции (косинусы или синусы) образуют N-мерный ортогональный базис в пространстве N-мерных векторов исходных сигналов.
- Весовые коэффициенты вычисляются как скалярное произведение сигнала на базисные функции

$$\text{ДПФ: } X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \quad k = 0, \dots, N-1$$

Запись в матричной
форме:

$$\vec{X} = \hat{A} \vec{x}$$

$$\hat{A} = \begin{pmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & e^{-\frac{2\pi i}{N}} & e^{-\frac{4\pi i}{N}} & e^{-\frac{6\pi i}{N}} & \dots & e^{-\frac{2\pi i}{N}(N-1)} \\ 1 & e^{-\frac{4\pi i}{N}} & e^{-\frac{8\pi i}{N}} & e^{-\frac{12\pi i}{N}} & \dots & e^{-\frac{2\pi i}{N}2(N-1)} \\ 1 & e^{-\frac{6\pi i}{N}} & e^{-\frac{12\pi i}{N}} & e^{-\frac{18\pi i}{N}} & \dots & e^{-\frac{2\pi i}{N}3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-\frac{2\pi i}{N}(N-1)} & e^{-\frac{2\pi i}{N}2(N-1)} & e^{-\frac{2\pi i}{N}3(N-1)} & \dots & e^{-\frac{2\pi i}{N}(N-1)^2} \end{pmatrix}$$

Формула вычисления
коэффициентов:

$$A(m, n) = \exp \left(-2\pi i \frac{(m-1)(n-1)}{N} \right)$$



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 N^2
 - ▶ Ограничение: большинство реализаций FFT принимают только массивы длиной $N = 2^m$
- Есть и быстрое обратное преобразование

J. W. Cooley and J. W. Tukey, “An algorithm for the machine calculation of complex Fourier series,” *Mathematics of Computation*, vol. 19, pp. 297–301, 1965.



Теорема о свёртке

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

$$F[g * h] = F[g]F[h]$$

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

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

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



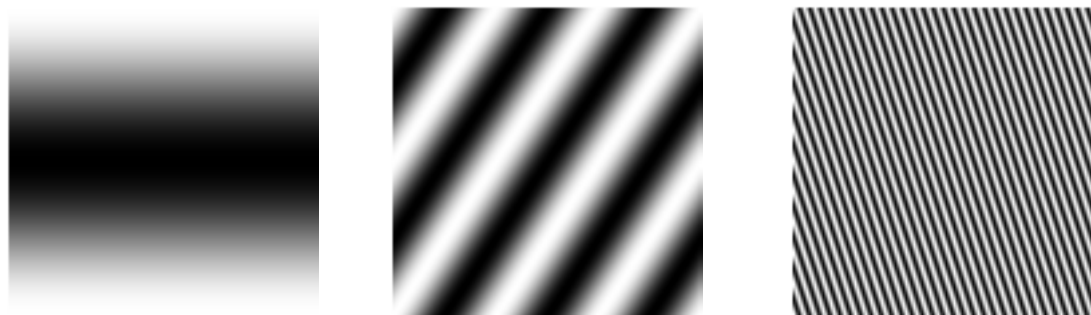
Быстрая свёртка

- Есть g (длиной N) и h (длиной M)
- Дополняем их нулями до $L = 2^k$ так, чтобы $L > N + M - 1$
- Вычисляем спектры для g и h с помощью быстрого преобразования Фурье
- Перемножаем спектры g и h как комплексные числа
- Применяем быстрое обратное преобразование Фурье
- Получаем сигнал длиной $N+M-1$, дополненный нулями до L , являющийся результатом свёртки



2D преобразование

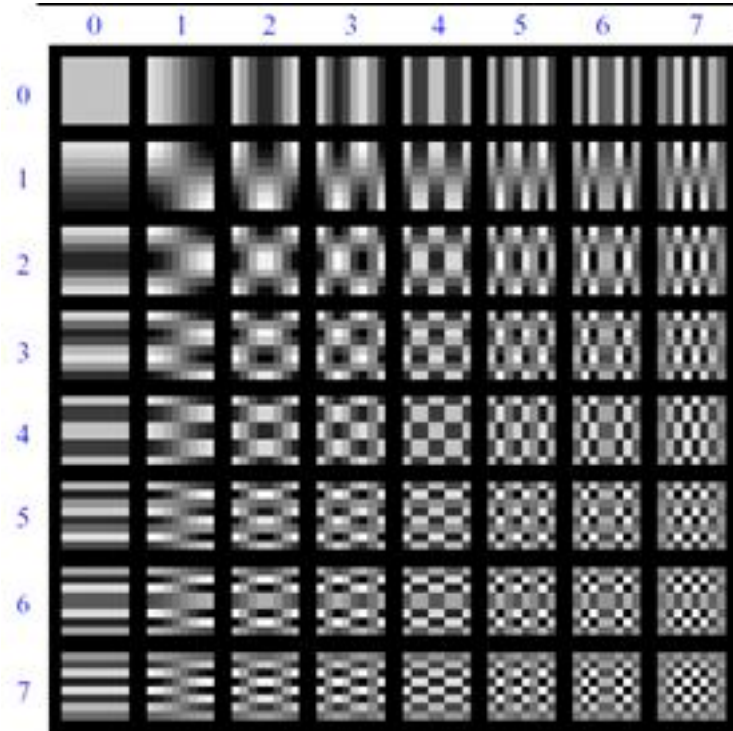
- Базисные функции имеют вид двумерных синусоид с разными углами наклона и фазами



- 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

+

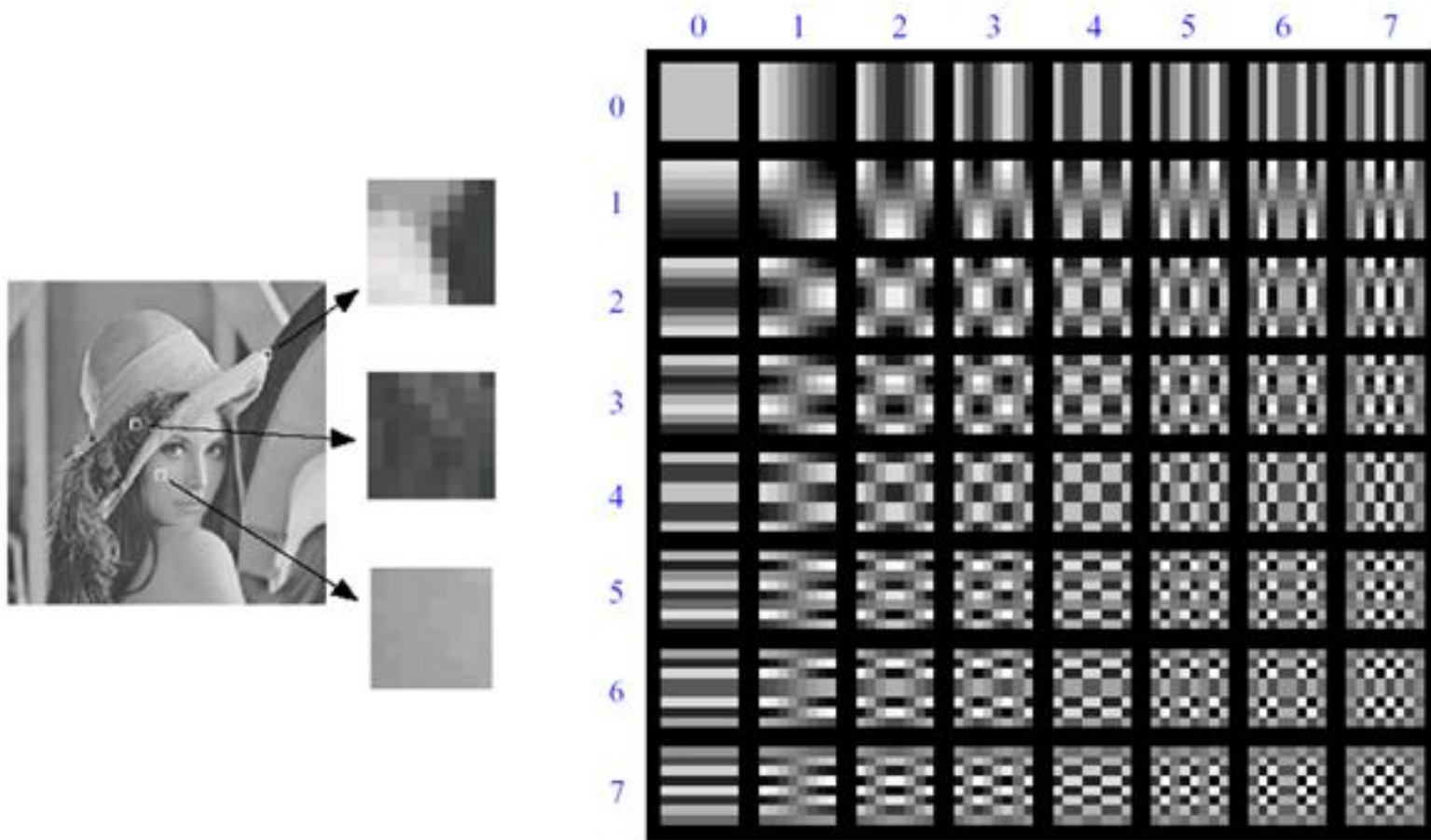
6.192 x

Реконструкция буквы после ДКП

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



Сжатие с потерями (JPEG)

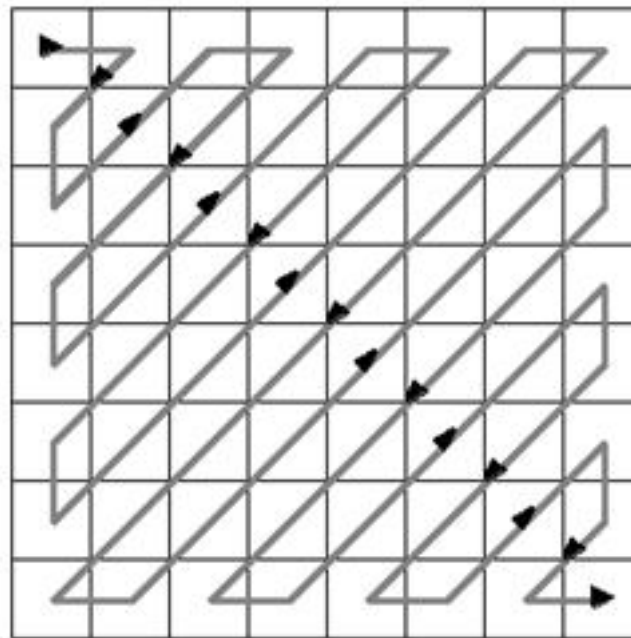


- Один из самых ярких примеров применения ДКП



Использование ДКП в JPEG

- 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:

Яндекс

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

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

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

$$B = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -3 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- Делим G на Q и округляем:
 - $\text{round} (G(i,j) / Q(i,j))$
- 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

Яндекс



89k

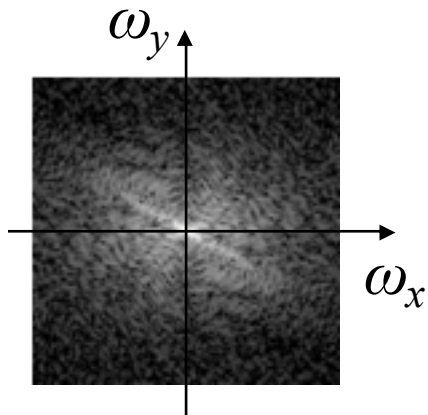


12k



Спектральный анализ для изображений

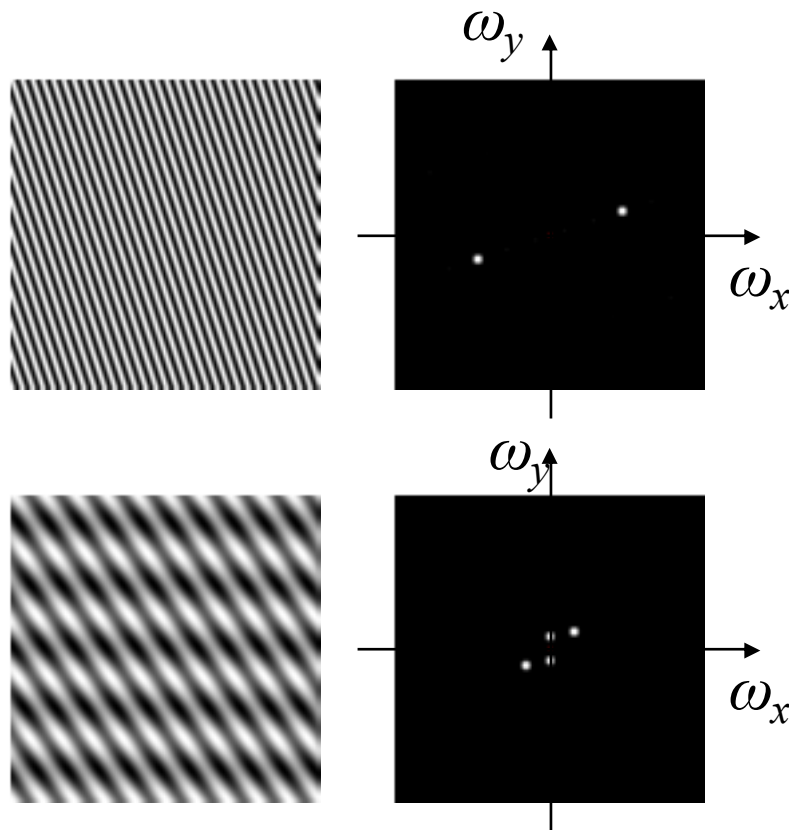
- Отображение спектров изображений
 - Спектр – это изображение, показывающая зависимость амплитуды от частоты и от направления синусоиды.
 - Амплитуды отображаются в виде яркостей.
 - Нулевая частота – в центре спектра, низкие частоты вокруг центра, высокие – дальше от центра.
 - Спектр обычно продублирован 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 самых мощных гармоник не скрыла остальные, менее мощные, но тоже существенные гармоники.





Спектральный анализ

- Примеры изображений и их спектров



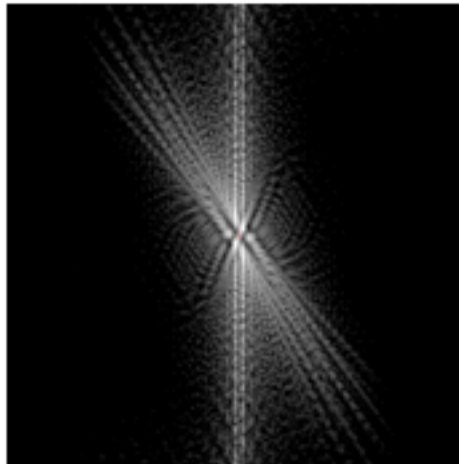
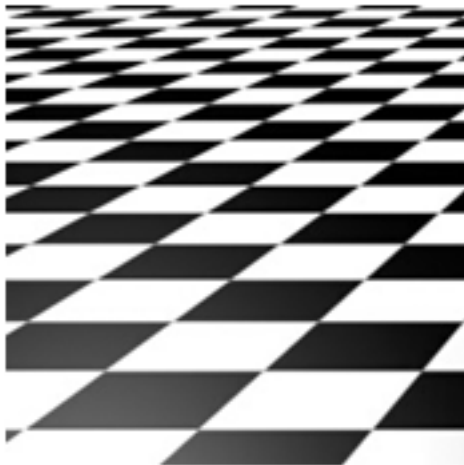
Видно, что спектр одной синусоиды – это точка (не забываем про симметричное отражение спектра)

Две синусоиды – две точки

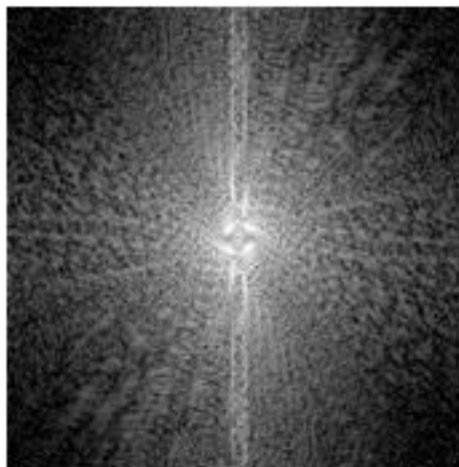
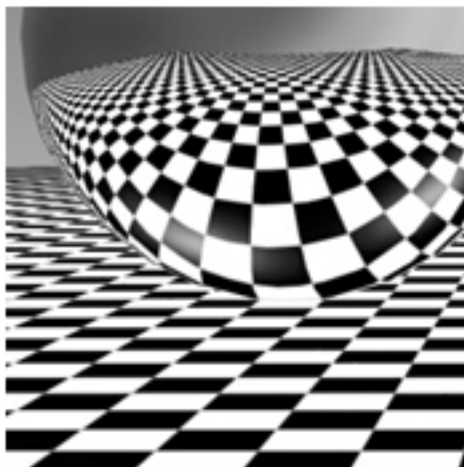


Спектральный анализ

- Примеры изображений и их спектров



По спектру
прослеживаются
преобладающие
направления в исходной
картинке

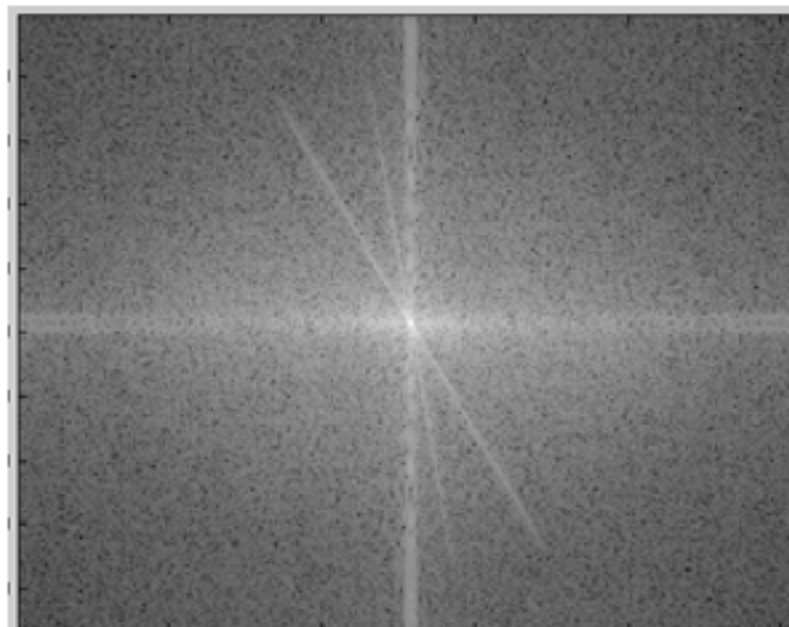


Много высоких частот в
спектре – много мелких
деталей в исходном
изображении



Искусственная сцена

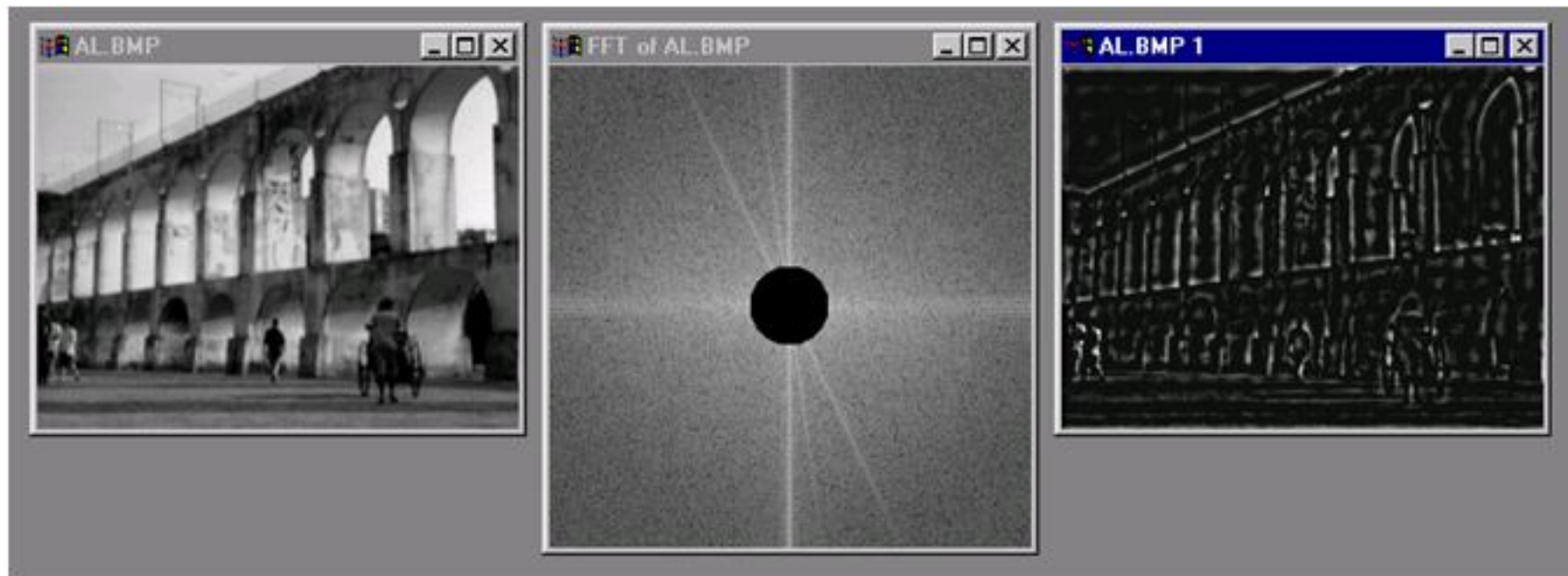
Яндекс





Края в изображении

Яндекс





Резюме

- 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
- Есть и другие виды представления изображений
 - На основе вейвлет-разложения
 - Разреженные представления на основе словаря
 - Некоторые из них будут рассмотрены во второй части курса



План лекции

- Частотная фильтрация изображений
 - DCT-разложение, JPEG
 - Теорема о свёртке
- Сегментация изображений
 - Простейшие методы сегментации
 - Текстура
 - Использование сегментации для анализа изображений



Из чего состоит изображение?

Яндекс



Slide by Alexey Efros



Из «кусков» - отдельных объектов

Яндекс

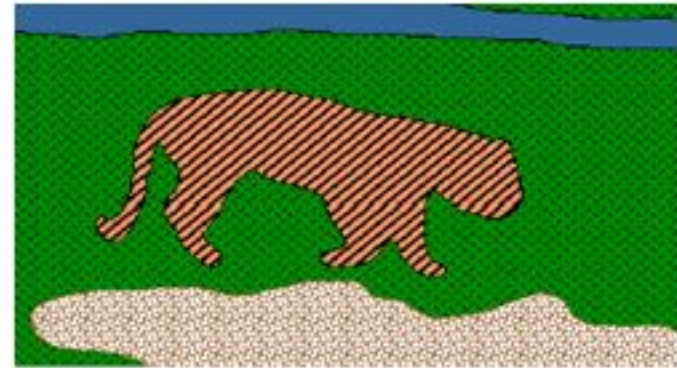


Slide by Alexey Efros



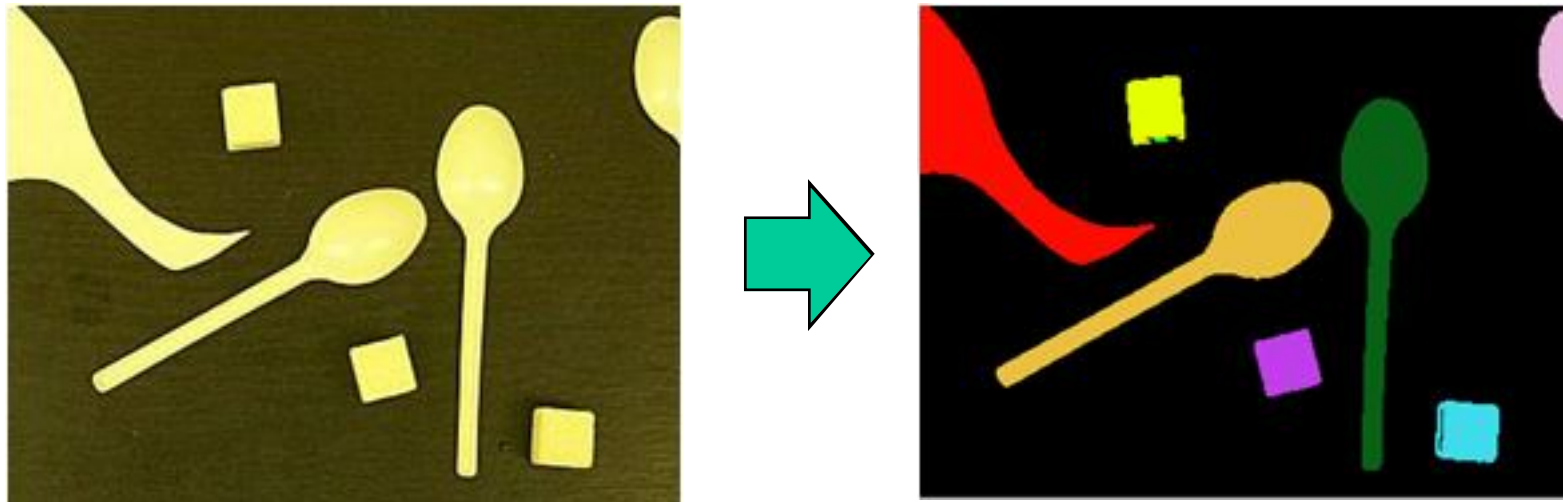
Сегментация

- 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

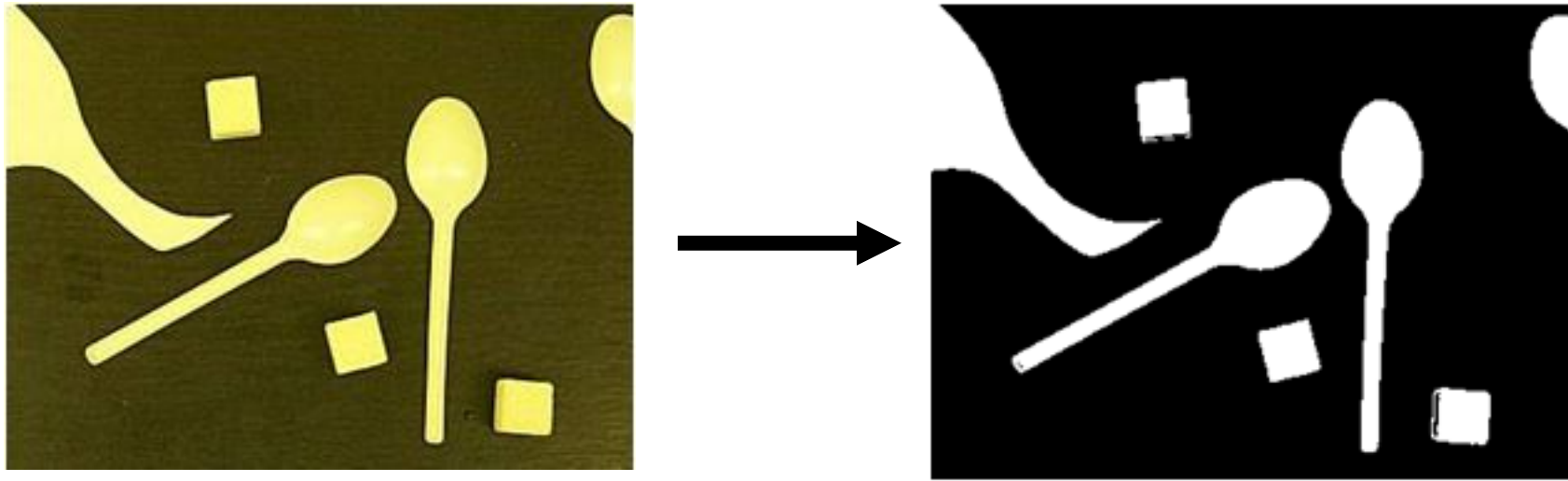
Чем отличаются объекты на этом изображении?



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



Пороговая бинаризация



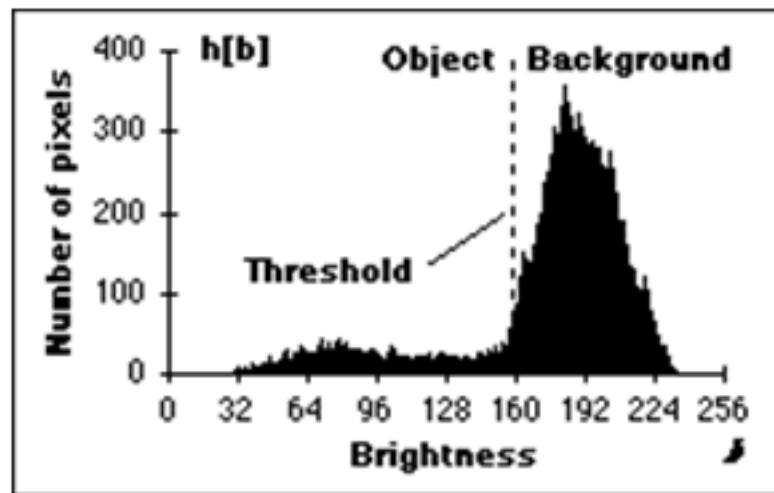
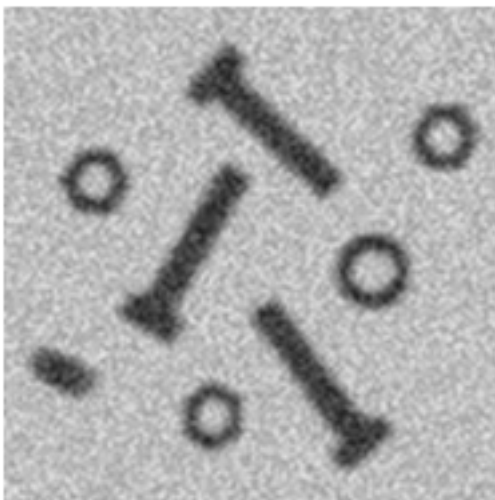
- Пороговая фильтрация (thresholding)
 - Пиксели, которых выше/ниже некоторого порога, заданного «извне», помечаются 1
 - Ниже порога помечаются 0
- Бинарное изображение – пиксели которого могут принимать только значения 0 и 1
- Бинаризация - построение бинарного изображения по полутоновому / цветному



Пороговая фильтрация

Более интересный способ – определение порога автоматически, по характеристикам изображения

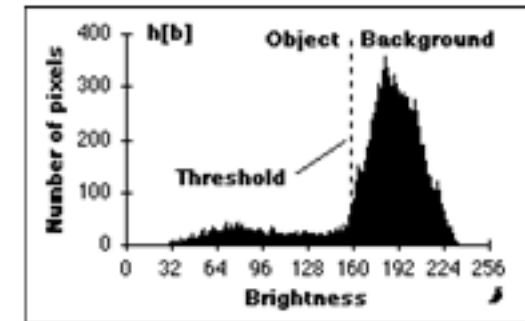
- Анализ гистограммы



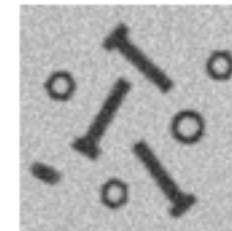


Анализ гистограммы

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



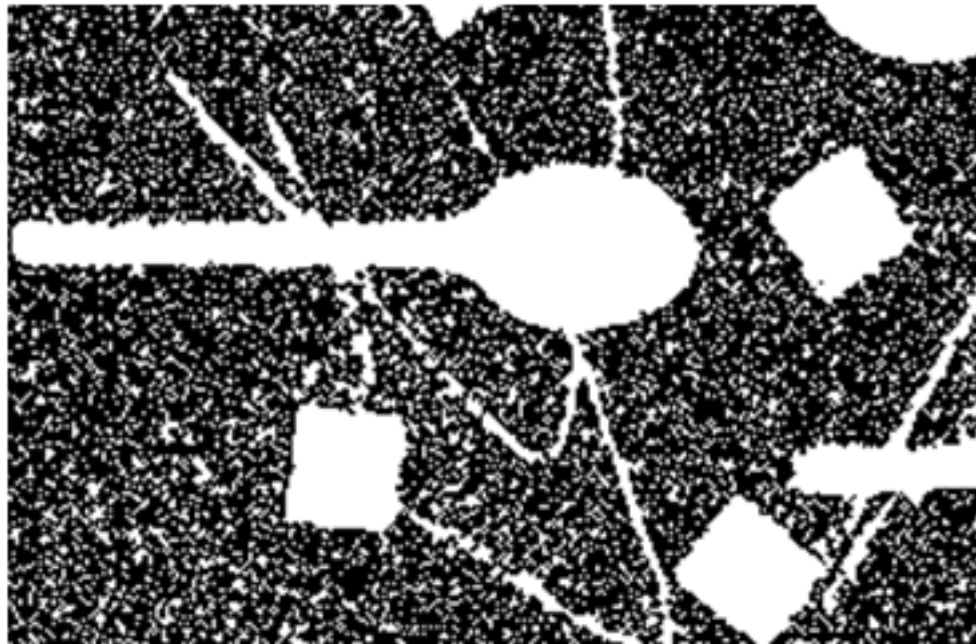
1. Сгладить гистограмму;
2. Найти ячейку гистограммы h_{\max} с максимальным значением;
3. На стороне гистограммы не относящейся к объекту (на примере – справа от пика фона) найти яркость h_p , количество пикселей с яркостью $\geq h_p$ равняется $p\%$ (например 5%) от пикселей яркости которых $\geq h_{\max}$;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





Шум в бинарных изображениях

Пример бинарного изображения с сильным шумом

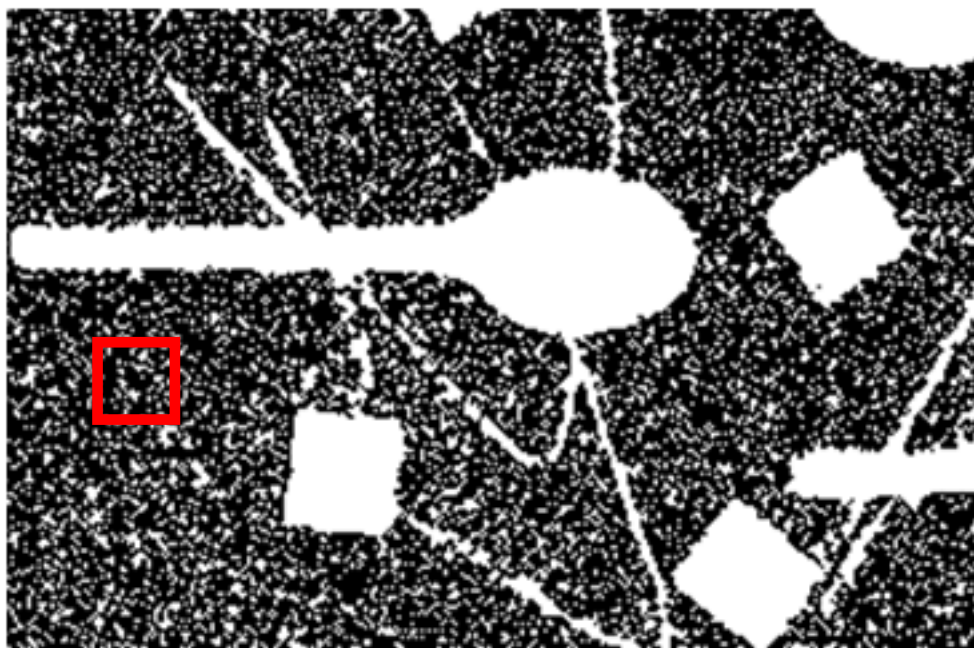


Часто возникает из-за невозможности полностью подавить шум в изображениях, недостаточной контрастности объектов и т.д.



Шум в бинарных изображениях

- ⑩ По одному пикселю невозможно определить – шум или объект?
- ⑩ Нужно рассматривать окрестность пикселя!





Подавление и устранение шума

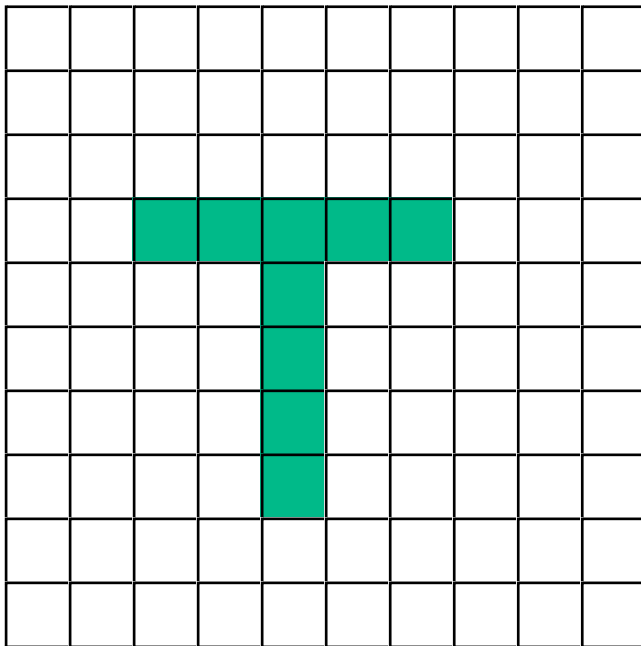
Широко известный способ - устранение шума с помощью операций математической морфологии:

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

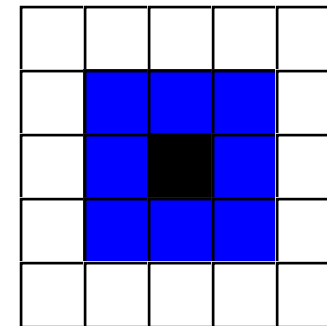


Математическая морфология

A



B

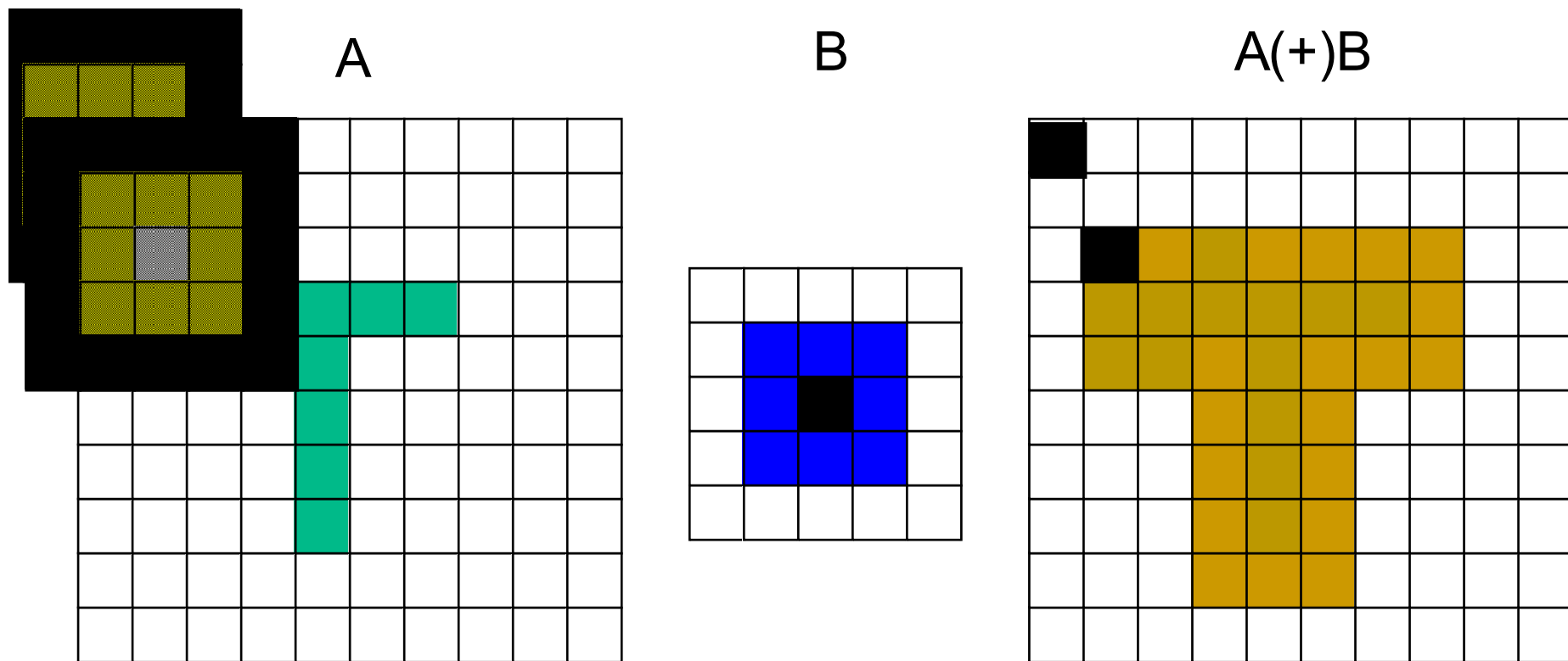


- Множество A обычно является объектом обработки
- Множество B (называемое структурным элементом) – инструмент обработки



Расширение в дискретном случае

Яндекс



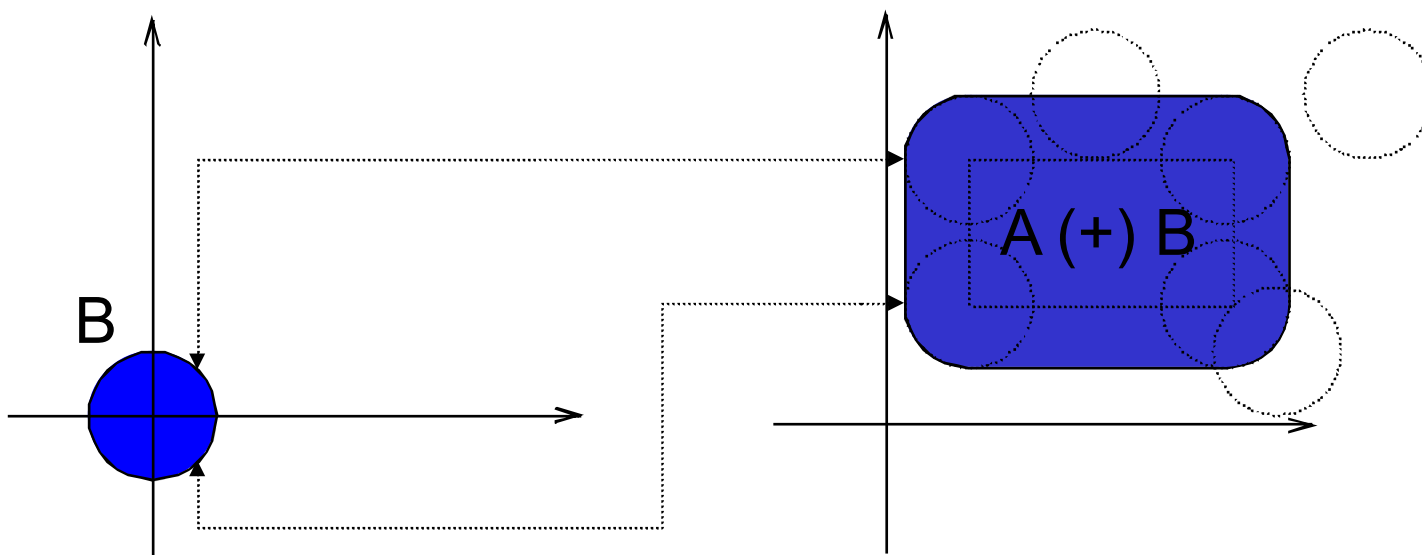
Операция «расширение» - аналог логического «или»



Расширение

Расширение (dilation)

$$A (+) B = \{t \in \mathbb{R}^2: t = a + b, a \in A, b \in B\}$$

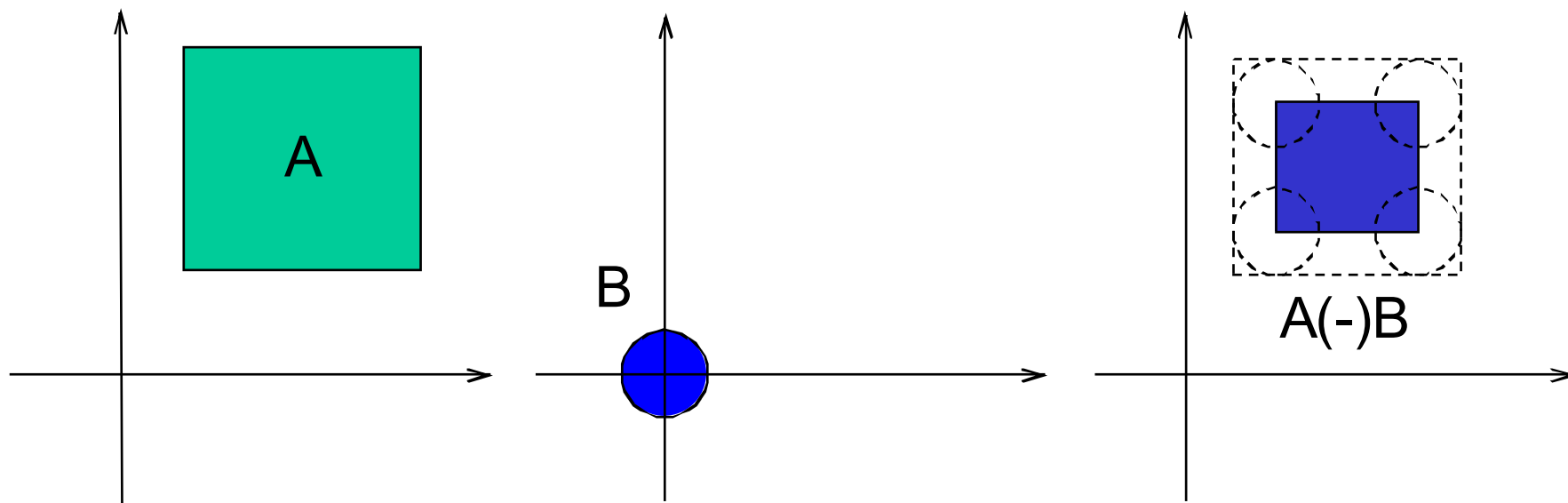




Сужение

Сужение (erosion)

$A (-) B = (A^C (+) B)^C$, где A^C – дополнение A





Результат операции сужения



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



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



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



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



Операции раскрытия и закрытия

Морфологическое раскрытие (opening)

- **$\text{open}(A, B) = (A (-) B) (+) B$**

Морфологическое закрытие (closing)

- **$\text{close}(A, B) = (A (+) B) (-) B$**

Попробуйте догадаться, что эти операции делают?



Применение открытия

Применим операцию открытия к изображению с сильным шумом:



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



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



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



Сужение vs Открытие

Яндекс



Сужение



Открытие



Дефекты бинаризации

Пример бинарного изображения с дефектами
распознаваемых объектов





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

Яндекс

Не во всех случаях математическая морфология так легко убирает дефекты, как хотелось бы...

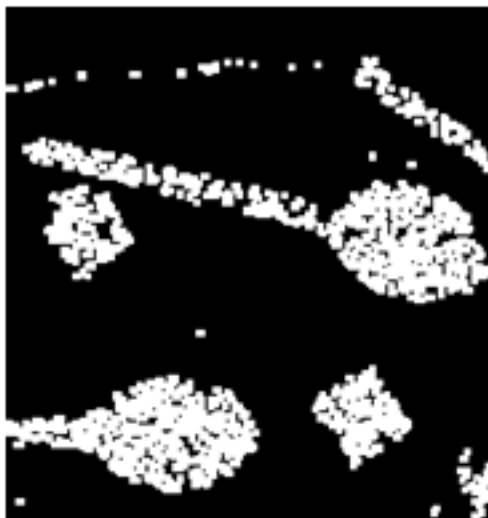




Применения операции открытия



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



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



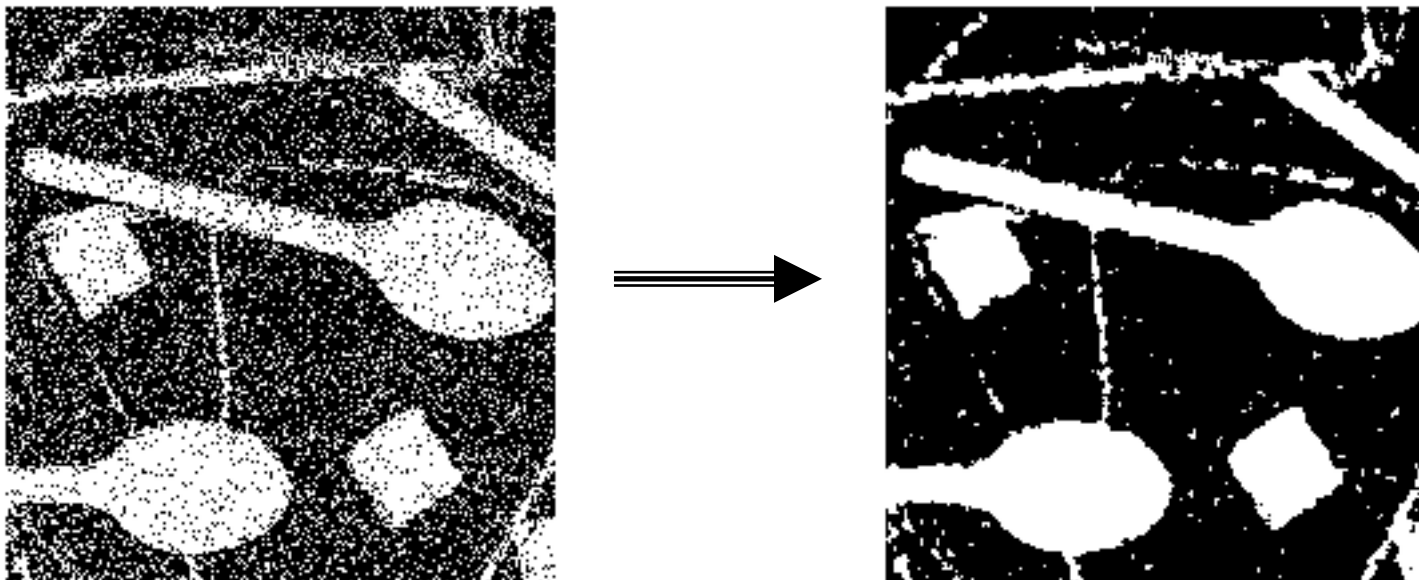
$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

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?

Яндекс



Получили бинарное
изображение



Нужна карта разметки



Выделение связанных областей

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



Соседи пикселей:

	1	
2	*	3
	4	

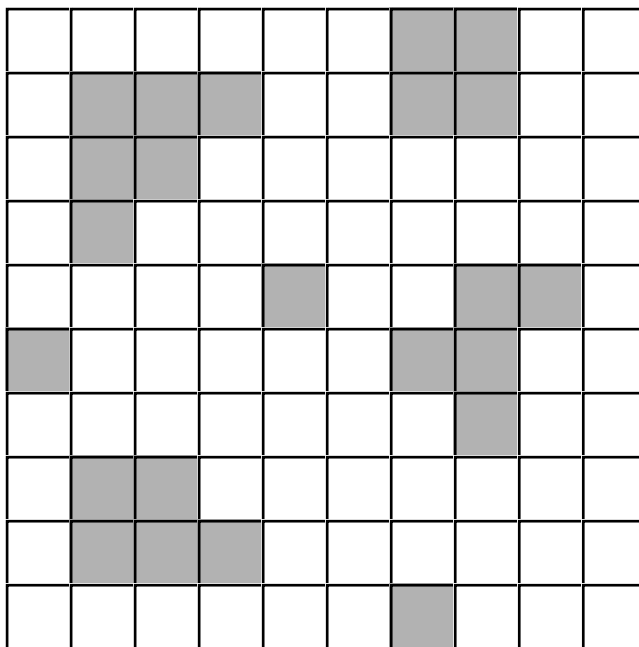
4-СВЯЗНОСТЬ

1	2	3
4	*	5
6	7	8

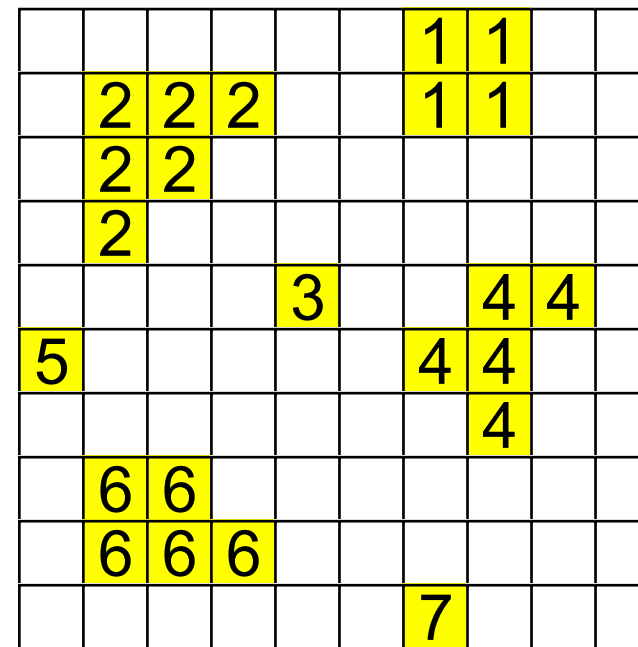
8-СВЯЗНОСТЬ



Разметка связанных 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

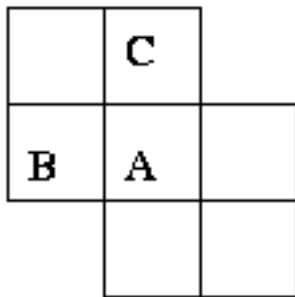
```
void Fill(BIT* img[], int* labels[], int x, int y, int L)
{
    if( (labels[x][y] == 0) && (img[x][y] == 1) )
    {
        labels[x][y] = L;
        if( x > 0 )
            Fill(img, labels, x - 1, y, L);
        if( x < W - 1 )
            Fill(img, labels, x + 1, y, L);
        if( y > 0 )
  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

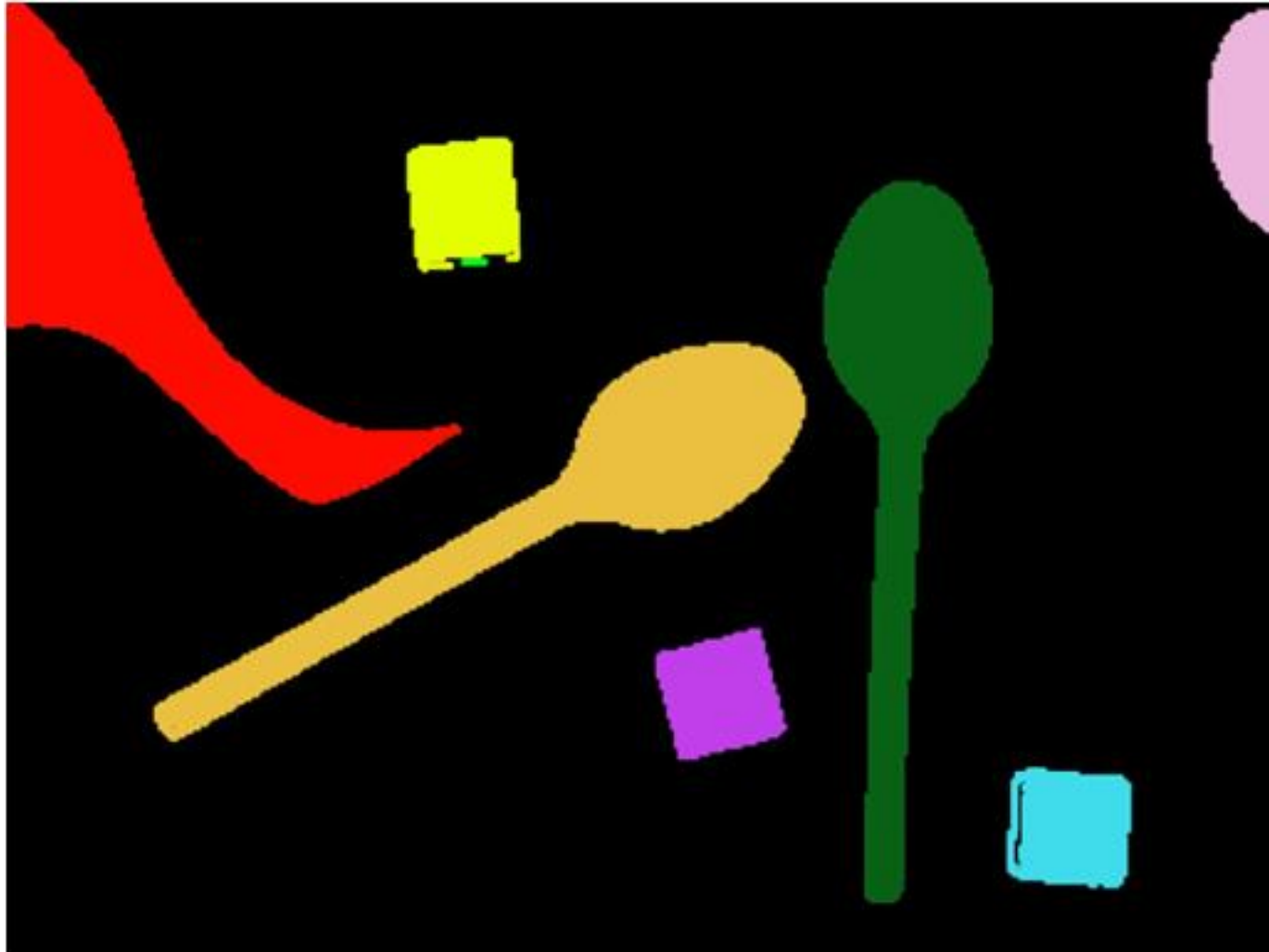
else if B and C labeled
    if B label = C label
        copy label to A
    else
        copy either B label or C label to A
        record equivalence of labels
```

За сколько операций мы разметим изображение?



Выделенные связанные компоненты

Яндекс





Сегментация



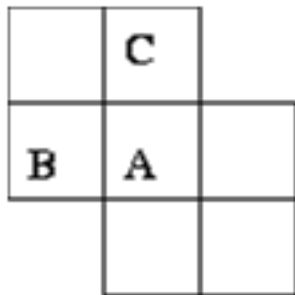
Тот же метод можно использовать для сегментации и обычных изображений на однородные области



Выделение однородных областей методом последовательного сканирования



Сканируем изображение сверху вниз, слева направо:



1. if $I(A) - I_{avg}(CI(B)) > \delta$ and $I(A) - I_{avg}(CI(C)) > \delta$ -
создаем новую область, присоединяем к ней пиксел А
2. if $I(A) - I_{avg}(CI(B)) < \delta$ xor $I(A) - I_{avg}(CI(C)) < \delta$ –
добавить А к одной из областей
3. if $I(A) - I_{avg}(CI(B)) < \delta$ and $I(A) - I_{avg}(CI(C)) < \delta$:
 1. $I_{avg}(CI(B)) - I_{avg}(CI(C)) < \delta$ –
сливаем области В и С.
 2. $I_{avg}(CI(B)) - I_{avg}(CI(C)) > \delta$ –
добавляем пиксел А к тому классу, отклонение от которого минимально.

$I(A)$ – яркость пиксела А

$CI(B)$ – область к которой принадлежит пиксел В

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



Выделение однородных областей методом последовательного сканирования

Яндекс

Пример:





Признаки изображения

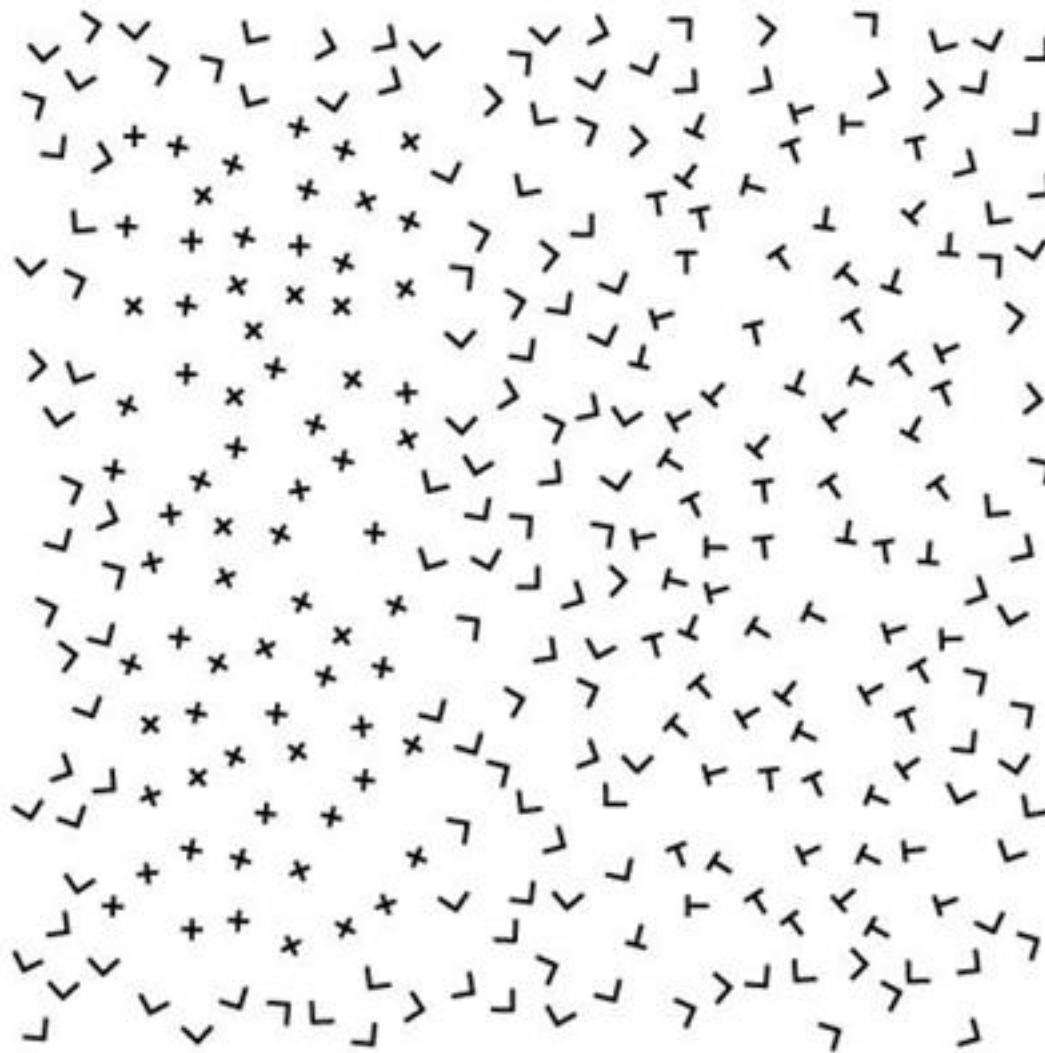
Какие признаки мы можем использовать для сравнения пикселей и регионов?

- Яркость
- Цвет
- ?



Пример

Яндекс



Видите отдельные области?

Image source: Landy & Graham (2004)



«Текстура»

Яндекс

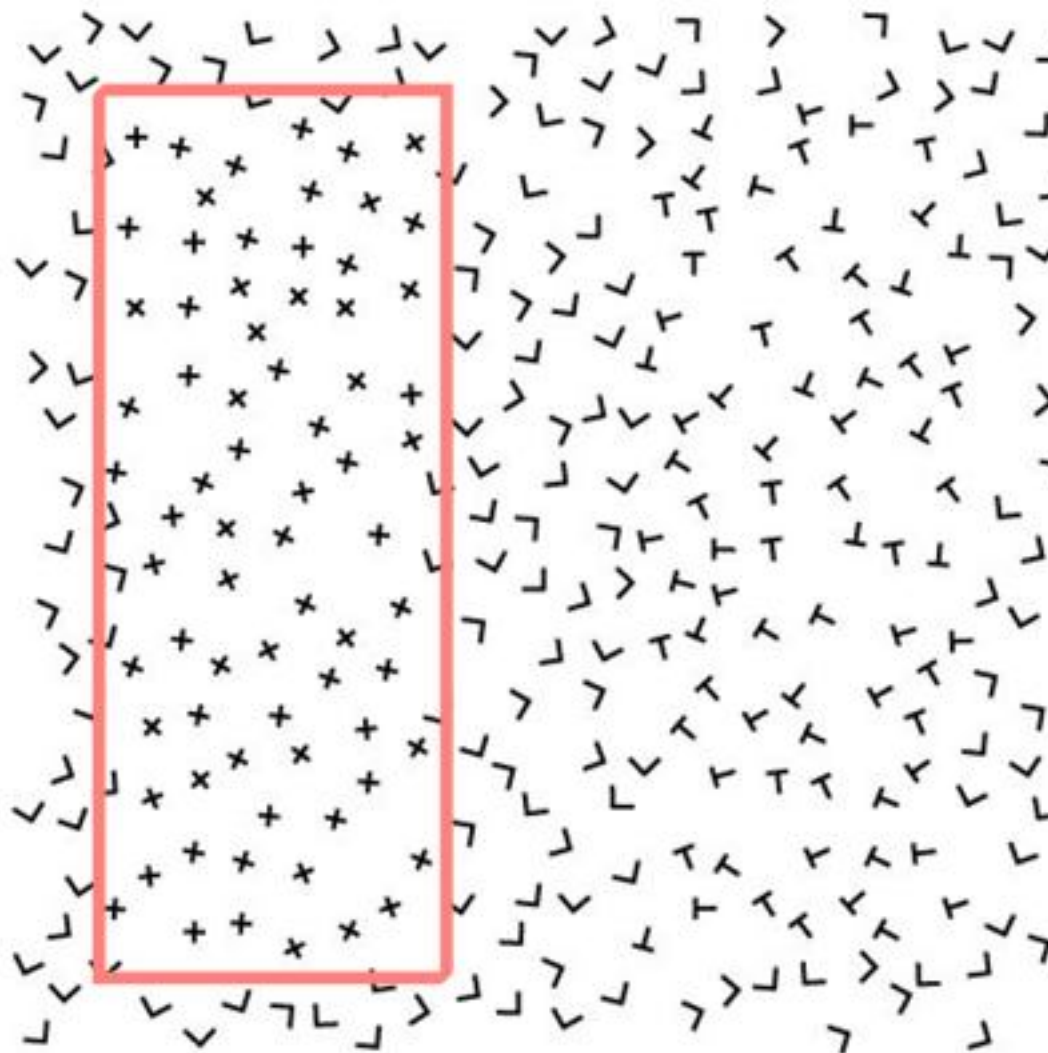


Image source: Landy & Graham (2004)



«Текстура»

Яндекс

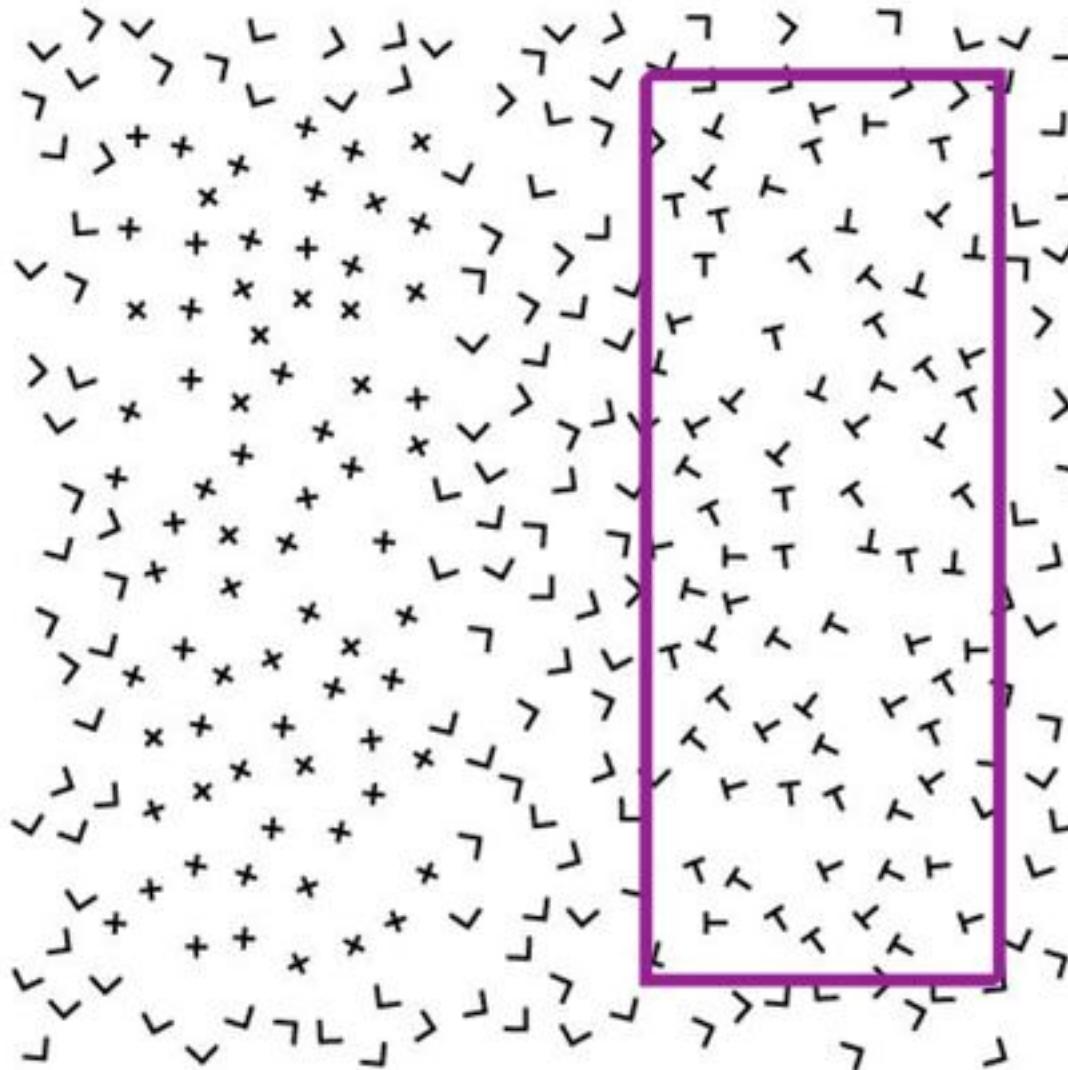
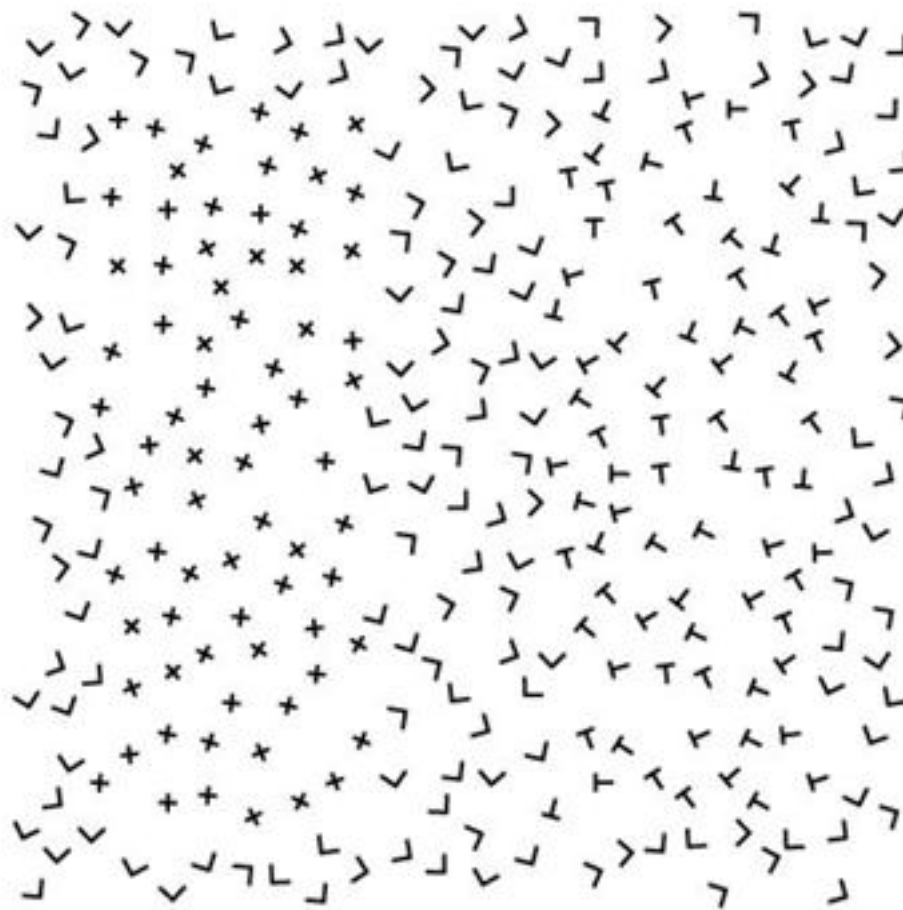


Image source: Landy & Graham (2004)



«Текстура»

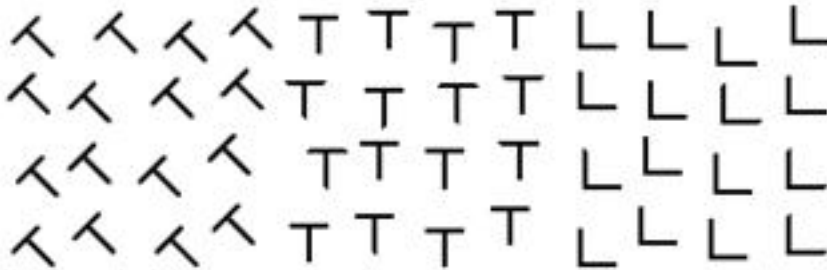
Яндекс



Типичный пример текстурного шаблона для исследований психофизиологического восприятия изображений



Текстура



(a)

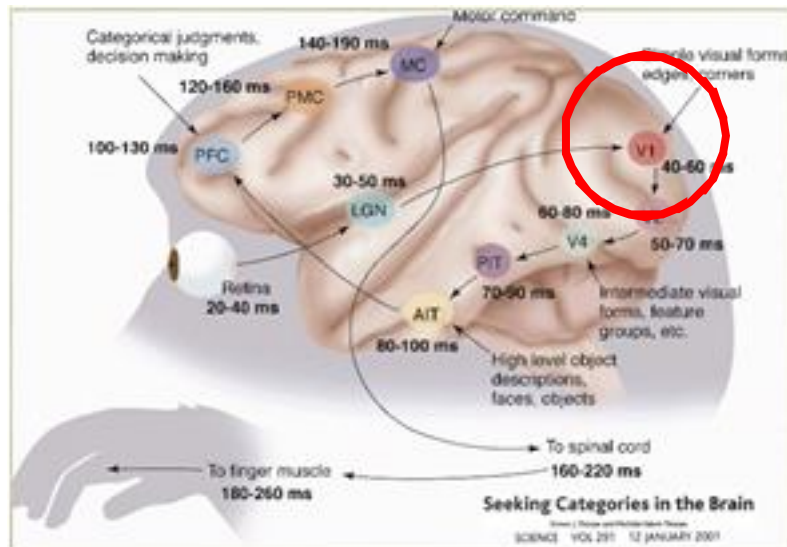


(b)

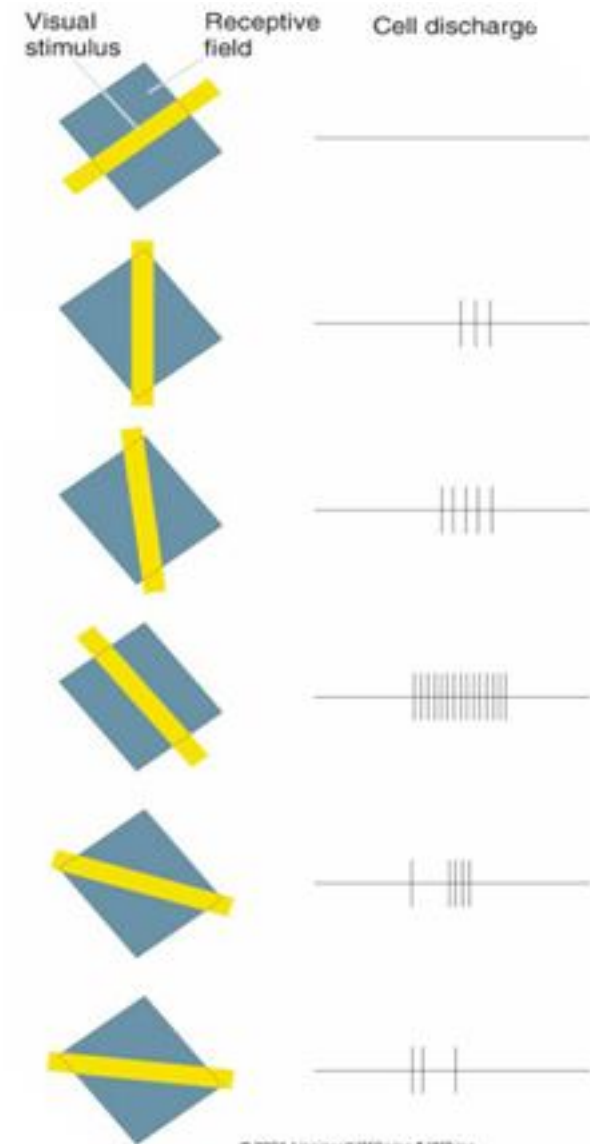
- Это типичные примеры текстурных шаблонов для исследований психофизиологического восприятия изображений
- Человек явно использует не только яркость и цвет, но и ориентацию краёв (градиентов изображения), их распределение, для анализа изображений



«Простые клетки» V1

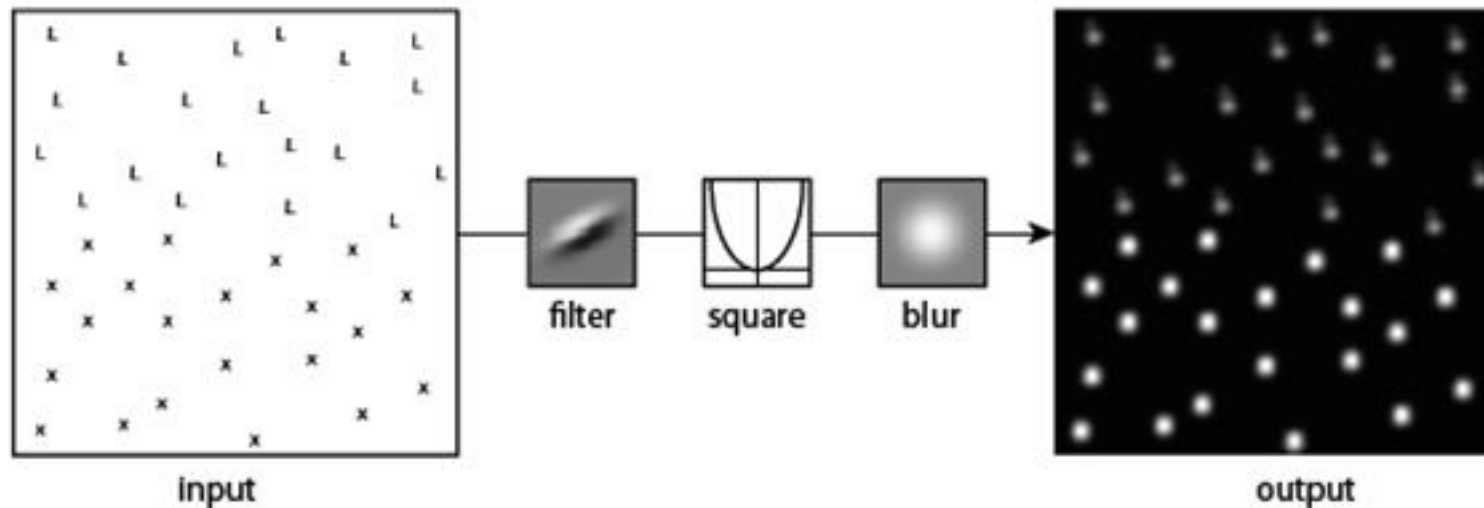


- В первичной визуальной коре головного мозга есть клетки, чувствительные к краям определенной ориентации
- Для каждой области есть набор таких клеток, чувствительные к краям разной ориентации





Анализ текстуры



- Выберем фильтр, чувствительный к краю определенной ориентации
- Результат фильтрации сгладим
- Будут «подсвечены» области, содержащие текстуру с краями заданной ориентации

Pietro Perona and Jitendra Malik «Detecting and Localizing edges composed of steps, peaks and roofs», ICCV 1990



Банки фильтров

- Возьмём теперь несколько фильтров разного масштаба и ориентации
- Такой набор называют «банк фильтров»
- Каждый пиксель изображения после обработки банком фильтров даёт вектор признаков
- Этот вектор признаков эффективно описывает локальную текстуру окрестности пикселя
- Активно используется в сегментации, распознавании изображений и т.д.





Фильтры Габора

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi\frac{x'}{\lambda} + \psi\right)$$

$$x' = x \cos(\theta) + y \sin(\theta)$$

$$y' = -x \sin(\theta) + y \cos(\theta)$$

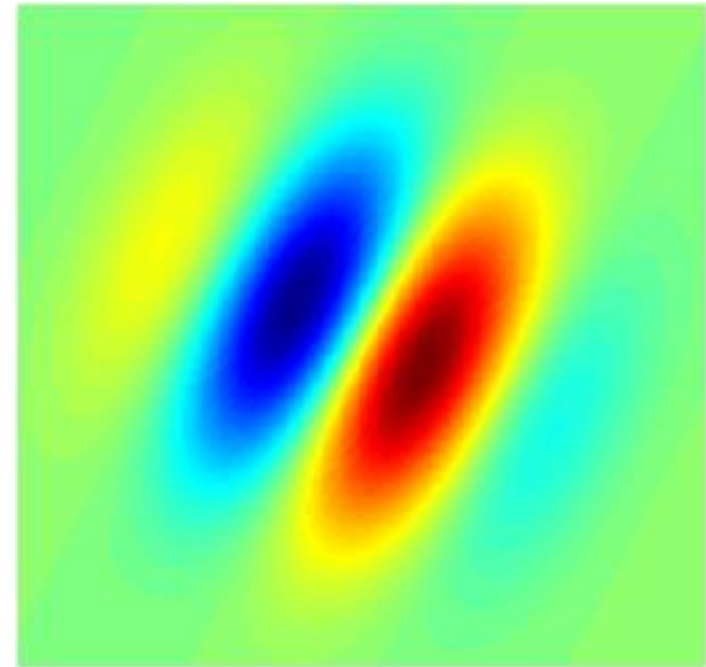
θ - ориентация

λ - длина волны

σ - сигма гауссиана

γ - соотношение размеров (aspect ratio), «эллиптичность фильтра»

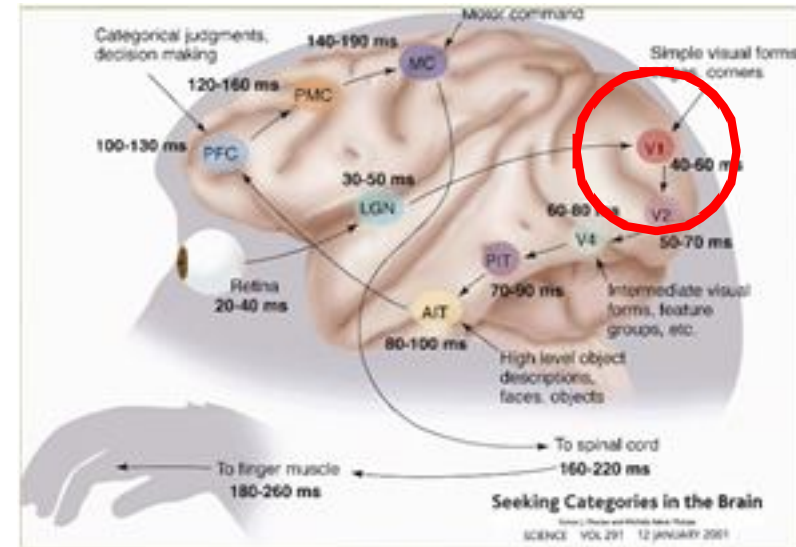
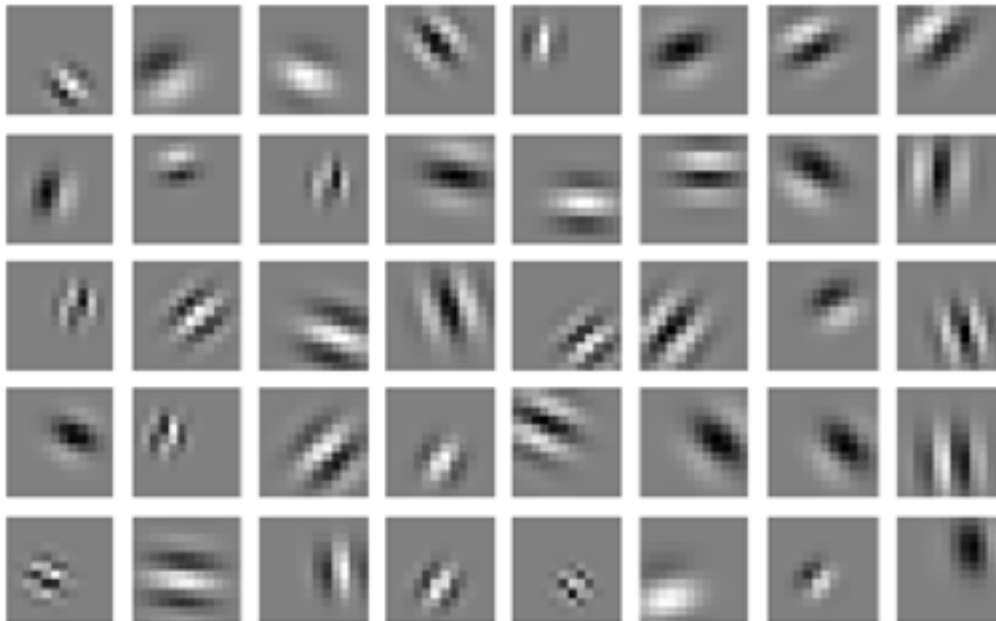
ψ - сдвиг фазы



- 2D фильтр Габора – ядро гауссиана, домноженное на синусоиду
- Предложены в 1947 Денисом Габором (нобелевским лауреатом), независимо переоткрыты в 1980 году



Связь со зрением человека



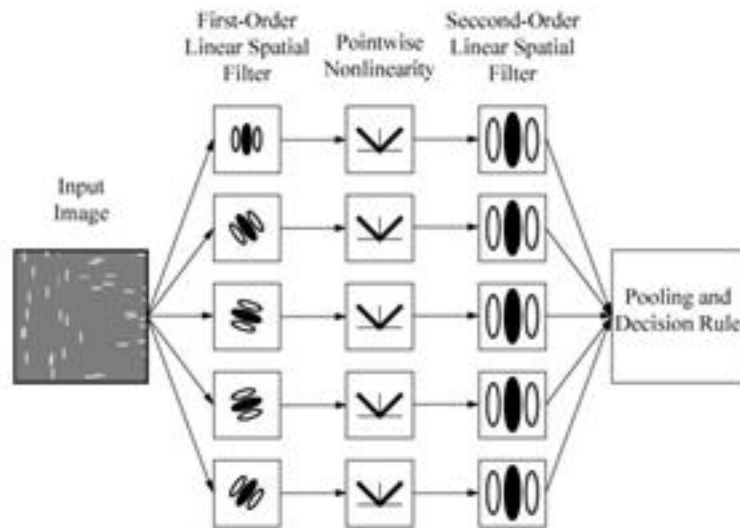
- Похожи на форму рецептивных полей простых клеток (simple cells) в визуальной коре мозга человека
- Стандартный банк фильтров для анализа текстуры, называют «биологически обусловленным»

J. G. Daugman, "Two-dimensional spectral analysis of cortical receptive field profiles.," *Vision research*, vol. 20, no. 10, pp. 847–856, 1980.

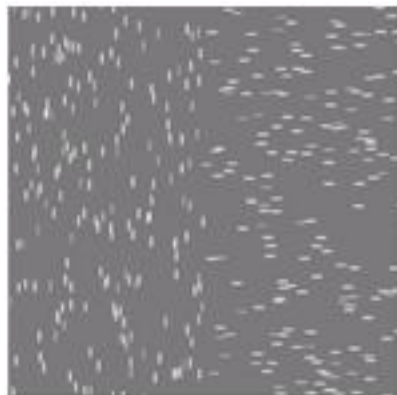
J. G. Daugman, "Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression," *IEEE Trans. Acoust. Speech Sigal Process.*, vol. 36, no. 7, pp. 1169–1179, 1988



Модель «back pocket»



- Такие модели текстуры называют «back pocket»
- Модель «биологически ВОЗМОЖНА»
- По подобным моделям опубликовано много работ.



Ввод



После 1 этапа



После 2 этапа



Выход

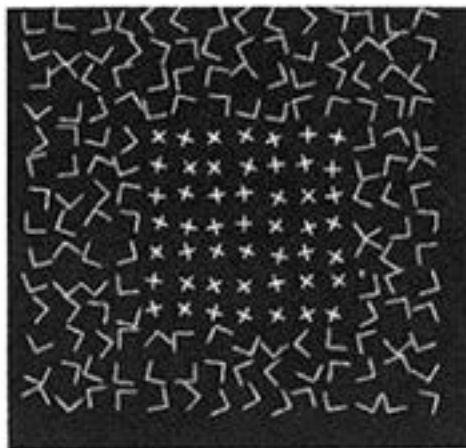


Пример, Bergen & Adelson (1988)

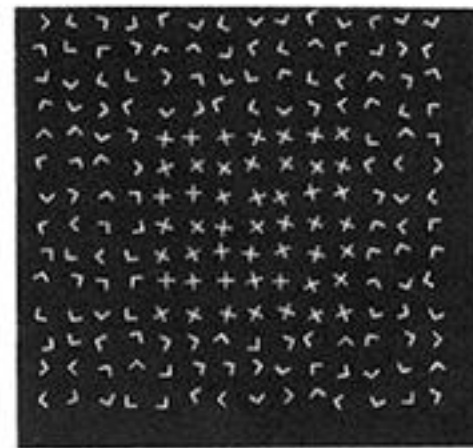
Яндекс



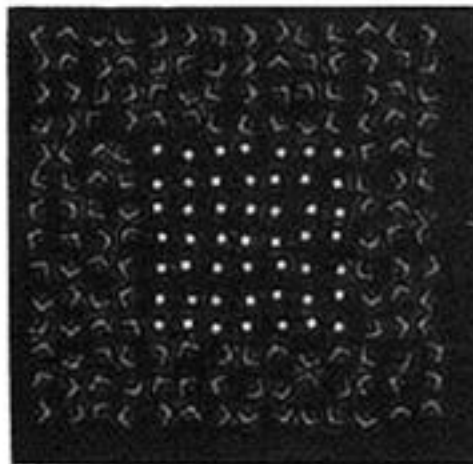
a



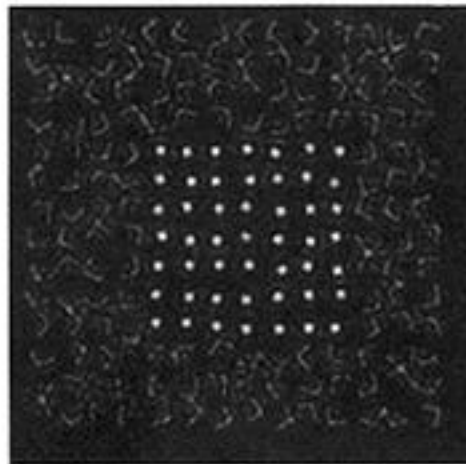
b



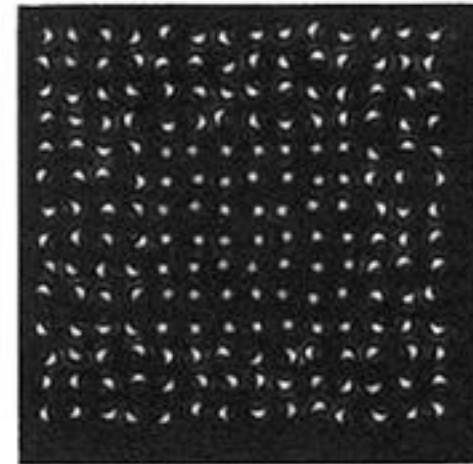
c



d



e

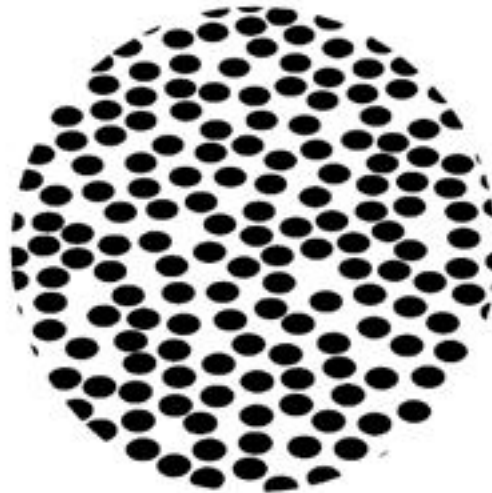


f



Психологическое свойство текстуры

Яндекс





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 поверхности

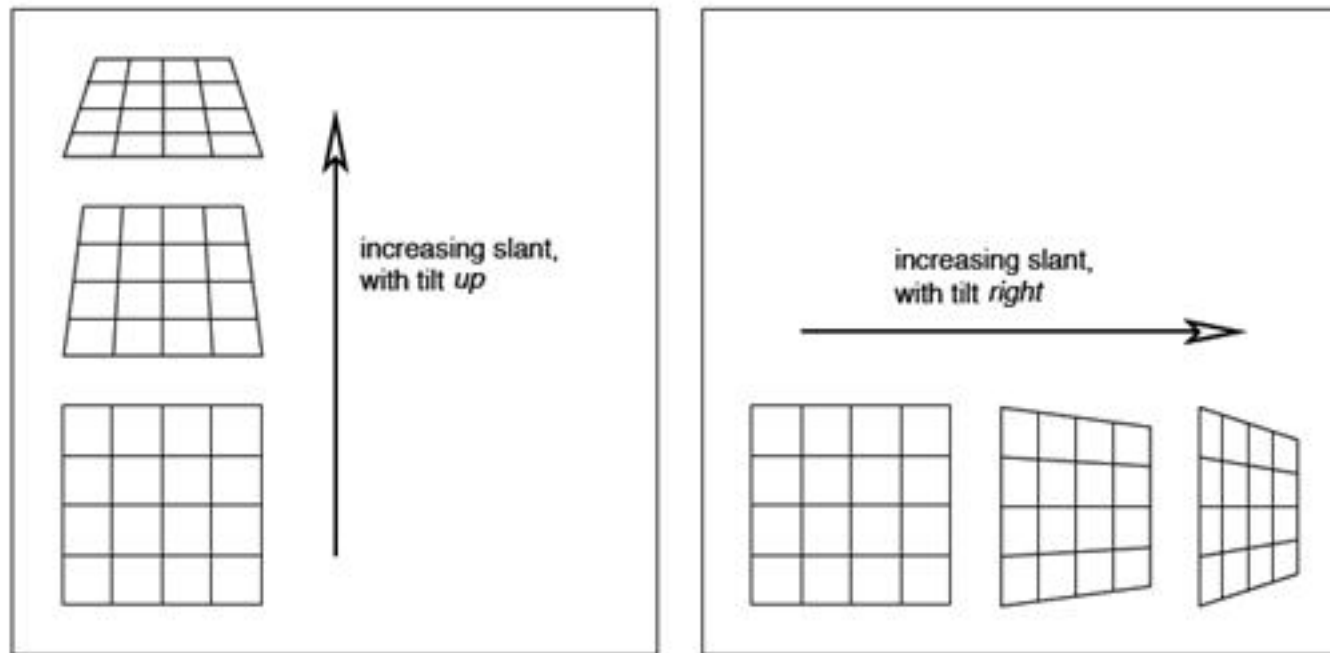


Figure 8.7. Surface orientation is often characterized in terms of *slant* and *tilt*.



Современные алгоритмы



- Задача сегментации изображений продолжает активно исследоваться.
- Иногда используется один признак, иногда множество
- Подробнее современные алгоритмы рассматриваются во второй части курса



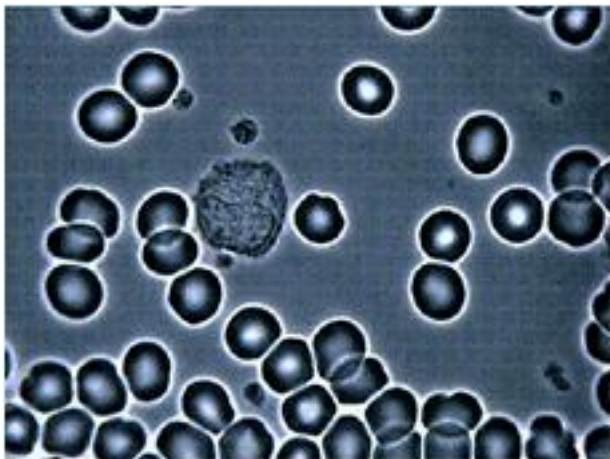
План лекции

- Частотная фильтрация изображений
 - DCT-разложение, JPEG
 - Теорема о свёртке
- Сегментация изображений
 - Простейшие методы сегментации
 - Текстура
 - Использование сегментации для анализа изображений



Распознавание через сегментацию

Яндекс



Клетки крови



Ложки и сахар



Монеты и купюры

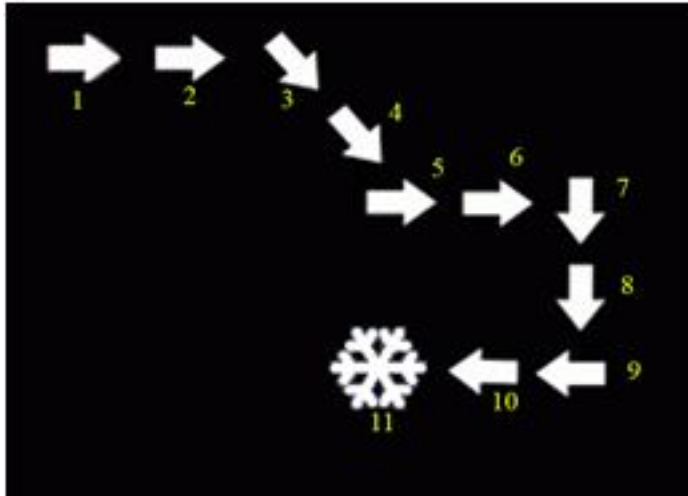


Номера

Все эти объекты сняты с
определенного, фиксированного
ракурса, на контрастном фоне



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 инерц
- Удлиненность (эксцентриситет)

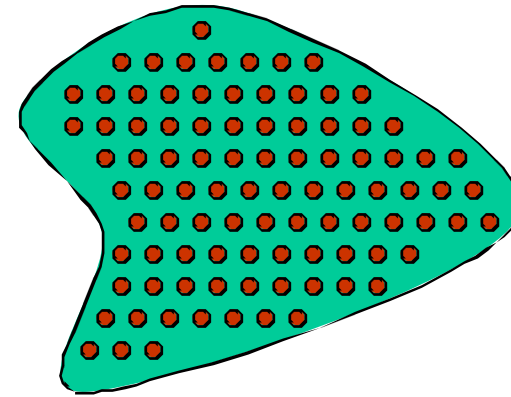




Площадь и центр масс

- Площадь – количество пикселей в области;

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



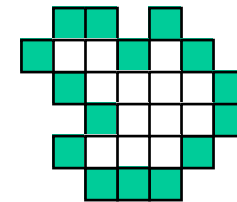
- Центр масс

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



Периметр и компактность

- Периметр – количество пикселей принадлежащих границе области;



- Компактность – отношение квадрата периметра к площади;

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

Наиболее компактная фигура – $C = 4\pi$ круг:





Подсчет периметра области

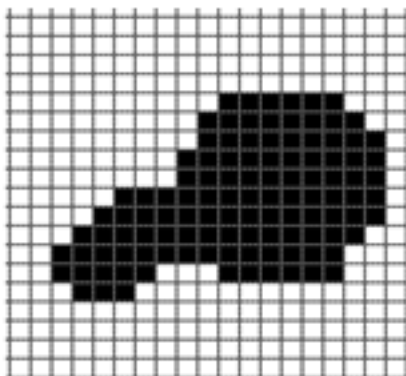
1. Пиксель лежит на границе области, если он сам принадлежит области и хотя бы один из его соседей области не принадлежит.
(внутренняя граница)
2. Пиксель лежит на границе области, если он сам не принадлежит области и хотя бы один из его соседей области принадлежит.
(внешняя граница)

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

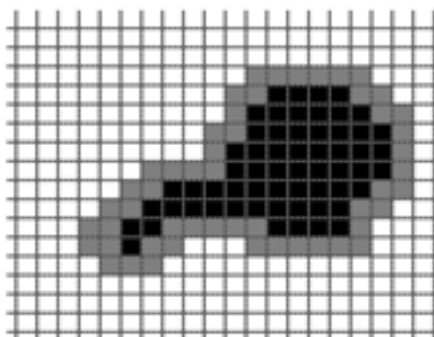


Пример периметров области

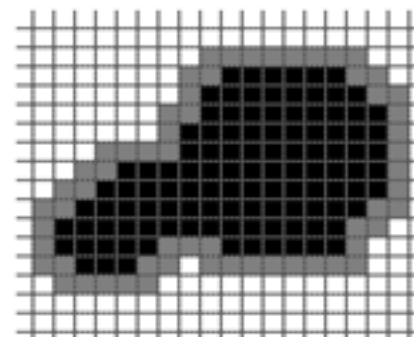
Яндекс



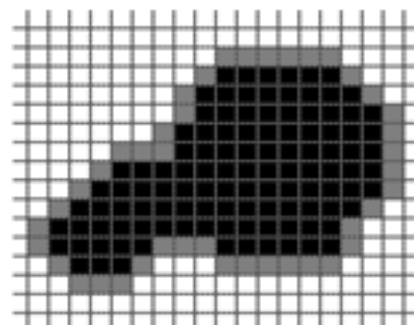
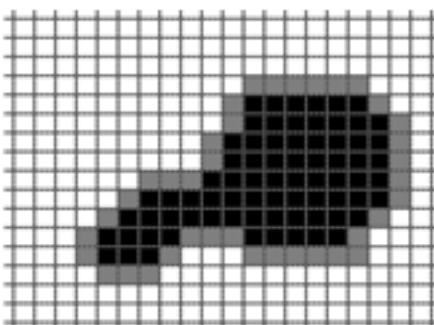
Область



Внутренняя граница



Внешняя граница





Операция оконтуривания объекта

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

Внутреннее оконтуривание

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

Внешнее оконтуривание

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



Пример оконтуривания объекта

Яндекс





Статистические моменты области



Дискретный центральный момент m_{ij} области определяется следующим образом:

$$m_{ij} = \sum_{x,y \in S} (x - \bar{x})^i (y - \bar{y})^j I(x, y)$$

Центр масс области



Инвариантные характеристики

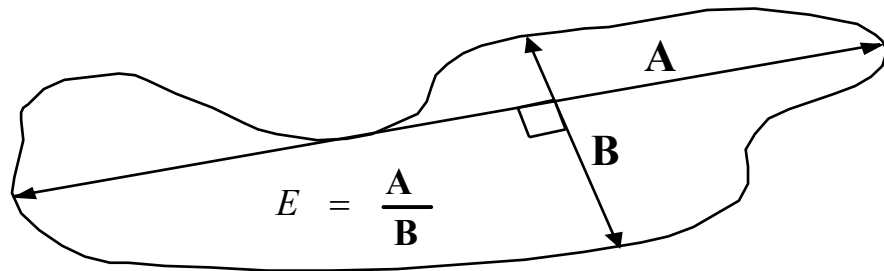
Для распознавания нас интересуют характеристики инвариантные по отношению к масштабированию, переносу, повороту:

- 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

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

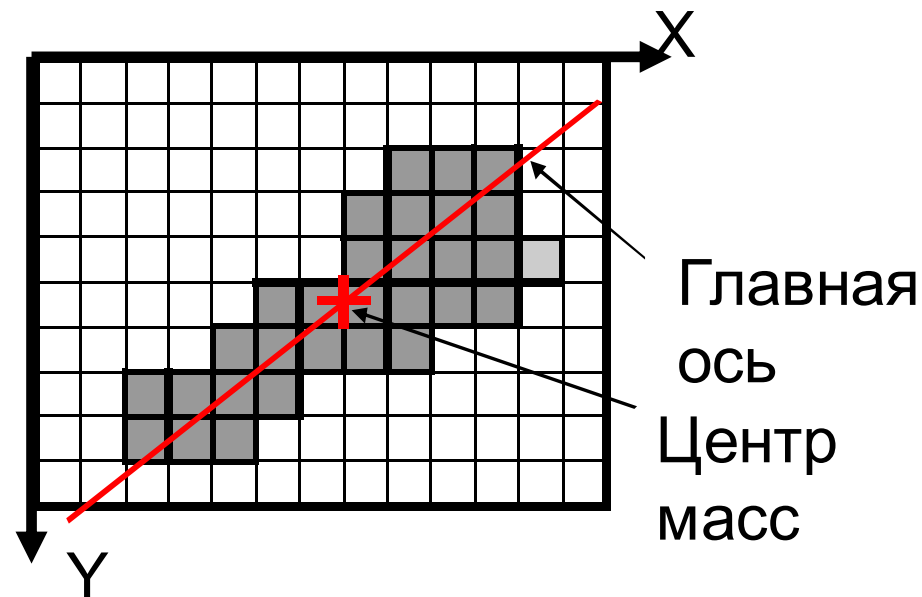




Ориентация главной оси инерции

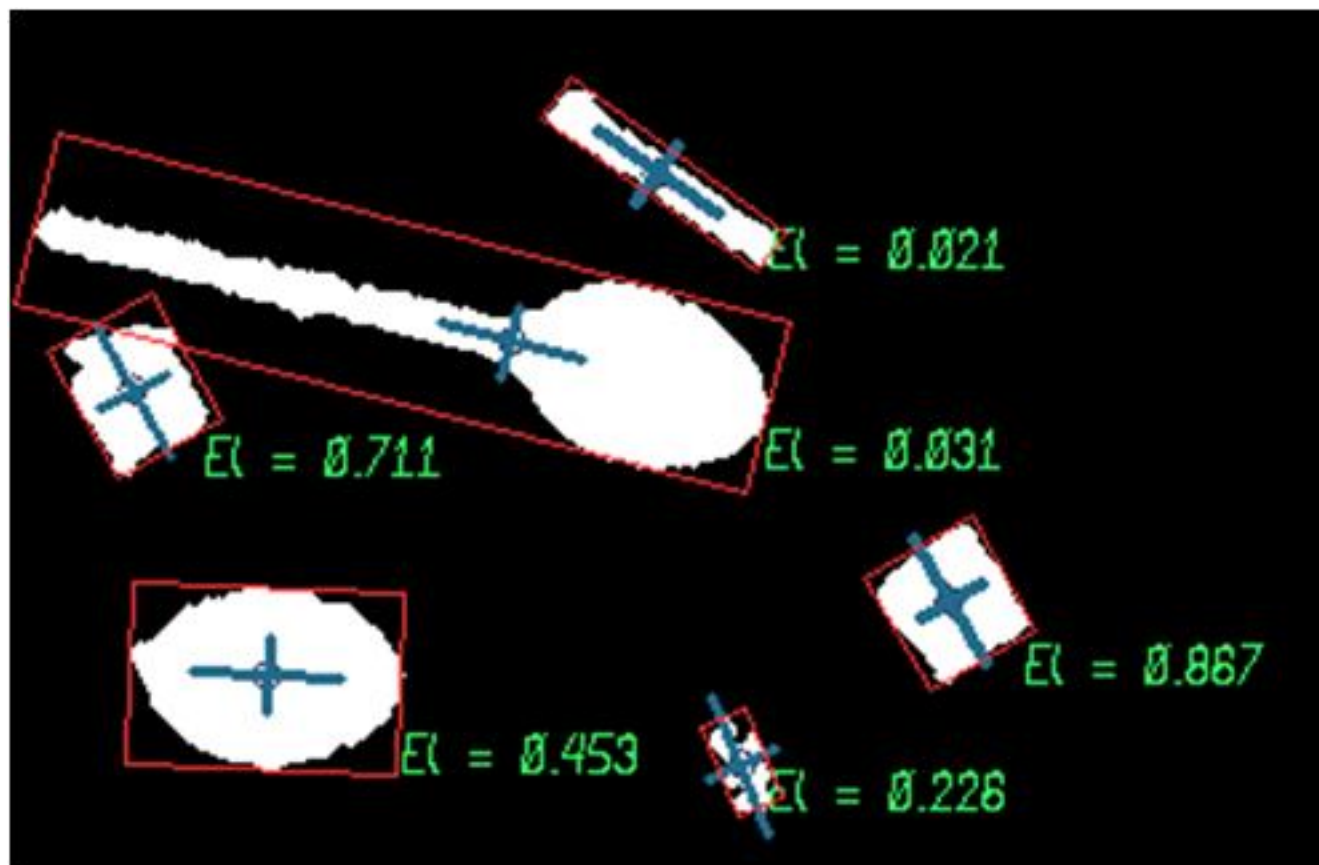
Не является инвариантной к повороту, но в ряде случаев предоставляет полезную информацию об ориентации объекта:

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





Пример



Вычисленные значения признаков



Фотометрические признаки

Для каждой области можно подсчитать некий набор простейших числовых характеристик:

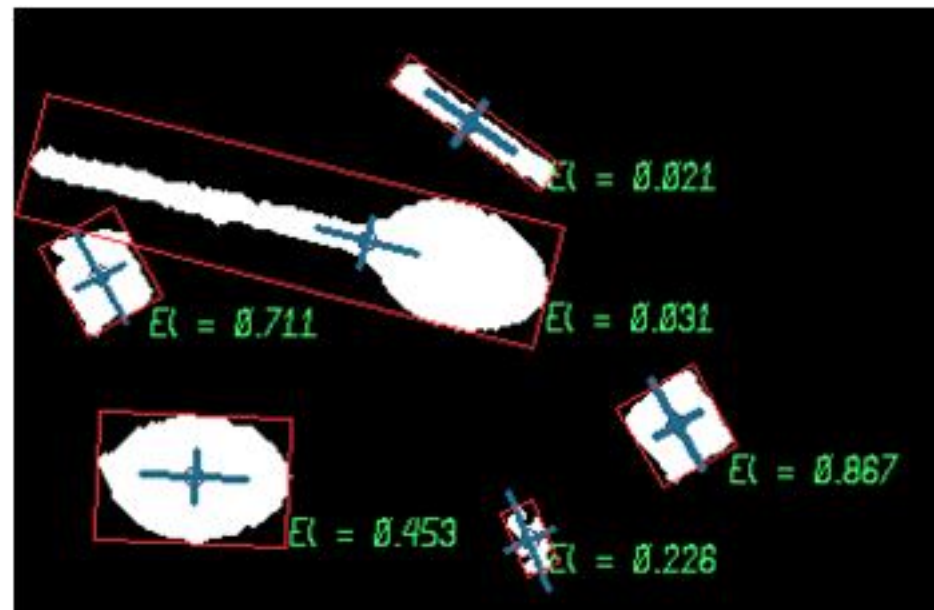
- Средняя яркость
- Средний цвет (если изображение цветное)
- Гистограмма распределения яркостей
(или три гистограммы распределения R, G, B)
- Дисперсию (разброс) яркостей или цвета

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



Как анализировать признаки

- Пример – ложки и сахар





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

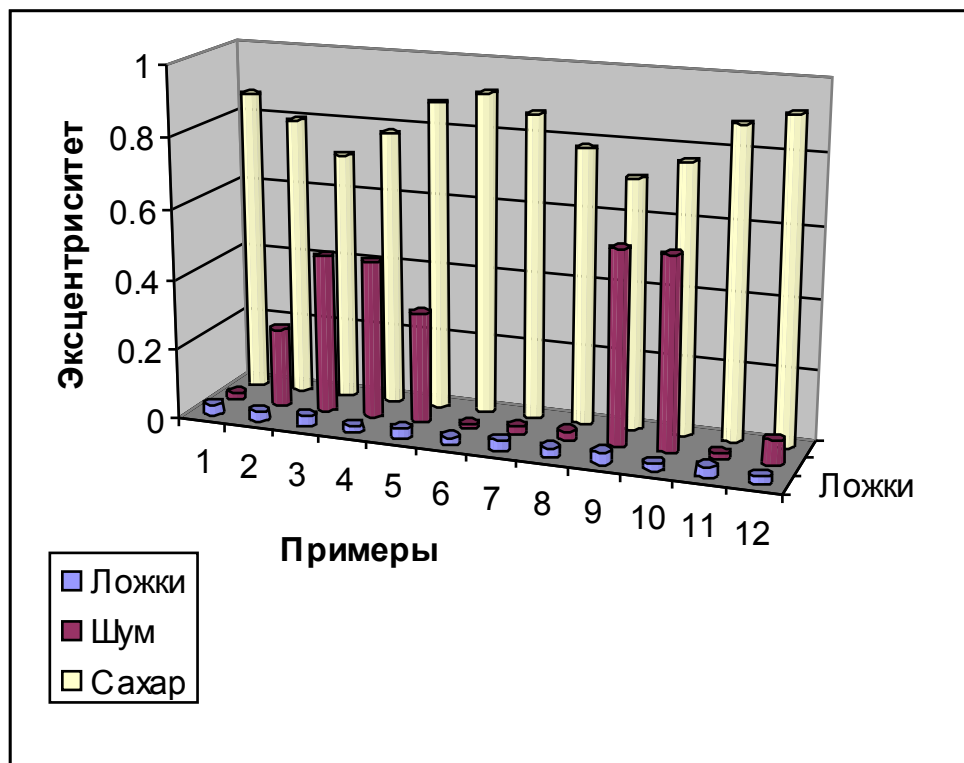
- Собрать тренировочную базу изображений
 - Где только ложки
 - Где только сахар
 - Где только шум

Как получить такие? Да просто закрасить все остальное.
- Брать признаки и строить графики



Графический анализ

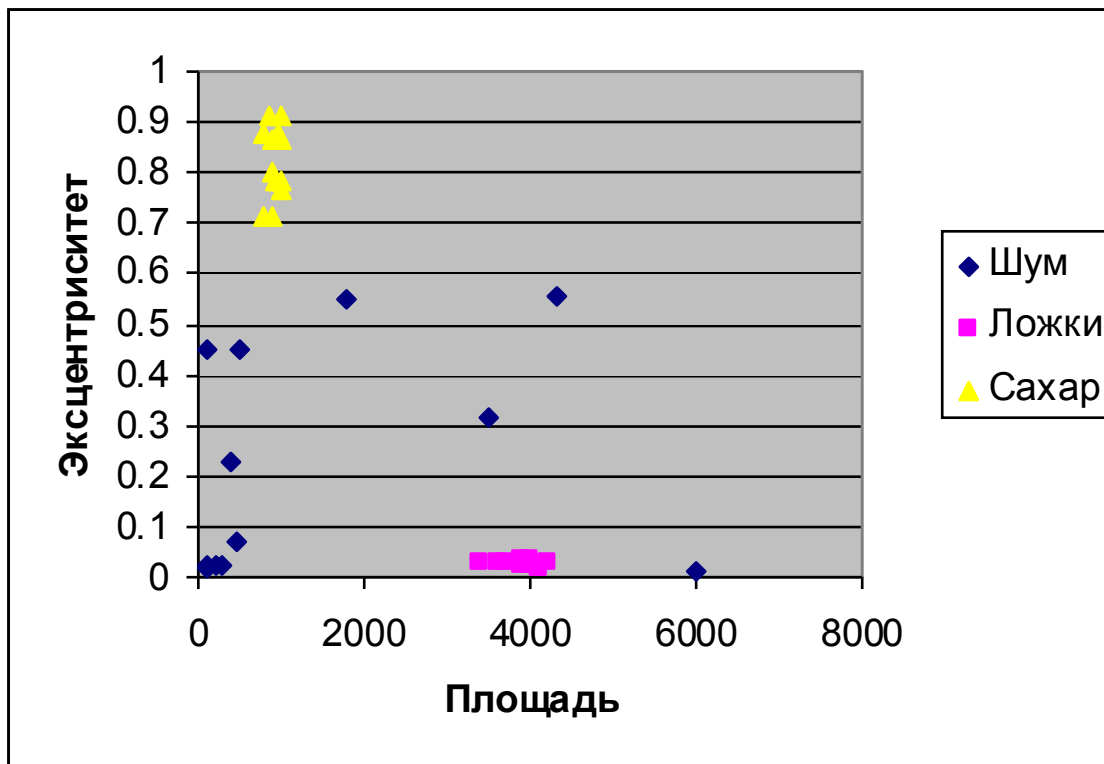
- Диаграмма распределения эксцентриситета (проблема – не получается отличить шум от ложек)





Графический анализ

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



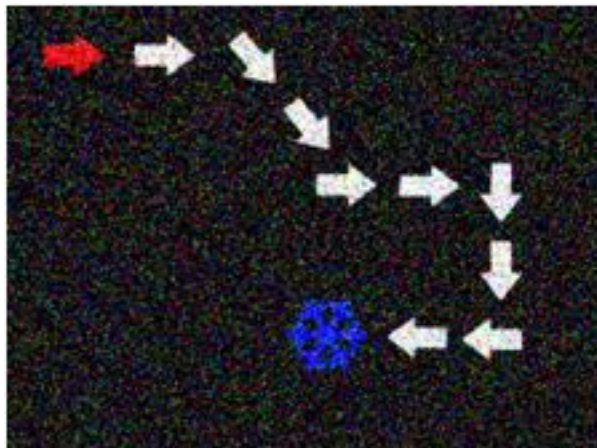


Машинное обучение

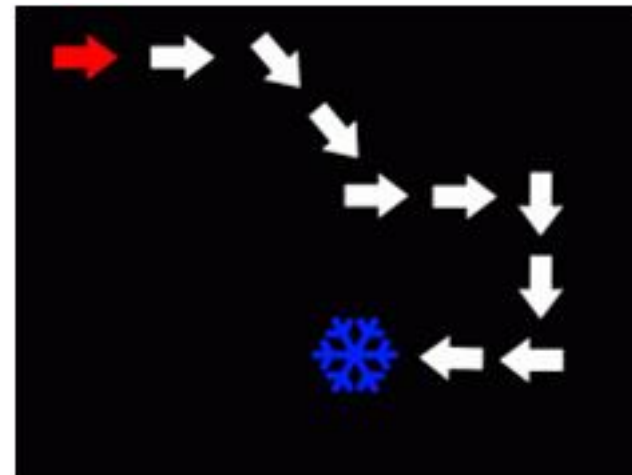
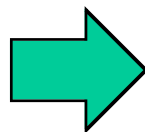
- Причина бурного развития компьютерного зрения в последние годы.
- Требуются большие коллекции примеров для обучения.
- Существенная часть оставшегося курса посвящена этому подходу!



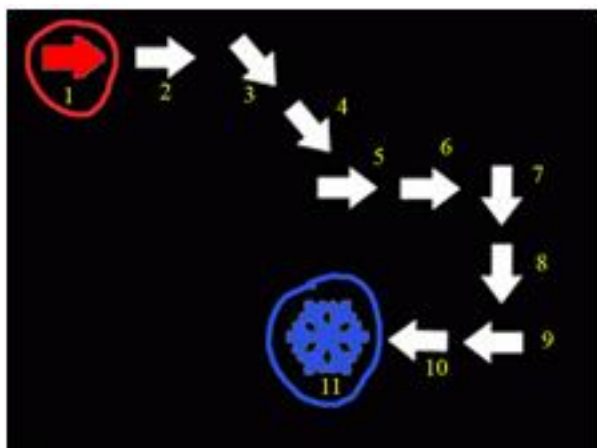
Схема простого алгоритма



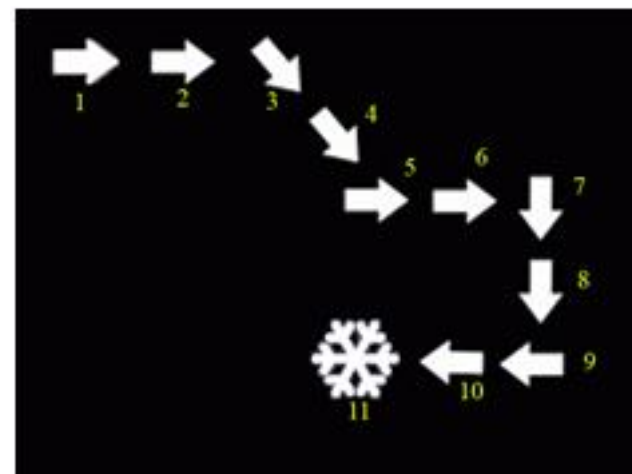
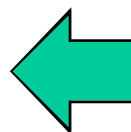
Предобработка
изображения



Сегментация
изображения



Вычисление
признаков
сегментов и
классификация





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