

Обробка інформації в складних технічних системах

УДК 004.627

Д.Э. Двухглавов, М.Ю. Струтинский

Национальный технический университет «ХПИ», Харьков

СПОСОБ ПЕРЕКОДИРОВКИ КООРДИНАТ МЕСТОПОЛОЖЕНИЯ ДЛЯ ПЕРЕДАЧИ В SMS-СООБЩЕНИЯХ

Рассматривается задача уменьшения размерности представления координат местоположения, актуальность которой обусловлена современными тенденциями использования средств сотовой связи. Рассмотрены особенности представления информации о местоположении объектов и предложено решение поставленной задачи на основе преобразования координат в набор символов из кодировки UTF. Разработан алгоритм перекодировки координат местоположения, представлен пример решения задачи с применением разработанного алгоритма и результаты анализа его эффективности, полученные путем сравнения результатов работы программной реализации алгоритма с результатами работы известных программ сжатия данных.

Ключевые слова: координаты местоположения, представление координат, передача координат местоположения в сетях сотовой связи.

Введение

В настоящее время неотъемлемым атрибутом человека является технические устройства, подключенные к сети мобильной связи. Все чаще при обмене информацией с использованием данных технических устройств возникает необходимость передачи SMS-сообщений, которые содержат местоположение различных объектов: собственных координат, координат достопримечательностей и других мест встречи. Передача такой информации на текущий момент требует формирования SMS-сообщений большого объема, что определяет высокую стоимость их отправки, а также увеличивает нагрузку на сеть. Представление одного набора координат в его чистом виде со стандартной, на данный момент, для систем GPS точностью занимает около 23 символов [1]. Если использовать известные алгоритмы сжатия на большом количестве наборов координат, то, несомненно, будет получен выигрыш в объеме. К таким алгоритмам следует отнести алгоритмы архивации и кодирования информации, которые представлены в [2 – 4]. Однако в этих алгоритмах не учитывается специфика представляемой информации, используется понятие "данные" без учета семантики. В тоже время очень часто эффективность сжатия данных зависит от их типа. Поэтому, изучив особенности представления координат, можно уменьшить занимаемый ими объем еще больше.

Целью статьи является разработка способа перекодировки координат местоположения, позволяющего формировать более компактные по занимаемой памяти SMS-сообщения при обмене информацией в системах сотовой связи за счет учета особенностей координат местоположения.

Основная часть

Для сжатия лучше всего подходит алгоритм, который учитывает свойства данных определенного вида и использует их для выполнения сжатия. На данный момент существует огромное множество типов информации, но не для всех из них существует алгоритм, позволяющий максимально эффективно уменьшать объем, занимаемый этой информацией. Для сжатия такой информации, как координаты местоположения используют алгоритмы для сжатия чисел. Но также необходимо учитывать и то, что координаты отличаются от других чисел тем, что изменяются в определенных пределах и необходимая точность, выражаемая количеством знаков после запятой, тоже ограничена [1]. Рассмотрим семантику информации о координатах, которая является важной для разрабатываемого способа.

Координаты записываются цифрами и по широте принимают значения от -180 до 180, а по долготе от -90 до 90 с точностью 8 знаков после запятой [5]. Эту специфику координат можно использовать для разработки алгоритма, который будет использовать эти особенности и проявит себя лучшим образом при обработке значений в указанных диапазонах. Пусть требуется передать n наборов координат, каждый из которых содержит широту (latitude) и долготу (longitude) местоположения некоторого объекта. Данную совокупность представим в виде матрицы координат X размерностью $n \times 2$:

$$X = \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \dots & \dots \\ x_n & y_n \end{pmatrix}, \quad (1)$$

Значение каждой координаты в этой матрице представлено с точностью асс знаков после запятой. Задача состоит в уменьшении объема этой матрицы за счет уменьшения места, занимаемого координатами без потерь точности самих координат.

Предлагается представить координаты в виде символов кодировки UTF, как одной из самых распространенных кодировок в мире. Для выполнения преобразования понадобится определенное количество code point, предоставляемое этой кодировкой. Для UTF максимальное количество code point составляет $\max = 1114112$. Элементы code point могут быть только положительными, следовательно, необходимо выполнить преобразование значений координат, которые изначально варьируются в пределах от -180 до 180 по широте и от -90 до 90 по долготе. Для этого прибавим к матрице X матрицу S, где матрица S имеет размер $n \times 2$, является статичной и состоит из n наборов (180, 90). Другими словами матрица S состоит из максимальных значений широты и долготы. При сложении с матрицей координат X будет получена матрица, которая содержит исключительно положительные числа от 0 до 360 по широте и от 0 до 180 по долготе. Теперь, когда значения координат положительные, необходимо избавиться от дробной части координаты – все code point являются целыми числами. Для этого все координаты умножаются на величину заданной точности асс. Для дальнейшей разработки необходимо определить ограничение:

$$(X + S) \cdot \text{асс} < \max^2. \quad (2)$$

Если условие выполняется, вычисляются матрицы INT и REST – матрицы целых частей и остатков от деления. Любое целое число можно разбить на три других числа, с помощью которых можно получить исходное число по формуле $x = a \cdot b + c$. В данном случае в качестве b будет выступать число max, следовательно, для представления одного большого числа нам понадобится 2 меньших числа, это и есть числа в матрицах INT и REST. Запишем формулы для расчета этих матриц:

$$\text{INT} = \lfloor (X + S) \cdot \text{асс} / \max \rfloor, \quad (3)$$

$$\text{REST} = (X + S) \cdot \text{асс} - \text{INT} \cdot \max. \quad (4)$$

Далее для всех элементов матриц INT и REST применяется операция приведения к символу, которую обозначают char() [6]. Затем к приведенным символам применяется операция конкатенации &: $s_1 \& s_2 \rightarrow s_1s_2$, т.е. символ s_2 записывается сразу за символом s_1 . Таким образом, элементы матрицы преобразованных координат X^* получаются в результате применения такой формулы:

$$X^* = \text{char}(\text{INT}) \& \text{char}(\text{REST}). \quad (5)$$

Матрица X^* , которая будет получена в результате перекодировки, будет иметь вид:

$$X^* = \begin{pmatrix} sx_1 & sy_1 \\ sx_2 & sy_2 \\ \dots & \dots \\ sx_n & sy_n \end{pmatrix}, \quad (6)$$

где sx_i, sy_i – преобразованные координаты, полученные в результате конкатенации INT и REST частей числа, представленных в виде символов. На этом алгоритм сжатия завершен, каждая координата теперь представлена двумя символами выбранной кодировки, а каждый набор координат [latitude, longitude] – четырьмя. Для формирования строки для отправки другому адресату выполняется конкатенация всех элементов матрицы в произвольном удобном порядке.

Далее рассмотрим порядок обратного перекодирования координат. Алгоритм получает на вход матрицу X^* и матрицу S, значения асс и max. Вначале матрица X^* раскладывается на матрицы INT и REST, после чего элементы матрицы X определяются с использованием формулы (7):

$$X = \frac{(\text{INT} \cdot \max + \text{REST})}{\text{асс}} - S. \quad (7)$$

Полученная матрица X представляет собой матрицу с наборами координат [latitude, longitude] с точностью асс.

Для случая с преобразованием координат с точностью до 10 знаков после запятой и используемой кодировкой UTF ограничение (2) всегда будет выполняться [3]. В случае, если условие (2) не выполняется, необходимо дополнительно построить нулевую матрицу повторений R размерностью матрицы координат и добавлять +1 к соответствующему значению в матрице повторений на каждой итерации алгоритма. Цикл повторяется, пока выполняется условие $\text{result}_i \% \max > 1$, где $\text{result}_0 = (X + S) \cdot \text{асс}$, а затем пересчитывается result.

Каждый новый result_{i+1} вычисляется на основе предыдущего result_i по формуле $\text{result}_{i+1} = \frac{\text{result}_i}{\max}$. Процесс продолжается, пока $\text{result}_i \% \max > 1$. По завершению вычислений, матрица R конкатенируется с матрицами INT и REST при вычислении X^* и, соответственно, учитывается при восстановлении X:

$$X^* = \text{char}(\text{INT}) + \text{char}(R) + \text{char}(\text{REST}), \quad (8)$$

$$X = \frac{(\text{INT} \cdot R \cdot \max + \text{REST})}{\text{асс}} - S. \quad (9)$$

Максимальную эффективность алгоритм показывает при применении к сжатию координат в SMS-сообщениях. SMS-сообщение можно отправить только в одной кодировке. При написании сообщения в кодировке ASCII максимальный объем одного сообщения составляет 160 символов, а при написании сообщения в кодировке UTF – 70 символов [7, 8]. При использовании кодировки UTF в сообщении

и необходимости отправки координат в стандартном виде координаты займут от 21 до 26 символов [7, 9]. Представленный в работе алгоритм сжатия позволяет сократить этот объем до 6-8 символов. Первый и последний символ занимают скобки, которые укажут на наличие координат в этой области, 4-6 символов посередине занимают координаты, множитель является небольшим числом и занимает 1 символ, а остаток от деления может занимать 1-2 символа. Алгоритм поддается улучшению, на что и будет направлена дальнейшая над ним работа. Если всегда занимать верхний и нижний суррогаты символа по-

лезными данными, можно сократить место, занимаемое координатами еще больше, но, скорее всего, в ущерб скорости перекодировки координат.

Для демонстрации работы алгоритма разработано приложение для обмена сообщениями с координатами для OS Android. Координаты в сообщении получаются после преобразования сообщений с использованием представленного в работе алгоритма. В табл. 1 представлена сравнительная характеристика сообщений с сжатыми координатами и координатами в их стандартном виде. Сообщения представлены в стандартном Android мессенджере.

Таблица 1

Сравнительная характеристика сообщений

Координаты без сжатия	Сжатые координаты
координаты: (77.3876548;47.7323097)(-73.01874 83;-69.5615096)(-13.8463783;-158.9 41968)	координаты: (€)(e F)(ċ Ū)
координаты: (52.9276801;-6.723555)(64.2710313; -95.7630305)(46.6667472;5.5833014)(51.3100771;11.6174341)(56.87923 77;14.8140666)(-23.1948387;-38.3654 514)(36.5497317;-98.5824271)	координаты: (ğ)(ċ ħ X)(ħ)(ğ)(ċ)(ħ)(ħ)
координаты: (15.6793058;-10.6685625)(57.90553 06;30.6420178)(-30.9245221;148.33 33529)(-1.6741411;-178.2073781)(66 .7182013;-47.9110102)(64.083982;-2. 1934907)(-33.708328;-25.5856746)(2 5.1240961;-103.8880994)(45.043471 8;19.3170263)(56.9255171;14.75583 12)(-23.5489399;163.0542349)(-55.7 76573;-168.0852215)(59.2492359;16 .1721288)(-6.2147269;22.5069935)(8 3.5269712;-151.8744113)	координаты: (ħ)(ċ)(ħ)(ħ)(ħ)(ħ)(ħ)(ħ) (ħ)(ħ)(ħ)(ħ)(ħ)(ħ)(ħ)(ħ)

Из таблицы видно, что сжатые сообщения занимает гораздо меньше места по сравнению с сообщениями без сжатия координат. Вместо отправки 6 сообщений пользователю будет достаточно только 2 для передачи 15 наборов координат. Координаты в сжатом виде нет необходимости разделять точкой с запятой, потому что 2 координаты всегда занимают 4 символа, первые два из которых относятся к широте, а два остальных – к долготе [10]. Для обозначения координат используются скобки, которые пользователь системы Android не сможет набрать в стандартных мессенджерах [11], эти скобки имеют специальный номер в UTF и нужны для выделения координат из сообщения.

К преимуществам алгоритма можно отнести высокую скорость сжатия данных, отсутствие потерь при сжатии, возможность интеграции с другими алгоритмами сжатия для еще большего сокращения объема, занимаемого координатами без потерь качества, простоту в реализации. К недостаткам алгоритма можно отнести необходимость обработки информации при ее отправке и получении, недостаточно высокая степень сжатия по сравнению с другими алгоритмами на больших объемах данных при оптимизации хранения данных в системе, невозможность

использования упрощенной версии алгоритма для больших чисел, необходимость поддержки системой, отправляющей и принимающей информацию выбранной в алгоритме кодировки. Чем больше в выбранной кодировке символов, тем эффективнее будет алгоритм сжимать большие числа, представляя их в виде символов выбранной кодировки.

Для снятия показателей времени работы алгоритма и сравнения объемов информации разработано приложение для сжатия координат различными методами, записи в файл и передачи файла по сети, в процессе работы приложения замеряется время сжатия, передачи и общее время с момента начала и окончания обработки данных [4]. Сравнительная характеристика отправки наборов координат в различном объеме с использованием разных комбинаций алгоритмов шифрования представлена в таблицах 2-4. Анализ показателей эффективности проводится на компьютере с такими характеристиками: процессор - Intel(R) Core(TM) i3-3110M, CPU 2400 МГц, ядер - 2; оперативная память (RAM) - 6,00 Гб; ОС Microsoft Windows 8.1 x64, версия 6.3.9600, сборка 9600; сетевой адаптер - Intel(R) Centrino(R) Wireless-N 135.

Таблица 2

Показатели эффективности (100 пар координат)

	RAW	CCA	zip	7z	CCA+7z
Занимаемый объем (байт)	2316	1318	1333	1361	1247
Время упаковки (мс)	0	4	20	29	31
Время распаковки (мс)	0	18	16	18	48
Время передачи (мс)	18.1	10.3	10.4	10.6	9.7

Таблиця 3

Показатели эффективности (10000 пар координат)

	RAW	CCA	zip	7z	CCA+7z
Занимаемый объем (байт)	230670	132269	109304	102966	99540
Время запаковки (мс)	0	54	76	162	111
Время распаковки (мс)	0	378	27	30	412
Время передачи (мс)	1802.1	1033.3	853.9	804.4	777.6

Таблиця 4

Показатели эффективности (1000000 пар координат)

	RAW	CCA	zip	7z	CCA+7z
Занимаемый объем (кб)	23055	13230	10664	9789	9766
Время запаковки (с)	0	0.673	4.848	31.009	6.512
Время распаковки (с)	0	11.052	0.549	1.010	13.347
Время передачи (с)	184.44	105.84	85.312	78.312	78.128

Выводы

В результате анализа особенностей представления координат местоположения был предложен способ их перекодировки, который позволяет уменьшить объем памяти, занимаемый ими в SMS-сообщениях. В ходе разработки алгоритма показано, что уменьшение объема может быть достигнуто за счет преобразования координат в символы кодировки UTF.

Практическая ценность результатов работы определена по результатам исследования эффективности обработки массивов координат разного размера. Использование предложенного способа обеспечивает выигрыш по всем параметрам обработки координат местоположения по сравнению с другими программами при обработке наборов координат количеством до 100. При обработке наборов большего объема наилучшие результаты обеспечивает интеграция предложенного способа с алгоритмом сжатия данных 7z.

Список литературы

1. Иванов А.В. Комплексные оптимальные алгоритмы обработки информации в навигационных системах подвижных наземных объектов / А.Б. Иванов. – Радиотехника, 2010. – № 5 – 87 с.
2. Nelson M. *The Data Compression Book*, 2nd edit / M. Nelson J. Gailly. – New York, 1996. – 541 p.
3. Khalid S. *Introduction to Data Compression* / S. Khalid. – San Francisco: MKPs, 1996. – 471 p.

4. Ватолин Д.А. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео / Д.А. Ватолин, А.А. Ратушняк, М.В. Смирнов, В.Е. Юкин. – Диалог-МИФИ, 2002. – 384 с.

5. Громаков Ю. А. Технологии определения местоположения в GSM и UMTS / Ю. А. Громаков, А. В. Северин, В. А. Швецов. – М.: Эко-Трендз, 2005. – 144 с.

6. Jebamalar E. *International Journal of Applied Information Systems* / E. Jebamalar, D. Asir, G. Singh. – FCS, New York, USA, 2013. – №4 – 240 p.

7. Ле-Бодик Г. Мобильные сообщения: службы и технологии SMS, EMS и MMS / Г. Ле-Бодик. – М.: КУДИЦ-ОБРАЗ, 2005. – 448 с.

8. Романова Ю.Д. Информатика и информационные технологии / Ю.Д. Романова, И.Г. Лесничая, В.И. Шестаков, И.В. Миссинг, П.А. Музычкин. – М.: Эксмо, 2008. – 592 с.

9. Габидулин, Э.М. Префиксный код. Лекции по теории информации / Э.М. Габидулин, Н.И. Пилипчук. – М.: МФТИ, 2007. – С. 43-49. – 214 с.

10. Иванов А.В. Комплексные оптимальные алгоритмы обработки информации в навигационных системах подвижных наземных объектов с контролем целостности навигационного обеспечения / А.Б. Иванов. – Радиотехника, 2010. – №12 – 91 с.

11. Спецсимволы в СМС ОС Android [Электронный ресурс]. – Режим доступа: <http://stackoverflow.com/questions/6903852/special-characters-in-android-sms>.

Поступила в редколлегию 27.02.2015

Рецензент: д-р техн. наук, проф. В.В. Баранник, Харьковский университет Воздушных Сил им. И. Кожедуба, Харьков.

СПОСІБ ПЕРЕКODУВАННЯ KOOPДИНАТ МІСЦЕПОЛОЖЕННЯ ДЛЯ ПЕРЕДАЧІ В SMS-ПОВІДОМЛЕННЯХ

Д.Е. Двухглавов, М.Ю. Струтинський

Розглядається задача зменшення обсягу, займаного координатами місцеположення, актуальність якої зумовлена сучасними тенденціями використання засобів стільникового зв'язку. Розглянуто особливості представлення інформації про місцеположення об'єктів та запропоновано рішення поставленої задачі на основі перетворення координат в набір символів кодування UTF. Розроблено алгоритм перекодування координат місцеположення, поданий приклад вирішення задачі із застосуванням розробленого алгоритму та результати аналізу його ефективності, отримані порівнянням результатів програмної реалізації алгоритму з результатами роботи відомих програм стиснення даних.

Ключові слова: координати місцеположення, представлення координат, передача координат розташування в мережах стільникового зв'язку.

THE WAY OF RECODING LOCATION COORDINATES FOR TRANSMISSION IN SMS-MESSAGES

D.E. Dvuhglavov, M.Yu. Strutynskiy

Considered the problem of representation of location coordinates which is relevant in modern trends in mobile telephony. Reviewed details of object location's representation and suggested the problem's solution based on transforming location coordinates into set of characters from UTF encoding. Developed algorithm for location coordinates recoding, introduced example of the problem and its solution using developed algorithm. Presented results of efficiency analysis of developed algorithm, which were composed through comparing results of program using developed algorithm with results of other known data compression programs.

Keywords: location coordinates, representation of location coordinates, location coordinates transmission in mobile networks.