

# Формат JPEG2000: структура, методы сжатия, применение для геопространственных данных

Герасименко Евгений, НИУ ВШЭ

НУГ Геоинформатики

<http://geolab.gis.land>

# JPEG

- Сжимается блоками 8 на 8 и кодируются с помощью кодов Хаффмана.
- При сжатии (квантовании) происходит огрубление мелких деталей
- Использование косинус преобразования значительно ускоряет процесс кодирования по сравнению с вейвлет преобразованием используемого в формате JPEG2000 (в ущерб потери качества).
- Для кодирования применяется кодер Хаффмана, который в отличие от адаптивного арифметического кодера (JPEG2000) работает лучше только при частоте появления символов пропорциональной степеням двойки, что редко встречается в реальных изображениях. В арифметическом кодировании эта проблема решена присвоением кодов последовательностям символов.

# Преимущества JPEG2000

- Большой коэффициент сжатия
- Цветовое преобразование с и без потерь
- Поддерживает и сжатие с потерями, и сжатие без потерь.
- Переход от сжатия с потерями к сжатию без потерь
- Поддерживается увеличение чёткости статически и динамически определённых областей (Regions of Interest) (в основном на видео)
- Помехоустойчивость
- Фиксированное и прогрессивное (по точности и по разрешению) кодирование
- Поддерживается кодирование тайлов разных размеров
- Многокомпонентные (составные) изображения
- Трансформации на основании (по) линий и блоков
- Методы изменения сжатых изображений

# Преимущества JPEG2000 (2)

- Эффективность сжатия в среднем превосходит JPEG на 15-20% при меньших потерях качества изображения.
- Сохранена обратная совместимость с JPEG
- При значительном сжатии с потерями (артефакты появляются при значительно большем сжатии чем при сжатии файла с помощью JPEG) (вместо сжатие Хаффмана используется арифметическое сжатие), на стыках тёмных и светлых (контрастных) участков появляется тонкая светлая (яркая) полоса, но нет характерных для формата JPEG секторов 8x8.

**JPEG** at 0.125 bpp



**JPEG2000** at 0.125 bpp



**JPEG** at 0.25 bpp



**JPEG2000** at 0.25 bpp



**JPEG** at 0.125 bpp



**JPEG2000** at 0.125 bpp



# Применение в GIS

- JPEG2000 применяется для лучшего сжатия данных с потерями и без потерь.
- Позволяет хранить изображения с привязкой к геоданным.
- При сжатии изображение остаётся бесшовным.
- Возможность декодировать изображения до определённого уровня (масштабируемость качества).

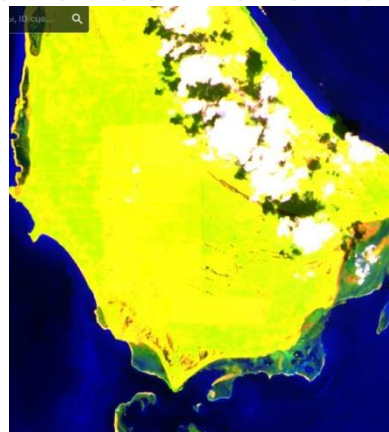
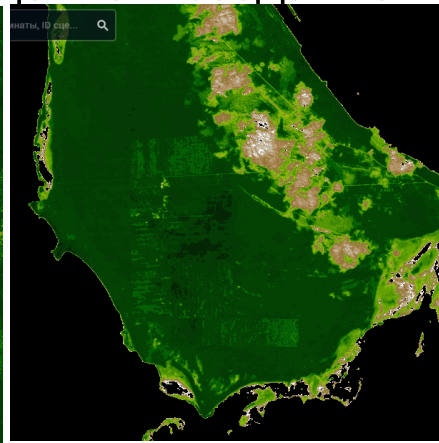
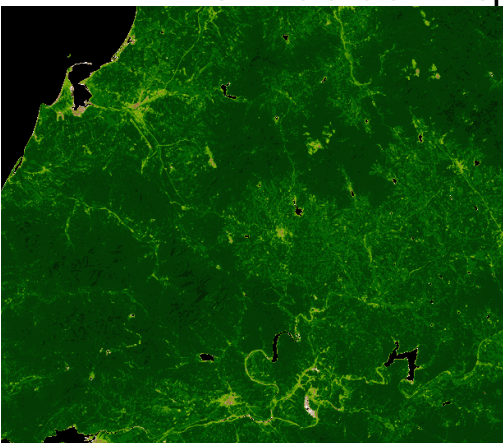
# JPEG2000 и Sentinel-2

JPEG2000 используется для хранения изображений проекта Sentinel-2(A,B).

Эти аппараты оснащены мультиспектральным сенсором для съёмок с разрешением от 10 до 60 метров в видимой, ближней инфракрасной (VNIR) и коротковолновой инфракрасной (SNIR) зонах спектра (всего 13 каналов). Публикуется 10,04 ТБ/сут. Общий объём данных 41,35 ПБ.

Данные каналы позволяют производить:

- Оценку состояния растительности
- Мониторинг чрезвычайных ситуаций
- Инвентаризацию и оценку состояния лесов и сельскохозяйственных угодий
- Точное земледелие и создание планов землепользования

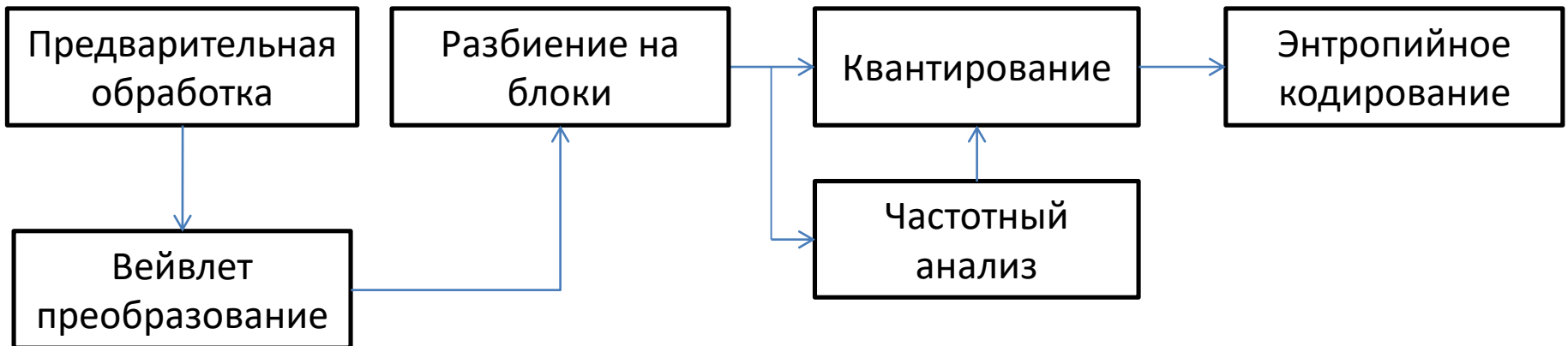


Багамские острова

# Сжатие

1. Предварительная обработка
2. ДВП
3. Сканирование
4. Квантование
5. Энтропийное кодирование

(Адаптивное арифметическое сжатие)





# Входное изображение

- Состоит из неотрицательных целых чисел (как правило)
- Многокомпонентные изображения:
  - До 256 компонент
  - Произвольные размеры ширины/битовой глубины для каждого компонента

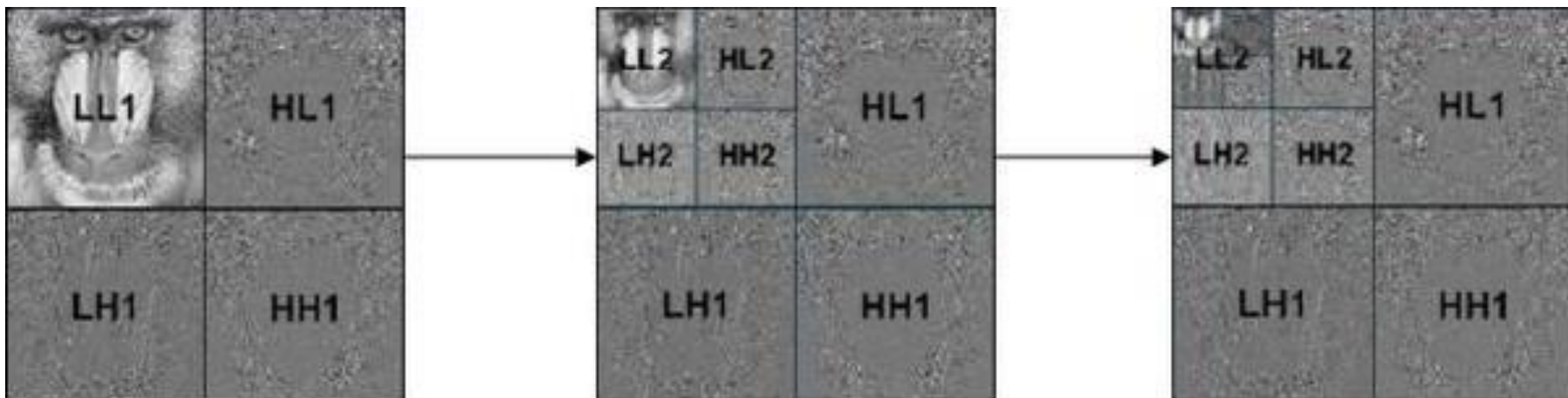
# Предварительная обработка

- Из значений в изображении вычитается среднее.
- При большом размере изображение дробится на части (тайлы имеют размеры от 256 до 4096 пикселей) (желательно максимального размера), затем каждая часть сжимается отдельно и затем они объединяются в единое изображение.

# Дискретное Вейвлет Преобразование

На каждом шаге мы применяем соответствующий фильтр (в стандарте – биортогональные фильтры (5,3) – без потерь и (9,7) – с потерями) (проходимся по столбцам и строкам) при этом применяется лифтинговая схема (а не свёртка) и получаем 4 части

Столбцы\Строки	Низкие	Высокие
Низкие	LL	HL
Высокие	LH	HH



# Биортогональный фильтр (5,3)

- Фильтры:  $\tilde{h}_n = \{-1/8, 1/4, 3/4, 1/4, -1/8\}$ ,  $\tilde{g}_n = \{1/4, -1/2, 1/4, 0, 0\}$ .

Декомпозиция производится следующим образом:

$$c_k^{1,0} = c_{2k}^0, \quad k = 0, \dots, N_1 - 1,$$

$$\begin{cases} d_k^1 = \text{int}\left(\frac{c_{2k}^0 + c_{2k+2}^0}{2}\right) - c_{2k+1}^0, & k = 0, \dots, N_1 - 2, \\ d_{N_1-1}^1 = c_{N_1-2}^0 - c_{N_1-1}^0, \end{cases}$$

$$\begin{cases} c_0^1 = c_0^{1,0} - \text{int}\left(\frac{d_0^1}{2}\right), \\ c_k^1 = c_k^{1,0} - \text{int}\left(\frac{d_{k-1}^1 + d_k^1}{4}\right), & k = 0, \dots, N_1 - 2, \\ c_{N_1-1}^1 = c_{N_1-2}^{1,0} - \text{int}\left(\frac{d_{N_1-2}^1 + d_{N_1-1}^1}{4}\right) \end{cases}$$

Реконструкция осуществляется по следующим формулам :

$$c_0^0 = c_0^1 + \text{int}\left(\frac{d_0^1}{2}\right),$$

$$c_{2k}^0 = c_k^1 + \text{int}\left(\frac{d_{k-1}^1 + d_k^1}{4}\right), \quad k = 1, \dots, N_1 - 2,$$

$$c_{N_1-2}^0 = c_{N_1-1}^1 + \text{int}\left(\frac{d_{N_1-2}^1 + d_{N_1-1}^1}{4}\right),$$

$$c_{2k+1}^0 = \text{int}\left(\frac{c_{2k}^0 + c_{2k+2}^0}{2}\right) - d_k^1, \quad k = 0, \dots, N_1 - 2,$$

$$c_{N_1-1}^0 = c_{N_1-2}^0 - d_{N_1-2}^1.$$

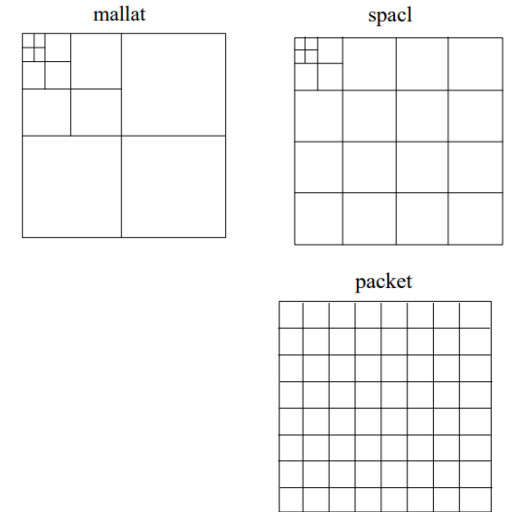
# Дискретное Вейвлет преобразование (2)

Поддерживаются:

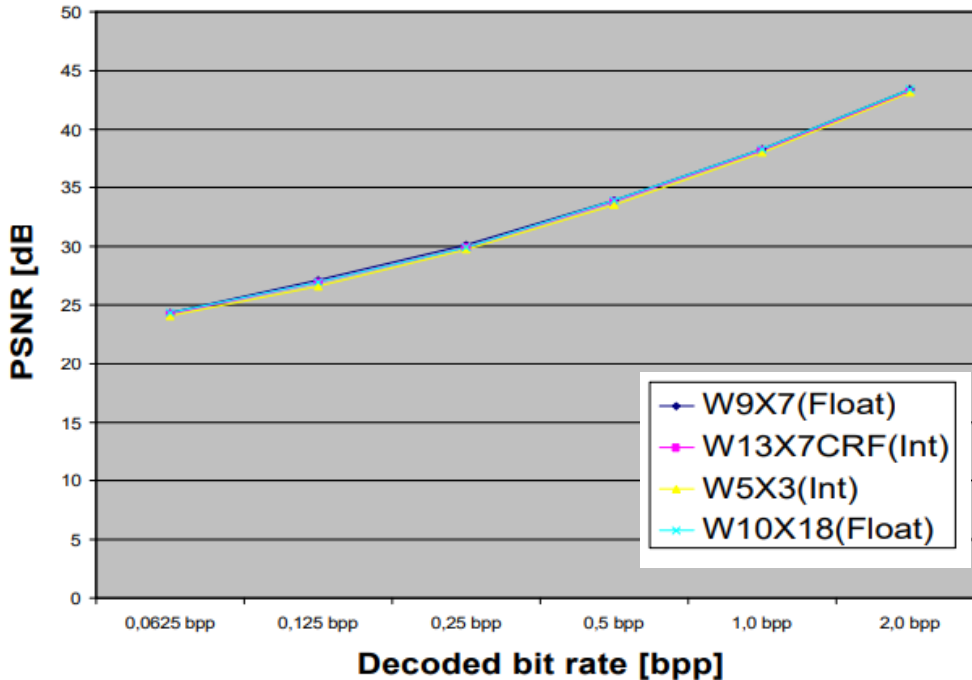
- Произвольные вейвлет разложения
- Произвольные вейвлет ядра
  - Ядра могут отличаться для каждого направления и для каждого уровня разложения
- Изображения с произвольной битовой глубиной
  - Везде от 1 до 32 бит на образец (знаковый и без знаковый)

# Поддерживаемые фильтры

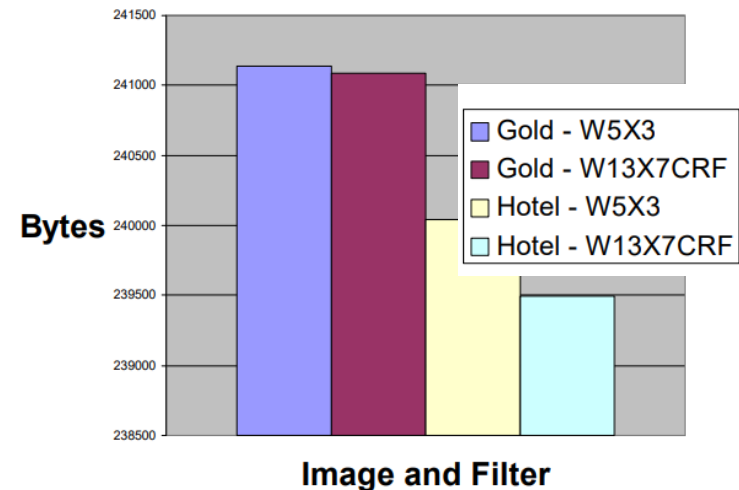
- Вейвлеты с плавающей точкой: **(9,7)** (10,18)
- Целочисленные: (13,7), CRF(13,7), **(5,3)**, (2,10)
- По умолчанию целочисленное (с потерями): CRF(13,7)
- По умолчанию целочисленное (без потерь): (5,3)
- (Пользовательские фильтры)



PSNR [dB] - Filter Comparison (Hotel)



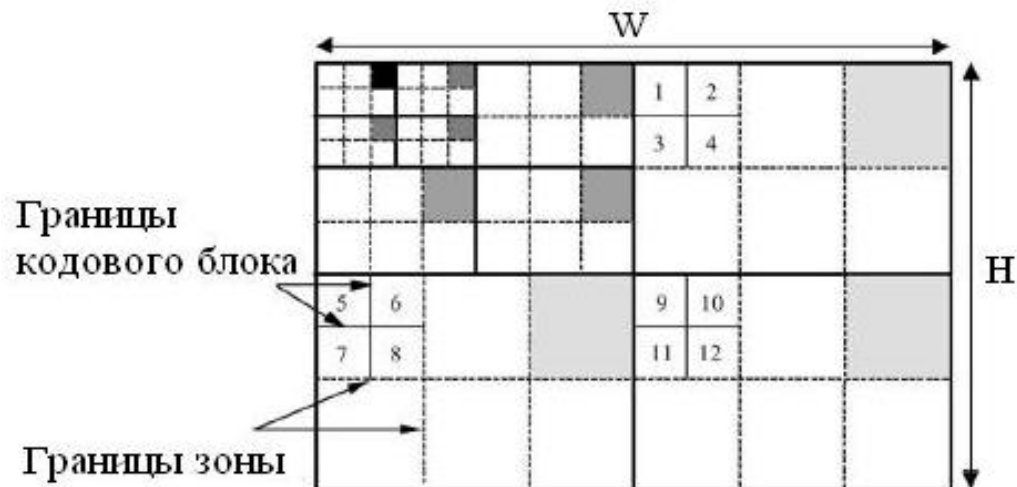
Lossless Results



# Разбиение на блоки

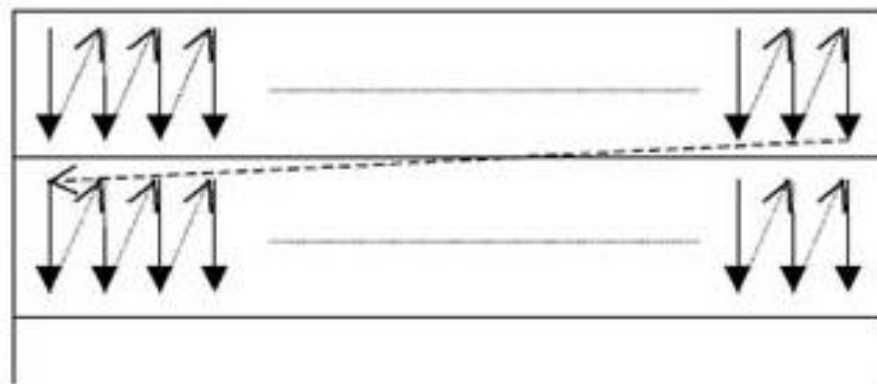
(Сканирование)

В JPEG2000 каждый уровень разрешения вейвлет-разложения разбивается на зоны и кодовые блоки (кодовые блоки образуются в результате разбиения подполос). Каждый из кодовых блоков обрабатывается (и квантуется) независимо друг от друга.



[http://www.autex.spb.su/download/wavelet/jpeg2000/jpeg2000\\_rabbani.pdf](http://www.autex.spb.su/download/wavelet/jpeg2000/jpeg2000_rabbani.pdf)

0	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60
1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61
2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62
3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63
64	...														
65	...														



# Квантирование

- Применяется равномерный квантователь (uniform quantizer) с мёртвой зоной.
- При сжатии без потерь шаг равен 1, при сжатии с потерями шаг зависит от необходимой степени сжатия.
- Шаг квантователя постоянен в пределах субполосы.
- Во второй части стандарта допустимо применение решётчатого квантователя (ТСQ).



# Энтропийное кодирование

- Применяется адаптивный арифметический кодер (из-за патентных ограничений используется не QM кодер (IBM) а MQ кодер).
- Кодирование происходит по кодируемым блокам. Максимальный размер блока не больше 4096 пикселей, высота не менее 4 пикселей. При этом снижается коэффициент сжатия, но повышается устойчивость сжатого потока к ошибкам канала связи (из-за малых размеров блоков). Блоки кодируются в три этапа битовыми плоскостями.



# Почему Энтропийное кодирование

- Использование локальных вариаций от блока к блоку
- Доступ к случайной области изображения
- Оптимизация использования памяти при сжатии и распаковке
- Возможность осуществления параллельных вычислений

# Энтропийное кодирование (2)

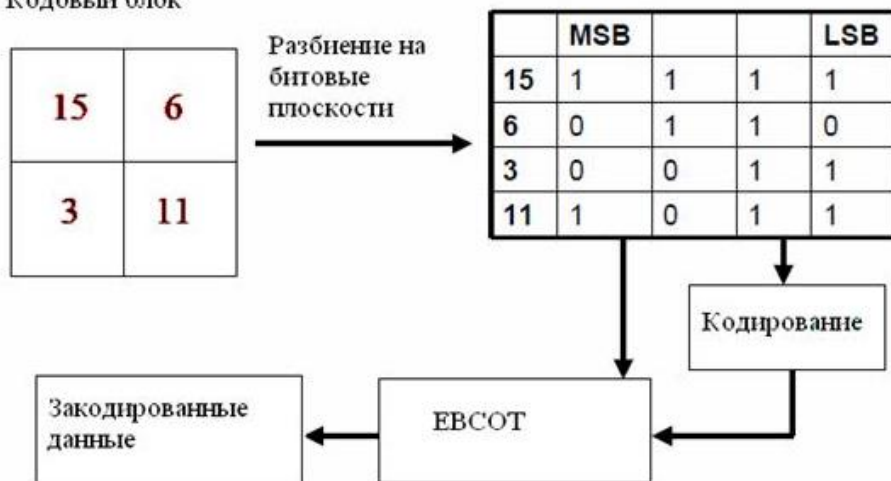
## Особенности

- Каждая субстрока разбивается на блоки
- Все блоки в пределах субстроки имеют одинаковый размер
- Блоки кодируются независимо в 3 прохода
  - Кодирование старших бит
  - Уточняющий проход
  - Очищающий проход
- Постобработка определяет степень усечения для каждого блока
- Итоговый битовый поток состоит из набора слоёв

# Типы операций кодирования

- Zero Coding (ZC)
- Run-Length coding (RLC)
- Sign coding (SC)
- Magnitude refinement (MR)

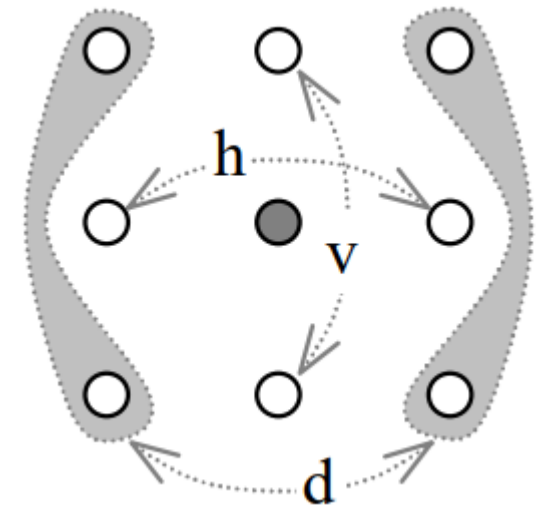
Кодовый блок



# Zero Coding

Используется 1 из 9 различных состояний окружения (соседей) для кодирования значения символа в зависимости от переменных состояния значимости

- Непосредственные горизонтальные соседи (h)
- Непосредственные вертикальные соседи (v)
- Непосредственные диагональные соседи (d)
- Не ближайшие соседи (f)



# Run-Length coding

- Используется в сочетании с примитивом ZС, чтобы уменьшить число двоичных символов, которые требуется закодировать с помощью арифметического кодирования
- Если образец ещё не является значимым, то сочетание ZС и RLC используется для кодирования значения символа в текущей битовой плоскости

# Sign coding

- Используется не более одного раза для каждого образца в блоке сразу же обнаруживается, что ранее незначимый символ является во время операций кодирования ZC и RLC
- Если это выполняется, то SC также должен быть вызван для отправки (кодирования) знака

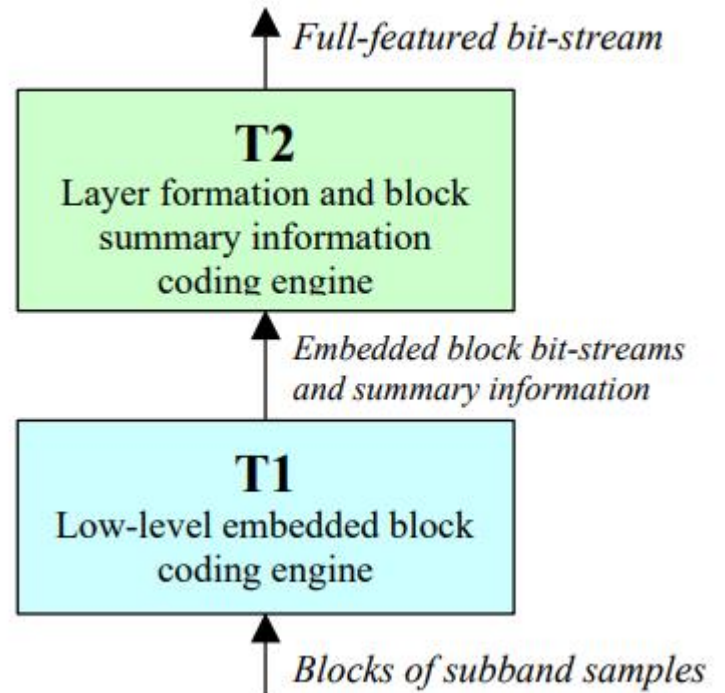
# Magnitude refinement

- Используется для кодирования уже значимого образца
- Если образец уже значим, то MR используется для кодирования новой битовой позиции



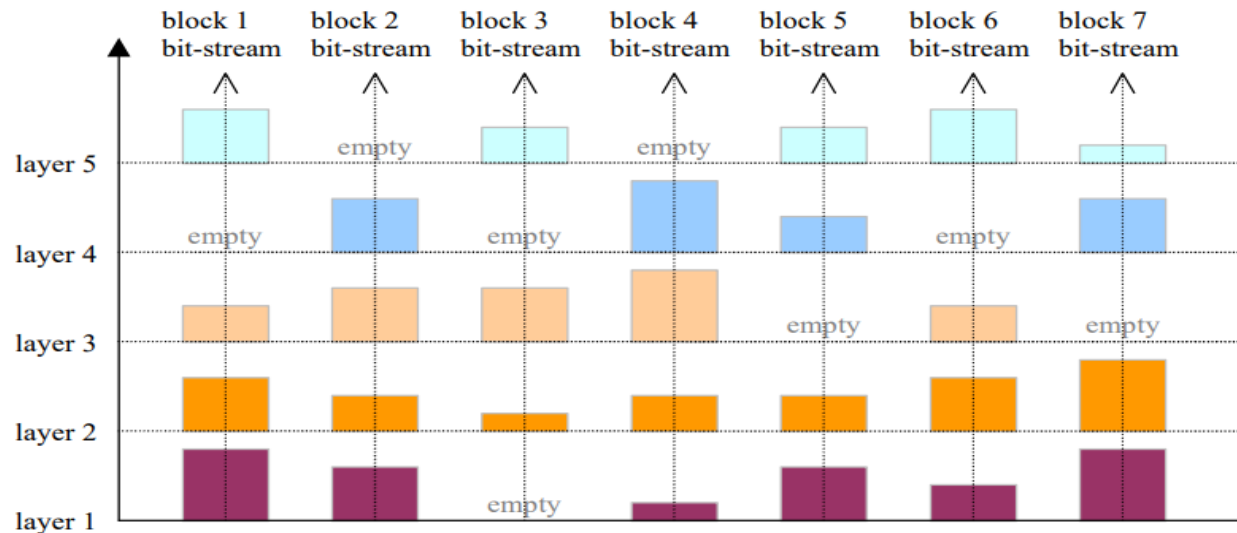
# Энтропийное кодирование (3)

- Формирование кодовых потоков из слоёв (пакетизация)
- Создание блочных битовых потоков (кодирование битовых плоскостей с помощью арифметического кодера)



# Структура потоков слоёв

- Каждый битовый поток организован, как последовательность слоёв
- Каждый слой содержит дополнительные вклады (вбросы) из каждого блока (возможно пустые)
- Точки усечения блоков, связанные с каждым слоем, являются оптимальными по искажениям скорости
- Оптимизация искажений скорости выполняется, но не должна стандартизироваться



# Структура файла JPEG2000

Кроме того, JPEG2000 выдает в результате работы кодера второго уровня сжатые данные из кодовых блоков в компонентах, называемых пакетами и слоями. Для каждой зоны сжатые данные из кодового блока сначала группируются в один или более пакетов (packet). Пакет представляет собой непрерывный сегмент в сжатом битовом потоке, который состоит из определенного количества кодовых проходов из каждого кодового блока в зоне [Rabban].

Заголовок пакета	n0 фрагментов битовой плоскости кодового блока 0	.....	n11 фрагментов битовой плоскости кодового блока 11
------------------	--	-------	--

Пакет состоит из тела [Marcellin] и заголовка. В заголовке содержится следующая информация: данные о каждом блоке в пакете, число нулевых битовых плоскостей для каждого блока, число фрагментов битовой плоскости, включенных в кодовый блок, количество бит, необходимых для хранения подполос и битовых плоскостей. Для корректного декодирования в каждом пакете содержится информация о предыдущих пакетах (такое представление и позволяет осуществить прогрессивное декодирование изображения).

# Структура файла JPEG2000 (2)

- Затем пакеты из каждой зоне на каждом уровне разложения и из каждого тайла группируются в более крупные единицы, называемые слоями (layer). На иллюстрации показано, как формируется структура кодового потока [V. Sanchez & A. Basu].
- Таким образом, пакет – это совокупность закодированных кодовых блоков, слой – совокупность пакетов, а битовый поток всего изображения – совокупность слоев, а в выходной поток передается двоичный код изображения (с учетом результатов кодирования каждой битовой плоскости) и кривая искажения уровня (R-D), позволяющая оценить производительность сжатия.



# Структура файла JPEG2000 (2)

Порядок, в котором пакеты появляются в битовом потоке, называется прогрессивным (progression order) и контролируется с помощью специальных маркеров. Для обрабатываемого тайла, необходимо задать четыре параметра для однозначного определения пакета: компонент (component), разрешение (resolution), слой (layer) и позиция (зона, precinct).

Базовая часть JPEG2000 предоставляет несколько вариантов появления пакетов:

- слой разрешение компонент позиция (LRCP);
- разрешение слой компонент позиция (RLCP);
- разрешение позиция компонент слой (RPCL);
- позиция компонент разрешение слой (PCRL);
- компонент позиция разрешение слой (CPRL)