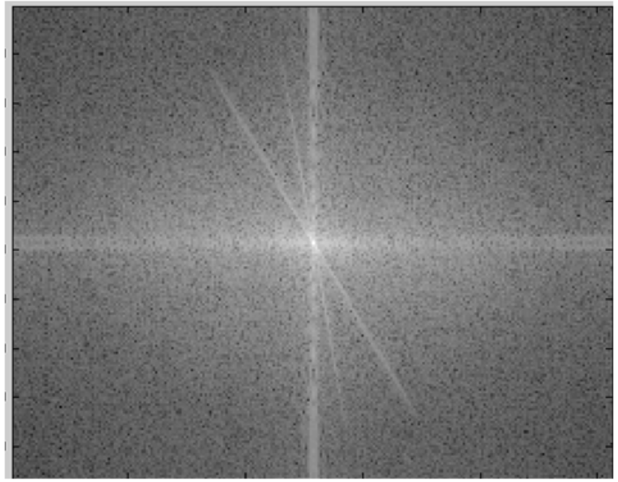
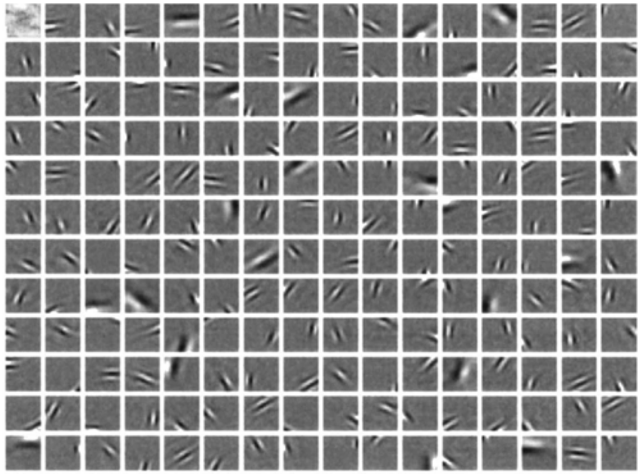
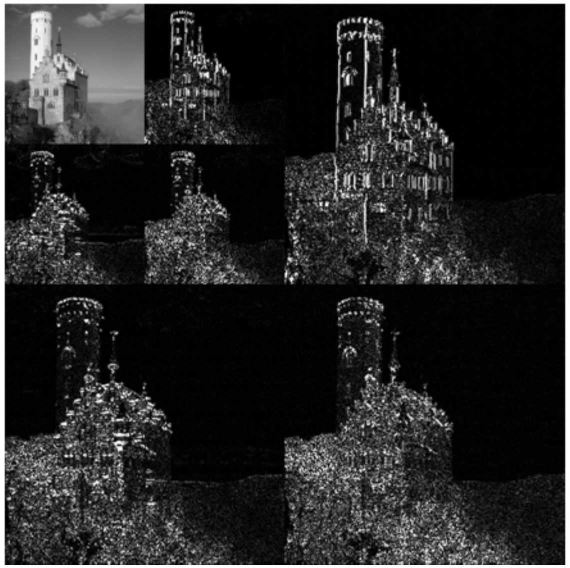
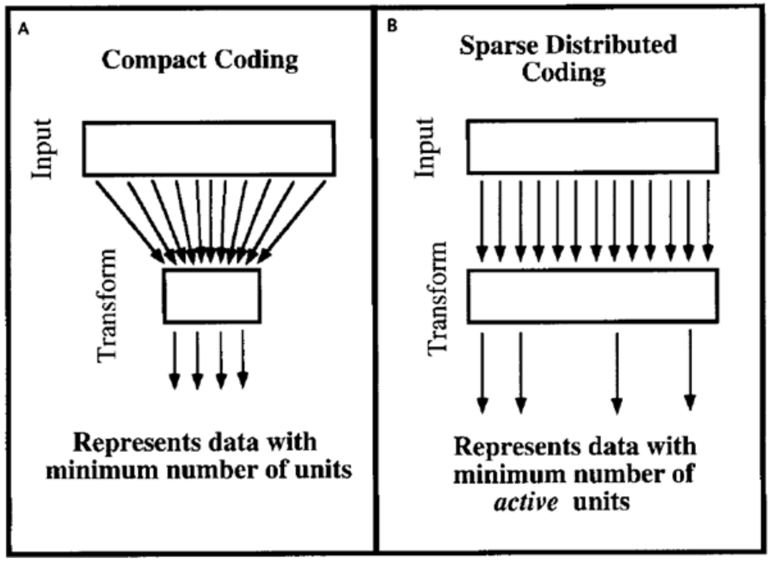
Представление изображений









Many slides adopted from Svetlana Lazebnik, Steve Seitz and Alexey Efros

Общая информация

Этот курс подготовлен и читается при поддержке Microsoft Research

- Страница курса http://courses.graphicon.ru/main/vision

Описание изображения

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

- Простая регулярная однородная сетка
- Никакой структурной информации

Формальная постановка

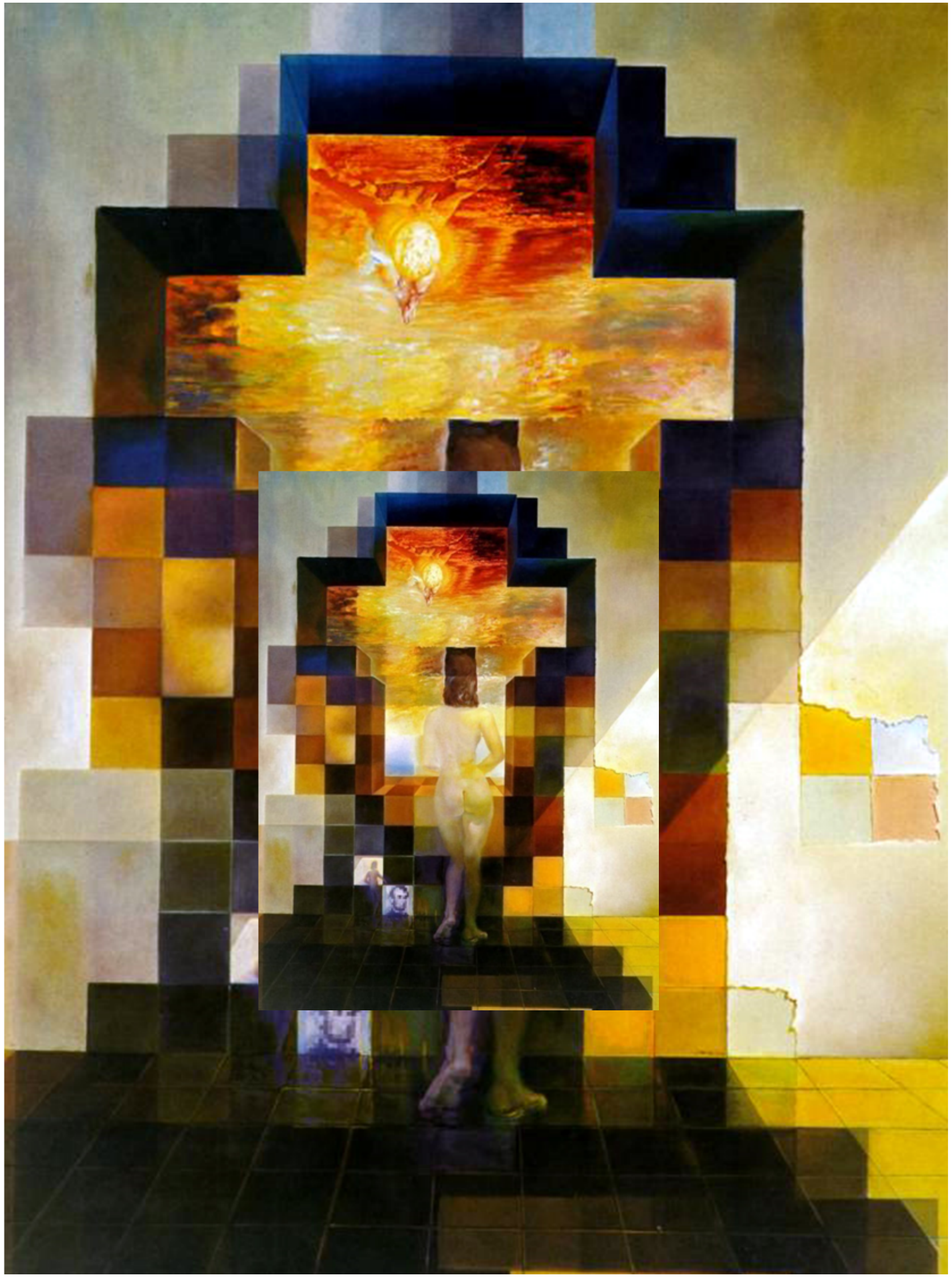
$y \in \Re^m$ - сигнал

$x \in \Re^n$ - представление

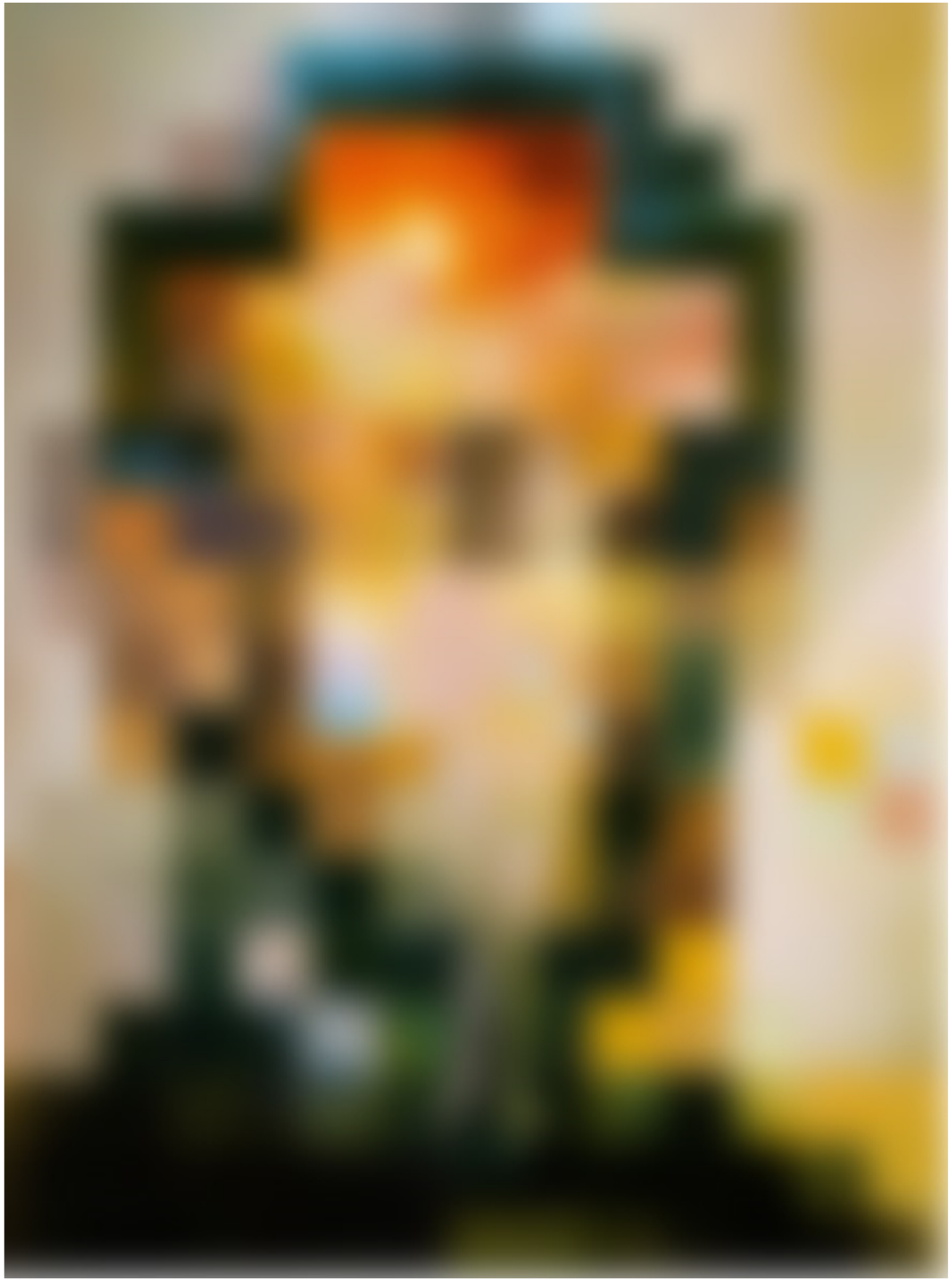
$$D \in \Re^{n \times m}$$
$$y = Dx$$

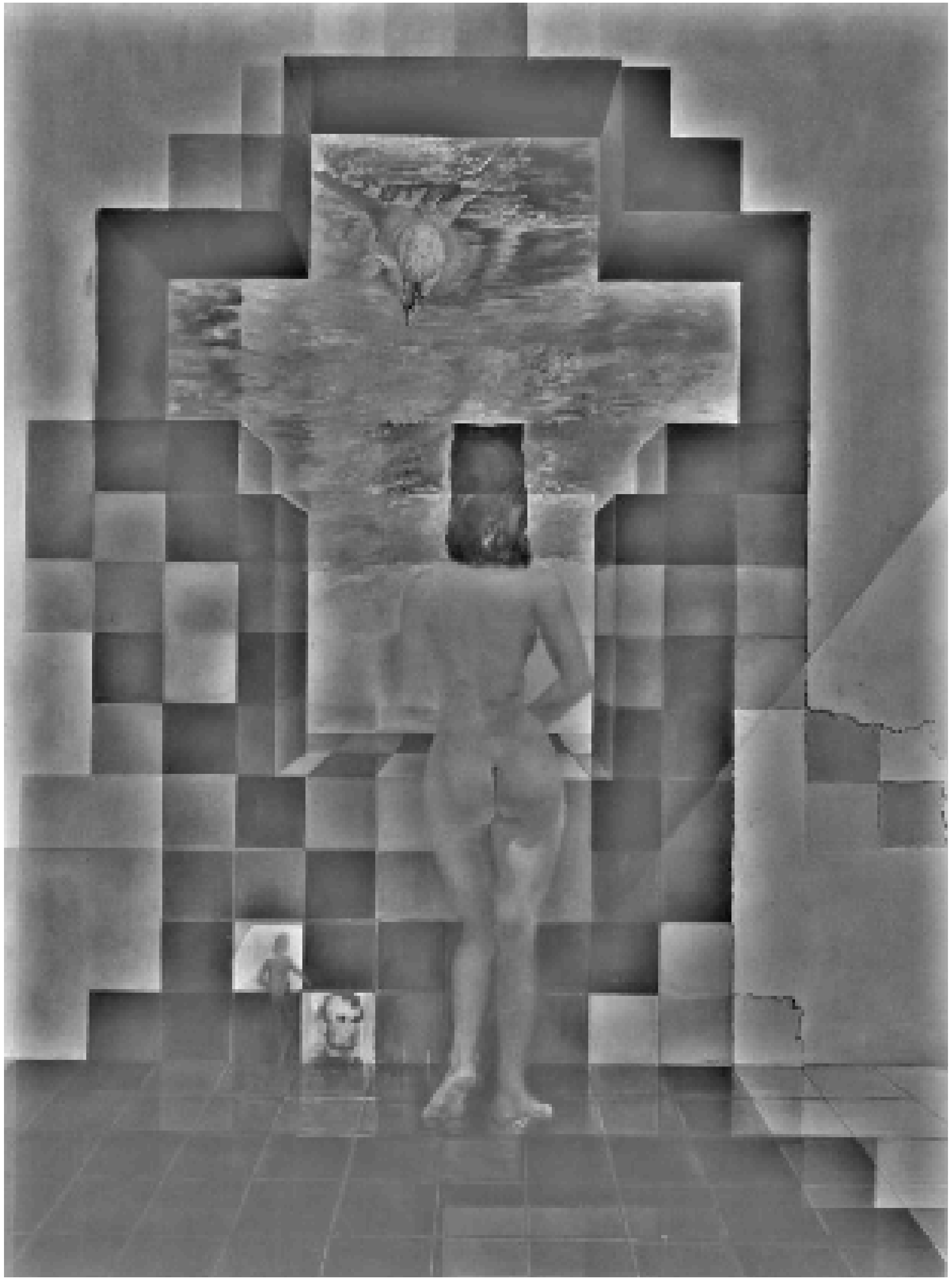
$D = [d_1, d_2, \ldots, d_m]$ $\quad d_i$ - атом (базисная функция)

• В случае изображения, y – обычно фрагмент изображения, чаще всего прямоугольной формы.
• n может быть как равно m, так и нет!
• Если D – ортогональная матрица, тогда преобразование в обе стороны выполняется просто



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 переодическая 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:

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

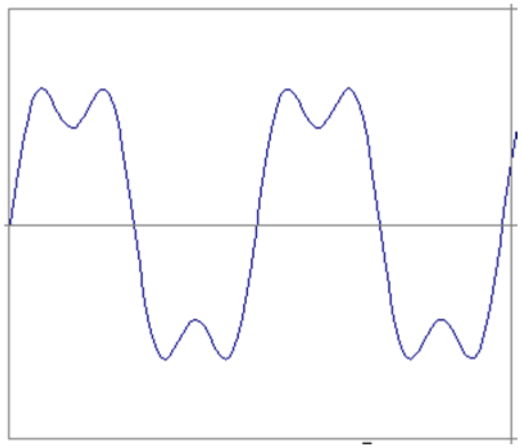
Быстрое преобразование Фурье

- 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 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.

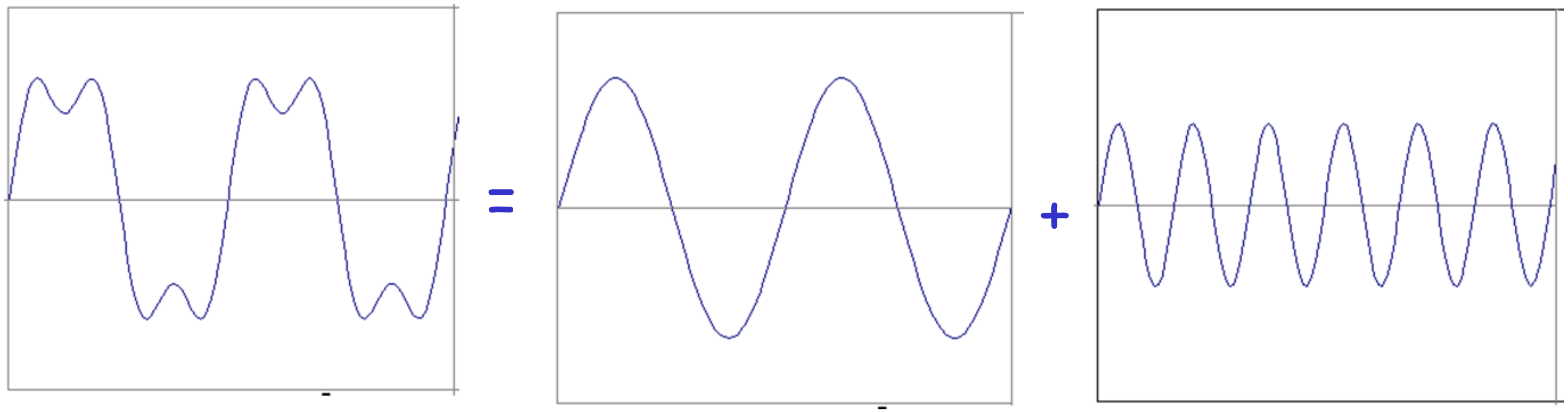
Пример

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



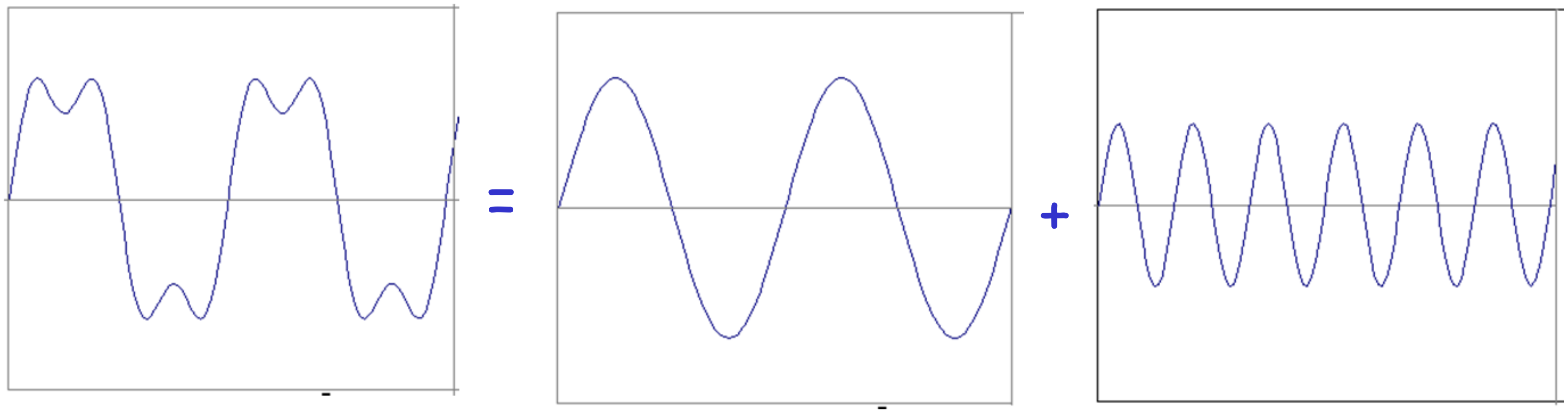
Пример

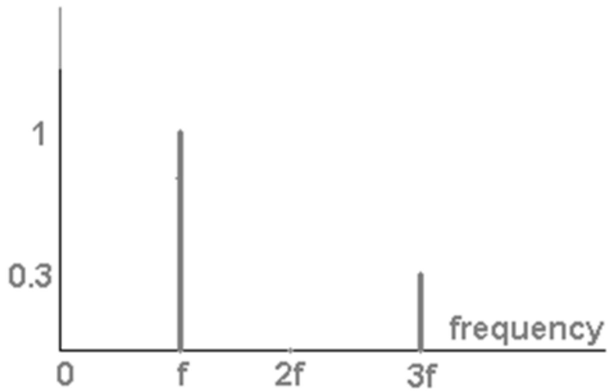
- $g(t) = \sin(2pf\ t) + (1/3)\sin(2p(3f)\ t)$



Частотный спектр

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





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

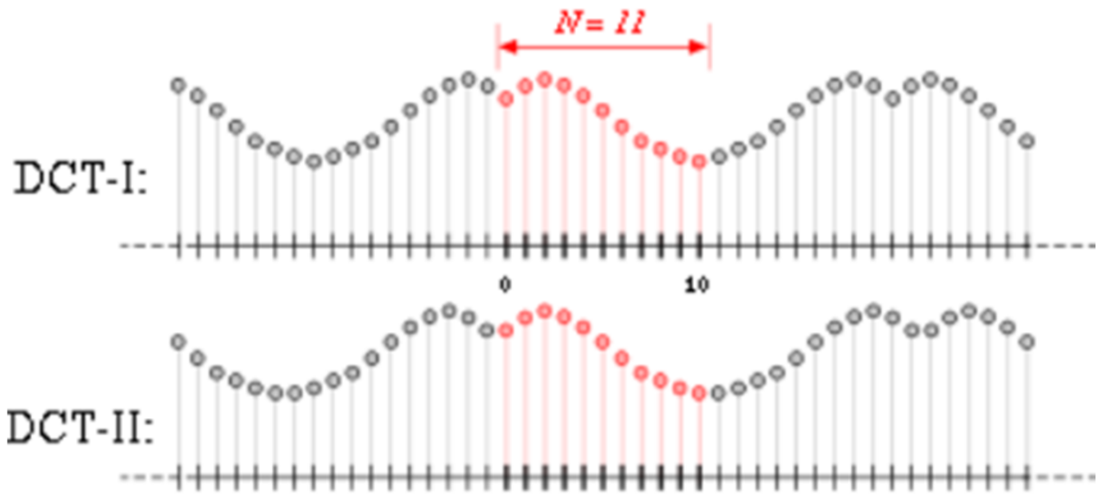
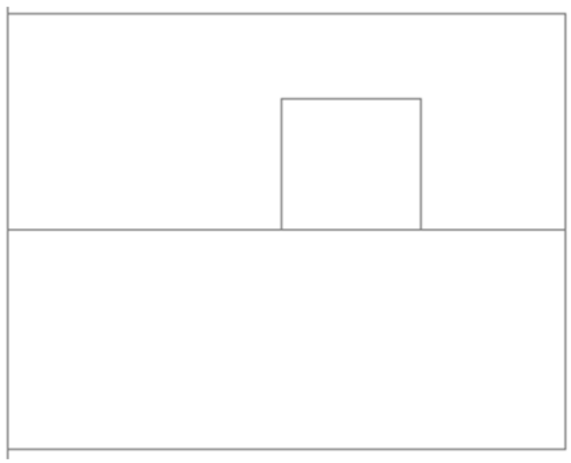
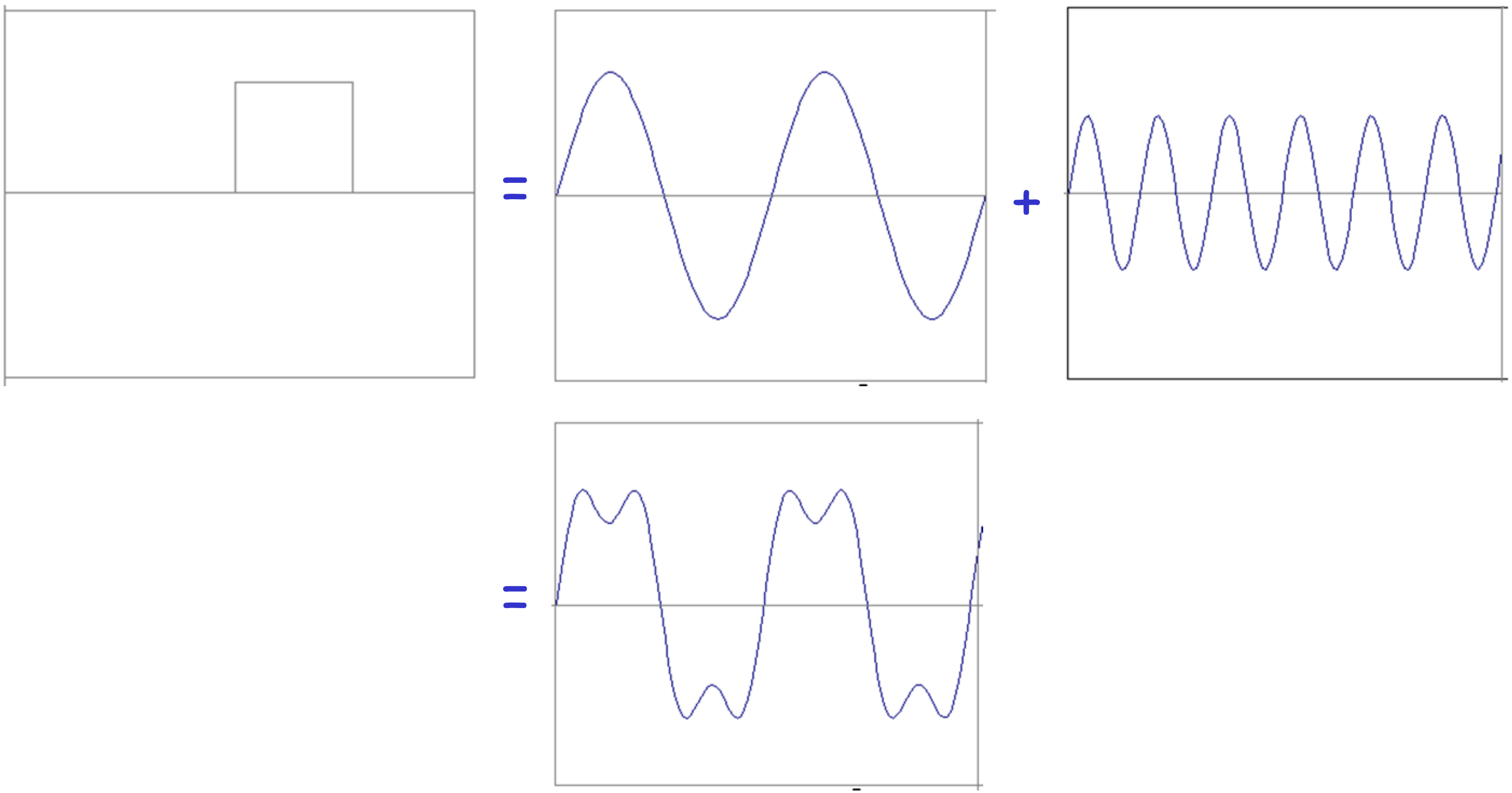


Image source: wiki

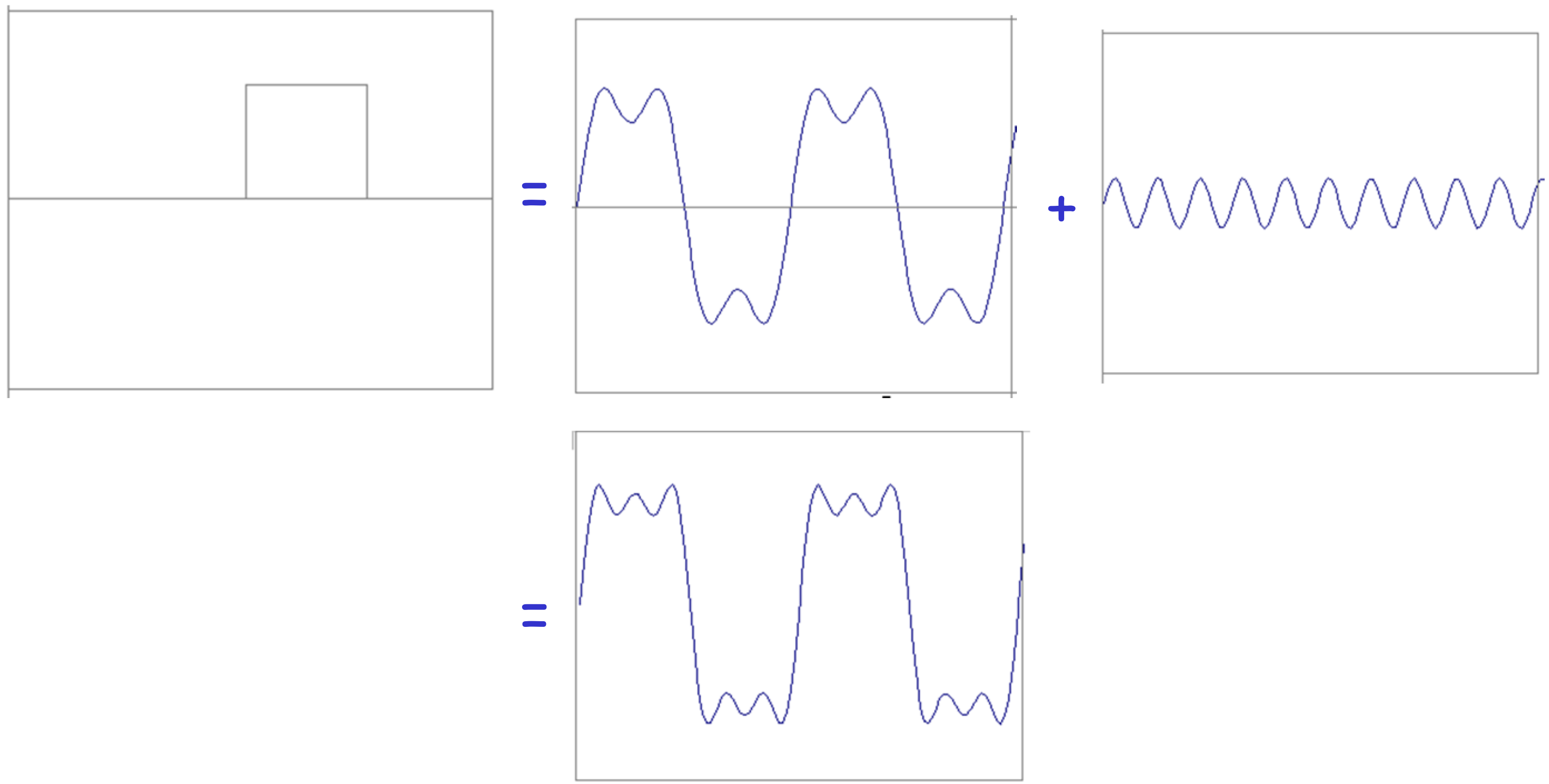
Прямоугольный сигнал



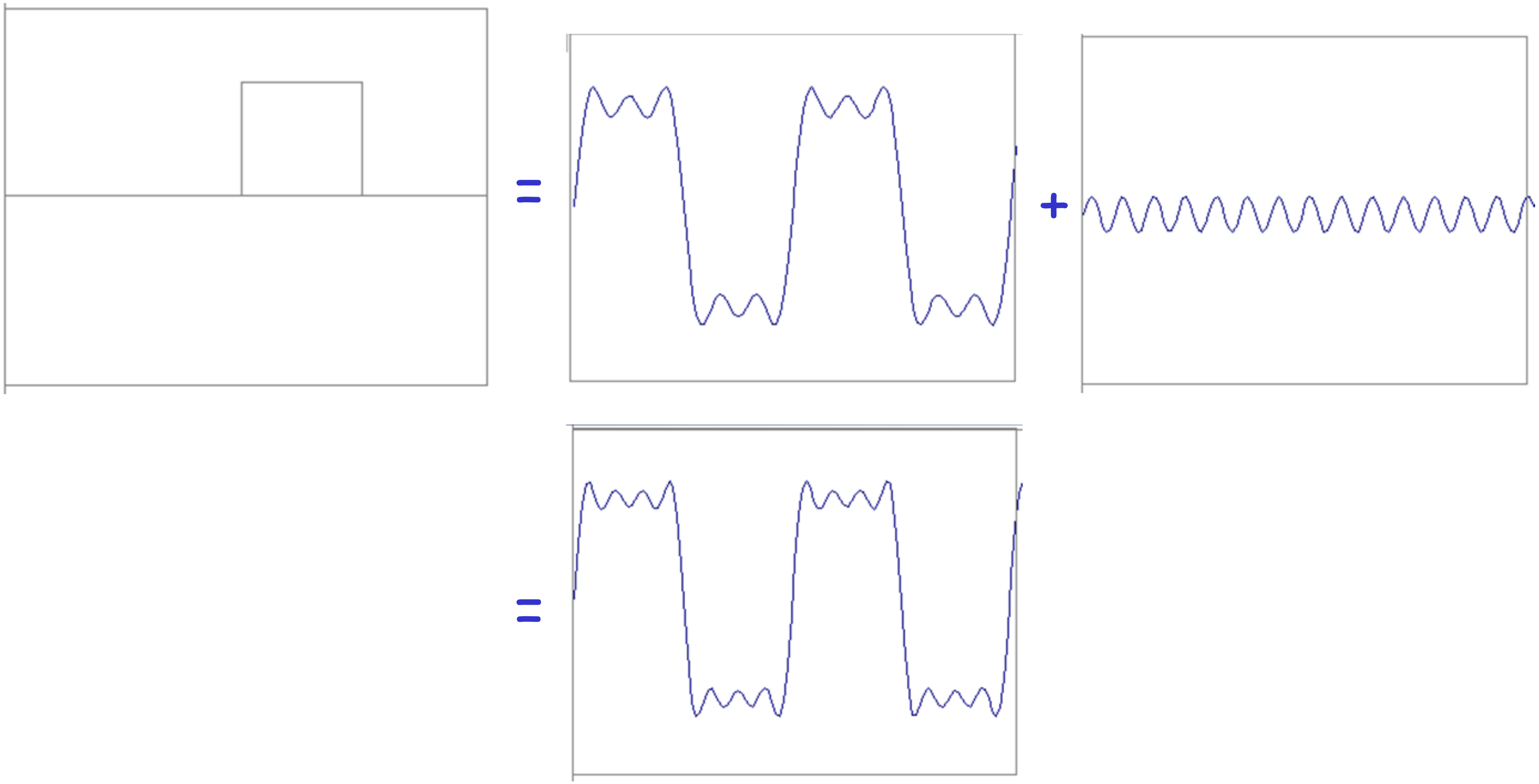
Прямоугольный сигнал



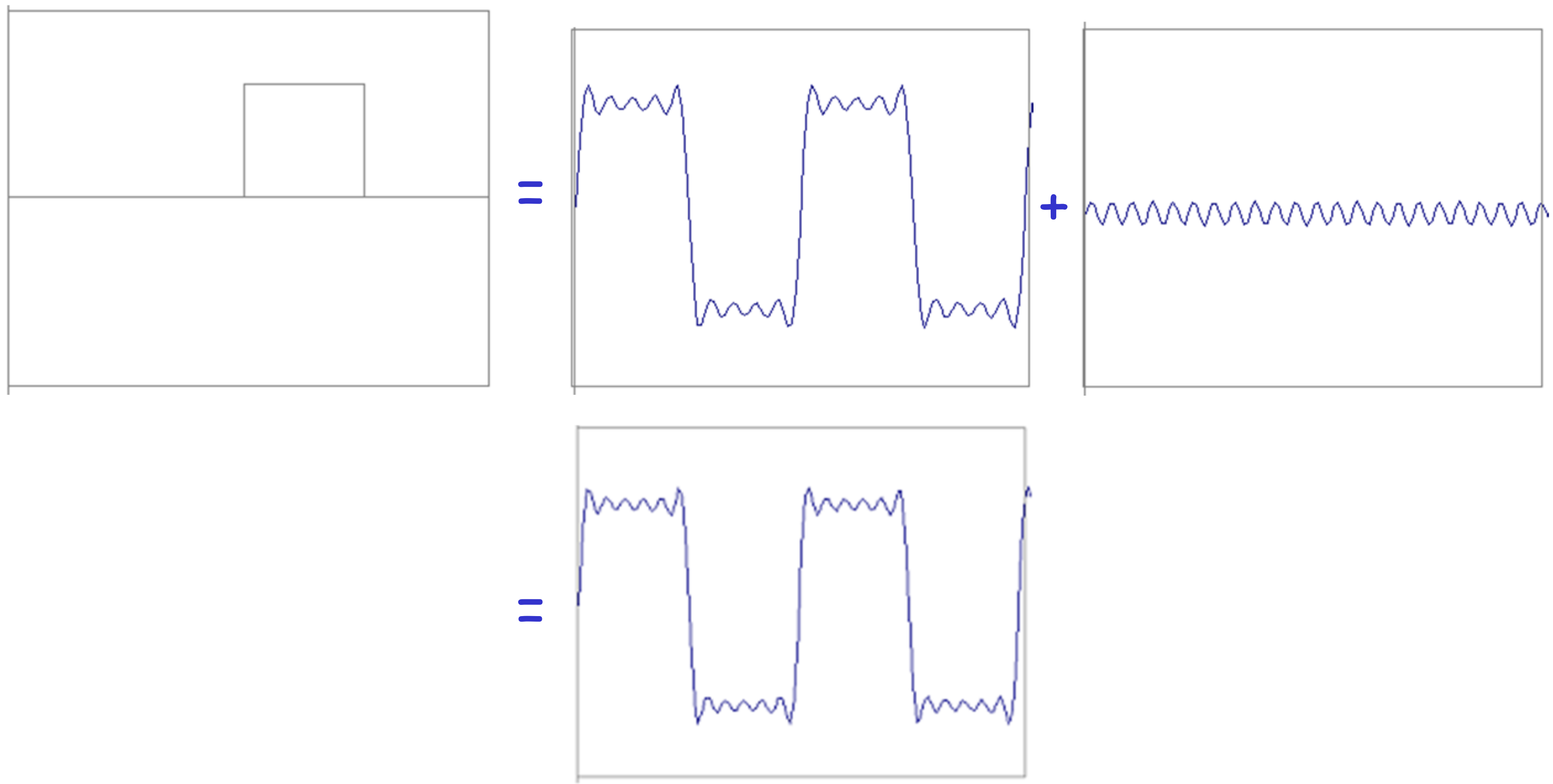
Прямоугольный сигнал



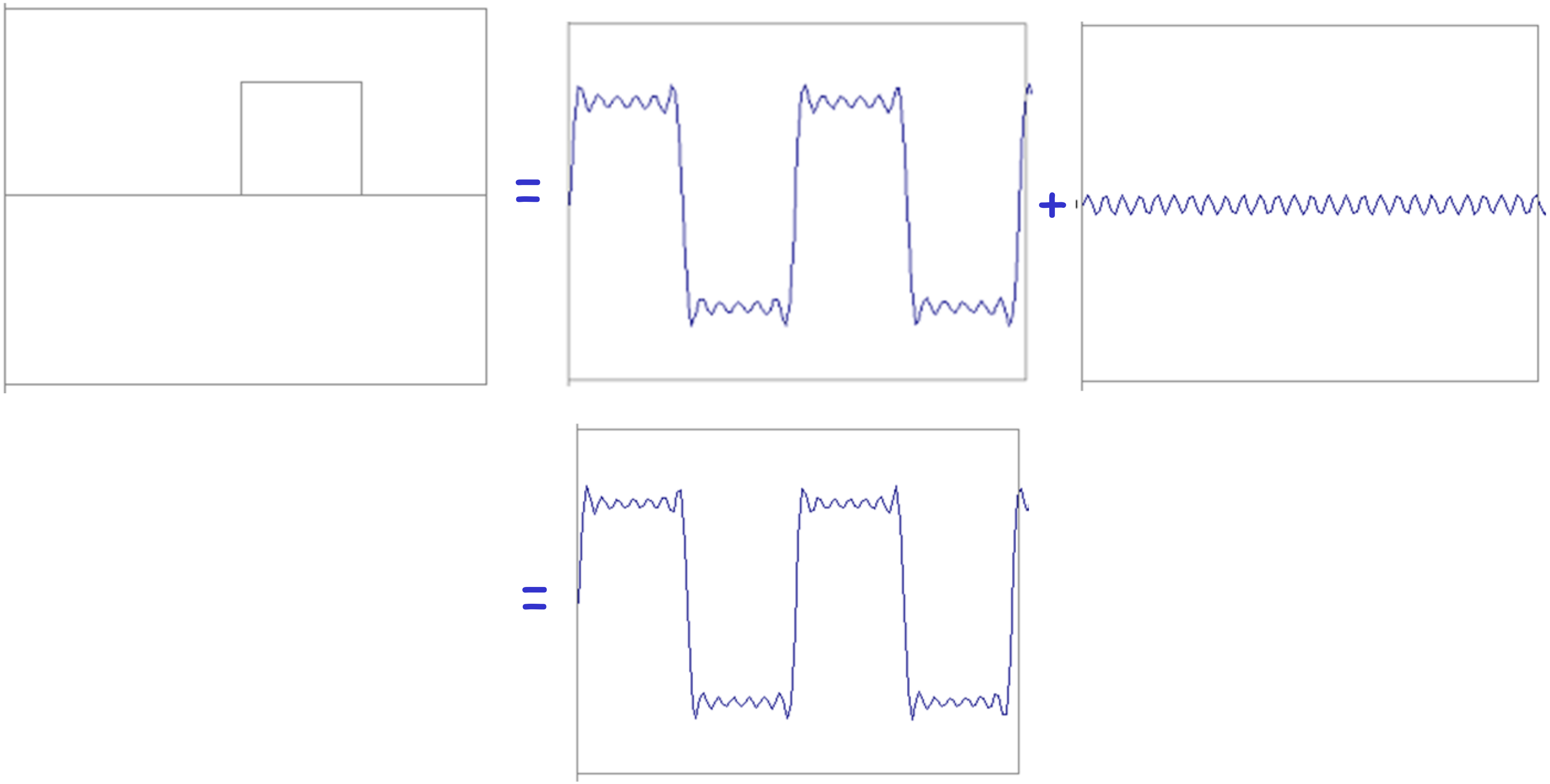
Прямоугольный сигнал



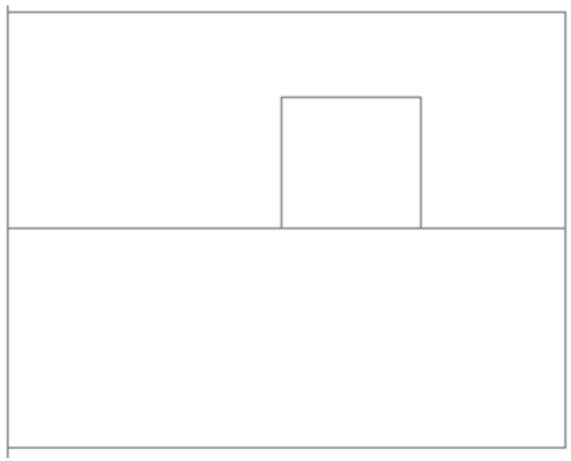
Прямоугольный сигнал



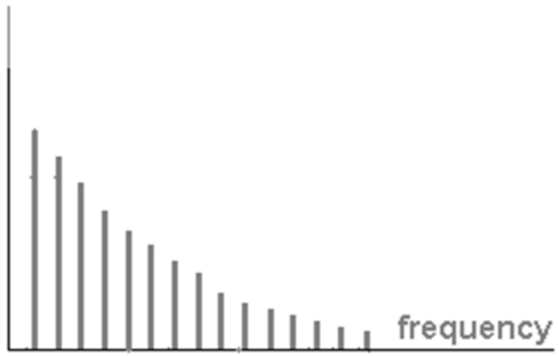
Прямоугольный сигнал



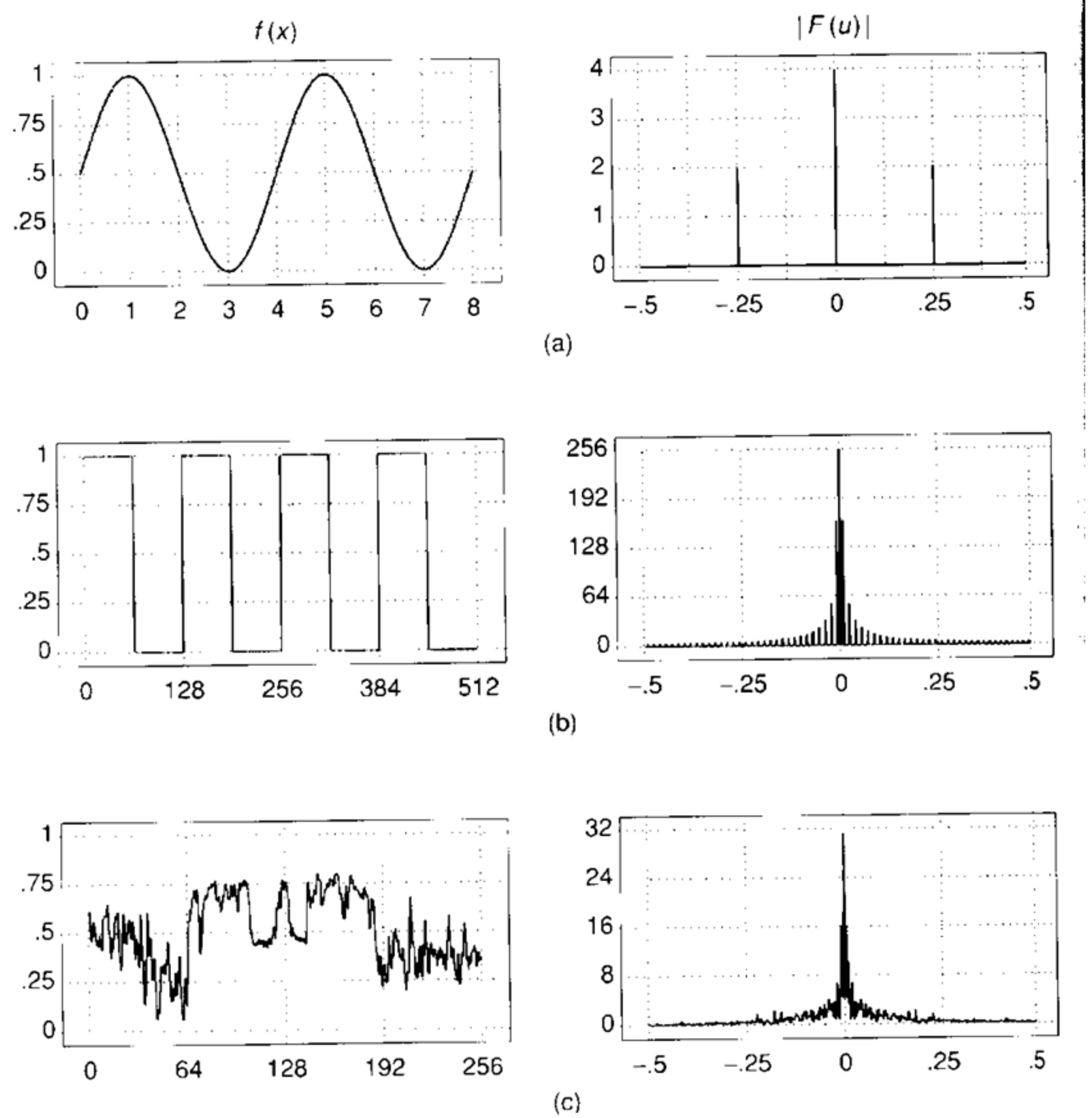
Прямоугольный сигнал



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



Спектр частот



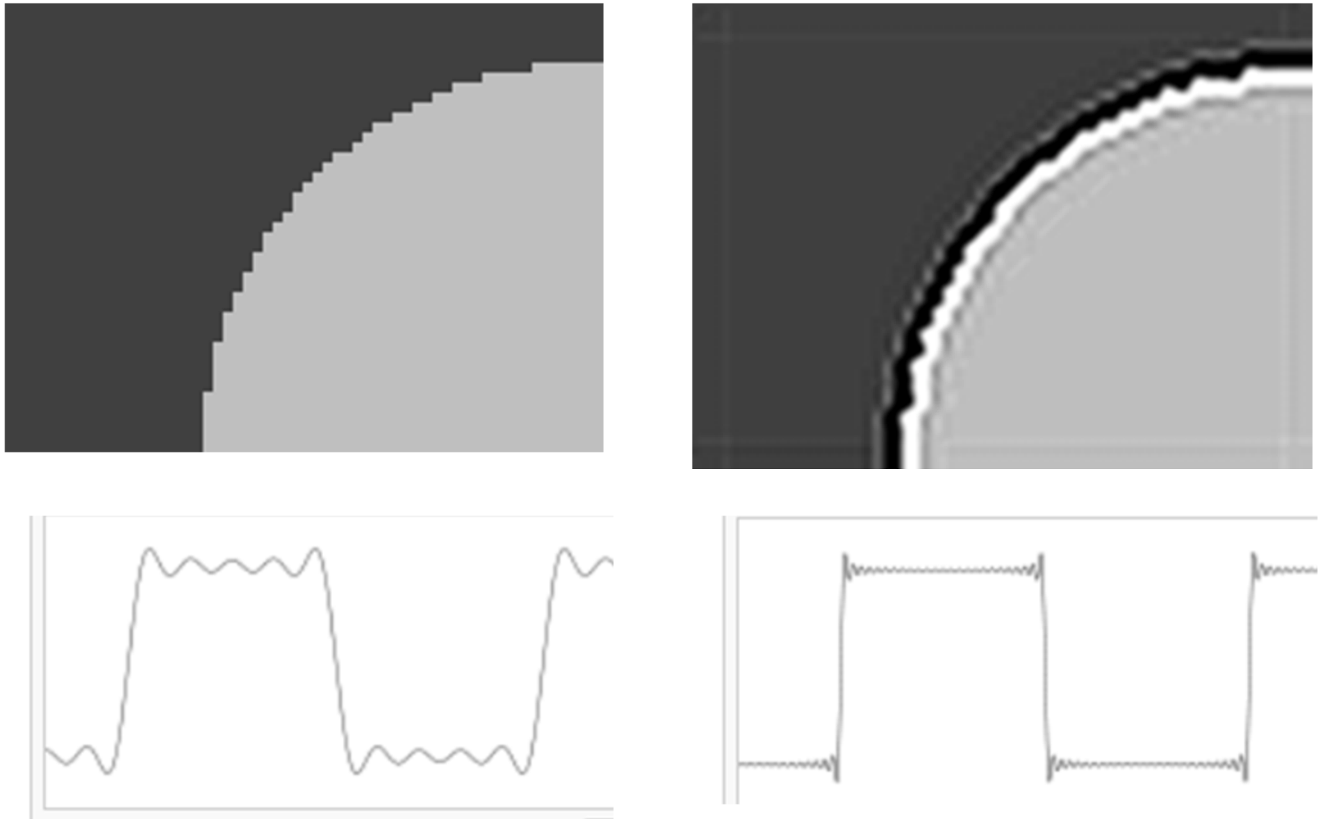
Повышение резкости

Ядро свертки

$$\frac{1}{10}\cdot\begin{Vmatrix}-1 & -2 & -1\\ -2 & 22 & -2\\ -1 & -2 & -1\end{Vmatrix}$$

Обратите внимание на ореолы вокруг границ

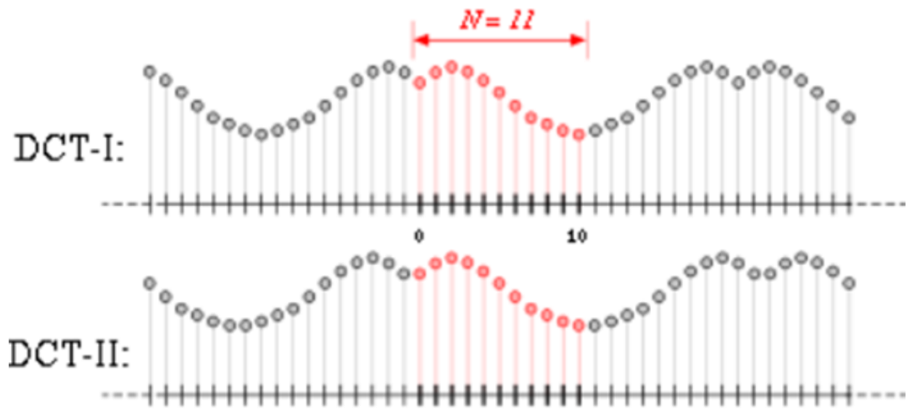
Ringing

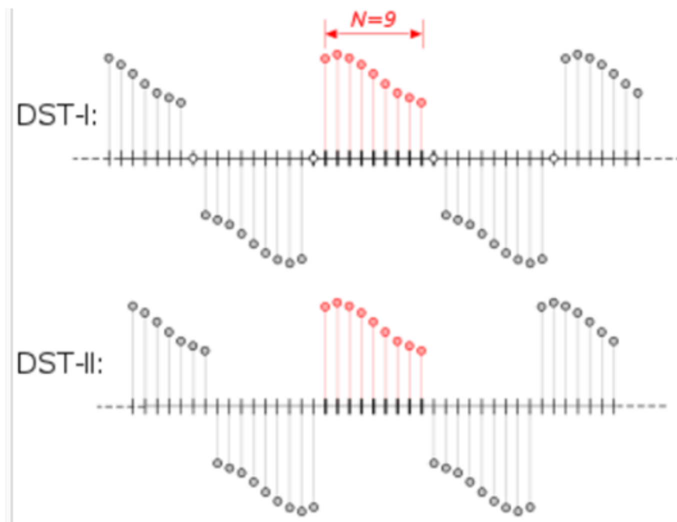


- Gibb's Phenomenon
- Частичная сумма ряда Фурье отличается от сигнала на резких границах
- Постоянно с этим борются

Свойства

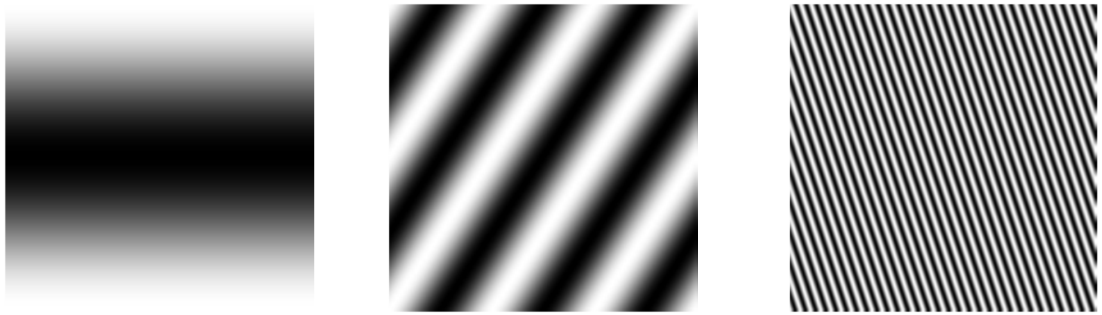
- Разрывы функции приводят к тому, что требуется больше слагаемых для достижения точности
- sin() – нечётная функция, поэтому продление должно быть нечётной функцией
- Поскольку у реального сигнала значение на конце и в начале сигнала обычно разное, то продление почти всегда с разрывом
- Для реальных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





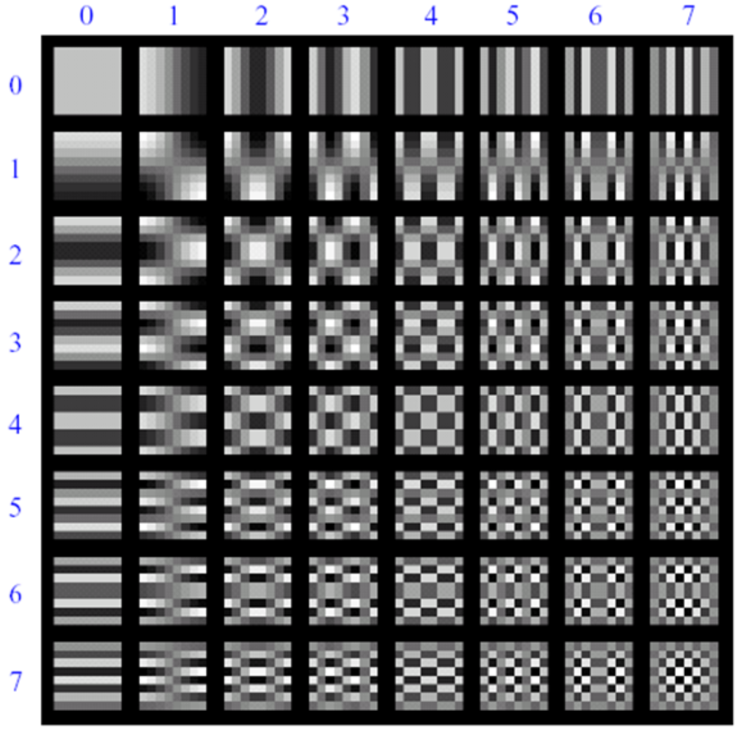
2D преобразование

- 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)

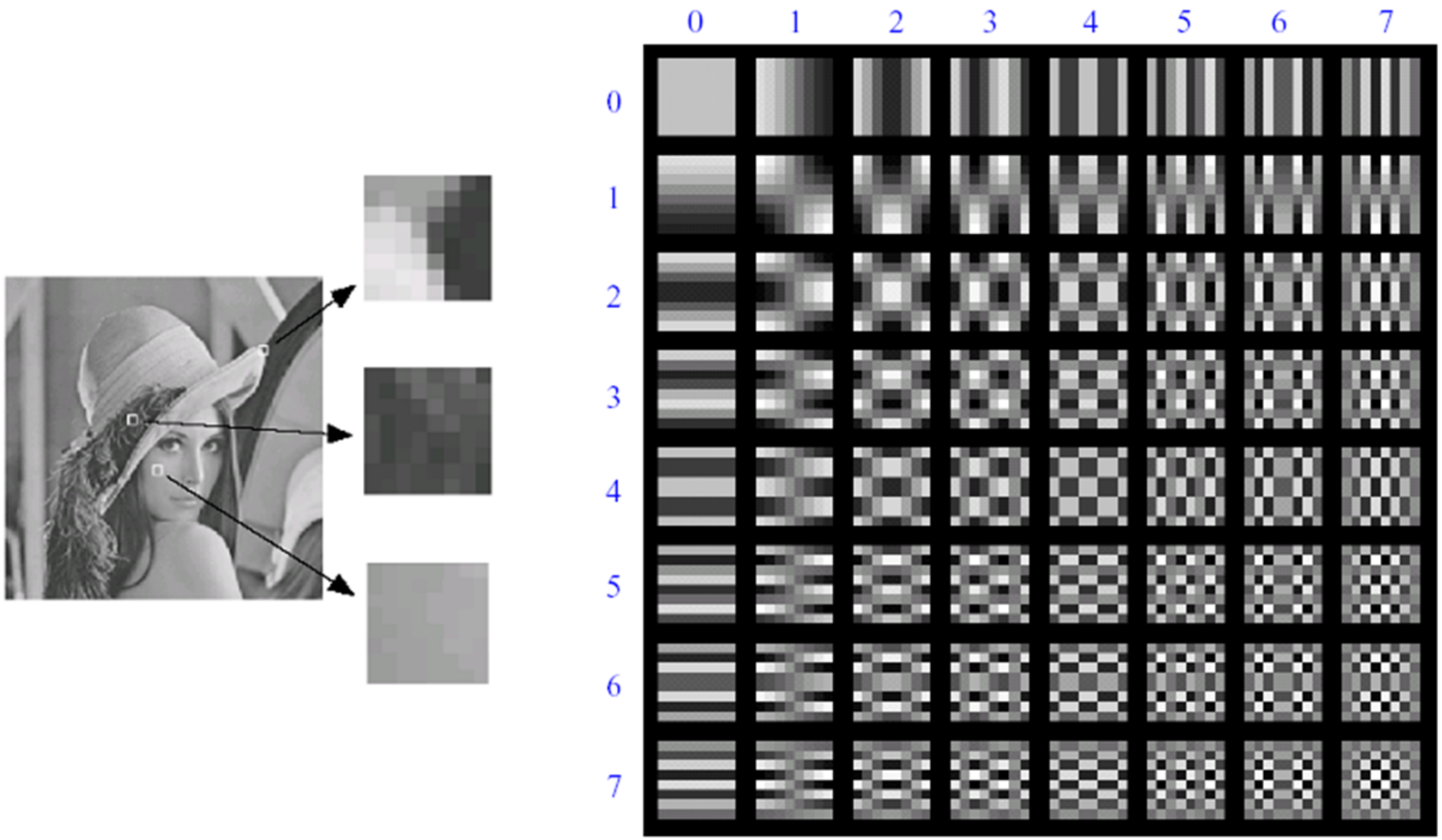
Буква А размером 8x8 пикселей

Реконструкция буквы после ДКП

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

Image source: wiki

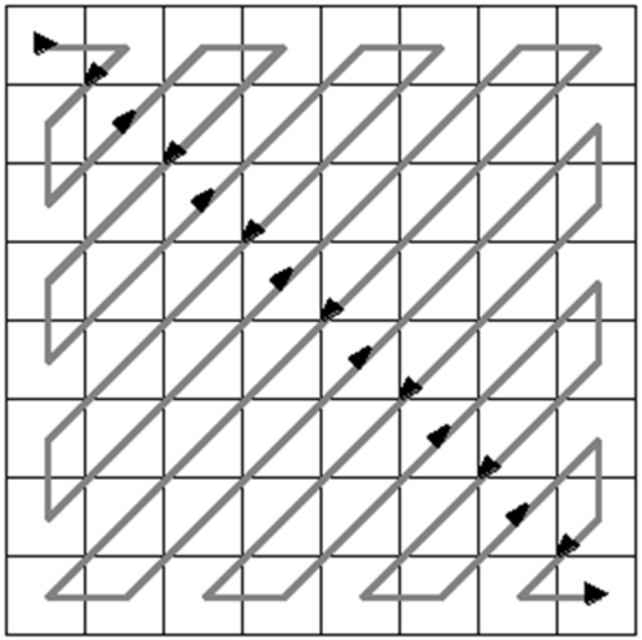
Сжатие с потерями (JPEG)



- 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:

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

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

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

- 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

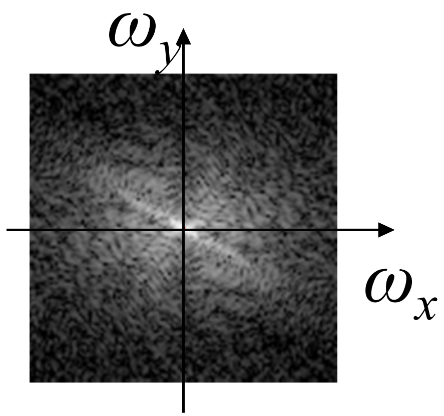
Пример сжатия

89k

12k

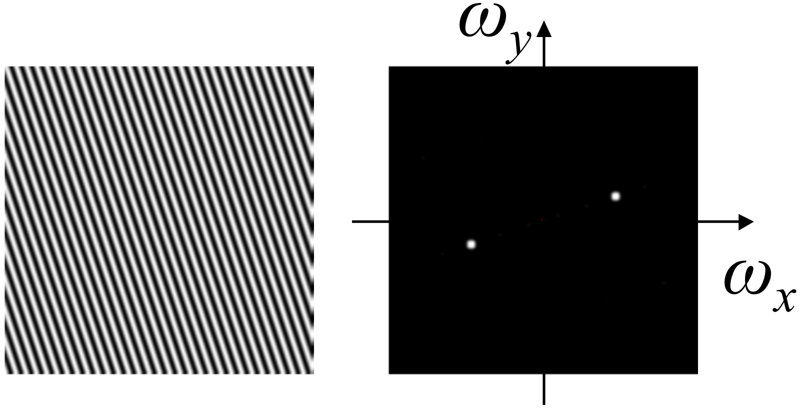
Спектральный анализ

- 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.

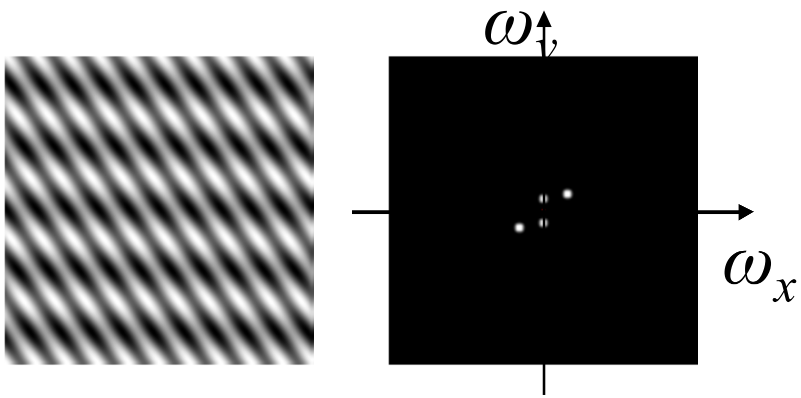


Спектральный анализ

- Примеры изображений и их спектров



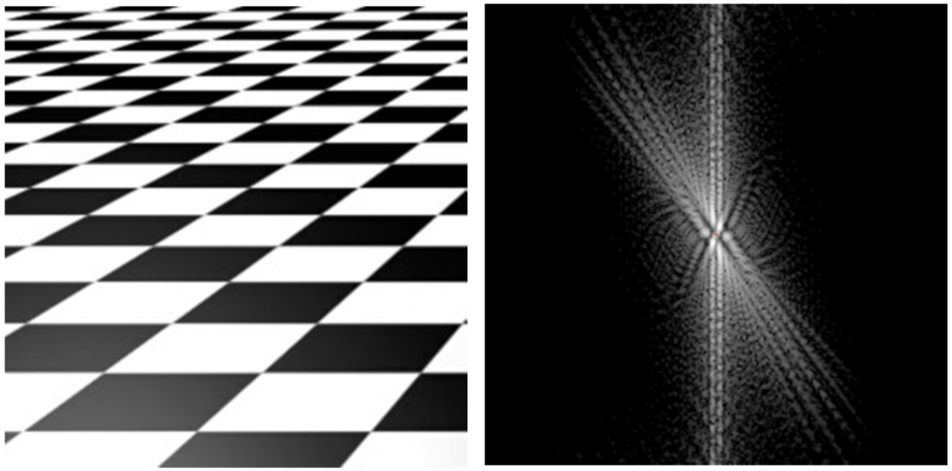
Видно, что спектр одной синусоиды – это точка
(не забываем про симметричное отражение спектра)



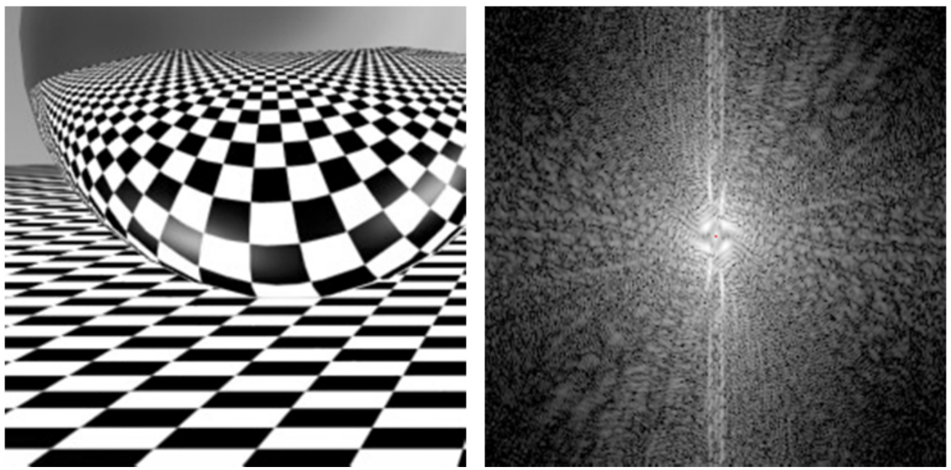
Две синусоиды – две точки

Спектральный анализ

- Примеры изображений и их спектров

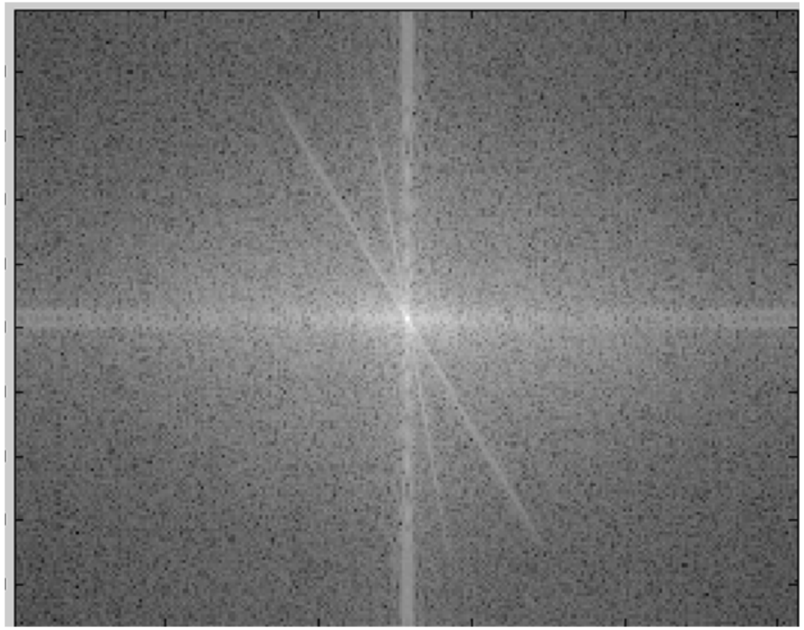


По спектру прослеживаются преобладающие направления в исходной картинке

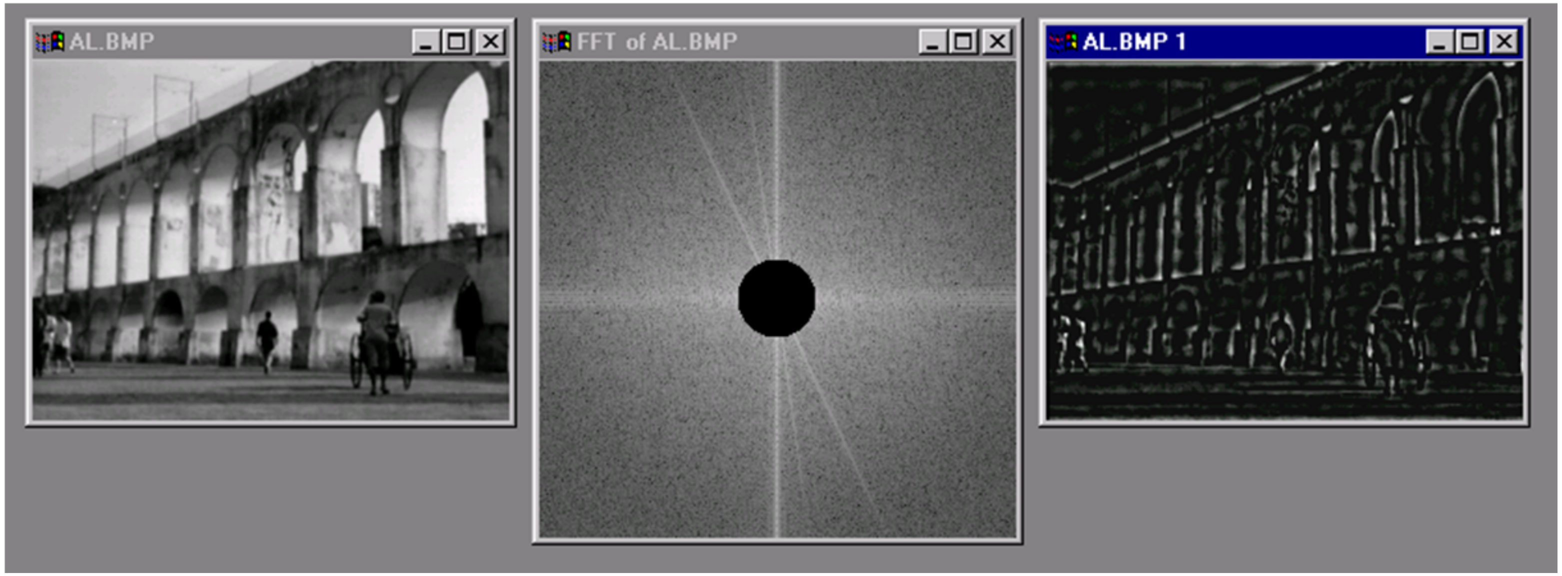


Много высоких частот в спектре – много мелких деталей в исходном изображении

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

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

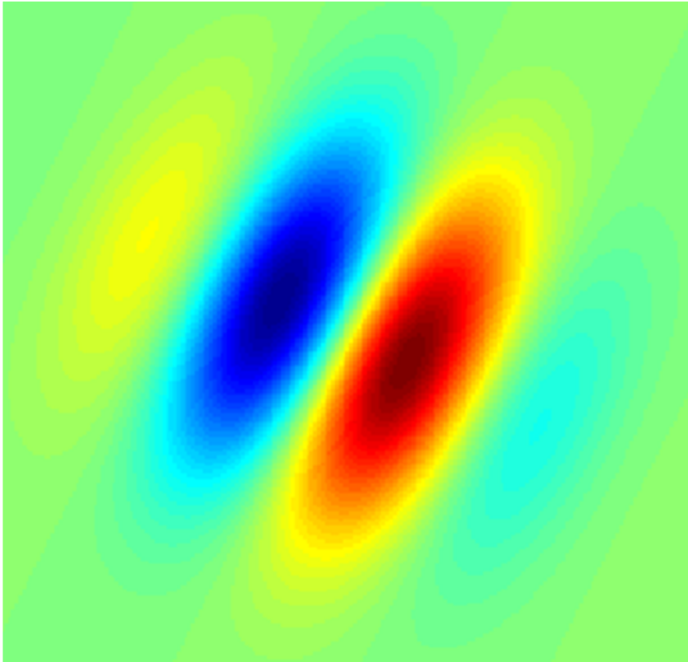
- Свёртка в пространстве эквивалентна произведению в частотном диапазоне
- Можно существенно ускорить многие операции свёртки!

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

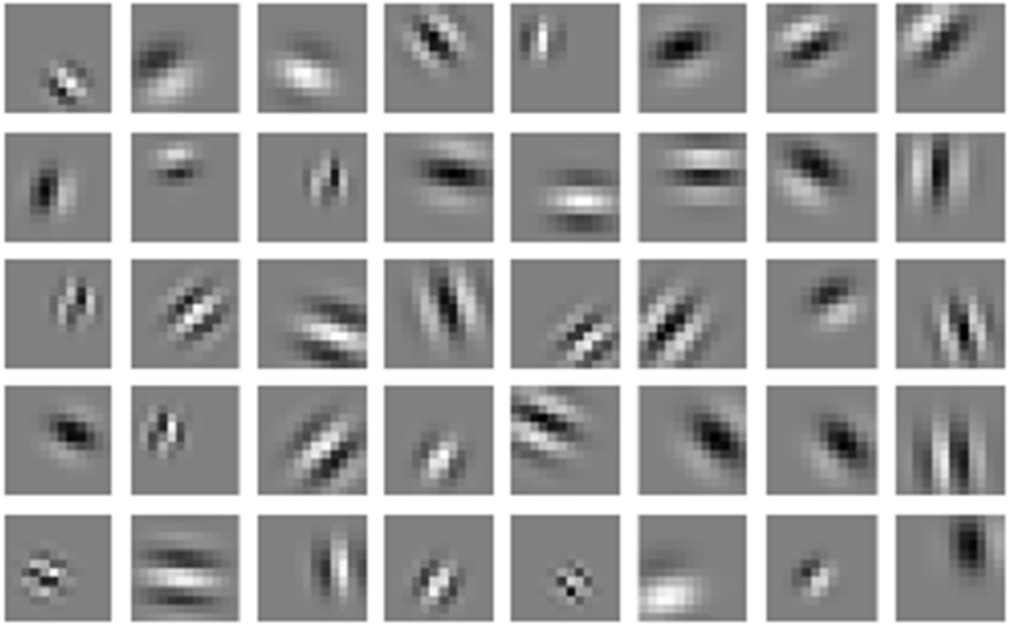
σ - сигма гауссиана

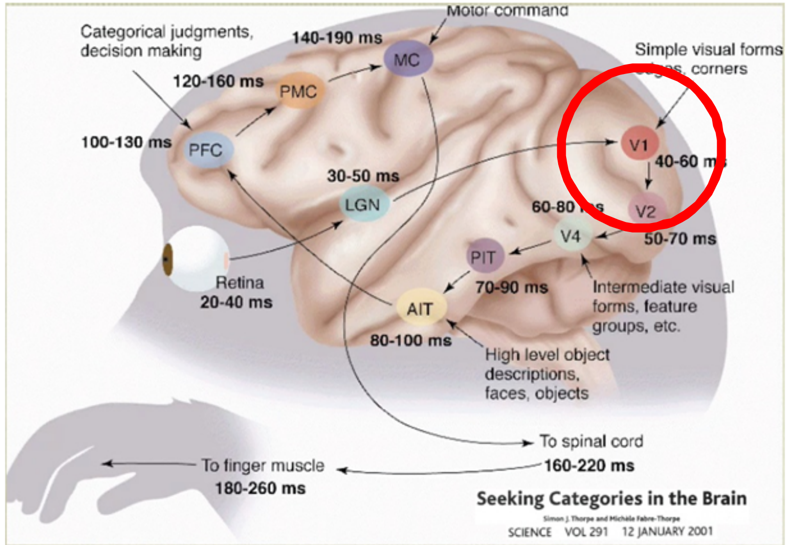
γ - соотношение размеров (aspect ratio), «эллиптичность фильтра»

ψ - сдвиг фазы

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

Связь со зрением человека



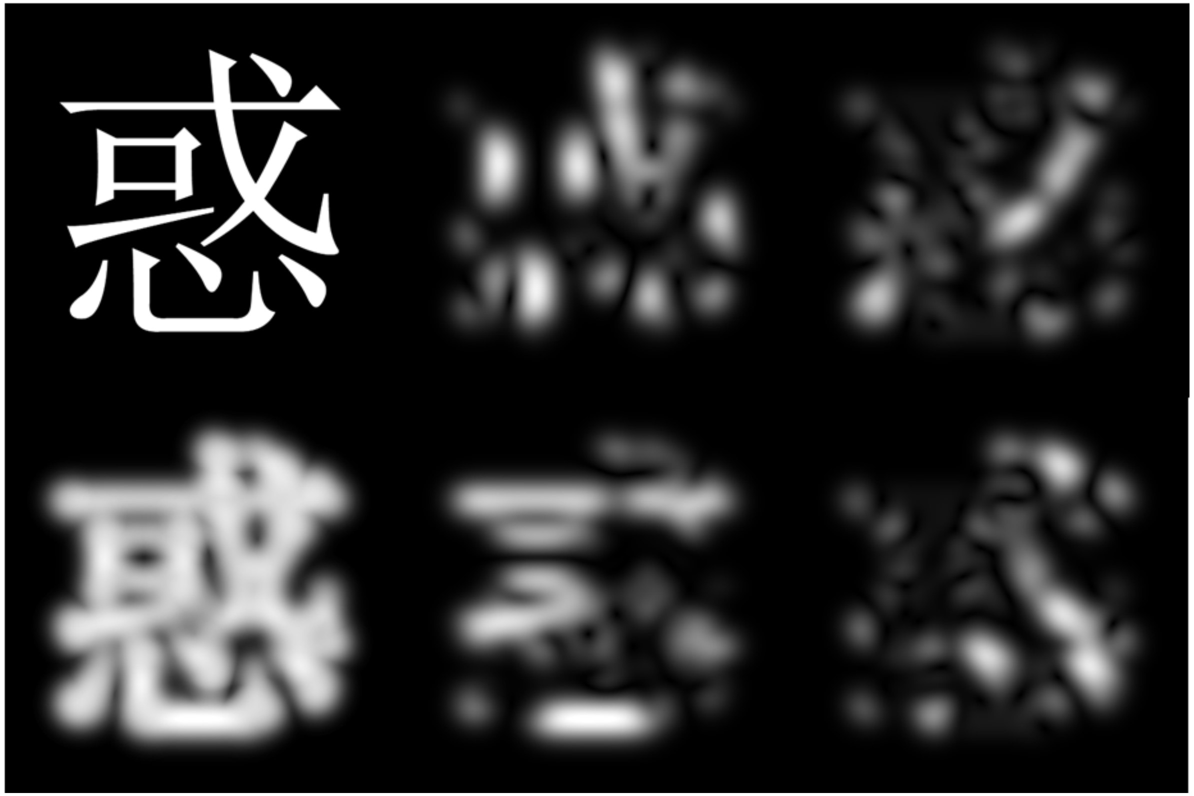


- Похожи на форму рецептивных полей простых клеток (simple cells) в визуальной коре мозга человека

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, “Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters,” *J. Opt. Soc. Am. A, vol. 2, no. 7, pp. 1160–1169, 1985.*

Поиск краёв с помощью Габора

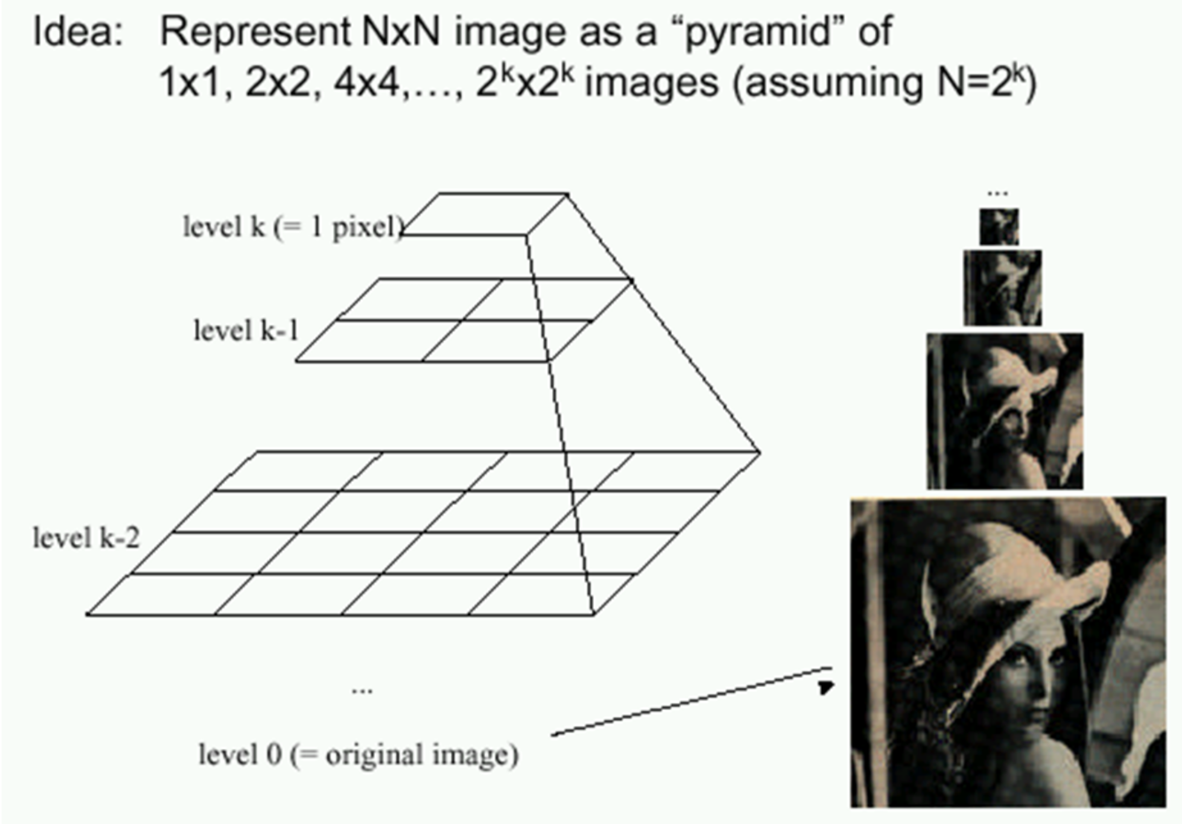


- Слева вверху – иероглиф
- 4 справа – применение фильтров Габора с ориентациями 0, 45, 90 и 135 градусов
- Слева внизу – совмещение результатов фильтрации

Банки фильтров

- Набор фильтров разного масштаба и ориентации образует «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.

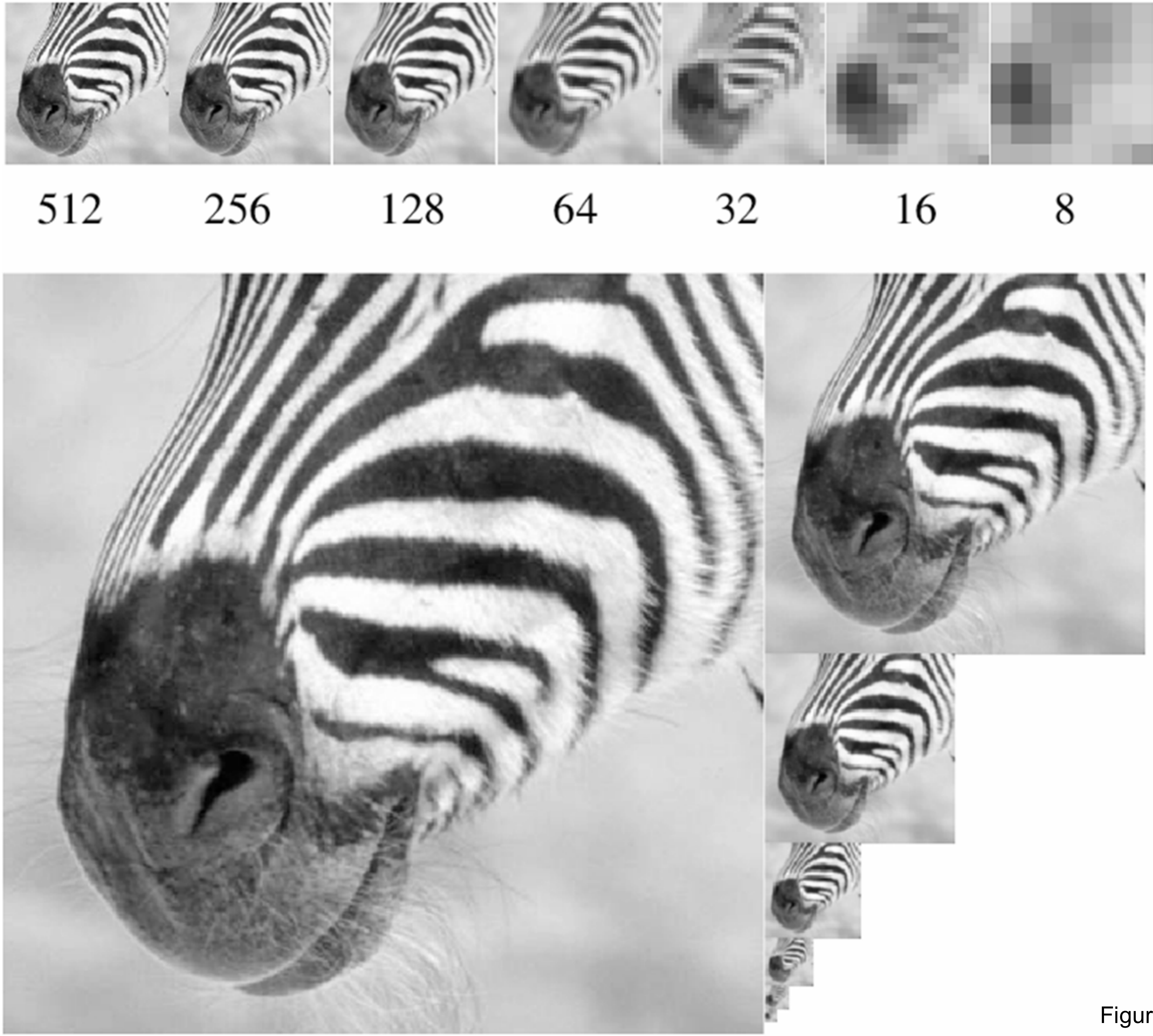
Пирамиды изображений



- Известна как Пирамида Гауссиан
- В компьютерной графике – “mip map” [Williams, 1983]

P. Burt and E. Adelson, “The Laplacian pyramid as a compact image code,” *IEEE Trans. Commun., vol. 31, no. 4, pp. 532–540, 1983.*

Slide by Steve Seitz



На высшем уровне полоса в пиксель – волос, на среднем – полоска, на нижнем - нос

Figure from David Forsyth

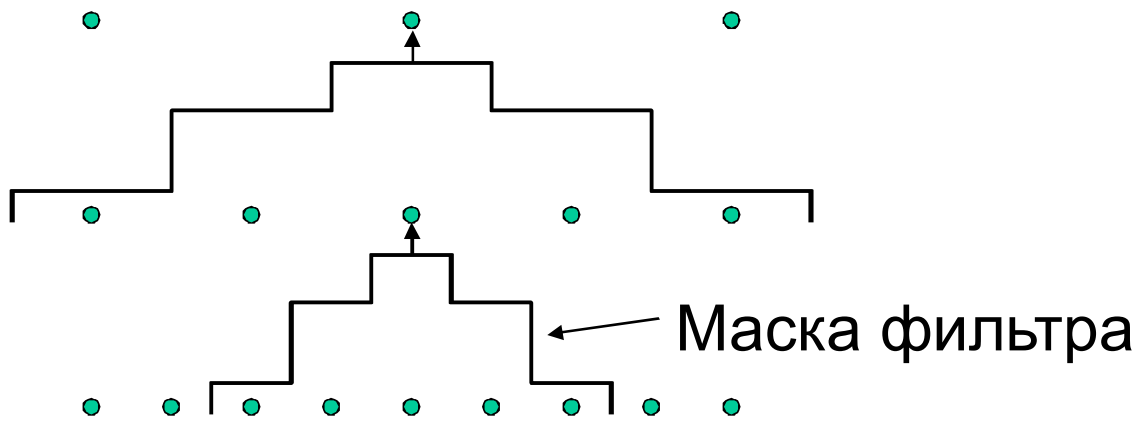
Применение пирамид

- Улучшение сопоставления шаблонов
  - Поиск сдвига
    - Классическая стратегия последовательного уточнения
    - «coarse-to-fine strategy»
  - Поиск по масштабу (размеру)
    - Сравнение шаблонов
    - Поиск объектов с разным масштабом
      - » Поиск лиц
      - » Ключевых точек

- Основа «кратномасштабного анализа» изображений
  - Multiresolution methods

- Одна из предпосылок вейвлет-анализа

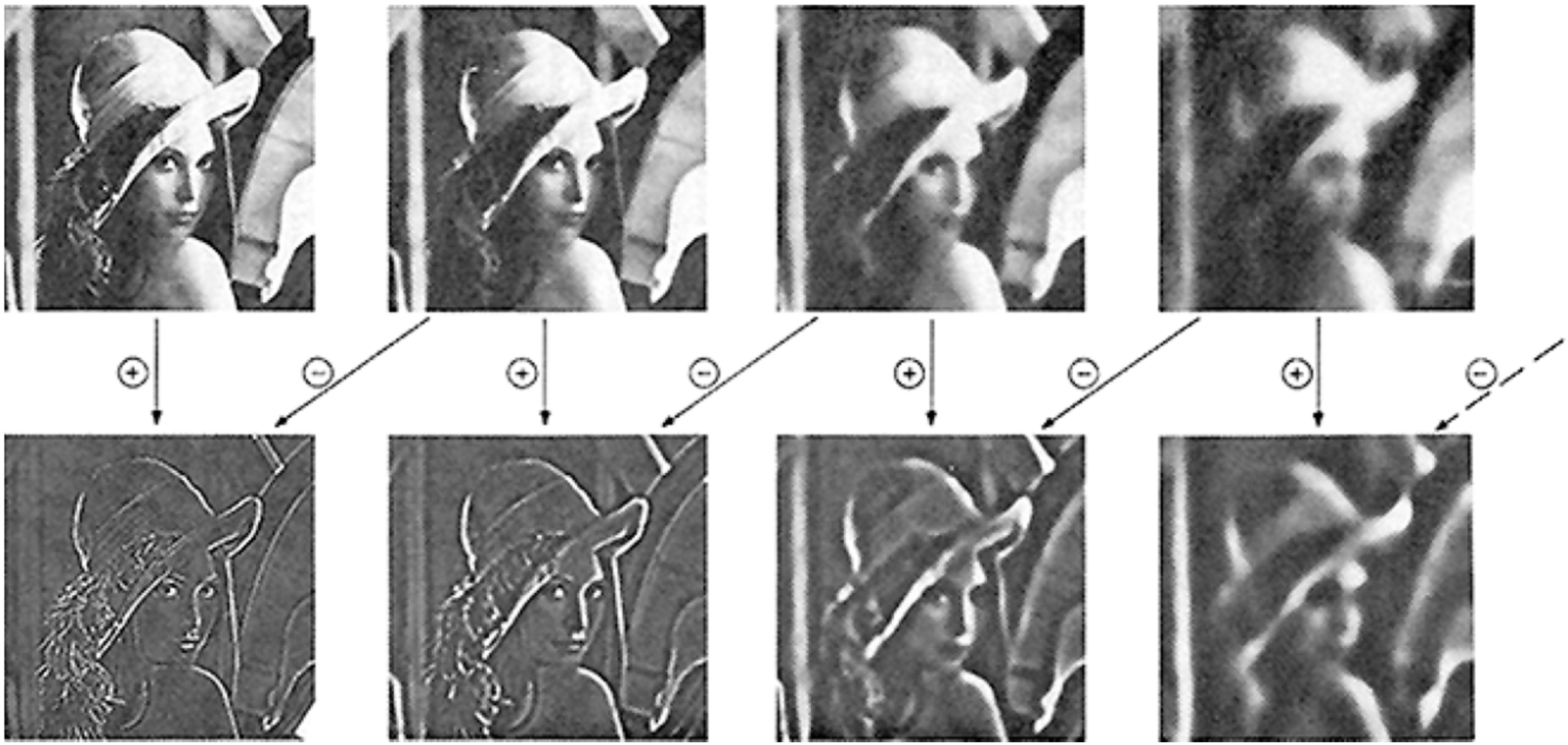
Построение пирамиды



- Повторяем
  - Фильтруем
  - Сэмплируем
- До достижения минимального разрешения
  - Можем задать необходимое количество уровней (пр.: 3)
- EXPAND
  - Обратное преобразование – повышение разрешения
  - Интерполяция промежуточных значений с помощью того же фильтра

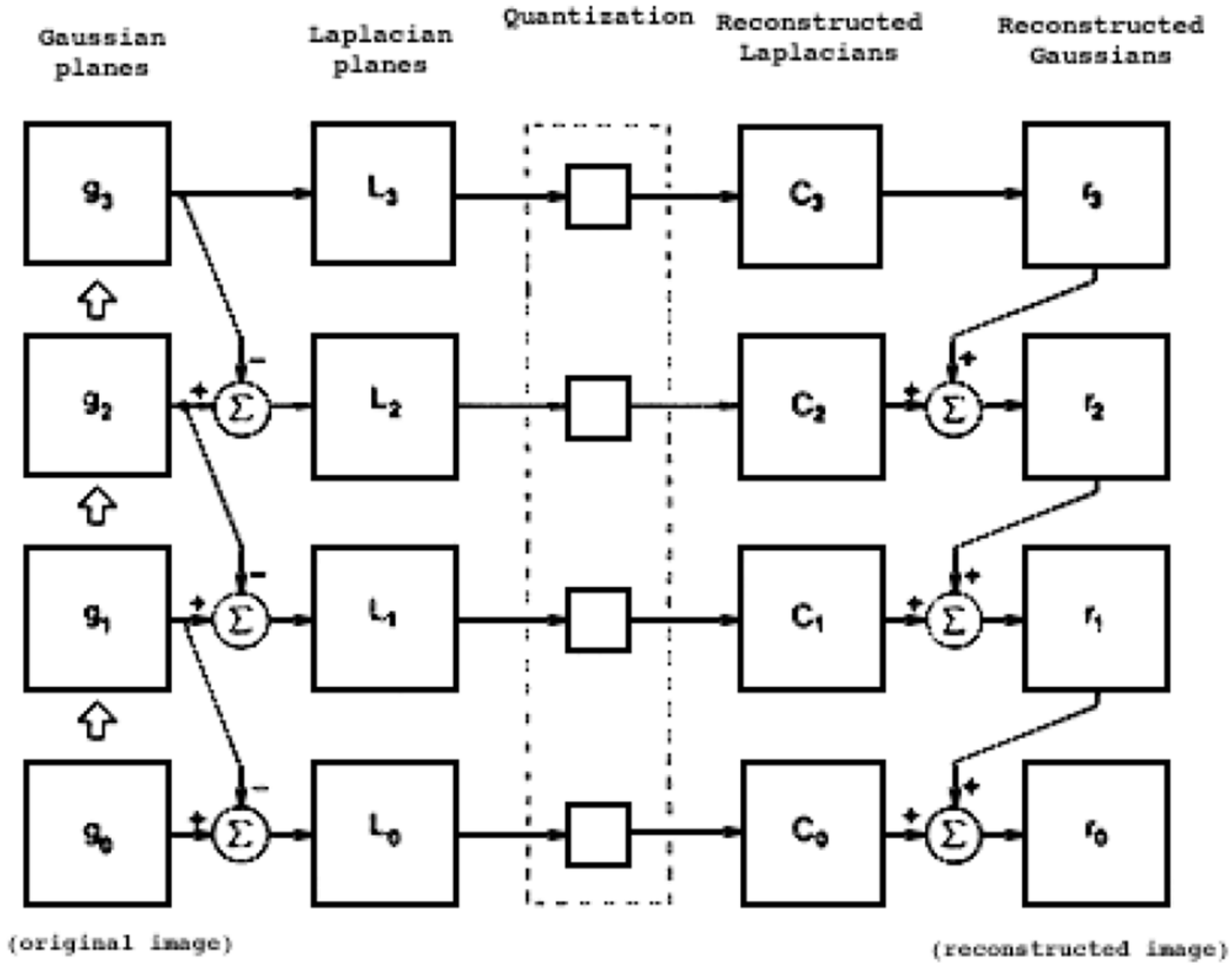
Вся пирамида занимает лишь 4/3 исходного размера!

Пирамиды лапласианов



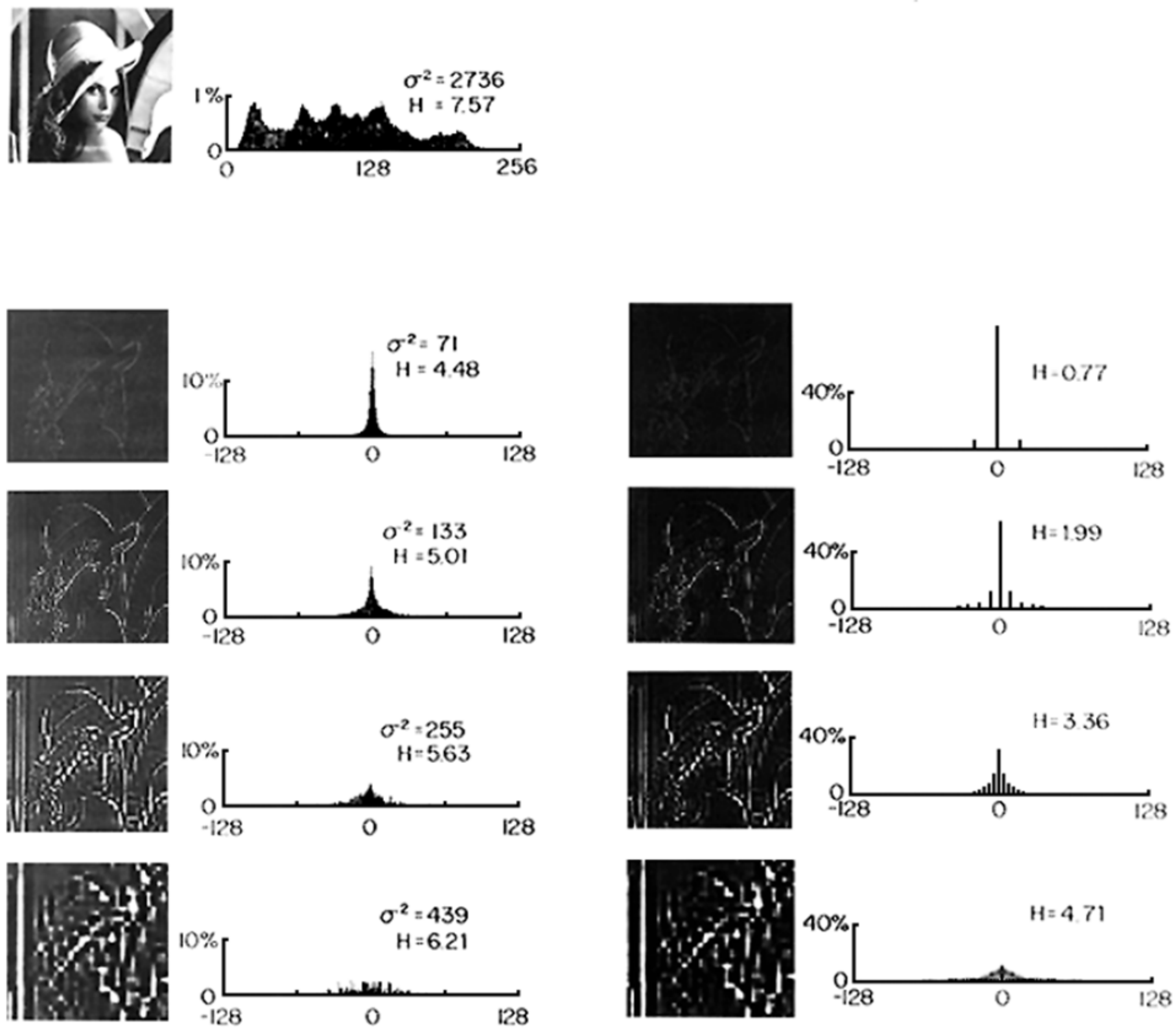
- $L_l = g_l - \mathrm{EXPAND}(g_{l+1}) = g_l - g_{l+1,1^*}$
- $g_n = L_n$
- $g_l = L_l + \mathrm{EXPAND}(g_{l+1})$

Блок-схема

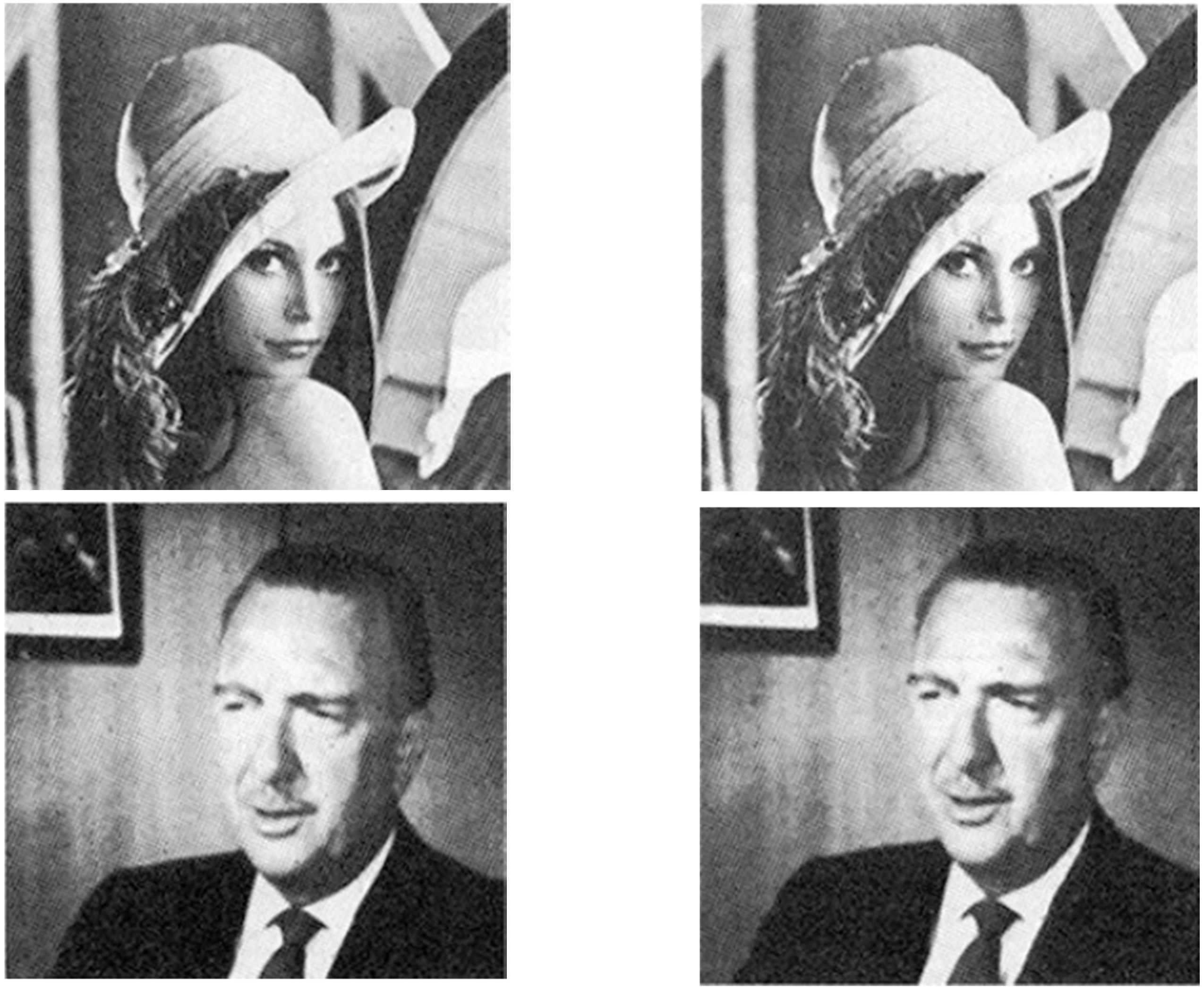


Gaussian planes
Laplacian planes
Quantization
Reconstructed Laplacians
Reconstructed Gaussians
g3
L3
C3
r3
g2
L2
C2
r2
g1
L1
C1
r1
g0
L0
C0
r0
(original image)
(reconstructed image)

Квантование уровней пирамиды



Сжатие с помощью пирамиды



1.58 bit/pixel

0.73 bit/pixel

Исходные

Сжатые

Применение пирамиды

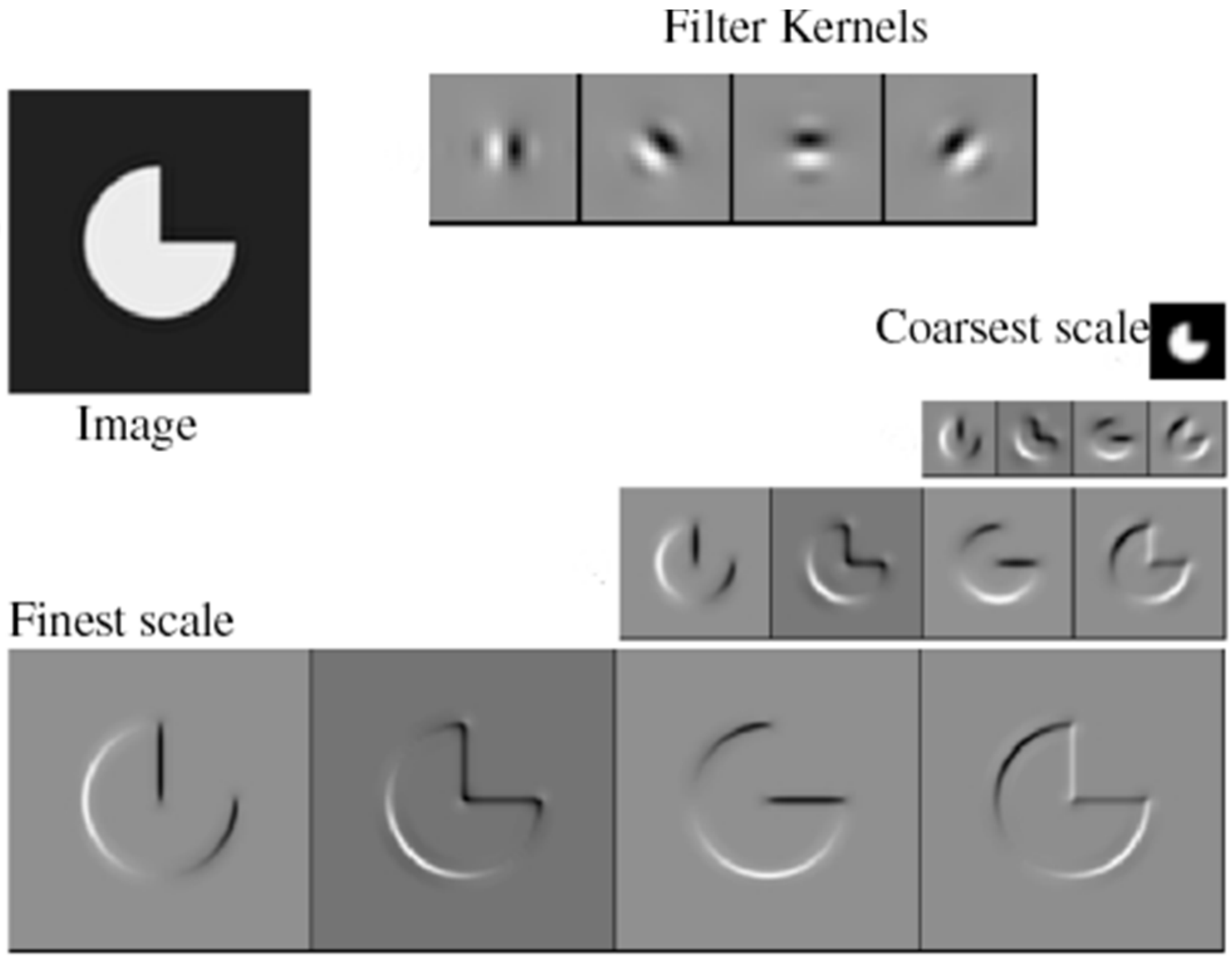
0.03 0.10 0.31 0.81 1.58

- Прогрессивная передача данных
  - Вначале передаются верхние уровни пирамиды
- Числа обозначают бит / пиксель

Ориентированные пирамиды

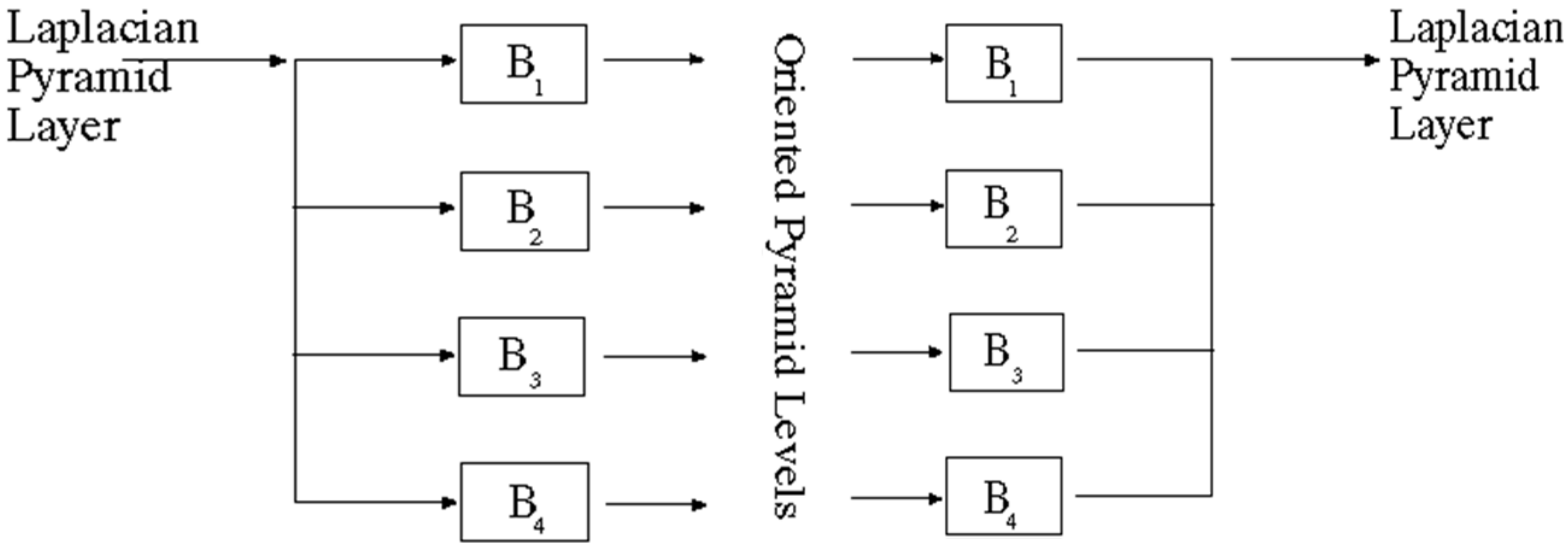
- Пирамида лапласианов не содержит информации об ориентации
- Применим ориентированный фильтр к каждому уровню пирамиды
  - Правильный выбор фильтра позволит упростить синтез изображения
  - Такая пирамида будет хранить информацию о структуре разного масштаба (частоты) и ориентации

Ориентированные пирамиды



E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, “Shiftable multiscale transforms,” *IEEE Trans. Inf. Theory, vol. 38, no. 2* part 2, pp. 587–607, 1992.

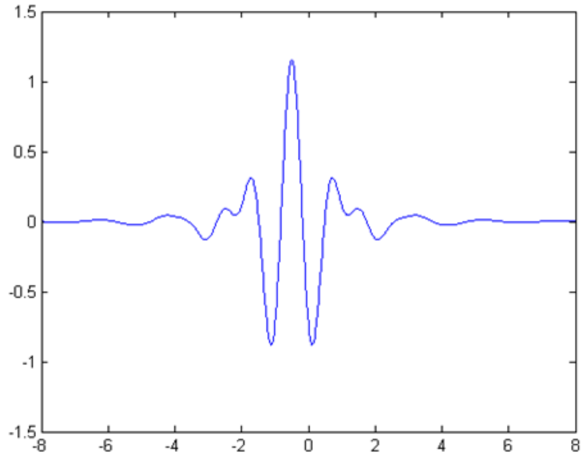
Построение и реконструкция



Анализ Синтез

Вейвлеты

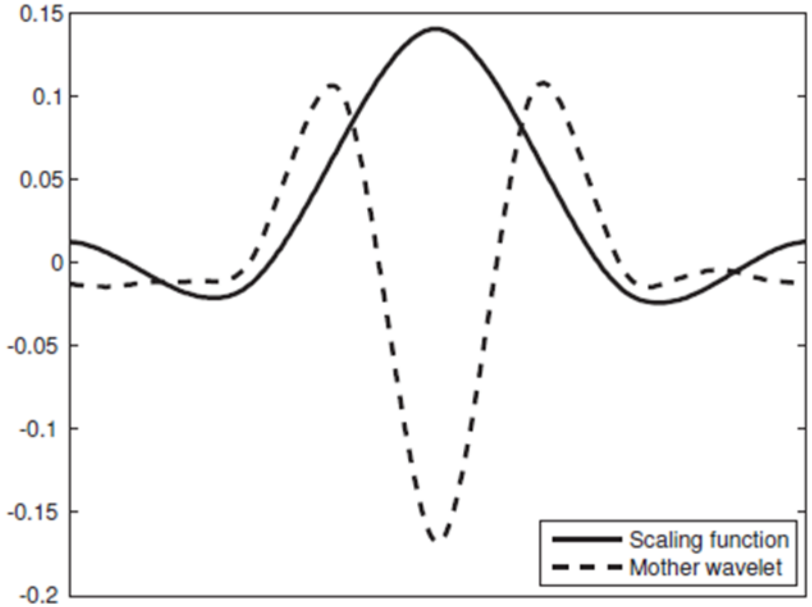
- Вейвлет (wavelet), он же «всплеск» – сигнал, амплитуда которого вначале равна 0, потом ненулевая, потом снова нулевая
- Вейвлет-разложение - разложение сигнала по набору масштабированных и смещённых версий одной базовой функции
- Вейвлет-преобразование может быть ортогональным



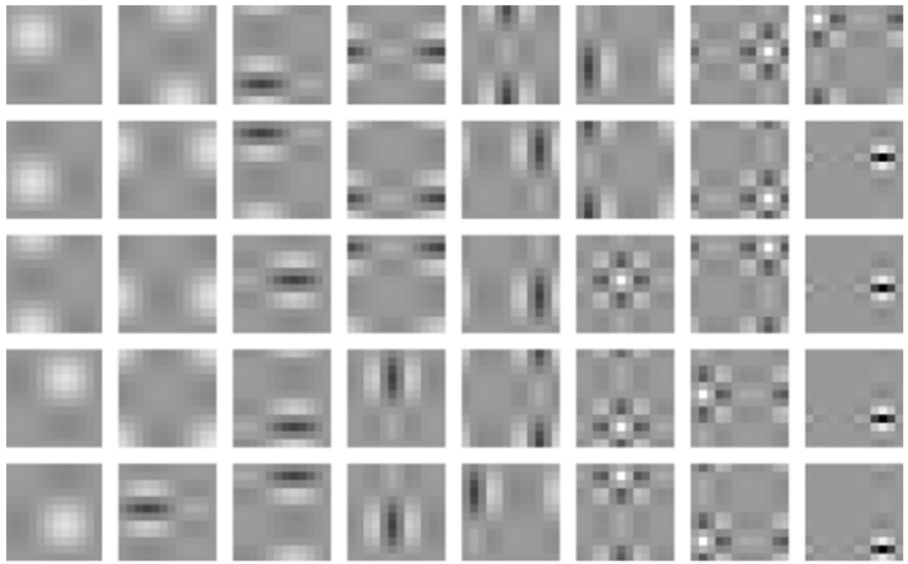
- Доказано, что вейвлет-преобразование оптимально для одномерного кусочно-непрерывного сигнала с конечным числом точек разрыва
- Но в 2D, для линейных разрывов, это не работает
  - Поэтому разложение оптимально лишь для сигналов с точечными разрывами

J. Morlet and A. Grossman, “Decomposition of Hardy functions into square integrable wavelets of constant shape,” *SIAM J. Math. Anal.,* vol. 15, pp. 723–736, 1984.

Кратномасштабный анализ



Базовые функции

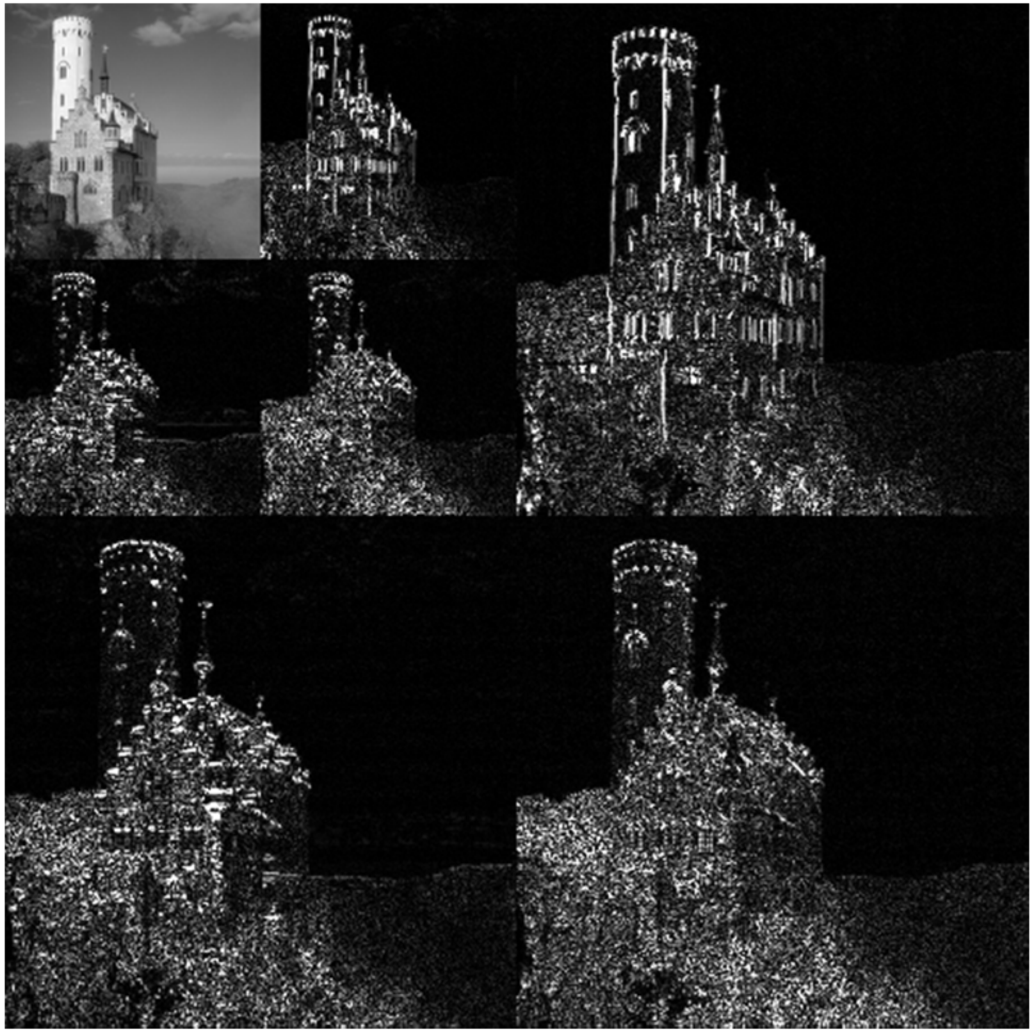


Примеры атомов

- Вейвлет-базис для кратномасштабного анализа можно построить с помощью «материнского вейвлета» и «масштабирующей функции»
  - Масштабирующая функция – низкочастотная функция, позволяет «огрубить сигнал»
  - Материнский вейвлет – высокочастотная функция, позволяет выделить детали сигнала нужного масштаба

S. G. Mallat, “A theory for multiresolution signal decomposition: The wavelet representation,” *IEEE Trans. Patt. Anal. Machine Intell., vol. 11,* no. 7, pp. 674–693, 1989.

JPEG2000



- Одно из самых известных применения вейвлет-преобразования
- 20% преимущество перед JPEG
- Больше на больших изображения с плавными изменениями яркости
- Оптимально для медицинских изображений
  - Высокое разрешение
  - Размытые границы

От преобразований к словарю

$y \in \Re^m$ - сигнал

$x \in \Re^n$ - представление

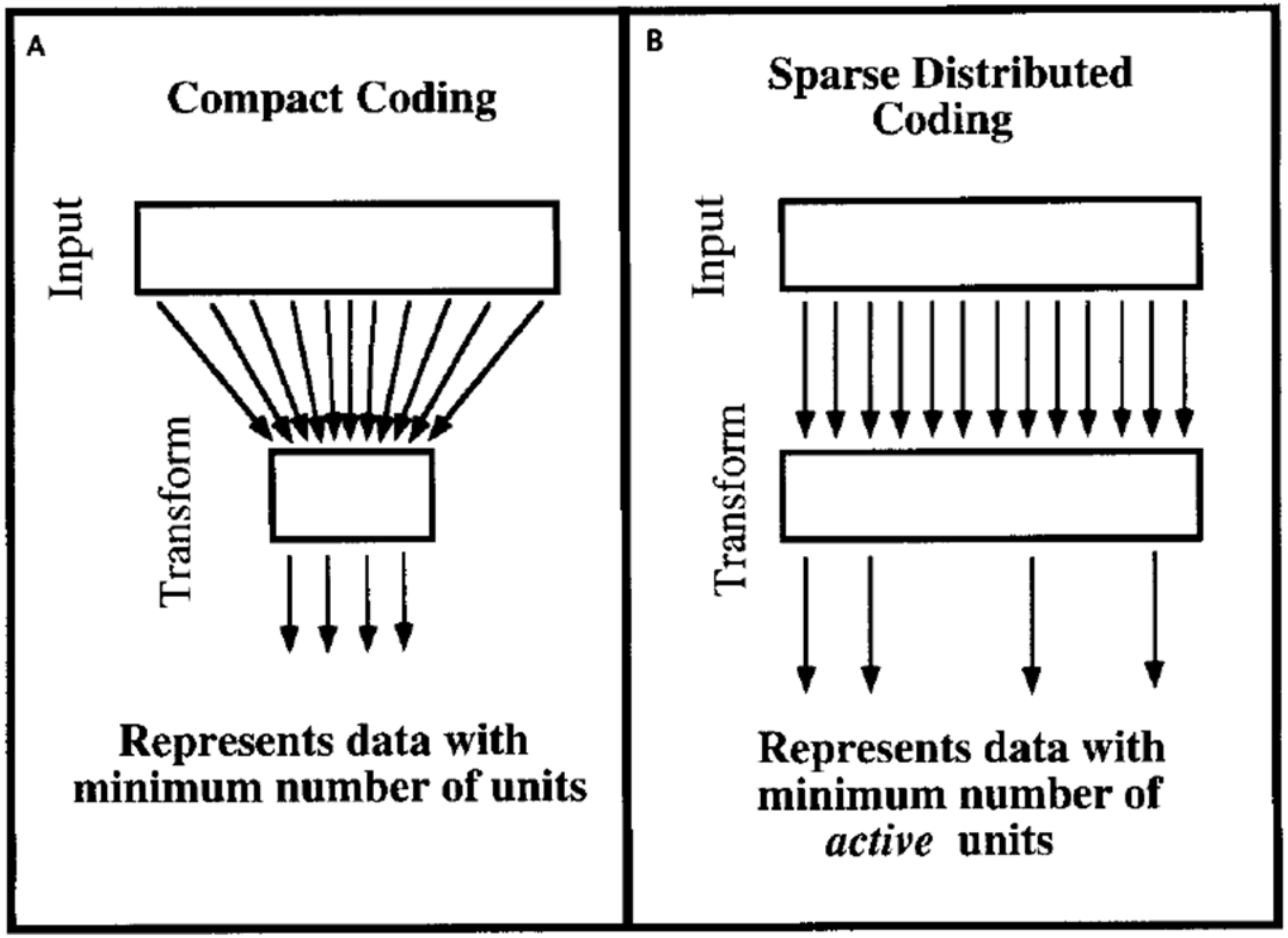
$$D \in \Re^{n \times m}$$

$$y = Dx$$

- Рассмотрим случай n >>m
- Сигнал y можно представить как комбинацию некоторого небольшого подмножества из D
- Причем x – уже неоднозначно!

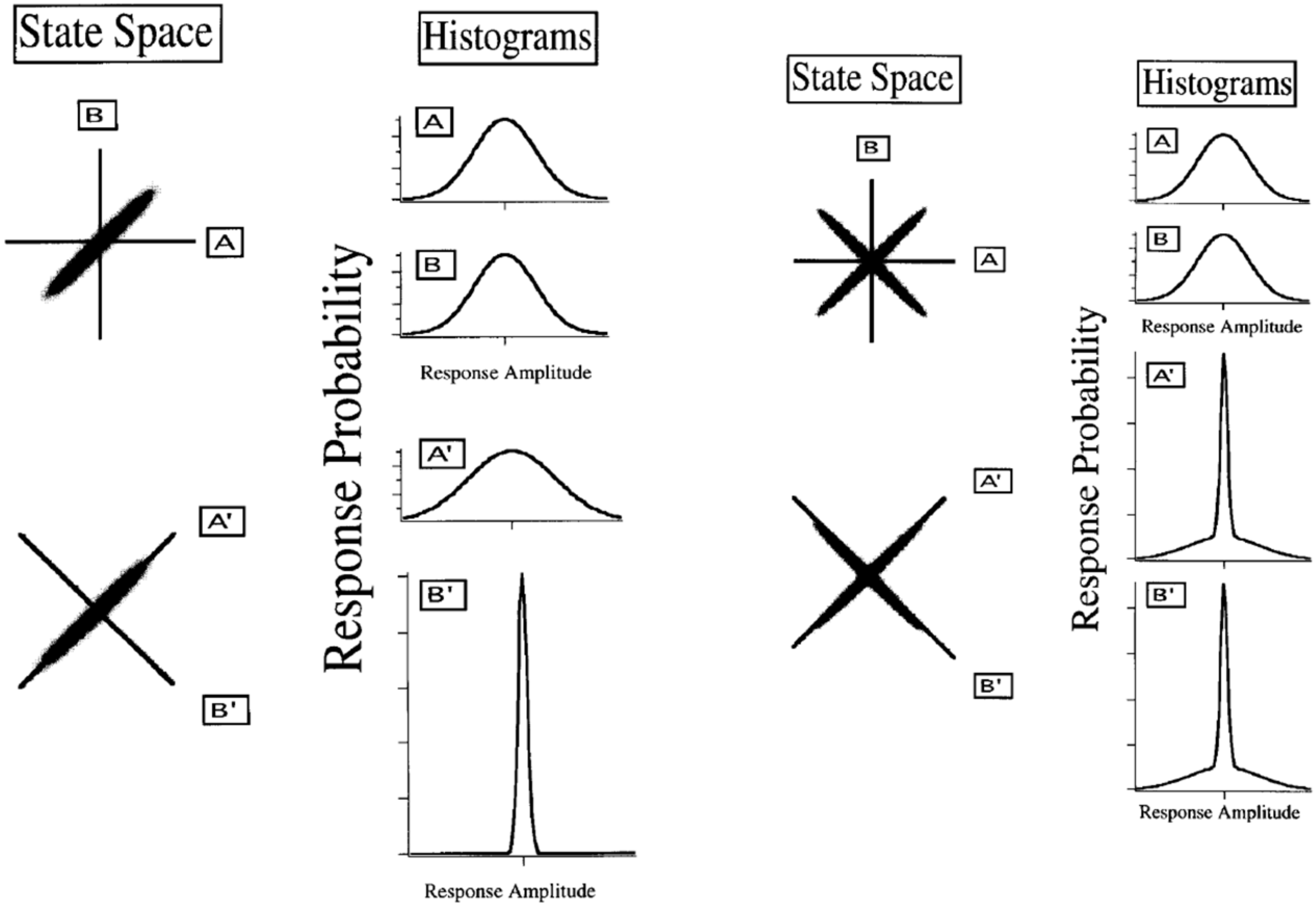
- В этом случае D называется словарём
- Мы можем накладывать разные ограничения на x
- Каждая строка D – «атом»

"Compact" vs "Sparse"



D. J. Field, "What is the goal of sensory coding?," *Neural Computation,* vol. 6, no. 4, pp. 559–601, 1994.

Разреженное кодирование



Биологические причины

- Поиск особенностей и сопоставление
  - Вероятность отклика каждой отдельной клетки (переменной) низка
  - Вероятность случайного (ложного) повторения того же самого шаблона тоже низка
- Хранение данных в ассоциативной памяти
  - Если данных разрежены (sparcified), то сети позволяют хранить больше данных, и извлекать их быстрее

Формализация

$y \in \Re^m$ - сигнал $\quad x \in \Re^n$ - представление $\quad D \in \Re^{n \times m}$

$$y = Dx$$

$$\|x\|_0 < \mathrm{L}$$

- 0-норма – «псевдонорма», равна количеству ненулевых элементов вектора x
- Поскольку точного равенства достичь нельзя, ищем минимум нормы ||Dx-y|| при условии разреженного представления x
- 0-норма очень сложна в использовании, поэтому её иногда заменяют L1-нормой

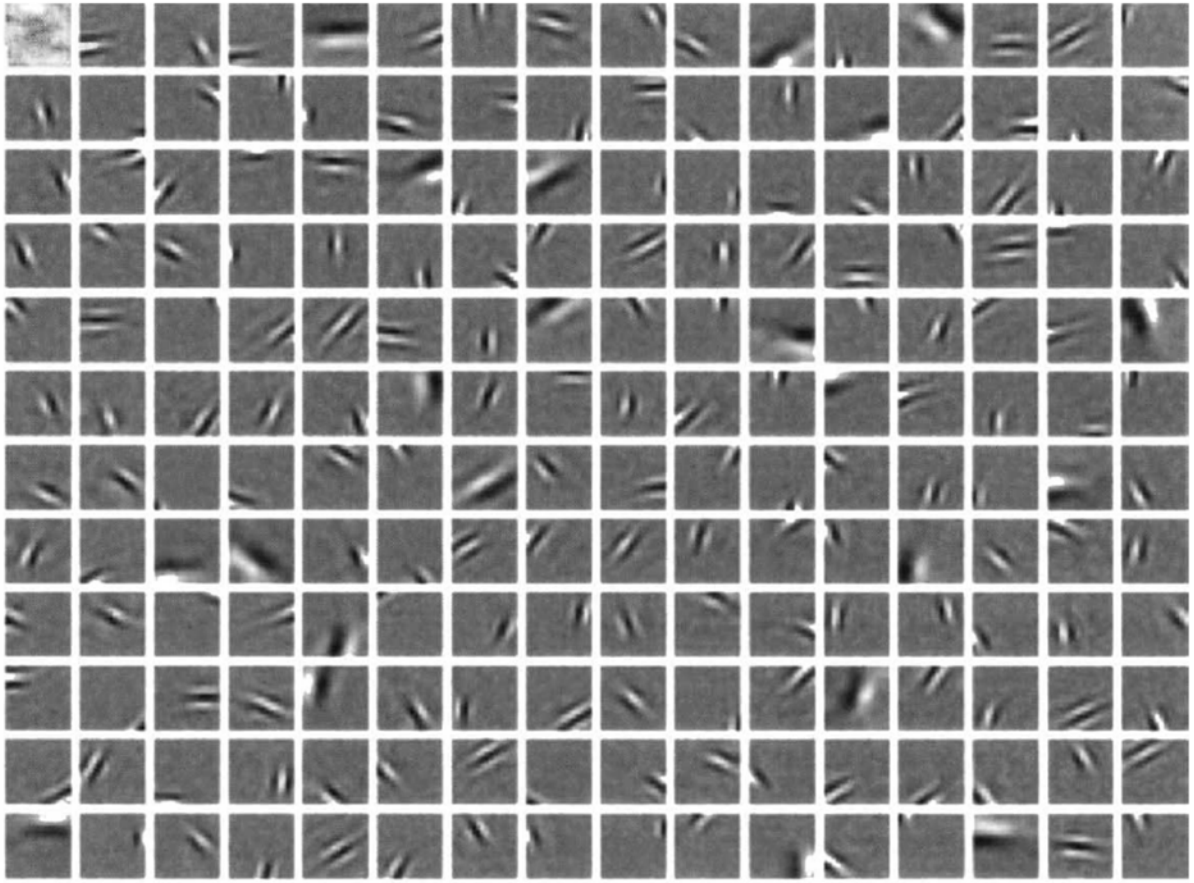
Свойства простых клеток

- Рецептивные поля простых клеток в визуальной коре головного мозга обладают следующими свойствами
  - Пространственно локализованные
  - Ориентированные
  - Заданная полоса частот (bandpass)
    - Чувствительны к структуре на определенном пространственном масштабе

- Гипотеза: стратегия кодирования сигнала, максимизирующая разреженность, достаточна для объяснения этих свойств

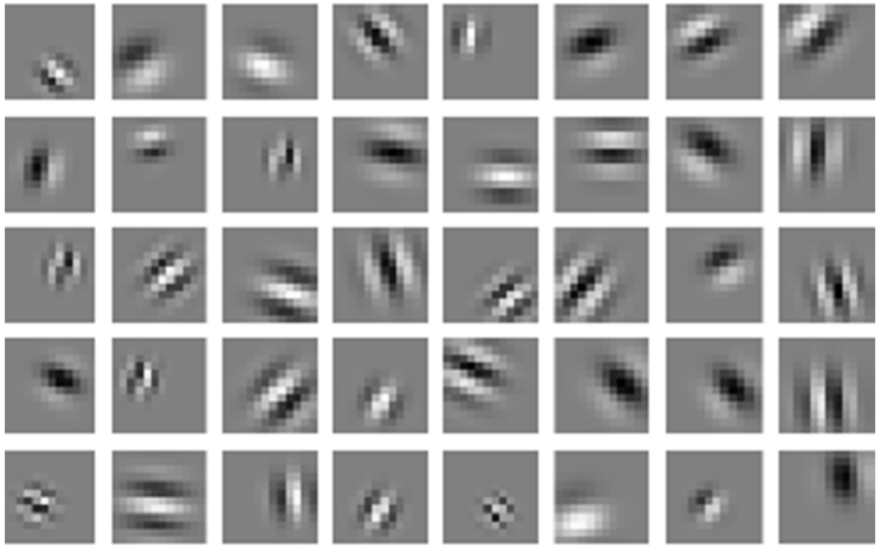
B.A. Olshausen and B.J. Field, *Emergence of simple-cell receptive field properties by learning a sparse code for natural images, Nature, 381 (1996), pp. 607–609.*

Разреженное представление

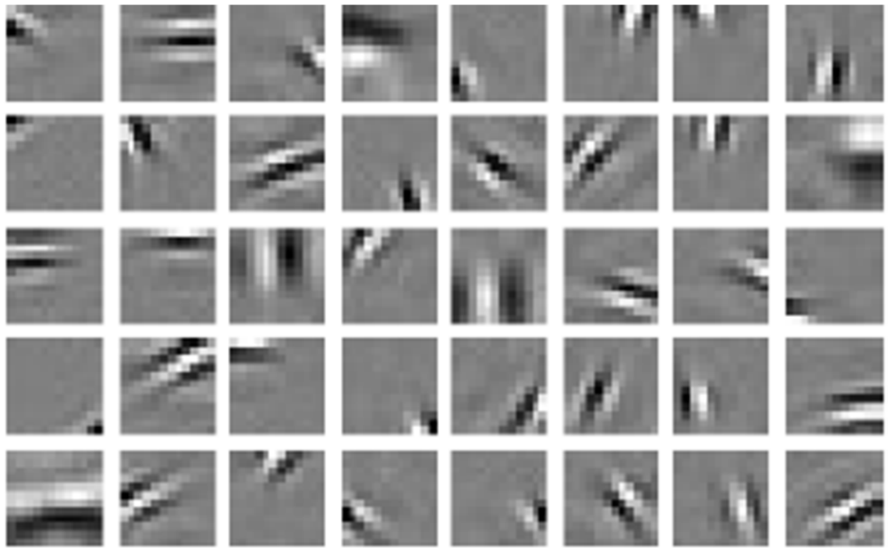


• Словарь, полученный с помощью обучения без учителя (кластеризации), с учётом предположения (условия) разреженности представления

Сравнение с фильтрами Габора



Фильтры Габора



Обученный «словарь»

Словарь, построенный автоматически для набора естественных изображений, с выполнением условия разреженности представления, оказался аналогичным биологическим свойствам рецептивных полей

Какие возникают задачи?

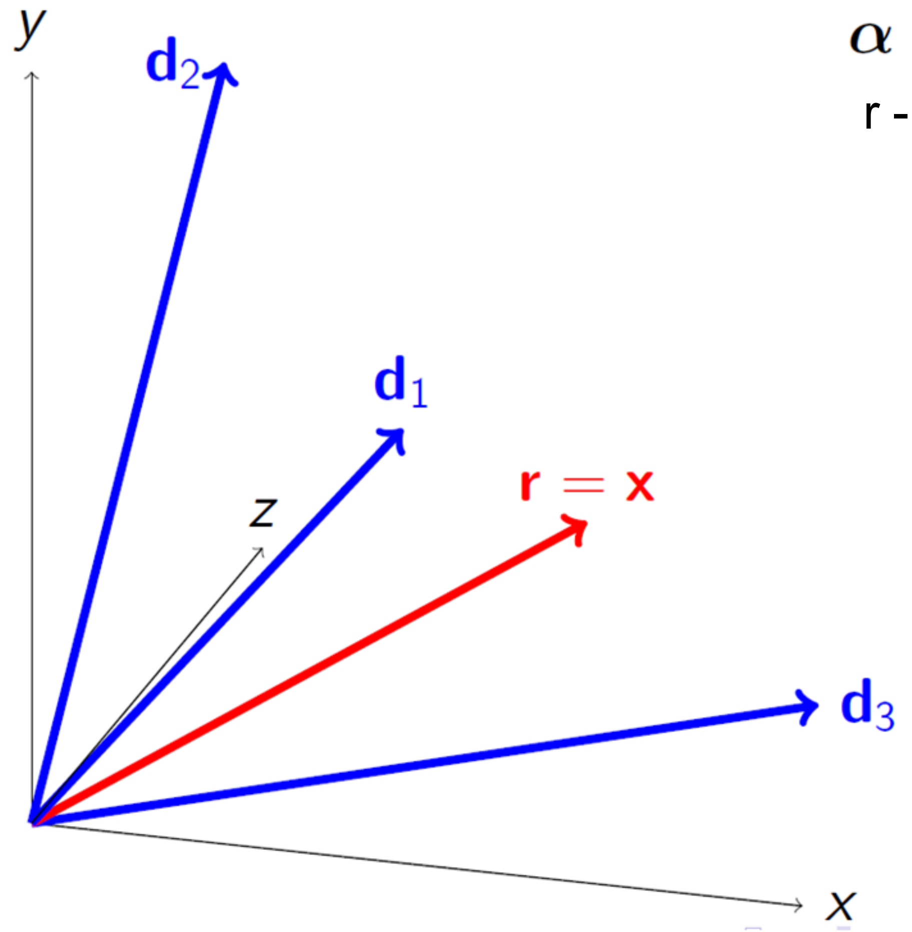
- Как получить представление x сигнала y по словарю D ?
- Какой словарь использовать?
- Можно ли обучить словарь D для задачи?
- Для чего применять разреженное представление?

Алгоритмы выбора x

$$y = Dx \quad \|x\|_0 < \mathrm{L}$$

- В общем случае при 0-норме задача NP-сложная
- Разработан ряд приближенных жадных методов
  - Basis Pursuit
  - Matching Pursuit (*)
  - Orthogonal Matching Pursuit (*)
  - LARS-Lasso

Matching Pursuit



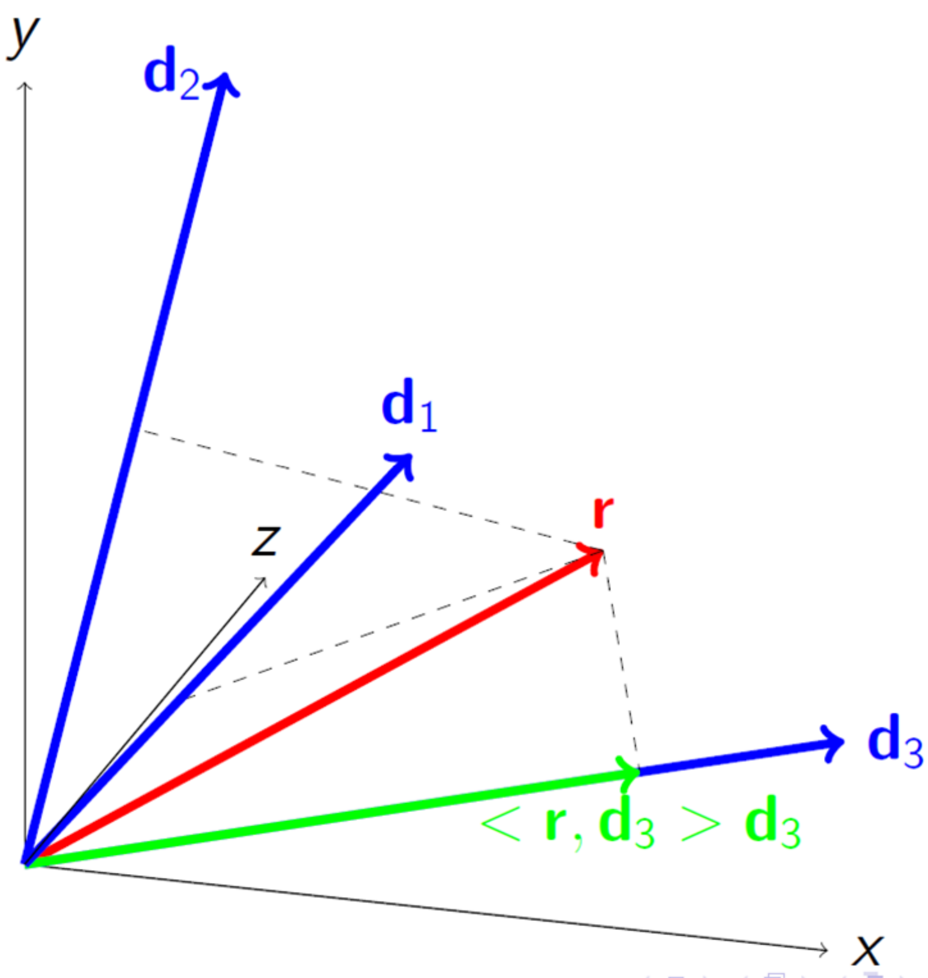
$$\boldsymbol{\alpha} = (0, 0, 0)$$

r - невязка

S. Mallat and Z. Zhang. Matching pursuit in a time-frequency dictionary. *IEEE Transactions on Signal Processing, 41(12):3397–3415, 1993.*

Slide by F.Bach

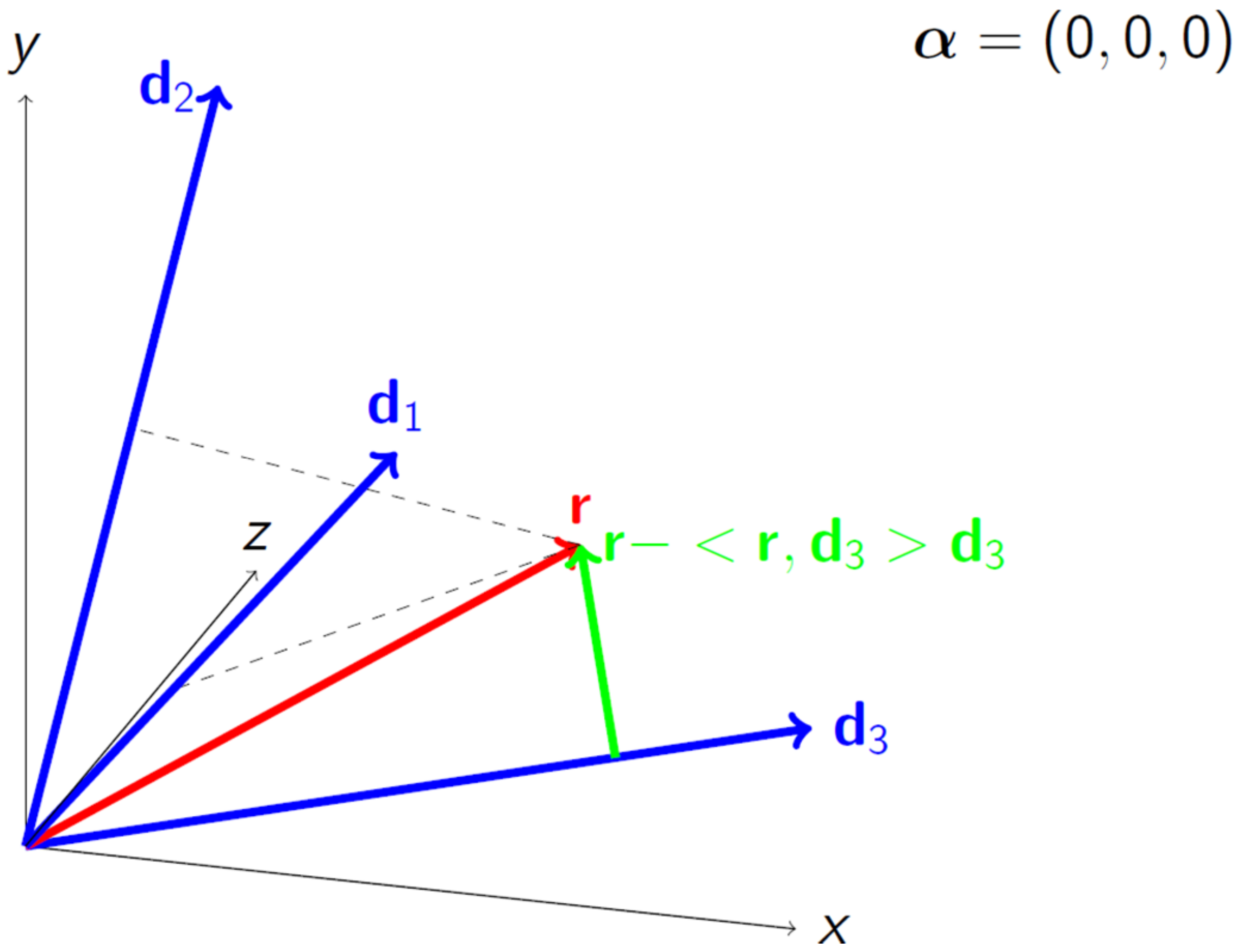
Matching Pursuit



Проецируем на все атомы, находим максимум

Slide by F.Bach

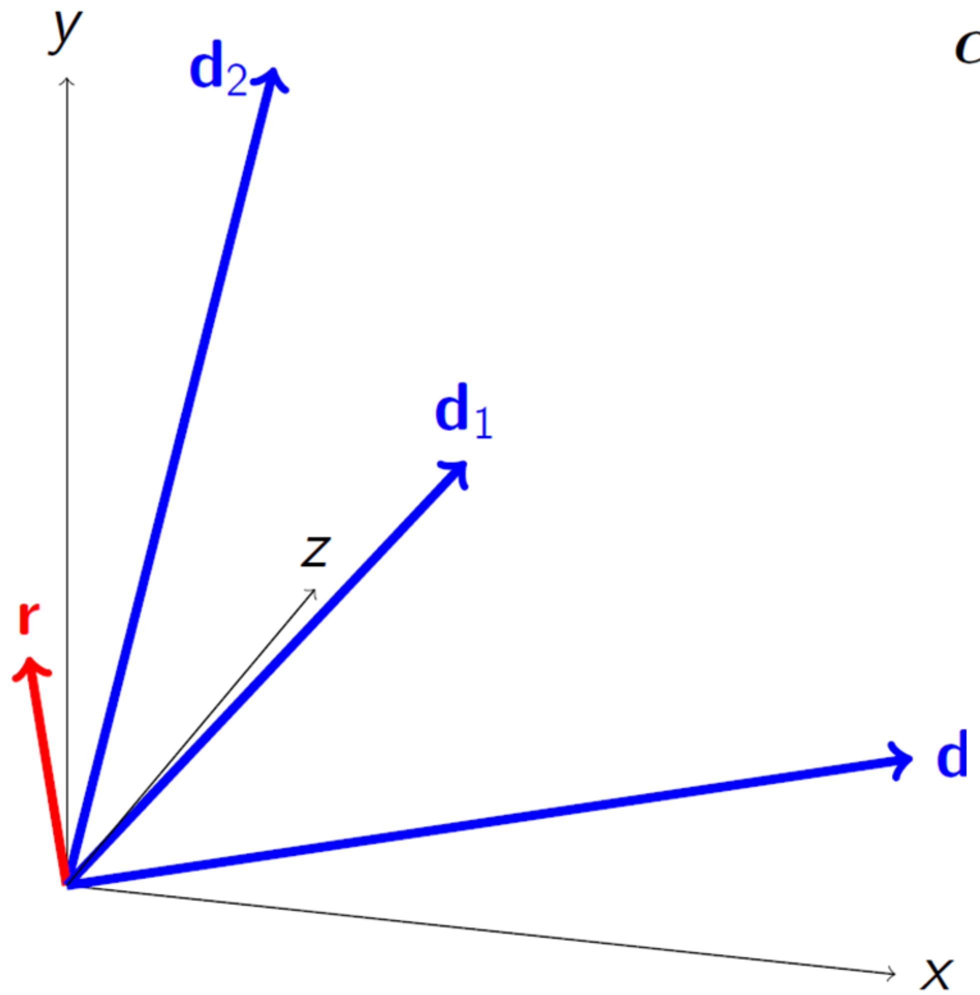
Matching Pursuit



- Обновляем невязку r

Slide by F.Bach

Matching Pursuit



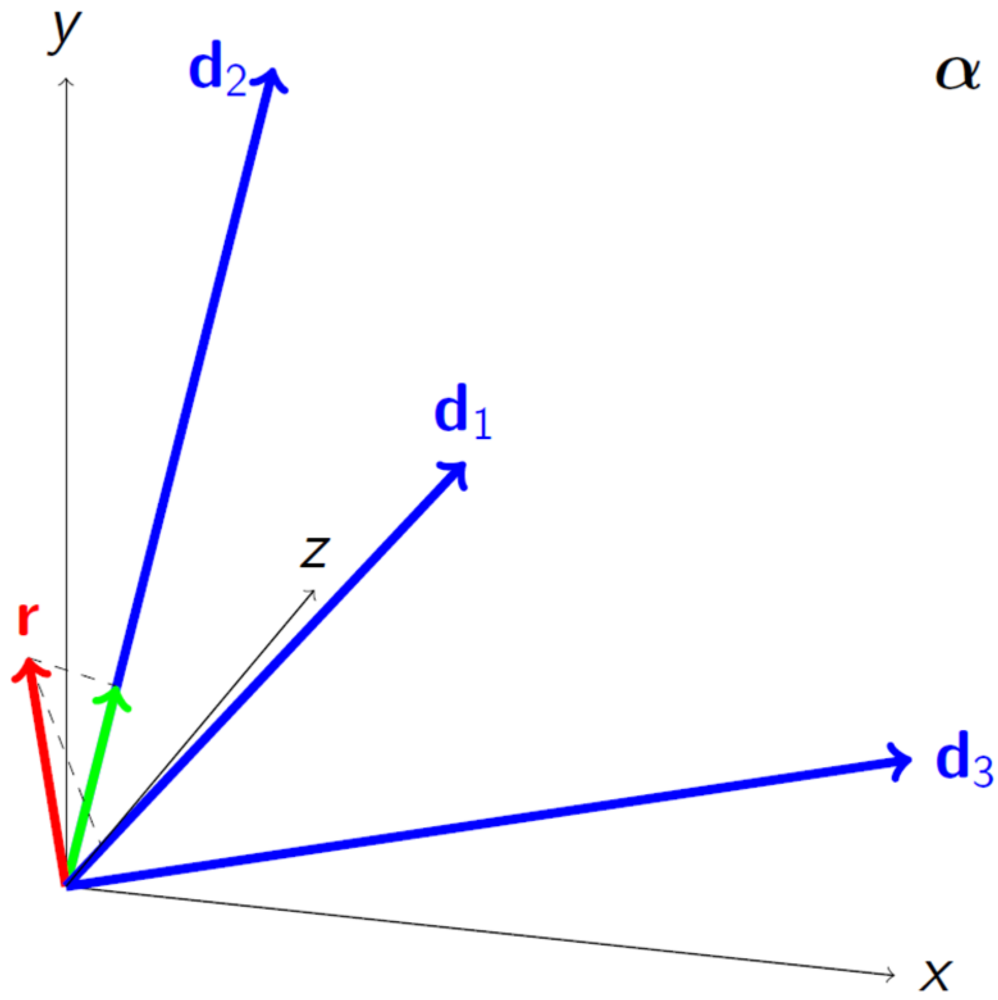
$$\alpha = (0, 0, 0.75)$$

- Обновляем веса

Slide by F.Bach

Matching Pursuit

$\alpha = (0, 0, 0.75)$

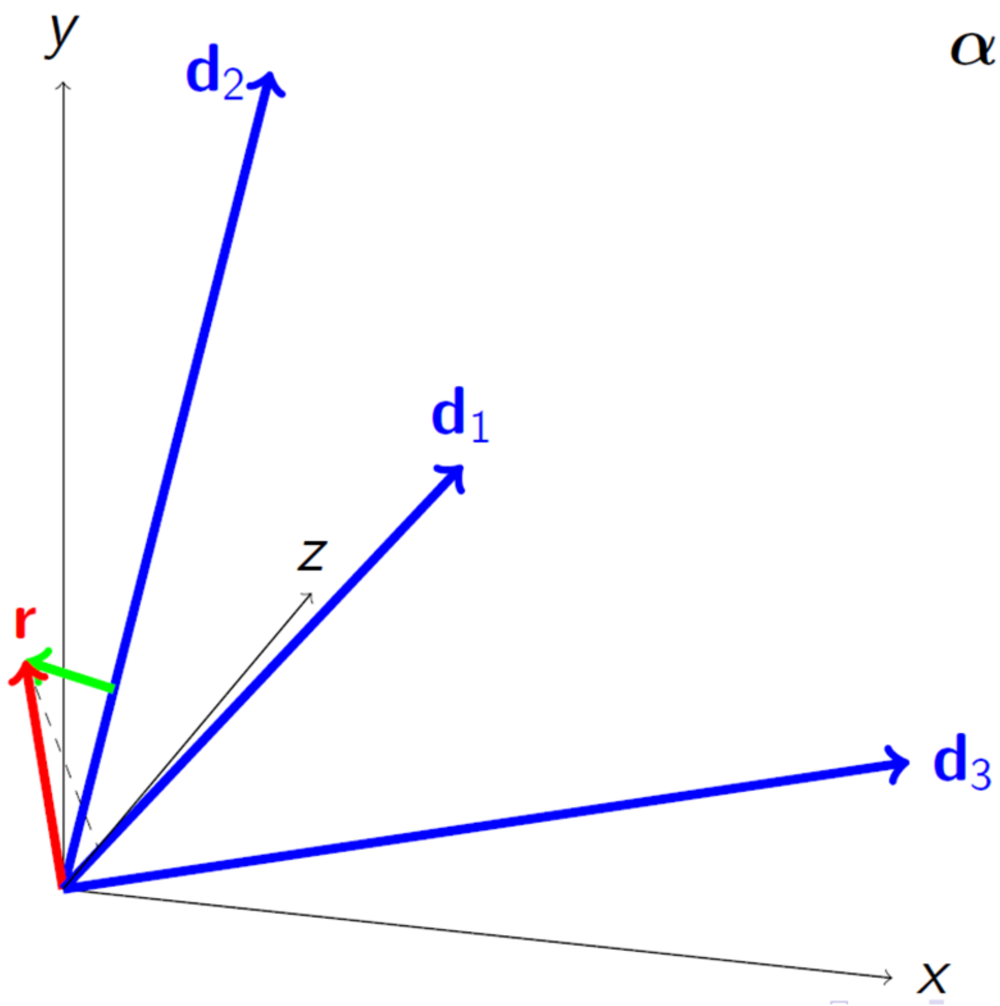


Снова проецируем на все атомы, ищем максимальную проекцию

Slide by F.Bach

Matching Pursuit

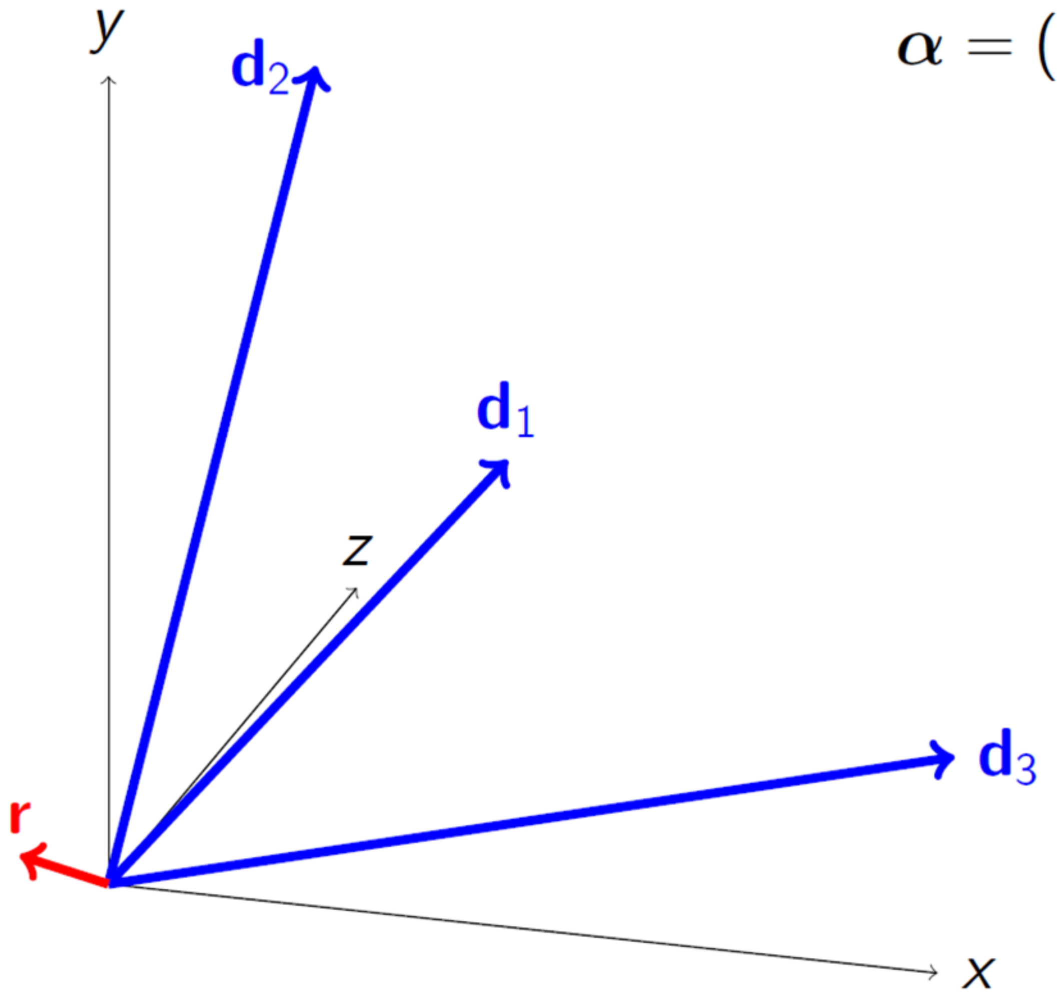
$$\alpha = (0, 0, 0.75)$$



Slide by F.Bach

Matching Pursuit

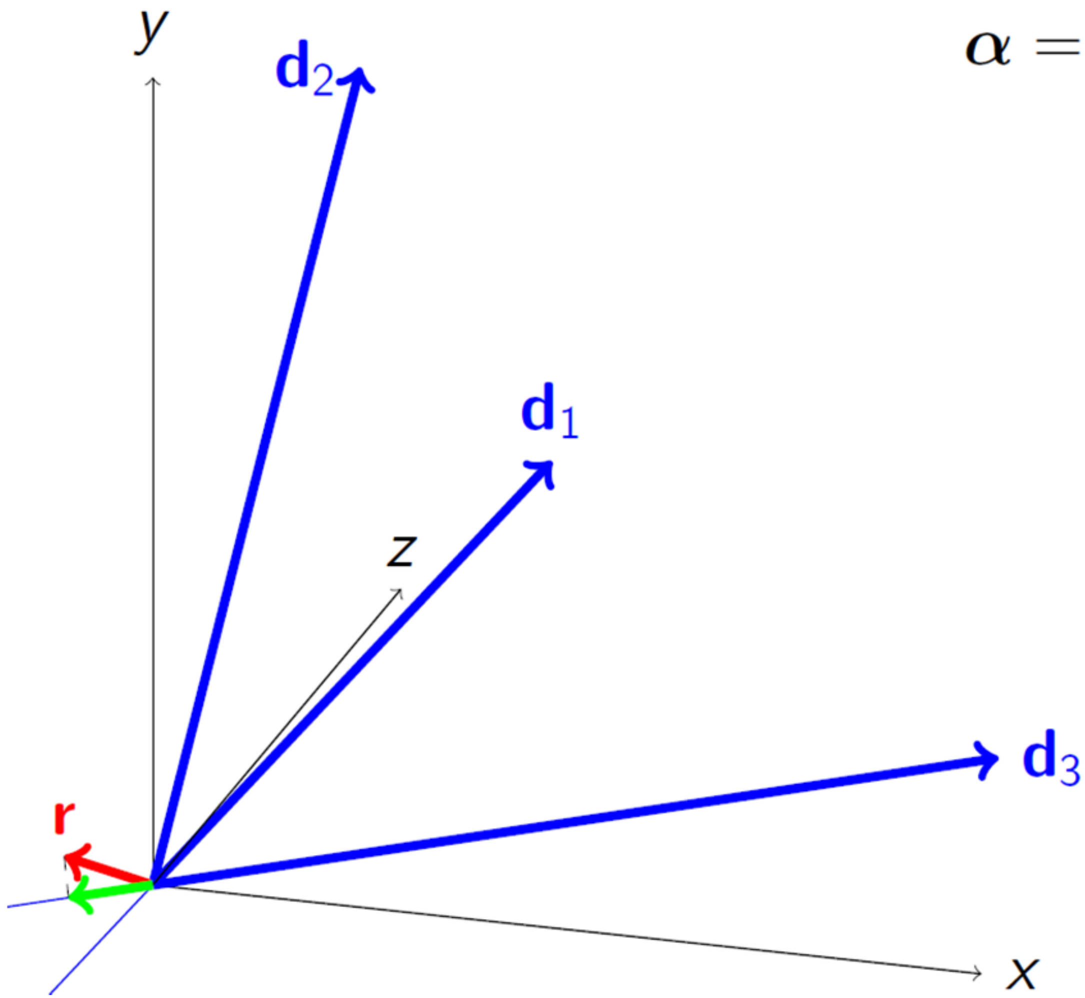
$\boldsymbol{\alpha} = (0, 0.24, 0.75)$



Slide by F.Bach

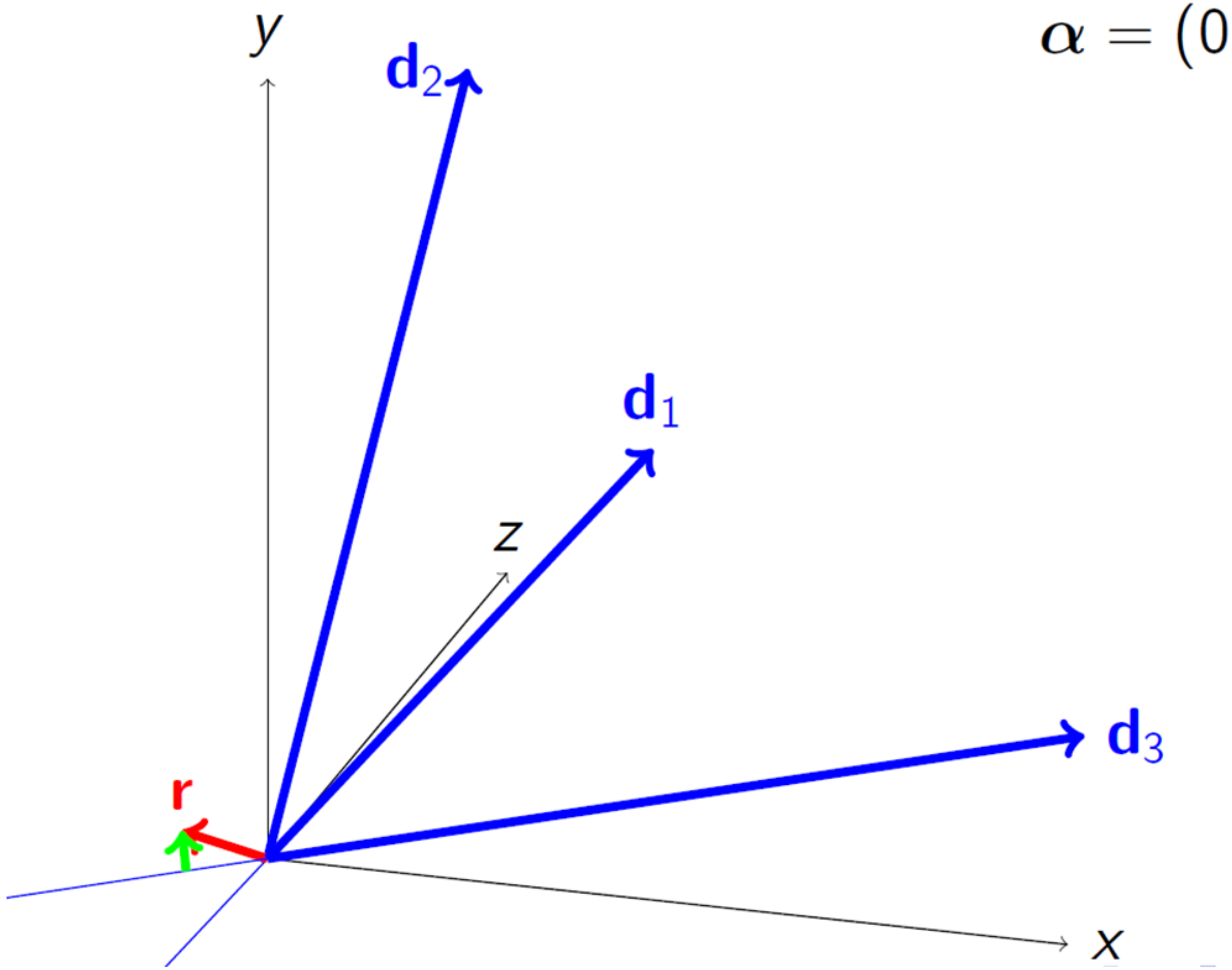
Matching Pursuit

$$\alpha = (0, 0.24, 0.75)$$



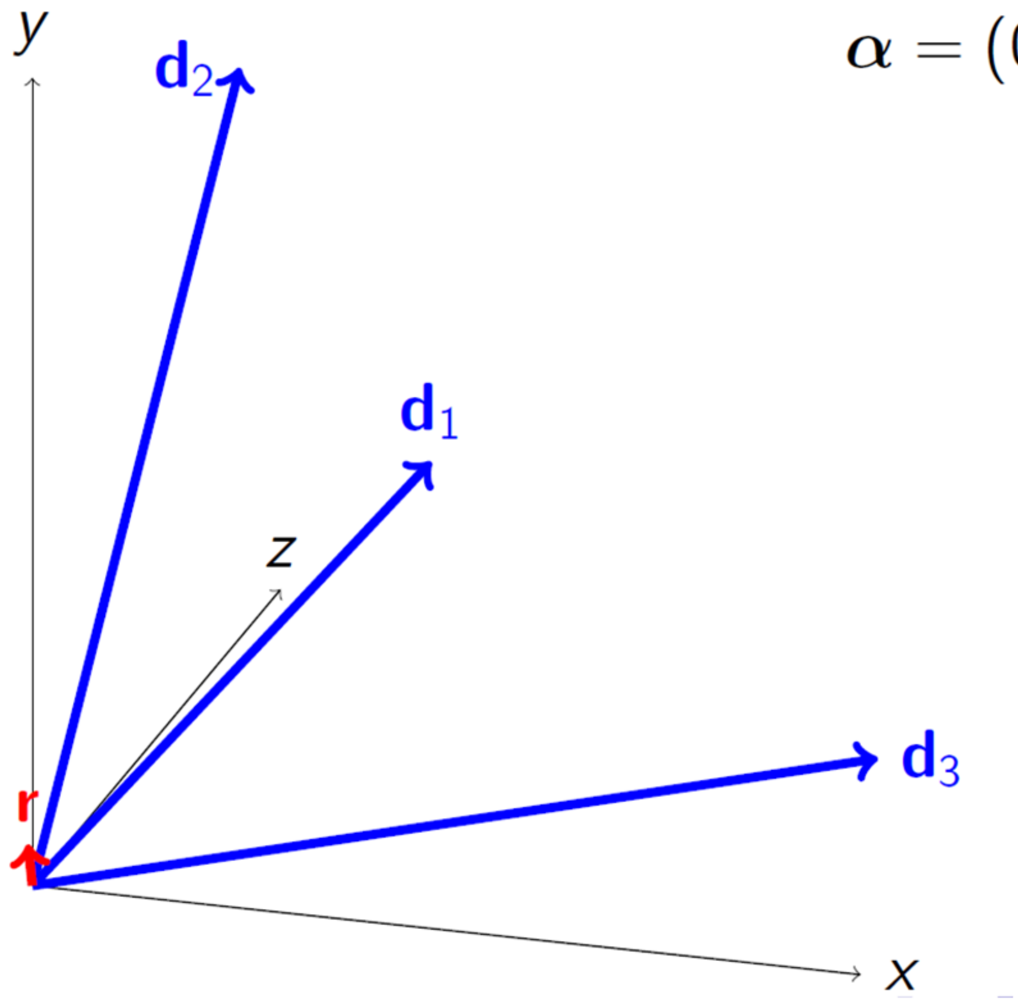
Matching Pursuit

$$\boldsymbol{\alpha} = (0, 0.24, 0.75)$$



Slide by F.Bach

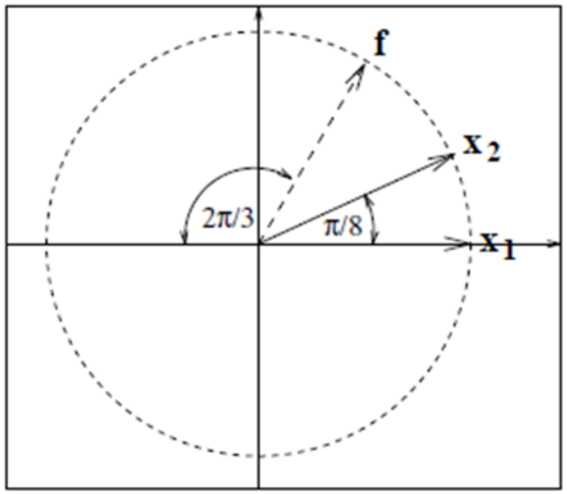
Matching Pursuit



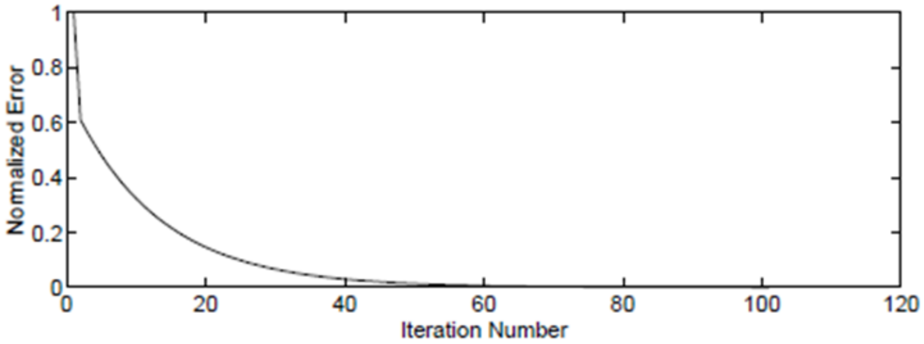
Slide by F.Bach

Недостатки метода

(a)



(b)



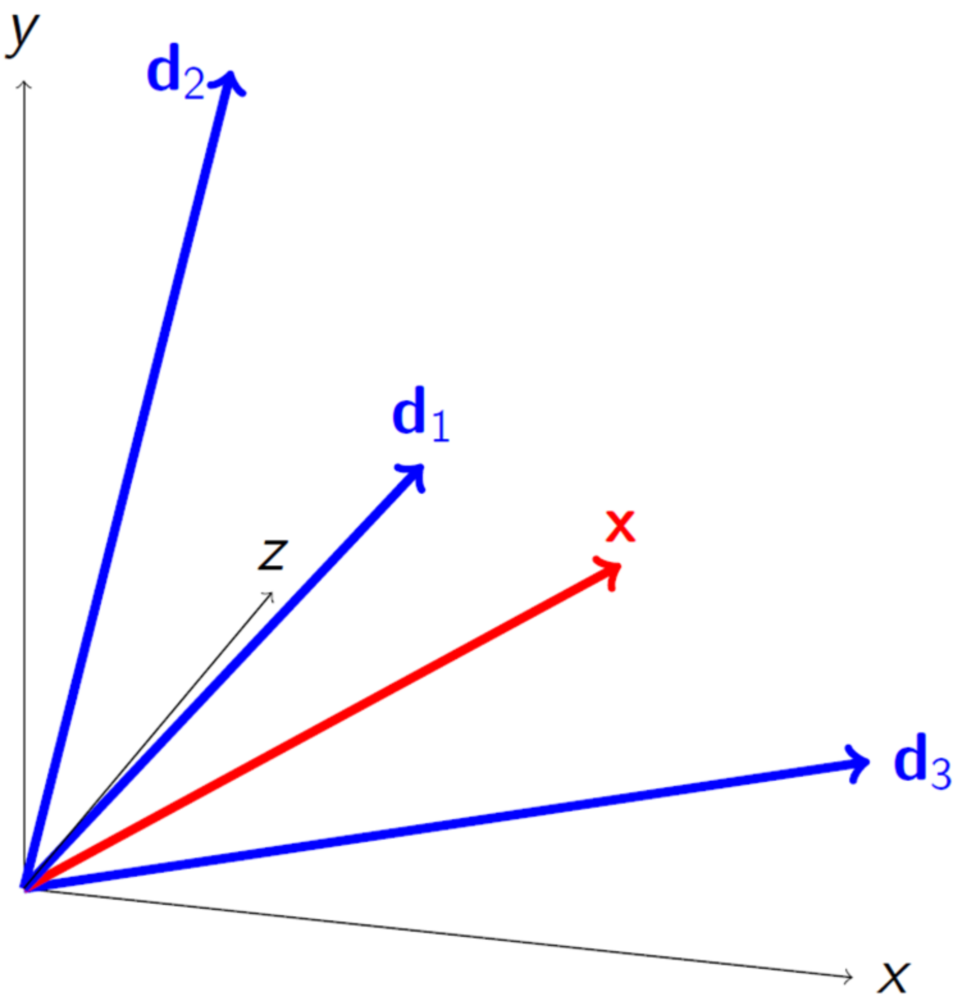
- Существует точное разложение f по x_1 и x_2
- Но после первых двух итераций будут использованы оба вектора x_1 и x_2, но ошибка будет ненулевой
- Метод не оптимален в том смысле, что разложение по уже использованным векторам не оптимально
- Проблема – невязка r не обязательно лежит в подпространстве, ортогональном к подпространству, образованном выбранными векторами

Orthogonal Matching Pursuit

- Идея:
  - Невязка r должна лежать в дополнительном подпространстве к подпространству, образованному выбранными атомами Г
  - Будем строить Г, добавляя в него по одному атому за итерацию
    - Каждый атом добавляется один раз
  - На каждой итерации выбираем такой атом d_i, чтобы $\Gamma_{i-1} + d_i$ минимизировало невязку r_i

Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition,” *Conference Record of The 27th Asilomar Conference on Signals, Systems and Computers, pp. 40–44, 1993.*

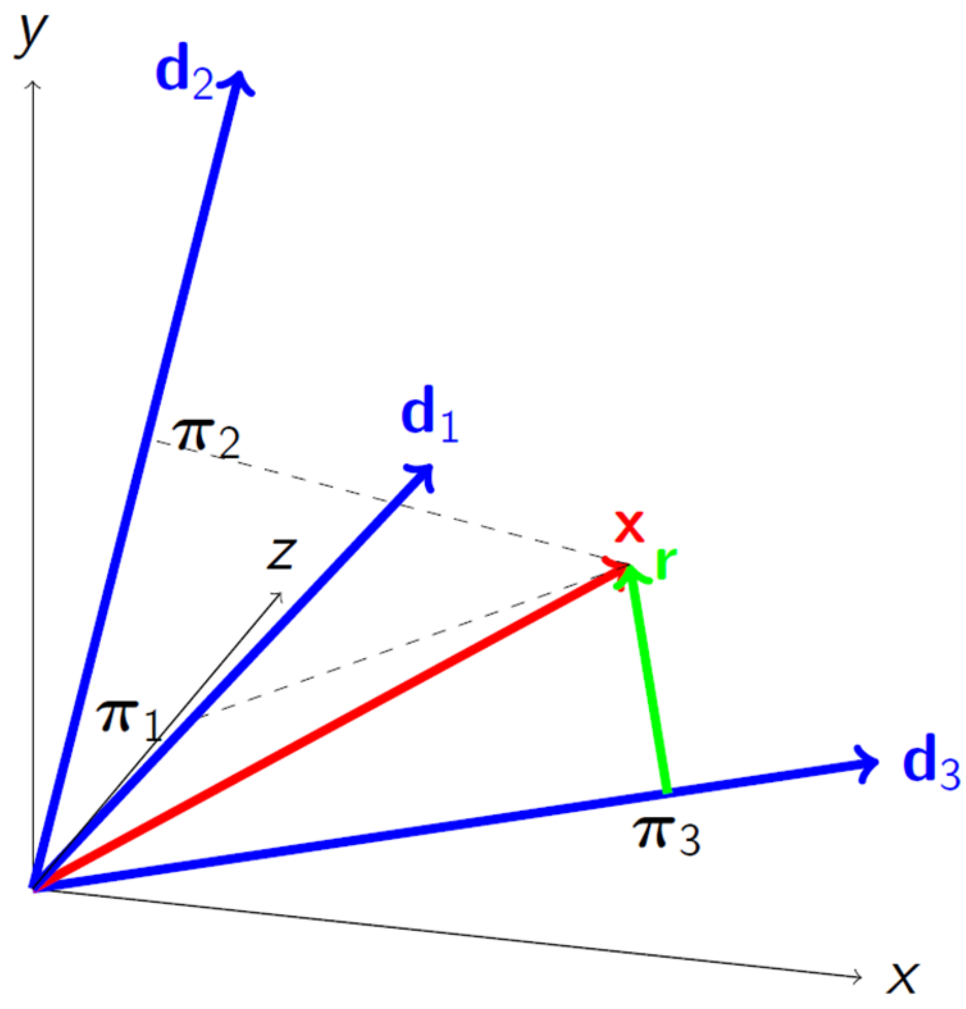
Orthogonal Matching Pursuit



$$\alpha = (0, 0, 0)$$
$$\Gamma = \emptyset$$

Slide by F.Bach

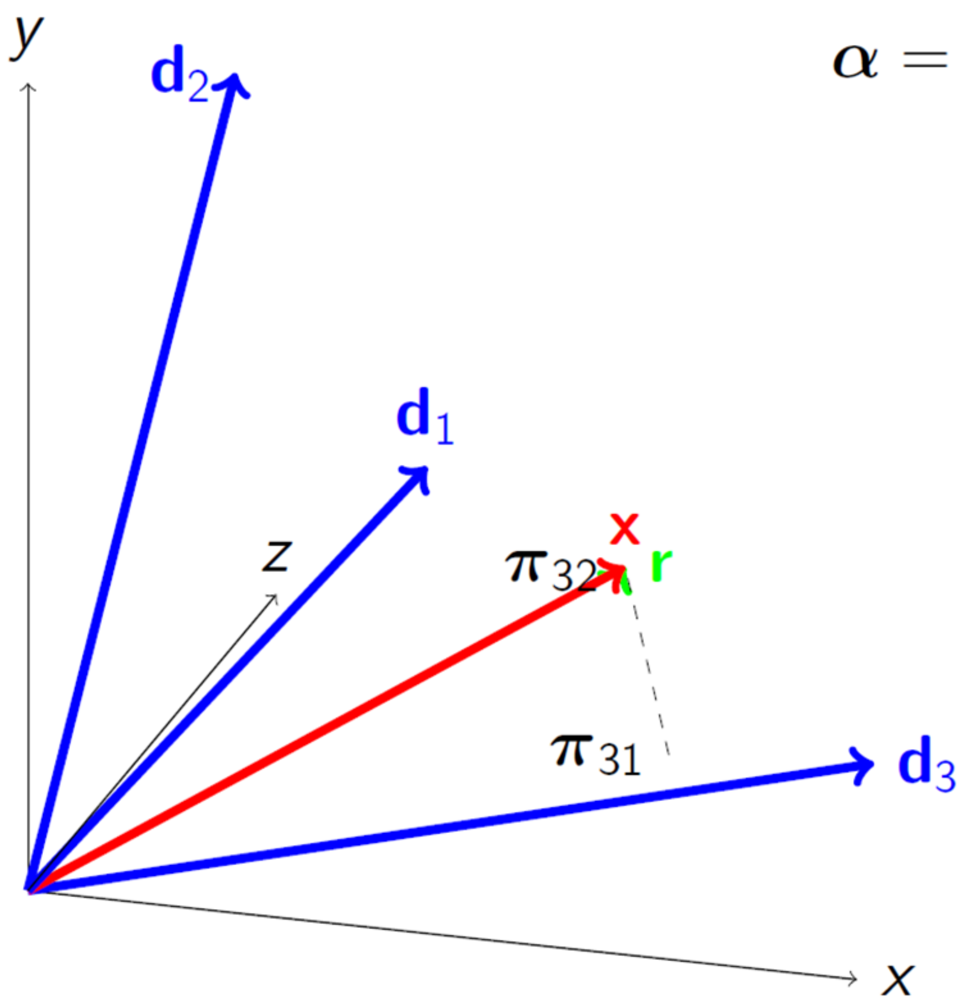
Orthogonal Matching Pursuit



$$\alpha = (0, 0, 0.75)$$
$$\Gamma = \{3\}$$

Slide by F.Bach

Orthogonal Matching Pursuit



$$\alpha = (0, 0.29, 0.63)$$
$$\Gamma = \{3, 2\}$$

Slide by F.Bach

Orthogonal Matching Pursuit

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^p} \ ||\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}||_2^2 \ \text{ s.t. } \ ||\boldsymbol{\alpha}||_0 \leq L$$

1: $\Gamma = \emptyset$.
2: **for** *iter* $= 1, \ldots, L$ **do**
3: Select the atom which most reduces the objective

$$\hat{\imath} \leftarrow \arg\min_{i \in \Gamma^C} \left\{ \min_{\boldsymbol{\alpha}'} ||\mathbf{x} - \mathbf{D}_{\Gamma \cup \{i\}} \boldsymbol{\alpha}'||_2^2 \right\}$$

4: Update the active set: $\Gamma \leftarrow \Gamma \cup \{\hat{\imath}\}$.
5: Update the residual (orthogonal projection)

$$\mathbf{r} \leftarrow (\mathbf{I} - \mathbf{D}_\Gamma (\mathbf{D}_\Gamma^T \mathbf{D}_\Gamma)^{-1} \mathbf{D}_\Gamma^T) \mathbf{x}.$$

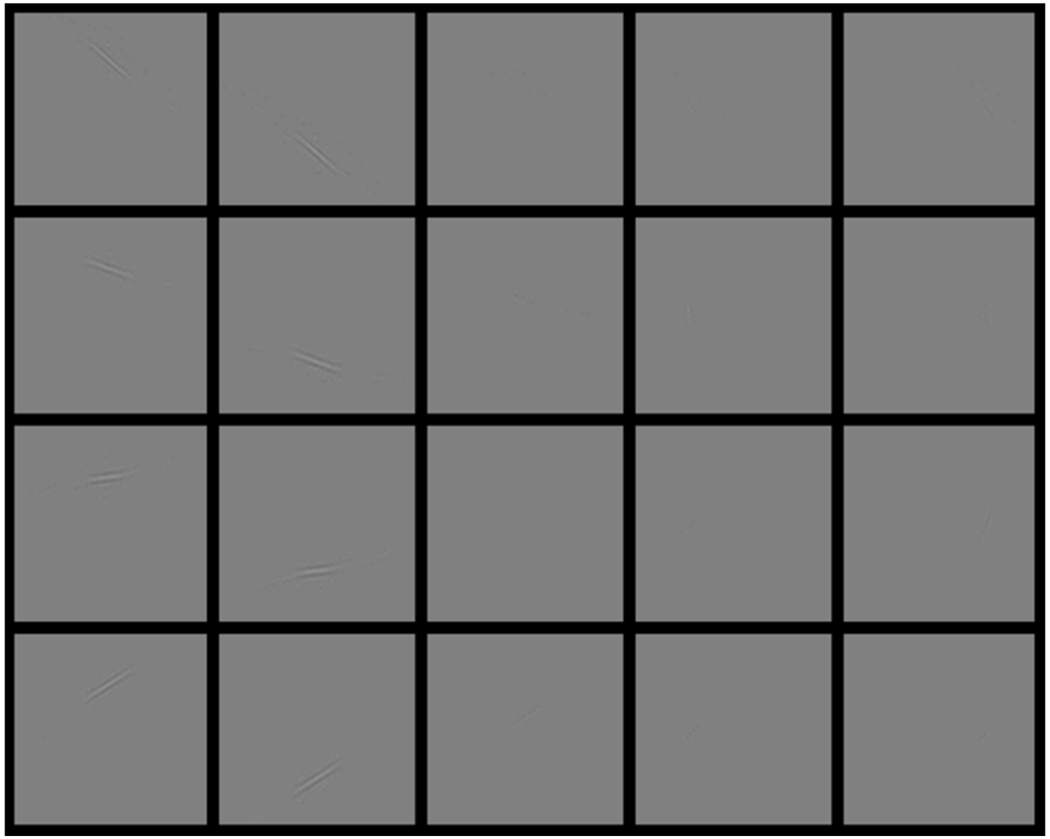
6: Update the coefficients

$$\boldsymbol{\alpha}_\Gamma \leftarrow (\mathbf{D}_\Gamma^T \mathbf{D}_\Gamma)^{-1} \mathbf{D}_\Gamma^T \mathbf{x}.$$

7: **end for**

Аналитические словари

- Curvelets
- Countourlets
  - Surfacelets
- Bandelets
- Complex wavelet transform
- Shearlet
- Grouplet
- Grouped bandelets



Surfacelets

Y. M. Lu and M. N. Do Multidimensional Directional Filter Banks and Surfacelets *IEEE Transactions on Image Processing*, vol. 16, no. 4, April 2007.

Построение словаря

- Пакетные методы
  - Для всего множества исходных данных строим сразу один словарь
- С обучением на лету
  - Анализируем элементы данных по одному, обновляя каждый раз словарь

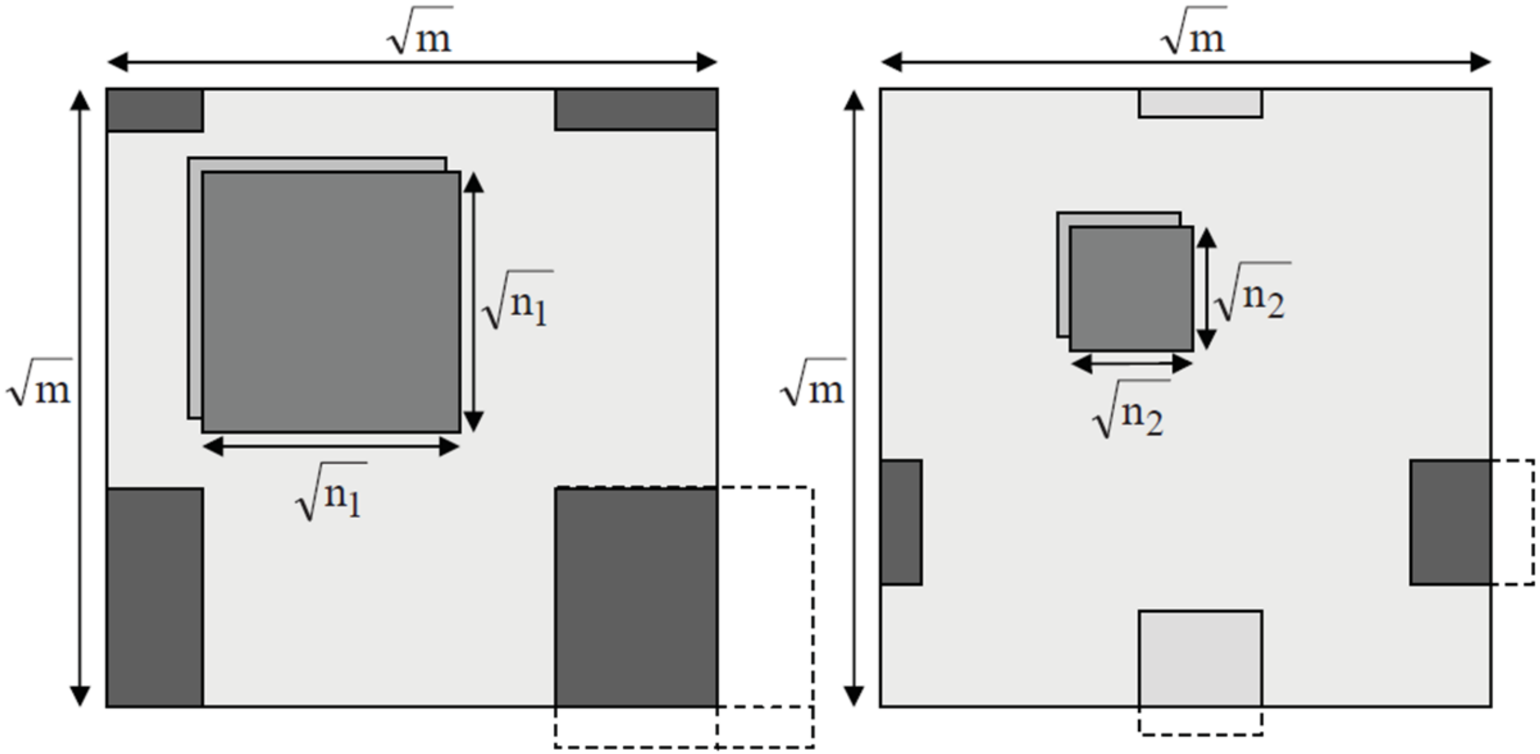
K-SVD

- Обобщение метода K-средних для обучения словаря
- Задача: $\min_{\mathbf{D},\mathbf{X}} \left\{ \|\mathbf{Y} - \mathbf{DX}\|_F^2 \right\} \quad \forall i,\ \|\mathbf{x}_i\|_0 \le T_0$

- Итеративный алгоритм:
  - Инициализация словаря
  - Повторяем:
    - – Получаем разреженное представление X по D, Y
      - » OMP
    - – Для каждого k=1,K
      - » Обновляем атом dk с помощью SVD разложения

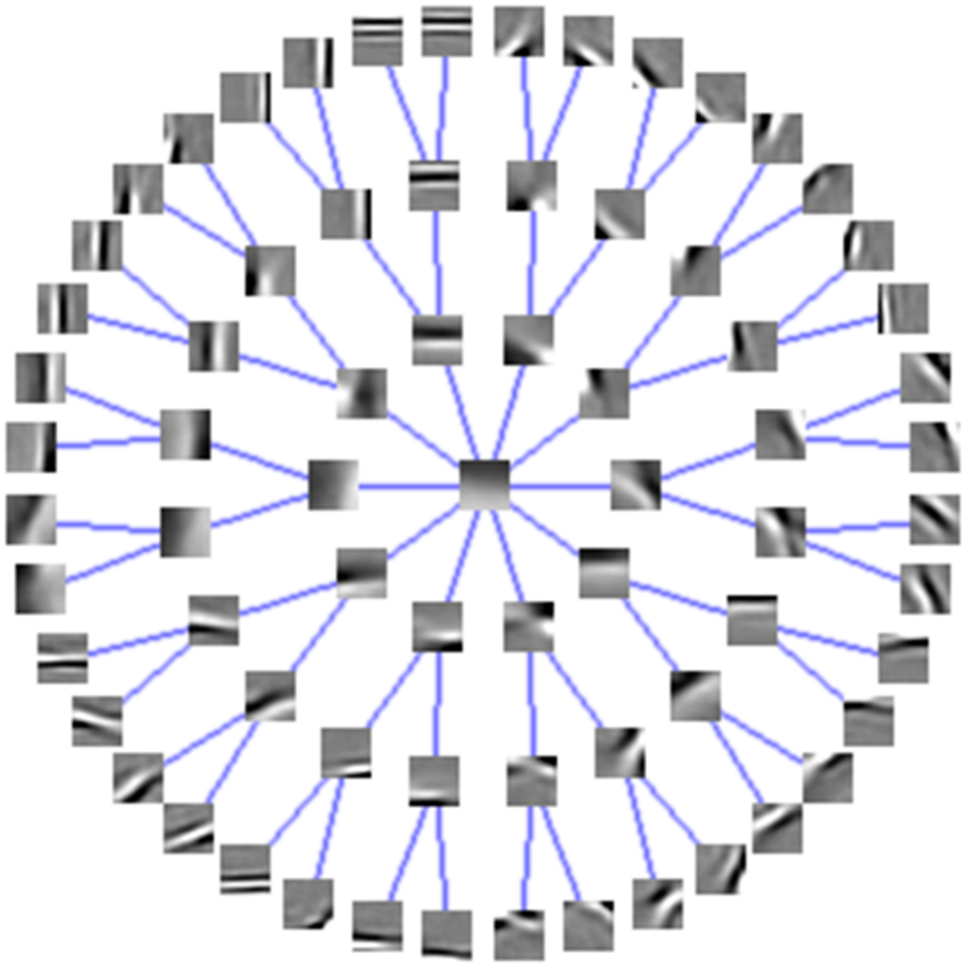
M. Aharon, M. Elad, and A. M. Bruckstein, “The K-SVD: an algorithm for designing of overcomplete dictionaries for sparse representation,” *IEEE Trans. Signal Process., vol. 54, no. 11, pp. 4311–4322, 2006.*

Image-Signature Dictionary



M. Aharon and M. Elad, "Sparse and redundant modeling of image content using an image-signature-dictionary," *SIAM Journal on Imaging Sciences, vol. 1, no. 3, pp. 228–247, 2008.*

Иерархические словари



Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for sparse hierarchical dictionary learning. In Proc. of the International Conference on Machine Learning (ICML), 2010.

Пример: подавление шума

- Построение словаря
  - Построение ISD размером 75x75, патчи 8x8
  - L=2
  - 110000 примеров
- Кодирование
  - Использование OMP
  - Добавляем атомы, пока ошибка не превысит 1.1 * sigma
  - Sigma – оценка шума в исходном изображении

- Вариант 2:
  - Обучение по зашумленным изображениям

Результат

Noisy images

PSNR=24.61dB PSNR=24.61dB

ISD results

PSNR=33.64dB PSNR=32.22dB

Результат

MOD Results

PSNR=33.23dB

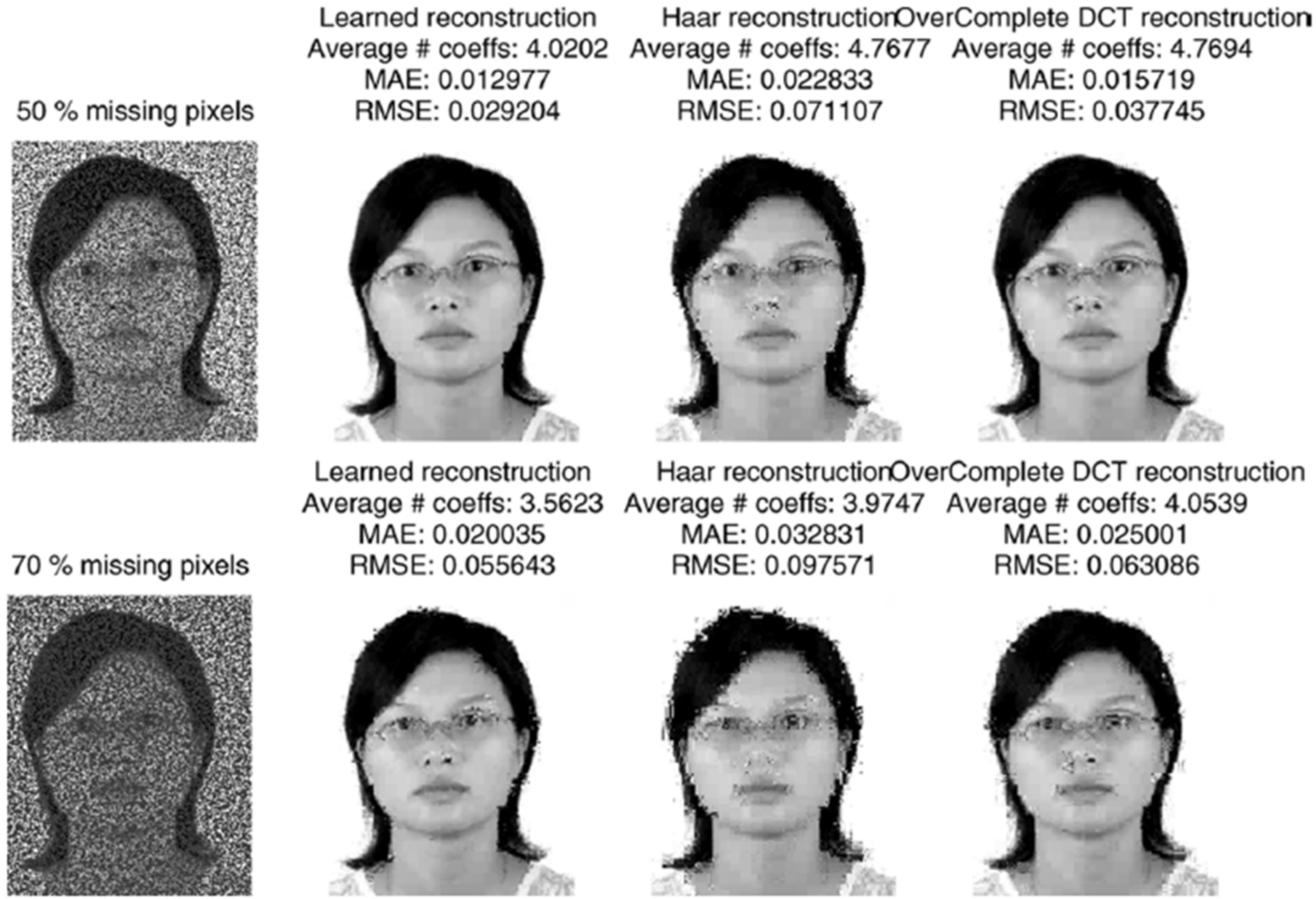
PSNR=31.71dB

Epitome results

PSNR=30.37dB

PSNR=27.38dB

Реконструкция изображения



Реконструкция изображения

Реконструкция изображения

Реконструкция изображения

Реконструкция изображения

Реконструкция изображений

Реконструкция изображений

Детектор краёв

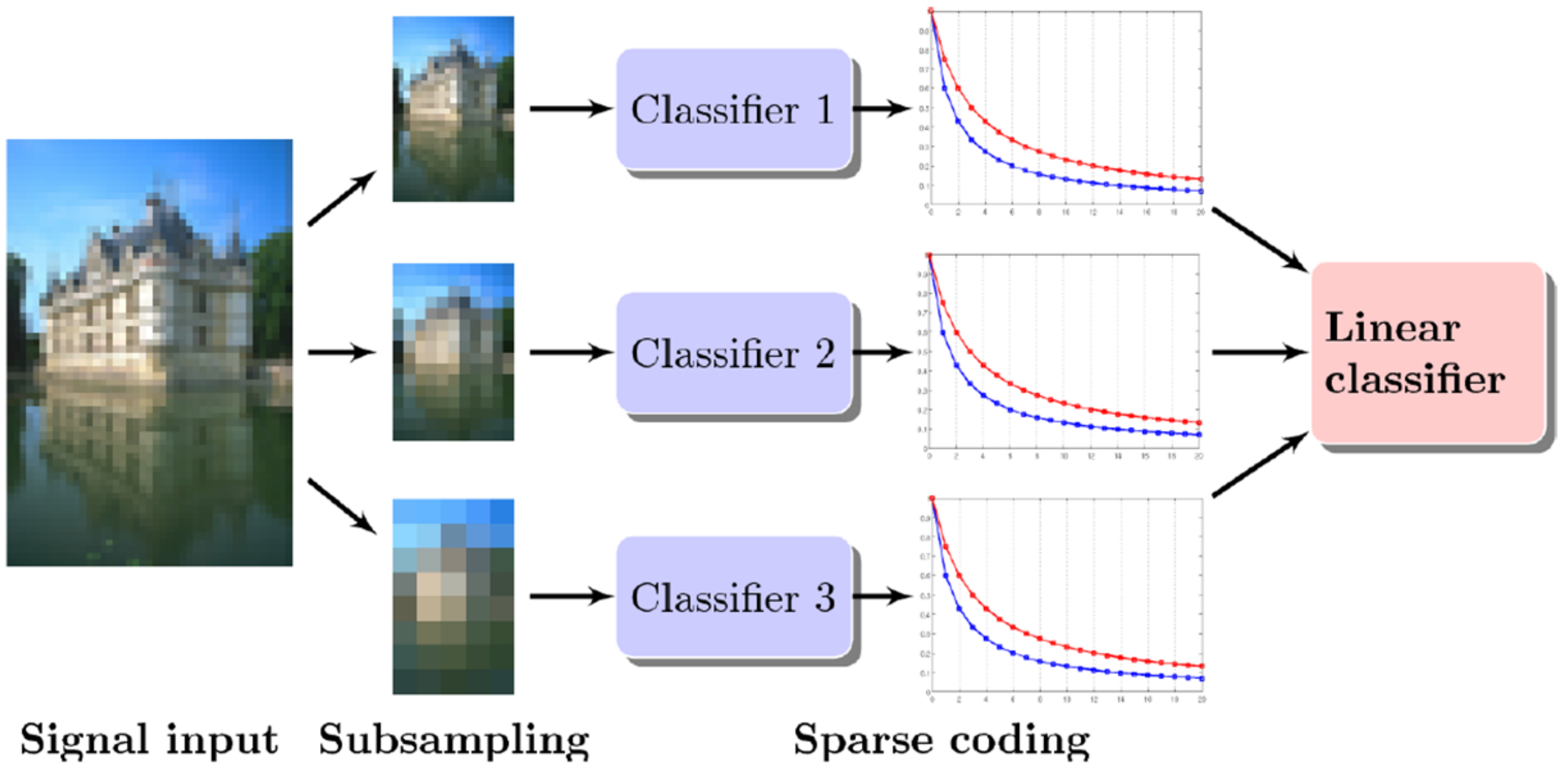
- Задачи:
  - Фильтрация ложных краёв
  - Фильтрация всех краёв, не относящихся к заданной категории объектов
- Идея:
  - Обучим по словарю для каждого класса объектов
  - Новому сигналу назначим класс, реконструкция по которому наиболее точная

$$\mathbf{R}^{\star}(\mathbf{x}, \mathbf{D}) = \min_{\boldsymbol{\alpha} \in \mathbb{R}^p} ||\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}||_2^2 \text{ s.t. } ||\boldsymbol{\alpha}||_0 \leq L.$$

$$\min(\mathbf{R}^{\star}(\mathbf{x}, \mathbf{D}_-), \mathbf{R}^{\star}(\mathbf{x}, \mathbf{D}_+))$$

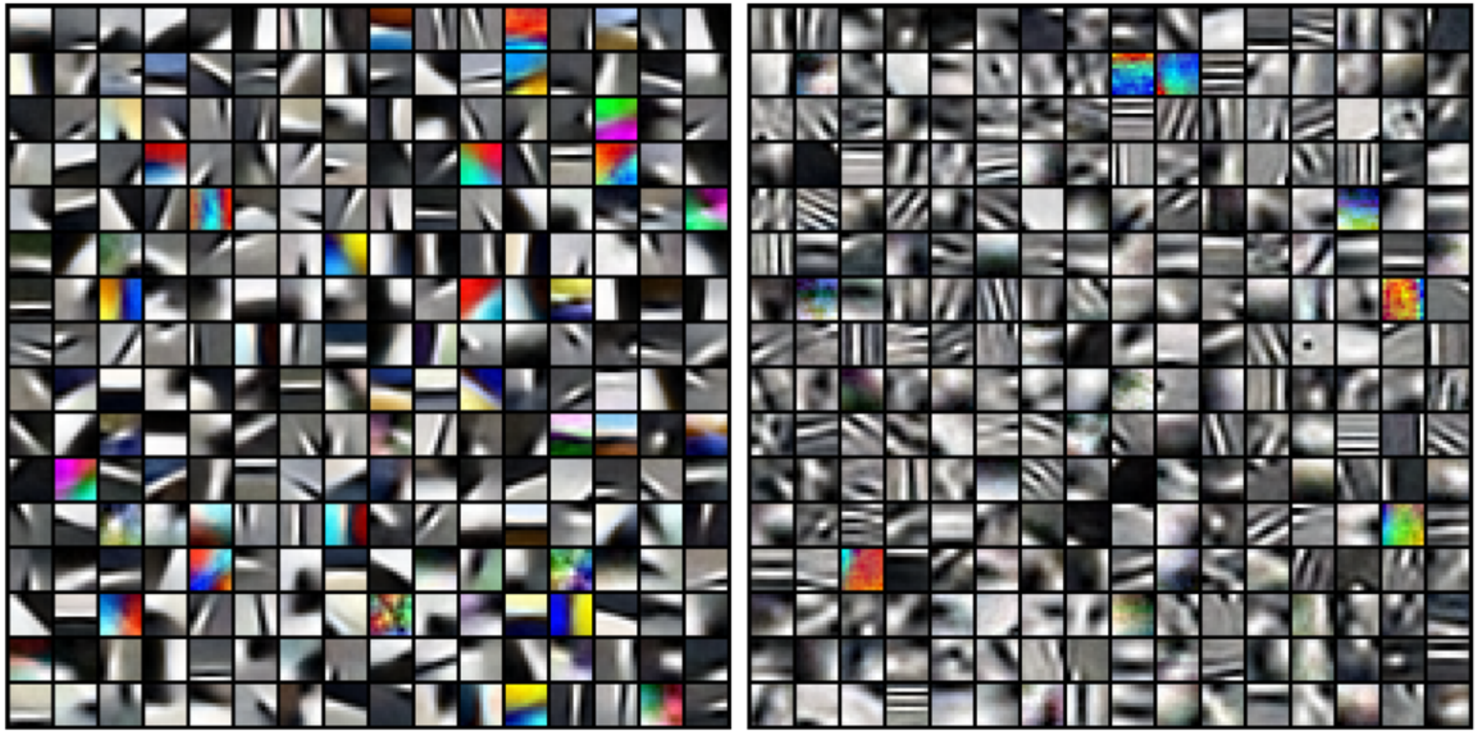
J. Mairal, M. Leordeanu, F. Bach, M. Hebert, and J. Ponce. Discriminative sparse image models for class-specic edge detection and image interpretation. ECCV, 2008

Многомасштабный классификатор



- Построим ROC-кривую по параметру L для реконструкции по каждому словарю
- Построим словари для разных разрешений и разных классов
- Набор ROC-кривых составит вектор-признак для классификации

Фильтр краёв

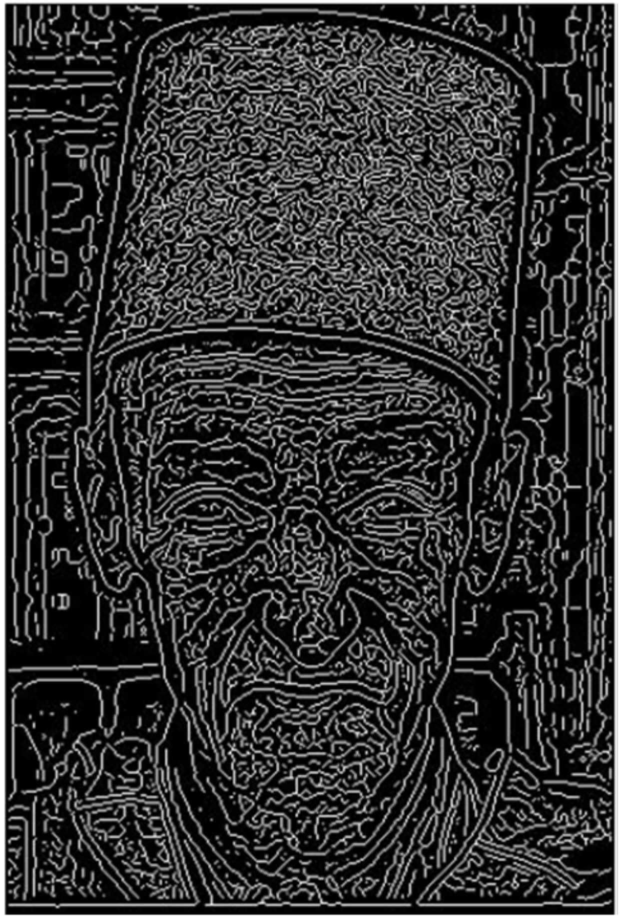


Good edges   Bad edges

- Обучающая выборка – результат Canny, выбор хороших/плохих краёв по отсегментированным пользователем изображениям

Фильтр краёв

Исходное



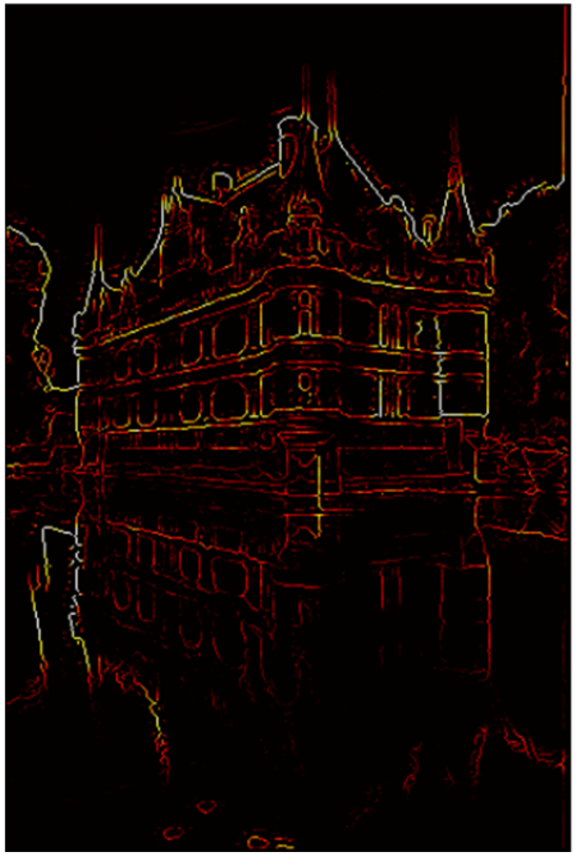
Канни

По словарю

Фильтр краёв

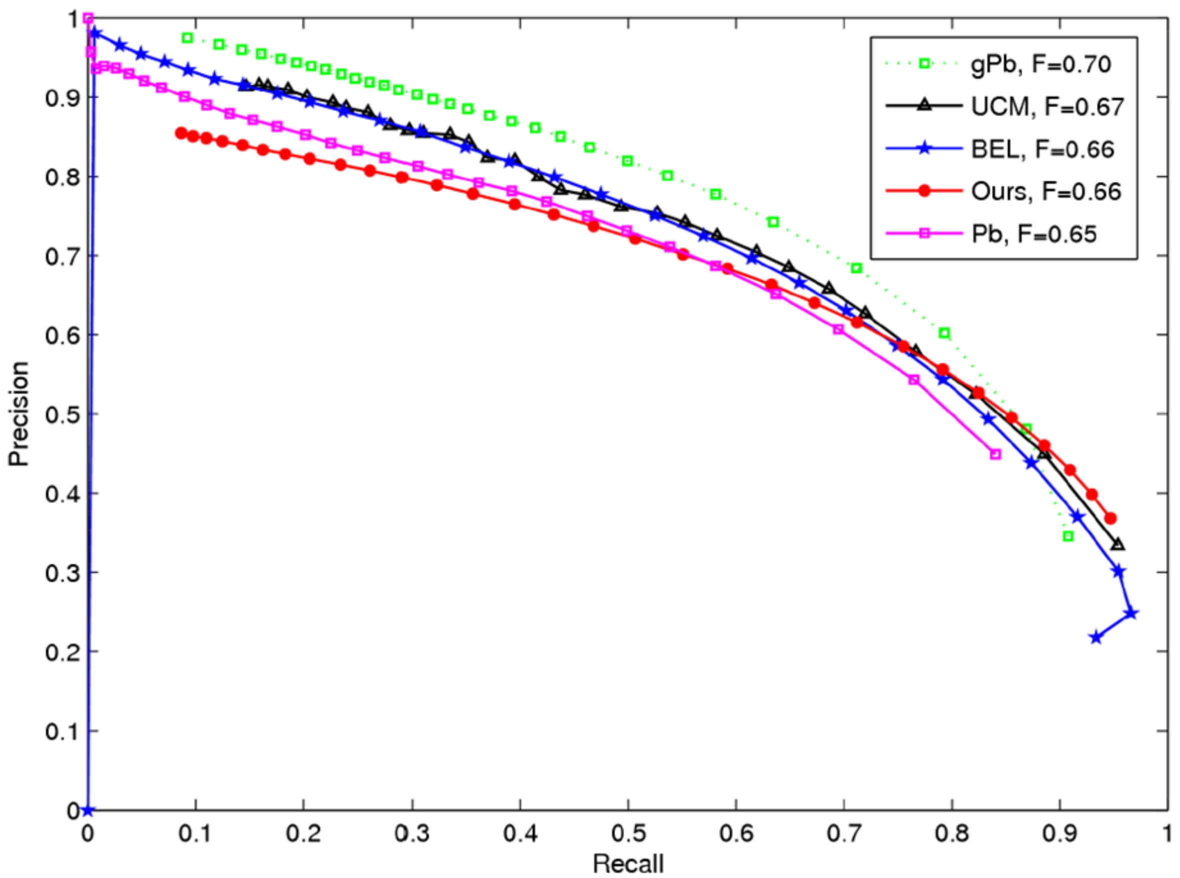
Исходное

Канни



По словарю

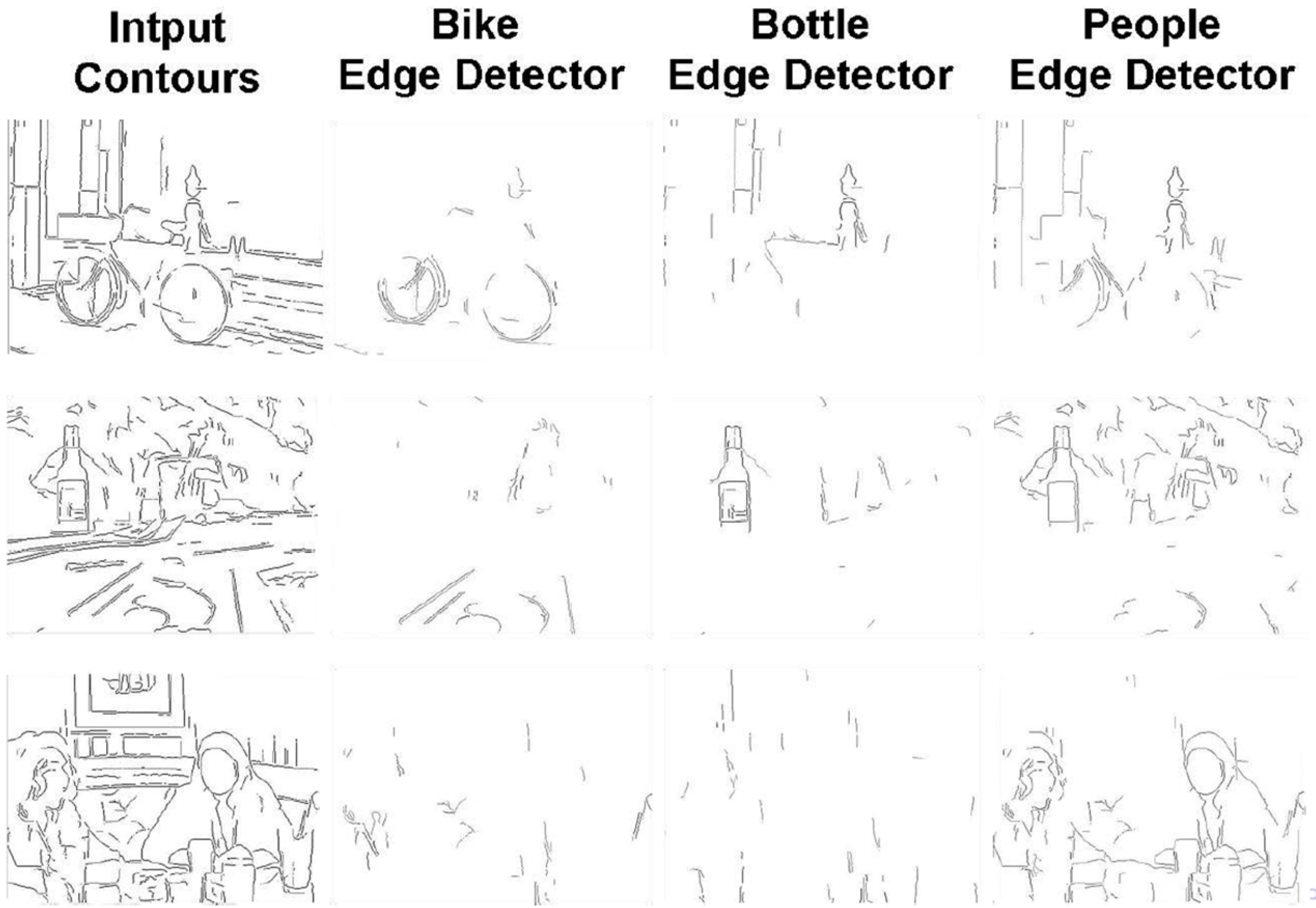
Сравнение с аналогами



- Precision – доля правильной определённых объектов
- Recall – доля найденных объектов из имеющихся в базе

- Чем выше кривая – тем лучше
- Pb, gPb, UCM, BEL – самые современные декторы краёв на изображении
- Большая часть обученных атомов – серые, цветные же обычно включают пару дополнительных цветов

Фильтр краёв для класса



Аутентификация картин

J. M. Hugues, D. J. Graham, and D. N. Rockmore. Quantication of artistic style through sparse coding analysis in the drawings of Pieter Bruegel the Elder. Proceedings of the National Academy of Science, TODO USA, 107(4):1279-1283, 2009

Аутентификация картин

Аутентификация картин

Authentic

Fake

Fake

Аутентификация картин

Authentic

Fake

Authentic

Коды

- Страница Minh N. Do
  - http://www.ifp.illinois.edu/~minhdo/software/
  - Разные преобразования, пирамида Лапласианов и т.д.
- Страница Michael Elad
  - http://www.cs.technion.ac.il/~elad/software/
  - K-SVD и другие алгоритмы для разреженных словарей
- SparseLab
  - http://sparselab.stanford.edu/
- SPArse Modeling Software
  - http://www.di.ens.fr/willow/SPAMS/

На следующей лекции

- Локальные особенности изображения
- Методы поиска локальных особенностей
- Дескрипторы локальных особенностей