

УДК 681.32

СЕМЕРЕНКО В. П.

ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА В ЗАДАЧАХ СЖАТИЯ РАСТРОВЫХ ИЗОБРАЖЕНИЙ

*Винницкий национальный технический университет
Хмельницкое шоссе, 95, Винница, 21021, Украина
Тел.: +38(0432)439002, vpsmerenko@mail.ru*

Анотація. Запропоновано метод ущільнення без втрат бінарних та напівтонових зображень з використанням булевої алгебри кубічних функцій. Початкове зображення розбивається на окремі блоки фіксованого розміру, які інтерпретуються як карти Карно, для мінімізації яких розроблено новий алгоритм поліноміальної складності. Перевагами запропонованого методу є можливість паралельної реалізації на мікро- та макрорівні, стійкість до помилок обчислень, і виконання різноманітних перетворень над зображенням.

Ключові слова: ущільнення зображень, ущільнення без втрат, мінімізація булевих функцій, карти Карно, паралельна обробка, кубічні покриття.

Abstract. A novel approach for the lossless compression of binary and gray level images using the Boolean algebra of cubic functions is presented. An initial image is divided into the separate blocks of constant size, which are interpreted as Karnaugh map, for minimization of which the new algorithm of polynomial complexity is suggested. The advantages of this method of data compression are the possibility of parallel realization at the micro- and macro levels, immunity against computational errors and performance of the different transformations with the images.

Keywords: compression of the images, lossless compression, minimization of Boolean function, Karnaugh map, parallel processing, cubic covering.

Аннотация. Предложен метод сжатия без потерь бинарных и полутоновых изображений с использованием булевой алгебры кубических функций. Исходное изображение разбивается на отдельные блоки фиксированного размера, которые интерпретируются как карты Карно, для минимизации которых разработан новый алгоритм полиномиальной сложности. Достоинствами предложенного метода сжатия являются возможность параллельной реализации на микро- и макроуровне, устойчивость к ошибкам вычислений, и выполнение различных преобразований над изображением.

Ключевые слова: сжатие изображений, сжатие без потерь, минимизация булевых функций, карты Карно, параллельная обработка, кубические покрытия.

ВВЕДЕНИЕ

Одним из важнейших направлений в сфере информационных технологий является переход к электронному документообороту. В системах передачи данных, в геоинформационных системах, в банковском деле, в медицине документы либо сразу получают в электронном виде, либо преобразуются в него путем сканирования бумажных материалов. В большинстве случаев электронные документы представляются в растровом формате и занимают огромные массивы памяти. Особую актуальность имеет сжатие без потерь и сжатие именно изображений, поскольку они требуют для хранения гораздо большего объема памяти, чем текст [1].

Для изображений можно использовать универсальные методы сжатия данных без потерь. В основе многих известных методов лежит простая идея: если для часто используемых элементов использовать короткие коды, а для редко используемых элементов – длинные коды, тогда для хранения исходных данных в закодированном виде потребуется меньший объем памяти. На этом принципе построены статистические методы (алгоритм Хаффмена, Q-кодирование), словарные методы (алгоритм Лемпела-Зива и его различные модификации), групповое кодирование (RLE) [2].

Однако, эти методы ориентированы на обработку именно текстовых данных и не учитывают главную особенность изображений – наличие связности не только в строках, но и в столбцах, т.е. изображения обладают избыточностью в двух измерениях. Именно для учета особенностей этой

структуры данных и были разработаны специальные методы сжатия изображений.

Здесь в первую очередь необходимо упомянуть международные стандарты кодирования ССІТТ (ныне ІТУ-Т) V.42bis, разработанные для передачи черно-белых изображений по телеграфным каналам и сетям передачи данных [2,3]. Эти стандарты сжатия являются неадаптивными – для кодирования всех изображений, независимо от их содержания, в них применяются одни и те же схемы и алгоритмы. Новейший метод сжатия изображений – метод JBIG [4] – использует адаптивный арифметический кодировщик, что позволяет в несколько раз лучше сжимать изображения.

Степень сжатия данных – очень важный, но не единственный критерий, по которому необходимо оценивать каждый конкретный метод. К числу важнейших критериев качества и эффективности методов сжатия данных следует отнести время сжатия и восстановления, сложность программной и аппаратной реализации, возможности распараллеливания вычислений, устойчивости к ошибкам вычислений.

С учетом комплексного подхода к критериям качества сжатия изображений перспективными являются методы сжатия без потерь с использованием минимизации булевых функций [5-7]. Каждый пиксел изображения рассматривается как одно из выходных значений булевой функции. Далее производится минимизация этой функции известными методами. Полученная минимальная дизъюнктивная нормальная форма (ДНФ) функции после кодирования и представляет собой сжатое представление исходного блока данных.

ПОСТАНОВКА ЗАДАЧИ

Все преимущества сжатия изображений на основе минимизации булевых функций непосредственно зависят от эффективности реализации выбранного метода минимизации. Основная проблема, возникающая в задаче минимизации, состоит в большом объеме вычислений на различных этапах преобразований, в результате чего эта задача принадлежит к числу NP -сложных задач [8]. Поскольку при гипотезе $P \neq NP$ точное решение задачи минимизации получить эффективно невозможно, это не означает запрета возможности аппроксимации задачи, т.е. нахождения приближенного решения, которого, возможно, будет достаточно во многих практических случаях. Поэтому актуальной является разработка полиномиальных приближенных алгоритмов минимизации булевых функций.

Существенно повысить качество сжатия данных можно также при переходе к параллельной обработке, что позволит получить высокопроизводительную программно-аппаратную реализацию разработанных алгоритмов.

АНАЛИЗ СЛОЖНОСТИ МИНИМИЗАЦИИ БУЛЕВЫХ ФУНКЦИЙ

Как правило, минимизация булевой функции состоит из двух этапов [9]. На первом этапе минимизации СДНФ преобразуется в сокращенную ДНФ (СокДНФ) путем нахождения всех минтермов максимальных рангов (простых импликант), а на втором этапе – в тупиковую ДНФ (ТДНФ). В общем случае, для заданной функции F имеется некоторое множество ТДНФ, среди которых находится минимальная ДНФ (МДНФ).

Первый этап минимизации достаточно формализован и имеет полиномиальную сложность, т.е. принадлежит к классу задач P .

Получение множества ТДНФ на втором этапе минимизации обычно выполняется переборным способом с помощью таблицы покрытий. Выбор МДНФ из множества ТДНФ также требует полного перебора. По оценке Ю. Журавлева [9], для больших n , число ТДНФ превышает $5^{2^{n-4}}$. Именно второй этап минимизации и превращает всю задачу минимизации в NP -сложную задачу.

В последние годы были предложены новые подходы к минимизации булевых функций, в частности, эвристические методы [10], двоичные диаграммы решений [11]. Однако большинство новых методов также берут за основу известную стратегию минимизации: вначале генерация всех простых импликант, а затем выбор из них минимального количества для МДНФ.

Основная причина, по которой известные методы минимизации булевых функций остаются в классе NP -сложных задач, состоит в наличии длинной цепочки преобразований:

$$\text{СДНФ} \rightarrow \text{СокДНФ} \rightarrow \text{ТДНФ} \rightarrow \text{МДНФ}.$$

Парадокс состоит в том, что полученная на первом этапе СокДНФ во многих случаях является более сложной, чем исходная СДНФ. Для “почти всех” логических функций асимптотическая сложность СДНФ составляет $n2^{n-1}$, тогда как число минтермов в СокДНФ превышает $n^{(1-\varepsilon)\log_2 \log_2 n} 2^n$. Таким

образом, вначале сложность минимизируемой функции значительно возрастает, и лишь затем начинается поиск минимального решения переборным способом.

Осознание причин экспоненциального роста сложности указанных преобразований привел к появлению новой философии в разработке алгоритмов минимизации, которая не требует нахождения всех простых импликант минимизируемой функции [12].

Близким к этой философии является и предлагаемый подход к минимизации, который заключается в итеративном уменьшении сложности минимизируемой функции по схеме СДНФ → ”почти” МДНФ. Другими словами, для практических потребностей можно ограничиться нахождением близкой по сложности к МДНФ, при условии существенного сокращения времени преобразования и экономии памяти.

ВЫБОР МЕТОДА БУЛЕВОЙ МИНИМИЗАЦИИ

Рассмотрим полутоновое изображение, в котором каждый пиксел представлен с помощью k бит. Тогда изображение может иметь 2^k градаций яркости и его можно разбить на k битовых плоскостей, в которых каждый пиксел имеет только два значения: нулевое и единичное. Одну битовую плоскость можно представить как бинарное изображение, в котором каждый пиксел представляется одним битом. Предлагаемый метод сжатия рассматривается далее для бинарных изображений, но он также пригоден и для всех тех изображений, которые можно свести к совокупности битовых плоскостей (рис.1).

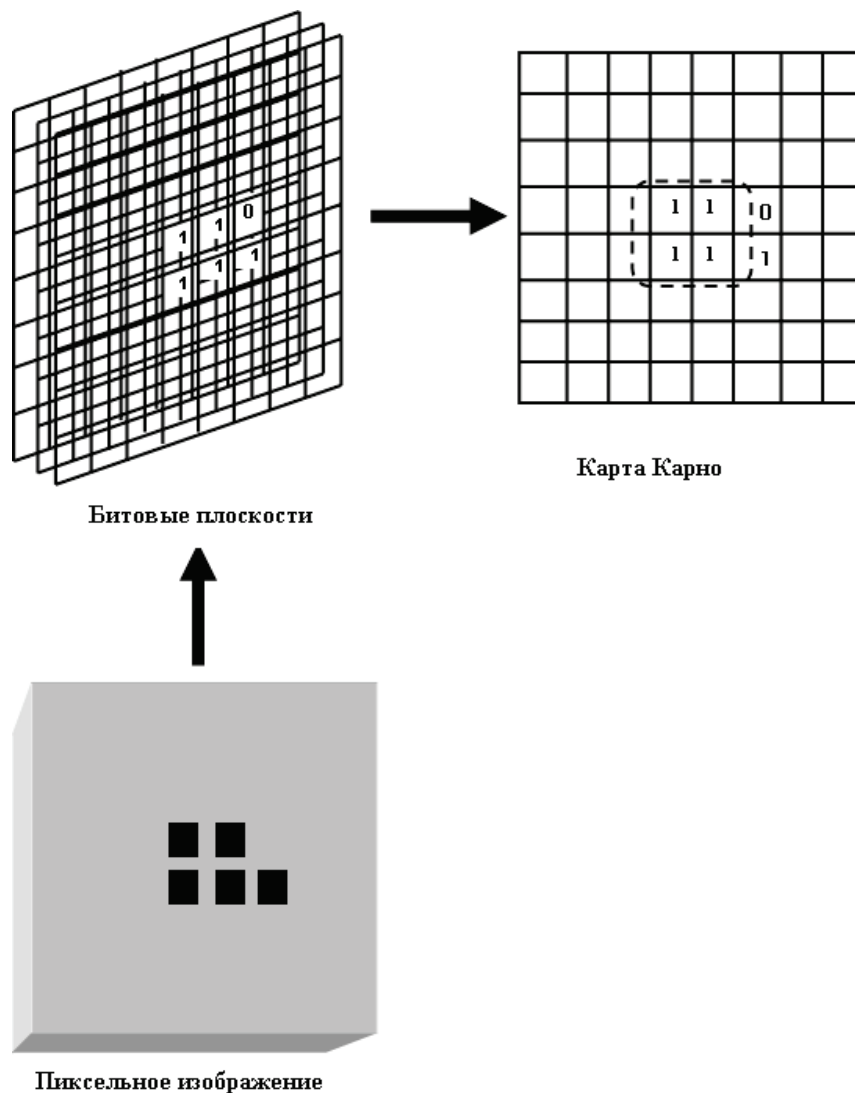


Рис. 1. Сжатие изображений на основе карт Карно

Теоретически все пиксели бинарного изображения (битовой плоскости) можно представить как

минтермы одной булевой функции. Однако минимизация такой функции займет много времени и для ускорения вычислений трудно применить параллельную обработку. Поэтому все изображение (битовая плоскость) подразделяется на квадратные блоки одного типа размером 4×4 , 8×8 или 16×16 пикселей.

Критерием выбора размера блока является степень сжатия, о чем далее будет сказано.

Сразу отметим, что минимизируются лишь блоки с пикселями разных цветов (смешанные), а для одноцветных блоков отмечается лишь их цвет.

Среди известных методов минимизации булевых функций лучше выбрать тот, который имеет удобное графическое представление каждого этапа преобразований. С этих позиций карты Карно не имеют конкурентов [9,13]. Выбранный нами квадратный блок изображения с черными и белыми пикселями уже фактически есть карта Карно с единицами и нулями (см. рис.1). Традиционно карты Карно применяются только для ручной минимизации, поэтому сложилось ошибочное мнение о непригодности этого метода для минимизации функций с большим числом переменных. Однако на самом деле не существует никаких ограничений на размер этих карт, поскольку для нумерации ячеек используется код Грея, который, как и двоичный код, легко может быть построен для любого целого числа. Достоинством карт Карно является также и то, что можно выполнять минимизацию как по единичным значениям функции, так и по нулевым значениям, следовательно, можно сделать выбор между ними в зависимости от соотношения числа единичных и нулевых значений.

Поскольку после минимизации данные содержат символы в троичном алфавите $\{0,1,x\}$ ($x \in \{0,1\}$), поэтому для обратного перехода в двоичный формат необходим этап кодирования. Например, можно выбрать один бит для символа x и по два бита для символов 0 и 1:

$$x \rightarrow 0, \quad 0 \rightarrow 10, \quad 1 \rightarrow 11.$$

Таким образом, для бинарных изображений можно выделить такие этапы сжатия информации:

1. Разбиение исходного изображения на блоки фиксированного размера, например, 16×16 пикселей.
2. Представление каждого блока картой Карно и нумерация каждого пикселя блока в коде Грея.
3. Для смешанных блоков выполнение минимизации соответствующей булевой функции либо по единичным, либо по нулевым значениям функции.
4. Кодирование и представление блоков в формате сжатых данных.

БУЛЕВЫ АЛГЕБРЫ ДЛЯ ЗАДАЧ СЖАТИЯ ИЗОБРАЖЕНИЙ

Теоретическим базисом для методов сжатия изображений с применением булевой минимизации являются булевы алгебры. В настоящее время в компьютерной схемотехнике используется булева алгебра логических функций с двузначными элементами, для которых определены операции конъюнкции, дизъюнкции и отрицания.

Основные определения для этой алгебры даны в [9]. Под логической функцией от m переменных будем понимать функцию $F\{x_1, x_2, \dots, x_m\}$ с областью логических значений $\{0,1\}$. Дизъюнктивной нормальной формой (ДНФ) называется дизъюнкция минтермов

$$h_1 \vee h_2 \vee \dots \vee h_n,$$

на которых булева функция F принимает значение логической 1. Минтерм содержит произведение переменных или их отрицаний, их количество определяет ранг минтерма. Количество минтермов определяет длину ДНФ, а общее количество рангов ее минтермов – сложность $L(F)$ ее ДНФ. Та ДНФ, которая состоит только из минтермов нулевого ранга, называется совершенной (СДНФ).

Для программно-аппаратной реализации задач сжатия изображений на базе современной микросхемотехники введем булеву алгебру кубических функций с тремя элементами

$$\{0, 1, x\}, \text{ где } x \in \{0, 1\},$$

которые могут образовывать p -кубы [14]. В общем случае p -куб представляет собой m -мерный набор элементов e_1, \dots, e_m , где $e_i \in \{0,1,x\}$, $i = 1 \div m$, а независимая координата x может принимать значения 0 или 1. Будем полагать, что число p независимых координат куба d_i определяет его размерность p_i ($p_i = 0 \div m$). 0 -куб не содержит независимых координат x и имеет нулевую размерность.

Кубическая функция $\Psi = \Psi(e_1, \dots, e_m)$ дает однозначное отображение множества 0 -кубов в множество $\{0,1\}$ и использует три основные операции (пересечения, объединения и дополнения кубов), а

также ряд дополнительных операций (вычитания и $*$ -произведения кубов). Определения этих операций даны в [13].

ОПРЕДЕЛЕНИЕ 1. Непустой результат операции $*$ -произведения

$$d_k = d_i * d_j \quad (1)$$

называется существенным, если размерность куба d_k будет превышать размерности обеих кубов d_i и d_j , или хотя бы одного из них.

Непустой результат операции $*$ -произведения соответствует в булевой алгебре логических функций операции склеивания минтермов Ab_i и $\overline{Ab_i}$:

$$Ab_i \vee \overline{Ab_i} = A \quad (2)$$

Различают единичный кубический комплекс $K^{(1)}$ и нулевой кубический комплекс $K^{(0)}$ – множества 0 -кубов, на которых функция Ψ принимает значение соответственно 1 и 0. Исходная карта Карно по сути есть объединение комплекса $K^{(1)}$ и комплекса $K^{(0)}$.

Каждый из кубических комплексов однозначно определяет кубическую функцию Ψ , поэтому в дальнейшем может использоваться только один из них, а именно, тот комплекс, который имеет меньшее число кубов. В ходе минимизации 0 -кубы объединяются в 1-кубы, 2-кубы и т.д.

В результате общее количество кубов уменьшается и кубические комплексы превращаются в кубические покрытия. Минимальное единичное D -покрытие (минимальное нулевое R -покрытие) кубической функции Ψ содержит минимальное количество кубов m_D (m_R) максимальной размерности, которые в совокупности содержат все кубы комплекса $K^{(1)}$ ($K^{(0)}$). D -покрытие и R -покрытие взаимно эквивалентны, поэтому вопросы минимизации в дальнейшем рассматривается применительно к D -покрытию, но полученные результаты применимы и к R -покрытию.

Если провести аналогию с булевой алгеброй логических функций, тогда минимальному D -покрытию кубической функции Ψ соответствует минимальная ДНФ логической функции F , один 0 -куб комплекса $K^{(1)}$ соответствует одному минтерму СДНФ, а s -куб d_j ($j = 1 \div m_D$) минимального D -покрытия соответствует импликанте ранга p минимальной ДНФ функции F . Например, D -покрытие

$$D = \begin{vmatrix} 0 & 1 & 0 & x \\ 1 & x & 0 & 1 \\ x & 1 & 1 & x \end{vmatrix}$$

соответствует следующей ДНФ:

$$f(a,b,c,d) = \overline{a} \& b \& \overline{c} \vee a \& \overline{c} \& d \vee b \& c.$$

МАШИНООРИЕНТИРОВАННЫЙ МЕТОД МИНИМИЗАЦИИ ПО КАРТАМ КАРНО

Под задачей минимизации логической функции F будем понимать задачу нахождения ее ДНФ минимальной сложности по критерию количества v ее минтермов и суммы рангов отдельных минтермов:

$$L_{\min}(F) = v + \sum_i^v p_i, \quad v = 1 \div 2^n. \quad (3)$$

Для оценки сложности кубических покрытий будем использовать величину цены покрытия [13]. Для D -покрытия, состоящего из v кубов, цена C_D будет равна

$$C_D = v + \sum_i^v (m - s_i), \quad v = 1 \div 2^n. \quad (4)$$

Формула (4) эквивалентна критерию минимальности (3) в алгебре логических функций.

Пусть имеется кубы d_i и d_j D -покрытия, имеющие размерности соответственно s_i и s_j , и в результате операции (1) получен непустой куб d_k , имеющий размерность s_k .

ОПРЕДЕЛЕНИЕ 2. Непустой куб d_k , сформированный в результате операции (1), поглощает (покрывает) куб d_i или куб d_j , если его размерность s_k превышает размерности s_i или s_j

соответственно куба d_i или куба d_j .

Согласно приведенным определениям, в результате операции (1) возможны следующие результаты.

1. Результирующий куб d_k поглощает оба исходных куба d_i и d_j : $s_k > s_i$, $s_k > s_j$. Результат операции (1) является существенным.

2. Результирующий куб d_k поглощает только один исходный куб d_i : $s_i > s_k$, $s_j \leq s_k$. Результат операции (1) является существенным.

3. Результирующий куб d_k не поглощает ни одного из исходных кубов d_i и d_j : $s_k \leq s_i$, $s_k \leq s_j$. Результат операции (1) не является существенным.

Предлагаемый алгоритм минимизации состоит из совокупности итераций, в ходе которых из $D^{(z)}$ -покрытия цены $C_D^{(z)}$, полученного на z -й итерации, формируется $D^{(z+1)}$ -покрытие цены $C_D^{(z+1)}$ на $(z+1)$ -й итерации: $C_D^{(z+1)} \leq C_D^{(z)}$. Формирование нового покрытия $D^{(z+1)}$ происходит в результате операции (1) над кубами d_i и d_j $D^{(z)}$ -покрытия на основе следующих правил ($d_i \in D^{(z)}$, $d_j \in D^{(z)}$, $d_k \in D^{(z+1)}$).

Правило добавления. В $D^{(z+1)}$ -покрытие включается новый куб d_k , полученный в результате операции (1) над кубами d_i и d_j , если куб d_k является существенным.

Правило удаления. Из покрытия $D^{(z)}$ удаляется тот куб d_i , который поглощается новым кубом d_k , полученного в результате операции (1) над кубами d_i и d_j .

Можно заметить, что существенный результат операции (1) эквивалентен операции склеивания (2).

Рассмотрим алгоритм минимизации для одного блока изображения в алгебре кубических функций на основе указанных правил.

АЛГОРИТМ

Исходные данные: Задан блок изображения размером $(n \times n)$ пикселей в виде ячеек карты Карно в алфавите $\{0,1\}$.

Результат: Блок сжатого изображения в виде минимального кубического покрытия в алфавите $\{0,1,x\}$.

1. Сформировать в коде Грея адреса ячеек карты Карно размера $(\log_2 n \times \log_2 n)$, соответствующей заданному блоку изображения.

2. Разбить совокупность ячеек карты Карно на две группы:

а) группу с ячейками, которые соответствуют единичным пикселям изображения, т.е. 0-кубам кубического комплекса $K^{(1)}$;

б) группу с ячейками, которые соответствуют нулевым пикселям изображения, т.е. 0-кубам кубического комплекса $K^{(0)}$.

3. Выбрать меньшую по размеру группу ячеек, пусть это будет $K^{(1)}$, и из соответствующих 0-кубов, сформировать текущее кубическое покрытие D_z . Задать номер итерации по строкам $z=1$.

4. Разбить покрытие D_z на t пар покрытий $D_z^{(i)}$ и $D_z^{(i+1)}$, содержащих кубы из t соседних пар по $2 \times w$ строк в паре ($i = 1, 2, \dots, w$, $w = 2^{z-1}$).

5. Выполнить операции $*$ -произведения между кубами покрытий $D_z^{(i)}$ и $D_z^{(i+1)}$ и сформировать покрытие $D_{z,row}^{(i,i+1)}$ согласно Правил добавления и удаления:

$$D_{z,row}^{(i,i+1)} = D_z^{(i)} * D_z^{(i+1)}, \quad (5)$$

а затем объединить полученные покрытия для всех t соседних пар соседних строк:

$$D_{z,row} = \bigcup_t D_{z,row}^{(i,i+1)}.$$

6. Если $t > 1$, то увеличить номер итерации ($z = z + 1$), и вернуться к п.4.

7. Сформировать минимальное покрытие по строкам $D_{\min, row}$:

$$D_{\min, row} = \bigcup_z D_{z, row}.$$

8. Задать номер итерации по столбцам $z = 1$.

9. Разбить покрытие D_z на t пар покрытий $D_z^{(i)}$ и $D_z^{(i+1)}$, содержащих кубы из t соседних пар по $2 \times w$ столбцов в паре ($i = 1, 2, \dots, w, w = 2^{z-1}$).

10. Выполнить операции $*$ -произведения между кубами покрытий $D_z^{(i)}$ и $D_z^{(i+1)}$ и сформировать покрытие $D_{z, col}^{(i, i+1)}$ согласно Правил добавления и удаления:

$$D_{z, col}^{(i, i+1)} = D_z^{(i)} * D_z^{(i+1)},$$

11. Объединить полученные покрытия для всех t соседних пар соседних столбцов:

$$D_{z, col} = \bigcup_t D_{z, col}^{(i, i+1)}.$$

12. Если $t > 1$, то увеличить номер итерации ($z = z + 1$), и вернуться к п.4.

13. Сформировать минимальное покрытие по столбцам $D_{\min, col}$, которое является искомым покрытием D_{\min} :

$$D_{\min, col} = \bigcup_z D_{z, col}, \quad D_{\min} = D_{\min, col}.$$

14. Конец.

Примечание. Если в п.3 будет выбран комплекс $K^{(0)}$, тогда п. 4-п.13 выполняются для нулевых кубических покрытий.

ВОСПРОИЗВЕДЕНИЕ СЖАТОЙ ИНФОРМАЦИИ

Задачей восстановления сжатой информации является получение значения каждого пикселя исходного изображения. В терминах алгебры кубических функций эта задача эквивалентна задаче получения комплекса $K^{(1)}$ по известному D_{\min} -покрытию или получения комплекса $K^{(0)}$ по известному R_{\min} -покрытию. Для перехода от D_{\min} -покрытию к комплексу $K^{(1)}$ можно каждый куб d_i размерности s_i этого покрытия ($d_i \in D_{\min}, i = 1 \div n_{\min}$) заменить 2^{p_i} 0-кубами комплекса $K^{(1)}$. Такая операция сопровождается получением большого количества одинаковых 0-кубов от различных кубов D_{\min} -покрытия, что требует соответствующей процедуры удаления лишних 0-кубов.

Гораздо эффективнее выполнить восстановление сжатой информации без преобразования кубов D_{\min} -покрытия. С этой целью понадобится два m -разрядных счетчика, которые будут вести счет в коде Грея от 0 до 2^m . Первый счетчик (счетчик строк) будет последовательно выдавать адреса строк карты Карно, а второй счетчик (счетчик столбцов) – адреса столбцов этой же карты Карно. Двоичный $2m$ -разрядный вектор v_l , полученный конкатенацией m -разрядных значений обоих счетчиков, представляет собой 0-куб комплекса $K^{(1)}$, следовательно, за тактов 2^m работы этих счетчиков будут сформированы все 0-кубы, т.е. адреса всех ячеек карты Карно. На каждом такте работы в процессе восстановления сжатой информации выполняется операция пересечения каждого куба d_i D_{\min} -покрытия с вектором v_l :

$$d_i \cap v_l, \quad l = 0 \div 2^n. \quad (6)$$

Непустой результат операции (6) будет свидетельствовать, что соответствующий вектору v_l 0-куб комплекса $K^{(1)}$ содержится в D_{\min} -покрытии, значит, соответствующий этому 0-кубу пиксель имеет единичное значение. Таким образом, за 2^n тактов работы будет восстановлен один блок исходного изображения. Отметим, что вывод изображения на экран дисплея может происходить одновременно с его восстановлением.

Рассмотренный способ восстановления изображения, кроме высокого быстродействия, имеет еще несколько очень важных преимуществ.

При восстановлении можно получить не только исходное, но и так называемое огрубленное изображение. Если упомянутые счетчики будут формировать только каждый k -й адрес 0-куба ($k = 2, 4, 8, \dots$), тогда можно получить только часть пикселей, т.е. изображение худшего качества, но в k раз быстрее. При этом граничные размеры огрубленного блока изображения могут быть такими же, как и в исходном блоке изображения, или в k^2 раз меньшими. В последнем случае произойдет уменьшение изображения в k^2 раз.

Аналогичные действия можно выполнить для всех блоков изображения, а затем, после несложных преобразований, получить уменьшенную копию всего изображения, например, в левом верхнем углу экрана дисплея.

ЭКСПЕРИМЕНТАЛЬНЫЕ РЕЗУЛЬТАТЫ

Предложенный метод минимизации булевых функций реализован в виде программы KARNO-MIN на языке C++ на персональном компьютере с тактовой частотой 1,40 ГГц и объемом оперативной памяти 1,25 Гб.

В качестве прототипа для сравнения с известными средствами минимизации была выбрана программа ESPRESSO [15], которая уже длительное время является мировым промышленным стандартом в этой сфере. Для сравнения использовались тестовые примеры для одновыходных булевых функций, заданных в виде двоичных массивов СДНФ с различным числом входов (табл. 1). Для учета временных затрат использовалась библиотека параллельного программирования OpenMP [16], в которой имеются специальные средства для измерения астрономического времени выполнения программного кода.

Как видно из таблицы 1, по степени минимизации булевых функций возможности известной и разработанных программ почти одинаковы, зато программа KARNO-MIN работает в несколько раз быстрее.

Таблица 1.

Сравнение результатов минимизации булевых функций

№	Исходные данные			Результаты минимизации			
	Число входов	Число выходов	Число минтермов СДНФ	ESPRESSO		KARNO-MIN	
				Число минтермов МДНФ	Среднее время выполнения в сек.	Число минтермов МДНФ	Среднее время выполнения в сек.
1	6	1	64	1	0,303	1	0,039
2	8	1	256	1	0,308	1	0,048
3	10	1	1024	1	0,331	1	0,080
4	6	1	20	3	0,317	4	0,023
5	0	1	30	14	0,315	13	0,025

Таблица 2.

Степени сжатия тестовых бинарных изображений

Тестовое изображение	Размер блока 16×16 пикселей			Размер блока 8×8 пикселей			Размер блока 4×4 пикселей		
	η	N	N_{not}	η	N	N_{not}	η	N	N_{not}
lena	1,727	1024	71	2,156	4096	371	1,908	16384	2427
tulips	1,452	1333	251	1,956	5334	539	1,837	21338	3916
baboon	0,560	156	144	0,801	625	404	1,021	2500	1185

Проведем теперь анализ степени сжатия для некоторых общеизвестных тестовых изображений (lena, baboon, tulips), представленных в виде черно-белых изображений формата bmp (табл. 2).

Как видно из табл.2, для изображений с большими компактными областями одного цвета

наибольшая степень сжатия получается с блоками размеров 8×8 и 16×16 пикселей. Для “мелкозернистых” изображений необходимо использовать блоки минимального размера 4×4 , поскольку для больших блоков эффект сжатия отсутствует.

Для более точного анализа можно подсчитать соотношение количества N_{not} блоков, которые не сжимаются, к общему количеству N блоков. Для изображений, которые могут быть сжаты в два и более раз, доля несжимаемых блоков, как правило, не превышает 10%. Эффект сжатия начинается, если доля несжимаемых блоков менее 45%.

ПРИМЕНЕНИЕ ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ ПРИ СЖАТИИ И ВОССТАНОВЛЕНИИ ИЗОБРАЖЕНИЙ

Методы обработки логических (бинарных) данных отвечают требованиям современной интегральной технологии и, поэтому, позволяют применить параллельную обработку данных на различных этапах преобразований изображений [17].

Будем различать параллелизм на макро- и микроуровне [18].

Параллелизм на микроуровне основан на математическом аппарате булевой алгебры кубических функций. Отличительной особенностью выполнения операций над кубами является возможность разбить весь процесс выполнения операций с кубами покрытий кубической функции на элементарные независимые друг от друга фрагменты вычислений и организовать параллельную обработку данных.

В процессе минимизации кубических функций выполняется операция $*$ -произведения (5) между кубами покрытий $D_z^{(i)}$ и $D_z^{(i+1)}$. Все эти операции можно выполнить одновременно, т.е. параллельно. В процессе восстановления сжатой информации необходимо выполнить операции пересечения (6) каждого куба d_i D_{\min} -покрытия с вектором v_l . Эти операции также можно выполнить одновременно, т.е. параллельно.

Указанные операции с кубами очень удобно реализовать аппаратно с помощью программированной систолической структуры (ПСС), которая использует сразу два принципа параллелизма: конвейерную и матричную обработку. Рассмотрим подробнее реализацию параллелизма на этапе восстановления сжатой информации.

Для восстановления блока бинарного изображения размером $n \times n$ пикселей понадобится ПСС, состоящая из матрицы $n \times n$ вычислительных ячеек 1, n -разрядного регистра результата 2 и n элементов ИЛИ 3 (Рис. 2).

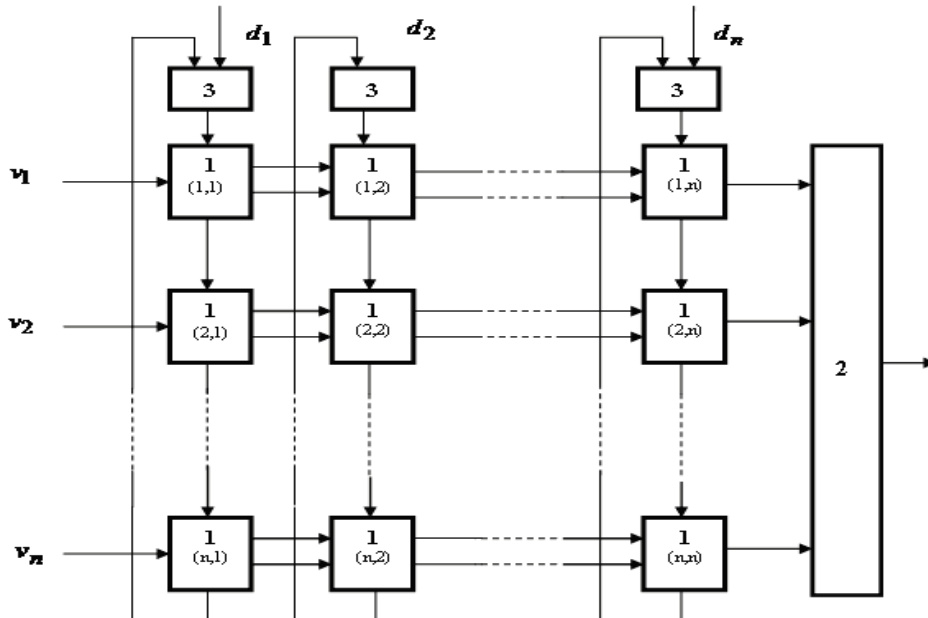


Рис. 2. ПСС для восстановления блока бинарного изображения

Перед началом работы ПСС программируется, т.е. через ее вертикальные входы записываются n кубов d_1, \dots, d_n D_{\min} -покрытия, причем со сдвигом на один разряд относительно друг друга. В рабочем режиме на горизонтальные входы ПСС поступают векторы v_1, \dots, v_n от счетчиков строк и

столбцов, также со сдвигом на один разряд относительно друг друга. На каждом микротакте работы компоненты кубов D_{\min} -покрытия движутся циклически по вертикальным шинам ПСС, компоненты векторов v_l – по горизонтальным шинам ПСС, а в каждой вычислительной ячейке ПСС выполняется элементарная операция (6). Начиная с $(2n-1)$ -го микротакта работы на выходных горизонтальных шинах справа появляются итоговые результаты операций пересечения каждого куба d_i D_{\min} -покрытия с соответствующими векторами v_l . Коэффициент ускорения работы η будет равен отношению числа микротактов при последовательной и параллельной работе:

$$\eta = \frac{n^2}{3(n-1)+1}.$$

Параллелизм на макроуровне означает возможность одновременной обработки более крупных фрагментов данных: с блоками изображений и битовыми плоскостями. Параллельную обработку на макроуровне можно организовать программно с помощью многопоточкового программирования или с помощью специальных библиотек (OpenMP, MPI). Такая программная реализация основана на аппаратной поддержке в виде многоядерных процессоров или мощных вычислительных кластеров [16].

ВЫВОДЫ

Представлен новый метод сжатия данных без потерь, использующий метод минимизации булевых функций. Этот метод является универсальным, но наибольший эффект достигается при сжатии бинарных и полутоновых изображений.

Разработан метод минимизации булевых функций полиномиальной сложности на основе формализованной обработки карт Карно большой размерности, который может иметь и самостоятельное применение. Отметим два его важнейших достоинства. Во-первых, значительно сокращается объем промежуточных вычислений благодаря непосредственному переходу от СДНФ к “почти” минимальной ДНФ минуя этап получения СоКДНФ. Во-вторых, ориентация на параллельную обработку на аппаратном уровне с помощью систолических (волновых) структур благодаря использованию в качестве теоретического базиса алгебры кубических функций. Параллельная обработка может использоваться также на программном уровне для одновременного сжатия блоков изображений и битовых плоскостей.

Восстановление изображения осуществляется быстрее его сжатия. При этом также можно получить “огрубленное” изображение за еще более короткий интервал времени и осуществить его масштабирование.

Большие перспективы открываются при использовании предлагаемого сжатия данных в оптоэлектронных системах [19]. Благодаря мгновенной передаче изображений одним световым пучком обеспечивается возможность двухуровневого распараллеливания процесса сжатия полученных изображений: одновременной работы со всеми блоками изображения и распараллеливания алгоритма минимизации по отдельным итерациям внутри каждого блока.

При передаче сжатых изображений их восстановление в конечном пункте также возможно с применением параллелизма.

СПИСОК ЛИТЕРАТУРЫ

1. Ватолин Д. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео: / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин. – М.: ДИАЛОГ-МИФИ, 2002. – 384 с.
2. CCITT, Standardization of Group 3 Facsimile Apparatus for Document Transmission, ITU Recommendation T.4, 1980.
3. CCITT, Facsimile Coding Schemes and Coding Control Functions for Group 4 Facsimile Apparatus, ITU Recommendation T.6, 1984.
4. Сэломон Д. Сжатие данных, изображения и звука / Д. Сэломон. – М.: Техносфера, 2004. – 368 с.
5. Augustine J., Feng W., Makur A. and Jacob J., “Switching theoretic approach to image compression,” *Journal of Signal Processing (Elsevier Science)*, vol. 44, June 1995, pp. 243-246.
6. Kumar N. and Gupta S., “Use of Local Minimization for Lossless Gray Image Compression,” *International Journal of Computer Science and Information Technologies*, vol. 1(4), 2010, pp. 203-207.
7. Falkowski B.J., “Lossless binary image compression using logic function and spectra,” *Computers and Electrical Engineering*, vol. 30, Issue 1, January 2004, pp. 17-43.
8. Гэри М. Вычислительные машины и труднорешаемые задачи: / М. Гэри, Д. Джонсон. – М.: Мир, 1982. – 419 с.

9. Поспелов Д.А. Логические методы анализа и синтеза схем: / Д.А. Поспелов. Изд. 3-е. – М., Энергия, 1974. – 368 с.
10. Закревский А.Д. Минимизация булевых функций многих переменных в классе ДНФ – итеративный метод и программная реализация / А.Д. Закревский, Н.Р. Торопов // Теоретические основы прикладной дискретной математики. – 2009. – №1(3). – С.5-14.
11. Смагин А.А. Применение методов минимизации булевых функций для оптимизации цифровых устройств / А.А. Смагин, А.В. Шиготаров // Известия Самарского научного центра Российской академии наук. – 2009, т. 11. – №3(2). – С. 343-349.
12. Biswas N.N., Srikanth S. and Jacob J., “Cubical CAMP for Minimization of Boolean Functions,” 9th International Conference on VLSI Design – January 1996, pp. 264-269.
13. Миллер Р. Теория переключательных схем: Пер. с англ. / Р. Миллер. – М.: Наука, 1970, – Т.1. – 416 с.
14. Семеренко В.П. Систолическая реализация кубических функций / В.П. Семеренко // Электронное моделирование. – 1992. – №1. – С. 21-25.
15. Brayton R.K., McGeer P., Sanghavi J., Sangiovanni-Vincentelli A. “A New Exact Minimizer for Two-Level Logic Synthesis,” Logic Synthesis and Optimization, Kluwer Academic Publishers, 1993, p.2-31.
16. Эхтер Ш., Многоядерное программирование: / Ш. Эхтер, Дж. Робертс. – СПб.: Питер, 2010. – 316 с.
17. Кухарев Г.А. Техника параллельной обработки бинарных данных на СБИС: / Г.А.Кухарев, В.П.Шмерко, С.Н.Янушкевич. – Минск: Выш. шк.,1991.–226с.
18. Семеренко В.П. Систолическая реализация конечных автоматов / В.П. Семеренко // Оптико-електронні інформаційно-енергетичні технології. – 2007. – № 2. – С. 154-162.
19. Кожемяко В.П. Оптикоелектронні системи і пристрої. Навчальний посібник: / В.П. Кожемяко, В.С. Гаркушевський, Петрук В.Г. – Вінниця.: ВНТУ, 2005. – 100 с.

Надійшла до редакції 15.10.2011р.

СЕМЕРЕНКО ВАСИЛЬ ПЕТРОВИЧ – к.т.н., доцент, доцент кафедри обчислювальної техніки, Вінницький національний технічний університет, м.Вінниця, Україна.