

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(национальный исследовательский университет)

Поиск нечетких дубликатов видео

при поддержке интернет-кинотеатра



Никитин Илья Константинович,
аспирант каф. 806 МАИ

twitter: @w_495

почта: w@w-495.ru

nikitin.i@tvzavr.ru

Что такое «нечеткие дубликаты»



оригинал

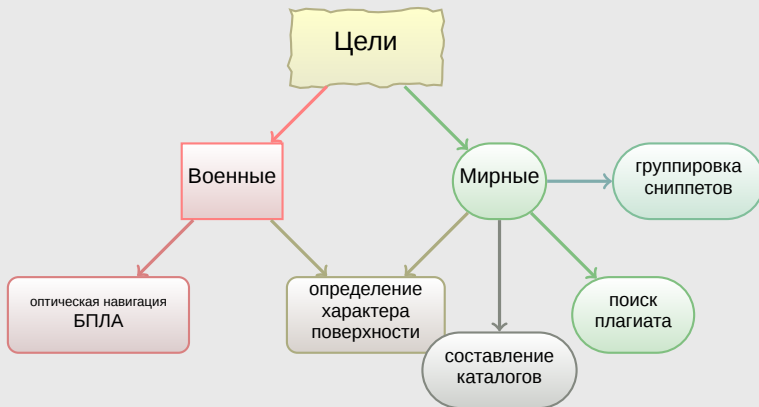


естественный дубликат

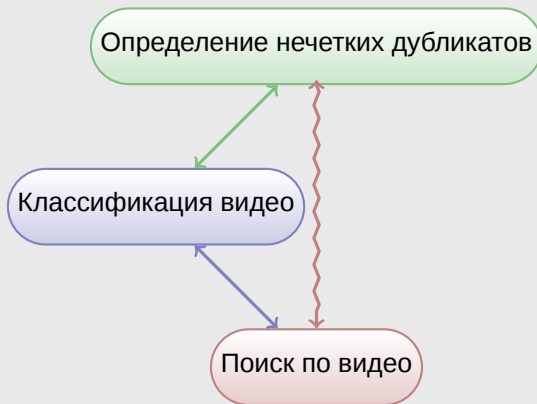


искусственный дубликат

Зачем искать «нечеткие дубликаты»



Связанные проблемы



Как пытаются искать

1. Сравнение глобальных особенностей видео.

- ▷ функция яркости, функция визуального потока;
- + относительно быстро, вычислительно просто;
- легко обмануть.

2. Сравнение отдельных кадров и их сумм:

- ▷ глобальные особенности (гистограммы, спектры, GIST);
- ▷ локальные особенности (PCA-SIFT, детектор Харриса);
- + точно;
- долго, затратно.

3. Сравнение звукового ряда ([youtube.com](https://www.youtube.com)):

- + быстро, просто;
- много ошибок, не применимо если нет звука.

4. Поиск и сравнение «визуальных (видео) слов» (licenzero.ru):

- + точно, если достаточная база «слов»;
- долго, нужно много размеченных данных.

5. Комбинация методов.

Алгоритм поиска нечетких дубликатов видео

- ▶ ν — новое видео;
- ▶ $\Pi = \{\pi_1, \pi_2, \dots, \pi_n\}$ — исходные видео:
 - ← Π может быть пустым;
 - ← для непустого Π вычислены дескрипторы сцен элементов.
- 1. Сравниваем дескриптор каждой сцены $\sigma_{\nu,i}$ из ν с дескриптором каждой сцены $\sigma_{\pi_k,j}$ из π_k в L_2 .
- 2. Если дескрипторы совпали ν с дескрипторами π_k . на некотором временном промежутке, то считаем эту часть ν — дубликатом π_k ,
 несовпавшие части ν помещаем в Π .
- 3. Если дескрипторы не совпали, то считаем ν уникальным и добавляем в Π .

Поиск на основе сцен (1)

Кадр — *frame*, фотографический кадр

- ⇐ отдельная статическая картинка;
- ⇐ обозначим φ .

Съемка — *shot*, кинематографический кадр

- ⇐ множество фотографических кадров, единство процесса съемки;
- ⇐ обозначим $\sigma, \varphi \in \sigma$;
- ⇐ часто называют «сценой», далее будем рассматривать, σ , называя сценой;

Сцена — *scene*, монтажный кадр

- ⇐ множество фотографических кадров, единство места и времени;
- ⇐ обозначим $s, \varphi \in \sigma \subset s$.

Поиск на основе сцен (2)

Сцена как «съемка», кинематографический кадр

— совокупность множества фотографических кадров φ внутри временной области τ , кадры, которой $\varphi_{\sigma,i}$ значительно отличается от кадров соседних областей.

$$\sigma = \{\varphi_{\sigma,i} | \text{diff}(\varphi_{\sigma,i}, \varphi_{\sigma,j}) < \varepsilon, \varphi_{\sigma,i}, \varphi_{\sigma,j} \in \tau\}$$

diff — функция разности кадров.

Аналогично можно ввести определение «звуковой сцены», предварительно разделив звуковой сигнал на отсчеты.

Выделение сцен

- ▶ сравнение гистограмм яркости кадров;
- ▶ сравнение спектров кадров;
- ▶ сравнение векторов движения кадров.

Первые кадры сцен рекламы МТС



Временные отметки перемены сцен (1)

Временные отметки перемены сцен для видео закодированного различными кодеками. Замеры проводились при низкой чувствительности.

Отметки в секундах		
<i>n</i>	vp6f	h264
1	0.094	0.04
2	1.654	1.6
3	6.574	6.52
4	11.654	11.6
5	14.254	14.2

Временные отметки перемены сцен (2)

Временные отметки перемены сцен для видео закодированного различными кодеками. Замеры проводились при высокой чувствительности.

Отметки в секундах			
<i>n</i>	cinepak	indeo5	h264
1	0.133333	0.133333	0.133333
2	11.3333	—	—
3	74	74	74
4	78.9333	—	—
5	87.9333	—	87.9333
6	88.2667	88.2667	88.2667
7	88.3333	—	—
8	94.5333	94.5333	94.5333
9	—	101.133	101.133
10	101.4	—	101.4
11	—	—	112

Относительные длины

- ▶ длина всех отрезков относительно всех, для рекламы МТС (для обоих вариантов) это будет представлять матрицу

$$\begin{pmatrix} 1.0000 & 0.3171 & 0.3071 & 0.6000 \\ 3.1538 & 1.0000 & 0.9685 & 1.8923 \\ 3.2564 & 1.0325 & 1.0000 & 1.9538 \\ 1.6667 & 0.5285 & 0.5118 & 1.0000 \end{pmatrix};$$

- ▶ длина отрезков относительно некоторых:
 - например 3 предыдущих,
 - такой вариант применим для реального времени.

Выравнивания длин

Время	1 сек	2 сек	3 сек	4 сек	5 сек	6 сек	7 сек
Видео V_1	$\sigma_{1,1}$		$\sigma_{1,2}$				$\sigma_{1,3}$
Видео V_2	$\sigma_{2,1}$						$\sigma_{2,2}$

Алгоритм Гейла-Черча для выравнивания длин предложений параллельных корпусов на разных языках

- ▶ требуется установить, что V_1 и V_2 , «переводы» друг друга;
- ▶ когда лучше выравнивать, *до* или *после* перехода к относительным длинам:
 - до*: перевычислять относительные длины,
 - после*: учитывать масштаб относительных длин;
- ▶ вычислительные затраты.

При совпадении относительных длин сцен

- ▶ нельзя делать вывод об одинаковости сцен;
- ▶ сравнить начальные и конечные кадры:
 - сравнивать попиксельно или на основе «знакового представления» с разными масштабами:
 - + просто;
 - не устойчиво к трансформациям.
 - искать особенности в каждой паре кадров, SIFT:
 - + надежно, устойчиво к искажениям;
 - долго, для каждой пары сцен придется перевычислять особенности.
 - вычислить GIST для нужных кадров проверяемой сцены;
 - воспользоваться «мешком слов» и MOB.

При сравнении удобно иметь набор уже сопоставленных сцен.

GIST

Изначально используется для поиска похожих изображений.

1. Считаем отклики детекторов краёв на 5 разных масштабах и 6 ориентациях края.
2. Получаем 33 «канала» — цвет и 30 откликов фильтров края.
3. Разбиваем изображение сеткой 4×4 на 16 ячеек.
4. В каждой ячейке усредняем значения всех каналов.

Мешок слов

Предполагаем, что есть некоторый набор изображений, на которых можно обучиться.

Обучение

- ▷ собираем множество фрагментов (на основе SIFT);
- ▷ кластеризуем и строим словарь;
- ▷ квантуем каждый фрагмент по словарю;
- ▷ считаем «мешки слов» для каждого изображения;
- ▷ обучаем MOV на мешках слов.

Сопоставление

- ▷ выбираем фрагменты из изображения (на основе SIFT);
- ▷ квантуем каждый фрагмент по словарю;
- ▷ строим «мешок слов» для изображения;
- ▷ применяем классификатор MOV.

Дескриптор сцены

1. Вектор отношений длины сцены к длинам других сцен;
 - ▶ удобно сразу хранить, с объединениями соседних сцен;
 - ▶ для относительных длин по трем предыдущим — 6 вариантов.
2. Характеристики начального и конечного кадров:
 - ▶ или «мешки слов» начального и конечного кадров:
 - + лучше соответствует предметной области,
 - потенциально бесконечный размер вектора гистограммы;
 - ▶ или GIST начального и конечного кадров;
 - + не требует какого либо обучения,
 - менее точен.

Алгоритм поиска нечетких дубликатов видео

- ▶ ν — новое видео;
 - ▶ $\Pi = \{\pi_1, \pi_2, \dots, \pi_n\}$ — исходные видео:
 - ← Π может быть пустым;
 - ← для непустого Π вычислены дескрипторы сцен элементов.
 - 1. Сравниваем дескриптор каждой сцены $\sigma_{\nu, i}$ из ν с дескриптором каждой сцены $\sigma_{\pi_k, j}$ из π_k в L_2 .
 - 2. Если дескрипторы совпали ν с дескрипторами π_k . на некотором временном промежутке, то считаем эту часть ν — дубликатом π_k ,
 несовпавшие части ν помещаем в Π .
 - 3. Если дескрипторы не совпали, то считаем ν уникальным и добавляем в Π .
- (—) Сравнивать дескрипторы явно — не эффективно,
 можно применить семантическое хеширование.

Семантическое хеширование

- ▶ Введем бинарные подписи.
- ▶ Подписи для близких в L_2 сцен должны быть близки.
- ▶ Локально чувствительное хеширование:
 1. Случайная проекция данных на прямую.
 2. Случайно выберем порог, пометив проекции 0 или 1.
 3. С увеличением числа бит подпись приближает L_2 -метрику в исходных дескрипторах.
- ▶ Обучаемое хеширование.

Обучаемое хеширование

Стимулирование (boosting)

- ⇐ BoostSSC;
- ⇐ Расстояние между дескрипторами вычисляется, как расстояние Хемминга.

Ограниченная машина Больцмана

- ⇐ связь только между слоями;
- ⇐ внутри слоев связи нет;
- ⇐ мощность слоев понижается от размера входного вектора до размера требуемого кода.

Результаты и перспективы

Результаты:

- ▶ предложен подход:
 - ▷ относительные длины,
 - ▷ выравнивания,
 - ▷ дескрипторы сцен;
- ▶ проведены эксперименты.

Дальнейшее развитие

- ▶ изменить алгоритм поиска перемены сцены,
 - ▷ сейчас используется ffmpeg;
- ▶ хеширование на основе машины Больцмана;
- ▶ эксперименты на реальном множестве исходных данных.