

2. Бабаш А.В., Шанкин Г.П. Криптография / Учебное пособие., 2007. – 512 с.
3. Карпушкин, Э. М. Радиосистемы передачи информации / Уч. метод. пособие для студентов учреждений, обеспечивающих получение высшего образования по спец. "Радиоэлектронные системы". – Минск: БГУИР, 2008. – 62 с.

## СПОСОБ СЖАТИЯ ТЕРМИНАЛЬНЫХ ИЗОБРАЖЕНИЙ

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Клепацкий В. И.*

*Лашкевич Е. М. – м-р техн. наук, ассистент*

Ставится задача уменьшения объема передаваемых данных при передаче изображения удаленного компьютера по каналам связи в режиме реального времени. Проведен анализ существующих алгоритмов сжатия, на основе которого обоснована необходимость применения предложенного комплексного способа, главным достоинством которого является невысокая нагрузка на канал связи.

Одним из типов видеоданных, необходимость сжимать который существует в реальном времени, является видео, происходящее на экране пользователя и передаваемое на экран удаленного компьютера. На сегодняшний день чаще всего это видео высокого разрешения, для которого требуется канал с высокой пропускной способностью, которой во многих случаях недостаточно. В связи с этим возникает задача минимизировать объем данных, передаваемых по сети для управления удаленным компьютером.

Существуют технологии, позволяющие на порядки сжимать изображение (фрактальное сжатие и др.), но требовательные к быстродействию, что не удовлетворит условиям передачи изображения в реальном времени.

В терминальном режиме работы пользователь чаще всего работает с базами данных или документами, поэтому основные изменения изображения сосредоточены вокруг курсора, и глобально изображение на экране меняется нечасто. Этим можно воспользоваться для уменьшения количества сетевого трафика.

На сегодняшний день существует несколько программных продуктов, позволяющих передавать изображение удаленного компьютера (Microsoft RDP, Citrix ICA, Remote Administrator, Team Viewer и др.), и использующих различные алгоритмы. Общие алгоритмы сжатия видео описаны в [3]. Исходя из изложенных в учебном пособии сведений, можно сделать вывод, что для сжатия терминального видео наиболее подходящими являются 2 алгоритма: попиксельное и поблочное сравнение кадров, т.к. кадры похожи между собой и сравнение необходимо осуществлять в реальном времени.

В [1] используется алгоритм, основанный на попиксельном сравнении изображений. Каждый десятый ключевой кадр кодируется и передается независимо. Для каждого промежуточного кадра записываются номера строк и столбцов, в которых есть изменившиеся пиксели относительно ключевого кадра, а затем – цвета пикселей, находящиеся на пересечении этих строк и столбцов. Цвета пикселей, находящиеся на пересечении этих строк и столбцов образуют изображение меньшего размера, которое можно сжимать так же, как и ключевой кадр. Остальную часть промежуточных кадров можно восстановить по ключевому кадру. Таким образом, получается алгоритм сжатия без потери информации с линейной трудоемкостью. Входные данные для этого алгоритма – 2 изображения, над которыми выполняется попиксельная операция хог. Выходные данные – массив, количество элементов которого равно количеству пикселей в исходном изображении. Каждый элемент массива является индикатором равенства или неравенства пикселей ключевого и анализируемого изображения.

Алгоритм, основанный на поблочном сравнении изображений, используется TightVNC [2]. Изображение делится на небольшие неперекрывающиеся квадратные области, называемые ранговыми блоками. По сути, разбивается на пронумерованные квадраты. Определяются номера блоков, в которых есть хотя бы один изменившийся пиксель относительно соответствующего блока ключевого кадра. Изображение меньшего размера, составленное из изменившихся блоков пикселей можно сжимать теми же алгоритмами, что и ключевой кадр. При увеличении размера блока уменьшаются накладные расходы, связанные с хранением номеров изменившихся блоков, но вместе с тем уменьшается и точность определения изменившейся области. Экспериментальным путем установлено, что при размере блока 8\*8 пикселей достигается максимальная степень сжатия с учетом количества передаваемых номеров блоков.

Оба этих алгоритма приблизительно одинаковы по скорости работы. Алгоритм, основанный на попиксельном сравнении, оказывается быстрее при вводе пользователем текста или при сворачивании/разворачивании окна. Алгоритм, основанный на поблочном сравнении демонстрирует лучшие результаты при скроллинге и перемещении окна на некоторое расстояние, близкое к диагональному. Ни один из этих алгоритмов не является безусловно лучшим по степени сжатия.

Для усовершенствования рассмотренных алгоритмов предлагается следующий способ:

Отказаться от ключевого кадра и каждый раз в качестве ключевого использовать предыдущий кадр, ведь он все равно уже отрисован на экране, т.е. полностью передавать только самый первый кадр:

FRAME1= First Bitmap Image;

Массив отличий каждого следующего кадра от предыдущего получаем по формуле

Differences Array(n+1) = FRAME(n) XOR FRAME(n+1), где

FRAME1 – первый кадр, который будет отрисован;

Differences Array(n+1) – массив, несущий информацию об отличии следующего кадра от текущего;

FRAME(n)– текущий кадр на экране;

FRAME(n+1)– расчётный, следующий за текущим кадр.

Сжимаем Differences Array(n+1)

Для сжатия данных после попиксельного сравнения целесообразно использовать алгоритм RLE. (англ. Run-length encoding, RLE) или Кодирование повторов — простой алгоритм сжатия данных, который оперирует сериями данных, то есть последовательностями, в которых один и тот же символ встречается несколько раз подряд. При кодировании строка одинаковых символов, составляющих серию, заменяется строкой, которая содержит сам повторяющийся символ и количество его повторов. Учитывая частоту передачи кадров массив в абсолютном большинстве случаев будет состоять из большого количества идущих подряд нулей, т.е. таким образом обеспечивается многократное сжатие без потерь.

При необходимости еще более уменьшить нагрузку на канал можно применять фильтрацию. В зависимости от типа данных, который предполагается использовать, можно выбрать предпочтительный алгоритм фильтрации (более подходящий для текста либо для изображений). Таким образом, перед отправкой при помощи фильтра изображение сжимается, а после при помощи другого фильтра восстанавливается практически без потери качества. Огрехи могут быть заметны лишь при использовании на экране мелкого шрифта.

Данный способ будет прекрасно работать на канале очень маленькой ширины (Dial UP, GPRS), что позволит иметь полноценный комфортный графический удаленный доступ к компьютерам без необходимости наличия широкополосного интернета. Вместе с тем за счет того, что операция XOR является очень быстрой, а массив различий хорошо сжимается простейшим алгоритмом сжатия, это все остается нетребовательным к вычислительным ресурсам компьютера.

Комплексное использование этих трех методов позволит минимизировать количество данных, передаваемых по сети, при приемлемом качестве получаемого изображения.

Список использованных источников:

1. Патент № 7224731. США. Motion estimation/compensation for screen capture video / S. Mehrotra. Опубликовано, 29.05.2007.
2. Tight VNC [Электронный ресурс]. – Электронные данные. – Режим доступа : <http://www.tightvnc.com/>
3. Сэлмон, Д. Сжатие данных, изображений и звука. / Д. Сэлмон. - Москва: Техносфера, 2006.

## SDL TRIDION – МОЩНАЯ И ГИБКАЯ СИСТЕМА УПРАВЛЕНИЯ КОНТЕНТОМ WEB-САЙТОВ

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Кокичев С. В.

Сечко Г.В. – канд. техн. наук, доцент

В современном мире уже ни одна уважающая себя компания не обходится без собственного web-сайта. Чем больше информации содержит сайт, тем сложнее им управлять и когда объем информации достигает определенного уровня, то уже не обойтись без системы управления контентом. Таких систем существует достаточно много и у каждой есть свои преимущества для разных степеней наполненности сайта информацией. Одной из мощнейших систем управления контентом является продукт голландской компании SDL Tridion

Языковая наследственность в Tridion'e



Данная система управления контентом замечательно подойдет для компании, которая продает свой продукт, а так же осуществляет поддержку покупателей по всему миру. Многоязычность и наследование – это основные преимущества SDL Tridion. Система многоязычности здесь реализована по принципу наследственности - материал, созданный на языке, с наибольшим приоритетом копируется во все остальные языковые версии сайта, после чего его можно переводить или нет. Контент создается на английском и автоматически клонируется во все лежащие ниже языковые версии (наследники), где дальше он может переводиться на соответствующие языки. Страницы из каждой языковой версии публикуются в отдельные папки на сервере, к которым можно привязать любые доменные имена.

Для того чтобы начать создавать контент в Tridion, нужно создать т.н. схемы. Схема — это набор полей с разными свойствами.

Например, для статей вам надо будет один набор полей, а для продукта другой. Далее, вся вводимая