**Лекция 1.**

**Лекция 2.**

**Лекция 3.**

Методы сжатия с частичной потерей качества оригинала.

Используется три основных избыточности:

1. Пространственная избыточность (внутри кадра/изображения между соседними пикселями)
2. Временная - //- (от кадра к кадру)
3. Цветовая (внутри кадра по количеству представляемых цветов)

Методы сжатия:

А) работают только со статичными изображениями(JPEG)

Б) только с видео (кросс-кадровая интерполяция)

В) универсальные

Кодек cinepac – устранение избыточности через векторную квантизацию, через квантовые дельты кадров и через палитрование изображения. Появился в 90-х годах, сжимает не очень, относится к методам с потерями качества, довольно прост.

Устранение избыточности через **векторную квантизацию**: разбиваем изображение на блоки (например, 16х16 пикселей), ищем среди них допустимо (разница должна быть не выше установленной планки качества) похожие по какой-то метрике. Изображение будет состоять из карты блоков, где допустимо похожие блоки группируются по номерам. После этого в выходной файл пишется кодовая книга (где пишется непосредственно растровой содержание этих блоков) и таблица с пронумерованными блоками:

таблица с пронумерованными блоками:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 4 | 6 | 4 |
| 5 | 5 | 1 | 6 | 2 |
| 2 | 4 | 3 | 2 | 2 |
| 4 | 1 | 5 | 1 | 1 |
| 3 | 6 | 2 | 1 | 3 |

кодовая книга:

|  |  |
| --- | --- |
| 1 | XX  XX |
| 2 | QQ  QQ |
| 3 | SS  SS |
| 4 | HH  HH |
| 5 | LL  LL |
| 6 | PP  PP |

Блоки в итоге усредняются, в зависимости от заданной планки качества при компрессии. Так работает векторная квантизация.

**Квантование дельты кадров:**

Берется дельта между соседними кадрами А и Б

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 255 | 254 |  | 255 | 250 | => | 0 | -4 |
| 254 | 252 |  | 254 | 255 |  | 0 | 3 |

Кусочек кадра А, кадра Б и разница между ними (вычитается последующий кадр из предыдущего).

Получаем некую дельта, которая после квантуется с заданным ранее шагом.

Пусть, например, шаг = 5, это значит, что округление до ближайшего будет приводиться к пятерке и кратным ей. Есть виды квантования с динамическим шагом(не в cinepak). А в выходной поток тогда запишется 0 -5 0 5. Потом, если сжимать по Хаффману, будет значительно проще, потому что будут только числа, кратные 5.

**Палитрование изображений**

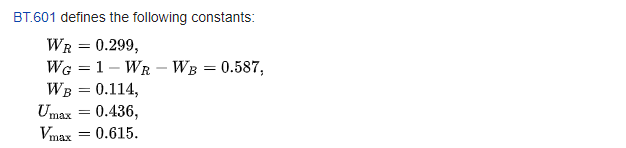
Полноцветный RGB переводится, как правило, в палитру из 256 цветов по критерию наиболее часто встречающихся на изображении. В сжатый файл пишем только кодовую книгу палитры – номер палитрованного цвета и его значение.

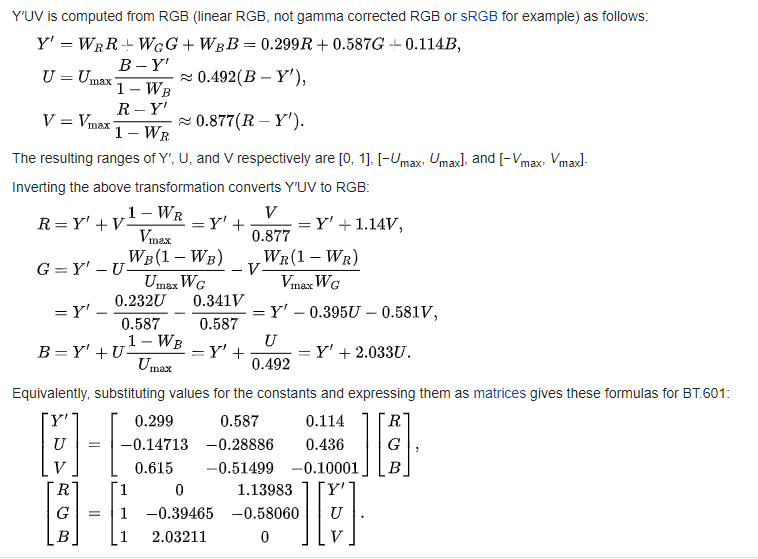
Есть другие идеи, например, контур основанного кодирования и прочие.

**Метод JPEG.**

**JPEG – Joint Picture Expert Group**, разработка – конец 80-х – начало 90-х (официально: 1992). Призван найти золотую середину между качеством и объемом. Разработан специально для 24-битных изображений. Работает следующим образом:

1. сначала осуществляется переход в другое цветовое пространство(RGB->YUV, или Y Cb Cr). YUV – разработано в 30-х годах, пришло из телевидения. Y – яркость для ч/б телевизоров, в них одна компонента – яркость. Но позже, когда появились цветные телевизоры, для обратной совместимости, чтобы сигнал подходил и для цветных, и для чб, ввели две цветоразностные компоненты U, V(Cb, Cr – хроматический синий/красный). Правила их перевода – с помощью матрицы перехода**(более подробно – у Ватолина, раздел 2, стр.52)**. Переход нужен для того, что, поскольку данный метод с потерей качества, мы можем цветовое сжатие осуществить за счет того, что человеческий глаз менее чувствителен к цвету, чем к яркости, то можно сохранить с бОльшими потерями и выиграть на сжатии. Есть 2 стандарта перехода в данную матрицу – телевизионный и компьютерный – REC.601(BT.601) и PC.601. Можно представить с помощью матрицы (у Ватолина), а можно – с помощью формул(<https://en.wikipedia.org/wiki/YUV>)





Стандарт определяет 3 константы: весовую для красного цвета, синего и зеленого, а также 2 взвешенные компоненты для получения двух цветоразностных компонент.

Формула для яркости: **0.3красного+0.6зеленого+0.1синего** – используется не только здесь, но и для вывода изображений на монохромный экран, при печати на чб принтерах и любых операциях при конвертации полноцветного изображения в оттенки серого

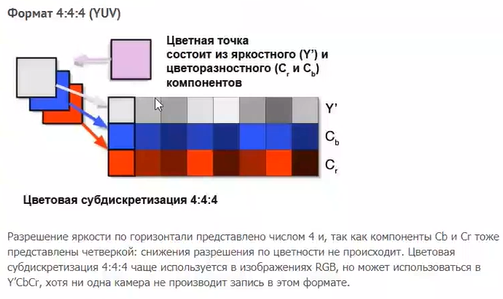
Две цветоразностные компоненты – определяются как разности (из синего/красного вычитаем яркость).

Нет цветоразностной компоненты по зеленому, тк формула для яркости подобрана таким образом, что максимум по яркости человеческого глаза как приходится на область зеленого цвета. Поэтому цветовой коэффициент в яркости у зеленого максимален, и нет нужды хранить зеленый (середину спектра) отдельно.

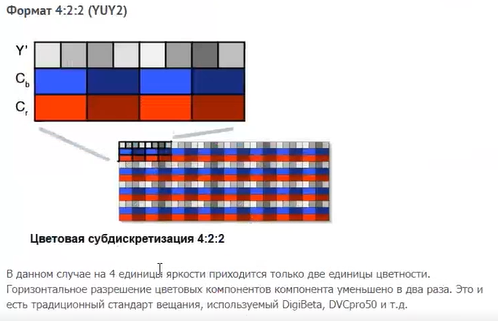
Формула для BT.601 приводит в диапазон [0,1]. Для компьютера удобнее пользоваться целочисленным.

1. Цветовое преобразование (**см. Ватолин**). Проредить цвета можно по вертикали, по горизонтали, соединить по 4 и тд – существует много видов цветового прореживания.

Например, полный вид(когда прореживания нет) – не происходит снижение разрешения по цветности:

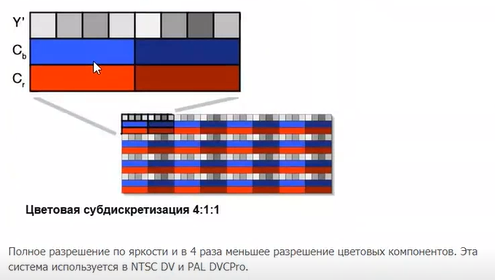


Если оно происходит, например, один из распространенных форматов 4:2:2 – здесь на один отсчет по горизонтали по яркости приходится один усредненный отсчет цветоразностных компонент.

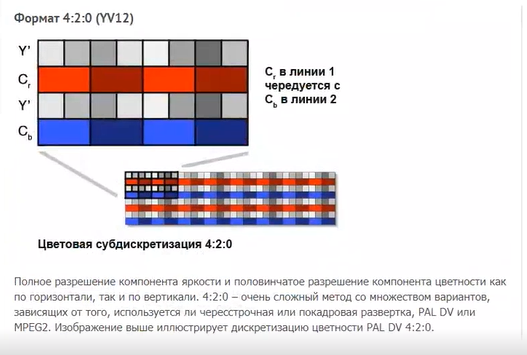


Часто встречается в MPEG, современных стандартах вещания, хорошо распространен.

4:1:1 – когда на 4 отсчета по яркости идет по одному отсчету цветоразностных компонент.

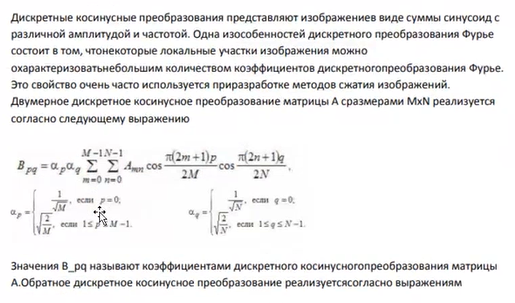


В аналоговом ТВ используется чересстрочная подача цветов. Это записывается как 4:2:0, тк в ТВ используется построение кадра за два прохода(четные/нечетные строки). 0 говорит о том, что на каждый отсчет по яркости происходит нелинейно(?).

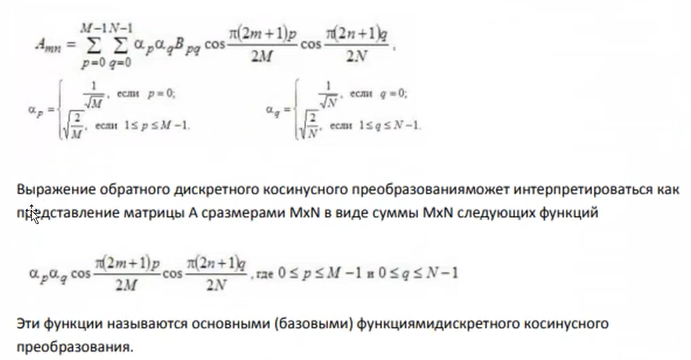


В стандарте JPEG нет четкого стандарта прореживания.

1. Разбиение на матрицы 8х8 или 16х16.
2. Применение ДКП – аналог дискретного преобразования Фурье



По формуле: сумма дискретных коэффициентов – косинусов, внутри косинусов m,n – исходные значения коэффициентов, на выходе также получаем матрицу 8х8.



Зачем это делать? Разлагая исходный сигнал, переходим в частотную плоскость. Это нужно для того, чтобы в (???)

**В итоге получаем матрицу, в которой коэффициенты усреднены с точки зрения частот – отсортированные по частоте коэффициенты в матрице:**

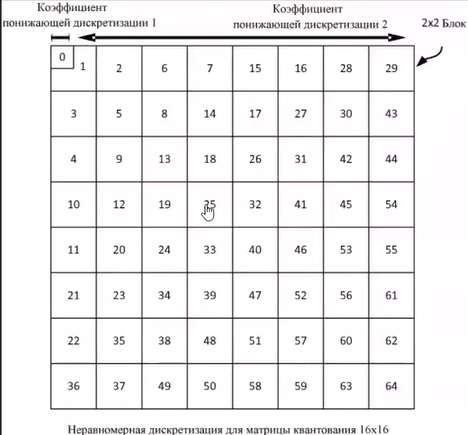
|  |
| --- |
| **Нижние частоты**  **Средние частоты**  **Высокие частоты** |

**Нижние частоты, в т.ч., первый член ряда – это средняя яркость блока. Ее терять не хочется, поэтому нижние частоты огрубляются(сжимаются) несильно.**

**Средние частоты можно огрубить сильнее, а верхние – почти полностью занулить. За счет этого мы избавимся от мелких деталей изображения.**

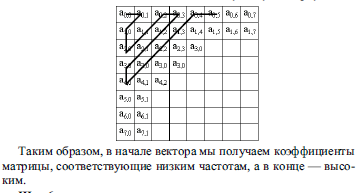
Если исходная матрица содержала в себе просто фрагмент изображения, то теперь этот фрагмент усреднен по частотам. Как в ряде Фурье: сначала идут члены, которые определяют основной рисунок (основное приближение аппроксимируемых функций), потом уточняющие, и уточняющие отдельные участки сигнала.

1. Квантование – деление рабочей матрицы на матрицу квантования поэлементно. Пример матрицы квантования (но для каждого коэффициента сжатия она своя)

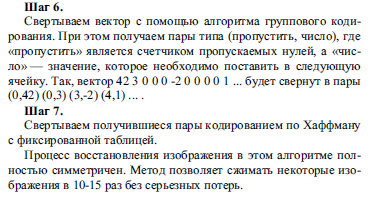


В данной матрице возрастают числа от верхнего левого до правого нижнего угла. -> при делении исходной матрицы коэффициентов ДКП на данную и последующем огрублении верхние значения не пострадают, средние – не сильно, а нижние – сильнее всего.

1. Зигзаг-сканирование: Полученную матрицу переводим в вектор следующим способом:



1. Применяем RLE (модифицированный вариант) для получения пар типа (пропустить; число) (????). Далее используем либо Хаффмана, либо арифметическое кодирование(чаще).

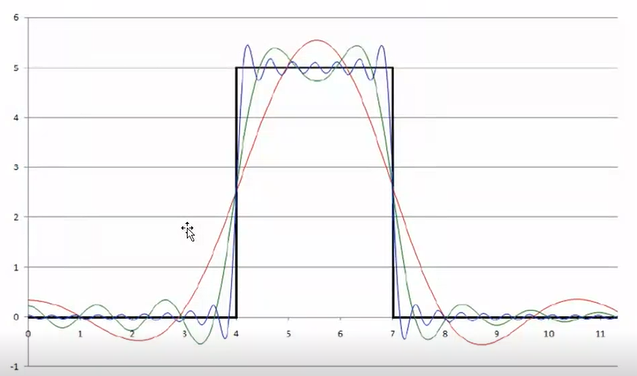


Симметричность – примерно 1 к 1, то есть, при декомпрессии примерно время то же самое, как и на компрессию.

Артефакты JPEG: (объяснение лабы с ~50той минуты)

1. Цветовое обеднение – в момент цветового прореживания – см.пункт 2.
2. Потеря мелких деталей (в лабе оба пункта объяснить, почему)
3. Проявление эффекта Гиббса – связан с тем, когда мы приближаем сигнал с помощью периодических функций прямоугольный, пилообразный сигналы

При попытке разложить такой сигнал в ряд Фурье, будет проблемы с черными краями(как их четко обрисовать), поэтому на границах с резким переходом цветов идет потеря качества, так и то, что потом быстро занулить осциллирование периодических сигналов



Эффект Гиббса – осцилляция(колыхание) периодического сигнала после перехода от полосы пропускания к полосе поглощения.

Приводит к определенным паразитарным эффектам на изображениях.

1. Блочная мозаичность. Поскольку блоки сжимаются независимо друг от друга, у одного происходит округление цветов в одну сторону, а у соседнего – на пару тонов в другую, и визуально начинают проявляться квадратики.

Изображения хранятся в тифф – формат без потерь качества.

Если меняется коэффициент сжатия – меняются значения в матрице квантования. Чем он выше, тем больше значения в матрице коэффициентов.

В сжатом изображении ЛР1Комплекс:

Градиентные переходы – раньше цвета были гладкими, теперь, в силу блочного усреднения (каждый из блоков имеет свою среднюю яркость), и стали проявляться переходы (возле кусков ткани) - Блочная мозаичность.

Потеря мелких деталей

Потеря цветов

Эффект Гиббса – все цифры стали размазаны, и у них появился двойной контур – косинусы, который должны перейти резко от черного к белому, они этого сделать не могут, поэтому грани начинают размываться и грязниться (на линейке)

В сжатом изображении ЛР1Тестовая:

Эффект Гиббса – все цифры стали размазаны, и у них появился двойной контур – косинусы, который должны перейти резко от черного к белому, они этого сделать не могут, поэтому грани начинают размываться и грязниться

Градиентные переходы – раньше цвета были гладкими, теперь, в силу блочного усреднения (каждый из блоков имеет свою среднюю яркость), и стали проявляться переходы (мишень) - Блочная мозаичность.

В сжатом изображении ЛР1Портрет:

Потеря мелких деталей

Градиентные переходы – раньше цвета были гладкими, теперь, в силу блочного усреднения (каждый из блоков имеет свою среднюю яркость), и стали проявляться переходы - Блочная мозаичность.

Каждый файл зажать тремя видами.

Привести хотя бы по одному примеру на каждый артефакт.

Подобрать тот коэффициент jpeg, при котором потери качества незаметны, и посчитать статистику.

Исходные размеры файлов без сжатия – не те, что указаны в тифф. Нужно их посчитать – длина х ширина х 3 байта х цветовая глубина (либо пересохранить файл в бмп)

Сказать, сколько весит в тиффе, и сколько тифф сэкономил места.

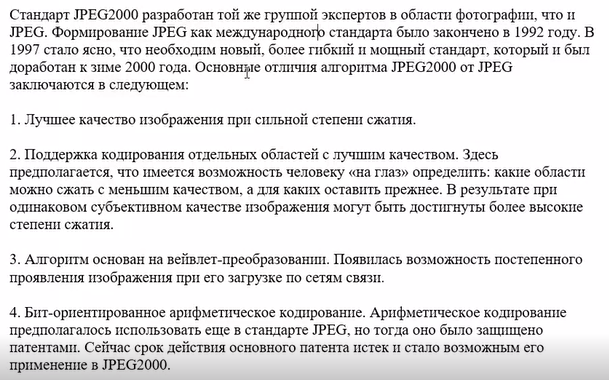
Размер сжатых в jpeg, и указать процент компрессии – отношение одного к другому.

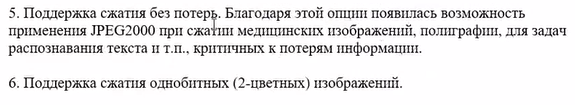
Привести в отчете оригинал портрета, кроме фрагментов.

**Лекция 4.**

**JPEG 2000**

В JPEG 2000 есть ряд нововведений, по сравнению с классическим: есть возможность адаптивного кодирования, т.е., отдельной области – мы выбираем область, например, где много мелких деталей, и кодируем ее отдельно, с лучшим качеством. Кроме того, используется другой алгоритм перехода частот на плоскость – вейвлет-преобразование. А также, если в ЖПГ на последнем шаге предполагается, что мы все еще дожимаем по Хаффману, то в ЖПГ 2000 используется однозначно арифметическое кодирование. Также есть опция сжатия без потерь. Специально адаптирован к сжатию ч/б изображений(однобитных).

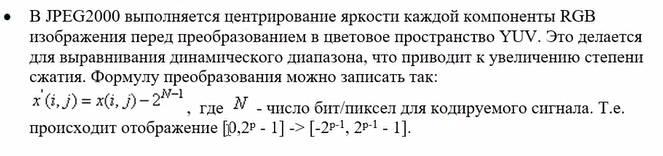




В целом, ЖПГ 2000 по крупным блокам своей работы не отличается от ЖПГ, то есть, сначала раскладываем, потом преобразовываем, применяем преобразования.

Различия:

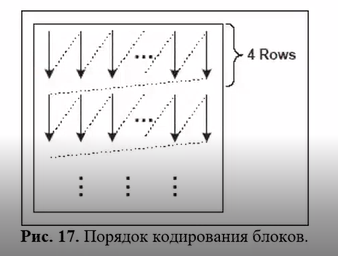
1. Производится предобработка – центрирование яркости перед тем, как перейти к цветовому пространству YUV: Если немного подвинуть значения пикселей к центру с точки зрения среднего цвета, то по битам получим экономию.



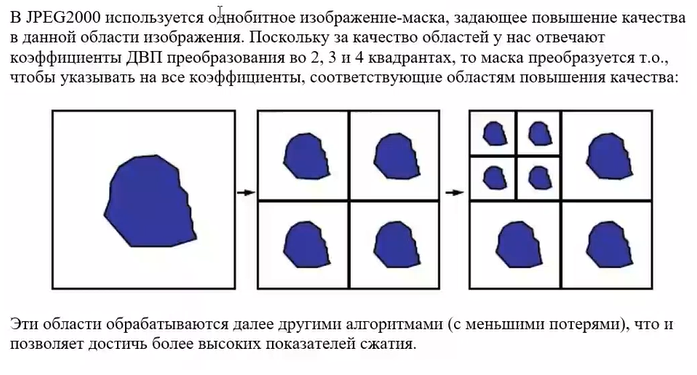
1. Вместо ДКП используется вейвлет-преобразование. (более подробно – на следующих лекциях – **см.лекция 5**).
2. После того, как разложили на частотные коэффициенты, происходит квантование. Квантователь не такой фиксированный, как в ЖПГ, - используется постоянный квантователь с мертвой зоной (трогаем интервал около нуля и зануляем его, а все коэффициенты дальше округляем более аккуратно).



1. Порядок кодирования блоков – каждый блок кодируется по отдельности. Алгоритм кодирования обходит матрицу коэффициентов округления каждого блока полосами:

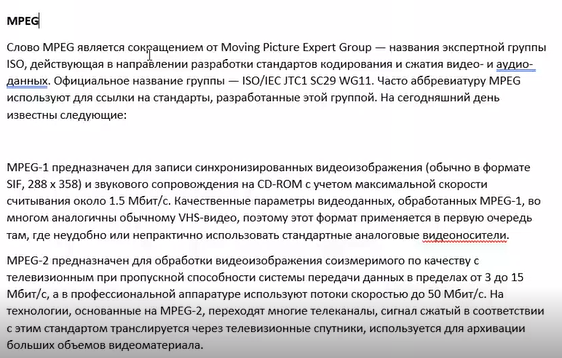


Блоки разбиваются на блоки с высотой 4. Далее полосы сканируются сверху вниз, а колонки в каждой полосе – слева направо.

1. 
2. Изображение разбивается на тайлы (непересекающиеся области), и каждый тайл представляется в виде слоев, каждый из слоев можно обрабатывать отдельно. Слои могут быть битовые. ЖПГ 2000 на порядок лучше по качеству, чем ЖПГ.

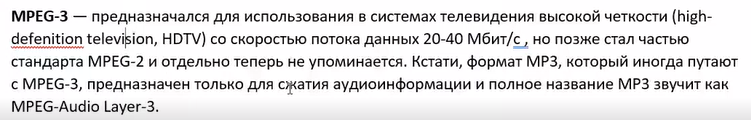
**Более подробно про ЖПГ 2000 – в файле JPEG2000V2**

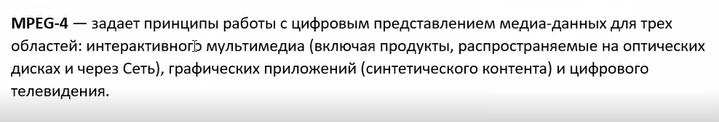
**MPEG**



**МПЕГ-1** – появился в 90-х и был предназначен для одной скорости CD-ROM-a (одна скорость передачи), т.е. компакт-диски, устройства для их чтения. Был заточен под ¼ стандартного телевизионного изображения (288 х 358). Есть стандарт Video CD – первые компакт-диски с видео.

**МПЕГ-2** – предусматривает различные качества, битрейты. Является основным форматом, в т.ч. для цифрового телевещания, также DVD – диски.



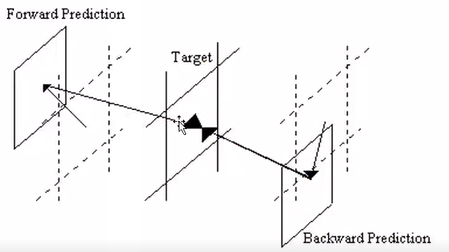


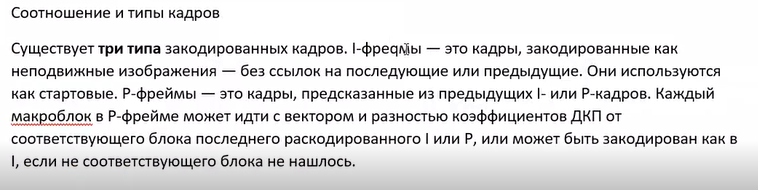
Содержит несколько томов(частей). Например, МПЕГ 4 парт 10 == Аш.264. МПЕГ-4 – очень много сделано для гибкости формата с точки зрения различных областей использования, возможности отдельно хранить компоненты.

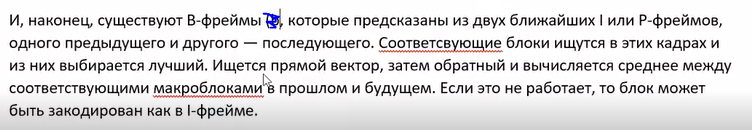
**HEVC(H.265) –** наиболее современный формат. (позже)

**Идеи в МПЕГ**

Основное, что есть в МПЕГ, - **компенсация движения**, которая позволяет предсказать движение: от кадра к кадру не повторять движение блока, а определить, куда смещается данный блок, и закодировать его вектором смещения и ссылкой на предыдущий кадр, где можно взять содержимое этого блока. При этом вектора смещения могут ссылаться как на предыдущий, так и на последующий кадр.



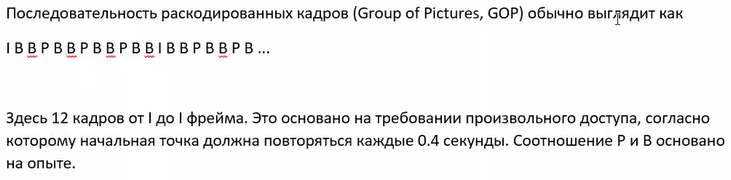




I-фреймы – intro – отдельные ЖПГ. Бывают видеопотоки, состоящие только из i-кадров, тогда, по сути, это Motion JPEG, M-JPEG. Этим пользуются, например, ряд производителей видеокамер (из-за того, что в любой момент можно нажать на стоп, и нельзя будет ссылаться на следующий кадр).

Р-фреймы – predicted.

B-фреймы – bidirectional predicted – предсказаны из двух или одного. **За счет эффекта послевидения мы подмешиваем за хорошим кадром пару плохих, а потом опять хороший. Это дает возможность обманывать зрение и существенно уменьшать объем хранимой информации.**



По факту: I -> P -> B B -> P -> B B и тд. В реальности, чтобы декодер не перепрыгивал через кадры, кодер переставляет их в нужном порядке, для увеличения скорости.

В-кадры наиболее плохи по информации – они содержат ссылки только вперед-назад, но это незаметно для человеческого глаза. Но если в процессе просмотра МПЕГ нажать на паузу, можно увидеть кадр очень низкого качества – В-кадр. А за счет того, что каждые 0.4 секунды показывается более хороший i-кадр, человеческий глаз в процессе просмотра этого не замечает. Зато сильно выигрывает по сжатию.

**Методы оценки и компенсации движения**

1. Полный перебор (full search) – берем картинку, режем ее на блоки, и для текущего блока перебираем все блоки соседнего кадра, ищем наиболее похожий.

Плюсы: гарантированно найдем смещенный блок, если он вообще был на следующем кадре

Минусы: наиболее вычислительно затратный метод

1. Блочные алгоритмы оценки движения (см. файл Blochnye algoritmy ocenki dvizhenija)

Алгоритмы оценки движения разрабатываются, исходя из компромисса между сложностью вычислений, качеством работы и объемом выходного потока данных.

Наиболее известные блочные алгоритмы оценки движения: трехшаговый, четырехшаговый, ортогональный, логарифмический и иерархический.

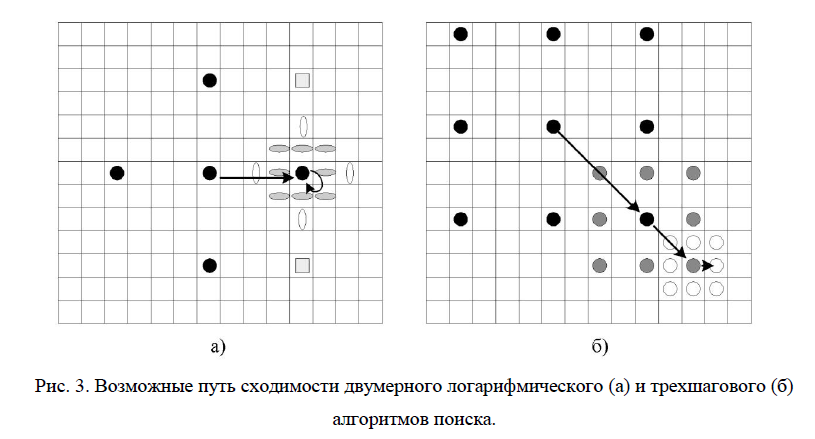
**В общем случае** задача сводится к поиску вектора d – поиска смещения функции яркости. В простом варианте ограничивается зона поиска (не по всему кадру).

**Есть алгоритм поиска с уточнениями сходимости** – сначала разбивает блоки в крупную клетку и ищем наиболее похожий блок. После того, как мы определили, например, что в предыдущем кадре блок был по центру, а в текущем кадре – в верхнем правом углу, мы смещаемся в угол, и уже начинаем искать вокруг него.

**Трехшаговый поиск** ищет вектора движения, переходя от более грубого к более точному шаблону поиска. Он состоит из следующих шагов:

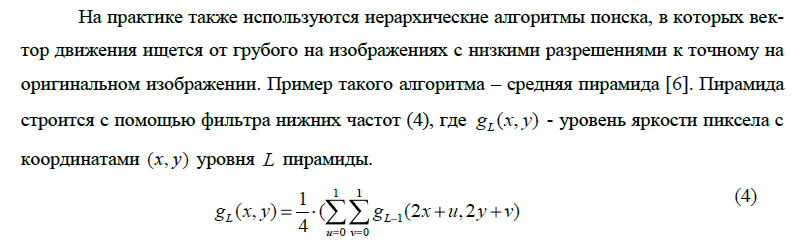
1 - рассматриваются восемь блоков на начальном расстоянии от центра (текущего блока).

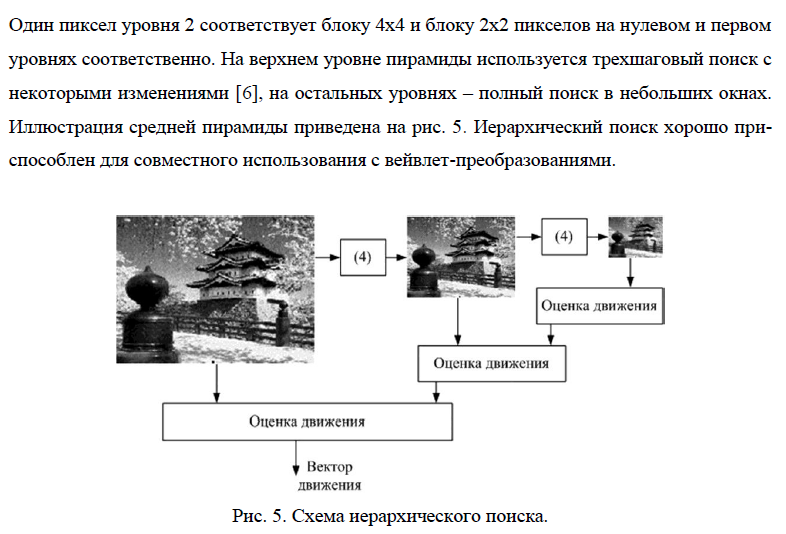
2 - начальное расстояние между центрами блоков уменьшается вдвое, центр поиска сдвигается в точку с минимальными искажениями. Этапы 1 и 2 повторяются до тех пор, пока начальный шаг не будет меньше единицы. На рис. 3б приведена возможная схема сходимости трехшагового алгоритма поиска. Недостатком трехшагового поиска является однородное расположение точек в первом шаге, которое неэффективно в случае медленного движения.



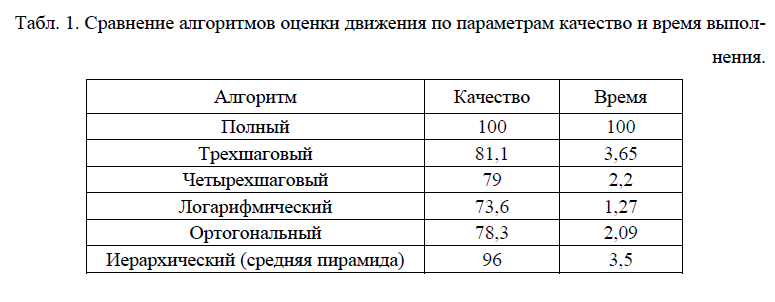
Разница с **логарифмическим** в том, что в лог. поиске зона поиска сужается логарифмическим шагом.

1. Телескопический поиск (продолжаем искать путем продления предыдущего кадра) – если известен вектор смещения на шаг назад, тогда предполагаем, что объект продолжает свою линейную скорость в том же направлении, и мы берем предыдущий вычтенный вектор смещения, сдвигаемся дальше на него, и начинаем искать требуемый блок по линии его движения, постепенно расширяя область поиска.
2. Иерархические алгоритмы поиска (сначала ищем на огрубленном изображении, постепенно повышая точность). Используют идеи блочных, но они ищут сначала по пирамиде.





**На картинке стрелки должны быть в другую сторону: от маленького к большому.**



1. Метод корреляции фаз (не блочные алгоритмы). Основан на теореме Фурье о сдвиге изображения – если одно изображение является копией предыдущего, сдвинутой на какое-то значение, то ДПФ также будут сдвинуты друг относительно друга на:



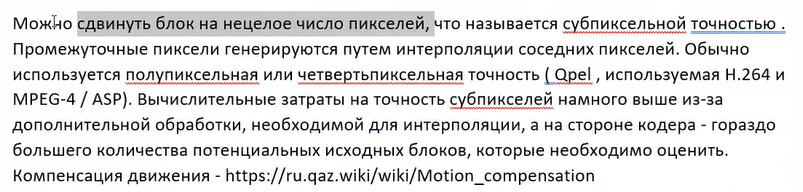
То есть, ДПФ картинки В будет равно дискретному преобразованию (ДПФ) картинки А, умноженному на е в степени (в формуле).

Основываясь на этой формуле, можно узнать дельта х, дельта у через преобразование, через взаимную корреляцию. Этот метод вычислительно более затратен, но устойчив к шумам.

([**https://ru.qaz.wiki/wiki/Phase\_correlation**](https://ru.qaz.wiki/wiki/Phase_correlation))

1. Методы 3Д-поиска (хороши при смещениях не в плоскости кадра, а ближе/дальше от камеры (от нас/к нам)).
2. Эвристические и др.методы…

**Методы, которые надо знать точно:** полный перебор, логарифмический, телескопический, метод корреляции фаз.



Когда реальный объект смещается, а на раскадровке получилось, что блок не попал в пиксели, а сместился на нецелое количество пикселей. Вычислительные затраты выше, но и качество тоже.

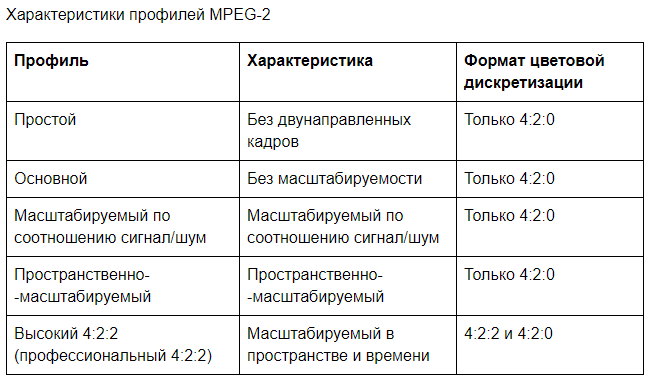


При глобальной компенсации на сцене ничего не происходит, но движется камера.

Плюсы: не надо делить на блоки, можно предсказывать линейно.

**Еще плюсы МПЕГ:**

Уровни и профили позволяют стандартизировать возможные потоки данных с точки зрения их качества, скорости, степени компрессии, правил цветового прореживания и пр..



Масштабируемый по отношению сигнал/шум – разный битрейт может быть.

Уровни – определяют пространственное разрешение кадров.

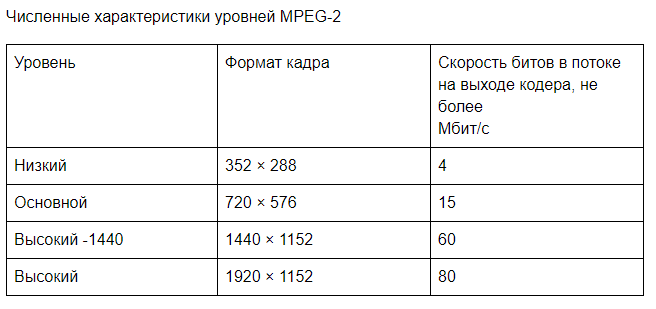




Таблица выше дает понимание об ожидаемом битрейте – та пропускная способность, которая должна обеспечиваться для видео без задержек и лагов.

В МПЕГ-4 уровней и профилей еще больше.

MPEG-4 Visual (группа Part 2 стандартов MPEG-4)

**1999й год**

MPEG-4 Visual (Part 2 ISO/IEC 14496, «кодирование аудиовизуальных объектов») представляет собой объемный документ, описывающий довольно широкий класс функций, относящихся к кодированию и представлению визуальной информации. Стандарт имеет дело со следующими типами данных:

* движущиеся изображения (прямоугольные кадры):
* видеообъекты (области произвольной формы с движущимися предметами);
* двухмерные и трехмерные сеточные объекты (представляющие деформируемые объекты);
* анимированные лица и фигуры людей;
* статические текстуры (неподвижные изображения).

КРОМЕ ПРЯМОУГОЛЬНЫХ КАДРОВ, ОСТАЛЬНОЕ ПРАКТИЧЕСКИ НЕ ИСПОЛЬЗУЕТСЯ

Несмотря на многообразие инструментов, задаваемых стандартом, в сердцевине MPEG-4 Visual находится довольно простой механизм видеокодирования, использующий кодек на основе блоков с компенсацией движения и с последующим преобразованием DCT, квантованием и энтропийным кодированием. Синтаксис этого стержневого кодека (при некоторых ограничениях) идентичен ядру Н.263.

Стандарт H.264 / AVC / MPEG-4 Part 10

**2003й год**

Группы экспертов ITU-Т и MPEG объединились в JVT (Joint Video Team) и разработали новый стандарт, который превосходит по степени сжатия видеоизображений опубликованные ранее стандарты MPEG-4 и Н.263. Этот стандарт получил название AVC (Advanced Video Coding, усовершенствованное кодирование видео) и был опубликован одновременно как стандарт MPEG-4 Part 10 и как рекомендация ITU-T H.264.

Стандарт Н.264 был разработан для эффективного кодирования видео прямоугольного формата. Его первоначальной задачей было обеспечение функциональности, схожей с Н.263, но с лучшим сжатием и более надежным механизмом передачи данных. Большинство функциональных блоков кодера (прогноз, преобразование, квантование, энтропийное кодирование) - такие же, как и в Н.261, Н.263, - сохранились, но подверглись существенной переработке.

Для энтропийного кодирования выбраны Exp-Golomb коды (экспоненциальные коды Golomb) - коды переменной длины с регулярной структурой, кодирование переменной длины с адаптацией на основе контекста (CAVLC), адаптивное двоичное арифметическое кодирование на основе контекста (CABAC). - **добавили**

Компенсация движения выбрана с точностью в четверть пикселя, при этом размер блока компенсации движения может изменяться от 16х16 до 4х4 пикселя.

Дополнительно для повышения сжатия и качества используются (ВИКИПЕДИЯ):

* внутрикадровое кодирование. Эта техника сжимает похожие цвета лучше, чем Н.263, особенно в сложных для кодирования сценах с быстрым движением. Благодаря этой возможности подобные фрагменты кодируются с меньшей битовой скоростью (битрейтом); - **можем ссылаться на вектора смещения не только между кадрами, но и на другие блоки текущего кадра**
* адаптивный фильтр для уменьшения артефактов блочности (Adaptive Deblocking Filter). В рекомендации H.264 используется фильтр удаления блочности для блоков размером 4x4.
* Многокадровое предсказание:
  + **Использование сжатых ранее кадров в качестве опорных** (то есть с заимствованием части материала из них) куда более гибко, чем в предыдущих стандартах. **Позволяется использование до 32 ссылок на другие кадры, тогда как в более ранних число ссылок ограничено одним или, в случае**[**B-кадров**](https://ru.wikipedia.org/wiki/%D0%A2%D0%B8%D0%BF%D1%8B_%D0%BA%D0%B0%D0%B4%D1%80%D0%BE%D0%B2)**, двумя кадрами**. Это поднимает эффективность кодирования, так как позволяет кодеру выбирать для компенсации движения между большим количеством изображений. В большинстве сцен данная функция обеспечивает не очень большое улучшение в качестве и не даёт заметного понижения [битрейта](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82%D1%80%D0%B5%D0%B9%D1%82" \o "Битрейт). Однако, для некоторых сцен, например, с частыми повторяющимися участками, возвратно-поступательным движением и т. п. данный подход при сохранении качества позволяет очень сильно снизить затраты битрейта.
  + [**Компенсация движения**](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B5%D0%BD%D1%81%D0%B0%D1%86%D0%B8%D1%8F_%D0%B4%D0%B2%D0%B8%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F)**с переменным размером блока (от 16x16 до 4x4 пикселя)** позволяет крайне точно выделять области движения.
  + Векторы движения, выводящие за границы изображения. В [MPEG-2](https://ru.wikipedia.org/wiki/MPEG-2) и предшествовавших ему стандартах векторы движения могли указывать только на пикселы, находящиеся в границах декодированного опорного изображения. Методика экстраполяции за границы изображения, появившаяся как опция в [H.263](https://ru.wikipedia.org/wiki/H.263), включена в новый стандарт.
  + Шеститочечная фильтрация компонента яркости для полупиксельного предсказания с целью уменьшения зубчатости краев и, в конечном счёте, обеспечения большей чёткости изображения.
  + **Точность до четверти пиксела (Qpel) при компенсации движения** обеспечивает очень высокую точность описания движущихся областей (что особенно актуально для медленного движения). Цветность, как правило, хранится с разрешением, уменьшенным вдвое по вертикали и горизонтали ([прореживание цвета](https://ru.wikipedia.org/w/index.php?title=%D0%9F%D1%80%D0%BE%D1%80%D0%B5%D0%B6%D0%B8%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D1%86%D0%B2%D0%B5%D1%82%D0%B0&action=edit&redlink=1)), поэтому компенсация движения для компонента цветности использует точность в одну восьмую пиксела цветности.
* Сжатие макроблоков без потерь:
  + Метод представления макроблоков без потерь в [PCM](https://ru.wikipedia.org/wiki/PCM), при котором видеоданные представлены непосредственно, позволяющий точно описывать определённые области и допускающий строгое ограничение на количество закодированных данных для каждого макроблока.
* Новые функции преобразования:
  + Точное целочисленное преобразование пространственных блоков 4×4 (концептуально подобное широко известному [DCT](https://ru.wikipedia.org/wiki/DCT), но упрощенное и способное обеспечить точное декодирование[[1]](https://ru.wikipedia.org/wiki/H.264#cite_note-1)), позволяющее точное размещение разностных сигналов с минимумом шума, часто возникающего в предыдущих кодеках.
  + Точное целочисленное преобразование пространственных блоков 8x8 (концептуально подобное широко известному DCT, но упрощенное и способное обеспечить точное декодирование; поддерживается не во всех профилях), обеспечивающее большую эффективность сжатия схожих областей, чем 4×4.
  + **Адаптивный выбор кодеком между размерами блока 4×4 и 8×8 (поддерживается не во всех профилях).**
  + Дополнительное [преобразование Адамара](https://ru.wikipedia.org/w/index.php?title=%D0%9F%D1%80%D0%B5%D0%BE%D0%B1%D1%80%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%90%D0%B4%D0%B0%D0%BC%D0%B0%D1%80%D0%B0&action=edit&redlink=1), применяемое к дискретно-косинусным коэффициентам основного пространственного преобразования (к коэффициентов яркости, и, в особом случае, цветности) для достижения большей степени сжатия в однородных областях.
* Квантование:
  + Логарифмическое управление длиной шага для упрощения распределения битрейта кодером и упрощенного вычисления обратной длины квантования.
  + Частотно-оптимизированные матрицы масштабирования квантования, выбираемые кодером для оптимизации квантования на основе человеческих особенностей восприятия (поддерживается не во всех профилях).
* Внутренний фильтр [деблокинга](https://ru.wikipedia.org/w/index.php?title=%D0%94%D0%B5%D0%B1%D0%BB%D0%BE%D0%BA%D0%B8%D0%BD%D0%B3&action=edit&redlink=1" \o "Деблокинг (страница отсутствует)) в цикле кодирования, устраняющий [артефакты](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D1%82%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D1%8B_%D1%81%D0%B6%D0%B0%D1%82%D0%B8%D1%8F) блочности, часто возникающие при использовании основанных на [DCT](https://ru.wikipedia.org/wiki/DCT) техниках сжатия изображений.
* [Энтропийное кодирование](https://ru.wikipedia.org/wiki/%D0%AD%D0%BD%D1%82%D1%80%D0%BE%D0%BF%D0%B8%D0%B9%D0%BD%D0%BE%D0%B5_%D0%BA%D0%BE%D0%B4%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) квантованных коэффициентов трансформации:
  + **Context-adaptive binary arithmetic coding** ([CABAC](https://ru.wikipedia.org/wiki/CABAC), контекстнозависимое адаптивное бинарное [арифметическое кодирование](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D0%B8%D1%84%D0%BC%D0%B5%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%BA%D0%BE%D0%B4%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) — алгоритм сжатия без потерь для синтаксических элементов видеопотока на основе вероятности их появления. Поддерживается только в Main Profile и выше. Обеспечивает более эффективное сжатие, чем [CAVLC](https://ru.wikipedia.org/w/index.php?title=CAVLC&action=edit&redlink=1), но требует значительно больше времени на декодирование.
  + Context-adaptive variable-length coding (CAVLC, контекстнозависимое адаптивное кодирование с переменной длиной кодового слова) — альтернатива CABAC меньшей сложности. Тем не менее, оно сложнее и эффективнее, чем алгоритмы, применяемые для тех же целей в более ранних технологиях сжатия видео (как правило это [алгоритм Хаффмана](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%A5%D0%B0%D1%84%D1%84%D0%BC%D0%B0%D0%BD%D0%B0)).
  + Часто используемое, простое и высоко структурированное кодирование словами переменной длины многих элементов синтаксиса, не закодированных CABAC или CAVLC, известное как [коды Голомба](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4%D1%8B_%D0%93%D0%BE%D0%BB%D0%BE%D0%BC%D0%B1%D0%B0) (экспоненциальное кодирование Голомба).
* Функции устойчивости к ошибкам:
  + Определение уровня сетевой абстракции ([NAL](https://ru.wikipedia.org/w/index.php?title=Network_Abstraction_Layer&action=edit&redlink=1)), позволяющее использовать один и тот же синтаксис видео в различных сетевых окружениях, включая наборы параметров последовательности (sequence parameter sets, SPSs) и наборы параметров изображения (picture parameter sets, PPSs), которые обеспечивают большую надёжность и гибкость, чем предыдущие технологии.
  + **Гибкое упорядочивание макроблоков (**[**FMO**](https://ru.wikipedia.org/w/index.php?title=Flexible_Macroblock_Ordering&action=edit&redlink=1)**),** также известное как группы частей (поддерживается не во всех профилях) и произвольное упорядочивание частей ([ASO](https://ru.wikipedia.org/w/index.php?title=Arbitrary_Slice_Ordering&action=edit&redlink=1)) — методы реструктурирования порядка представления фундаментальных областей (макроблоков) в изображениях. При эффективном использовании гибкое упорядочивание макроблоков может существенно повысить устойчивость к потере данных.

Благодаря ASO, так как каждая часть изображения может быть декодирована независимо от других (при определённых ограничениях кодирования), **новый стандарт позволяет посылать и получать их в произвольном порядке друг относительно друга**. Это может снизить задержку в приложениях реального времени, особенно при использовании на сетях, имеющих режим работы **доставка вне очереди**. Эти функции могут также использоваться для множества других целей помимо восстановления ошибок.

**H.263/264 – стандарты проведения видеоконференций. Но при задержках вдиео, чтобы не сыпалась картинка, в H.264 добавилась помехозащита и устойчивость, чтобы картинка портилась меньше.**

Помехозащита

Низкая устойчивость к равновероятно распределенным случайным ошибкам при передаче видеоданных по каналам связи. Это обусловлено прежде всего явлением размножения ошибок при восстановлении подвижных изображений. Проведенные эксперименты по декодированию изображений по видеоданным, имеющим искаженные двоичные разряды, позволили установить, что даже одиночные ошибки в видеоданных приводят к потере основного содержания определенного фрагмента изображений.

Профили

**Constrained Baseline Profile (Ограниченный базовый профиль)**

Рассчитан на применение в недорогих продуктах. Включает набор возможностей, общих для профилей Baseline, Main, и High профилей.

**Baseline Profile (Базовый профиль)**

Применяется в недорогих продуктах, требующих дополнительной устойчивости к потерям. Используется для видеоконференций и в мобильных продуктах. Включает все возможности Constrained Baseline Profile и, дополнительно, возможности для большей устойчивости к потерям при передаче. С появлением Constrained Baseline Profile отошёл на второй план, так как все потоки Constrained Baseline Profile соответствуют Baseline Profile, и оба этих профиля имеют общий код идентификатора.

**Main Profile (Основной профиль)**

Применяется для цифрового телевидения стандартной четкости в трансляциях, использующих сжатие MPEG-4 в соответствии со стандартом DVB.

**Extended Profile (Расширенный профиль)**

Предназначен для потокового видео, имеет относительно высокую степень сжатия и дополнительные возможности для повышения устойчивости к потере данных.

**High Profile (Высокий профиль)**

Является основным для цифрового вещания и видео на оптических носителях, особенно для телевидения высокой четкости. Используется для Blu-Ray видеодисков и DVB HDTV вещания.

**High 10 Profile (Высокий профиль 10)**

Дополнительно поддерживает 10-битовую глубину кодирования изображения.

**High 4:2:2 Profile (Hi422P)**

В основном нацелен на профессиональное использование при работе с чересстрочным видеопотоком. Поддерживает дополнительный вариант кодирования цветности.

**High 4:4:4 Predictive Profile (Hi444PP)**

Базируясь на Hi422P, включает ещё один вариант кодирования цветности и работу с 14-битной глубиной кодирования.

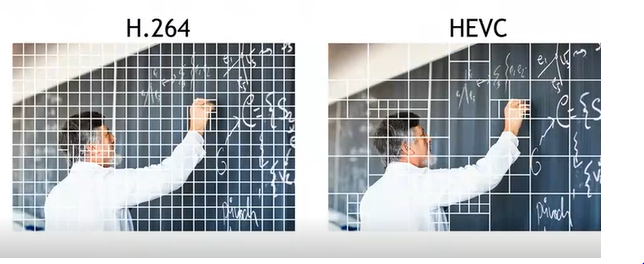
Функции поддержки в отдельных профилях

|  |
| --- |
|  |
| **Функции** | **CBP** | **BP** | **XP** | **MP** | **HiP** | **Hi10P** | **Hi422P** | **Hi444PP** |
| **Chroma formats(Формат цветового прореживания)** | 4:2:0 | 4:2:0 | 4:2:0 | 4:2:0 | 4:2:0 | 4:2:0 | 4:2:0/4:2:2 | 4:2:0/4:2:2/4:4:4 |
| **Sample depths (bits) – цветовая глубина** | 8 | 8 | 8 | 8 | 8 | 8 to 10 | 8 to 10 | 8 to 14 |
| **Flexible macroblock ordering (FMO)** | Нет | Да | Да | Нет | Нет | Нет | Нет | Нет |
| **B slices** | Нет | Нет | Да | Да | Да | Да | Да | Да |
| **CABAC entropy coding – адаптивное арифметическое кодирование** | Нет | Нет | Нет | Да | Да | Да | Да | Да |
| **8×8 vs. 4×4 transform adaptivity** | Нет | Нет | Нет | Нет | Да | Да | Да | Да |
| **Quantization scaling matrices – матрица квантизации с логарифмическим шагом** | Нет | Нет | Нет | Нет | Да | Да | Да | Да |
| **Monochrome (4:0:0)** | Нет | Нет | Нет | Нет | Да | Да | Да | Да |

H.265 или HEVC ( High Efficiency Video Coding). MPEG-H Part 2

2010й год

Вместо применяющихся в H.264 макроблоков в HEVC используются блоки с древовидной структурой кодирования. Выигрыш кодера HEVC — в применении блоков большего размера.

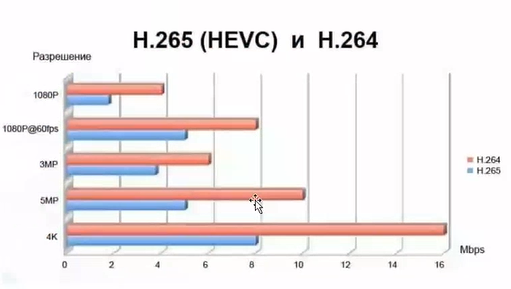


Цель лабы: насколько эффективно будет сжиматься в одном и в другом

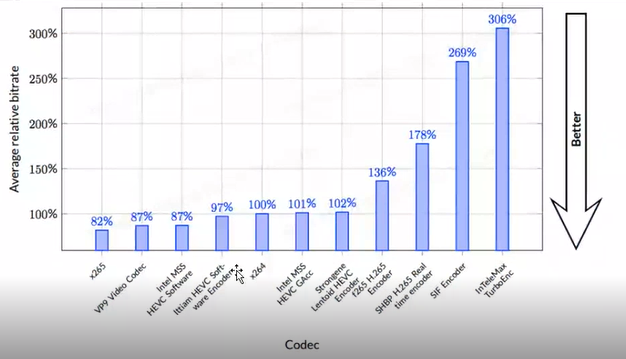
Другие дополнения:

* [Компенсация движения](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B5%D0%BD%D1%81%D0%B0%D1%86%D0%B8%D1%8F_%D0%B4%D0%B2%D0%B8%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F) с точностью до 1/8 пикселя (Qpel)
* Адаптивное предсказание ошибок кодирования (APEC) в пространственной и частотной областях – чтобы блоки не сыпались
* Адаптивный выбор матрицы квантования (AQMS)

В теории:



На практике:



Многое зависит от кодека. На диаграмме – H.264 – 100% и различные вариации H.265

По лабе: **битрейт(общую скорость потока) нужно порезать в два раза**

**Лекция 5.**

**Физиология зрения (гл.8, со стр.171 «Технологии мультимедиа и восприятие ощущений»)**

Есть несколько типов движений глаз:

Первое из них — это **медленное следящее движение**, требующее сознательного внимания. Оно удерживает область наилучшего видения направленной на объект интереса при его движении или перемещении самого наблюдателя. Движения глаз (сокращения глазных мышц) могут быть достаточно быстрыми — до 100 м/с, но могут замедляться наркотическими веществами, утомлением или алкоголем.

Например, провожаем глазами автомобиль/самолет/другой быстро движущийся объект. – когда сознательно управляем мышцами глазных яблок и поворачиваем глаза вслед за объектом.

Второе фиксирующее движение — более быстрое и **называется саккадическим**. Когда объект уходит из области наилучшего зрения, глаз остается неподвижным около 200 мс, а затем быстро (со скоростью около 900 м/с) смещается, чтобы вновь восстановить высокое качество изображения на сетчатке. Такое движение может быть недостаточным или избыточным, тогда центрирование доводится микросаккадическими движениями.

Можно рассматривать все эти движения как «подстройку». Глазодвигательная система беспрерывно находится в таком состоянии до тех пор, пока максимально четкое изображение не попадет в область лучшего видения (на центральную ямку). Поэтому при уходе объекта из этой области за счет времени реакции (латентного периода) человек не способен отслеживать движение еще около 200 мс.

Наконец, третий тип фиксирующих движений — это **вергентные движения**. Во всех предшествующих случаях оба глаза двигаются в одном и том же направлении, т. е. их движения являются содружественными. При вергентных же движениях — другая ситуация. Если объект приближается к лицу или удаляется от него, то глаза согласованно движутся в противоположных направлениях — к переносице или вискам, т. е. несодружественно. Вергентные движения побуждаются различиями положения изображений объекта на сетчатках двух глаз — диспаратностью (сетчаточным несовпадением). Вергентные движения также согласованы с аккомодацией хрусталика: когда меняется расстояние до объекта, мышцы ресничного тела, контролирующие аккомодацию, и глазные мышцы, выполняющие

вергентные движения, работают совместно, что содействует поддержанию точной фокусировки.

**Смещение глаз (нистагм**): если пальцами вытянутой руки быстро размахивать перед собственными глазами, то различить отдельные пальцы становится невозможно. Если же с такой же скоростью вертеть головой, то пальцы остаются вполне различимыми (включаются саккадические движения). Следует учитывать при моделировании пространства, когда смещается и объект, и камера.

Инерция зрения (лат. inertia — «бездействие», «вялость») — это физиологическое явление, заключающееся в отставании возникновения и исчезновения зрительного ощущения от воздействия светового раздражителя. Инерция зрения проявляется наличием латентного периода, возникновением последовательных образов, слиянием световых мельканий и т. п.

Для создания ощущения движения объекта необязательно, чтобы его проекция на сетчатку смещалась непрерывно. Это ощущение не нарушается и в случае, если на сетчатку глаза проецировать ряд

быстро сменяющих друг друга изображений объекта, на которых представлены последовательные фазы его движения. Такое явление называется **стробоскопическим эффектом.**

Именно это свойство зрения используется для создания иллюзий движения в кино и в телевидении. Чтобы движение воспринималось плавным, частота смены изображений с различным фазами

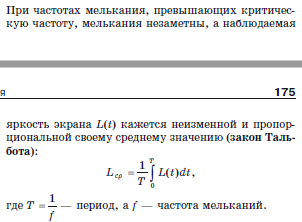
движения должна быть достаточно большой, — иначе движения будут казаться прерывистыми. Чем больше яркость экрана, тем выше эта частота, однако при используемых обычно яркостях экрана она

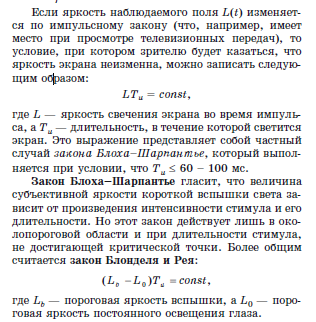
не превышает 60–70 Гц.

**Инерционность зрительной системы обусловлена двумя причинами: инерционностью фотохимических процессов, протекающих в фоторецепторах сетчатки, и инерционностью процессов распространения сигналов по ее нейронным сетям.**

Первый вид инерционности проявляется при изменении средней освещенности наблюдаемой сцены, в результате которого нарушается динамическое равновесие между скоростями распада и регенерации зрительных пигментов в фоторецепторах. Для восстановления этого равновесия требуется от нескольких минут до нескольких десятков минут в зависимости от начального и конечного уровней освещенностей, при которых протекает адаптация. Инерционность же, определяемая процессами распространения сигналов в нейронных сетях зрительной системы, проявляется на более коротких временных интервалах, не превышающих одной секунды.

**Эффект «послевидения» (persistence of vision)** – угасание возбуждения зрительных рецепторов происходит не сразу.





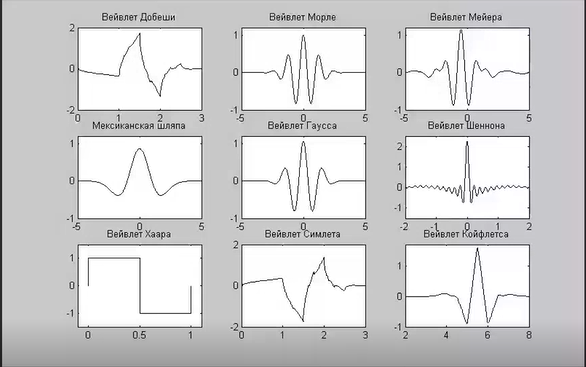
Кинофильмы, как правило, снимаются с частотой 24 кадра в секунду, но и яркость показа в этом случае существенно выше. Телевизионный стандарт NTSC, распространенный в Северной Америке и большей части стран Тихоокеанского бассейна, определяет частоту чуть меньше 30 кадров в секунду, а два других распространенных в остальном

мире стандарта—PAL и SECAM (последний используется, например, в России и Франции), — частоту 25 кадров в секунду. **Каждый кадр при этом строится за два прохода — с помощью двух полей, содержащих четные и нечетные строки изображения**. Таким образом, общая частота смены изображений в телевизионных стандартах составляет от 50 до 60 Гц. – этот режим называется Interlaced Mode. Non-Interlaced Mode – каждый кадр показывается в 2 Гц, без появления эффектов расчески (???????).

Вейвлет – преобразование - **ДВП**

JPEG2000 использует *дискретное вейвлетное преобразование* (*Discrete Wavelet Transformation*) – некий ортогональный базис(функции), по которому можно разложить исходный сигнал, для разбиения изображения на высокочастотные и низкочастотные области. ДВП обрабатывает каждую строку и столбец исходного изображения с помощью частотного фильтра.

Вместо синуса и косинуса (как в ДКП) используются другие функции. Наиболее типичные вайвлеты:



Они могут гладкими или нет(не везде непрерывны). Но это не мешает, так как эти вейвлеты задаются в матричной форме, и матрицы обладают свойством ортогональности.

**Основные св-ва вейвлетов:**

1. **Средний интеграл от –оо до +оо по данной функции= 0,** то есть, если провести линию по оси Х и померить площадь вейвлета над ней и под ней, то площади должны совпасть.
2. **Быстрое затухание при стремлении к бесконечности,** то есть, ее значимая область представляет собой достаточно короткий отрезок.

**Вейвлет Хаара**

Представляет из себя полусумму(верхний пик) и полуразность(нижний пик)

**НЧ = (a+b)/2**

**ВЧ = (a-b)/2**

Пример:

Характерный для видео сигнал исходный**(одномерный):** 255 254 254 255 253 255 252 251 – яркости

Разложение по вейвлету Хаара:

Первый проход:

НЧ = 254.5 254.5 254 251.5

ВЧ = 0.5 0.5 1 0.5

Второй проход (итеративно по первому):

НЧ = 254.5((254.5+254.5) /2) 252.75

ВЧ = 0 1.25

3 проход:

НЧ = 253.625

ВЧ = 0.875

Тогда закодированное изображение без потерь качества можно представить следующим образом:

НЧ3 ВЧ1 ВЧ2 ВЧ3 – исходные 8 чисел также представлены 8 числами.

Обратным ходом можно это восстановить.

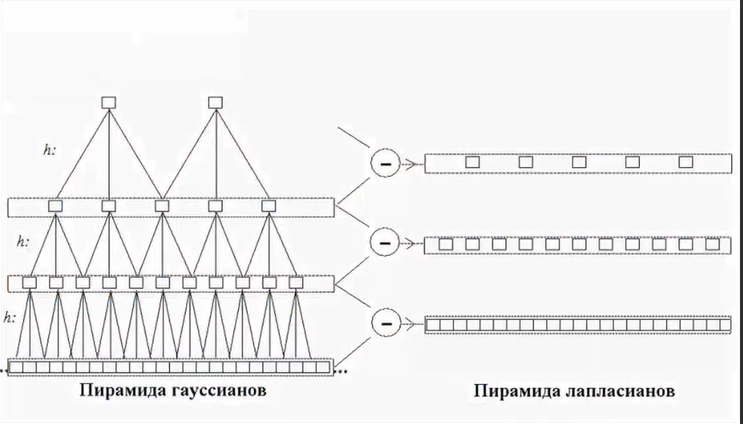
Мы добились того, что низкая частота всего одна, все остальное – высокие частоты, близкие к нулю, которые удобно сжимать, квантовать и тд.

Для двумерного сигнала будет 4 фильтра: полусумма/полуразность всех, а также полусумма/полуразность по вертикали/диагонали.

Графически: исходный сигнал с помощью 4ех фильтров получить 4 более мелких изображения, из которых верхнее левое будет содержать уменьшенную исходную картинку, справа – полусумму по горизонтали, полуразность по вертикали, слева внизу – наоборот, справа – полуразность по диагонали (содержит наименьшее количество значимой информации, поэтому сразу отбрасывается). Далее, берем левый верхний и проделываем то же самое (**ниже есть картинка**).

После окончания этого процесса смотрим, какой квадрат с каким качеством пойдет в сжатый файл путем округления, квантования высоких и низких частот.

Иерархическое представление данного подхода – **пирамида лапласианов и гауссианов.**



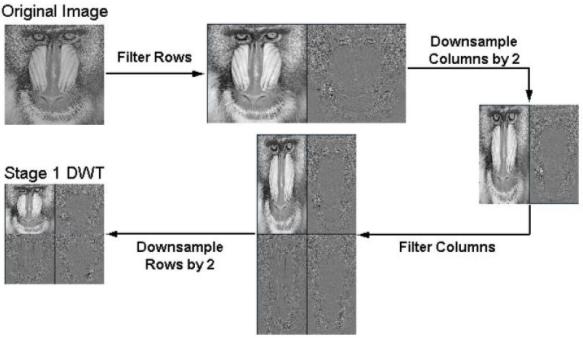
Это была **общая идея вейвлет-сжатия.**

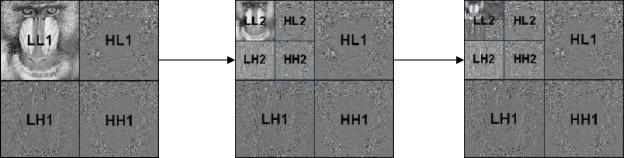
**Хаар в реальности не используется, хоть он и самый быстрый, а используются ортогональные и биортогональные вейвлеты (вейвлет Добеши)**

В связи с тем, что каждый проход с использованием частотного фильтра на выходе увеличивает объем информации в два раза, после обработки размер изображения уменьшается в два раза. После одного этапа ДВП обрабатываемый фрагмент делится на четыре сегмента:

* LL – низкие частоты по строкам и столбцам
* HL – высокие частоты по строкам и низкие по столбцам
* LH – низкие частоты по строкам и высокие по столбцам
* HH – высокие частоты по строкам и столбцам

По стандарту количество этапов может быть от 0 до 32. Для обычного изображения используют от 4-х до 8-ми этапов. На каждом следующем этапе обрабатывается только низкочастотная область (LL), так как в высокочастотных областях обычно не содержится важной информации.

  
  
**Рис. 10.** Схема выполнения одного этапа ДВП.

  
  
**Рис. 11.** Последовательное применение ДВП к изображению.

Для вычисления вейвлет преобразования в стандарте определены два биортогональных вейвлет фильтра – фильтр Добеши(9,7) для сжатия с потерями и фильтр с целочисленными коэффициентами для сжатия без потерь(5,3). В расширениях стандарта разрешается применение любых фильтров, а также не только традиционное разбиение, но и произвольное.

//Коэффициенты приведены <http://sernam.ru/cod_17.php> (не открывается)

***Фильтр с целочисленными коэффициентами***

В случае сжатия без потерь прямое ВП в одномерном случае будет иметь вид

http://sernam.ru/htm/lect_cod/files/cod_17.files/image010.gif,

http://sernam.ru/htm/lect_cod/files/cod_17.files/image011.gif,

что соответствует лифтинговой схеме для вычисления низкочастотной и высокочастотной составляющих. Так как преобразование должно быть без потерь, то сделаем так, чтобы выходные значения http://sernam.ru/htm/lect_cod/files/cod_17.files/image012.gif были целочисленными. Это достигается следующими формулами:

http://sernam.ru/htm/lect_cod/files/cod_17.files/image013.gif

***Преобразование Добеши***

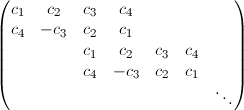
Будем брать по четыре значения, смещаясь каждый раз на два. То есть, если исходная последовательность — 1, 2, 3, 4, 5, 6,…, N-1, N, то будем брать четвёрки (1, 2, 3, 4), (3, 4, 5, 6) и т. д. Последняя четвёрка «кусает последовательность за хвост»: (N-1, N, 1, 2).

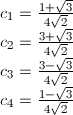
Построим два фильтра: высокочастотный и низкочастотный. Каждую четвёрку будем заменять на два числа. Так как четвёрки перекрываются, то количество значений после преобразования не изменится.

Для удобства подсчёта обратной матрицы потребуем также ортогональности преобразования. Тогда поиск обратной матрицы сведётся к транспонированию.

Пусть значения яркостей в четвёрке равны x, y, z, t. Тогда первый фильтр запишем в виде:  
http://habrastorage.org/storage2/234/86f/ff2/23486fff2b0e3c009671073e7d83d770.png

Четыре коэффициента, образующих вектор-строку матрицы преобразования, пока нам неизвестны.  
  
Чтобы вектор-строка коэффициентов второго фильтра был ортогонален первому, возьмём те же коэффициенты но переставим их и поменяем знаки:  
http://habrastorage.org/storage2/912/c56/4df/912c564dfcfc2774a537338ca8a99a6c.png

Матрица преобразования будет иметь вид.  
  


Требование ортогональности выполняется для первой и второй строк автоматически. Потребуем, чтобы строки 1 и 3 тоже были ортогональны:  
  
http://habrastorage.org/storage2/dec/49c/0c7/dec49c0c7db1844ea1c86387692937b0.png  
  
Векторы должны иметь единичную длину (иначе определитель будет не единичным):  
  
http://habrastorage.org/storage2/1c7/eb5/c04/1c7eb5c04c25eddcbb6a0b0a553d22b7.png  
  
Преобразование должно обнулять цепочку одинаковых значений (например, (1, 1, 1, 1)):  
  
http://habrastorage.org/storage2/eba/c60/ab8/ebac60ab84de7f47f9419c7a093e2c52.png  
  
Преобразование должно обнулять цепочку линейно растущих значений (например, (1, 2, 3, 4)):  
  
http://habrastorage.org/storage2/483/773/83e/48377383ece7e8e838b859071454b4a1.png  
  
Получили 4 уравнения, связывающие коэффициенты. Решая их, получаем:  
  
  
  
Подставив их в матрицу, получаем искомое преобразования. После его применения к фотографиям получим больше нулей и малых коэффициентов, что позволит сжать изображение сильнее.

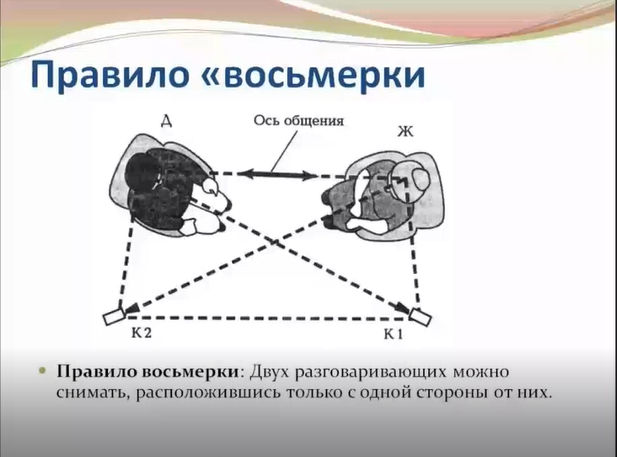
**Дефекты**

Поскольку вейвлет-сжатие – сжатие с потерями, для него характерны аналогичные с JPEG дефекты – потеря мелких деталей, эффект Гиббса (так как раскладываем на функции), дефекты цветового прореживания (в зависимости от кодека).

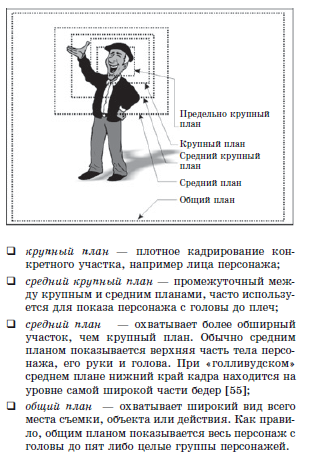
Правила съемки диалогов:

**Дмитрий Синецкий. Видеокамеры и видеосъемка.**

Правило 180 градусов (Правило восьмерки)



Планы съемки: **стр 178 Технологии мультимедиа**



В лабе – общий, средний, крупный.

Шкала – 1 – плохо, 5 – не заметны потери качества, нужно 2-4.

Описать, какой кодек, был использован, какие планы съемки и приемы и найти информацию о том, что за вейвлет применялся.

**Лекция 6.**

**Виды методов оценки:**

* **Субъективные(эксперты)**
* **Объективные**
* **Объективные, основанные на модели человеческого зрения**

Объективные методы оценки: (лабораторная 4)

Часто не показывают артефакты, которые видны человеческому глазу.

Метрики: с предъявлением оригинала/без. (Пример: фильм по телевизору)

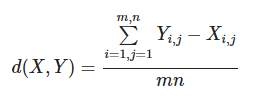
* Метрика sup(x-y) – позволяет увидеть существенные изменения по яркости => если sup(x-y) = 255, при 8-битной глубине цвета, значит, что точка полностью перекрасилась. Если sup > 200, то цвет существенно меняется.
* Матожидание-среднее арифметическое(MSAD/delta) – разница в средней яркости двух изображений.

MSAD – считается по модулю

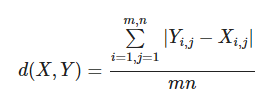
Delta – без

MSAD - в отличие от Delta, покажет реальную разницу между изображениями, 0 - означает полностью эквивалентные изображения, 255 или 1 - разные

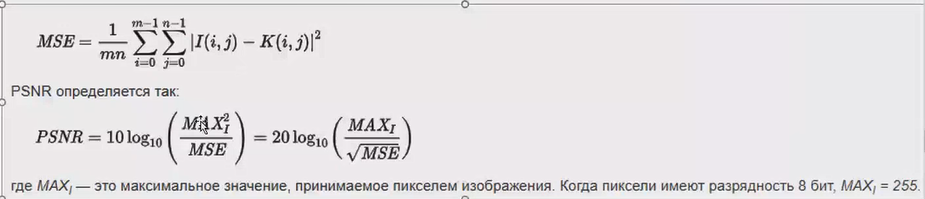
Delta - Эта метрика не показывает потери качества, потому что она может быть равна 0 для совершенно разных изображений, но вы можете определить общие сдвиги яркости с помощью этой метрики, если уверены, что изображения имеют одинаковую структуру.

Delta - 

Эту формулу можно переписать следующим образом: средняя искаженная яркость минус средняя исходная яркость.

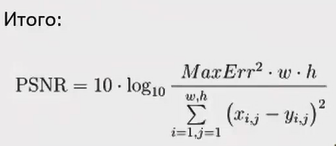
MSAD – 

Метрика зависит только от разницы оригинала и искажения.

* PSNR – отношение пикового сигнала к шуму, измеряется в Дб

MSE – cркв ошибка

Логарифм берем для максимального вытягивания графика возможных величин.



MaxErr – максимальный сигнал/максимальная ошибка (максимальная разница, до 255).

Плюсы:

* + 1. Есть возможность получать одинаковые диапазоны значений pnsr для сигнала, представленного в разном диапазоне значений => можно его измерить для сигнала в диапазоне 0..255 и можно измерить для сигнала из диапазона 0..1, и psnr будет показывать одинаково корректные значения для обоих. То есть, не важно, был ли полноцвет, какая глубина цвета и тд. За счет того, что мы нормируем.
    2. **Подумайте сами))))))))))))))))))))))))))))))))))**

**Чем эта формула лучше, чем просто MSE**

Общий PSNR - это агрегированное значение, которое рассматривает все обработанные кадры как одно огромное изображение, а затем вычисляет PSNR. Общий PSNR учитывает суммарное искажение по всем кадрам и не различает ситуацию, когда все искажения в одном кадре, от ситуации, когда они распределяются по всем кадрам. В то время как агрегированное среднее арифметическое значение зависит от среднего геометрического MSE каждого кадра, общий PSNR зависит от их среднего арифметического.

PSNR < 20 – качество очень плохое

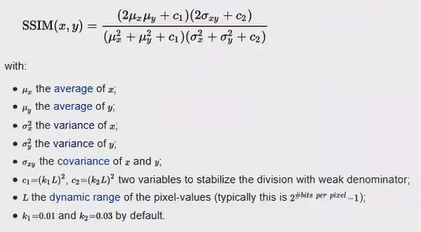
PSNR > 35-40 – качество почти не отличимо от исходного

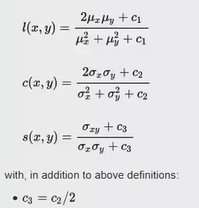
Основная метрика, предназначенная для быстрой и понятной оценки качества. – для выходных оценок.

Для оценок внутри алгоритма м. использовать MSE или delta

* SSIM – строится карта заметности, вычисляется индекс ssim – чем он ближе к 0, тем хуже, 1 – различий нет.

1. Карта светимости
2. Карта контрастности
3. Карта структурного сходства





Вывод по лабе : получили такое-то pnsr/ssim, какое качество у экспертов должно быть на основании этих метрик.

Субъективные методы оценки (лабораторная 5):

Аналитически вариабельны – зависят от очень многих факторов.

Есть стандарт, в котором расписаны методы и рекомендации по формированию групп для оценивания.

Возможное число участников – от 4 до 40(по статистике). Должны быть одинаковые условия просмотра, а эксперты не должны ранее участвовать в подобных упражнениях и быть с одинаковым цветовосприятием и остротой зрения.

Методы:

* Шкала деградации с двумя стимулами – сначала показывают эталонную сцену, а затем – после обработки. Участник оценивает деградацию изображения по 5-балльной шкале (1 – очень раздражают, 5 – не заметны). Можно не говорить, где исходное.
* Непрерывная оценка качества с одним стимулом – предназначено для длинных роликов – зритель раз в минуту сообщает о качестве показываемого материала, исходник не показывают.
* Метод попарного сравнения – чтобы сравнить несколько видов обработки – можно подмешать исходник – составляются все возможные пары из них, пары показываются одновременно, на 50% сером фоне, и зрителю не сообщается, где что.

**Лекция 7.**

**Eye-tracking –** отслеживание движений зрачков глаз. Есть системы различных технологий, которые это делают, для разных целей.

Пример системы(**оптическая**): зафиксировать голову наблюдателя щадящим образом, рядом с экраном поставить лампу, которая будет подсвечивать зрачки глаз + какая-либо видеокамера, направленный на глаза, которые снимает движения зрачков. Чтобы понимать в каком диапазоне движутся зрачки глаз, сначала производится калибровка (калибровочный ролик, в котором зрителя просят посмотреть на концы экрана – 9 точек) – таким образом запоминаются крайние отклонения зрачков.

Есть системы, основанные на **инфракрасной подсветке глаз**; **контактные** (используются в медицинских целях).

**Применение:**

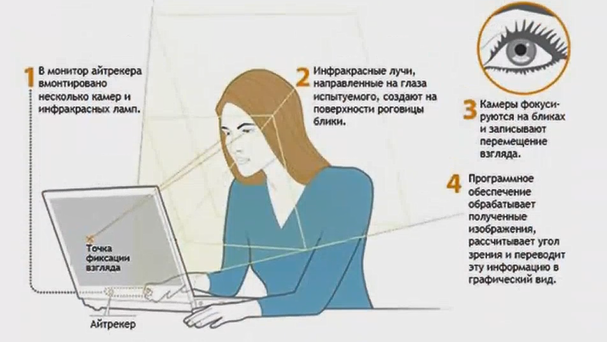
* Системы проектирования компьютерных интерфейсов **–** эти интерфейсы применяются на сайтах магазинов, особенно с большим охватом покупателей: каждый см экрана стоит больших денег, поэтому разработчикам очень важно понимать, где концентрируется взгляд пользователя при работе с таким интерфейсом – чтобы понимать, где какую информацию лучше разместить. Аналогично – с соцсетями, поисковыми системами, массмедиа и тд.
* Военное дело **–** есть системы наведения ракет. Когда времени мало, руки заняты приборами управления, нужно показать цель, и показать ее можно глазами. Есть специальные шлемы у пилотов боевых самолетов и вертолетов, в которые сбоку встраивается инфракрасная подсветка и видеокамера, которая распознает и фиксирует, куда в данный момент смотрит пилот, и наводит боевые системы огня на точку зрения.
* Люди с ограниченными возможностями **–** для частично/полностью парализованных людей глаза являются основным средством управления чем-либо, поэтому есть специальные медицинские системы.
* Выкладка товаров в розничных магазинах **–** правильная выкладка товаров поднимает продажи на 10-15% => постоянно проводится исследование и анализ того, как товары должны лежать на полках так, чтобы они привлекали максимальное внимание покупателя: есть тестовые покупатели; товар раскладывается на витрины, и система eye-tracking отслеживает, куда смотрит покупатель.
* Интерфейсы интернет-магазинов
* Отслеживание артефактов цифрового видео (цепляется ли глаз за них или нет) в целях оценки качества

**Критерии оценивания:** куда смотрит человек; сколько по времени он смотрит; насколько плотно он вокруг объекта делает уточняющие движения зрачками.

Время фиксации глаза и плотность траектории позволяют делать выводы о том, каким элементам рассматриваемого объекта уделяется наибольшее внимание.

**Eye-tracking системы**

* **Бесконтактные**
  + **Оптические**
  + **Инфракрасные**
  + **Гибридные (самые точные) – и инфракрасная, и оптическая**
* **Контактные – установка датчиков непосредственно на органы человека**



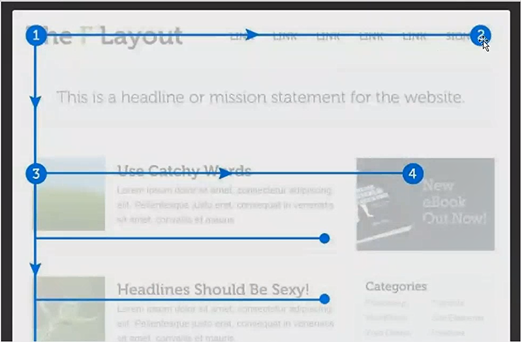
Задача (с точки зрения реализации):

* + - 1. Определение положения зрачка в каждом кадре
      2. Сопоставление положения зрачка на видео точкам в экране, в которые смотрит пользователь

В оптической системе МАИ используется следующий алгоритм: По красному каналу (наиболее контрастному в данном случае) происходит пороговая бинаризация; потом идет операция замыкания с точки зрения дорисовки зрачков; выделение контуров и накладывание эллипсов (аппроксимация эллипсами), выбор наиболее подходящего эллипса; в косоугольных координатах (т.к. от глаз до монитора получается пирамида) этот эллипс совместить с определенной точкой экрана.

Результаты подобных исследований:

1. **Правило буквы F** (Якоб Нильсен): если пользователь серфит интернет без какой-либо конкретной цели, его взгляд скользит по траектории, напоминающей букву F.



Первые строчки дочитываем до конца, потом пару строчек пропускаем. Цепляемся взглядом за еще какую-либо строку, но читаем уже не так внимательно, как первую. Далее, процесс повторяется, но с укорачиванием прочитанной строки.

1. Взгляд движется слева направо
2. Затем возвращается назад и скользит вниз
3. После этого снова направляется в правую сторону.
4. И, наконец, опускается в самый низ страницы.

Результат работы ай-трекера:



Поэтому все разработчики поисковых систем вставляют в первые строчки рекламу, и это наиболее дорогие места для рекламы.

Самые ценные ссылки – в верхнем левом углу, в правом верхнем – еще какая-то контекстная справка (или реклама), затем идут не очень значительные результаты поиска, после – снова релевантные (Google поиск).

1. **Принцип буквы F работает не всегда.** Если пользователь приходит на сайт с определенной целью, траектория его взгляда меняется. Скорость движения по странице увеличивается, так как он знает, что хочет найти. Поэтому траектория меняется.

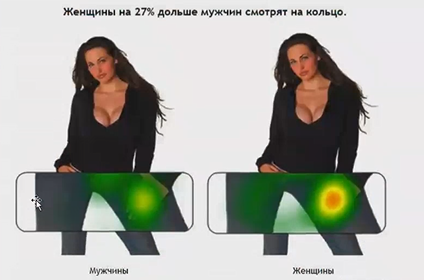
В этом случае eye-tracking нужен на примере конкретного контента, чтобы понять, где что нужно выводить с точки зрения создателя сайта.

1. **Баннерная слепота** (Якоб Нильсен) - в большинстве случаев пользователи игнорируют рекламу; люди исключительно редко смотрят на элементы сайта, напоминающие рекламу, даже если они таковыми не являются. В действительности, люди довольно редко смотрят на рекламу, особенно на анимированные gif.



1. **Статическая реклама более эффективна анимированной** – из-за предыдущего пункта. Пользователи тратят в среднем 6.5% времени на просмотр рекламы в интернете.
2. **Гендерные различия**





1. **Некоторые вещи невозможно игнорировать –** люди практически всегда смотрят на лицо, грудь и область гениталий, этим фактом часто пользуются рекламщики.

**Организация видеовещания в локальной сети**

Для начала, нужно понимать цель видеораздачи.

Есть офф-лайн: ютуб, вк, облака и тд

Есть он-лайн: трансляции в твиче, перископ, ютуб трансляции.

Также можно, например, подключиться к веб-камерам на улице – телеканал может подключиться к какой-либо веб-камере и посмотреть что-либо. – непрерывная трансляция видеопотока с одной камеры на много зрителей.

Другой вариант – он-лайн репортаж.

3 вариант – если организуется собственная станция вещания, причем, как и интернет-радио, и сервер видеовещания, который постоянно или по какому-либо расписанию крутит записи из видеобазы и/или съемку real-time.

Для разных случаев настройки сети и сетевого ПО совершенно разные.

**Виды организаций видеовещания:**

По способу доставки данных:

* Системы с одноадресным вещанием(юникаст) – вещание «точка-точка»
* -||- многоадресным (мультикаст) – вещание «точка-тире»

По степени интерактивности:

* Вещание по запросу(он-деманд) – пользователь сам должен изъявить желание – ютуб
* Широковещательные системы (бродкаст) – пользователь подключается к идущему широкому вещанию и не имеет права менять программу вещания – телеканалы

Есть следующие сочетания видов вещания:

Unicast on-demand – одноадресная система по запросу – ютуб – именно для пользователя, в выбранное им время начинается одноадресный юникаст-поток.

Unicast broadcast – одноадресная система вещания на всех – домашняя веб-камера, которую пользователь может включить, настроить, а затем сообщить ее адрес кому-либо для подключения. По умолчанию, без специализированного сетевого оборудования организовать данный тип вещания не получится, поэтому эта веб-камера будет каждому подключившемуся отдавать свой поток.

Multicast broadcast – многоадресное вещание на всех – любое кабельное ТВ;

Multicast on-demand – **не бывает, тк on-demand подразумевает, что пользователь сам выбирает, что и когда ему смотреть; а если много адресов вещания, то какой из пользователей будет выбирать? По идее, каждый. Но невозможно нескольким пользователям договориться, что они будут смотреть один и тот же ролик в одно и то же время с точностью до мс. Поэтому для каждого из них видеовещание будет индивидуальное. – Так же, как группа зрителей смотрит один телевизор, и они выхватывают друг у друга пульт.**

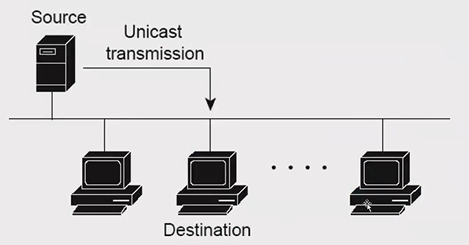
Multicast broadcast разгружает сеть, тк в случае юникаст бродкаст приходится каждому подключившемуся отдавать свой одинаковый поток, и нагрузка возрастает в разы.

Также, у всех пользователей разная скорость интернета, поэтому один и тот же поток нужно готовить в разном качестве, и отдавать поток соответствующего качества в зависимости от скорости интернета подключенного.

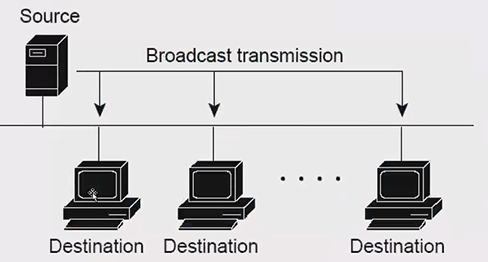
Мелкомягкие разработали формат ASF (Advanced Streaming Format). Он может работать как с файлами, так и в реал-тайм содержать потоки. Разница между файлом и потоком в том, что поток бесконечен, в отличие от файла. **Главное в АСФ то, что он поддерживает возможность вещания с разными качествами.** Для этого нужен специальный софт – Windows Media Encoder – в нем указывается, с какого ресурса брать видео и качество/количество выходных потоков. Берется исходный видеопоток, его, в режиме реального времени раскладывает на упакованные, сжатые потоки требуемого качества и отдает их в формате АСФ, а сервер Windows Media уже раздает их клиентам как по юникаст, так и по мультикаст. У клиента должен стоять проигрыватель Windows Media, для воспроизведения.

**Передача Unicast/Broadcast/Multicast трафика**

**Unicast –** используется для сервисов персонального характера, каждый пользователь может запросить персональный видеоконтент в произвольное время. Трафик направляется из одного источника к одному ip-адресу назначения.



**Broadcast –** использует ip-адрес(маску), чтобы посылать один и тот же поток данных всем абонентам данной ip-подсети(например, адрес может оканчиваться на 255). Бродкаст трафик принимается всеми включенными пользователями сети вне зависимости от их желания. По этой причине используется в основном для передачи служебной информации.



**Multicast –** используется, когда необходимо доставить видеоконтент неограниченному числу абонентов, не перегружая сеть: используется специальный класс ip-адресов (например, адреса в определенном диапазоне), на них должно быть настроено сетевое оборудование (маршрутизаторы, коммутаторы, хабы, свитчи). Сервер отдает на коммутатор один поток, а коммутатор уже маршрутизирует этот поток своим подписчикам, причем только тем, кто это действительно смотрит.

Для этого маршрутизатор должен поддерживать протокол **IGMP (Internet Group Management Protocol) –** для отслеживания текущего состояния групп рассылки (членства в той или иной группе того или иного конечного узла сети, то есть, клиента) – кто из абонентов сейчас висит и какой канал хочет смотреть.

Основные правила работы протокола:

- Периодический опрос: прокотол опрашивает клиента, не отвалился ли он

- Клиент сам сообщает момент подключения – заявку на подключение в группу

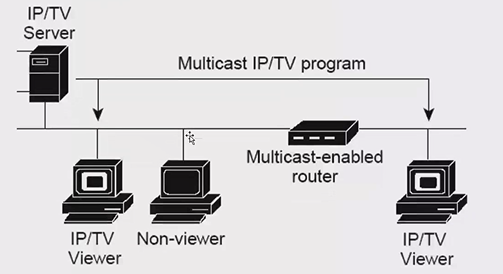
**По-научному))))) :**

**-** конечный узел сети посылает пакет IGMP типа report для обеспечения запуска процесса подключения к группе рассылки

**-** узел не посылает дополнительных пакетов при отключении

**-** маршрутизатор мультикаст с определенной периодичностью посылает в сеть запросы IGMP, которые позволяют определить текущее состояние групп рассылки

**-** узел посылает ответный пакет IGMP для каждой группы до тех пор, пока есть хотя бы один клиент данной группы



**Сервер отдает поток на коммутатор, который поддерживает протокол ИГМП, который, в свою очередь, управляет подписчиками. Если на локальной машине включен плеер, и выбран соответствующий канал, то мультикаст роутер включает подписчика в список рассылки. Если кто-то из подписчиков не хочет ничего смотреть, то, даже находясь физически в той же подсети, он этот трафик получать не будет(non-viewer).**

Это серьезно разгружает сеть, помогает повысить пропускную способность и тд.

Пример:

Как работает IGMP:

Есть сервер, который включен в роутер. Этот сервер вещает несколько ТВ-передач, например:

224.12.0.1 канал1 Cartoon Network

224.12.0.2 канал2 News

224.12.0.3 канал3 Culture

Клиент включает канал1, тем самым отправляя запрос на роутер для подключения к группе 224.12.0.1. С точки зрения протокола, это запрос “JOIN 224.12.0.1”

Если он переключается на другой канал, то отправляется сначала уведомление, что он покидает 224.12.0.1, а затем повторяет запрос join для другого канала.

Через определенный период времени роутер опрашивает, кто из подписчиков в какой группе, используя запрос query. В ответ он получает membership report – список всех групп, в которых состоит данный подписчик.

**Типовая схема сети IPTV(промышленное вещание)**

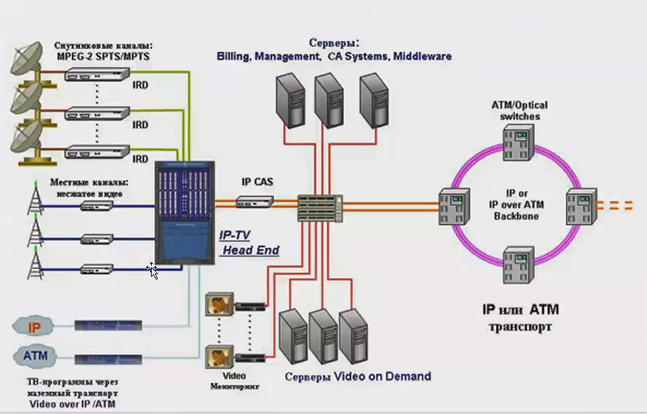
**Компоненты головной аппаратной IP-TV системы:**

* **Головная станция**
* **Система условного доступа**
* **Видеосерверы**
* **Серверы биллинговой системы**
* **Серверы системы менеджмента**
* **Серверы промежуточного ПО**

**Компоненты опорной(магистральной) транспортной сети:**

* **Опорная(backbone) оптическая сеть на базе IP-технологии**
* **Высокопроизводительные маршрутизаторы с оптическими интерфейсами**

**Транспортный уровень доступа, состоящий (например, для xDSL сети) из головного DSL-устройства DSLAM и медной пары (телефонной линии), непосредственно заведенной в дом к абоненту.**





Очень мощный сервер (IPTV Head End), который заботится об источнике видеосигнала (спутниковые каналы, местные каналы, источники через IP, также, как пример, аналоговое видео, веб-камеры и тд).

На центральный маршрутизатор, кроме Head End, подключаются еще сервера, содержащие большие базы видеофильмов, для оказания услуги video on-demand (если кто-то потребует какой-то конкретный фильм), а также много служебных серверов, которые предназначены для управления пользователями, биллингом и тд

Далее, подается на backbone или какую-то другую сеть для вещания.

От backbone сигнал поступает на ящики последней мили – DSLAMs (DSL Access Multiplexor – разделитель сигнала) – они уже по конкретным абонентам раздают видеопотоки.

Все маршрутизаторы в сети должны поддерживать протокол ИГМП для снижения трафика (разгрузки сети).