

Алгоритм автоматического построения карт перемонтажа разных версий фильма

Молодецких И.А., Ватолин Д.С.

Московский государственный университет им. М.В. Ломоносова

Некоторые фильмы перевыпускаются в двух и более версиях. Одна из частых причин — сокращение фильма для показа в кинотеатрах с последующим выпуском как театральной (сокращённой), так и режиссёрской (оригинальной, расширенной) версий. Автоматическое нахождение различий в монтаже между разными версиями фильма затрудняется возможными изменениями цветовой гаммы, соотношения сторон, сокращением или увеличением длины отдельных сцен, добавлением или удалением объектов в сценах. Подобные трудности не позволяют применить существующие подходы напрямую к построению карт перемонтажа. В данной работе описан алгоритм полностью автоматического построения карты перемонтажа между двумя версиями фильма.

Ключевые слова: карты перемонтажа, сопоставление видео, режиссёрская версия

Automatic Editing Map Construction to Detect Differences Between Film Versions

Molodetskikh I., Vatolin D.

Lomonosov Moscow State University

Some movies are released in two or more versions. One of the frequent causes is shrinking the film for theatrical showcase with subsequent release of both the theatrical (shrunk) cut and the director's (original, extended) cut. Automatic editing difference detection is complicated by potential changes to color gamut, aspect ratio, reduced or increased length of individual scenes, object addition or removal. These difficulties make it impossible to directly apply existing approaches to editing map construction. We propose an algorithm for fully automatic construction of an editing map of two film versions.

Keywords: editing maps, video matching, director's cut, frame-level accuracy

Введение

Часто фильм сокращается для театрального показа, и впоследствии выпускаются как театральная, так и режиссёрская (расширенная) версии. Между разными версиями фильма встречаются такие изменения, как:

- Удаление отдельных менее важных для сюжета сцен;
- Сокращение сцен на несколько кадров с начала или с конца;
- Изменение порядка сцен;
- Добавление новых сцен;
- Цветокоррекция кадров;
- Изменение соотношения сторон;
- Частичная перерисовка существующих сцен, например, удаление или добавление отдельных объектов в сцене, замена заднего плана.

Цель данной работы — создание алгоритма полностью автоматического построения карты перемонтажа между двумя версиями фильма, устойчивого к возможным изменениям. Алгоритм должен корректно распознавать добавленные, удалённые и

переставленные сцены и распознавать незначительно изменённые сцены как совпадающие.

В качестве областей применения можно выделить анализ различий между разными версиями фильма, анализ архивных копий фильмов, контроль показа рекламы на телеканалах.

Исследования в данной области

Сопоставление видео — активно изучаемая область со множеством применений (устранение дубликатов в системах хранения данных; обнаружение материалов, нарушающих авторские права; контроль за показом рекламы в телевизионных сетях) [6].

Во многих методах сопоставления для увеличения точности применяется разбиение видео на сцены. Для этого существует несколько подходов [10]. В работе [4] описан метод, использующий значения яркости каждого пикселя. В статье [1] каждый кадр делится на некоторое число фрагментов, и затем используется средняя яркость пикселей в каждом фрагменте. Также существуют алгоритмы, вычисляющие и применяющие данные о границах объектов [11] или движении объектов [2] в кадре. Эти подходы, как правило, исходят из предположения, что резкое изменение некоторого параметра (например, резкий скачок средней яркости кадра) является признаком смены сцены.

При сопоставлении видео, как правило, используются признаки как отдельных кадров, так и видео в целом. Такой подход повышает устойчивость к искажениям и скорость работы и уменьшает затраты памяти. В качестве признаков кадров могут выступать средние яркости областей кадра [5] или гистограммы цветовых каналов [7]. Авторы [12] предлагают вычислять и использовать перцептивные хеши кадров. Как особенности сцен или видео в целом применимы длины и углы векторов движения объектов в кадрах видео [8], длины сцен и изменения в средней яркости между сценами [13, 14], гистограммы средней яркости [9]. В данных подходах между двумя видеофрагментами строится корреляция по некоторым признакам, и в случае достаточного соответствия видеофрагменты помечаются как совпадающие.

Предложенный алгоритм

Поставим задачу формально. Даны две последовательности кадров $\{a_i\}_{i=1}^{N^a}$ и $\{b_i\}_{i=1}^{N^b}$ длиной N^a и N^b кадров соответственно. Необходимо найти множество индексов I наибольшей мощности и отображение $F: I \rightarrow \mathbb{N}$ такое, что для каждого индекса $i \in I$ кадр a_i первой последовательности соответствует кадру $b_{F(i)}$ второй последовательности, а каждый кадр $a_j, j \notin I$ не соответствует никакому из кадров b_i . Построение такого множества и отображения мы проведём в несколько этапов. За основу взят алгоритм построения и сравнения деревьев сцен из работы [14].

Разбиение на сцены

На первом этапе обе видеопоследовательности разбиваются на сцены. Для этого вычисляется средняя яркость каждого кадра: $\{l_i\}_{i=1}^n$. Затем вычисляются разности между последовательными элементами l_i , образуя новую последовательность $\{d_i\}_{i=1}^{n-1}$:

$$d_i = l_{i+1} - l_i.$$

К полученной последовательности применяется высокочастотный фильтр. Множество индексов кадров, соответствующих границам сцен определяется как $\{i \mid d_{i-1} > \theta\}$, где

$\theta = 1$ была найдена экспериментально. Таким образом, мы получили множества сцен $\{A\}$ и $\{B\}$ последовательностей a_i и b_i .

Сопоставление сцен

На следующем этапе строится сопоставление между сценами из $\{A\}$ и $\{B\}$. Из множества $\{A\}$ выбирается по 7 последовательно идущих сцен (число сцен выбрано экспериментально), и во множестве $\{B\}$ происходит поиск последовательности сцен, схожей по длинам сцен и разностям средних яркостей между сценами с выбранной из множества $\{A\}$. Сравнение последовательностей сцен производится следующим образом: для каждой границы между сценами вычисляются стоимости сопоставления

$$\begin{aligned} CostT(p, q) &= (t_p - t_q)^2, \\ CostL(p, q) &= (l_p - l_q)^2, \end{aligned}$$

где t_p и t_q — время в секундах от начала последовательности сцен до границ p и q соответственно, l_p и l_q — разности средних яркостей между сценами. Стоимость сопоставления последовательностей сцен:

$$Cost = wS_{CostT}^2 + S_{CostL}^2,$$

где S_{CostT} и S_{CostL} — стандартные отклонения значений $CostT$ и $CostL$ по всем границам сцен, $w = 0.2997$ — вес, выбранный экспериментально.

Таким образом строится первое приближение сопоставления между сценами. Затем каждая сцена из $\{A\}$, не сопоставленная на предыдущем шаге (в $\{B\}$ не нашлось последовательности сцен, стоимость сопоставления с которой была меньше порогового значения), находится в $\{B\}$ при помощи гистограмм сцен [1]: каждый кадр S_i сцены S разбивается на M непересекающихся областей, и в каждой области вычисляется средняя яркость. По полученным значениям строится гистограмма кадра $Hist(S_i)$ с M корзинами. Корзины гистограммы сцены $SceneHist(S)$ содержат усреднённые значения корзин гистограмм кадров сцены:

$$\begin{aligned} SceneHist_j(S) &= \frac{1}{|S|} \sum_{i=1}^{|S|} Hist_j(S_i), \\ j &= \overline{1..M}. \end{aligned}$$

Сравнение гистограмм сцен производится с помощью коэффициента Жаккара:

$$J(HistA, HistB) = \frac{\sum_l \min(HistA_l, HistB_l)}{\sum_l \max(HistA_l, HistB_l)}.$$

Покадровое выравнивание

На третьем этапе выполняется покадровое выравнивание сопоставленных сцен. Задачу покадрового выравнивания можно определить следующим образом. Пусть сцена $A = \{a_k, a_{k+1}, \dots, a_{k+|A|}\}$ одной видеопоследовательности соответствует сцене $B = \{b_m, b_{m+1}, \dots, b_{m+|B|}\}$ второй видеопоследовательности. Так как при перемонтаже могут быть обрезаны края сцен, длины A и B могут отличаться, и i -й кадр A не обязательно соответствует i -му кадру B . Задача покадрового выравнивания — найти такой сдвиг s , что для $i \in [\max(0, s), \min(|A|, |B| - \max(0, s))]$ кадр a_{k+i} соответствует кадру b_{m+i+s} .

Оптимальное выравнивание строится с использованием перцептивных хешей кадров на основе яркостей блоков кадра [12]. Хеш H_j кадра S_i вычисляется следующим образом: кадр разбивается на M непересекающихся блоков, в каждом из которых

считается средняя яркость: $\{L_1, L_2, \dots, L_M\}$. Затем находится медиана полученного множества L_d и сравнивается с каждым из значений, образуя итоговый хеш кадра:

$$H_j(S_i) = \begin{cases} 0, & L_j < L_d, \\ 1, & L_j \geq L_d, \end{cases} \\ j = \overline{1..M}.$$

Наилучший сдвиг s — минимизирующий суммарное расстояние между хешами кадров:

$$s = \arg \min_s \sum_{i=\max(0,s)}^{\min(|A|,|B|-\max(0,s))} \rho(H(a_{k+i}), H(b_{m+i+s})), \\ \rho(H(a), H(b)) \text{ — расстояние Хэмминга.}$$

Результаты

Для тестирования алгоритма были составлены две выборки из короткометражного фильма «Синтел» (2010, реж. Колин Леви), каждая из которых состояла из 80 последовательностей. Для этого фильм разбивался на сцены, у некоторых сцен с начала и с конца обрезалось случайное количество кадров, и из результирующей последовательности изменённых сцен бралась случайная перестановка фрагментов длиной от 1 до 20 сцен. Во второй выборке к видео дополнительно применялись случайные искажения, такие как увеличение яркости на 8%, изменение цветового оттенка, обрезание кадра со всех сторон на 5%, размытие кадра.

На вход алгоритма подавалась каждая из тестовых последовательностей и фильм целиком. Затем вывод алгоритма (информация, какому кадру фильма соответствует каждый кадр тестовой последовательности) сравнивалась с исходной, использованной при создании последовательности. Кадры разбивались на корректно сопоставленные (алгоритм показал, что кадр присутствует в исходном фильме, и правильно указал исходный кадр), некорректно сопоставленные (алгоритм показал, что кадр присутствует в исходном фильме, но исходный кадр был указан неправильно), и не сопоставленные (алгоритм показал, что кадр в исходном фильме отсутствует).

Оценку результатов работы алгоритма на первой и второй выборках можно видеть на рис. 1.

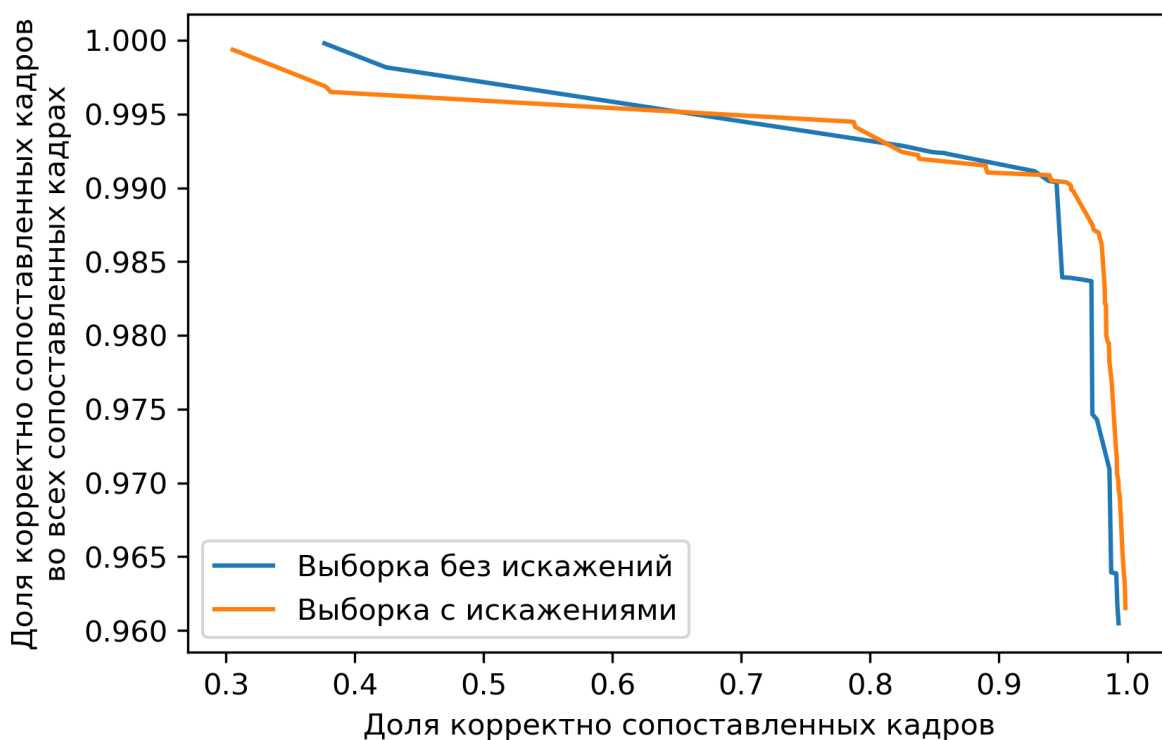


Рис. 1. Результат работы алгоритма на тестовых выборках.

Для визуализации сопоставления, найденного алгоритмом, была создана страница с интерактивной картой перемонтажа (пример на рис. 2). Здесь можно видеть разбиение двух видеопоследовательностей на фрагменты разных типов. Фрагменты тёмно-зелёного цвета — совпадающие между двумя последовательностями. Фрагменты светло-зелёного цвета — присутствующие в одной из видеопоследовательностей, но отсутствующие в другой. Фрагменты светло-синего цвета — присутствующие в обеих последовательностях, но переставленные в одной относительно другой. При наведении курсора на фрагмент отображается информация о фрагменте (временные метки начала и конца), а также подсвечивается соответствующий фрагмент из другой видеопоследовательности. Также карту перемонтажа можно увеличивать, чтобы детально просмотреть конкретное место. Ниже в визуализацию интегрированы сами видеопоследовательности, и при нажатии на фрагмент начинается их проигрывш с места, соответствующего началу данного фрагмента.

Рассмотрим некоторые примеры построенных карт перемонтажа. На рис. 3 можно видеть карту перемонтажа между театральной и режиссёрской версиями фильма «Аватар» (2009, реж. Джеймс Кэмерон). В режиссёрской версии добавлено несколько фрагментов на протяжении фильма, а также переставлено несколько сцен в начале и в середине фильма.

На рис. 4 изображена карта перемонтажа между режиссёрской и расширенной версиями фильма «Хранители» (2009, реж. Зак Снайдер). На данных примерах алгоритм даёт хорошие результаты, благодаря высокому качеству изображения и отсутствию различий в яркости и области кадра между версиями. Можно видеть, что в режиссёрской версии добавлено несколько новых фрагментов.



Рис. 2. Интерактивная визуализация карты перемонтажа.

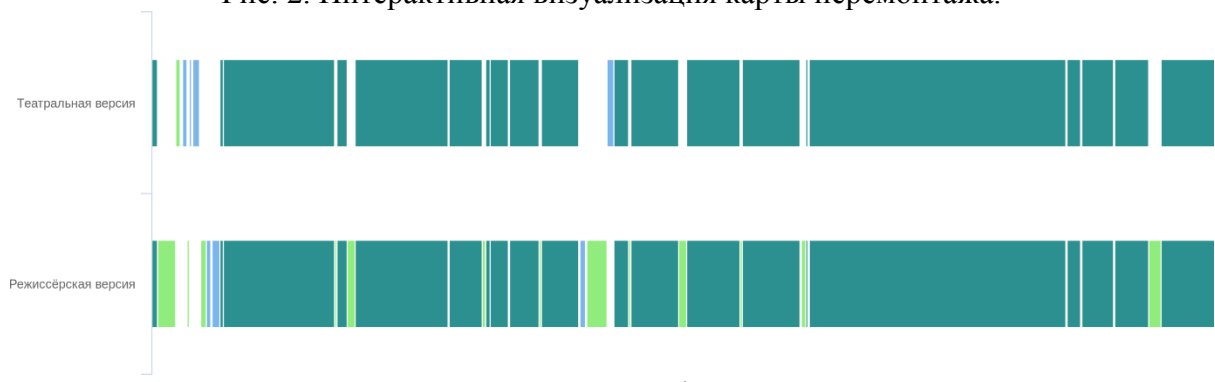


Рис. 3. Карта перемонтажа фильма «Аватар».



Рис. 4. Карта перемонтажа фильма «Хранители».

На рис. 5 приведена карта перемонтажа фильма «Андрей Рублёв» (1966, реж. Андрей Тарковский). Это — сложный случай для алгоритма, так как в одной из версий присутствуют постоянные скачки яркости; между версиями яркости кадров неравномерно изменены (например, одна часть кадра становится ярче, а другая часть кадра — темнее), кроме того качество записи не очень хорошее. Часть сцен сопоставляется неверно (алгоритм считает совпадающими кадры разных сцен), а часть кадров не сопоставляется при фактическом наличии в другой версии фильма. Сопоставление сильно фрагментировано, так как многие сцены в одной из версий обрезаны на несколько кадров. По результатам сопоставления 10 фильмов была составлена статистика переставленных сцен (таблица 1).

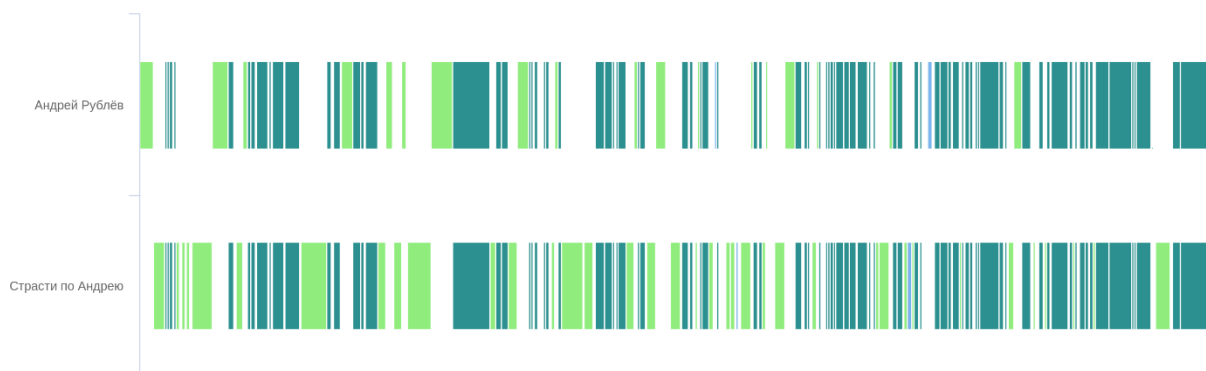


Рис. 5. Карта перемонтажа фильма «Андрей Рублёв»

Таблица 1. Статистика по числу переставленных сцен между театральной и расширенной версиями фильмов.

Фильм	Число переставленных сцен
«Царство небесное» (2005, реж. Ридли Скотт)	35
«Аватар» (2009, реж. Джеймс Кэмерон)	31
«Советник» (2013, реж. Ридли Скотт)	21
«Андрей Рублёв» (1966, реж. Андрей Тарковский)	15
«Человек Паук 2» (2004, реж. Сэм Рэйми)	4
«Пикник у Висячей скалы» (1975, реж. Питер Уир)	2
«Чужой» (1979, реж. Ридли Скотт)	0
«Хранители» (2009, реж. Зак Снайдер) между театральной и режиссёрской	0
«Хранители» (2009, реж. Зак Снайдер) между режиссёрской и расширенной	0
«Вспомнить всё» (2012, реж. Лен Уайзман)	0

Заключение

Предложен алгоритм автоматического сопоставления видеопоследовательностей для построения карт перемонтажа. Проведённое объективное тестирование показывает, что алгоритм корректно определяет переставленные и удалённые фрагменты, а также устойчив к изменениям, встречающимся между разными версиями фильмов. С помощью описанного алгоритма были построены карты перемонтажа нескольких фильмов, и в ходе анализа результатов были выявлены сложные случаи для алгоритма. Планируемое развитие алгоритма включает улучшение работы на трудных местах, внутрикадровое сопоставление для выявления различий в сопоставленных сценах

(таких как удаление объектов или перерисовка фона), разбиение фильма на семантические фрагменты для более глубокого анализа влияния перемонтажа на структуру фильма.

Литература

- [1] Ahmed M., Karmouch A., Abu-Hakima S. Key frame extraction and indexing for multimedia databases //Vision Interface. – 1999. – Т. 99. – С. 1-1.
- [2] Bouthemy P., Gelgon M., Ganansia F. A unified approach to shot change detection and camera motion characterization //IEEE transactions on circuits and systems for video technology. – 1999. – Т. 9. – №. 7. – С. 1030-1044.
- [3] Chiu C. Y., Yang C. C., Chen C. S. Efficient and effective video copy detection based on spatiotemporal analysis //Multimedia, 2007. ISM 2007. Ninth IEEE International Symposium on. – IEEE, 2007. – С. 202-209.
- [4] Chung M. G., Kim H., Song S. M. H. A scene boundary detection method //Image Processing, 2000. Proceedings. 2000 International Conference on. – IEEE, 2000. – Т. 3. – С. 933-936.
- [5] Kim K. R., Jang W. D., Kim C. S. Frame-level matching of near duplicate videos based on ternary frame descriptor and iterative refinement //Image Processing (ICIP), 2015 IEEE International Conference on. – IEEE, 2015. – С. 31-35.
- [6] Liu J. et al. Near-duplicate video retrieval: Current research and future trends //ACM Computing Surveys (CSUR). – 2013. – Т. 45. – №. 4. – С. 44.
- [7] Nian F. et al. Efficient near-duplicate image detection with a local-based binary representation //Multimedia Tools and Applications. – 2016. – Т. 75. – №. 5. – С. 2435-2452.
- [8] Taşdemir K., Cetin A. E. Content-based video copy detection based on motion vectors estimated using a lower frame rate //Signal, Image and Video Processing. – 2014. – Т. 8. – №. 6. – С. 1049-1057.
- [9] Wu X., Hauptmann A. G., Ngo C. W. Practical elimination of near-duplicates from web video search //Proceedings of the 15th ACM international conference on Multimedia. – ACM, 2007. – С. 218-227.
- [10] Yuan J. et al. A formal study of shot boundary detection //IEEE transactions on circuits and systems for video technology. – 2007. – Т. 17. – №. 2. – С. 168-186.
- [11] Zabih R., Miller J., Mai K. Feature-based algorithms for detecting and classifying scene breaks. – Cornell University, 1995.
- [12] Zauner C. Implementation and benchmarking of perceptual image hash functions. – 2010.
- [13] Zobel J., Hoad T. C. Detection of video sequences using compact signatures //ACM Transactions on Information Systems (TOIS). – 2006. – Т. 24. – №. 1. – С. 1-50.
- [14] Паршин А. Е., Глазистов И. В. Алгоритм поиска дубликатов в базе видеопоследовательностей на основе сопоставления иерархии смен сцен //НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ. – 2009. – № 12. – С. 51-61.

Молодецких Иван Андреевич

Студент

Московский Государственный Университет им. М.В. Ломоносова

ivan.molodetskikh@graphics.cs.msu.ru

Ватолин Дмитрий Сергеевич

к.ф.-м.н.

Старший научный сотрудник
Московский Государственный Университет им. М.В. Ломоносова
dmitriy@graphics.cs.msu.ru