

АДАПТИВНЫЙ АЛГОРИТМ ИНТЕРПОЛЯЦИИ ДЛЯ ИЕРАРХИЧЕСКОЙ КОМПРЕССИИ ИЗОБРАЖЕНИЙ

М.В. Гашиков, Н.И. Глузов, В.В. Сергеев

Институт систем обработки изображений РАН

Самарский государственный аэрокосмический университет

Аннотация

В работе предлагается новый адаптивный алгоритм интерполяции для иерархической компрессии изображений. Алгоритм основан на использовании параметризованного решающего правила, которое настраивается на каждое конкретное изображение в процессе его обработки. Производится сравнение разработанного алгоритма интерполяции с известными.

Введение

Иерархическая компрессия изображений [1-3] основана на многократном прореживании исходного изображения, использовании прореженного изображения для интерполяции пропущенных отсчетов и последующем кодировании интерполяционных ошибок. Поэтому эффективность иерархической компрессии во многом определяется эффективностью используемого алгоритма интерполяции. Применяемые в настоящее время алгоритмы интерполяции [4, 6], как правило, заключаются в простом усреднении по ближайшим отсчетам и никак не учитывают особенности обрабатываемого изображения. Поэтому актуальной является задача разработки нового адаптивного алгоритма интерполяции, который был бы более эффективен за счет настройки на каждое конкретное изображение.

В данной работе предлагается новый адаптивный алгоритм интерполяции, который настраивается на изображение за счет использования нескольких интерполирующих функций и параметризованного решающего правила. Рассматривается использование этого алгоритма для конкретного метода компрессии, реализующего иерархический подход.

Метод компрессии на основе ИСИ

Будем рассматривать метод компрессии на основе иерархической сеточной интерполяции (ИСИ) [5,6]. Пусть $\mathbf{X} = \{x(m, n)\}$ – исходное изображение. Рассмотрим представление изображения \mathbf{X} в виде объединения иерархических уровней \mathbf{X}_l :

$$\mathbf{X} = \bigcup_{l=0}^{L-1} \mathbf{X}_l, \quad (1)$$

$$\mathbf{X}_{L-1} = \{x_{L-1}(m, n)\}, \quad (2)$$

$$\mathbf{X}_l = \{x_l(m, n)\} / \{x_{l+1}(m, n)\}, \quad l < L-1, \quad (3)$$

где L – количество уровней,
а $\{x_l(m, n)\}$ – массив отсчетов изображения, взятых с шагом 2^l по каждой координате.

Представление (1)-(3) дает возможность при компрессии и декомпрессии обрабатывать уровни последовательно, начиная со старшего уровня \mathbf{X}_{L-1} , причем отсчеты каждого более старшего (менее детального) уровня будут использоваться для интерполяции отсчетов младшего (более детального) уровня.

Количество уровней выбирается таким образом, чтобы объем старшего (наиболее прореженного) уровня был пренебрежимо мал по сравнению с объемом данных всего изображения, поэтому старший уровень может быть записан в сжатое изображение и без компрессии. Процедура компрессии любого из следующих младших уровней $\mathbf{X}_l, l < L-1$ включает следующие этапы (см. также рис. 1):

1) Интерполяция.

Производится интерполяция отсчетов $\{x_l(m, n)\}$ уровня \mathbf{X}_l на основе отсчетов $\{\bar{x}_k(m, n), k > l\}$ уровней $\{\bar{\mathbf{X}}_k, k > l\}$, уже прошедших компрессию и восстановление:

$$\hat{x}_l(m, n) = P \left(\bigcup_{k=l+1}^{L-1} \{\bar{x}_k(m, n)\} \right), \quad (4)$$

где $\hat{x}_l(m, n)$ – интерполирующие значения отсчетов, а $P(\dots)$ – функция, задающая в общем виде некоторый интерполятор. Конкретные примеры интерполяторов приведены в следующем разделе.

2) Вычисление постинтерполяционных остатков

Вычисляется массив разностей истинных и интерполирующих значений отсчетов («постинтерполяционных остатков»):

$$f_l(m, n) = x_l(m, n) - \hat{x}_l(m, n). \quad (5)$$

3) Квантование постинтерполяционных остатков

Выполняется квантование постинтерполяционных остатков (5). В данной работе используется квантователь, гарантирующий, что декомпрессированное изображение будет отличаться от исходного не более, чем на величину заданной максимальной ошибки ε_{\max} :

$$q_l(m, n) = \text{sign}(f_l(m, n)) \left[\frac{|f_l(m, n)| + \varepsilon_{\max}}{2\varepsilon_{\max} + 1} \right], \quad (6)$$

где $[..]$ обозначает выделение целой части.

4) Вычисление восстановленных значений отсчетов

По квантованным значениям постинтерполяционных остатков вычисляются восстановленные значения $\bar{x}_l(m, n)$ отсчетов уровня. Очевидно, что для этого необходимо сначала вычислить восстановленные значения постинтерполяционных остатков:

$$\bar{f}_l(m, n) = q_l(m, n)(2\varepsilon_{\max} + 1), \quad (7)$$

а затем осуществить собственно восстановление:

$$\bar{x}_l(m, n) = \bar{f}_l(m, n) + \hat{x}_l(m, n). \quad (8)$$

Эти восстановленные отсчеты будут нужны для интерполяции более младших иерархических уровней $\{X_k, k < l\}$.

5) Статистическое кодирование

Осуществляется статистическое кодирование квантованных постинтерполяционных остатков. Поскольку распределение их вероятностей, как правило, является существенно неравномерным, в результате кодирования можно достичь значительного сокращения объема данных, которое и является целью выполняемой обработки изображения. На этом описании процедуры компрессии закончено.

Процедура декомпрессии уровня $X_l, l < L-1$ включает следующие этапы:

1) Декодирование

Производится декодирование квантованных значений постинтерполяционных остатков (6).

2) Интерполяция

Производится интерполяция вида (4) по восстановленным отсчетам.

3) Восстановление

Как и при компрессии, производится деквантование (7) и вычисление восстановленных значений отсчетов (8). На этом описании процедуры декомпрессии закончено.

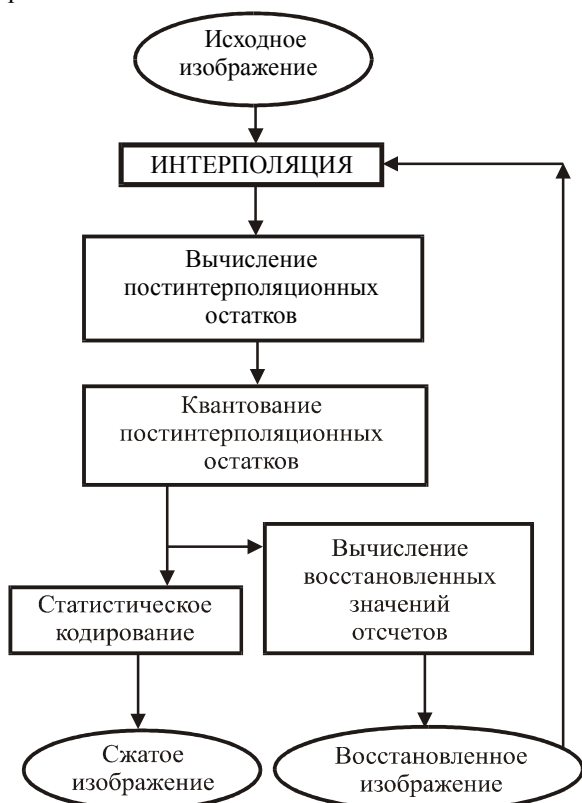


Рис. 1. Схема процедуры компрессии

Известные схемы интерполяции

Основными требованиями к интерполятору являются как можно более высокая точность при ма-

лой вычислительной сложности. Поэтому обычно для интерполяции используют алгоритмы (схемы), которые заключаются в простом усреднении по ближайшим восстановленным отсчетам (см. рис 2). Эти схемы работают одинаково и при компрессии и при декомпрессии.

Будем отдельно рассматривать два типа интерполируемых отсчетов. Отсчеты вида $x_l(2m+1, 2n+1)$ будем называть «центральными» (см. рис. 3). Нетрудно видеть, что четыре ближайших восстановленных отсчета более старшего уровня находятся по диагонали от центрального. Отсчеты вида $x_l(2m+1, 2n)$ и $x_l(2m, 2n+1)$ будем называть "крайними" (см. рис. 4). Два ближайших восстановленных отсчета более старшего уровня находятся по вертикали или по горизонтали от крайнего. Укажем способы интерполяции одного крайнего и центрального отсчетов для каждой схемы.

Схема (I):

$$\hat{x}_l(2m+1, 2n) = \langle \bar{x}_l(2m, 2n), \bar{x}_l(2m+2, 2n) \rangle, \quad (9)$$

$$\hat{x}_l(2m+1, 2n+1) = \langle \bar{x}_l(2m, 2n), \bar{x}_l(2m+2, 2n), \bar{x}_l(2m, 2n+2), \bar{x}_l(2m+2, 2n+2) \rangle \quad (10)$$

где $\langle \dots \rangle$ обозначает усреднение с последующим округлением до ближайшего целого.

Схема (II):

Сначала с помощью выражения вида (9) интерполируются все крайние отсчеты. Затем для них вычисляются восстановленные значения, которые используются для интерполяции центральных отсчетов

$$\hat{x}_l(2m+1, 2n+1) = \langle \bar{x}_l(2m+1, 2n), \bar{x}_l(2m, 2n+1), \bar{x}_l(2m+1, 2n+2), \bar{x}_l(2m+2, 2n+1) \rangle \quad (11)$$

Схема (III):

Сначала с помощью выражения вида (10) интерполируются все центральные отсчеты. Затем для них вычисляются восстановленные значения, которые используются для интерполяции крайних отсчетов

$$\hat{x}_l(2m+1, 2n) = \langle \bar{x}_l(2m, 2n), \bar{x}_l(2m+2, 2n), \bar{x}_l(2m+1, 2n-1), \bar{x}_l(2m+1, 2n+1) \rangle \quad (12)$$

Достоинством рассмотренных известных схем является низкая вычислительная сложность. Общим недостатком этих схем является то, что они всегда работают одинаково, т.е. никак не учитывают ни особенностей обрабатываемого изображения, ни локальных особенностей обрабатываемого участка.

Адаптивная схема интерполяции

Основной идеей адаптивного интерполятора является использование не одной, а нескольких интерполирующих функций. Решение о том, какая именно из этих функций будет использована в каждой конкретной точке изображения, принимается на основе решающего правила, зависящего от парамет-

ров. При этом собственно обработке данных предшествует процесс обучения (настройки решающего правила), который заключается в отыскании оптимальных значений этих параметров.

При работе этого алгоритма, также как и для схемы (III), сначала производится интерполяция всех центральных отсчетов, а затем, на основе восстановленных значений центральных отсчетов, осуществляется интерполяция крайних отсчетов. Рассмотрим три интерполирующие функции центрального отсчета (см. рис. 3).

$$\hat{x}_l^{(0)}(2m+1, 2n+1) = \langle \bar{x}_l(2m, 2n), \bar{x}_l(2m+2, 2n+2) \rangle \quad (13)$$

$$\hat{x}_l^{(1)}(2m+1, 2n+1) = \langle \bar{x}_l(2m, 2n), \bar{x}_l(2m+2, 2n), \bar{x}_l(2m, 2n+2), \bar{x}_l(2m+2, 2n+2) \rangle \quad (14)$$

$$\hat{x}_l^{(2)}(2m+1, 2n+1) = \langle \bar{x}_l(2m+2, 2n), \bar{x}_l(2m, 2n+2) \rangle \quad (15)$$

Выбор функции, которая будет использована для интерполяции в каждой точке, осуществляется на основе признака

$$\mu_l(2m+1, 2n+1) = |\bar{x}_l(2m, 2n) - \bar{x}_l(2m+2, 2n+2)| - |\bar{x}_l(2m, 2n+2) - \bar{x}_l(2m+2, 2n)| \quad (16)$$

с помощью зависящего от параметров α и β решающего правила

$$F_R(\mu, \alpha, \beta) = \begin{cases} 0, & \mu < \alpha; \\ 1, & \alpha \leq \mu \leq \beta; \\ 2, & \mu > \beta; \end{cases} \quad (17)$$

которое в каждой точке по значению признака определяет номер используемой интерполирующей функции. Предполагается, что оба параметра целочисленные, причем $\alpha \leq 0$, а $\beta \geq 0$. Собственно адаптивный интерполятор задается выражением

$$\hat{x}_l(m, n) = \hat{x}_l^{(j)}(m, n), \quad j = F_R(\mu_l(m, n), \alpha, \beta). \quad (18)$$

Поясним такое построение интерполятора. Нетрудно видеть, что большие значения признака (18) соответствуют, например, ситуации, когда через интерполируемый центральный отсчет проходит диагональный контур, имеющий направление «слева снизу вправо вверх». Причем чем лучше выражен контур, тем больше значение признака. Если признак больше порогового значения β , то будет выбрана интерполирующая функция (15), реализующая интерполяцию «вдоль контура» (см. рис. 3-(3)). В контурных точках такая интерполяция является более точной, за счет чего и достигается выигрыш.

Аналогичным образом рассматривается ситуация с диагональным контуром направления «слева сверху вправо вниз» (см. рис. 2-(1)), которая «регулируется» параметром α . В случае, если выраженного диагонального контура не присутствует, то используется «обычное» усреднение вида (14).

Параметры α и β играют роль пороговых величин, на которых происходит переключение на интерполяцию «вдоль контура». Чем меньше на изо-

бражении контуров, тем больше по модулю значения этих параметров.

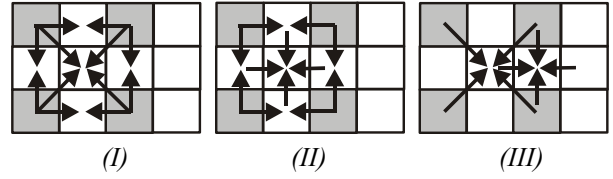


Рис. 2. Известные схемы интерполяции

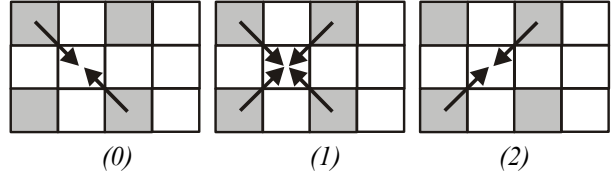


Рис. 3. Адаптивная интерполяция центрального отсчета

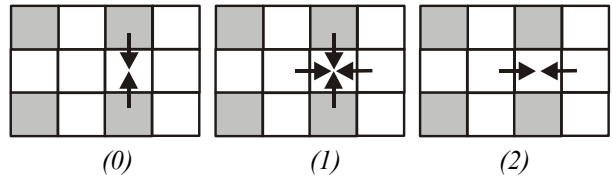


Рис. 4. Адаптивная интерполяция крайнего отсчета

Перед собственно применением интерполятора (18) производится его обучение (настройка на конкретное изображение), которое заключается в отыскании оптимальных значений параметров α и β . Для этого в данной работе используется критерий минимизации суммарной ошибки интерполяции

$$E_l^{(\Sigma)}(\alpha, \beta) = \sum_{(m,n) \in I_l} |\hat{x}_l(m, n) - x_l(m, n)| \rightarrow \min_{\alpha, \beta} \quad (19)$$

где I_l – множество индексов центральных отсчетов уровня X_l .

Рассмотрим алгоритм поиска параметров. Двухпараметрическая задача оптимизации (19) допускает декомпозицию на две однопараметрические задачи. Для этого рассматриваются множества индексов $I_l^{(-)}$, $I_l^{(0)}$ и $I_l^{(+)}$, для которых значение признака $\mu_l(m, n)$ соответственно отрицательно, равно нулю и положительно. Тогда суммарная ошибка интерполяции (19) может быть записана в виде

$$E_l^{(\Sigma)}(\alpha, \beta) = E_l^{(-)}(\alpha) + E_l^{(0)} + E_l^{(+)}(\beta), \quad (20)$$

где $E_l^{(+)}(\beta)$ – суммарная ошибка интерполяции по точкам с положительным значением признака

$$E_l^{(+)}(\beta) = \sum_{(m,n) \in I_l^{+}} |\hat{x}_l(m, n) - x_l(m, n)|, \quad (21)$$

которая зависит только от одного параметра β . Величины $E_l^{(-)}(\alpha)$ и $E_l^{(0)}$ вводятся аналогичным образом через множества $I_l^{(-)}$ и $I_l^{(0)}$.

Теперь оптимальное значение параметра β можно определить как минимум одномерной функ-

ции (21) на интервале $[0, M)$, где $M = \max\{x(m, n)\} + 1$. Для этого в процессе обучения заполняется матрица $\mathbf{E}^{(+)} = \{E_{i,k}^{(+)} : i = 1, 2; k \in [1, M]\}$:

$$E_{i,k}^{(+)} = \sum_{(m,n) \in I_i^{(k)}} \left| \tilde{x}_l^{(i)}(m, n) - x_i(m, n) \right|, \quad (22)$$

где $I_i^{(k)}$ – множество индексов, для которых значение признака $\mu_i(m, n)$ равно k . После того, как эта матрица заполнена, значение минимизируемой функции (21) вычисляется для всех значений аргумента с помощью следующих выражений:

$$E_i^{(+)}(0) = \sum_{\mu=1}^{M-1} E_{2,\mu}^{(+)}, \quad (23)$$

$$E_i^{(+)}(\beta) = E_i^{(+)}(\beta - 1) - E_{2,\beta}^{(+)} + E_{1,\beta}^{(+)}, \quad (24)$$

Следует отметить, что при проведении расчета (23)-(24) и при собственно поиске минимума функции (21) производится работа не с изображением, а со вспомогательной матрицей \mathbf{E} , благодаря чему вычислительная сложность этого этапа пренебрежимо мала и не зависит от размера изображения.

Определение оптимального значения параметра α осуществляется аналогичным образом, исходя из минимизации функции $E_i^{(-)}(\alpha)$, с использованием матрицы $\mathbf{E}^{(-)}$, которая строится аналогично матрице $\mathbf{E}^{(+)}$. Заполнение обеих матриц осуществляется за один проход по изображению и добавляет к вычислительной сложности 4 операции на отсчет. После того, как оптимальные значения параметров определены, они сохраняются в сжатом изображении и используются для обработки интерполятором (18).

Таким образом, алгоритм интерполяции центрального отсчета на этапе компрессии включает следующие шаги:

- 1) Обучение: вычисление оптимальных значений параметров (первый проход по изображению):
 - а) вычисление интерполирующих значений (13)-(15) и признака (16) для всех центральных отсчетов;
 - б) заполнение матриц $\mathbf{E}^{(-)}$ и $\mathbf{E}^{(+)}$;
 - в) отыскание оптимальных значений параметров α и β как минимумов функций $E_i^{(-)}(\alpha)$ и $E_i^{(+)}(\beta)$ с использованием выражений вида (23)-(24);
- 2) Сохранение оптимальных значений параметров α и β в сжатом изображении.
- 3) Собственно интерполяция (второй проход по изображению), которая заключается в применении функции (18) ко всем центральным отсчетам при оптимальных значениях параметров α и β .

В отличие от известных интерполяционных схем, разработанный алгоритм на этапе декомпрессии работает не так, как на этапе компрессии. А

именно, при декомпрессии интерполяция не требует обучения и включает следующие шаги:

- 1) Чтение из сжатого изображения оптимальных значений параметров α и β .
- 2) Собственно интерполяция (единственный проход по изображению), которая заключается в применении функции (18) ко всем центральным отсчетам при оптимальных значениях параметров α и β .

Как при компрессии, так и при декомпрессии, после интерполяции центральных отсчетов для них вычисляются восстановленные значения, которые используются для интерполяции крайних отсчетов. Интерполятор крайних отсчетов работает в целом аналогично интерполятору центрального отсчета. Интерполяция по-прежнему производится по четырем ближайшим отсчетам, но теперь они расположены не по диагонали, а по вертикали и по горизонтали от интерполируемого (см. рис. 4). Другими словами, вся ситуация «поворачивается» на 45 градусов. Соответственно, при интерполяции будут отслеживаться не диагональные, а вертикальные и горизонтальные контура.

Важно подчеркнуть также тот факт, что из построения адаптивного интерполятора видно, что в вырожденном случае (при $\alpha = -M + 1$, $\beta = M - 1$) он сводится к схеме (III). Другими словами, по точности построенный интерполятор гарантированно будет работать не хуже, чем интерполятор (III).

Таким образом, можно сформулировать следующие преимущества разработанного адаптивного интерполятора:

1. Учет локальных особенностей изображения за счет использования нескольких интерполирующих функций;
2. Настройка на каждое конкретное изображение за счет использования параметризованного решающего правила;
3. Гарантия точности не хуже, чем у схемы (III);
4. Структурно простая схема обработки.

Сравнение интерполяционных схем

Для определения эффективности разработанного адаптивного интерполятора, который далее будем называть схемой (IV), было проведено его сравнение с известными схемами (I), (II), (III), которые описываются выражениями (9)-(12). В задаче компрессии эффективность любой интерполяционной схемы определяется ее вычислительной сложностью и степенью сжатия, которая может быть достигнута с применением этой схемы.

Если считать, что деление на степени двойки не требует арифметических операций, то все четыре описанные схемы реализуются исключительно аддитивными операциями. Вычислительные сложности для всех схем в аддитивных операциях на отсчет приведены в таб. 1.

Из таблицы видно, что вычислительная сложность разработанной схемы (IV) больше, чем сложность из-

вестных схем интерполяции, однако для большинства приложений является вполне допустимой.

Таблица 1. Вычислительная сложность интерполяционных схем

Схема интерполяции	(I)	(II)	(III)	(IV)
Вычислительная сложность при компрессии	$\frac{5}{3}$	$\frac{5}{3}$	3	11
Вычислительная сложность при декомпрессии	$\frac{5}{3}$	$\frac{5}{3}$	3	6

Для сравнения схем интерполяции по степени компрессии изображений в рамках метода ИСИ был проведен вычислительный эксперимент. Для эксперимента использовались изображения типа «космический снимок», одно из которых показано на рис. 5.



Рис. 5. Изображение вида «космический снимок»

Для всех изображений схема (IV) обеспечила меньший объем сжатого изображения, чем схемы (I), (II), (III). Для количественного описания этого выигрыша использовались величины вида

$$\Delta^{(I)} = \frac{S^{(I)} - S^{(IV)}}{S^{(I)}} 100\%, \quad (25)$$

где $S^{(I)}$, $S^{(IV)}$ - размеры сжатых изображений для схем (I) и (IV) соответственно. Аналогичным образом вычислялись величины $\Delta^{(II)}$ и $\Delta^{(III)}$. Типичные результаты, полученные при различных максимальных ошибках для изображения на рис. 5, показаны на рис. 6.

Как видно из рис. 6, величины вида (25) для всех трех известных схем всегда положительны. Следовательно, на всем рассматриваемом диапазоне максимальных ошибок разработанная схема (IV) имеет устойчивое преимущество, которое, вообще говоря, увеличивается с ростом ошибки. При этом преимущество над самой лучшей из известных схем достигает 6-7% на половине рассматриваемого диапазона, что на практике позволяет обеспечить значительную экономию емкости запоминающих устройств.

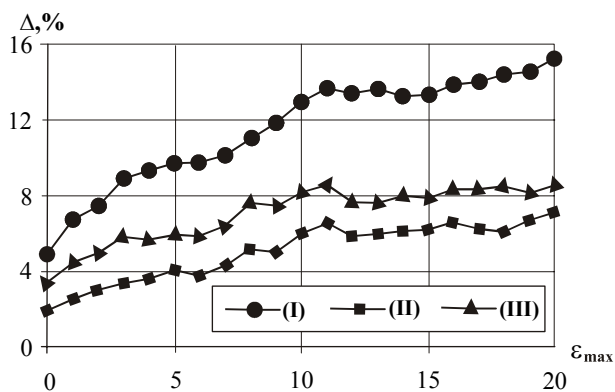


Рис. 6. Оценка выигрыша в размере сжатого изображения разработанной схемы интерполяции над известными схемами

Заключение

В работе предложен новый адаптивный алгоритм интерполяции для иерархической компрессии изображений. Данный алгоритм обеспечивает выигрыш по степени сжатия по сравнению с известными алгоритмами интерполяции. Разработанный алгоритм может быть эффективно использован в прикладных задачах, связанных с обработкой, хранением и передачей сжатых цифровых изображений.

Литература

1. Александров В.В., Горский Н.Д. Представление и обработка изображений: Рекурсивный подход // Л.: Наука, 1985. 192 с.
2. Kortman C. M. Redundancy Reduction – A Practical Method of Data Compression, Proc. IEEE, 55, No. 3 (Marth, 67). P. 253-263.
3. Васин Ю. Г., Бакарева В. П. Рекуррентные алгоритмы адаптивного сжатия с использованием хорошо приспособленных локальных восстанавливающих функций. – Математическое обеспечение САПР: Межвуз. сб. Горький: ГГУ, 1978, вып. I
4. Ефимов В. М., Колесников А. Н. Оценка эффективности иерархических и построчных алгоритмов сжатия полутоновых изображений без потерь // Тезисы докладов III конференции. Распознавание образов и анализ изображений: новые информационные технологии. Н. Новгород, 1997. Часть I. С. 157-161.
5. Гашников М. В., Глумов Н. И., Сергеев В. В. Информационная технология компрессии изображений в системах оперативного дистанционного зондирования // Известия Самарского научного центра РАН, 1999 № 1, С.99-107.
6. M. Gashnikov, N. Glumov, V. Sergeev Compression Method for Real-Time Systems of Remote Sensing // Proceedings of 15th International Conference on Pattern Recognition. Barcelona 2000. September 3-7. Vol.3. P. 232-235.

Adaptive interpolation algorithm for hierarchical image compression

M.V. Gashnikov^{1,2}, N.I. Glumov^{1,2}, V.V. Sergeev^{1,2}

¹*Image Processing Systems Institute of RAS,*

²*Samara State Aerospace University*

Abstract

The paper proposes a new adaptive interpolation algorithm for hierarchical image compression. The algorithm is based on the use of a parameterized decision rule, which is adjusted for each particular image during its processing. The developed interpolation algorithm is compared with the existing ones.

Keywords: adaptive interpolation algorithm, image compression, parameterized decision rule, image processing.

Citation: Gashnikov MV, Glumov NI, Sergeev VV. Adaptive interpolation algorithm for hierarchical image compression. Computer optics 2002; 23: 89-93.

References

- [1] Alexandrov RV, Gorsky ID. Image representation and processing: a recursive approach. Dordrecht: Springer Science+Business Media; 1993.
- [2] Kortman CM. Redundancy reduction – A practical method of data compression. Proc IEEE 1967; 55(3): 253-263.
- [3] Vasin YG, Bakareva VP. Recurrent algorithms for adaptive compression using well-adapted local restoration functions [In Russian]. In Book: CAD software: Interuniversity collection of papers. Vol I. Gorky: GSU Publisher; 1978: 13.
- [4] Efimov VM, Kolesnikov AN. Effectiveness estimation of the hierarchical and line-by-line lossless compression algorithms [In Russian]. Proc 3rd All-Russian Conference “Pattern Recognition and Image Analysis: New Information Technologies” 1997; I: 157-161.
- [5] Gashnikov MV, Glumov NI, Sergeev VV. Information technology for image compression in online remote sensing systems. Izvestia of Samara Scientific Center of the Russian Academy of Sciences 1999; 1: 99-107
- [6] Gashnikov M, Glumov N, Sergeev V. Compression method for real-time systems of remote sensing. Proc 15th Int Conf on Pattern Recognition 2000; 3: 232-235.