Основы цифровой обработки сигналов (DSP)

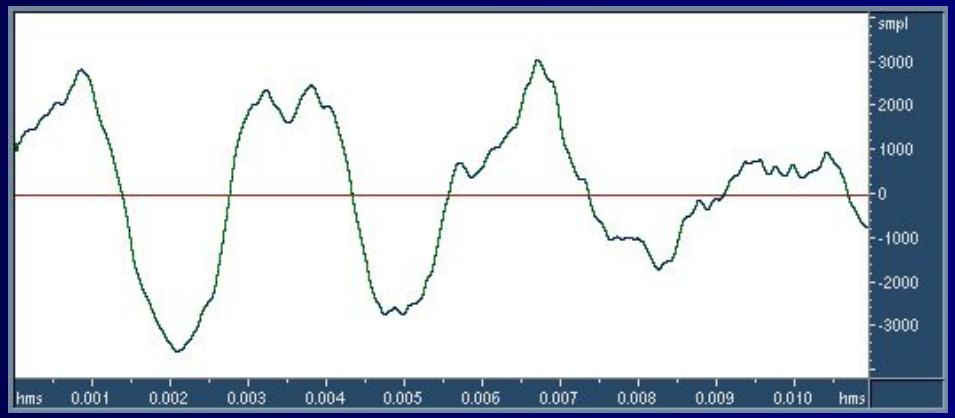
План лекции

- Основные определения
- Дискретизация, теорема Котельникова
- Линейные системы
- Дискретное преобразование Фурье
- Спектральный анализ
- Фильтрация, быстрая свертка
- Приложения

Сигналы

- Сигнал – скалярная функция от одного или нескольких аргументов.

Примеры сигналов



s(t) – звук

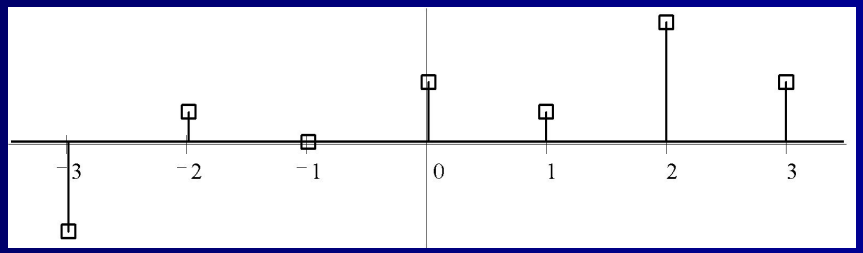
f(x,y) – изображение

Сигналы

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Аналоговые (непрерывные)
  - Примеры:
    - звук в воздухе или в проводе, идущем от микрофона
    - изображение (до ввода в компьютер)
    - запись показаний датчика
- Цифровые (дискретные)
  - Примеры:
    - звук в компьютере (одномерный массив чисел)
    - изображение в компьютере (двумерный массив чисел)
    - запись показаний датчика в компьютере (одномерный массив)

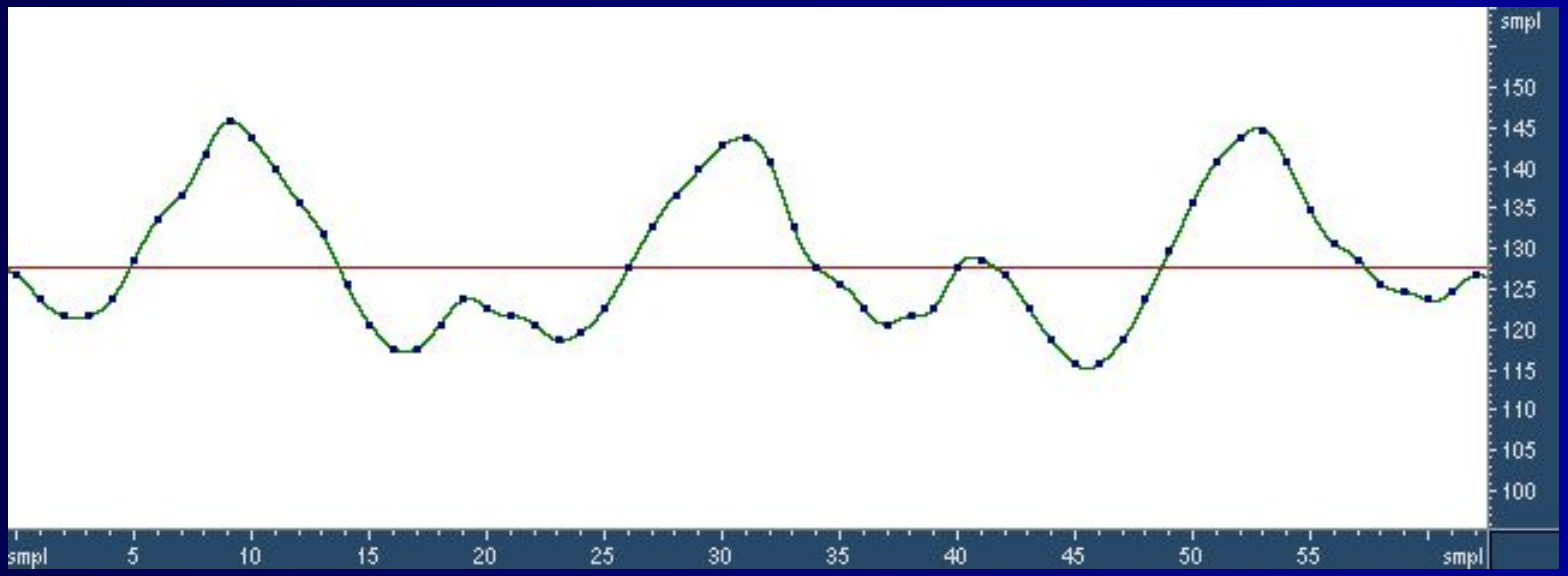
Одномерный
цифровой сигнал



Оцифровка сигналов

GRAPHICS & MEDIA LAB
VIDEO GROUP

1. Дискретизация по времени
2. Квантование по амплитуде



Оцифровка сигналов

GRAPHICS & MEDIA LAB
VIDEO GROUP

- При каких условиях по цифровому сигналу можно точно восстановить исходный аналоговый?
- Предположим, что значения амплитуд в цифровом сигнале представлены точно.
- Введем понятие спектра аналогового сигнала:

$$X(v) = \int_{-\infty}^{+\infty} x(t) \cdot e^{-2\pi i v t} dt \qquad x(t) = \int_{-\infty}^{+\infty} X(v) \cdot e^{2\pi i v t} dv$$

(разложение на синусоиды с различными частотами)

x(t) – исходный сигнал
X(v) – спектр, т.е. коэффициенты при гармониках с частотой v

Теорема Котельникова

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Пусть

1. спектр сигнала x(t) не содержит частот выше F, т.е. X(v)=0 за пределами отрезка [-F, F]
2. дискретизация сигнала x(t) производится с частотой F_s , т.е. в моменты времени nT, здесь T= F_s^{-1}
3. $F_s \geq 2F$

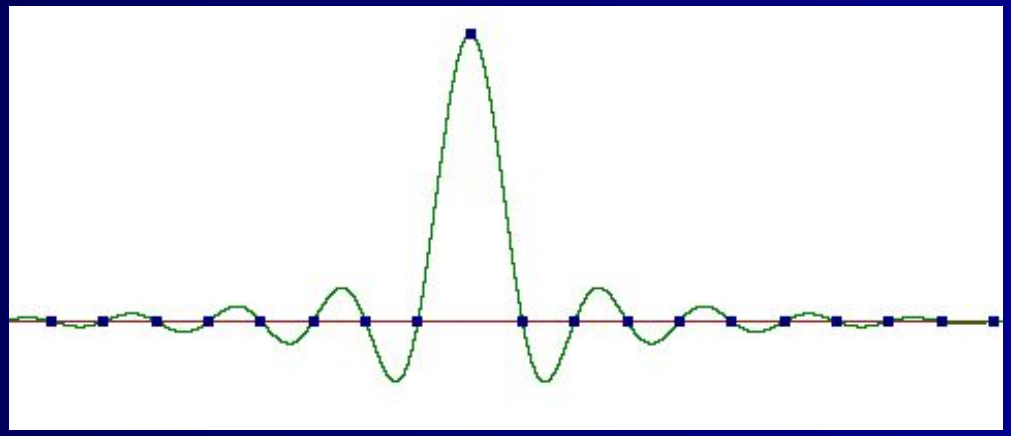
- Тогда исходный аналоговый сигнал x(t) можно точно восстановить из его цифровых отсчетов x(nT), пользуясь интерполяционной формулой

$$x(t) = \sum_{n=-\infty}^{+\infty} x(nT) \cdot Sinc(t - nT) \qquad Sinc(t) = \frac{\sin \pi F_s t}{\pi F_s t}$$

Теорема Котельникова

- Как выглядят интерполирующие sinc-функции?

$$x(t) = \sum_{n=-\infty}^{+\infty} x(nT) \cdot Sinc(t - nT) \qquad Sinc(t) = \frac{\sin \pi F_s t}{\pi F_s t}$$

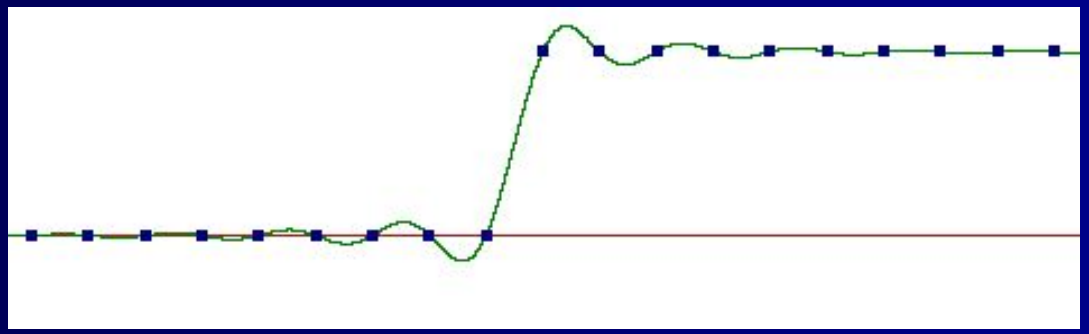


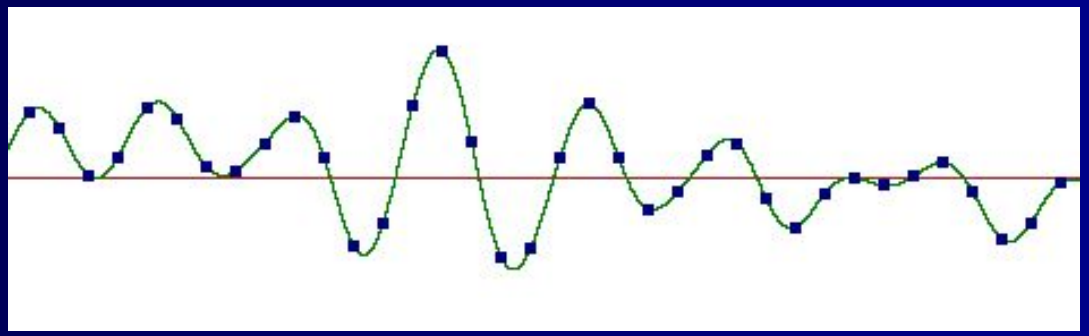
Бесконечно затухающие колебания

Теорема Котельникова

- Реконструкция аналоговых сигналов. Sinc-интерполяция.

$$x(t) = \sum_{n=-\infty}^{+\infty} x(nT) \cdot Sinc(t - nT)$$



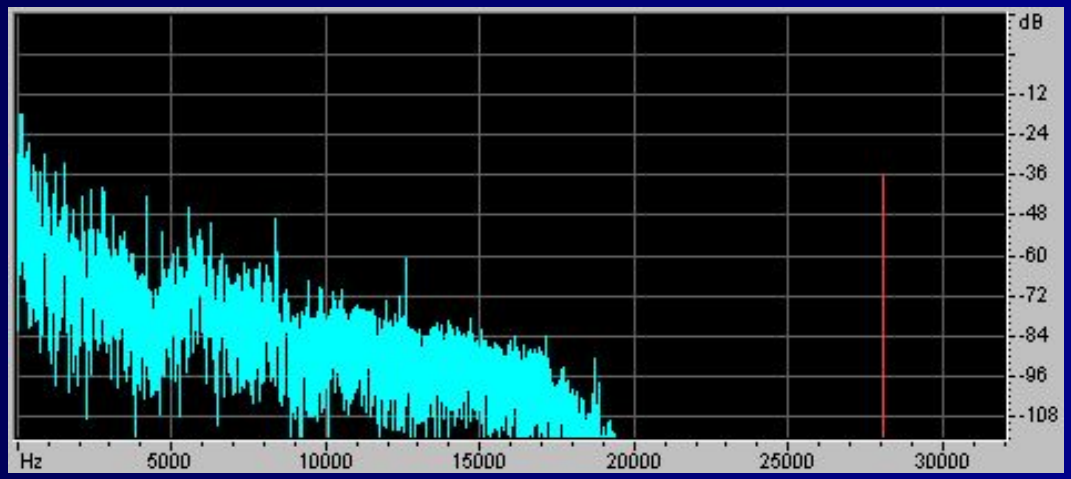


GRAPHICS & MEDIA LAB
VIDEO GROUP

Алиасинг

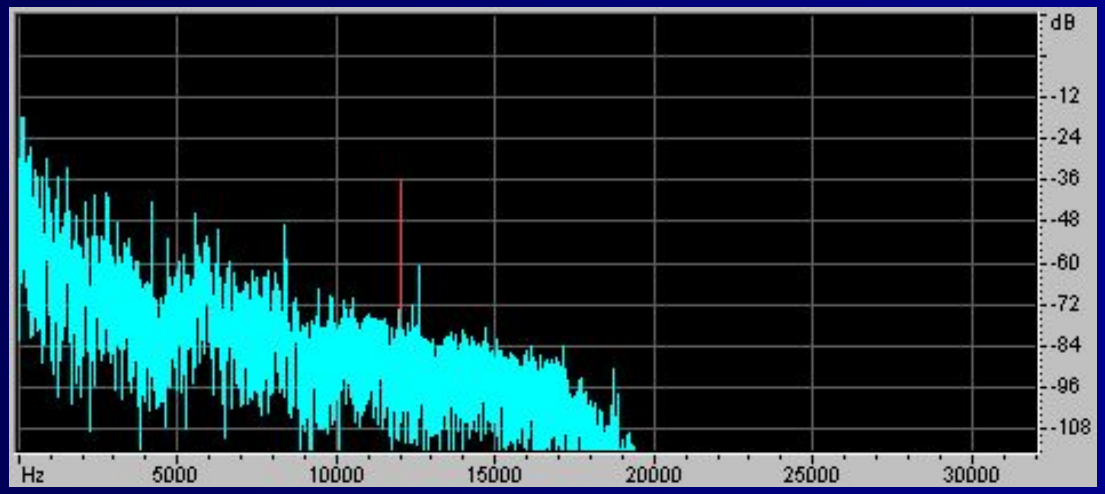
(наложение спектров)

- Что будет, если условия теоремы Котельникова не выполнены?
- Пусть звук не содержит частот выше 20 кГц. Тогда, по теореме Котельникова, можно выбрать частоту дискретизации 40 кГц.
- Пусть в звуке появилась помеха с частотой 28 кГц. Условия теоремы Котельникова перестали выполняться.



Алиасинг

- Проведем дискретизацию с частотой 40 кГц, а затем – восстановим аналоговый сигнал sinc-интерполяцией.



- Помеха отразилась от половины частоты дискретизации в нижнюю часть спектра и наложилась на звук. Помеха переместилась в слышимый диапазон. Алиасинг.

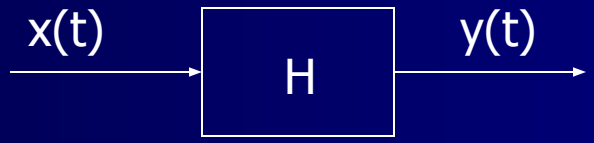
Алиасинг

- Как избежать алиасинга?
- Применить перед оцифровкой анти-алиасинговый фильтр
  - Он подавит все помехи выше половины частоты дискретизации (выше 20 кГц) и пропустит весь сигнал ниже 20 кГц.
  - После этого условия теоремы Котельникова будут выполняться и алиасинга не возникнет.
  - Следовательно, по цифровому сигналу можно будет восстановить исходный аналоговый сигнал.

Линейные системы

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Система – преобразователь сигнала.



$$y(t) = H(x(t))$$

- Линейность:

$$H(\alpha \cdot x(t)) = \alpha \cdot H(x(t))$$

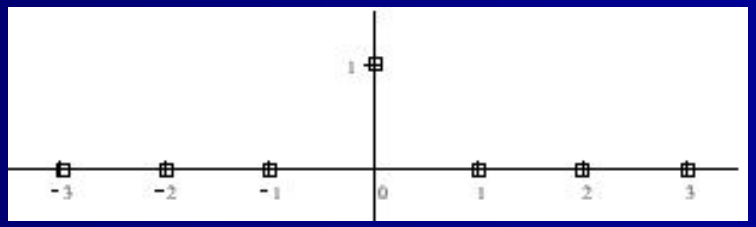
$$H(x(t) + z(t)) = H(x(t)) + H(z(t))$$

- Инвариантность к сдвигу:

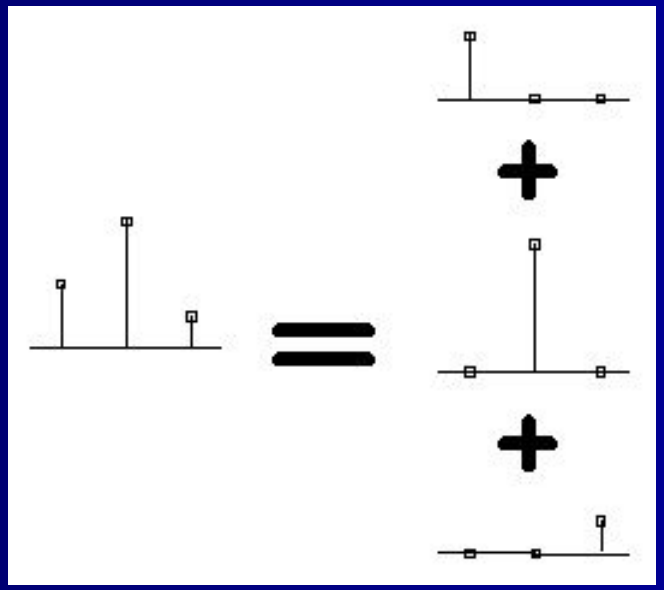
$$H(x(t - t_0)) = y(t - t_0)$$

Импульсная характеристика

- Единичный импульс δ[n]

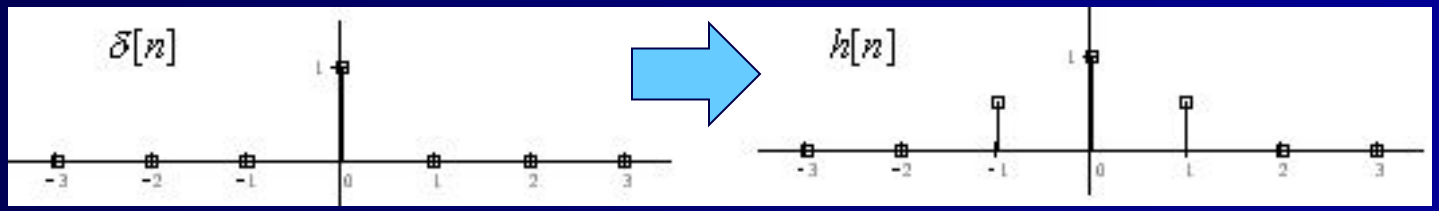


- Разложение произвольного сигнала на взвешенную сумму единичных импульсов



Импульсная характеристика

- Отклик системы на единичный импульс



- h[n] – импульсная характеристика системы (импульсный отклик системы)

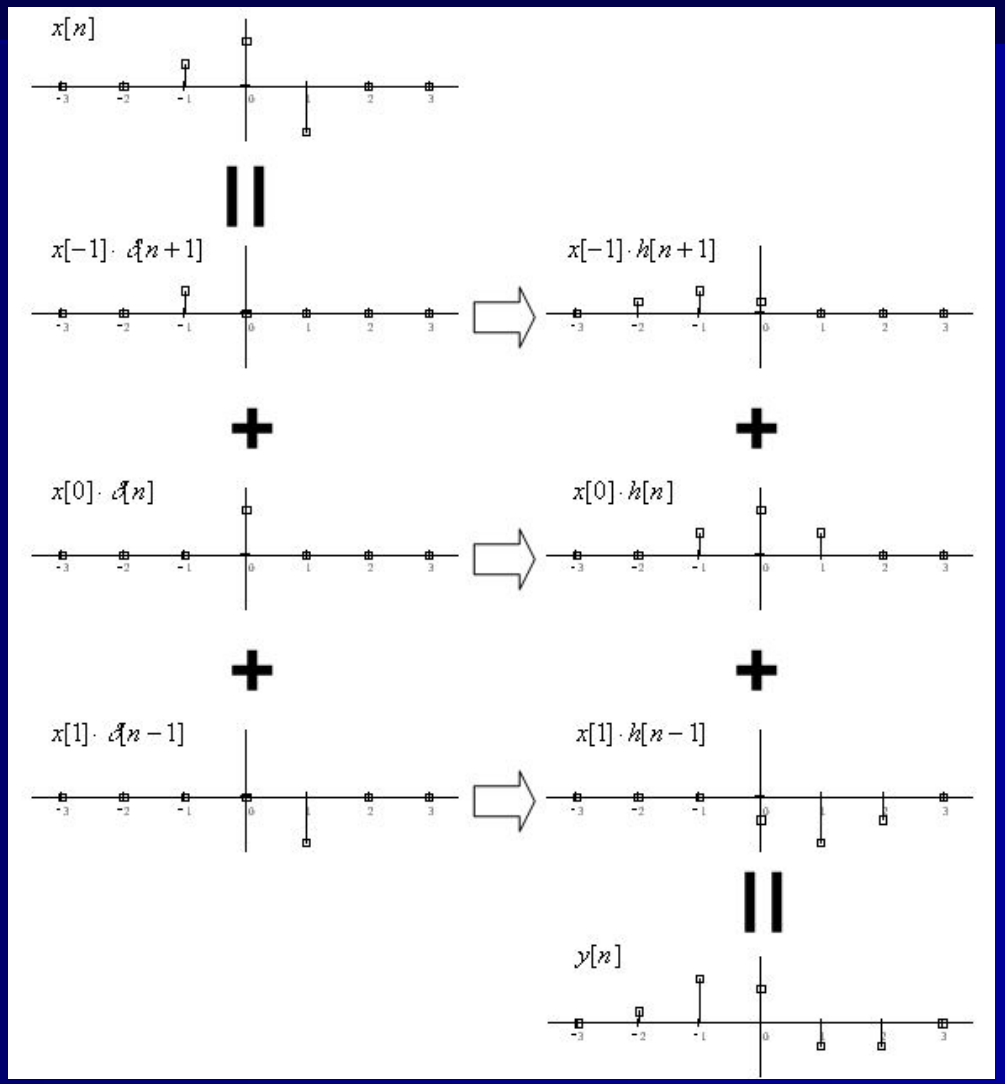
Импульсная характеристика

- Вычисление отклика линейной системы на произвольный входной сигнал
- Свертка

$$y[n] = h[n] * x[n]$$

$$y[n] = \sum_{k=-\infty}^{+\infty} x[n-k] \cdot h[k]$$

h[n] – ядро свертки



Линейные системы

- Итак, любая линейная инвариантная к сдвигу система производит операцию свертки входного сигнала со своей импульсной характеристикой.
- Важное свойство линейных систем:

  При подаче на любую линейную систему синусоиды, на выходе получается синусоида той же частоты, что и на входе. Измениться могут только ее амплитуда или фаза.
- Следствие: линейные системы удобно анализировать, раскладывая любые входные сигналы на синусоиды.

Преобразование Фурье

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Зачем раскладывать сигналы на синусоиды?
  - Анализ линейных систем
  - Слух и синусоиды
  - Хорошо разработана теория и практика
- Дискретное преобразование Фурье (ДПФ)

  Ряд Фурье

$$x[n] = \sum_{k=0}^{N/2} C_k \cos \frac{2\pi k (n + \varphi_k)}{N}$$

  Частоты и амплитуды

$$x[n] = \sum_{k=0}^{N/2} A_k \cos \frac{2\pi k n}{N} + \sum_{k=0}^{N/2} B_k \sin \frac{2\pi k n}{N}$$

- Прямое и обратное преобразования Фурье

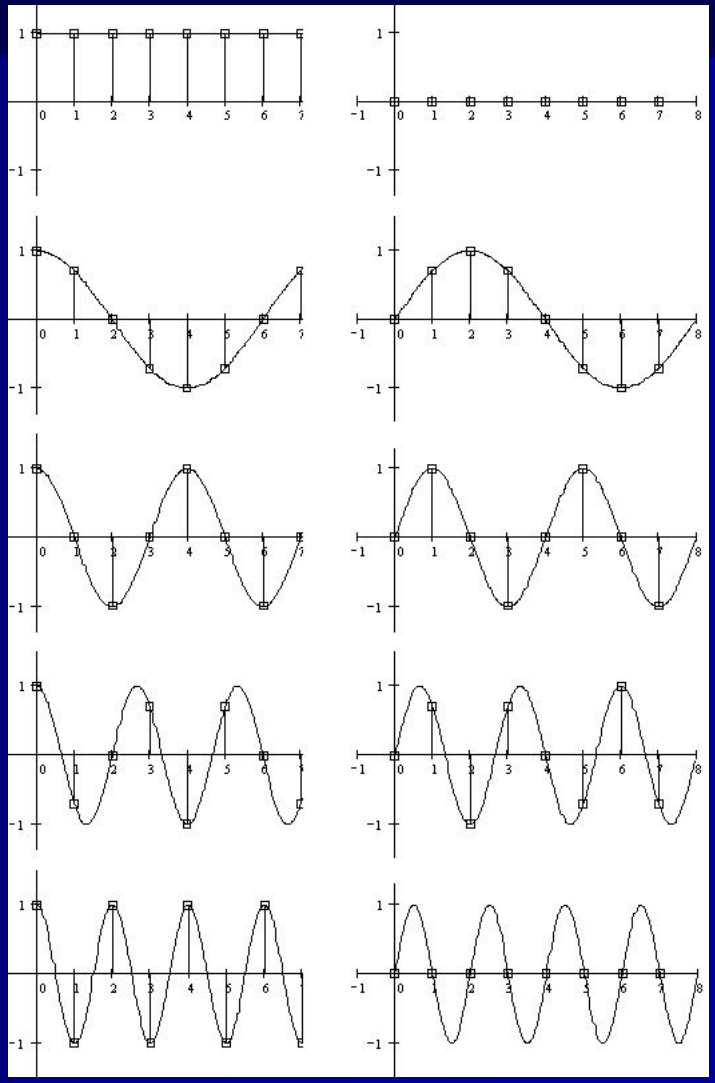
Преобразование Фурье

- Базисные функции дискретного преобразования Фурье для сигнала длины N = 8.

  Имеем N/2 + 1 = 5 различных базисных частот.

  Имеем N+2 базисные функции, 2 из которых тождественно равны нулю.

  Количество информации не изменяется: N чисел



Преобразование Фурье

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Базисные функции образуют N-мерный ортогональный базис в пространстве N-мерных векторов исходных сигналов.
- Следовательно, разложение обратимо, т.е. по коэффициентам разложения (A_k, B_k) можно точно восстановить исходный дискретный сигнал.
- Обратное преобразование Фурье – вычисление суммы конечного ряда Фурье (сложить N штук N-точечных синусоид со своими коэффициентами).

Преобразование Фурье

- Прямое преобразование Фурье – вычисление скалярных произведений сигнала на базисные функции:

$$A_k = \frac{2}{N}\sum_{i=0}^{N-1} x[i]\cos\frac{2\pi ki}{N} \quad k = 1,\ldots,\frac{N}{2}-1$$

$$A_k = \frac{1}{N}\sum_{i=0}^{N-1} x[i]\cos\frac{2\pi ki}{N} \quad k = 0,\frac{N}{2}$$

$$B_k = \frac{2}{N}\sum_{i=0}^{N-1} x[i]\sin\frac{2\pi ki}{N} \quad k = 0,\ldots,\frac{N}{2}$$

- Для вычисления всех коэффициентов по этому алгоритму требуется примерно N^2 умножений: очень много при больших длинах сигнала N.

Преобразование Фурье

- 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 $N = 2^m$

- Существует и обратное БПФ (IFFT) – такой же быстрый алгоритм вычисления обратного ДПФ.

Преобразование Фурье

GRAPHICS & MEDIA LAB
VIDEO GROUP

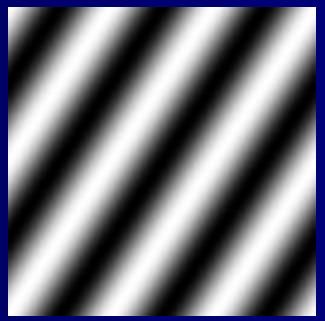
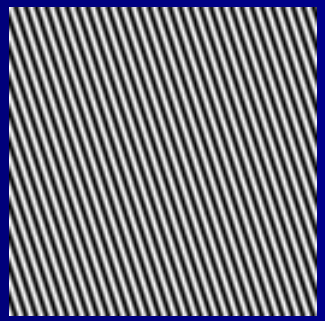
- Входные данные FFT
  - $N = 2^m$, размер FFT
  - Входной вектор длины N, иногда в комплексном представлении
- Выходные данные FFT
  - Коэффициенты A_k и B_k, иногда записанные в комплексном представлении $A_k + iB_k$

Преобразование Фурье

GRAPHICS & MEDIA LAB
VIDEO GROUP

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

$$\sin\left(\frac{2\pi ni}{N}+\frac{2\pi mj}{M}\right)$$

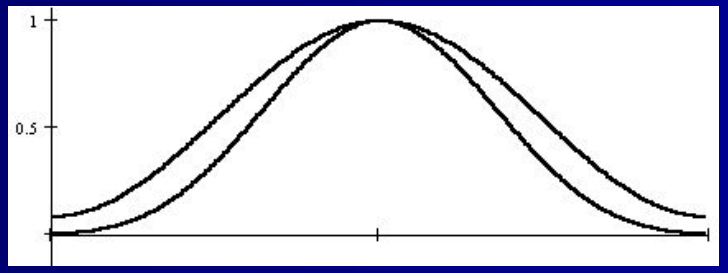
- Вычисление двумерного ДПФ
  1. Прямой способ – скалярные произведения со всеми базисными функциями. Очень много операций.
  2. Быстрый способ – декомпозиция на одномерные ДПФ

Спектральный анализ

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Как вычислить и отобразить спектр сигнала?
  1. Взять нужный отрезок сигнала длины 2^m; если нужный отрезок короче – дополнить его нулями.
  2. Если нужно – устранить из сигнала постоянную составляющую (вычесть константу – среднее значение).
  3. Если нужно – домножить сигнал на весовое окно, плавно спадающее к краям (для уменьшения размытия спектра).
  4. Вычислить FFT.
  5. Перевести комплексные коэффициенты в полярную форму: получить амплитуды.
  6. Отобразить график зависимости амплитуды от частоты.

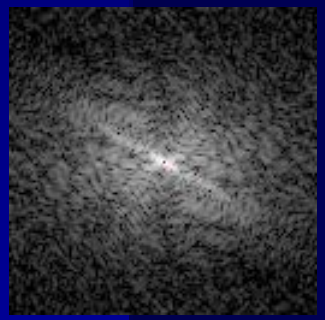
Примеры весовых окон



Спектральный анализ

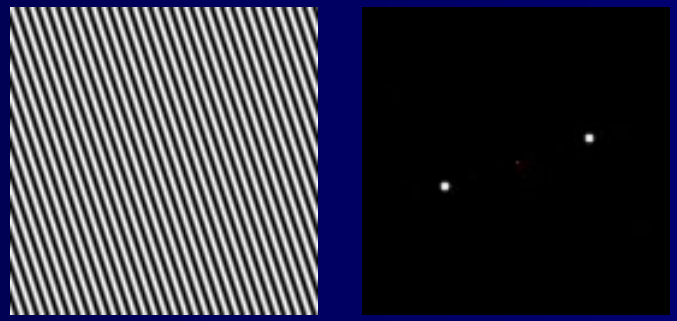
GRAPHICS & MEDIA LAB
VIDEO GROUP

- Отображение спектров изображений
  - Спектр – это картинка, 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.

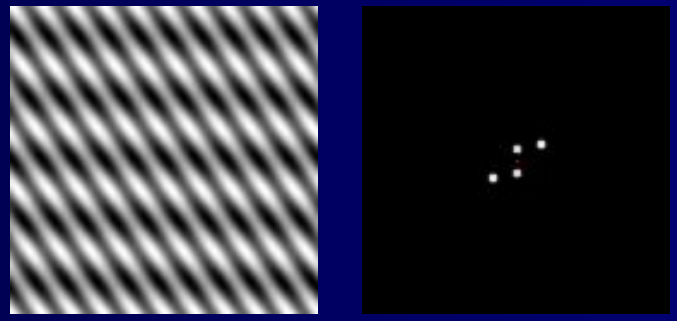


Спектральный анализ

- Примеры изображений и их спектров



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

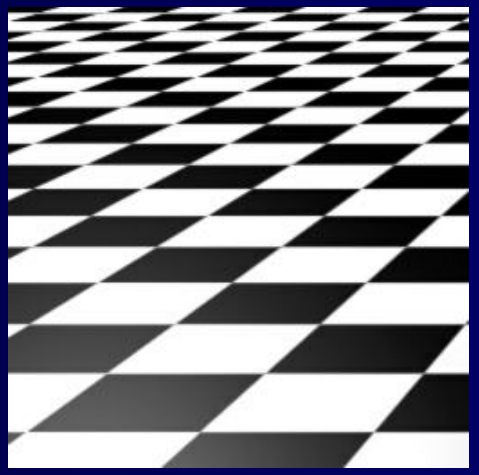


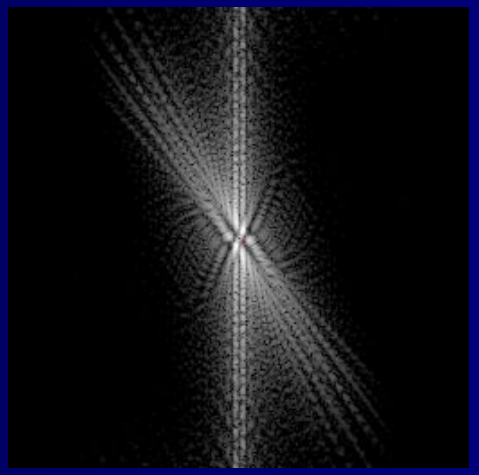
Две синусоиды – две точки

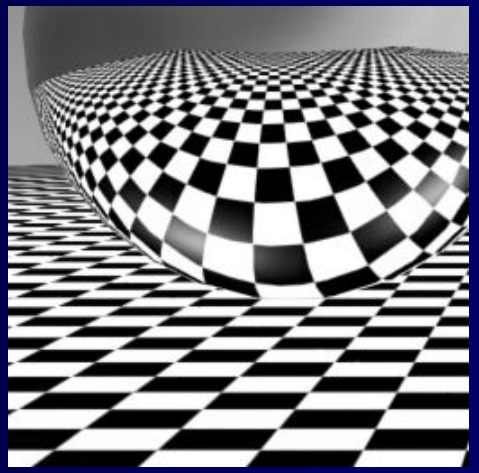
Спектральный анализ

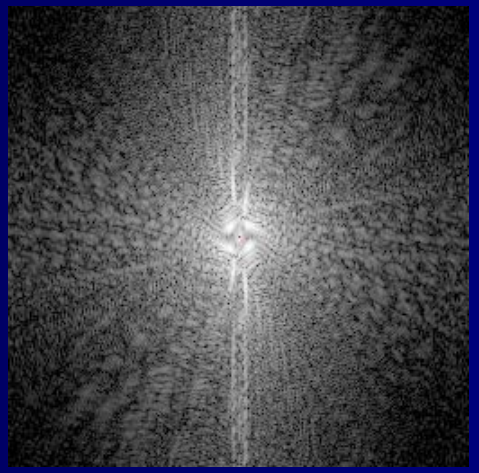
GRAPHICS & MEDIA LAB
VIDEO GROUP

- Примеры изображений и их спектров









По спектру прослеживаются преобладающие направления в исходной картинке

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

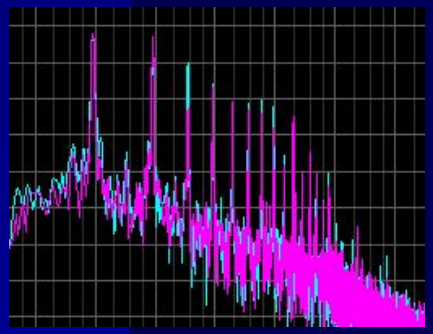
Спектральный анализ

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Отображение спектра звука: спектрограмма
  - Спектрограмма – график зависимости амплитуды от частоты
  - Низкие частоты – слева, высокие – справа
  - Часто применяется логарифмический масштаб частот и амплитуд: “log-log-спектрограмма”
  - Временное и частотное разрешение спектрограммы

Децибелы: $D = 20 \lg \frac{A_1}{A_0}$

A_1 – амплитуда измеряемого сигнала,
A_0 – амплитуда сигнала, принятого за начало отсчета (0 дБ)



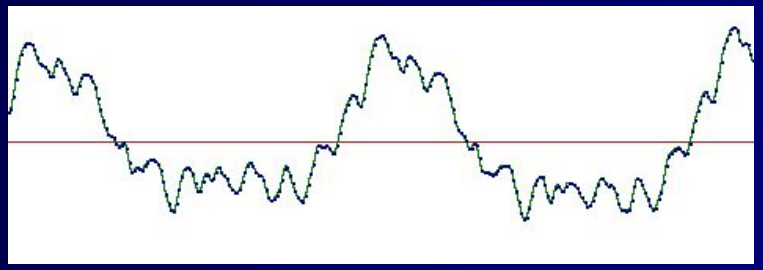
Разница на 6 дБ – разница по амплитуде в 2 раза,
разница на 12 дБ – разница по амплитуде в 4 раза.

Часто за 0 дБ принимается либо самый тихий слышимый звук, либо самый громкий звук, который может воспроизвести аудио-устройство.

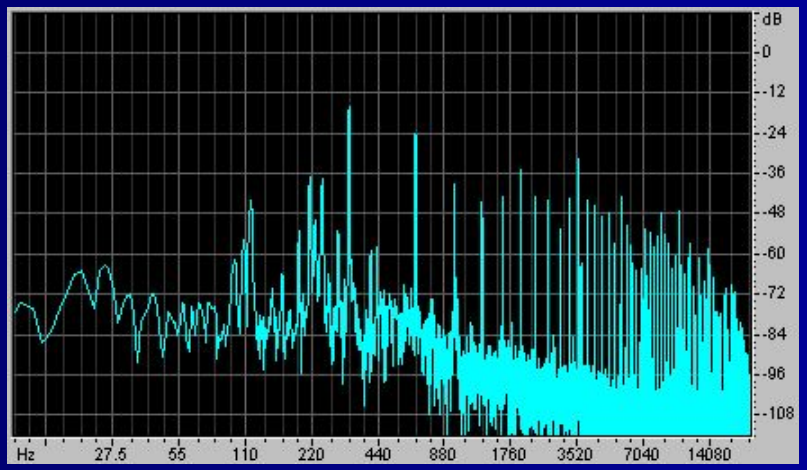
Спектральный анализ

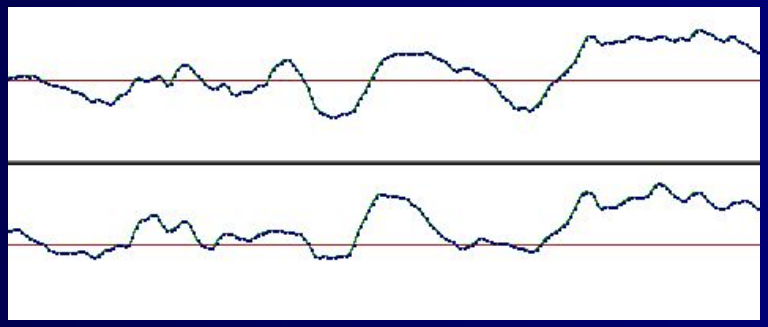
GRAPHICS & MEDIA LAB
VIDEO GROUP

- Примеры звуков и их спектров

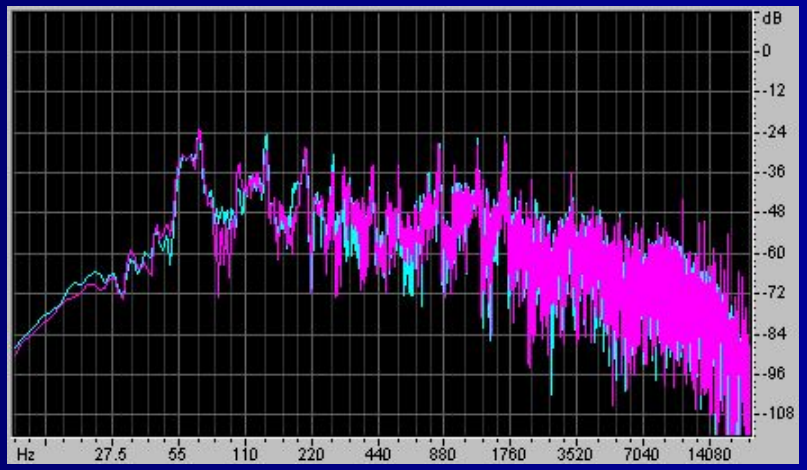


Нота на гитаре





Песня (стерео запись)

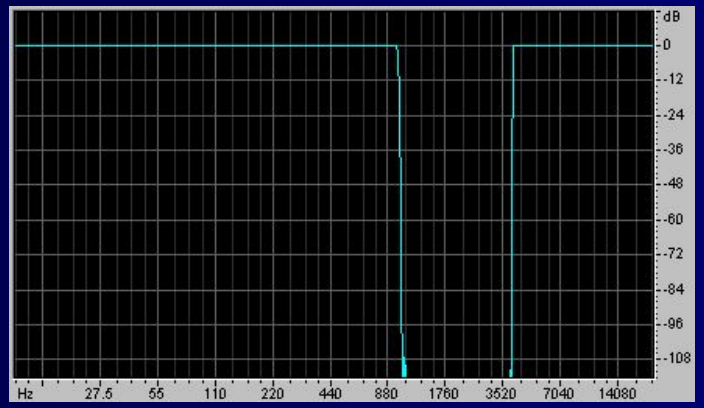


Быстрая свертка

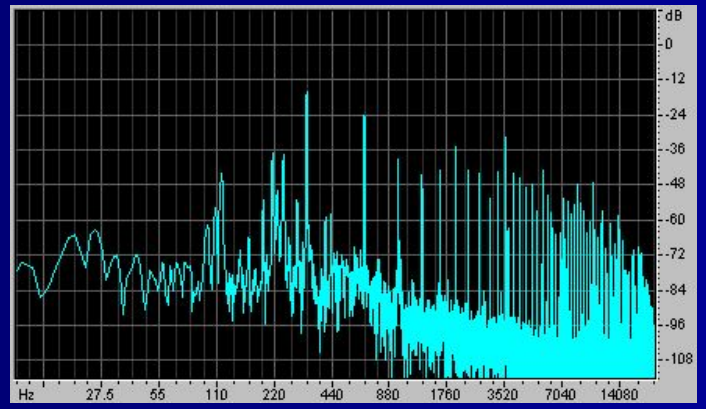
- Прямое вычисление: M·N умножений (M – размер ядра свертки, N – длина сигнала)
- Теорема свертки: свертка во временной области эквивалентно умножению в частотной области, умножение во временной области эквивалентно свертке в частотной области.
- Алгоритм быстрой свертки:
  1. Вычислить спектры сигнала и ядра свертки (FFT)
  2. Перемножить эти спектры
  3. Вернуть полученный спектр во временную область (IFFT)
- Почему это быстрее? Потому что переход в частотную область и обратно быстрый: FFT

Фильтрация

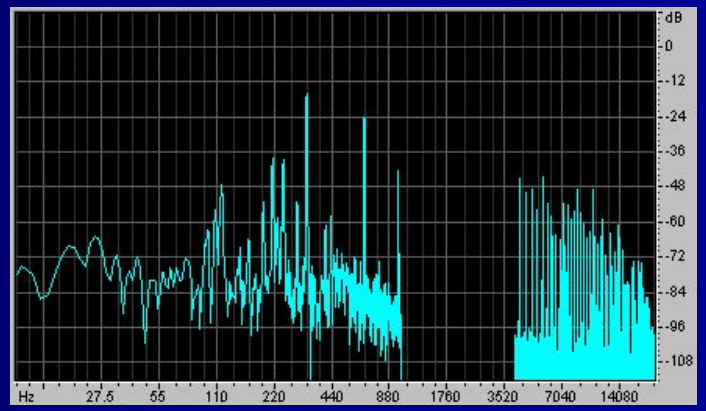
- Спектры сигналов при свертке перемножаются
- Следовательно, свертка (фильтрация) меняет спектр сигнала



*

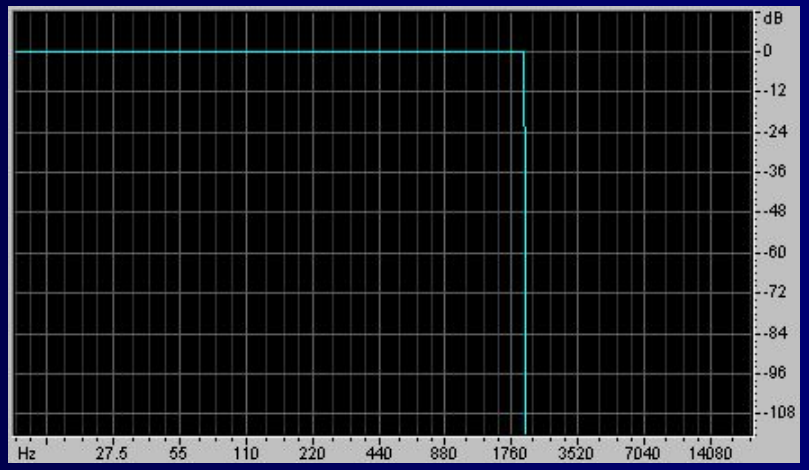


Перемножение амплитуд – сложение децибелов

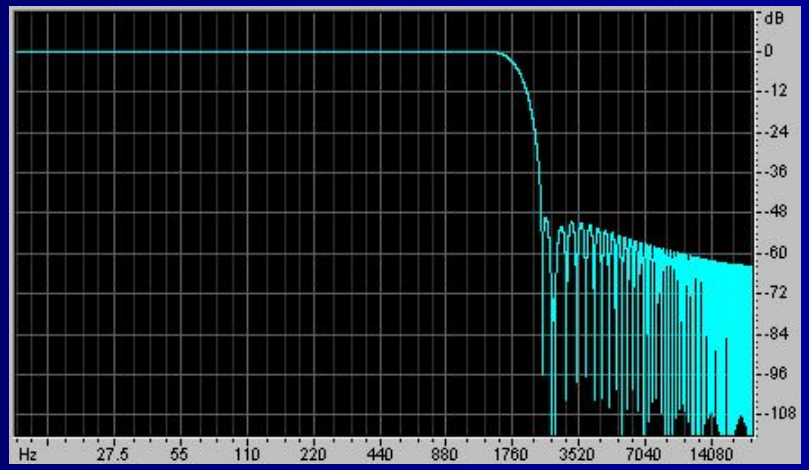


Фильтрация

- Частотная характеристика фильтра (АЧХ)
  - Полосы пропускания (pass-band), подавления (stop-band), среза (transition band)
- Линейность фазы
- Длина фильтра
- Проектирование фильтров



Идеальный НЧ-фильтр



Один из реальных НЧ-фильтров

Фильтрация

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Применения фильтрации
  - Подавление помех и шумов
  - Анти-алиасинг
  - Улучшение качества звука, компенсация искажений звуковой аппаратуры, творческие задачи в звукозаписи
  - Обработка изображений: эффекты, коррекция
  - Фильтрация – составная часть многих других, более сложных алгоритмов

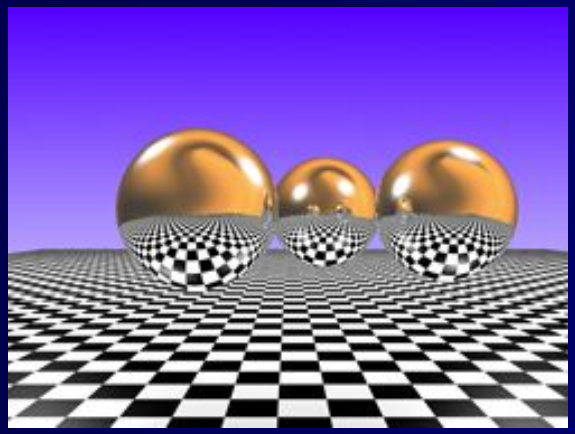
Другие применения DSP

- Компрессия изображений (JPEG, JPEG-2000)
- Компрессия аудио (mp3)
- Мобильная телефония
- Звукозапись
- Шумоподавление
- Обработка и распознавание речи

- и многое другое

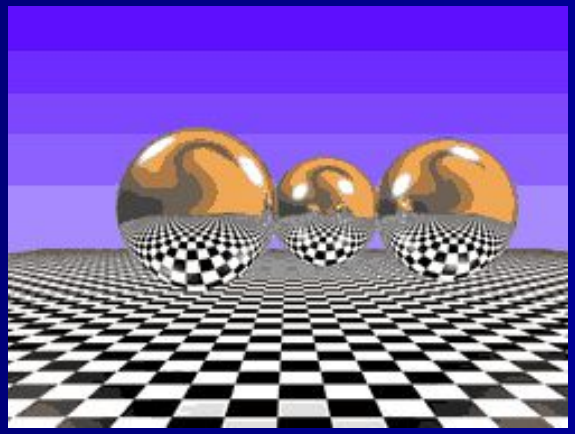
Псевдотонирование

- Цель: уменьшить видимые артефакты палитризации

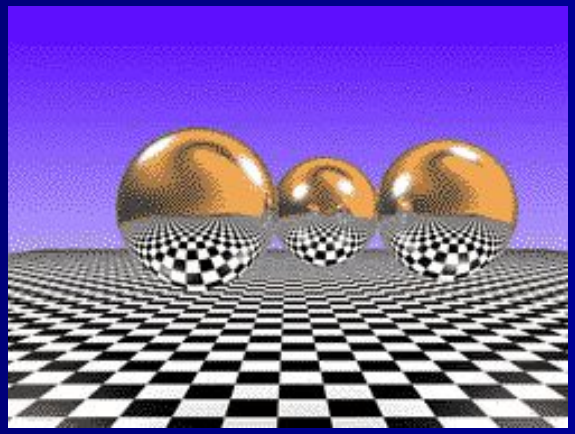


RGB

Округление



Псевдотонирование



16 цветов

Псевдотонирование

- 1-й шаг – сведение к градациям серого

$$Mono = 0.3 \cdot red + 0.59 \cdot green + 0.11 \cdot blue$$

Псевдотонирование

- 1-й шаг – сведение к градациям серого

$$Mono = 0.3 \cdot red + 0.59 \cdot green + 0.11 \cdot blue$$

Псевдотонирование

- Методы
  - Округление

$$Dst[x, y] = truncate(Mono[x, y])$$

Псевдотонирование

- Методы
  - Dithering (добавление шума)

$$Dst[x,y] = truncate(Src[x,y] + Noise[x,y])$$

Белый шум – случайные числа с нулевым мат. ожиданием

Псевдотонирование

- Методы
  - Упорядоченное псевдотонирование

1. Изображение разбивается на блоки
2. В каждом блоке вычисляется средняя интенсивность
3. В зависимости от интенсивности выбирается нужный шаблон
4. Шаблон записывается в блок

Примеры шаблонов
с разными степенями заполнения:

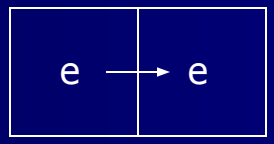
Псевдотонирование

- Методы
  - Диффузия ошибки

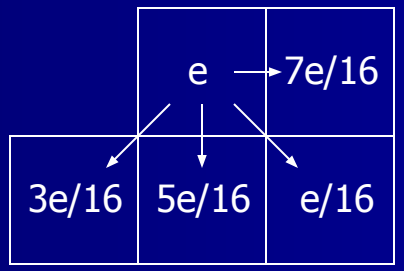
```
for (i=0; i<Height; i++)
  for (j=0; j<Width; j++) {
    Dest[i][j] = quantize(Src[i][j]);
    e = Dest[i][j] - Src[i][j];
    Src[i][j+1] -= e;
  }
```

Идея алгоритма: ошибка, внесенная при квантовании текущего пикселя, распределяется между соседними (еще не квантованными) пикселями.

Примеры видов распределения ошибки:



простейший



Floyd-Steinberg

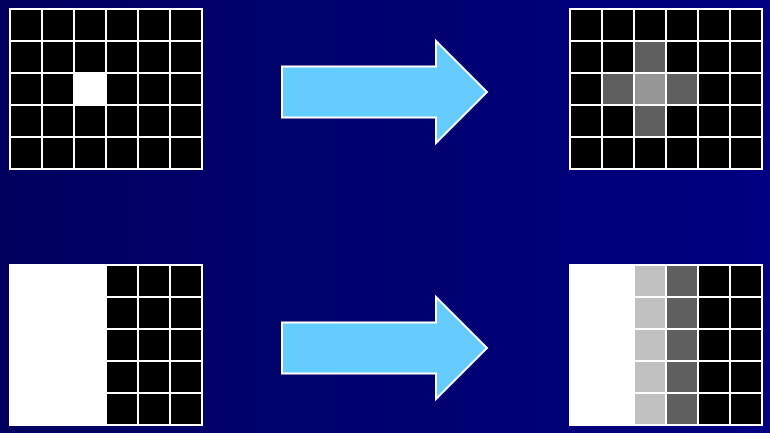
Фильтры

- Как работают фильтры

Коэффициенты фильтра,
ядро свертки 3x3,
«функция размытия точки»

$$Ker[k,p] = \frac{1}{6} \cdot \begin{Vmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{Vmatrix}$$

$-1 \le k \le 1$,
$-1 \le p \le 1$



Фильтры

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Свертка

$$Dst[i,j] = Src[i,j] * Ker[k,p]$$

$$Dst[i,j] = \sum_{k,p} Src[i-k, j-p] \cdot Ker[k,p]$$

```
// Обнулить изображение Dest[i][j]
...
// Выполнить свертку
for (i=0; i<Height; i++)        // Для каждого пикс. Dest[i][j]...
  for (j=0; j<Width; j++)
    for (k=-1; k<=1; k++)       // ...превратить его в ядро свертки
      for (p=-1; p<=1; p++)
        Dest[i+k][j+p] += Src[i][j] * Ker[k][p];    // и сложить
```

Подводные камни:

- Выход за границы массива
- Выход за пределы допустимого диапазона яркости пикселей
- Обработка краев.

Фильтры

- Свойства фильтров
  1. Результат фильтрации однотонного (константного) изображения – константное изображение. Его цвет равен

$$Dest = Src \cdot \sum_{k,p} Ker[k, p]$$

  2. Следствие: чтобы фильтр сохранял цвет однотонных областей, нужно чтобы

$$\sum_{k,p} Ker[k, p] = 1$$

  3. Следствие: если сумма коэффициентов фильтра равна нулю, то он переводит однотонные области в нулевые.

Примеры фильтров

- Размытие (blur)

Примеры фильтров

- Повышение четкости (sharpen)

Примеры фильтров

- Нахождение границ (edges)

Примеры фильтров

- Тиснение (embossing)

Примеры фильтров

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Простейшее размытие

$$Ker[k,p] = \frac{1}{15} \cdot \begin{Vmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 1 \end{Vmatrix}$$

- Константное размытие
  “box-фильтр”
  (любой размер фильтра)

$$Ker[k,p] = \frac{1}{Sum}$$

- Гауссово размытие
  (любой размер фильтра)

$$Ker[k,p] = \frac{1}{Sum} \cdot \exp \frac{k^2 + p^2}{-2\sigma^2}$$

Примеры фильтров

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Повышение резкости

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

- Нахождение границ

$$\begin{Vmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{Vmatrix}$$

+ модуль, нормировка, применение порога…

- Тиснение

$$\begin{Vmatrix} 0 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & 0 \end{Vmatrix}$$

+ сдвиг яркости, нормировка…

Фильтры

- Некоторые свойства свертки

  Пусть *X* и *Y* – изображения, *H* – ядро свертки

1. Линейность

$$(const \cdot X) * H = const \cdot (X * H)$$

$$(X + Y) * H = (X * H) + (Y * H)$$

2. Инвариантность к сдвигу

$$(X[i - i_0, j - j_0] * H = (X * H)[i - i_0, j - j_0]$$

Фильтры

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Сепарабельные (разделимые) фильтры

$$Ker[k, p] = F(k) \cdot G(p)$$

Если фильтр сепарабельный, то фильтрацию можно производить быстрее:

1. Отфильтровать все столбцы одномерным фильтром F(k)
2. Отфильтровать все строки одномерным фильтром G(p)

Гауссиан – сепарабельный фильтр, т.к.

$$Gauss[k, p] = \frac{1}{Sum} \cdot \exp\frac{k^2}{-2\sigma^2} \cdot \exp\frac{p^2}{-2\sigma^2}$$

Еще один сепарабельный фильтр – box-фильтр

Фильтры

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Медианный фильтр
  - Каждый пиксель принимает значение, являющееся медианой значений пикселей в окрестности
  - Медиана – средний элемент в отсортированном массиве
  - Позволяет подавить шум (особенно, единичные «выпадающие» пиксели), не размывая границ
  - Медианный фильтр нелинейный (как доказать?)

  - Векторная медиана – такой элемент массива, для которого сумма L1-расстояний до остальных элементов минимальна (для одномерного случая – совпадает с предыдущим определением)

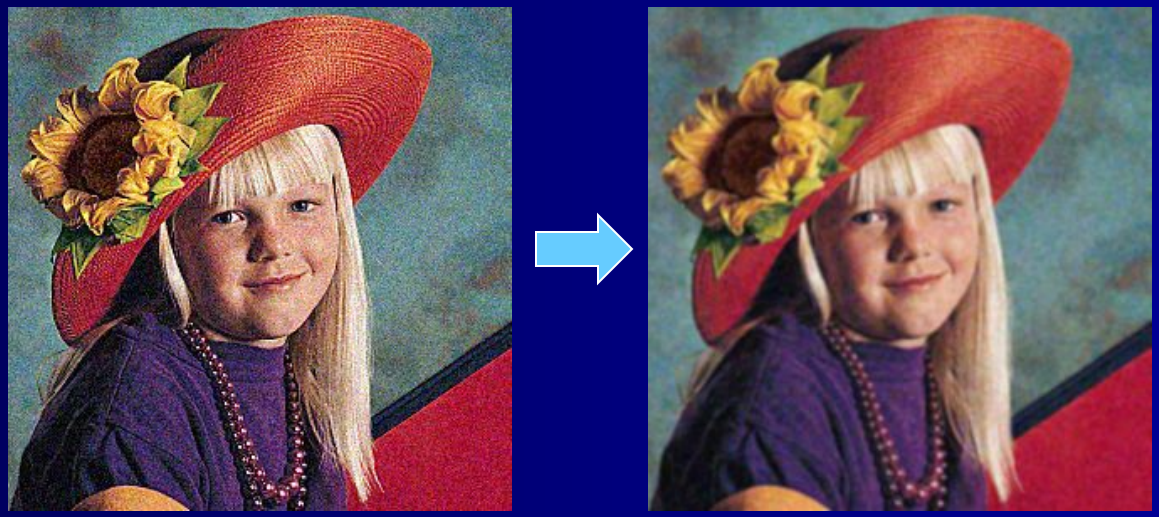
Фильтры

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Понятие о частотах в изображении и звуке
  - Частоты и гармонические колебания (звук)
  - Частоты и детали (изображение)
  - Постоянная составляющая
  - Действие фильтров
    - Фильтр размытия – НЧ-фильтр
    - Фильтр повышения четкости – ВЧ-фильтр
    - Фильтр нахождения границ – ВЧ-фильтр
    - Фильтры и обработка звука

Шумоподавление

- Простейшие методы
  - Размытие изображения – вместе с шумом размывает детали



  - Размытие в гладких областях – остается шум вблизи границ
  - Медианная фильтрация – хорошо подавляет импульсный шум, но удаляет мелкие детали

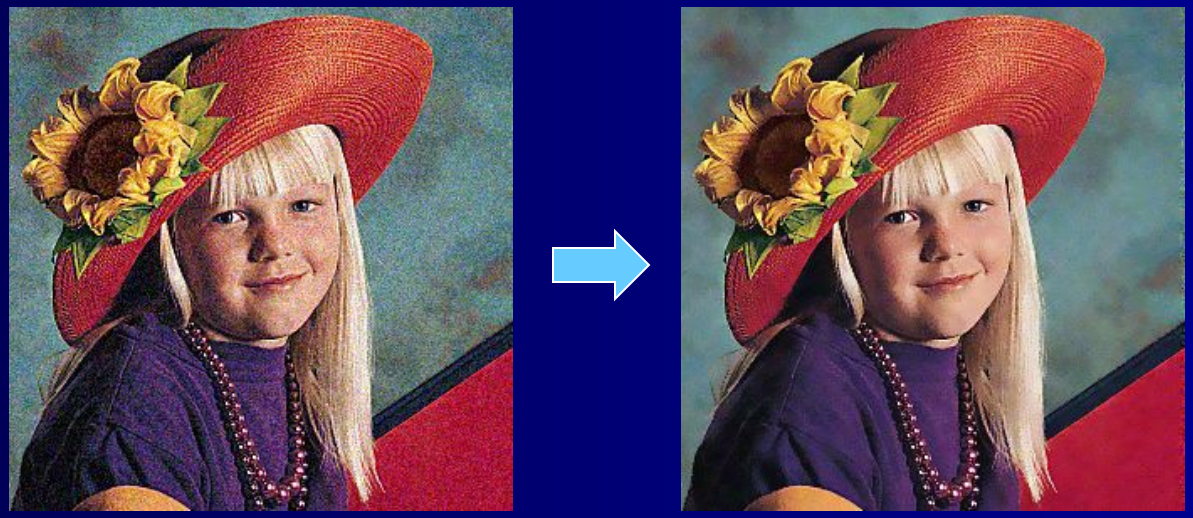
Шумоподавление

- Адаптивные алгоритмы
  - K nearest neighbors (K-NN) усреднение окружающих пикселей с весами

$$y_{i,j} = \sum_{k,m \in Q} x_{i+k,j+m} \cdot W(i,j,k,m)$$

$$W(i,j,k,m) \approx \exp\left(-\frac{(x_{i,j} - x_{i+k,j+m})^2}{h^2}\right) \cdot \exp\left(-\frac{k^2+m^2}{\rho^2}\right)$$

фотометрическая близость — пространственная близость



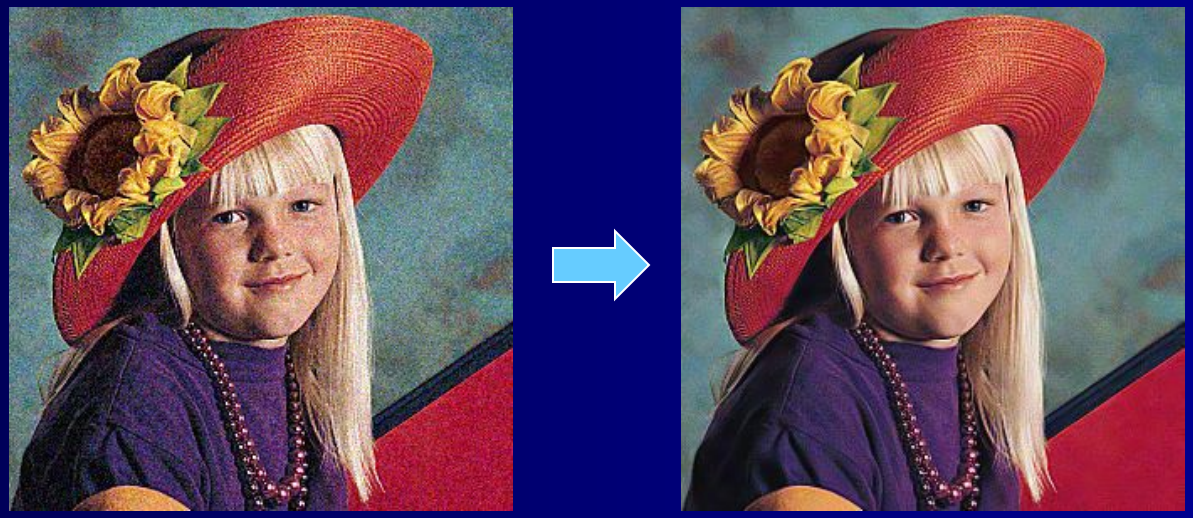
Шумоподавление

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Адаптивные алгоритмы
  - Non-local means (NL-means) – веса зависят от близости целых блоков, а не отдельных пикселей

$$W(i,j,k,m) \approx \exp\left(-\frac{\|v(x_{i,j}) - v(x_{i+k,j+m})\|^2}{h^2}\right)$$

$v(x_{i,j})$ – блок вокруг пикселя $x_{i,j}$



Метрики качества

- Как измерить похожесть двух изображений?

исходное изображение

искаженное изображение

Метрики качества

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Среднеквадратичная ошибка (MSE)

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2$$

N – число пикселей

- Пиковое отношение сигнал/шум (PSNR)

$$PSNR_{dB} = 10\lg\frac{M^2}{MSE}$$

M – максимальное значение пикселя

Метрики качества

- PSNR и MSE не учитывают особенности человеческого восприятия!

Оригинал

Далее будут использованы рисунки из статьи
Wang, Bovik, Lu “WHY IS IMAGE QUALITY ASSESMENT SO DIFFICULT?”

Метрики качества

- У этих изображений одинаковые PSNR с оригиналом (примерно 25 dB)

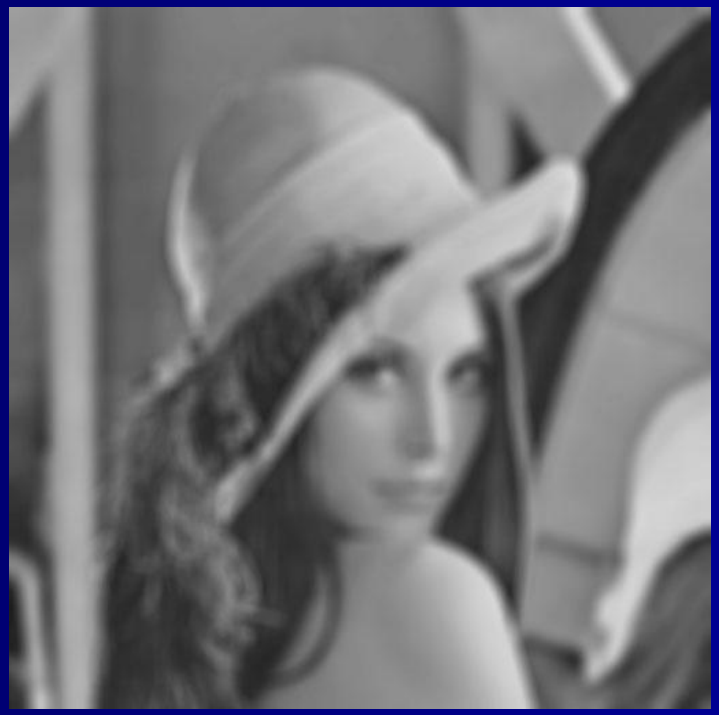
Повышена контрастность

Добавлен белый гауссов шум

Метрики качества

- И у этих – тоже примерно 25 dB!

Добавлен импульсный шум



Размытие

Метрики качества

- И у этого – тоже!

Артефакт блочности после JPEG

Метрики качества

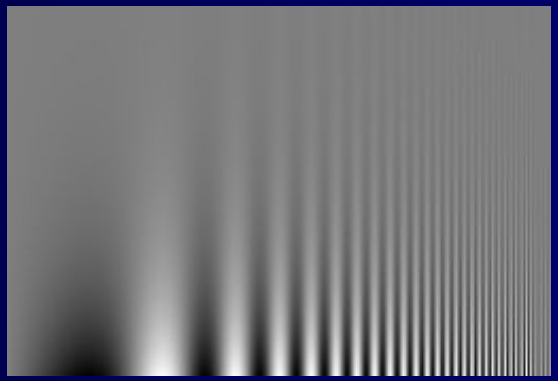
- Вывод: PSNR не всегда отражает реальный видимый уровень искажений.
- Как улучшить?

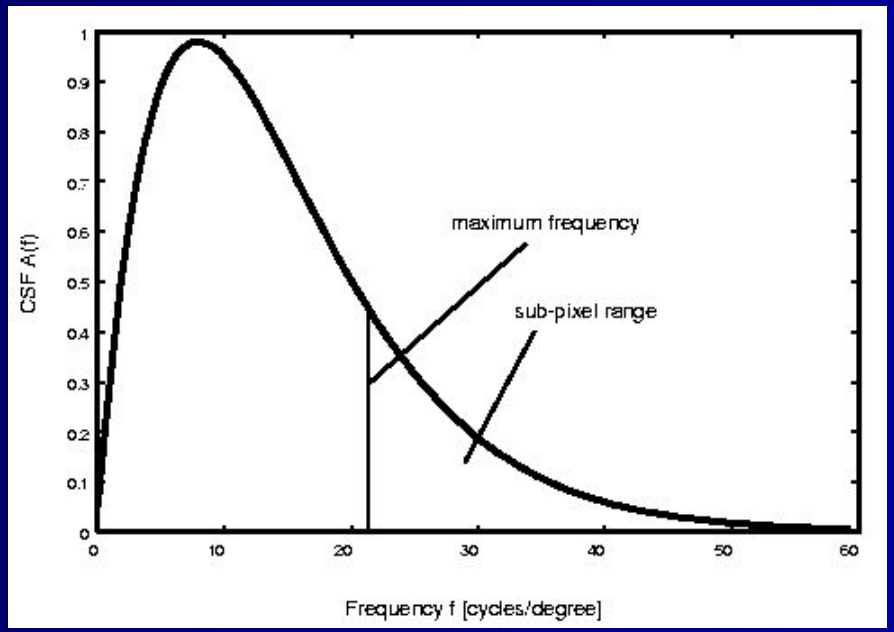
HVS models
(human visual system)

  - Использовать функцию чувствительности глаза к различным частотам (CSF)
  - Использовать свойство маскировки
  - Использовать равномерные к восприятию цветовые пространства (CIE Lab, CIEDE2000)

Метрики качества

- Contrast sensitivity function (CSF)
  - Показывает чувствительность глаза к различным частотам

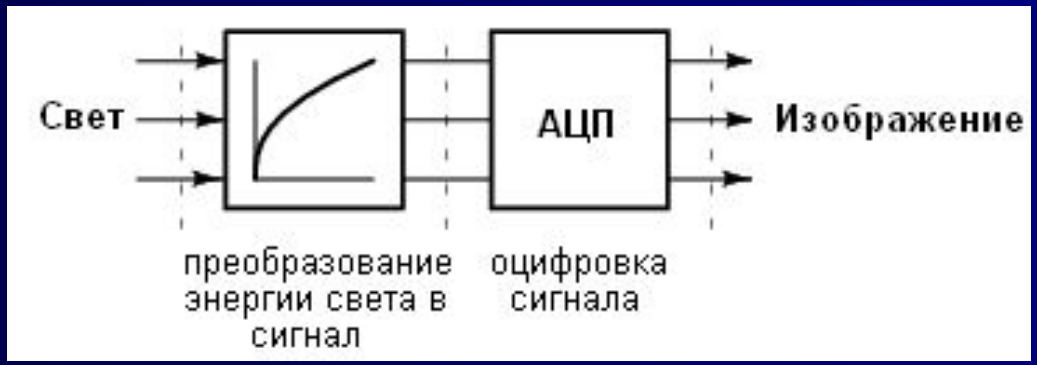




Абсцисса – пространственная частота
(колебаний / градус угла обзора)

Как получается цифровое изображение?

- Свет, падая на светочувствительный элемент преобразуется в электрические сигналы
- Сигналы оцифровываются, превращаются в массив чисел



$f(x)=y$

x – характеристика яркости света
y – яркость пиксела изображения

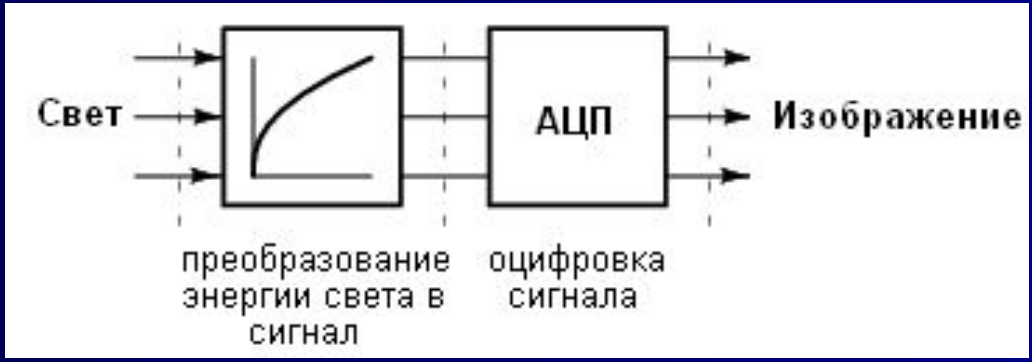
Почему оно может получиться плохо?

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Ограниченный диапазона чувствительности датчика
- “Плохой” функции передачи датчика

«Улучшение» изображения

- Изменение контраста изображения
  - Компенсация:
    - Ограниченного диапазона яркостей датчика
    - “Плохой” функции передачи датчика



Что такое гистограмма?

Гистограмма – это график распределения тонов на изображении. На горизонтальной оси - шкала яркостей тонов от белого до черного, на вертикальной оси - число пикселей заданной яркости.

Изменение контраста изображения

Что может не устраивать в полученном изображении:

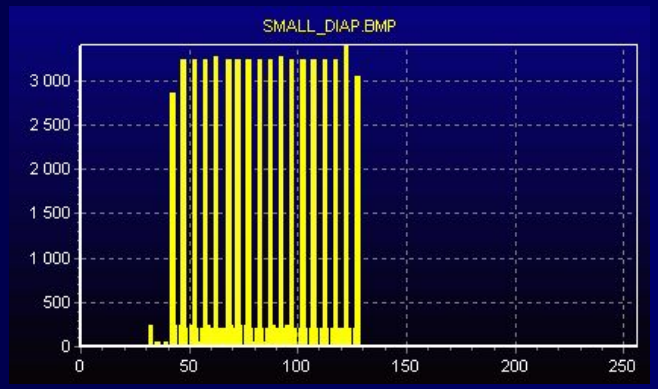
- Узкий или смещенный диапазон яркостей пикселей (тусклое или «пересвеченное» изображение)
- Концентрация яркостей вокруг определенных значений, неравномерное заполнение диапазона яркостей (узкий диапазон - тусклое изображение)
- Коррекция - к изображению применяется преобразование яркостей, компенсирующий нежелательный эффект:

$f^{-1}(x)=y$

y – яркость пиксела на исходном изображении,
x – яркость пиксела после коррекции.

Линейная коррекция

Компенсация узкого диапазона яркостей – линейное растяжение:



$$f^{-1}(y) = (y - y_{\min}) * \frac{(255 - 0)}{(y_{\max} - y_{\min})}$$

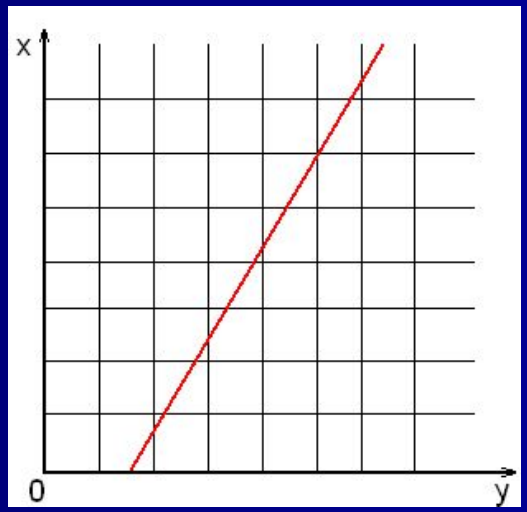
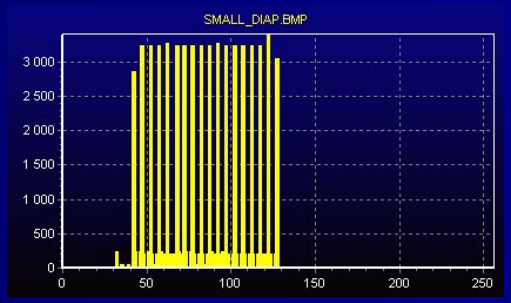


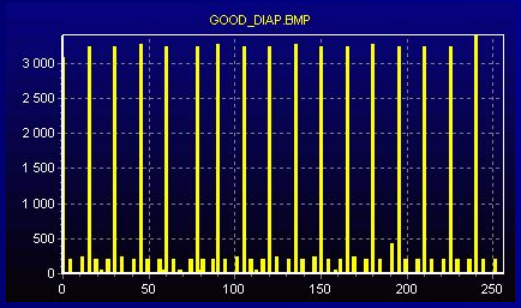
График функции $f^{-1}(y)$

Линейная коррекция

GRAPHICS & MEDIA LAB
VIDEO GROUP

Компенсация узкого диапазона яркостей – линейное растяжение:





Линейная коррекция

Линейное растяжение – «как AutoContrast в Photoshop»

Линейная коррекция

Линейная коррекция помогает не всегда!

Нелинейная коррекция

- Нелинейная компенсация недостаточной контрастности
- Часто применяемые функции:
- Гамма-коррекция
  - Изначальная цель – коррекция для правильного отображения на мониторе.
- Логарифмическая
  - Цель – сжатие динамического диапазона при визуализации данных

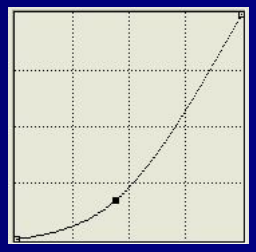
Гамма-коррекция

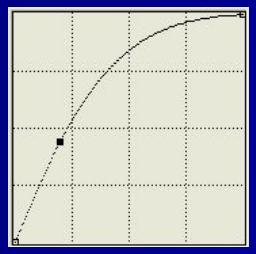
GRAPHICS & MEDIA LAB
VIDEO GROUP

Гамма-коррекция

– Изначальная цель – коррекция для правильного отображения на мониторе. Так называют преобразование вида:

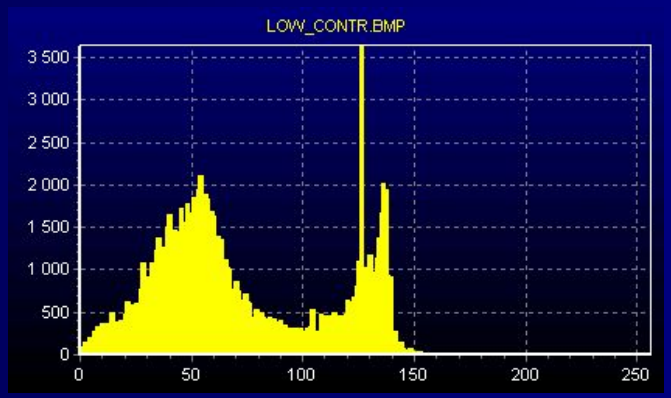
$$y = c \cdot x^{\gamma}$$

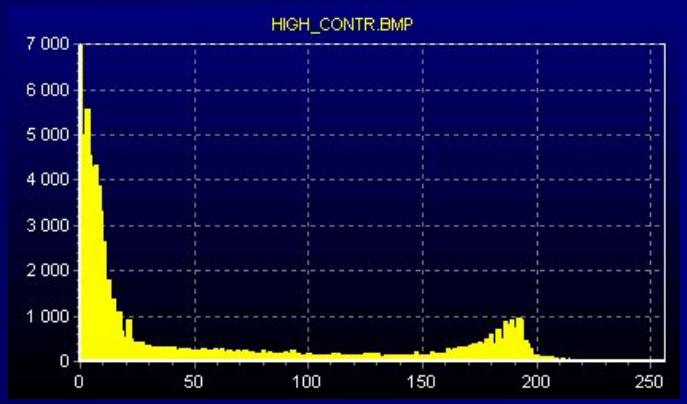




Графики функции $f^{-1}(y)$

Нелинейная коррекция





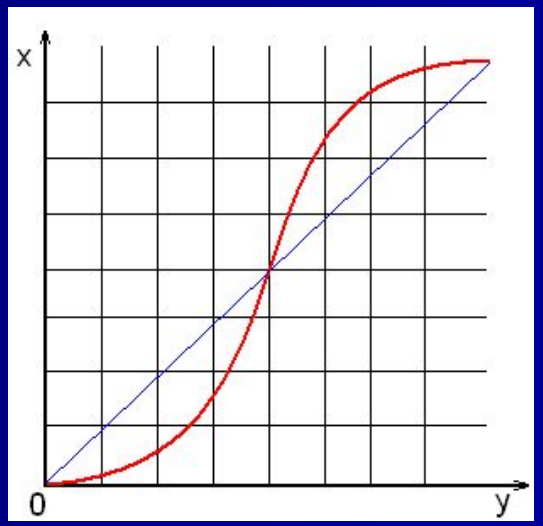


График функции $f^{-1}(y)$

Нелинейная коррекция

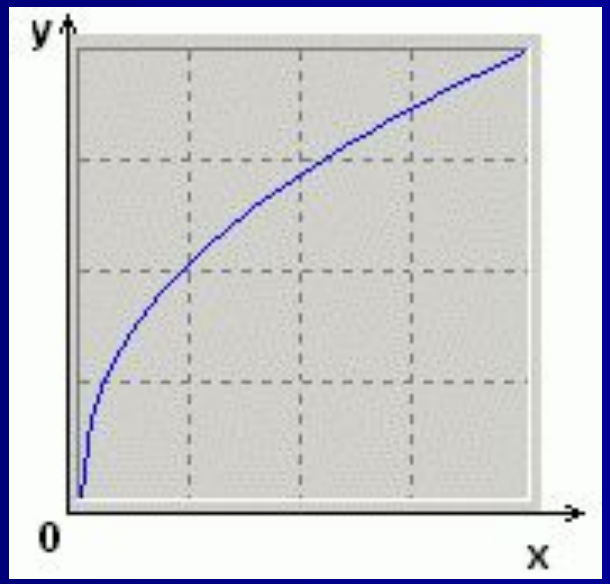


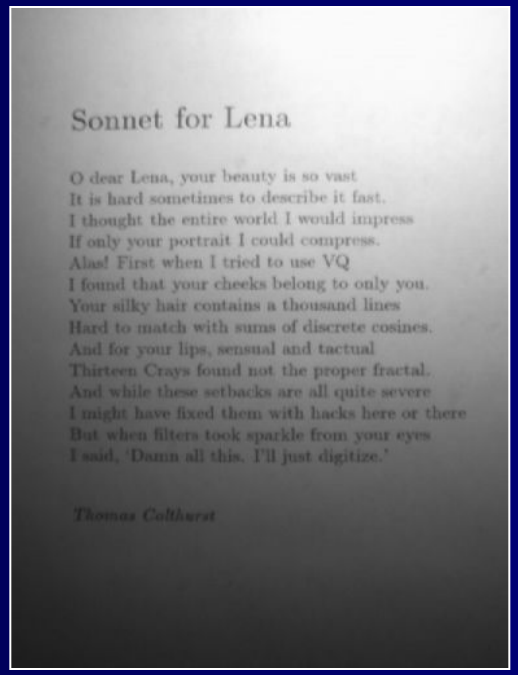
График функции $f^{-1}(y)$

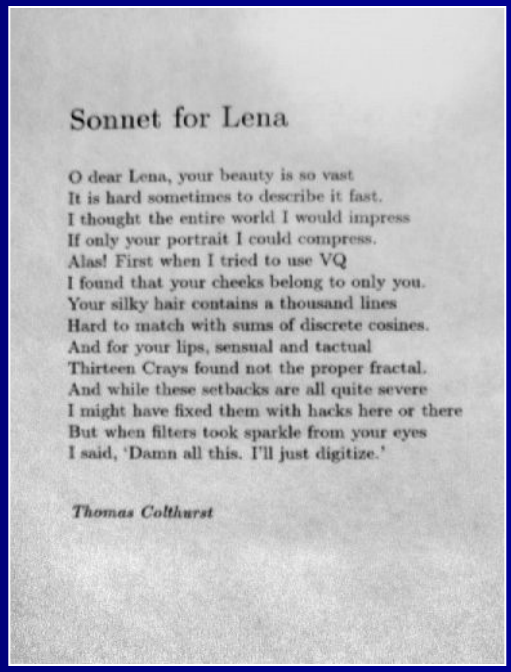
Сравнение линейной и нелинейной коррекции

GRAPHICS & MEDIA LAB
VIDEO GROUP

Компенсация разности освещения

Пример



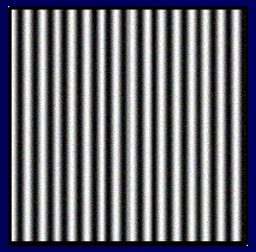


Компенсация разности освещения

Идея:

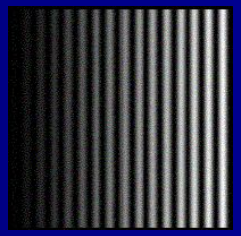
- Формирование изображения:

- Плавные изменения яркости относятся к освещению, резкие - к объектам.



объект $f(i,j)$

освещение $l(i,j)$



Изображение освещенного объекта $I(i,j)$

Выравнивание освещения

Алгоритм

- Получить приближенное изображение освещения путем низочастотной фильтрации

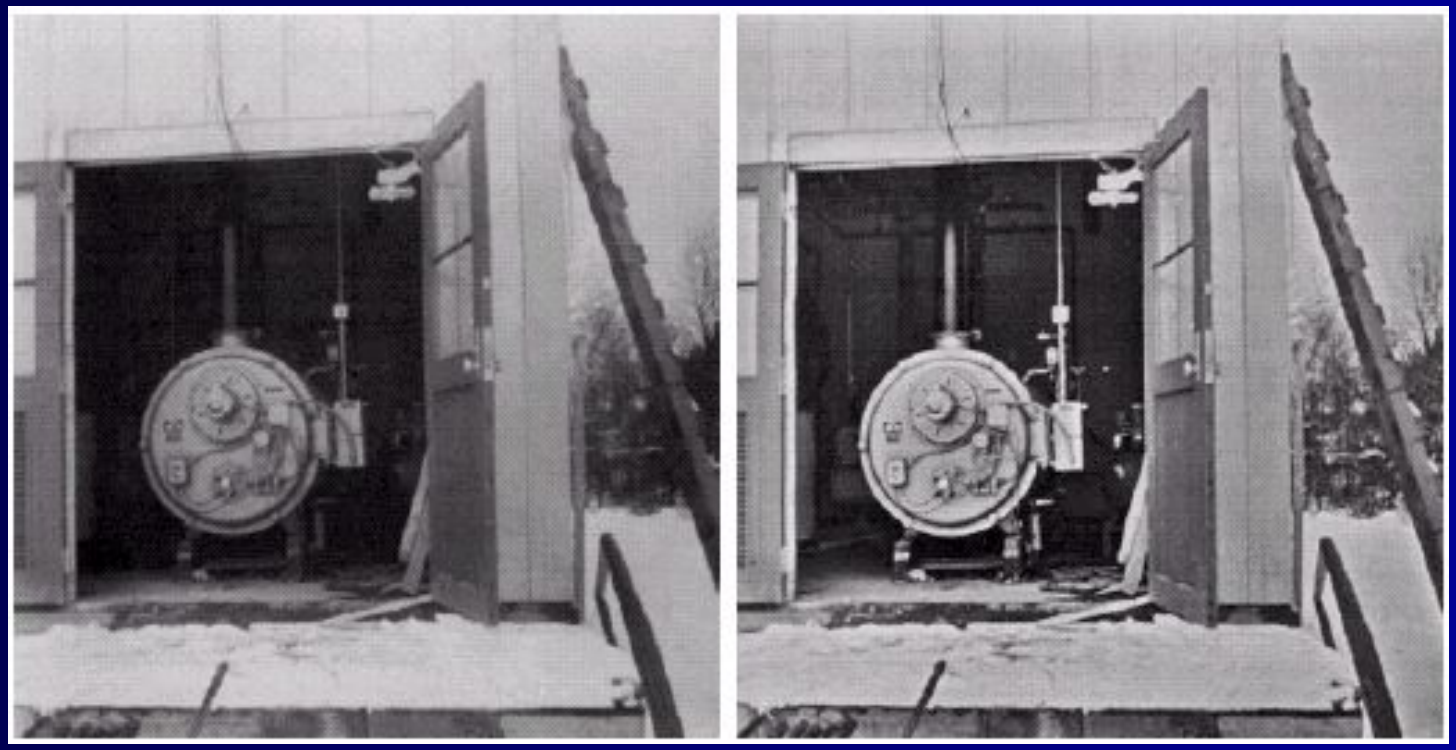
$$l'(i,j) = I(i,j) * G$$

- Восстановить изображение по формуле

$$f'(i,j) = \frac{I(i,j)}{l'(i,j)}$$

Выравнивание освещения

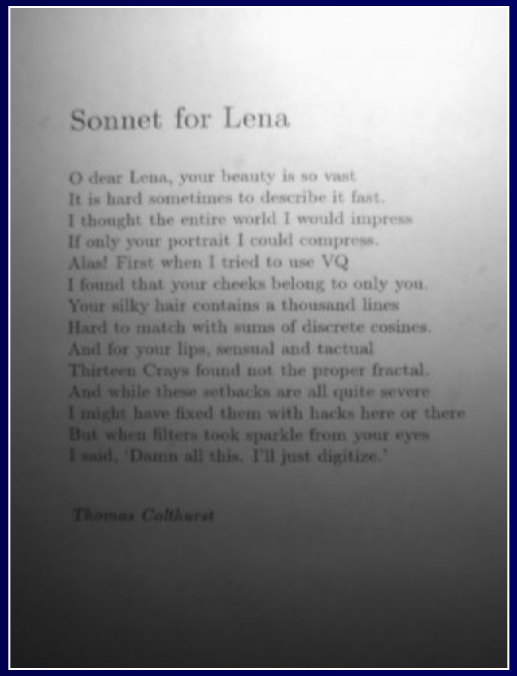
Пример



Компенсация разности освещения

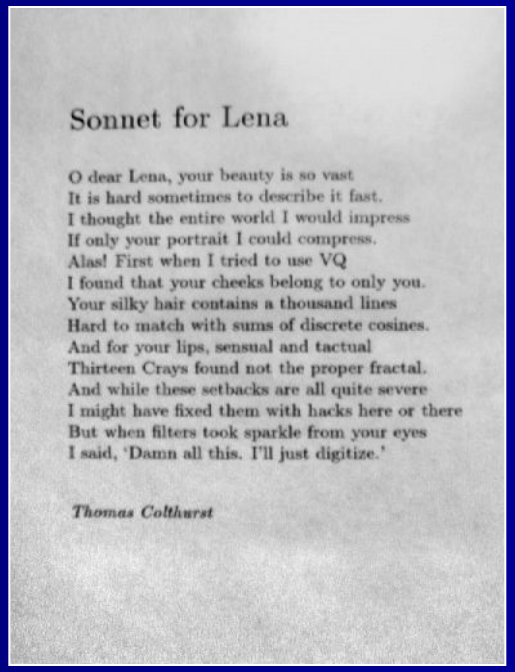
GRAPHICS & MEDIA LAB
VIDEO GROUP

Пример



/

=



Gauss 14.7 пикселей

Цветовая коррекция изображений

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Изменение цветового баланса
  - Компенсация:
    - Неверного цветовосприятия камеры
    - Цветного освещения

«Серый мир»

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Предположение:
  - Сумма всех цветов на изображении естественной сцены дает серый цвет;
- Метод:
  - Посчитать средние яркости по всем каналам:

$$\overline{R} = \frac{1}{N}\sum R(x,y); \quad \overline{G} = \frac{1}{N}\sum G(x,y); \quad \overline{B} = \frac{1}{N}\sum B(x,y); \quad Avg = \frac{\overline{R} + \overline{G} + \overline{B}}{3};$$

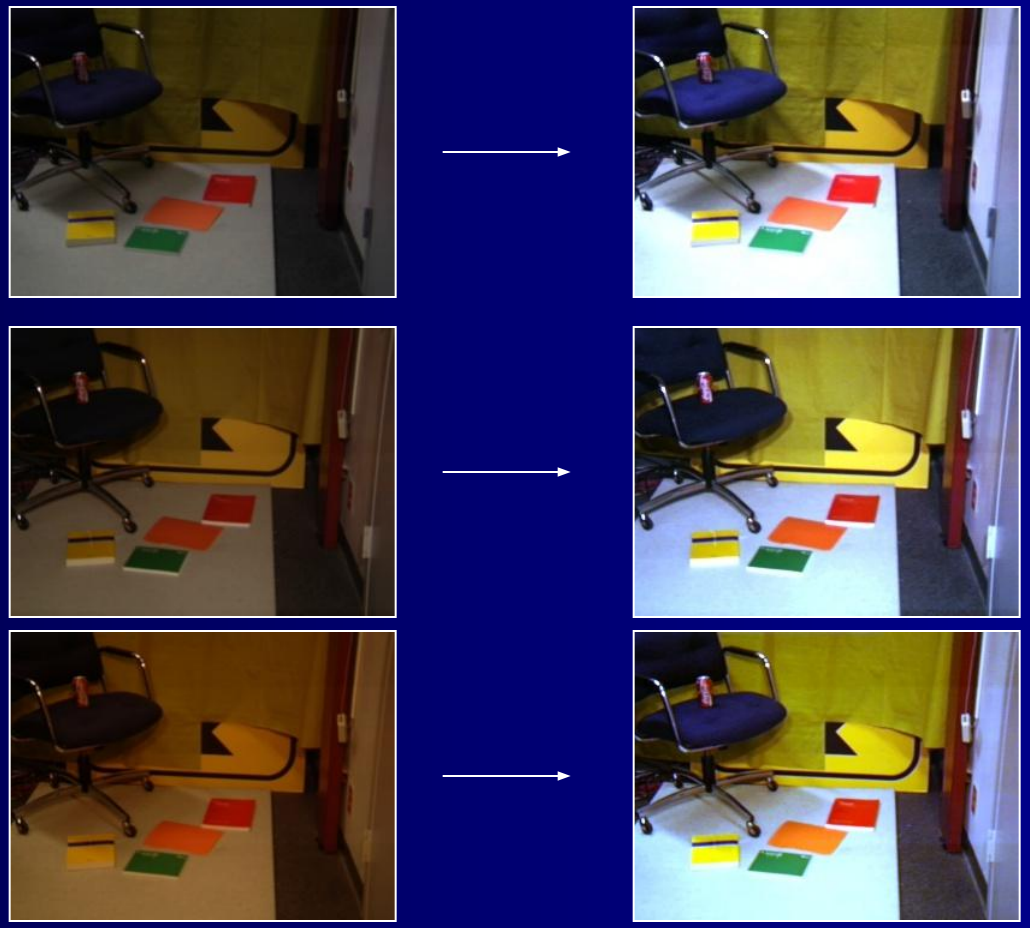
  - Масштабировать яркости пикселей по следующим коэффициентам:

$$R' = R \cdot \frac{Avg}{\overline{R}}; \quad G' = G \cdot \frac{Avg}{\overline{G}}; \quad B' = B \cdot \frac{Avg}{\overline{B}};$$

«Серый мир» - примеры

«Серый мир» - примеры

«Серый мир» - примеры



«Идеальный отражатель»

- Предположение:
  - Наиболее яркие области изображения относятся к бликам на поверхностях, модель отражения которых такова, что цвет блика = цвету освещения;
    (дихроматическая модель)
- Метод
  - Обнаружить максимумы по каждому из каналов:

    $R_{\max}, G_{\max}, B_{\max}$
  - Масштабировать яркости пикселов:

$$R * \frac{255}{R_{\max}}; \quad B * \frac{255}{B_{\max}}; \quad G * \frac{255}{G_{\max}};$$

Цветовая коррекция изображений

- Растяжение контрастности (“autolevels”)
  - Идея – растянуть интенсивности по каждому из каналов на весь диапазон;
- Метод:
  - Найти минимум, максимум по каждому из каналов:

$$R_{\min}, R_{\max}, G_{\min}, G_{\max}, B_{\min}, B_{\max}$$

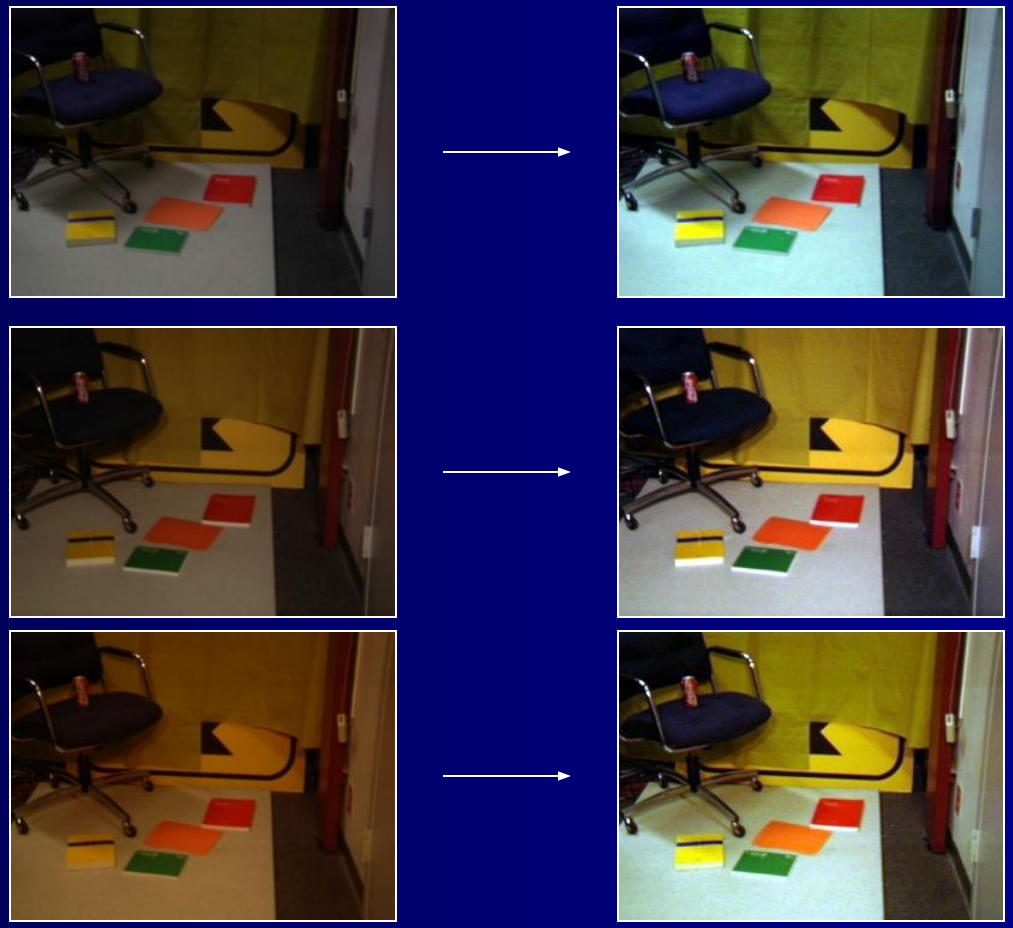
  - Преобразовать интенсивности:

$$(R - R_{\min}) * \frac{(255-0)}{(R_{\max} - R_{\min})}; \quad (G - G_{\min}) * \frac{(255-0)}{(G_{\max} - G_{\min})};$$

$$(B - B_{\min}) * \frac{(255-0)}{(B_{\max} - B_{\min})};$$

Растяжение контрастности всех каналов (“autolevels”)

Растяжение контрастности (“autolevels”)



Коррекция с опорным цветом

- Предположение
  - Пользователь указывает цвет вручную;
- Источник:
  - Априорные знания – «облака – белые»
  - Хорошая фотография этой же сцены
- Метод
  - Преобразовать по каждому из каналов цвета по формуле:

$$R * \frac{R_{dst}}{R_{src}}; \qquad G * \frac{G_{dst}}{G_{src}}; \qquad B * \frac{B_{dst}}{B_{src}};$$

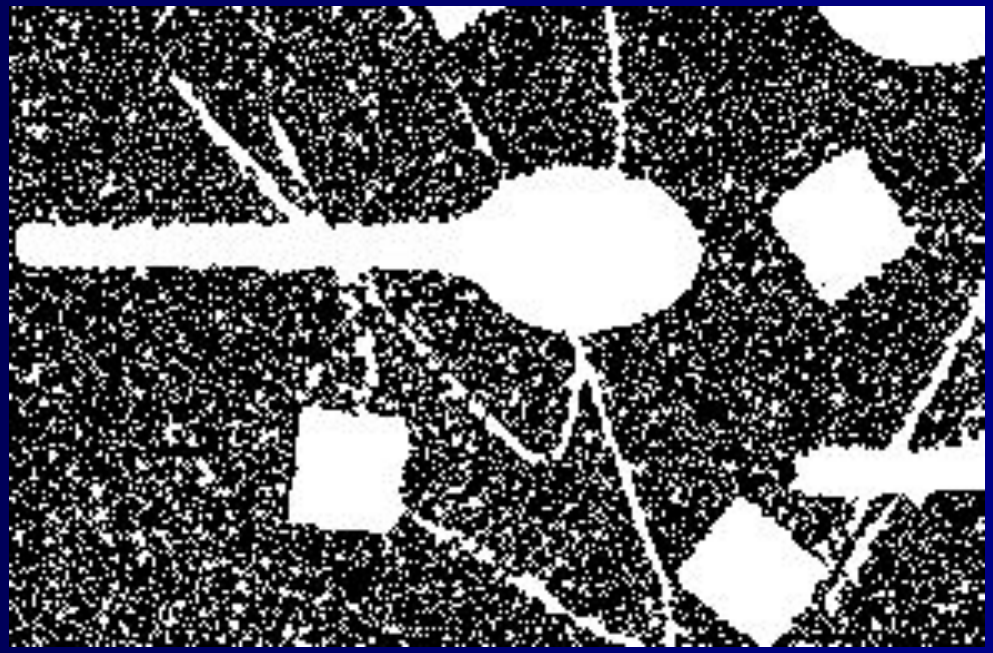
Коррекция с опорным цветом

GRAPHICS & MEDIA LAB
VIDEO GROUP

Примеры:

Шум в бинарных изображениях

Пример бинарного изображению с сильным шумом



Подавление и устранение шума

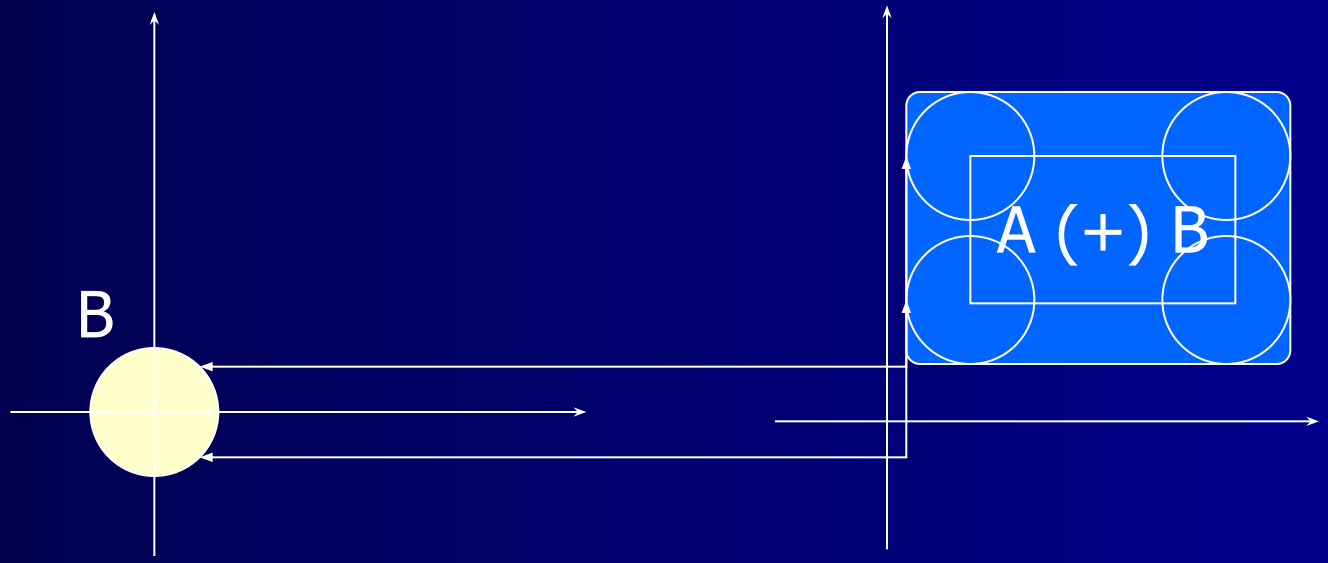
- Бинарное изображение – изображение, пиксели которого принимают всего два значения (0 и 1).

- Широко известный способ - устранение шума с помощью операций математической морфологии:
- Сужение (erosion)
- Расширение (dilation)
- Закрытие (closing)
- Раскрытие (opening)

Операции математической морфологии

Расширение

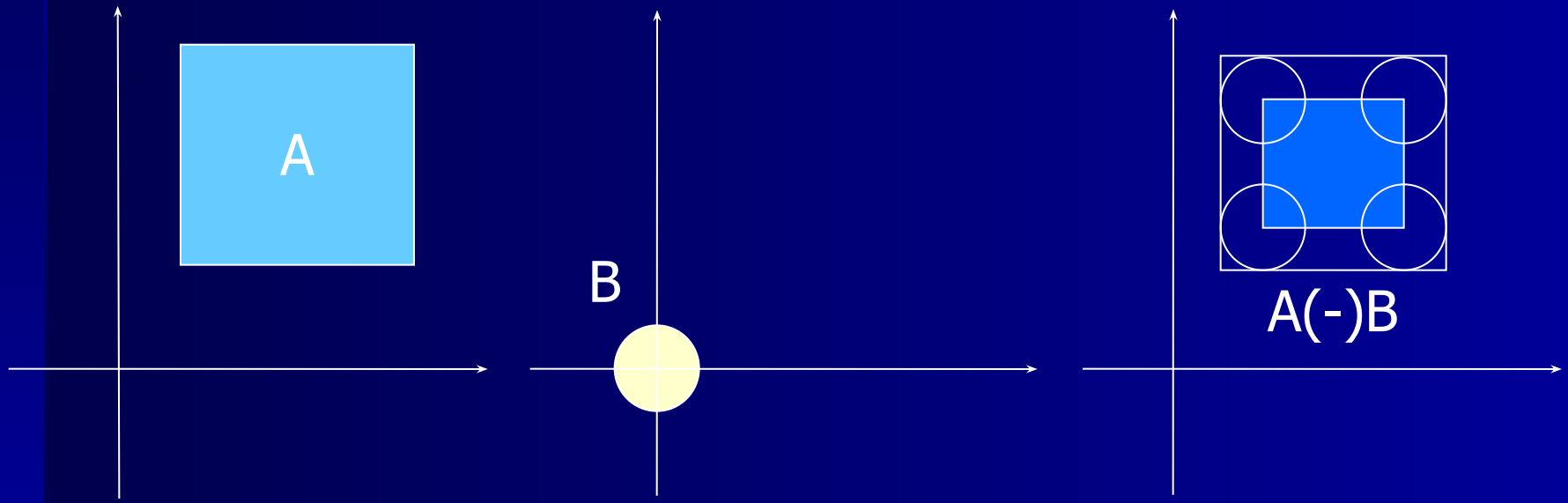
$$A (+) B = \{t \in R^2 : t = a + b, a \in A, b \in B\}$$



Операции математической морфологии

Сужение

A (-) B = (AC (+) B)C, где **AC –** дополнение **A**



Свойства морфологических операций

Коммутативный закон

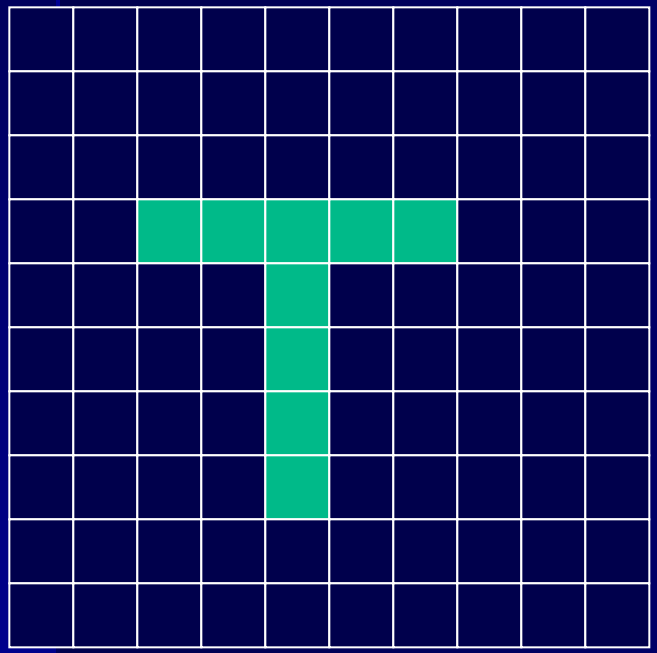
- A (+) B = B (+) A
- A (-) B < > B (-) A

Ассоциативный закон

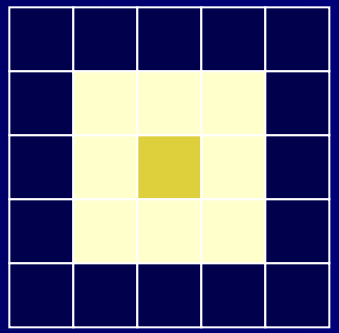
- A (+) (B (+) C) = (A (+) B) (+) C
- A (-) (B (-) C) = (A (-) B) (-) C

Дискретные операции морфологии

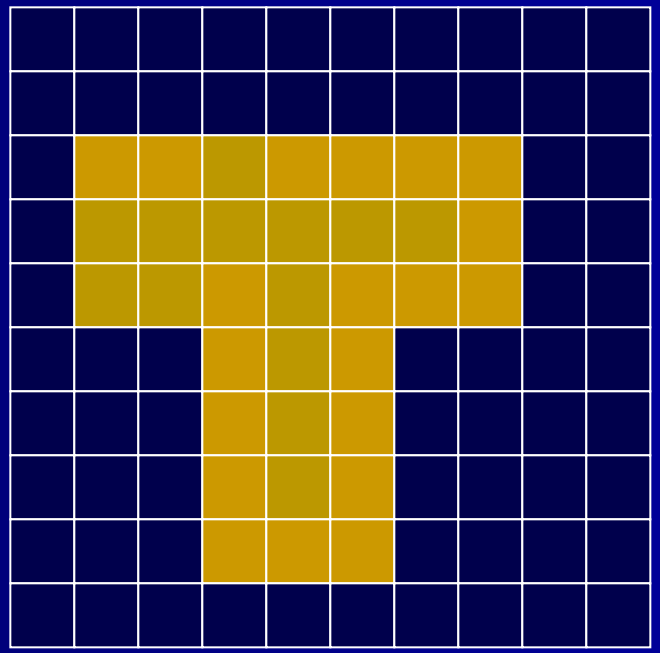
A



B



A(+)B



Операции раскрытия и закрытия

Морфологическое раскрытие (opening)

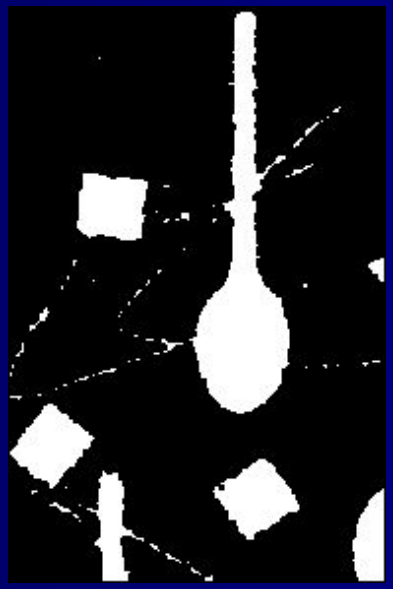
- **open(A, B) = (A (-) B) (+) B**

Морфологическое закрытие (closing)

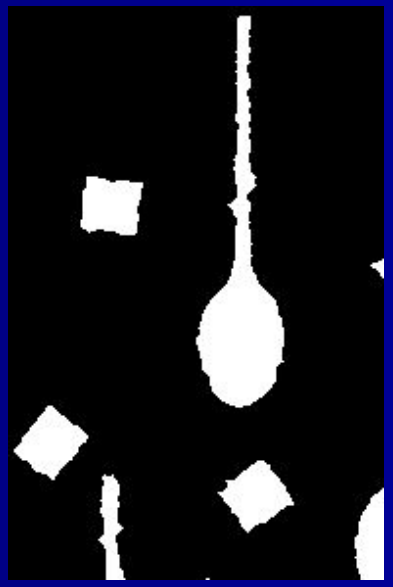
- **close(A, B) = (A (+) B) (-) B**

Применения сужения к бинарному изображению с сильным шумом

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



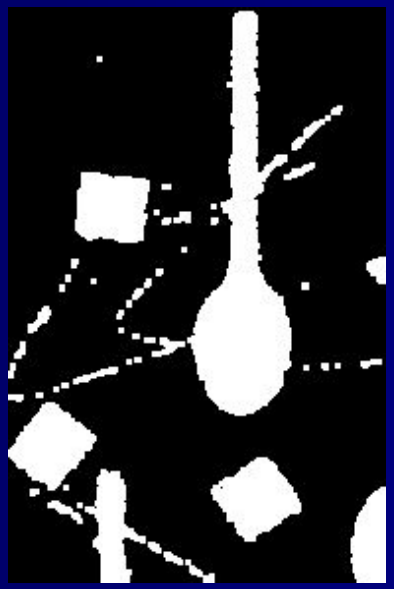
$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & [1] & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



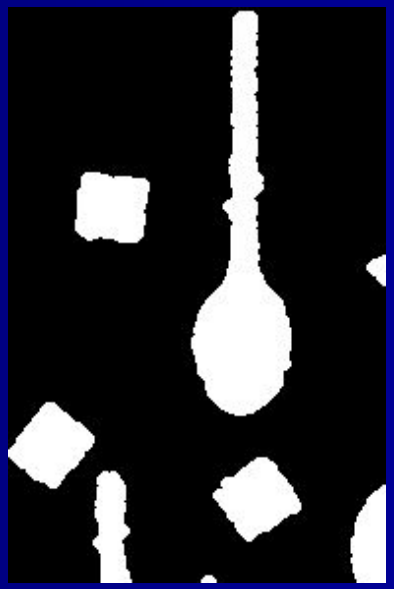
$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & [1] & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Применения открытия к бинарному изображению с сильным шумом

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



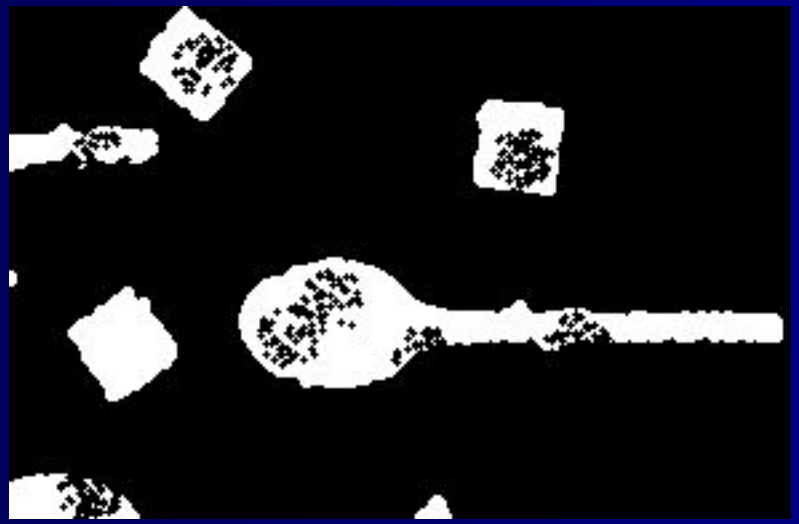
$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



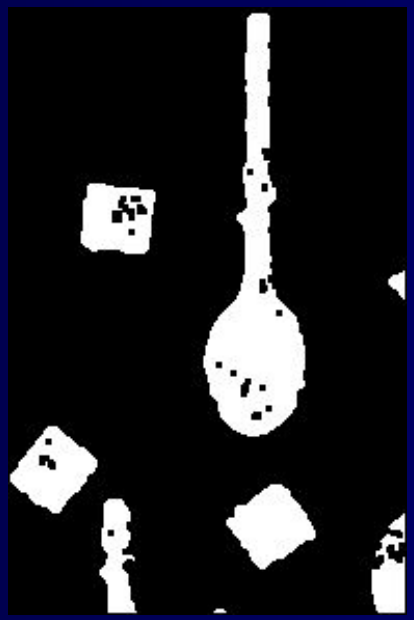
$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Устранение шума в бинарных изображениях

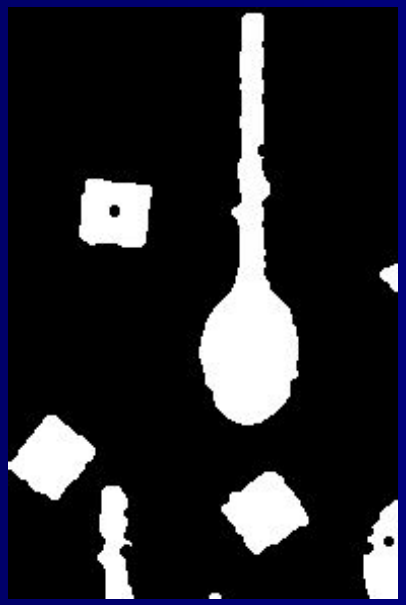
Пример бинарного изображению с дефектами распознаваемых объектов



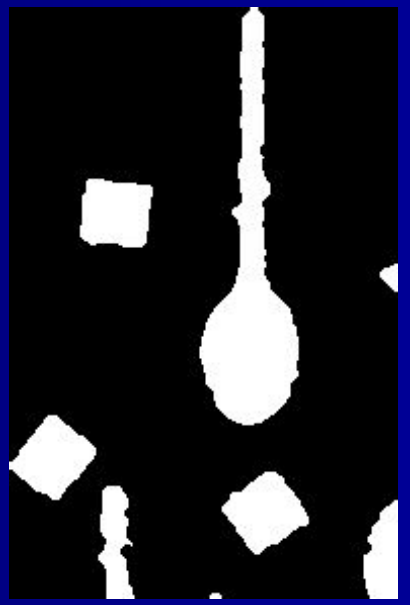
Применения закрытия к бинарному изображению с дефектами объектов



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



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

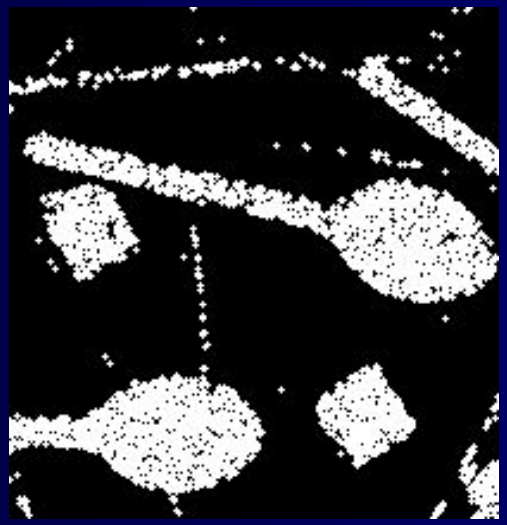


$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

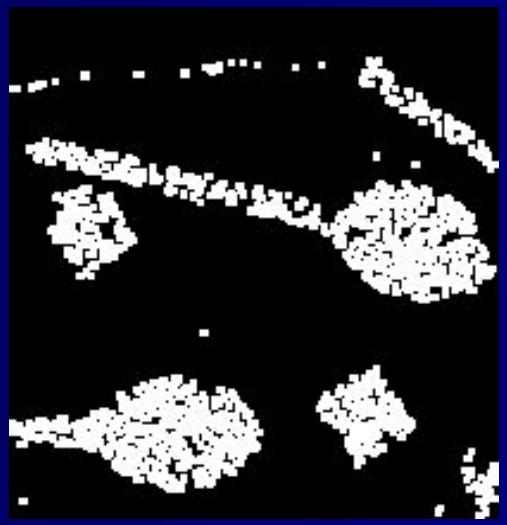
Не лучший пример для морфологии

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

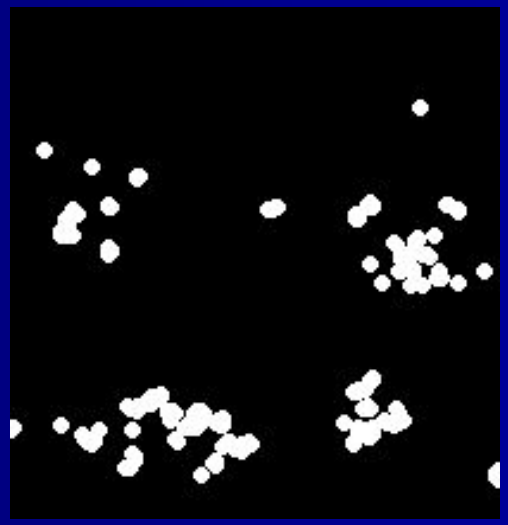
Применения операции открытия



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



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



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