

КРАТКИЕ СООБЩЕНИЯ

УДК 004.93'1; 004.932

Д.В. Прадун, С.В. Овсянников, А.А. Кравцов

**БЛОЧНО-ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ
С ПОМОЩЬЮ БИБЛИОТЕКИ INTEL THREADING BUILDING BLOCKS**

Описываются основные понятия и компоненты библиотеки блочно-параллельной обработки данных Intel Threading Building Blocks. Исследуется возможность использования библиотеки для реализации параллельных алгоритмов обработки мультиспектральных изображений. Приводятся примеры параллельной фильтрации, бинаризации и определения градиента изображений. Дается сравнение скорости обработки мультиспектральных изображений в последовательном и параллельном режимах.

Введение

Большинство операций при обработке изображений предполагает выполнение одних и тех же вычислений над разными данными и получение результатов обработки отдельного блока данных, не зависящих от результатов других операций. Это, в свою очередь, предусматривает широкие возможности для распараллеливания, но выполнение распараллеливания путем непосредственной работы с потоками для каждой отдельной операции представляет собой сложную и трудоемкую задачу. Библиотека языка C++ Threading Building Blocks (TBB) от компании Intel позволяет разработчику создавать многопоточные приложения, абстрагируясь от непосредственного управления потоками. Данная библиотека поддерживает масштабируемое параллельное программирование с использованием стандартного языка C++ и не требует применения специализированных языков или компиляторов.

В настоящей работе показаны результаты использования библиотеки TBB при параллельной реализации алгоритмов бинаризации, фильтрации и определения градиента мультиспектральных изображений.

1. Общее описание библиотеки

Компоненты библиотеки можно разделить на несколько типов [1]:

- компоненты, непосредственно отвечающие за выполнение параллельных вычислений;
- компоненты-контейнеры, обеспечивающие возможность одновременного доступа к данным из нескольких потоков;
- компоненты контроля параллельного выполнения и доступа к данным;
- распределители памяти;
- планировщик задач;
- внутренние типы данных.

К компонентам, непосредственно отвечающим за выполнение параллельных вычислений, относятся простые циклы (`parallel_for`, `parallel_reduce`) и сложные циклы (`parallel_while`, `pipeline`).

Все параллельные вычисления в рамках библиотеки производятся посредством вызова функций простых либо сложных циклов. Доступ к компонентам-контейнерам библиотеки TBB может осуществляться как при использовании самой библиотеки, так и при использовании стандартных потоков. Параллельные контейнеры позволяют считывать и записывать элементы контейнера нескольким потокам одновременно [1]. К компонентам-контейнерам относятся: хэш-таблица (`concurrent_hash_map`), вектор (`concurrent_vector`), очередь (`concurrent_queue`).

Компоненты контроля параллельного выполнения и доступа к данным подразделяются на два вида: смешанные (`mutex`) и атомарные (`Atomic Operations`). Суть функционирования первых заключается в контроле последовательности выполнения потоков. Вторые представляют собой ограниченный набор функций, выполнение которых для других потоков выглядит мгновенным.

Распределители памяти, входящие в состав библиотеки, обеспечивают следующие возможности:

– многопоточную работу с памятью для приложений, активно выделяющих и освобождающих память;

– исключение ошибок, связанных с изменением данных в кэше только одного ядра.

Планировщик заданий является мощным инструментом, обеспечивающим многопоточную работу всех циклов [1].

2. Использование библиотеки для реализации алгоритмов обработки изображений

Так как библиотека ТВВ предназначена для параллельной обработки входных данных, это подразумевает под собой тот факт, что ни одно значение отдельного блока данных, представляющих собой результат той или иной обработки, не зависит ни от одного значения другого блока данных. Поэтому данная библиотека удобна при выполнении практически любых блочно-параллельных процессов обработки изображений, например бинаризации, определения градиента, фильтрации, кластеризации, классификации и т. д.

2.1. Бинаризация изображений

Для бинаризации, т. е. представления мультиспектральных изображений в черно-белом формате, использовался кластерный метод автоматического определения порога бинаризации Отсу [2] (рис. 1). Данный метод основан на вычислении порога с помощью вероятностных оценок. Значение порога T_{opt} в алгоритме Отсу определяется как

$$T_{opt} = \arg \max \left\{ \frac{P_b(T)[1 - P_b(t)][m_f(T) - m_b(T)]^2}{P_b(T)\sigma_b^2 + [1 - P_b(T)]\sigma_f^2(T)} \right\}, \quad (1)$$

где $P_b(T)$ – суммарная функция распределения вероятности принадлежности пиксела заднему фону изображения при заданном пороге T ; $m_f(T)$ и $m_b(T)$ – математические ожидания функции вероятности переднего плана и заднего фона изображения соответственно; $\sigma_f(T)$ и $\sigma_b(T)$ – дисперсии [2].

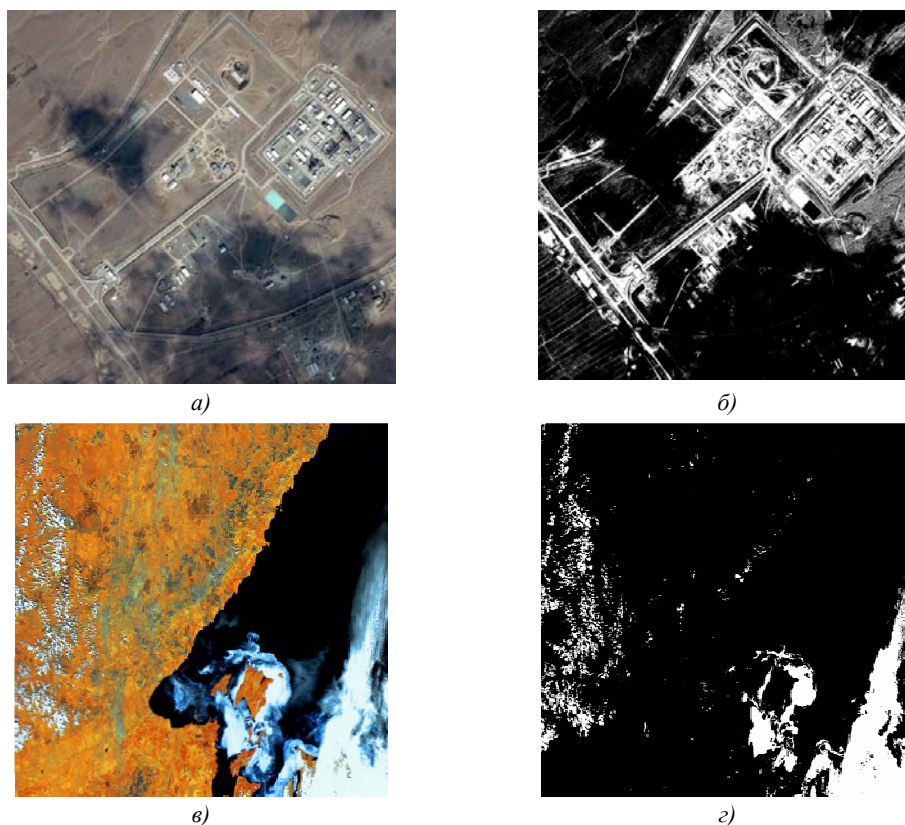


Рис. 1. Бинаризация изображений с помощью алгоритма Отсу: а), в) исходные изображения; б), з) результаты бинаризации

В табл. 1 представлены временные характеристики выполнения бинаризации в последовательном и параллельном режимах.

Таблица 1

Результаты бинаризации

Характеристики изображений	Время обработки, с	
	в последовательном режиме	в параллельном режиме
3 канала, 2638 × 2546 пикселей	2,9	2,7
3 канала, 5000 × 5000 пикселей	3,8	3,6

2.2. Вычисление градиента

Градиентными изображениями являются ребра, разделяющие соответствующие соседние пиксели исходного изображения. В нашем случае выполнялось вычисление градиента с помощью масок в вертикальном, горизонтальном и диагональных направлениях [3]:

1	2	1	1	-1	1	2	1	0	0	-1	-2	1
-1	-2	-1	2	-2	0	-1	-2	-1	1	2	1	0
			1	-1								

На рис. 2 и в табл. 2 представлены результаты вычисления градиента и временные характеристики данных вычислений.

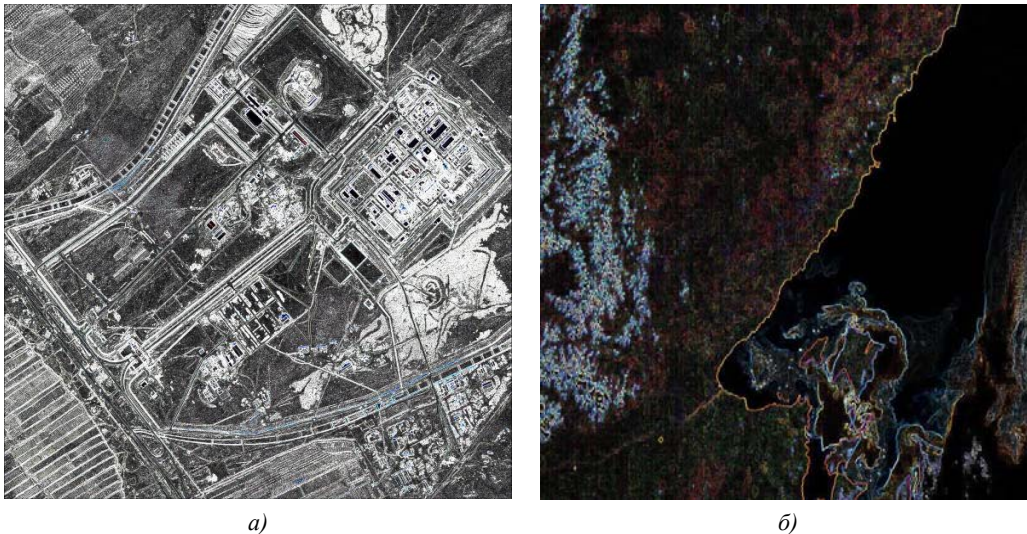


Рис. 2. Примеры вычисления градиента: а) изображения на рис. 1, а; б) изображения на рис. 1, в

Таблица 2

Результаты вычисления градиента в последовательном и параллельном режимах

Характеристики изображений	Время обработки, с	
	в последовательном режиме	в параллельном режиме
3 канала, 2638 × 2546 пикселей	7,1	5,4
3 канала, 5000 × 5000 пикселей	17,7	10,0

2.3. Фильтрация изображений

Алгоритм фильтрации мультиспектральных изображений описан в [3–5]. Он включает в себя следующие основные процедуры:

- определение градиента в четырех направлениях;
- бинаризацию градиента;
- вычисление локальной либо глобальной дисперсии изображения для определения коэффициентов последующей фильтрации гауссовским оператором;
- непосредственно фильтрацию с помощью гауссовского оператора.

Результаты фильтрации и временные затраты на выполнение представлены на рис. 3 и в табл. 3 соответственно. Все тесты проводились на персональном двухъядерном компьютере Intel Core 2 Duo 1.8 GHz.

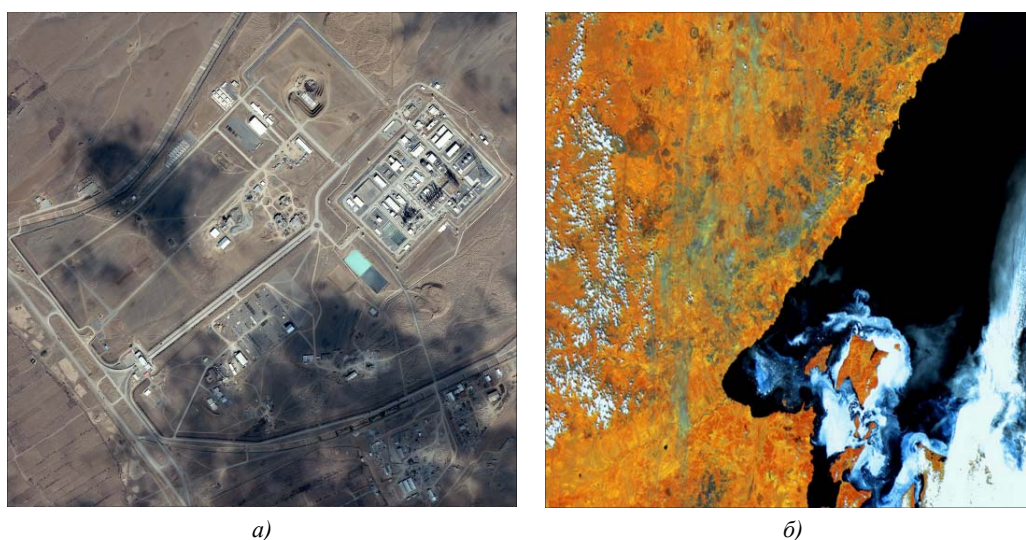


Рис. 3. Примеры адаптивной фильтрации: а) изображения на рис. 1, а; б) изображения на рис. 1, в

Таблица 3
Результаты адаптивной фильтрации в последовательном и параллельном режимах

Характеристики изображений	Время обработки, с	
	в последовательном режиме	в параллельном режиме
3 канала, 2638 × 2546 пикселей	412	52
3 канала, 5000 × 5000 пикселей	1 604	177

Заключение

Проведенные тесты показали, что использование библиотеки Intel TBB при реализации блочно-параллельной обработки данных особенно эффективно в случае выполнения сложных вычислений над снимками большого размера. Эффективность параллельных вычислений может быть увеличена за счет увеличения количества вычислительных ядер как на одном компьютере, так и в пределах вычислительной сети. Однако при параллельной реализации более сложных алгоритмов обработки изображений может потребоваться разработка нового или модификация старого алгоритма. Это обусловлено спецификой использования библиотеки TBB: каждый отдельный блок данных из набора всех блоков, на которые разбивается исходное изображение, рассматривается как самостоятельное изображение, без учета остальных данных.

Список литературы

1. IntelThreading Building Blocks – Documentation Threading Building Blocks – Documentation [Electronic resource]. – Mode of access : <http://software.intel.com/en-us/articles/intel-threading-building-blocks-documentation>. – Date of access : 12.10.2009.

2. Прадун, Д.В. Автоматическая бинаризация градиентных изображений на основе кластерного метода определения пороговой величины / Д.В. Прадун, Б.А. Залесский // Информатика. – 2010. – № 1 (25). – С. 5–12.

3. Залесский, Б.А. Алгоритм адаптивной фильтрации мультиспектральных изображений / Б.А. Залесский, Д.В. Прадун // Информатика. – 2009. – № 2 (22). – С. 31–38.

4. Прадун, Д.В. Фильтрация мультиспектральных изображений с помощью адаптивного оператора Гаусса на основе автоматической бинаризации градиента / Д.В. Прадун // Четвертый Белорусский космический конгресс, Минск, 25–27 октября, 2009. – Минск : ОИПИ НАН Беларуси, 2009. – С. 165–169.

5. Pradun, D. Adaptive filtration of multispectral images / D. Pradun, B. Zalesky, A. Kravtsov // Proc. of the 10th Intern. Conf. PRIP'2009, Minsk, May 19–21, 2009 / Belarusian State University. – Minsk, 2009. – P. 104–107.

Поступила 19.11.10

*Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: pradundv@gmail.com*

D.V. Pradun, S.V. Ovsyannikov, A.A. Kravtsov

**BLOCK-PARALLEL IMAGE PROCESSING
USING THE INTEL THREADING BUILDING BLOCKS LIBRARY**

The basic concepts and components of a block-parallel data processing library known as the Intel Threading Building Blocks are described. The possibility of library usage for implementing parallel algorithms of multispectral image processing is investigated. Examples of a parallel filtering, thresholding and gradient calculation are shown. A comparison of calculation speed of multispectral image processing in consecutive and parallel modes is provided.