

УДК 681.327

В.В. Прончев, А.В. Тузиков

СЖАТИЕ ИЗОБРАЖЕНИЙ БЕЗ ПОТЕРЬ ДЛЯ УДАЛЕННОГО УПРАВЛЕНИЯ КОМПЬЮТЕРОМ С МОБИЛЬНЫХ УСТРОЙСТВ

Предложен простой и эффективный алгоритм сжатия изображений без потерь для применения в системах «удаленного рабочего стола». Алгоритм рассчитан на применение в сотовых телефонах. Используются две особенности изображений, формируемых приложениями, работающими в среде Windows: наличие прямоугольных областей одного цвета и относительно небольшое число одновременно использующихся цветов.

Введение

Одним из наиболее универсальных способов удаленного контроля и управления персональным компьютером является организация «виртуального удаленного рабочего стола» (Remote Desktop). Удаленный рабочий стол позволяет пользователю видеть приложения, запущенные на удаленном компьютере, и управлять ими по сети. Существует два основных подхода к передаче и формированию изображений в таких системах.

Первый подход заключается в передаче клиенту параметров вызова функций GDI (Graphics Device Interface). Клиент сам формирует изображение, отображает текст и т. п. Данный подход реализован компанией Microsoft в протоколе RDP (remote desktop protocol [1]) и используется в Terminal Server.

Более простым подходом является передача изображения, которое уже сформировано сервером. Представителями данного класса систем являются RAdmin, RealVNC и многие другие.

С появлением карманных персональных компьютеров (КПК) и беспроводных средств связи стала актуальной возможность использования КПК в качестве клиентов систем Remote Desktop. Компания Microsoft встроила Terminal Server Client в свою операционную систему для КПК Pocket PC 2002. Изучив возможности данной программы, можно отметить следующие ее недостатки:

- Скорости передачи данных по сотовой сети (2-4 кб/с) недостаточно для комфортной работы.
- Передаваемый трафик имеет большой объем.
- Операции скроллинга виртуального рабочего стола связаны с получением подтверждения от сервера; это вносит задержку, в которой нет необходимости.
- Отсутствует возможность поворота изображения на 90°.
- Элементы управления занимают часть площади экрана КПК.

Одной из популярных систем удаленного управления компьютером является open-source проект RealVNC [2]. К преимуществам данной системы можно отнести наличие клиентов под большое число платформ. Существует клиент для КПК под управлением Windows CE 3.0. На данный момент средства сжатия изображения в этом пакете ограничены алгоритмом RLE (Run Length Encoding [3], кодирование длин повторов) и операцией копирования фрагментов изображения. Операция копирования используется при движении окон и прокрутке. Протокол передачи кадров не позволяет организовать работу с «нулевым трафиком» в случае, если изображение не изменяется.

1. Мобильный телефон как средство удаленного управления компьютером

Существенным недостатком большинства моделей КПК является отсутствие поддержки нового стандарта беспроводной связи Bluetooth. В большинстве случаев единственным средст-

вом связи с сотовым телефоном является ИК-порт. Держать телефон в пределах прямой видимости с КПК на практике очень неудобно.

С развитием современных технологий стало возможным перенести задачу управления компьютером со связки КПК–телефон на сам сотовый телефон. Рассмотрим некоторые преимущества такого подхода:

- Телефон класса Nokia 3650 существенно дешевле пары КПК–телефон.
- Не требуется решать проблему коммуникации между КПК и телефоном.
- Добавление еще одной возможности в ПО сотового телефона, в отличие от постоянного ношения дополнительного устройства (КПК), не создает никаких неудобств пользователю.

Кратко опишем возможности Nokia 3650/7650 [4]:

- экран с разрешением 176x208, 4096 цветов;
- процессор семейства ARM9, 104 МГц, 4 Мб RAM, до 256 Мб Flash;
- операционная система Symbian 6.1;
- органы управления – джойстик и кнопки;
- видеокамера 640x480, поддержка беспроводной связи Bluetooth, ИК-порт;
- кодирование и декодирование MPEG4 видео и звука MP3 в реальном времени, имитация большинства пультов дистанционного управления и многое другое.

Разрабатываемый программный комплекс ориентирован на работу в сотовых телефонах под управлением операционной системы Symbian (Nokia 7650/3650/3660/6600, Siemens SX1, Samsung SGH-D700). В перспективе возможен перенос ПО клиента на другие платформы. Основные особенности и возможности программного комплекса:

- Исключение любого рода избыточности в передаваемой информации. Передача изображения в формате клиента (сотового телефона, 12 бит/пиксель для Nokia 3650/7650).
 - Эффективный алгоритм сжатия изображения без потерь.
 - Максимальная простота ПО клиента.
 - Гибкая настройка, контроль потока передаваемого трафика и скорости обновления кадров со стороны сервера. Возможность передачи нескольких кадров до получения подтверждения для компенсации задержек в сети.
 - Протокол, позволяющий работать в режиме с «нулевым трафиком», если нет изменений в изображении. Находясь в режиме ожидания, телефон не тратит энергию аккумулятора на отправку пакетов по сотовой сети, а также экономит дорогостоящий интернет-трафик.
 - Совмещение скроллинга (прокрутки) видимой части виртуального рабочего стола с управлением курсором мыши. Телефон «видит» ту часть кадра, где в данный момент находится мышь.
 - Передача и хранение в памяти телефона только части кадра, содержащей видимый фрагмент и небольшой запас для прокрутки. Данная возможность позволяет работать с кадром любого разрешения.
 - Эмуляция клавиатуры на телефоне.
- В перспективе возможна реализация следующих алгоритмов:
- Удобное масштабирование (уменьшение) изображения для упрощения работы с высокими разрешениями.
 - Реализация алгоритма сжатия с потерями, что позволило бы увеличить производительность при показе видеопоследовательностей и дало бы возможность превратить, например, телефон в систему видеонаблюдения. На экране телефона появится возможность наблюдать в реальном времени изображения, получаемые компьютером с видеокамер.

В сотовых сетях стандарта GSM возможности по передаче интернет-трафика серьезно ограничены: низкая скорость, малая емкость сетей и высокая стоимость трафика. Именно в данной области становятся востребованными эффективные алгоритмы компрессии.

Можно выделить следующие особенности формируемых изображений, характерные для большинства приложений Windows и других оконных сред:

- наличие прямоугольных областей одного цвета;
- относительно небольшое число одновременно используемых цветов.

Поставленная задача имеет ряд отличий от решавшихся ранее задач кодирования изображений в реальном времени:

- Малое разрешение сжимаемых изображений и высокие вычислительные мощности современного компьютера позволяют использовать сложные и трудоемкие алгоритмы кодирования.
- Производительность и объем оперативной памяти компьютера существенно превосходят возможности современных сотовых телефонов. Алгоритм декодирования должен обладать достаточной скоростью для работы на сотовом телефоне в реальном времени. Желательно также максимально снизить объем оперативной памяти, необходимый для декодирования.

2. Сжатие изображения

В основе изложенного далее алгоритма лежит поиск и кодирование монотонных прямоугольных областей максимальной площади. Для кодирования положения и размеров применяются модификации алгоритма адаптивного арифметического кодирования [3, 5]. Для кодирования значений цвета используется стратегия Move To Front (MTF) [6], после чего применяется локально-адаптивное арифметическое кодирование.

Сжатие изображения выполняется в два этапа:

- поиск прямоугольников одного цвета, кодирование их положения, размера и цвета;
- кодирование цвета точек, не входящих ни в один из прямоугольников; используется информация о цвете ранее закодированных смежных точек.

2.1. Поиск прямоугольников

Для каждой точки, не включенной ни в один из ранее закодированных прямоугольников, выполняется поиск прямоугольной области одного цвета с левым верхним углом в данной точке. Выбирается область, содержащая наибольшее число незакодированных точек. В случае если число незакодированных точек в найденной области меньше некоторой константы (S_{min}), то данная точка пропускается и счетчик пропущенных точек увеличивается на единицу.

Параметр S_{min} определяет минимальную полезную площадь, при которой прямоугольник кодируется алгоритмом. На рис. 1 показан результат работы алгоритма при $S_{min}=16$.



Рис. 1. Оригинальное изображение (слева) и частично декодированное изображение, полученное на основе прямоугольных областей, закодированных на первом этапе работы алгоритма (справа)

2.2. Кодирование положения прямоугольника

Положение определяется количеством пропущенных незакодированных точек до левого верхнего угла прямоугольника.

Пусть X – количество пропущенных точек. Используется адаптивное арифметическое кодирование с диапазоном кодируемых значений $[0, N]$. Если $X < N$, то кодируем значение X . В противном случае на вход арифметического кодера поступает значение N , $X = X - N$ и повторяется процесс кодирования для нового значения X .

Допустим, $N = 47$. Кодируя, например, значение $X = 100$, алгоритм построит последовательность символов 47, 47, 6.

Особенностями кодируемой величины является широкий диапазон возможных значений и малая вероятность появления значений, превышающих некоторый предел N . Рассмотренное тривиальное преобразование позволяет добиться высокой степени компрессии подобных данных при относительно небольших затратах памяти.

2.3. Кодирование размеров

Максимальные значения таких величин, как длина и ширина прямоугольника зависят от состояния алгоритма и могут быть вычислены как алгоритмом сжатия, так и алгоритмом декомпрессии. Максимальные значения длины и ширины – расстояния до краев изображения.

В реализованном алгоритме адаптивного арифметического кодирования количество символов входного алфавита может быть динамическим. При кодировании размеров прямоугольника алгоритм меняет размер алфавита, отсекая недопустимые варианты.

2.4. Кодирование значений цвета

Алгоритм Move To Front (MTF, «двигайся в начало списка») [6] является простейшим представителем семейства алгоритмов обновления списка [6, 7]. Алгоритмы этого семейства содержат список, в котором находятся символы входного алфавита. Символ входного алфавита кодируется индексом символа в списке, после чего список обновляется. Для декодирования достаточно взять символ из списка с нужным индексом и обновить список.

В случае MTF символ ставится в начало списка в позицию «0». Остальные символы увеличивают свою позицию в списке на единицу. Таким образом, повторяющийся символ будет иметь индекс «0», а часто встречающиеся символы будут находиться в начале списка. Например:

$$\begin{aligned} \text{MTF}(2, 1, 1, 1, 1, 0, 2, 0, 2) &= (2, 2, 0, 0, 0, 2, 2, 1, 1) \\ \text{Начальный список} &: (0, 1, 2) \\ \text{MTF}(2) &= 2, \text{ список } (2, 0, 1) \\ \text{MTF}(1) &= 2, \text{ список } (1, 2, 0) \\ \text{MTF}(1) &= 0, \text{ список } (1, 2, 0) \end{aligned}$$

Еще одним популярным представителем данного семейства является алгоритм MTF-1. Его отличие от алгоритма MTF заключается в том, что новый символ занимает позицию «0», только в том случае, если он занимал ранее позицию «0» или «1», и занимает позицию «1» в противном случае.

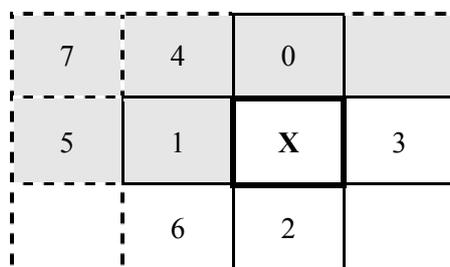


Рис. 2. Порядок занесения значений цвета смежных пикселей в список, X – текущий пиксель

Преобразование MTF позволяет кодировать цвет в контексте ранее закодированных значений цвета. Чем ближе встречалось значение ранее, тем меньше будет индекс, полученный алгоритмом. Для рассматриваемых в данной статье изображений малые значения на выходе

преобразования будут встречаться с большими частотами. При энтропийном кодировании результата преобразования MTF достигается степень сжатия выше, чем при кодировании самих значений цвета.

Стратегия MTF была адаптирована к двумерной структуре кодируемых данных. На втором этапе работы алгоритм использует цвета ранее закодированных смежных пикселей (рис. 2). На каждом шаге цвета пикселей 0, 1 переносятся в начало списка. Цвета пикселей 2, 3 также переносятся в начало списка, если их значения известны декодеру (были включены ранее в один из прямоугольников)

Указанное преобразование позволяет упорядочить список цветов в порядке уменьшения вероятности совпадения с кодируемым значением. Как указывалось выше, кодированию подвергается индекс цвета в списке. Для кодирования результата преобразования используется локально-адаптивная схема. После обработки каждой строки раstra накопленные частоты делятся на два путем сдвига значений счетчиков на 1 бит. Данная операция препятствует неограниченному росту значений счетчиков, а также позволяет алгоритму быстро адаптироваться к изменениям частот кодируемых символов. Значение приращения счетчиков частот выбирается достаточно высоким, чтобы обеспечить быстрое накопление статистики.

3. Арифметическое кодирование

В энтропийном кодировании основную роль играют два основных алгоритма: кодирование по Хаффману и арифметическое кодирование [3, 5]. Арифметическое кодирование позволяет эффективно сжимать данные, в которых один из символов входного алфавита имеет вероятность, близкую к единице, достигая степени сжатия менее 1 бит/символ.

Ниже приведен алгоритм арифметического кодирования последовательности символов. Частоты символов входного алфавита предполагаются заданными.

Берем отрезок $[0,1)$. Делим отрезок пропорционально частотам символов входного алфавита. Из полученных отрезков выбираем тот, который соответствует кодируемому символу. Для следующего символа разбиваем полученный отрезок аналогичным образом. Выбираем отрезок, соответствующий следующему символу, и т. д. Получаем некоторый отрезок $[low, high)$. Результатом работы алгоритма станет двоичное представление дробной части числа, лежащего внутри $[low, high)$.

Зная любое число из диапазона $[low, high)$, а также частоты, на основании которых проводилось разбиение на отрезки, декодер может найти, какому отрезку принадлежит число, и, соответственно, декодировать символ.

В данной работе реализована адаптивная схема арифметического кодирования. При запуске алгоритма все символы считаются равновероятными. Всем частотам присвоено значение единица. После обработки очередного символа кодировщик увеличивает его частоту на некоторую величину, от значения которой зависит скорость «адаптации» алгоритма на начальном этапе. Аналогично поступает декодер после получения символа.

Просмотр таблицы частот связан с большим числом обращений к памяти. Данные, поступающие на вход арифметического кодировщика, имеют неравномерное распределение. Алгоритм сжатия изображения построен так, что данные с меньшими абсолютными значениями встречаются с большей частотой. Это справедливо как для индекса цвета, так и для положения, длины и ширины прямоугольников. Арифметический кодировщик и декодер существенно сокращают время поиска, начиная просматривать таблицу частот с меньших абсолютных значений символа.

4. Сжатие последовательности кадров

При обработке последовательности кадров осуществляется кодирование «разности» между кадрами следующим образом:

- Вводится дополнительный специальный код для цвета – код совпадения значений цвета на текущем и предыдущем кадре.

- Осуществляется сдвиг предыдущего кадра (при необходимости).

- При совпадении значений цвета цвет данной точки заменяется на данный код.

Такой подход позволяет использовать всего один кадровый буфер для декодирования. Декодер, встречая код совпадения, оставляет в кадровом буфере старое значение цвета.

Выполняя сжатие очередного кадра последовательности, арифметический кодировщик использует частоты, накопленные на предыдущих кадрах. Алгоритм MTF использует список индексов цветов предыдущего кадра. Чтобы избежать неограниченного роста значений счетчиков частот, а также улучшить локальную адаптивность алгоритма, значения частот делятся на два.

Алгоритм декомпрессии выполняет два прохода: на первом проходе декодируются прямоугольники – крупные объекты, затем декодируются оставшиеся точки изображения – мелкие детали. Фоном является изображение предыдущего кадра. Данная особенность позволяет показать частично сформированное изображение, получив только часть сжатых данных.

5. Сравнение алгоритмов сжатия

Несмотря на относительную простоту и отсутствие в алгоритме средств поиска повторяющихся последовательностей, описанный метод показал высокую степень компрессии, существенно опередив алгоритмы семейства LZ77, и показал степень сжатия выше, чем один из самых мощных универсальных алгоритмов сжатия – PPMd [8].

Результаты теста на фрагменте цветного изображения (см. рис. 1), содержащем окна, текст, элементы управления и иконки, приведены в табл. 1. Данные для архиваторов RAR, 7-Zip и PKZIP приведены без учета заголовков архивов.

Таблица 1

Сравнение алгоритмов сжатия (для изображения на рис. 1)

Метод	Полученный размер, байт	Отношение, %
Рассматриваемый алгоритм, первый запуск	2196	0
Рассматриваемый алгоритм, повторный запуск, используются накопленные таблицы частот и список цветов	2124	-3.3
GIF	4792	+118
PKZIP v4.00	3195	+45.5
RAR 3.0, default	2982	+35.8
7-Zip 3.12, default	2855	+30.0
RAR 3.0 (-m5, maximum compression)	2655	+20.9
PPMd (7-Zip 3.12, -m0=PPMd)	2274	+3.6

Алгоритм показал степень сжатия 0,51 бит/пиксель, превзойдя широко применяемые на практике алгоритмы, работающие с одномерными последовательностями данных. Следует отметить, что для работы алгоритмов семейства PPM (prediction by partial matching) требуются большие объемы памяти (для хранения дерева контекстов), а также большие вычислительные мощности при декодировании (арифметическому кодированию подвергается каждый кодируемый символ). Указанные недостатки ограничивают возможности применения алгоритмов PPM в мобильных устройствах.

Заключение

Можно выделить следующие преимущества предложенного алгоритма:

- Компактность и простота реализации (около 300 строк кода на языке C++ для декомпрессора).

- Невысокие требования к оперативной памяти. Алгоритм использует память только для частотных коэффициентов и списка цветов.
- Высокая скорость работы декодера на изображениях с высокой избыточностью.
- Возможность декодирования частично полученного кадра (прогрессивная декомпрессия), при этом крупные объекты будут показаны раньше, чем мелкие детали.
- Эффективное кодирование «разности» кадров при скроллинге (прокрутке). Сжимаемый кадр будет содержать большой пустой прямоугольник, соответствующий части предыдущего кадра. Алгоритм идеально подходит для его кодирования.

Рассмотренный алгоритм реализован в предварительной версии системы удаленного управления компьютером для сотовых телефонов Nokia 3650/7650. Производительность тестовой версии декодера в несколько раз превышает скорость приема данных в сотовой сети. Серверная часть работает под управлением ОС Windows. Программный комплекс успешно применяется на практике.

Список литературы

1. T.128 Draft – Application Sharing Protocol. <http://www.rdesktop.org/docs/t128.zip>.
2. Virtual Network Computing: the original open-source cross-platform remote control solution. <http://www.realvnc.com/>.
3. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин. – М.: Диалог-МИФИ, 2002. – 384 с.
4. Technical specs of Symbian OS 6 devices. <http://my-symbian.com/7650/techdata.php>.
5. Ghanbari M. Video coding: an introduction to standard codecs. – London, The Institution of Electrical Engineers, 1999. – 264 p.
6. Move To Front (MTF). <http://www.data-compression.info/Algorithms/MTF/index.htm>.
7. Albers S., Mitzenmacher M. Average Case Analyses of List Update Algorithms, with Applications to Data Compression // *Algorithmica*. – 1998. – № 21(3). – P. 312-329.
8. Шкарин Д.А. Повышение эффективности алгоритма РРМ // *Проблемы передачи информации*. – 2001. – Т. 37. – № 3. – С. 44-54.

Поступила 26.05.04

*Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: pronchev@mpen.bas-net.by*

V.V. Pronchev, A.V. Tuzikov

LOSSLESS IMAGE COMPRESSION FOR MOBILE REMOTE DESKTOP

Modern cell phones are used in a wide variety of applications. A lossless image compression algorithm for remote desktop systems is presented. The algorithm is simple, efficient and suitable for using in cell phones. It uses two particular features of images generated by most applications running in windows environment: a large amount of rectangular areas with the same color and relatively a small amount of colors used at the same time.