Математические модели обработки сигналов

Тема 16: Дискретное всплеск-преобразование

Лектор: Кривошеин А.В.

Ортогональный КМА

Ортогональным КМА для пространства $L_2(\mathbb{R})$ называется последовательность замкнутых подпространств $\{V_j\}$, обладающая свойствами:

КМА1. Они вложены друг в друга $\ldots \subset V_{-1} \subset V_0 \subset V_1 \subset \ldots$

КМА2. Замыкание их объединения совпадает с $L_2(\mathbb{R})$: $\overline{\bigcup_{j\in\mathbb{Z}} V_j} = L_2(\mathbb{R})$.

КМА3. Пересечение всех подпространств состоит из нулевой функции: $\bigcap_{j\in\mathbb{Z}} V_j = \{0\}$.

КМА4. Масштабируемость: $f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1}$.

КМА5. Существует функция $\varphi(x) \in V_0$ с компактным носителем, что ее целочисленные сдвиги $\varphi_{0,n}(x) = \varphi(x-n)$ образуют ОНБ в пространства V_0.

Тогда $V_0 = \overline{\operatorname{span}\{\varphi_{0,k},\ k \in \mathbb{Z}\}}$.

Из КМА4 : $V_j = \overline{\operatorname{span}\{\varphi_{j,k},\ k \in \mathbb{Z}\}}$,

где $\varphi_{j,k} = 2^{j/2}\,\varphi\left(2^j - k\right)$.

Основное свойство функции φ: поскольку $\varphi \in V_0 \subset V_1$, то φ раскладывается по базису пространства V_1, то есть φ представима в виде линейной комбинации собственных сдвинутых, сжатых копий

$$\varphi(x) = \sum_{n\in\mathbb{Z}} h[n]\ \varphi_{1,n} = \sqrt{2} \sum_{n\in\mathbb{Z}} h[n]\ \varphi(2x-n).$$

Всплеск-функция ψ получается в виде

$\psi(x) = \sum_{n\in\mathbb{Z}} g[n]\ \varphi_{1,n}$ с некоторым набором коэффициентов g,

при этом система $\{\psi_{j,k}\}_{j,k\in\mathbb{Z}}$ образует ОНБ в $L_2(\mathbb{R})$.

Дискретное Всплеск-Преобразование

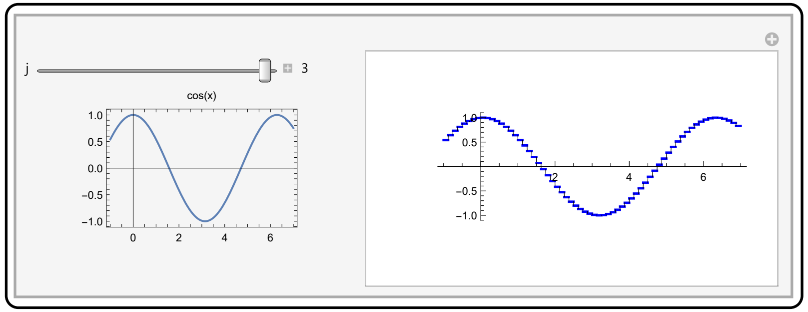
Пусть $f \in L_2(\mathbb{R})$. Построим начальное приближение функции f с помощью одного из пространств V_j, используя для этого оператор проектирования функции f на это пространство

$P_j : L_2(\mathbb{R}) \to V_j$,

$$P_j(f) = \sum_{k \in \mathbb{Z}} \langle f, \varphi_{j,k} \rangle \varphi_{j,k} = \sum_{k \in \mathbb{Z}} a_j[k] \varphi_{j,k} \in V_j.$$

На практике функция $f(x)$ представляется массивом значений $y_n = f(x_n)$ на достаточно мелкой сетке $x_n = n \Delta x$. В качестве начального приближения и берется этот массив значений, то есть $a_j[n] = y_n$.

Приближения с помощью $\mathrm{V_j}$



Дискретное Всплеск-преобразование

Далее, сравним коэффициенты начального приближения a_j
с коэффициентами более “грубого” приближения a_{j-1} на более низком уровне для пространства V_{j-1}.

Поскольку подпространство V_j можно разложить в прямую сумму подпространств $V_j = V_{j-1} \oplus W_{j-1}$, то проекция $P_j(f)$ является суммой проекций $P_{j-1}(f)$ на V_{j-1} и $P^W_{j-1}(f)$ на W_{j-1}. Причём в W_{j-1} система $\{\psi_{j-1,k}\}_{k\in\mathbb{Z}}$ является ОНБ, а значит можно записать

$$P_j(f) = P_{j-1}(f) + P^W_{j-1}(f) = \sum_{k\in\mathbb{Z}} a_{j-1}[k]\ \varphi_{j-1,k} + \sum_{k\in\mathbb{Z}} d_{j-1}[k]\ \psi_{j-1,k}, \text{ где}$$

$$a_{j-1}[k] = \langle f, \varphi_{j-1,k}\rangle, \quad d_{j-1}[k] = \langle f, \psi_{j-1,k}\rangle,$$

Фактически наборы коэффициентов a_{j-1}, d_{j-1} дают ту же информацию, что и набор a_j. Причем набор a_{j-1} отвечает за более "грубое" приближение функции f, а набор d_{j-1} отвечает за разницу между приближениями.

Всплеск-преобразование

Процесс можно продолжать и далее.

$V_j = V_{j-1} \oplus W_{j-1} = V_{j-2} \oplus W_{j-2} \oplus W_{j-1}$

$P_j(f) = P_{j-1}(f) + P^W_{j-1}(f) = P_{j-2}(f) + P^W_{j-2}(f) + P^W_{j-1}(f) =$

$$\sum_{k\in\mathbb{Z}} a_{j-2}[k]\,\varphi_{j-2,k} + \sum_{k\in\mathbb{Z}} d_{j-2}[k]\;\psi_{j-2,k}\,k + \sum_{k\in\mathbb{Z}} d_{j-1}[k]\;\psi_{j-1,k},$$

где $a_{j-2}[k] = \langle f, \varphi_{j-2,k}\rangle$, $d_{j-2}[k] = \langle f, \psi_{j-2,k}\rangle$.

После N шагов получим набор коэффициентов $\{a_{j-N},\ d_{j-N},\ \ldots,\ d_{j-1}\}$, соответствующий разложению

$V_j = V_{j-N} \oplus W_{j-N} \ldots \oplus W_{j-1}$,

$P_j(f) = P_{j-N}(f) + P^W_{j-N}(f) + P^W_{j-N+1}(f) + \ldots + P^W_{j-1}(f)$

Всплеск-преобразование

Графически этот процесс можно изобразить так

$$
\begin{tikzcd}
V_j \arrow[r] \arrow[dr] & V_{j-1} \arrow[r] \arrow[dr] & V_{j-2} \arrow[dr] & \ldots\ V_{j-N+1} \arrow[r] \arrow[dr] & V_{j-N} \\
& W_{j-1} & W_{j-2} & \ldots & W_{j-N}
\end{tikzcd}
$$

Или с точки зрения коэффициентов

$$
\begin{tikzcd}
P_j(f) \arrow[r] \arrow[dr] & a_{j-1} \arrow[r] \arrow[dr] & a_{j-2} \arrow[r] \arrow[dr] & \ldots & a_{j-N+1} \arrow[r] \arrow[dr] & a_{j-N} \\
& d_{j-1} & d_{j-2} & \ldots & & d_{j-N}
\end{tikzcd}
$$

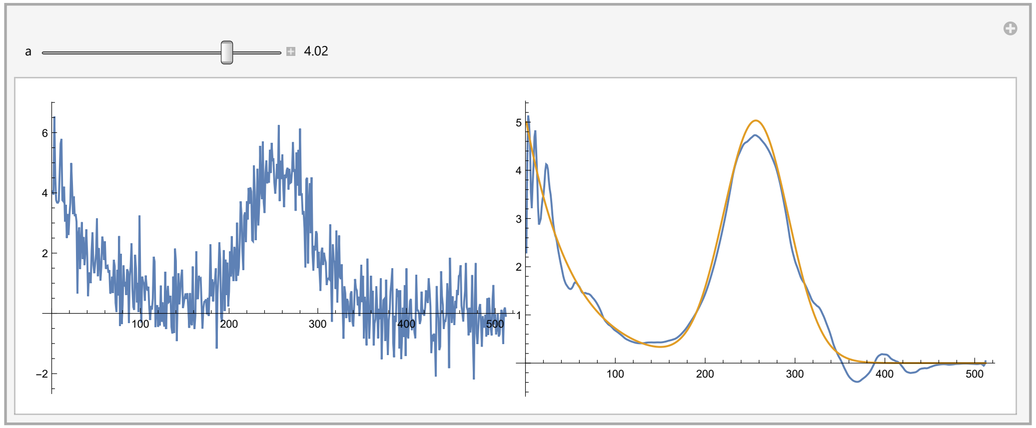
Всплеск-анализ

Набор коэффициентов $\{a_{j-N},\ d_{j-N},\ \ldots,\ d_{j-1}\}$ называют результатом всплеск-преобразованием функции f.
Всплеск-анализ сигнала заключается в следующих шагах:

1. Построение начального приближения функции f с помощью $P_j(f)$.
2. Нахождение коэффициентов разложения $\{a_{j-N},\ d_{j-N},\ \ldots,\ d_{j-1}\}$.
3. Анализ или преобразование коэффициентов (удаление шума, компрессия и др.)
4. Восстановление функции по коэффициентам с помощью формулы

$$f(x) \simeq \sum_{k\in\mathbb{Z}} a_{j-N}[k]\ \ \varphi_{j-N,k}(x) + \sum_{k\in\mathbb{Z}} d_{j-N}[k]\ \ \psi_{j-N,k}(x) + \ldots + \sum_{k\in\mathbb{Z}} d_{j-1}[k]\ \ \psi_{j-1,k}(x)$$

Например, восстановление зашумленного сигнала:



Быстрые алгоритмы

Для пересчёта коэффициентов при всплеск-преобразовании имеются быстрые алгоритмы. Формально, коэффициенты разложения определяются с помощью скалярных произведений, что очень затратно с точки зрения вычислений.

$$a_{j-m}[k] = \langle f, \varphi_{j-m,k} \rangle = \int_{\mathbb{R}} f(x)\, \overline{\varphi_{j-m,k}(x)}\, dx, \quad d_{j-m}[k] = \langle f, \psi_{j-m,k} \rangle = \int_{\mathbb{R}} f(x)\, \overline{\psi_{j-m,k}(x)}\, dx.$$

Быстрые алгоритмы для пересчёта коэффициентов не требуют вычисления скалярных произведений. Эти алгоритмы были предложены С.Малла (1986). Пусть у нас есть начальное приближение функции f с помощью $P_j(f)$ и нам известны коэффициенты a_j. Каким образом найти набор коэффициентов a_{j-1}, d_{j-1} и сделать первый шаг разложения

$$P_j(f) = \sum_{k \in \mathbb{Z}} a_{j-1}[k]\ \varphi_{j-1,k} + \sum_{k \in \mathbb{Z}} d_{j-1}[k]\ \psi_{j-1,k}.$$

без интегрирования и вычисления скалярных произведений?

Быстрые алгоритмы

Воспользуемся масштабирующим уравнением и определением всплеск-функции

$$\varphi(x) = \sum_{n\in\mathbb{Z}} h[n]\ \ \varphi_{1,n}(x) \quad \text{и} \quad \psi(x) = \sum_{n\in\mathbb{Z}} g[n]\ \ \varphi_{1,n}(x).$$

Последовательность h называют **масштабирующей маской**, g — **всплеск-маской**.

Из этих соотношений с помощью сдвигов и растяжения можно вывести соотношения связывающие функции $\varphi_{j-1,k}(x)$ и $\psi_{j-1,k}(x)$ с функциями $\varphi_{j,n}(x)$

$$\varphi_{j-1,k}(x) = \sum_{n\in\mathbb{Z}} h[n]\ \ \varphi_{j,n+2k}(x) \quad \text{и} \quad \psi_{j-1,k}(x) = \sum_{n\in\mathbb{Z}} g[n]\ \ \varphi_{j,n+2k}(x),$$

Действительно, если $\varphi(x) = \sum_{n\in\mathbb{Z}} h[n]\ 2^{1/2}\varphi(2x-n)$, то заменив $x \to 2^{j-1}x-k$ и домножив на $2^{(j-1)/2}$ получим :

$$2^{(j-1)/2}\varphi\left(2^{j-1}x-k\right) = \sum_{n\in\mathbb{Z}} h[n]\ \ 2^{j/2}\varphi\left(2^{j}x-n-2k\right).$$

Тогда новые коэффициенты для всплеск-разложения определяются как

$$a_{j-1}[k] = \langle f, \varphi_{j-1,k}\rangle = \left\langle f, \sum_{n\in\mathbb{Z}} h[n]\ \ \varphi_{j,n+2k}\right\rangle = \sum_{n\in\mathbb{Z}} \overline{h[n]}\, a_j[n+2k],$$

$$d_{j-1}[k] = \langle f, \psi_{j-1,k}\rangle = \left\langle f, \sum_{n\in\mathbb{Z}} g[n]\ \ \varphi_{j,n+2k}\right\rangle = \sum_{n\in\mathbb{Z}} \overline{g[n]}\, a_j[n+2k].$$

Быстрые алгоритмы

То есть мы свели подсчёт новых коэффициентов к операциям сложения и умножения. Изменим немного вид этих формул, записав их с помощью свертки. Обозначим $\overline{h[-n]} =: h^*[n]$, $\overline{g[-n]} =: g^*[n]$. Тогда

$$a_{j-1}[k] = \sum_{n\in\mathbb{Z}} \overline{h[n]}\, a_j[n+2k] = \sum_{n\in\mathbb{Z}} h^*[-n]\, a_j[n+2k] = \sum_{n\in\mathbb{Z}} h^*[n]\, a_j[2k-n] = h^* * a_j[2k]$$

$$d_{j-1}[k] = \sum_{n\in\mathbb{Z}} \overline{g[n]}\, a_j[n+2k] = \sum_{n\in\mathbb{Z}} g^*[-n]\, a_j[n+2k] = \sum_{n\in\mathbb{Z}} g^*[n]\, a_j[2k-n] = g^* * a_j[2k]$$

То есть для нахождения коэффициентов a_{j-1} и d_{j-1} надо подсчитать свертку коэффициентов a_j с h^* и g^* с последующим down-sampling. Схематично это можно представить так

$$a_j \longrightarrow h^* * a_j \xrightarrow{\downarrow 2} a_{j-1}.$$

$$a_j \longrightarrow g^* * a_j \xrightarrow{\downarrow 2} d_{j-1}.$$

Этот способ вычисления новых коэффициентов называется также **каскадными алгоритмами** или алгоритмами Малла.

Число коэффициентов в наборе $\{a_j[k]\}$ и в наборах $\{a_{j-1}[k]\}$, $\{d_{j-1}[k]\}$ практически одинаковое.

Процесс можно продолжить и далее, взяв в качестве начальных коэффициентов $\{a_{j-1,k}\}$

$$a_{j-1} \longrightarrow h^* * a_{j-1} \xrightarrow{\downarrow 2} a_{j-2}.$$

$$a_{j-1} \longrightarrow g^* * a_{j-1} \xrightarrow{\downarrow 2} d_{j-2}.$$

и так далее, получив в итоге набор $\{a_{j-N},\ d_{j-N},\ \ldots,\ d_{j-1}\}$.

Быстрые алгоритмы: восстановление

Рассмотрим теперь вопрос о восстановлении исходных коэффициентов a_j по коэффициентам, полученным в ходе всплеск-преобразования $\{a_{j-N},\ d_{j-N},\ \ldots,\ d_{j-1}\}$. Рассмотрим случай восстановления a_j по $\{a_{j-1},\ d_{j-1}\}$.

Коэффициенты $\{a_j\}$ участвуют в разложении функции f в подпространстве V_j по базису $\{\varphi_{j,k}\}$

$$P_j(f) = \sum_{k\in\mathbb{Z}} a_j[k]\ \varphi_{j,k}\,.$$

Коэффициенты $\{a_{j-1},\ d_{j-1}\}$ также участвуют в разложении функции f по базису $\{\varphi_{j-1,k},\ \psi_{j-1,k}\}$ в том же пространстве V_j в соответствии с формулой $V_j = V_{j-1} \oplus W_{j-1}$ или

$$P_j(f) = \sum_{k\in\mathbb{Z}} a_{j-1}[k]\ \varphi_{j-1,k} + \sum_{k\in\mathbb{Z}} d_{j-1}[k]\,\psi_{j-1,k}\,.$$

Поэтому для нахождения коэффициентов a_j перейдем от второй формулы к первой, используя следующие соотношения

$$\varphi_{j-1,k}(x) = \sum_{n\in\mathbb{Z}} h[n-2\,k]\ \ \varphi_{j,n} \quad \text{и} \quad \psi_{j-1,k}(x) = \sum_{n\in\mathbb{Z}} g[n-2\,k]\ \ \varphi_{j,n}.$$

которые получены заменой переменной суммирования из уже использованных соотношений:

$$\varphi_{j-1,k}(x) = \sum_{n\in\mathbb{Z}} h[n]\ \ \varphi_{j,n+2\,k}(x) \quad \text{и} \quad \psi_{j-1,k}(x) = \sum_{n\in\mathbb{Z}} g[n]\ \ \varphi_{j,n+2\,k}(x),$$

Быстрые алгоритмы: восстановление

$$P_j(f) = \sum_{k\in\mathbb{Z}} a_{j-1}[k]\ \varphi_{j-1,k} + \sum_{k\in\mathbb{Z}} d_{j-1}[k]\,\psi_{j-1,k} =$$

$$\sum_{k\in\mathbb{Z}} a_{j-1}[k] \sum_{n\in\mathbb{Z}} h[n-2\,k]\ \ \varphi_{j,n} + \sum_{k\in\mathbb{Z}} d_{j-1}[k] \sum_{n\in\mathbb{Z}} g[n-2\,k]\ \ \varphi_{j,n} =$$

$$\sum_{n\in\mathbb{Z}} \left(\sum_{k\in\mathbb{Z}} a_{j-1}[k]\ h[n-2\,k] + \sum_{k\in\mathbb{Z}} d_{j-1}[k]\ g[n-2\,k] \right) \varphi_{j,n} = \sum_{k\in\mathbb{Z}} a_j[k]\ \varphi_{j,k} = P_j(f)$$

Таким образом,

$$a_j[n] = \sum_{k\in\mathbb{Z}} a_{j-1}[k]\ h[n-2\,k] + \sum_{k\in\mathbb{Z}} d_{j-1}[k]\ g[n-2\,k],$$

что и даёт способ для восстановления коэффициентов a_j по a_{j-1}, d_{j-1}.

Быстрые алгоритмы: восстановление

Операцию восстановления можно записать с помощью свёртки с предварительным up-sampling последовательностей a_{j-1}, d_{j-1}. То есть, обозначив

$$\tilde{a}_{j-1}[m] = \begin{cases} a_{j-1}[k], & m = 2k \\ 0, & m = 2k+1 \end{cases}, \qquad \tilde{d}_{j-1}[m] = \begin{cases} d_{j-1}[k], & m = 2k \\ 0, & m = 2k+1 \end{cases},$$

формула принимает вид

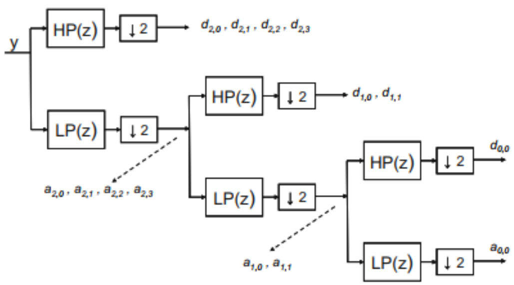
$$a_j[n] = \sum_{m\in\mathbb{Z}} \tilde{a}_{j-1}[m]\; h[n-m] + \sum_{m\in\mathbb{Z}} \tilde{d}_{j-1}[m]\; g[n-m] = \tilde{a}_{j-1} * h + \tilde{d}_{j-1} * g.$$

Или же, схематично,

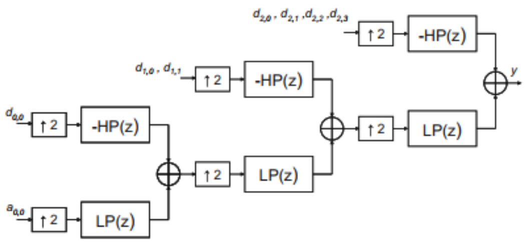
$$\left.\begin{array}{l} a_{j-1} \xrightarrow{\uparrow 2} \tilde{a}_{j-1} \longrightarrow h * \tilde{a}_{j-1} \\ a_{j-1} \xrightarrow{\uparrow 2} \tilde{d}_{j-1} \longrightarrow g * \tilde{d}_{j-1} \end{array}\right\} \xrightarrow{+} a_j$$

Быстрые алгоритмы: схема

Тогда разложение по ДВП соответствует схеме



А схема восстановления:



Быстрые алгоритмы: пример Хаара

Рассмотрим схему пересчёта коэффициентов на примере всплесков Хаара. В этом случае

$h[0] = h[1] = \dfrac{1}{\sqrt{2}}$ и $g[0] = \dfrac{1}{\sqrt{2}}$, $g[1] = -\dfrac{1}{\sqrt{2}}$, остальные коэффициенты равны нулю.

$$a_{j-1}[k] = \sum_{n\in\mathbb{Z}} \overline{h[n]}\, a_j[n+2k] = \frac{1}{\sqrt{2}} a_j[2k] + \frac{1}{\sqrt{2}} a_j[2k+1],$$

$$d_{j-1}[k] = \sum_{n\in\mathbb{Z}} \overline{g[n]}\, a_j[n+2k] = \frac{1}{\sqrt{2}} a_j[2k] - \frac{1}{\sqrt{2}} a_j[2k+1].$$

Восстановление имеет вид

$$a_j[2k] = \frac{1}{\sqrt{2}} a_{j-1}[k] + \frac{1}{\sqrt{2}} d_{j-1}[k],$$

$$a_j[2k+1] = \frac{1}{\sqrt{2}} a_{j-1}[k] - \frac{1}{\sqrt{2}} d_{j-1}[k].$$

Если сигнал имеет конечное число значений, например $N = 8$, то на первом уровне разложения останется 4 приближающих коэффициента и 4 коэффициента будут отвечать за детали. На следующем уровне разложения останется 2 приближающих коэффициента и 2 коэффициента будут отвечать за детали на этом уровне и т.д.

Сложность быстрых алгоритмов

Подсчитаем количество операций, требуемых для подсчёта всплеск-преобразования по быстрым алгоритмам, то есть по формулам

$$a_{j-1}[k] = \sum_{n\in\mathbb{Z}} \overline{h[n]}\, a_j[n+2k],$$

$$d_{j-1}[k] = \sum_{n\in\mathbb{Z}} \overline{g[n]}\, a_j[n+2k].$$

Будем считать, что фильтры h, g имеют N коэффициентов. А общее число отсчётов в сигнале равно $M = 2^J$.

Для разложения одного уровня требуется подсчитать 2^J коэффициентов. На каждый коэффициент требуется N умножений.

На следующем уровне остаётся 2^{J-1} коэффициентов и т.д. Тогда число операций

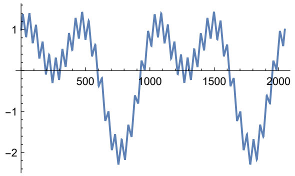
$$O = 2^J N + 2^{J-1} N + \ldots + 2N =$$

$$M N \left(1 + 1/2 + \ldots + 2^{-J+1}\right) = O(M).$$

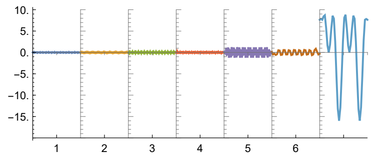
Таким образом, сложность линейно зависит от длины входной последовательности.

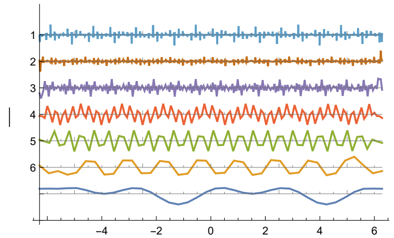
Пример всплеск-анализа сигнала

График сигнала:

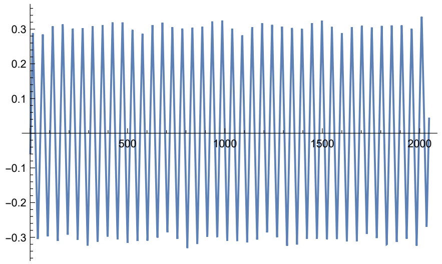


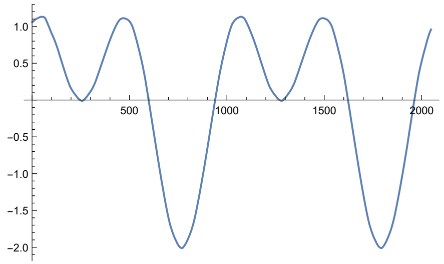
Графики коэффициентов ДВП





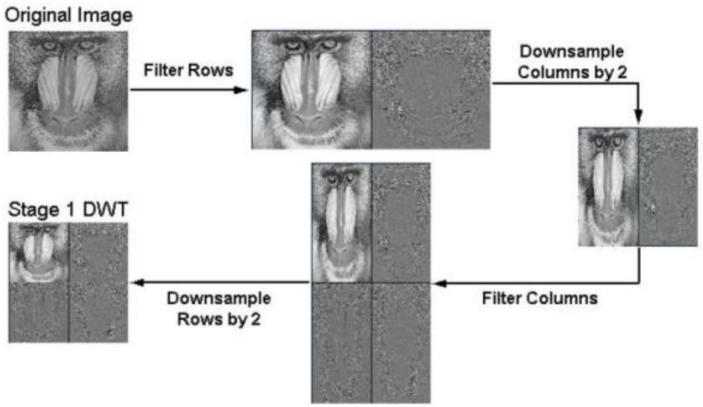
Восстановление по деталям и по приближению





Пример всплеск-анализа изображения

Пример разложения изображения по системе всплесков.



Алгоритм JPEG 2000

Напомним основные шаги алгоритма сжатия изображений в формата JPEG для серых BMP-изображений.

Для цветных изображений сначала изображение преобразуется в оптимальное цветовое пространство. Известно, что в случае применения цветового пространства яркость/цветность (YCbCr) достигается лучшая степень сжатия. Кроме того, это изображение в этом цветовом пространстве лучше соответствует человеческой визуальной системе.

Кодирование в формате JPEG состоит из следующих этапов:

1. Изображение локально преобразуется с помощью DCT (дискретного косинусного преобразования).
2. Полученные коэффициенты со значениями ниже некоторого порога отбрасываются (thresholding).
3. Оставшиеся коэффициенты кодируются эффективным образом (Huffman code).

Алгоритм JPEG 2000

Алгоритм JPEG2000 разработан той же группой экспертов в области изображений, что и JPEG. Формирование JPEG как международного стандарта было закончено в 1994 году. Вскоре стало ясно, что необходим новый, более гибкий и мощный стандарт, который и был доработан к 2000 году.

Основные отличия алгоритма в JPEG2000 от алгоритма в JPEG заключаются в следующем:

1) Лучшее качество изображения при большей степени сжатия. Заметное уменьшение размеров графики “Web-качества” при том же визуальном качестве.

2) Основной алгоритм сжатия заменен на дискретное всплеск-преобразование (ДВП). Преобразуется всё изображение, что избавляет от 8-пиксельной блочности, возникающей при повышении степени сжатия. Кроме того, просто организуется плавное проявление изображения, от размытого к более чёткому.

3) Поддержка сжатия без потерь.

4) Поддержка сжатия однобитных (2-цветных) изображений. Сжатие с использованием ДКП весьма неэффективно к изображениям с резкими переходами цветов.

5) На уровне формата поддерживается прозрачность.

Кроме того, на уровне формата поддерживаются включение в изображение информации о копирайте, поддержка устойчивости к битовым ошибкам при передаче, можно включать в изображение его описание, информацию для поиска и т.д.
.j2k .jp2 .jpx .j2c .jpf

Алгоритм JPEG 2000

Кодирование в формате JPEG 2000 состоит из следующих шагов:

1. Сдвиг значения среднего уровня яркости и ДВП.
2. Полученные коэффициенты со значениями ниже некоторого порога обнуляются (thresholding)
3. Оставшиеся коэффициенты кодируются эффективным образом (арифметическое кодирование)

ПЕРВЫЙ ШАГ. Сдвиг значения среднего уровня яркости достигается вычитанием 128 из значений каждого пикселя. Также производится смена цветового пространства RGB на YCrCb.

JPEG2000 не требует разбиения изображения на малые квадратные блоки, так как используемое всплеск-преобразование работает на фрагментах любого размера. Но если объем памяти, доступный кодеру для работы, меньше, чем объем памяти, необходимый для кодирования всего изображения, выполняется разбиение изображения на квадратные блоки, которые кодируются независимо друг от друга.

Всплеск-преобразование осуществляется с помощью 2-х различных наборов фильтров:

Обратимое всплеск-преобразование без потерь (целые значения в целые, всплески Ле Галла 5\3.)

Необратимое всплеск-преобразование с потерями (вещественные значения в вещественные, всплески 9\7.)

Всплески Ле Галла 5\3.

$$H = \left\{-\frac{1}{8}, \frac{2}{8}, \frac{6}{8}, \frac{2}{8}, -\frac{1}{8}\right\},\ G = \left\{-\frac{1}{2}, 1, -\frac{1}{2}\right\}$$

$$H^d = \left\{\frac{1}{2}, 1, \frac{1}{2}\right\},\ G^d = \left\{-\frac{1}{8}, -\frac{2}{8}, \frac{6}{8}, -\frac{2}{8}, -\frac{1}{8}\right\}$$

Всплески Cohen-Daubechies-Feauveau 9\7.

$$H = \{0, -0.091, -0.058, 0.591, 1.115, 0.591, -0.058, -0.091, 0\},$$

$$G = \{0.026, 0.016, -0.078, -0.266, 0.603, -0.266, -0.078, 0.016, 0.026\}$$

$$H^d = \{0.026, -0.016, -0.078, 0.266, 0.603, 0.266, -0.078, -0.016, 0.026\},$$

$$G^d = \{0, 0.091, -0.058, -0.591, 1.115, -0.591, -0.058, -0.091, 0\}.$$

Алгоритм JPEG 2000

Для того, чтобы преобразование можно было применять к крайним пикселям изображения, оно симметрично достраивается в обе стороны на несколько пикселей. В худшем случае (сжатие с потерями) нам необходимо достроить изображение на 4 пикселя.

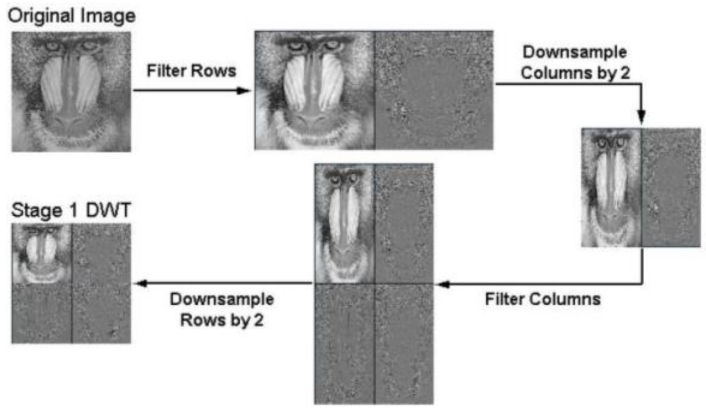
К строкам изображения применяется ДВП и формируется строка коэффициентов. При этом в начале строки размещаются коэффициенты a_{j-1} — это "сглаженная копия" всей строки (низкочастотная составляющая), а в конце строки размещаются коэффициенты d_{j-1} — это информация о колебаниях значений промежуточных пикселей (высокочастотная составляющая).

Это преобразование применяется сначала ко всем строкам изображения, а затем ко всем столбцам изображения. В результате изображение делится на 4 квадрата.

В первом квадрате будет сформирована уменьшенная копия изображения.

В остальных трех квадратах находится высокочастотная информация.

После это преобразование повторно применяется уже только к первому квадрату изображения по тем же правилам.



Алгоритм JPEG 2000

Это позволяет получить “изображение для предварительного просмотра”, прочитав небольшой участок данных из начала файла, а не распаковывая весь файл. Иерархичность преобразования может также использоваться для плавного улучшения качества изображения при передаче его по сети.

Пока что никакого сжатия ещё не происходит. Информация в изображении реорганизована и декоррелирована, чтобы сконцентрировать энергию изображения в левом верхнем углу. Это подготовка в шагу сжатия, поскольку многие коэффициенты деталей близки к нулю.

ВТОРОЙ ШАГ. Как и в алгоритме JPEG, после всплеск-преобразования применяется квантование. Коэффициенты квадрантов делятся на заранее заданное число. При увеличении этого числа снижается динамический диапазон коэффициентов, они становятся ближе к 0, и мы получаем большую степень сжатия. Варьируя эти числа для разных уровней преобразования, для разных цветовых компонент и для разных квадрантов, мы очень гибко управляем степенью потерь в изображении. Рассчитанные оптимальные коэффициенты квантования также сохраняются для однозначной распаковки.

ТРЕТИЙ ШАГ. Полученные коэффициенты кодируются эффективным образом при помощи арифметического кодирования. Кодирование полученных округленных коэффициентов выполняется поблочно. По стандарту JPEG2000 непосредственно перед кодированием фрагменты разбиваются на достаточно малые блоки (например, размером 32x32 или 64x64) так, чтобы все блоки одного фрагмента были одинакового размера. Разбиение на блоки выполняется для того, чтобы осуществить более гибкую организацию сжатой информации для повышения помехоустойчивости и так далее.