

ИССЛЕДОВАНИЕ БЫСТРОДЕЙСТВИЯ АЛГОРИТМОВ ПОИСКА НА ИЗОБРАЖЕНИЯХ ОДНОИМЕННЫХ ТОЧЕК С ИСПОЛЬЗОВАНИЕМ СОВРЕМЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

*Кузнецов А.Е., зам. директора НИИ «Фотон» РГРТУ, д.т.н., профессор; foton@rsreu.ru;
Рыжиков А.С., инженер НИИ «Фотон» РГРТУ; foton@rsreu.ru.*

THE ANALYSIS OF THE SPEED OF SEARCH ALGORITHMS ON IMAGES OF THE CORRESPONDING POINTS USING MODERN COMPUTER

Kuznetsov A.E., Ryzhikov A.S.

The analysis of the speed of search algorithms on images of the corresponding points using modern computing techniques is conducted. The main stages of the work of the correlation-extreme algorithm and the algorithm based on SURF descriptors are considered. A theoretical and experimental evaluation of their performance is carried out using a multi-core CPU and video card. The recommendations on the choice of the finding reference points algorithm for the technology of automatic control of ERS materials geodesic reference are formulated.

Key words: points coregistration, correlation, SURF, graphics card.

Ключевые слова: идентификация одноименных точек, корреляция, SURF, видеокарта.

Введение

Базовым элементом технологии наземной обработки изображений от спутниковых систем высокого пространственного разрешения является контроль геодезической привязки маршрутов съемки. Эта операция выполняется путем распознавания на анализируемом и опорном изображениях одноименных объектов и измерения их координат [1, 2].

Известны два подхода к поиску одноименных точек на паре снимков. Первый, традиционный, основан на корреляционном сопоставлении фрагментов анализируемого снимка с множеством его образов в зоне поиска другого. Второй подход предполагает сопоставление дескрипторов локальных особенностей анализируемого и опорного изображений. Поскольку маршрут съемки представляет собой массив данных сверхбольшого объема (десятки и сотни гигабайт), то предъявляются высокие требования по скорости выполнения операции отождествления на снимках одноименных объектов. В этой связи возникает задача исследования быстродействия рассмотренных алгоритмов на современной вычислительной технике, включающей многоядерный центральный процессор (ЦП, CPU) и видеокарту (GPU). Результатом такого исследования будут рекомендации по выбору алгоритма поиска опорных точек и его практической реализации в технологии автоматического контроля геопривязки материалов съемки.

Архитектура современной ЭВМ

Современный ЦП состоит из нескольких ядер (1 – 36), оптимизированных для обработки последовательных инструкций с максимальной производительностью.

Выполнено исследование быстродействия алгоритмов поиска на изображениях одноименных точек с использованием современной вычислительной техники. Рассмотрены основные этапы работы корреляционно-экстремального алгоритма и алгоритма на основе SURF-дескрипторов. Проведена теоретическая и экспериментальная оценка их производительности при реализации на многоядерном центральном процессоре и видеокarte. Сформулированы рекомендации по выбору алгоритма поиска опорных точек для технологии автоматического контроля геодезической привязки материалов ДЗЗ.

GPU строится на основе масштабируемого массива потоковых мультипроцессоров и имеет тысячи упрощенных и энергоэффективных ядер, оптимизированных для массивно-параллельной обработки данных. Блоки векторных вычислений ЦП реализуют принцип SIMD (одна инструкция выполняется над многочисленными данными), а GPU – SIMT (одна инструкция и несколько потоков). SIMT не требует векторизации данных и допускает произвольные ветвления в потоках. Упрощенная архитектура ЭВМ приведена на рис. 1.

Скорость доступа к данным в памяти GPU превосходит или соответствует скорости доступа к ОП, а чтение/запись разделяемой памяти (Shared Mem) на несколько порядков быстрее. Выполнение расчетов на GPU эффективно, когда одну и ту же последовательность математических операций применяют к большому объему данных. При этом лучшие результаты достигаются, если отношение числа арифметических инструкций к числу обращений к памяти достаточно велико. Это предъявляет меньшие требования к управлению исполнением, вырожденному в GPU, а высокая плотность арифметических инструкций и большой объем входных данных уменьшает ценность больших кэшей, присутствующих в CPU.

С учетом этих особенностей, исследование быстродействия анализируемых алгоритмов будем выполнять применительно к их реализации на CPU и GPU.

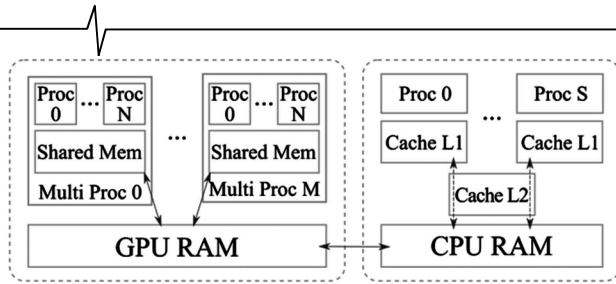


Рис. 1. Архитектура современной ЭВМ

Алгоритмы поиска одноименных точек

Алгоритм пирамидального корреляционно-экстремального (КЭ) поиска и алгоритм, основанный на дескрипторах SURF (далее алгоритм SURF), подробно рассмотрены в работах [2, 3]. Здесь приведем лишь краткое их описание.

Алгоритм поиска одноименных точек на основе корреляционного отождествления состоит из следующих шагов:

Формирование множества координат точек $C = \{c_i\}$, $c_i = \{x_i, y_i\}$ анализируемого изображения $b(x, y)$, для которых необходимо найти одноименные на опорном изображении $b'(x', y')$.

Определение для каждой точки c_i зоны поиска – множества возможных координат одноименной точки в системе координат опорного изображения $C'_i = \{c'_{it}\}$, $c'_{it} = \{x'_{it}, y'_{it}\}$.

Принятие решения о отождествлении всех пар (c_i, c'_{kt}) , для которых выполняется условие:

$$\begin{cases} k = \arg \max_i r(c_i, c'_{it}) \\ r(c_i, c'_{kt}) > r_{\min} \end{cases}$$

где $r(c_i, c'_{it})$ – функция, определяющая нормированную корреляцию фрагмента анализируемого и опорного изображения с центрами в точках с координатами c_i и c'_{it} соответственно; r_{\min} – пороговое значение корреляционной функции.

В работах [3, 4], предложены подходы, позволяющие снизить вычислительную сложность и повысить надежность идентификации одноименных точек в условиях значительных координатных рассогласований и сюжетных различий анализируемого и опорного изображения. Эти оптимизации прямо влияют на способ организации вычислений.

Число операций сопоставления фрагментов при корреляционно-экстремальном отождествлении составляет $T_1 = (M'-M+1) \cdot (N'-N+1)$, где M' – высота фрагмента, N' – ширина, M' и N' – высота и ширина зоны поиска соответственно. Квадратичный рост числа операций с увеличением зоны поиска ограничивает возможность непосредственного применения алгоритма при больших взаимных рассогласованиях снимков. Для повышения быстродействия процесса идентификации образа фрагмента в большой зоне поиска применяется механизм пирамидального представления изображений, позволяющий осуществить последовательное уточнение начальных приближений. Пирамида представляет собой последовательность изображений, в которой каждое последующее изображение является прорежен-

ной вдвое копией предыдущего. Очевидно, что на i -ом уровне пирамиды область поиска сокращается в $2^{2(i-1)}$ раз. Число операции сопоставления фрагментов соответственно составит $T_2 = \sum_i \frac{T_1}{2^{2(i-1)}}$. В работе [3] отме-

чено, что оптимальное число уровней пирамиды 2 – 3. Увеличение числа уровней не имеет смысла, т.к. при каждом прореживании происходит снижение информативности изображений, что повышает вероятность попадания в ложный локальный экстремум корреляционной функции при поиске образа заглубленного фрагмента.

Рассмотрим теперь основные шаги алгоритма SURF. Обозначим $\tau(b, c_i)$ – критерий, определяющий, является ли точка c_i характерной точкой изображения $b(x, y)$. Дескриптором G_{c_i} точки c_i назовем вектор чисел, характеризующий распределение интенсивностей пикселей фрагмента с центром в точке c_i . В качестве меры близости дескрипторов точек c_1, c_2 применяется Евклидово расстояние $d(G_{c_1}, G_{c_2})$.

Алгоритм поиска одноименных точек на основе дескрипторов SURF состоит из следующих шагов:

Формирование множества точек $C = \{c_i\}$ анализируемого изображения $b(x, y)$, для которых выполняется $\tau(b, c_i)$, и множества соответствующих дескрипторов $F = \{G_{c_i}\}$.

Формирование множества точек $C' = \{c'_i\}$ опорного изображения $b'(x', y')$, для которых выполняется $\tau(b', c'_i)$, и множества соответствующих дескрипторов $F' = \{G_{c'_i}\}$.

Формирование множества пар одноименных точек $O = \{(c_i, c'_i)\}$, для которых выполняется условие

$$\begin{cases} \forall G_{c_k} \in F \rightarrow d(G_{c_k}, G_{c'_i}) > d(G_{c_i}, G_{c'_i}) \\ \forall G_{c'_k} \in F' \rightarrow d(G_{c_i}, G_{c'_k}) > d(G_{c_i}, G_{c'_i}) \end{cases}$$

Авторами SURF предложен ряд оптимизаций вычислительного процесса [5].

При детектировании характерных точек для вычисления элементов матрицы Гессе и фильтра Хаара используется интегральное представление изображения, по которому можно быстро найти сумму яркостей пикселей в произвольной прямоугольной области изображения.

Традиционно элементы матрицы Гессе вычисляются путем свертки фрагмента изображения с фильтрами, изображенные на рис. 2 а. В детекторе SURF применяется аппроксимация лапласиана гауссиан (Fast-Hessian), рассчитываемая с использованием фильтров, изображенных на рис. 2 б. Такие свертки эффективно вычисляются с привлечением интегрального изображения.

Для быстрого сопоставления дескрипторов анализируемого и опорного изображения строится древовидная структура, как показано на рис. 3. Далее производится вычисление меры близости только для пар дескрипторов, обладающих одинаковым значениями знака Гессиа-на, масштаба и сходными направлениями градиента.

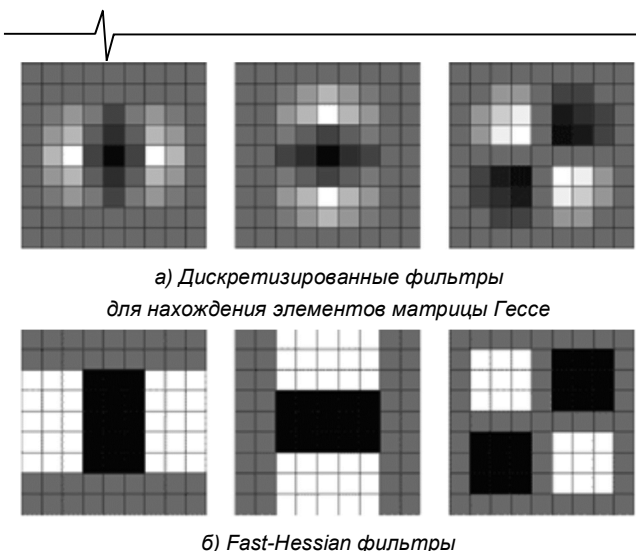


Рис. 2. Фильтры для вычисления лапласиана гауссиан

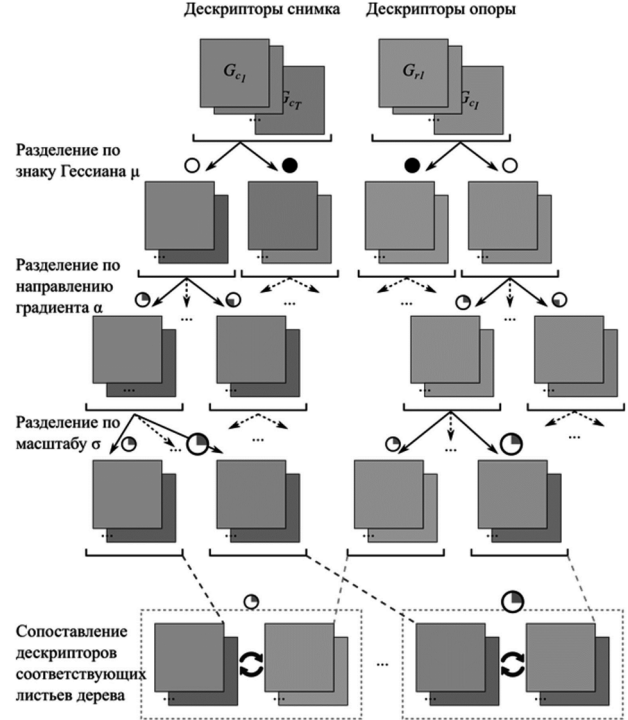


Рис. 3. Дерево сопоставления дескрипторов

Оценка быстродействия алгоритмов поиска одноименных точек

Выполним оценку количества тактов процессора Intel с архитектурой Skylake при однопоточном выполнении векторизованного (AVX) кода, необходимых для поиска пары одноименных точек корреляционно-экстремальным отождествлением и с использованием SURF-дескрипторов. Примем допущение об отсутствии кэш-промахов и накладных расходов на векторизацию данных. Процесс поиска характерных точек в обоих случаях рассматривать не будем.

Пусть размер фрагмента составляет 128 пикселей, допустимое рассогласование координат 512 пикселей. Число операций сопоставления фрагментов при наличии трех уровней пирамиды приведено в табл. 1 Таблица (принято допущение, что ошибка определения положения фрагмента на i -ом уровне составляет не более 2 пиксе-

лей на $i-1$ -ом уровне). Число арифметических операций для вычисления корреляционной функции фрагментов размером $M \times N$ приведено в табл. 2. (Опущено вычисление средней яркости, т.к. эту операцию можно свести к расчету среднего значения одной строки/столбца для каждого положения фрагмента). В соответствии с приведенными данными, для обнаружения образа фрагмента необходимо по меньшей мере $5 \cdot (32^2 \cdot 9216 + 64^2 \cdot 64 + 128^2 \cdot 64) \approx 5,3 \cdot 10^7$ операций сложения и $3 \cdot (32^2 \cdot 9216 + 64^2 \cdot 64 + 128^2 \cdot 64) \approx 3,2 \cdot 10^7$ операций умножения целых чисел. В табл. 3 приведено число тактов, за которое выполняется каждая из используемых операций. Общее число тактов для обнаружения образа фрагмента составит не менее $4,3 \cdot 10^7$.

Таблица 1. Параметры пирамидального поиска при КЭ отождествлении

Уровень пирамиды	Размер зоны поиска	Размер фрагмента	Число операций сопоставления фрагментов
3	128x128	32x32	9216
2	72x72	64x64	64
1	136x136	128x128	64

Таблица 2. Число арифметических операций, необходимых для расчета корреляционной функции

Операция	Число операций
+/-	$5MN$
*	$3MN$
$\sqrt{\quad}$	1

Таблица 3. Производительность векторизованных команд архитектуры Skylake

Операция	Число тактов (AVX)
+/-	0,5
*	0,5
$\sqrt{\quad}$	12

Оценим теперь число операций, необходимых для построения дескрипторов и вычисления меры близости между ними.

Дескриптор формируется путем расчета 25 фильтров Хаара в 16 квадрантах. Для расчета фильтра Хаара с использованием интегрального изображения необходимо 12 операций сложения. Таким образом, для построения дескриптора требуется произвести 4800 операций сложения при вычислении фильтров Хаара и 960 операций сложения при обобщении точечных градиентов, всего $4800 + 960 \approx 5,7 \cdot 10^3$ операций сложения, $2,9 \cdot 10^3$ тактов. Вычисление меры близости пары дескрипторов (Евклидово расстояние между 64-компонентными векторами) предполагает 127 операций сложения, 64 умножения и одно извлечение квадратного корня, 121 такт.

При выделении четырех уровней масштаба и четырех секторов направления градиента (рис. 2), дескрипторы анализируемых изображений в лучшем случае будут разбиты на 32 непересекающиеся группы равных размеров. Число операций сопоставления A дескрипто-

ров одного изображения с тем же числом дескрипторов другого изображения составит $32\left(\frac{A}{32}\right)^2 = \frac{A^2}{32}$. В худшем случае, если все признаки, по которым осуществляется выделение групп, вырождены, сопоставление полным перебором потребует A^2 операций.

Примем среднее число операций сопоставления дескрипторов за $\left(\frac{A^2}{32} + A^2\right)/2 = \frac{A^2}{16}$. Тогда число тактов для формирования A дескрипторов каждого изображения и их сопоставления составит $2 \cdot A \cdot (2,9 \cdot 10^3) + A^2/16 \cdot 121$.

Согласно приведенным расчетам, время выполнения КЭ поиска одного фрагмента эквивалентно времени создания и сопоставления $2 \cdot 10^3$ пар дескрипторов характерных точек. Ожидаемая производительность рассмотренных алгоритмов приведена на рис. 4 (ось числа тактов в логарифмическом масштабе).

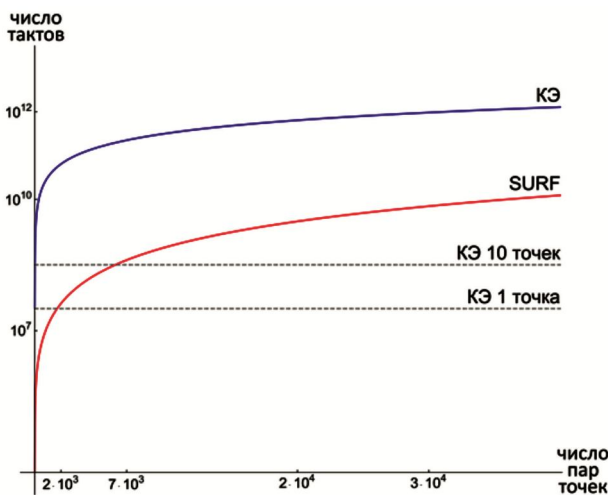


Рис. 4. График теоретической производительности рассмотренных алгоритмов в однопоточной реализации

Полученные оценки быстродействия соответствуют однопоточному выполнению алгоритмов поиска одноименных точек на центральном процессоре. Однако средства современной вычислительной техники обладают многоядерным центральным процессором и графическим сопроцессором (видеокартой). Точная теоретическая оценка быстродействия алгоритмов на таких устройствах практически невозможна в связи со сложным конкурентным доступом к разделяемым данным и многоуровневой иерархией кэш-памяти.

Для получения экспериментальных данных по быстродействию выполнена многопоточная реализация алгоритмов на языке C++ с использованием примитивов OpenCV. Производительность реализованных алгоритмов тестировалась на конфигурации CPU Intel Xeon 2.4 Ghz x 36, GPU NVIDIA Tesla k20c (706 MHz x 2496).

Для КЭ алгоритма в каждом потоке параллельно выполняется построение пирамидального представления эталона и фрагмента зоны поиска и КЭ поиск. График, иллюстрирующий быстродействие многопоточной реализации в зависимости от числа потоков, приведен на рис. 5.

Из графика следует, что увеличение числа задействованных для вычисления ядер центрального процессора не приводит к пропорциональному ускорению. Алгоритм КЭ отождествления масштабируется слабо. Вероятно, это связано с тем, что время, затрачиваемое на обращение к данным в ОП, преобладает над затратами на собственно вычисления. С ростом числа одновременно выполняющихся потоков эта особенность приводит к большому числу кэш-промахов и не позволяет задействовать все ресурсы CPU.

Изучена возможность реализации алгоритма КЭ отождествления на GPU. Проведенные эксперименты выявили принципиальную проблему. Для определения истинного положения фрагмента необходимо на завершающем этапе выполнить поиск максимального / минимального значения корреляционной функции, что сводится к последовательному проходу по всем пикселям раstra, определяющего величину КЭ функции в каждой точке зоны поиска. Проведение этой последовательной по своей природе операции «на месте», в GPU, крайне неэффективно, т.к. при этом простаивает большая часть ядер. Пересылка раstra на CPU и дальнейший поиск экстремумов на ресурсах ЦП дала прирост скорости 40%. Тем не менее это на порядок более медленная, чем вычисление КЭ функции в памяти GPU, операция определяет верхнюю границу скорости обработки данных. Анализ реализованной программы показал, что 81% времени уходит на пересылку данных CPU ↔ GPU, а быстродействие в целом эквивалентно реализации КЭ поиска в 8 потоков на CPU.

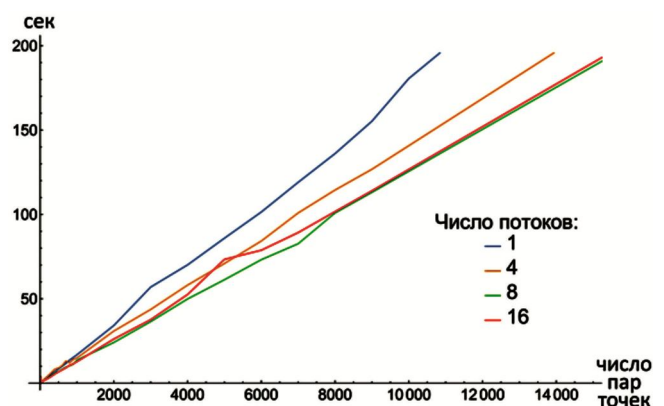


Рис. 5. График быстродействия многопоточной реализации КЭ алгоритма

В многопоточной реализации алгоритма SURF на центральном процессоре построение пирамиды разностей гессиан производится последовательно, по слоям. Параллельно выполняется детектирование характерных объектов и построение их дескрипторов, при этом удается задействовать все 36 ядер процессора. Сопоставление дескрипторов из соответствующих листьев дерева каждого изображения так же проводится в отдельных потоках с эффективным использованием 8 – 16 ядер процессора в зависимости от структуры дерева дескрипторов.

В отличие от GPU-версии алгоритма на графическом процессоре организовано параллельное вычисление гессиан и их сглаживание, детектирование характерных объектов и построение дескрипторов. Сопоставление дескрипторов выполняется параллельно на центральном процес-

соре, как и в CPU реализации. Результаты выполнения многопоточной реализации алгоритма SURF и КЭ (ЦП, 8 потоков) приведены на рис. 6 (ось времени представлена в логарифмическом масштабе). Столь высокой производительности алгоритма SURF удалось добиться в первую очередь за счет высокой локальности и малого объема данных, необходимого для формирования дескриптора. Это позволяет естественным образом параллельно рассчитывать большое их количество с использованием быстродействующей кэш памяти процессора (CPU-версия) или разделяемой памяти блоков (GPU).

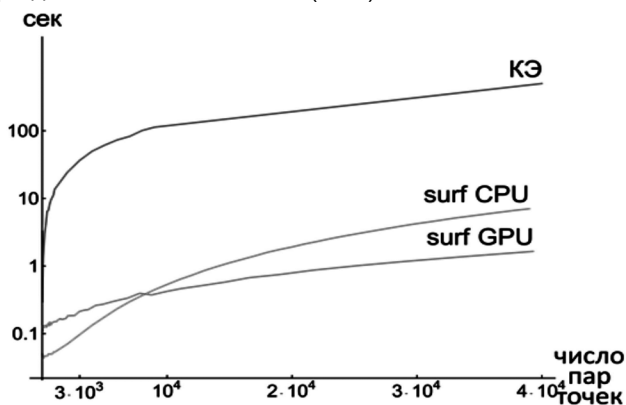


Рис. 6. График быстродействия многопоточной CPU и GPU реализации рассмотренных алгоритмов

Таблица 4. Число точек, сопоставленных за фиксированное время

Длительность, с	Число сопоставленных пар точек		
	КЭ, 8 ядер	SURF, CPU	SURF, GPU
1	68	14170	24940
2	159	20400	45700
3	198	25500	60000

Вывод: соотношение производительности многопоточных реализаций алгоритмов КЭ и SURF сильно коррелирует с теоретически рассчитанными величинами для однопоточных вариантов. Многопоточный алгоритм SURF, реализованный на GPU, производительнее многопоточной CPU-версии более чем в 2 раза. Число пар точек, сопо-

ставленных за фиксированное время в 1, 2 и 3 секунды реализованными алгоритмами, приведено в табл. 4.

Заключение

Алгоритм, основанный на дескрипторах SURF, может быть эффективно распараллелен на современной вычислительной технике. Его массивно-параллельная реализация как на многоядерном центральном процессоре, так и с привлечением ресурсов видеокарты, позволяет обеспечить формирование массивов координат одноименных точек со скоростью, на два порядка превышающей скорость корреляционно-экстремального поиска, так же реализованного параллельно. Другой проблемой, определяющей возможность практического применения алгоритма SURF, остается вопрос надежности сопоставления. Это требует дополнительного исследования, которому планируется посвятить следующую работу.

Литература

1. Кузнецов А.Е. Конвейерные технологии потоковой обработки данных от высокодетальных систем ДЗЗ// Тез. докл. четвертой международной науч.-техн. конф. «Актуальные проблемы создания космических систем дистанционного зондирования земли» – М.:АО «Корпорация «ВНИИЭМ», 2016. – 220 с.
2. Кузнецов А.Е., Пошехонов В.И., Рыжиков А.С. Технология автоматического контроля точности геопривязки спутниковых изображений по опорным снимкам от КА «Landsat-8» // Цифровая обработка сигналов. – 2015. – № 3. – С. 37-42.
3. Кузнецов А.Е., Светелкин П.Н. Оперативное совмещение спектральнональных изображений при цветосинтезе // Вестник Рязанского государственного радиотехнического университета. Выпуск 22, 2007. С. 3-7.
4. Прэтт У. Цифровая обработка изображений. В 2-х кн.: пер. с англ. М.: Мир, 1982. Кн.1: 312 с., кн.2: 480 с.
5. H. Bay, T. Tuytelaars, L. Van Gool, Surf: Speeded up robust features // Computer vision—ECCV. – 2006. – С. 404-417.

НОВЫЕ КНИГИ



Витязев В.В. Многоскоростная обработка сигналов – М.: Изд-во «Горячая линия-Телеком», 2017 г. – 336 с.: ил.

Рассмотрена эволюция теории и технологий многоскоростной обработки сигналов в период с начала 70-х гг. прошлого столетия до наших дней с позиции вклада, который внесли в их развитие работы российских ученых и специалистов в области цифровых информационных технологий реального времени. Описаны методы и алгоритмы многоступенчатой и многокаскадной реализаций цифровых узкополосных фильтров и банков цифровых фильтров на основе эффектов прореживания по времени (децимация во временной области) и по частоте (децимация в частотной области).

Приведена методика оптимизации параметров многоступенчатых и многокаскадных структур цифровых полосовых фильтров. Построение оптимальных структур и расчет параметров фильтров частотной селекции иллюстрируется многочисленными примерами.

Для специалистов, научных работников, преподавателей вузов, аспирантов; будет полезна студентам информационных и инфокоммуникационных направлений подготовки.