

Стандарт JPEG2000

5 января 2023 г. 1:29

Стандарт JPEG2000 основан на дискретном вейвлетном преобразовании.

Стандарт JPEG использует дискретное косинусное преобразование, там оно применяется к блокам, на которое делится изображение. Появляются блочные артефакты, т.к. при кодировании блока практически не используется информация о других блоках (используется только DC-коэффициент предыдущего блока, т.к. каждый раз кодируется разность DC текущего блока и предыдущего)

Вейвлетное преобразование:

1. Не требует деления изображения на блоки
2. Выделяет набор различных частот из входного изображения, после чего можно квантовать этот набор, как угодно (другими словами, появляется возможность точно контролировать, какую информацию сохраняем, какую - отбрасываем из каждого поддиапазона частот)
3. Декоррелирует изображение при помощи банка фильтров (т.е. сначала делается преобразование, которое уменьшает зависимости между коэффициентами, затем происходит кодирование)
4. Позволяет осуществить прогрессивное кодирование изображения в рамках одного битового потока (можно сжать изображение без потерь, потом положить на сервер, отрезать от него, например, половину битового потока, и куда-нибудь передать, только уже в более плохом качестве)

Как работает вейвлетное преобразование:

Пусть имеется входной вектор и по два фильтра для кодера и для декодера: по одному низкочастотному и по одному высокочастотному.

Кодер:

- Осуществляется свертка входного изображения с фильтрами кодера (свертка: умножается первый коэффициент фильтра на первое значение входного вектора, второй - на второе)
- Выполняется децимация результатов свертки с шагом 2 (из результатов свертки берется каждое второе значение).

Декодер:

- Восстанавливает свертки путем вставки нулей (перед каждым значением добавляем нули)
- Выполняет свертку фильтрами декодера
- Складывает полученные векторы - должен получиться исходный вектор

Пример

1. Дискретное вейвлетное преобразование

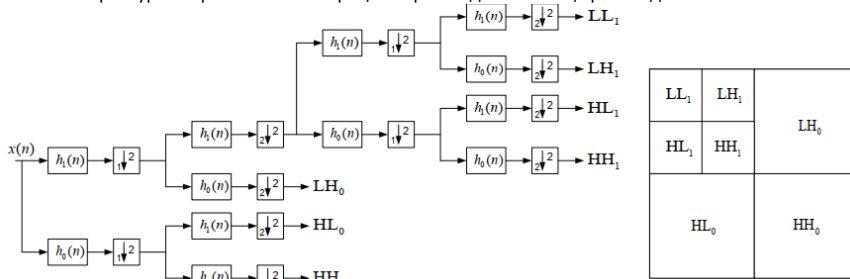
Преобразование Хаара (Haar transform). Пример.

- $h_0 = \frac{1}{\sqrt{2}} \{1, 1\}$, $h_1 = \frac{1}{\sqrt{2}} \{1, -1\}$, $g_0 = \frac{1}{\sqrt{2}} \{1, 1\}$, $g_1 = \frac{1}{\sqrt{2}} \{-1, 1\}$.
- Входной вектор $x = \{1, 2, 3, 4, 5, 6, 7, 8\}$
- Кодер:
 - ▶ Свертка фильтрами h_0 и h_1 :
 - * $y_0 = x \otimes h_0 = \frac{1}{\sqrt{2}} \{2, 3, 5, 7, 9, 11, 13, 15\}$
 - * $y_1 = x \otimes h_1 = \frac{1}{\sqrt{2}} \{0, -1, -1, -1, -1, -1, -1, -1\}$
 - ▶ Децимация y_0 и y_1 с шагом 2:
 - * $q_0 = y_0 \downarrow 2 = \frac{1}{\sqrt{2}} \{3, 7, 11, 15\}$
 - * $q_1 = y_1 \downarrow 2 = \frac{1}{\sqrt{2}} \{-1, -1, -1, -1\}$
- Декодер:
 - ▶ Восстановление y_0 и y_1 путём вставки нулей:
 - * $\hat{y}_0 = q_0 \uparrow 2 = \frac{1}{\sqrt{2}} \{0, 3, 0, 7, 0, 11, 0, 15\}$
 - * $\hat{y}_1 = q_1 \uparrow 2 = \frac{1}{\sqrt{2}} \{0, -1, 0, -1, 0, -1, 0, -1\}$
 - ▶ Свертка фильтрами g_0 и g_1 :
 - * $z_0 = \hat{y}_0 \otimes g_0 = \frac{1}{2} \{3, 3, 7, 7, 11, 11, 15, 15\}$
 - * $z_1 = \hat{y}_1 \otimes g_1 = \frac{1}{2} \{-1, 1, -1, 1, -1, 1, -1, 1\}$
 - ▶ Вычисление выходного вектора:
 - * $x = z_0 + z_1 = \{1, 2, 3, 4, 5, 6, 7, 8\}$

Применительно к изображениям (т.е. не к одномерным, а двумерным данным), фильтры будут применяться сначала по строкам, потом по столбцам

JPEG2000

1. Выполняется преобразование цветового пространства (при сжатии с потерями RGB -> YCbCr, без потерь RGB -> RCT)
2. Используются низкочастотный и высокочастотный фильтры
3. На первом уровне преобразования каждая цветовая компонента раскладывается при помощи вейвлетного преобразования (свертка, децимация, применение фильтров, снова децимация - уже по столбцам) - в итоге появляется первый уровень разложения: 4 матрицы: низкочастотная LL0, высокочастотная HH0, еще LH0 и HL0.
4. На втором уровне разложения матрица LL0 раскладывается еще раз и т.д.



В стандарте используется два варианта вейвлетных фильтров: 5/3 ("пять-три") вейвлет и 9/7 вейвлет.

Используемые в стандарте преобразования

- Daubechies 5/3 wavelets (для сжатия с потерями и без потерь):

$$h_0(n) = \begin{cases} 0.75, & \text{if } n = 0 \\ 0.25, & \text{if } n = \pm 1 \\ -0.125, & \text{if } n = \pm 2 \end{cases}$$

$$h_1(n) = \begin{cases} 0.5, & \text{if } n = 0 \\ -0.25, & \text{if } n = \pm 1 \end{cases}$$

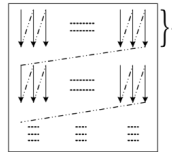
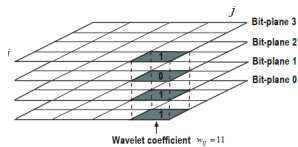
- Daubechies 9/7 wavelets (только для сжатия с потерями):

$$h_0(n) = \begin{cases} 0.602949018236, & \text{if } n = 0 \\ 0.266864118443, & \text{if } n = \pm 1 \\ -0.078223266529, & \text{if } n = \pm 2 \\ -0.016864118443, & \text{if } n = \pm 3 \\ -0.026748757411, & \text{if } n = \pm 4 \end{cases}$$

$$h_1(n) = \begin{cases} 0.557543526229, & \text{if } n = 0 \\ -0.295635881557, & \text{if } n = \pm 1 \\ -0.028771763114, & \text{if } n = \pm 2 \\ -0.045635881557, & \text{if } n = \pm 3 \end{cases}$$

После дискретного вейвлетного преобразования имеем матрицы. Каждая из них должна быть закодирована.

- Каждая матрица ДВП представляется в виде битовых плоскостей и кодируется независимо от остальных.
- Сначала, передаётся число ненулевых битовых плоскостей.
- Затем начинается кодирование от старшей значимой битовой плоскости к младшим.
- Бит с номером n каждого вейвлет-коэффициента принадлежит битовой плоскости n .



Находим самый большой коэф. по амплитуде, и он будет определять самую старшую ненулевую битовую плоскость

Каждая битовая плоскость кодируется в 3 прохода.

- I. Significance propagation pass.
 - Используется для незначимых коэффициентов с ненулевым контекстом.
 - Соответствующий бит незначимого коэффициента кодируется контекстным адаптивным арифметическим кодером. Всего используются 9 контекстов, которые определяются в зависимости от значимости соседних коэффициентов.
 - Если текущий бит коэффициента равен единице, то такой коэффициент становится значимым.
 - Знак кодируется при помощи пяти двоичных контекстов. Номер контекста определяется по знаку и значимости соседних коэффициентов.
 - II. Magnitude refinement pass.
 - Биты значимых коэффициентов кодируются при помощи трёх двоичных контекстов.
 - III. Clean-up pass.
 - Кодируется наиболее значимая битовая плоскость.
 - Кодируются биты незначимых коэффициентов, соответствующих нулевому контексту, на других битовых плоскостях.
- В результате генерируется прогрессивный битовый поток:

$$d_3 \geq d_2 \geq d_1 \geq d_0 = 0.$$

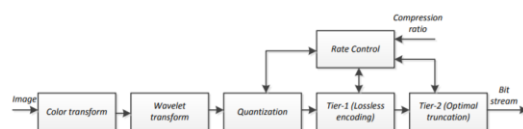
$$r_3 \leq r_2 \leq r_1 \leq r_0.$$

- Этот битовый поток может быть отсечен после каждого прохода.

Для сжатия без потерь нужно сжать и передать все плоскости.

Для сжатия с потерями можно не передавать например младшую битовую плоскость.

Общая схема кодера



Основные этапы:

- Преобразование цветового пространства;
- Дискретное вейвлетное преобразование;
- Сжатие без потерь каждой вейвлетной матрицы (Tier-1).
- Достижение заданного уровня сжатия путём выбора точек отсечения (Tier-2).

Tier-1 - арифметический кодер

JPEG2000 - на вход передается степень сжатия (compression ratio), а не Quality Factor как в обычном JPEG.

Несколько деталей еще раз:

Характеристики алгоритма JPEG-2000:

- Степень сжатия: 2-200 (Задается пользователем). Возможно сжатие без потерь.
- Класс изображений: Полноцветные 24-битные изображения. Изображения в градациях серого без резких переходов цветов (фотографии). 1-битные изображения.
- Характерные особенности: Позволяет удалять визуально неприятные эффекты, повышая качество в отдельных областях. При сильном сжатии появляется блочность и большие волны в вертикальном и горизонтальном направлениях.

