

# Цифровая обработка сигналов

## Глава 1. Введение в курс цифровой обработки сигналов

- 1.1. Элементы теории сигналов
- 1.2. Классификация сигналов
- 1.3. Математические модели сигналов
- 1.4. Параллельные и векторные методы цифровой обработки сигналов
- 1.5. Вопрос о сравнении 2-х сигналов
- 1.6. Некоторые характерные сигналы
- 1.7. Интенсивность сигнала
- 1.8. Примеры базисных функций

## Глава 2. Цифровое представление сигналов

- 2.1. Дискретизация сигналов
- 2.2. Квантование сигнала
- 2.3. Практические вопросы дискретизации и квантования

## Глава 3. Преобразование сигналов

- 3.1. Преобразование Фурье
- 3.2. Преобразование Уолша
- 3.3. Линейные и нелинейные фильтры
- 3.4. Дискретные и спектральные преобразования
- 3.5. Цифровое представление преобразований
- 3.6. Свойства дискретных спектральных преобразований (на примере ДПФ)

## Глава 4. Быстрые алгоритмы ДПФ и ДПУ

- 4.1. Быстрые алгоритмы ДПФ и ДПУ
- 4.2. Двумерное БПФ
- 4.3. Совмещенные алгоритмы
- 4.4. Рекуррентный алгоритм вычисления БПФ

## Глава 5. Использование быстрых алгоритмов для цифровой фильтрации и интерполяции сигналов

- 5.1. Цифровая реализация линейного фильтра-цифровой фильтр
- 5.2. Вычисление цифровой свертки с использованием БПФ
- 5.3. Последовательная свертка
- 5.4. Секционированная свертка

## Глава 6. Примеры линейных одномерных и двумерных фильтров

- 6.1. Одномерные фильтры
- 6.2. Базисные фильтры
- 6.3. Нерекурсивные КИХ-фильтры
- 6.4. Фильтры низких частот

## Глава 7. Цифровое моделирование случайных сигналов

- 7.1. Реальный сигнал
- 7.2. Генерация случайных сигналов
- 7.3. Вычисление статистических характеристик случайных сигналов
- 7.4. Вычисление корреляционных функций и спектров
- 7.5. Основные принципы моделирования помех
- 7.6. Измерение параметров случайных помех

## Глава 8. Синтез линейных фильтров

- 8.1. Оптимальные линейные адаптивные фильтры
- 8.2. Адаптивные линейные фильтры для подавления аддитивного независимого и
- 8.3. Коррекция линейных искажений, вносимых линейными блоками систем пре-

- 8.4. Линейные фильтры для препарирования изображений (выделения деталей)
- 8.5. Итеративный алгоритм фильтрации импульсных помех
- 8.6. Рекурсивный алгоритм фильтрации импульсных помех
- 8.7. Обнаружение и фильтрация импульсных помех и сбоев
- 8.8. Синтез линейных фильтров методом регуляризации Тихонова

## Глава 9. Нелинейные алгоритмы обработки сигналов

- 9.1. Нелинейные алгоритмы обработки сигналов
- 9.2. Алгоритмы сглаживания сигналов
- 9.3. Увеличение детальности изображения
- 9.4. Обнаружение деталей и границ
- 9.5. Стандартизация изображений

## Глава 10. Параллельные методы цифровой обработки сигналов

- 10.1. Введение в параллельные методы цифровой обработки сигналов
- 10.2. Параллельные методы вычисления свертки (аппаратно-программные)
- 10.3. Параллельное вычисление по группам разрядов (разрядным срезам) изображений
- 10.4. Локальная эквализация
- 10.5. Параллельное вычисление по группам элементов входного изображения
- 10.6. Медианные фильтры
- 10.7. Медианные фильтры на основе сортирующих сетей

## Глава 11. Модели изображений

- 11.1. Модели изображений
- 11.2. Основные цветовые модели
- 11.3. Форматы файлов изображения

## Глава 12. Алгоритмы сжатия изображений

- 12.1. Алгоритмы сжатия изображений
- 12.2. Алгоритм JPEG
- 12.3. Алгоритм Хаффмана
- 12.4. Вейвлеты
- 12.5. Алгоритм JPEG2000 (2000 г.)

## Глава 13. Распознавание образов

- 13.1. Распознавание образов
- 13.2. Методы распознавания
- 13.3. Вероятностный подход
- 13.4. Метод минимального расстояния
- 13.5. Метод разделяющих функций
- 13.6. Линейные разделяющие функции
- 13.7. Метод потенциалов
- 13.8. Выделение признаков
- 13.9. Нейронные сети
- 13.10. Многослойные нейронные сети

# Элементы теории сигналов

В настоящее время методы цифровой обработки сигналов, digital signal processing (DSP) находят все более широкое применение, вытесняя постепенно методы, основанные на аналоговой обработке. В данном курсе рассматриваются основы теории и наиболее употребляемые алгоритмы обработки. [1,2]

Основная цель-создание представления об основных приемах и методах, аппаратных и программных средствах цифровой обработки сигналов.

Под **сигналом** понимается физический процесс, отображающий сообщение и служащий для его передачи по каналу связи. Или сигнал- это форма, в которой воплощена передаваемая, хранимая или перерабатываемая информация.

В общем случае сигналы рассматриваются как функции, заданные в физических координатах, т.е.  $U(x,y,z)$

# Классификация сигналов

При описании сигналов в первую очередь выделяются две наиболее общие характеристики:

- множество значений, которые могут принимать сами сигналы;
- множество значений, которые могут принимать их математические представления (формулы).

Исходя из этого, можно представить следующую классификацию сигналов:

(Для классификации могут быть использованы разные критерии, поэтому можно предположить и разные классы сигналов):

- Пространственный и временной:

*Пространственный сигнал* - зависит только от пространственных координат. Наиболее часто используемый, изображение - двумерный сигнал.

*Временной сигнал* - зависит только от времени. Электрический сигнал в ЭВМ.

- Финитный и инфинитный:

*Финитный сигнал* - множество значений координат ограничено.

*Инфинитный сигнал* - множество значений координат неограниченно.

- Аналоговый, непрерывный, дискретный, квантованный, цифровой:

*Аналоговый сигнал* - сигнал и его аргументы принимают непрерывные значения на числовых осях координат и значений.

*Дискретный сигнал* - аргументы сигнала принимают лишь счетное множество значений.

*Квантованный сигнал* - сигнал, который принимает счетное множество значений.

*Цифровой сигнал* - дискретный, квантованный сигнал.

С точки зрения математического описания различают также детерминированные (регулярные) и случайные (недетерминированные) сигналы.

*Детерминированный сигнал*, это сигнал, который может быть задан точными функциональными зависимостями от пространственных координат и времени.

*Случайный сигнал* - сигнал, который нельзя описать точными математическими соотношениями. Эти сигналы рассматриваются как выборочные значения или реализации из некоторого ансамбля сигналов, и математическое описание не для каждого отдельного сигнала (который есть случайная величина или функция), а для ансамбля(процесса) в целом. Эти сигналы описываются не точными выражениями, а при помощи осредненных характеристик.

# Математические модели сигналов

(Основы математического описания сигналов)

При математическом описании сигнал рассматривается как точка в некотором пространстве сигналов, которое представляет **линейное метрическое пространство**. В этой связи описать представление сигнала и его основных характеристик можно, используя свойства метрического линейного пространства.

Амплитуда сигнала- $U$ - комплексная

$$U = U_1 + iU_2 \text{ или } U = U_0 e^{i\varphi},$$

$U_0$ - собственно амплитуда

$\varphi$ - фаза сигнала

# **Параллельные и векторные методы цифровой обработки сигналов**

Векторные компьютеры, концепция конвейеризации. Параллельные компьютеры, различные схемы соединения процессоров. Основные принципы векторизации алгоритмов цифровой обработки **сигналов** для векторных ЭВМ. Реализация алгоритмов цифровой обработки на параллельных компьютерах: параллельные вычисления по группам элементов изображений, по группам разрядов изображений, по группам элементов свертки. Систолические матрицы.

# Вопрос о сравнении 2-х сигналов

Сравнение сигналов осуществляется путем сравнения их математических моделей (комплексных амплитуд или интенсивностей) по различным критериям. Здесь приведены сравнения сигналов по их амплитудам.

- Среднеквадратичное отклонение

$$d = \sqrt{\frac{1}{\Omega} \int_{\Omega} |U(x) - U'(x)|^2 dx}$$

Критерий-плохой:

а)если яркость понизится всего на 5%, то качество сигнала (по этому критерию) будет сильно испорчено. Однако, глаз это не воспринимает. б) Если наблюдается изображение со "снегом" - резкое мерцание белых точек, то данный критерий почти не изменится. На глаза этот сигнал имеет сильное отрицательное воздействие.

- Максимальное отклонение

$$d = \max_{\Omega} |U(x) - U'(x)|$$

Эта мера, наоборот, чувствительна к отдельным всплескам сигналов.

- Мера *отношение сигнал/шум (PSNR)*

$$d = 10 \lg \frac{255^2 n^2}{\sum_{i,j=1}^n |U_{ij} - U'_{ij}|^2}$$

Этой мере присущи те же недостатки, что и среднеквадратичному отклонению.

# Некоторые характерные сигналы

## 1. Точечный сигнал

$$U(x,y) = A\delta_{x-x_0, y-y_0}$$

2. Сигнал с амплитудой A, локализованный в точке  $(x_0, y_0)$

$$\varphi(t) = \text{rect}\left[\frac{(t-T)}{T}\right]A,$$

где  $\text{rect}(x) = \begin{cases} 1, & 0 \leq x \leq 1 \\ 0, & \text{в противном случае} \end{cases}$

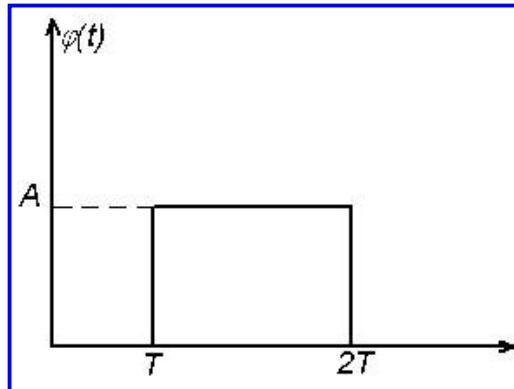


Рис. 1.

## 3. Гауссов сигнал

$$U(x,y) = A \exp\left(-\frac{x^2+y^2}{\sigma^2}\right)$$

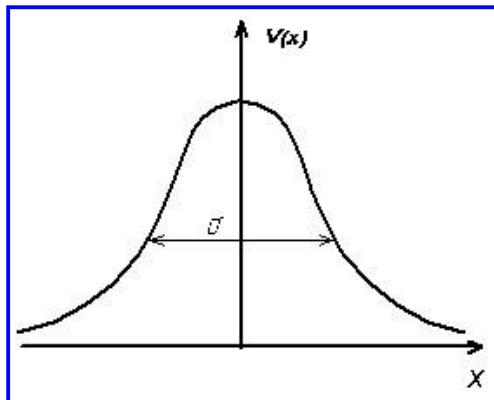


Рис. 2.

# Интенсивность сигнала

$$I=U \cdot U^* = |U|^2 - \text{интенсивность} \text{ сигнала.}$$

Если определять отличие одного сигнала от другого по их интенсивности, используются следующие критерии:

- Мера отличия одного сигнала от другого:

$$d=\int_0^T (U_1(t)-U_2(t))(U_1(t)-U_2(t))^* dt$$

T-область определения сигнала.

Физический смысл энергии - определение разности двух сигналов.

- Принцип суперпозиции: результат действия двух сигналов равен их геометрической сумме:

$$U_3=U_1+U_2$$

- Любой сигнал можно представить в виде разложения по базисным функциям:

$$U=\sum_{k=0}^N U_k \phi_k \quad (1)$$

где  $\phi_k$ -базисные функции;  $(\phi_k, \phi_e)=\delta_{k,e}$

$U_k$ -коэффициенты

$$U_k=(U, \phi_k)$$

Пример для временного сигнала:

$$U_k=\frac{1}{T} \int_0^T U(t) \phi_k(t) dt$$

# Примеры базисных функций

## 1. Импульсные

$$\varphi_k = \text{rect}(t - kT)/T \quad (1)$$

$$\text{rect}(t) = \begin{cases} 1 & 0 \leq t \leq 1 \\ 0 & \text{в противном случае} \end{cases}$$

$$U_k = \frac{1}{T} \int_{kT}^{(k+1)T} U(t) dt$$

## 2. Гармонические (экспоненциальные)

$$\varphi_k(t) = \sin\left(\frac{\pi kt}{T}\right)\pi, \quad (2)$$

$$\varphi_k(t) = \exp\left(\frac{i\pi kt}{T}\right),$$

$$U_k = \frac{1}{2T} \int_0^T U(t) \exp\left(-\frac{i\pi kt}{T}\right) dt$$

## 3. Функции отсчетов

$$\varphi_k(t) = \text{sinc}2\pi T \frac{t-k}{2T} \quad (3)$$

$$\text{sinc}x = \frac{\sin(x)}{x}$$

$$U_k = \frac{1}{2T} \int_{-\infty}^{+\infty} U(t) \text{sinc}\left[\frac{t-k}{2T} 2\pi T\right] dt = U \frac{k}{2T}$$

## 4. Функция Уолша

$$Wal_k(\xi) = \frac{1}{\sqrt{T}} (-1)^{\sum_{m=0}^{\infty} k_m \xi_{m+1}}$$

где  $\xi = \frac{t}{T}$ ;  $\xi = \sum_{m=0}^{\infty} \xi_{m+1} 2^{-(m+1)}$  - двоичное представление числа  $\xi$

$K_m^r = k_m \oplus k_{m+1}$  m-й разряд кода Грея номера k.

$k_m$  - m-й номер двоичного разряда числа k (справа налево)

$$k = \sum_{m=0}^{\infty} k_m 2^m, \oplus - \text{сложение по модулю 2.}$$

Пример функций Уолша:

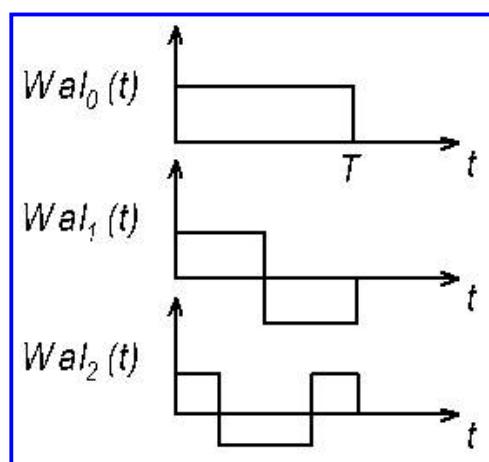


Рис. 1.

# Дискретизация сигналов

Необходимость дискретизации и квантования — ЭВМ работает только с дискретным набором чисел.

**Дискретизация сигнала** — замена непрерывного сигнала последовательностью чисел, являющихся представлением этого сигнала по какому-либо базису.

**Постановка задачи.**

Пусть имеется непрерывный сигнал  $x(t)$ , заданный на интервале  $[0, +\infty)$ . При переходе к оцифровке происходит следующая операция. Выбирается шаг дискретизации  $T$ , и вместо исходного сигнала получается последовательность  $y[n]=x(nT)$ ,  $n=0, 1, \dots$ . Далее, выбирается формат оцифровки  $r$ . Обычно он бывает кратным 8, хотя это необязательно. Предположим что существует такое число  $M$ , что выполнены неравенства:  $-M \leq y[n] \leq M$  для всех  $n$ . Интервал  $[-M, M]$  разбивается на  $2^r$  частей. После этого каждое значение  $y[n]$  заменяется новой последовательностью  $z[n]$ , но теперь каждый новый член последовательности принимает значение из интервала  $[0, 2^r - 1]$ . При желании вместо указанного представления можно перейти к представлению сигнала целыми числами со знаком.

На каждом из упомянутых шагов происходит углубления сигнала. первая задача цифровой обработки заключается в оценке искажения исходного сигнала. дальнейшая обработка состоит в извлечении из полученного сигнала нужной информации и подавлении шумов. Это осуществляется с помощью цифровой фильтрации. Даже оцифрованный сигнал занимает много места, и следующий шаг обработки заключается в сжатии сигнала. Обычно имеется ввиду сжатие с потерей информации. Здесь важно установить критерий допустимой потери информации в зависимости от выбранного критерия выбирается способ сжатия. Хотя последовательность бесконечна, в реальных условиях мы имеем дело лишь с конечными последовательностями. В этой связи нужна оценка потерь, связанных с усечением последовательностей.

**Теорема отсчетов (теорема Котельникова):**

Самый распространенный способ дискретизации.

Сигналы, спектр Фурье которых равен нулю за пределами интервала  $(-F, F)$ , могут восстановлены по своим отсчетам, взятым с шагом  $\Delta t = \frac{1}{2F}$

$$U(t) = \int_{-\infty}^{+\infty} U(k\Delta t) \operatorname{sinc}\left(2\pi F\left(t - \frac{k}{2F}\right)\right), \quad (1)$$

где

$$U(k\Delta t) = 2F \int_{-\infty}^{+\infty} U(t) \operatorname{sinc}\left(2\pi F\left(t - \frac{k}{2F}\right)\right) dt \quad (2)$$

$$\text{или } U(k\Delta t) = \sum_{n=-\infty}^{+\infty} U(t_n) S\left(t - \frac{k}{2F}\right) \text{ — отсчеты в точках } t = \frac{k}{2F}$$

Определение: спектр Фурье:

$$U_\Phi = \int_{-\infty}^{+\infty} U(t) \exp[2\pi ft] dt \quad (3)$$

$$U(t) = \int_{-\infty}^{+\infty} U_\Phi(f) \exp(-2\pi ft) df$$

Доказательство:

Так как спектр  $U_\Phi(f)$  ограничен интервалом  $(-F, F)$  то его можно разложить в ряд Фурье

$$U(t) = \int_{-F}^F u_\Phi(f) \exp(-2\pi ft) df = \int_{-F}^F \sum_{l=-\infty}^{+\infty} x(l) \exp(+i\pi fl\Delta t) \exp(-2\pi ft) df = \\ \sum_{l=-\infty}^{+\infty} x(l) \int_{-F}^F \exp[2\pi if(t - \frac{l}{2F})] df = \sum_{l=-\infty}^{+\infty} x(l) \frac{2\sin[2\pi F(t - \frac{l}{2F})]}{2\pi(t - \frac{l}{2F})}$$

По определению коэффициентов Фурье

$$x(l) = \frac{1}{2} F^{-1} \int_{-F}^F U_\Phi(f) \exp(-\frac{2\pi ifl}{2F}) df = \frac{1}{2} F^{-1} (\frac{1}{2F}) = \frac{1}{2} FU(l\Delta t)$$

Таким образом, если взять отсчеты функции  $U(t)$  в точках  $t = \frac{k}{2F}$ , удаленных друг от друга на величину  $\Delta t = \frac{1}{2F}$  — частота Найквиста, то функцию  $U(t)$  можно абсолютно точно восстановить по этим отсчетам.

Двумерный случай:

$$U(x,y) = \sum_{k_1=-\infty}^{+\infty} \sum_{k_2=-\infty}^{+\infty} U\left(\frac{k_1}{2F_1}, \frac{k_2}{2F_2}\right) \operatorname{sinc}2\pi F_1(t_1 - \frac{k_1}{2F_1}) \operatorname{sinc}2\pi F_2(t_2 - \frac{k_2}{2F_2}) \quad (4)$$

где  $U\left(\frac{k_1}{2F_1}, \frac{k_2}{2F_2}\right)$  — отсчеты на двумерном прямоугольном реестре с шагом  $\frac{1}{2F_1}$  по оси  $x$  и  $\frac{1}{2F_2}$  по оси  $y$ .

### Дискретизация функций с реальным спектром

Реальные функции имеют спектр, как правило вида рис. 1

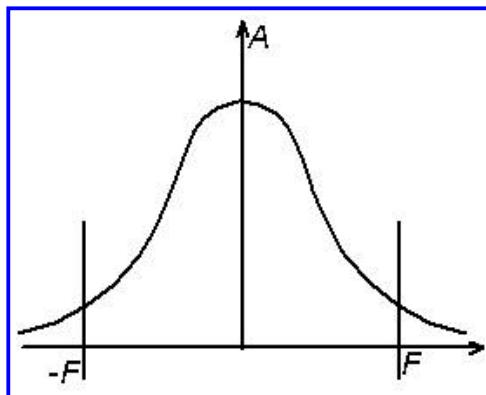


Рис. 1.

Учет так называемых "хвостов" резко увеличивает вычислительные затраты. Нельзя точно восстановить сигнал по теореме Котельникова.

Для того, чтобы не возникало искажений, нужно пропустить изображение через фильтр низких частот — "окно", который устраняет высокочастотные составляющие и ограничивает спектр.

Примеры фильтров "окон".

1. Прямоугольное окно

$$\operatorname{rect}\left(\frac{f+F}{2F}\right) = \begin{cases} 1, & |f| \leq F \\ 0, & |f| \geq F \end{cases} \quad (5)$$

2. Окно Хэннинга

$$W(f) = \begin{cases} \frac{1}{2}(1 + \cos(\frac{\pi f}{F})), & \text{при } |f| \leq F \\ 0 \text{ в остальных случаях} & \end{cases} \quad (6)$$

3. Окно Кайзера

$$W(f) = \begin{cases} \frac{I_0(\alpha \sqrt{1 - (\frac{f}{F})^2})}{I_0(\alpha)} & |f| \leq F \\ 0 & \text{в остальных случаях} \end{cases} \quad \alpha \text{ может варьироваться} \quad (7)$$

Рассмотрим, что будет, если а) спектр сигнала не является конечным, т.е. как на рисунке рис. 1. и б) если восстанавливающий фильтр имеет полосу пропускания  $[-F, F]$  шире, чем спектр сигнала

а) Процедуру восстановления сигнала по его отсчетам  $a(k\Delta t)$  можно представить, как результат пропускания через идеальный фильтр низких частот  $\lambda_d(t) = \frac{1}{\Delta t} \operatorname{sinc}(\frac{\pi t}{\Delta t})$  и частотной характеристикой

$$H_d(t) = \operatorname{rect}\left(f + \frac{1}{2\Delta t}\right) \quad (8)$$

непрерывного сигнала  $\tilde{U}(t) = \sum_{k=-\infty}^{+\infty} U(k\Delta t) f(t - k\Delta t)$ ,

спектр которого — периодически продолженный с периодом  $\frac{1}{\Delta t}$  спектр сигнала  $U(t)$

$$U_\phi(f) = \sum_{m=-\infty}^{+\infty} U_\phi(f - \frac{m}{\Delta t}) \quad (9)$$

Если спектр  $U_\phi(f)$  ограничен  $(-F, F)$ , то выделяется строго один период спектра, соответствующий  $m=0$  и равный спектру  $U(t)$ .

Если спектр  $U(t)$  шире  $[-F, F]$ , то происходит перекрытие соседних периодов спектра сигнала  $\tilde{U}(t)$  и уже невозможно выделить спектр сигнала в чистом виде. Это явление называется **строб-эффектом**

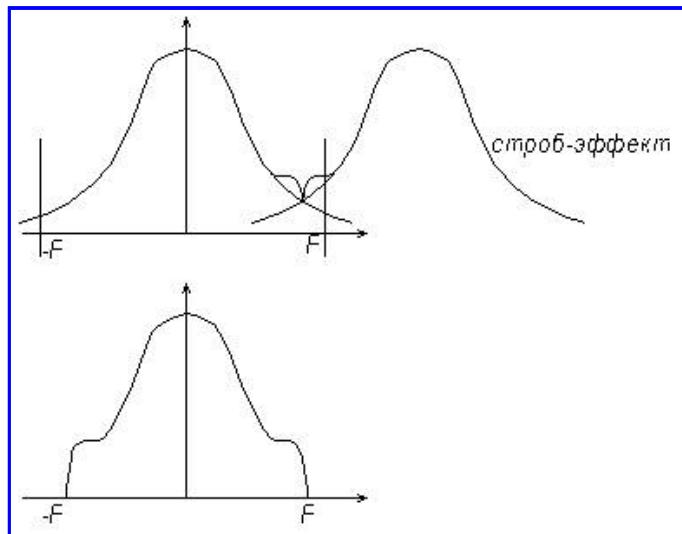


Рис. 2. Строб-эффект

6) Во втором случае в восстановленный сигнал после растирования с шагом  $\Delta t$  непрерывный сигнал попадают компоненты исходного сигнала из плюс-минус первого порядка периодически продолжительного спектра.

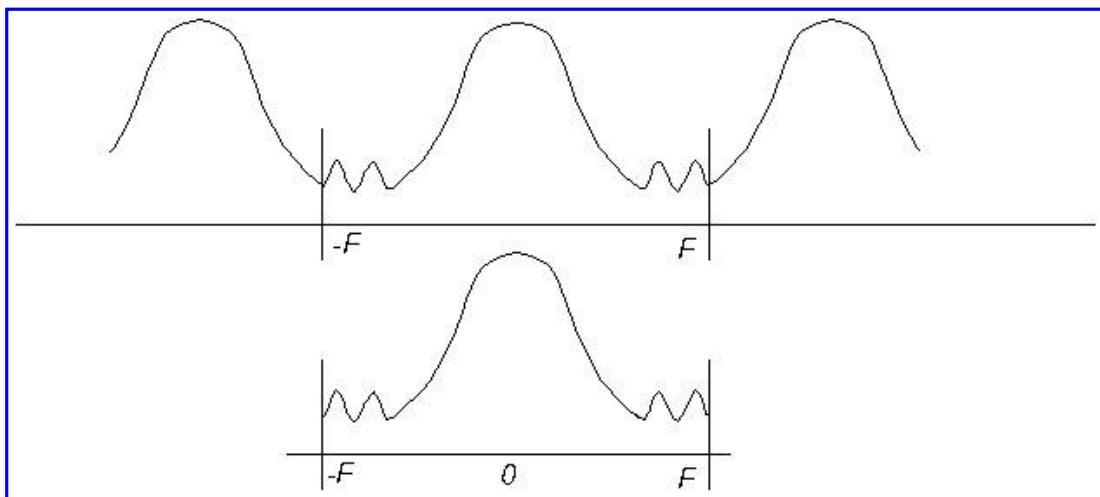


Рис. 3. Муар-эффект

Это явление называется **муар-эффектом**.

# Квантование сигнала

Поэлементное квантование: в области значений сигнала выбирается отрезок конечной длины, который разбивается на интервалы квантования, и значения, попадающие в каждый интервал, обозначаются одним числом- номером интервала. При восстановлении сигнала номер заменяется значением-представителем данного интервала. Способ разбиения на интервалы и значение представителя выбирается так, чтобы удовлетворялись требования к точности представления непрерывного сигнала цифровым.

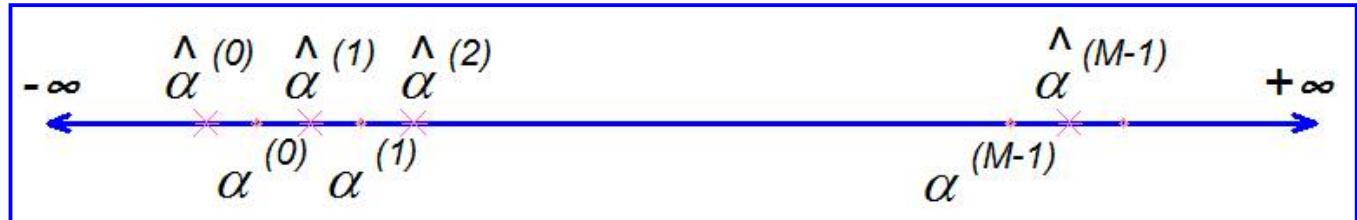


Рис. 1. Равномерное квантование

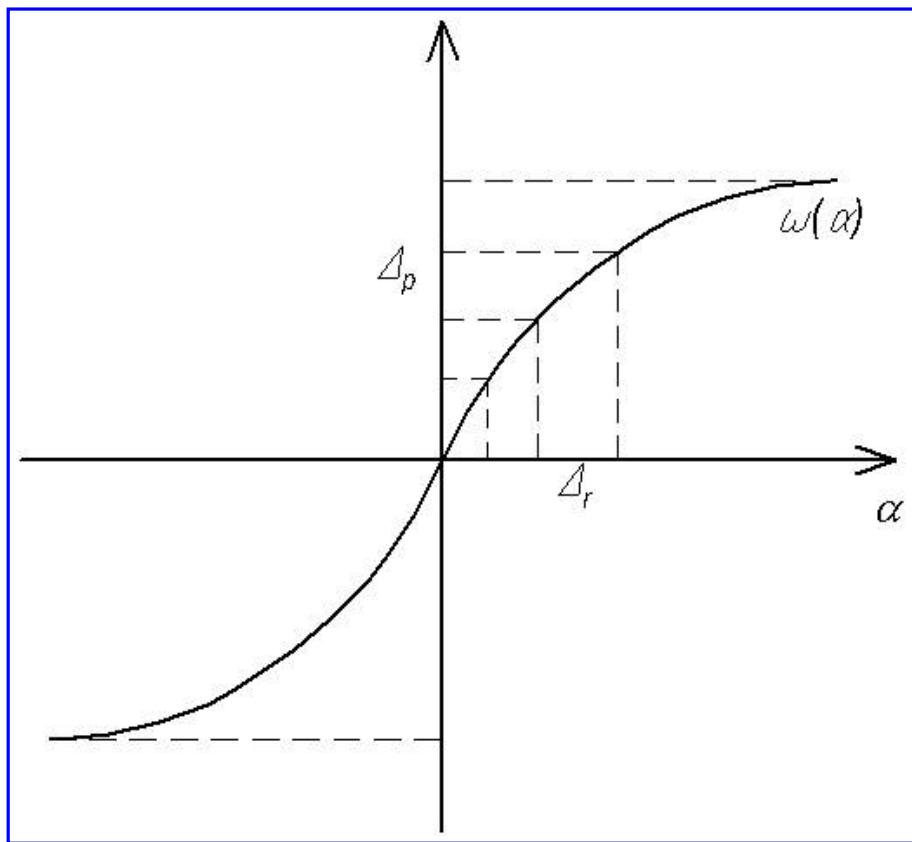


Рис. 2. Неравномерное квантование

$\alpha$  — коэффициент дискретного представления сигнала

$\hat{\alpha}^{(r)}$  значение представителя  $r$ -го интервала квантования значения  $\alpha$

Ошибка квантования  $\varepsilon^{(r)} = \alpha - \hat{\alpha}^{(r)}$

Подход:  $\alpha$  и  $\varepsilon^r$  — случайная величина.  $D_0(\varepsilon^{(r)})$  — функция потерь, связанная с отличием  $\alpha$  от  $\hat{\alpha}^{(r)}$ . Точность представления характеризуется

$$Q_0 = \sum_{r=0}^{M-1} \int_{\alpha^{(r)}}^{\alpha^{(r+1)}} P(\alpha) D_0(\varepsilon^r) d\alpha \quad (1)$$

$P(\alpha)$  — распределение вероятности потерь.

С учетом ошибок, связанных с конечностью интервала, точность квантования дополнится функциями

$$Q_{1\text{р}} = \int_{-\infty}^{\alpha^{(1)}} P(\alpha) D_{1\text{р}} \varepsilon^0 d\alpha \quad (2)$$

$$Q_{2\text{р}} = \int_{\alpha^{(M)}}^{\alpha^{(M)}} P(\alpha) D_{2\text{р}} \varepsilon^{(M)} d\alpha \quad (3)$$

Тогда **Оптимальное квантование** - это квантование, при котором достигается  $\min(Q_0, Q_{1\text{р}}, Q_{2\text{р}})$

Существующие устройства обычно выполняют равномерное квантование, при котором границы интервалов квантования размещаются равномерно в заданном диапазоне значений сигнала, а представители уровней располагаются посередине.(2)

Используя такие устройства, оптимальное квантование можно обеспечить, если перед квантованием сигнал подвергнуть нелинейному преобразованию (предискажению) и на выходе при восстановлении подвергнуть нелинейной коррекции. Для определения предискажающей функции удобно ошибку квантования оценить по средней величине потерь в пределах интервала квантования  $D(\Delta_r)$ ;  $\Delta_r = \alpha^{(r+1)} - \alpha^{(r)}$ , т.е.

$$Q_0 = \int_{\alpha^{(1)}}^{\alpha^{(M)}} P(\alpha) D(\Delta_r) d\alpha$$

Пусть  $\omega(\alpha)$  — функция предискажения,  $\Delta_p$ -интервал равномерного квантования, тогда

$$\frac{\Delta_p}{\omega(\alpha)} = \Delta_r \text{ и}$$

$$Q_0 = \int_{\alpha'}^{\alpha^{(M)}} P(\alpha) D\left(\frac{\Delta_p}{\omega(\alpha)}\right) d\alpha$$

Наиболее часто используется пороговый критерий

$$D(\Delta_r) = \begin{cases} 0, |\Delta_r| \leq \Delta_{\text{пор}} \\ 1, |\Delta_r| > \Delta_{\text{пор}} \end{cases}$$

В этом случае, чтобы  $Q_0 = \min$  ширина  $r$ -го интервала =  $2\Delta_{\text{пор}}$ , а представитель  $r$ -го уровня — значение  $\alpha$  в центре  $r$ -го интервала.

а) Если  $\Delta_{\text{пор}}$  не зависит от  $\alpha$ , то получается равномерная шкала квантования

$$M = \frac{\alpha_{\max} - \alpha_{\min}}{2\Delta_{\text{пор}}}$$

б)  $\Delta_{\text{пор}} = \delta_0 \alpha$  (ограниченная относительная ошибка квантования) В этом случае равномерному квантованию нужно подвергать нормированный логарифм.

$$\frac{\omega(\alpha) - \omega(\alpha_{\min})}{\omega(\alpha_{\max}) - \omega(\alpha_{\min})} = \frac{\ln(\frac{\alpha}{\alpha_{\min}})}{\ln(q)}$$

$$q = \frac{\alpha_{\max}}{\alpha_{\min}}, M = \frac{\ln q}{\delta_0}$$

(Так можно описать по закону Вебера-Фехнера требование к точности квантования, предъявляемые зрительной системой человека)

Так поступают на практике: перед равномерным квантованием сигнал подвергают логарифм-предискажению (компрессия), при восстановлении синтезированный сигнал подвергают потенцированию (экспандирование).

Количество уровней: Данные психофизических измерений показывают:  $\delta_0 \approx 1,5\%-2\%$  (порог относительной контрастной чувствительности зрения в обычных условиях освещенности) Существующие устройства воспроизведения изображений обеспечивают диапазон  $q \approx 100$ , отсюда  $M \approx 230$ . В настоящих устройствах квантования обычно  $M \div 128 \div 256$  уровней.

# Практические вопросы дискретизации и квантования

а) **Дискретизация** на практике спектр сигнала часто не известен. Поэтому параметры дискретизации определяют задаваясь, например, критерием максимального значения отличия сигнала от результата его восстановления по дискретному представлению.

Практический выбор: на интервале дискретизации функция не должна сильно меняться.

**Квантование:** Наиболее часто- задача квантования при обработке сигналов в цифровом процессоре.

Для ввода в ЭВМ отсчетов видеосигнала отводится 8 бит, что соответствует 256 уровням квантования. При обработке в ЭВМ -16 или 32 разряда. Перевод целых и вещественных чисел в байтовый формат и их обработка- типичная задача квантования. для этого необходимо в математическом и программном обеспечении иметь следующий блок:

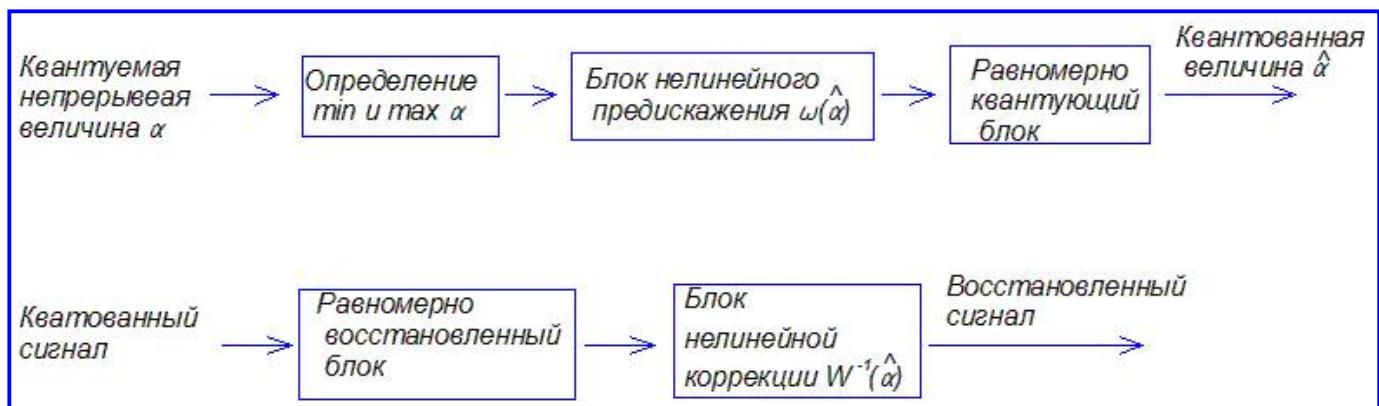


Рис. 1.

# Преобразование Фурье

Важнейшей характеристикой исходного сигнала является его преобразование Фурье. Если исходный сигнал задан функцией  $f(t)$ , заданной на всей вещественной оси, то его преобразование Фурье задается формулой:

$$F(\omega) = \int_{-\infty}^{+\infty} U(t) \exp(-i2\pi\omega t) dt \quad (1)$$

Обратное

$$U(t) = \int_{-\infty}^{+\infty} U_\phi(f) \exp(i2\pi\omega t) d\omega \quad (2)$$

Функция  $F(\omega)$  или ее модуль трактуется как интенсивность исходного сигнала на частоте  $\omega$ . Обратное преобразование задается аналогичной формулой:

$$f(t) = \int_{-\infty}^{+\infty} \exp(2\pi i\omega t) F(\omega) d\omega$$

Справедливость указанных формул возможна лишь при определенных ограничениях на исходные функции. В зависимости от наложенных ограничений данным формулам придают различный смысл. Мы не будем уточнять данное обстоятельство, предполагая, что все выполняемые операции типа изменения порядка интегрирования законны. Однако в любом случае при обычном понимании интегрирования необходимым условием является убывание функций на бесконечности. В реальных условиях это ограничение не имеет места, поэтому предварительно нужно ознакомиться со специальным математическим аппаратом, позволяющим в некоторых случаях обойти данное ограничение.

Прежде, чем переходить к изложению этого аппарата, напомним основные свойства преобразования Фурье. Для краткости связь между функцией и ее преобразованием Фурье будем обозначать так:  $f(t) \Rightarrow F(\omega)$ .

Если  $f(t) \Rightarrow F(\omega)$ ,  $g(t) \Rightarrow G(\omega)$  то  $af(t) + bg(t) \Rightarrow aF(\omega) + bG(\omega)$

$f(t-a) \Leftrightarrow \exp(2i\pi\omega a)F(\omega)$

Двойственное соотношение имеет вид:  $f(t)g(t) \Rightarrow F(\omega) * G(\omega)$

Вообще говоря, не предполагается, что функция  $f(t)$  - вещественная. Если же это так, то  $F(-\omega) = \bar{F}(\omega)$ . Эта формула получается формальным дифференцированием под знаком интеграла.

## Обобщенные функции

Как уже отмечалось, для того, чтобы в обычном смысле существовало преобразование Фурье от функции, необходимо ее убывание на бесконечности. Очевидно, что это не выполнено для стационарного сигнала. Для того, чтобы иметь возможность работать с преобразованием Фурье и от таких функций нужен вспомогательный аппарат.

Обозначим через  $K$  множество бесконечно дифференцируемых функций с компактным носителем. По определению, последовательность  $\varphi_n(x) \rightarrow \varphi(x)$ , если все эти функции имеют общий компактный носитель, принадлежат  $K$  и в каждой точке имеет место обычная сходимость. Функционал это отображение  $f: K \rightarrow C$ , причем  $f(\varphi_n) \rightarrow f(\varphi)$ . Если  $u(x)$  - интегрируемая функция, то ей соответствует функционал  $f(\varphi) = \int \varphi(x) u(x) dx$ . Однако существуют функционалы, не представимые в указанной форме. Например,  $\Delta\varphi = \varphi(0)$ . Этот функционал записывают в форме  $\int \varphi(x) \delta(x) dx$ . Наряду с указанным функционалом определяют функционалы  $\delta(x-a)$ ,  $\delta(tx)$ , исходя из формального правила замены переменных в интеграле. Хотя этот функционал нельзя представить с помощью обычной функции, можно ввести  $\delta$ -образную последовательность. Положим  $\delta_n(x) = n/2$  при  $-1/n \leq x \leq 1/n$  и 0 в остальных точках. Интеграл от нее равен 1. При больших  $n$  функция  $\varphi \in K$  представима в виде

$$\varphi(x) = \varphi(0) + x\varphi'(0) + O(n^{-2}) \text{ при } -1/n \leq x \leq 1/n, \text{ поэтому } \int_{-1/n}^{1/n} \varphi \delta_n(x) dx = \varphi(0) + O(1/n) \text{ (второе слагаемое исчезает в силу симметричности).}$$

Лемма. Пусть  $\varphi(x)$  имеет интегрируемую производную. Тогда  $\int \varphi(x) \cos(nx) dx \rightarrow 0$ ,  $n \rightarrow \infty$ .

Доказательство проводится интегрированием по частям. Аналогичное утверждение справедливо и для  $\sin(nx)$ .

## Преобразование Фурье и обобщенные функции

### Вспомогательные утверждения

## Вспомогательные утверждения

Лемма. Справедлива формула

$$\int_0^\infty \frac{\sin(x)}{x} dx = \pi/2 \quad (3)$$

Доказательство. Хотя формула (3) хорошо известна, мы приведем ее доказательство, поскольку она является основой многих дальнейших выкладок. Рассмотрим контур, изображенный на рис.1

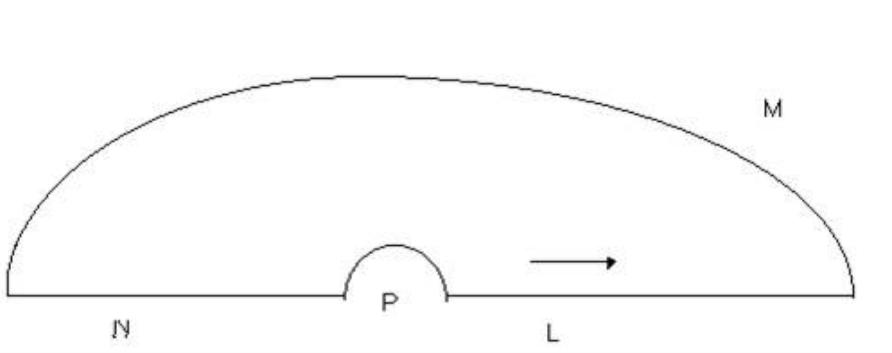


Рис 1. Контур интегрирования.

и интеграл по контуру в указанном направлении от аналитической функции  $\frac{\exp(iz)}{z}$ . Имеем  $\int \frac{\exp(iz)}{z} dz = 0$ , поскольку у функции нет особенностей внутри области интегрирования. Здесь контур **M** - дуга окружности радиуса **R**, а контур **P** - дуга окружности радиуса **ε**. Обе дуги имеют центр в начале координат. За исключением крайних точек, на контуре **M** выполнено неравенство  $\operatorname{Im} z > 0$ , поэтому с ростом **R** интеграл по этому контуру стремится к 0. Интегралы по контурам **P**, **M**, **L** в сумме дают  $2i \int_{\epsilon}^{\infty} \frac{\sin(x)}{x} dx$ . Найдем теперь интеграл по контуру **P**. Сделаем замену  $z = \epsilon \cdot \exp(it)$ . В результате

интеграл по этому контуру примет вид  $\int_{\pi}^0 \exp(\epsilon \cdot \exp(it)) idt = -\pi + o(\epsilon)$ . Последняя оценка получена в результате разложения подынтегральной функции в ряд. Устремляя **ε** к 0, завершаем доказательство.

Следствие 1.

$$\int_0^\infty \frac{\sin(Nx)}{x} dx = \pi/2$$

при любом **N**.

Доказательство проводится путем замены переменной

Следствие 2.

$$\int_0^a \frac{\sin(Nx)}{x} dx \rightarrow \pi/2, \quad N \rightarrow \infty.$$

Для любого **a > 0**.

Доказательство.  $\left( \int_0^a + \int_a^\infty \right) \frac{\sin(Nx)}{x} dx = \pi/2$ . Второе слагаемое стремится к 0 когда **N → ∞**.

Из соображений симметрии вытекает формула

$$\int_{-a}^a \frac{\sin(Nx)}{x} dx \rightarrow \pi, \quad N \rightarrow \infty, \quad a > 0 \quad (4)$$

## Пример отыскания обобщенных функций

Под обобщенной функцией понимается непрерывный функционал. Примером такой функции является **δ**-функция.

Предложение 1.  $1 = \delta(\omega)$ .

Доказательство. Очевидно, что обычное преобразование Фурье от 1 не существует.

Положим  $F_N(\omega) = \int_{-N}^N \exp(-2\omega\pi t) dt$ . Не существует обычного предела у этой функции при **N → ∞**. Найдем

функционал  $\int_a^b \phi(\omega) F_N(\omega) d\omega = \int_a^b \phi(\omega) \frac{\sin(2\pi\omega N)}{\pi\omega} d\omega$ . Если 0 не попадает в интервал интегрирования,

подынтегральная функция не имеет особенностей, и весь интеграл стремится к 0. В противном случае, интеграл стремится к  $\int_{-c}^c [\phi(0) + \omega\phi'(0) + O(\omega^2)] \frac{\sin(2\pi\omega N)}{\pi\omega} d\omega$ , где  $c$  произвольное малое положительное число. Второе слагаемое исчезает в силу симметричности, и при  $c \rightarrow 0, N \rightarrow 0$  получаем, используя (4), конечный результат.

Следствие 3.  $\exp(2\pi at) = \delta(\omega - a)$ . Доказательство. Формально утверждение есть следствие общего правила:  $\delta(\omega) = 1 \rightarrow \delta(\omega - a) = \exp(2\pi at)$ , но фактически надо доказать, что это правило распространяется и на обобщенные функции. Проще всего, дать прямое доказательство.

### Производные от обобщенных функций.

Производная определяется путем формального применения интегрирования по частям с учетом компактности носителя функций из  $\mathcal{K}$ :  $\int_{-\infty}^{\infty} \phi(\omega) f'(\omega) d\omega = - \int_{-\infty}^{\infty} \phi'(\omega) f(\omega) d\omega$ . В качестве примера рассмотрим обобщенную функцию  $\eta(\omega)$ , заданную равенством:  $\int_{-\infty}^{\infty} \phi(\omega) \eta(\omega) d\omega = \int_0^{\infty} \phi(\omega) d\omega$  и найдем производную от нее. Имеем  $\int_{-\infty}^{\infty} \phi(x) \eta'(x) dx = - \int_{-\infty}^{\infty} \phi'(x) dx$ . Это означает, что  $\eta'(x) = \delta(x)$ .

Замечание. Следует быть очень осторожным применяя к обобщенным функциям формулы, связывающие производную от функции и ее преобразование Фурье. В качестве примера рассмотрим отыскание преобразование Фурье от  $\eta(x)$ . Действуя формально, можем получить:  $\eta = F(\omega)$ ,  $\eta' = \delta \Rightarrow 2\pi i \omega F(\omega) = 1$ , откуда  $F(\omega) = \frac{1}{2\pi i \omega}$ . Теперь, исходя из определения, найдем правильный ответ.

Положим  $F_N(\omega) = \int_0^N \exp(-2\omega i \pi t) dt = \frac{1 - \exp(-2\omega i \pi N)}{2\omega i \pi}$  и подсчитаем  $\int_a^b \phi(\omega) F_N(\omega) d\omega$ . Если точка 0 не входит в интервал интегрирования, то интеграл стремится к  $\int_a^b \frac{\phi(\omega)}{2\pi i \omega} d\omega$ , то есть ожидаемый результат. Если же точка 0 принадлежит интервалу интегрирования, то наряду с указанным слагаемым появится еще одно.  $\int_{-c}^c \phi(\omega) F_N(\omega) d\omega = \int_{-c}^c \phi(\omega) \frac{1}{2\pi i \omega} [1 - \cos(2\pi\omega N) + i * \sin(2\pi\omega N)] d\omega$

Второе слагаемое исчезает в силу симметрии, а из третьего слагаемого получаем  $\delta$ -функцию. Окончательный результат выглядит так:  $\eta \Rightarrow \frac{1}{2\pi i \omega} + \delta(\omega)/2$ . Отметим, что отсюда получается правильный результат для преобразования Фурье от  $\delta$ -функции, поскольку  $\omega \delta(\omega) = 0$ .

Замечание. Интеграл  $\int \frac{\phi(\omega)}{\omega} d\omega$  существует в смысле главного значения для функции из  $\mathcal{K}$ . Это означает существование соответствующего функционала.

### Для двумерных функций.

$$U_\phi(f_1, f_2) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} U(x, y) \exp[-i2\pi(f_1 x + f_2 y)] dx dy$$

Основные свойства:

Теорема о свертке:

определение:

$$U(t) = \int_{-\infty}^{+\infty} a(\tau) b(t-\tau) d\tau \quad - \text{свертка} \tag{5}$$

Фурье-образ свертки равен произведению фурье-образов свертываемых функций, т.е.

$$U_\phi(f) = a_\phi(f) b_\phi(f)$$

Доказательство:

$$U_\phi(f) = \int_{-\infty}^{+\infty} U(t) \exp(i2\pi ft) dt = \int_{-\infty}^{+\infty} \exp(i2\pi ft) \int_{-\infty}^{+\infty} a(\tau) b(t-\tau) dt d\tau e^{i2\pi f\tau} e^{-i2\pi f\tau} =$$

$$*\zeta = t\tau$$

$$= \int_{-\infty}^{+\infty} d\tau e^{i2\pi f\tau} a(\tau) \int_{-\infty}^{+\infty} d\xi b(\xi) e^{i2\pi f\xi} = a_\phi(f) b_\phi(f)$$

# Преобразование Уолша

*Преобразование Уолша*

$$U(t) = \sum_{k=0}^{\infty} C_k W_{ak}(t), \quad m \quad (1)$$

где  $0 \leq t \leq 1$

$$C_k = \frac{1}{T} \int_0^T U(t) W_{ak}(t) dt \quad (2)$$

Если функция  $U$  определена на более широком интервале, то можно ввести  $\xi = \frac{t}{T}$  и тогда перейдем к нужному интервалу.

$$U(t) = \sum_{k=0}^{\infty} C_k W_{ak}\left(\frac{t}{T}\right) \quad (3)$$

$$C_k = \frac{1}{T} \int_0^T U(t) W_{ak}\left(\frac{t}{T}\right) dt$$

Для свертки справедливы те же свойства, что и для Фурье.

# Линейные и нелинейные фильтры

Преобразование **сигнала** каким-либо устройством в общем случае описывается оператором **линейный фильтр** - динамическая система, применяющая некий линейный оператор ко входному сигналу для выделения или подавления определённых частот сигнала и других функций по обработке входного сигнала

$$U_{\text{вых}} = T[U_{\text{вх}}] \quad (1)$$

Большой класс преобразующих систем-линейные.

Определение 1:

Преобразование называется линейным, если

$$T[aU_{1\text{вх}} + bU_{2\text{вх}}] = aT[U_{1\text{вх}}] + bT[U_{2\text{вх}}] = aU_{1\text{вых}} + bU_{2\text{вых}} \quad a, b - \text{const} \quad (2)$$

Определение:

Оператор, для которого условия (1) не выполнено называется нелинейным

$$T[U_{\text{вх}}] = U_{\text{вх}}^2 = U_{\text{вых}}$$

Определение:

Преобразователь называется инвариантным к сдвигу, если

$$T[U_{\text{вх}}(t-T)] = U_{\text{вых}}(t-T) \quad (3)$$

Наиболее часто используемый класс- линейные инвариантные к сдвигу системы (ЛИС)

Математическое описание ЛИС-систем

Имеем:

$$U_{\text{вх}}(t) = \int_{-\infty}^{+\infty} U_{\text{вх}}(t') \delta(t-t') dt'$$

Применим оператор Т к обоим частям:

$$T[U_{\text{вх}}(t)] = T \left[ \int_{-\infty}^{+\infty} U_{\text{вх}}(t') \delta(t-t') dt' \right] =$$

(из линейности Т)

$$= \int_{-\infty}^{+\infty} U_{\text{вх}}(t') T[\delta(t-t')] dt' = \int_{-\infty}^{+\infty} U_{\text{вх}}(t') H(t-t') dt'$$

Итак, для ЛИС-систем

$$U_{\text{вых}}(t) = \int_{-\infty}^{+\infty} U_{\text{вх}}(t') H(t-t') dt' \quad (4)$$

H(t)-называется функцией импульсного отклика или импульсным откликом системы.

Применив преобразование Фурье к обеим частям (4)

$$U_{\text{вых}}^\Phi(f) = U_{\text{вх}}^\Phi(f) H^\Phi(f) \quad (5)$$

H^\Phi(f)- передаточная функция системы.

# Дискретные и спектральные преобразования

**Дискретное преобразование сигнала**- представление непрерывного сигнала в форме конечного числа ординат  $N$

**Спектральное преобразование сигнала** - представление данного сигнала через его проекции на базис, образованный некоторой системой ортогональных функций.

Определение: последовательность называется периодической с периодом  $N$ , если

$$\tilde{U}(n+N)=\tilde{U}(n)$$

Теорема

Любая периодическая последовательность  $\tilde{U}(n)$  с периодом  $N$  может быть представлена дискретным рядом Фурье (ДФП)

$$\tilde{U}(n)=\frac{1}{N} \sum_{k=0}^{N-1} \tilde{V}(k) \exp(i \frac{2\pi}{N} kn), \quad (1)$$

$$\text{где } \tilde{V}(k) \sum_{n=0}^{N-1} \tilde{U}(n) \exp(-i \frac{2\pi}{N} kn) \text{ - коэффициенты ДПФ.} \quad (2)$$

Любой конечный сигнал можно периодически продолжить на всю числовую ось, т.е.

$$\tilde{U}(n)=\sum_{k=-\infty}^{+\infty} U(n-kN), \quad (3)$$

где  $N$ -период,  $\tilde{U}(n)$ -периодическое продолжение  $U(n)$ . Тогда:

$$U(n)=\begin{cases} \tilde{U}(n), & n \in \mathbb{R}_N = [0, N-1] \\ 0 & \text{в остальных случаях} \end{cases} \quad (4)$$

Тогда ДПФ можно записать:

$$V(k)=\sum_{n=0}^{N-1} U(n) \exp(-2\pi i \frac{kn}{N}) \quad (5)$$

$$U(n)=\frac{1}{N} \sum_{k=0}^{N-1} V(k) \exp(2\pi i \frac{kn}{N}),$$

где

$$V(k)=\begin{cases} \tilde{V}(k), & k \in [0, N-1] \\ 0 & \text{в остальных случаях} \end{cases}$$

## Восстановление дискретного сигнала

Наша цель - найти необходимые условия, при которых сигнал может быть восстановлен по дискретной выборке

Прежде всего, отметим часто используемый факт:  $\cos(2\pi\omega_0 t)=[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]/2$

## Преобразование Фурье от последовательности

Пусть имеется сигнал  $x(t)$ , и выбран шаг дискретизации  $S$ . Функция заменяется последовательностью  $y[n]=x(nS)$ .

Определение. Преобразованием Фурье от последовательности называется функция

$$Y(\omega)=\sum_{n=-\infty}^{\infty} y[n] \exp(-2\pi\omega iSn) \quad (6)$$

Отметим, что функция  $Y(\omega)$  является периодической. Часто ради простоты обозначений полагают  $S=1$ , и в этом случае период функции равен 1. Это принципиальное различие между преобразованиями Фурье от функции и последовательности. В то же время, оба преобразования тесно связаны. Положим

$$u(t)=\sum_{n=-\infty}^{\infty} \delta(t-nS). \text{ Тогда}$$

$$Y(\omega)=\int_{-\infty}^{\infty} u(t)x(t) \exp(-2\pi\omega it) dt \quad (7)$$

то есть является преобразованием Фурье от произведения двух функций, из которых одна -

обобщенная функция. Согласно общей теории, преобразование Фурье от произведения двух функций равно свертке образов сомножителей. Здесь мы отступаем от строгого изложения, поскольку уже справедливость (7) требует обоснования. Для упрощения обозначений положим  $S=1$ . Найдем  $U(\omega)=\int_{-\infty}^{\infty} u(t)\exp(-2\pi\omega t) dt$ . Снова положим  $U_N(\omega)=\sum_{n=-N}^{N} \delta(t-n)\exp(-2\pi i\omega t) dt = \sum_{n=-N}^{N} \exp(-2\pi i\omega n)$ . Обратим внимание на то, что это периодическая функция с периодом 1, представленная суммой геометрической прогрессии. Имеем:  $U_N(\omega)=\frac{\exp(-2\pi i N\omega)-\exp(-2\pi i(N+1)\omega)}{1-\exp(-2\pi i\omega)}$ . Умножим числитель и знаменатель на  $\exp(-\pi i\omega)$ . Получим:  $U_N(\omega)=\frac{\sin(2\pi(N+1/2)\omega)}{\sin(\pi\omega)}$ . В окрестности 0  $\int_{-c}^c \phi(\omega)U_N(\omega)d\omega = \int_{-c}^c \frac{\phi(\omega)\sin(2\pi(N+1/2)\omega)}{\pi i\omega} d\omega$  стремится при  $N\rightarrow\infty$  к  $\phi(0)$ . Таким образом в окрестности 0  $U_N(\omega)\rightharpoonup\delta(\omega)$ . В силу периодичности, имеем окончательный результат:  $U(\omega)=\sum_{k=-\infty}^{\infty} \delta(\omega-k)$ . Для произвольного S можем написать формулу:

$$\sum \delta(T-Sn) \rightharpoonup \sum \delta(\omega - \frac{n}{S}) / S \quad (8)$$

### **Связь между непрерывным и дискретным преобразованиями Фурье. Частота Найквиста.**

Используя формулы (7) и (8) и, предполагая верным утверждение о преобразовании Фурье от произведения функций, получаем:  $Y(\omega)=U(\omega)*X(\omega)$ , где  $x(t)\rightharpoonup X(\omega)$ , откуда вытекает

$$Y(\omega)=\sum X(\omega - \frac{n}{S}) / S \quad (9)$$

Эта формула устанавливает связь между непрерывным и дискретным преобразованиями Фурье. Как и следовало ожидать,  $Y(\omega)$  имеет период  $1/S$ , что согласуется с (6).

Предположим, что спектр исходного сигнала ограничен:  $|\omega| \geq M \Rightarrow X(\omega)=0$  для некоторого  $M$ . Выберем  $S$  таким образом, чтобы выполнялось неравенство

$$S < \frac{1}{2M} \quad (10)$$

В этом случае функция  $X(\omega)$  однозначно определяется функцией  $Y(\omega)$ . Значение  $2M$  называется частотой выборки Найквиста. Если частота выборки больше указанной величины, спектр непрерывного сигнала может быть восстановлен по спектру дискретного. Позже будет показано, что и сам непрерывный сигнал восстанавливается по дискретному.

### **Теорема Котельникова-Шеннона**

Эта теорема уточняет результат предыдущего пункта.

Если исходный сигнал имеет ограниченный спектр и выполнено условие (10), то непрерывный сигнал можно восстановить по дискретному.

Доказательство. Пусть спектр сигнала находится в интервале  $[-P, P]$ . Выберем произвольное  $M > P$ .

Тогда  $x(t)=\int_{-\infty}^{\infty} X(\omega)\exp(2\pi i\omega t) d\omega = \int_{-M}^{M} X(\omega)\exp(2\pi i\omega t) d\omega$ . Функцию, заданную на конечном интервале,

можно разложить в ряд Фурье:  $X(\omega)=\sum c_k \exp(-2\pi i k\omega/2M)$ , где  $c_k=\int_{-M}^{M} X(\omega)\exp(-2\pi i k\omega/2M) d\omega/2M$ .

Отсюда следует, что  $c_k=\frac{1}{2M}x(k/2M)$ . Теперь  $x(t)=\sum c_k \int_{-M}^{M} \exp(2\pi\omega(t-k/2M)i) d\omega$ . Положив  $S=\frac{1}{2M}$ .

Получим

$$x(t)=S \sum x(kS) \frac{\sin(2\pi M(t-kS))}{\pi(t-kS)} \quad (11)$$

Замечание. Обратим внимание, что в (10) должно выполняться строгое неравенство, если мы хотим, чтобы утверждение оставалось верным и для сигналов с преобразованием Фурье в виде обобщенной функции. В качестве примера рассмотрим  $x(t)=\sin(2\pi t)$ . Спектр сигнала сосредоточен на интервале  $[-1, 1]$ . Положим  $M=1$ , тогда  $S=1/2$ , но последовательность  $x(kS)$  оказывается нулевой. То есть непрерывный сигнал не удается восстановить по дискретным значениям. Если же  $S < 1/2$ , то можно воспользоваться формулой (11).

Как уже отмечалось, ДПФ является периодической функцией. В дальнейшем при изложении свойств ДПФ будем предполагать, что  $S=1$ . В этом случае период ДПФ равен 1. Обратное преобразование получается почлененным интегрированием ряда. Если  $X(\omega)\rightharpoonup x[n]$ , то обратное преобразование задается

формулой  $x[n] = \int_0^T X(\omega) \exp(2\pi i n \omega) d\omega$ . Данная формула вытекает из соотношения: интеграл  $\int_0^T \exp(2\pi i n t) dt$  равен 0 при  $n \neq 0$  и 1 иначе.

### Свертка

Свертка двух последовательностей определяется формулой:  $z_n = x_n \cdot y_n$ ,  $z_n = \sum_k x_{n-k} y_k$

Предложение. ДПФ от свертки двух последовательностей равняется произведению из преобразований Фурье, а ДПФ от произведения двух последовательностей есть свертка их преобразований Фурье.

Доказательство. Найдем преобразование от произведения последовательностей. Имеем  $x[n]y[n] = \int_0^T X(u) \exp(2\pi i n u) du \int_0^T Y(v) \exp(2\pi i n v) dv = \int_0^T \int_0^T Y(v)X(u) \exp(2\pi i n(u+v)) du dv$ .

В силу периодичности подынтегральных функций, получим  $\int_0^T \exp(2\pi i n \omega) \int_0^T Y(\omega-u)X(u) du$ .

Найдем ДПФ от свертки. По определению  $X(\omega) = \sum_n x[n] \exp(-2\pi i \omega n)$ ,  $Y(\omega) = \sum_k y[k] \exp(-2\pi i \omega k)$ .

Перемножая эти ряды и собирая коэффициенты при одинаковых степенях, получим  $X(\omega)Y(\omega) = \sum_n z[n] \exp(-2\pi i \omega n)$

Отметим очевидные следствия вещественности исходной последовательности:  $\bar{X}(\omega) = X(-\omega)$ .

### Пример вычисления ДПФ

Ранее было подсчитано ДПФ от единичной последовательности. В реальных условиях полагают, что в отрицательные моменты времени сигнал отсутствует. В этой связи интересно найти ДПФ от дискретного аналога функции  $\eta$ .

Предложение.  $\sum_{k=0}^{\infty} \delta(t-k) \Rightarrow \frac{1}{1 - \exp(2\pi i \omega)} + \sum \delta(\omega - k)/2$

Доказательство. Положим  $F_N(\omega) = \sum_{k=0}^N \exp(-2\pi i \omega k) = \frac{1 - \exp(2\pi i \omega(N+1))}{1 - \exp(-2\pi i \omega)}$ . Теперь

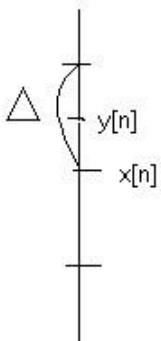
$$\int \phi(\omega) F_N(\omega) d\omega \rightarrow \int \phi(\omega) \frac{1}{1 - \exp(-2\pi i \omega)} + \sum \phi(k)/2$$

### Дискретный сигнал

Начиная с этого момента дальнейшее изложение ориентируется на компьютерную обработку сигнала. Основное отличие состоит в отсутствии понятия непрерывности, на котором базировался предыдущий материал.

### Шум от дискретизации

В результате перехода от непрерывного сигнала к дискретному возникает искажение. Реальный сигнал  $y[n] = x[n] + e[n]$ .



Здесь первое слагаемое - дискретный сигнал, а второе - ошибка. Пусть  $\Delta$  - длина интервала между соседними дискретными значениями. Предположим, что для представления сигнала используются  $B$  битов, а весь интервал возможных значений входного сигнала это  $[-X_M, X_M]$ . Тогда имеет место равенство  $\Delta = \frac{2X_M}{B}$ . В процессе дискретизации вместо самого сигнала берется ближайшее возможное

равенство  $\Delta = 2^B$ . В процессе дискретизации вместо самого сигнала беретсяближайшее возможное дискретное значение. В силу этого,  $-\Delta/2 < e[n] < \Delta/2$ . Согласно простейшей модели,  $e[n]$  имеет равномерное распределение на интервале изменения, поэтому дисперсия  $\sigma^2 = \Delta^2/12$ . Качество процедуры дискретизации определяется величиной  $Q = \frac{S^2}{\sigma^2}$ , где в числителе стоит дисперсия исходного сигнала. Заменяя  $S=X_M/4$ , получим  $Q = \frac{X_M^2/12}{16\Delta^2} = \frac{12 \cdot 2^{2B}}{16 \cdot 4} = \frac{3 \cdot 2^{2B}}{16}$ . На практике используется величина  $10 \cdot \lg Q$  и получается результат в децибелах. В нашем случае это  $20B \cdot \lg 2 - 10 \cdot \lg \frac{16}{3} \approx 6B - 7.2$ . Хороший уровень качества равен 90дБ, который достигается при  $B=16$ .

# Цифровое представление преобразований

При описании и построении цифровых преобразований непрерывных сигналов необходимо соблюдать принцип соответствия между непрерывными и цифровыми преобразователями.

Непрерывное и цифровое преобразования соответствуют друг другу, если одинаковые входные сигналы они преобразуют в одинаковые выходные.

Найдем дискретное представление **преобразования Фурье**.

При машинной обработке вместо интеграла Фурье приходится пользоваться его приближением, подсчитанным с помощью конечной суммы. В результате возникают дополнительные эффекты, а теория дискретного преобразования Фурье становится самостоятельной дисциплиной.

Рассмотрим  $N$ -мерное пространство последовательностей длины  $N$ . Каждый элемент этого пространства имеет вид  $(f(0), f(1), \dots, f(N-1))$ , где  $f(x)$  - некоторая функция, принимающая комплексные значения. В этом пространстве рассмотрим набор векторов, составленный из последовательностей  $\chi_k$ , построенных по функциям  $\exp(2\pi ikx/N)$ ,  $k=0,1,\dots,N-1$ . В пространстве определено скалярное произведение:  $\langle f, g \rangle = \sum_{k=0}^{N-1} f(k) \bar{g}(k)$ . Имеет место равенство  $\langle \chi_m, \chi_n \rangle = \delta_{m,n} N$ . Это означает что последовательности  $\chi_k$  составляют базис пространства. при этом для каждой произвольной функции  $f(x)$   $(f(0), f(1), \dots, f(N-1)) = \sum a_k \chi_k$ , где  $a_k = \langle f, \chi_k \rangle / N$ . Эти две формулы обычно записываются в виде

$$f(x) = \sum a_k \exp(2\pi i kx/N), \quad a_k = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp(-2\pi i kx/N) \quad (1)$$

и называют дискретным преобразованием Фурье. Из последней формулы следует, что  $a_k$  есть аналог значения преобразования Фурье исходной функции, вычисленного в точке  $k/N$ .

## Связь ряда Фурье и дискретного преобразования Фурье

Пусть периодическая на  $[0,1]$  функция задана формулой  $f(x) = \sum_{k=-\infty}^{\infty} f_k \exp(2\pi i kx)$ . Выберем  $N$  и найдем дискретное преобразование, используя значения функции в точках  $m/N$ . Легко видеть, что  $\sum_{k=0}^{N-1} \exp(2\pi i k(m-p)/N)$  равно  $N$  если  $N|(p-m)$  и 0 в противном случае. Отсюда следует, что коэффициент  $a_k$  в формуле (1), найденный по последовательности  $f(m|N)$ ,  $m=0,1,\dots,N-1$ , равен  $\sum_{p=-\infty}^{\infty} f_k p N$ . Этот эффект называют эффектом подмены частот, поскольку вместе с ожидаемой частотой в этот коэффициент вносят вклад и другие частоты.

## Преобразование вещественных последовательностей.

Если исходная последовательность вещественная, то в дискретном преобразовании Фурье присутствует избыточность, так как из вещественных чисел получается вещественных чисел. Из определения следует, что  $\bar{a}_k = \frac{1}{N} \sum f(x) \exp(2\pi i kx/N) = \frac{1}{N} \sum \exp(2\pi i (k-N)x/N) = a_{N-k}$ .

В этой связи рассматривают только коэффициенты  $a_k$ ,  $k=0,1,\dots,[N/2]$  (целая часть  $N/2$ ).

В качестве примера рассмотрим  $f(x) = \sin(2\pi kx)$ . У нее два обычных коэффициента:  $f_k = 1/2i$ ,  $f_{-k} = -1/2i$ . Учитывая эффект подмены, получим, что дискретные коэффициенты это  $a_k = f_k$ ,  $a_{-k} = f_{-k}$ . Согласно принятому соглашению, будет найден коэффициент с наименьшим индексом. Для того, чтобы с помощью дискретного преобразования найти истинную частоту  $k$  надо выбирать  $N > 2k$ . Поскольку значения истинных частот заранее не известны, сигнал нужно пропустить через фильтр низких частот, оставив лишь частоты из нужного диапазона.

Имеем для **сигнала** со спектром  $(-F, F)$

$$\begin{aligned} U(t) &= \sum_{k=-\infty}^{+\infty} U(k\Delta t) \operatorname{sinc}[\pi\Delta t(t-k\Delta t)] \quad \Delta t = \frac{1}{2F} \\ U^\Phi(f) &= \int_{-\infty}^{+\infty} U(t) \exp(i2\pi ft) dt = \sum_{k=-\infty}^{+\infty} U(k\Delta t) \int_{-\infty}^{+\infty} \operatorname{sinc}[\pi\Delta t(t-k\Delta t)] \exp(i2\pi ft) dt = \\ &= \frac{1}{\Delta t} \sum_{k=-\infty}^{+\infty} U(k\Delta t) \exp(i2\pi f \frac{k}{\Delta t}) \end{aligned} \quad (2)$$

$$2F \sum_{k=-\infty}^{+\infty} U(k) e^{-j2\pi k f} = \sum_{k=-\infty}^{+\infty} U(k) \delta(f - \frac{k}{T})$$

Поскольку сигнал  $U(t)$  ограничен  $(-\frac{T}{2}, \frac{T}{2})$ , то  $U^\Phi(f)$  можно тоже разложить по теореме Котельникова с шагом  $\Delta f = \frac{1}{T}$ , т.е.

$$U^\Phi(f) = \sum_{l=-\infty}^{+\infty} U^\Phi(l\Delta f) \operatorname{sinc}\left[\pi T(f - \frac{l}{T})\right], \quad (3)$$

где  $U^\Phi(l\Delta f) = T \int_{-\infty}^{+\infty} U^\Phi(f) \operatorname{sinc}\left[\pi T(f - \frac{l}{T})\right] df$  подставим (2)

$$\begin{aligned} &= \frac{1}{2F} \sum_{k=-\infty}^{+\infty} \exp(i2\pi f \frac{k}{2F}) T \int_{-\infty}^{+\infty} \operatorname{sinc}\left[\pi T(f - \frac{l}{T})\right] df U(k\Delta t) = \\ &= \frac{1}{2F} \sum_{k=-\infty}^{+\infty} U(k\Delta t) T \int_{-\infty}^{+\infty} \exp(i2\pi f \frac{k}{2F}) \operatorname{sinc}\left[\pi T(f - \frac{l}{T})\right] df = \\ &= \frac{1}{2F} \sum_{k=-\infty}^{+\infty} U(k\Delta t) \exp(i2\pi \frac{kl}{2FT}) \end{aligned}$$

Количество отсчетов  $N = 2FT \rightarrow (m.k. |f| < F \rightarrow |\frac{l}{T}| \leq F \quad |l| \leq 2FT \quad -FT \leq l \leq FT)$

Если обозначить

$$\begin{aligned} U^\Phi(l\Delta f) &= V(l) \frac{1}{2F} \exp(-i2\pi \frac{k_0 l}{N}) \\ N &= 2FT; \quad k_0 = \frac{N}{2}; \text{ то} \end{aligned} \quad (4)$$

$$V(l) = \sum_{k=0}^{N-1} U(k) \exp(i2\pi \frac{kl}{N}), \text{ это и т.д.}$$

Таким образом, осуществляется переход от непрерывного к дискретному ПФ для отсчетов сигнала и спектра.

Аналогично осуществляется переход к дискретному преобразованию Уолша:

$$U(n) = \sum_{k=0}^{N-1} C(k) W_k(\frac{n}{N}) \quad (5)$$

$$C(k) = \frac{1}{N} \sum_{n=0}^{N-1} U(n) W_k(\frac{n}{N})$$

$W_k(\frac{n}{N})$  - отсчеты функций Уолша.

# Свойства дискретных спектральных преобразований (на примере ДПФ)

1. Цикличность: (периодичность)

$$\tilde{V}(k+N) = \tilde{V}, \quad \tilde{U}(n+N) = \tilde{U}(n)$$

Доказательство:

$$V(k+N) = \sum_{n=0}^{N-1} U(n) \exp\left(-2\pi i \frac{kn}{N}\right) \exp\left(-2\pi i \frac{nN}{N}\right) = V(k) \exp\left(-2\pi i \frac{nN}{N}\right) = 1 \quad (1)$$

2. Симметрия:

$$V(N-k) \leftrightarrow \pm U(N-n) \quad (2)$$

$$V^*(k) \leftrightarrow U^*(N-n); \quad U^*(n) \leftrightarrow V^*(N-k) \quad (3)$$

$$V(k) = \pm V^*(k) \leftrightarrow U(n) = \pm U^*(N-n) \quad (4)$$

Доказательство: (на примере (3))

$$U^*(N-n) = \sum_{k=0}^{N-1} V^*(k) \exp\left(\frac{(N-n)k2\pi i}{N}\right) = \sum_{k=0}^{N-1} V^*(k) \exp\left(\frac{2\pi ikn}{N}\right) = \text{ДПФ}^{-1}(V^*(k))$$

3. Сдвиг

$$V[(k+k_0)\bmod N] \leftrightarrow U(n) \exp\left(\frac{-i2\pi k_0 n}{N}\right) \quad (5)$$

$$V(k) \exp\left(2\pi i \frac{k n_0}{N}\right) \leftrightarrow U[(n+n_0)\bmod N] \quad (6)$$

Доказательство:

$$\begin{aligned} & \sum_{n=0}^{N-1} U[(n+n_0)\bmod N] \exp\left(-2\pi i \frac{k(n)}{N}\right) = \\ & = \sum_{n=0}^{N-1} U[(n+n_0)\bmod N] \exp\left(-2\pi i \frac{k(n+n_0)}{N}\right) \exp\frac{kn_0}{N} 2\pi i = \\ & = \sum_{\xi=0}^{N-1} U(\xi) \exp\left(-2\pi i \frac{\xi k}{N}\right) \exp\frac{kn_0}{N} 2\pi i = V(k) \exp\frac{kn_0}{N} 2\pi i \end{aligned}$$

4. Дискретная теория отсчетов: Периодическое  $L$ -кратное продолжение последовательности приводит к раздвиганию отсчетов ее **ДПФ** с шагом  $L$ , причем появляющиеся при этом дополнительные отсчеты имеют нулевое значение. Аналогичное соотношение имеются и для периодического продолжения ДПФ-спектра.

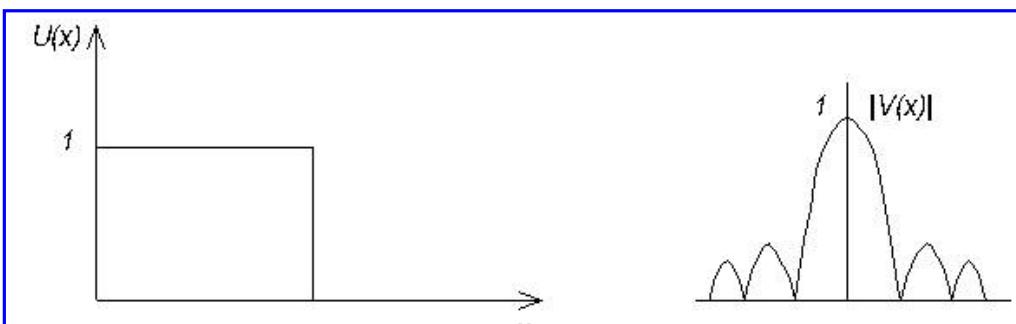
## Основные свойства ДПФ

$$V(k) = \sum_{n=0}^{N-1} U(n) e^{-\frac{2\pi i kn}{N}} \quad \text{— прямое ДПФ}$$

$$U(n) = \frac{1}{N} \sum_{k=0}^{N-1} V(k) e^{\frac{2\pi i kn}{N}} \quad \text{— обратное ДПФ}$$

1. Соотношение шага  $\Delta x$  по координате и шага  $\Delta V$  по частоте  $\Delta Y = \frac{1}{N \Delta x}$

2. Пример спектра:



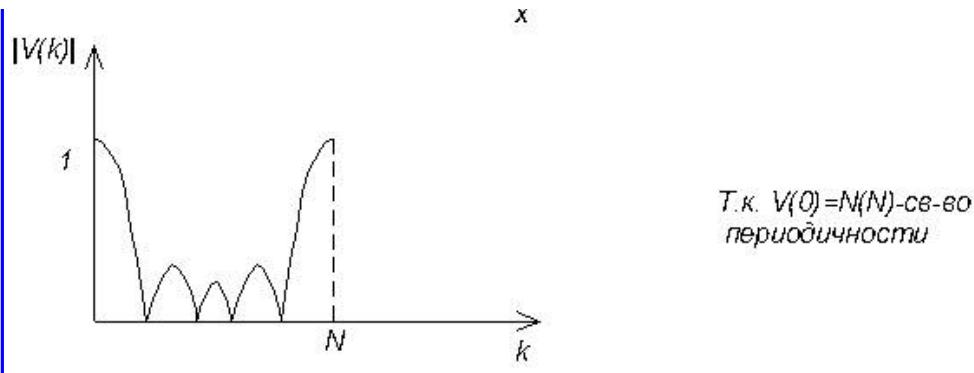


Рис. 1.

Для получения симметричной картины нужно умножить на  $(-1)^x$  перед вычислением ДПФ : центр  $\frac{N}{2}$ . Для двумерного ДПФ нужно умножить на  $(-1)^{x+y} \rightarrow$  центр смещается в точки  $(\frac{N}{2}, \frac{N}{2})$

### Гауссовы фильтры низких частот

$$H(v) = \exp\left(-\frac{D^2(v)}{2\sigma^2}\right) \text{ в одномерном}$$

$$H(v, V) = \exp\left(-\frac{D^2(v, V)}{2\sigma^2}\right) \text{ в двумерном}$$

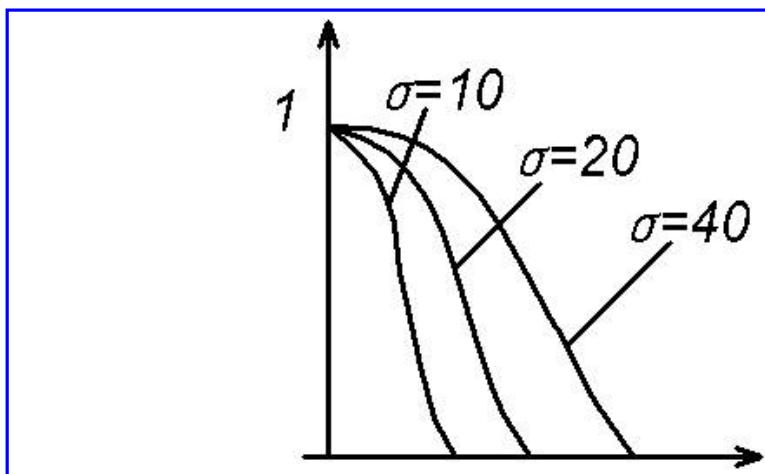


Рис. 2.

Пространственный гауссов фильтр не имеет "звонов"

### Частотные фильтры повышения резкости (фильтры высоких частот)

Подавляют низкочастотные составляющие, но подчеркивают контуры, которые связаны с высокочастотными составляющими. Передаточные функции могут быть получены с помощью следующего соотношения:

$$H_{B.C.}(v) = 1 - H_{N.C.}(v)$$

#### 1. Фильтры высоких частот Баттерворта

$$H(v) = \frac{1}{1 + [D_0/D(v)]^{2n}}$$

Теже самые эффекты, связанные со звонами, что и для низких частот.

$$2. Гауссов фильтр высоких частот H(v) = 1 - e^{-\frac{D^2(v)}{2\sigma^2}}$$

#### 3. Лапласиан

$$H(v) = -U^2$$

$$H(v, U) = -(U^2 + v^2)$$

В пространственной области

$$H(v, V) \leftrightarrow \nabla^2 f(x, y)$$

[www.preuhall.com/gonzalezwoods](http://www.preuhall.com/gonzalezwoods)

[www.imageprocessingbook.com/index\\_dipre.htm](http://www.imageprocessingbook.com/index_dipre.htm)

# Быстрые алгоритмы ДПФ и ДПУ

Впервые **БПФ** разработан Кули и Тьюки в 1965г. и дал толчок к использованию ЭВМ для спектральных преобразований.

Обычные формулы для вычисления ДПФ требуют большого количества умножений:  $(N-1)^2$ , где  $N$  — число точек в ДПФ. Существуют приемы, позволяющие уменьшить это количество. Они называются быстрыми схемами (БПФ). Простейшая относится к случаю  $N=2^n$ .

## Случай $N=2^n$ .

Любое число в интервале  $[0, N-1]$  однозначно представляется двоичным вектором длины  $n:m=m_0+2m_1+\dots+2^{n-1}m_{n-1}$ . Если последовательность  $a(0), a(1), \dots, a(N-1)$  задана, то положим  $c(k)=\sum_{m=0}^{N-1} a(m) \exp(-2\pi i km/N)$ . В дальнейшем, что упростить изложение, введем обозначение

$W=\exp(-2\pi i km/N)$ , откуда  $W^N=1$ . Имеем  $c(k)=\sum a(m_{n-1}, m_{n-2}, \dots, m_0) W^{k(m_0+2m_1+\dots+2^{n-1}m_{n-1})}$ . Основное замечание заключается в следующем: суммирование по индексу  $m$  равносильно суммированию по всем двоичным индексам  $m_j$ .  $c(k)=\sum_{m_0} W^{km_0} \cdot \sum_{m_1} W^{km_1} \dots \cdot \sum_{m_{n-1}} a(m_{n-1}, m_{n-2}, \dots, m_0) W^{k2^{n-1}m_{n-1}}$ , каждый из которых принимает два значения.

Для числа  $k$  существует аналогичное двоичное представление:  $k=k_0+2k_1+\dots+2^{n-1}k_{n-1}$ . Рассмотрим самую внутреннюю сумму.  $\sum_{m_{n-1}} a(m_{n-1}, m_{n-2}, \dots, m_0) W^{k2^{n-1}m_{n-1}}$ . Нетрудно видеть, что это некоторая функция  $F_1(m_{n-2}, \dots, m_0, k_0)$ . Следующая сумма принимает вид  $\sum_{m_{n-2}=0}^1 F_1(m_{n-2}, \dots, m_0, k_0) W^{k2^{n-1}m_{n-1}} = F_2(m_{n-3}, \dots, m_0, k_0)$ . Этот процесс продолжается. Окончательно имеем  $c(k)=F_n(k_{n-1}, k_{n-2}, \dots, k_0)$ . Количество сумм равняется  $n=\log_2 N$ , в каждой из которых лишь одно умножение. Для вычисления всех коэффициентов нужно  $N \log_2 N$  умножений. Другое преимущество этой схемы - экономный расход оперативной памяти.

## Случай $N=N_1N_2$ с взаимно простыми сомножителями.

Рассмотрим другой крайний случай, когда  $N=N_1N_2$  и  $(N_1, N_2)=1$ . В этом случае существуют целые  $n_1$ ,  $n_2$ , для которых  $n_2N_1+n_1N_2=1$ . Отсюда следует, что

$$n_2N_1+n_1N_2=1 \pmod{N} \quad (1)$$

При этом можно считать выполненными неравенства

$$0 \leq n_1 \leq N_1, 0 \leq n_2 \leq N_2 \quad (2)$$

Если такое неравенство для  $n_1$ , например, не имеет места, можно разделить на  $N_1$ . Для любого целого  $k$ ,  $0 \leq k \leq N$  из (1) вытекает  $k_2N_1+k_1N_2 \equiv k \pmod{N}$ . При ограничениях типа (2)  $k_1, k_2$  находятся однозначно. Имеем  $c(k)=\frac{1}{N} \sum_{x=0}^{N-1} a(x) W^{kx} = \frac{1}{N} \sum_{x=0}^{N-1} a(x) W^{(k_2N_1+k_1N_2)x}$ . Числа  $n_2, N_1, n_1, N_2$  — взаимно простые. Следовательно, имеем для любого целого  $x$ ,  $0 \leq x \leq N$   $x_2n_2N_1+x_1n_1N_2 \equiv x \pmod{N}$ . Теперь  $W^{kx}=W^{(x_2n_2N_1+x_1n_1N_2)(k_2N_1+k_1N_2)}$ . Раскрывая скобки и отбрасывая члены кратные  $N$ , получим показатель вида  $k_2x_2n_2N_1^2+k_1x_1n_1N_2^2$ .

Из равенства  $n_2N_1+n_1N_2=1$  следует, что  $n_2N_1^2 \equiv N_1(N)$ , поэтому весь показатель сравним с  $k_2x_2N_1+k_1x_1N_2$ . Это означает, что  $W^{kx}=W^{k_2x_2N_1} \cdot W^{k_1x_1N_2}$ . Вводя обозначения  $W_1=\exp(-2\pi i/N_1)$ ,  $W_2=\exp(-2\pi i/N_2)$  окончательно получим  $\frac{1}{N} \sum a(x) W^{kx} = \frac{1}{N_1} \sum_{x_1} W_1^{k_1x_1} \cdot \frac{1}{N_2} \sum_{x_2} W_2^{k_2x_2} a(x_1, x_2)$ . Это означает, что преобразование Фурье для  $N$  точек свелось к последовательному выполнению преобразования Фурье по  $N_2$  точкам, а затем — по  $N_1$  точкам результатов предыдущего преобразования. При этом потребуется не более, чем  $(N_1-1)^2(N_2-1)^2$  умножений. По сравнению с  $(N_1N_2-1)^2$  выигрыш небольшой. Если же для какого-либо из промежуточных случаев есть своя быстрая схема, выигрыш может получиться значительным.

## Рассмотрим более подробно основную идею:

$$V(k)=\frac{1}{N} \sum_{n=0}^{N-1} U(n) \exp\left(-2\pi i \frac{nk}{N}\right)$$

IV  $n=0$

Если принять, что экспоненты вычислены заранее, то количество умножений  $\sim N$ , количество сложений  $\sim N$

Для вычисления всего спектра  $0 \leq k \leq N-1$

количество умножений  $\sim N^2$

количество сложений  $\sim N^2$

Рассмотрим преобразование:

$$V(k) = \frac{1}{N} U(n) \exp\left(-2\pi i \frac{nk}{N}\right)$$

$$n=2m \quad m \div 0.. \frac{N}{2}-1=0..N_1-1; \quad N_1=\frac{N}{2}$$

$$V(k) = \sum_{m=0}^{N_1-1} U(2m) \exp\left(-2\pi i \frac{km}{N_1}\right) + \left(\sum_{m=0}^{N_1-1} U(2m+1) \exp\left(-2\pi i \frac{km}{N_1}\right)\right) \exp\left(-\frac{k2\pi i}{N_1}\right)$$

$$\sum_{m=0}^{N_1-1} U(2m) \exp\left(-2\pi i \frac{km}{N_1}\right) = S_0(k); \quad \sum_{m=0}^{N_1-1} U(2m+1) \exp\left(-2\pi i \frac{km}{N_1}\right) = S_1(k)$$

Эта процедура называется децимацией по времени.

Таким образом:

$$V(k) = S_0(k) + S_1(k) \exp\left(-\frac{k2\pi i}{N}\right) \quad (3)$$

$$V(k+N_1) = S_0(k) - S_1(k) \exp\left(-\frac{k2\pi i}{N}\right)$$

При этом  $0 \leq k \leq N-1$

(Использовали  $\exp(-\pi i) = -1$ )

Этот процесс принято представлять графически

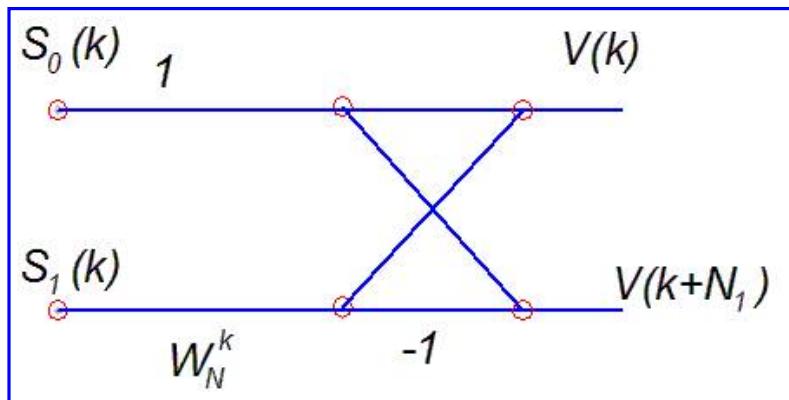


Рис. 1.

$$\text{где } W_N^k = \exp\left(-\frac{k2\pi i}{N}\right)$$

Схема получила название "бабочка"

Для вычисления одной бабочки нужно вычислить:

- 2 преобразования Фурье по основанию  $N_1$ ;
- одно умножение и два сложения.

Количество "бабочек" для вычисления всего спектра равно:

- умножения  $\sim (2N_1+1)N_1 = (N+1)\frac{N}{2}$

- сложения  $\sim (2N_1+2)N_1 = (N+2)\frac{N}{2}$

Если выбрать  $N=2^e$ ;  $l=\log_2 N$ , то процедуру прореживания можно повторить  $\log_2 N$  раз, пока не получаем двухточечное преобразование Фурье. На каждом шаге будет вычисляться  $\frac{N}{2}$  "бабочек", каждая из которых требует 1 умножение и 2 сложения. Таким образом, общее число умножений равно  $\sim \frac{N}{2} \log_2 N$ , т.е. выигрыш по сравнению с прямым вычислением в  $\frac{2N}{\log_2 N}$  раз.



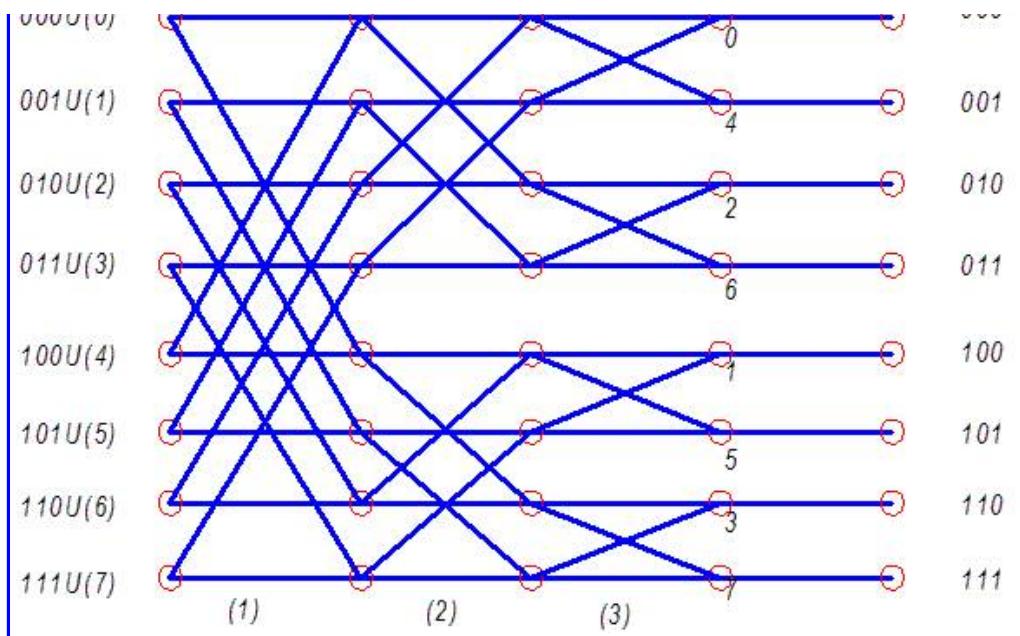


Рис. 2. прореживание по 4 бабочкам

Тестовый пример БПФ:

```

procedure PPT(var x:mas;sign, NP:integer; T:real);
var
nmax,i,nn,mm,lr,nw,ii,j,loc,nwl:integer;
zz,W,delta:real;
MSK:array[1..20] of integer;
cs:array[1..2] of real;
cxcs;xa,hold:complex;
label 1,2;
#include <stdio.h>
#include <math.h>
struct complex {float re, im;};
void fft(struct complex x[], int sign, int np,float t);
struct complex cmplx(float x,float y);
struct complex add(struct complex x,struct complex y);
struct complex red(struct complex x,struct complex y);
struct complex mult(struct complex x,struct complex y);
int st(int n,int np);
void main()
{
struct complex mas[32];
float remas[32];
int i;
for(i=0;i<32;i++)
{
mas[i].re=0;mas[i].im=0;
if(i>=8 && i<=23)
mas[i].re=1;
}
fft(mas,1,5,1);
for(i=0;i<32;i++)
{
remas[i]=sqrt(mas[i].re*mas[i].re+mas[i].im*mas[i].im);
printf("%f",remas[i]);
}
}
void fft(struct complex x[],int sign,int np,float t)
{
int msk[19];

```

```

float cs[2];
struct complex cxcs,xa,hold;
int nmax,i,j,nn,lr,nw,nwl,ii,ij,loc,ll;
float zz,pi,delta,w;
pi=M_PI;
nmax=st(2,np);
zz=2*pi*sign/nmax;
delta=t/nmax;
if(sign<0)
delta=1/t;
msk[0]=nmax/2;
for(i=1;i<np;i++)
msk[i]=msk[i-1]/2;
nn=nmax;
mm=2;
for(lr=1;lr<=np;lr++)
{
nn=nn/2;
nw=0;
for(i=1;i<=mm;i=i+2)
{
ii=nn*i;
w=nw*zz;
cs[0]=cos(w);
cs[1]=sin(w);
cxcs=cmplx(cs[0],cs[1]);
for(j=1;j<=nn;j++)
{
ii=ii+1;
ij=ii-nn;
xa=mult(cxcs,x[ii-1];
x[ii-1]=red(x[ij-1],xa);
x[ij-1]=add(x[ij-1],xa);
}
for(loc=2;loc<=np;loc++)
{
ll=nw-msk[loc-1];
if(ll<=0) break;
else nw=ll;
}
if (ll==0)
nw=msk(loc);
else
nw=nw+msk[loc-1];
}
mm=2*mm;
}
nw=0;
for(i=1;i<=nmax;i++)
{
nwl=nw+1;
hold=x[nwl-1];
if(nwl-i>0)
{
x[nwl-1].re=x[i-1].re*delta;
x[nwl-1].im=x[i-1].im*delta;
}
if(nwl-i>=0)
{
x[i-1].re=hold.re*delta;
x[i-1].im=hold.im*delta;
}
}
}

```

```

-----,
}
for(loc=1;loc<=np;loc++)
{
ll=nw-msk[loc-1];
if(ll<=0)
break;
else
nw=ll;
}
if(ll==0)
nw=msk[loc];
else
nw=nw+msk[loc-1];
}
}

struct complex cmplx(float x,float y)
{
struct complex z;
z.re=x;
z.im=y;
return(z);
}
struct complex add(struct complex x,struct complex y)
{
struct complex z;
z.re=x.re+y.re;
z.im=x.im+y.im;
return(z);
}
struct complex add(struct complex x,struct complex y)
{
struct complex z;
z.re=x.re+y.re;
z.im=x.im+y.im;
return(z);
}
struct complex red(struct complex x,struct complex y)
{
struct complex z;
z.re=x.re-y.re;
z.im=x.im-y.im;
return(z);
}
struct complex mult(struct complex x,struct complex y)
{
struct complex z;
z.re=x.re*y.re-x.im*y.im;
z.im=x.re*y.im+x.im*y.re;
return(z);
}
int st(int n,int np)
{
int m,i;
m=1;
for(i=1;i<=np;i++)
m=m*n;
return(m);
}

```

3. Прит выполнении процесса (3) вплоть до двухточечного ДПФ происходит процесс двоичного разложения частоты к вплоть до значения  $\kappa_0=0$  на начальном шаге (см NW=0) На каждом следующем

этапе к значению  $k$  добавляется следующий двоичный разряд. Этому отвечает процедура цикла по параметру "LOC".

При этом окончательное значение параметра  $k$  получается в инверсном порядке.

4. На последнем завершающем этапе происходит упорядочение массива значений в правильном порядке, т.е. инверсия двоичного кода числа  $k$ .

## Двумерное БПФ

Для двумерного **БПФ** можно получить следующий аналог "бабочки"

$$V(k_1, k_2) = S_{00}(k_1, k_2) + W_N^{k_2} S_{01}(k_1, k_2) + W_N^{k_1} S_{10}(k_1, k_2) + W_N^{k_1+k_2} S_{11}(k_1, k_2);$$

$$V(k_1 + \frac{N}{2}, k_2) = S_{00}(k_1, k_2) + W_N^{k_2} S_{01}(k_1, k_2) - W_N^{k_1} S_{10}(k_1, k_2) - W_N^{k_1+k_2} S_{11}(k_1, k_2); \quad (1)$$

$$V(k_1, k_2 + \frac{N}{2}) = S_{00}(k_1, k_2) - W_N^{k_2} S_{01}(k_1, k_2) + W_N^{k_1} S_{10}(k_1, k_2) - W_N^{k_1+k_2} S_{11}(k_1, k_2);$$

$$V(k_1 + \frac{N}{2}, k_2 + \frac{N}{2}) = S_{00}(k_1, k_2) - W_N^{k_2} S_{01}(k_1, k_2) - W_N^{k_1} S_{10}(k_1, k_2) + W_N^{k_1+k_2} S_{11}(k_1, k_2)$$

где

$$S_{00}(k_1, k_2) = \sum_{m_1=0}^{N_1-1} \sum_{m_2=0}^{N_1-1} U(2m_1, 2m_2) W_{N_1}^{m_1 k_1 + m_2 k_2},$$

$$S_{01}(k_1, k_2) = \sum_{m_1=0}^{N_1-1} \sum_{m_2=0}^{N_1-1} U(2m_1, 2m_2+1) W_{N_1}^{m_1 k_1 + m_2 k_2}; \quad (2)$$

$$S_{10}(k_1, k_2) = \sum_{m_1=0}^{N_1-1} \sum_{m_2=0}^{N_1-1} U(2m_1+1, 2m_2) W_{N_1}^{m_1 k_1 + m_2 k_2},$$

$$S_{11}(k_1, k_2) = \sum_{m_1=0}^{N_1-1} \sum_{m_2=0}^{N_1-1} U(2m_1+1, 2m_2+1) W_{N_1}^{m_1 k_1 + m_2 k_2}$$

Соответствующая этому преобразованию "бабочка"

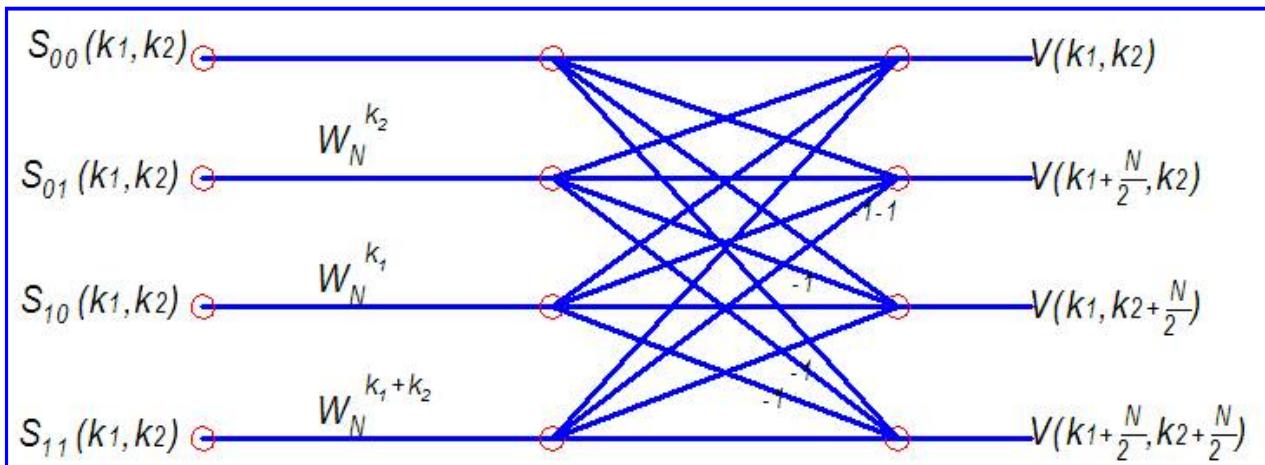


Рис. 1.

Каждая бабочка требует 3-х умножений и 8 сложений.

Для вычисления всех отсчетов  $V$  необходимо вычислить  $\frac{N^2}{4}$  "бабочек". Т.к. процедуру прореживания выполняют  $\log_2 N$  раз, то количество операций умножения необходимо выполнить  $\sim \frac{3N^2}{4} \log_2 N$

Нетрудно подсчитать, что при прямом вычислении двумерного **ДПФ** потребуется  $\sim N^4$  операций комплексного умножения.

# Совмешенные алгоритмы

Если последовательность  $U(n)$  — вещественна, то в соответствии с формулой **(Свойства дискретных спектральных преобразований (на примере ДПФ)3)**

$$V(k)=V^*(N-K) \quad (1)$$

Таким образом, достаточно вычислить  $V(k)$  только для  $k=0,1,\dots,\frac{N}{2}$ , а остальные найти по формуле (1).

Два способа реализации выигрыша:

1. Из двух последовательностей  $U(n)$  и  $W(n)$  длиной  $N$  образуют последовательность

$$Q(n)=U(n)+iW(n)$$

И вычисляют ее **БПФ**

$$\chi(k)=\sum_{n=0}^{N-1} Q(n)\exp(i2\pi\frac{kn}{N})$$

Тогда БПФ исходных последовательностей  $V(k)$  и  $P(k)$  находят:

$$V(k)=\frac{\chi(k)+\chi^*(N-k)}{2}; P(k)=-i\frac{\chi(k)+\chi^*(N-k)}{2} \quad (2)$$

Причем вычисления по формуле (2) достаточно вычислить только для  $k=0,1,\dots,\frac{N}{2}$ , а остальные можно найти по формуле (1).

Удобно для двумерных:  $U(n)$  и  $W(n)$  — соседние строки.

2. При преобразовании одномерных последовательностей вещественных чисел используется 2-ой способ, вытекающий из представления:

$$V(k)=\sum_{n=0}^{N/2-1} a_{2n}\exp(i2\pi\frac{nk}{(N/2)})+\sum_{n=0}^{N/2-1} a_{2n+1}\exp(i2\pi\frac{nk}{(N/2)})\exp[i\pi\frac{k}{(N/2)}] \quad (3)$$

Таким образом БПФ всей последовательности можно выполнить, вычислив БПФ от двух ее подпоследовательностей, составленных из четных и нечетных членов исходной последовательности, т.е.

$$V(k)=[V_k^{(ч)}+\exp(\frac{i\pi k}{(N/2)})V_k^{(н)}],$$

при этом коэффициенты  $V_k^{(ч)}$  +  $V_k^{(н)}$  можно найти, пользуясь 1-м способом по формулам (2)

# Рекуррентный алгоритм вычисления БПФ

При обработке сигналов, в особенности при обработке изображений, часто приходится вычислить спектры следующих друг за другом сильно перекрывающихся фрагментов сигнала (локальные спектры). В этом случае вычисление каждого фрагмента следует проводить рекуррентно, используя спектр предшествующего фрагмента.

Пусть  $V^{(0)}(k)$ -спектр 1-го фрагмента

$V^{(p)}(k)$ - спектр 2-го фрагмента, смещенного на  $p$  отсчетов

$$\begin{aligned} V^{(p)}(k) = & \sum_{n=0}^{N-1} U(n+p) \exp\left[i2\pi \frac{kn}{N}\right] = \left\{ \sum_{n=0}^{N-1} U(n) \exp\left(i2\pi \frac{kn}{N}\right) - \sum_{n=0}^{P-1} U(n) \exp\left(i2\pi \frac{kn}{N}\right) - \right. \\ & - \sum_{n=p}^{N-1} U(n) \exp\left(i2\pi \frac{kn}{N}\right) + \sum_{n=p}^{N-1} U(n) \exp\left(i2\pi \frac{kn}{N}\right) + \\ & \left. + \sum_{n=N}^{N+P-1} U(n) \exp\left(i2\pi \frac{kn}{N}\right) \right\} \exp\left(-i2\pi \frac{kP}{N}\right) = \\ = & V^{(0)}(k) \exp\left(-i2\pi \frac{kP}{N}\right) + \left\{ \sum_{n=0}^{P-1} [U(n+N) - U(n)] \exp\left(i2\pi \frac{kn}{N}\right) \right\} \exp\left(-i2\pi \frac{kP}{N}\right) \end{aligned} \quad (1)$$

Если  $P=1$  (скользящая обработка), то

$$V^{(1)}(k) = \left\{ V^{(0)}(k) + U(N) - U(0) \right\} \exp\left(-i2\pi \frac{k}{N}\right) \quad (2)$$

Использование формул (1) и (2) позволяет сократить число операций, необходимых при вычислении локальных спектров.

Тоже и для БПУ.

# Цифровая реализация линейного фильтра-цифровой фильтр

В соответствии с теоремой отсчетов:

$$U_{\text{вых},n} = \int_{-\infty}^{+\infty} U_{\text{вых}}(t) \operatorname{sinc}[2\pi F(t - \frac{n}{2F})] dt = \int_{-\infty}^{+\infty} U_{\text{вых}}(t) \lambda_n(t - n\Delta t) dt ,$$

$$\text{где } \Delta t = \frac{1}{2F} , \lambda_n = 2F \operatorname{sinc}[2\pi F(t - \frac{n}{2F})]$$

Выразим  $U_{\text{вых}}$  через свертку

$$\begin{aligned} U_{\text{вых},n} &= \int_{-\infty}^{+\infty} dt \lambda_n(t - n\Delta t) \int_{-\infty}^{+\infty} d\xi U_{\text{вх}}(\xi) h(t - \xi) d\xi = \\ &= \sum_{k=-\infty}^{+\infty} U_{\text{вх},k} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h(t - \xi) \lambda_n(t - n\Delta t) \lambda_k(\xi - k\Delta t) dt d\xi = \sum_{k=-\infty}^{+\infty} U_{\text{вх},k} h_{n-k} , \end{aligned} \quad (1)$$

где

$$h_{n-k} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h(t - \xi) \lambda_n(t - n\Delta t) \lambda_k(\xi - k\Delta t) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h(t - \xi + (n-k)\Delta t) \lambda_n(t) \lambda_k(\xi) \quad (2)$$

-дискретный аналог функции **импульсного отклика**. При цифровой обработке количество слагаемых ограничено, что связано либо с протяженностью сигнала, либо импульсного отклика.

Таким образом

$$U_{\text{вых},n} = \sum_{k=0}^{N-1} U_{\text{вх},k} h_{n-k} , \quad (3)$$

или

$$= \sum_{k=0}^{N-1} U_{\text{вх},n-k} h_n$$

Тогда импульсную реакцию непрерывного фильтра, соответствующую данному цифровому, можно представить в виде:

$$\tilde{h}(t, \xi) = \sum_{k=0}^{N-1} \sum_{n=0}^{N-1} h_n \lambda[t - (k-n)\Delta t] \lambda(\xi - n\Delta t) \quad (4)$$

Таким образом, по аналогии с формулами

$$W(t) = \begin{cases} \frac{I_0(a\sqrt{1-(\frac{f}{F})^2})}{I_0(a)} & |f| \leq F \\ 0 & \text{в остальных случаях} \end{cases} \quad \text{а может варьироваться}$$

для сигнала, если ввести

$$\tilde{h}(t, \xi) = \sum_{k=0}^{N-1} \sum_{n=0}^{N-1} h_{k,n} \delta(t - k\Delta t) \delta(\xi - n\Delta t) , \quad (5)$$

то спектр сигнала (5) есть периодически продолженной спектр функции  $\tilde{h}(t, \xi)$ . Если спектр ограничен, то есть умножен на функцию  $\operatorname{rect}$ , то получим формулу (4) восстановленного сигнала.

# Вычисление цифровой свертки с использованием БПФ

В основе лежит свойство преобразования Фурье от свертки, которые, как оказывается, справедливо соответственно и для дискретной свертки.

## Теорема о циклической свертке

$$\sum_{n=0}^{N-1} U_n h_{(k-n) \bmod N} \xleftrightarrow{\text{ДПФ}} V_e^\Phi H_e^\Phi , \quad (1)$$

т.е.  $\text{ДПФ}$  от циклической свертки равно произведению Фурье-образов свертываемых функций.

## Доказательство:

Пусть  $H_e^\Phi$  - ДПФ  $\{h_n\}$ ,  $V_e^\Phi$  - ДПФ  $\{U_n\}$

Рассмотрим, чему будет равно  $\text{ДПФ}^{-1}$  от произведения  $H_e^\Phi$  и  $V_e^\Phi$

Имеем:

$$\text{ДПФ}^{-1}(H_e^\Phi V_e^\Phi) = \frac{1}{N} \sum_{l=0}^{N-1} V_e^\Phi H_e^\Phi \exp(2\pi i \frac{kl}{N}) = \frac{1}{N} \sum_{l=0}^{N-1} \sum_{n=0}^{N-1} U_n H_e^\Phi \exp[2\pi i \frac{l(k-n)}{N}] =$$

Используя свойство (3) ДПФ о сдвиге, нетрудно увидеть

$$= \frac{1}{N} \sum_{n=0}^{N-1} U_n h_{(k-n) \bmod N} , \text{ что и т.д.}$$

Таким образом получается циклическая свертка, т.е. свертка двух функций, периодически продолженных за пределы заданного интервала.

## Отличие от обычной свертки:

Рассмотрим

$$U_{\text{вых},k} = \sum_{n=0}^{N-1} U_{\text{вх},n} h_{k-n} - \text{обычная свертка}$$

$$\tilde{U}_{\text{вых},k} = \sum_{n=0}^{N-1} U_{\text{вх},n} h_{(k-n) \bmod N} - \text{циклическая свертка}$$

Получим, что при  $k$  близких к 0 или  $N-1$  в свертке участвуют не соседние отсчеты  $h$ , а удаленные от него на всю длину последовательности  $N$ . Для борьбы с этими краевыми эффектами принимают принудительное удлинение сигнала и импульса отклика с доопределением их значений.

Принцип доопределения:

Пусть  $W_k$  - линейная свертка

$$W_k = \sum_{n=0}^{N-1} h_{(k-n)} U(n)$$

Очевидно, что  $W_k$  определена для всех  $k$ :

$$0 \leq k \leq 2N-1$$

Рассмотрим циклическую свертку:

$$y_k = \sum_{n=0}^{N-1} U(n) h_{(k-n) \bmod N} \quad (2)$$

Очевидно, что последовательность  $h(n)$  и  $U(n)$  необходимо доопределить до длины  $2N-1$ . Доопределение производится нулями. В этом случае циклическая свертка переходит в линейную.

Итак, алгоритм вычисления следующий:

1. Пусть  $N_1$ -размер сигнала  $U$ , а  $N_2$ -размер импульсного отклика  $h$ . Выберем  $N = N_1 + N_2 - 1$
2. Дополним  $U$  и  $h$  необходимым количеством нулей для заполнения области  $N$ .
3. Вычислим  $N$  точечное БПФ  $U$  и  $h$ .
4. Вычислим произведение  $U^\Phi(k) h^\Phi(k)$
5. Найдем обратное  $N$  точечное БПФ от произведения П.Ч.

Дополнение нулями приводит к увеличению числа операций. Поэтому использование БПФ приводит к выигрышу лишь для больших значений  $N$ .

## Последовательная свертка

В большинстве систем цифровой обработки сигналов (см. например, ОЭП) реализовано несколько последовательно соединенных фильтров. В этом случае процесс цифровой фильтрации можно записать следующим образом:

$$\begin{aligned} U_{\text{вых},1} &= U_{\text{вх},1} * * h_1; \\ U_{\text{вых},2} &= U_{\text{вых},1} * * h_2; \\ U_{\text{вых},p} &= U_{\text{вх},p-1} * * h_p; \end{aligned} \quad (1)$$

где  $* *$  - операция свертки,  $p$  - количество фильтров. Выполним **БПФ** от равенств (1) и осуществим подстановки:

$$U_{\text{вых},p}^{\Phi}(k) = U_{\text{вх},p}^{\Phi} * h_1^{\Phi}(k) * h_2^{\Phi}(k) * \dots * h_p^{\Phi}(k) \quad (2)$$

или

$$U_{\text{вых},p}^{\Phi}(k) = U_{\text{вх}}^{\Phi}(k) * H^{\Phi}(k) \quad (3)$$

где  $H^{\Phi}(k)$  — произведение передаточных функций всех фильтров. Таким образом, использование БПФ облегчает цифровую фильтрацию для каскада фильтров.

# Секционированная свертка

На практике часто имеет место ситуация, при которой размер изображения (сигнала) значительно превосходит размер **импульсного отклика**  $N_c \gg N_U$ . В этом случае используется операция свертки над секциями или блоками данных.

## Метод перекрытия с суммированием

Разделим массив сигнала  $U_{bx}(n)$  на  $N_U$  — точечные секции, определив секцию с номером 1 следующим образом

$$U_{bx,l}(n)=\begin{cases} U_{bx}(n) & l \cdot N_U \leq n \leq (l+1)N_U \\ 0 & \text{в остальных случаях} \end{cases} \quad (1)$$

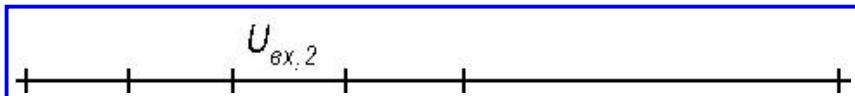


Рис. 1.

Пример опорной области такой секции. Т.к. опорные области секций не перекрываются, и покрывают всю опорную область массива  $U_{bx}(n)$ , то:

$$U_{bx}(n)=\sum_l U_{bx,l}(n)$$

Из линейности свертки можно записать:

$$\begin{aligned} U_{вых}(n) &= \sum_{k=0}^{N_c-1} U_{bx}(k)h(n-k) = \sum_{k=0}^{N_c-1} \sum_l \{U_{bx,l}(k)\}h(n-k) = \\ &= \sum_l \left\{ \sum_{k=0}^{N_c-1} U_{bx,l}(k)h(n-k) \right\} = \sum_l U_{вых,l}(n) \end{aligned}$$

Выходная секция  $U_{вых,l}(n)$  представляет результат свертки  $h(n)$  с секцией  $l$  последовательности  $U_{bx}(n)$ . Т.к. опорная область секции  $U_{вых,l}(n)$  больше опорной области  $U_{bx,l}(n)$  выходные секции должны перекрываться.

Свертки  $U_{bx,l}(n)$  с  $h(n)$  можно вычислить, используя БПФ.

# Одномерные фильтры

## Классификация фильтров

Исторически, фильтры описываются дифференциальными уравнениями с постоянными коэффициентами. Для цифровых фильтров это уравнение переходит в разностного типа:

$$y(i) = \sum_{k=0}^K b_k x(i-k) - \sum_{l=1}^L a_l y(i-l), \quad (1)$$

где  $y$  - выходной,  $x$  - входной сигналы.

1. Фильтры, у которых отличен от 0 хотя бы один коэффициент  $a_i$  называются рекурсивными или фильтры с бесконечной импульсной характеристикией (**БИХ-фильтры**)
2. Фильтры, у которых все  $a_i=0$  называются нерекурсивными или с конечной импульсной характеристикией (**КИХ-фильтры**)

Наиболее часто принимаются фильтры 2-го порядка вида:

$$y(i)=b_0x(i)-a_1y(i-1)-a_2y(i-2) \quad (2)$$

Передаточную функцию такого фильтра можно найти, выполнив преобразование Фурье.

$$Y(f)=b_0X(f)-a_1 Y(f)\exp(-i2\pi ft)-a_2 Y(f)\exp(-i4\pi ft) \quad (3)$$

$$\tau: Y(f)=\tau \sum_{j=-\infty}^{+\infty} y(j)\exp(-i2\pi jf\tau)$$

$\tau$ - шаг дискретизации. Передаточная функция:

$$\frac{Y(f)}{X(f)}=H(f)=\frac{b_0}{1+a_1\exp(-i2\pi f\tau)+a_2\exp(-i4\pi f\tau)} \quad (4)$$

Это передаточная функция любого рекурсивного фильтра 2-го порядка.

# Базисные фильтры

Наиболее широко применяются при обработке данных и сигналов.

## 1. Фильтр низких частот (низкополосный фильтр).

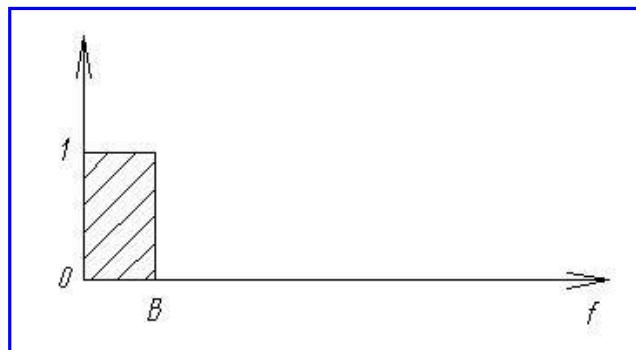


Рис. 1.

На практике нельзя реализовать фильтр с такой характеристикой, так как это неосуществимо физически. Ребра или углы графика обычно несколько сглажены, и имеется информация с частотами, не превосходящими  $B$ .

Реальная аппроксимация таких фильтров - синусный НЧ фильтр Баттеруорта.

$$|H(f)|^2 = \frac{1}{1 + \left(\frac{\sin \pi f t}{\sin \pi B t}\right)^4} \quad (1)$$

Вычисление коэффициентов  $a_1$ ,  $a_2$  и  $b_0$  такого фильтра выполняется по следующей программе.

```
Program LP2;
var
  real a1,a2,b0,p,q,c,d,E,B,T,fact,Fi
begin
  Fi:=pi/4;
  fact:=sin(pi*B*T);
  q:=fact*sin(Fi);
  c:=1-sqr(fact);
  d:=-c+sqrt(sqr(c)+4*sqr(q))/2;
  E:=(sqrt(d)+sqrt(d+1));
  a1:=(2*sqr(q)/d-1)*2/sqr(E);
  a2:=1/(sqr(E)+sqr(E));
  b0:=1+a1+a2;
end
```

## 2. Фильтр высоких частот (2-го порядка).

Фильтр высоких частот пропускает частоты в диапазоне  $(F-B; F)$  и подавляет низкие частоты.

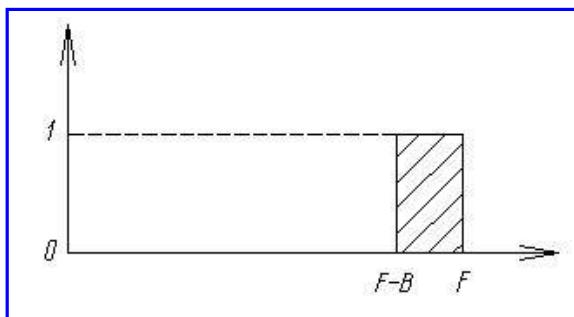


Рис. 2.

$$F \cdot \tau = \frac{1}{\omega_n}$$

... 2F

Хорошо аппроксимируется синусным фильтром высоких частот Баттеруорта:

$$|H(f)|^2 = \frac{1}{1 + \left(\frac{\sin(\pi f\tau + \pi/2)}{\sin \pi B\tau}\right)^4} \quad (2)$$

В диапазоне от 0 до F-B почти вся информация гасится, а в интервале (F-B, F) проходит почти беспрепятственно.

Как видно из (2), эта передаточная функция только углом  $\pi/2$  отличается от НЧ фильтра. В результате в программе для фильтра ВЧ по сравнению с предыдущей изменится только одна формула:

$$a1 := (2 * \text{sqr}(q) / d - 1) * 2 / \text{sqr}(E);$$

то есть изменяется знак на противоположный.

### 3. Полосопропускающий фильтр (2-й порядок).

Полосопропускающий фильтр пропускает частоты в диапазоне  $(f_c - B; f_c + B)$ .

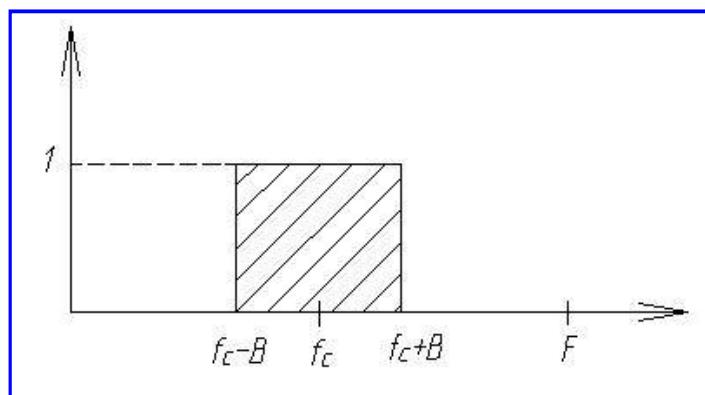


Рис. 3.

Он реализуется в виде полосопропускающего фильтра Баттеруорта 2-го порядка с передаточной функцией:

$$|H(f)|^2 = \frac{1}{1 + \left(\frac{\cos(2\pi f_c \tau) - c}{S}\right)^2} \quad (3)$$

где  $c = \cos(2\pi f_c \tau) \cos(\pi B\tau)$ ;  $S = \sin(2\pi f_c \tau) \sin(\pi B\tau)$ . Максимум этой функции достигается для частоты  $f_m$ , определяемой равенством:

$$\cos(2\pi f_m \tau) = \cos(2\pi f_c \tau) \cos(\pi B\tau),$$

что при  $B \ll \frac{1}{\tau}$  эквивалентно

$$f_c \approx f_m.$$

Программа по реализации подобного фильтра имеет вид:

```
program BNPS;
var
  T, B, FC, fact, ang2, cc, freq, ang, p, q, c, d, E, G, H, a1, a2, b0;
begin
  fact:=pi*T*B;
  ang2:=2*pi*FC*T;
  cc:=cos(ang2)*cos(fact);
  freq:=atan(sqrt(1.-cc*cc),cc)/(2*pi));
  ss:=sin(ang2)*sin(fact);
  ang:=pi/2;
  p:=ss*cos(ang)+cc;
  q:=ss*sin(ang);
  c:=1-(p*p+q*q);
  d:=0.5*(-c+sqrt(c*c-4*q*q));
  E:=sqrt(d+1)+sqrt(d);
  G:=2*sqrt(1-q*q/d)/E;
  if (p<0) then G:=-G;
  H:=-1/(E*E);
```

```
--'--'
a1:=-G;
a2:=-H;
b0:=1+a1+a2;
end;
```

a1 и a2 определить из условий нормировки.

#### 4. Заграждающий фильтр.

*Заграждающий фильтр* поглощает всю частотную информацию в диапазоне  $(f_c - B; f_c + B)$ , оставляя всю информацию вне этого диапазона без изменений.

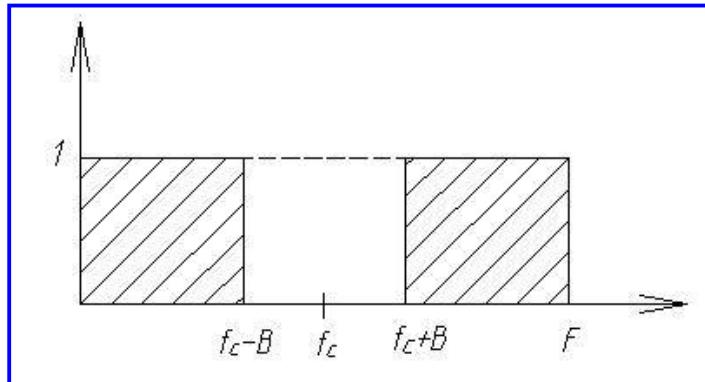


Рис. 4.

Передаточная функция заграждающего фильтра второго порядка:

$$|H(f)|^2 = 1 - \frac{1}{1 + \left(\frac{\cos(2\pi f t) - c}{S}\right)^2} \quad (4)$$

Нетрудно видеть, что заграждающий фильтр имеет еще и нули. Поэтому ему отвечает разностное уравнение

$$y(i) = b_0 X(i) + b_1 X(i-1) + b_2 X(i-2) - a_1 y(i-1) - a_2 y(i-2). \quad (5)$$

Коэффициенты  $a_1$ ,  $a_2$  определяются как и в предыдущем случае.  $b_0$ ,  $b_1$ ,  $b_2$  оказываются равными:  $b_0=1$ ,  $b_1=-2c$ ,  $b_2=1$ . Их необходимо отнормировать, исходя из того, что  $|H(0)|^2=1$ .

# Нерекурсивные КИХ-фильтры

Предложил Ормсби. Фильтр НЧ.

$$y(i) = \sum_{k=-M}^M b_k x(i-k), \quad (1)$$

где  $b_k = \tau \int_{-B}^B \exp(i2\pi f k \tau) H(f) df$  - веса фильтра.

Сначала в качестве весовой функции  $H(f)$  выбирается прямоугольник. При этом

$$b_k = \frac{\sin(2\pi B k \tau)}{k \tau}.$$

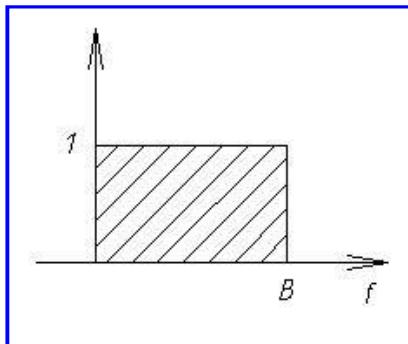


Рис. 1.

Однако веса убывают медленно, поэтому  $M$  может быть большим.

Фильтр улучшила замена прямоугольного импульса на трапециевидный.

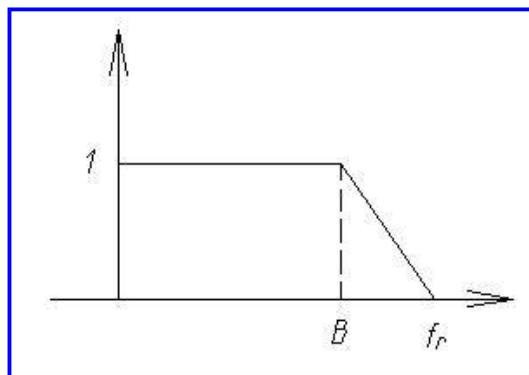


Рис. 2.

$$b_k \sim \frac{1}{k^2}$$

## Фильтр Поттера.

Программа получения весов. Принцип:

1. Сначала вычисляются веса прямоугольного импульса.
2. Умножаются на функцию, определяющую фильтр. Она улучшает характеристику фильтра за счет выхода за пределы **полосы пропускания**. Можно использовать почти все окна (Хэннинга, Хэмминга и т.д.)

Поттер использовал окно Р310.

$$\omega_k = \frac{c_k}{\omega} [d_0 + 2 \sum_{p=-3}^3 d_p \cos \frac{\pi p k}{M}]$$

$$c_k = \begin{cases} 1/2 & \text{при } k = \pm M \\ 1 & \text{в остальных случаях} \end{cases}$$

$$d_0 = 1, d_1 = d_{-1} = 0,684988, d_{-2} = d_2 = 0,202701, d_3 = d_{-3} = 0,0177127$$

$$\omega = d_0 + 2 \sum_{p=-3} d_p = 2,8108034$$

```

program LPSPBG;
const d0=0.35577019;
d:array[1..3] of real;
var bk:array[1..100] of real;
fact,sumg,B,T,sum:real;
M1,k,i,M:integer;
begin
fact:=2*B*T;
bk[1]:=fact;
fact:=fact*pi;
for i:=1 to M do
begin
bk[i+1]:=sin(fact*i)/(i*pi);
end;
bk[M+1]:=bk[M+1]/2; /*трапециевидное сглаживание в конце*/
sumg:=bk[1];
for i:=1 to M do
begin
sum:=d0;
fact:=pi*i/M;
for k:=1 to 3 do
sum:=sum+2*d[k]*cos(fact*k);
bk[i+1]:=bk[i+1]*sum;
sumg:=sumg+2*bk[i+1];
end;
M1:=M+1;
for i:=1 to M1 do
bk[i]:=bk[i]/sumg;
end.
```

Можно определить массив следующим образом:

```

d1=0.2436983;
d2=0.07211497;
d3=0.00630165;
```

# Фильтры низких частот

Фильтры для сглаживания изображений.

1. Фильтры низких частот Баттервортта:

$$H(v) = \frac{1}{1 + [D(v)/D_0]^{2n}},$$

где  $D(v)=v$ .

В двумерном случае

$$H(v,v) = \frac{1}{1 + [D(v,v)/D_0]^{2n}}$$

$$D(v,v) = [v^2 + v^2]^{1/2}$$

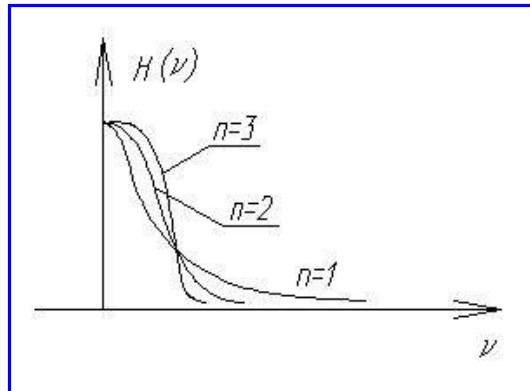


Рис. 1. Вид функции

При использовании **функций Баттервортта** более высоких порядков возникают "звоны" (эффекты Гиббса).

Рассмотрим ФБ в пространственной области.

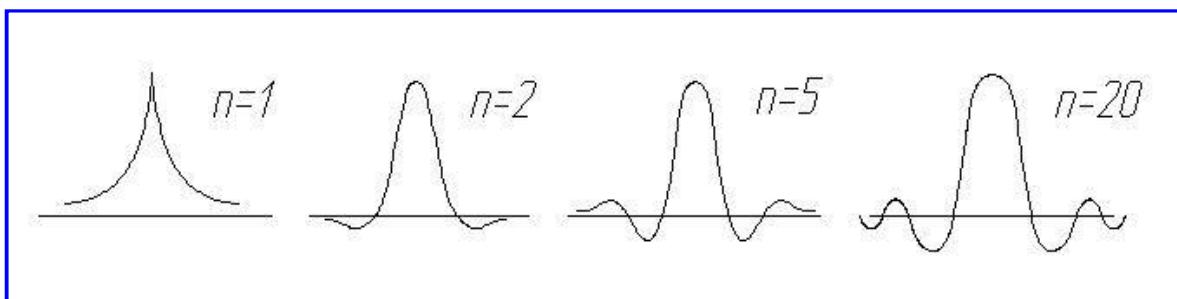


Рис. 2.

С увеличением порядка "звон" увеличивается.

$$3. V(0) = \sum_{n=0}^{N-1} u(n) = \langle u \rangle N$$

$\langle u \rangle$  - среднее значение.

4. Сдвиг:

$$u(x) e^{-\frac{2\pi i u_0 x}{N}} \leftrightarrow V(u-u_0)$$

и

$$u(x-x_0) \leftrightarrow V(u) e^{-\frac{2\pi i u_0 u}{N}}$$

Отсюда ясно домножение на  $(-1)^x$ .

$$(-1)^x = e^{i\pi x} = e^{-\frac{2\pi i u_0 x}{N}}; u_0 = N/2$$

$$u(x)(-1)^x \leftrightarrow V(u-N/2)$$

5. Дистрибутивность и изменение масштаба:

$$F(u_1+u_2) = F(u_1) + F(u_2)$$

$$u(ax) \leftrightarrow \frac{1}{|a|} V\left(\frac{v}{a}\right)$$

6. Периодичность и симметрия:

$$u(v+N)=V(v)$$

$$u(v)=V^*(-v)$$

для действительных функций  $u(N)$ . Поэтому  $|u(v)|=|u(-v)|$ .

# Реальный сигнал

*Реальный сигнал*-результат наложения детерминированного сигнала и случайного помехового сигнала.

Случайный сигнал(помеха)-результат (с точки зрения описания) *статистического моделирования*. При статистическом моделировании результат получается путем того или иного усреднения характеристик и данных, получаемых в наборе отдельных экспериментов. Поэтому статистические модели характеризуются:

- наличием генератора случайных чисел;
- наличием блоков вычисления статистических характеристик.

Случайные сигналы генерируются в виде реализаций псевдослучайных последовательностей с заданными статистическими свойствами.

# Генерация случайных сигналов

Стандартный способ

*Стандартный способ генерации*- генерация независимых псевдослучайных чисел и их линейное или нелинейное преобразование для получения заданных статистических свойств.

1. Генерация независимых псевдослучайных чисел с равномерным законом распределения на заданном интервале.

2. Линейное или нелинейное преобразование для получения заданных статистических свойств.

Генерация случайных независимо и равномерно распределенных чисел:

$$\xi_k = (c_1 \xi_{k-1} + c_2) \bmod c_3 \quad (1)$$

Здесь  $\xi_k$  — k-ое псевдослучайное число,  $\xi_{k-1}$  — предыдущее число,  $c_1, c_2, c_3$  — константы

$\xi_0$  — обычно выбирается произвольно.

$c_3$  — определяется величиной разрядной сетки применяемого процессора — максимально возможно целое число.

$c_2$  — незначительно влияет на качество последовательности (обычно  $c_2=0$ )

$c_1$  — выбор этой величины критичен.

Пример стандартной программы:

Генерация гауссовой последовательно псевдослучайных чисел.

Пусть  $\{\xi_k^{re}\}_{12}, \{\xi_k^{im}\}$  — два отрезка последовательности одинаково распределенных (равномерно) некоррелированных (независимых) чисел, полученных с помощью датчика.

Вычислим ДПФ:

$$\eta_e = \eta_e^{re} + i\eta_e^{im} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} h_k (\xi_k^{re} + i\xi_k^{im}) \exp\left(\frac{i2\pi kl}{N}\right) \quad (2)$$

Можно показать, что т.к.  $\xi_k^{re}$  и  $\xi_k^{im}$  имеют независимые равномерные распределения, то при больших  $N$  в силу центральной предельной теоремы  $\xi_k^{re}$  и  $\xi_k^{im}$  будут иметь распределения, близкие к гауссовому.

Найдем корреляционные функции:

$$C_{\eta}^{re}(l_1, l_2) = \left( \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} h_k (\xi_k^{re} \cos\left(\frac{2\pi kl_1}{N}\right) - \xi_k^{im} \sin\left(\frac{2\pi kl_1}{N}\right)) \right) \left( \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} h_n (\xi_n^{re} \cos\left(\frac{2\pi nl_2}{N}\right) - \xi_n^{im} \sin\left(\frac{2\pi nl_2}{N}\right)) \right)$$
$$= \left( \frac{\langle \xi^2 \rangle}{N} \right) \sum_{k=0}^{N-1} h_k^2 \cos\left[2\pi k \frac{l_1 - l_2}{N}\right],$$

где  $\langle \xi^2 \rangle = \langle (\xi^{re})^2 \rangle = \langle (\xi^{im})^2 \rangle$  а

$$\langle \xi_{k_1}^{re} \xi_{k_2}^{re} \rangle = \langle \xi_{k_1}^{re} \xi_{k_2}^{im} \rangle = \langle \xi_{k_1}^{im} \xi_{k_2}^{re} \rangle = 0 \text{ т.к.}$$

Числа исходной последовательности некоррелированы.

Аналогично можно показать, что

$$\langle \eta_{l_1}^{im} \eta_{l_2}^{im} \rangle = C_{\eta}^{im}(l_1, l_2) = C_{\eta}^{re}(l_1, l_2) \text{ и}$$

$$C_{\eta}^{re/im}(l_1, l_2) = \langle \eta_{l_1}^{re}, \eta_{l_2}^{im} \rangle = \frac{\langle \xi^2 \rangle}{N} \sum_{k=0}^{N-1} h_k^2 \sin\left[\frac{2\pi k(l_2 - l_1)}{N}\right]$$

Если выполнено условие симметрии  $h_k = h_{N-k}$ ,

то можно получить:

$$\langle \eta_{l_1}^{re}, \eta_{l_2}^{re} \rangle = \langle \eta_{l_1}^{im}, \eta_{l_2}^{im} \rangle = \left( \frac{\langle \xi^2 \rangle}{\sqrt{N}} \right) \left( \frac{1}{\sqrt{N}} \right) * \sum_{k=0}^{N-1} h_k^2 \exp[i2\pi ik(l_2 - l_1)],$$

$$\langle \eta_{l_1}^{re}, \eta_{l_2}^{im} \rangle = 0$$

Таким образом, получаем, что корреляционная функция вещественной и мнимой части преобразованной последовательности есть дискретная ПФ от набора коэффициентов  $\{h_k^2\}$ ;  $k=0, 1, 2, \dots, N-1$ . Таким образом:

.. ? ..

коэффициенты  $\{h_k\}$  следует выбирать как отсчеты требуемого энергетического спектра последовательности.

Итак, из двух последовательностей некоррелированных одинаково распределенных чисел получили две последовательности гауссовых чисел с заданным энергетическим спектром.

Почему гауссовые? В силу ЦПТ. Сходимость к гауссовому закону  $\sim \frac{1}{\sqrt{N}}$ . Чем больше  $N$ , тем больше степень приближения.

# Вычисление статистических характеристик случайных сигналов

1. Функции плотности распределения вероятностей:  $p(x)$

$p(x) = \frac{dP(x)}{dx}$ , где  $P(x)$  — функция распределения случайной величины  $x$ - вероятность того, что случайная величина не превышает значения  $X$ .  $p(x)$  -вероятность нахождения в интервале  $(x+dx)$ .

Дискретный аналог-гистограмма распределения значений  $h(m)$ - относительная доля отсчетов, имеющих значение  $m$ .

**Одномерная гистограмма** — описывает частоту появления значений отсчетов независимо от значений других отсчетов.

**Многомерная гистограмма** -описывает частоту совместного появления значений нескольких отсчетов сигналов.

$$h(\vec{m}) = \frac{1}{N} \sum_{k=0}^{N-1} \delta(\vec{m} - \vec{m}_k); \quad (1)$$

$\vec{m}_k$ -вектор значений компонент  $k$ -ого отсчета сигнала;  $\vec{m}$  — вектор значений компонент аргумента гистограммы  $\vec{m} = (m_0, m_1, \dots, m_{k-1})$

Определение средних значений величины  $m$ :

По определению:

$$\bar{m} = \sum_{m=0}^M mh(m) = \frac{1}{N} \sum_{k=0}^{N-1} m_k \quad (2)$$

Вычисление локального среднего

$$\bar{m}(r) = \frac{1}{2N_1+1} \sum_{k=N_1}^{N_1} m_{r-k} \quad (3)$$

по окрестности вблизи отсчета  $r$  размера  $2N_1+1$  элементов.

# Вычисление корреляционных функций и спектров

По определению, **корреляционная функция** сигнала

$$R_a(\xi) = \frac{1}{X} \int_x^{x+X} a(x)a^*(x+\xi)dx;$$

энергетический спектр

$$A(f) = \int_{-\infty}^{+\infty} R_a(\xi) \exp(i2\pi f \xi) d\xi.$$

Нетрудно получить дискретные аналоги этих соотношений:

$$R_a(n) = \frac{1}{N} \sum_{k=0}^{N-1} a_k a^*_{k+n}, \quad (1)$$

который отличается от свертки наличием знака "+", а не "-". Таким образом, можно использовать операции типа свертки.

Энергетический спектр

$$A(r) = \frac{1}{M} \sum_{k=0}^{M-1} R_a(k) \exp\left(-\frac{i2\pi kr}{M}\right), \quad (2)$$

т.е. дискретное преобразование Фурье от корреляционной функции.

# Основные принципы моделирования помех

В общем случае сигнал и шум аddитивны. Таким образом, зашумленный сигнал равен сумме  
 $u(t)=u_0(t)+\xi(t)$ ,

$u(t)$  - зашумленный сигнал,  $u_0(t)$  - неискаженный сигнал,  $\xi(t)$  - шум. Исходя из статистической независимости сигнала и шума, для **корреляционных функций** искаженного, идеального сигнала и шума, получаем:

$$R_u(n)=R_{u_0}(n)+R_\xi(n) \quad (1)$$

# Измерение параметров случайных помех

Для выполнения операции фильтрации искаженного сигнала необходимо определить характеристики шума, не отделяя его от сигнала. Выход заключается в том, чтобы разделить характеристики сигнала и шума на основе измерения характеристик зашумленного сигнала.

Основное предположение — предположение о гладкости характеристик неискаженного сигнала.

## 1. Оценка параметров аддитивного не зависящего от сигнала флюктуационного широкополосного шума

а) Шум, как правило, некоррелирован или слабокоррелирован. Тогда:

$$C_H(r,s) = C_0(r,s) + C_\xi(r,s), \quad (1)$$

где

- $C_H$  — корреляционная функция зашумленного сигнала,
- $C_0$  — корреляционная функция незашумленного сигнала,
- $C_\xi$  — корреляционная функция шума.

Это следует из аддитивности и независимости сигнала и шума.

$$C_\xi(r,s) = C_H(r,s) - C_0(r,s).$$

Так как шум слабокоррелирован, то

$$C_\xi(r,s) = \sigma_\xi^2 \delta(r,s)$$

$\sigma_\xi^2$  — дисперсия шума,  $\delta$  — *символ Кронекера*. Таким образом,  $C_H$  отличается от  $C_0$  только в начале координат или вблизи небольшой окрестности начала координат.

$$\sigma_\xi^2 = C_H(0,0) - C_0(0,0) \quad (2)$$

В остальных точках  $C_H(r,s) = C_0(r,s)$ .

Как правило,  $C_0(r,s)$  — гладкая функция, поэтому ее можно с высокой степенью точности экстраполировать в начало координат. Поэтому для определения  $\sigma_\xi^2$  достаточно:

- измерить  $C_H(r,s)$  в окрестности  $(0,0)$
- найти экстраполяцию  $C_0(0,0)$  величины  $C_0(0,0)$
- оценить  $\sigma_\xi^2 = C_H(0,0) - C_0(0,0)$ .

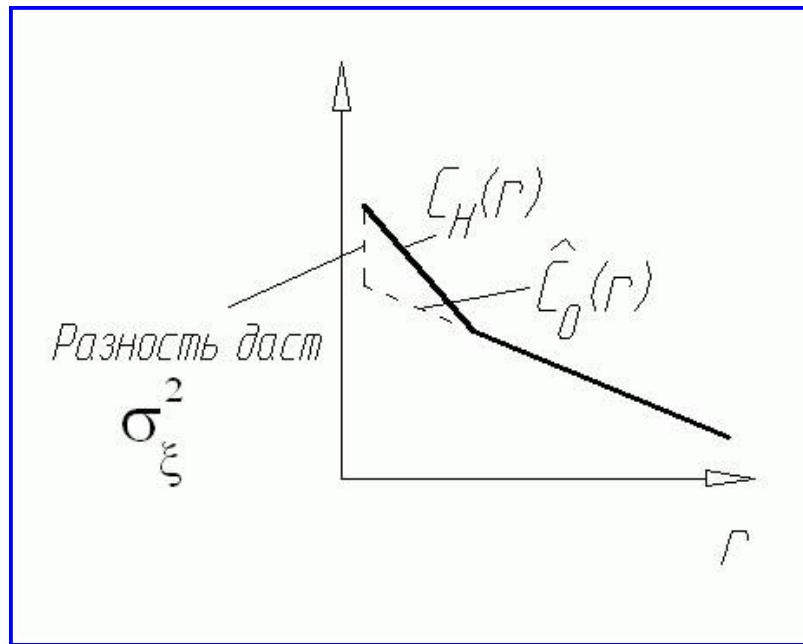


Рис. 1.1

## 2. Оценка интенсивности и частоты гармонических составляющих периодических и других помех с узким спектральным составом

Периодические (муаровые) помехи имеют место в теле- и фототелесистемах (следствие дискретизации). В базисе Фурье они имеют несколько компонент, заметно отличных от нуля.

Пример:

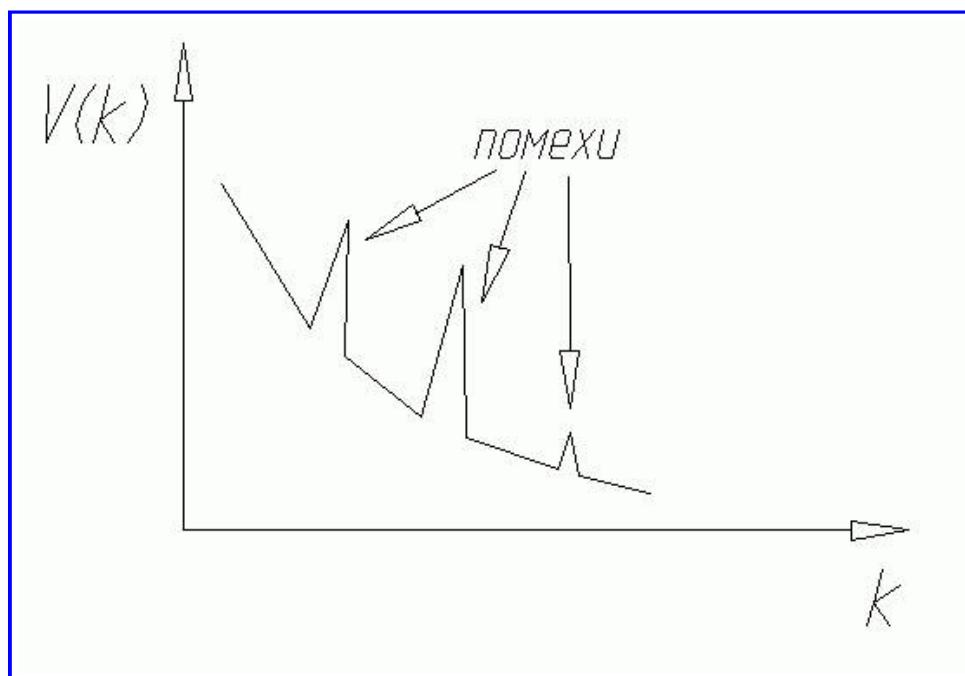


Рис. 2. 1

Спектр сигнала, как правило, гладкий и монотонный. Наличие узкополосного шума проявляется в виде аномально больших и локализованных отклонений в спектре искаженного сигнала. Расположение этих выбросов корреляционной функции обычно неизвестно.

**Локализация** — метод предсказания или "голосования".

Принцип: находим среднее значение квадрата модуля спектральных компонент зашумленного сигнала  $\langle |\beta_{r,s}|^2 \rangle$  с использованием БПФ.

**Метод "голосования"** : каждый элемент анализируемой последовательности рассматривается одновременно с некоторым количеством 2<sup>п</sup>его ближайших соседей. Эта выборка упорядочивается по возрастанию и проверяется, не попало ли значение данного элемента в число  $k$  крайних значений. При положительном ответе принимается решение о наличии аномально большого (малого) значения в данном элементе. Величины  $n$  и  $k$  заданы априори.

Тогда из аддитивности сигнала и шума

$$\langle |\chi_{r,s}|^2 \rangle = \langle |\beta_{r,s}|^2 \rangle - \langle |\alpha_{r,s}|^2 \rangle,$$

где  $\chi_{r,s}$  — шум,  $\alpha_{r,s}$  — чистый сигнал.

Величину  $\langle |\alpha_{r,s}|^2 \rangle$  получают интерполяцией по ближайшим точкам  $\langle |\beta_{r,s}|^2 \rangle$ , которые не затронуты искажением.

### 3.Оценка параметров импульсного шума

Осуществляется так же, как и для узкополосного, но в пространственной или временной области.

# Оптимальные линейные адаптивные фильтры

Одним из наиболее известных способов обработки сигналов является линейная фильтрация сигналов. Параметры требуемых фильтров находят, пользуясь принципами оптимальной (винеровской) фильтрации, разработанной для непрерывных сигналов и среднеквадратичного критерия качества фильтрации. При этом фильтры являются оптимальными в среднем и не являются адаптивными, поскольку алгоритмы обработки не будут зависеть от фоновой части сигнала. Однако, желательно, чтобы искомый сигнал был лучше не в среднем, а для данного конкретного вида. С этой точки зрения нужно описывать преобразования при фиксированной фоновой части, а не производить усреднение результата по случайнм параметрам фона.

Сформулируем критерий качества для адаптивных фильтров.

1. Считаем, что случайная природа сигналов определяется шумом датчика и случайнм фоном.
2. В качестве меры отличия результата обработки от "идеального" сигнала используем квадратичную меру.
  - $\mathbf{b} = \{b_k\}$  - N-мерный вектор отсчетов обрабатываемого сигнала;
  - $\mathbf{a} = \{a_k\}$  - N-мерный вектор отсчетов "идеального" сигнала;
  - $\hat{\mathbf{a}} = \{\hat{a}_k\}$  - N-мерный вектор отсчетов результата обработки.

Потребуем, чтобы величина

$$\overline{\langle |\varepsilon|^2 \rangle} = \overline{\sum_{k=0}^{N-1} |a_k - \hat{a}_k|^2}, \quad (1)$$

где  $\langle \rangle$  - усреднение по ансамблю реализаций изображения,  $\overline{\langle \rangle}$  - усреднение по ансамблю шума датчика, была минимальной. Этот критерий называется МСКО (минимальная среднеквадратичная ошибка).

При линейной фильтрации

$$\hat{a}_k = \sum_{n=0}^{N-1} h_{k-n} b_n. \quad (2)$$

Используя соотношение Парсеваля, критерий качества (1) можно выразить через отсчеты спектров сигналов -  $\{a_k\} \Leftrightarrow \{\hat{a}_k\}$  и  $\{\alpha_k\} \Leftrightarrow \{\hat{\alpha}_k\}$ , т.е.

$$\overline{\langle |\varepsilon|^2 \rangle} = \overline{\sum_{k=0}^{N-1} |a_k - \hat{a}_k|^2}. \quad (3)$$

(Соотношение Парсеваля:

$$\sum_{k=0}^{N-1} x_k^2 = \sum_{k=0}^{N-1} \chi_k^2,$$

где  $\chi_k$  - отсчеты разложения вектора  $\mathbf{x} = (x_0, \dots, x_N)$  по базису Фурье. Норма вектора не зависит от выбора ортогонального базиса.)

Подставим в (3) исходя из (2) значения спектра сигнала  $\hat{\alpha}_k$ :

$$\hat{\alpha}_k = \beta_k \cdot H_k, \quad (4)$$

где  $\beta_k$  - спектр сигнала на входе,  $H_k$  - спектр импульсного отклика.

Таким образом,

$$\overline{\langle |\varepsilon|^2 \rangle} = \overline{\sum_{k=0}^{N-1} |a_k - \beta_k \cdot H_k|^2}. \quad (5)$$

$$H_k = H_k^{\text{re}} + i \cdot H_k^{\text{im}}$$

Вычислим производные по вещественной и мнимой частям и приравняем нулю:

$$\begin{aligned} -\overline{\langle \alpha_k \beta_k^* \rangle} - \overline{\langle \alpha_k^* \beta_k^* \rangle} + 2 \overline{\langle H_k^{\text{re}} |\beta_k|^2 \rangle} &= 0 \\ i \overline{\langle \alpha_k \beta_k^* \rangle} - i \overline{\langle \alpha_k^* \beta_k \rangle} + 2 \overline{\langle H_k^{\text{im}} |\beta_k|^2 \rangle} &= 0 \end{aligned} \quad (6)$$

Или

$$2 \overline{\langle H_k |\beta_k|^2 \rangle} - 2 \overline{\langle \alpha_k \beta_k^* \rangle} = 0$$

Отсюда имеем

$$H_1 = \frac{\overline{\langle \alpha_k \beta_k^* \rangle}}{\overline{\langle H_k |\beta_k|^2 \rangle}} \quad (7)$$

$$\alpha_k = \sqrt{\langle |\beta_k|^2 \rangle}$$

Другим полезным *критерием оптимизации* параметров линейных фильтров является критерий минимума среднеквадратичной ошибки восстановления энергетического спектра сигнала (ВСС-критерий). Оптимальный по этому критерию фильтр определяется соотношением

$$H_k = \left[ \langle |\alpha_k|^2 \rangle / \langle |\beta_k|^2 \rangle \right]^{1/2} \quad (8)$$

Характеристики (7) и (8) зависят от спектров обрабатываемых изображений. Фильтры являются адаптивными.

Если ошибка усредняется по всему изображению, то фильтры глобально-адаптивные, если в пределах отдельных фрагментов - локально-адаптивные.

# Адаптивные линейные фильтры для подавления аддитивного независимого шума

Аддитивный шум:

$$\beta_r = \alpha_r + \varepsilon_r, \quad (1)$$

$\varepsilon_r$  - отсчеты спектра независимого шума.

Подставим в (7), получим:

$$H_k = \frac{\overline{|\beta_k|^2} - \overline{|\varepsilon_k|^2}}{\overline{|\beta_k|^2}}. \quad (2)$$

Это дискретный аналог классической формулы винеровской фильтрации.

С помощью такого фильтра оказывается возможной фильтрация помех, в энергетических спектрах которых лишь небольшое число компонент заметно отлично от нуля (периодическая помеха). Это сильнокоррелированный узкополосный шум. Из формулы (2) видно, что он пропускает те компоненты, где интенсивность шума равна 0, и ослабляет компоненты, где интенсивность шума велика.

Частный случай - режекторный фильтр:

$$H_k = \begin{cases} 1, & \overline{|\varepsilon_k|^2} = 0 \\ 0, & \overline{|\varepsilon_k|^2} = \overline{|\beta_k|^2} \end{cases}$$

Результаты использования близки к (2), но возможно появление периодических искажений в районе резких границ.

Другим примером **аддитивных помех** с сосредоточенным спектром, хорошо поддающихся линейной фильтрации, являются помехи типа полос и неравномерности фона. Их дискретный спектр Фурье сосредоточен в области очень низких пространственных частот.

Подавление этих помех производится с помощью цифровой фильтрации рекурсивными фильтрами по следующим формулам:

$$\hat{a}_k = \left[ - \sum_{m=-N}^N a_{k+m} \cdot \frac{1}{2N+1} + a_k + \bar{a} \right], \quad (3)$$

где  $\bar{a}$  - константа, равная половине максимального значения сигнала по всему кадру сигнала.

Для двумерного случая эта формула приобретает вид:

$$\hat{a}_{k,l}^{(1)} = \left[ a_{k,l} - \frac{1}{2N_1+1} \sum_{m=-N_1}^{N_1} a_{k+m,l} \right] + \bar{a};$$

$$\hat{a}_{k,l}^{(2)} = \left[ \hat{a}_{k,l}^{(1)} - \frac{1}{2N_2+1} \sum_{m=-N_2}^{N_2} \hat{a}_{k_1,l+m}^{(1)} \right] + \bar{a}.$$

Каждый из одномерных фильтров подавляет полосы в соответствующем направлении. При этом параметры  $N_1$  и  $N_2$  выбираются, сообразуясь с размером фона.

# Коррекция линейных искажений, вносимых линейными блоками систем преобразования сигнала

Примерами таких блоков могут быть оптическая система (дефокусировки и т.д.), атмосфера и т.д.

Пусть  $\mathbf{b}=\{b_k\}$  — вектор отсчетов искаженного сигнала,  $\beta=\{\beta_k\}$  — его спектр.

Соответственно  $\{a_k\}$ ,  $\{a_k^*\}$  — сигнал и спектр неискаженного сигнала,  $\{n_k\}$ ,  $\{\chi_k\}$  — аддитивный независимый шум и его спектр.

Линейная искажающая система в базисе Фурье задана диагональной матрицей  $\{\lambda_k\}$ . Найдем скалярный линейный фильтр, корректирующий искажения в такой системе  $\{H_k\}$ .

В данном случае

$$\beta_k = \lambda_k a_k + \alpha_k,$$

тогда из формулы  $H_k = \frac{\overline{a_k} \overline{\beta_k^*}}{\langle |\beta_k|^2 \rangle}$  (см. **Оптимальные линейные адаптивные фильтры**) получаем:

$$H_k = (\lambda_k^* \langle |\alpha_k|^2 \rangle + \langle a_k \rangle \bar{\alpha}_k) / \langle |\beta_k|^2 \rangle,$$

или, т.к.  $\bar{\alpha}_k = 0$ , то

$$H_k = \lambda_k^* \langle |\alpha_k|^2 \rangle / \langle |\beta_k|^2 \rangle. \quad (1)$$

Имея в виду связь между  $a_k$  и  $\beta_k$ , получаем:

$$H_k = \begin{cases} (\langle |\beta_k|^2 \rangle - \langle |\alpha_k|^2 \rangle) / \lambda_k \langle |\beta_k|^2 \rangle, & \lambda_k \neq 0 \\ 0, & \lambda_k = 0 \end{cases}$$

(для критерия МСКО).

Соответственно для критерия ВСС имеем:

$$H_k = \begin{cases} ((\langle |\beta_k|^2 \rangle - \langle |\alpha_k|^2 \rangle) / |\lambda_k|^2 \langle |\beta_k|^2 \rangle)^{1/2}, & \lambda_k \neq 0 \\ (\langle |\alpha_k|^2 \rangle / \langle |\beta_k|^2 \rangle)^{1/2}, & \lambda_k = 0 \end{cases}$$

При этом спектр неискаженного изображения  $\langle |\alpha_k|^2 \rangle$  при  $\lambda_k = 0$  должен быть известен априори, либо найден **интерполяцией** величины  $(\langle |\beta_k|^2 \rangle - \langle |\alpha_k|^2 \rangle) / |\lambda_k|^2$  по соседним точкам, как это делалось для диагностики помех с узким спектром.

Таким фильтром корректируют системы наблюдения через турбулентную атмосферу, системы с дефокусировкой (с гауссовой апертурой).

# Линейные фильтры для препарирования изображений (выделения деталей)

Для подчеркивания мелких деталей наиболее часто используется подавление **нижних и усиление верхних пространственных частот**.

Эти фильтры целесообразно трактовать как линейную фильтрацию полезного сигнала на фоне помех. При этом под полезным сигналом понимаются детали изображения под помехами - фоновая часть, мешающая интерпретации деталей.

Следуя методике (см. **Оптимальные линейные аддитивные фильтры** ), найдем характеристику фильтра, минимизирующую средний квадрат модуля ошибки между полезным сигналом и результатом фильтрации искаженного сигнала:

$$\overline{[\varepsilon]^2} = \overline{\sum_{k=0}^{N-1} [\alpha_k - H_k \beta_k]^2},$$

где  $\overline{\cdot}$  означают уже усреднение по таким случайным параметрам, как форма, ориентация, масштаб,  $[\cdot]$  - усреднение по случайным положениям отсчета на картинке сигнала. По аналогии с формулой

$$H_k = \frac{\overline{\alpha_k \beta_k^*}}{\overline{|\beta_k|^2}}$$
 можно получить:

$$H_k = [\overline{\alpha_k \beta_k^*}] / [\overline{|\beta_k|^2}] \quad (1)$$

Имея в виду, что  $\beta_k^* = \alpha_k^* = \xi_k^*$  - аддитивный шум (спектр), получим

$$H_k = (\overline{\alpha_k^2}) + \overline{\alpha_k \xi_k} / \overline{|\beta_k|^2}$$

$$\text{Так как } \alpha_r = \sum_{k=0}^{N-1} \alpha_k \phi_k(r), \text{ где } \{\alpha_k\} - \text{отсчеты каждого сигнала, то } \overline{\alpha_r} = \sum_{k=0}^{N-1} \overline{\alpha_k} \phi_k(r) = \overline{\alpha} \sum_{k=0}^{N-1} \phi_k(r), \text{ так}$$

как среднее не зависит от величины  $k$  (ориентирован произвольно).

Так как базис ортогонален, то сумма отлична от нуля на одной из базисных функций, ответственной за передачу постоянной составляющей сигнала. Эту постоянную учитывать не будем.

Таким образом

$$H_k = [\overline{\alpha_k^2}] / [\overline{|\beta_k|^2}] \quad (2)$$

или

$$H_k = [\overline{\alpha_k^2}] / (\overline{|\beta_k|^2} + \epsilon)$$

Это МСКО-фильтр.

Если использовать критерий восстановления энергетического спектра, то придем к фильтру ВСС:

$$H_k = (\overline{\alpha_k^2} / \overline{|\beta_k|^2})^{1/2} \quad (3)$$

Наконец, если задаться целью получить максимум отношения величины сигнала на искомом объекте в точке его локализации к среднеквадратичному значению сигнала на фоновом изображении, то придем к МОСШ-фильтру (максимальное отношение сигнал/шум):

$$H_k = \overline{\alpha_k^*} / \overline{|\beta_k|^2} \quad (4)$$

Кроме того, можно считать, что формула (2) несет простой физический смысл. Если нужно выделить крупномасштабные детали, следовательно, нужно подавить высокие частоты, если мелкомасштабные - низкие частоты.

# Итеративный алгоритм фильтрации импульсных помех

*Импульсные помехи* - объекты размером в 1 элемент изображения. Для обнаружения используется фильтр вида:

$$\hat{a}_k = b_k - \frac{1}{2N_1+1} \sum_{n=-N_1}^{N_1} b_{k-n}, \quad (1)$$

где отсчеты берутся в ближайшей окрестности сбоев, т.е. обычно  $N_1=2$ , и окрестность суммирования включает 5 элементов.

Если сигнал на выходе такого фильтра превышает по модулю некоторый порог  $\delta$ , то принимается решение о наличии выброса шума. Все обнаруженные таким образом искаженные отсчеты сигнала отмечаются, после чего они исправляются путем замены значениями, усредненными по их неотмеченным соседним отсчетам.

Отсчеты сигнала, используемые при предсказании в окрестности 3-х элементов и попадающие в апертуру фильтра (1), в свою очередь, могут быть искажены. Поэтому алгоритм фильтрации должен быть итеративным с понижением порога  $\delta$  в процессе итераций.

Каждая итерация осуществляется в 2 подхода:

- обнаружение выброса;
- исправление искажений.

Обычно бывает 3-4 итерации.

# Рекурсивный алгоритм фильтрации импульсных помех

Алгоритм (1 предыд. модуля) требует 2 прохода. Его можно упростить, если для вычисления локального среднего использовать не 4 отсчета, соседних с данным на растре, а только 2 уже обработанных отсчета, предшествующих данному. В результате получаем следующий *рекурсивный* алгоритм фильтрации:

$$\hat{a}_k = \begin{cases} b_k, & \text{если } |d| < \delta_1 \\ \bar{b}_k + \delta_2 \cdot \text{sign}(d), & \text{в противном случае,} \end{cases}$$

где  $d = b_k - \bar{b}_k$ ,  $\bar{b}_k = (b_{k-2} + b_{k-1})/2$ .

Порог  $\delta_2$  вводится, чтобы обеспечить устойчивость алгоритма, и выбирается равным 3-4% размаха значений сигнала.

# Обнаружение и фильтрация импульсных помех и сбоев

Действие импульсных помех сказывается не на всем изображении, а в случайно расположенных точках, где значение сигнала заменяется случайной величиной.

Сбои приводят к тому, что на отдельных участках сигнал пропадает, и вместо него — другой сигнал, определяемый причиной сбоя: царапина или повреждение фотоматериала, сбой канала передачи и т.д.

Алгоритм фильтрации включает 2 этапа:

1. обнаружение выбросов;
2. исправление обнаруженных искаженных отсчетов сигнала.

## Итеративный алгоритм фильтрации импульсных помех

Импульсные помехи — объекты размером в 1 элемент изображения. Для обнаружения используется фильтр вида:

$$\hat{a}_k = b_k - \frac{1}{2N_1+1} \sum_{n=-N_1}^{N_1} b_{k-n}, \quad (1)$$

где отсчеты берутся в ближайшей окрестности сбоев, т.е. обычно  $N_1=2$ , и окрестность суммирования включает 5 элементов.

Если сигнал на выходе такого фильтра превышает по модулю некоторый порог  $\delta$ , то принимается решение о наличии выброса шума. Все обнаруженные таким образом искаженные отсчеты сигнала отмечаются, после чего они исправляются путем замены значениями, усредненными по их неотмеченным соседним отсчетам.

Отсчеты сигнала, используемые при предсказании в окрестности 3-х элементов и попадающие в апертуру фильтра, в свою очередь, могут быть искажены. Поэтому алгоритм фильтрации должен быть итеративным с понижением порога  $\delta$  в процессе итераций.

Каждая итерация осуществляется в 2 подхода:

- в первый подход происходит обнаружение выброса;
- во второй — исправление искажений.

Обычно бывает 3-4 итерации.

## Рекурсивный алгоритм фильтрации импульсных помех

Алгоритм (1) требует 2 прохода. Его можно упростить, если для вычисления локального среднего использовать не 4 отсчета, соседних с данным на *растре*, а только 2 уже обработанных отсчета, предшествующих данному. В результате получаем следующий рекурсивный алгоритм фильтрации:

$$\hat{a}_k = \begin{cases} b_k, & \text{если } |d| < \delta_1 \\ \bar{b}_k + \delta_2 \cdot \text{sign}(d) & \text{в противном случае,} \end{cases}$$

где  $d = b_k - \bar{b}_k$ ,  $\bar{b}_k = (b_{k-2} + b_{k-1})/2$ .

Порог  $\delta_2$  вводится, чтобы обеспечить устойчивость алгоритма, и выбирается равным 3–4% размаха значений сигнала.

# Синтез линейных фильтров методом регуляризации Тихонова

Метод регуляризации Тихонова применяется в следующей постановке задачи синтеза:

Дано интегральное уравнение типа свертки:

$$u_2(x) = \int_{-\infty}^{+\infty} H(t-x)u_1(t)dt, \quad (1)$$

где  $u_1$  - сигнал на входе в фильтр,  $u_2$  - сигнал на выходе фильтра,  $H$  - сигнал импульсного отклика. Требуется по известным функциям  $u_1(x)$  и  $u_2(x)$  определить  $H(x)$ . Подобная постановка задачи имеет место в том случае, когда имеется возможность получения результатов искажения тестового сигнала, заранее известного исследователю. Синтезируется фильтр  $H$  и затем используется для восстановления сигналов, искаженных той же средой, что и тестовый сигнал.

Уравнение (1) можно свести к виду

$$u_2(x) = \int_{-\infty}^{+\infty} u_1(t-x)H(t)dt. \quad (2)$$

Казалось бы, решение этого уравнения найти просто, вычислив **БПФ** от обеих частей и найдя отношение

$$\tilde{H}(\omega) = \frac{\tilde{u}_2(\omega)}{\tilde{u}_1(\omega)};$$

где  $\tilde{H}$ ,  $\tilde{u}_1$  и  $\tilde{u}_2$  - Фурье-образы соответствующих функций.

Однако, задача (2) относится к классу так называемых некорректных задач, то есть задач, в которых малым возмущениям в исходных данных (то есть в функциях  $u_1$  и  $u_2$ ) соответствуют заметные отклонения в решениях.

Примером таких возмущений в численных расчетах могут быть всегда погрешности, связанные с конечностью разрядов сетки ЭВМ. Действительно, пусть  $\tilde{u}_1(\omega)$  имеет погрешность  $\varepsilon_1$ , а  $\tilde{u}_2(\omega)$  -  $\varepsilon_2$ . Поскольку функции  $u_1$  и  $u_2$  имеют спектр ограниченный, то при больших частотах, где  $\tilde{u}_1$  и  $\tilde{u}_2=0$ , оказываются погрешности  $\varepsilon_1$  и  $\varepsilon_2$ , что приводит к ошибкам в определении  $\tilde{H}(\omega)$ .

Теория и методы решения некорректно поставленных задач получили интенсивное развитие после основополагающей работы А.Н. Тихонова, вышедшей в 1943 г.

Пусть  $\delta$  - погрешность задания функции  $u_2$ , а  $E$  - погрешность задания  $u_1$ .

Сглаживающим (регуляризационным) функционалом уравнения (2) называется функционал вида

$$M^\alpha[H] = \|\hat{A}H - u_2\| + \alpha \|H\|^2, \quad (3)$$

где нормы

$$\|\hat{A}H - u_2\| = \int_{-\infty}^{+\infty} \left\{ \int_{-\infty}^{+\infty} u_1 E(x-t)H(t)dt - u_2(x) \right\}^2 dt,$$

$$\|H\|^2 = \int_{-\infty}^{+\infty} \{H^2(x) + (H'(x))^2\} dx,$$

$\alpha$  - параметр регуляризации.

Основной идеей метода регуляризации является сведение задачи нахождения решения уравнения (2) к нахождению минимума функционала (3).

Доказано, что задача нахождения  $\min\{M^\alpha[H]\}$  разрешима и имеет единственное решение  $H^\alpha(x)$  при любых  $\alpha > 0$ , непрерывных функций  $u_1$  и  $u_2$ . При этом полученное решение  $H^\alpha(x)$  сводится к точному решению  $\bar{H}(x)$  при  $\alpha \rightarrow 0$ .

Таким образом, решение задачи минимизации функционала тем точнее отвечает точному решению, чем меньше  $\alpha$ . В то же время нельзя уменьшать  $\alpha$  до нуля, так как решение перестает быть устойчивым.

## Выбор параметра регуляризации

Основан на принципе обобщенной невязки.

Обобщенная невязка:

$$\rho(\alpha) = \|A_\varepsilon H^{(\alpha)} - u_2\| - (\delta + \varepsilon \|H^{(\alpha)}\|^2). \quad (4)$$

Первое слагаемое (4) определяет меру отличия левой и правой частей уравнения

$$u_2(x) = \int_{-\infty}^{+\infty} u_1(t-x) H(t) dt \quad (5)$$

второе — максимальную ошибку, связанную с возмущением  $\delta$  и  $\varepsilon$ . Действительно,

$$\|\hat{A}_\varepsilon H - u_2\| \leq \|\hat{A}_\varepsilon - \hat{A}\| + \|u_\delta - u\| \leq \delta^2 + (\varepsilon \|H\|)^2.$$

В соответствии с принципом обобщенной невязки в качестве параметра регуляризации  $\alpha^*$  принимается корень уравнения

$$\rho(\alpha) = 0.$$

В этом случае  $H^{(\alpha)}(x)$  — решение уравнения (5).

Физический смысл: Все функции  $H^{(\alpha)}(x)$ , удовлетворяющие уравнению (5) с ошибкой, не превышающей  $\delta + \varepsilon \|H^{(\alpha)}\|^2$ , могут считаться приближенным решением уравнения (5).

В то же время доказано, что при  $\alpha = \alpha^*$  решение является устойчивым по отношению к малым возмущениям, то есть задача корректная.

### Реализация метода регуляризации для уравнения (5)

Дискретный аналог уравнения  $u_2(x) = \int_{-\infty}^{+\infty} u_1(t-x) H(t) dt$

$$\Delta x \sum_{j=0}^{n-1} u_2(k-j) H(j) = u_1(k), \quad (6)$$

где  $n$  — число отсчетов.

Конечно-разностная аппроксимация (6) имеет следующий вид:

$$M^\alpha[H] = \sum_{k=0}^{n-1} \left( \sum_{j=0}^{n-1} u_2(k-j) H(j) \right)^2 \Delta x + \alpha \sum_{k=0}^{n-1} \Delta x [H^2(k) + (H'(k))^2]. \quad (7)$$

В частотной области функционал (7) запишется в виде:

$$M^\alpha[H] = \frac{\Delta x}{n} \sum_{m=0}^{n-1} (|\tilde{u}_2(m)|^2 (\Delta x)^2 \tilde{H}(m) \tilde{H}^*(m) - 2\Delta x \tilde{u}_2(m) \tilde{u}_1(m) \tilde{H}(m) + |\tilde{u}_1(m)|^2 + \alpha(1 + \omega_m^2) H(m) H^*(m)), \quad (8)$$

где  $\omega_m = \frac{2\pi m}{T}$ ,  $T$  — размер сигнала,  $\tilde{u}_1, \tilde{u}_2$  и  $\tilde{H}$  — Фурье-образы.

Проварыровав функционал по  $H(m)$  и приравняв результат варыирования к нулю, получим решение уравнения (6):

$$H(k) = \frac{\Delta x}{n} \sum_{m=0}^{n-1} \frac{e^{-i2\pi km/n} \tilde{u}_2(m) \tilde{u}_1(m)}{|\tilde{u}_2(m)|^2 (\Delta x)^2 + \alpha(1 + (2\pi m/T)^2)} \quad (9)$$

Для нахождения параметра регуляризации  $\alpha$  используем принцип обобщенной невязки: ищем решение уравнения

$$\rho(\alpha) = \beta(\alpha) - ((\delta + \varepsilon \sqrt{\gamma(\alpha)})^2) = 0, \quad (10)$$

где:

$$\gamma(\alpha) = \frac{\Delta x}{n} \sum_{m=0}^{n-1} \frac{|\tilde{u}_2(m)|^2 (\Delta x)^2 |\tilde{u}_1(m)|^2 (1 + (\frac{\pi m}{T})^2)}{[|\tilde{u}_2(m)|^2 (\Delta x)^2 + \alpha(1 + (\frac{2\pi m}{T})^2)]^2} \quad (11)$$

$$\beta(\alpha) = \frac{\Delta x}{n} \sum_{m=0}^{n-1} \frac{\alpha^2 (1 + (\frac{2\pi m}{T})^2) |\tilde{u}_1(m)|^2}{[|\tilde{u}_2(m)|^2 (\Delta x)^2 + \alpha(1 + (\frac{2\pi m}{T})^2)]^2}.$$

Простой способ решения (10) может быть предложен исходя из монотонности функции  $\rho(\alpha)$ .



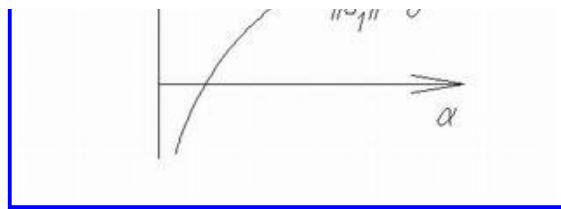


Рис. 1.

В качестве начального приближения взять достаточно большое  $\alpha$ , чтобы  $\rho(\alpha) > 0$ , и уменьшать  $\alpha$  с некоторым шагом, пока не выполнится неравенство:

$$|\rho(\alpha)| < \sigma, \quad \sigma - \text{малое}, \quad \sigma > 0.$$

# Нелинейные алгоритмы обработки сигналов

Основной класс нелинейных алгоритмов - *ранговые алгоритмы*, основанные на быстрых алгоритмах вычисления локальных гистограммных характеристик. Ранговые алгоритмы осуществляют нелинейное преобразование сигнала

$$b_k = \Phi_k(a_k),$$

причем  $\Phi_k$  - функция, вид которой определяется некоторым подмножеством так называемых рангов или ранговых выборок отсчетов, образованных данным отсчетом сигнала и отсчетами в заданной окрестности сигнала.

*Ранг* - это номер элемента в вариационном ряду, составленном из отсчетов в заданной окрестности. Он показывает, сколько элементов выборки имеет значение, меньшее, чем значение данного элемента.

Ранг называется локальным, если выборка образована фрагментом сигнала (окрестностью).

# Алгоритмы сглаживания сигналов

Двойкий смысл.

- При коррекции искажений **сигнала**, внесенных системой передачи, — это подавление помех, связанных с несовершенством системы (аддитивных, флуктуационных, импульсных и т.д.) При этом сглаживанию подвергается сигнал на выходе системы передачи.
- При препарировании изображений, т.е. при подчеркивании одних деталей и подавлении мешающих, это — устранение деталей (обычно малоразмерных), мешающих восприятию нужных (генерализация сигнала).

При этом сглаживание может применяться на любой стадии препарирования.

## Сглаживание для аддитивной модели шума

Цель **сглаживания сигнала** — сигнал, описываемый кусочно-постоянной моделью, т.е. "лоскутный" сигнал с пятнами — деталями, имеющими постоянное значение в пределах каждого пятна (сегментация сигнала). Для такого сигнала любая S-окрестность "идеального" кусочно-постоянного сигнала имеет гистограмму распределения вида:

$$h_s^0(q) = \sum_n H_n q_n q_{n+1}, \quad (1)$$

где  $q_n$  — значение сигнала с номером  $n$ , имеющим площадь  $H_n$ . Итак,  $h_s(a)$  — мультимодальная функция, состоящая из суммы локальных максимумов (кластеров).

Гистограмма распределения значений сигнала несглаженного изображения отличается от (1) тем, что в ней чистые меры, определяемые  $\delta$ -символом, размыты, т.е.

$$h_s(q) = \sum_n H_n \Delta_n(q - q_n).$$

$\Delta_n$  — унимодальная функция, характеризующая размытие моды.

Итак,  $h_s(q)$  — мультимодальная функция, в которой первичные моды  $\delta_{qq_n}$  проявляются в виде кластеров или локальных максимумов.

Сглаживание можно теперь определить как оценку параметров кластеров, к которым принадлежит заданный элемент. Для этого нужно определить границы кластеров.

Два способа:

1. Аддитивное квантование мод.
2. "Выращивание" кластеров.

## Аддитивное квантование мод

Анализируется локальная гистограмма, и в ней отыскиваются границы между локальными максимумами. Эти границы рассматриваются как границы интервалов квантования, и все значения сигнала, попавшие в тот или иной интервал, заменяются значением, равным положению максимума моды.

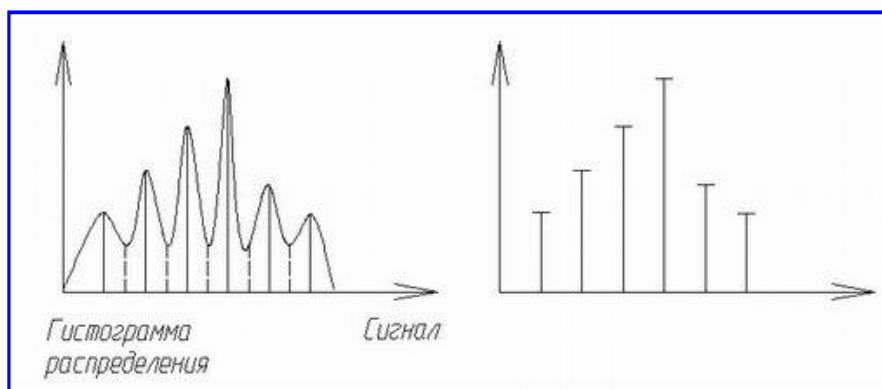


Рис. 1. 1

Качество зависит от того, насколько хорошо разделяются моды гистограммы. Поэтому аддитивное квантование целесообразно проводить по отдельным фрагментам, размер которых выбирается так, чтобы они включали небольшое число деталей сигнала (небольшое число мод в гистограмме).

Может оказаться, что выделяются моды, площадь которых невелика (разбросанные точки). Они разбивают границы деталей, образующих более мощные моды. Следовательно, нужно выполнять отбраковку мод по их мощности: если площадь моды меньше заданной величины, ее объединяют с соседней более мощной.

Недостатки: трудоемкая задача — определить границы кластеров при скользящей обработке, когда нужно принять решение о принадлежности к тому или иному кластеру одного центрального элемента.

### Метод "выращивания" кластеров

Кластер, к которому принадлежит элемент, определяется путем последовательных приближений.

Выбирается центр выравнивания — предварительная оценка положения моды. Около нее строится M-окрестность.

По M-окрестности выбирается следующее приближение к центру кластера и строится новая окрестность и т.д.

### Способы выбора M-окрестности и центрального элемента

Таблица 1

Центр выравнивания	KNV-окрестность	$\varepsilon_V$ -окрестность	$\varepsilon_r$ -окрестность
$V_K$	KNV( $V_K$ )	$\varepsilon_V(V_K)$	$\varepsilon_r(V_K)$
Среднее арифметическое по элементам M	KNV(MEAN(M))	$\varepsilon_V(MEAN(M))$	$\varepsilon_r(MEAN(M))$
Медиана по M	KNV(MED(M))	$\varepsilon_V(MED(M))$	$\varepsilon_r(MED(M))$
Срезка по M	KNV(CUT(M))	$\varepsilon_V(CUT(M))$	$\varepsilon_r(CUT(M))$

KNV-окрестность — окрестность по ближайшим K соседям  $V_K$  по величине

$$M_{KNV} = \{V_S(i) : \sum_{i=p}^{p+k} |V_i - V_S(i)| = \min\}$$

$V_S(i)$  — элементы сигнала в вариационном ряду

$\varepsilon_V$ -окрестность

$$M_{\varepsilon_V} = \{V_S(i) : V_1 - \varepsilon_V \leq V_S(i) \leq V_1 + \varepsilon_V\}$$

$\varepsilon_r$ -окрестность

$$M_{\varepsilon_r} = \{V_S(i) : r_S(V_1) - \varepsilon_r \leq i \leq r_S(V_1) + \varepsilon_r\}$$

$r_S(V_K)$  — ранг элемента с величиной сигнала  $V_K$  в вариационном ряду, построенном по S-окрестности

Медиана по M (MED(M)) — средний элемент в вариационном ряду из элементов окрестности M

$$MED(M) = V_M(r=(N_M+1)/2)$$

$N_M$  — число элементов в M-окрестности

Срезка по M (CUT(M))

$$CUT(M) = \begin{cases} V_P, & V_M(L) \leq V_P \leq V_M(R) \\ V_M(L), & V_P < V_M(L) \\ V_M(R), & V_P > V_M(R) \end{cases},$$

где  $V_M(R)$  ( $V_M(L)$ ) — значение элемента в вариационном ряду по M-окрестности с рангом R (L)

Среднее арифметическое по элементам M

$$MEAN(M) = \frac{1}{N_M} \sum_{k \in M} V_k$$

Выбор оценки центра выращивания определяется наличием априорной информации.

Выбор вида и размера окрестности определяется:

- KNV-окрестность позволяет учесть априорную информацию о размерах деталей; размер должен быть порядка площади деталей, которые нужно сохранить;
- $\varepsilon_V$ -окрестность позволяет учитывать априорную информацию о минимальных перепадах, которые нужно сохранить, или дисперсии шума, которую нужно подавить.
- $\varepsilon_r$ -окрестность — фильтрация импульсных помех и алгоритмы выделения границ.

S-окрестность равна двум минимальным размерам деталей, которые нужно сохранить. S-окрестность — заданное определенным образом множество элементов вокруг центрального.

Для оценки  $\hat{V}_K$  текущего элемента после определения границ кластера рекомендуют MEAN, MED или CUT по границам кластера.

1. Если шум гауссов, лучше выбирать MEAN.
2. Если у шума более "тяжелые хвосты", то лучше выбирать MED или CUT.

Некоторые из возможных алгоритмов:

$$\left. \begin{array}{l} \hat{V}_K = \text{MEAN}(\varepsilon_V(V_K)), \text{ где } \varepsilon_V = 10, S = 7*7 \\ \hat{V}_K = \text{MEAN}(KNV(MED(M))), \text{ где } S = 17*17, M = 3*3, K = 90 \end{array} \right\} \text{для обработки изображений} \quad (2)$$

$\varepsilon_V = 1,5\delta$ ,  $\delta$  — стандартное отклонение гауссова шума,  $K$  — количество ближайших соседей.

Для сглаживания речевых сигналов:

$$\hat{V}_K = \text{MEAN}(\varepsilon_V(\text{MED}(\text{KSN}(V_K)))) \quad (3)$$

KSN — определенная окрестность из  $K$  ближайших на растре соседей. Наиболее известной является медианная фильтрация:

$$V_K = \text{MED}(S),$$

где  $S$  — некоторая заданная определенным образом окрестность.

### Сглаживание для модели импульсных помех

Сглаживание импульсной помехи — 2 подхода:

- местонахождение искаженного элемента
- оценка сглаженного значения

Разметка — местонахождение, то есть проверка попадания ранга  $r_S(V_K)$  центрального элемента S-окрестности в  $\varepsilon_r$ -окрестность медианы

$$|r_S(V_K) - (N_S + 1)/2| < \varepsilon_r \quad (4)$$

(неравенство Вилкохсона).

Если неравенство (4) выполнено, то принимается решение об отсутствии помехи, в противном случае элемент ( $K$ ) искажен помехой.  $\varepsilon_r$  задается в зависимости от вероятности появления импульсных помех на элементе изображения.

Проверку гипотезы о наличии выброса в центральном элементе S-окрестности можно проводить и по значению элемента.

Пусть

$$\Delta = \varepsilon_V - |V_K - \text{SMT}(V_K)|, \quad (5)$$

где  $\text{SMT}(V_K)$  — сглаженное значение, полученное одним из алгоритмов сглаживания для аддитивной модели.

$\varepsilon_V$  зависит от разброса значений сигнала и распределения помехи.

Если  $\Delta < 0$ , то имеется помеха.

Второй этап — сглаживание помехи по окрестности отмеченных элементов. При этом из окрестности удалены отмеченные элементы.

Два типа алгоритмов:

$$\hat{V}_K = \begin{cases} V_K, \varepsilon_r \geq |r_S(V_K) - (N_S + 1)/2| \\ \text{SMT}(M), \varepsilon_r < |r_S(V_K) - (N_S + 1)/2| \end{cases} \quad (6)$$

$$\hat{V}_K = \begin{cases} V_K, \varepsilon_V \geq |V_K - \text{SMT}(V_K)| \\ \text{SMT}(M), \varepsilon_V < |V_K - \text{SMT}(V_K)| \end{cases} \quad (7)$$

$\text{SMT}(M)$  — сглаживание по окрестности, из которой исключены точки, подлежащие исправлению.

Часто используют  $S:3*3$ ,  $\text{SMT}(M) = \text{MED}(S)$ .

Основные ошибки:

— ошибки гладкости обработка (изменение искаженного изображения в результате сглаживания)

- ошибки ложного обнаружения (приводят к нежелательному сглаживанию деталей обнаружения)  $\Rightarrow$  нужно увеличить пороги;
  - ошибки пропуска (несглаженные выбросы помех)  $\Rightarrow$  нужно уменьшить пороги.
- Сглаживание итеративно, начинается с больших порогов и по мере удаления больших выбросов порог понижается на каждом шаге итерации.

# Увеличение детальности изображения

Увеличение **детальности изображения**- операция, противоположная сглаживанию (повышение локальных контрастов)

1. Наиболее известный метод определения и усиления различий-нерезкая маска.

$$\hat{V}_k = g(V_k - \bar{V}_k) + \bar{V}_k, \quad (1)$$

где  $\bar{V}_k$  сумма элементов в S-окр , взятых с некоторыми весами (среднее). g- коэффициент усиления. Это линейный алгоритм.

Ранговое расширение:

$$\hat{V}_k = gd_k + SMT(H(M)) \quad (2)$$

где  $d_k = V_k - SMT(H(M))$

Преимущество- адаптивность и меньшая пространственная инерционность.

В качестве сглаживания используется:

$$SMT(H(M)) = MED(KNV(S_k))$$

2. Скользящая эквализация гистограмм

При этом сигнал подвергается поэлементному нелинейному преобразованию  $F(V_k)$  , крутизна которого в точке  $V_k$  пропорциональна значению локальной гистограммы в точке  $q=V_k$  , т.е.

$$\frac{\Delta F(V_k)}{\Delta V_k} = \frac{h_s(q)}{q_2 V_k} \quad (3)$$

Решив это разностное уравнение, получим

$$\hat{V}_k = F(V_k) = \sum_{q=0}^{V_k} h_s(q) \quad (4)$$

откуда вытекает, что

$V_k = r_s(V_k)$  , т.е. ранг в вариационном ряду, построенная по S-окрестности.

Степенная интенсификация — более общий случай, когда

$$\Delta F(V_k) = [h_s(q)]^p |_{q=V_k} \quad (5)$$

Дает хорошие результаты при  $0 < p < 1$  в медицине.

Наконец, наиболее общий случай, когда крутизна пропорциональна

$$\frac{\Delta F(V_k)}{\Delta V_k} = f(h_s(q))$$

f-эквализация. Линейная эквализация

$$f(h_s(q)) = g_1 h_s(q) + q_2 \quad (6)$$

В этом случае

$$\hat{V}_k = g_1 r_s(V_k) + g_2 V_k \quad (7)$$

$g_1$  и  $g_2$  можно выбирать, например  $g_1=g_2=0.5$ .

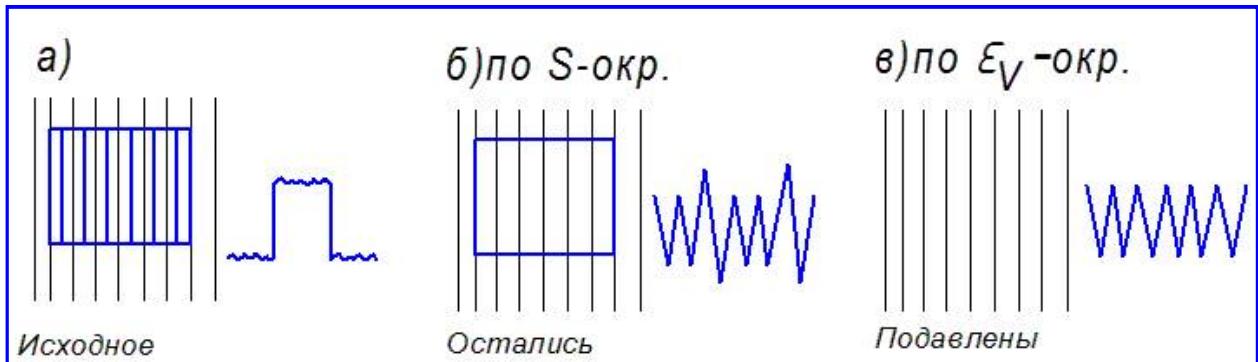
В качестве окрестности эквализации можно выбирать не S-окрестность, а  $\mathcal{E}_V$  или KNV.

При этом получим

$$\hat{V}_k = r_{\mathcal{E}_V}(V_k) \quad (8)$$

$$\hat{V}_k = r_{KNV}(V_k) \quad (9)$$

Эквализация  $\mathcal{E}_V$  или KNV окрестность позволяет избежать деталей, принадлежащих другому кластеру.



изображение

границы  
квадрата

границы  
квадрата

Рис. 1.

Выбор  $\Sigma_V$  и  $KNV$  окрестностей определяется информацией о разбросе ( $\Sigma_V$ ) значений сигнала или размерах ( $KNV$ ) деталей изображения.

Другая возможность-замена  $V_k$  некоторой функцией [ранга](#), т.е.

$$\hat{V}_k = f(r_s(V_k)) \quad (10)$$

а) Гиперболизация гистограмм

$$\hat{V}_k = \exp(r_s(V_k)) \quad (11)$$

при этом  $\ln(\hat{V}_k)$  будет иметь равномерное распределение.

б) Степенная интенсификация при  $p=0$  имеем:

$$\hat{V}_k = V_k - \text{MIN}(S) = V_k - V_S(r=1) \quad (12)$$

Алгоритм линейной коррекции [сигнала](#)

если не  $S$  а  $M_{KNV}$ -окрестность, то

$$\hat{V}_k = V_k - \text{MIN}(KNV(V_k)) \text{ или} \quad (13)$$

$$\hat{V}_k = V_k - \text{MIN}(\Sigma_V(V_k)) -$$

не имеет смысла, т.к.  $\text{MIN}(\Sigma_V) = V_k - \Sigma_V$  и  $\hat{V}_k = \Sigma_V$ - не зависит от  $V_k$

Следующая возможность: использование в вышеприведенных формулах вместо  $V_k$  сглаженных оценок SMT(H(M)). Получаем следующую группу алгоритмов:

$$\hat{V}_k = r_s(SMT(H(M))) \quad (14)$$

$$\hat{V}_k = g_1 r_s(SMT(H(M))) + g_2 SMT(H(M)) \quad (15)$$

$$\hat{V}_k = V_k - \text{MIN}(KNV(SMT(H(M)))) \quad (16)$$

$$\hat{V}_k = V_k - \text{MIN}(\Sigma_V(SMT(H(M)))) \quad (17)$$

Описанные выше алгоритмы повышения детальности соответствуют представлению о деталях, аддитивно наложенных на фон.

Лучше соответствуют понятию деталей и фона, представление о деталях, как бы врезанных в фон. Увеличению контраста деталей должно предшествовать их обнаружение.

## Пример 1

$$V_k = \begin{cases} SMT(H(M)), |d_k| \leq \Sigma_V \\ gd_k + SMT(H(M)), |d_k| > \Sigma_V \end{cases}$$

по аналогии с фильтрацией [импульсных помех](#). При такой процедуре которая пока выделяет все детали, нужно провести последующее сглаживание, чтобы устранить ненужные детали.

В зависимости от конечной цели -2 алгоритма:

1. выделение деталей с сохранением фона

2. Отделение деталей от фона

Алгоритмы- сочетание алгоритма повышения контраста в  $S_c$  окрестности и сглаживания в  $S_s$ -окрестности. При этом  $S_s$  окрестность составляет часть  $S_c$  окрестности. Получим следующие алгоритмы:

1. Выделение деталей с сохранением фона

$$\hat{V}_k = g_1 SMT(S_s(f_{S_c} r_{S_c}(SMT(S_c)))) + g_2 SMT(S_c) \quad (18)$$

$$\hat{V}_k = SMT(S_s(V_k - \text{MIN}(KNV(SMT(S_c))))) \quad (19)$$

$$\hat{V}_k = SMT(S_s(gd_k + SMT(S_c))) \quad (20)$$

2. Процедуры с отделением деталей от фона

$$\hat{V}_k = SMT(S_s(f(r_{S_c}(SMT(S_c))))) \quad (21)$$

$$\hat{V}_k = SMT(S_s(d_k)) = SMT(S_s(V_k - SMT(S_c))) \quad (22)$$

алгоритм эквалайзации-сглаживание

Некоторые характерные случаи общих алгоритмов:

$$\hat{V}_k = \text{MED}(\text{KNV}(\text{MED}(M)))$$

здесь  $S=15 \times 15$

Алгоритм Вилкоксона

$$\hat{V}_k = \text{MEAN}(S_s(r_{S_c}(V_k)))$$

Алгоритм Тамуры при  $f(t)=r^t$

$$\hat{V}_k = \text{MEAN}(S_S(r_{S_c}^t(V_k)))$$

Медианный критерий:  $f(t)=\text{sign}(r_{S_c}(V_k) - \frac{(N_{S_c}+1)}{2})$

$$\hat{V}_k = \text{MEAN}(S_s(\text{sign}(r_{S_c}(V_k) - \frac{(N_{S_c}+1)}{2}))$$

# Обнаружение деталей и границ

Принцип обнаружения границ (деталей) базируется на определении степени статистического несоответствия распределения значений элементов анализируемой окрестности заданному распределению значений сигнала вблизи деталей. При этом размер окрестности выбирают порядка размеров деталей, которые нужно обнаруживать, или, соответственно, порядка размеров окрестности границ деталей.

Само по себе обнаружение состоит в сравнении измеренной степени соответствия с порогом.

Часто при препарировании имеет смысл также предъявить для визуализации и саму величину несоответствия, а не только результат сравнения с порогом. При этом обнаружение осуществляет оператор.

## Виды ранговых алгоритмов обнаружения границ

1. Для кусочно-постоянной модели признаком границы служит двухмодовость локальной гистограммы. Критерий двухмодовости- характеристики разброса-квазиразмах.

$$QDISP(S)=V_s\left[\frac{(N_s+1)}{2}+\varepsilon_r\right]-V_s\left[\frac{(N_s+1)}{2}-\varepsilon_r\right] \quad (1)$$

$N_s$ -медиана выборки по S-окрестности

2. Для более тонкого выделения границ рекомендуется алгоритм вида:

$$\Delta \hat{V}_k = \max(\varepsilon_r(SMTM(M))) - \min(\varepsilon_r(SMTM(M))) \quad (2)$$

SMTM(M)-сглаживание по M-окрестности

3. Алгоритм вычисления локальной **дисперсии**:

$$\hat{V}_{ke} = (\text{MEAN}(V_{ke} - \text{MEAN}(S))^2)^{1/2} \quad (3)$$

Недостаток этого алгоритма- инерционность, что приводит к размазыванию деталей и границ.

1,2- в этих нелинейных алгоритмах  $\varepsilon_r=\sqrt{N_s}$

# Стандартизация изображений

Метод обработки, основанный на использовании известной гистограммы распределения идеального изображения.

**Стандартизация гистограмм**- преобразование сигнала, делающее гистограмму распределения его значений заданной. Затем полученное модельное преобразование распространяют на сигнал, который необходимо обработать.

**Пример алгоритма:**

$$\hat{V}_k = V_{ST}(r_s(V_k))$$

$V_{ST}$  ( $r$ )-значение  $r$ -ого по рангу элемента вариационного ряда, построенного по заданной гистограмме.

Стандартизация — 2-х видов: локальная и глобальная.

Стандартизация среднего арифметического и дисперсии:

$$\hat{V}_k = \frac{g_1 \sigma}{g_1 V_k + \sigma} \{V_k - \text{MEAN}(S)\} + g_2 m + (1-g_2) * \text{MEAN}(S)$$

$\sigma, m$  — эталонные среднеквадратичное отклонение и среднее арифметическое;  $g_1, g_2$ — константы.

# Введение в параллельные методы цифровой обработки сигналов

Использование достижений СБИС- технологий позволило подойти к задаче конструирования алгоритмов с иной стороны, а именно: возродить старые принципы реализации, которые были более трудоемки с точки зрения обычных ЭВМ, и отказаться от вычисления с помощью **БПФ**.

Недостатки вычисления с помощью БПФ:

1. необходимо дополнять нулями сигнал и импульсный отклик.
2. Жестко заданы отсчеты спектра. Отсчеты спектра и отсчеты сигнала связаны соотношениями:

$$\Delta t \cdot \Delta f = \frac{1}{N}, \text{ где } N \text{ общее число отсчетов.}$$

Это не всегда удобно при спектральном анализе.

# Параллельные методы вычисления свертки (аппаратно-программные)

1. Параллельное вычисление по группам элементов изображений.

$$U_{ij}^{(\text{вых})} = \sum_{m=-M}^M \sum_{n=-N}^N H_{mn} U_{i-m, j-n}^{(\text{вх})} \quad (1)$$

$H_{mn}$  - отсчеты импульсного отклика

$m = -\bar{M}, n = \bar{N}$  (малоформатный)

$U_{ij}^{(\text{вх})}$  - отсчеты входного сигнала  $i = \bar{I}, j = \bar{J}$  — крупноформатный объект.

а) Представим (1) в виде:

$$U_{ij}^{(\text{вых})} = \sum_{m=-M}^M S_{ijm} \quad (2)$$

где  $S_{ijm} = \sum_{n=-N}^N U_{i-m, j-n}^{(\text{вх})} H_{mn}$ , т.е. виде суммы одномерных строк сверток, строк изображения и соответствующих строк ядра свертки.

$S_{ijm}$  можно вычислять параллельно. Нужно иметь  $(2M+1)$  процессоров одномерной свертки и на выходе  $2M+1$  — входовой сумматор.

Необходимо такие буферные ЗУ, обеспечивающие хранение  $(2M+1)$  текущих строк изображения.

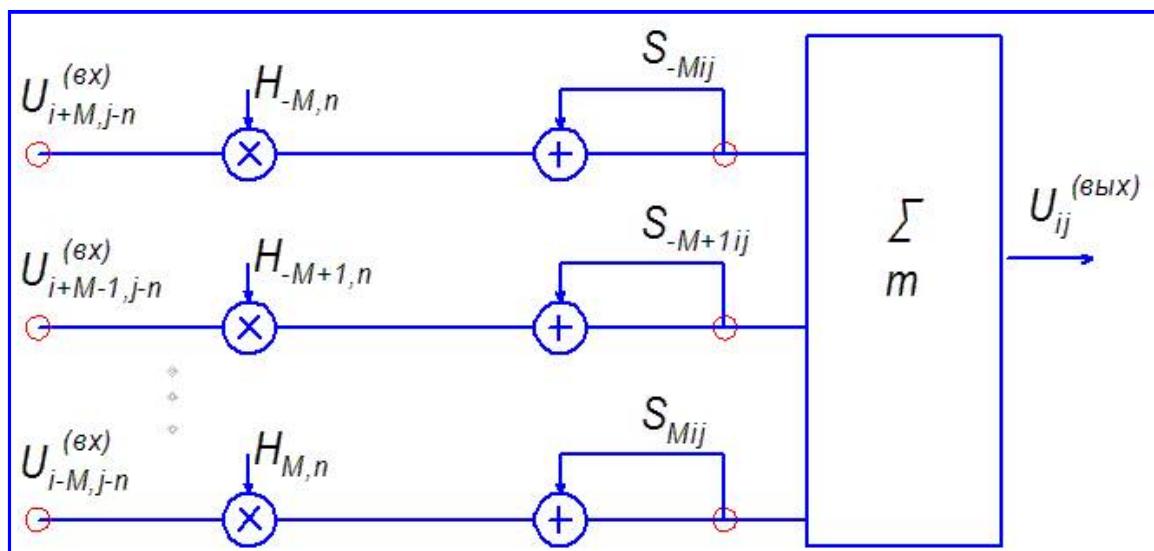
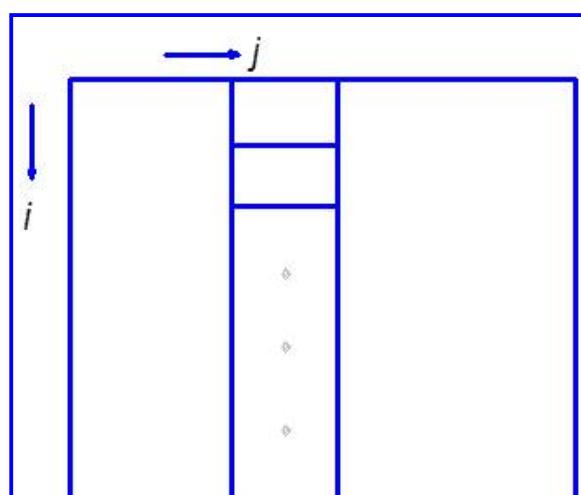


Рис. 1. Схема алгоритма

$(2M+1)$  умножителей и  $(2M+1)$  сумматоров



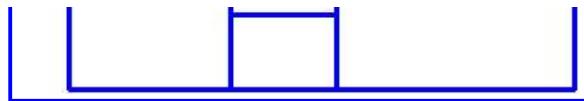


Рис. 2.

на каждом шаге суммирования в операции участвует один столбец изображения.

Суммирование выполняется по направлению по строкам.

Общее время вычислений:  $(2N+1)t + (2N+1)t + (2M+1)t$

б) Более быстродействующая схема, если есть  $L = (2M+1)(2N+1)$  умножителей и 1 сумматор на  $L$  входов:

В этом случае алгоритм вычисления можно представить в виде:

$$\begin{aligned} U_{i-m, j-n}^{(bx)} &= H_{mn} \quad \forall m, n \\ S := \sum_{m=-M}^M \sum_{n=-N}^N R_{mn} & \\ U_{ij}^{(вых)} &:= S \end{aligned} \quad (3)$$

Нетрудно определить, что потребуется всего  $T = tx + Lt +$  операций.

Поскольку время умножения  $tx < lt$ , то производительность будет больше.

в) Для увеличения быстродействия можно организовать параллельную работу умножителей и сумматора в конвейерном режиме.

Общий вид:

$$\forall i = \overline{1, I}$$

$$\forall j = \overline{1, J}$$

$$\left\{ \begin{array}{l} R_{mn} := U_{i-m, j-n}^{(bx)} H_{mn} \\ S := \sum_{m=-M}^M \sum_{n=-N}^N R_{mn} \end{array} \right.$$

$$R_{mn} := R_{mn}$$

$$U_{ij}^{(вых)} := S$$

Структура конвейерного процессора двумерной свертки с ядром размером 3x3 (импульсным откликом

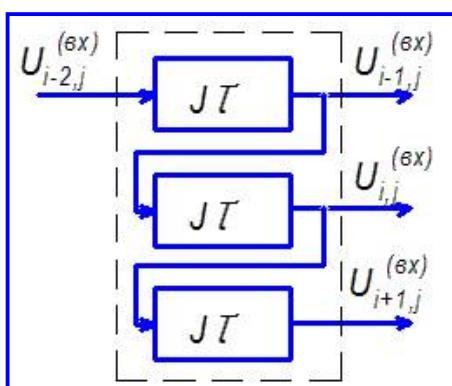
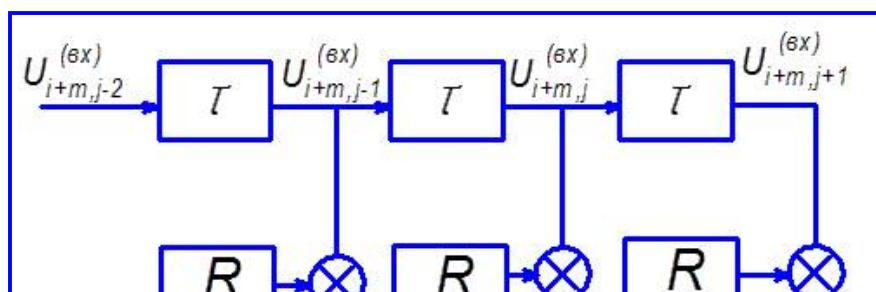


Рис. 3. Линия задержки



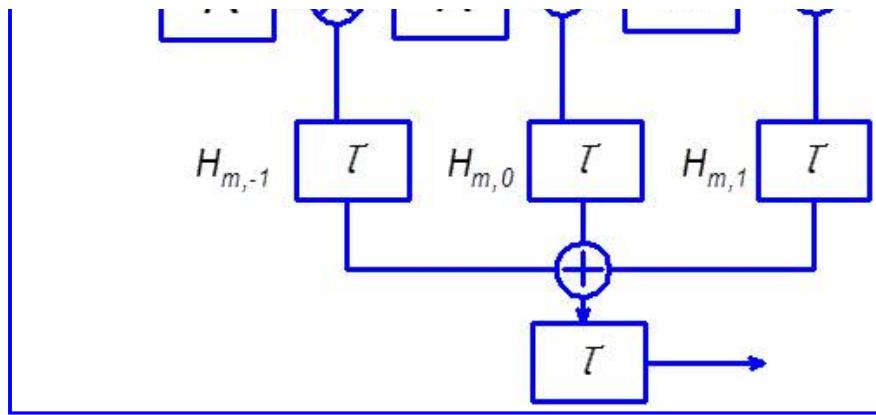


Рис. 4. Процессорный элемент

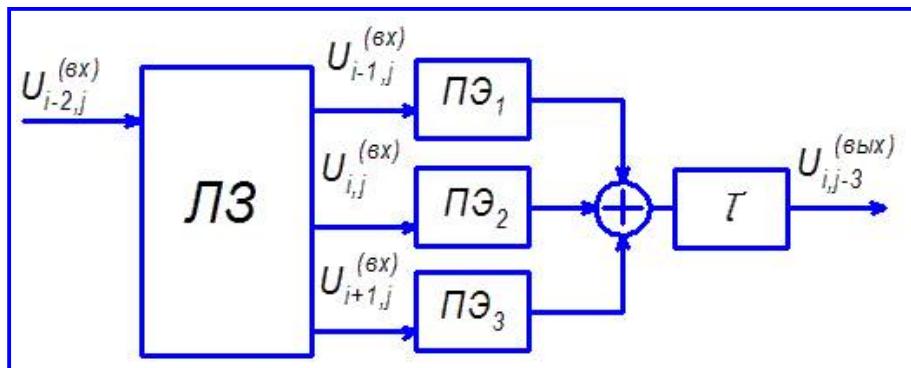


Рис. 5. Процессор свертки

После каждого умножителя и 2-х входового сумматора нужно использовать конвейерные регистры.

После начальной загрузки весовых коэффициентов  $H_{mn}$ , загрузки сдвигающих и конвейерных регистров в каждом такте вводится один элемент изображения и выводится один элемент свертки.

Структура древовидного процессора была использована в процессоре **ЦДС** с импульсным откликом 3x3 для систем технического зрения роботов и работала в темпе телевизионной развертки. Изображение размером 256x256 обрабатывалось за 16,7 мс.

# Параллельное вычисление по группам разрядов (разрядным срезам) изображения

Рассмотрим одномерную свертку

$$Y_i = \sum_{m=-M}^M H_m X_{i-m} \quad (1)$$

При этом будем для простоты считать, что числа сигнала неотрицательны, Q-разрядные.

Тогда можно представить

$$H_m = \sum_{q=0}^{Q-1} a_{mq} 2^q, a_{mq} = 0,1 \quad (2)$$

$$X_i = \sum_{r=0}^{Q-1} X_{ir} 2^r, X_{ir} = 0,1$$

Подставив (2) в (1) получим:

$$Y_i = \sum_{m=-M}^M \sum_{q=0}^{Q-1} a_{mq} 2^q \sum_{r=0}^{Q-1} X_{i-m,r} 2^r$$

или

$$Y_i = \sum_{r=0}^{Q-1} \sum_{m=-M}^M \sum_{q=0}^{Q-1} a_{mq} X_{i-m,r} 2^{q+r} \quad (3)$$

## Метод распределенной арифметики

$$Y_i = \sum_{r=0}^{Q-1} \sum_{q=0}^{Q-1} 2^{q+r} \sum_{m=-M}^M a_{mq} X_{i-m,r} \quad (4)$$

Выполним суммирование по q

$$Y_i = \sum_{r=0}^{Q-1} 2^r \sum_{m=-M}^M H_m X_{i-m,r} \quad (5)$$

Каждая сумма по m есть свертка последовательности 2M+1 весовых коэффициентов  $H_m$  и последовательности бит, сформированных из r-х разрядов 2M+1 отсчетов сигнала  $X_{i-m}$  (т.е. r-м разрядным срезом отсчетов). Индекс r определяет разрезной срез и значение коэффициентов  $2^r$ . Умножение на  $2^r$  — сдвиг частичного результата на r разрядов.

а) Разновидности метода-с записью предварительно вычисленных частичных результатов в ЗУ — т.е. сумм по m для каждого r. (Значения промежуточных результатов выбираются из ЗУ по адресу, определяемому значениями разрядов  $X_{i-m,r}$ )

Таким образом:

$$Y_i = \sum_{r=0}^{Q-1} 2^r f(X_{i+M,r}, X_{i+M-1,r} \dots X_{i-M,r}) \quad (6)$$

где

$$f(z_M, z_{M-1}, \dots z_{-M}) = \sum_{m=-M}^M H_m z_m$$

$$z_m = 0,1$$

f-табличная функция 2M+1 двоичных переменных. Объем ЗУ для хранения f -2M+1 ячеек. При параллельной выборке частичных результатов из ЗУ производительность увеличивается в Q раз.

Если совокупность 2M+1 отсчетов разделена на S блоков (каждый блок имеет  $P = \frac{2M+1}{S}$  отсчетов) и выполнить вычисления (6) отдельно для каждого блока, то (6) преобразуется:

$$\begin{aligned} Y_i &= \sum_{r=0}^{Q-1} 2^r \left\{ \sum_{p=0}^{P-1} H_{-M+p} X_{i+M-p,r} + \sum_{p=0}^{P-1} H_{-M+P+p} X_{i+M-P-p,r} + \dots \right\} = \\ &= \sum_{r=0}^{Q-1} 2^r \sum_{k=0}^S \sum_{p=0}^{P-1} H_{-M+kP+p} X_{i+M-kP-p,r} = \\ &= \sum_{r=0}^{Q-1} 2^r f_k(X_{i+M-kP}, X_{i+M-(k+1)P}, \dots X_{i+(M-1)P}) \end{aligned} \quad (7)$$

$$\text{где } f_k(z_0, z_1, \dots, z_{p-1}) = \sum_{p=0}^M H_{-M+kP+p} z_p \quad z=(0,1)$$

В этом случае используется  $S$  блоков ЗУ, общий объем составит  $S2^P < 2^{2M+1}$  ячеек. Во всех блоках - значение частичных результатов, которые нужно просуммировать.

**Последовательно-параллельные вычисления** в соответствии с (6) выполнены в элементе (PIPE) для импульсного отклика 3x3 или 1x9. Одноименные разряды 9 элементов (разрядный срез) с выходов регистров образуют 9-разрядный адрес, по которому из ЗУ частичных результатов считывается 12-разрядный код суммы весовых коэффициентов. Последовательно считываемые коды суммируются в сдвигателе с соответствующим сдвигом и 20-разрядный результат → в *регистр*.

Максимальная частота, формирующая результат-1,28 МГц для 8-разрядных элементов.

#### **Другая модификация. Разряды каждого из отсчетов разбиваются на группы по L разрядов**

$$\begin{aligned} Y_i &= \sum_{r=0}^{L-1} \sum_{R=0}^{Q/L-1} 2^{r+kl} \sum_{m=-M}^M H_m X_{i-m, r+kL} = \\ &= \sum_{K=0}^{Q/L-1} 2^{KL} \sum_{r=0}^{L-1} 2^r \sum_{m=-M}^M H_m X_{i-m, r+kL} = \\ &= \sum_{k=0}^{Q/L-1} 2^{KL} f_k \begin{pmatrix} X_{i+M, KL} & X_{i+M, 1+KL}, \dots, X_{i+M, (K+1)L-1} \\ X_{i+M-1, KL} & X_{i+M-1, 1+KL}, \dots, X_{i+M-1, (K+1)L-1} \\ X_{i-M-1, KL} & X_{i-M-1, 1+KL}, \dots, X_{i-M, (K+1)L-1} \end{pmatrix}, \end{aligned}$$

где

$$f_k \begin{pmatrix} Z_{M,0} & Z_{M,1}, \dots, Z_{M,L-1} \\ Z_{M-1,0} & Z_{M-1,1}, \dots, Z_{M-1,L-1} \\ Z_{-M,0} & Z_{-M,1}, \dots, Z_{-M,L-1} \end{pmatrix} = \sum_{r=0}^{L-1} 2^r \sum_{m=-M}^M H_m Z_{mr} \quad (8)$$

Происходит увеличение объема ЗУ до  $2^{(2M+1)L}$  ячеек и сокращение времени вычисления свертки в  $L$  раз. Комбинирование (8) и (6) позволяет сократить объем памяти до  $S2^{(2M+1)L/S}$  ячеек и ускорить свертку в  $L$  раз по сравнению с (6).

Еще большее ускорение- при параллельной выборке из ЗУ.

#### **Свертка по двоичным разрядам чисел**

$$Y_i = \sum_{q=0}^{Q-1} \sum_{r=0}^{Q-1} S_{iqr} 2^{q+r} \quad (9)$$

где  $S_{iqr} = \sum_{m=-M}^M a_{mq} X_{i-m, r}$  - вычисляется с помощью счетчиков.

Выражения

$$C_{iq} = \sum_{r=0}^{Q-1} S_{iqr} 2^r \quad \text{и} \quad z_i = \sum_{q=0}^{Q-1} C_{iq} 2^q$$

Имеют одинаковую структуру и их можно также вычислить с помощью счетчиков.

# Локальная эквализация

*Эквализация гистограмм* -нелинейное преобразование, осуществляющееся по формуле:

$$D_{ij} = \sum_{q=0}^{D_{ij}} H(q)$$

Если гистограммы вычисляются по некоторому S- фрагменту (локальная эквализация), то

$$D_{ij} = \sum_{q=0}^{D_{ij}} H_S(q), \quad (1)$$

где гистограмма строится по S-окрестности, т.е.

$$H_S(q) = \sum_{m=-M}^M \sum_{n=-N}^N \delta(q, D_{i-m, j-n}) \frac{1}{(2M+1)(2N+1)} \quad (2)$$

S-символ Крошкера.

В таком виде алгоритм вычисления характеризуется низким быстродействием, связанным с последовательным механизмом формирования гистограмм, это ограничивает применение параллельных методов.

Безгистограммный алгоритм основан на следующем соотношении

Подставив (2) в (1) получим:

$$\begin{aligned} D_{ij} &= \frac{1}{(2M+1)(2N+1)} \sum_{q=0}^{D_{ij}} \sum_{m=-M}^M \sum_{n=-N}^N \delta(q, D_{i-m, j-n}) = \\ &= \sum_{m=-M}^M \sum_{n=-N}^N \frac{1}{(2M+1)(2N+1)} \delta(D_{i-m, j-n} \leq D_{ij}) \end{aligned} \quad (3)$$

$$\text{где } \delta(a \leq b) = \begin{cases} 0, \text{если } a > b \\ 1, \text{если } a \leq b \end{cases} \quad (4)$$

Операция (3) может быть использована для построения различных видов параллельного выполнения.

# Параллельное вычисление по группам элементов входного изображения

a) Представим соотношение  $D'_{ij} = \sum_{m=-M}^M \sum_{n=-N}^N \frac{1}{(2M+1)(2N+1)} \delta(D_{i-m,j-n} \leq D_{ij})$

$$D'_{ij} = \sum_{m=-M}^M S'_{ijm}, \quad (1)$$

где

$$S'_{ijm} = \sum_{n=-N}^N \delta(D_{i-m,j-n} \leq D_{ij}) \quad (2)$$

Вычисления по формуле (2)  $S'_{ijm}$  можно выполнять параллельно, так как количество сумм равно  $(2M+1)$ , то во столько раз увеличится быстродействие.

б) Если ввести в состав процессора  $(2M+1)(2N+1)$  **компараторов** и один сумматор, имеющих  $(2M+1)$   $(2N+1)$  одноразрядных входов, то алгоритм можно представить

$$V_i := \overline{1, j}$$

$$V_j := \overline{1, J}$$

$$R_{m,n} := \delta(D_{i-m,j-n} \leq D_{ij})$$

$$D'_{ij} = \sum_{m=-M}^M \sum_{n=-N}^N R_{m,n}$$

Для повышения быстродействия можно организовать параллельную работу компараторов и сумматора **S**, с использованием конвейерной организации, как это было использовано для процессора свертки.

# Медианные фильтры

Алгоритмы **медианных фильтров** основаны на вычислении медианы либо по всему изображению — глобальная медианная фильтрация, либо по некоторой S-окрестности — локальный алгоритм.

$$U_{ij}^{вых} = \text{MED}(S)$$

S — окрестность элемента  $U_{ij}^{вх}$

Один из алгоритмов поиска медианы — гистограммный алгоритм.

## Гистограммный алгоритм

Пусть элементы сигнала Q-разрядные числа. Для S-фрагмента сигнала, в котором элемент  $\{D_{i+m,j+n} | m=-M, M; n=-N, N\}$   $L=(2N+1)(2M+1)$

гистограмма  $\{H_q : q=0, G-1; G=2^Q\}$

строится по формуле

$$H_q = \sum_{m=-M}^M \sum_{n=-N}^N S(q, D_{i+m,j+n}) \frac{1}{(2N+1)(2M+1)}$$

$H_q$ -число(доля) элементов фрагмента, равных  $q$ .

**Медиана** — это средний элемент в упорядоченной по возрастанию последовательности элементов, т.е. для нахождения медианы нужно суммировать элементы гистограммы до тех пор, пока их сумма не превысит или не станет равной величине  $\frac{(L_s+1)}{2}$ . Индекс последнего просуммированного элемента гистограммы и даст медиану.

Для построения гистограммы фрагмента нужно вычислить  $L=(2M+1)(2N+1)$  операций суммирования типа  $H_q = H_q + 1$ , для нахождения медианы — не более  $G=2^Q$  циклов сравнения и суммирования

$$\begin{aligned} S &\geq S + H_q \\ S &< \frac{(N_s+1)}{2} \quad N_s = 1 \end{aligned}$$

## Мультигистограммный алгоритм МФ

Изменяя точность представления сигнала от 1 до Q двоичных разрядов можно определить Q гистограмм фрагмента  $S = \{D_{i+m,j+n} | m=-M, M; n=-N, N\}$

следующим образом

$$H_q^e = \sum_{m=-M}^M \sum_{n=-N}^N \left\{ 2^e \left( \sum_{k=1}^e d_{i+m,j+n,k} 2^{-k} \right) = q \right\}$$

$q=0, G-1$  двоичные цифры элемента  $D_{ij}$ . Поиск медианы осуществляется за Q циклов сравнения: по гистограмме  $H_q^1$  — определяется значение старшего разряда медианы, по гистограмме  $H_q^2$  — второго разряда медианы...

Каждый из Q циклов сравнения может быть выполнен параллельно.

С целью сокращения объема вычислений, необходимого для фильтрации, предложен приближенный метод вычисления медианы:

-определяются старшие разряды медианы

$$d_1^{'}, d_2^{'}, \dots, d_{Q-t}^{'}$$

-младшие разряды задаются равными

$$d_{Q-t+1}^{'}, d_V^{'}, \dots, d_Q^{'}$$

Отличие между точными и приближенными медианами равно  $2^{t-1}$

## Конвейерные алгоритмы МФ

Недостаток предыдущего — сложность формирования Q гистограмм.

Метод основан на последовательном исключении элементов изображения больших и меньших медиан.

Q-разрядные элементы изображения  $\{U_{i+m,j+n} | m=-M, M; n=-N, N\}$  можно представить в виде матрицы  $L \times Q$ , где  $L=(2M+1)(2N+1)$ . Пусть  $B_q, q=1, Q$  столбцы матрицы размером  $L \times Q$  (составлены из q-х разрядов элементов фрагмента изображения).

$\{S_q : q=1, \bar{Q}\}$  - векторы, указывающие исключенные элементы на  $q$ -м шаге алгоритма (включая предыдущие циклы).

$B_0, S_0$ -единичные векторы.

$\{B_q^0, B_q^1, q=1, \bar{Q}\}$  — число нулей (единиц) в векторе  $B_q$ , отмеченных нулями в соответствующих позициях  $S_{q-1}$

Алгоритм поиска:

$$F_1 := B_1^0; t = \frac{[(2M+1)(2N+1)]}{2}$$

$\forall q=1, \bar{Q}$

$S_q = S_{q-1}$

if  $F_q < t$ , then  $\{S_q := S_q \& B_q; F_{q+1} = F_q + B_{q+1}^0\}$

else

$\{S_{q1} = S_q \& \bar{B}_q; F_{q+1} := F_q - B_{q+1}^1\}$

end

$U'_{ij} = S_Q \& \{U_{i+m, j+n} \mid m=-\bar{M}, \bar{M}; n=-\bar{N}, \bar{N}\}$

где  $S_q \& B_q$  — конъюнкция между соответствующими элементами  $S_q$  и  $B_q$ .

$S_Q \& U_{i+m, j+n}$  - выбор элемента фрагмента изображения, соответствующего позиции единицы в векторе  $S_Q$ .

В результате последовательного исключения элементов изображения, больших и меньших медиан, за  $Q$  циклов, включающих в себя операции сравнения и суммирования(вычитания), определяется значение медианы.

Алгоритм особенно интересен для разработки специальных вычислителей.

# Медианные фильтры на основе сортирующих сетей

Медиана квадратного фрагмента  $(2M+1) \times (2M+1)$  элементов вычисляется с помощью одномерного **медианного фильтра** в течение 2-х этапов

1. Вычисление медианы строк (столбцов) фрагмента

$$U'_{-M} = \text{med}(U_{-M,j-M}, \dots, U_{-Mj}, \dots, U_{-M,j+M})$$

$$U'_{-i} = \text{med}(U_{i,j-M}, \dots, U_{ij}, \dots, U_{i,j+M})$$

$$U'_{M} = \text{med}(U_{M,j-M}, \dots, U_{M,j}, \dots, U_{M,j+M})$$

2. По медианам строк определяется псевдомедиана

$$U'^*_{ij} = \text{med}(U'_{-M}, \dots, U'_i, \dots, U'_M)$$

Для определения медианы  $U'^*_{ij} = \text{med}(S)$  нужно выполнить сортировку  $(2M+1)^2$  элементов.

Псевдомедиана:

- Сортировка  $(2M+1)$  элементов (при этом определяется медиана строки фрагмента  $U'_i$ , т.к. перекрывающиеся фрагменты изображения отличаются на одну строку) параллельно и сортировки  $(2M+1)$  элементов изображения, т.е. медиана строк. При этом вычисляется медиана строк.

## Алгоритм быстрой сортировки для медианной фильтрации

Обычная сортировка одномерного массива размерности  $n$  простого выбора (поиск  $\min$  и его перестановка) требует:

$\frac{1}{2}(n^2-n)$  операций сравнения;

max:  $\frac{n^2}{4} + 3(n-1)$  операций пересылок.

## Алгоритм быстрой сортировки (К.Хоара 1971г.)

(Сортировка с разделением)

$a_1, a_2, a_3, \dots, a_x, \dots, a_{n-1}, a_n$

Просматриваем массив слева направо, пока не встретим  $a_i > x$ , а затем справа налево, пока не встретим  $a_j < x$ . Тогда поменяем их местами.

Продолжим просмотр с обменом, пока не встретимся где-то вблизи  $x$  ( $i > j$ ). В этом случае исходный массив разделится на 2 части:

левая-элементы  $< x$

правая-элементы  $> x$

Применяя указанную процедуру рекурсивно к каждой из частей получим упорядоченный по возрастанию массив.

```
a:array[1..n] of real;
procedure quicksort;
procedure sort(l,r:integer);
var i,j:integer ;
k,w:real;
begin i:=l; j:=r;
x:=a[(l+r) div L];
repeat
  while a[i]<x do i:=i+1;
  while a[j]>x do j:=j+1;
  if i<=j then
  begin
    w:=a[i]; a[i]:=a[j];a[j]:=w;
    i:=i+1; j:=j-1;
  end
end;
```

```
until i>j
if l<j then sort(l,j);
if i<r then sort(r);
end;
begin
sort(1,n);
end;
```

Общее количество сравнений  $n \cdot \log_2 n$ . Общее количество перестановок max:  $n \cdot \log_2 n$ ,  
среднее:  $\frac{n}{6} \cdot \log_2 n$ .

# Модели изображений

1. *Черно-белая модель Bitmap*. Каждому пикселу ставится в соответствие 1 бит-0, если цвет белый и 1, если цвет черный — битовая карта
2. *Полутоновые изображения. (grayscale)* Пиксел полутонового изображения кодируется 8 битами(1 байт). Глубина цвета такого изображения составляет 8 бит, каждый пиксел может принимать 256 различных значений. Значения, принимаемые пикселями называются серой шкалой. Она имеет 256 градаций серого цвета.
3. *Индексированный цвет*. Первые цветные мониторы работали с ограниченной цветовой гаммой. Они кодировались или 4 битами (16 цветов) или 8 (256 цветов). Такие цвета получили название индексированных цветов. Индексированные цвета кодируются обычно 4 или 8 битами в виде цветовых таблиц. Например, графическая среда Windows 95 поддерживает цветовую таблицу из 8 бит/пиксел.
4. *Полноцветные изображения*. К ним относятся модели с глубиной цвета не менее 24 байт на пиксел (не менее 16,7 млн. оттенков). Поэтому их иногда называют True Color (истинный цвет). Битовый объем каждого пикселя распределяется по цветовым составляющим. Каждый цвет кодируется 8 битами. Цветовые составляющие обычно в программах организуются в виде каналов, совмещенное отображение каналов и определяет цвет изображения. К таким моделям относятся: RGB,CMYK и другие.

# Основные цветовые модели

## Модель RGB

Основана на том, что человеческим глазом непосредственно воспринимаются 3 цвета-красный (red), зеленый (green) и синий (blue). Остальные цвета образуются при смешении этих 3 основных.

Смешение 3 базовых цветов производится в разных пропорциях. В результате получается все многообразие оттенков. В модели RGB количество каждого компонента-256 градаций. Цветовые компоненты называются каналами. RGB-3 канальная цветовая модель.

Эту модель можно представить в виде 3-мерной системы координат

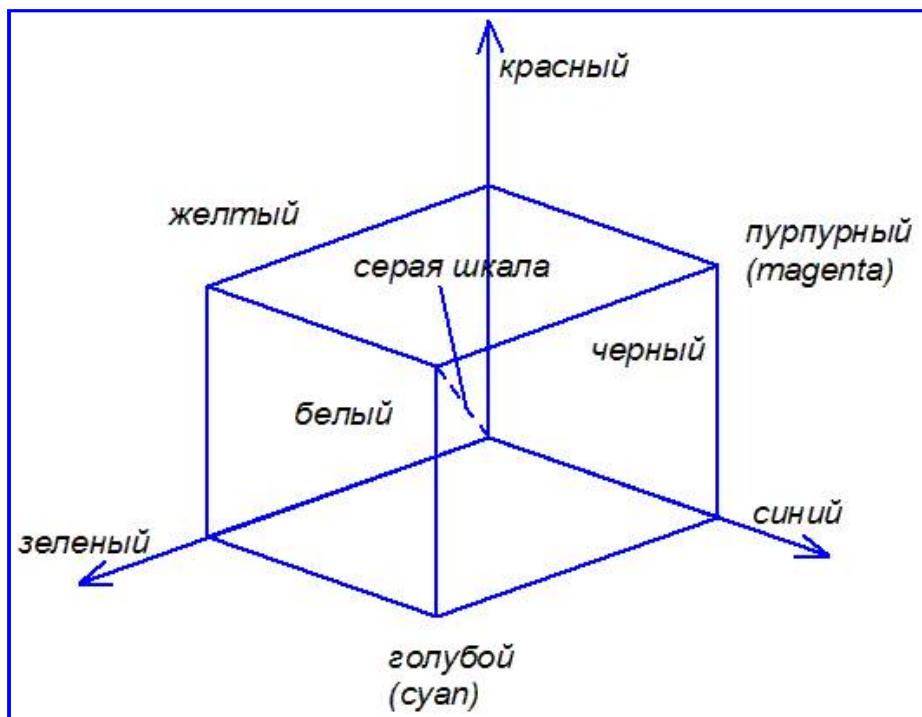


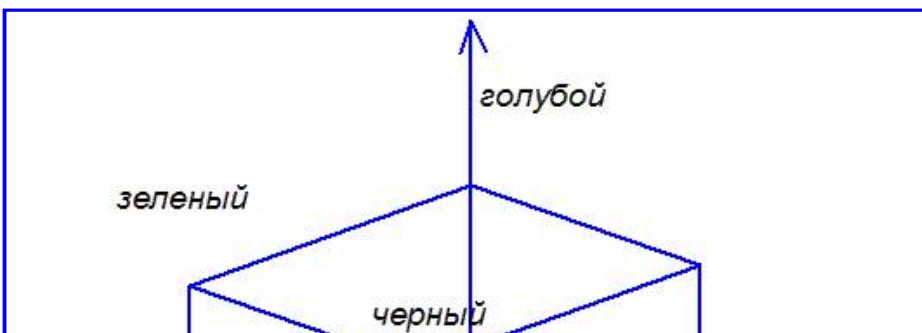
Рис. 1. Модель RGB

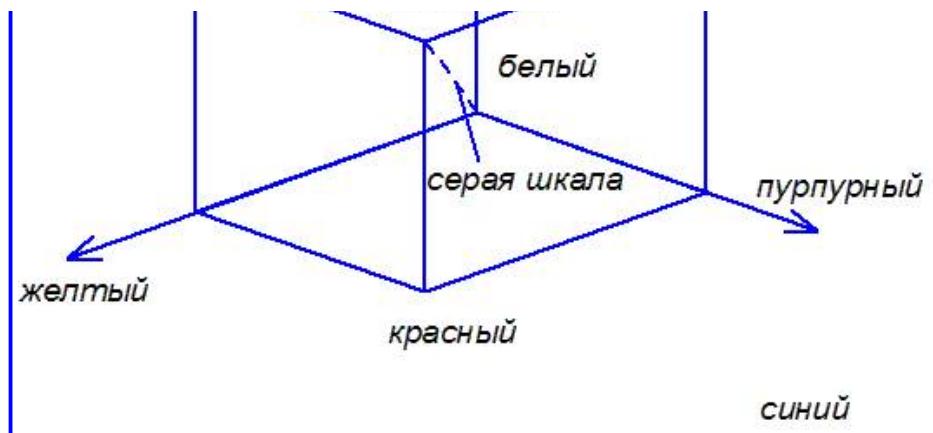
## Модель CMYK

В этой модели основные цвета образуются путем вычитания из белого цвета основных аддитивных цветов RGB. В результате основными цветами будут три: голубой (белый-красный), пурпурный (белый-зеленый), желтый (белый-синий).

Эти цвета составляют полиграфическую триаду. При печати этими цветами они поглощают красную, зеленую и синюю составляющую белого цвета так, что большая часть видимого цветового спектра может быть reproducedирована на бумаге.

Каждому пикселу **CMYK** изображения присваивается значение, определяющее процент содержания триадных цветов. При смешении 2-х составляющих результирующий цвет затемняется (в отличии от RGB, где засветляется). При смешении max значений всех 3-х цветов получается черный цвет.





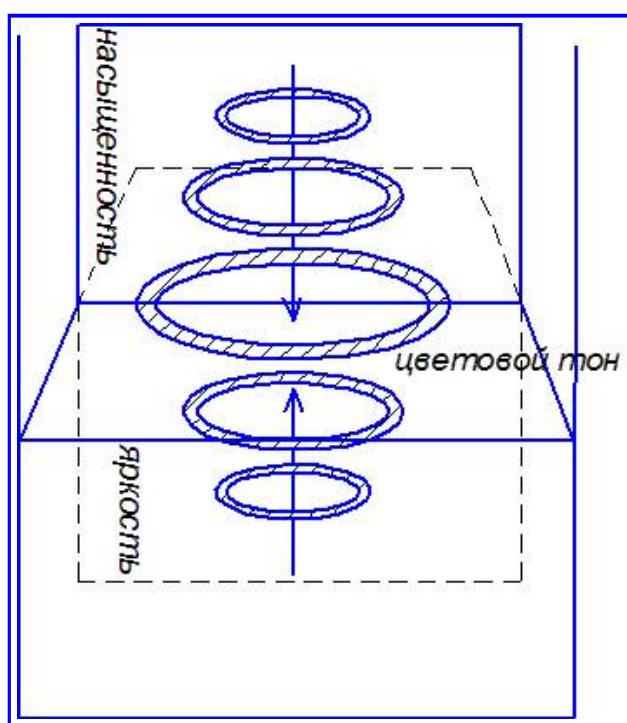
**Рис. 2. Модель CMYK**

Данная модель описывает реальные полиграфические краски, которые не так идеальны, как луч. Они имеют примеси. Поэтому при смешении 3 основных получается не черный, а темно-грязный цвет. Для компенсации взяли черную краску (black). 4-канальная модель (субстративные цвета)

### Модель HSB

Любой цвет в **HSB** модели определяется своим цветовым тоном (собственно цвет), насыщенностью (процентом добавленной к цвету белой краски), и яркостью (процентом добавленной черной краски).

Тон(hue), насыщенность (saturation), яркость (brightness)-3 канальная модель.



**Рис. 3. Цветовая модель HSB**

Спектральные цвета (тона) располагаются на цветовом круге и характеризуются величиной угла в диапазоне от 0 до 360°. Эти цвета обладают max насыщенностью и яркостью (100%)

С уменьшением насыщенности цвет осветляется. Если насыщенность =0%, то цвет белый. Насыщенность определяет чистоту цвета.

Яркость - определяет освещенность или затемненность цвета. Цвета на круге- max яркость(100%). Уменьшение яркости означает его затемнение. Яркость может изменяться от 0 до 100%. Работа с яркостью- добавление черной краски.

В общем случае, любой цвет получается из спектрального цвета добавлением определенного процента белой и черной краски (т.е. серой).

# Форматы файлов изображения

Формат PCX (фирма z-soft). Создан для программы PC PaintBrush. Практически любое приложение, работающее с графикой, импортирует его.

Однако, есть трудности при работе с CMYK → невозможно применение в издательских системах при создании цветных публикаций.

Формат BMP- предназначен для Windows. Поддерживается всеми приложениями Windows. Используется [индексированные цвета](#) → неприменим в издательской работе.

**Формат TIFF** (Tagged Image File Format)-универсальный формат (файл с расширением TIFF)

Достоинство- переносимость на различные платформы- IBM или Macintosh. Импортируется во всех издательских программах.

**Формат PSD** (Adobe Photo Shop Document)- внутренний для программы Adobe Photoshop. Поддерживает все модели изображения. Не универсален, не импортируется программами верстки и объектной графики.

**Формат JPEG** (Joint Photographic Experts Group) Сохранение точечных файлов со сжатием (расширение .jpg). Поддерживает [полноцветные изображения](#) в моделях RGB и CMYK.

Алгоритм сжатия с потерями. Не существенен для Internet или принтерных распечаток, но катастрофичен для типографии (сжатие от 5 до 15 раз).

**Формат GIF** (Graphics Interchange Format)- в основном для работы в Internet (Compuserve-разработчик). Поддерживает только индексированные изображения.

**Формат PCD** (Photo CD) (Eastman Kodak Corp)-для настольных редакторских систем. Используется для создания каталогов изображений. Полезно- определение требуемого разрешения при импорте.

**Формат EPS** (Encapsulated Post Script) формат описания изображений на языке Post Script для полиграфических целей (универсальный формат для точечных и векторных изображений, текстовой информации)

# Алгоритмы сжатия изображений

Основные требования к алгоритмам:

- высокая степень сжатия
- высокое качество изображения
- высокая скорость сжатия(компрессии)
- высокая скорость декомпрессии
- возможность масштабирования (изменение размеров до размеров активного окна). Например, некоторые алгоритмы могут вносить существенные искажения: JPEG и изображение-пиджак в мелкую клетку. Увеличение или уменьшение изображения-неприятные искажения
- устойчивость к ошибкам (локальность нарушений при порче или потере фрагмента)
- редактируемость-минимальная степень ухудшения качества при его повторном сохранении после редактирования.

Первая задача-метод обхода плоскости

Цель-создание одномерного массива D из двумерного массива S. Основное требование-при создании D желательно допускать как можно "разрывов". Каждый следующий элемент, заносимый на i-ом шаге должен являться соседним для предыдущего, занесенного в D на (i-1)-м шаге.

Типичный пример: алгоритм "змейка"(зигзаг-сканирование).

Такой алгоритм применяется в JPEG при обходе квадрантов.

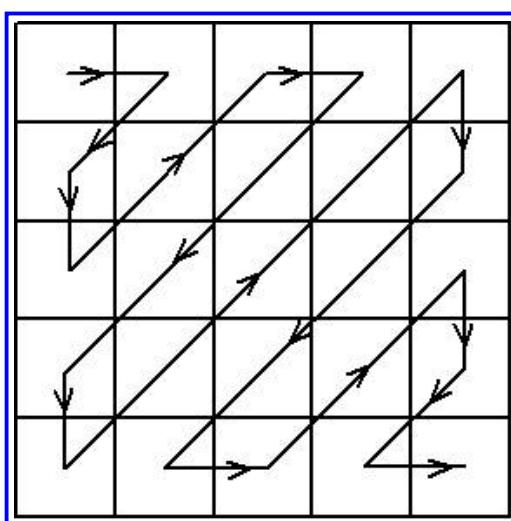


Рис. 1. Квадратная змейка

Рекурсивный метод для обхода квадратных областей.

Пусть левый верхний элемент 1-й. Для квадрата 2x2- два вида обхода:

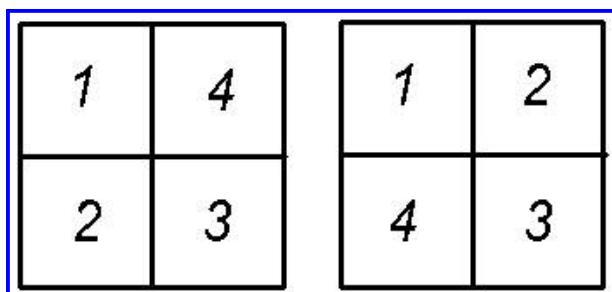


Рис. 2. виды обхода

Если 1-й проход был сверху вниз ,то 5-й шаг - переход к левому верхнему элементу квадрата справа (или к правому нижнему элементу квадрата сверху). Наоборот, если 1-й проход был вправо, то 5-й шаг переход к квадрату снизу (или слева)

Основные правила

- если нужно выйти к правому (или верхнему) квадрату то 1-й шаг-вправо если к нижнему(или

- если нужно выйти к правому (или верхнему) квадрату, то 1-й шаг вправо, если к нижнему (или левому), то 1-й шаг-влево.
- пройденным путем однозначно задается, к какому квадрату нужно выйти в каждый конкретный момент
- только в самом начале есть выбор одного из 2 вариантов обхода.

### Пример 1

1	2	15	16	17	20	21	22
4	3	14	13	18	19	24	23
5	8	9	12				
6	7	10	11	32		28	
					36		
60		56				40	
			52				
64				48		44	

Рис. 3.

1-й шаг внутри 2x2 -вправо → после его обхода мы должны выйти к нижнему квадрату 2x2.

1-й шаг от 2x2 к 2x2 внутри 4x4 был вниз → мы должны выйти к правому 4x4.

1-й шаг внутри 16x16 прийти к нижнему и т.д.

Разрывов нет.

Для 3x3 можно взять

Основные алгоритмы сжатия-это алгоритмы сжатия с потерями, т.к. старые алгоритмы (без потери) практически не сжимали изображение.

Основная проблема- найти компромисс между степенью сжатия и качеством. До сих пор адекватный критерий не найден.

Некоторые критерии:

#### 1. Среднеквадратичное отклонение (RMS)

$$d = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^n \sqrt{(x_{ij} - y_{ij})^2}$$

где  $x_{ij}$ -отсчет точного сигнала,  $y_{ij}$ - отсчет сжатого сигнала. По этому критерию изображение будет сильно испорчено при понижении яркости на 5% (глаз этого не заметит). В то же время изображение со снегом, полосами будут признаны хорошими.

#### 2. Максимальное отклонение

$$d = \max_{i,j} |x_{ij} - y_{ij}|$$

Эта мера чувствительна к биению отдельных пикселов.

#### 3. Мера отношение сигнал/шум (PSNR)

$$d = 10 \lg \frac{255^2 n^2}{\sum_{i=1}^n \sum_{j=1}^n (x_{ij} - y_{ij})^2}$$

Лучше всего идти по пути оценки качества изображения глазами.

# Алгоритм JPEG

Оперирует с областями 8x8 на которых яркость и цвет меняются плавно. (Joint Photographic Expert Group)-подразделение в рамках ISO.

## 1 шаг

Перевод изображения из стандарта RGB в цветовое пространство YCrCb (YUV).

Y-яркостная составляющая, Cr,Cb- красный и синий цвета.

Преобразование связано с тем, что глаз менее чувствителен к цвету, чем к яркости. Поэтому массивы Cr и Cb можно архивировать с большими потерями.

Матрица переход:

$$\begin{vmatrix} Y \\ Cb \\ Cr \end{vmatrix} = \begin{vmatrix} 0,299 & 0,587 & 0,144 \\ 0,5 & -0,4187 & -0,0813 \\ 0,1687 & -0,3313 & 0,5 \end{vmatrix} * \begin{vmatrix} R \\ G \\ B \end{vmatrix} + \begin{vmatrix} 0 \\ 128 \\ 128 \end{vmatrix}$$

Обратное преобразование:

$$\begin{vmatrix} R \\ G \\ B \end{vmatrix} = \begin{vmatrix} 1 & 0 & 1,402 \\ 1 & -0,34414 & -0,71414 \\ 1 & 1,772 & 0 \end{vmatrix} * \left( \begin{vmatrix} Y \\ Cb \\ Cr \end{vmatrix} - \begin{vmatrix} 0 \\ 128 \\ 128 \end{vmatrix} \right)$$

## 2 шаг

Разбиваем исходное изображение на матрицы размером 8x8. Из каждой матрицы формируем 3 рабочие матрицы **ДКП** (по 8 бит отдельно для каждой компоненты)

Для сильных сжатий- компонента Y- полностью, а компоненты Cr и Cb — через столбец.

## 3 шаг

Вычисляем коэффициенты ДПФ

$$Y[U,V] = \frac{1}{4} \sum_{i=0}^7 \sum_{j=0}^7 C(i,u)C(j,v)y(i,j)$$

где

$$C(i,u) = A(u) \cos\left(\frac{(2i+1)u\pi}{2n}\right),$$

$$A(u) = \begin{cases} 1/\sqrt{2}, & u=0 \\ 1, & u \neq 0 \end{cases}$$

ДКП применяется к каждой рабочей матрице 8x8. В результате- матрица ДКП- коэффициентов. Коэффициенты в левом верхнем углу -низкие частоты, в правом нижнем-высокие частоты.

## 4 шаг

Квантование для каждой компоненты

$$Y_q(U,V) = \text{InterRound}\left( \frac{Y(U,V)}{q(U,V)} \right)$$

q(U,V)-пороговое значение, которое задается для каждой рабочей матрицы Y,U,V-матрица квантования(МК). Здесь максимальные потери.

В стандарт JPEG- специальные MR, построенные опытным путем. Коэффициент у позволяет получить большее или меньшее сжатие умножением на них.

Специальные эффекты: при больших значениях у потери в низких частотах могут быть так велики, что изображение распадается на квадраты 8x8. Потери в высоких частотах- эффект Гиббса. Где резкий переход цвета образуется "нимб".

## 5 шаг

Матрицу 8x8 переводим в 64-элементный одномерный массив (алгоритм "змейка") В начале вектора коэффициенты, соответствующие низким частотам-НЧ, а в конце -ВЧ.

## 6 шаг

Свертываем вектор с помощью алгоритма группового кодирования: результат- пары чисел типа <пропустить число>, где <пропустить>-счетчик пропускаемых нулей, <число>-значение , которое необходимо поставить в следующую ячейку.

## Пример 1

$$42,3,0,0,0,-2,0,0,0,0,1 \Rightarrow (0,42),(0,3),(3,-2),(4,1)$$

### 7 шаг

Свертываем получившиеся пары кодированием по Хаффману с фиксированной таблицей.

Обратное ДКП

$$y[i,j] = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 Y[u,v] C(i,u) C(j,v)$$

# Алгоритм Хаффмана

Пусть есть алфавит-исходные  $\Psi = \{a_1, \dots, a_N\}$

Алфавит для кодировки  $\Omega = \{0, 1\}$

Задача заключается в сопоставлении символу алфавита  $\Psi$  слова из алфавита  $\Omega$ .

$a_1 \rightarrow B_1$

$a_2 \rightarrow B_2$

$a_N \rightarrow B_N$

1 шаг

Упорядочиваем все буквы входного алфавита в порядке убывания вероятности. Считаем, что все слова  $B_i$  из алфавита  $\Omega = \{0, 1\}$  пустые.

2 шаг

Объединяем 2 символа  $a_{ir-1}$  и  $a_{ir}$  с наименьшими вероятностями  $P_{ir-1}$  и  $P_{ir}$  в один псевдосимвол  $a' \{a_{ir-1}, a_{ir}\}$  с вероятностью  $P_{ir-1} + P_{ir}$ . Дописываем 0 в начало слова  $B_{ir-1}$  ( $B_{ir-1} = \overline{0}B_{ir-1}$ ) и 1 в начало слова  $B_{ir}$  ( $B_{ir} = \overline{1}B_{ir}$ )

3 шаг

Удаляем из списка упорядоченных символов  $a_{ir-1}$  и  $a_{ir}$ , заносим туда псевдосимвол  $a' \{a_{ir-1}, a_{ir}\}$ . Проводим 2 шаг, добавляя по необходимости 1 и 0 для всех слов  $B_i$ , соответствующих псевдосимволам до тех пор, пока в списке не останется 1 псевдосимвол.

## Пример 1

В  $\Psi$  - 4 буквы  $\Psi = \{a_1, a_2, a_3, a_4\}$   $r=4$ , вероятность их встречи  $P_1=0.5, P_2=0.24, P_3=0.15, P_4=0.11$

Процесс построения схемы рис. 1

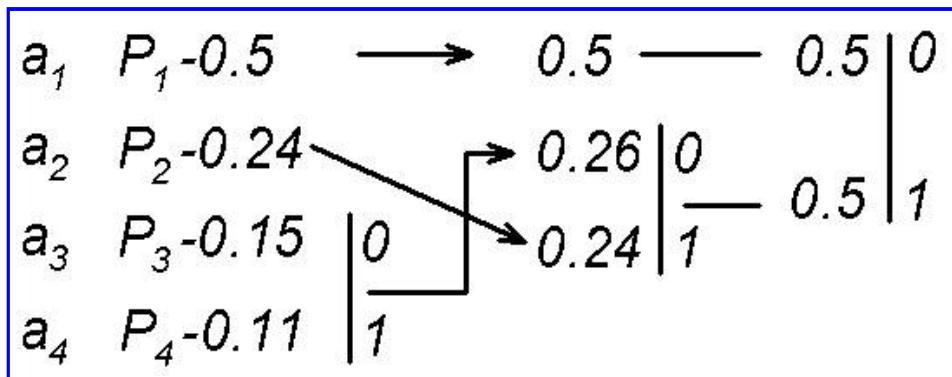


Рис. 1.

Чтобы получить кодирующее слово, нужно пройти по стрелкам от начальных символов к концу получившегося бинарного дерева. Получим схему:

$a_1-0$   
 $a_2-11$   
 $a_3-100$   
 $a_4-101$

код Хаффмана

Самый часто встречающийся символ-0, самый редко встречающийся -101

Помимо самого файла получается таблица соответствия кодируемым символам и кодирующих цепочек-код Хаффмана с фиксированной таблицей.

# Вейвлеты

## Введение в вейвлеты

Вейвлеты - математические функции, позволяющий анализировать частотные компоненты данных  $\Gamma_{i-x}$  пиксельное изображение

[9735]

Усредним попарно значения пикселей

[84]

Чтобы восстановить величины 4-х исходных пикселей составим разность 1-го и среднего (1) 3-го и 2-го среднего: [8 4 1 -1]

Применим указанную процедуру для 2-х первых средних

[6] [2] [1 -1]

## Вейвлеты Хаара

Будем говорить об изображении, как о функции, заданной на отрезке  $[0,1]$ . (функции кусочно-постоянны)

Изображение в 1 пиксель- функция, постоянная на всем отрезке. Пространство таких функций-  $V^0$ . Двухпиксельные функции образуют пространство  $V^1$ . Эти функции кусочно-постоянны на отрезках  $[0,1/2][1/2,1]$  и т.д.  $V^j$ -пространство кусочно-постоянных функций, которые постоянны на каждом из интервалов, число которых равно  $2^j$ .

Очевидно, что функции, являющиеся элементом  $V^j$ , являются элементом и  $V^{j+1}$ , т.е.

$V_0 \in V_1 \in V_2 \in \dots \in$

Базис пространства  $V^j$  -min бином векторов  $V^j$ , линейная комбинация которых даст  $\forall$  вектор пространства  $V^j$ . Эти функции называются масштабирующие функции и обозначаются  $\Phi$ . Пример базиса для  $V^j$

### Пример 1

Пример базиса для  $V^j$

$$\Phi_i^j(x) = \Phi(2^j x - i) \quad i=0,1,\dots,2^j-1$$

$$\Phi(x) = \begin{cases} 1 & 0 \leq x \leq 1 \\ 0 & \text{в противном случае} \end{cases}$$

### Пример 2

Пример для  $V^2$

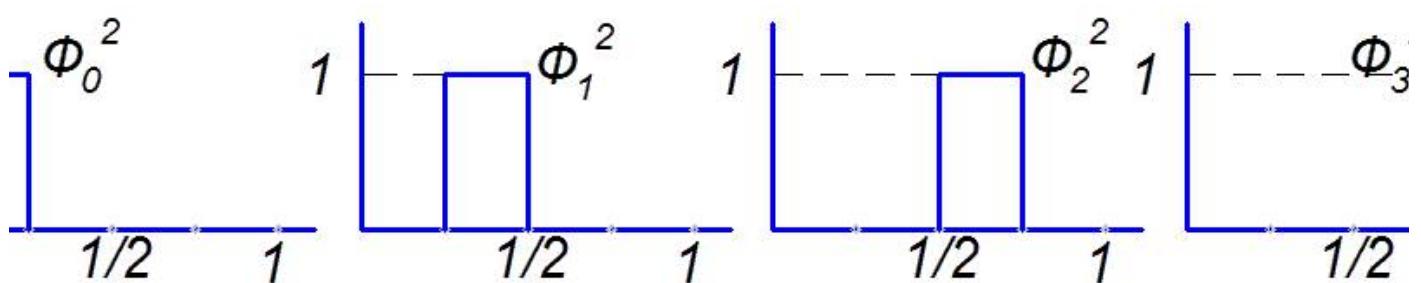


Рис. 1.

базис-  $V^2$

скалярное произведение 2-х функций  $f(x)$  и  $g(x)$

$$\langle f|g \rangle = \int_0^1 f(x)g(x)dx$$

2 функции  $U$  и  $V$  ортогональны, если  $\langle U|V \rangle = 0$

$\Phi_i^j$ -ортогональные функции.

### Пример 3

Пример разложения

4-х пиксельное изображение {9 7 3 5}

$$T(x)=\Phi_0^2(x)9+\Phi_1^2(x)7+\Phi_2^2(x)3+\Phi_3^2(x)5=C_0^2\Phi_0^2(x)+C_1^2\Phi_1^2(x)+C_2^2\Phi_2^2(x)+C_3^2\Phi_3^2(x)$$

Определим новое векторное пространство  $W^j$  (разложен в  $V^2$ ).  $W^j$ -ортогональное дополнение для  $V^j$  в пространстве  $V^{j+1}$ . Это пространство всех функций в  $V^{j+1}$ , ортогональных всем функциям в  $V^j$ . Базис  $\Psi_i^j(x)$  пространства  $W^j$  называется множеством вейвлетов.

Свойства:

1. Базисные функции  $\Psi_i^j$  из  $W^j$  вместе с базисными функциями  $\Phi_i^j$  из  $V^j$  образуют базис  $V^{j+1}$ .
2.  $\forall$  базисная функция  $\Psi_i^j$  из  $W^j$  ортогональна  $\forall$  любой базисной функции  $\Phi_i^j$  из  $W^j$ .

Вейвлеты Хаара

$$\Psi_i^j = \Psi(2^j x - i), i=0,1,\dots,2^{j-1}$$

где

$$\Psi(x)=\begin{cases} 1 & 0 \leq x < 1/2 \\ -1 & 1/2 \leq x < 1 \\ 0 & \text{в остальных случаях} \end{cases}$$

Вейвлеты Хаара для  $W^1$  рис. 2

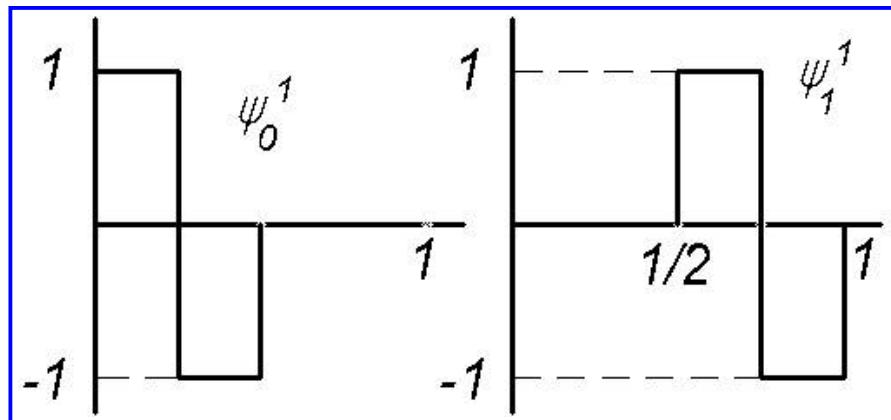


Рис. 2.

$$T(x)=8\cdot\Phi_0^1(x)+4\cdot\Phi_1^1(x)+1\cdot\Psi_0^1(x)+(-1)\cdot\Psi_1^1(x)$$

Разложение по базисным функциям в пространстве  $W^1V^1$

Опять выполним усреднение и попарные разности:

$$T(x)=6\cdot\Phi_0^0(x)+2\cdot\Psi_0^0(x)+1\cdot\Psi_0^1(x)+(-1)\cdot\Psi_1^1(x)$$

Разложение по функциям  $V^0$ ,  $W^0$ ,  $W^1$

Таким образом это разложение является точным представлением изображения.

Обращает???? тот факт, что большинство коэффициентов малы.

Используя обратное преобразование по этим коэффициентам можно точно! восстановить изображение.

### Вейвлет-преобразование для двумерного случая

1. Стандартное разложение: применяем преобразование к каждой строке пикселов. Затем выполним преобразование для каждого столбца.

2. Пусть есть квадрат из 4-х точек:

$$a_{2i,2j}, a_{2i+1,2j}, a_{2i,2j+1}, a_{2i+1,2j+1}$$

$$b_{ij}^1 = (a_{2i,2j} + a_{2i+1,2j} + a_{2i,2j+1} + a_{2i+1,2j+1})/4$$

$$b_{ij}^2 = (a_{2i2j} + a_{2i+12j} - a_{2i2j+1} - a_{2i+12j+1})/4$$

$$b_{ij}^3 = (a_{2i2j} - a_{2i+12j} + a_{2i2j+1} - a_{2i+12j+1})/4$$

$$b_{ij}^4 = (a_{2i2j} - a_{2i+12j} - a_{2i2j+1} + a_{2i+1,2j+1})/4$$

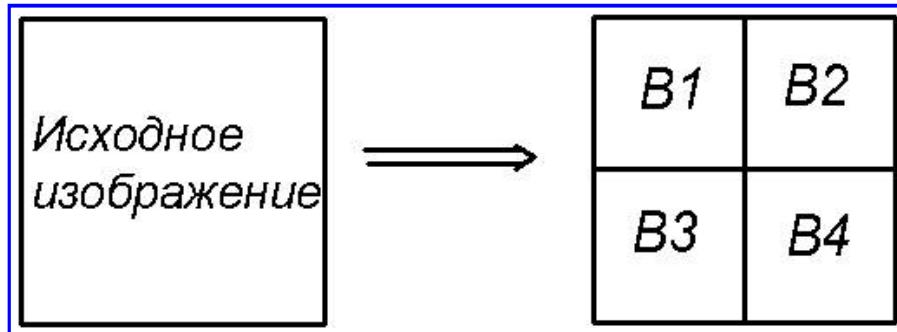


Рис. 3.

Из изображения 512x512 получим 4 (256x256)

1-я матрица - уменьшенная копия изображения

2-я -матрица - уменьшенная копия изображения

Дихотомический поиск:

1. Одномеризовать двумерную матрицу коэффициентов

2. Расположить по убыванию

3. Отбросить все коэффициенты:  $\sum_{i=m}^N C_m^2 < \varepsilon$

Алгоритм:

1. Находим  $\tau_{\min}$  и  $\tau_{\max}$

2.  $\tau = \frac{\tau_{\min} + \tau_{\max}}{2}$

3. Находим сумму  $C^2[i] < \tau \rightarrow S = S + C^2[i]$  Если  $S < \varepsilon \rightarrow \tau_{\min} = \tau$ , else  $\tau_{\max} = \tau$

4. Повторяем 2,3 пока  $\tau_{\min} \approx \tau_{\max}$

Удалены: если  $\tau[i] < \tau \rightarrow C[i] = 0$

Нестандартная процедура:

попарно усредним по строкам, затем попарно - по столбцам и т.д.

# Алгоритм JPEG2000 (2000 г)

Две модификации: с потерями и без потерь.

## 1 шаг

Сдвиг яркости каждой компоненты *RGB* для выравнивания динамического диапазона (гистограмма приближается к 0) → увеличение степени сжатия.

$$I'(x,y) = I(x,y) - 2^{ST-1}$$

ST- свое для каждой компоненты

$$\text{При восстановлении } I(x,y) = I'(x,y) + 2^{ST-1}$$

## 2 шаг

Перевод изображения из *RGB* → *YUV*

а) для сжатия с потерями- обычная процедура

б) для сжатия без потерь:

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} \frac{R+2G+B}{4} \\ R-G \\ B-G \end{pmatrix}$$

обратное преобразование:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} U+G \\ Y - (\frac{U+V}{4}) \\ V+G \end{pmatrix}$$

## 3 шаг

Дискретное вейвлет-преобразование (DWT) 2-х видов:

Сжатие с потерями:

Таблица 1

i	низкочастотное $h_L(i)$	высокочастотное $h_H(i)$
0	1,11508	
$\pm 1$		
$\pm 2$		
$\pm 3$		
$\pm 4$		

Коэффициенты при восстановлении:

Таблица 2

i	низкочастотные $g_L(i)$	высокочастотные $g_H(i)$
0		
$\pm 1$		
$\pm 2$		
$\pm 3$		
$\pm 4$		

Для сжатия без потерь:

Таблица 3

i	сжатие	$h_L(i)$	$h_H(i)$	восстановление	$g_L(i)$	$g_H(i)$
0	.	6/8	1	.	1	6/8
$\pm 1$	.	2/8	-1/2	.	1/2	-2/8
$\pm 2$	.	-1/8	0	.	0	-1/8

Четные выходящие значения (низкочастотные преобразования)

$$Y_{out}(2n) = \sum_{j=0}^{N-1} x_{in} h_L(j-2n)$$

Нечетные-высокочастотный фильтр

$$Y_{out}(2n) = \sum_{j=0}^{N-1} x_{in} h_H(j-2n-1)$$

При этом отсчеты, лежащие в начале и конце последовательностей, симметрично достраиваются.

### 3.1 шаг

Далее применяется черезстрочное преобразование:

Четные- в начало, нечетные в конец. Это преобразование - сначала для строк, затем для столбцов.

### 4 шаг

Квантование (также, как и в JPEG)

### 5 шаг

Арифметическое сжатие (MQ-кодер)

# Распознавание образов

*Распознавание образов* — это совокупность средств и методов, позволяющих достигнуть и превзойти естественные средства восприятия и анализа окружающего мира.

Основные решаемые задачи:

- автоматическое чтение рукописного и машинописного текста
- восприятие речи
- медицинская диагностика
- дистанционная идентификация объекта
- интерпретация социологических данных
- криминалистика

Последовательность решения задачи:

1. Восприятие и обработка (предварительная) информации
2. Анализ информации
3. Классификация известных образов
4. Отнесение неизвестного образа к одному из классов

Восприятие и предварительная обработка:

Это- методы и приборы (аппаратные) получения информации об интересующих объектах и ее предварительная обработка (фильтрация), устранение помех.

## Анализ информации

*Анализ информации* — выделение существенных с точки зрения данной задачи характеристик объектов-признаков.

Признаки:

- количественные (рост, вес и т.д.)
- качественные (красный, синий)
- двоичные (есть свойство или нет)

Целесообразно приводить все признаки к одному виду:

## Пример 1

Цвет-красный-характеризовать в длинах волн.

Цвета: последовательность чисел красный-1, оранжевый-2, желтый-3 и т.д.

Веса- также градуировать: 0-10=1, 10-20=2 и т.д.

При этом шкала может быть неравномерной.

Результат анализа: получение таблицы "объект-признак"

$$\left\{ \begin{array}{l} \vec{x}_1 = x_{11}x_{12}x_{13}\dots x_{1m} \\ \vec{x}_2 = x_{21}x_{22}x_{23}\dots x_{2m} \\ \vec{x}_3 = x_{31}x_{32}x_{33}\dots x_{3m} \\ \vdots \\ \vec{x}_n = x_{n1}x_{n2}x_{n3}\dots x_{nm} \end{array} \right. \quad \vec{x}_n - \text{объекты}, x_{nm} - \text{признаки}$$

## Классификация

Операция классификации состоит в том, чтобы распределить данные по категориям (или классам), включающим совокупность образов имеющих одни и те же признаки(или близкие).

Классификация:

1. С "учителем"
2. Автоматическая (без учителя)
  1. С "учителем" -пространство признаков разделяется на известное число классов человеком на основе априорной информации.
  2. Автоматическая-количество и местоположение классов заранее не известно (пример-классификация живой природы)

## Основные требования, предъявляемые к классам

1. Компактность-точки в пространстве, представляющие объекты одного класса, расположены друг к другу ближе, чем к точкам, представляющим объекты других классов.

Расстояние между точками:

$$d_1(\vec{x}_i, \vec{x}_j) = \left\{ \sum_{k=1}^m (x_{ik} - x_{jk})^2 \right\}^{1/2} \text{-евклидова метрика}$$

$$d_2(\vec{x}_i, \vec{x}_j) = \sum_{k=1}^m |x_{ik} - x_{jk}| \text{-расстояние по манхеттону}$$

$$d_3(\vec{x}_i, \vec{x}_j) = \max_k |x_{ik} - x_{jk}| \text{-по Чебышеву}$$

Расстояние между точкой и классом  $C_0$ :

$$d(\vec{x}, C_0) = \inf \{ d(\vec{x}, \vec{P}), \vec{P} \in C_0 \}$$

2. Сепарабельность (разделимость)-классы являются ограниченными и не пересекаются между собой.  
Эти свойства выполняются не всегда (сильно зависит от удачного выбора признаков)

## 1 метод-Иерархическая группировка

Искомое описание- иерархическая структура. Иерархия строится на основе понятия расстояния → выбор метрики. Метод состоит в том, чтобы разработать последовательность разделений множества всех объектов на подгруппы, одна из которых обладает искомым свойством, а другая -не обладает.

Построение иерархии (Алгоритм Ланса и Вильямса)

Пусть  $X$ - множество реализаций, пронумерованное от 1 до  $m$  (множество объектов)

$$X = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_m\}$$

1. Вычисляется множество всех расстояний между объектами  $\{d(ij), 1 \leq i \leq m, 1 \leq j \leq m\}$ -таблица расстояний.

2. Определяется  $d(p,q) = \min_i (d(ij))$ -т.е. минимальное расстояние — это 1-е иерархическое объединение.

Столбцы и строки  $p$  и  $q$  из таблицы удаляются и вводятся (столбец и строка) г расстояний, определяемых по закону:

$$d(i,r) = \min_i \{d(i,p), d(i,q)\} \text{ — критерий min}$$

или

$$d(i,r) = \max_i \{d(i,p), d(i,q)\} \text{ — критерий max]$$

3. Если число столбцов и строк в сокращенной таблице  $> 1 \rightarrow$  идти к 2, иначе  $\rightarrow$  4.

4. Конец

## Пример 2

Пример иерархической группировки.

Дано: 4 объекта  $\{\vec{x}_1, \dots, \vec{x}_4\}$

1. Расстояние между объектами

Таблица 1

	$x_1$	$x_2$	$x_3$	$x_4$
$x_1$	0	5	0.5	2
$x_2$	5	0	1	0.6
$x_3$	0.5	1	0	2.5
$x_4$	2	0.6	2.5	0

2.  $d(x_1, x_3) = 0.5 \rightarrow \min \rightarrow$  1-е иерархическое объединение  $\{x_1, x_3\} = a$

По правилу  $\min d\{a, x_2\} = \min\{d(x_1, x_2), d(x_3, x_2)\} = 1$

$d\{a, x_4\} = \min\{d(x_1, x_4), d(x_3, x_4)\} = 2$

Таблица 2

	$x_2$	$x_4$	$a$
$x_2$	0	0.6	1
$x_4$		0	2
$a$			0

Новая группировка  $b=\{x_2, x_4\}$

Таблица 3

	a	b
a	0	1
b	1	0

Окончательное объединение  $C=\{a, b\}$

Построение удобно отображать на графике (рис. 1):

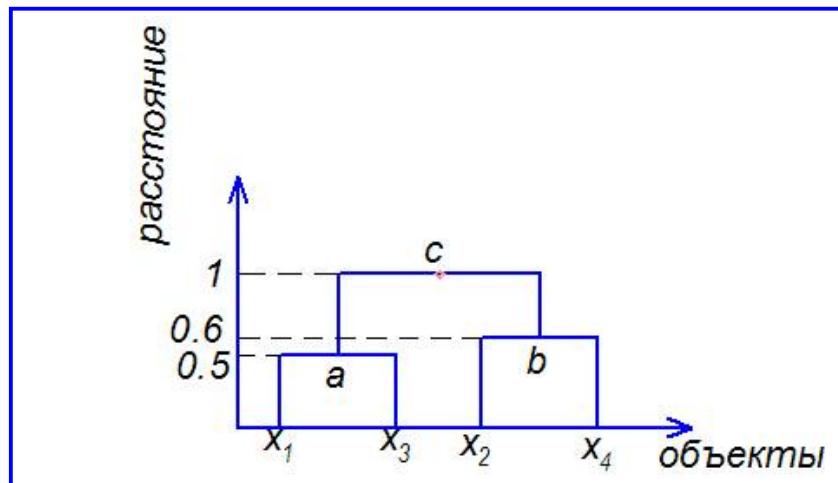


Рис. 1.

Таким образом - разбиение на 2 класса.

Классификация по max дает :

$$a=\{x_1, x_3\} \quad b=\{x_2, x_4\} \quad c=\{a, b\}$$

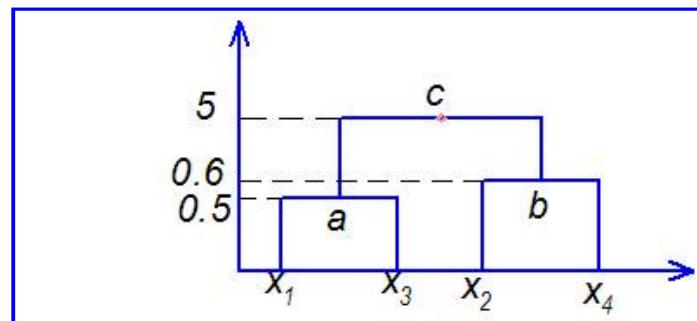


Рис. 2.

# Методы распознавания

Пусть задано разбиение  $\{c_k\}$  на  $k$  известных классов. Проблема заключается в том, чтобы отнести неизвестный предъявленный отчет к одному из известных классов.

# Вероятностный подход

Пусть  $\vec{x}_m$  - предъявляемый объект. Решение об отнесении неизвестного объекта к классу с номером  $k$  можно считать оправданным, если для любого  $j$  выполняется условие:

$$P(c_k/\vec{x}_m) \geq P(c_j/\vec{x}_m),$$

где  $P(a/b)$  - условная вероятность, то есть вероятность события  $a$  при условии того, что произошло событие  $b$ .

По **теореме Байеса**

$$P(a/b) = \frac{P(a)P(b/a)}{P(b)};$$

таким образом:

$$P(c_k/\vec{x}_m) = \frac{P(c_k)P(\vec{x}_m/c_k)}{P(\vec{x}_m)}.$$

Отсюда вытекает следующее решающее правило:

$$P(c_k)P(\vec{x}_m/c_k) \geq P(c_j)P(\vec{x}_m/c_j).$$

Частный случай - разделение на 2 класса:

$$\frac{P(\vec{x}_m/c_1)}{P(\vec{x}_m/c_2)} \geq \frac{P(c_2)}{P(c_1)}.$$

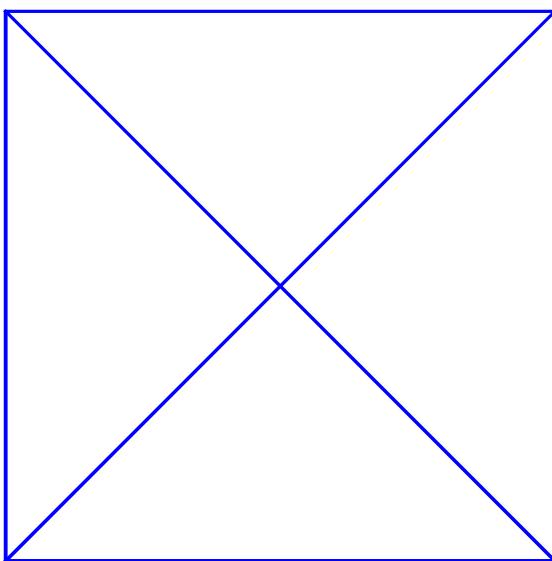


Рис. 1.

На прямой АВ неравенство выполняется, следовательно,  $\vec{x}_m \in c_1$ . Точка, где  $x_m < x$  - класс  $c_1$ ,  $x_m > x$  - класс  $c_2$ . Однако, вероятность того, что в первом случае  $x_m \in c_2$ , отлична от 0 (область 1).

Для класса  $c_1$  область 1 - зона ложной тревоги:

$$P_{л.т.} = \int_{-\infty}^x P(c_2)P(\vec{x}_m/c_2) d\vec{x}_m$$

Зона 2 - зона пропуска обнаружения:

$$P_{п.о.} = \int_x^{\infty} P(c_1)P(\vec{x}_m/c_1) d\vec{x}_m$$

Ошибка распознавания

$$P = P_{л.т.} + P_{п.о.}$$

# Метод минимального расстояния

Пусть  $\vec{x}_m$  - вектор исходных измерений,  $\vec{R}_i (i=1,2,\dots,k)$  - множество векторов, характеризующих исходные образцы.

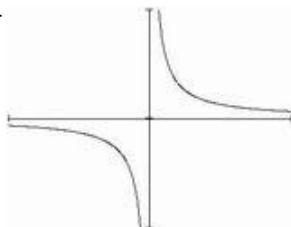
Решение принимается

$\vec{x}_m \in c_i$ , если  $\|\vec{x}_m - \vec{R}_i\| = \min$ ,

где  $\|\vec{x}_m - \vec{R}_i\|$  - расстояние между  $\vec{x}_m$  и  $\vec{R}_i$ .

Применяется:

1. Метрика Евклида  $\left\{ \sum_{j=1}^k (x_{mj} - r_{ij})^2 \right\}^{1/2}$
2. Метрика Минковского  $\left[ \sum_{k=1}^n |x_{mk} - r_{ik}|^{1/\lambda} \right]^\lambda$ ,  $\lambda$  - целое
3. Метрика Чебышева  $\max_k |x_{mk} - r_{ik}|$
4. Метрика Камберра  $\sum_{k=1}^n \frac{|x_{mk} - r_{ik}|}{|x_{mk} + r_{ik}|}$



# Метод разделяющих функций

Если функции сепарабельны,  $\Rightarrow$  существуют гиперповерхности, окаймляющие эти области,  $\Rightarrow$  границы раздела.

Пусть  $\vec{x}_m$  — вектор измерений.

**Разделяющей функцией** называется функция, которая обуславливает принятие решения, обеспечивающего выполнение условия:

$$\vec{x}_m \in C_i \Leftrightarrow g_i(\vec{x}_m) \geq g_j(\vec{x}_m); j=1,2,\dots,k$$

$g(\vec{x})$  — разделяющая функция.

# Линейные разделяющие функции

Пусть  $k$  - число классов,  $\vec{w}_i = \{w_{i1}, w_{i2}, \dots, w_{in}\}$  - вектор весовых коэффициентов. **Линейная разделяющая функция**

$$g_i(\vec{x}) = w_{i0} + \vec{w}_i \cdot \vec{x}$$

или, если ввести обозначения:

$$\vec{v}_i = \{w_0, w_1, \dots, w_n\}$$

$$\vec{y} = \{x_0, x_1, \dots, x_n\},$$

где обычно выбирают  $x_0 = 1$ , разделяющая функция имеет вид:

$$g_i(\vec{y}) = \vec{v}_i \cdot \vec{y},$$

а решающее правило записывается в виде:

$$\vec{y}_m \in c_i \Leftrightarrow g_i(\vec{y}_m) \geq g_j(\vec{y}_m) \quad \forall j.$$

Разделяющая поверхность описывается в этом случае уравнением:

$$g(\vec{y}) = g_i(\vec{y}) - g_j(\vec{y}) = 0.$$

Таким образом, необходимо найти коэффициенты  $w_{ij}$ . Нахождение коэффициентов происходит методом обучения и коррекции. Рассмотрим его на примере 2-х классов.

Пусть имеется 2 класса -  $c_1$  и  $c_2$ , а также векторы  $\vec{v}$  и  $\vec{y}$ . Критерий классификации:

$$\vec{v} \cdot \vec{y} > 0 \Rightarrow \text{класс } c_1$$

$$\vec{v} \cdot \vec{y} \leq 0 \Rightarrow \text{класс } c_2.$$

Принцип обучения: берутся известные объекты. Если результат классификации неверный, то применяют "наказание", состоящее в том, что изменяют веса, увеличивая или уменьшая их. Если результат верный, применяют "поощрение", веса оставляют неизменными.

Таблица 1

Решение	Отклик	Решение	Верно?	Изменение весов
$\vec{x}(i) \in c_1$	$\leq 0$	$c_2$	нет	$\vec{w}(i+1) = \vec{w}(i) + c \vec{x}(i)$
$\vec{x}(i) \in c_1$	$> 0$	$c_1$	да	$\vec{w}(i+1) = \vec{w}(i)$
$\vec{x}(i) \in c_2$	$> 0$	$c_1$	нет	$\vec{w}(i+1) = \vec{w}(i) - c \vec{x}(i)$
$\vec{x}(i) \in c_2$	$\leq 0$	$c_2$	да	$\vec{w}(i+1) = \vec{w}(i)$

$\vec{w}(i), \vec{w}(i+1)$  - выборы коэффициентов на  $i$  и  $i+1$  шаге коррекции,  $\vec{x}(i)$  - выбор измерений,  $c$  - некоторая константа.

Задача нахождения  $\vec{w}$  считается решенной, когда все предъявленные объекты классифицированы верно. Показано, что процедура сводится к конечному числу этапов, если классы линейно сепарабельны.

## Выбор $c$

$$\text{Пусть } \vec{\eta} = \begin{cases} +\vec{y}, & \vec{x} \in c_1 \\ -\vec{y}, & \vec{x} \in c_2 \end{cases}.$$

Процедура коррекции:

$$\vec{v}(i+1) = \begin{cases} \vec{v}(i) + \rho \vec{\eta}(i), & \vec{v}(i) \cdot \vec{\eta}(i) < 0 \\ \vec{v}(i) \text{ в остальных случаях}, & \rho > 0 \end{cases}$$

Умножим обе части на  $\vec{\eta}(i)$ :

$$\vec{v}(i+1) \cdot \vec{\eta}(i) = [\vec{v}(i) + \rho \vec{\eta}(i)] \cdot \vec{\eta}(i).$$

При правильном распознавании эта величина положительна, т.е.

$$\vec{v}(i) \cdot \vec{\eta}(i) + \rho |\vec{\eta}(i)|^2 > 0.$$

Коррекция производится только тогда, когда первый член неравенства меньше или равен нулю,  $\Rightarrow$  должно выполняться

$$\rho > \frac{|\vec{v}(i)\vec{\eta}(i)|}{|y(i)|^2},$$

откуда

$$\rho = \lambda \frac{|\vec{v}(i)\vec{\eta}(i)|}{|y(i)|^2}, \quad 0 < \lambda < 2$$

где для сходимости  $1 < \lambda < 2$ . Итак,

$$\vec{v}(i+1) = \begin{cases} \vec{v}(i) + \lambda \frac{|\vec{v}(i)\vec{\eta}(i)|}{|y(i)|^2}, & \vec{v}(i)\vec{\eta}(i) < 0; (-\lambda \frac{\vec{v}(i)\vec{\eta}(i)}{|y(i)|^2}) \\ \vec{v}(i) & \text{в противном случае} \end{cases}.$$

# Метод потенциалов

рассматривается идея разделения на 2 класса —  $C_1$  и  $C_2$ . Аналогия с полем электрических зарядов.

Представим облако точек, отображающих один класс, как некоторое потенциальное поле, отделенное от другого поле, отображающего другой класс, глубокой долиной, где потенциал минимален.

Определение минимальной эквипотенциали позволит найти границу раздела.

Пусть  $K(\vec{x}, \vec{x}_k)$  — потенциальная функция, центрированная относительно  $\vec{x}_k$  — выборки  $k$ -го объекта. Разделяющая функция находится с помощью суммарного потенциала  $K(\vec{x})$ , вычисляемого как сумма частных потенциалов  $K(\vec{x}, \vec{x}_i)$ , связанных с отдельными источниками  $i$ .

$K(\vec{x})$  определяется следующим алгоритмом, где  $i$  — номер этапа (соответствует номеру предъявляемого образца):

$$K_{i+1}(\vec{x}) = K_i(\vec{x}) + \rho_{i+1} K(\vec{x}, \vec{x}_{i+1}),$$

где

$$\rho_{i+1} = \begin{cases} 1, & \text{если } \vec{x}_{i+1} \in C_1 \text{ и } K_i(\vec{x}_{i+1}) \leq 0 \\ -1, & \text{если } \vec{x}_{i+1} \in C_2 \text{ и } K_i(\vec{x}_{i+1}) > 0 \\ 0 & \text{в остальных случаях, т.е. при правильной классификации} \end{cases}$$

**Правильная классификация:**  $K(\vec{x}) > 0$  при  $\vec{x} \in C_1$   
 $K(\vec{x}) \leq 0$  при  $\vec{x} \in C_2$ .

Если потенциал определяется с помощью ортогональных функций, то есть

$$K(\vec{x}, \vec{x}_i) = \sum_{n=1}^{\infty} \lambda_n^2 \phi_n(\vec{x}) \phi_n(\vec{x}_i),$$

то

$$R_{i+1}(\vec{x}) = \sum_{n=1}^{\infty} C_n(i+1) \phi_n(\vec{x})$$

и

$$\sum_{n=1}^{\infty} C_n(i+1) \phi_n(\vec{x}) = \sum_{n=1}^{\infty} C_n(i) \phi_n(\vec{x}) + \sum_{n=1}^{\infty} \lambda_n^2 \rho_{i+1} \phi_n(\vec{x}) \phi_n(\vec{x}_{i+1}).$$

$C_n(i+1)$  определяются обучением, т.е. предъявлением известных объектов.

Из ортогональности  $\phi_n(\vec{x})$

$$C_n(i+1) = C_n(i) + \rho_{i+1} \lambda_n^2 \phi_n(\vec{x}_{i+1}).$$

Наиболее часто в качестве  $\phi_n$  берутся **полиномы Эрмита**.

Рассмотрим пример для четырех объектов.

## Пример 1

$\vec{x}_1, \vec{x}_2 \in C_1, \vec{x}_3, \vec{x}_4 \in C_2$ .

$\vec{x}_1 = (-1, 0), \vec{x}_2 = (1, 1), \vec{x}_3 = (2, 0), \vec{x}_4 = (1, -2)$

$\phi_1(\vec{x}) = \phi_1(x_1, x_2) = H_0(x_1)H_0(x_2) = 1 \cdot 1 = 1$

$\phi_2(\vec{x}) = H_1(x_1)H_0(x_2) = 2x_1 \cdot 1 = 2x_1$

$\phi_3(\vec{x}) = H_0(x_1)H_1(x_2) = 2x_2$

$\phi_4(\vec{x}) = H_1(x_1)H_1(x_2) = 4x_1x_2$

Потенциальная функция

$$K(\vec{x}, \vec{x}_i) = \sum_{n=1}^4 \phi_n(\vec{x}) \phi_n(\vec{x}_i) = 1 + 4x_1x_1^{(i)} + 4x_2x_2^{(i)} + 16x_1x_2x_1^{(i)}x_2^{(i)}.$$

Определение разделяющей функции обучением. Начальное приближение:  $K_0(\vec{x}) = 0$ .

1.  $\vec{x}_1 \in C_1$   $K_0(\vec{x}_1) = 0$   $\rho_1 = 1$

$$K_1(\vec{x}) = K(\vec{x}, \vec{x}_1) = 1 - 4x_1$$

2.  $\vec{x}_2 \in C_1$

$$K_1(\vec{x}_2) = 1 - 4 < 0 \implies$$

$$K_2(\vec{x}) = K_1(\vec{x}) + K(\vec{x}, \vec{x}_2) = 2 + 4x_2 + 16x_1x_2$$

3.  $\vec{x}_3 \in C_2$

$$K_2(\vec{x}_3)=2>0 \implies$$

$$K_3(\vec{x})=K_2(\vec{x})-K(\vec{x}, \vec{x}_2)=1-8x_1+4x_2+16x_1x_2$$

4. Наконец,  $\vec{x}_4 \in C_2$

$$K_3(\vec{x}_4)=-47 \text{ верно} \implies$$

$$K_4(\vec{x})=K_3(\vec{x}).$$

При этом точка  $\vec{x}_1$  классифицирована верно,

$$K_4(\vec{x}_1)=9>0.$$

Разделяющая функция

$$K_4(\vec{x})=1-8x_1+4x_2+16x_1x_2=0.$$

$$x_2 = \frac{8x_1 - 1}{16x_1 + 4} \text{ - гипербола.}$$

# Выделение признаков

## 1. Распознавание кривых

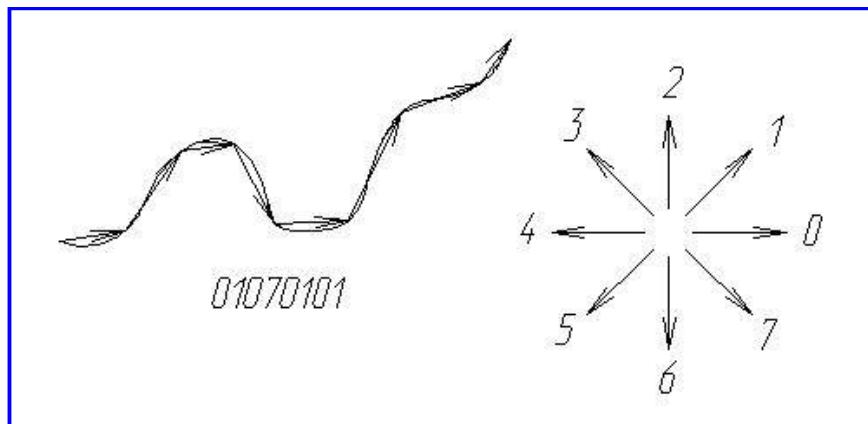


Рис. 1.

## 2. Электрокардиограммы

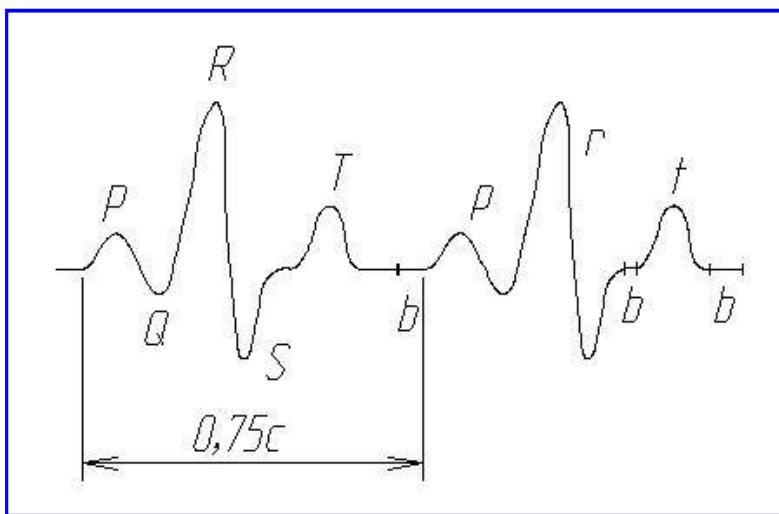


Рис. 2.

Эйтховен: PQRST

В автоматическом распознавании: p,r,b,t

## 3. Распознавание фигур (самолетов)

а) Инвариантные характеристики контура самолета — центральные моменты:

$$M_{pq} = \frac{1}{N} \sum_{i=1}^N (u_i - u)^p (v_i - v)^q,$$

где  $u_i, v_i$  — координаты  $i$ -ой точки контура,  $u, v$  — средние значения по ??? из  $N$  точек.

Для распознавания применяли следующие моменты:

$$M_1 = M_{20} + M_{02}$$

$$M_2 = (M_{20} - M_{02})^2 + 4M_{11}$$

$$M_3 = (M_{30} - 3M_{12})^2 + (3M_{21} - M_{03})^2$$

б) Представление самолета вектором, составляющие которого являются коэффициентами разложения контура в ряд Фурье.

$$z(\theta) = \sum_{k=-\infty}^{+\infty} c_k e^{-j2\pi k\theta}$$

$$Z(\Theta) = \sum_{k=-\infty}^{\infty} C_k e^{ik\Theta}, \quad 0 \leq \Theta \leq 2\pi$$

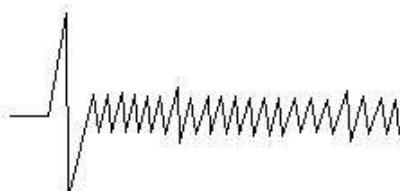
$\Theta$  — угол отклонения радиуса-вектора  $Z$ ,  $C_k$  — комплексные коэффициенты ряда Фурье.

$$C_k = \frac{1}{2\pi} \int_0^{2\pi} Z(\Theta) e^{2\pi ik\Theta} d\Theta$$

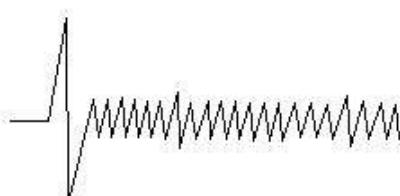
При соответствующей нормировке и выборе начала радиуса-вектора  $Z(\Theta)$  такое описание не зависит от изменения масштаба, угла поворота и сдвига.

#### 4. Спектральный анализ

а) **сейсмология**:



— сейсмограмма землетрясения



— ядерные взрывы

Каждый отрезок — двухкомпонентный вектор:  $x_1$  — энергия,  $x_2$  — число переходов через 0.

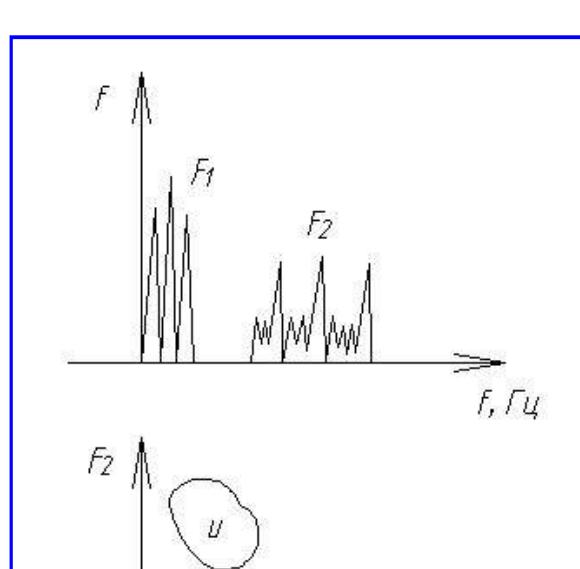
#### 5. Распознавание речи

3 вида мод:

- гласные — на выдохе — основной тон
  - согласные звонкие
  - согласные глухие
- а) Прямой метод:
- непосредственное наблюдение голосовых связок
  - оптическое — ввод внешнего источника и фототранзистора
  - ультразвуковой (2 преобразователя на шее)
  - электрический (оценка сопротивления мышечных тканей)
- б) Косвенные.

Наиболее простой — кодирование **сигнала** набором гласных, выбор определяется исходя из надежности их разделения 1 частотной области.

Форматы И:



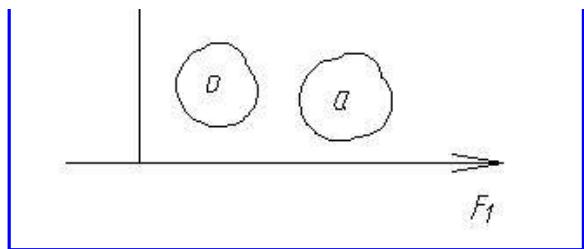


Рис. 3.

# Нейронные сети

Модель однослойной нейронной сети для двух классов:

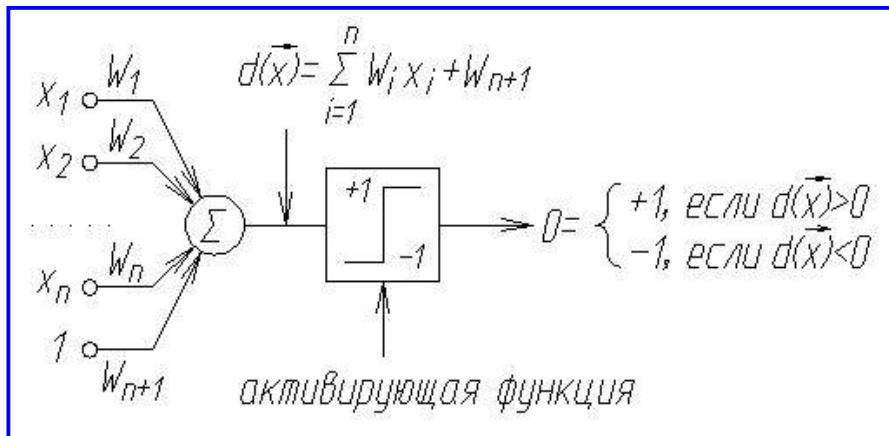


Рис. 1.

$w_i$  — веса

Выходной сигнал базируется на линейной дискриминантной функции  $d(\vec{x})$ .

Если  $d(\vec{x}) > 0 \Rightarrow$  на выходе +1 и  $\vec{x} \in C_1$ , если  $d(\vec{x}) < 0 \Rightarrow$  на выходе -1 и  $\vec{x} \in C_2$ .  $d(\vec{x}) = 0 \Rightarrow$  улучшение разделяющей поверхности (гиперплоскости).

**Обучение:** метод Уидроу-Хоффа — дельта-правило наименьшего соседнего квадрата.

**Целевая функция**  $J(\vec{w}) = \frac{1}{2}(r - \vec{w} \cdot \vec{y})^2$ , где  $r$  — желаемая реакция,  $r=1$ , если  $\vec{y} \in C_1$  и  $r=-1$ , если  $\vec{y} \in C_2$ .

Обучение состоит в том, чтобы найти  $\min$  этой функции, который достигается при  $r = \vec{w} \cdot \vec{y}$ .

$$\vec{w}(k+1) = \vec{w}(k) + \alpha [r(k) - \vec{w}(k) \cdot \vec{y}(k)] \vec{y}(k),$$

где  $\vec{w}(k+1)$  — вектор весов на шаге  $k+1$ ,  $\vec{w}(k)$  — вектор весов на шаге  $k$ ,  $0 < \alpha < 1$ .

# Многослойные нейронные сети

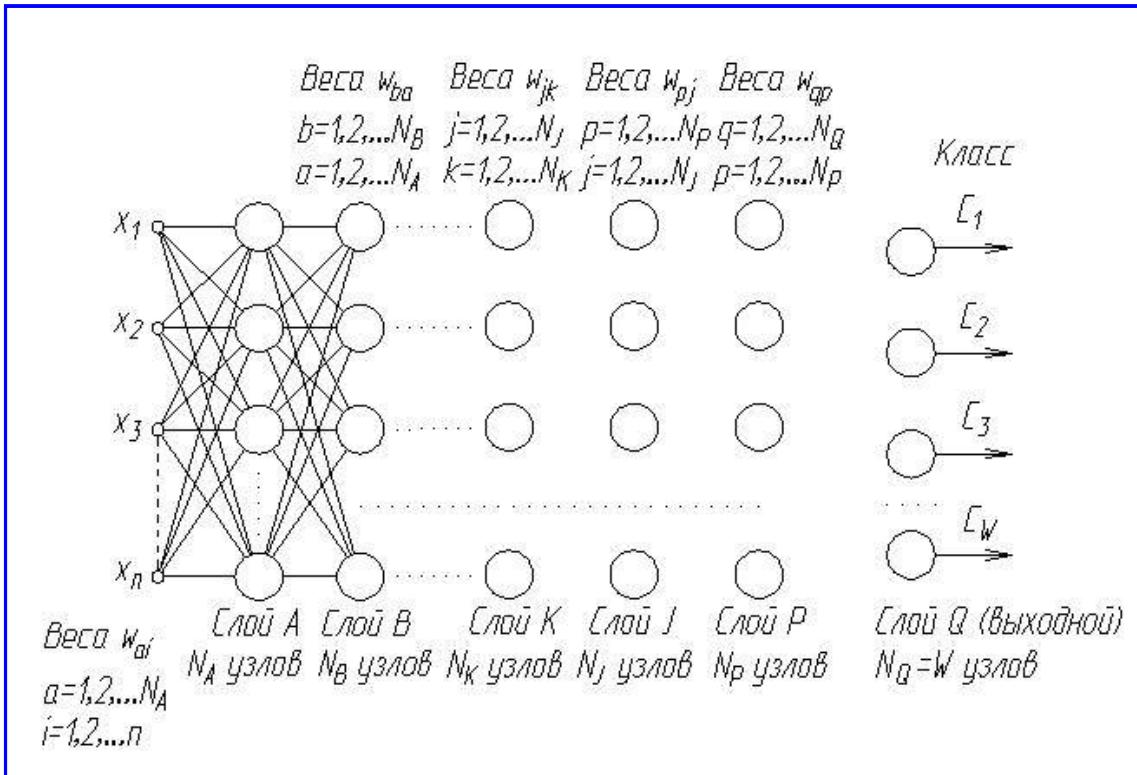


Рис. 1.

Активирующая функция со сглаженным порогом

$$h_j(I_j) = \frac{1}{1 + e^{-(I_j + \Theta_j)/\Theta_0}}$$

(*сигмоидальная функция*).

$I_j$ ,  $j=1,2,\dots,N_I$  — значения на входе активирующей функции каждого узла  $J$ -го слоя,  $\Theta_j$  величина смещения,  $\Theta_0$  — крутизна.

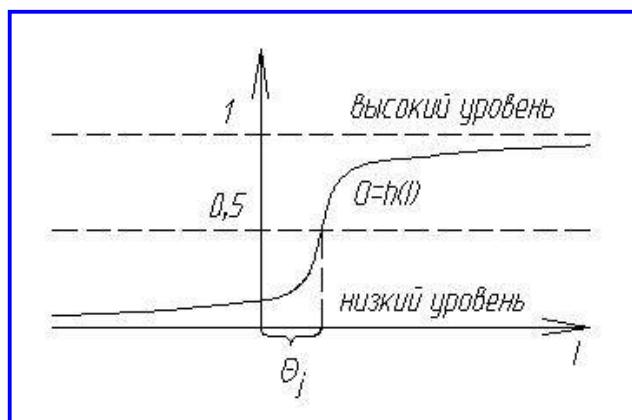


Рис. 2.

$\Theta_j$  аналогично коэффициенту  $w_{n+1}$ .

**Обучение:** метод обратного распространения ошибки.

$$I_j = \sum_{k=1}^{N_k} w_{jk} O_k,$$

где

- $I_j$  — сигнал на входе активирующего элемента каждого узла слоя  $J$

- $J=1,2,\dots,N_J$
- $w_{jk}$  — веса, модифицирующие выходные сигналы
- $O_k$  узлов слоя  $K$  на вход узлов слоя  $J$   
 $O_k = h_k(I_k), k=1,2,\dots,N_k$

Выходной слой:

суммарный квадрат ошибки между желаемыми реакциями узлов  $r_q$  выходного слоя  $Q$  и соответствующими фактическими реакциями  $O_q$  равен

$$E_Q = \frac{1}{2} \sum_{q=1}^{N_q} (r_q - O_q)^2 \quad (1)$$

Необходимо так построить процесс обучения, чтобы функция ошибки стремилась к минимальному значению.

$$\Delta w_{qp} = \alpha (r_q - O_q) h'_q(I_q) O_p, \quad (2)$$

где  $h'_q(I_q) = \frac{\partial h_q(I_q)}{\partial I_q}, 0,1 < \alpha < 1$ .

Продолжим рассмотрение, двигаясь назад от выходного слоя.

$$\Delta w_{pj} = \alpha \delta_p O_j, \quad (3)$$

где

$$\delta_p = h'_p(I_p) \sum_{q=1}^{N_Q} \delta_q w_{qp}; \quad (4)$$

$$\delta_q = (r_q - O_q) h'_q(I_q).$$

Далее

$$\Delta w_{jk} = \alpha \delta_j O_k,$$

где

$$\delta_j = h'_j(I_j) \sum_{p=1}^{N_p} \delta_p w_{pj},$$

и т.д.