

ЧИСЛЕННЫЕ МЕТОДЫ И АНАЛИЗ ДАННЫХ

Обратимая целочисленная аппроксимация преобразований цветового пространства для сжатия без потерь цветных растровых данных большого объёма

А.Е. Хмельнов¹

¹ Институт динамики систем и теории управления имени В.М. Матросова СО РАН, 664033, Россия, г. Иркутск, ул. Лермонтова, д. 134

Аннотация

Обратимые целочисленные преобразования имеют большое значение для алгоритмов сжатия без потерь. Для выполнения обратимой декорреляции цветовых каналов предложен алгоритм вычисления параметров обратимого целочисленного преобразования, аппроксимирующего такие непрерывные отображения, как дискретное преобразование Карунена–Лоэва. Предложен способ оценивания ошибок аппроксимации, позволяющий выбрать оптимальную аппроксимацию исходного преобразования, минимизирующую эти ошибки. На примере формата файлов MRG, предназначенного для хранения больших объёмов целочисленных растровых данных, показано, что после применения декорреляции получается повысить степень сжатия многоканальных растровых изображений при использовании алгоритма сжатия без потерь.

Ключевые слова: декорреляция, обратимое целочисленное преобразование, сжатие без потери информации, специализированный алгоритм сжатия.

Цитирование: Хмельнов, А.Е. Обратимая целочисленная аппроксимация преобразований цветового пространства для сжатия без потерь цветных растровых данных большого объёма / А.Е. Хмельнов // Компьютерная оптика. – 2022. – Т. 46, № 3. – С. 492-505. – DOI: 10.18287/2412-6179-CO-1052.

Citation: Hmelnov AE. Reversible integer approximation of color space transforms for lossless compression of big color raster data. Computer Optics 2022; 46(3): 492-505. DOI: 10.18287/2412-6179-CO-1052.

Введение

Разработанный в ИДСТУ СО РАН формат MRG (Multi Resolution Grid) [1], предназначенный для представления без потери информации растровых данных большого объёма, был адаптирован для хранения многоканальных изображений. Для каждого блока изображения данные нескольких каналов последовательно размещаются в памяти, что позволяет использовать для их обработки алгоритмы, разработанные для одноканальных изображений. Кроме того, разделение каналов в памяти повышает эффективность использования алгоритмов сжатия, поскольку при этом рядом оказываются более близкие данные, что особенно важно для применяемых в формате MRG способов сжатия разностных целочисленных последовательностей [2]. Эффективность этих алгоритмов зависит от того, насколько близко к нулю будут находиться получаемые разности.

Дальнейшее улучшение результатов сжатия может быть получено за счёт учёта сходства разных цветовых каналов. Действительно, если визуально сравнить два канала одного изображения, то легко заметить сходство между ними: области высокой и низкой яркости на таких изображениях будут в основном

находиться на одних и тех же местах, т.е. эти изображения обычно сильно коррелированы. Таким образом, должно быть возможно улучшить степень сжатия по сравнению с независимой упаковкой каналов за счёт учёта этой корреляции.

В данной работе рассматривается подход, основанный на использовании декорреляции цветовых каналов, при которой вектора цвета переводятся в другое цветовое пространство, наиболее подходящее для обрабатываемых данных. В этом цветовом пространстве результирующие каналы уже не являются коррелированными, за счёт чего снижается разброс значений для большинства из них.

При этом все рассматриваемые алгоритмы сжатия работают без потерь, поэтому преобразование декорреляции цветовых каналов также должно быть обратимым. В работе рассматривается универсальный подход, позволяющий построить обратимое целочисленное преобразование на основе непрерывного линейного преобразования над действительными числами.

Заметим, что, с одной стороны, от обратимого преобразования $F(x)$ требуется только, чтобы выполнялось $F^{-1}(F(x)) \equiv x$ для обрабатываемых целочисленных векторов x , т.е. при этом не требуется такая же обратимость $F^{-1}(x)$, т.к. нам достаточно, чтобы

можно было восстанавливать x после преобразования. С другой стороны, нас не устраивают тривиальные решения, увеличивающие норму F , когда, например, за счёт умножения матрицы с рациональными коэффициентами на общий знаменатель этих коэффициентов мы заменим матрицу преобразования на целочисленную, поскольку таким способом мы увеличиваем диапазоны результирующих значений, что может пропорционально увеличить объём получаемых данных при использовании рассматриваемых алгоритмов сжатия.

Предложен способ оценки ошибки округления получаемого целочисленного преобразования и алгоритм выбора наилучшего варианта дискретизации преобразования, минимизирующего эту ошибку.

1. Примеры обратимых целочисленных преобразований

При создании алгоритмов сжатия без потерь и соответствующих форматов файлов, в которых выполняется преобразование целочисленных векторов, например, значений цветовых каналов, разработчики давно используют обратимые целочисленные преобразования, которые в какой-то степени аппроксимируют непрерывные. Хотя эти аппроксимации могут быть очень грубыми, их использование даёт ощутимый эффект. Рассмотрим примеры таких преобразований.

1.1. Преобразование цветов в JPEG 2000

Известный пример простого обратимого целочисленного преобразования используется, например, в формате JPEG 2000 [3]. Когда в этом формате применяется сжатие без потерь, требуется обратимость преобразования цветового пространства. В документации по формату описывается "Reversible multiple component transformation" (RCT) – прямое преобразование цветового пространства RGB:

$$Yr = \lfloor (R + 2G + B) / 4 \rfloor; Ur = R - G; Vr = B - G, \quad (1)$$

которому соответствует следующее обратное преобразование ("Inverse RCT"):

$$G = Yr - \lfloor (Ur + Vr) / 4 \rfloor; R = Ur + G; B = Vr + G.$$

Легко увидеть, что это преобразование действительно является обратимым, поскольку

$$\begin{aligned} \lfloor (Ur + Vr) / 4 \rfloor &= \lfloor (R - 2G + B) / 4 \rfloor = \\ &= \lfloor (R + 2G + B) / 4 \rfloor - G = Yr - G. \end{aligned}$$

Преобразование $YrUrVr$ является довольно грубой аппроксимацией точных формул для перехода от цветового пространства RGB в цветовое пространство $YCrCb$, задаваемых стандартом CCIR 601 и его более актуальными версиями [4]:

$$\begin{aligned} Y &= 0,299R + 0,587G + 0,114B, \\ Cr &= 0,5R - 0,4187G - 0,0813B + 128, \\ Cb &= -0,1687R - 0,3313G + 0,5B + 128. \end{aligned}$$

А именно: Ur аппроксимирует $2Cr$, а $Vr - 2Cb$, в которых коэффициенты округляются до ближайшего целого (при этом меньшие коэффициенты обнуляются). На самом деле точные формулы CCIR 601 также используются в JPEG 2000 при сжатии с потерями и обозначаются в стандарте аббревиатурой ICT (irreversible component transformation).

Из того, что в формате JPEG 2000 используется настолько грубое приближение желаемого преобразования цветового пространства, можно сделать вывод, что разработчикам этого формата не были известны более эффективные способы построения обратимых целочисленных аппроксимаций непрерывных преобразований.

1.2. Новое обратимое преобразование для спутниковых снимков

После анализа матрицы декорреляции для конкретного спутникового снимка для этой матрицы удалось вручную найти новое обратимое преобразование цветового пространства, которое больше подходит для таких данных:

$$\begin{aligned} Cr_1 &= \lfloor (R + G + B) / 3 \rfloor; Cr_2 = \lfloor (R + B - 2G) / 2 \rfloor; \\ Cr_3 &= B - R. \end{aligned} \quad (2)$$

Соответствующее обратное преобразование:

$$\begin{aligned} C &= 2Cr_2 + Cr_3 \bmod 2; G = Cr_1 - \lfloor C / 3 \rfloor; \\ R &= G + (C - Cr_3) / 2; B = G + (C + Cr_3) / 2. \end{aligned} \quad (3)$$

Покажем, что преобразование (3) является обратным для (2).

Теорема 1. Преобразование цветового пространства для спутниковых данных (2) обратимо, его обратное преобразование задаётся формулой (3).

Доказательство. Будем рассматривать операцию $\lfloor \cdot \rfloor$ как округление вниз, а значения вида $X \bmod D$ будем считать всегда положительными, т.е., например, $-5 \bmod 3 = -2 + 3 = 1$, а не -2 . Тогда

$$\lfloor X / D \rfloor = (X - X \bmod D) / D.$$

Поэтому

$$\begin{aligned} C &= 2Cr_2 + Cr_3 \bmod 2 = 2 \lfloor (R + B - 2G) / 2 \rfloor + \\ &+ (B - R) \bmod 2 = (R + B - (R + B) \bmod 2) - 2G + \\ &+ (B + R - 2R) \bmod 2 = R + B - 2G. \end{aligned}$$

И отсюда

$$\begin{aligned} Cr_1 - \lfloor C / 3 \rfloor &= \lfloor (R + G + B) / 3 \rfloor - \lfloor (R + B - 2G) / 3 \rfloor = \\ &= \lfloor (R + G + B) / 3 \rfloor - \lfloor (R + B + G) / 3 - G \rfloor = G, \\ G + (C - Cr_3) / 2 &= G + (R + B - 2G - (B - R)) / 2 = R, \\ G + (C + Cr_3) / 2 &= G + (R + B - 2G + (B - R)) / 2 = B, \quad \square \end{aligned}$$

На самом деле для получения таких обратимых преобразований нужна определённая удача. Этот процесс не может выполняться на регулярной основе, кроме того, в данном примере, так же как и для RCT из JPEG 2000, требуется существенно исказить аппроксимируемое преобразование, чтобы получилось его обратить.

2. Основные принципы предлагаемого подхода

Сначала рассмотрим основные идеи, лежащие в основе предлагаемого подхода: использование преобразования Карунена–Лоэва для декорреляции цветных каналов и общую схему дискретизации непрерывных преобразований.

2.1. Получение непрерывного преобразования для декорреляции цветных каналов

Главные компоненты изображения можно найти при помощи дискретного преобразования Карунена–Лоэва (ДКЛ): построения ковариационной матрицы и вычисления её собственных векторов [5]. После перехода в эту систему координат цветные компоненты оказываются некоррелированными, что устраняет избыточность информации и способствует сокращению числа бит, используемых для представления второстепенных цветных каналов за счёт небольшого увеличения средней битовой глубины главного канала. Однако элементы матрицы преобразования оказываются действительными числами, что требует использования округления для получения целочисленных результатов и не позволяет в общем случае рассчитывать на обратимость преобразования.

2.2. Получение обратимого целочисленного преобразования на базе непрерывного

Для обратимого поворота растрового изображения на угол α широко известна схема лифтинга [6]:

$$\begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1-\cos \alpha}{\sin \alpha} & 1 \end{bmatrix} \times \begin{bmatrix} 1 & -\sin \alpha \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{1-\cos \alpha}{\sin \alpha} & 1 \end{bmatrix}. \quad (4)$$

Здесь представлен один из её вариантов. Эта схема вычислений позволяет заменить поворот на ряд последовательных сдвигов вдоль осей координат, где величина сдвига вычисляется без участия изменяемой координаты, что позволяет выполнять вычисление и округление величины сдвига в ходе как прямого, так и обратного преобразований, имея в распоряжении все необходимые входные данные.

Этот подход можно обобщить для построения обратимых целочисленных аппроксимаций произвольных непрерывных преобразований при том условии,

что преобразование может быть представлено в необходимой для его применения форме. Рассмотрим основную идею подхода в несколько более общей форме, чем то, как это было изначально предложено в [7].

Непрерывное преобразование $\bar{y} = \bar{F}(\bar{x})$ должно быть представимо как последовательность шагов следующего вида:

$$\begin{aligned} y_0 &= k_0 x_n + f_0(x_1, \dots, x_{n-1}); \\ y_1 &= k_1 x_1 + f_1(y_0, x_2, \dots, x_{n-1}); \\ y_i &= k_i x_i + f_i(y_0, \dots, y_{i-1}, x_{i+1}, \dots, x_{n-1}); \\ y_n &= k_n y_0 + f_n(y_1, \dots, y_{n-1}), \end{aligned} \quad (5)$$

где $k_i \in \mathbb{Z}$, $k_i \neq 0$ и y_0 – вспомогательное промежуточное значение, не входящее в результирующий вектор \bar{y} . Скорее всего, коэффициенты $k_i \in \{-1; 1\}$, поскольку иначе будет расширен диапазон значений y_i , что неблагоприятно скажется на последующем сжатии соответствующей последовательности.

Тогда получится определить обратимую целочисленную аппроксимацию $y' = \bar{F}'(\bar{x})$ исходного непрерывного преобразования, где оба вектора \bar{x} и y' являются целочисленными:

$$\begin{aligned} y'_0 &= k_0 x_n + [f_0(x_1, \dots, x_{n-1})] \\ y'_1 &= k_1 x_1 + [f_1(y'_0, x_2, \dots, x_{n-1})] \\ y'_i &= k_i x_i + [f_i(y'_0, \dots, y'_{i-1}, x_{i+1}, \dots, x_{n-1})] \\ y'_n &= k_n y'_0 + [f_n(y'_1, \dots, y'_{n-1})], \end{aligned} \quad (6)$$

здесь скобки $[]$ обозначают некоторую операцию округления, например *round*, *ceil*, *floor*.

В результате получится обратить целочисленное преобразование, выражая из равенств значения y'_0 и x_i и меняя порядок выполнения вычислений на обратный:

$$\begin{aligned} y'_0 &= (y'_n - [f_n(y'_1, \dots, y'_{n-1})]) / k_n; \\ x_i &= (y'_i - [f_i(y'_0, \dots, y'_{i-1}, x_{i+1}, \dots, x_{n-1})]) / k_i; \\ x_1 &= (y'_1 - [f_1(y'_0, x_2, \dots, x_{n-1})]) / k_1; \\ x_0 &= (y'_0 - [f_0(x_1, \dots, x_{n-1})]) / k_0. \end{aligned} \quad (7)$$

Обращение преобразования таким образом становится возможным благодаря тому, что f_i зависит лишь от значений, доступных как при прямом, так и при обратном порядке вычислений, и не зависит от x_i и y'_i . Хотя рассматриваемый подход не делает никаких предположений о виде функции f_i и может быть применён для произвольных нелинейных преобразований, далее будем использовать только линейные, поскольку только это требуется в данной задаче.

Хотя при $abs(k_i) > 1$ обратное преобразование перестает быть целочисленным, поскольку для значений

$$y'_i - \left[f_i(y'_0, \dots, y'_{i-1}, x_{i+1}, \dots, x_{n-1}) \right],$$

которые не делятся на k_i , оно даёт не целые x_i , это не делает $F'(\bar{x})$ необратимым, т.к. для $y'_i = F'(\bar{x})$ делимость этих выражений на k_i обеспечена.

Заметим, что величина ошибки целочисленной аппроксимации $\bar{d}y = y' - \bar{y}$ складывается из ошибок округления f_i и ошибок, вызванных заменой на y'_i в аргументах f_i . Далее мы проанализируем этот вопрос более подробно.

3. Представление линейного преобразования как последовательности сдвигов по одной из координат

Теперь покажем конструктивно, каким образом линейное преобразование может быть аппроксимировано по общей схеме (6).

3.1. Элементарные обратимые матрицы

Для получения обратимого целочисленного преобразования, близкого к матрице перехода к главным компонентам, был адаптирован подход, описанный в [7]. В этой работе вводится понятие *элементарных обратимых матриц* (ERM – elementary reversible matrices). Задаваемые этими матрицами преобразования могут быть вычислены обратимо. Рассматриваются *треугольные элементарные обратимые матрицы* (TERM – triangular elementary reversible matrices) и *однострочные элементарные обратимые матрицы* (SERM – single-row elementary reversible matrices).

Матрицы вида SERM отличаются от тождественной матрицы I только значениями недиагональных ячеек одной строки и, возможно, знаком соответствующего диагонального элемента. Таким образом, такие матрицы задают линейный сдвиг вдоль одной из координат, величина которого не зависит от самой этой координаты. Именно это требуется для того, чтобы можно было применить ранее рассмотренный подход для получения обратимого целочисленного преобразования.

В работе [7] доказано, что матрицу A размерности $n \times n$ с $\det A = \pm 1$ можно разложить в произведение матрицы перестановки строк P и $n+1$ матрицы вида SERM. При доказательстве используется промежуточное разложение на треугольные матрицы вида TERM, и это доказательство не получается использовать как непосредственное руководство к действию для написания алгоритма SERM-факторизации. Поэтому далее рассмотрим альтернативное конструктивное доказательство аналогичной теоремы, которое описывает конкретный алгоритм SERM-факторизации.

3.2. Алгоритм факторизации на матрицы сдвига по одной координате

Покажем конструктивно, что для матрицы A размерности $n \times n$ с определителем по модулю, равным 1, можно построить разложение на $n+1$ матрицу сдвига

вдоль осей системы координат $B^{(i)}$ с единичными по модулю элементами на диагонали и единственной ненулевой строкой:

$$M = P_L A P_R = B^{(n)} \cdot \dots \cdot B^{(1)} \cdot B^{(0)}, \quad (8)$$

где

$$B^{(0)} = \begin{bmatrix} 1 & & & \\ & \dots & & \\ & & 1 & \\ b_{01} & \dots & b_{0,n-1} & k_0 \end{bmatrix} B^{(1)} = \begin{bmatrix} k_1 & b_{12} & \dots & b_{1n} \\ & 1 & & \\ & & \dots & \\ & & & 1 \end{bmatrix} \\ \dots \\ B^{(n)} = \begin{bmatrix} 1 & & & \\ & \dots & & \\ & & 1 & \\ b_{n1} & \dots & b_{n,n-1} & k_n \end{bmatrix}. \quad (9)$$

Здесь матрица P_L задаёт перестановку строк, а P_R – столбцов, в итоге получаем матрицу M – результат перестановки строк и столбцов матрицы A .

В матрицах $B^{(i)}$ используются коэффициенты $k_i \in \{-1, 1\}$, $i \in \{1..n\}$, которые могут изменять направление i -й оси. Для заданных значений k_i выбираем k_0 так, чтобы выполнялось:

$$\prod_{i=0}^n k_i = \det M = \det P_L A P_R = \pm 1. \quad (10)$$

Необходимость использования отрицательных значений k_i мы покажем позже при рассмотрении примера с лифтингом матрицы поворота в параграфе 5.1. На самом деле, если использовать знаки k_i , $i \in \{1..n\}$, то можно всегда задавать $k_0 = 1$, поскольку изменение знака k_0 можно выполнить путём изменения знака всех b_{in} для $i \in \{1..n\}$ и k_n . Всё же далее мы рассмотрим общий случай с выбором знака и u k_0 .

В матрицах сдвига $B^{(i)}$ для $i \in \{1..n\}$ от строк единичной матрицы отличается только i -я (и n -я у $B^{(0)}$). Таким образом, при использовании такого разложения за один шаг выполняется изменение одной из координат. Всего происходит $n+1$ пересчёт координат, что потребует $n+1$ округления при выполнении целочисленной аппроксимации. Каждая комбинация перестановки строк и столбцов и выбора знаков k_i определяет свой способ разложения. Эти способы разложения отличаются порядком вычисления результирующих координат (зависит от перестановки строк), выбором исходных координат, соответствующих результирующим (зависит от перестановки столбцов), а также тем, происходит ли при этом изменение знака координаты (знак k_i).

Теорема 2. Каждому сочетанию результата перестановки строк и столбцов $M = P_L A P_R$ и выбора знаков $k_i \in \{-1, 1\}$, $i \in \{1..n\}$ может соответствовать способ факторизации на $n+1$ матрицу сдвига по одной координате.

Доказательство. Для начала дадим общие соображения, подтверждающие возможность решения рассматриваемой задачи. Рассмотрим преобразование $y = Mx$. Обратим внимание, что пространство матриц M имеет размерность $n^2 - 1$ (поскольку матрица задаётся в n^2 -мерном пространстве значений элементов ограничением на величину детерминанта). Пространство коэффициентов b_{ij} также имеет размерность $(n+1)(n-1) = n^2 - 1$. Таким образом, размерности двух рассматриваемых пространств совпадают при любом n . Поскольку $|\det B^{(0)}| = 1$, равенство модуля определителя произведения матриц единице с необходимым знаком обеспечивается по построению с использованием условия (10) посредством выбора знака элемента со значением k_0 матрицы $B^{(0)}$.

Покажем, как может быть найдено сдвиговое разложение матрицы M . Обозначим

$$v = \sum_{j=1}^{n-1} b_{0j} x_j + k_0 x_n, \tag{11}$$

n -я координата вектора после применения $B^{(0)}$.

Дальше начинают вычисляться координаты результирующего вектора y .

$$y_1 = k_1 x_1 + \sum_{j=2}^{n-1} b_{1j} x_j + b_{1n} v = (k_1 + b_{1n} b_{01}) x_1 + \sum_{j=2}^{n-1} (b_{1j} + b_{1n} b_{0j}) x_j + b_{1n} k_0 x_n, \tag{12}$$

при этом, с другой стороны,

$$y_i = \sum_{j=1}^n m_{ij} x_j.$$

Сравнивая коэффициенты при x_j в двух выражениях для y_1 , получаем:

$$k_1 + b_{1n} b_{01} = m_{11}, \quad b_{1j} + b_{1n} b_{0j} = m_{1j} \tag{13}$$

для $j \in \{2..n-1\}$, $b_{1n} k_0 = m_{1n}$.

Из последнего равенства находим b_{1n} , после чего из первого находится b_{01} , а остальные равенства становятся линейными по ещё не известным b_{0j} и b_{1j} (здесь и далее ещё не известные коэффициенты выделены жирным шрифтом). Эти остальные равенства будут использованы для вычисления b_{1j} после нахождения значений b_{0j} на следующих шагах (см. далее (20)).

Далее рассмотрим i -й шаг:

$$y_i = \sum_{j=1}^{i-1} b_{ij} y_j + k_i x_i + \sum_{j=i+1}^{n-1} b_{ij} x_j + b_{in} v = \sum_{j=0}^{i-1} b_{ij} \sum_{l=1}^n m_{jl} x_l + k_i x_i + \sum_{j=i+1}^{n-1} b_{ij} x_j + b_{in} \left(\sum_{j=1}^{n-1} b_{0j} x_j + k_0 x_n \right). \tag{14}$$

Здесь в качестве значений y_j сразу используем выражения с коэффициентами из матрицы M . Группируя слагаемые с x_j , получим:

$$y_i = \sum_{j=1}^{i-1} \left(\sum_{l=1}^{i-1} b_{il} m_{lj} + b_{in} b_{0j} \right) x_j + \left(\sum_{l=1}^{i-1} b_{il} m_{li} + b_{in} b_{0i} + k_i \right) x_i + \sum_{j=i+1}^{n-1} \left(\sum_{l=1}^{i-1} b_{il} m_{lj} + b_{in} b_{0j} + b_{ij} \right) x_j + \left(\sum_{l=1}^{i-1} b_{il} m_{ln} + b_{in} k_0 \right) x_n. \tag{15}$$

Если предположить, что по результатам анализа выражений для y_j для $j < i$ нам уже известны значения для b_{0j} для $j < i$, то по первому и последнему слагаемым в выражении (15) для y_i получим систему из i линейных уравнений:

$$\begin{cases} \sum_{l=1}^{i-1} b_{il} m_{lj} + b_{in} b_{0j} = m_{ij} & \text{для } j \in \{1..i-1\} \\ \sum_{l=1}^{i-1} b_{il} m_{ln} + b_{in} k_0 = m_{in}, \end{cases} \tag{16}$$

для i неизвестных: b_{il} для $l \in \{1..i-1\}$ и b_{in} .

После решения этой системы второе слагаемое в выражении (15) даёт уравнение:

$$\sum_{l=1}^{i-1} b_{il} m_{li} + b_{in} b_{0i} + k_i = m_{ii}. \tag{17}$$

Отсюда

$$b_{0i} = \left(m_{ii} - \sum_{l=1}^{i-1} b_{il} m_{li} - k_i \right) / b_{in}.$$

Таким образом, предположение индукции о возможности найти b_{0j} для $j < i$ по результатам анализа выражений для y_j для $j < i$ распространяется и на i . Заметим, что база индукции также выполнена, т.к. по результатам анализа y_1 было вычислено значение b_{01} .

Из третьего слагаемого в выражении (15) получим:

$$\sum_{l=1}^{i-1} b_{il} m_{lj} + b_{in} b_{0j} + b_{ij} = m_{ij} \text{ для } j \in \{i+1..n-1\}. \tag{18}$$

После подстановки в эти равенства вычисленных значений b_{il} для $l \in \{1..i-1\}$ и b_{in} они позволяют выразить ещё не известные b_{ij} (этими выражениями можно будет воспользоваться после вычисления b_{0j}):

$$b_{ij} = m_{ij} - \sum_{l=1}^{i-1} b_{il} m_{lj} - b_{in} b_{0j} \text{ для } j \in \{i+1..n-1\}. \tag{19}$$

Также теперь после вычисления b_{0i} можно определить ещё не вычисленные значения b_{ij} с использованием второго равенства из (13) (даёт уравнение для $j=1$) и части равенств (19) для ранее выполненных шагов (дают уравнения для $j \in \{2..i-1\}$):

$$b_{ji} = m_{ji} - \sum_{l=1}^{j-1} b_{jl} m_{li} - b_{jn} b_{0i} \text{ для } j \in \{1..i-1\}. \tag{20}$$

Наконец, на n -м шаге:

$$y_n = \sum_{j=1}^{n-1} b_{nj} y_j + k_n v = \sum_{j=0}^{n-1} b_{nj} \sum_{l=1}^n m_{jl} x_l + k_n \left(\sum_{j=1}^{n-1} b_{0j} x_j + k_0 x_n \right). \tag{21}$$

Аналогично общему случаю, группируя слагаемые с x_j , получим:

$$y_n = \sum_{j=1}^{n-1} \left(\sum_{l=1}^{n-1} b_{nl} m_{lj} + k_n b_{0j} \right) x_j + \left(\sum_{l=1}^{n-1} b_{nl} m_{ln} + k_n k_0 \right) x_n. \quad (22)$$

Отсюда следует система из n линейных уравнений:

$$\begin{cases} \sum_{l=1}^{n-1} b_{nl} m_{lj} + k_n b_{0j} = m_{nj} & \text{для } j \in \{1..n-1\} \\ \sum_{l=1}^{n-1} b_{nl} m_{ln} + k_n k_0 = m_{nn}, \end{cases} \quad (23)$$

для поиска $n-1$ неизвестного b_{nl} для $l \in \{1..i-1\}$. Т.е. одно из этих уравнений будет лишним, а система – избыточной. Это происходит из-за того, что условие (10) на знак $\det M$ выполняется автоматически в силу строения матриц, участвующих в разложении.

Поскольку при избыточности системы сразу не ясно, какое из уравнений лишнее, такую систему необходимо решать именно как переопределённую с использованием соответствующих алгоритмов, принимающих на вход прямоугольную матрицу. Например, можно применить метод Гаусса с выбором главного элемента для прямоугольных матриц.

После нахождения значений b_{nl} из (23) также требуется найти последние ещё не вычисленные коэффициенты по формуле (20).

Таким образом, при условии, что все рассматриваемые системы линейных уравнений будут разрешимыми и коэффициенты, на которые происходит деление в ряде формул, не будут равны 0, будут вычислены все коэффициенты матриц линейных сдвигов по одной координате, используемые в разложении. При этом все эти коэффициенты определяются однозначно.

Табл. 1. Порядок вычисления коэффициентов для матрицы 5×5

| B | 1 | 2 | 3 | 4 | 5 |
|----------|----------|----------|----------|----------|----------|
| 0 | 1 | 2 | 3 | 4 | k_0 |
| 1 | k_1 | 2 | 3 | 4 | 1 |
| 2 | 2 | k_2 | 3 | 4 | 2 |
| 3 | 3 | 3 | k_3 | 4 | 3 |
| 4 | 4 | 4 | 4 | k_4 | 4 |
| 5 | 5 | 5 | 5 | 5 | k_5 |

Но здесь возможен и отрицательный результат, когда для данного порядка вычислений решение не получится найти из-за деления на 0 или неразрешимости системы линейных уравнений. Поэтому в формулировке теоремы содержится фраза «может соответствовать». □

В табл. 1 показан порядок вычисления коэффициентов матрицы B на примере расчётов для матрицы M с размерностью 5×5 . Цифра в ячейке таблицы – номер шага алгоритма, на котором эта ячейка вычисляется; k_i в ячейке означает, что её значением является k_i , которое известно ещё до начала работы алгоритма.

4. Поиск наилучшего способа вычислений

Напомним, что матрица $M = P_L A P_R$, т.е. она является результатом перестановки строк и столбцов матрицы A . Кроме того, на получаемые выражения влияет выбор знаков k_i . Таким образом, остаётся вопрос о поиске наилучшего способа вычисления коэффициентов b_{ij} . Более качественным будем считать такое разложение матрицы, которое минимизирует среднеквадратичное отклонение значений целочисленного преобразования от исходного непрерывного. Таким образом, для небольших размерностей можно выполнить полный перебор всех $2^n (n!)^2$ вариантов. В частности, если в ходе вычислений возникает деление на 0, то это означает, что просто требуется рассмотреть другой вариант порядка переменных и равенств.

4.1. Среднеквадратичная ошибка округления

Рассмотрим способ оценивания среднеквадратичного отклонения результатов целочисленного преобразования от результатов непрерывного. Обозначим через v' целочисленную версию v . Здесь и далее $\{x\}$ обозначает округление x , а $\{x\} = [x] - x$ – разность между результатом округления и исходным значением. Если считать, что округление выполняется к ближайшему целому, то для больших диапазонов изменения x (т.е. при $\max(|x|) \gg 1$) можно считать, что $\{x\}$ является случайной величиной, равномерно распределённой на интервале $[-0,5; 0,5]$. Тогда

$$\overline{\{x\}^2} = \int_{-0,5}^{0,5} x^2 dx = \frac{1}{12}. \quad (24)$$

Заметим, что, например, при обработке растровых изображений значения цветовых компонент обычно представлены целочисленными значениями из интервала 0..255, поэтому здесь явно можно считать такой диапазон изменения значений большим.

При работе с матрицами, возникающими в задачах обработки измерений об объектах реального мира, например матриц ДКЛ, оценка (24) всегда будет выполнена, т.к. получение рациональных коэффициентов с небольшими знаменателями здесь крайне маловероятно. Однако для преобразований, задаваемых аналитически, такие коэффициенты вполне возможны, поэтому далее рассмотрим и эти случаи.

4.2. Среднеквадратичная ошибка округления для рациональных чисел с небольшим знаменателем

Когда все коэффициенты в линейном выражении над целочисленными параметрами являются рациональными числами с небольшим знаменателем, оценку (24) может потребоваться уточнить. Действительно, если для знаменателей d_i у коэффициентов из некоторого равенства $\text{НОК}(d_i) = D$, то разность между результатом округления и исходным значением может быть равна одной из D дробей со знаменателем D :

$$\{x\} \in \left\{ -\frac{1}{2}, -\frac{n-1}{2n}, \dots, \frac{n-1}{2n} \right\} \text{ для чётных } D = 2n \text{ и}$$

$$\{x\} \in \left\{ -\frac{n}{2n+1}, \dots, \frac{n}{2n+1} \right\} \text{ для нечётных } D = 2n + 1 .$$

Поэтому, считая все эти дроби равновероятными, получаем:

$$\overline{\{x\}^2} = \left(\left(\frac{1}{2} \right)^2 + 2 \sum_{i=1}^{n-1} \left(\frac{i}{2n} \right)^2 \right) / (2n) =$$

$$= \left(n^2 + 2 \sum_{i=1}^{n-1} i^2 \right) / (2n)^3 \text{ для } D = 2n$$

и

$$\overline{\{x\}^2} = \left(2 \sum_{i=1}^n \left(\frac{i}{2n+1} \right)^2 \right) / (2n+1) =$$

$$= 2 \sum_{i=1}^n i^2 / (2n+1)^3 \text{ для } D = 2n + 1 .$$

Используя формулу для суммы квадратов первых n целых чисел

$$S(n) = \sum_{i=1}^n i^2 = \frac{1}{6} (2n+1)(n+1)n$$

и упрощая выражение

$$n^2 + 2S(n-1) = \frac{1}{3} n(2n^2 + 1),$$

получаем:

$$\overline{\{x\}^2} = (n^2 + 2S(n-1)) / (2n)^3 = \frac{1}{3} n(2n^2 + 1) / (2n)^3 =$$

$$= \frac{1}{6} (2n^2 + 1) / (2n)^2 \text{ для } D = 2n$$

и

$$\overline{\{x\}^2} = 2S(n) / (2n+1)^3 =$$

$$= \frac{1}{3} (2n+1)(n+1)n / (2n+1)^3 \text{ для } D = 2n + 1 .$$

Таким образом

$$\overline{\{x\}^2} = \begin{cases} \frac{1}{12} \left(1 + \frac{1}{2n^2} \right) & \text{для } D = 2n , \\ \frac{1}{3} (n+1)n / (2n+1)^2 & \text{для } D = 2n + 1 . \end{cases} \quad (25)$$

Легко заметить, что с увеличением n оба выражения с разных сторон стремятся к $1/12 = 0,08(3)$, однако при небольших n они существенно отклоняются от этой величины. Несколько первых значений показаны в табл. 2 (в третьем столбце приводится отклонение в % от $1/12$).

Таким образом, для $D=2$ отклонение будет больше в 1,5 раза, чем при округлении непрерывных ве-

личин, для $D=3$ – на 11 % меньше, а далее, например, только значения $D=4$ и $D=6$ дают отклонение более 5%. Поэтому, чтобы сократить ошибку оценки до этого уровня, достаточно проверить коэффициенты на то, что все они являются целыми или рациональными числами со знаменателями 2, 3, 4, 6. Поэтому необходимым условием существования таких знаменателей у коэффициента C является то, чтобы $12C$ было целым числом. Для определения знаменателя достаточно проанализировать остаток от деления этого целого числа на 12.

Табл. 2. Ошибки округления для рациональных знаменателей

| $D = 2n$ | $\overline{\{x\}^2}$ | в % от 1/12 | $D = 2n + 1$ | $\overline{\{x\}^2}$ | в % от 1/12 |
|----------|----------------------|-------------|--------------|----------------------|-------------|
| 2 | 0,125 | 50 | 3 | 0,07407 | -11,111 |
| 4 | 0,09375 | 12,5 | 5 | 0,08 | -4 |
| 6 | 0,08796 | 5,5556 | 7 | 0,08163 | -2,0408 |
| 8 | 0,08594 | 3,125 | 9 | 0,08230 | -1,2346 |
| 10 | 0,085 | 2 | 11 | 0,08265 | -0,8265 |
| 12 | 0,08449 | 1,3889 | 13 | 0,08284 | -0,5917 |
| 14 | 0,08418 | 1,0204 | 15 | 0,08296 | -0,4444 |
| 16 | 0,08398 | 0,7813 | 17 | 0,08305 | -0,3460 |
| 18 | 0,08385 | 0,6173 | 19 | 0,08310 | -0,2770 |
| 20 | 0,08375 | 0,5 | 21 | 0,08314 | -0,2268 |

4.3. Оценивание ошибок целочисленной аппроксимации линейного выражения

При использовании в одном линейном выражении нескольких ошибок округления можно считать эти ошибки независимыми случайными величинами в том случае, когда округляемые выражения различаются.

Если U и V – независимые случайные величины, а $W = aU + bV$, то

$$\overline{W^2} = \overline{(aU + bV)^2} = a^2 \overline{U^2} + b^2 \overline{V^2} .$$

Если же $U = V$, то ошибка округления возрастет:

$$\overline{W^2} = \overline{(aV + bV)^2} = (a + b)^2 \overline{V^2} .$$

Отдельно необходимо учесть, что в том случае, когда все коэффициенты целочисленные, ошибка округления равна нулю.

Обозначим ошибку округления для вспомогательной величины v в (11):

$$dv = v' - v = \left\{ \sum_{j=1}^{n-1} b_{0j} x_j \right\} = \delta_0 . \quad (26)$$

Далее рассмотрим выражение для целочисленной аппроксимации i -й результирующей координаты из (14):

$$y'_i = k_i x_i + \left[\sum_{j=1}^{i-1} b_{ij} y'_j + \sum_{j=i+1}^{n-1} b_{ij} x_j + b_{in} v' \right] . \quad (27)$$

Перепишем выражения для y_i через целочисленные аппроксимации переменных y'_j :

$$\begin{aligned}
 y_i &= k_i x_i + \sum_{j=1}^{i-1} b_{ij} y_j + \sum_{j=i+1}^{n-1} b_{ij} x_j + b_m v = \\
 &= k_i x_i + \sum_{j=1}^{i-1} b_{ij} y'_j + \sum_{j=i+1}^{n-1} b_{ij} x_j + b_m v' - \sum_{j=1}^{i-1} b_{ij} dy_j - b_m dv.
 \end{aligned} \quad (28)$$

Отсюда получаем выражение для общей ошибки аппроксимации, учитывающей и ошибки аппроксимации используемых в выражении y_i :

$$\begin{aligned}
 dy_i &= y'_i - y_i = \left\{ \sum_{j=1}^{i-1} b_{ij} y'_j + \sum_{j=i+1}^{n-1} b_{ij} x_j + b_m v' \right\} + \\
 &+ \sum_{j=1}^{i-1} b_{ij} dy_j + b_m dv = \delta_i + \sum_{j=1}^{i-1} b_{ij} dy_j + b_m dv,
 \end{aligned} \quad (29)$$

здесь для ошибки округления, выполняемого при вычислении y'_i , мы вводим обозначение

$$\delta_i = \left\{ \sum_{j=1}^{i-1} b_{ij} y'_j + \sum_{j=i+1}^{n-1} b_{ij} x_j + b_m v' \right\}. \quad (30)$$

Из (29) видно, что dy_i зависит от dy_j для j от 1 до $i-1$, а также от δ_i и δ_0 . Подставляя аналогичные выражения для dy_j , получим, что

$$dy_i = \delta_i + \sum_{j=0}^{i-1} D_{ij} \delta_j, \quad (31)$$

здесь D_{ij} – коэффициенты, которые ещё требуется найти.

Подставляя выражения для dy_j из (31) в (29), получим:

$$\begin{aligned}
 dy_i &= \delta_i + \sum_{j=1}^{i-1} b_{ij} \left(\delta_j + \sum_{l=0}^{j-1} D_{jl} \delta_l \right) + b_m \delta_0 = \\
 &= \delta_i + \left(\sum_{j=1}^{i-1} b_{ij} D_{j0} + b_m \right) \delta_0 + \sum_{j=1}^{i-1} b_{ij} \delta_j + \sum_{l=1}^{i-2} \delta_l \sum_{j=l+1}^{i-1} b_{ij} D_{jl},
 \end{aligned} \quad (32)$$

отсюда

$$\begin{aligned}
 dy_i &= \delta_i + \left(b_m + \sum_{j=1}^{i-1} b_{ij} D_{j0} \right) \delta_0 + \\
 &+ \sum_{j=1}^{i-2} \left(b_{ij} + \sum_{l=j+1}^{i-1} b_{il} D_{lj} \right) \delta_j + b_{i,i-1} \delta_{i-1}.
 \end{aligned} \quad (33)$$

Сопоставляя коэффициенты перед δ_j в (33) и (31), получим выражения для D_{ij} , задающие процесс их вычисления по коэффициентам, уже вычисленным для j от 0 до $i-1$. Отдельно требуется рассмотреть начальное значение $i=1$:

$$dy_1 = \delta_1 + b_{1n} \delta_0. \quad (34)$$

При оценке среднего квадрата этих выражений будем считать, что $\overline{\delta_i^2} = 1/12$, если среди b_{ij} есть действительные (т.е. рациональные с большим знаменателем) коэффициенты, если же все коэффициенты рациональные с небольшим знаменателем, то использу-

ем оценки для рациональных чисел, иначе (все b_{ij} – целочисленные) считаем, что $\overline{\delta_i^2} = 0$.

4.4. Учёт зависимостей между координатами результирующего вектора

В том случае, когда в правой части выражений для вычисления y'_i и y'_j используются одинаковые округляемые выражения, ошибки округления для этих координат уже нельзя считать независимыми случайными величинами, что должно повлиять на вычисление оценки для тех y'_k , в правой части выражений для которых используются y'_i и y'_j . При этом $\overline{\delta_i^2}$ и $\overline{\delta_j^2}$ будут совпадать. Поэтому при наличии таких совпадений после вычисления D заменяем D_{ki} на $D_{ki} + D_{kj}$ и D_{kj} на 0. Более точно: среди всех координат с совпадающими округляемыми выражениями выбираем одну (скажем, с наименьшим индексом i), записываем в D_{ki} сумму элементов D_{kj} по всем j , для вычисления которых используется то же самое округляемое выражение, что и для i , а остальные D_{kj} заменяем на 0.

Также возможна ситуация, что округляемые выражения для y'_i и y'_j отличаются только знаком. Если считать, что операция $[\]$ соответствует функции $round()$ с округлением к ближайшему чётному числу, тогда $[-E] = -[E]$. Поэтому дополнительно необходимо учитывать и такие совпадения, только теперь для тех y'_j , для которых знак округляемого выражения отличается, надо вычитать D_{kj} из D_{ki} .

4.5. Учёт наличия целочисленных коэффициентов при поиске совпадений

При наличии в округляемых выражениях целочисленных коэффициентов может возникнуть желание вынести целочисленные слагаемые I из-под операции округления, рассчитывая на то, что $[I+E] = I+[E]$, после чего могут обнаружиться дополнительные совпадения округляемых выражений. Однако если используется округление к ближайшему чётному числу, то на результат не влияет вынос из-под округления только чётных I . Вынос же нечётного приведёт к изменению знака $\{E\}$ для $E = n + 1/2$ (n – целое), т.е. направление округления для чисел вида $XXX,5$ будет изменено на противоположное. Такой эффект будет существенным только в том случае, когда все нецелочисленные коэффициенты в E являются рациональными и НОД знаменателей этих коэффициентов – небольшое чётное число, поскольку для действительных коэффициентов вероятностью получения дробной части 0,5 можно пренебречь, а для нечётных знаменателей дробная часть 0,5 невозможна. Таким образом, для $D = 2n$ ошибка округления до ближайшего чётного принимает значения

$$\{x\} \in \left\{ -\frac{1}{2}, -\frac{n-1}{2n}, \dots, \frac{n-1}{2n}, \frac{1}{2} \right\},$$

где вероятность всех промежуточных значений равна $1/2n$, а значения $\pm 1/2$ возникают с вероятностью $1/4n$.

В итоге слагаемые с чётными коэффициентами сразу выносим из-под операции округления. Если после этого у двух округляемых выражений различаются только целочисленные слагаемые, а все рациональные совпадают, то округляемые величины из двух таких выражений будут всегда совпадать, когда они принимают промежуточные значения, а для крайних значений $-1/2$ и $1/2$ они могут либо совпадать, либо различаться с вероятностью $1/2$.

Т.е. случайные события округления вниз и вверх для двух рассматриваемых подвыражений являются независимыми при условии того, что дробная часть округляемых чисел равна $1/2$. Поэтому при вычислении корреляции между δ_i и δ_j слагаемые для $-1/2$ и $1/2$ будут равны нулю. Таким образом:

$$\overline{\delta_i \delta_j} = 2S(n-1)/(2n)^3 = (2n-1)(n-1)/(24n^2). \quad (35)$$

При вычислении оценок dy_i^2 придётся вычислять выражения вида

$$\begin{aligned} \overline{W^2} &= \overline{(a_i \delta_i + a_j \delta_j)^2} = a_i^2 \overline{\delta_i^2} + \\ &+ 2a_i a_j \overline{\delta_i \delta_j} + a_j^2 \overline{\delta_j^2} = (a_i^2 + a_j^2) \frac{1}{12} \left(1 + \frac{1}{2n^2} \right) + \\ &+ 2a_i a_j \frac{(2n-1)(n-1)}{24n^2}, \end{aligned} \quad (36)$$

$$\overline{W^2} = (a_i^2 + a_j^2) \frac{1}{8n} + (a_i + a_j)^2 \frac{(2n-1)(n-1)}{24n^2}. \quad (37)$$

Аналогично можно получить такое же выражение для большего числа слагаемых:

$$\overline{W^2} = \left(\sum_i a_i^2 \right) \frac{1}{8n} + \left(\sum_i a_i \right)^2 \frac{(2n-1)(n-1)}{24n^2}. \quad (38)$$

Таким образом, первое слагаемое описывает δ_j как независимые случайные величины, а второе – как совпадающие. Видно, что с ростом n доля независимой составляющей быстро падает, но при $n=1$ (знаменатель $D=2$), наоборот, δ_i являются полностью независимыми при наличии любых различий в целочисленных слагаемых у округляемых выражений. В итоге получаем следующие уточнения к алгоритму проверки на совпадение округляемых выражений:

- слагаемые с чётными целочисленными коэффициентами всегда можно пропускать при сравнении выражений;
- если в округляемом выражении присутствуют действительные коэффициенты или все знаменатели рациональных являются нечётными, то можно пропускать и нечётные целочисленные коэффициенты;
- если все знаменатели рациональных коэффициентов равны 2, то любые различия в слагаемых с не-

чётными целочисленными коэффициентами делают ошибки округления независимыми;

- иначе (чётное НОК знаменателей больше 2) используем схему вычислений, комбинирующую независимые и зависимые составляющие по формуле (38).

4.6. Алгоритм поиска наилучшего порядка вычислений

Как уже упоминалось ранее, существует $(n!)^2$ перестановок строк и столбцов матрицы $n \times n$ и 2^n вариантов выбора знаков k_j . С ростом размерности матрицы произведение этих величин быстро даёт очень большие числа. Поэтому при необходимости обрабатывать более 4–5 каналов требуется использовать методы ускорения полного перебора.

Ускорить вычисления позволяет наблюдение, что рассматриваемый в Теореме 2 алгоритм работает построчно: первые i его шагов обрабатывают первые i строк матрицы и никак не зависят от значений элементов следующих её строк. Поэтому в ходе перебора можно существенно повысить скорость вычислений разложением матриц за счёт повторного использования вычислений для всех начальных строк матрицы, которые не подвергались перестановке.

Для генерации перестановок строк и столбцов используем идею алгоритма перестановок Хипа, который минимизирует число обменов переставляемых объектов (строк или столбцов). При этом изменяем этот алгоритм так, чтобы он чаще обменивал объекты с большими индексами.

Цикл верхнего уровня выполняет перебор перестановок столбцов. Вложенный цикл выполняет перебор перестановок строк. При этом он повторно использует результаты вычислений для неизменившихся строк, так как метод вычисления разложения получает на вход номер строки, начиная с которой необходимо продолжить расчёт. Кроме того, этот метод возвращает номер строки, на которой произошла ошибка вычислений (деление на 0 или ошибка в методе Гаусса): новые вычисления в ходе перебора продолжатся только после изменения ошибочной строки.

Ещё необходимо учесть, что первый шаг алгоритма зависит от знака определителя, который меняется на противоположный при каждой перестановке. Поэтому нельзя непосредственно повторно использовать результаты от предыдущего состояния вычислений. Вместо этого алгоритм работает с двумя состояниями вычислений: для положительного и отрицательного значений определителя.

5. Примеры использования алгоритмов

Приведём несколько примеров, демонстрирующих применение предложенных алгоритмов. Сначала аналитически исследуем задачу обратимой целочисленной аппроксимации операции поворота в 2D с использованием предложенных в статье алгоритмов факторизации и оценки ошибок аппроксимации. Далее рассмотрим несколько примеров применения разработан-

ного с использованием этих алгоритмов программного обеспечения для обработки матриц с действительными и рациональными коэффициентами, а также при сжатии многоканальных растровых изображений.

5.1. Поворот в 2D

Вернёмся к рассмотрению схемы лифтинга для матрицы поворота (4). Применим к этой матрице алгоритм факторизации из Теоремы 2.

Сначала вычислим результат работы алгоритма для произвольной двумерной матрицы M . При этом выполняется $\det M = k_0 k_1 k_2$, и отсюда $k_0 = \det M / (k_1 k_2)$. Из (13) получаем:

$$b_{12} = m_{12} / k_0,$$

$$b_{01} = (m_{11} - k_1) / b_{12} = (m_{11} - k_1) k_0 / m_{12}.$$

Далее сразу переходим к последнему шагу алгоритма (23), на котором используется переопределённая система. Из последнего уравнения:

$$b_{21} = (m_{22} - k_0 k_2) / m_{12}.$$

Из первого можно получить тот же результат, т.к. $\det M = m_{11} m_{22} - m_{21} m_{12}$:

$$b_{21} = (m_{21} - k_2 b_{01}) / m_{11} =$$

$$= (m_{21} - k_2 (m_{11} - k_1) k_0 / m_{12}) / m_{11} =$$

$$= (m_{21} m_{12} - k_0 k_2 m_{11} + k_0 k_1 k_2) / (m_{11} m_{12}) =$$

$$= (m_{21} m_{12} - k_0 k_2 m_{11} + (m_{11} m_{22} - m_{21} m_{12})) / (m_{11} m_{12}) =$$

$$= (m_{11} m_{22} - k_0 k_2 m_{11}) / (m_{11} m_{12}) = (m_{22} - k_0 k_2) / m_{12}.$$

В результате получаем следующее разложение матрицы M :

$$M = \begin{bmatrix} 1 & 0 \\ \frac{(m_{22} - k_2 k_0)}{m_{12}} & k_2 \end{bmatrix} \begin{bmatrix} k_1 & \frac{m_{12}}{k_0} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{(m_{11} - k_1) k_0}{m_{12}} & k_0 \end{bmatrix}.$$

Подставим выражение для k_0

$$M = \begin{bmatrix} 1 & 0 \\ \frac{(m_{22} - \det M / k_1)}{m_{12}} & k_2 \end{bmatrix} \begin{bmatrix} k_1 & \frac{k_1 k_2 m_{12}}{\det M} \\ 0 & 1 \end{bmatrix} \times$$

$$\times \begin{bmatrix} 1 & 0 \\ \frac{(m_{11} - k_1) \det M}{k_1 k_2 m_{12}} & \frac{\det M}{k_1 k_2} \end{bmatrix}. \quad (39)$$

и ещё упростим эту формулу посредством умножения/деления соответствующих строк и столбцов соседних матриц на одинаковую величину

$$M = \begin{bmatrix} 1 & 0 \\ \frac{(m_{22} - \det M / k_1)}{m_{12}} & \frac{\det M}{k_1} \end{bmatrix} \begin{bmatrix} k_1 & m_{12} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{m_{11} - k_1}{m_{12}} & 1 \end{bmatrix}, \quad (40)$$

при этом из неё уйдут k_0 и k_2 . Здесь k_0 изначально был избыточным, а k_2 уходит потому, что знак k_2 определяется знаком k_1 .

Применим формулу (40) к четырём возможным перестановкам строк и столбцов матрицы поворота (4) (в номере варианта младший бит – признак перестановки столбцов, старший – строк). В результате получаем четыре альтернативных разложения, показанных в табл. 3. Здесь ещё возможны одновременные перестановки столбцов матрицы, стоящей слева, и строк матрицы – её соседа справа, а также смены знака строк и соответствующих столбцов, которые не влияют на конечный результат. Среди этих перестановок в табл. 3 приведены такие варианты, которые делают выражения более симметричными.

Теперь вычислим ошибки округления, соответствующие каждому из вариантов разложения с использованием алгоритма из параграфа 4.3. Поскольку минимальный возможный общий знаменатель 5 при рациональных значениях косинуса и синуса угла поворота соответствует острым углам египетского треугольника, и уже это значение знаменателя согласно табл. 2 даёт лишь небольшое отклонение (4%) от непрерывного случая, будем всегда использовать оценку для непрерывного случая (24), согласно которой все $\delta_i^2 = 1/12$. Из (26) $dv = \delta_0$. Из (34) $dv = \delta_0$. По (29)

$$dy_2 = \delta_2 + b_{21} dy_1 + k_2 dv = \delta_2 + k_2 \delta_0 + b_{21} (\delta_1 + b_{12} \delta_0) =$$

$$= (k_2 + b_{12} b_{21}) \delta_0 + b_{21} \delta_1 + \delta_2. \quad (41)$$

Отсюда определяем среднеквадратичные ошибки округления по отдельным координатам:

$$\overline{dy_1^2} = \frac{1}{12} (1 + b_{12}^2), \quad (42)$$

$$\overline{dy_2^2} = \frac{1}{12} ((k_2 + b_{12} b_{21})^2 + b_{21}^2 + 1), \quad (43)$$

и суммарную среднеквадратичную ошибку

$$E = \frac{1}{12} (2 + b_{12}^2 + b_{21}^2 + (k_2 + b_{12} b_{21})^2). \quad (44)$$

Подставляя в (44) значения из Табл. 3 вариантов разложения матрицы поворота, получаем оценки ошибок округления для каждого из вариантов, представленные в Табл. 4. Видно, что для вариантов 0 и 3 эти ошибки совпадают, а для остальных они отличаются. На рис. 1 представлены графики функций ошибок в зависимости от угла поворота для различных вариантов факторизации. Для сравнения приведён график функции $E = 1/6 = 0,16$ (6), соответствующий согласно (24) среднему квадрату суммарной ошибки округления двух координат при необратимой целочисленной аппроксимации.

Таким образом, при использовании схемы лифтинга для получения обратимой целочисленной аппроксимации операции поворота для $\alpha \in [-$

$(1/4)\pi, (1/4)\pi]$ лучше использовать варианты 0^+ или 3^+ , для $\alpha \in [-(3/4)\pi, -(1/4)\pi]$ – варианты 1^+ и 2^- , для $\alpha \in [(1/4)\pi, (3/4)\pi]$ – варианты 1^- и 2^+ , а для $\alpha \in [(3/4)\pi, -(3/4)\pi]$ (на круге) – варианты 0^- или 3^- . В результате для квадрата ошибки округления E будет выполняться

$$\frac{1}{4} = 0,25 \leq E \leq \frac{3-\sqrt{2}}{6} \cong 0,2643.$$

Таким образом, использование обратимой целочисленной аппроксимации в двумерном случае увеличивает E на 50–59% по сравнению с простым округлением. Заметим, что для углов поворота, кратных $\pi/2$, выполняется $E=0$, т.к. при этом все элементы матрицы M равны ± 1 , но здесь мы рассматриваем только нетривиальные случаи, требующие использования округления.

Табл. 3. Варианты разложения матрицы поворота

| № | detM | b_{01} | b_{12} | b_{21} | Разложение |
|---|------|--|---------------|--|---|
| 0 | 1 | $\frac{k_1 - \cos\alpha}{\sin\alpha}$ | $-\sin\alpha$ | $\frac{k_1 - \cos\alpha}{\sin\alpha}$ | $\begin{bmatrix} 1 & 0 \\ \frac{k_1 - \cos\alpha}{\sin\alpha} & k_1 \end{bmatrix} \begin{bmatrix} k_1 & -\sin\alpha \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{k_1 - \cos\alpha}{\sin\alpha} & 1 \end{bmatrix}$ |
| 1 | -1 | $-\frac{\sin\alpha + k_1}{\cos\alpha}$ | $\cos\alpha$ | $\frac{\sin\alpha + k_1}{\cos\alpha}$ | $\begin{bmatrix} 1 & 0 \\ \frac{k_1 + \sin\alpha}{\cos\alpha} & k_1 \end{bmatrix} \begin{bmatrix} k_1 & -\cos\alpha \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & \frac{k_1 + \sin\alpha}{\cos\alpha} \end{bmatrix}$ |
| 2 | -1 | $\frac{\sin\alpha - k_1}{\cos\alpha}$ | $\cos\alpha$ | $-\frac{\sin\alpha - k_1}{\cos\alpha}$ | $\begin{bmatrix} \frac{k_1 - \sin\alpha}{\cos\alpha} & k_1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} k_1 & -\cos\alpha \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{k_1 - \sin\alpha}{\cos\alpha} & -1 \end{bmatrix}$ |
| 3 | 1 | $\frac{\cos\alpha - k_1}{\sin\alpha}$ | $\sin\alpha$ | $\frac{\cos\alpha - k_1}{\sin\alpha}$ | $\begin{bmatrix} k_1 & \frac{\cos\alpha - k_1}{\sin\alpha} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \sin\alpha & k_1 \end{bmatrix} \begin{bmatrix} 1 & \frac{\cos\alpha - k_1}{\sin\alpha} \\ 0 & 1 \end{bmatrix}$ |

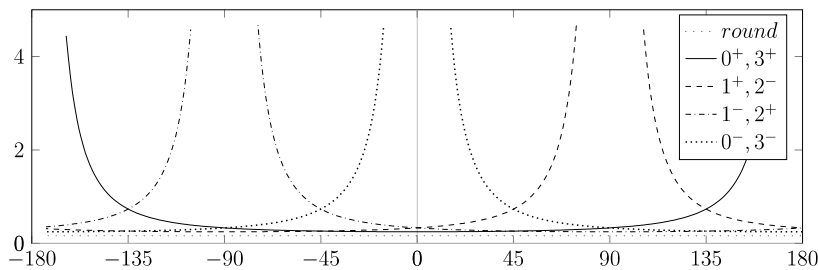


Рис. 1. Графики ошибок округления. Обозначения: N^\pm , где N – номер варианта из табл. 3, индексы: $+\Rightarrow k_1 = 1, -\Rightarrow k_1 = -1$; линия round – ошибка округления по двум координатам, равная $1/6$

Табл. 4. Ошибки округления для вариантов разложения матрицы поворота

| № | Ошибка |
|---|--|
| 0 | $\frac{1}{6} \left(1 + \frac{1 - k_1 \cos\alpha}{\sin^2\alpha} \right) = \frac{1}{6} \left(1 + \frac{1}{1 + k_1 \cos\alpha} \right)$ |
| 1 | $\frac{1}{6} \left(1 + \frac{1 + k_1 \sin\alpha}{\cos^2\alpha} \right) = \frac{1}{6} \left(1 + \frac{1}{1 - k_1 \sin\alpha} \right)$ |
| 2 | $\frac{1}{6} \left(1 + \frac{1 - k_1 \sin\alpha}{\cos^2\alpha} \right) = \frac{1}{6} \left(1 + \frac{1}{1 + k_1 \sin\alpha} \right)$ |
| 3 | $\frac{1}{6} \left(1 + \frac{1 - k_1 \cos\alpha}{\sin^2\alpha} \right) = \frac{1}{6} \left(1 + \frac{1}{1 + k_1 \cos\alpha} \right)$ |

Если же не использовать при переборе изменение знака k_i (т.е. при $k_1 = k_2 = 1$), то с этого графика придётся удалить последний вариант, представленный

$$M_{test3} = \begin{bmatrix} 0,167578095807093 & -0,832080011626454 & -0,528734750188967 \\ 0,970365522982484 & 0,233924320504726 & -0,0605818791663982 \\ -0,174092887886319 & 0,50291377643501 & -0,846622347838325 \end{bmatrix}.$$

пунктирной линией. В этом случае наибольшая ошибка $E = 1/3 = 0,33$ (3) будет достигаться в окрестности $\alpha = \pm\pi$. Этот пример показывает, что без использования отрицательных k_i для некоторых матриц не получится достичь минимальной ошибки округления. С ростом размерности вектора вероятность таких событий будет возрастать.

5.2. Факторизация случайных матриц поворота

Для тестирования алгоритмов факторизации матриц реализовано приложение, в котором генерируются матрицы M заданной размерности с $\det M = 1$ с использованием последовательности случайных поворотов (также можно считать и обработать готовые матрицы).

Приведём пример вычислительного эксперимента. Была сгенерирована следующая матрица:

В результате поиска наилучшей факторизации этой матрицы без использования смены знака (все $k_i = 1$) был найден следующий порядок вычислений

$$\begin{aligned} v &= -x_3 + [-0,587140446781492 * x_2 - 0,103131497526781 * x_1], \\ y_3 &= x_2 + [-0,0867794573141128 * x_1 + 0,846622347838325 * v], \\ y_2 &= x_1 + [0,269494392105347 * y_3 - 0,167578095807093 * v], \\ y_1 &= v + [-0,569295788686185 * y_3 + 0,176839800421345 * y_2], \end{aligned}$$

который даёт среднеквадратичную ошибку округления $\delta = 0,6038768$.

При использовании смены знака был найден следующий способ вычислений:

$$\begin{aligned} v &= -x_3 + [0,31758833387352 * x_2 + 0,036428232395694 * x_1], \\ y_1 &= -x_2 + [0,14831722345153 * x_1 + 0,528734750188967 * v], \\ y_2 &= x_1 + [-0,214684222437343 * y_1 + 0,174092887886319 * v], \\ y_3 &= v + [-0,270580419465551 * y_1 - 0,170222215147965 * y_2], \end{aligned}$$

позволяющий сократить среднеквадратичную ошибку округления до $\delta^K = 0,5866457$.

В табл. 5 приведены среднеквадратичные ошибки округления по каналам, полученные при этих порядках вычислений. Здесь δ^{round} – ошибка округления для непрерывного преобразования, δ^{exp} – экспериментальные данные, полученные сравнением результатов обратимой целочисленной аппроксимации с результатами вычислений на действительных числах для всех $2^{24} = 16777216$ значений векторов из трёх байтов; δ^{calc} – оценки этих отклонений, вычисленные по формулам из параграфа 4.3. Индекс K отмечает результаты, полученные при смене знака. Видно, насколько близки оценки к экспериментально полученным значениям.

Табл. 5. Ошибки округления для матрицы M_{est3} по каналам

| Канал | 1 | 2 | 3 |
|---------------------|-----------------|-----------------|-----------------|
| δ_i^{round} | 0,2886751370780 | 0,2886749943275 | 0,2886750796816 |
| δ_i^{exp} | 0,378233137697 | 0,2994850696208 | 0,3631997618950 |
| δ_i^{calc} | 0,378238174508 | 0,2994853072995 | 0,3631963917705 |
| $\delta_i^{exp,K}$ | 0,3265365557316 | 0,2957695715349 | 0,3873472743671 |
| $\delta_i^{calc,K}$ | 0,3265425490470 | 0,2957701077198 | 0,3873540625016 |

5.3. Факторизация матрицы с рациональными коэффициентами с небольшими знаменателями

Рассмотрим пример, демонстрирующий необходимость использования уточнений оценок, сделанных в параграфах 4.2 и 4.5, при наличии рациональных коэффициентов. В формате MRG используется обратимое целочисленное преобразование, позволяющее получить значение пикселя растра более низкого разрешения H по четырём пикселям растра L более высокого разрешения, и сохранить информацию, необходимую для восстановления значений этих исход-

ных пикселей, с использованием всего трёх дополнительных запоминаемых значений ($U, V, \text{ и } W$):

$$\begin{aligned} H_{i,j} &= \left[\frac{1}{4} (L_{2i,2j} + L_{2i,2j+1} + L_{2i+1,2j} + L_{2i+1,2j+1}) \right], \\ U_{i,j} &= \left[\frac{1}{2} (L_{2i,2j} + L_{2i,2j+1} - L_{2i+1,2j} - L_{2i+1,2j+1}) \right], \\ V_{i,j} &= \left[\frac{1}{2} (L_{2i,2j} - L_{2i,2j+1} + L_{2i+1,2j} - L_{2i+1,2j+1}) \right], \\ W_{i,j} &= L_{2i,2j} - L_{2i,2j+1} - L_{2i+1,2j} + L_{2i+1,2j+1}. \end{aligned} \quad (45)$$

Приведём соответствующее обратное преобразование:

$$\begin{aligned} w &= W_{i,j}; u = 2U_{i,j} - w \text{ mod } 2; v = 2V_{i,j} - w \text{ mod } 2, \\ h &= (2 - u - v - w) \text{ mod } 4 - 2, \\ L_{2^*i,2^*j} &= H_{i,j} + (h + u + v + w) \div 4, \\ L_{2^*i,2^*j+1} &= H_{i,j} + (h + u - v - w) \div 4, \\ L_{2^*i+1,2^*j} &= H_{i,j} + (h - u + v - w) \div 4, \\ L_{2^*i+1,2^*j+1} &= H_{i,j} + (h - u - v + w) \div 4. \end{aligned} \quad (46)$$

Заметим, что данное преобразование не укладывается в схему (5) и его было достаточно сложно придумать. Поскольку в (45) выполняется три округления рациональных выражений: два из них с общим знаменателем 2 и одно – со знаменателем 4, среднеквадратичная ошибка округления согласно Табл. 2 составляет $\delta^{HL} = \sqrt{2 * 0,125 + 0,09375} = 0,58630197$.

Посмотрим, какой способ вычислений будет найден рассматриваемым методом для соответствующей преобразованию матрицы

$$M_{HL} = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/2 & 1/2 & -1/2 & -1/2 \\ 1/2 & -1/2 & 1/2 & -1/2 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

Получаем следующую последовательность вычислений (для этой матрицы изменение знаков не позволяет улучшить результат, поэтому дальше не рассматривается)

$$\begin{aligned} v &= x_4 + x_1 - 2 * x_2 + x_3, \\ y_1 &= x_1 + [-1,5 * x_2 + x_3 - 0,5 * v], \\ y_2 &= x_2 - 2 * x_3 + v, \\ y_3 &= x_3 + y_1 + y_2 - v, \\ y_4 &= v + [-1,5 * y_1 - 0,75 * y_2 + 1,5 * y_3], \end{aligned}$$

и ошибки, показанные в Табл. 6. Из таблицы видно, что данная схема вычислений позволяет получить ошибки δ^{exp} , равные минимально возможным ошибкам округления для результатов непрерывного преобразования δ^{round} . При этом вычисленные с учётом рациональности знаменателей оценки δ_i^{calc} точно соответствуют экспериментальным данным. Итоговое

среднеквадратичное отклонение $\delta=0,58630197$ точно совпадает с ошибкой для схемы (45). Таким образом, рассматриваемый метод позволяет автоматически найти способ вычислений, который оказывается не хуже по точности, чем исходный, полученный посредством более сложных рассуждений.

Всего было найдено 32 способа вычислений с этой же точностью, здесь был приведён первый из них. Среди этих способов можно выбрать и те, которые выглядят несколько проще, например следующий:

$$\begin{aligned} v &= x_2 - x_3, \\ y_4 &= x_4 + x_1 - 2 \cdot x_3 - v, \\ y_3 &= x_1 + [-0,5 \cdot y_4 - x_3 - v], \\ y_1 &= x_3 + [0,25 \cdot y_4 + 0,5 \cdot v], \\ y_2 &= v + y_3. \end{aligned}$$

Табл. 6. Ошибки округления для матрицы M_{NL} по каналам

| Канал | 1 | 2 | 3 | 4 |
|--------------------|--------------|---|--------------|--------------|
| δ_i^{round} | 0,3535533906 | 0 | 0,3535533906 | 0,3061862178 |
| δ_i^{exp} | 0,3535533906 | 0 | 0,3535533906 | 0,3061862178 |
| δ_i^{calc} | 0,3535533906 | 0 | 0,3535533906 | 0,3061862178 |

5.4. Применение при сжатии изображений

Для анализа эффективности применения алгоритмов декорреляции цветочных каналов в программной библиотеке для работы с форматом MRG были реализованы режимы сохранения изображений, различающиеся применяемым методом декорреляции цветочных каналов (жирным выделены названия соответствующих столбцов в таблице):

- **Нет** – без декорреляции;

Табл. 7. Результаты сжатия изображений при различных вариантах декорреляции каналов

| Изображение | Сжатие, каналы | Свойства, ширина × высота, каналы × бит | Размер, МиБ | Нет, % | JP2, % от Нет | ДЗЗ, % от Нет | ДКЛ, % от Нет |
|----------------------------------|--------------------|---|-------------|--------|---------------|---------------|---------------|
| po_483237_rgb | нет RGB | 0,5 м / пикс, 15544×11332×16 | 1008,08 | 30,60 | 77,33 | 75,70 | 74,31 |
| mosaic_8bit_part2 | нет RGB | 0,5 м / пикс, 19058×39388×8 | 2148,25 | 55,16 | 82,76 | 82,37 | 81,44 |
| LC08_L2SP_133023_20210727 | deflate B2, B3, B4 | LandSat 30 м / пикс, 7881×7971,3×16 | 230,96 | 53,39 | 96,91 | 96,31 | 95,67 |
| LC81290182020146LGN00 | deflate B2, B3, B4 | LandSat 30 м / пикс, 8201×826,3×16 | 211,64 | 69,54 | 95,77 | 95,79 | 94,15 |
| S2B_MSIL1C_20210609T040549_N030 | LZW B2, B3, B4 | Sentinel 10 м / пикс, 10980×10980,3×16 | 475,33 | 66,58 | 97,99 | 96,55 | 95,33 |
| 20160819_053332_0e14_3A_Analytic | LZW RGB | PlanetScope, 3 м / пикс, 9310×6065,3×16 | 136,98 | 58,18 | 91,22 | 91,51 | 90,59 |

- **JP2** – с декорреляцией, как в JPEG 2000 (1) для трёх каналов;
- **ДЗЗ** – с новым обратимым преобразованием (2) для трёх каналов спутникового снимка;
- **ДКЛ** – с использованием обратимой целочисленной аппроксимации преобразования Карунена–Лоэва для цветочных каналов рассматриваемым в работе методом.

В текущей реализации декорреляция цветочных каналов выполняется после преобразования (45) и применяется к разностным данным U, V, W из (45). В табл. 7 приведены результаты применения различных методов декорреляции цветочных каналов для ряда тестовых спутниковых изображений, полученных из разных источников. Значения, описывающие размер сжатого файла, приведены в % к размеру исходного файла TIFF для изображений без декорреляции (столбец «Нет»), а для остальных методов – в процентах к размеру файла без декорреляции. Были использованы изображения в формате TIFF с разными способами сжатия, поэтому в столбце «Нет» значения сильно различаются. Используемый в формате MRG метод сжатия разностных целочисленных последовательностей хорошо реагирует на улучшение статистических характеристик обрабатываемых значений и позволяет судить об эффективности преобразований цветового пространства, которая, в свою очередь, зависит от степени корреляции каналов исходных данных. Для большей части обработанных изображений преобразование «ДЗЗ» оказалось эффективнее, чем преобразование «JP2», а использование целочисленной аппроксимации ДКЛ позволяет дополнительно сократить объём получаемого файла.

Заключение

В отличие от предыдущих работ, рассматривающих построение обратимых целочисленных преобразований, в данной статье дано конструктивное доказательство возможности построения обратимой целочисленной аппроксимации для линейных преобразований, матрица которых имеет единичный по модулю определитель.

Предложен метод оценивания среднеквадратичных ошибок округления, связанных с конкретным способом аппроксимации. При этом отдельно исследован вопрос вычисления оценок для округляемых выражений с рациональными и целочисленными коэффициентами. Хотя приведённые в статье вычислительные эксперименты продемонстрировали 100% точности этих оценок, возможны более сложные зависимости между округляемыми выражениями, которые всё-таки могут потребовать их дополнительного уточнения. Заметим, что все такие уточнения касаются аналитически задаваемых матриц с рациональными и целочисленными коэффициентами, а также с повторяющимися фрагментами и не имеют отношения к матрицам, получаемым в результате статистического анализа реальных данных.

Разработан алгоритм поиска наилучшего способа аппроксимации линейного преобразования, который минимизирует суммарную среднеквадратичную ошибку округления. При этом рассматриваются все варианты перестановок строк и столбцов матрицы, а также возможность смена знака изменяемой координаты. Приведены примеры, демонстрирующие то, что в ряде случаев только смена знака некоторых координат позволяет получить минимально возможную ошибку округления.

Рассмотрен пример использования обратимых аппроксимаций преобразования Карунена–Лоэва для повышения эффективности сжатия без потерь многоканальных растровых данных.

Благодарности

Работа выполнена в рамках гранта № 075-15-2020-787 Министерства науки и высшего образования РФ на выполнение крупного научного проекта по приоритетным направлениям научно-технологического развития (проект «Фундаментальные основы, методы и технологии цифрового мониторинга и прогнозирования экологической обстановки Байкальской природной территории»).

References

- [1] Hmelnov AE. The MRG file format for compact representation and fast decompression of large digital elevation models [In Russian]. Computational Technologies 2015; 20(1): 63-74.
- [2] Hmelnov AE. A lossless compression algorithm for integer differences sequences by optimization of their division into intervals of constant bit depth values [In Russian]. Computational Technologies 2015; 20(3): 75-98.
- [3] Christopoulos C, Skodras A, Ebrahimi T. The JPEG2000 still image coding system: an overview. IEEE Trans Consum Electron 2000; 46(4): 1103-1127. DOI: 10.1109/30.920468.
- [4] Recommendation ITU-R BT.601-7 (03/2011). Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios. ITU; 2017.
- [5] Dony RD. Karhunen-Loève transform. In Book: Rao KR, Yip PC, eds. The transform and data compression handbook. Boca Raton: CRC Press LLC; 2001.
- [6] Daubechies I, Sweldens W. Factoring wavelet transforms into lifting steps. J Fourier Anal Appl 1998; 4: 247-269. DOI: 10.1007/BF02476026.
- [7] Hao P, Shi Q. Matrix factorizations for reversible integer mapping. IEEE Trans Signal Process 2001; 49(10): 2314-2324. DOI: 10.1109/78.950787.
- [8] Hao P, Shi Q. Invertible linear transforms implemented by integer mapping [In Chinese]. Science in China, Series E 2000; 30(2): 132-141.
- [9] Hao P, Shi Q. Comparative study of color transforms for image coding and derivation of integer reversible color transform. 15th Int Conf on Pattern Recognition (ICPR) 2000; 3: 224-227.
- [10] Hao P, Shi Q. Reversible integer KLT for progressive-to-lossless compression of multiple component images. Proc 2003 Int Conf on Image Processing (Cat No 03CH37429) 2003: I-633. DOI: 10.1109/ICIP.2003.1247041.
- [11] Heap BR. Permutations by interchanges. Comput J 1963; 6(3): 293-294. DOI: 10.1093/comjnl/6.3.293.

Сведения об авторе

Хмельнов Алексей Евгеньевич, 1968 года рождения, в 1991 году окончил Московский физико-технический институт по специальности «Прикладная математика и физика», кандидат технических наук (2000 год), работает заведующим лабораторией Комплексных информационных систем в ИДСТУ СО РАН. Области научных интересов: обработка пространственных данных, бинарные форматы данных, автоматизация разработки приложений с использованием спецификаций.

E-mail: hmelnov@icc.ru.

ГРНТИ: 27.41.41

Поступила в редакцию 21 сентября 2021 г. Окончательный вариант – 9 февраля 2022 г.

Reversible integer approximation of color space transforms for lossless compression of big color raster data

A.E. Hmelnov¹

¹ISDCT SB RAS – Matrisov Institute for System Dynamics and Control Theory of Siberian Branch of RAS,
664033, Irkutsk, Russia, Lermontova 134

Abstract

Reversible integer transforms are of great importance for lossless compression algorithms. To perform reversible decorrelation of color channels we propose an algorithm for calculating parameters of a reversible integer transform, which approximates such continuous mappings as a discrete Karunen-Loeve transform. We propose a method for estimating the approximation errors, which makes it possible to choose an optimal approximation of the original transform that minimizes these errors. Using the MRG file format, intended for storing large amounts of integer raster data, as an example, we show that after applying decorrelation, it is possible to increase the compression ratio of multichannel raster images using lossless compression algorithms.

Keywords: decorrelation, reversible integer transform, lossless compression, data-specific compression algorithms.

Citation: Hmelnov AE. Reversible integer approximation of color space transforms for lossless compression of big color raster data. *Computer Optics* 2022; 46(3): 492-505. DOI: 10.18287/2412-6179-CO-1052.

Acknowledgements: The work was supported by the Ministry of Science and Higher Education of the Russian Federation within a framework of major scientific projects in priority areas of scientific and technological development under grant No. 075-15-2020-787 (the project "Fundamentals, methods and technologies for digital monitoring and forecasting of the environmental situation on the Baikal natural territory")

Author's information

Alexei Evgenievich Hmelnov (b. 1968) graduated from Moscow Institute of Physics and Technology 1991, majoring in Applied Mathematics and Physics. He received his PhD in Technical Sciences in 2000. Currently he works as the head of Laboratory of Complex Information Systems at the Institute for System Dynamics and Control Theory of Siberian Branch of RAS. Research interests are spatial data processing, binary data formats and automation of application development using specifications. E-mail: hmelnov@icc.ru.

Received September 21, 2021. The final version – February 09, 2022.
