

# ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ DIGITAL PROCESSING OF SIGNALS AND IMAGES

УДК 004.042

**В. П. Потапов**, д-р техн. наук, зам. директора — директор филиала, potapov@ict.sbras.ru,

**С. Е. Попов**, канд. техн. наук, ст. науч. сотр., popov@ict.sbras.ru

Федеральное государственное бюджетное учреждение науки Институт вычислительных технологий  
Сибирского отделения Российской академии наук, Новосибирск

## Многопоточный алгоритм конвертации чередования данных каналов гиперспектральных снимков

*Представлен высокопроизводительный алгоритм преобразования данных спектральных каналов (Band Interleave Conversion) сенсора EO-1 Hyperion, предусматривающий возможность запуска его на многопроцессорных платформах в мультипоточковом режиме и обеспечивающий эффективное выполнение на низкопроизводительных системах ввода/вывода. Предложена реализация порционного считывания данных снимка в оперативную память с последующим размещением значений (Digitals Number) в целевых буферных массивах в несколько потоков, рассчитанных по числу спектральных каналов или по числу линий снимка. Представлено также расширение для программного продукта Exelis ENVI, реализующее разработанный алгоритм, на базе технологии GUI-WIDGETS в интеграции с пакетами Java SwingX. Приведены результаты тестирования представленных алгоритмов с их базовыми аналогами программного комплекса Exelis ENVI.*

**Ключевые слова:** преобразование порядка следования данных, спектральные каналы, гиперспектральные изображения, многопотоковость, Java, IDL-Bridge, EO-1 Hyperion

### 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 улучшении 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 рассчитывают исходя из усредненных показателей и не учитывают особенности 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 пред- и постобработки мульти- и гиперспектральных изображений, а также на сравнение разрешающей способности алгоритмов. Так, в работе [10] представлен алгоритм *Multi-Scale*

*Retinex (MSR)*, обеспечивающий сжатие динамического диапазона, снижение зависимости от условий освещения, а также улучшение пространственного разрешения мультиспектральных данных. В частности, результаты моделирования показывают, что, несмотря на степень снижения качества изображения вследствие изменения атмосферного излучения, ошибки классификации могут быть существенно уменьшены путем предварительной обработки данных изображения с помощью MSR. В работе [8] сравниваются алгоритмы атмосферной коррекции *Atmospheric CORrection Now (ACORN)*, *the Fast Line-of-sight Atmospheric Analysis of Spectral Hypercubes (FLAASH)*, and *ATmospheric CORrection (ATCOR 2-3)* для гиперспектральных данных сенсора EO-1 Hyperion для кросс-корреляции и приводятся особенности поглощения в определенных длинах волн. Показано, что в зависимости от различных диапазонов длин волн алгоритм атмосферной коррекции ACORN показал лучшую корректирующую способность для снимков EO-1 Hyperion для определения литологии и минералогического состава природных земельных геоматериалов. В работе [7] описаны результаты использования моделей алгоритмов ATREM, HATCH, ACORN и FLAASH и приведены 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 делает невозможным 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 число артефактов и отклонение от сглаженной кривой отражения. Обнаружено, что использование алгоритма MATCH-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 пред- и постобработки спектральных изображений. В работах [1] и [6] рассматривается программный прототип системы автоматизированного, высокоточного алгоритма с малой задержкой для расчета атмосферной коррекции на основе C++ для модуля FLAASH. Предложенный в работах [1] и [6] подход заключается в замене расчетов моделей переноса радиации модуля MODTRAN на предварительно рассчитанные справочные таблицы, используя для этого параллельную обработку на многоядерных/многoproцессорных 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 ее интеграции 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, работа модуля *Band Interleave* с опцией подготовки данных для атмосферной коррекции алгоритмами FLAASH в программном комплексе ENVI для гиперспектрального снимка (7,7 × 185 км) сенсора EO-1 составляет около 2,5 ч. При этом некорректно используются ресурсы файловой подсистемы вследствие неоптимизированного порядка считывания файлов данных спектральных каналов. Конечно, стоит сказать, что в том же программном продукте Exelis ENVI есть встроенная технология мультипоточности (The IDL Thread Pool: [сайт]. URL: [http://www.exelisvis.com/docs/The\\_Thread\\_Pool.html](http://www.exelisvis.com/docs/The_Thread_Pool.html)), но она представлена по принципу *hard coded*, т. е. разработчик ее не может настраивать и ею управлять, 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 описана разработка высокопроизводительного алгоритма для предваритель-

ной обработки (подготовки) гиперспектральных данных с последующей их передачей на вход модуля атмосферной коррекции программного продукта Exelis ENVI.

## Постановка задачи

Разработать алгоритм преобразования 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 объемом оперативной памяти до 2 Гбайт.

## Используемые данные и программные продукты

Для тестирования разрабатываемых алгоритмов использованы данные гиперспектральных изображений, полученных с ресурса *EarthExplorer* (EarthExplorer: [сайт]. URL: <http://earthexplorer.usgs.gov/>) Геологической службы США. В работе рассматривались изображения сенсора EO-1 Hyperion с полосами съемки 7,7 × 42 км и 7,7 × 185 км. Гиперспектральный сенсор Hyperion предназначен для получения цифровых изображений земной поверхности с пространственным разрешением до 30 м в 242 спектральных диапазонах видимой и инфракрасной областей.

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 EOlsppprrrYYYYDDDXXML\_BBBB\_TTT.TIF (Department of the Interior U. S. Geological Survey, 2006). Размер снимка определяется числом линий (параметр PRODUCT\_LINES) и числом пикселей одной линии (параметр PRODUCT\_SAMPLES). Эта информация размещается в файле метаописания (EarthExplorer: [сайт]. URL: <http://earthexplorer.usgs.gov/>).

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 указывают 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 (EarthExplorer: [сайт]. URL: <http://earthexplorer.usgs.gov/>), имеют формат преобразования порядка следования данных каналов BSQ (*band sequential*) (рис. 1, см. четвертую сторону обложки). 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 (число каналов)  $\times$  (число линий)  $\times$  (число пикселей в линии)  $\times 2$  и так далее по числу спектральных каналов. Тип данных для каждого значения пикселя — Short, размером 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.

### 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 LIT 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 являются операции ввода/вывода файловой подсистемы, а конкретнее — компромисс между использованием оперативной памяти и временем загрузки исходных данных в нее.

Согласно [4] исходное изображение для модуля атмосферной коррекции FLAASH должно иметь значение светимости для каждого пикселя в спектральном канале и быть в формате преобразования BIL (*band-interleaved-by-line*) или VIP (*band-interleaved-by-pixel*).

В формате BIL данные хранят значение для каждой линии снимка в порядке следования "канал за каналом", соответственно, для каждой строки (см. рис. 1, на четвертой стороне обложки). Формат VIP аналогичен BIL за исключением того, что данные в порядке следования "канал за каналом" хранятся для каждого пикселя (рис. 1) [2].

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

*Первый этап* аналогичен для BSQ  $\rightarrow$  BIL и BSQ  $\rightarrow$  VIP. Здесь осуществляется проверка общей длины

снимка в байтах в целях использования порционного считывания данных из файлов. Пороговое значение для порции установлено в 1 Гбайт. Выбрано оно опытным путем исходя из минимальных требований, необходимых для исполнения программного кода виртуальной машины Java, максимальной длины массива в Java и минимизации использования файла подкачки операционной системы Windows. На этом этапе вводятся следующие понятия: число линий снимка, размещенных в "heap size"  $^{heap}L$ , целое число таких размещений для общего числа линий  $^{num}L$  и остаток линий с учетом полного покрытия снимка  $^{remain}L$ :

$$^{heap}L = \frac{1024^3}{^{byte}L \cdot n}, \quad (1)$$

где  $^{heap}L \in \mathbb{Z}$  — число линий в "heap size" (в программном коде задано как тип Integer, поэтому результат деления — всегда целое число),  $^{byte}L \in \mathbb{Z}$  — длина одной линии в байтах,  $n \in \mathbb{Z}$  — числе спектральных каналов;

$$^{num}L = \frac{^{heap}L \cdot ^{heap}L}{^{num}L} + ^{remain}L, \quad (2)$$

где  $^{num}L \in \mathbb{Z}$  — число линий снимка для каждого спектрального канала,  $^{heap}L \in \mathbb{Z}$  — число размещений для общего числа линий,  $^{remain}L \in \mathbb{Z}$  — остаток линий с учетом полного покрытия снимка.

*Второй этап* также аналогичен для двух типов конвертаций BSQ  $\rightarrow$  BIL и BSQ  $\rightarrow$  VIP. На данном этапе выполняются самые длительные по времени операции. Здесь происходит порционное считывание данных из файлов спектральных каналов в массив типа BufferArray, содержащий данные  $i$ -й порции  $^{byte}A_k^i$  и остаток данных  $^{byte}A_k^{remain}$  для каждого из спектральных каналов:

$$^{byte}A_k^i = [_{offset}L + ^{heap}L \cdot ^{byte}L \cdot i, _{offset}L + ^{heap}L \cdot ^{byte}L \cdot (i + 1)]_k; \quad (3)$$

$$^{byte}A_k^{remain} = [_{offset}L + ^{heap}L \cdot ^{byte}L \cdot \frac{^{heap}L}{^{num}L}, _{offset}L + ^{heap}L \cdot ^{byte}L \cdot \frac{^{heap}L}{^{num}L} + ^{remain}L]_k, \quad (4)$$

где  $^{byte}A_k^i$  — элемент массива порции данных и  $^{byte}A_k^{remain}$  — остаток данных  $k$ -го 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 смещения байтов в TIFF-файле  $k$ -го спектрального канала, соответствующие считываемой  $i$ -й порции данных или остатка данных.

Далее будем рассматривать  $^{byte}A_k^i$  и  $^{byte}A_k^{remain}$  как массивы длиной  $^{heap}L \cdot ^{byte}L$  и  $^{remain}L \cdot ^{byte}L$  соответственно.

На *третьем этапе* происходит размещение данных  $\text{byte}_{buffer}A$  в массив типа BufferArray согласно выбранному типу конвертации  $\text{BSQ} \rightarrow \text{BIL}$  [см. ниже формулы (7)–(10)] или  $\text{BSQ} \rightarrow \text{VIP}$  [см. ниже формулы (11)–(14)] с последующим сохранением его на диск. Здесь применяется мультипоточковый подход к одновременной записи данных  $\text{byte}_{buffer}A^i$  в выделенную память в соответствии со схемой расположения для  $k$ -го спектрального канала (см. рис. 1). Вводится переменная "число процессорных ядер", для конвертации  $\text{BSQ} \rightarrow \text{VIP}$  — переменная "число обрабатываемых спектральных каналов на ядро в порции данных"  $\text{core}B \in \mathbb{Z}$ , для конвертации  $\text{BSQ} \rightarrow \text{BIL}$  — переменная "разбивки порции данных по вычислительным ядрам"  $\text{core}(\text{heap}L) \in \mathbb{Z}$ :

$$\text{core}B = \frac{n}{\text{core}N}, n = \text{core}B \cdot \text{core}N + \text{core}_{\text{remain}}B, \quad (5)$$

где  $\text{core}N \in \mathbb{Z}$  — число процессорных ядер;  $\text{core}_{\text{remain}}B \in \mathbb{Z}$  — остаток обрабатываемых спектральных каналов на ядро в порции данных;

$$\text{core}(\text{heap}L) = \frac{\text{heap}L}{\text{core}N},$$

$$\text{heap}L = \text{core}(\text{heap}L) \cdot \text{core}B + \text{core}_{\text{remain}}(\text{heap}L), \quad (6)$$

где  $\text{core}_{\text{remain}}(\text{heap}L) \in \mathbb{Z}$  — остаток разбивки порции данных по вычислительным ядрам;

$$\text{BIL}^i[k \cdot \text{byte}L + j \cdot \text{byte}L \cdot n] = \lfloor \text{byte}_{buffer}A_k^i[\text{byte}L \cdot j],$$

$$\text{byte}_{buffer}A_k^i[\text{byte}L \cdot j + \text{byte}L], \quad (7)$$

где  $\text{BIL}^i$  — сконвертированная  $i$ -я ( $i = 0 \dots \frac{\text{heap}L}{\text{num}} - 1$ ) порция данных,  $j = c \cdot \text{core}(\text{heap}L) \dots c \cdot \text{core}(\text{heap}L) + \text{core}_{\text{remain}}(\text{heap}L)$ , где  $c = 0 \dots \frac{\text{heap}L - \text{core}_{\text{remain}}(\text{heap}L)}{\text{core}(\text{heap}L)}$ ,  $k = 0 \dots n - 1$ ,  $k, c, i, j \in \mathbb{Z}$ ;

$$\text{remainBIL}^i[k \cdot \text{byte}L + j \cdot \text{byte}L \cdot n] =$$

$$= \lfloor \text{byte}_{buffer}A_k^i[\text{byte}L \cdot j], \text{byte}_{buffer}A_k^i[\text{byte}L \cdot j + \text{byte}L], \quad (8)$$

где  $\text{remainBIL}^i$  — остаток сконвертированной  $i$ -й порции данных после многопоточковой обработки,  $j = \text{heap}L - \text{core}_{\text{remain}}(\text{heap}L) \dots \text{heap}L$ ,  $k = 0 \dots n - 1$ ;

$$\text{remainBIL}[k \cdot \text{byte}L + j \cdot \text{byte}L \cdot n] =$$

$$= \lfloor \text{byte}_{buffer}A_k^{\text{remain}}[\text{byte}L \cdot j], \text{byte}_{buffer}A_k^{\text{remain}}[\text{byte}L \cdot j + \text{byte}L], \quad (9)$$

где  $\text{remainBIL}$  — остаток данных после разбиения на порции согласно (2),  $j = 0 \dots \text{remain}L$ ,  $k = 0 \dots n - 1$ ;

$$\text{BIL} = \left( \bigcup_{i=0}^{\frac{\text{heap}L}{\text{num}} - 1} (\text{BIL}^i \cup \text{remainBIL}^i) \right) \cup \text{remainBIL}, \quad (10)$$

где  $\text{BIL}$  — полный файл гиперспектрального снимка с преобразованием  $\text{BIL}$ .

Соответственно, для конвертации  $\text{BSQ} \rightarrow \text{VIP}$   $i$ -й порции данных ( $i = 0 \dots \frac{\text{heap}L}{\text{num}} - 1$ ) применяли следующие формулы:

$$\text{VIP}^i[2j \cdot n + 2k] =$$

$$= \text{byte}_{buffer}A_k^i[2j], \text{VIP}^i[2j \cdot n + 2k + 1] =$$

$$= \text{byte}_{buffer}A_k^i[2j + 1], \quad (11)$$

где  $\text{VIP}^i$  — сконвертированная  $i$ -я порция данных,  $j = 0 \dots \frac{\text{heap}L \cdot \text{width}L}{\text{width}L}$ ;  $\text{width}L$  — ширина снимка, равная значению параметра PRODUCT\_SAMPLES в метаописании снимка,  $k = c \cdot \text{core}B \dots c \cdot \text{core}B + \text{core}B$ ,

$$c = 0 \dots \frac{n - \text{core}_{\text{remain}}B}{\text{core}B}, k, c, i, j \in \mathbb{Z};$$

$$\text{remainVIP}^i[2j \cdot n + 2k] = \text{byte}_{buffer}A_k^i[2j],$$

$$\text{remainVIP}^i[2j \cdot n + 2k + 1] = \text{byte}_{buffer}A_k^i[2j + 1], \quad (12)$$

где  $\text{remainVIP}^i$  — остаток сконвертированной  $i$ -й порции данных после многопоточковой обработки,  $j = 0 \dots \frac{\text{heap}L \cdot \text{width}L}{\text{width}L}$ ,  $k = n - \text{core}_{\text{remain}}B \dots n$ ,  $k, j \in \mathbb{Z}$ ;

$$\text{remainVIP}[2j \cdot n + 2k] = \text{byte}_{buffer}A_k^{\text{remain}}[2j],$$

$$\text{remainVIP}[2j \cdot n + 2k + 1] = \text{byte}_{buffer}A_k^{\text{remain}}[2j + 1], \quad (13)$$

где  $\text{remainVIP}$  — остаток данных после разбиения на порции согласно (2),  $j = 0 \dots \text{remain}L \cdot \text{width}L$ ,  $k = 0 \dots n$ ;

$$\text{VIP} = \left( \bigcup_{i=0}^{\frac{\text{heap}L}{\text{num}} - 1} (\text{VIP}^i \cup \text{remainVIP}^i) \right) \cup \text{remainVIP}, \quad (14)$$

где  $\text{VIP}$  — полный файл гиперспектрального снимка с преобразованием  $\text{BIL}$ .

Программная реализация алгоритма представлена как расширение для программного продукта Exelis ENVI (рис. 2, см. четвертую сторону обложки). 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, использовали технологию Java-Bridge IDL. Данная технология позволила инкапсулировать сложные процедуры расчетов от конечного пользователя и представить конечный интерфейс как обычное расширение ENVI IDL. Благодаря технологии 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, "на лету".

Ниже представлены результаты тестирования алгоритма (рис. 3, см. четвертую сторону обложки). Тестирование проводили на пяти гиперспектральных снимках, полученных с ресурса [9]. Были выбраны снимки размерами 1960 × 6930, 242 спектральных каналов. Для каждого снимка проводили по 25 "проходов" алгоритмов (базового и многопоточного). На графиках рис. 3 (см. четвертую сторону обложки) представлены средние значения времени работы алгоритмов. Значения для двух типов конвертаций BSQ → VIP и BSQ → BIL оказались приблизительно одинаковы, отличие составило около ±5 с.

### Заключение

Применение мультипоточного подхода позволило существенно сократить время (в 10 раз) загрузки снимка и преобразований (BIL, VIP, BSQ). Этого удалось добиться путем разбиения процесса обработки гиперспектрального изображения по каналам, либо попиксельно, где в первом случае общее число каналов разделено равномерно по числу вычислительных ядер процессора(-ов), во втором случае разбиение идет по ширине и/или по высоте снимка, а также путем порционной загрузки данных снимка в оперативную память, что позволило существенно снизить нагрузку на файловую систему ввода/вывода и 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 файло-

вой системой и емкостью оперативной памяти (до 2 Гбайт).

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

### Список литературы

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, VIP, and BSQ raster files.** // ESRI. ArcGIS 9.2 Desktop Help. URL: [http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=BIL,\\_VIP,\\_and\\_B\\_SQ\\_raster\\_files](http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=BIL,_VIP,_and_B_SQ_raster_files) (дата обращения 22.07.2014).
3. **EarthExplorer.** // USGS. URL: <http://earthexplorer.usgs.gov/> (дата обращения 22.07.2014)
4. **Fast Line-of-sight Atmospheric Analysis of Hypercubes (FLAASH)** // 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.** // Exelis ENVI. URL: [http://www.exelisvis.com/docs/TheThread\\_Pool.html](http://www.exelisvis.com/docs/TheThread_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.

**V. P. Potapov**, Deputy Managing Director, Branch Manager, potapov@ict.sbras.ru,

**S. E. Popov**, Chief Scientist, popov@ict.sbras.ru

Institute of Computational Technologies SB RAS, Novosibirsk, 630090, Russia

## Multithreaded Algorithm for Band Interleave Conversion of Hyperspectral Images

*This paper presents a high-performance data conversion algorithm of spectral bands (Band Interleave Conversion) of the sensor EO-1 Hyperion, providing the ability to run it on multiprocessor platforms in multi-threaded mode and ensures the effective implementation on low I/O systems. Proposed implementation of proportional reading image data into RAM, followed by the placement of values (Digitals Number) in the target buffer arrays in multiple threads, calculated by the number of spectral channels or the number of lines of the image. Results of testing the algorithm presented with their basic counterparts software package Exelis ENVI. It is shown that the speed (second) of the multithreaded algorithm hundreds of times higher than for the native algorithm. In particular, the process of multithreaded band interleave conversion on the 8-core architecture with the use RAMDisk I/O-subsystem was only 45 seconds, while the native algorithms needed in similar environment of about 6800 seconds. The paper presents an Exelis ENVI extension, realizing the developed algorithm based on the GUI-WIDGETS integration with packages Java SwingX using IDL-Bridge technology.*

**Keywords:** converting the order of the data, spectral channels, hyperspectral images, multithreading, java, IDL-Bridge, EO-1 Hyperion

## References

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), pp. 111707 (1–10).
2. BIL, BIP, and BSQ raster files. *ESRI. ArcGIS 9.2 Desktop Help*. Retrieved from. 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)
3. **EarthExplorer**. *USGS*. Retrieved from <http://earthexplorer.usgs.gov/>.
4. **Fast** Line-of-sight Atmospheric Analysis of Hypercubes (FLAASH), *Exelis ENVI*. Retrieved from <http://www.exelisvis.com/docs/FLAASH.html>
5. **Hyperion** level 1 gst (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, pp. 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), pp. 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), pp. 392–397.
9. The IDL Thread Pool. *Exelis ENVI*. Retrieved from [http://www.exelisvis.com/docs/The\\_Thread\\_Pool.html](http://www.exelisvis.com/docs/The_Thread_Pool.html).
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, pp. 34–44.

УДК 621.391

С. В. Дворников, д-р техн. наук, проф., профессор каф.,  
А. В. Пшеничников, канд. техн. наук, доц., доц. каф.,  
С. С. Манаенко, канд. техн. наук, ст. преподаватель каф.,  
Военная академия связи, г. Санкт-Петербург, e-mail: manaenkoss@mail.ru

## Помехоустойчивая модель сигнала КАМ-16 с трансформированным созвездием

*Представлены результаты исследования, направленные на разработку модели сигнала КАМ-16 с трансформированной конstellационной диаграммой. Определяются показатели помехоустойчивости, по которым оценивается разработанная модель и модель с трансформированным созвездием в соответствии со стандартом цифрового телевидения DVB-T2.*

**Ключевые слова:** помехоустойчивость, конstellационная диаграмма, вектор сигнального созвездия, квадратурная амплитудная манипуляция

### Введение

В настоящее время в радиотехнике широко используются сигнальные конструкции (СК), формируемые квадратурным методом на основе синфазной ( $I$ ) и квадратурной ( $Q$ ) составляющих сигнала. В частности, к таковым относятся сигналы квадратурной амплитудной манипуляции (КАМ). Каждый сигнал КАМ можно изобразить в виде сигнальной точки, координаты которой определяются значениями  $Q$  и  $I$ . Совокупность таких точек образует так называемое сигнальное созвездие (конstellационную диаграмму). При этом наибольшее распространение в радиотехнике получили созвездия, имеющие прямоугольную диаграмму из 16 сигнальных точек (КАМ-16) [1, 2].

Как правило [3], помехоустойчивость СК определяется минимальным евклидовым расстоянием (МЕР)  $d_E$ . Указанный показатель представляет собой наименьшее геометрическое расстояние между точками сигнального созвездия. Другим важным

энергетическим показателем СК является отношение максимальной мощности СК к ее среднему значению, определяемому в работе [4] как пик-фактор (ПФ). По показателю МЕР сигналы КАМ являются более помехоустойчивыми по отношению к сигналам фазовой манипуляции такой же размерности (позиционности), однако по показателю ПФ они проигрывают последним.

Между тем в телевизионном стандарте DVB-T2 для повышения помехоустойчивости рекомендуется использовать модели СК с трансформированными конstellационными диаграммами, полученными путем поворота всех точек сигнального созвездия на одинаковый угол. В результате применения указанной процедуры каждая точка сигнального созвездия получает независимые координаты, которые дают возможность на приемном конце восстановить позиции точки сигнального вектора конstellационной диаграммы СК даже по одной координате, и тем самым повысить помехоустойчивость процедур демодуляции.