

# Высокопроизводительный алгоритм Band Interleave Conversion для данных гиперспектральных снимков сенсора EO-1 Hyperion

В.П. Потапов, С.Е. Попов

**Аннотация.** В работе представлен высокопроизводительный алгоритм преобразования данных спектральных каналов (Band Interleave Conversion) сенсора EO-1 Hyperion, предусматривающий возможность запуска его на многопроцессорных платформах в мультипоточном режиме и обеспечивающий эффективное выполнение на низкопроизводительных системах ввода/вывода. Предложена реализация порционного считывания данных снимка в оперативную память, с последующим размещением значений (Digital Number) в целевых буферных массивах в несколько потоков, рассчитанных по количеству спектральных каналов или по количеству линий снимка. В работе представлено расширение для программного продукта Exelis ENVI, реализующее разработанные алгоритмы, на базе технологии GUI-WIDGETS в интеграции с пакетами Java SwingX.

**Ключевые слова:** Преобразование порядка следования данных, спектральные каналы, многопоточность, java, IDL-Bridge.

## Введение

Данные зондирования земли поступают в виде изображений, как правило, в цифровой форме, обработка ведется на ЭВМ, поэтому технологии дистанционного зондирования тесно связаны с цифровой обработкой изображений. Предварительная обработка данных дистанционного зондирования заключается в геометрической коррекции спутниковых изображений, радиометрической и атмосферной коррекции, восстановлении пропущенных пикселей и улучшения изображений путем изменения контраста. Одним из наиболее важных этапов предварительной обработки космических снимков является атмосферная коррекция. В настоящее время, большая часть данных попадающих в руки конечного пользователя уже атмосферно скорректирована. Однако в связи с тем, что процесс коррекции

больших объемов данных полностью автоматизирован, используемые для коррекции алгоритмы рассчитываются исходя из усредненных показателей и не учитывают особенностей получения каждого конкретного снимка, а также возможность оптимизации вычислительного процесса для каждого конкретного программного продукта или отдельных сторонних специализированных модулей.

Существует ряд работ, направленных на оптимизацию алгоритмов пре- и пост-обработки мульти- и гиперспектральных изображений, а также сравнение разрешающей способности алгоритмов. Так в работе [1] представлен алгоритм Multi-Scale Retinex (MSR), обеспечивающий сжатие динамического диапазона, снижение зависимости от условий освещения, а также улучшение пространственного разрешения мультиспектральных данных. В частно-

сти, результаты моделирования показывают, что, несмотря на степень снижения качества изображения из-за изменения атмосферного излучения, ошибки классификации могут быть существенно уменьшены путем предварительной обработки данных изображения с помощью MSR. В работе [2] обсуждается сравнение алгоритмов атмосферной коррекции Atmospheric CORrection Now (ACORN), the Fast Line-of-sight Atmospheric Analysis of Spectral Hypercubes (FLAASH), and ATmospheric CORrection (ATCOR 2-3) на примере гиперспектральных данных сенсора EO-1 Hyperion. Рассматриваются процедуры кросс-корреляции снимков, приводится сравнение особенностей поглощения определенных длин волн. В зависимости от различных диапазонов длин волн алгоритм ACORN показывает наилучшую корректирующую способность на снимках EO-1 Hyperion в задачах определения литологии и минералогического состава природных земельных геоматериалов. В работе [3] описываются результаты использования моделей алгоритмов ATREM, HATCH, ACORN и FLAASH, приводятся оценки коэффициентов отражения на верхней границе атмосферы для водяного пара. Показано, что существуют значительные различия между моделями атмосферной коррекции протестированных алгоритмов. Отличия моделей равны или больше, чем ошибки, вносимые калибровкой длин волн или шумами сенсоров. Этот результат не дает возможности узнать, какие модели работают лучше. В обоих случаях (по данным AVIRIS и Hyperion) графики коэффициентов отражения, полученные из сцен с наиболее высокими показателями для водяного пара, содержали значительное количество артефактов и имели отклонение от сглаженной кривой отражения. Обнаружено, что использование алгоритма HATCH-2D с опцией «column-by-column wavelength calibration», позволило получить более «сглаженные» спектры отражения, чем при использовании метода «average spectral calibration» для всего массива данных. Однако недостатком данного подхода явилось значительное увеличение времени вычислений.

Не менее важными являются вопросы, связанные с вычислительной производительностью алгоритмов пре- и пост-обработки спектральных

изображений. В работах [4, 5] рассматривается программный прототип системы автоматизированного, высокоточного алгоритма с малой задержкой для расчета атмосферной коррекции на основе C++ для модуля FLAASH. Предложенный подход заключается в замене расчетов моделей переноса радиации модуля MODTRAN на предварительно рассчитанные данные, приведенные в справочных таблицах, используя для этого параллельную обработку на многоядерных/многопроцессорных вычислительных системах.

В настоящее время на рынке программных продуктов есть готовые решения для визуализации и обработки данных дистанционного зондирования Земли (ДЗЗ), которые включают в себя набор инструментов для проведения полного цикла обработки данных от ортотрансформирования и пространственной привязки изображения до получения необходимой информации и интеграции ее с данными ГИС. Однако программная реализация алгоритмов, интегрированных в такой софт, не оптимизирована, требует достаточно много машинных ресурсов и длительного времени на обработку изображений. Например, модуль Band Interleave с опцией подготовки данных для атмосферной коррекции алгоритмами FLAASH в программном комплексе ENVI для гиперспектрального снимка (7.7 км×185 км) сенсора EO-1 составляет порядка двух с половиной часов. При этом некорректно используются ресурсы файловой подсистемы из-за неоптимизированного порядка считывания файлов данных спектральных каналов. Конечно, стоит сказать, что в том же программном продукте Exelis ENVI есть встроенная технология мультипотокности [6], но она представлена по принципу «hard coded», т.е. не может настраиваться и управляться разработчиком, а самое главное, она реализована только для простейших математических операций и создания массивов, что не позволяет в полной мере применять технологию объектно-ориентированного подхода Java.

В данной работе описывается разработка высокопроизводительного алгоритма для предварительной обработки (подготовки) гиперспектральных данных с последующей их передачей на вход модулю атмосферной коррекции программного продукта Exelis ENVI.

### 1. Постановка задачи

Разработать алгоритм преобразование порядка следования данных отдельных каналов, их программную реализацию в виде расширений для программного комплекса ENVI с использованием технологии IDL-Java Bridge. Программный функционал алгоритмов должен предусматривать возможность запуска их на многопроцессорных платформах в мультипоточковом режиме и обеспечивать эффективное выполнение на низкопроизводительных системах ввода/вывода, с количеством оперативной памяти до 2 Гб.

### 2. Используемые данные и программные продукты

Для тестирования разрабатываемых алгоритмов использованы данные гиперспектральных изображений, полученные с ресурса EarthExplorer [7] Геологической службы США. В работе рассматривались изображения сенсора EO-1 Hyperion с полосами съемки 7.7 км × 42 км и 7.7 км × 185 км соответственно. Гиперспектральный сенсор Hyperion предназначен для получения цифровых изображений земной поверхности с пространственным разрешением до 30 м в 242 спектральных диапазонах видимой и инфракрасной области.

Гиперспектральные снимки сенсора EO-1 Hyperion уровня L1T представляют собой набор из 242 файлов, соответствующих спектральным каналам в графическом формате GeoTIFF, а также файл метаданных с расширением (.TXT или .MET). Файлы имеют нативный формат именования, соответствующий схеме **EO1spprrrYYYYDDDDXXXML\_BBBB\_TTT.TIF** [8]. Размер снимка определяется количеством линий (параметр PRODUCT\_LINES) и количеством пикселей в одной линии (параметр PRODUCT\_SAMPLES). Эта информация размещается в файле метаописания [7].

Каждый TIFF-файл спектрального канала содержит теги с информацией о типе, количестве и размере данных в нем размещенных. В работе были использованы стандартные теги TIFF-формата:

- StripOffsets – содержит массив значений смещений (по одному на линии), которые ука-

зывают позицию первого байта каждой линии в файле TIFF. Первый элемент массива указывает смещение первой линии, второй — смещение второй линии и т.д.

- StripByteCount - описывает массив значений, указывающих размер каждой линии в байтах.

Гиперспектральные снимки, загружаемые с ресурса EarthExplorer [7] имеют формат преобразование порядка следования данных каналов BSQ (band sequential) (Рис. 1). Данный формат хранит информацию о каждом спектральном канале изображения за раз. Другими словами, данные для всех пикселей первого канала расположены первыми (с нулевым смещением), затем идут данные всех пикселей второго канала со смещением (количество каналов)\*(количество линий)\*(количество пикселей в линии)\*2 и так далее по количеству спектральных каналов. Тип данных для каждого значения пиксела – Short, размеров 16 бит.

Высокопроизводительные расчетные алгоритмы разработаны с применением объектно-ориентированного языка Java, расширения для программного комплекса Exelis ENVI созданы на базе языка IDL, связующим интерфейсом послужила технология Java-IDL Bridge.

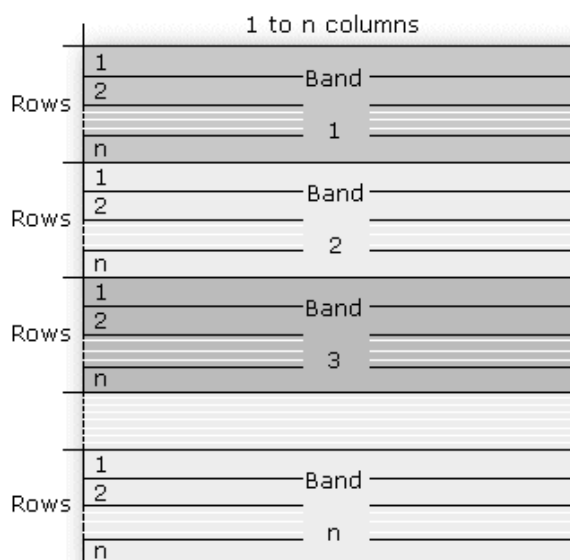


Рис. 1. Схематичное изображение данных в формате BSQ

### 3. Алгоритм и его программная реализация

Основной акцент в работе ставился на создание алгоритма препроцессинга (преобразование порядка следования данных отдельных каналов Band Interleave Conversion) полного снимка с данными уровня L1T без выделения на нем областей интересов (ROI) за относительно небольшое время выполнения. Как показал опыт использования программных решений (в частности ENVI), основным «узким» местом в производительности их алгоритмов является операции ввода/вывода файловой подсистемы, а конкретнее – компромисс между использованием оперативной памяти и временем загрузки исходных данных в нее.

Согласно [9] исходное изображение для модуля атмосферной коррекции FLAASH должно иметь значение светимости для каждого пиксела спектрального канала и быть в формате преобразования BIL (band-interleaved-by-line) или BIP (band-interleaved-by-pixel).

В формате BIL данные хранят значение для каждой линии снимка в порядке следования «канал за каналом» соответственно для каждой строки (Рис. 2). Формат BIP аналогичен BIL за исключением того, что данные в порядке следования «канал за каналом» хранятся для каждого пиксела (Рис. 3) [10].

Поэтому, учитывая схему расположения данных в исходном изображении, их запись в файл назначения для конвертируемого формата, битность и тип данных, процесс конвертации форматов можно разбить на 3 этапа с целью оптимизации операции ввода/вывода, размещения и работы с массивами в оперативной памяти. Далее для краткости соответствующую конвертацию будем обозначать как BSQ→BIL или BSQ→BIP.

Первый этап аналогичен для BSQ→BIL и BSQ→BIP. Здесь осуществляется проверка общей длины снимка в байтах с целью использования порционного считывания данных из файлов. Пороговым значением для порции установлено 1Гб. Выбрано оно опытным путем, исходя из минимальных требований необходимых для исполнения программного кода виртуальной машины Java, максимальной длины массива в Java и минимизации использования

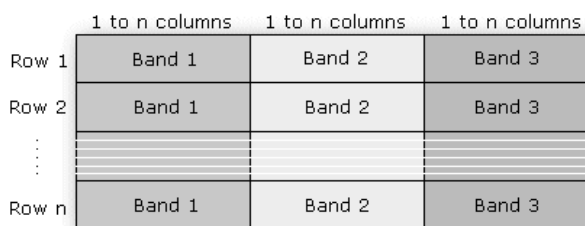


Рис. 2. Схематичное изображение данных в формате BIL

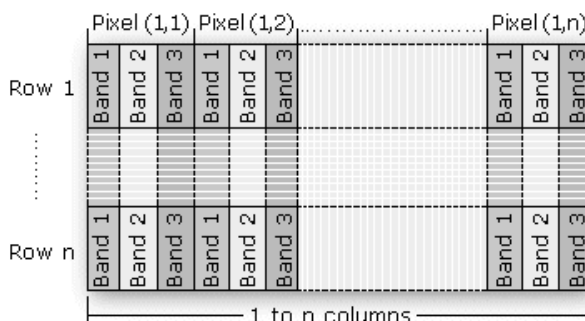


Рис. 3. Схематичное изображение данных в формате BIP

файла подкачки операционной системы Windows. На этом этапе вводятся понятия: количество линий снимка, размещённых в «heap size» (1), целое число таких размещений для общего количества линий (2) и остаток линий с учетом полного покрытия снимка (2).

$$heapL = \frac{1024^3}{byteL \cdot n}, \tag{1}$$

где  $heapL \in \mathbb{Z}$  – число линий в «heap size» (в программном коде задано как тип **Integer**, поэтому результат деления – всегда целое число),  $byteL \in \mathbb{Z}$  – длина одной линии в байтах,  $n \in \mathbb{Z}$  – количество спектральных каналов.

$$numL = \frac{heapL}{numL} \cdot heapL + remainL, \tag{2}$$

где  $numL \in \mathbb{Z}$  – число линий снимка для каждого из спектрального канала,  $\frac{heapL}{numL} \in \mathbb{Z}$  – число размещений для общего количества линий,  $remainL \in \mathbb{Z}$  – остаток линий с учетом полного покрытия снимка.

Второй этап также аналогичен для 2 типов конвертаций BSQ→BIL и BSQ→BIP. На данном этапе выполняются самые длительные по времени операции. Здесь происходит порционное считывание данных из файлов спектральных каналов в массив типа BufferArray, содержащий

данные  $i$ -ой порции (3) и остаток данных (4) для каждого из спектральных каналов.

$$buffer_{A_k}^{byte_i} = \left[ offset_L + heap_L \cdot byte_L \cdot i, offset_L + heap_L \cdot byte_L \cdot (i + 1) \right)_k, \quad (3)$$

и

$$\left[ \begin{array}{c} buffer_{A_k}^{byte_{remain}} = \\ offset_L + heap_L \cdot byte_L \cdot \frac{heap_L}{num_L} \\ offset_L + heap_L \cdot byte_L \cdot \frac{heap_L}{num_L} + remain_L \end{array} \right]_k \quad (4)$$

где  $buffer_{A_k}^{byte_i}$  - элемент массива порции данных и  $buffer_{A_k}^{byte_{remain}}$  - остаток данных  $i$ -ого спектрального канала,  $offset_L \in \mathbb{Z}$  - значение тега StripOffset TIFF-файла,  $k = 0 \dots n - 1$ ,  $i = 0 \dots \frac{heap_L}{num_L} - 1$ ,  $k, i \in \mathbb{Z}$ . В формулах (3) и (4) в виде отрезков с границами показаны начальное и конечное смещение байт в TIFF файле  $i$ -ой порции данных или остатка данных. Далее будем рассматривать  $buffer_{A_k}^{byte_i}$  и  $buffer_{A_k}^{byte_{remain}}$ , как массивы длиной  $heap_L \cdot byte_L$  и  $remain_L \cdot byte_L$  соответственно.

На третьем этапе происходит размещение данных  $buffer_{A_k}^{byte}$  в массив типа BufferArray согласно выбранному типу конвертации BSQ→BIL (7-10) или BSQ→BIP (11-14) с последующим сохранением его на диск. Здесь применяется мульти-поточковый подход к одновременной записи данных  $buffer_{A_k}^{byte_i}$  в выделенную память в соответствие со схемой расположения для  $k$ -ого спектрального канала (Рис. 2 и Рис. 3). Вводится переменная «количество процессорных ядер», для конвертации BSQ→BIP - переменная «количество обрабатываемых спектральных каналов на ядро в порции данных» (5), для конвертации BSQ→BIL - переменная «разбивки порции данных по вычислительным ядрам» (6).

$$core_B = \frac{n}{core_N}, \quad n = core_B \cdot core_N + core_{remain}_B, \quad (5)$$

где  $core_B \in \mathbb{Z}$  - количество обрабатываемых спектральных каналов на ядро в порции данных,  $core_N \in \mathbb{Z}$  - количество процессорных ядер,  $core_{remain}_B \in \mathbb{Z}$  - остаток обрабатываемых спектральных каналов на ядро в порции данных.

$$core(heap_L) = \frac{heap_L}{core_N},$$

$$heap_L = core(heap_L) \cdot core_B + core_{remain}(heap_L), \quad (6)$$

где  $core(heap_L) \in \mathbb{Z}$  - разбивка порции данных по вычислительным ядрам,  $core_N$  - количество процессорных ядер,  $core_{remain}(heap_L) \in \mathbb{Z}$  - остаток разбивки порции данных по вычислительным ядрам.

$$BIL^i[k \cdot byte_L + j \cdot byte_L \cdot n] = [buffer_{A_k}^{byte_i}[byte_L \cdot j], buffer_{A_k}^{byte_i}[byte_L \cdot j + byte_L]] \quad (7)$$

где  $BIL^i$  - сконвертированная  $i$ -ая ( $i = 0 \dots \frac{heap_L}{num_L} - 1$ ) порция данных,  $j = c \cdot core(heap_L) \dots c \cdot core(heap_L) + core(heap_L)$ ,

где  $c = 0 \dots \frac{heap_L - core_{remain}(heap_L)}{core(heap_L)}$ ,  $k = 0 \dots n - 1$ ,  $k, c, i, j \in \mathbb{Z}$ .

$$remain_{BIL}^i[k \cdot byte_L + j \cdot byte_L \cdot n] = [buffer_{A_k}^{byte_i}[byte_L \cdot j], buffer_{A_k}^{byte_i}[byte_L \cdot j + byte_L]], \quad (8)$$

где  $remain_{BIL}^i$  - остаток сконвертированной  $i$ -ой порции данных после многопоточкового обработки,  $j = heap_L - core_{remain}(heap_L) \dots heap_L$ ,  $k = 0 \dots n - 1$

$$remain_{BIL}[k \cdot byte_L + j \cdot byte_L \cdot n] = \left[ \begin{array}{c} buffer_{A_k}^{byte_{remain}}[byte_L \cdot j], \\ buffer_{A_k}^{byte_{remain}}[byte_L \cdot j + byte_L] \end{array} \right], \quad (9)$$

где  $remain_{BIL}$  - остаток данных после разбиения на порции согласно (2),  $j = 0 \dots remain_L$ ,  $k = 0 \dots n - 1$ .

$$BIL = \left( \bigcup_{i=0}^{\frac{heap_L}{num_L}-1} (BIL^i \cup remain_{BIL}^i) \right) \cup remain_{BIL}, \quad (10)$$

где  $BIL$  - полный файл гиперспектрального снимка с преобразованием BIL.

Соответственно, для конвертации BSQ→BIP  $i$ -ой порции данных ( $i = 0 \dots \frac{heap_L}{num_L} - 1$ ) применялись формулы (11), (12).

$$BIP^i[2 \cdot j \cdot n + 2 \cdot k] = buffer_{A_k}^{byte_i}[2 \cdot j],$$

$$BIP^i[2 \cdot j \cdot n + 2 \cdot k + 1] = buffer_{A_k}^{byte_i}[2 \cdot j + 1] \quad (11),$$

где  $BIP^i$  – сконвертированная  $i$ -ая порция данных,  $j = 0 \dots heap_L \cdot width_L$ , где  $width_L$  – ширина снимка, равна значению параметра PRODUCT\_SAMPLES в метаописании снимка,  $k = c \cdot core_B \dots c \cdot core_B + core_B$ , где  $c = 0 \dots \frac{n - remain_B}{core_B}$ ,  $k, c, i, j \in \mathbb{Z}$ .

$$\begin{aligned} remain_{BIP}^i[2 \cdot j \cdot n + 2 \cdot k] &= buffer_{A_k}^{byte} A_k^i[2 \cdot j], \\ remain_{BIP}^i[2 \cdot j \cdot n + 2 \cdot k + 1] &= buffer_{A_k}^{byte} A_k^i[2 \cdot j + 1], \end{aligned} \quad (12)$$

где  $remain_{BIP}^i$  – остаток сконвертированной  $i$ -ой порции данных после многопоточкового обработки,  $j = 0 \dots heap_L \cdot width_L$ ,  $k = n - remain_B \dots n$ ,  $k, j \in \mathbb{Z}$ .

$$\begin{aligned} remain_{BIP}[2 \cdot j \cdot n + 2 \cdot k] &= buffer_{A_k}^{byte} A_k^{remain}[2 \cdot j], \\ remain_{BIP}[2 \cdot j \cdot n + 2 \cdot k + 1] &= \\ &= buffer_{A_k}^{byte} A_k^{remain}[2 \cdot j + 1] \end{aligned} \quad (13)$$

где  $remain_{BIP}$  – остаток данных после разбиения на порции согласно (2),  $j = 0 \dots remain_L \cdot width_L$ ,  $k = 0 \dots n$ .

$$BIP = \left( \bigcup_{i=0}^{heap_L-1} (BIP^i \cup remain_{BIP}^i) \right) \cup remain_{BIP}, \quad (14)$$

где  $BIP$  – полный файл гиперспектрального снимка с преобразованием BIL.

Программная реализация алгоритма представлена как расширение для программного продукта Exelis ENVI (Рис. 4). Графический интерфейс пользователя разработан собственными средствами пакета на базе технологии GUI-WIDGETS в интеграции с пакетами Java SwingX. Для взаимодействия с Java-классами, реализующими логику представленного алгоритма использовалась технология Java-Bridge IDL. Данная технология позволила инкапсулировать сложные процедуры расчетов от конечного пользователя,

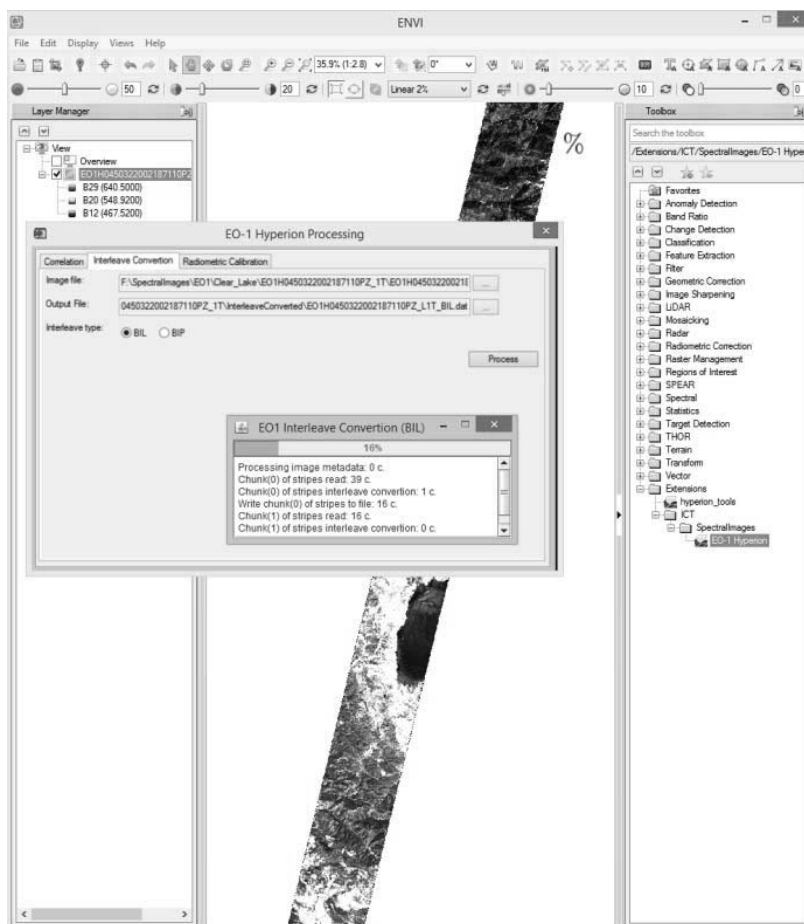


Рис. 4. Графический интерфейс расширения ENVI на базе многопоточкового алгоритма преобразование порядка следования данных отдельных каналов сенсора EO-1 Hyperion

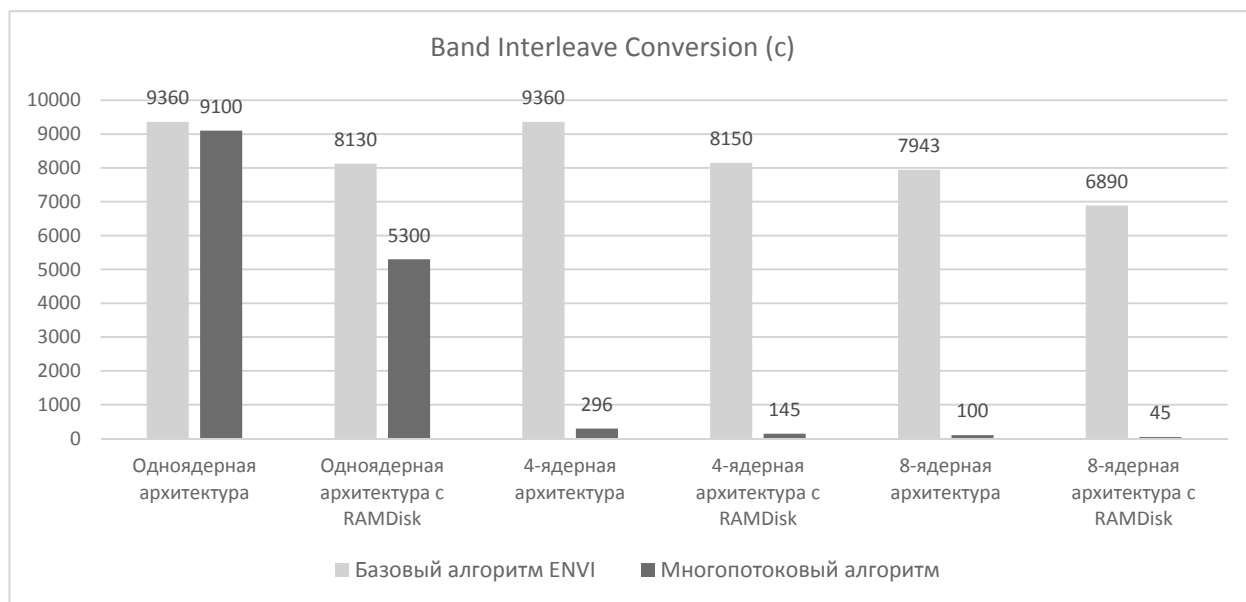


Рис. 5. Сравнительные результаты тестирования многопоточкового алгоритма по отношению к аналогичному базовому алгоритму ENVI

и представить конечный интерфейс, как обычное расширение ENVI IDL. Благодаря использованию Java-Bridge IDL разработанные программные компоненты Java могут быть использованы в сторонних программных продуктах, поддерживающих IDL. Гибкая настройка конфигурационных файлов Java-Bridge IDL позволила контролировать распределение машинных ресурсов на программном уровне, за счет передачи параметров запуска виртуальной машины Java «налету».

На Рис. 5 представлены результаты тестирования алгоритма. Тестирование проводилось на пяти гиперспектральных снимках, полученных с ресурса [6]. Были выбраны снимки размером 1960×6930, соответственно значения параметров PRODUCT\_SAMPLES и PRODUCT\_LINES 242 спектральных каналов. Для каждого снимка проводилось по 25 «проходов» алгоритмов (базового и многопоточкового). На графиках представлены средние значения времени работы алгоритмов. Значения для двух типов конвертаций BSQ→BIP и BSQ→BIL оказались приблизительно одинаковы, отличие составило порядка ±5 секунд.

### Заключение

Применение мульти-поточкового подхода позволило существенно сократить время (в 10 раз) загрузки снимка и преобразований (BIL, BIP,

BSQ). Этого удалось добиться путем разбиения процесса обработки гиперспектрального изображения по каналам, либо попиксельно. В первом случае, общее количество каналов разделено равномерно по количеству вычислительных ядер процессора(-ов), во втором случае, разбиение идет по ширине и/или по высоте снимка. А порционная загрузка данных снимка в оперативную память, позволила существенно снизить нагрузку на файловую систему ввода/вывода, что привело к снижению времени выполнения данных операций.

Использование технологии IDL-Java Bridge дало возможность интеграции разработанных программных компонентов в среде ENVI IDL. Предложенный алгоритм способен эффективно выполняться на платформах с низкопроизводительной файловой системой и количеством оперативной памяти (до 2 Гб).

Предложенный подход может быть реализован на вычислительных кластерах, что резко уменьшает время пакетной обработки гиперспектральных снимков.

### Литература

1. Adler-Golden S.M., Perkins T., Matthew M.W., Berk A., Bernstein L.S., et al. Speed and accuracy improvements in FLAASH atmospheric correction of hyperspectral imagery

- // SPIE Optical Engineering. 2012. Vol. 51(11). P. 111707(1-10).
2. BIL, BIP, and BSQ raster files. [Электронный ресурс] // ESRI. ArcGIS 9.2 Desktop Help. URL: [http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=BIL,\\_BIP,\\_and\\_BSQ\\_raster\\_files](http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=BIL,_BIP,_and_BSQ_raster_files) (дата обращения 22.07.2014).
  3. EarthExplorer [Электронный ресурс] // USGS. URL: <http://earthexplorer.usgs.gov/> (дата обращения 22.07.2014).
  4. Fast Line-of-sight Atmospheric Analysis of Hypercubes (FLAASH) [Электронный ресурс] // Exelis ENVI. URL: <http://www.exelisvis.com/docs/FLAASH.html> (дата обращения 24.07.2014).
  5. Hyperion level 1gst (L1GST) product output files data format control book. Earth Observing-1 (EO-1). Version 1.0. Department of the Interior U.S. Geological Survey. 2006. 24 P.
  6. Perkins T., Adler-Golden S.M., Cappelaere P., Mandl D. High-speed Atmospheric Correction for Spectral Image Processing // SPIE Proceeding: Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVIII. 2012. Vol. 8390. P. 245-252
  7. Qu Z., Goetz A. F. H., Kindel B. High-accuracy atmospheric correction for hyperspectral data (HATCH) model // Geoscience and Remote Sensing. 2003. Vol. 41(6). P. 1223 - 1231.
  8. San B. T., Suzen M. L. Evaluation of different atmospheric correction algorithms for EO-1 Hyperion imagery // International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science. Tokyo. 2010. Vol. 38(8), P. 392-397.
  9. The IDL Thread Pool [Электронный ресурс] // Exelis ENVI. URL: [http://www.exelisvis.com/docs/The\\_Thread\\_Pool.html](http://www.exelisvis.com/docs/The_Thread_Pool.html) (дата обращения 29.09.2014)
  10. Thompson B.J., Rahman Z., Park S.K. Multiscale retinex for improved performance in multispectral image classification // SPIE Proceedings: Visual Information Processing IX. 2000. Vol. 4041. P. 34-44.

**Потапов Вадим Петрович.** Заместитель директора Федерального государственного бюджетного учреждения науки Институт вычислительных технологий Сибирского отделения Российской академии наук (ИВТ СО РАН). Окончил Томский политехнический институт в 1974 году. Доктор технических наук, профессор. Автор 152 печатных работ, в том числе 6 монографий. Область научных интересов: распределенные вычислительные системы и облачные технологии, геоинформационные системы. E-mail: [potapov@ict.sbras.ru](mailto:potapov@ict.sbras.ru)

**Попов Семен Евгеньевич.** Старший научный сотрудник ИВТ СО РАН. Окончил Кемеровский государственный университет в 2002 году. Кандидат технических наук. Автор 26 печатных работ. Область научных интересов: математическое моделирование, распределенные вычислительные системы и облачные технологии. E-mail: [popov@ict.sbras.ru](mailto:popov@ict.sbras.ru)