

# Программа расчета метрики MSE качества переданного по сети видеопотока\*

Дмитрий Банков  
МФТИ (ГУ), ИППИ РАН  
bankov@iitp.ru

Андрей Гуштин  
ИППИ РАН  
guschin@iitp.ru

Евгений Хоров  
ИППИ РАН  
khorov@iitp.ru

## Аннотация

В статье решается задача вычисления метрики MSE видеопотока, повреждённого в процессе передачи по сети. Представлен алгоритм сопоставления кадров между оригинальным и повреждённым видеопотоком, учитывающий возможные потери при передаче видеофайлов по сети.

## 1. Введение

Объем передаваемого по сети видеотрафика стремительно растет. Согласно прогнозам компании Cisco [1], в 2017 году доля видеоданных будет составлять почти 70% от всего трафика. Передача видеопотоков в режиме реального времени налагает ограничения на средства, используемые при передаче данных. Например, видеопотоки должны передаваться с ограниченной задержкой. Помимо этого видеопотоки являются нестационарными (интенсивность видеопотоков непостоянна во времени), а также неоднородными (потери пакетов, включающих в себя различные части видеопотока, по-разному влияют на качество сформированного видеоизображения).

Перечисленные особенности видеопотоков делают актуальной задачу оптимизации процесса передачи видеотрафика по сети. Из-за того, что в процессе передачи по сети видеопотоки повреждаются, исследователям необходим инструмент оценки качества видеопотоков. Существующие инструменты, о которых пойдет речь в разделе 2, обладают рядом недостатков, ограничивающих область их применения. Для преодоления этих недостатков реализована программа, описанная в разделе 4. Основными требованиями к программе являются возможность сравнения видеоизображений, состоящих из различного количества статичных изображений (кадров), а также независимость времени работы программы

от количества потерянных кадров. Для того, чтобы удовлетворить поставленным требованиям, разработан метод сопоставления кадров, описанный в разделе 4.2.2. Помимо этого, в разделе 4.2.4 описан метод, с помощью которого можно определять применимость программы для сравнения конкретных видеоизображений.

## 2. Существующие методы оценки качества видеоизображений

Большинство метрик качества видеоизображений основано на метриках качества статичных изображений, и поэтому видеоизображения сравнивают покадрово. В связи с этим метрики применимы только для сравнения видеоизображений, имеющих одинаковое количество кадров.

Примером метрики, широко используемой для сравнения качества двух видеоизображений, является среднеквадратичная ошибка (от англ. Mean Squared Error – MSE). Для кадров  $X$  и  $Y$  размером  $M \times N$  пикселей значение метрики MSE задается формулой:

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

где  $X(i, j)$  – это значение компоненты яркости пиксела  $(i, j)$  изображения  $X$ .

Значение метрики MSE для видеофрагментов рассчитывается как среднее значение MSE для соответствующих кадров, из которых состоят сравниваемые видеофрагменты.

Аналогом MSE является метрика PSNR (Peak Signal-to-Noise Ratio или отношение пикового сигнала к шуму), связанная с MSE следующим отношением:

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

где  $A$  – максимальное значение яркости для пиксела, например 255 для 8-битного YUV-изображения.

\*Исследование выполнено при финансовой поддержке РФФИ в рамках гранта 12-07-33067 мол\_а\_вед.

Для двух одинаковых изображений значение PSNR равно бесконечности.

В процессе передачи видеопотока по сети могут быть потеряны некоторые кадры. Из-за этого метрики MSE и PSNR неприменимы для оценки качества таких видеопотоков.

Для видеофайлов, имеющих различное количество кадров, нужен алгоритм сопоставления кадров между оригинальным видеофайлом и повреждённым. Один из таких алгоритмов разработан в Hewlett-Packard Lab [2]. Этот алгоритм основан на предположении, что значение метрики PSNR для поврежденного видеофайла максимально, тогда и только тогда, когда соответствие между парами кадров составлено верно. Однако время выполнения данного алгоритма растёт с количеством потерянных кадров: программа, реализующая данный алгоритм выполняет оценку качества видеофрагмента длительностью всего 8 секунд с частотой 24 кадра в секунду и разрешением  $640 \times 360$  за 7 минут, если удалены 10 кадров подряд. Тесты проводились на компьютере с процессором Intel Core i3 3.3 ГГц.

Ещё один метод сопоставления кадров разработали в Институте проблем передачи информации РАН [3]. Этот метод основан на сравнении блоков данных, начинающихся с заголовков закодированных кадров в видеофайлах. Программа, реализующая данный метод, работает быстрее, чем упомянутая в данном разделе программа от Hewlett-Packard Lab, и время её выполнения не зависит от числа потерянных кадров. Однако недостатком этой программы является то, что она использует функции для работы с видеофайлами, которые являются частью пакета Matlab [4], и которые не всегда правильно декодируют поврежденные видеофайлы.

Другой метод оценки качества видеопотоков, переданных по сети, используется в наборе инструментов *evalvid*, разработанном в Technische Universität Berlin [5]. В этом наборе присутствует программа, которая по данным об отправленных и о полученных по сети пакетах определяет кадры, утерянные или повреждённые в процессе передачи видеофайла и удаляет эти кадры из оригинального видеофайла так, чтобы получился видеофайл, сформированный на приёмнике. Затем требуется декодировать оригинальный и повреждённый видеофайл при помощи подходящего декодера так, чтобы количество кадров в получившихся видеоизображениях было одинаковым. После этого можно использовать ещё одну программу, включённую в инструментарий *evalvid*, которая вычисляет значение PSNR для декодированных видеофайлов. Идея использовать информацию о процессе передачи видеофайлов значительно упрощает процесс определения исчезнувших кадров, однако утилита, которая генерирует повреждённый видеофайл не работает при потере определенных кад-

ров.

### 3. Постановка задачи

Цель нашей работы состоит в реализации программы расчета метрики качества переданного по сети видеопотока, которая:

1. может сравнивать видеофайлы имеющие разное количество кадров;
2. работает за время, линейно зависящее от количества кадров в неповрежденном видеопотоке и не зависящее от количества потерянных кадров;
3. правильно определяет номера потерянных кадров и правильно считает значение метрики MSE для декодированных кадров.

### 4. Описание программы

#### 4.1. Краткое описание программы

Описываемая программа считает метрику MSE для двух видеофайлов, закодированных согласно стандарту MPEG-4 part 2, первый из которых считается оригинальным, а второй – поврежденным видеофайлом. В качестве аргументов программа принимает:

1. имя оригинального видеофайла;
2. имя поврежденного видеофайла;
3. название файла, в который следует вывести отчет;
4. параметры  $\Delta$  и  $L_{min}$ , о которых рассказывается в разделах 4.2.3 и 4.2.4.

В процессе работы определяется соответствие между кадрами из разных видеофайлов, и для сопоставленных кадров вычисляется значение метрики. Программа выводит суммарное значение метрики на стандартный поток вывода, а в указанный файл – подробный отчет, представляющий собой таблицу, имеющую следующие графы:

**Nr** – номер кадра в оригинальном видеофайле в порядке воспроизведения;

**type** – тип кадра;

**dec1** – номер кадра в оригинальном видеофайле в порядке декодирования;

**dec2** – номер кадра в поврежденном видеофайле в порядке декодирования, соответствующего кадру из оригинального видеофайла. Если кадр был пропущен, то вместо номера выводится  $-1$ ;

**pos1** – положение в видеопотоке видеопакета, из которого получен кадр в оригинальном видеофайле;

**pos2** – положение в видеопотоке видеопакета, из которого получен кадр в поврежденном видеофайле, соответствующий кадру в оригинальном видеофайле. Если кадр был пропущен, то выводится  $-1$ ;

**MSE** – значение MSE для данных кадров. Если кадр был пропущен, то выводится 0.

## 4.2. Описание алгоритма

**4.2.1. Основные определения.** При кодировании видеоизображения с помощью MPEG 4, получается бинарный видеопоток, который при обработке разбивается на фрагменты. Фрагменты, содержащие информацию о кадрах, называются видеопакетами, причем каждому кадру можно поставить в соответствие ровно один видеопакет (обратное утверждение, вообще говоря, не верно).

Порядок видеопакетов в видеопотоке и кадров в видеоизображении может отличаться. Однако декодер видеопотока позволяет определить видеопакет, из которого был получен тот или иной кадр. Каждый видеопакет можно идентифицировать положением его первого байта в видеопотоке. Вместе с очередным кадром декодер выдает некоторую метаинформацию, содержащую в том числе и положение в видеопотоке видеопакета, из которого получен данный кадр.

**4.2.2. Алгоритм сопоставления кадров.** Для того, чтобы сравнивать изображения, предоставленные декодером, необходимо сопоставить номера изображений из разных видеофайлов. Искомое соответствие формируется в два этапа:

1. построение соответствия между видеопакетами оригинального и поврежденного видеопотока;
2. построение соответствия между кадрами и видеопакетами.

Построение соответствия схематически представлено на рисунке 1.

На первом этапе из обоих видеофайлов считываются видеопакеты. Видеопакеты сравниваются методом, описанным в разделе 4.2.3. Если видеопакеты признаны соответствующими друг другу, то соответствие запоминается, и считываются новые видеопакеты из обоих видеофайлов. В противном случае видеопакет из оригинального видеофайла запоминается как пропущенный, и считывается новый видеопакет из оригинального видеофайла. Считывание видеопакетов и их сравнение повторяется до

тех пор, пока хотя бы один из видеофайлов не будет считан полностью. В результате описанных действий получается таблица, дающая соответствие между видеопакетами из разных видеофайлов.

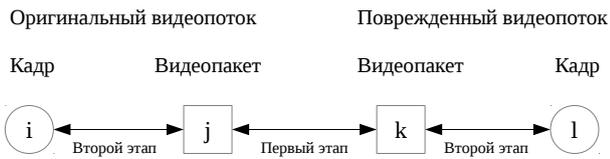
На втором этапе из обоих видеофайлов получают кадры. Для каждого кадра известно, из какого видеопакета он получен. Из соответствия между видеопакетами строится соответствие между кадрами, и для сопоставленных кадров считается значение MSE.

**4.2.3. Сравнение видеопакетов.** Данные в каждом видеопакете представляют собой одномерный массив, размер которого известен. Массив из оригинального видеофайла делится на отрезки длиной  $\Delta$  байт. В массивах из оригинального видеофайла и из поврежденного видеофайла ищется последовательность байт  $\psi = 00 : 00 : 01 : B6 : XX$ , где  $XX$  – произвольное значение от  $00$  до  $FF$ , которая является заголовком закодированного кадра в видеопотоке MPEG-4 part 2. Такая последовательность существует, т.к. в каждом видеопакете закодирован ровно один кадр.

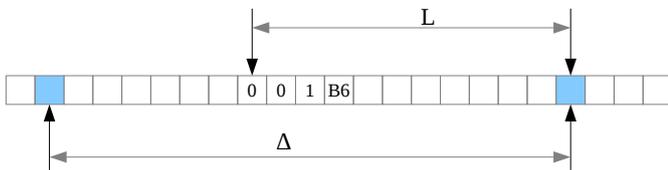
Будем сравнивать последовательности байт, начинающиеся с заголовка закодированного кадра, имеющие длину  $L$ , зависящую от положения последовательности  $\psi$  в данных из оригинального файла. Если последовательность  $\psi$  находится внутри отрезка (см. рис. 2), то длина буфера принимается равной количеству байт от начала последовательности до конца отрезка. Если последовательность находится на границе отрезков (см. рис. 3), то длина буфера принимается равной количеству байт от начала последовательности до конца следующего отрезка.

Буферы длины  $L$ , начинающиеся с последовательности  $\psi$  сравниваются. Если они совпадают, то видеопакеты считаются одинаковыми. Если они не совпадают, то ищется такое  $L^*$ , при котором буферы будут совпадать. Если такое  $L^*$  найдено, то видеопакеты считаются одинаковыми и выводится сообщение о возможной ошибке сравнения, в котором указано значение  $L^*$ . Сообщение о возможной ошибке сравнения также выводится в случае, если  $L$  меньше, чем  $L_{min}$ . В случае, если  $L^*$  не найдено, видеопакеты считаются различными.

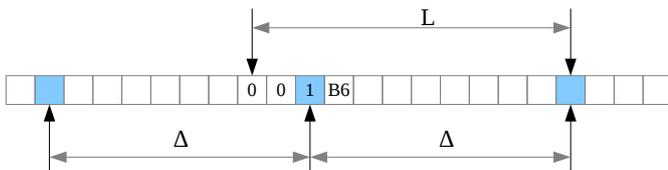
**4.2.4. Выбор параметров  $\Delta$  и  $L_{min}$ .** Программа разработана для сравнения видеофайлов, поврежденных в процессе передачи по сети, с исходными неповрежденными видеофайлами. Рассмотрим процесс передачи видеопотока с использованием стека из протоколов MPEG-TS, RTP, UDP и IP. При использовании этих протоколов видеопоток передается в TS-пакетах длины 184 байта. Это значит, что в поврежденном видеофайле данные могут пропадать только блоками по 184 байта. Таким образом, значе-



**Рис. 1. Построение соответствия между кадрами из разных видеофайлов.**



**Рис. 2. Определение сравниваемого буфера в случае, если заголовок находится внутри отрезка.**



**Рис. 3. Определение сравниваемого буфера в случае, если заголовок находится на границе отрезков.**

ние  $\Delta$ , принимаемое функцией сравнения видеопакетов, в случае использования протокола MPEG-TS, следует принять равным 184.

Значение  $L_{min}$  следует выбирать экспериментально исходя из количества кадров в оригинальном видеофайле и из того, насколько кадры похожи друг на друга.  $L_{min}$  следует принять равным наименьшему числу, для которого все блоки данных длины  $L_{min}$ , входящие в видеопакеты из оригинального видеофайла и начинающиеся с последовательности  $\Psi$ , попарно различны. Если такого числа нет, то при использовании представленной в статье программы для оценки качества видеопотока возможно неправильное сопоставление видеопакетов.

Для определения  $L_{min}$  была написана утилита, которая считывает пакеты из видеофайла, сохраняет из пакетов блоки данных заданной длины, начинающиеся на последовательность  $\Psi$ , и попарно их сравнивает. В случае, если одинаковые блоки найдены, выводятся номера пакетов, из которых получены блоки.

## 5. Численные результаты

В этом разделе сравниваются времена выполнения программы, описанной в статье (BVM), программы, реализованной в Hewlett-Packard Lab (HPMSE), и программы, созданной в Институте проблем передачи информации РАН (FMSE). Исследована зависимость времени выполнения программ от количества кадров, вырезанных из видеофайла подряд, а также от количества точек на один кадр.

### 5.1. Зависимость времени выполнения программы от количества вырезанных подряд кадров

На графиках 4, 5, 6 изображена зависимость времени выполнения программ BVM, FMSE и HPMSE соответственно от количества вырезанных подряд кадров. Использовались видеофайлы трёх типов, параметры которых представлены в таблице 1. Разрешение видеоизображений  $640 \times 360$  пикселей. В первую очередь следует отметить то, что время выполнения программы BVM уменьшается с ростом числа вырезанных кадров, что связано с тем, что для меньших пар кадров нужно считать значение MSE. Время выполнения программы HPMSE растёт линейно, что объясняется тем, что используется алгоритм скользящего окна, а величина скользящего окна растёт с количеством потерянных кадров. Линия FMSE имеет ярковыраженную ступень при изменении числа вырезанных кадров с 6 до 8. Этот эффект объясняется тем, что расчет MSE в случае, если два изображения совпадают, почти не требует времени. 7-й по счету удаляемый кадр имеет номер

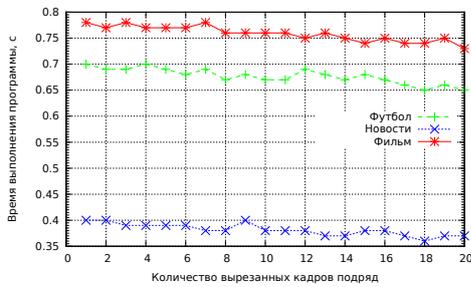


Рис. 4. Зависимость времени выполнения программы BVM от количества вырезанных подряд кадров.

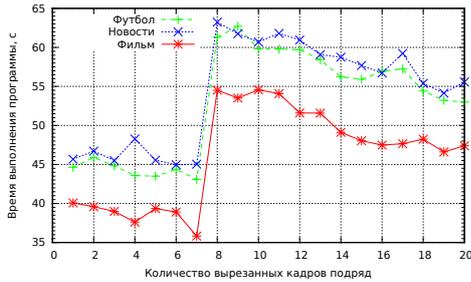


Рис. 5. Зависимость времени выполнения программы FastMSE от количества вырезанных подряд кадров.

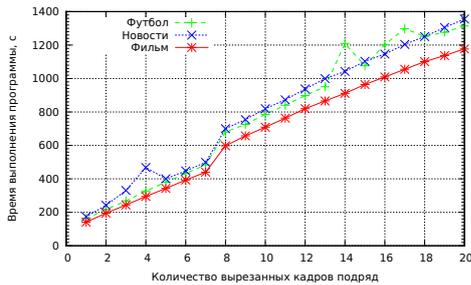


Рис. 6. Зависимость времени выполнения программы НРМСЕ от количества вырезанных подряд кадров.

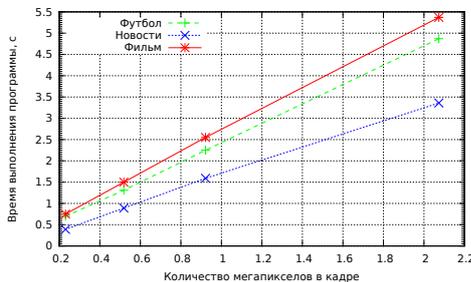


Рис. 7. Зависимость времени выполнения программы BVM от количества точек в кадре.

99 во фрагменте “Фильм”. Когда он удаляется, соответствующая GoV повреждается, и время расчета метрики возрастает.

Сравнение времени выполнения программ показывает, что время выполнения программы BVM на порядок меньше времени выполнения программы FMSE и на два порядка меньше времени выполнения программы НРМСЕ.

## 5.2. Зависимость времени выполнения программы от количества точек в кадре

На графиках 7, 8, 9 изображена зависимость времени выполнения программ BVM, FMSE и НРМСЕ соответственно от количества точек в кадре. Использовались видеофайлы, указанные в таблице 1. Время выполнения всех сравниваемых программ линейно возрастает с количеством точек. Линия BVM имеет меньший наклон, чем линии FMSE и НРМСЕ. Как и в предыдущей главе, видно, что время выполнения программы BVM на порядок меньше времени выполнения программы FMSE и на два порядка меньше времени выполнения программы НРМСЕ.

## Список литературы

- [1] Cisco Visual Networking Index: Forecast and Methodology, 2012–2017. — URL: [http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white\\_paper\\_c11-481360\\_ns827\\_Networking\\_Solutions\\_White\\_Paper.html](http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html).
- [2] 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.
- [3] Гуцин А. С. Кирьянов А. Г. Ляхов А. И. Хоров Е. М. Быстрая оценка качества видеопотоков, передаваемых по сети в условиях помех // Электросвязь. — 2013. — no. 8.
- [4] The Matlab environment. — URL: <http://www.matlab.com/>.
- [5] Klaue Jirka, Rathke Berthold, Wolisz Adam. Evalvid— A framework for video transmission and quality evaluation // Computer Performance Evaluation. Modelling Techniques and Tools. — Springer, 2003. — P. 255–272.

Название видеофайла	Футбол	Фильм	Новости
Размер GOV	50	25	50
Количество B-VOP между двумя P-VOP	2	1	2
Число кадров в секунду	25		
Длительность	8 секунд		
Разрешение	Различное: 640 × 360, 960 × 540, 1280 × 720, 1920 × 1080		

Таблица 1. Параметры использованных видеофайлов.

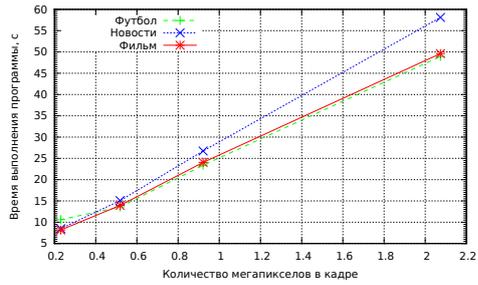


Рис. 8. Зависимость времени выполнения программы FastMSE от количества точек в кадре.

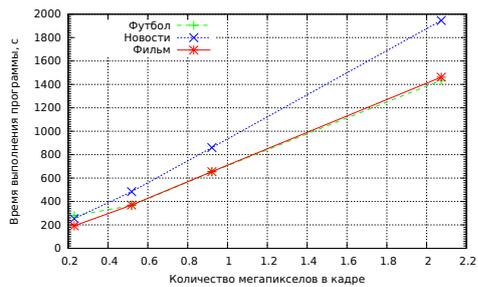


Рис. 9. Зависимость времени выполнения программы NPMSE от количества точек в кадре.