Современные проблемы дистанционного зондирования Земли из космоса.
2012. Т.9. № 1. С. 32–38

Математическая модель бортового процессора для субпиксельной обработки данных ДЗЗ с целью повышения разрешающей способности ЦКИ

С.В. Блажевич[1], В.Н. Винтаев[2], А.Л. Греков[1], А.В. Секирин[1], Н.Н. Ушакова[2]

[1]Белгородский государственный университет
ул. Победы, д.85, 308015, г. Белгород, Россия (4722)-30-11-98
blazh@bsu.edu.ru
[2]Белгородский университет потребительской кооперации
ул. Садовая, 116 а, г. Белгород, Россия (4722)-26-38-31
viktor.vn2010@yandex.ru
natush2006@yandex.ru

В работе представлены результаты математического моделирования процессора, ориентированного на решение задачи синтеза цифровых изображений на основе субпиксельной технологии с использованием методов повышения эффективности и производительности расчетов для применения процессора на борту космического аппарата.

Ключевые слова: бортовой процессор, обработки цифровых изображений, синтез цифровых космических изображений, субпиксельное сканирование.

Введение

При разработке программно-аппаратного бортового комплекса, решающего задачу синтеза одного улучшенного по четкости и разрешению изображения из нескольких изображений одного и того же участка местности, сдвинутых относительно друг друга в соответствии с требованиями субпиксельной технологии [1] на доли апертуры пиксела, основной процедурой является решение системы линейных уравнений, связывающей распределение яркости по пикселам в комплексе исходных изображений с искомым распределением по пикселям уменьшенной апертуры в синтезируемом изображении. Необходимым условием разрешимости системы уравнений относительно неизвестных – набора значений пикселов синтезируемого изображения, является равенство количества пикселов в синтезируемом изображении суммарному количеству пикселов в исходных изображениях. Применяя сквозную нумерацию пикселов синтезированного изображения и пикселов в упорядоченном наборе исходных изображений, получим простую формулировку задачи в виде системы линейных уравнений, в которой число неизвестных и количество уравнений может быть, например, 10000х10000 для синтезируемого изображения размером в сотню мегабайт для каждого из трех цветов популярной модели RGB реалистичной палитры в 24 разряда. С таким представлением работать удобнее, чем использовать двухиндексную нумерацию искомых пикселов и двухиндексную нумерацию пикселов – правых частей системы уравнений с дополнительным третьим индексом принадлежности к какому либо исходному изображению. Просто необходимо при формировании изображений после выполнения вычислений обращаться к таблицам соответствий принадлежности номеров пикселов в сквозной нумерации соответствующей строке изображения, что определяется значением кратности деления общего числа пикселов на длину строки. В такой системе уравнений все пикселы синтезируемого изображения включаются в каждое уравнение, при этом далеко не все коэффициенты уравнения не раны нулю, более того, матрица коэффициентов системы уравнений для случая субпиксельного сканирования [1] представляет собой ленточную матрицу, собранную по главной диагонали из подматриц размера 2х2, 3х3 и т.д. в зависимости от соотношения апертур исходных и синтезируемых пикселов.

Математическая модель процессора

При аналитическом сравнении методов решения указанной системы (метод обращения матриц, Гаусса, Гаусса-Зейделя,), было выяснено, что метод Гаусса-Зейделя имеет преимущества в связи с тем, что для случая решения системы уравнений по технологии [1] может быть построен на вычислениях кросс-сверток строк и столбцов [2], т.е. позволяет активно применять процедуры быстрого преобразования Фурье и добиваться высокой степени снижения вычислительной сложности алгоритма. Но есть еще одно существенное преимущество у этого метода – метод Гаусса-Зейделя это рекуррентное решение системы уравнений, при том на каждом шаге с все более улучшенным или уточненным результатом, представляемом в виде полного изображения. Если существует точное решение системы, то и метод Гаусса-Зейделя позволяет его найти. Если не существует, то легко сформулировать критерий остановки процесса решения системы в общеизвестном виде – модуль попиксельной разности изображений на соседних шагах процесса не должен превышать некоторого значения, но весьма существенно то, что на каждом этапе можно «видеть», что получается и если необходимо, можно корректировать ход решения.

В соответствии методом Гаусса-Зейделя, если задана система линейных уравнений

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1; \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{1n}x_1 + a_{2n}x_2 + \ldots + a_{\boldsymbol{n}}\ x_n = b_n, \end{cases} \tag{1}$$

то сначала ее представляют в виде:

$$\begin{cases} x_1 = \dfrac{1}{a_{22}}(b_1 - a_{12}x_2 - a_{13}x_3 - \ldots - a_{1n}x_n); \\ x_2 = \dfrac{1}{a_{22}}(b_2 - a_{21}x_2 - a_{23}x_3 - \ldots - a_{2n}x_n); \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_n = \dfrac{1}{a_{nn}}(b_n - a_{n1}x_2 - a_{nn-2}x_{n-2} - \ldots - a_{\boldsymbol{n}\ -1}x_{n-1}); \end{cases} \tag{2}$$

тогда решение системы производится по формулам

$$\begin{cases} x_1^{k+1} = \dfrac{1}{a_{11}}(b_1 - a_{12}x_2^{\ k} - a_{13}x_3^{\ k} - \ldots - a_{1n}x_n^{\ k}); \\ x_2^{k+1} = \dfrac{1}{a_{22}}(b_2 - a_{21}x_2^{\ k} - a_{23}x_3^{\ k} - \ldots - a_{2n}x_n^{\ k}); \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_n^{k+1} = \dfrac{1}{a_{nn}}(b_n - a_{n1}x_2^{\ k+1} - a_{nn-2}x_{n-2}^{\ \ k+1} - \ldots - a_{\boldsymbol{n}\ -1}x_{n-1}^{\ \ k+1}); \end{cases} \tag{3}$$

где x_i^k, x_i^{k+1} – значения искомых переменных соответственно на k-том и $k+1$-ом шаге.

При разработке алгоритмов, с последующей разработкой программ для универсальных или специализированных или проблемно-ориентированных процессоров стратегия снижения вычислительных сложностей алгоритмов позволяет сократить количество выполняемых операций арифметико-логического класса в десятки, сотни и даже сотни тысяч раз, в соответствии с размерами решаемых задач и исходными сложностями первичных алгоритмов [2, 3]. Применяя методы распараллеливания или (и) конвейеризации аппаратных средств поддержки и замечая, что количество аппаратных операционных единиц в вычислителе (точнее в построенном на указанных принципах коллективном вычислителе) помноженное на количество оставшихся после распараллеливания и конвейеризации последовательных вычислений как шагов выполнения алгоритма есть величина постоянная для данной вычислительной сложности реализуемого алгоритма, выраженной в элементарных операциях, которые в свою очередь могут быть реализованы таблично, выборками из памяти, можно количество параллельных ветвей (аппаратных затрат) минимизировать применением стратегии снижения сложности самого алгоритма. Выборки уместно рассматривать и реализовать для выравнивания длительности выполнения всех операций независимо от их собственных сложностей (измеренных, скажем, в количестве булевых операций, необходимых для построения каждой из операций). В целом, целесообразно для всех элементарных операций (в алгебрах: числовой, векторной, матричной – это операции сложения, умножения), применяемых для представления алгоритма, принять следующее положение:

элементарная операция – это всегда таблица двухместной операции (подобная таблице истинности в алгебре Буля) и алгоритм использования этой таблицы, если размер решаемой в этой операции задачи превышает размер таблицы, задающей операцию. Все это в принципе реализуется в известных нам арифметиках: многоразрядные и сумматоры, и умножители обычно набираются из одноразрядных, которые и задаются соответствующими таблицами истинности (порождающими нормальные конъюнктивно-дизъюнктивные формы логического описания устройств), при этом можно одной таблицей подобно школьным вычислениям реализовать последовательно выполнение всей многоразрядной операции или привлечь различные формы параллелизма в вычислениях при посредстве нескольких таблиц, удовлетворяя при этом закону постоянства произведения числа параллельных ветвей коллективного вычислителя на число последовательных вычислений в соответствии описанному выше.

В [4, 5] показано, что наиболее употребительные при обработке изображений алгоритмы класса Фурье преобразований, сверток, линейных фильтраций, реализуемые, если можно так выразиться в классической технологии в алгебре с операциями «сложить» и «умножить», более эффективно осуществляются на основе табличных операций, таких, например, как двухточечное преобразование Фурье, Адамара, Уолша, табличная двухточечная или четырехточечная свертка и так далее. При этом оказывается, что вычислительное устройство, реализующее вычислительные процессы на основе «классической» арифметики, работает на порядки эффективнее, если в этой арифметической системе заменить двухместную операцию умножения на перечисленные выше билинейные, то есть удовлетворяющие условиям дистрибутивности и тому подобным условиям табличные операции. В [2] показано, что возврат вычислителя в традиционную арифметическую систему не только реализуется элементарно, но и обеспечивается при этом гораздо более эффективная реализация той же операции умножения. Примером тому наглядная и красивая теорема Шёнхаге-Штрассена [2] и теорема о замене классического алгоритма умножения целых чисел (алгоритма сложности n^2) тремя быстрыми преобразованиями Фурье (БПФ) (с вычислительной сложностью $3n\log_2 n$). В [3] показано, что стратегию Шёнхаге-Штрассена можно распространить и на построенный обобщенный вариант операции умножения – на обобщенную бинарную билинейную операцию (ОББО), причем распространить более эффективно, сократив некоторое количество не совсем уместных шагов при таком переходе: выразив обобщенную операцию в виде свертки двух векторов (сомножителей), умноженной скалярно на специфический нормирующий вектор. Использование системы остаточных классов не только для преобразования операции умножения, но и преобразования процедуры свертки, подобно описанному выше применению БПФ позволяет сводить вычислительные сложности алгоритмов с уровня n^2

до уровня для $const \cdot n$ одномерных задач размера n, и с уровня n^4 до уровня n^2 для двумерных задач, которыми в подавляющем большинстве являются задачи поточной обработки изображений [6, 7].

Пусть в «традиционной» арифметической системе построен некоторый алгоритм (например, из упомянутого класса алгоритмов), имеющий конечную длину n и являющийся двухместным на входе, с подачей дополнительных входов на некоторых этапах. Если группы последовательных описываемых этапов по длине (по числу операций) равны нулю, то эта ситуация приводится к эффективному этапу, определяющему некоторую k-местную операцию и этот этап может быть представлен комбинационно на базе совокупности двухместных операций, в древовидном графе, как это принято в теории и практике ЭВМ. Алгоритм, для которого подбираются операции более компактного его выполнения, приводится (если это удается сделать) к виду с минимально возможными показателями вычислительной сложности. Следует отметить, что общего плана рекомендаций и методик для любых алгоритмов не существует, однако и процессы минимизации дизъюнктивных форм, и методы оптимизации, и собственно стратегии снижения вычислительных сложностей, да и вся история математики, посвященная не только выводам основных теорем и положений, но и приведению различных задач и их решений к эффективной форме [8, 9] в подавляющем большинстве случаев для задач обработки изображений позволяют достичь неплохих результатов, поднимая путем снижения вычислительных сложностей задач эффективную производительность процессоров без увеличения их тактовой рабочей частоты на высокий уровень. Известно, что любую многоместную операцию и любой многовходовый алгоритм можно представить в виде каскадов двухместных операций и двухвходовых алгоритмов, соответвтвенно. При этом для списков операций и алгоритмов, обеспечивающих математическое представление задач поточной обработки изображений, разработаны формы их табличного представления [3]. При использовании таких форм, во-первых, длинные фрагменты исходного алгоритма просто заменяются заранее просчитанными таблицами с однотактным их выполнением, во- вторых, исходный алгоритм оказывается представлен в операциях не представляющих собой операции традиционных алгебр.

При этом стоит отметить, что наращивание размера задачи по реализации, скажем, БПФ от четырехточечной до 32-точечной осуществляется более экономично [5], в смысле наращивания вычислительных затрат, при использовании табличной четырехточечной операции БПФ, нежели прямая реализация 32-точечной процедуры.

Имеет смысл различать три вида постановок задач на синтез или разработку оптимальных операций:

- разработка табличных операций, реализующих минимально возможные наращивания вычислительной сложности алгоритмов или (и) групп алгоритмов при наращивании размеров задач;
- разработка табличных операций, обеспечивающих минимально возможное наращивание вычислительных сложностей при расширении групп алгоритмов;
- разработка табличных операций, обеспечивающих минимально возможное наращивание вычислительных сложностей при наращивании размеров задач и расширении номенклатуры алгоритмов одновременно.

Таким образом, рассматривая операционные элементы, реализующие элементарные операции как элементарные машины (ЭМ) и не акцентируясь на том, аппаратно или программно (или аппаратно-программно) выполнена отдельная ЭМ (такое послабление и обобщение вопроса нам предоставляет декларированное выше положение о постоянстве и равенстве вычислительной сложности реализуемой операции произведения числа аппаратных ЭМ (в концепции параллелизма) на число последовательных операций), мы можем рассматривать реализуемую операцию как распределенную (в пространстве или (и) времени, то есть в четырехмерном пространстве) вычислительную среду, рассмотрению которой были посвящены работы [10].

При этом вопросы надежности, живучести, реконфигурации, оптимального размещения задач в вычислительной среде (распределенной) практически не связаны с выбором обсуждаемых операций, и исследованы достаточно глубоко как в теоретическом, так и в практическом плане [9], для распределенных вычислительных сред с ЭМ, список, состав, структура и взаимодействие операций в которых могут быть произвольными.

Концепция оптимальности алгоритмов и операций, рассматриваемая ниже, определяет постановки задач на оптимизацию (в частности, параметров, определяющих и задающих алгоритм и операции), вызванную проблемой верификации, описанной в [11] и заключающейся в том, что алгебры, с дискретной сигнатурой, с фиксированными операциями, не обеспечивают полноты пространства операций. Варьирование параметрами фильтров, например, в задачах фильтрации сигналов, изображений на фоне помех и шумов, исследуемое уже на протяжении достаточного количества лет, более того постановки и решения вариационных задач [12] дают красивые и действенные результаты в оптимизации соотношений вида сигнал \ шум и тому подобных, а также различных параметров задач, но оставляют решения задач в гиперплоскости, отличающейся от гиперплоскости решений с участием варьирования операций алгебр, привлекаемых к решаемым задачам. Речь идет, прежде всего, о классической малой вариации операций алгебр. Решение таких задач в общем виде достаточно близко перекликается с теорией возмущения операторов, но при переходе к вычислимым, реализуемым на ЭВМ задачам, все сводится к алгоритмам в арифметических системах с классическими фиксированными операциями. Тогда как любой такой алгоритм, то есть вычислимое представление задачи, записанный (реализованный) в возмущенных операциях, скажем умножения, и сложения, не есть линейное преобразование исходного алгоритма и не получается как процедура, записанная «другим программистом» по-другому. Исходные алгоритмы, записанные в возмущенных на произвольную малую величину (норму операции) операциях, порождают самостоятельную, если можно так выразиться, и совершенно невыводимую из предыдущих теорем гиперплоскость алгоритмов, свойства которых являются, если использовать терминологию объектно-ориентированного программирования наследными от свойств исходных алгоритмов. Но чтобы варьировать на малый параметр операцию, скажем, умножения, необходимо такую операцию, прежде всего, построить.

Варианты построения такого рода операций реализованы в [3, 13], экспериментальные результаты использования компьютерных арифметик на базе указанных операций при обработке изображений говорят о расширении возможностей по решению задач фильтрации, распознавания и улучшения репрезентативных свойств изображений.

Следующей задачей при разработке процессора является экономия потребляемой энергии, объема, веса при создании бортового спецпроцессора, противоречащая принципам резервирования (мажорирования и т.п.), которая для обеспечения заданного уровня живучести может быть реализована путем снижения числа выполняемых операций (в единицу времени) на единицу объема данных при выполнении алгоритма обработки.

При работе над указанным направлением применялась следующая методика исследований:

- математическое моделирование алгоритма обработки данных (в среде программирования С++);
- разработка эффективных (быстрых) модификаций алгоритма и табличных (однотактных) реализаций операций и модулей алгоритма;
- компьютерное имитационное моделирование работы блоков процессора в варьируемой фоноцелевой обстановке;

Алгоритм обработки данных представляет собой конвейер:

- цифровая нормализация передаточных характеристик ячеек сенсорного поля кусочно-линейной аппроксимацией по массиву задаваемых тангенсов углов их наклона на интервалах аппроксимации $(tg\alpha_{ij})_k$ и смещений $(b_{ij})_k$ для компенсации дисперсий и уходов параметров ПЗС, реализации функционального преобразования двумерного поля данных [14]: установки порогов, введения компенсационных констант расширения динамического диапазона, управления средним корреляционным коэффициентом передачи (усиления);

- деконволюция изображения на матричном транспаранте с функцией регрессии точки задаваемой параметрами предварительной расфокусировки сцены, проецируемой на ПЗС [15]. Расфокусировка проецируемой на матрицу ячеек ПЗС сцены поддерживает устойчивость работы сенсорного поля при отказе его ячеек за счет соответствия координат локализуемых объектов;
- решение системы линейных уравнений по основной задаче процессора с выполнением всех видов сверток на блоках БПФ на основе комплексирования полноразмерных БПФ из малоточечных табличных БПФ.

Процессор, соответствующий предлагаемой вычислительной модели может быть организован в конвейерно-параллельную структуру [16] из идентичных соответствующих конвейеру алгоритма вычислителей, входы и выходы которых соединены с лифтным (конвейерным тактируемым) коммутатором, обеспечивающим управление конфигурацией вычислителя и коммутацией его на сенсорное поле ПЗС.

Литература

1. *Блажевич С.В., Винтаев В.Н. и Ушакова Н.Н.* Синтез космического изображения с улучшенной разрешающей способностью на основе субпиксельного сканирования // Современные проблемы дистанционного зондирования Земли из космоса, 2010. Т.7. №4. С. 9–13.
2. *Ахо А.* Построение и анализ вычислительных алгоритмов / А. Ахо, Д. Хопкрофт, Д. Ульман.- М.:Мир, 1979, 536 с.
3. *Винтаев В.Н.* Вычислительное устройство на основе проблемно-ориентированной компьютерной арифметики / В.Н.Винтаев.- Диссертация на соиск. уч.ст. канд.техн. наук. Москва, 1989, 118 с.
4. *Прэтт У.* Цифровая обработка изображений / У.Прэтт.- М.:Мир, Кн.2, 1982, 790 с.
5. *Каппелини В.* Цифровые фильтры и их применение/В. Каппелини, А.Д. Константинидис, П. Эмилиани.- М.: Энергоатомиздат, 1983, 360 с.
6. *Zhang Y.* Processing Logic of the Radar Map Ensuring High-Res in a Millimeterwave / Y.Zhang, L. Xing-Guo // Hongwai yu haomibo.J.Infrared and Millimeter Waves, 1998.-17.-№4.-P.293–298.
7. *Makarov O.N.* On the Realization Between the Fast Fourier and Hadamart Transform Algorithms of Karatsuba, Vinograd and Strassen / O.N. Makarov // 15 Zh. Vych. Mat.Fis. July 1975.- P. 1095–1105.
8. *Казмирчук А.А.* Методы восстановления изображений по блокам двумерного спектра / А.А.Казмирчук // Распараллеливание обработки информации: Сб. докладов VI Всесоюзн. шк.-семинара, Львов, 1987.- Ч.2.- С. 71–72.
9. *Морен К.* Методы Гильбертова пространства / К.Морен.- М.:Мир, 1965, 570 с.
10. *Евреинов Э.В., Хорошевский В.Г.* Однородные вычислительные системы. - Новосибирск, 1978, 318 с.
11. *Винтаев В.Н., Уразбахтин А.И., Ушакова Н.Н.* Критерий допустимости необходимой деградации разрешающей способности изображений, синтезируемых некоторыми радиотехническими системами // «Телекоммуникации», теоретический и научно-методический журнал, Москва, 2004, №11, с. 34–36.
12. *Гулд С.* Вариационные методы в задачах о собственных значениях / С.Гулд.- М.:Мир, 1970, 328 с.
13. *Ушакова Н.Н. Коррекция* цифровых космических изображений на основе верифицирующего моделирования. Диссертация на соискание уч. степени кандидата техн. наук. – Курск, 2004, 256 с.
14. *Винтаев В.Н., Гамидов В.В., Исмаилов К.Х.* Задача цифровой нормализации матриц фотоприемных устройств. Сообщения НПОКИ №1. Изд. «ЭЛМ», Баку, 1984, с.
15. *Алиева М.А., Исмаилов К.Х.* Моделирование архитектуры бортового процессора с проблемной ориентацией // Исследование Земли из космоса, №2, 1987, с. 112–117.

Mathematical model of on-board processor for subpixel remote sensing data processing for improvement of CCS resolution

S.V. Blazevich[1], V.N. Vintaev[2], A.L. Grekov[1], A.V. Sekirin[1], N.N. Ushakova[2]

[1]Belgorod State University
str. Pobedy, 85, 308015, Belgorod, Russia (4722) -30-11-98
blazh@bsu.edu.ru
[2]Belgorod University of Consumer Cooperatives
str. Sadovaya, 116 and, Belgorod, Russia (4722) -26-38-31
viktor.vn2010 @ yandex.ru
natush2006@yandex.ru

This paper presents the results of mathematical modeling of the processor which is oriented on solution problem of synthesizing digital images based on sub-pixel technology using methods of increase of the efficiency and productivity counting for the application of the processor on the board of a space vehicle.

Keywords: on board processor, digital image processing, digital satellite image synthesis, sub-pixel scanning.