

Технология иерархического кодирования видеоинформации

Васин Ю.Г., Жерздев С.В.

Аннотация

Предлагается информационная технология иерархического кодирования изображений различной природы с использованием локальных однородных хорошо приспособленных базисных функций (ЛОХПБФ). Рассматривается сжатие без потерь и с контролируемым по чебышевской метрике уровнем погрешности. Предложены способы для эффективной практической реализации предложенных алгоритмов и приведены сравнительные результаты сжатия для различных типов изображений.

1 Введение

В работе предлагаются информационные технологии адаптивного сжатия видеоинформации различной природы (полутонные и полноцветные изображения, изображения с индексацией цвета) на основе иерархических структур усеченных двоичных деревьев (УДД), построенных на базе локальных однородных хорошо приспособленных базисных функций (ЛОХПБФ) [1 – 6].

Рассматриваемая иерархическая структура имеет ряд преимуществ. Во-первых, она обеспечивает восстановление изображения с заданной степенью детализации (разрешением) по нескольким первым уровням, с последующим, в случае необходимости, уточнением. Это свойство полезно в задачах удаленного доступа, поскольку позволяет оперативно начать отображение изображения еще в процессе передачи, с постепенным увеличением уровня детализации по мере поступления данных. Кроме того, становится возможной эффективная реализация отображения изображения с различным коэффициентом масштабирования для, например, обеспечения навигации по большеформатному изображению.

Во-вторых, существуют различные эффективные алгоритмы обработки данных, представленных в иерархическом виде. К ним можно отнести как алгоритмы для решения топологических и геометрических задач, так и алгоритмы обработки изображения в целом [5, 6].

Адаптивное сжатие на основе ЛОХПБФ обеспечивается с любым наперед заданным уровнем допустимой поэлементной погрешности, т.е. погрешности в каждой точке. В отличие от интегральных критериев, это позволяет очень точно контролировать качество восстанавливаемого изображения. Необходимо отметить возможность задания различной допустимой погрешности для различных уровней дерева. Так, задавая для верхних уровней несколько меньшую погрешность, можно получить восстановленное изображение гораздо лучшего (по субъективной оценке) качества.

Целью работы является развитие информационной технологии иерархического сжатия видеоинформации на ба-

Работа выполнена при финансовой поддержке РФФИ, грант поддержки ведущих научных школ №00-15-96108 и ФУП “Интеграция” проект К-03392.

зе иерархической структуры УДД для повышения эффективности сжатия изображений.

2 Полутонные изображения

Будем считать, что полутонное изображение задано растром пикселей из H строк и W столбцов. Представим его матрицей O , в которой каждый отсчет (пиксель) изображения представлен числовым значением интенсивности $o_{i,j}$, $1 \leq i \leq H$, $1 \leq j \leq W$, квантованным на 2^b уровней.

Для контроля уровня потерь при сжатии изображения с потерями будем использовать метрику Чебышева. Будем говорить, что изображение сжато с ошибкой ε , если восстановленная версия изображения задана матрицей отсчетов \tilde{O} и

$$(1) \quad \varepsilon = \max_{1 \leq i \leq H, 1 \leq j \leq W} |o_{i,j} - \tilde{o}_{i,j}|$$

Алгоритм построения иерархической структуры предполагает, что $H = W = 2^L + 1$, поэтому далее будем считать, что производится упаковка фрагмента изображения, представленного матрицей отсчетов P , элементы которой $p_{i,j}$, $1 \leq i, j \leq 2^{ls} + 1$ являются целыми числами в диапазоне $[0, 2^b - 1]$.

Предлагаемая информационная технология состоит из нескольких последовательных операций.

1. Построение иерархической структуры отсчетов – двоичного дерева.
2. Интерполяция отсчетов по отсчетам предыдущих уровней иерархии и, возможно, переход к разностному дереву.
3. Преобразование структуры двоичного дерева отсчетов к усеченному двоичному дереву (УДД).
4. Кодирование, если необходимо, структуры УДД.
5. Кодирование значений отсчетов или погрешностей интерполяции.

2.1 Построение двоичного дерева отсчетов

Построим иерархическую последовательность изображений P^k , $0 \leq k \leq 2ls$. Положим $P^{2ls} \equiv P$, т.е.

$$(2) \quad p_{i,j}^{2ls} = p_{i,j}, \quad 1 \leq i, j \leq 2^{ls} + 1$$

Изображения вышележащих уровней иерархии определим следующим образом

$$(3) \quad p_{i,j}^{2k+1} = p_{2i-1,j}^{2k+2}, \quad 1 \leq i \leq 2^k + 1, \quad 1 \leq j \leq 2^{k+1} + 1, \quad 0 \leq k < ls$$

$$(4) \quad p_{i,j}^{2k} = p_{i,2j-1}^{2k+1}, \quad 1 \leq i \leq 2^k + 1, \quad 1 \leq j \leq 2^k + 1, \quad 0 \leq k < ls$$

Таким образом, изображение каждого уровня иерархии представляет собой прореженную вдвое по одной из осей копию изображения следующего уровня иерархии, а изображение нулевого (верхнего) уровня представлено четырьмя угловыми отсчетами исходного изображения.

Можно упорядочить отсчеты исходного изображения в иерархическую структуру, соответствующую двоичному дереву. При этом каждый уровень такого дерева будет содержать узлы, помеченные значением одного (в некоторых случаях двух) отсчета. Отсчеты, принадлежащие одному уровню двоичного дерева составляют множество отсчетов, которые необходимы для получения изображения соответствующего уровня иерархии из изображения предыдущего уровня.

Будем обозначать узел с номером i на уровне k двоичного дерева N_i^k . Тогда узел N_1^1 является корнем двоичного дерева, а сыновьями узла N_i^k являются узлы N_{2i-1}^{k+1} и N_{2i}^{k+1} . При этом существует следующее соответствие узлов двоичного дерева и отсчетов

$$(5) \quad N_{2^k(i-1)+j-1}^{2k} : P_{2i,j}^{2k}, \\ 1 \leq i \leq 2^{k-1}, 2 \leq j \leq 2^k, 1 \leq k \leq ls$$

$$(6) \quad N_{2^k(i-1)+1}^{2k} : P_{2i,1}^{2k}, \\ 1 \leq i \leq 2^{k-1}, 1 \leq k \leq ls$$

$$(7) \quad N_{2^k(j-1)+i-1}^{2k-1} : P_{i,2j}^{2k-1}, \\ 2 \leq i \leq 2^k, 1 \leq j \leq 2^{k-1}, 1 \leq k \leq ls$$

$$(8) \quad N_{2^k(j-1)+1}^{2k-1} : P_{1,2j}^{2k}, \\ 1 \leq j \leq 2^{k-1}, 1 \leq k \leq ls$$

Пример соответствия узлов двоичного дерева и отсчетов исходного изображения для случая $ls = 2$ (размеры изображения 5×5 отсчетов) приведен на Рис. 1.

Отсчеты, приписанные узлам на каждом уровне двоичного дерева являются отсчетами исходного изображения, а не результатом некоторой усредняющей функции. Благодаря этому упрощается контроль погрешности восстановления исходного изображения. Действительно, пусть отсчеты уровней $1, 2, \dots, 2ls$ двоичного дерева сжаты с ошибками $\hat{\varepsilon}_1, \hat{\varepsilon}_2, \dots, \hat{\varepsilon}_{2ls}$ соответственно. Тогда изображение P^{k*} может быть восстановлено с ошибкой

$$(9) \quad \varepsilon_{k*} = \max_{1 \leq k \leq k*} \hat{\varepsilon}_k$$

На практике выбирается

$$(10) \quad \hat{\varepsilon}_k = \varepsilon_{hi}, 1 \leq k \leq 2ls - 2$$

$$(11) \quad \hat{\varepsilon}_k = \varepsilon_{max}, 2ls - 1 \leq k \leq 2ls$$

причем $\varepsilon_{hi} \leq \varepsilon_{max}$. Очевидно, исходное изображение будет восстановлено с ошибкой ε_{max} . Выбор $\varepsilon_{hi} < \varepsilon_{max}$ позволяет получить больший коэффициент сжатия при несущественно худшем субъективно воспринимаемом качестве восстановленного изображения.

В другой интерпретации каждому узлу N_i^k дерева соответствует некоторая прямоугольная область исходного изображения D_i^k , причем дочерним узлам N_{2i-1}^{k+1} и N_{2i}^{k+1} соответствуют области D_{2i-1}^{k+1} и D_{2i}^{k+1} соответственно, полученные разбиением области D_i^k пополам. Т.е. двоичное

дерево задает рекурсивное разбиение исходного изображения

$$(12) \quad D_i^k = D_{2i-1}^{k+1} \cup D_{2i}^{k+1}$$

Как можно заметить, каждое поддерево содержит пространственно локализованные группы отсчетов (Рис. 2) и можно принять гипотезу об относительно сильной взаимной корреляции отсчетов одного поддерева.

Необходимо отметить, что общее количество отсчетов в двоичном дереве совпадает с числом отсчетов исходного изображения с точностью до четырех отсчетов, принадлежащих изображению верхнего уровня иерархии P^0 . Таким образом, преобразование изображения к иерархической структуре не увеличивает объема кодируемых данных.

2.2 Интерполяция отсчетов и построение разностного дерева

Известно, что для интерполяции широко используются методы разложения сигналов по тем или иным системам базисных функций (в частности, наиболее известно разложение Фурье). При этом в большинстве случаев выбор систем базисных функций производится полуэмпирически с использованием опыта и интуиции исследователя. Наряду с требованием высокого коэффициента сжатия, необходимо учитывать емкостные и временные ограничения (сколько требуется времени для вычисления коэффициентов разложения по базису, сколько требуется памяти для хранения базисных функций). Поэтому выбор базисных функций всегда был процессом творческим и искусство исследователя заключалось в эффективности этого выбора, как правило, из известного набора систем базисных функций (тригонометрический базис, функции Уолша, функции Адамара и т.д.). В этом плане очень важна разработка объективных методов построения систем базисных функций, которые опираются не на интуицию исследователя, а исходят из свойств совокупности исходных данных.

В решение перечисленных проблем было предложено [2] синтезировать различные хорошо приспособленные базисные функции (ХПБФ), удовлетворяющие заданным свойствам. С этой целью была введена простая геометрическая модель представления множества исходных сигналов в виде точек многомерного пространства. В качестве ХПБФ выбирались орты построенного подпространства меньшей размерности, которое покрывало исходное множество точек с определенной заранее точностью в заданной метрике [2, 7]. Были доказаны теоремы и разработаны алгоритмы, позволяющие осуществлять переход от ортонормированной системы ХПБФ к системе сглаживающих ХПБФ (ХПСБФ) и восстанавливающих ХПБФ (ХПВБФ). В этом случае, кодирование сводилось к выбору существенных отсчетов на исходном сигнале с использованием ХПСБФ и декодированию с использованием ХПВБФ [7, 8].

С целью дальнейшей оптимизации ХПБФ, достижения локальности в обработке, сокращения емкостных и временных ограничений было предложено [1 – 6] синтезировать однородные (совершенно одинаковые), но различные по времени локальные (короткие) сглаживающие

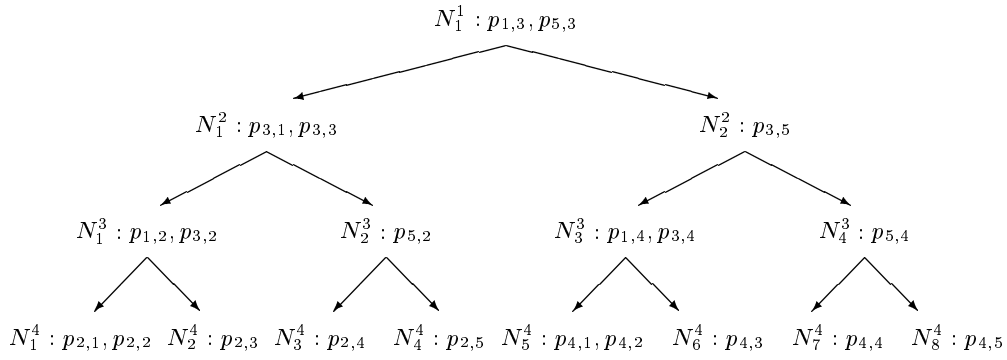


Рис. 1: Соответствие отсчетов узлам двоичного дерева ($ls = 2$)

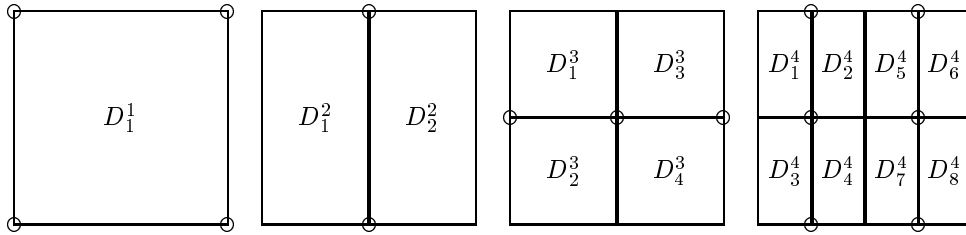


Рис. 2: Рекурсивное разбиение изображения и положение отсчетов соответствующего уровня иерархии

и восстанавливающие функции. При этом сигнал раскладывается на сумму ЛОХПСФ разного размера и местоположения. Были предложены конструктивные методы синтеза локальных однородных хорошо приспособленных сглаживающих и восстанавливающих функций (ЛОХПСФ, ЛОХПВФ) для заданного множества сигналов. Процедура синтеза ЛОХПСФ и ЛОХПВФ сведена к решению некоторой многоэкстремальной оптимизационной задачи. Была показана возможность синтеза ЛОХПВФ для ЛОХПСФ заданного вида. Задав ЛОХПСФ таким образом, чтобы существенными отсчетами являлись нечетные отсчеты сигнала, получим ЛОХПВФ для интерполяции четных отсчетов по нескольким соседним существенным отсчетам.

Пусть x_t , $1 \leq t \leq T$ — отсчеты исходного сигнала. Продлим его влево и вправо некоторым образом, например $x_t = x_1$, $t < 1$ и $x_t = x_T$, $t > T$. Тогда восстановленное (интерполированное) значение четного отсчета будет выражено следующим образом

$$(13) \quad \hat{x}_{2t} = \sum_{i=1}^m \gamma_i x_{2t-m+2i-1},$$

m — количество ненулевых коэффициентов γ_i ЛОХПВФ. Значение m выбирается четным и на практике лежит в диапазоне $[2 \dots 16]$. Приведем несколько примеров вектора $\bar{\gamma}$ для различных m :

- $m = 2$, $\bar{\gamma} = \langle 0.5, 0.5 \rangle$,
- $m = 4$, $\bar{\gamma} = \langle -0.125, 0.625, 0.625, -0.125 \rangle$,

- $m = 4$, $\bar{\gamma} = \langle -0.096, 0.596, 0.596, -0.096 \rangle$

Уровень k двоичного дерева содержит отсчеты, принадлежащие изображению P^k , но отсутствующие в изображениях предыдущих уровней иерархии (P^0, P^1, \dots, P^{k-1}). Таким образом, естественной будет попытка интерполяции отсчетов текущего уровня двоичного дерева по отсчетам вышележащих уровней, что будет соответствовать переходу от изображения P^{k-1} к изображению с большим уровнем детализации P^k .

Однако, поскольку при декодировании исходные значения отсчетов будут недоступны, для обеспечения эквивалентных результатов интерполяции при кодировании и декодировании следует использовать восстановленные значения отсчетов, а кодирование и декодирование производить по уровням иерархии от верхних к нижним. Кроме того, поскольку входные значения являются целочисленными, имеет смысл искать целочисленные интерполированные значения. При этом погрешность интерполяции (см. далее) всегда будет целым числом, что упрощает дальнейшую обработку и особенно кодирование. Полагая, что четыре угловых отсчета, которые не входят в двоичное дерево и составляют изображение P^0 будут переданы без искажений, т.е. $\tilde{P}^0 \equiv P^0$, можно привести выражение (13) к окончательному виду.

$$(14) \quad \hat{p}_{i,2j}^{2k+1} = \left[\sum_{h=1}^m \gamma_h \tilde{p}_{i,j-\frac{m}{2}+h}^{2k} \right],$$

$$1 \leq i \leq 2^k + 1, 1 \leq j \leq 2^k, 0 \leq k < ls$$

$$(15) \quad \hat{p}_{2i,j}^{2k+2} = \left[\sum_{h=1}^m \gamma_h \hat{p}_{i-\frac{m}{2}+h,j}^{2k+1} \right],$$

$$1 \leq i \leq 2^k, 1 \leq j \leq 2^{k+1} + 1, 0 \leq k < ls$$

Введем понятие погрешности интерполяции для отсчета, как разности исходного и интерполированного значений отсчета

$$(16) \quad \delta_{i,2j}^{2k+1} = p_{i,2j}^{2k+1} - \hat{p}_{i,2j}^{2k+1},$$

$$1 \leq i \leq 2^k + 1, 1 \leq j \leq 2^k, 0 \leq k < ls$$

$$(17) \quad \delta_{2i,j}^{2k+2} = p_{2i,j}^{2k+2} - \hat{p}_{2i,j}^{2k+2},$$

$$1 \leq i \leq 2^k, 1 \leq j \leq 2^{k+1} + 1, 0 \leq k < ls$$

Если узлам двоичного дерева вместо значений отсчетов поставить в соответствие погрешности интерполяции, получим так называемое разностное дерево. Хранение погрешности интерполяции для отсчетов позволяет восстановить их значение без потерь, таким образом разностное дерево равноценно исходному. В случае сжатия с потерями можно приписать узлам дерева величины $\tilde{\delta}_{i,j}^k$, которые представляют погрешность интерполяции отсчетов с заданной точностью. Точное определение таких величин можно задать различными способами. Так, в работе [4] предлагается следующее определение

$$(18) \quad \tilde{\delta}_{i,j}^k = \begin{cases} 0, & \text{если } |\delta_{i,j}^k| \leq \hat{\varepsilon}_k \\ \delta_{i,j}^k, & \text{в остальных случаях} \end{cases}$$

Подобный подход позволяет получить разностное дерево с бóльшим числом узлов с нулевым значением погрешности интерполяции, что активно используется при построении УДД (см. раздел 2.3).

Другое, более эффективное определение (19) кроме увеличения числа узлов с нулевым значением погрешности интерполяции существенно уменьшает мощность множества хранимых значений погрешности интерполяции, за счет чего можно улучшить эффективность последующего кодирования этих значений (см. раздел 2.5).

$$(19) \quad \tilde{\delta}_{i,j}^k = \left[\frac{\delta_{i,j}^k}{2\hat{\varepsilon}_k + 1} \right] (2\hat{\varepsilon}_k + 1)$$

При этом для упрощения алгоритмов можно хранить величины

$$(20) \quad \check{\delta}_{i,j}^k = \left[\frac{\delta_{i,j}^k}{2\hat{\varepsilon}_k + 1} \right]$$

Очевидно, значения $\check{\delta}_{i,j}^k$ можно восстановить по значениям $\delta_{i,j}^k$ при декодировании, поскольку величины $\hat{\varepsilon}_k$ полагаются известными.

Восстановленные значения отсчетов будут определяться по следующим соотношениям

$$(21) \quad \check{p}_{i,2j}^{2k+1} = \hat{p}_{i,2j}^{2k+1} + \check{\delta}_{i,2j}^{2k+1},$$

$$1 \leq i \leq 2^k + 1, 1 \leq j \leq 2^k, 0 \leq k < ls$$

$$(22) \quad \check{p}_{2i,j}^{2k+2} = \hat{p}_{2i,j}^{2k+2} + \check{\delta}_{2i,j}^{2k+2},$$

$$1 \leq i \leq 2^k, 1 \leq j \leq 2^{k+1} + 1, 0 \leq k < ls$$

Отметим, что как при использовании выражения (18), так и при использовании выражения (19) ошибка составит не более $\hat{\varepsilon}_k$. Таким образом, каждое из изображений

иерархической последовательности может быть восстановлено по разностному двоичному дереву с ошибкой не больше заданной.

2.3 Приведение двоичного дерева к УДД различной структуры

Для “реальных” изображений большая часть отсчетов будет интерполирована с погрешностью не более заданной. Это отсчеты $p_{i,j}^k$, для которых выполняется соотношение

$$(23) \quad |\delta_{i,j}^k| \leq \varepsilon_k$$

или, что эквивалентно при использовании выражения (18) или (19),

$$(24) \quad \tilde{\delta}_{i,j}^k = 0$$

Для таких отсчетов можно не хранить собственно значение отсчета или погрешность интерполяции, тем не менее их можно будет восстановить с заданной ошибкой.

Необходимо учитывать наличие узлов двоичного дерева, содержащих два отсчета. Поскольку число их относительно невелико, для упрощения алгоритмов обработки можно положить, что отсчеты в таких узлах отбрасываются только парами (если оба они удовлетворяют условию (24)), либо оба они будут сохранены. В таком случае можно ввести единый флаг наличия отсчета (отсчетов) для узла УДД

$$(25) \quad f_i^k = \begin{cases} 1, & \text{если узел } N_i^k \text{ содержит отсчет(ы)} \\ 0, & \text{в остальных случаях} \end{cases}$$

Следующий вид структуры УДД получается путем удаления из двоичного дерева поддеревьев из узлов, не содержащих отсчетов. Введем флаг наличия узла N_i^k в УДД

$$(26) \quad e_i^k = \begin{cases} 1, & \text{если узел } N_i^k \text{ присутствует в УДД} \\ 0, & \text{в противном случае} \end{cases}$$

$$= \begin{cases} f_i^k, & k = 2ls \\ 1, & f_i^k + e_{2i-1}^{k+1} + e_{2i}^{k+1} > 0, 1 \leq k < 2ls \\ 0, & f_i^k + e_{2i-1}^{k+1} + e_{2i}^{k+1} = 0, 1 \leq k < 2ls \end{cases}$$

Можно ввести дополнительное ограничение на удаление узлов и потребовать, чтобы УДД оставалось насыщенным. В этом случае может несколько снизиться число удаленных из дерева узлов, но упрощается структура УДД и, соответственно, уменьшаются затраты на ее кодирование.

Несомненно, описанные методы усложняют структуру двоичного дерева, делают ее нерегулярной. Это приводит к необходимости применения различных подходов для хранения помимо значений отсчетов или погрешностей интерполяции самой структуры УДД. Тривиальный подход состоит в хранении для каждого узла флага наличия отсчета (отсчетов) и флага (флагов) наличия дочерних узлов. Более эффективные методы приводятся в разделе 2.4.

2.4 Кодирование структуры УДД

Примем, что УДД является насыщенным и каждый узел содержит все приписанные ему отсчеты либо не содержит отсчетов вовсе. При этом структура такого УДД полностью описывается поставленными в соответствие каждому узлу N_i^k УДД флагом наличия отсчета (отсчетов) f_i^k и флагом наличия дочерних узлов

$$(27) \quad s_i^k = \begin{cases} 0, & k = 2ls \\ 0, & 1 \leq k < 2ls, e_{2i-1}^{k+1} e_{2i}^{k+1} = 0 \\ 1, & 1 \leq k < 2ls, e_{2i-1}^{k+1} e_{2i}^{k+1} = 1 \end{cases}$$

При фиксированном прямом порядке обхода УДД достаточно хранить только значения этих флагов, упорядоченные в соответствии с порядком обхода соответствующих им узлов УДД. Аналогичная ситуация наблюдается и при обходе узлов УДД по уровням. В обоих случаях по упорядоченной последовательности значений флагов можно восстановить структуру УДД.

Первое, что можно применить для более эффективного кодирования структуры дерева - это записать число полных уровней k_f , определяемое выражением

$$(28) \quad k_f = \max_{1 \leq k^* \leq 2ls} (k^* : \sum_{i=1}^{2^{k^*}-1} e_i^{k^*} = 2^{k^*}-1)$$

Кроме этого, можно хранить номер последнего уровня, содержащего хотя бы одну вершину.

$$(29) \quad k_m = \max_{1 \leq k^* \leq 2ls} (k^* : \sum_{i=1}^{2^{k^*}-1} e_i^{k^*} > 0)$$

Хранение этих величин позволит отказаться от хранения флага s_i^k на всех уровнях $k < k_f$ и на уровне k_m .

Для более эффективного действия этих методов можно дополнить некоторые уровни до полных фиктивными вершинами без отсчетов. Можно показать, что дополнять до полных нужно уровни, непосредственно следующие за полными и, при отсутствии статистического кодирования значений флагов, заполненные не менее чем на 75%. Исключение составляет последний уровень $k = k_m$. Поскольку дополнительные узлы на этом уровне не требуют флага наличия дочерних узлов, дополнение до полного становится оправданным при заполнении его на 50%.

С целью дальнейшего уменьшения объема хранимых данных применяется статистическое кодирование. Для этого флаги группируются по типам (коды наличия дочерних вершин и коды наличия отсчетов). Каждая такая группа рассматривается как последовательность бит, которые объединяются в байты.

Хранения флага наличия отсчета в вершине можно избежать, храня значения отсчетов или погрешности интерполяции для всех заданных вершин, в том числе и тех, где значение ошибки не превышает допустимого уровня. В простейшем случае число фиктивных отсчетов должно составлять не более $1/b$ от общего числа узлов УДД, т.к. значения отсчетов квантованы на 2^b уровней и хранение значения отсчета требует b бит, а флаг наличия отсчета - 1 бит. Применение разностного дерева и/или

статистического кодирования в общем случае изменяет это соотношение.

Необходимо отметить, что применение статистического кодирования к отсчетам изменяет приведенные соотношения до такой степени, что вполне оправданным становится хранение полного дерева и всех отсчетов. Такой подход, являясь оптимальным или очень близким к оптимальному по объему выходного потока на абсолютном большинстве реальных входных данных, обладает существенными преимуществами. Так, он позволяет отказаться от работы с древовидными структурами в памяти, заменяя обход дерева простым обходом отсчетов в заданном порядке. Это заметно упрощает как кодирующий, так и декодирующий алгоритмы, уменьшает объем необходимой для их работы памяти и увеличивает производительность.

2.5 Кодирование значений отсчетов и ошибок интерполяции

По аналогии с кодированием структуры, для кодирования отсчетов можно применить статистическое кодирование. Существуют различные алгоритмы статистического кодирования можно применять различные алгоритмы. В настоящее время наиболее часто употребляемыми являются алгоритм Хаффмана [9] и различные реализации арифметического кодирования [10, 11]. Каждый из методов имеет свои преимущества. Так, кодирование Хаффмана имеет несколько более высокую производительность, но коэффициент сжатия ограничен 8 (для входного алфавита мощностью 256). Кроме того, производительность кодирования Хаффмана теряется при работе с адаптивными статистическими моделями. Арифметическое кодирование, в свою очередь позволяет произвольно близко подойти к теоретическому пределу коэффициента сжатия для алфавитного статистического кодирования и более применимо к адаптивным статистическим моделям и входным алфавитам большой мощности.

Существенно лучшего эффекта можно достичь, если перейти от кодирования абсолютных значений отсчетов к кодированию ошибок интерполяции (к разностному дереву). При этом можно проводить вычисление ошибок интерполяции по модулю 2^b , что позволяет получать значения в диапазоне $[0, 2^b - 1]$ вместо $[1 - 2^b, 2^b - 1]$. Такой подход позволяет уменьшить входной алфавит алгоритма статистического кодирования и, соответственно, повысить его производительность. Другой, более эффективный с точки зрения степени сжатия подход, — использование расширенного входного алфавита $[1 - 2^b, 2^b - 1]$. Заметное уменьшение стоимости кодирования может быть получено при использовании совместно со статистическим кодированием статистической модели, учитывающей текущее интерполированное значение при кодировании ошибки интерполяции. Так, например, при $b = 8$ и текущем интерполированном значении отсчета $\hat{p}_{i,j}^k = 253$ значение ошибки интерполяции может находиться только в диапазоне $\delta_{i,j}^k \in [-253, 2]$. Вероятность остальных значений должна быть принята равной нулю. С соответствующими поправками этот метод применим к значениям

$\tilde{\delta}_{i,j}^k$ и $\check{\delta}_{i,j}^k$.

$$(30) \quad \check{\delta}_{i,j}^k \in \left[\left[\frac{-\hat{p}_{i,j}^k}{2\hat{\varepsilon}_k + 1} \right], \left[\frac{2^b - 1 - \hat{p}_{i,j}^k}{2\hat{\varepsilon}_k + 1} \right] \right]$$

В случае кодирования 16-разрядных значений отсчетов непосредственное применение статистического кодирования затруднено вследствие большой мощности входного алфавита. При этом возникают следующие проблемы.

- При полуадаптивном кодировании объем накладных расходов на хранение информации о структуре дерева Хаффмана или таблицы частот недопустимо велик.
- Применение адаптивных алгоритмов статистического кодирования неоправданно из-за слишком медленного заполнения таблицы частот и больших затрат системных ресурсов на поддержание и модификацию таковой таблицы.

В качестве приемлемой альтернативы для разностных деревьев при $b = 16$ предлагается следующий подход. Подлежащая хранению величина $\check{\delta}_{i,j}^k$ разбивается на старшую и младшую части.

$$(31) \quad \lambda_{i,j}^k = \left\lfloor \frac{\check{\delta}_{i,j}^k}{256} \right\rfloor$$

$$(32) \quad \mu_{i,j}^k = \check{\delta}_{i,j}^k \bmod 256$$

При этом старшая часть $\lambda_{i,j}^k$ байт кодируется с применением адаптивной статистической модели, а в отношении младшей $\mu_{i,j}^k$ принимается гипотеза о равномерном распределении вероятностей при $\lambda_{i,j}^k \neq 0$ или используется собственная адаптивная статистическая модель при $\lambda_{i,j}^k = 0$.

3 Полноцветные изображения

В случае полноцветных изображений атрибутом пикселя является вектор в некотором цветовом пространстве (RGB, YCbCr, CMYK и др.). В этом случае производится разбиение изображения на цветовые слои с последующей независимой обработкой каждого слоя как полутонового изображения. Этот подход обладает следующими важными преимуществами.

- Унифицированная обработка как полутоновых, так и полноцветных изображений.
- Независимость алгоритма от количества цветовых слоев (размерности цветового пространства) в исходном изображении.
- Возможность независимого определения для каждого цветового слоя допустимой погрешности интерполяции и уровня детализации.

В случае, если имеется возможность выбора применяемого цветового пространства, рекомендуется выбирать цветное пространство, осуществляющее разбиение на составляющие яркости и цветности, например, YUV. Психологические особенности восприятия изображений человеком таковы, что есть возможность сжатия слоев цветности с меньшим уровнем детализации и большей допустимой ошибкой по сравнению со слоем яркости при аналогичном субъективном качестве восстановленного изображения. Кроме того, слои яркости и цветности менее коррелируют между собой, чем цветовые слои в пространстве RGB, что позволяет достичь большего коэффициента сжатия при том же уровне допустимых ошибок.

физические особенности восприятия изображений человеком таковы, что есть возможность сжатия слоев цветности с меньшим уровнем детализации и большей допустимой ошибкой по сравнению со слоем яркости при аналогичном субъективном качестве восстановленного изображения. Кроме того, слои яркости и цветности менее коррелируют между собой, чем цветовые слои в пространстве RGB, что позволяет достичь большего коэффициента сжатия при том же уровне допустимых ошибок.

4 Изображения с механизмом индексации цвета

Изображения с механизмом индексации цвета обладают определенной спецификой, делающей метод интерполяции по ЛОХПБФ применимым только в некоторых частных случаях. Интерполяция отсчетов такой природы должна учитывать следующие особенности.

- В общем случае логическая связь между значением индекса и соответствующим вектором цветового пространства отсутствует. Как следствие, в пространстве индексов погрешность интерполяции не имеет четко выраженной смысловой нагрузки. Сходные или часто встречающиеся совместно цвета могут иметь произвольно различные индексы.
- На практике наиболее востребовано сжатие изображений такого класса без потерь.
- Дальнейшие методы обработки изображений этого класса могут не допускать переназначения соответствий между индексами и векторами в цветовом пространстве (индексы могут нести произвольную смысловую нагрузку).

С учетом вышеперечисленных особенностей был предложен следующий подход к интерполяции отсчетов.

Рассматривается кодирование такого класса изображений только без потерь.

Интерполированным значением отсчета считается значение, имеющее наибольшую условную вероятность появления в данном контексте, где контекстом считается значение соответствующих существенных отсчетов, т.е. отсчетов вышележащих уровней структуры, по которым производится интерполяция. Поскольку таблица таких значений для каждого контекста должна храниться вместе со сжатым изображением и объем ее растет как экспонента числа существенных отсчетов, ограничим контекст значением двух существенных отсчетов.

Поскольку объем таблицы растет как квадрат мощности алфавита индексов, будем рассматривать в качестве контекста не само значение индексов существенных отсчетов, а их классы. Число таких классов ограничим некоторой величиной N_{cc} и будем строить эти классы следующим образом. Если число различных индексов в изображении не превышает величины N_{cc} , каждый класс будет содержать строго один индекс. В противном случае первые $N_{cc} - 1$ классов будут содержать $N_{cc} - 1$ самых часто встречающихся индексов, а один класс - все остальные индексы.

При построении разностного дерева вместо погрешностей интерполяции будем сохранять собственно значение исходного индекса, но при кодировании его будем применять условные вероятности, рассматривая в качестве контекста интерполированное значение. Наличие большого количества таблиц частот при таком подходе делает особенно привлекательным применение различного рода адаптивных методов статистического кодирования. Таким образом, при кодировании непосредственных значений индексов применяется одна из адаптивных статистических моделей, работающих с входным алфавитом соответствующей мощности. В случае разностного дерева применяется модель с условными вероятностями, количество значений контекста которой равно

$$(33) \quad N_{con} = \frac{(N_{cc} - 1)N_{cc}}{2}$$

5 Практические результаты

На практике было показано, что для любых типов входных данных наиболее эффективное кодирование обеспечивается при переходе от абсолютных значений отсчетов к ошибкам интерполяции (к разностному дереву).

Рассмотрим эффективность предложенной технологии адаптивного сжатия на примере аэрофотоснимка местности (полутонное изображение), закодированного с различными уровнями допустимой погрешности по сравнению с другими методами сжатия изображений. Исходная матрица имеет размеры 8192×8192 пиксель, объем $67\,108\,864$ байт, значения отсчетов рассматриваются как целочисленные в диапазоне $[0, 255]$. Объемы файлов, полученных различными методами и время сжатия/восстановления, представлены в таблице 1, файл FBTR получен с использованием описанных методов. Тестирование проводилось на машине с процессором Pentium-III 750 MHz, 512Mb RAM.

Таблица 1: Сравнительная эффективность различных алгоритмов при кодировании полутонного изображения

Допустимая ошибка ε_{max}	0	2	20
PNG	40 063 210 64 с/32 с	—	—
JPEG [13]	—	39 010 088 31 с/32 с	12 604 326 30 с/32 с
FBTR	36 162 352 36 с/47 с	17 805 144 29 с/36 с	1 644 523 25 с/32 с

Необходимо отметить возможность задания различной допустимой погрешности для различных уровней дерева. Так, для рассмотренного полутонного изображения в задачах визуализации было принято вполне приемлемым качество, полученное при кодировании с допустимой погрешностью $\varepsilon_{max} = 30$ при условии ограничения погрешности на верхних уровнях значением $\varepsilon_{hi} = 10$. Объем закодированного с такими параметрами изображения составил $2\,228\,579$ байт. Сходные результаты получены и

для полноцветных изображений применением алгоритма к каждому из слоев яркости и цветности.

Хорошие результаты были получены и для кодирования без потерь изображений с механизмом индексации цвета. Рассмотрим это на примере картографического изображения в разрешении 423 dpi (межточечный интервал 60 микрон). Характеристики исходного изображения: размер 5900×7251 пиксель, размер атрибута пикселя в битах — 8, размер дампа изображения в байтах — $42\,780\,900$. Результаты сжатия этого изображения различными алгоритмами приведены в таблице 2.

Таблица 2: Сравнительная эффективность различных алгоритмов при кодировании изображения с механизмом индексации цвета

Формат	Объем файла
Targa	11 069 500
GIF	6 503 903
PIRH [14]	5 411 601
RRE [15]	5 338 772
FBTR	3 670 506

Список литературы

- [1] Васин Ю.Г. Сжатие исходного описания ЭКГ-сигналов. В кн. "Теория и практика разработки автоматизированных медицинских информационных систем" Труды Республиканской конференции, Киев, 1975. С.69–75.
- [2] Васин Ю.Г. Хорошо приспособленные базисы и задачи обработки экспериментальной информации: учебное пособие. Горький: ГГУ, 1979.
- [3] Васин Ю.Г. Оптимизация описания исходных данных в диалоговых системах решения задач классификации. — В кн. "Современное состояние теории исследования операций" Москва:Наука, 1979. С.424–450.
- [4] Васин Ю.Г., Бакараева В.П. Рекуррентные алгоритмы адаптивного сжатия с использованием хорошо приспособленных локальных восстанавливающих функций. — Математическое обеспечение САПР: Межвуз.сб. Горький: ГГУ, 1978, вып.1.
- [5] Васин Ю.Г. Хорошо приспособленные локальные однородные методы обработки графической информации. — Автоматизация обработки сложной графической информации: Межвуз.сб. Горький: ГГУ, 1984.
- [6] Васин Ю.Г. Эффективность различных стратегий обработки видеoinформации на базе локальных однородных функций. — Методы и средства обработки графической информации: Межвуз.сб. Горький: ГГУ, 1986. С.4–47.
- [7] Неймарк Ю.И., Васин Ю.Г. Кодирование больших массивов информации в связи с задачами распознавания образов. — Изв. вузов. Радиофизика, 1968. С.1081–1085.

- [8] Васин Ю.Г., Савина Т.В. Кодирование ЭКГ неравномерными отсчетами. — Сборник “Биологическая и медицинская кибернетика” Ч.4. Москва, 1974. С.73–74.
- [9] Huffman D.A. 1952. A method for the construction of minimum redundancy codes. In Proceedings of the Institute of Electrical and Radio Engineers 40, 9 (Sept.), p.1098–1101.
- [10] Langdon G.G. 1984. An introduction to arithmetic coding. IBM J.Res.Dev. 28, 2 (Mar.), p.135–149.
- [11] Rubin F. 1979. Arithmetic stream coding using fixed precision registers. IEEE Trans.Inf.Theory IT-25, 6 (Nov.), p.672–675.
- [12] Рябко Б.Я., Фионов А.Н. Эффективный метод адаптивного арифметического кодирования для источников с большими алфавитами — Проблемы передачи информации. 1999. Т.35. Вып.4. С.95–108.
- [13] Pennebaker W.B., Mitchell J.L. JPEG: Still Image Data Compression Standard. — International Thomson Computer Press, London, UK, 1993.
- [14] Кудин А.В. Алгоритм PIRH-кодирования технических изображений. — Вестник Нижегородского государственного университета. Математическое моделирование и оптимальное управление. Нижний Новгород: Изд-во Нижегородского ун-та, 1998. Вып.1 (18).
- [15] Кудин А.В., Жерздев С.В. Алгоритм RRE-кодирования технических изображений. — Вестник Нижегородского университета. Математическое моделирование и оптимальное управление. Н.Новгород: Изд-во ННГУ. Вып.2 (19), 1998.

Об авторах

Васин Юрий Григорьевич, директор НИИ Прикладной Математики и Кибернетики ННГУ, д.т.н., профессор, член-корр. АТН РФ.

Жерздев Сергей Владимирович, аспирант ННГУ, НИИ ПМК ННГУ.