

УДК 004.627

А.С. Погодина, Т.А. Колесникова, О.Н. Кадушкевич

*Харьковский национальный университет радиоэлектроники, Харьков*

## ПРОЦЕДУРА СЖАТИЯ ВИДЕОИНФОРМАЦИИ НА ОСНОВЕ КОДИРОВАНИЯ ЦВЕТОВЫХ СОСТАВЛЯЮЩИХ ВИДЕОСИГНАЛА

*При решении задачи сжатия естественным является вопрос, насколько эффективна та или иная система сжатия. Процесс сжатия приводит к уменьшению объема представления информации на информационном носителе. В результате сжатия видеоинформации наряду с объективными факторами ошибок возникают искажения. Исходя из этого, существуют два типа сжатия: сжатие с искажениями представляемой информации, и сжатие, не допускающее такие искажения. Целью данного исследования является предложить оптимальную процедуру сжатия, основой которой является кодирования цветовых составляющих, а также первоначальная цветовая коррекция видеофрагмента при помощи разработанной программы на языке программирования Java. Таким образом, цель сжатия данных – обеспечить компактное и целостное представление данных для их экономного хранения или передачи по сетям.*

**Ключевые слова:** сжатие, видеосигнал, цветовое пространство, цветовая коррекция

### Введение

Развитие современной аудио-, видео- и компьютерной техники за последние годы привели к тому, что общение перешло в режим реального времени. Современная техника позволяет создавать и работать с медиафайлами, обмениваться ими со своими друзьями, размещать в интернет пространстве или хранить на устройствах памяти.

Изучение различных алгоритмов и процедур сжатия медиаинформации является областью научных исследований, которая остается популярной уже больше четверти века.

### Постановка цели исследования

Скорости передачи данных через сети и Интернет все время возрастают, объемы флеш-накопителей и жестких дисков все время увеличиваются, но вопрос о хранении информации остается острым. Не смотря на все это передавать информацию в исходном виде не разумно, это занимает много времени и уменьшает место хранения, поэтому актуален вопрос сжатие видеоданных.

Сжатие видеоданных имеет несколько достоинств. Во-первых, оно дает возможность использовать цифровое видео в среде передачи и хранения видеоконтента, которая не поддерживает несжатое видео. Например, пропускная способность современного Интернета недостаточна для обращения с несжатым видео в реальном масштабе времени даже при низкой частоте кадра и малом его размере [1].

Во-вторых, сжатие видеоданных делает более эффективным использование ресурсов при передаче и хранении видеоданных. Если доступен высокоскоростной канал, то более привлекательным пред-

ставляется решение, позволяющее передавать сжатое видео высокого разрешения вместо несжатого видео низкого разрешения.

Существуют два типа систем сжатия данных:

- системы сжатия без потерь информации (неразрушающее сжатие);
- системы сжатия с потерями информации (разрушающее сжатие).

Выбор системы неразрушающего или разрушающего сжатия зависит от типа данных, подлежащих сжатию. При сжатии речи, музыкальных данных, изображений и видео, наоборот, чаще используется разрушающее сжатие, поскольку при практически незаметных искажениях оно обеспечивает на порядок, а иногда и на два меньшую скорость.

Технологии сжатия видеоизображения – это сокращение и удаление избыточных видеоданных с целью оптимизации хранения файлов цифрового видео и их передачи по сети. Эффективные технологии сжатия позволяют значительно уменьшить размер файла при полном или частичном отсутствии потерь качества. Однако качество видеоизображения может снизиться при последующем уменьшении размера файла путем повышения уровня сжатия для конкретной технологии.

Цветовое пространство – графическое представление размерностей цвета. Цветовые пространства описываются набором цветовых координат и правилами построения цветов. В цветовом пространстве, каждый цвет описан набором из соответствующих координат

Известно, что цветовое изображение требует не менее трех чисел на один пиксел для точной передачи его цвета. Метод, выбранный для представления яркости и цвета, называется цветовым простран-

вом. При использовании некоторого цветового пространства каждый пиксель представляет собой запись (структуру), полями которой являются компоненты цвета.

Есть три наиболее популярные цветовые модели – это RGB (использующееся в компьютерной графике); YIQ, YUV или YCbCr (использующейся в видеосистемах); и CMYK (использующейся в цветовой печати). Все цветовые пространства могут быть получены из RGB пространства извлекаемое камерами и сканерами [2].

Целью исследования являлось предложить такую процедуру сжатия видеосигнала, в основе которой лежит кодирование цветовых составляющих, за счет чего уменьшается конечный объем файла. Упор исследования делался на цветовую составляющую видео, а не изменение его динамических составляющих.

### Выбор цветового пространства

Известно, что органы зрения человека менее чувствительны к цвету предметов, чем к их яркости (светимости).

В цветовом пространстве RGB все три цвета считаются одинаково важными, и они обычно сохраняются с одинаковым разрешением. Однако можно отобразить цветное изображение более эффективно, отделив светимость от цветовой информации и представив ее с большим разрешением, чем цвет.

Цветовое пространство YCbCr и его вариации (иногда их обозначают YUV) является популярным методом эффективного представления цветных изображений в видеоинформации.

Элементы цветового пространства: Y - яркость или интенсивность (размер 8 бит; значения от 16 до 235), C<sub>b</sub> - "цветность синего" или более точно отклонение цвета от серого на оси blue-yellow, C<sub>r</sub> - "цветность красного" или более точно отклонение цвета от серого на оси red-cyan.

Яркостная компонента содержит "черно-белое" (в оттенках серого) изображение, а оставшиеся две компоненты содержат информацию для восстановления требуемого цвета.

Полное описание цветного изображения задается величиной Y (компонентой светимости) и тремя хроматическими разностями C<sub>b</sub>, C<sub>r</sub> и C<sub>g</sub>, которые выражают разность между интенсивностью соответствующего цвета и средней светимостью каждого пиксела изображения.

Возникает вопрос о значении нового представления цвета, поскольку получается четыре компоненты вместо трех в пространстве RGB. Однако число C<sub>b</sub> + C<sub>r</sub> + C<sub>g</sub> является константой, поэтому только две из трех хроматических компонент необходимо хранить или предавать, поскольку

третью компоненту всегда легко вычислить, зная две другие.

Преимущество пространства YCbCr по сравнению с RGB заключается в том, что компоненты C<sub>b</sub> и C<sub>r</sub> можно представлять с меньшим разрешением, чем Y, так как глаз человека менее чувствителен к цвету предметов, это позволяет сократить объем информации, требуемый для представления хроматических компонент, без заметного ухудшения качества передачи цветовых оттенков изображения.

Простой зритель не заметит никаких различий между изображением RGB и его представлением в цветовом пространстве YCbCr с округленным разрешением хроматических компонент. Использование большего разрешения для компоненты яркости по сравнению с хроматическими компонентами является простым, но весьма эффективным приемом при сжатии видеоизображений.

Самый очевидный формат, это так называемый формат 4:4:4, который означает полную точность в передаче хроматических компонент, т.е. на каждые 4 световые отсчеты Y передаются по 4 отсчета компонент C<sub>b</sub> и C<sub>r</sub>. Другой формат 4:2:2 (YUY2) предполагает, что на каждые 4 отсчета компоненты Y приходится по два отсчета хроматических компонент. Данный формат используется для высококачественного цветного видео и используется в стандартах MPEG-4 и H.264.

Формулы для прямого и обратного преобразования выглядят следующим образом:

$$Y = k_r R + (1 - k_b - k_r) \times G + k_b B;$$

$$C_b = \frac{0.5}{1 - k_b} (B - Y);$$

$$C_r = \frac{0.5}{1 - k_r} (R - Y).$$

Для обратного преобразования формула выглядит:

$$R = Y + \frac{1 - k_r}{0.5} \times C_r;$$

$$G = Y - \frac{2k_b(1 - k_b)}{1 - k_b - k_r} \times C_b - \frac{2k_r(1 - k_r)}{1 - k_b - k_r} \times C_r;$$

$$B = Y + \frac{1 - k_b}{0.5} \times C_b/$$

Рекомендация ITU-T с идентификатором BT.601 предлагает следующие коэффициенты: k<sub>b</sub>=0.114, k<sub>r</sub>=0,299. С этими коэффициентами получаем следующие формулы:

$$Y_{601} = 0.299 \times R' + 0.587 \times G' + 0.114 \times B';$$

$$C_b = -0.172 \times R' - 0.339 \times G' + 0.511 \times B' + 128;$$

$$C_r = 0.511 \times R' - 0.428 \times G' + 0.083 \times B' + 128.$$

Данные формулы используются для кодирования восьмимбитного сигнала RGB с диапазоном возможных значений от 16 до 235, то есть 16 соответ-

ствуется полностью белому, а 236 – полностью черному цвету. Это сделано с целью улучшения передачи по сетевым каналам [3].

Кодирование по соответствующим формулам уменьшает объем информации, но не в такой степени как предполагалось изначально. Поэтому было предложено проводить первоначальную цветовую коррекцию видеосигнала, а только затем дополнительно кодировать цветовые составляющие.

### Цветовая коррекция видеосигнала

Суть такой коррекции заключается в том, чтобы проводить покомпонентную коррекцию отдельных составляющих цветового пространства RGB. Одним из этапов коррекции является задание желаемых цветов на выбранном фрагменте видео при помощи разработанного программного средства.

На основе полученного скорректированного фрагмента видео, проводится последующая цветовая коррекция для видео в целом.

Очевидным недостатком предложенной цветовой коррекции является то, что она применима для определенной категории видеoinформации. Это видео – содержащее те цвета, которые нам априори известны. А также то, что проводить такую коррекцию вручную довольно сложно.

Особенностью данной цветовой коррекции, является то, что пользователь выбирает некоторую небольшую совокупность малых фрагментов, цвет которых ему известен (например, участки лица, листья цветов, цвет неба, участки знакомой одежды и др.). Далее, используя эти тестовые фрагменты и соответствующие им фрагменты на исходном кадре, решается задача идентификации модели цветовой коррекции.

Модель цветовой коррекции изображений строится в цветовом пространстве RGB в виде соотношений, реализующих покомпонентные преобразования вида:

$$\begin{cases} R^* = f_r(R) = a_r R^3 + b_r R^2 + c_r R + d \\ G^* = f_g(G) = a_g G^3 + b_g G^2 + c_g G + d \\ B^* = f_b(B) = a_b B^3 + b_b B^2 + c_b B + d \end{cases}$$

где  $R, G, B$  – координаты цвета фрагментов на исходном кадре,

$R^*, G^*, B^*$  – «желаемые» координаты цвета на тех же фрагментах,

$a_r, a_g, \dots, d_g, d_b$ , – пока неизвестные коэффициенты, которые должны быть определены в результате решения задачи идентификации [4].

Программа разработана на языке программирования Java. При открытии программы создается файл, в который сохраняются рассчитанные коэффициенты. В верхней части открывшегося окна находятся пиктограммы команд «Выбрать изображение...» и «Сохранить изображение...». Выбранное изображение для коррекции появляется в середине окна программы. Значение координат  $R, G, B$  выбранной точки на изображении отображается, выше поля ввода новых значений координат. После ввода желаемых координат следует нажать пиктограмму «Корректировать». В командной строке программа выводит рассчитанные координаты.

**Пример:** выбираем видефрагмент и извлекаем из него кадр, цвета на котором нам известны (рис. 1). В данном случае: первая точка – это красный цвет с координатами:  $R=250, G=21, B=28$ ; вторая точка – это зеленый цвет с координатами:  $R=100, G=112, B=0$ ; третья точка – это белый цвет с координатами:  $R=244, G=230, B=201$ .



Рис. 1. Задание желаемых цветов

Зная исходные координаты цвета, задаем желаемые цвета, в данном случае пусть они будут следующие.

Первая точка - это красный цвет с координатами: R=255, G=0, B=0.

Вторая точка – это зеленый цвет с координатами: R=99, G=119, B=0.

Третья точка – это белый цвет с координатами: R=255, G=255, B=255.

Задаем желаемые цвета в разработанной программе.

После сохранения, получаем откорректированный файл уже с переведенным цветовым пространством, объем которого уменьшился примерно в два раза.

### Вывод

Сжатие видеoinформации необходимая часть корректного хранения данных. Но сжатие информации возможно не только за счет сжатия внутрикадрового и межкадрового расстояния, но и за счет кодирования цветовых составляющих видеосигнала. Цветовая коррекция является составной частью в процедуре сжатия, которая позволяет не только улучшать визуальное восприятие информации, но и реализовывать по каналное кодирование цветовых составляющих, что приводит к уменьшению информационного объема файла.

Данное исследование выявило недостатки, как первоначальный отбор обрабатываемого видео;

трудозатраты в проведении коррекции, поскольку в программе существуют ограничения.

Для достижения лучших результатов необходимо комплексное сжатие видеоданных, учитывая, как и динамические, так и цветовые составляющие.

### Список литературы

1. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео [Текст] : учеб. / В. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин. – М.: ДИАЛОГ-МИФИ, 2003. -384 с.

2. Поташиников А.М. Методы адаптивного кодирования цветовых составляющих видео изображений с применением преобразования Каранена-Лозва. [Текст] / А.М. Поташиников // Журн. МИРЭА. – 2012, №6 – С. 127-130.

3. Со И.А. Измерение цветовых искажений видеоизображений с использованием характеристик зрительной системы. [Текст] / И.А. Со // Журн. Компьютерная оптика. – 2011. – Т. 35, №3. –С. 395-407.

4. Бибииков, С.А. Цветовая коррекция на основе идентификации моделей по тестовым фрагментам изображений [Текст] / С.А. Бибииков, В.А. Фурсов // Журн. Компьютерная оптика. – 2008. – Т. 32, №3. – С 302-306.

Поступила в редколлегию 11.12.2013

**Рецензент:** д-р техн. наук, проф. А.М. Синотин, Харьковский национальный университет радиоэлектроники, Харьков.

### ПРОЦЕДУРА СТИСКАННЯ ВІДЕОІНФОРМАЦІЇ НА ОСНОВІ КОДУВАННЯ КОЛІРНИХ СКЛАДОВИХ ВІДЕОСИГНАЛА

А.С. Погодіна, Т.А. Колесникова, О.М. Кадушкевич

*При вирішенні задачі стиснення природним є питання, наскільки ефективно та чи інша система стиснення. Процес стиснення призводить до зменшення обсягу подання інформації на інформаційному носії. У результаті стиску відеоінформації поряд з об'єктивними факторами помилок виникають спотворення. Виходячи з цього, існують два типи стиснення: стиснення з спотвореннями представляється інформації, і стиснення, не допускає такі спотворення. Метою даного дослідження є запропонувати оптимальну процедуру стиснення, основою якої є кодування колірних складових, а також первісна колірна корекція відеофрагменту за допомогою розробленої програми на мові програмування Java. Таким чином, мета стиснення даних - забезпечити компактне і цілісне уявлення даних для їх економного зберігання або передачі по мережах.*

**Ключові слова:** стиснення, відеосигнал, колірний простір, колірна корекція.

### PROCEDURE FOR VIDEO COMPRESSION BASED ON CODING OF COLOR VIDEO SIGNAL COMPONENTS

A.S. Pogodina, T.A. Kolesnikova, O.N. Kadushkevich

*The video compression technique covers many applications from interactive systems to delivery of video information over telecommunications networks. The video compression algorithm relies on two basic techniques: compression distortion of the information, and compression that doesn't allow such distortions. Simulation results show that algorithm ensures optimal accuracy. As a result, video compression, along with objective factors should cause errors. Proceeding from this, there are two types of compression: compression distortion of the information, and compression that does not allow such distortions. The purpose of this study is to propose the optimal compression procedure, which is the basis of coding color components, as well as the original movie color correction using the program developed in the programming language Java. The purpose of data compression - to provide a compact and complete representation of data for efficient storage or transmission networks.*

**Keywords:** compression, video, color space, color correction.