

Быстрый алгоритм выравнивания кадров для оценки качества MPEG-4 видеопотоков, передаваемых по беспроводным сетям*

Андрей Гущин, Антон Кирьянов, Андрей Ляхов, Евгений Хоров
ИППИ РАН

{guschin, kiryanov, lyakhov, khorov}@iitp.ru

Аннотация

Анализируя возможности сетевых протоколов по передаче видеопотоков в режиме реального времени, многие исследователи столкнулись с проблемой оценки качества видеоизображения, для чего широко используются метрики MSE и PSNR. Однако, эти метрики не учитывают случающиеся время от времени потери пакетов при передаче видеопотока через беспроводную сеть и не могут быть применены в случае, когда в полученном видеофрагменте часть кадров потеряна целиком. Чтобы адаптировать метрики качества видеоизображения к такому сценарию, разработаны различные методы выравнивания кадров в двух видеофрагментах. Оценка схожести каждого кадра в полученном видеофрагменте с рядом кадров из оригинального фрагмента для определения соответствия между ними требует значительного времени, что ограничивает применимость этих методов. В данной работе предлагается новый метод выравнивания кадров, позволяющий существенным образом ускорить процедуру оценки качества видеоизображения, полученного при передаче через беспроводную сеть.

1. Введение

Растущая потребность пользователей в передаче видео в ближайшем будущем превысит возможности существующих средств связи. Согласно недавнему отчету компании Alcatel [1], только в США к 2020 году объем передаваемого видеотрафика увеличится в 12 раз. Вместе с тем за последнее десятилетие произошло стремительное развитие мобильных беспроводных устройств, предоставляющих пользователю повсеместный доступ в интернет. Данный факт делает беспроводные технологии наиболее востребованной частью рынка телекоммуникаций и вследствие

этого встает вопрос: *смогут ли современные беспроводные сети справиться с возрастающим объемом видеотрафика и как улучшить передачу видеоданных по беспроводным сетям?*

В отличие от других типов трафика передача видео в режиме реального времени налагает ограничения на используемые средства передачи данных. Помимо высокой пропускной способности потоковое видео должно быть передано с ограниченной задержкой и с крайне невысокой долей потерянных пакетов. Кроме этого, передача видеопотоков также осложняется их нестационарностью (интенсивность видеопотока меняется во времени) и неоднородностью (различные части видеопотока вносят разный вклад в качество доставленного видео).

Множество исследовательских групп по всему миру изучают возможности различных беспроводных технологий по передаче видео и предлагают соответствующие улучшения стандартов. Работая в этой области, мы столкнулись со следующей проблемой: *как оценивать качество видеопотока, переданного по беспроводной сети?* Большинство метрик оценки качества видеоизображения, найденные в литературе, были разработаны для сравнения различных видеокодексов. Такие метрики достаточно точно (относительно оценки, высказанной пользователем) оценивают качество видеоизображения, но они не могут быть применены в рассматриваемой нами задаче, так как ими не учитывается факт частичной потери данных из видеопотока в ходе передачи через беспроводную сеть. Существуют также и метрики, учитывающие возникающие потери целых видеокадров [2–5]. Метрики такого типа находят множество потерянных кадров, определяют соответствие между полученными и переданными кадрами (выравнивают видеофрагменты) и сравнивают их. Именно такие метрики широко используются при оценке качества видео, переданного по сети.

Оценка производительности беспроводных сетей требует проведения большого количества экспериментов в различных сценариях для достижения статистической значимости полученных результатов. К нашему удивлению, из-за высокой вычис-

*Данная работа выполнена при финансовой поддержке РФФИ в рамках гранта №12-07-33067 мол_а_вед.

лительной сложности алгоритмов выравнивания видеофрагментов, существующие метрики, учитывающие потери кадров, слишком медленны для того, чтобы оценивать качество видеоизображения, переданного по сети, так как для обработки результатов экспериментов (выравнивание и расчет метрики) требуется гораздо больше времени, чем для проведения экспериментов (непосредственно передача видеоданных через беспроводную сеть).

Для решения этой проблемы в данной работе мы предлагаем метод выравнивания кадров, позволяющий существенно ускорить вычисление некоторых метрик оценки качества видеоизображения, с учетом потерь кадров.

2. Обзор литературы

2.1. Структура видеопотока

Видеоданные могут рассматриваться с двух точек зрения. С одной стороны, видеоизображение это последовательность статических изображений, называемых кадрами, каждое из которых представляется в виде матрицы пикселей. Например, в цветовой модели YUV пиксель представляется в виде трех числовых компонент: одна компонента яркости и две компоненты цветности. В таком виде видеоданные занимают большой объем памяти для хранения и требуют большой емкости сети при их передаче.

Для того, чтобы оптимизировать процессы хранения и передачи видеоданных, их предварительно сжимают (или, другими словами, кодируют). Таким образом, с другой стороны видеопоток можно рассматривать в виде закодированного (сжатого) потока байт. В данной работе мы сосредоточимся на технологии кодирования MPEG-4, которая в настоящий момент являются одной из наиболее широко востребованных технологий кодирования мультимедийных данных. Данная технология стандартизована ISO и включает в себя ряд документов (называемых разделами от англ. *"parts"*). Ключевыми разделами этого стандарта являются MPEG-4 Part 2 и MPEG-4 Part 10 (технически идентичен стандарту H.264 [6], разработанному ITU-T). В нашей статье под термином MPEG-4 мы имеем в виду MPEG-4 Part 2 [7].

Далее мы приведем упрощенное описание видеопотока, закодированного по технологии MPEG-4, акцентируя внимание читателя на деталях, которые обычно опускаются в литературе.

Закодированный видеопоток MPEG-4 обладает иерархической структурой вложенных сущностей, показанной на рисунке 1. Каждая сущность начинается с уникальной последовательности бит, которая определяет тип данной сущности. Поскольку данные последовательности не могут встречаться в содер-

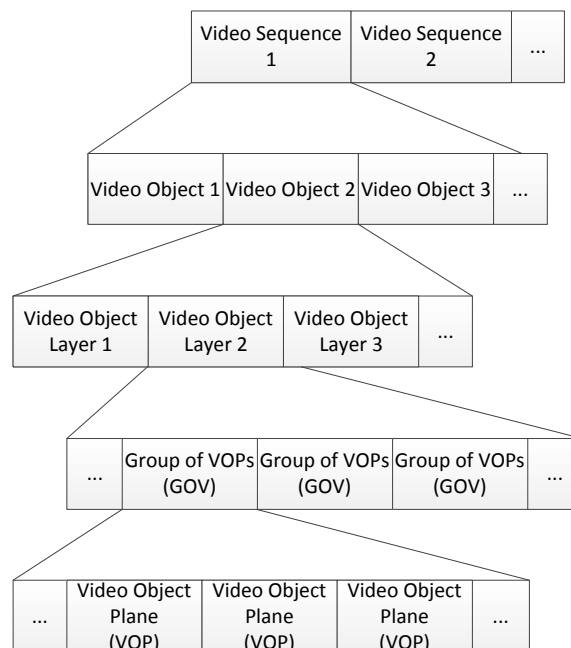


Рис. 1. Упрощенная структура MPEG-4 видеопотока.

жимом сущностей, это дает возможность случайного доступа к закодированному видеопотоку: нет необходимости разбирать всю структура потока с самого начала, вместо этого разбор может быть начат с произвольной сущности.

Сущности, называемые Video Object Plane (VOP), наиболее важны для нашего исследования, так как именно они соответствуют статичным изображениям или кадрам. Каждый VOP начинается с шестнадцатеричной последовательности 00:00:01:B6, и следующие два бита определяют его тип: intra-coded (I), predictive-coded (P), bidirectionally predictive-coded (B) VOP-ы¹ и sprite VOP-ы (S), которые не рассматриваются в данной статье. Последовательность VOP-ов, начинающаяся с I-VOP, называется Group of VOPs (GoV), но термин Group of Pictures (GoP) более распространен в литературе.

I-VOP закодирован с использованием информации только о самом себе. Для кодирования P-VOP используется алгоритм компенсации движения. С целью уменьшения избыточности алгоритм использует схожесть следующих друг за другом кадров и кодирует только информацию об изменениях в изображении, относительно предыдущего кадра. При ко-

¹Чтобы избежать неверного толкования, мы используем терминологию стандарта MPEG-4 и несжатые изображения обозначаем термином «кадр», а закодированные изображения термином «VOP».

дировании B-VOP для компенсации движения использует не только предыдущий кадр, но и последующий. Таким образом, достигается многократная степень сжатия по сравнению с обычными методами и, как следствие, P- и B-VOP-ы имеют гораздо меньший размер, чем I-VOP. Недостатком такого механизма кодирования, как показано на рисунке 2, в том, что невозможно декодировать B- или P-VOP, которые ссылаются на потерянный VOP (заштрихован) или на VOP, который еще не был декодирован. Так, например, если потерян I-VOP (зачеркнут), то все VOP-ы до следующего I-VOP не могут быть декодированы.

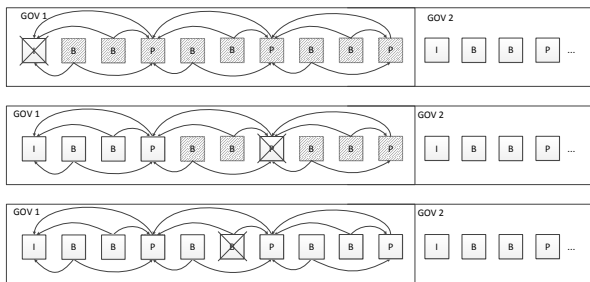


Рис. 2. Зависимости в закодированном MPEG-4 потоке.

Для декодирования B-VOP должны быть предварительно получены и декодированы как предыдущий, так и последующий VOP, информация из которых была использована при кодировании. Отсюда следует, что порядок, в котором сущности VOP хранятся в файле или передаются по сети, так называемый порядок кодирования, отличается от порядка воспроизведения, в котором соответствующие этим VOP-ам кадры изображаются при проигрывании видеофрагмента. Такой эффект называется переупорядочиванием (от англ. *reordering*). См. рисунок 3.

Для передачи видеопотока в режиме реального времени через сеть могут быть использованы различные технологии. Обычно для предоставления такого сервиса, как IPTV, используют стек из протоколов MPEG-TS, RTP, UDP и IP. При использовании этих протоколов видеопоток передается в TS-пакетах одинаковой фиксированной длины в 140 байт. 7 TS-пакетов агрегируются в пакет RTP, как показано на рисунке 4.

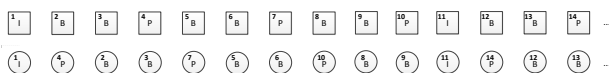


Рис. 3. Порядок воспроизведения (наверху) и порядок декодирования (внизу).



Рис. 4. Агрегирование RTP пакетов.

При передаче видеопотока по сети некоторые пакеты RTP могут быть потеряны и, следовательно, соответствующие VOPы будут повреждены. Как показано выше и как подтверждено в литературе, различные типы VOPов вносят различный вклад в качество видеоизображения [8, 9]. Более того, различные части одного VOP-а могут иметь разную ценность для всего изображения. Например, потеря середины или конца VOP-а ведет только лишь к искажению изображения, в то время как потеря начальной части VOP-а вместе с начальной последовательностью ведет к потере целого VOP-а: он фактически не воспроизводится на приемнике, так как не может быть корректно декодирован. Для оценки качества полученного видео, могут быть использованы различные методы. Их описание приведено в следующем разделе.

2.2. Метрики качества видеоизображения

Вне зависимости от рассматриваемой системы обработки или передачи видеоданных, необходим метод, позволяющий оценить качество видеоизображения, полученного на выходе из системы. Такие методы бывают объективными и субъективными.

Метод субъективной оценки качества характеризует мнение пользователя о качестве просматриваемого им видеоизображения. Хотя именно субъективное качество является основным показателем при сравнении разного рода система, процедура измерения субъективного качества является нетривиальной задачей. Например, метод средней экспертной оценки (от англ. *Mean Opinion Score – MOS*), стандартизованный ITU [10], требует привлечения определенного числа экспертов для просмотра тестовых видеофрагментов в специальных условиях, вследствие чего эксперименты по методологии MOS требуют значительных временных и административных ресурсов. Данный недостаток, ограничивающий применение методов субъективной оценки, способствовал развитию методов объективной оценки качества видеоизображения (называемых также метриками качества), которые бы предсказывали значения MOS, основываясь на некоторых математических моделях или алгоритмах. Будучи откалиброванными, такие методы дают возможность получать оценку качества без при использования существенных ресурсов.

Исторически, большинство метрик качества видеоизображений произошло от метрик качества статичных изображений, и поэтому видеофрагменты сравниваются покадрово. Обычно такие метрики используются в предположении, что оба сравниваемых видеофрагмента имеют одинаковую длину в кадрах (и, как следствие, длительность) и разрешение. Такое допущение является вполне естественным для обычной области применимости таких метрик: сравнение различных видеокодеков и технологий сжатия изображений.

В качестве примера такой метрики рассмотрим, среднеквадратичную ошибку (от англ. *Mean Squared Error* – MSE). Для двух кадров, оригинального X и искаженного Y , размера $M \times N$ каждый, значение метрики равно:

$$MSE = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [X(m,n) - Y(m,n)]^2, \quad (1)$$

где $X(i,j)$ – значение пикселя² изображения X с координатами (i,j) . Если изображения совпадают, то значение MSE равно 0. Очевидно, значение метрики возрастает с количеством искажений в изображении. Значение MSE для видеофрагмента обычно принимается равным среднему значению MSE по всем кадрам или их сумме.

Другая метрика, PSNR (англ. *Peak Signal-to-Noise Ratio* или отношение пикового сигнала к шуму), связана с MSE следующим отношением:

$$PSNR = 10 \cdot \log_{10} \left(\frac{A^2}{MSE} \right), \quad (2)$$

где A – максимально возможное значение яркости для пикселя, например 255 для 8-битного RGB-изображения. Для двух одинаковых изображений значение PSNR равно бесконечности (в литературе часто шкалу PSNR ограничивают сверху значением 100 децибел). Заметим, что PSNR, по сути, является MSE в логарифмическом масштабе, поэтому в большинстве случаев PSNR и MSE взаимозаменяемы.

В силу низкой вычислительной сложности, а также простой физической интерпретации, обе эти метрики используются очень широко. Однако они обладают двумя недостатками: (i) недостаточно точно предсказывают MOS и (ii) не принимают во внимание возможные потери некоторых кадров из-за передачи видеофрагмента по каналу с потерями, то есть сценарий, в котором полученный видеофрагмент имеет меньшую длину, чем исходный передаваемый фрагмент. Данные недостатки обусловили дальнейшее развитие метрик качества видеоизображения по двум направлениям.

²Здесь и далее под значением пикселя имеется в виду числовое значение компоненты яркости рассматриваемого пикселя.

В первом направлении пошли исследователи, чьей задачей являлось улучшение корреляции между оценкой качества, полученной с помощью разрабатываемой метрики, и значением MOS. Большинство из разработанных в этой области метрик существенно усложнились (в том числе и в вычислительном смысле), а некоторое из них даже получили возможность учитывать при оценке качества эффекты, возникающие при передаче видеофрагмента по сети, такие как задержка пакетов или потеря некоторых из них. Однако именно высокая вычислительная сложность ограничивает область применимости множеством видеофрагментов не длиннее нескольких секунд, что недостаточно для исследования эффективности механизмов передачи видеоданных, так как переходные процессы в начале и конце передачи видеопотока могут длиться единицы секунд. В нашей статье мы рассмотрим только метрику SSIM (Structural SIMilarity index), основной идеей которой является то, что значения близко расположенных пикселей коррелируют друг с другом. Заинтересованный читатель может ознакомиться с работой [11], где описан ряд других метрик.

Обозначим за x некоторую подматрицу (или, иными словами, подкадр) оригинального кадра X и за y соответствующую подматрицу полученного кадра Y . Для таких подматриц значение метрики SSIM рассчитывается следующим образом:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \quad (3)$$

где μ_x, μ_y – средние значения элементов матриц x и y ; σ_x, σ_y – дисперсии значений элементов матриц x и y ; σ_{xy} – ковариация x и y ; $c_1 = (k_1A)^2$, $c_2 = (k_2A)^2$ – две константы, добавляемые для исключения деления на малый знаменатель; A – максимально возможное значение яркости пикселя, аналогично (1); $k_1 = 0,01$ и $k_2 = 0,03$ – константы по умолчанию. В работе [12] предлагается использовать квадратные подкадры и симметричная двумерная весовая гауссова функция для расчета $\mu_x, \mu_y, \sigma_x, \sigma_y, \sigma_{xy}$.

После расчета значений SSIM для K пар подматриц, полученных попиксельным сдвигом по всему изображению, значение метрики для целых кадров X и Y вычисляется так:

$$SSIM(X;Y) = \frac{1}{K} \sum_{i=1}^K SSIM(x_i, y_i), \quad (4)$$

x_i, y_i – соответствующие подматрицы. SSIM принимает значение от -1 до 1, где 1 означает полное совпадение изображений.

Как и для MSE и PSNR, для использования SSIM необходимо, чтобы сравниваемые видеофрагменты имели одинаковую длину в кадрах.

Второе направление развития метрик качества видеоизображения определила необходимость срав-

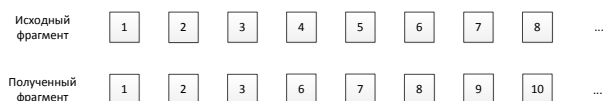


Рис. 5. Потери кадров (№4 и №5) приводят к неправильному покадровому сравнению при расчете метрики.

нивать видеофрагменты, имеющие разную длину, например, из-за потери части кадров при передаче фрагмента по сети. Как показано на рисунке 5, кадр с номером 6 получен сразу после 3-го, что приводит к сдвигу полученного видеоизображения относительно передаваемого и к неверному покадровому сравнению полученного фрагмента с оригинальным. Таким образом, метрики данного типа объединяют в себе простые способы оценки качества такие, как MSE и PSNR, с алгоритмами, позволяющими определить соответствие между кадрами полученного фрагмента и кадрами оригинального фрагмента.

2.3. Существующие методы выравнивания

Наиболее простым способом определения соответствия между кадрами полученного и оригинального видеофрагментов является использование скользящего окна для поиска соответствующего кадра. Для каждого кадра с номером n из полученного видеофрагмента ведется поиск соответствующего ему кадра в оригинальном видеофрагменте в отрезке $[n; n + w]$, где w – размер окна. Кадр из окна, наиболее похожий на кадр n , и выбирается соответствующим. Степень схожести определяется при помощи подсчета значения MSE или PSNR для двух сравниваемых кадров. Минимум MSE (или максимум PSNR) означает высокую схожесть кадров.

В данном методе каждый кадр поврежденного видеофрагмента сравнивается с w кадрами из оригинального фрагмента. В работе [5] размер окна w устанавливается вручную, исходя из структуры GoP для конкретных рассматриваемых в этой работе видеофрагментов. В работе [2] авторы определяют w , как увеличенное на единицу число потерянных при передаче кадров, которое может быть найдено как разность между числом кадров в оригинальном видеофрагменте и числом кадров в полученном фрагменте.

Вычислительная сложность такого метода растет линейно от размера окна w . Для ускорения процесса выравнивания видеофрагментов в работе [4] предлагается пересчитывать значение w с каждым найденным соответствием между двумя кадрами. Изначально w так же равно числу потерянных кад-



Рис. 6. Выравнивание видеофрагментов с использованием скользящего окна.

ров плюс один. На каждом шаге работы алгоритма кадр из окна в оригинальном фрагменте с максимумом PSNR относительно рассматриваемого кадра из полученного фрагмента выбирается как соответствующий. Кадры, находящиеся между этим выбранным кадром и кадром, выбранным на предыдущем шаге алгоритма, считаются потерянными. После этого размер окна w уменьшается на число кадров, которые были определены, как потерянные. Однако, такой метод выравнивания не может гарантировать установление правильного соответствия между кадрами и ошибка, произошедшая в начале процесса, приведет в дальнейшем к установлению соответствия между сдвинутыми относительно друг друга кадрами. Рассмотрим изображенный на рисунке 6 пример. Кадры с одинаковыми номерами на рисунке в действительности соответствуют друг другу. В полученном видеофрагменте 4-й и 5-й кадры отсутствуют, и за 3-м кадром следует 6-й. Однако из-за искажений в видеоизображении этот кадр наиболее похож на 8-й кадр из оригинального фрагмента, и они ставятся в соответствие друг другу. На следующем шаге алгоритм ищет соответствующий кадру с номером 7 из полученного фрагмента кадр из окна, начинающегося с кадра №9, что совершенно неверно.

Избежать ошибок подобного рода позволяет алгоритм, разработанный в Hewlett-Packard Lab [3] при допущении о том, что значение PSNR для поврежденного видеофрагмента будет максимально только тогда, когда соответствие между всеми парами кадров установлены совершенно верно. На первый взгляд, такой подход к решению задачи выравнивания видеофрагментов существенным образом увеличит вычислительную сложность процесса, поскольку для определения соответствия между кадрами необходимо будет рассмотреть все возможные паросочетания кадров. Однако, с использованием методов динамического программирования авторам работы удалось добиться вычисления PSNR за полиномиальное время, то есть с той же сложностью, что и метод, предложенный в работе [4]. Для еще большего ускорения алгоритма авторы предлагают ограничить размер окна значением $w = 5$, предполагая, что этого будет достаточно для верного выравнива-

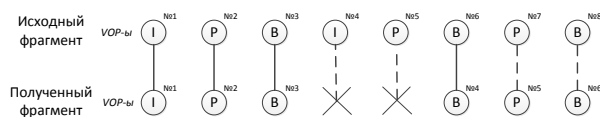


Рис. 7. Выравнивание видеофрагментов на первом шаге предлагаемого метода.

ния. Однако, данное допущение неверно, например, в случае коррелированных ошибок.

3. Предлагаемый метод

В данном разделе мы приведем описание предлагаемого нами метода определения соответствия между кадрами полученного и кадрами оригинального видеофрагментов.

В отличие от методов, описанных в предыдущем разделе, рассматривавших только несжатый (декодированный) поток изображений, предлагаемый метод анализирует и закодированный MPEG-4 и декодированный «сырой» видеопоток. Анализ закодированного видеопотока позволяет определить множество потерянных в ходе передаче кадров. Затем эта информация будет использована для определения соответствия между кадрами.

Наш метод включает в себя три этапа.

На первом этапе проводится поиск в оригинальном и полученном видеофрагментах уникальных последовательностей байт, с которых начинаются VOP-ы. Каждая найденная последовательность предваряет собой VOP, который в свою очередь соответствует кадру видеоизображения. Затем VOP-ы занумеровываются по порядку. Далее, используя допущение о том, что несколько байт, следующие сразу за начальной последовательностью, уникальным образом характеризуют VOP, и, сравнивая эти несколько байт, для каждого VOP из полученного видеофрагмента определяется соответствующий ему VOP (VOP, с которым совпадают указанные байты) из оригинального фрагмента, а также номера потерянных при передаче VOP-ов. На рисунке 7 изображен пример. В ходе передачи два VOP-а были потеряны. Предположим, что соответствующие VOP-ы для первых трёх VOP-ов полученного видеофрагмента уже определены. Для нахождения в оригинальном фрагменте VOP, соответствующего 4-му VOP из поврежденного фрагмента, необходимо сравнить несколько байт, следующих за стартовой последовательностью VOP №4, с соответствующими байтами VOP-ов из оригинального фрагмента, начиная с VOP №4. Таким образом будет установлено, что рассматриваемому VOP соответствует VOP из оригинального потока с номером 6.

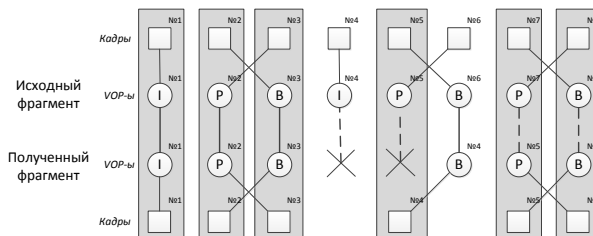


Рис. 8. Выравнивание видеофрагментов на втором шаге предлагаемого метода.

На втором этапе, зная номера потерянных при передаче VOP-ов и учитывая тот факт, что порядок (де)кодирования отличается от порядка воспроизведения, как было описано в разделе 2.1, определяется соответствие между кадрами поврежденного и исходного видеофрагментов. Для определения номеров потерянных кадров, мы рассматриваем по одному каждый потерянный VOP. Допустим очередной потерянный VOP имеет номер i . Если это B-VOP, то, как показано на рисунке 8, кадр из оригинального видеофрагмента, которому соответствует рассматриваемый потерянный VOP, имеет номер $i - 1$. В противном случае, если рассматриваемый VOP принадлежит типу I или P, то номер соответствующего кадра $i + k$, где k – это количество B-VOP-ов, которые следуют непосредственно за VOP с номером i в закодированном потоке. Так, например, кадру с номером 4 в полученном видеофрагменте соответствует 5-й кадр из исходного фрагмента.

На третьем этапе покадрово сравнивая оригинальный и полученный видеофрагменты с учетом установленных соответствий, мы вычисляем значение метрики для каждой пары кадров, исключая те кадры, которые были потеряны. В этом месте есть по меньшей мере два возможных способа для оценки вклада потерянных кадров в качество видеоизображения.

Первый способ подразумевает сравнение кадра из оригинального фрагмента, который был потерян при передаче, с кадром, который видит пользователь, просматривающий полученный видеофрагмент, то есть фактически с предыдущим успешно декодированным кадром. Но поскольку кадры, следующие друг за другом, как правило, отличаются незначительно, то если, кадр потерян, пользователь не увидит существенных искажений в видеоизображении, хотя метрика предсказала бы плохое качество.

Однако в данной работе мы используем второй способ и не вычисляем значение метрики для потерянных кадров из оригинального видеофрагмента. Данный подход является оправданным в случае, ес-

ли передача происходила без длительных последовательных потерь кадров.

Следует уточнить, что стандарт MPEG-4 определяет в заголовке VOP поля переменной длины, зависящей от значений полей сущностей верхних уровней (Video Object, Video Sequence). То есть заголовки VOP-а не обладают фиксированным размером или каким-либо маркером конца заголовка, что затрудняет анализ видеофрагмента. Число начальных байт заголовка, которые используются для определения соответствия между VOP-ми оригинального и полученного видеофрагментов, является настраиваемым параметром. Его значение зависит от структуры рассматриваемых видеофрагментов, в частности оно должно быть меньше размера самого короткого (в байтах) VOP-а во фрагменте. В проведенных в данной работе экспериментах это число было равно 6.

4. Численные результаты

В этом разделе мы подтвердим эффективность предложенного нами метода. Для этого мы реализуем метод в программной среде MATLAB [13], выберем несколько видеофрагментов, внесем в них искажения и затем оценим качество полученных видеофрагментов. Мы исследуем влияние, оказываемое используемым алгоритмом выравнивания видеофрагментов на значение метрики качества видео, а также какое время требуется для работы алгоритма и расчета метрики. В данном исследовании сравниваются следующие способы расчета метрики MSE с различными механизмами выравнивания:

1. NAMSE (Not Aligned MSE) – метрика MSE без использования какого-либо алгоритма выравнивания видеофрагментов. Если оригинальный фрагмент содержит больше кадров, чем поврежденный, то его окончание игнорируется.
2. HPMSE (Hewlett Packard MSE) – метрика MSE с использованием алгоритма выравнивания, разработанного в HP Lab [3] и описанного в разделе 2.3. Поскольку размер скользящего окна, ограниченный значением, меньшим, чем количество потерянных кадров, может привести к неверному определению соответствия между кадрами, то данная оптимизация предложенная авторами алгоритма нами не рассматривается.
3. FMSE (Fast MSE) – метрика MSE с алгоритмом, предложенным в нашей работе.

Для демонстрации того факта, что предложенный нами алгоритм совместим с другими типами метрик, мы также используем его вместе с метрикой SSIM и далее будем обозначать такой метод оценки качества видео FSSIM (Fast SSIM).

Большинство баз видеофрагментов, на которые ссылаются в литературе, содержат короткие видеофрагменты (их длительность не превышает несколько секунд) в силу того, что они ориентированы на сравнение видеокодеков. Однако, для исследования эффективности различных механизмов передачи видеопотоков в беспроводных сетях необходимо использовать видеофрагменты длительностью порядка нескольких десятков секунд (до одной минуты) для исключения влияния переходных процессов. Для нашего исследования мы выбрали три видеофрагмента: рекламный ролик фильма «Аватар», обозначаемый далее “Movie”, отрывок из новостной программы, обозначаемый далее “News”, и фрагмент футбольного матча, обозначаемый далее “Football”. Параметры используемых видеофрагментов перечислены в таблице 1.

Видеофрагменты и исходный программный код доступен по адресу <http://lab18.iitp.ru/pbl/vow.html>

4.1. Валидационные эксперименты

Для валидации предложенного нами алгоритма проведем две серии экспериментов, внося различным образом искажения в видеофрагменты и сравнивая затем качество полученных видеофрагментов при помощи разных методов, описанных выше.

Первая серия экспериментов проводится с целью установления того, что алгоритм находит верное соответствие между кадрами оригинальных и поврежденных фрагментов и полученные таким образом значения метрики являются ожидаемыми. Для этого мы выберем фрагменты с наименьшим разрешением 640 на 360 пикселей и длительностью 8 секунд. Затем мы повреждаем фрагменты путем удаления одного VOP какого-либо типа. Номер удаляемого VOP выбирается случайным образом.

Если из фрагмента был удален B-VOP, то пользователь может не заметить никаких искажений в видеоизображении. Так как на удаленный B-VOP нет ссылок из других VOP (информация из B-VOP не используется для декодирования других VOP-ов), то все остальные VOP будут декодированы успешно. Единственным эффектом, вызванным отсутствием данного B-VOP, будет удвоение временного интервала между двумя соседними кадрами в месте удаления VOP. Как было сказано в разделе 3, данный эффект не вносит никакого вклада в значение метрики, которое в случае использования MSE будет равно 0, а в случае использования SSIM, равно 1 для всех поврежденных таким образом видеофрагментов.

При удалении P-VOP, как было отмечено в разделе 2.1, все VOP-ы от следующего за удаленным до следующего I-VOP не будут корректно декодированы и следовательно видеоизображение будет иска-

Название фрагмента	Football	Movie	News
Размер GoV	50	25	50
Число B-VOP-ов между двумя P-VOP-ми	2	1	2
Число кадров в секунду (FPS)	24		
Длительность, с	8, 15, 30, 60, 120,		
Разрешение	640x360, 960x540, 1280x720, 1920x1080		

Таблица 1. Значения параметров видеофрагментов.

жено. То есть для этих VOP-ов значение MSE будет больше 0, а значение SSIM меньше 1.

В случае удаления I-VOP наблюдаются эффекты аналогичные предыдущему случаю. Единственным отличием является то, что повреждается вся GoV и значение метрики показывает качество хуже, чем в предыдущем случае.

Рисунок 9 демонстрирует значения качества видеоизображения, полученные с помощью разных метрик. Данные результаты доказывают необходимость принятия во внимание потери целых кадров при оценки качества видеоизображения, так как значения NAMSE совершенно не коррелируют с количеством внесенных искажений и сильно завышены вне зависимости от типа удаляемого VOP. В частности, значение NAMSE не равно 0 в случае удаления B-VOP, в отличие от HPMSE и FMSE. Кроме этого, можно заметить, что значения HPMSE и FMSE равны, из чего следует, что наш метод верно определяет соответствие между кадрами. Значения метрик FMSE, HPMSE и FSSIM соответствуют нашим предположениям, описанным выше, во всех случаях, кроме фрагмента “News”. Так как видеоизображение в этом видеофрагменте большей частью статично (движение есть только на лице ведущего и время от времени появляется текст), то удаление одного P- или I-VOP ведет к незначительному изменению качества. В используемом нами фрагменте новостной передачи I-VOP встречаются достаточно редко и они содержат не такой значительный объем информации, в то время как, объем информации в P-VOP выше среднего. В такой ситуации при удалении I-VOP следующий за ним P-VOP(s) быстро восстановит(-ят) качество изображения, но удаление P-VOP приводит к более сильному ухудшению качества. Согласованность результатов, полученных при помощи FMSE, HPMSE и FSSIM доказывает корректность работы нашего алгоритма в рассматриваемых сценариях.

В описанных выше экспериментах мы валидируем наш метод в случае удаления одного целого VOP из закодированного видеопотока. Хотя и данные эксперименты дают предсказуемый результат, это не гарантирует работоспособность предлагаемого метода в случае передачи видеопотока через сеть с возможными потерями пакетов. Поэтому во *второй серии экспериментов* мы рассматриваем пере-

дачу видеоизображения в режиме реального времени между двумя Wi-Fi станциями. Мы организуем сервер потокового вещания при помощи программы VLC на первой станции для передачи видеофрагмента длительностью одна минута, используя IP/UDP/RTP/MPEG-TS инкапсуляцию, описанную в разделе 2.1. На второй станции программа VLC выполняет роль клиента и записывает принимаемый поток в файл, который и рассматривается в виде поврежденного видеофрагмента.

Данный сценарий моделируется при помощи имитационного комплекса, часто используемого для исследований процесса передачи видеопотоков по беспроводным сетям. Комплекс состоит из сетевого эмулятора netns [14] и среды имитационного моделирования ns-3 [15], взаимодействующих при помощи контроллера экспериментов пері [16]. Netns представляет собой сетевой эмулятор, который позволяет создавать виртуальные сетевые интерфейсы и выполнять приложения, установленные в операционной системе. При этом netns использует реальные протоколы сетевого, транспортного уровня и уровня приложения из системы Linux. Эмулятор netns позволяет запустить приложение VLC, с помощью которого происходит вещание видеопотока в сеть, а также воспроизведение данного видеопотока на получателе. Для моделирования протокола MAC уровня IEEE 802.11, а также беспроводной среды передачи данных, используется дискретно-событийная среда имитационного моделирования ns-3. Потери пакетов и временные задержки при передаче пакета моделируются с помощью моделей распространения сигнала, являющихся частью данной среды. Таким образом, организованный имитационный комплекс предоставляет довольно обширные и гибкие возможности для моделирования реальных сетей передачи видеопотоков.

Мы настраиваем модель распространения радиосигнала в среде ns-3 таким образом, чтобы доля потерянных пакетов составляла 0.5%. Кроме случайных потерь в полученных видеофрагментах могут встречаться несколько последовательных потерь целых кадров в начале и конце фрагмента, которые вызваны инициализацией и завершением процесса потокового вещания: клиент запускается чуть позже относительно сервера и поэтому первые несколько кадров не получают клиентом. Потеря I-VOP в

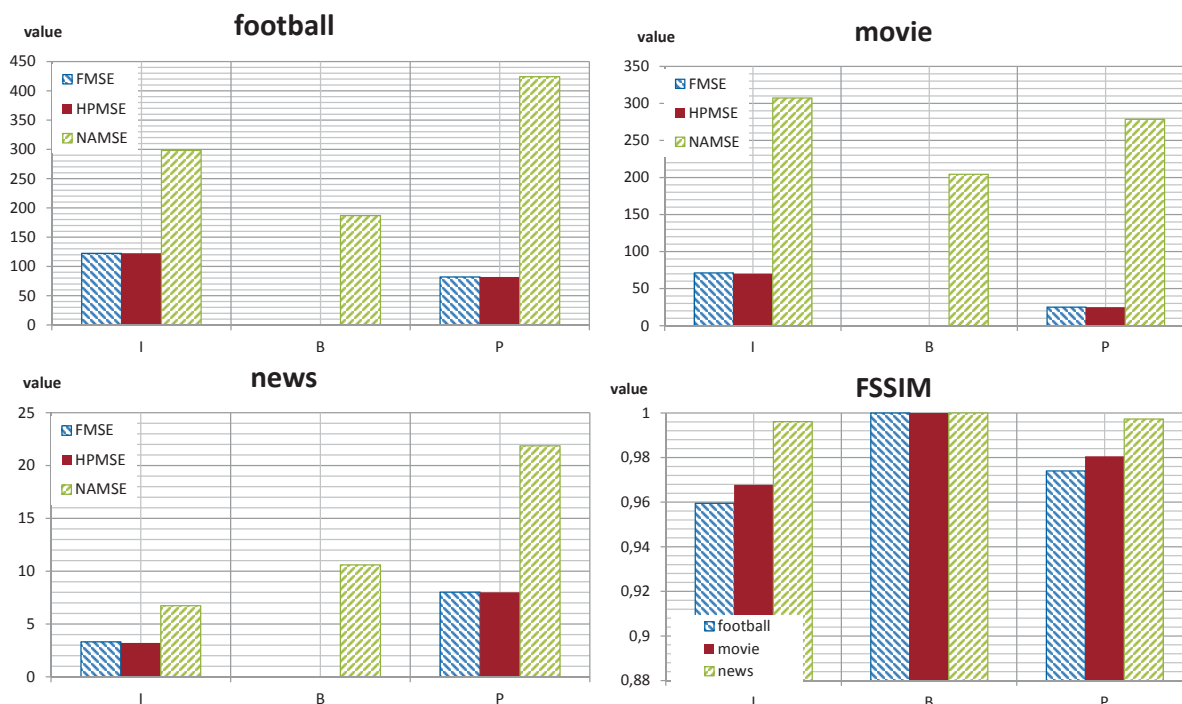


Рис. 9. Качество видеоизображения в зависимости от типа удаленного VOP.

начале фрагмента существенным образом сказывается на качестве видеоизображения. Видеофрагменты, содержащие большое количество движения (например, “Football”) восстанавливают свое качество быстрее, чем фрагменты, где движения меньше (например, “News”). Это объясняет значения MSE, продемонстрированные на рисунке 10. Как мы знаем из предыдущих экспериментов, на качестве фрагмента “News” потери VOP-ов сказываются меньше, чем на фрагменте “Football”. Однако, потеря первого I-VOP более существенная для видеофрагмента “News” и полностью искажает видеоизображение в течение первой GoV. Схожесть результатов, полученных при помощи FMSE и HPMSE, доказывает корректность предлагаемого нами метода.

Однако метрика SSIM показывает иные результаты: качество фрагмента “News” выше, чем качество других видеофрагментов. Данный факт не является противоречием, так как метрика SSIM основана на другом принципе сравнения изображений: рассматривается структура изображения, а не квадратичная ошибка.

4.2. Анализ производительности

Основным недостатком метода HPMSE является его высокая вычислительная сложность. Например, в разделе 4.1 мы оцениваем качество видеофрагментов длительностью в одну минуту, переданного по

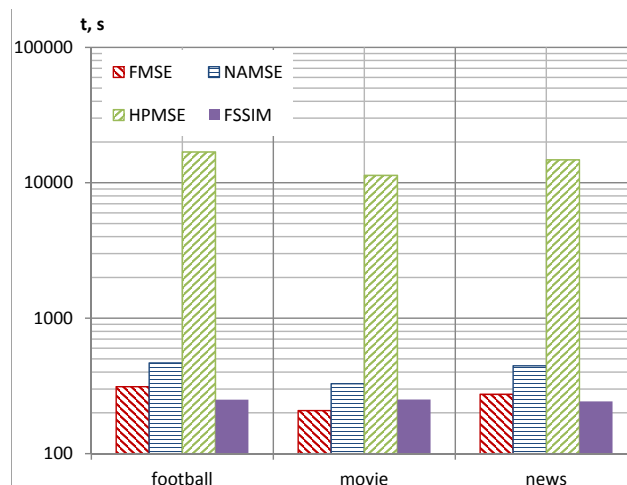


Рис. 11. Время расчета значений метрик качества видеоизображения для различных видеофрагментов, переданных по сети.

беспроводной сети и при помощи HPMSE это требует 3-5 часов (!) на компьютере с процессором Intel Core i3 3.3 ГГц. В то же время, как показано на рисунке 11 расчет значений FMSE и FFSIM занимает 3-5 минут.

Для объяснения данного результата проведем анализ сложности алгоритмов NAMSE, HPMSE и

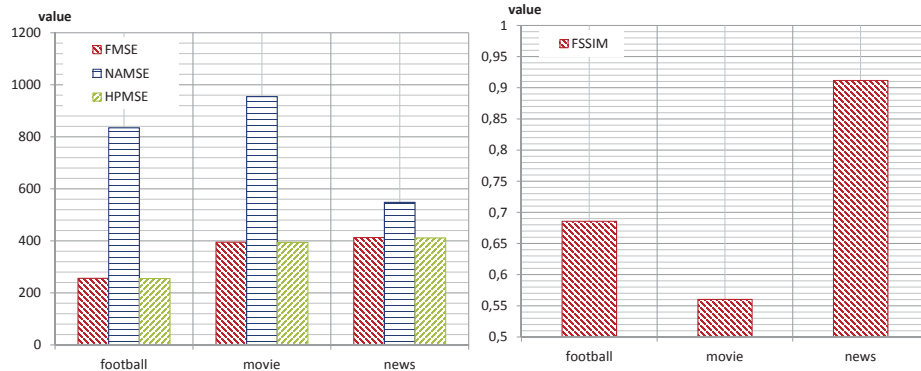


Рис. 10. Значения метрик качества видеоизображения для различных видеофрагментов.

FMSE. Введем следующие обозначения. Обозначим за S количество кадров (или VOP-ов) в оригинальном фрагменте и за L число потерянных кадров. Разрешение изображения $M \times N$.

Расчет MSE для пары кадров требует $O(MN)$ операций, то есть без использования какого-либо алгоритма соотнесения кадров друг с другом полученное значение метрики NAMSE для всего видеофрагмента потребует $O(MN(S-L))$ операций.

По умолчанию, метрика HPMSE, предложенная в работе [3], требует $O(MN(S-L)(L+1))$ операций. Даже с размером окна $w < L+1$ это в $w+1$ раз больше операций, чем для NAMSE.

Первый этап предлагаемого в этой работе алгоритма выравнивания кадров требует времени $O(s)$, где s – размер в байтах закодированного оригинального видеофрагмента. Поскольку кодирование предполагает сжатие, то $s \ll MNS$. Второй этап занимает $O(S)$ времени. Как и в случаях, описанных выше, значение MSE для двух кадров вычисляется за $O(MN(S-L))$ операций. Таким образом, в отличие от HPMSE, сложность предлагаемого метода не отличается значительным образом от NAMSE.

Проведем более подробный анализ производительности. В первую очередь исследуем зависимость времени расчета метрики от числа подряд потерянных кадров. Для этого рассмотрим фрагмент “Movie” с разрешением 640 на 360 пикселей и длительностью 8 секунд³ и будем удалять из него L VOP-ов, делая это последовательно и начиная с некоторого фиксированного номера, например 90 (при размере GOV равном 25, 90-й VOP находится в середине GOV, что дает возможность исследовать эффекты, возникающие при потере, как только середины, так конца GOV). Как видно из левой части рисунка 12, рассуждения выше подтвер-

ждаются и расчет значений FSSIM и NAMSE занимает приблизительно одинаковое время, незначительно уменьшающееся с ростом L . Время расчета HPMSE линейно растет вместе с количеством удаленных VOP-ов⁴. Однако линия FMSE имеет ярко выраженную ступень при изменении L с 8 до 10. Этот эффект объясняется тем, что расчет MSE в случае, если два изображения совпадают, не почти не требует времени. 5-й по счету удаляемый имеет номер 99 во фрагменте “Movie”. Когда он удаляется, соответствующая GoV повреждается и время расчета метрики возрастает. Правая часть рисунка 12 показывает, что время необходимо для соотнесения кадров незначительно по сравнению с временем необходимым для подсчета MSE для всех соответствующих пар кадров. Таким образом, мы доказали, что предложенный нами метод позволяет ускорить время расчета метрики в 10-80 раз в зависимости от числа потерянных кадров.

Наконец, мы исследуем зависимость времени расчета метрики от длительности рассматриваемого видеофрагмента и его разрешения. Зафиксировав $L = 4$ и повторяя предыдущий эксперимент, мы получаем результаты, изображенные на рисунке 13, которые еще раз подтверждают факт значительного ускорения времени расчета значения метрики, что делает возможным оценивать качество видеоизображений высокого разрешения (HD-видео) и длительных видеофрагментов, переданных по беспроводным сетям.

5. Заключение

Качество потокового видео, передаваемого по беспроводным сетям, существенно падает при значительных потерях пакетов. Эти потери приводят не только к частичному искажению видеоизображе-

³В данном эксперименте длительность и разрешение фрагмента не имеют существенного значения, поэтому они выбираются минимальными для сокращения времени проведения эксперимента.

⁴Для того, чтобы построить все линии на одном графике, мы выбрали логарифмический масштаб для оси ординат.

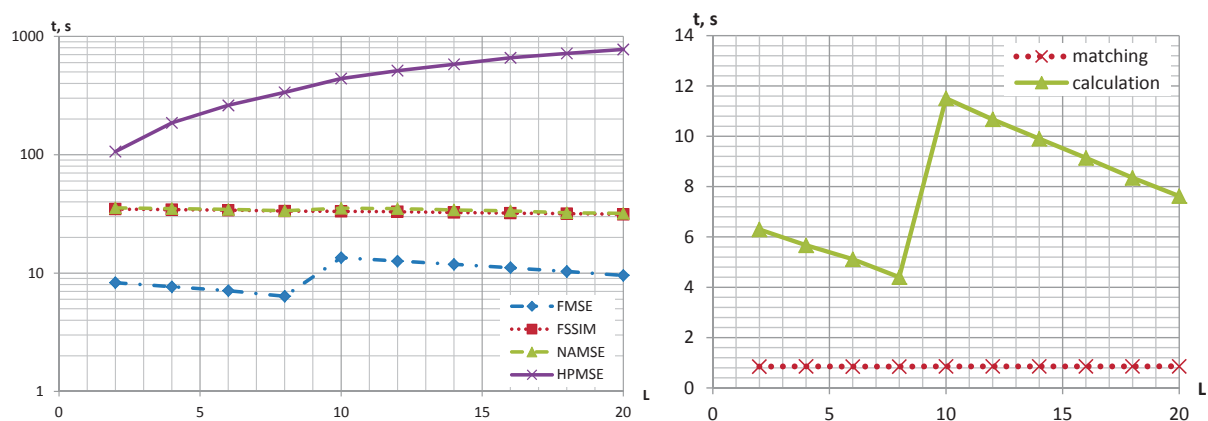


Рис. 12. Время расчета значений метрик качества видеоизображения в зависимости от числа подряд потерянных кадров.

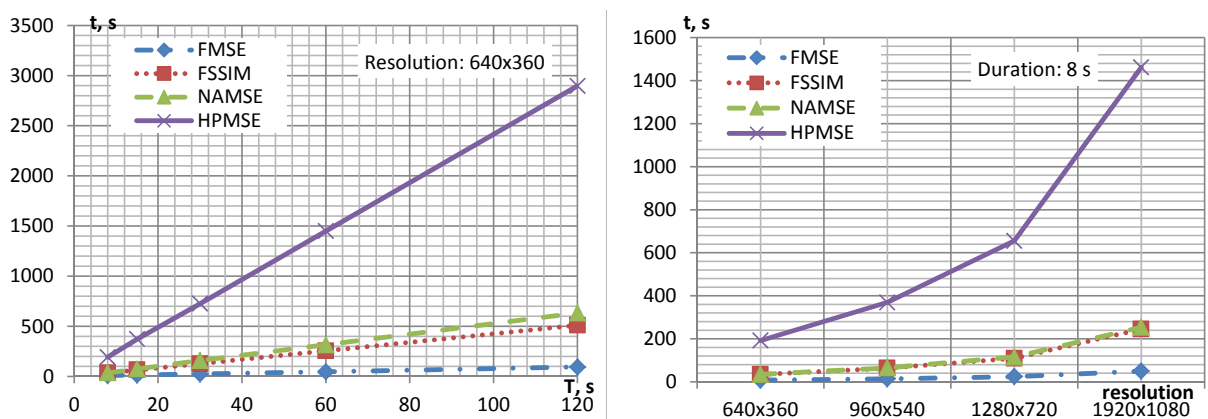


Рис. 13. Время расчета значений метрик качества видеоизображения в зависимости от длительности и от разрешения рассматриваемого видеофрагмента.

ния и к появлению артефактов, но и к потере целых кадров. Без использования какого-либо метода соотнесения кадров оригинального и поврежденного видеофрагментов традиционно используемые метрики оценки качества видеоизображения не учитывают эффекты, вызванные потерями кадров и следовательно предоставляют неверную оценку субъективного качества изображения. Существующие методы соотнесения кадров видеофрагментов оперируют только несжатым видеопотоком и слишком медленны, что ограничивает исследователей в возможностях по оценке качества видеопотоков, переданных по сети. Мы разработали новый метод выравнивания видеофрагментов и определения соответствия между кадрами исходного и полученного видеофрагментов, который может быть использован для улучшения различных типов метрик. В качестве примера, мы рассмотрели работу нашего метода совместно с метриками MSE и SSIM и доказали, что метод позволяет существенно снизить время необходимое для получения оценки качества видеофрагмента. Это позволяет проводить оценку качества видеофрагментов высокого разрешения в широком спектре сценариев передачи видеопотоков по беспроводным сетям, что в свою очередь даст возможность исследовать и реализовывать инновационные решения по улучшению способности беспроводных сетей по передаче все возрастающего количества видеотрафика.

Список литературы

- [1] Bell Labs Video Traffic Study "Video Shakes Up IP Edge". — URL: http://www.alcatel-lucent.com/wps/DocumentStreamerServlet?LMSG_CABINET=Docs_and_Resource_Ctr&LMSG_CONTENT_FILE=White_Papers/Video_Shakes_Up_IP_Edge_EN_Whitepaper.pdf.
- [2] Pullano Valentina, Vanelli-Coralli Alessandro, Corazza Giovanni E. PSNR evaluation and alignment recovery for mobile satellite video broadcasting // Advanced Satellite Multimedia Systems Conference (ASMS) and 12th Signal Processing for Space Communications Workshop (SPSC), 2012 6th / IEEE. — 2012. — P. 176–181.
- [3] Metrics for evaluating video streaming quality in lossy IEEE 802.11 wireless networks / An Chan, Kai Zeng, Prasant Mohapatra et al. // INFOCOM, 2010 Proceedings IEEE / IEEE. — 2010. — P. 1–9.
- [4] Aligned-PSNR (APSNR) for Objective Video Quality Measurement (VQM) in video stream over wireless and mobile network / YA Syahbana, AA Rahman, KA Bakar et al. // Information and Communication Technologies (WICT), 2011 World Congress on / IEEE. — 2011. — P. 330–335.
- [5] Estimation of the Visual Quality of Video Streaming Under Desynchronization Conditions / AA Atayero, OI Sheluhin, YA Ivanov, AA Alatishe // International Journal. — 2011. — Vol. 2.
- [6] ITU-T. — Recommendation H. 264, Advanced video coding for generic audiovisual services, 2012.
- [7] ISO/IEC. — 14496-2 Coding of audio-visual objects Part 2: Visual, 1999.
- [8] Greengrass Jason, Evans John, Begen Ali C. Not all packets are equal, part i: Streaming video coding and sla requirements // Internet Computing, IEEE. — 2009. — Vol. 13, no. 1. — P. 70–75.
- [9] Greengrass Jason, Evans John, Begen Ali C. Not all packets are equal, part 2: The impact of network packet loss on video quality // Internet Computing, IEEE. — 2009. — Vol. 13, no. 2. — P. 74–82.
- [10] ITU-R Recommendation BT.500-11. — Methodology for the subjective assessment of the quality of television pictures, 2002.
- [11] Wireless video quality assessment: A study of subjective scores and objective algorithms / A. Moorthy, K. Seshadrinathan, R. Soundararajan, A. Bovik // IEEE Transactions Circuits and Systems for Video Technology. — 2010. — April. — Vol. 20, no. 4.
- [12] Image quality assessment: From error visibility to structural similarity / Zhou Wang, Alan C Bovik, Hamid Rahim Sheikh, Eero P Simoncelli // IEEE Transactions on Image Processing. — 2004. — Vol. 13, no. 4. — P. 600–612.
- [13] The Matlab environment. — URL: <http://www.matlab.com/>.
- [14] Netns emulator. — URL: <http://nepihome.org/wiki/netns/>.
- [15] The ns-3 network simulator. — URL: <http://www.nsnam.org/>.
- [16] Network Experimentation Programming Interface (NEPI). — URL: <http://www.nepihome.org/>.