

УДК 004.021

## АНАЛИЗ КАДРОВ ВИДЕОРЯДА И ВЫЧИСЛЕНИЕ ПРОДОЛЖИТЕЛЬНОСТИ СЦЕНЫ ИСПОЛЬЗУЯ АЛГОРИТМ ПЕРЦЕПТИВНОГО ХЭША

Стадник А. С.

Донецкий национальный технический университет  
enemis@mail.ru

*В статье рассматривается методика создание подсистемы детектирования сцен, в видеоматериалах. Описывается алгоритм детектирования с использованием перцептивного хэша.*

### Введение

В настоящее время популярность социальных сетей обусловлена тем, что данные сети предоставляют пользователям большое количество разнообразных сервисов и давно перестали быть только средствами общения групп и сообществ, а стали виртуальными мирами, в которых можно общаться с людьми со схожими интересами, создавать группы или клубы по интересам, играть в on-line игры смотреть фильмы, видео клипы слушать музыку. Мультимедийные материалы в данных сетях загружают пользователи для того чтобы с любимой музыкой, любимым фильмом, клипом мог ознакомиться любой другой пользователь сети. В результате загрузке материалов на сайт данный контент копируется и храниться на серверах социальной сети. Из всего загруженного контента большую часть (примерно 85%), дискового пространства серверов занимают загруженные видеоматериалы. В большинстве случаев пользователь не является автором видеоматериала, и сам его получает из третьих источников. В результате некоторые материалы загружают разные незнающие друг друга люди. Происходит дублирование материалов, часто бывает, что источники получения исходного материала, у различных пользователей различны. К примеру один и тот же клип сжатый с одним разрешением с одними характеристиками битрейта звука и видео, но с разными версиями одного и того же кодека после выгрузки и обработки на сервере будут считаться абсолютно разными, для системы поиска но абсолютно идентичными для человека. Я рассказывал об этом в своей предыдущей статье [1], также были описаны существующие способы устранения проблемы дублирования и минусы этих способов. Так же я предложил вариант алгоритма поиска материалов основанный на системе распознавания образов. В данном методе поиска главной задачей была реализация детектирования сцены. О реализации этой задачи на практике, будет данная статья.

### Общая постановка проблемы

Нахождения сцены проще всего производить на основании сравнения каждого следующего кадра с предыдущим.

В таком случае постановка проблемы будет выглядеть следующим образом: в видеоматериале последовательно сравниваются текущий и предыдущий кадры, полученные и преобразованные определенным образом. Необходимо на основании сравнения определить насколько отличаются два кадра, и принять решения относится ли текущий кадр к новой сцене или к текущей, и в зависимости от решения выполнить определенные действия.

### Перцептивный хэш, что это?

В задаче выявления дублируемых материалов, вычисления длительности сцен и сохранения данной информации в базе данных, позволит сократить время выполнения программы сравнения, так как отпадает необходимость сравнения двух материалов равных по длительности, но с неравными отрезками длин сцен, классический пример различные серии одного сериала.

Вычисление длительности сцены, главная задача для выявления схожих видеоматериалов,

но так же необходимо учитывать, что видеоматериалов очень много, и необходимо, чтобы данный алгоритм был максимально быстрым, и использовался только один раз для каждого из материалов, а результаты его работы хранились в базе.

Вследствие всех перечисленных требований было решено использовать перцептивный хеш для сравнения изображений.

Перцептивные хэш-алгоритмы описывают функции для генерации сравнимых хэшей. Те, в зависимости от требования задачи можно определять степень схожести объектов на изображении, вычислять положение, определять расстояние между объектами.

Изображения используются для генерации индивидуального (но не уникального) отпечатка, и эти отпечатки можно сравнивать друг с другом.

Перцептивные хэши – это другая концепция по сравнению с обычным хешированием вроде MD5 и SHA2. В обычном хешировании каждый хэш, это преобразованный входной массив данных произвольной длины в выходную битовую строку фиксированной длины. Такие преобразования является случайным. Данные, которые используются для генерации хэша, выполняют роль источника для генератора случайных чисел, так что одинаковые данные дадут одинаковый результат, а разные данные — разный результат. В общем случае однозначного соответствия между исходными данными и хеш-кодом нет в силу того, что количество значений хеш-функций меньше, чем вариантов входного массива; существует множество массивов, дающих одинаковые хэш-коды, это вызывает так называемые коллизии. В отличие от них, перцептивные хэши можно сравнивать между собой и делать вывод о степени различия двух наборов данных.

### Алгоритм детектирование сцен, используя перцептивный хеш

Как известно всем математикам, имеющем представления о преобразованиях Фурье [2, 3], что изображение это всего лишь двумерный (зависимость яркости от горизонтальной и вертикальной координаты) непериодический сигнал. Для RGB изображения таких сигналов будет сразу три: яркость в каналах Red, Green и Blue [4]. Так как в обработке сигналов и связанных областях преобразование Фурье обычно рассматривается как декомпозиция сигнала на частоты и амплитуды разделим условно изображение на 3 частоты:

1. На низкой частоте будут содержаться самые крупные детали, общее распределение яркости и цвета, то есть форма объекта.
2. На средней частоте находится средняя и мелкая детализация, которая скрывается за выражением «локальный контраст» и для снятых крупным планом объектов является фактурой поверхности.
3. На высокой частоте находится сверх-мелкая детализация, про которую часто говорят «микроконтраст» и которая отвечает за резкость.

Очевидно, что для сравнения изображений необходимо использовать низкие частоты. Запишем алгоритм сравнения на псевдокоде и проиллюстрируем его работу на примере 3 подряд полученных кадров, для наглядности работы алгоритма будем сразу выполнять действия для 3 кадров:

```

АЛГ CompareImages
НАЧ
Загрузка _ видеофайла(имя _ файла)
предыдущий _ хеш[64]
номер _ кадра=0
ПОКА (кадр=Получение _ следующего _ кадра()) (1)
    НЦ
        номер _ кадра++
        кадр=Уменьшение _ размера (кадр, размер=8x8) (2)
        кадр=перевод _ в _ градации _ серого(кадр) (2)
        ср _ яркость=расчет _ средней _ яркости _ каждого _
            _ пикселя(кадр) (3)
        НЦ ДЛЯ i ОТ 1 ДО кол _ ва пикселей (3)

```

```

ЕСЛИ яркость _ кадр[i] < ср_ яркость ТО
  хеш[i]=0
ИНАЧЕ
  хеш[i]=1
КЦ
ЕСЛИ номер_ кадра!=1
  расстояние=Расстояние_ Хеминга(предыдущий_ хеш,хеш) (4)
ЕСЛИ расстояние > 10 ТО
  ВЫВОД «Изображения относятся к разным сценам» (5)
  предыдущий_ хеш=хеш
ИНАЧЕ
  ВЫВОД «Изображения относятся к одной сцене» (5)
  предыдущий_ хеш=хеш
ИНАЧЕ
  предыдущий_ хеш=хеш
КЦ
КОН АЛГ CompareImages

```

1. Опустив цикл, получим сразу 3 подряд идущих кадра (Рис. 1). Очевидно, что для человека определить, что первые 2 кадра относятся к одной сцене, а третий к другой не составит особого труда. Для принятия решения компьютером проследим за состояниями изображения при работе алгоритма сравнения изображений.



Рисунок 1. Последовательные кадры видеоматериала

2. Как уже упоминалась, для сравнения необходимы низкие частоты. Самый быстрый способ избавиться от высоких частот — уменьшить изображение. В данном случае мы уменьшаем его до 8x8, так что общее число пикселей составляет 64 (Рис. 2).



Рисунок 2. Изображения, приведённые к размеру 8x8 пикселей

Маленькое изображение переводится в градации серого, при этом хэш уменьшается втрое: с 64 пикселей (64 значения красного, 64 зелёного и 64 синего) всего до 64 значений цвета. Информация о кадре теперь равна всего 64 байта.

3. Далее для каждого из кадров вычисляется среднее значение пикселей, а затем формируется хэш, каждый пиксель сравнивается со средним значением и если он больше среднего значения, то в ячейку хэша записывается 1 иначе 0. В результате получаются примерно следующие изображения (Рис. 3)

4. Далее записывается переводится 64 отдельных бита в одно 64-битное значение. Порядок не имеет значения, если он сохраняется постоянным.

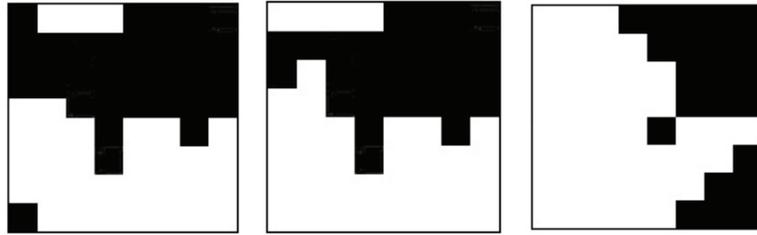


Рисунок 3. Визуальное представление вычисленного перцептивного хэша

- Изображение 1 = 8FFFFFF3F09081000
- Изображение 2 = 3FFECF3F09080000
- Изображение 3 = 3E1E0E0E08010307

5. Затем вычисляется расстояние Хемминга для каждой пары кадров.

В результате получается:

- для первого и второго изображения расстояние равно 5;
- для второго, третьего изображения расстояние равно 19;

6. На основании вычисленного расстояния делается заключение какой и сцен относится следующей кадр, если меньше 10 то к текущей, иначе к новой.

### Выводы

Высокая степень дублирования видеоматериалов, располагающихся на видеосервисах, приводит к их излишней загруженности. Т.о., задача определения и удаления схожих видеоматериалов является актуальной.

Сделан вывод что, при разработке приложения поиска, основной задачей, является, реализация подсистемы обнаружения сцен. Сделано предположение, что наиболее быстро и эффективно справиться с этой задачей алгоритм построенный на базе перцептивного хэша.

Предложенный алгоритм решения задачи позволит, значительно сократить общее время выполнения программы распознавания, тем самым позволив уменьшить время потраченное процессорами компаний социальных сетей и видеосервисов на поиск и удаления дублируемого контента, что в свою очередь приведет к повышению эффективности их работы.

### Литература

- [1] Сборник материалов II всеукраинской научно-технической конференции студентов, аспирантов и молодых ученых «Комп'ютерний моніторинг та інформаційні технології (ИУС и КМ 11-13 апрель - 2011)». III том. Донецк, ДонНТУ - 2011. с. 150-155.
- [2] EqWorld. Мир математических уравнений. Режим доступа: <http://eqworld.ipmnet.ru/ru/auxiliary/aux-inttrans.htm> [электронный ресурс].
- [3] Википедия - свободная энциклопедия. Режим доступа: [http://ru.wikipedia.org/wiki/Преобразования\\_Фурье](http://ru.wikipedia.org/wiki/Преобразования_Фурье)
- [4] 36 Кадров. Обработка изображения на основе частотного разложения изображения. Режим доступа: <http://36k.com.ua/photoarticles/processing/1> [электронный ресурс].
- [5] OpenCV - Сравнение изображений и генерация картинки отличий. Режим доступа: <http://robocraft.ru/blog/computervision/506.html> [электронный ресурс].
- [6] Поиск по изображениям — гугл и не только. Режим доступа: [http://habrahabr.ru/blogs/image\\_processing/126136](http://habrahabr.ru/blogs/image_processing/126136) [электронный ресурс].
- [7] «Выглядит похоже». Как работает перцептивный хэш. Режим доступа: [http://habrahabr.ru/blogs/image\\_processing/120562/](http://habrahabr.ru/blogs/image_processing/120562/) [электронный ресурс].