

Формат файлов MRG для компактного представления и высокоскоростной декомпрессии матриц высот большого объёма

А. Е. ХМЕЛЬНОВ

Институт динамики систем и теории управления СО РАН, Иркутск, Россия

e-mail: hmelnov@icc.ru

Рассматривается новый формат файлов MRG, предназначенный для хранения больших объёмов целочисленных растровых данных. Разработанный формат поддерживает быструю выборку фрагментов растра с различными уровнями детализации за счёт представления вместе с наиболее детальными данными растров более низкого разрешения. Предложена структура данных, позволяющая не только не увеличить расход дисковой памяти на хранение дополнительных растров более низкого разрешения, но и сократить её. Для упаковки данных разработан оригинальный алгоритм сжатия разностных целочисленных последовательностей, позволяющий как увеличить коэффициент сжатия, так и сократить время, необходимое для упаковки данных (по сравнению с использованием популярной библиотеки ZLib). Для чтения информации из файлов MRG реализована динамическая библиотека, что даёт возможность применять представленные в них данные в сторонних программах. Программное обеспечение для чтения файлов MRG способно эффективно работать при использовании ограниченного объёма оперативной памяти (от 16 Мб).

Ключевые слова: формат данных, растровые данные большого объёма, сжатие без потери информации.

Введение

В настоящее время увеличился объём некоммерческих информационных ресурсов, содержащих достаточно подробные сведения о рельефе для различных территорий Земного шара. Одним из наиболее востребованных из таких ресурсов являются данные SRTM (Shuttle Radar Topography Mission) [1], содержащие в том числе сведения о рельефе значительной части территории России на сетке с шагом 3" (< 92.6 м). В работе [2] проведено сравнение точности данных SRTM с другими моделями рельефа (GTOPO30, цифровые топографические карты различных масштабов) на примере различных территорий Латинской Америки и показано, что:

- SRTM существенно лучше GTOPO30;
- средняя ошибка SRTM в определении высоты относительно измерений GPS составляет 8 м против 20 м на топографической модели рельефа;
- данные SRTM превосходят по точности топографические карты начиная с масштаба 1 : 50 000.

Если предположить, что российские карты не могут быть на порядок точнее латиноамериканских, то данные SRTM должны содержать значительно более точную информацию о рельефе, чем, например, векторные электронные карты масштаба 1 : 200 000. Таким образом, данные SRTM являются ценным информационным ресурсом, который может быть использован для решения различных практических задач.

Данные SRTM с разрешением 3'' распространяются блоками 1° по широте $\times 1^\circ$ по долготе, содержащими 1201×1201 точек (плюс одна точка из-за включения границ между блоками в оба блока). Высоты задаются 2-байтовыми целыми числами и измеряются в метрах относительно эллипсоида WGS-84. Файлы блоков не содержат ничего, кроме таблицы высот, и занимают одинаковый объём $1201 \times 1201 \times 2b = 2884802b$.

В работе [1] описывается приём, позволяющий открыть файл SRTM в ArcView: необходимо его распаковать, изменить расширение на `bil` и подготовить одноимённый текстовый файл с расширением `hdr`, содержащий описание данных. Однако для того чтобы, например, увидеть весь Байкал и территории вокруг него, необходимо открыть 110 блоков в диапазоне $50^\circ - 60^\circ$ по широте $\times 100^\circ - 109^\circ$ по долготе (далее будем использовать эти данные для тестирования программного обеспечения). Выполнение таких действий вручную для 110 блоков не представляется разумным, и кроме того в дальнейшем проявятся сложности с их совместной обработкой в ArcView. Это определяет потребность в создании программного обеспечения, позволяющего эффективно работать с информацией большого количества блоков. В рамках такого обеспечения должно поддерживаться быстрое отображение на экране любого фрагмента растра, а также вырезание фрагментов растра с различным разрешением, что позволило бы обрабатывать информацию по интересующей территории, не превышая возможности используемого для этого программного обеспечения. Поскольку данные большого числа блоков в упакованном виде занимают много места, необходимо для их хранения применять алгоритмы сжатия информации. Для растровых изображений, например космоснимков, допустимо использовать алгоритмы сжатия с потерями, такие как JPEG, в том случае, если их применение не слишком меняет визуальное восприятие изображения человеком. В отличие от такого рода снимков для упаковки матриц высот необходимо применять алгоритмы сжатия без потерь, поскольку эти данные в первую очередь являются исходными для различных алгоритмов их автоматической обработки и анализа.

В процессе решения поставленной задачи были разработаны формат MRG (Multi Resolution Grid — матрица уровней разрешения) и соответствующее ему программное обеспечение. Далее будут рассмотрены принципы организации файла в формате MRG, его общая структура, разработанные алгоритмы сжатия без потерь растра высот и возможности созданного программного обеспечения.

1. Общие принципы организации файла в формате MRG

Для того чтобы быстро прочитать в память фрагмент растра с низким разрешением, содержащий информацию по большой территории, в файле MRG предусмотрено хранение не только исходного, но и менее детальных растров (отсюда происходит название формата). Необходимость в получении фрагментов с меньшим разрешением возникает, например, при просмотре на экране больших частей растра. Вычисление растров низкого разрешения непосредственно по исходным данным может быть достаточно трудоёмким для фрагментов, отображающих большую территорию.

В формате MRG разрешение растра при переходе на следующий уровень детализации (будем называть его более высоким) падает вдвое. Иначе говоря, значение каждого пикселя более высокого уровня является результатом усреднения значений соответствующих ему четырёх пикселей более низкого уровня. Каждый уровень разбивается на квадратные блоки. Разбиение сплошного растра на блоки позволяет более эффективно вырезать из него фрагменты: для получения фрагмента необходимо загрузить в оперативную память лишь те блоки, которые он затрагивает. Число уровней детализации определяется габаритами растра: усреднение продолжается до тех пор, пока весь растр не начнёт помещаться в один блок.

При тестировании алгоритмов применялись блоки с размером 400×400 пикселей. Такой размер блока был выбран исходя из типичных габаритов окна отображения карты на экране: для получения изображения при любом разрешении потребуется использовать информацию около десятка блоков, что вполне приемлемо. Кроме того, учитывалась кратность размеру градусного блока данных SRTM (1200). Вообще в качестве размера блока может быть задано любое чётное число. Чётность размера блока важна для упрощения алгоритмов генерации блоков более низкого разрешения: при этом каждый пиксель вычисляется по значениям пикселей только одного блока более низкого уровня, что позволяет обрабатывать эти блоки в любом порядке.

2. Структура файла

Файл в формате MRG состоит из следующих структур данных:

- заголовок;
- список уровней детализации;
- таблица описаний блоков для каждого уровня;
- упакованные данные для каждого блока;
- список свободных фрагментов файла.

Заголовок содержит информацию об общих размерах растра в пикселях, о его географических координатах, размере блока и числе уровней детализации растра. В нём также представлена информация об используемом способе сжатия и о смещении в файле начала списка свободных фрагментов.

Список уровней детализации содержит для каждого уровня информацию о числе строк и столбцов в таблице блоков данного уровня, а также о ширине блоков последнего столбца таблицы и высоте блоков последней строки (крайние блоки могут иметь меньшие размеры, поскольку размер растра может быть не кратен стороне блока (рис. 1)).

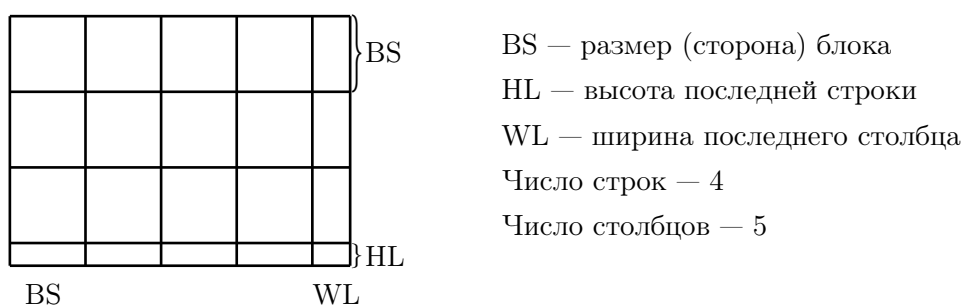


Рис. 1. Информация о разбиении растра уровня детализации на блоки

Таблица описаний блоков содержит по одной записи для каждого блока (всего число строк \times число столбцов элементов). В каждой записи содержатся сведения о размещении информации блока в файле (смещение и размер), а также флаги, отражающие состояние обработки блока и признак использования разностного кодирования для хранения данных (см. далее).

3. Разностное кодирование содержимого блоков

Рассмотренный способ представления информации об уровнях детализации растра вряд ли существенно отличается от того, который используется, например, в программе GoogleEarth [3]. Однако файлы формата MRG предназначены для локального представления всей информации, в силу чего в них возможна реализация более эффективного её хранения.

Поскольку в файле явно представлены блоки растров более низкого разрешения, для хранения этих блоков тратится дополнительное место. Однако блоки низкого разрешения могут быть использованы не только для быстрого получения менее детальных растров, но и для частичного восстановления блоков с более высоким разрешением при помощи различных алгоритмов интерполяции. Полученная в результате интерполяции аппроксимация содержимого блока, естественно, будет отличаться от истинного содержимого этого блока, однако в большинстве точек отличия будут не слишком велики, поскольку рельеф в основном изменяется достаточно плавно.

В реализации формата MRG для аппроксимации значений более низкого уровня используется линейная интерполяция. При этом считаем, что пиксели задаются своими центрами, поэтому (рис. 2) значения четырёх пикселей A, B, C, D уровня $L + 1$, влияющие на аппроксимацию значения пикселей P' уровня L , складываются со следующими коэффициентами, зависящими от положения P' относительно A, B, C, D :

$$P' = (9 * A + 3 * B + 3 * C + D) / 16.$$

Таким образом, при наличии блока более низкого разрешения для блоков, из которых он образован, можно вычислить разностные растры (разность между интерполированными и истинными данными). Разностный растр будет проще исходного, так как его элементы в основном принимают значения, близкие к нулю, а многие равны нулю (на тех участках, где высота изменяется линейно). Поэтому большинство алгоритмов сжатия будут давать значительно лучшие результаты на разностных данных. В результате

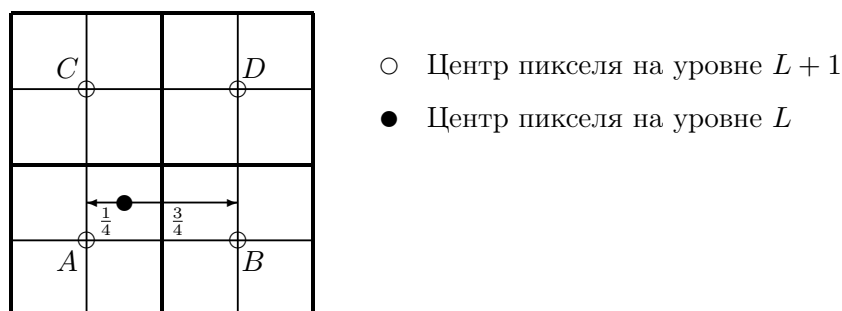


Рис. 2. Линейная интерполяция на более низкий уровень разрешения

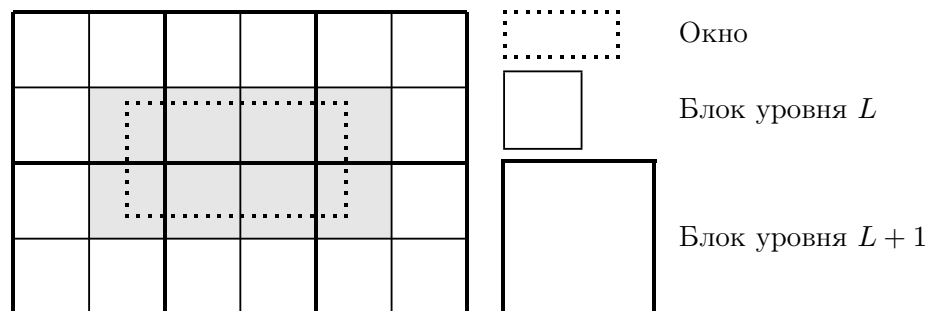


Рис. 3. Блоки, в худшем случае затрагиваемые окном на двух уровнях детализации

хранение промежуточных блоков может привести к сокращению суммарного расхода дисковой памяти (см. таблицу на с. 69).

Необходимо отметить, что хранение разностных растров для восстановления исходных значений пикселей блока требует загрузки в оперативную память не только самого блока, но и всех блоков более высоких уровней, которые его содержат. Если фрагмент растра, для которого необходимо получить данные, затрагивает $M \times N$ блоков на уровне L , то в худшем случае на более высоком уровне $L + 1$ он затрагивает $(M \bmod 2 + 1) \times (N \bmod 2 + 1)$ блоков (рис. 3). В силу этого при выборке большого количества блоков (для которого наличием слагаемых $+1$ в формуле можно пренебречь) с уровня L со следующего уровня $L + 1$ необходимо загрузить почти в 4 раза меньше блоков, что не слишком увеличивает объём работы (в целом приблизительно на треть). Если же с уровня L выбирается небольшое число блоков (для которого слагаемые $+1$ являются существенными), то и с уровня $L + 1$ будет загружено меньше или в худшем случае (при $M = N = 2$) столько же, т. е. тоже немного, блоков. Таким образом, необходимость в загрузке блоков верхних уровней при использовании разностных растров не приводит к существенному повышению трудоёмкости и требований относительно памяти алгоритма считывания блока даже в том случае, когда все используемые блоки ещё не считаны в память. При этом после выполнения нескольких оконных запросов блоки более высоких уровней скорее всего будут уже содержаться в памяти.

Функция распределения значений разностных растров имеет свои особенности, которые могут быть использованы для разработки более эффективных алгоритмов сжатия, учитывающих специфику такого рода данных. В результате разработки автором алгоритма сжатия BCRL (Bit Count Run Length — кодирование разбиением на интервалы постоянной глубины) удалось дополнительно повысить степень сжатия данных, сохраняемых в файлах MRG.

4. Работа в ограниченном объёме памяти

При сохранении в формате MRG данных по достаточно большой территории объём этих данных даже в упакованном виде начинает превышать объём всей оперативной памяти современных компьютеров. Кроме того, при считывании блока в память его содержимое распаковывается, после чего для разностных растров выполняется сложение с интерполированными данными, что позволяет в дальнейшем быстрее получать информацию по задействованным в текущих операциях блокам. Поэтому нельзя рассчитывать на возможность одновременной загрузки в оперативную память всего содержимого файла.

В программном обеспечении для чтения и подготовки данных в формате MRG реализованы механизмы, позволяющие работать в ограниченном объеме оперативной памяти. При создании экземпляра класса для работы с растром задается предельный объем памяти, используемой для хранения распакованных блоков этого раstra. Ограничение на объем используемой памяти соблюдается не совсем жестко: только после превышения допустимого объема запускается механизм освобождения ненужных блоков. Иначе говоря, при чтении предельный объем памяти может быть в какой-то момент превышен на объем данных одного блока. При создании нового раstra пока не завершённые блоки не могут быть освобождены, поэтому расход памяти может превысить предел ещё в большей степени.

Механизм освобождения памяти оценивает, насколько в дальнейшем может быть полезен каждый загруженный в память блок. Для определения степени полезности блока на уровне L учитывается его расстояние, измеряемое в количестве блоков, до фрагмента таблицы блоков уровня L , с которым в настоящее время выполняется работа. В первую очередь освобождаются наиболее далёкие блоки независимо от того, на каком уровне они находятся. Освобождение продолжается до тех пор, пока объем используемой памяти не сократится до некоторой доли от максимально допустимого или пока не закончатся незащищённые блоки.

При используемой для тестирования стороне блока величиной 400 пикселей программное обеспечение успешно соблюдает ограничение 16 Мб при чтении файлов MRG.

5. Алгоритм сжатия BCRL

Хотя проблемная область сжатия данных очень хорошо изучена [4–6], исследования в этой сфере не прекращаются, поскольку рост объемов накапливаемых данных едва ли не опережает рост объемов носителей информации. Различные алгоритмы сжатия информации основаны на использовании тех или иных особенностей сжимаемых данных. Алгоритмы сжатия без потерь в основном используют статистические и/или словарные методы [5]. Статистические методы (например, кодирование Хаффмана или арифметическое кодирование) эффективны для данных, имеющих неравномерность функции распределения значений, и позволяют представить более вероятные значения меньшим числом битов. Словарные методы (например, алгоритм LZW) эффективны при наличии в сжимаемых данных повторяющихся подпоследовательностей.

Особенностью разностных растров является то, что пиксели этих растров принимают значения преимущественно вблизи нуля, поэтому большинство их значений может быть представлено гораздо меньшим, чем в исходных данных, количеством бит (битовой глубиной). Если бы удалось представить каждое значение с минимально необходимой для него глубиной, то занимаемый данными объем мог бы существенно сократиться. Однако при этом пришлось бы ещё как-то закодировать служебную информацию о глубине каждого значения. Информацию о глубине значений также можно рассматривать в качестве специального раstra. Предлагаемый алгоритм BCRL предназначен для быстрого сжатия без потерь целочисленных данных, значения которых распределены преимущественно вблизи нуля, и основан на том, что растр исходных значений и растр их глубины не являются независимыми и в некоторых случаях глубину значения можно увеличить с той целью, чтобы последовательность глубин стала проще.

Алгоритм сжатия оптимизирует разбиение последовательности на интервалы постоянной глубины с тем, чтобы сократить суммарный объем, занимаемый данными вместе

со служебной информацией. Для оптимизации используется динамическое программирование: после добавления каждого значения пересчитывается таблица стоимости лучшего разбиения в зависимости от глубины, используемой для представления добавленного значения. Размер такой таблицы равен глубине исходных данных, поэтому на каждое значение тратится время, ограниченное константой, и вычислительная сложность всего алгоритма линейна. Конкретный алгоритм получается после выбора способа кодирования заголовков интервалов, содержащих информацию о длине и глубине каждого интервала. Алгоритм будет находить истинно оптимальное разбиение лишь в том случае, когда размер заголовка не зависит от длины интервала, иначе для получения точного решения пришлось бы учитывать не только глубину, но и длину последнего интервала. Вычислительный эксперимент показал, что субоптимальное решение, пренебрегающее зависимостью размера заголовка от длины, отличается от оптимального лишь на доли процента. Кроме того, предложен способ приближения к оптимальному решению за счёт увеличения размеров таблиц.

Что касается алгоритма декодирования данных, упакованных алгоритмом BCRL, то он является максимально простым: необходимо читать из битового потока заголовки интервала и следующее за ним указанное в этом заголовке число значений с указанным числом бит на значение до тех пор, пока не будет прочитан необходимый объём данных.

Более подробное рассмотрение алгоритма BCRL представляет собой тему отдельной статьи.

6. Оценка производительности алгоритма сжатия

Для оценки производительности предложенного алгоритма сжатия BCRL при кодировании разностных растров выполнено его тестирование на 110 блоках SRTM, описывающих территорию вокруг Байкала. Для сравнения была использована упаковка средствами библиотеки ZLib [7] с различными значениями уровня сжатия. Нижеприведённая таблица содержит описание результатов выполненного тестирования.

В первых двух строках таблицы приведена информация об исходных данных, остальные строки относятся к файлу MRG, полученному из этих данных с использованием различных алгоритмов сжатия. Отметим, что файлы SRTM распространяются в архивах ZIP, которые созданы со степенью сжатия, превышающей максимальную для имеющейся у автора версии архиватора PKZIP. Например, архив N50E100.hgt.zip имеет размер 1 376 212 байт, а после упаковки его содержимого при помощи архиватора PKZIPC получаем 1 418 255 байт. Этот же файл после максимального сжатия средствами библиотеки ZLib, используемой для сравнения, занимает 1 430 423 байт. Таким образом, исходные данные были сжаты очень хорошей версией архиватора ZIP.

Тем не менее при помощи разностного кодирования блоков эти результаты превосходит даже ZLib с минимальным для библиотеки уровнем сжатия Fastest. Из таблицы также видно, что алгоритм BCRL улучшил результаты ZLib с максимальным уровнем сжатия ещё почти на 5 % от объёма после сжатия ZLib'ом.

По времени работы алгоритм BCRL уступает только результатам ZLib с уровнем сжатия Fastest, однако при этом по степени сжатия BCRL существенно превосходит ZLib в этом режиме (на 22 % от объёма упакованных ZLib данных).

Попробуем неформально объяснить, в чём состоит преимущество алгоритма BCRL при сжатии разностных растров. В алгоритме deflate, реализованном в ZLib, словарные методы используются в сочетании со статистическими. Словарные методы не слишком

Информация о разбиении раstra уровня детализации на блоки

Сжатие	Размер, байт	Время, мин	Время, с	Коэффициент сжатия, %
Исходные данные без сжатия	317 328 220	—	—	100.00
Исходные данные в ZIP	113 427 015	—	—	35.74
MRG с ZLib fastest	111 536 958	01:11	71.516	35.15
MRG с ZLib default	94 608 633	02:44	164.422	29.81
MRG с ZLib max	91 674 463	20:25	1225.687	28.89
MRG с BCRL	87 402 922	02:04	124.750	27.54

хорошо подходят для рассматриваемых данных, поскольку последовательности высот не содержат сколько-нибудь существенных повторений (не считая водных поверхностей). Иными словами, словарные методы рассчитывают на маловероятные для результатов измерения высоты рельефа совпадения. В основном эти методы помогают присвоить более короткие коды двухбайтным целочисленным значениям.

При использовании статистического метода Хаффмана каждому значению присваивается код, позволяющий однозначно определить его длину. Таким образом, можно считать, что длина кода запоминается для каждого значения в отдельности. В любом случае при кодировании по Хаффману невозможно представить значение менее чем одним битом. В алгоритме BCRL длина кода задаётся один раз для всех значений интервала постоянной глубины, причём наиболее вероятные для рассматриваемого вида данных значения так же, как и в случае метода Хаффмана, в основном кодируются более короткими последовательностями битов. Иначе говоря, при использовании BCRL удаётся сэкономить на представлении информации о длине кодов. Важным частным случаем является последовательность из нулевых значений, длина представления которой либо не зависит от длины исходной последовательности, либо зависит от неё логарифмически, т. е. нулевые данные иногда могут быть представлены значительно меньше, чем одним битом на значение.

7. Программное обеспечение

Для создания и просмотра файлов в формате MRG разработана программа MRGView. В настоящее время реализовано создание файлов MRG на основе исходных данных SRTM. В этом процессе необходимо прежде всего определиться с диапазоном географических координат, который будет содержаться в файле, после чего в файл могут сразу или по частям импортироваться данные непосредственно из тех архивов ZIP, где они содержатся. Иными словами, допускается добавление недостающих блоков в существующий файл MRG, но не поддерживается изменение диапазона географических координат файла.

Для отображения раstra высот реализована отмывка рельефа с использованием нескольких цветовых палитр. Допускается выбор диапазона высот, соответствующего цветовой палитре и направлению на источник света.

Водные поверхности в данных SRTM не представлены явно, однако отдельно опубликованы данные SWBD [1], содержащие в формате ArcView Shape векторные контуры водных объектов, выделенных при обработке данных SRTM. В текущей версии программы MRGView данные SWBD не используются. Вместо этого реализован режим отображения водных поверхностей, в котором пиксель считается относящимся к вод-

ной поверхности, если его соседи имеют такую же высоту, как и он. В большинстве случаев данная эвристика работает правильно, в силу чего такой режим действует по умолчанию, но при необходимости его можно отключить.

С использованием предложенного формата все данные SRTM по Сибири и Дальнему Востоку помещаются на один компакт-диск. Для подготовки этого файла был использован 921 блок $1^\circ \times 1^\circ$ исходных данных.

8. Динамическая библиотека для доступа к файлам MRG

Для применения файлов MRG в сторонних программах реализована динамическая библиотека MRGLib, позволяющая:

- открыть файл MRG с указанием предельного расхода оперативной памяти;
- получить информацию о диапазоне географических координат, для которых в файле имеются данные;
- получить размер в пикселях самого подробного слоя растра;

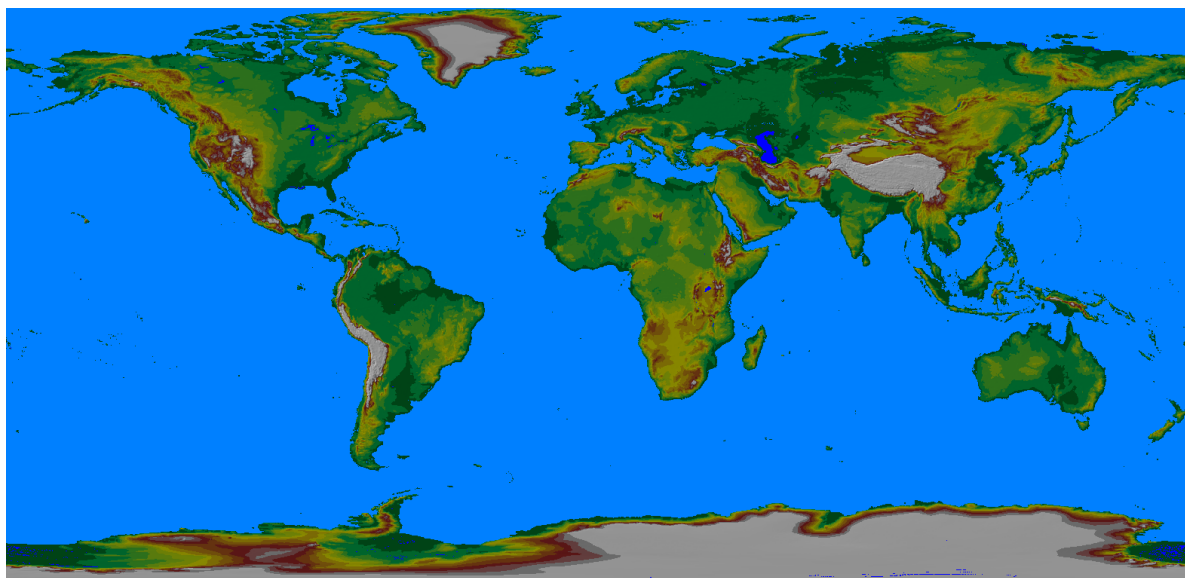


Рис. 4. Все данные SRTM3 занимают 10.8 Гб (11 645 149 616 байт)



Рис. 5. Данные, которые помещаются на один компакт-диск (700 Мб) в формате MRG

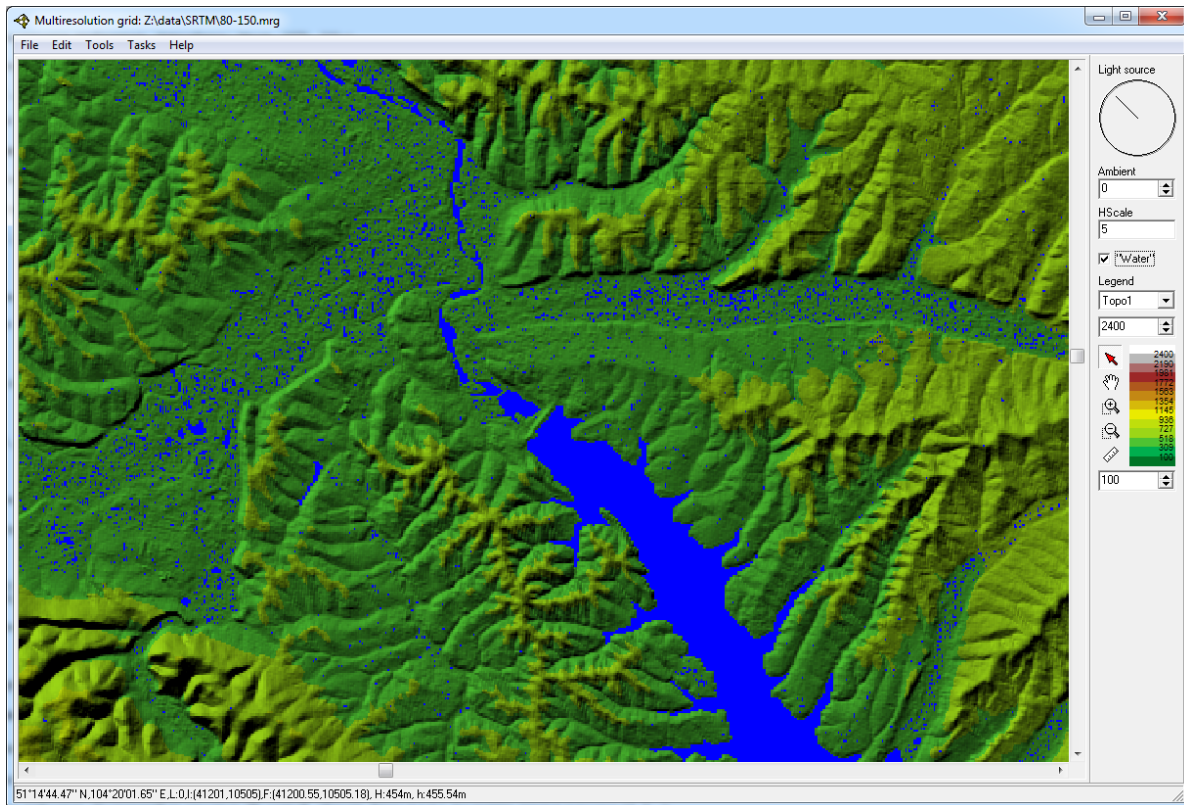


Рис. 6. Разрешение, которое можно получить в любой точке растра SRTM3

- рассчитать уровень детализации растра, необходимый для отображения указанного фрагмента растра нижнего уровня на экране с указанной шириной (при этом также определяется фрагмент растра на вычисленном уровне, соответствующий исходному);
- прочесть фрагмент растра на заданном уровне детализации в заданный буфер;
- получить географические координаты точки растра на уровне 0;
- получить координаты точки растра на уровне 0 по географическим координатам;
- сохранить фрагмент растра на заданном уровне детализации в файл изображения ArcView (*.bil);

Таким образом, с использованием библиотеки MRGLib можно достаточно просто получать информацию о рельефе с необходимой степенью детализации из файлов MRG для решения различных специальных задач. Примеры изображений, характеризующих возможности MRGLib, приведены на рис. 4–6.

Заключение

Рассмотренные в статье формат MRG и программное обеспечение для работы с ним дают возможность организовать более эффективное хранение и использование больших объёмов растровых данных. Применение предложенного подхода позволяет реализовать в дальнейшем централизованное хранилище растровой пространственной информации с быстрым получением необходимых фрагментов данных, а также распространять достаточно большие объёмы данных на сменных носителях информации.

Представленный алгоритм BCRL для сжатия разностных целочисленных последовательностей может быть использован и для хранения других видов данных, имеющих аналогичные особенности. Преимуществами BCRL являются низкая вычислительная сложность, высокая скорость алгоритма сжатия и максимальная простота алгоритма распаковки.

Возможность извлечения произвольных фрагментов данных из файлов MRG в формате BIL позволяет рассматривать такие файлы в качестве источника информации о рельефе, пригодной для проведения исследований средствами популярных ГИС.

Список литературы / References

- [1] **Дубинин М.** Описание и получение данных SRTM. Адрес доступа: <http://gis-lab.info/qa/srtm.html> (дата обращения 23.09.2014).
Dubinin, M. The SRTM data description and sources. Available at: <http://gis-lab.info/qa/srtm.html> (accessed 23.09.2014).
- [2] **Jarvis, A., Rubiano, J., Nelson, A. et al.** Practical use of SRTM data in the tropics — Comparisons with digital elevation models generated from cartographic data. Working document no. 198. Centro Internacional de Agricultura Tropical (CIAT), — Cali, CO. 32 pp. Available at: <http://srtm.csi.cgiar.org/PDF/Jarvis4.pdf>
- [3] Google Earth. Available at: <http://www.google.com/earth/> (accessed 23.09.2014).
- [4] **Ватолин Д., Ратушняк А., Смирнов М., Юкин В.** Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. М.: ДИАЛОГ-МИФИ, 2003.
Vatolin, D., Ratushnyak, A., Smirnov, M., Yookin, V. Data compression methods. The structure of archivers, image and video compression. Moscow: Dialog-MEPhI, 2003.
- [5] **Ватолин Д.** Всё о сжатии данных, изображений и видео. Адрес доступа: <http://www.compression.ru> (дата обращения 23.09.2014).
Vatolin, D. Everything about the data compression. Available at: <http://www.compression.ru> (accessed 23.09.2014).
- [6] **Salomon, D.** A guide to data compression methods. Springer-Verlag, 2002.
- [7] **Gailly, J.-L., Adler, M.** ZLib home site. Available at: <http://www.zlib.net> (accessed 23.09.2014).

*Поступила в редакцию 29 сентября 2014 г.,
с доработки — 10 ноября 2014 г.*

The MRG file format for compact representation and fast decompression of large digital elevation models

HMELOV, ALEXEI E.

Institute for System Dynamics and Control Theory SB RAS, Irkutsk, 664033, Russia

Corresponding author: Hmelnov, Alexei E., e-mail: hmelnov@icc.ru

Purpose. A new file format MRG (MultiResolution Grid), which is designed for lossless storage and fast decompression of large integer raster images, is considered. The format developed allows to quickly fetch raster fragments of various levels of details on account of storing both the most detailed data and the data of lower resolution.

Design\methodology\approach. The data of higher resolution are represented by storing the difference between actual data and their approximation from the corresponding lower resolution image block. For compression of differential integer sequences an original algorithm BCRL (Bit Count Run Length encoding) has been developed. The algorithm allows both to increase the data compression ratio and to reduce the data compression and especially decompression time (in comparison with the popular library ZLib). The BCRL decompression algorithm can be expressed by few lines of code: it just reads intervals of values of constant bit depths. The compression algorithm finds the close to optimal subdivision of the sequence of integer values into such intervals using the dynamic programming approach.

Findings. The proposed data structure allows not only lower usage of additional memory for storing the additional low resolution raster images, but conversely, proposes how to decrease the overall memory use. The BCRL data compression algorithm can efficiently utilize the features of differential sequences.

Originality\value. The file format developed is completely original and based on different principles. Compared to the known file formats, it can be used for lossless compression of integer raster images, such as MrSID, ECW, JPEG 2000. The proposed compression algorithm is significantly simpler, but still very efficient. A dynamic library has been implemented for reading data from the MRG files by external programs. The developed software can effectively work using a limited amount of RAM (down to 16 Mb).

Keywords: data format, big raster data, lossless compression.

Received 29 September 2014

Received in revised form 10 November 2014