

Изучение возможности использования векторов движения сжатого видеопотока для его идентификации

Р.Ш. Фахрутдинов¹, А.Ю. Мирин¹

¹Санкт-Петербургский Федеральный исследовательский центр Российской академии наук,
Санкт-Петербург, 199178, Российская Федерация

*Адрес для переписки: fahr@cobra.ru

Информация о статье

Поступила в редакцию 10.08.2021

Поступила после рецензирования 26.10.2021

Принята к публикации 10.01.2022

Ссылка для цитирования: Фахрутдинов Р.Ш., Мирин А.Ю. Изучение возможности использования векторов движения сжатого видеопотока для его идентификации // Труды учебных заведений связи. 2022. Т. 8. № 1. С. 57–64. DOI:10.31854/1813-324X-2022-8-1-57-64

Аннотация: Использование векторов движения для идентификации видеопоследовательностей хорошо изучено в рамках исследований на тему CBCD (аббр. от англ. Content-Based Copy Detection – определение копий видео на основе анализа контента). Это дает возможность проверки степени сходства двух фрагментов видео или поиска фрагмента в большей видеопоследовательности. Существующие методы формирования идентификационных наборов данных обычно используют полное декодирование видеопотока. Авторы предлагают использовать векторы движения, которые создают видеокодек при сжатии видеопоследовательности. Это позволяет уменьшить вычислительные затраты для идентификации видеопоследовательности и применять более простые алгоритмы для формирования идентификационных данных. В отличие от ранее предложенных методов, использующих либо модифицированные видеокодеки, либо устаревшие, авторы предлагают использовать данные, формируемые при сжатии кодеками наиболее распространенных видеохостингов (Youtube, Vimeo и т. д.) В последующих работах будет изучена возможность формирования автоматизированной системы сравнения видеопоследовательностей и определены ее возможности и ограничения.

Ключевые слова: сравнение видеопоследовательностей, CBCD, определение степени сходства видеоматериалов, поиск видеопоследовательностей.

Введение

Широкое распространение видеoinформации в том или ином виде (видеосвязь, видеохостинги, виртуальные кинотеатры, дистанционное обучение, телемедицина, видео-по-запросу, видеонаблюдение) делает актуальным вопрос об идентификации видеопоследовательности. Существуют различные подходы к решению этой проблемы.

Самый простой – это установка так называемых метаданных, например: имени файла (набора данных), служебных записей в потоке хранения видеоданных со специальными идентификаторами и другие способы записи дополнительных данных в ассоциированный с видеопотоком набор, хэш видеофайла или идентифицирующая запись в файловой системе хранения видеоматериалов. Однако

данные идентификаторы неустойчивы и легко изменяются с помощью самого простого программного обеспечения.

Второй по стойкости способ – внедрение водяных знаков (специальных меток, которые записываются в видеопоток, незначительно ухудшая его качество). Этот способ часто используют при идентификации оригинальных видеопрограмм (фильмов, клипов) для контроля «пиратского» копирования. Однако и этот метод имеет невысокую стойкость к трансформации путем перекодирования или изменения геометрических / визуальных характеристик видеоряда. Кроме того, он вообще не работает в случае так называемой «экранной копии», когда зритель в кинотеатре снимает на камеру только что вышедшую в прокат картину. Дополнительным недостатком является нахождение

компромисса между ухудшением качества изображения и стойкостью метки к различным воздействиям (более стойкие метки, как правило, заметно ухудшают качество изображения). Также этот метод не позволяет идентифицировать те видеопрограммы, которые начали распространяться без водяных знаков.

Наконец, самый стойкий способ заключается в получении идентификационных данных непосредственно из самого видеоряда. Смысл заключается в том, что видео само по себе является идентификатором, и из видеоряда могут быть извлечены идентификационные наборы данных. Под этим понимается получение из исходной видеопоследовательности сравнительно короткого (по сравнению с полной длиной видео) набора данных, который позволяет сравнивать различные видеопоследовательности. Идентификационные данные должны позволить определить степень сходства видеопоследовательностей независимо от их исходного качества (например, видео с Blu-ray диска и оцифрованная VHS-копия) и того, каким преобразованиям они подверглись (изменение яркости/контраста, кадрирование, пересжатие, изменение геометрических характеристик и пропорций, применение различных видеофильтров и т. д.).

Получение набора данных для идентификации является сложной задачей из-за большого объема информации в видеопоследовательности, наличия артефактов сжатия, возможности сильной трансформации исходного видеоматериала, различий в количестве кадров в секунду (видеостандарт), различного монтажа и т. д. Проблематика, связанная с определением копий видеоинформации (например, пиратских видеокопий или определение одинаковых видео разного качества на видеохостинге) на базе идентификационных наборов данных из самого видеоряда, описывается в литературе как CBCD (*аббр. от англ. Content-Based Copy Detection*) [1].

Есть и еще один аспект при использовании данных исследований в нашей стране. Министерство юстиции Российской Федерации формирует список материалов, которые, по решению суда, признаны экстремистскими [2]. В этом списке (кроме текстовых и звуковых) есть и видеоматериалы, которые имеют ссылки на конкретный запрещенный видеоролик с адресом сайта, где он представлен и его именем. Если какое-либо лицо выложит запрещенный видеоролик на другом ресурсе (или на том же ресурсе, но под другим именем), то данный видеоролик формально (до решения суда по его поводу) не будет считаться запрещенным. Кроме того, экстремистский ролик может подвергнуться трансформации или редактированию, наложению текста, совмещению с другими роликами и т. д. Решение данной проблемы может лежать в области

идентификации видеопоследовательностей, которая позволит определять степень схожести автоматизированно (например, при выкладывании видео на видеохостинге или при попытке скачивания видео по каналам связи).

Для формирования набора идентификационных данных используют гистограммы цветности некоторых кадров, временные расстояния между сценами, график изменения яркости по времени, вектора движения, поиск объектов и траектории их движения, а также некоторые другие характеристики видеоряда. Для обработки этих данных, кроме традиционных подходов (например, SURF/SIFT [3, 4] или использование смешанных пространственно-временных идентификаторов [5]), предлагается использовать нейронные сети [6], технологию распознавания лиц [7], дескрипторы активности движения MPEG-7 [8]. Однако для получения этих параметров нужно полностью разжать (декодировать) сжатый видеопоток.

В данной статье рассматривается возможность использования векторов движения сжатого видеопотока в качестве идентификационной характеристики. В случае получения положительного результата это позволит существенно снизить вычислительные затраты на получение векторов движения и формирование идентификационного набора данных (нет необходимости полностью разжимать видеопоток и заново искать в нем вектора движения).

Векторы движения сжатого видеопотока

Одним из основных способов уменьшения избыточности при сжатии видеопоследовательностей является использование векторов движения. Вектор движения представляет собой ссылку блока изображения на один из предыдущих или последующих кадров, т. е. вместо кодирования исходного блока мы указываем, где находится его копия (в другом кадре). Таким образом, векторы движения показывают, как перемещаются блоки между кадрами. Алгоритм сжатия сохраняет список векторов движения, чтобы при восстановлении (декодировании) исходного видеоряда можно было воссоздать полное изображение.

Характеризуя движение между кадрами, вектора движения отражают последовательность перемещения блоков изображения во времени, т. е. отражают динамику всего видеоряда. Например, сцены погонь или панорамирования изображений в фильмах содержат много сонаправленных с движением в кадре векторов, а в сценах диалогов векторов движения мало, и они разнонаправлены.

Особенностью анализа векторов движений является то, что независимо от качества изображения (в известных пределах) и разницы в разрешении видеопоследовательности, векторы движения сохраняют свое доминирующее направление (там, где оно есть), среднюю длину вектора, дистанцию

между сменами направления и прочие характеристики. На этой закономерности основаны методы получения идентификационных наборов данных, описанные в [9].

Наиболее успешный представляется метод [9] получения идентификаторов из векторов движения искусственно уменьшает частоту кадров (пропускает кадры) для получения большей длины вектора и более точного сравнения с использованием 2-х метрик – средней длины вектора и среднего направления вектора в кадре. По сути, это способ сглаживания данных путем уменьшения выборки. Поиск векторов движения при этом выполняется заново, с полным разжатием видеопотока.

Сравним направление векторов движения (см. рисунки 1а, 1б), полученных из 2-х копий одного и того же видеоряда, один из которых является оцифрованной копией записи видео с магнитной пленки в формате VHS (разрешение 480×360, соотношение сторон 4:3), а другой – HD-копией с Blu-ray диска (разрешение 1280×720, соотношение сторон 16:9). Оба видеоряда были выложены на видеохостинге Youtube, сжаты оператором видеосервиса (Google) с помощью одного и того же алгоритма h264, который позволяет получить векторы движения, не выполняя полного декодирования видеопотока (и не выполняя заново поиск векторов движения).

Для удобства VHS-копия увеличена, а HD-копия чуть уменьшена (это видно по размерам стрелок векторов движения). Для просмотра векторов движения использована программа ffplay (из пакета ffmpeg).

Хорошо видно, что, несмотря на существенную разницу в качестве, сохранены доминирующие направления векторов движения и доля сонаправленных векторов относительно их общего количества. Аналогичным образом были проверены несколько различных пар видеопоследовательностей (также использовалась пара VHS-копия и HD-копия), в которых эта закономерность подтвердилась.

Таким же образом можно проверить, как сохраняются эти характеристики векторов движений при кодировании одного и того же видеофрагмента разными видеокодеками (см. рисунки 1с, 1д). При кодировании разными видеокодеками, векторы движения ожидаемо сохраняют свое основное направление и соотношения по количеству, несмотря на большую разницу в поколениях между кодеками. Современный кодек находит большее количество векторов движения и дает меньше ошибок (ложных срабатываний относительно предсказания положения макроблока). Однако и очень старый кодек, несмотря на меньшую эффективность, достаточно точно предсказывает направление движения.



a)



b)



c)



d)

Рис. 1. Сравнение векторов движения: VHS (а), HD (б), mpeg1 (с) и h264 (д)
Fig. 1. Comparison of Motion Vectors of VHS (a), HD (b), mpeg1 (c) and h264 (d)

Получение векторов движения из сжатого видеопотока

Для извлечения векторов движения из сжатого видео можно использовать пакет открытого программного обеспечения для обработки видео ffmpeg, который позволяет извлекать векторы движения для нескольких видеоформатов (mpeg1, mpeg2, mpeg4, mpeg4/avc (h264)). Для каждого полученного вектора доступна следующая информация: положение исходного кадра относительно текущего (прошлого/будущее); ширина/высота блока; позиция (x, y) в исходном и текущем кадре.

Таким образом, для каждого вектора движения можно вычислить такие характеристики, как длина и направление вектора движения, размер блока, длина межкадрового предсказания.

Кодек вычисляет вектора движения почти для всех кадров видеоряда. Исключение составляют редкие опорные (reference) кадры и кадры без движения. Большинство промежуточных кадров содержат от единиц до нескольких сотен векторов движения, для которых можно рассчитать более общую информацию. Например, самые простые метрики: общая длина вектора, средняя длина вектора и среднее направление вектора в кадре.

Гистограмма общей длины вектора в кадре

В качестве примера рассмотрим гистограмму общей длины вектора движения в кадре, т. е. зависимости общей длины вектора движения в каждом кадре (y) от времени кадра (x).

В качестве объекта исследования используем несколько фрагментов фильмов разной длины, размещенных на видеохостинге Youtube, алгоритм сжатия – mpeg4/AVC (h264):

1) малая видеопоследовательность – сцена из фильма «Терминатор» HD-версия (1:39) 24 к/с и VHS-версия (1:33) 29,97 к/с;

2) средняя видеопоследовательность – сцена из фильма «Правонарушитель Морант» HD-версия (3:35) 23,976 к/с и VHS-версия (3:25) 30 к/с;

3) длинная последовательность – мультипликационный фильм «Покахонтас» HD-версия (1:17:51) 25 к/с и VHS-версия (1:21:20) 29,97 к/с;

4) малая последовательность – сцена из фильма «Трансформеры» HD-версия (1:07) 23,976 к/с и «экранная копия» (1:07) 25 к/с.

Выбор обусловлен наличием одинаковых фрагментов в различном качестве, выложенных на Youtube. Сравнение HD- и VHS-версий дает возможность сравнить почти «оригинальное» видео с сильно искаженной копией, которую, впрочем, вполне еще можно смотреть. Различия в длине последовательностей обусловлены наличием разных вариантов видео (кинотеатральная, BD/DVD-версия, версия для VHS, для ТВ и т. д.), сложным процессом оцифровки с VHS-носителя, наличием версий с разным количеством кадров в секунду.

Для визуальной оценки качества сравниваемых фрагментов ниже приведены части стопкадров (см. рисунок 2). Размеры частей изменены (версии с низким разрешением увеличены, версии с высоким разрешением уменьшены). Выбраны опорные (reference) кадры, которые в обеих версиях закодированы с максимально возможным качеством. В левом столбце представлены версии с низким разрешением и качеством (VHS или экранная копия), в правом – версии с высоким качеством и высоким разрешением (HD-версия).

Видно, что версии с низким разрешением имеют искаженную цветопередачу и малую детализацию. Выясним, достаточно ли той информации, которая содержится в векторах движения, для сравнения данных видеопоследовательностей.

Для корректного сравнения пришлось (по возможности) совместить временную линию сюжета, скорректировав значения по оси X с подобранным множителем. Временной сдвиг возникает как в связи с разными версиями видеопрограммы (монтаж), так и в процессе оцифровки (наложение фонограмм с переводом, изменение длительности при процедурах inverse-telecine, deinterlacing и т. д.).

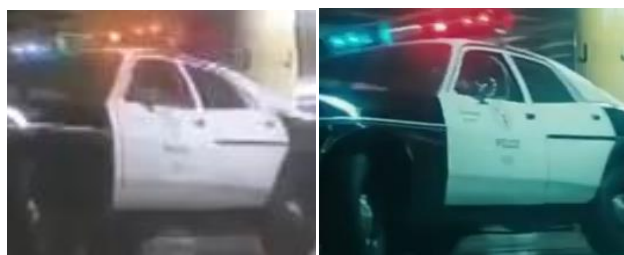
Ввиду разной геометрии изображения (HD-версия содержит в несколько раз большее количество точек, чем VHS), пришлось также подобрать множитель для Y -компоненты графика, чтобы компенсировать различия в длине вектора движения. Способы совмещения временных осей и компенсация средних длин векторов будут предметом рассмотрения в последующих работах.

Рассмотрим гистограмму сцены из фильма «Терминатор» (рисунок 3). По оси X – время, по оси Y – длина вектора в точках кадра. Последовательности разной длины отличаются монтажом (в VHS-версии некоторые части отсутствуют, некоторые части переставлены местами). По гистограмме видно, что определенное сходство есть. Длины векторов HD-версии ожидаемо на порядок больше, чем вектора VHS-версии.

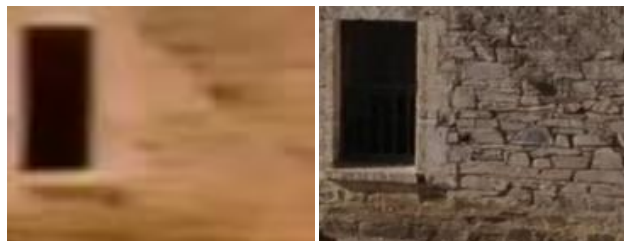
В данном случае при использовании векторов движения каждого кадра, необходимо применить сглаживание данных, чтобы уменьшить возможный шум (ошибки кодека, малое перемещение объектов в кадре). Были рассмотрены различные способы – всевозможные сплайны, кривые Безье, ядерная оценка плотности. Наиболее предпочтительным оказалось использование сглаживания методом ядерной оценки плотности (KDE, аббр. от англ. Kernel Density Estimation) [10].

Сглаживание гистограммы методом KDE

Пусть имеется дискретно определенная последовательность x_i, y_i , где x_i – время кадра; y_i – общая длина вектора движения в нем; $i = 1, 2, \dots, n$, n – количество кадров в последовательности.



a) фрагмент 1



b) фрагмент 2



c) фрагмент 3



d) фрагмент 4

Рис. 2. Сравнение качества фрагментов*Fig. 2. Comparison of Quality of Video Fragments*

Для формирования сглаженных отчетов используется специальная функция:

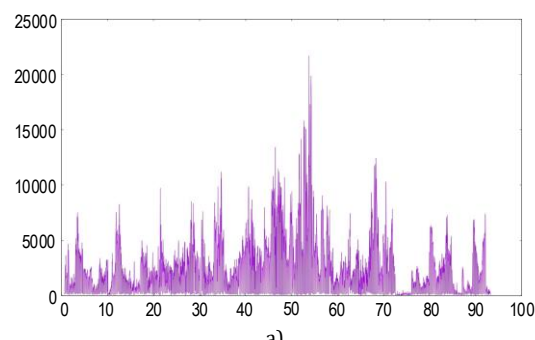
$$f'(x) = \sum_{j=k}^m \frac{1}{h} y_j K\left(\frac{x - x_j}{h}\right),$$

где каждый отчет $f'(x)$ получается суммой отчетов y_i в некотором «окне» значений ($j = k \dots m, 1 \leq k \leq m \leq n$), умноженных на функцию сглаживания.

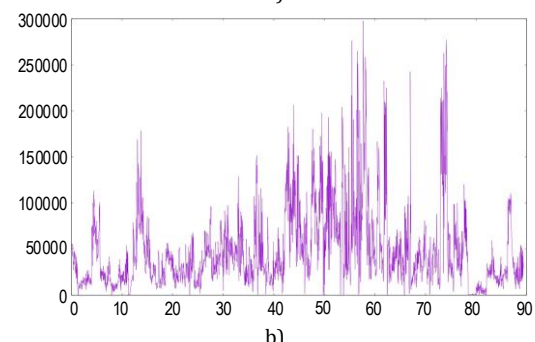
Значения x для функции $f'(x)$ вычисляются в произвольных точках (в т. ч. между точками x_i). «Окно» значений выбирается (по возможности) в некоторой окрестности от центральной точки x и для вычисления значения функции необязательно обходить все кадры последовательности.

На рисунке 4 представлена последовательность x_i, y_i длин векторов нескольких кадров, по оси x – время в секундах, по оси y – общая длина вектора в кадре (в точках кадра). Видно, что для расчета функции $f'(x)$ в точке x используются значения от x_k до x_m (выделено штриховкой), причем точка x лежит примерно посередине между x_k и x_m . Если

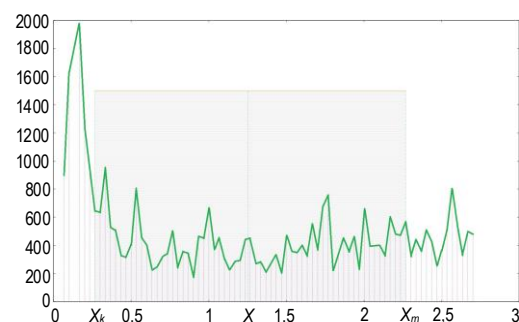
бы точка x лежала левее (ближе к началу оси координат), то для вычисления $f'(x)$ пришлось бы использовать «несимметричное» окно, где $x_k = 0$. Аналогично, если бы x лежала правее (ближе к концу последовательности), то пришлось бы использовать $x_m = x_n$. Значения $f'(x)$ при этом рассчитываются менее точно, чем если бы мы применили более длинную исходную последовательность. Параметр h называется полосой пропускания ядра и оказывает сильное влияние на степень сглаживания данных.



a)



b)

Рис. 3. Гистограмма общих длин векторов VHS- (a) и HD-версий (b) фрагмента 1*Fig. 3. Histogram of Total Vectors Length for VHS (a) and HD (b) Version of Fragment 1***Рис. 4. Выбор интервала для вычисления $f'(x)$** *Fig. 4. Choice of Span for Calculating $f'(x)$*

Для сглаживания используется ядерная функция K , при которой соблюдаются следующие условия:

$$K(x) \geq 0, \int K(x)dx = 1, \int xK(x)dx = 0, \\ \sigma_K^2 \equiv \int x^2 K(x)dx > 0.$$

Наилучшие результаты для сглаживания были получены с использованием функции Гауссова (нормального) ядра [10]:

$$K(x) = \frac{1}{\sqrt{2\pi}} (e^{-\frac{x^2}{2}}).$$

Результат сглаживания (см. рисунок 3) гистограммы с различными значениями h представлен на рисунке 5, где ось X – время в секундах, ось Y – общая длина вектора движения в кадре (в точках кадра), фиолетовая линия – VHS-версия (для кор-

ректности сравнения, длина вектора движения умножена на 12), зеленая линия – HD-версия. Видно, что для сравнения данного короткого видеофрагмента лучше всего подходит сглаживание с $h = 1$ или $h = 2$. При $h = 0,1$ гистограмма больше похожа не несглаженный набор данных, при $h = 5$ при сглаживании было потеряно много деталей.

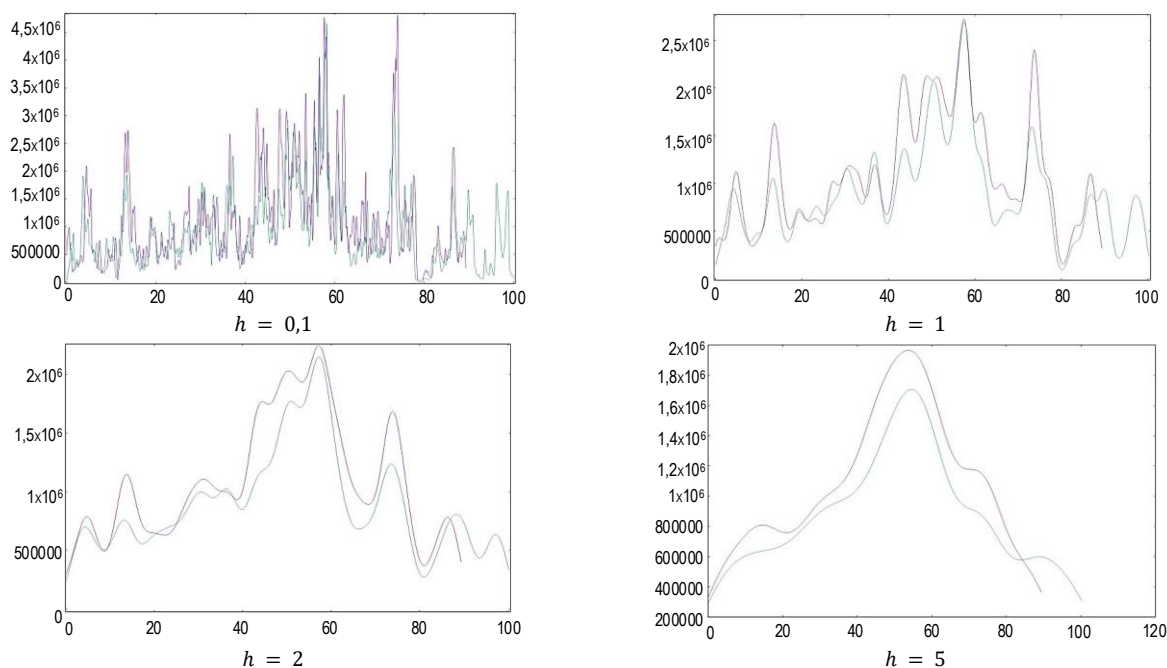


Рис. 5. Результат сглаживания при использовании KDE с различной полосой пропускания ядра для фрагмента 1

Fig. 5. A Smoothing Result by Using KDE with Several Values of Kernel Bandwidth for Fragment 1

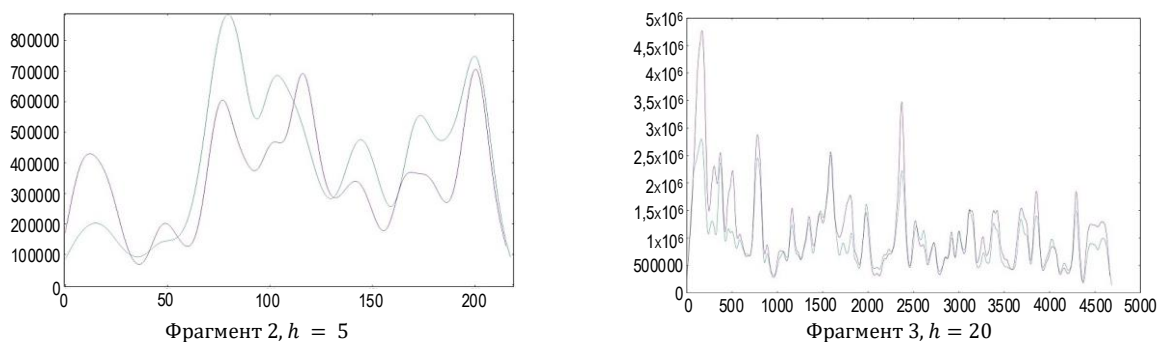


Рис. 6. Сравнение VHS- и HD-версий фрагментов 2 и 3

Fig. 6. Comparison of VHS and HD Version of Fragments 2 and 3

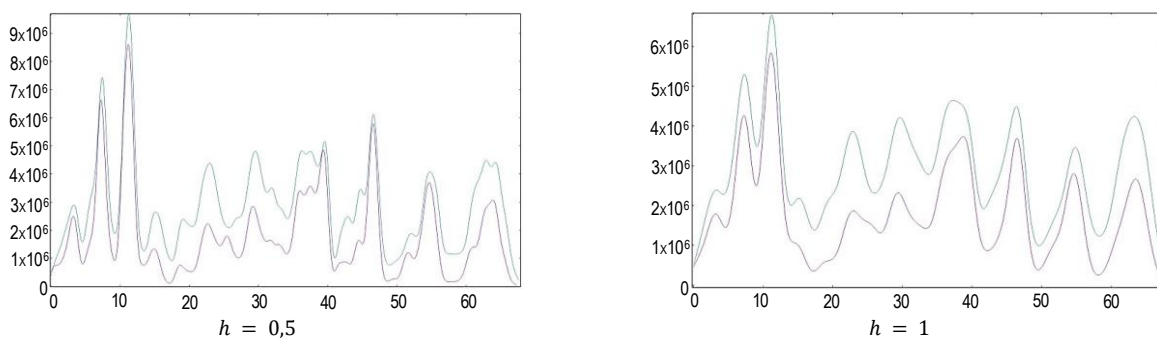


Рис. 7. Сравнение «экранной копии» и HD-версии фрагмента 4

Fig. 7. Comparison of «Screencopy» and HD Version of Fragment 4

При сравнении двух других фрагментов (см. рисунок 6), оказалось необходимым еще увеличить значения h , чтобы получить приемлемый для сравнения результат. В целом можно отметить, что чем больше информации для сравнения (длиннее видео), тем проще найти сходство между последовательностями, если они одинаковы по содержанию, но разные по качеству.

Вектора движения сжатых VHS- и HD-видеопоследовательностей, несмотря на сильное различие в качестве картинки, содержат достаточно информации для сравнения друг с другом. Даже простое суммирование длин векторов в кадре, сглаженное с помощью KDE, дает возможность оценки их сходства. Аналогичным образом можно сравнить так называемую «экранную копию» с оригинальной HD-версией (см. рисунок 7).

Закключение

Использование для идентификации видеороликов готовой информации о векторах движения от видеокодека более эффективно, чем ранее использовавшийся подход, требующий полностью декодировать видео и оперирующий данными большого объема, к тому же содержащими артефакты сжатия.

Направлением дальнейших исследований будет изучение возможности использования других параметров, получаемых с помощью векторов движения сжатого видеопотока: средней длины вектора в кадре и информации о направлении векторов, а также способы вычисления масштабных коэффициентов, позволяющих корректно сравнивать сглаженные данные видеопоследовательностей разного разрешения и качества с целью получения способа автоматизированного сравнения и оценки ее возможностей и ограничений.

Список используемых источников

1. Hampapur A., Bolle R.M. Comparison of Distance Measures for Video Copy Detection: IBM Research Report: RC 22056 (W0105-007), 14 May 2001. URL: <https://dominoweb.draco.res.ibm.com/reports/RC22056.pdf> (Accessed 08th February 2022)
2. Экстремистские материалы. Министерство юстиции Российской Федерации. URL: <https://minjust.gov.ru/ru/extremist-materials> (дата обращения 08.02.2022)
3. Yang X., Zhu Q., Cheng K.T. Near-Duplicate Detection for Images and Videos // Proceedings of the 1-st ACM workshop on Large-scale multimedia retrieval and mining (LS-MMRM'09, Beijing, China, 23rd October 2009). New York: Association for Computing Machinery, 2009. PP. 73–80. DOI:10.1145/1631058.1631073
4. Chiu C.Y., Tsai T.H., Hsieh C.Y. Efficient video segment matching for detecting temporal-based video copies // Neurocomputing. 2013. Vol. 105. PP. 70–80. DOI:10.1016/j.neucom.2012.04.036
5. Thomas R.M., Sumesh M.S. A Simple and Robust Colour Based Video Copy Detection on Summarized Videos // Procedia Computer Science. 2015. Vol. 46. PP. 1668–1675. DOI:10.1016/j.procs.2015.02.106
6. Hu Y., Lu X. Learning spatial-temporal features for video copy detection by the combination of CNN and RNN // Journal of Visual Communication and Image Representation. 2018. Vol. 55. PP. 21–29. DOI:10.1016/j.jvcir.2018.05.013
7. Lu Z.M., Li B., Ji Q.G., Tan Z.F., Zhang Y. Robust video identification approach based on local non-negative matrix factorization // International Journal of Electronics and Communications (AEÜ). 2015. Vol. 69. Iss. 1. PP. 82–89. DOI:10.1016/j.aue.2014.07.021
8. Roopalakshmi R., Guddeti R.M.R. A Novel CBCD Approach Using MPEG-7 Motion Activity Descriptors // Proceedings of International Symposium on Multimedia (Dana Point, USA, 5–7 December 2011). PP. 179–184. IEEE, 2011. DOI:10.1109/ISM.2011.36
9. Taşdemir K., Çetin A.E. Content-based video copy detection based on motion vectors estimated using a lower frame rate // Signal, Image and Video Processing. 2014. Vol. 8. PP. 1049–1057. DOI:10.1007/s11760-014-0627-6
10. Wasserman L. All of Statistics: A Concise Course in Statistical Inference // Springer Texts in Statistics. New York: Springer, 2004. DOI:10.1007/978-0-387-21736-9

* * *

A Study of the Possibility of Usage Motion Vectors of Compressed Videos to Create Video Identification

R. Fahrutdinov¹, A. Mirin¹

¹Saint Petersburg Federal Research Center of the Russian Academy of Sciences, St. Petersburg, 199178, Russian Federation

Article info

DOI:10.31854/1813-324X-2022-8-1-57-64

Received 10th August 2021

Revised 26th October 2021

Accepted 10th January 2022

For citation: Fahrutdinov R., Mirin A. A Study of the Possibility of Usage Motion Vectors of Compressed Videos to Create Video Identification. *Proc. of Telecom. Universities*. 2022;8(1):57–64. (in Russ.) DOI:10.31854/1813-324X-2022-8-1-57-64

Abstract: The use of motion vectors for identifying video sequences has been well studied (in the framework of research on the topic CBCD – Content-Based Copy Detection – detecting copies of videos based on content analysis). This makes it possible to check the similarity of two video fragments or search for a fragment in a larger video sequence. Existing and well-known methods for forming identification datasets typically use complete video stream decoding. The authors suggested using the motion vectors of a compressed video stream, which reduces the computational costs for identifying video sequences and uses simplified algorithms to generate identification data. Unlike the previously proposed methods, which implement either modified video codecs or obsolete ones, the authors propose using data formed by compression codecs that are used in the most common video hosting platforms (Youtube, Vimeo, etc.) The possibility of forming an automated system of comparing video sequences, along with its possibilities and limitations, will be studied in the following works.

Keywords: content based copy detection, video sequence comparison, similarity of video sequences, search of video patterns.

References

1. Hampapur A., Bolle R.M. *Comparison of Distance Measures for Video Copy Detection*. IBM Research Report. Report number: RC 22056 (W0105-007). 14 May 2001. Available from: <https://dominoweb.draco.res.ibm.com/reports/RC22056.pdf> [Accessed 08th February 2022]
2. Ministry of Justice of Russian Federation. *Extremist Materials*. (in Russ.) Available from: <https://minjust.gov.ru/ru/extremist-materials> [Accessed 08th February 2022]
3. Yang X., Zhu Q., Cheng K.T. Near-Duplicate Detection for Images and Videos. *Proceedings of the 1-st ACM workshop on Large-scale multimedia retrieval and mining, LS-MMRM'09, 23rd October 2009, Beijing, China*. New York: Association for Computing Machinery; 2009. p.73–80. DOI:10.1145/1631058.1631073
4. Chiu C.Y., Tsai T.H., Hsieh C.Y. Efficient video segment matching for detecting temporal-based video copies. *Neurocomputing*. 2013;105:70–80. DOI:10.1016/j.neucom.2012.04.036
5. Thomas R.M., Sumesh M.S. A Simple and Robust Colour Based Video Copy Detection on Summarized Videos. *Procedia Computer Science*. 2015;46:1668–1675. DOI:10.1016/j.procs.2015.02.106
6. Hu Y., Lu X. Learning spatial-temporal features for video copy detection by the combination of CNN and RNN. *Journal of Visual Communication and Image Representation*. 2018;55:21–29. DOI:10.1016/j.jvcir.2018.05.013
7. Lu Z.M., Li B., Ji Q.G., Tan Z.F., Zhang Y. Robust video identification approach based on local non-negative matrix factorization. *International Journal of Electronics and Communications (AEÜ)*. 2015;69(1):82–89. DOI:10.1016/j.aeue.2014.07.021
8. Roopalakshmi R., Guddeti R.M.R. A Novel CBCD Approach Using MPEG-7 Motion Activity Descriptors. *Proceedings of International Symposium on Multimedia, USA, 5–7 December 2011, Dana Point*. IEEE; 2011. p.179–184. DOI:10.1109/ISM.2011.36
9. Taşdemir K., Çetin A.E. Content-based video copy detection based on motion vectors estimated using a lower frame rate. *Signal, Image and Video Processing*. 2014;8:1049–1057. DOI:10.1007/s11760-014-0627-6
10. Wasserman L. All of Statistics: A Concise Course in Statistical Inference. *Springer Texts in Statistics*. New York: Springer; 2004. DOI:10.1007/978-0-387-21736-9

Сведения об авторах:

ФАХРУТДИНОВ
Роман Шафкатович

кандидат технических наук, заведующий лабораторией кибербезопасности и постквантовых криптосистем Санкт-Петербургского Федерального исследовательского центра Российской академии наук, fahr@cobra.ru
 <https://orcid.org/0000-0003-0233-3434>

МИРИН
Анатолий Юрьевич

кандидат технических наук, старший научный сотрудник лаборатории кибербезопасности и постквантовых криптосистем Санкт-Петербургского Федерального исследовательского центра Российской академии наук, mirin@cobra.ru
 <https://orcid.org/0000-0002-3114-458X>