

РЕАЛИЗАЦИЯ АЛГОРИТМА SPIHT НА СИГНАЛЬНОМ ПРОЦЕССОРЕ

Черников А.В., Плахов А.Г., Чобану М.К.

МОСКОВСКИЙ ЭНЕРГЕТИЧЕСКИЙ ИНСТИТУТ (ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ)

В работе рассматриваются вопросы реализации алгоритма сжатия изображений SPIHT на сигнальном процессоре TMS320C62x фирмы Texas Instrument. Данный алгоритм принадлежит к классу кодеров «с потерями» и работает с вейвлет-декомпозицией сигнала. SPIHT показывает хорошие характеристики качества на больших степенях сжатия, обладая при этом простой структурой. Генерируемый при кодировании поток имеет ряд положительных свойств:

1) возможность точного регулирования скорости передачи, а при записи в файл его размер может быть задан с точностью до байта;

2) возможность восстановления всего изображения при прекращении приема декодером бит в любой точке;

3) возможность быстрого просмотра изображений в удаленной базе данных.

В работе рассмотрены методы повышения скорости кодирования, как алгоритмические, так и программные; разработана аппаратно ориентированная версия кодера, предназначенная для использования в системах на базе сигнальных процессоров.

1. Вейвлет-декомпозиция.

Как известно, свертка с разделимыми фильтрами, широко применяемыми при выполнении вейвлет-декомпозиции, происходит последовательно по строкам, затем по столбцам. В первом случае обращения к памяти за коэффициентами происходят последовательно, что позволяет в полной мере использовать возможности кэш-памяти, а также при необходимости организовывать буфер во внутренней памяти процессора, что значительно ускоряет вычисление свертки. Для случая столбцов обращения к памяти происходят не последовательно, а с шагом равным размеру строки исходного изображения, что не желательно для кэш-памяти, а также требует дополнительных вычислений смещения адреса коэффициента. Также, в случае организации стандартного строкового буфера во внутренней памяти, возникают сложности при расширении сигнала по столбцам, которое необходимо для выполнения вейвлет-декомпозиции.

Решить эти проблемы можно с помощью DMA контроллера процессора, используя его для организации транспонирования группы столбцов исходного изображения из внешней памяти во внутренний строковый буфер. Операция транспонирования позволяет перейти от свертки столбца к свертке строки, что существенно увеличивает быстродействие. С точки зрения повышения качества, появляется возможность простой реализации периодического расширения сигнала по столбцам при использовании внутреннего буфера, объем которого меньше исходного изображения.

2. Алгоритм SPIHT.

Для адаптации алгоритма SPIHT к реализации на сигнальном процессоре используются специально разработанные методики: матрица значимости и безсписковый SPIHT. Под организацию матрицы значимости и хранение рабочих бит алгоритма отводится дополнительная память равная $W \cdot H$ байт, где W и H – ширина и высота изображения соответственно.

В процессе анализа элементов LIS, кодер должен просматривать все дерево в поисках значимых коэффициентов. Этот процесс, неоднократно повторяясь, занимает значительную часть вычислительных ресурсов. Для ускорения процесса кодирования используется матрица значимости (MS).

MS формируется таким образом, что значение каждого ее элемента (i,j) равно числу бит, занимаемому максимальным по модулю коэффициентом дерева, выстроенного из точки (i,j) с учетом корня. Имея в распоряжении такую структуру для анализа элемента типа A достаточно рассмотреть только точки с координатами «детей», обращаясь при этом к MS. Анализируя же элементы типа B необходимо рассмотреть только точки с координатами «внуков» в структуре MS. Это простое инженерное решение позволило значительно сократить время работы кодера и сделало возможным его применение в системах критичных к скорости кодирования, например, таких как обработка видеосигнала в реальном времени.

Разработан простой алгоритм формирования MS, позволяющий за один проход по коэффициентам вейвлет-разложения собрать в матрице значимости информацию по всем деревьям.

Алгоритм SPIHT использует в работе списковые структуры. При реализации в виде приложения списковая организация удобна для программиста и не имеет аппаратных ограничений. Для реализации же на сигнальных процессорах необходимо исключить динамическую работу с памятью. Следовательно, использование списков не желательно.

Безсписковый SPIHT строит иерархические деревья, не используя списковые структуры. Вся информация о ветвях деревьев и о значимых точках хранится в рабочей памяти алгоритма. Обработка строится таким образом, что все необходимые точки (значимые точки, точки входа в дерево) анализируются один раз, как и в списковом алгоритме, благодаря чему переход на безсписковый алгоритм проходит без потерь в быстродействии. Также появляется возможность раздельной обработки деревьев (в том числе параллельной).

Начиная работу с низкочастотной части, алгоритм рассматривает каждую точку, как отдельный коэффициент и как точку входа в дерево. Каждый коэффициент содержит два поля типа: тип «как точка» и тип «как дерево». Сначала коэффициент анализируется как точка: если он стал значим на предыдущих этапах, то передается его текущий бит, иначе выполняется проверка на значимость. Точка может быть двух типов:

IP - незначимая точка (insignificant point);

SP - значимая точка (significant point).

Затем точка анализируется как вершина дерева. Если дерево выстроенное из данной точки было раскрыто на предыдущих этапах, поле типа «как дерево» содержит тип «проход». В таком случае кодер переходит к анализу «детей», которые аналогичным образом обрабатываются как точки, затем как вершины деревьев. Таким образом, последовательно анализируются все «отброшенные» на предыдущих этапах точки. Кодер в этом случае осуществляет движение вглубь дерева. Если поле типа «как дерево» содержит А или В, кодер анализирует значимость дерева выстроенного из данной точки. Если значимо дерево с типом А, дети анализируются как точки, тип дерева заменяется на В и сразу же анализируется как В. В случае значимости дерева с типом В тип заменяется на «проход», «дети» получают тип А и сразу же анализируются.

Каждый байт рабочей памяти алгоритма соответствует точке на вейвлет-декомпозиции и содержит в своих битах:

с 3 по 7 - матрица значимости деревьев;

2 - флаг принадлежности к IP или SP (значимые и не значимые точки);

0,1- тип точки в дереве (А, В, D (от англ. «deeper»)).

Безсписковый алгоритм на границах передачи битовых полей абсолютно совпадает по переданной информации со списковым алгоритмом, сам же процесс передачи битового поля идет несколько в другом порядке. Как показали тесты, это различие вызывает некоторое снижение PSNR. Что можно увидеть на следующем рисунке.

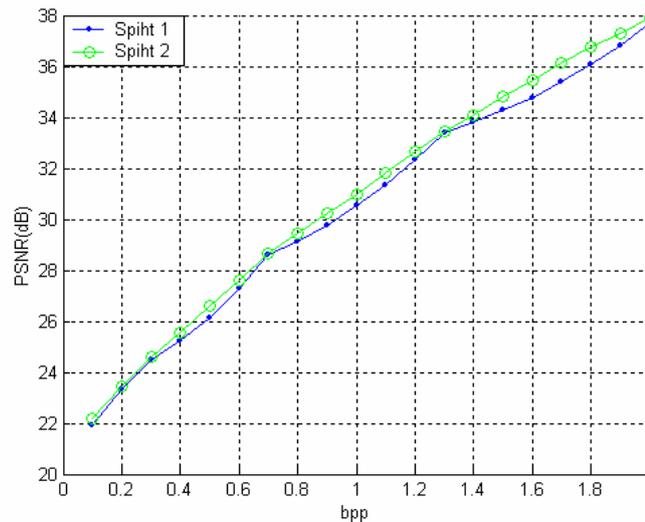


Рис. 1: Тест качества: Spiht 1 - бессписковый алгоритм; Spiht 2 - списковый алгоритм.

Отказ от списковых структур позволяет разрабатывать простую высокоскоростную аппаратную реализацию кодера для его применения в системах обработки изображений.

3. Выводы.

Предлагаемые методики позволяют реализовывать современный алгоритм сжатия SPIHT на сигнальных процессорах. Обозначены решения таких проблем как: декомпозиция столбцов, поиск значимых коэффициентов, отказ от использования динамической памяти.

Литература

1. В.И. Воробьев и В.Г. Грибунин. "Теория и практика вейвлет-преобразования.", ВУС, 1999.
2. A. Said and W.A. Pearlman, "A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees", IEEE Trans. on Circuits and Systems for Video Technology, vol. 6., pp. 243--250, June 1996.

REALIZATION OF ALGORITHM SPIHT ON SIGNAL PROCESSOR

Chernikov A., Plakhov A., Tchobanu M.

MOSCOW POWER ENGINEERING INSTITUTE (TECHNICAL UNIVERSITY)

The realization on signal processors TMS320C6x of algorithm of images compression named SPIHT is considered in the paper. The given algorithm belongs to a class of coders « with losses » and works with wavelet-

decomposition of a signal. SPIHT shows good characteristics of quality on the big degrees of compression, having thus simple structure. The stream generated at coding has a line of positive properties.

The methods of coding speed-up, both algorithmic, and program are considered in the paper; it is developed the hardware focused version of the coder intended for use in systems on the basis of signal processors.

1. Wavelet decomposition.

Convolution with separable filters occurs consistently in the lines, then on columns. In the first case the requests of coefficients from memory occur consistently. It allows to use opportunities a cache-memory, and also to organize the buffer in internal memory of the processor, that considerably accelerates calculation of convolution. For a case of columns the requests of coefficients from memory occur not consistently. In work it is offered to use DMA controller for the organization of transposing of group of columns of the initial image from external memory to the internal line buffer. Operation of transposing allows to pass from convolution of a column to convolution of a line, that essentially increases speed.

2. SPIHT algorithm.

During the analysis of LIS elements, the coder should look through all tree finding the significant coefficients. This process, repeatedly repeating, occupies a significant part of computing resources. For acceleration of coding process the new structure was put into the algorithm - the matrix of significance (MS), that has allowed to reduce the operating time of the coder considerably. The simple algorithm of formation MS allowing for one pass on the wavelet-decomposition coefficients to collect in the matrix of significance the information about all trees is developed.

Algorithm SPIHT uses list structures in its work. At realization as the application the list organization is convenient for the programmer and has no hardware restrictions. For realization on signal processors it is necessary to exclude dynamic work with memory. Hence, the use of lists is not desirable. For this reason for hardware realization was developed non-list SPIHT. The offered algorithm builds hierarchical trees, not using list structures. The processing is constructed in such a manner that all necessary points (significant points, points of the entrance in a tree) are analyzed once, as well as in list algorithm that's why the transition to the non-list algorithm passes without speed loss.

Refusal of list structures allows to develop simple high-speed hardware realization of the coder for its application in images processing systems.

The literature

1. V.I.Vorobiov and V.G.Gribunin. " The theory and the practice of wavelet-transform. ", VUS, 1999.
2. A. Said and W.A. Pearlman, " A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees ", IEEE Trans. on Circuits and Systems for Video Technology, vol. 6., pp. 243 - 250, June 1996.

