«Оптика атмосферы и океана», **14,** № 1 (2001)

УДК 535.36

А.А. Мицель[1,2], М. Ю. Катаев[1,3], И.Г. Окладников[1]

Сжатие банка коэффициентов поглощения атмосферных газов

[1] *Томский государственный университет систем управления и радиоэлектроники*
[2] *Институт оптического мониторинга СО РАН,*
[3] *Институт оптики атмосферы СО РАН, г. Томск*

Поступила в редакцию 20.12.2000 г.

Рассмотрены методы сжатия табличных данных. Показано, что для таблиц коэффициентов поглощения наиболее оптимальным методом является компрессия с помощью SVD-преобразования матриц. Приводятся примеры сжатия данных различными методами и их сравнение.

1. Постановка задачи

Необходимость решения задач оптики газовой атмосферы возникает при проектировании и инженерной проработке оптических систем мониторинга воздушного бассейна, оптической связи и других приборов, в основе которых лежит идея измерения излучения, прошедшего через атмосферу. Базовой характеристикой для решения задач переноса ИК-излучения в полосах поглощения является коэффициент поглощения. Эту характеристику требуется многократно рассчитывать для различных значений давления и температуры атмосферы на разных частотах. Расчет коэффициента поглощения может занимать до 75% общего вычислительного времени [1].

Одним из способов существенного сокращения времени счета является подход, основанный на однократном вычислении КП в узлах некоторой оптимизированной трехмерной сетки по переменным ν, *T*, *P*. Полученные значения хранятся в виде одного или нескольких структурированных файлов с возможностью последующего вычисления *k* в произвольной точке (ν, *T*, *P*) с использованием интерполяции по ближайшим узлам сетки [1, 2]. Такие файлы называются поисковыми таблицами (Look-Up Tables – LUT). Поскольку созданные таблицы занимают значительный объем памяти, то их необходимо сжать. На сегодняшний день существует широкий класс разнообразных методов сжатия, архиваторов и утилит сжатия информации. В данной статье рассмотрены некоторые математические алгоритмы разложения и преобразования данных, а также популярные архиваторы.

2. Методы сжатия и архивирование наборов данных

2.1. Спектральные методы сжатия информации

Дискретное Фурье-преобразование

Пусть N – произвольное натуральное число. Дискретное преобразование Фурье-вектора

$$X = \begin{pmatrix} x_0 \\ x_1 \\ \text{M} \\ x_{N-1} \end{pmatrix}$$

имеет вид

$$Y_n = \frac{1}{N}\sum_{k=0}^{N-1} x_k \exp\left(\frac{-2\pi i}{N} kn\right), \; n = 0, 1, \ldots, N-1. \qquad (1)$$

Чтобы получить преобразование, обратное преобразованию (1), умножим n-е уравнение в (1) на $\exp\left(\frac{2\pi i}{N} nl\right)$, $n = 0, 1, \ldots, N-1$ и полученные равенства сложим. Тогда коэффициенты при x_k в правой части этой суммы будут равны:

$$\frac{1}{N}\sum_{k=0}^{N-1} x_k \exp\left(\frac{-2\pi i}{N} kn\right)\exp\left(\frac{2\pi i}{N} nl\right) =$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} x_k \exp\left(\frac{-2\pi i}{N} n(l-k)\right) = \begin{cases} 0 l \neq k \\ 1 l = k \end{cases}.$$

Поэтому преобразование, обратное (1), будет иметь вид

$$x_k = \sum_{k=0}^{N-1} Y_n \exp\left(\frac{2\pi i}{N} kn\right), \; k = 0, 1, \ldots, N-1. \qquad (2)$$

Это преобразование обладает той особенностью, что для многих сигналов X спектральные характеристики $\{y_n\}$ могут концентрироваться в окрестности начала координат. Другими словами, для восстановления исходного сигнала с заданной точностью достаточно использовать лишь несколько первых элементов в преобразовании Фурье [3]. Таким образом, для сжатия необходимо сохранять только отобранные элементы. Чтобы выявить, какие коэффициенты оставить, а какие нет, используется некоторая пороговая функция, благодаря которой из всех элементов преобразования сохраняются только те, чье абсолютное значение больше некоторого порога. Очевидно, что выбор порога влияет на степень сжатия и погрешность при восстановлении спектра.

Этот метод не рекомендуется использовать для сжатия инфракрасных спектров высокого разрешения из-за их сильно выраженной нелинейности, однако может быть использован для сжатия гладких спектров УФ-диапазона либо спектров низкого разрешения.

Дискретное вэйвлет-преобразование (DWT)

Вэйвлет-преобразование включает в себя разложение функции сигнала или вектора (например, спектра) на более простые фиксированные строительные блоки с различными масштабами и положениями [4].

Так же, как и Фурье-преобразование, вэйвлет-преобразование работает с сигналом $f(\lambda)$ и преобразует его из пространства сигнала, которым является длина волны для инфракрасного спектра, в другое пространство. Однако в отличие от Фурье-преобразования, чье частотное пространство является одномерным, вэйвлет-преобразование создает двухмерное пространство с двумя параметрами: масштабным параметром a и пространственным параметром b. Это свойство является преимуществом вэйвлет-преобразования (BW) по сравнению с Фурье-преобразованием, тем не менее BW-преобразование дает неполное описание сигнала при данных частотах вдоль всего пространства длин волн. С другой стороны, в то время как для Фурье-преобразования в качестве базисных функций используются функции синуса и косинуса, для вэйвлет-преобразования существует множество способов выбора материнского вэйвлета $\psi(\lambda)$ и базисных функций $\psi_{a,b}(\lambda)$, которые могут быть получены следующим образом:

$$\psi_{a,b}(\lambda) = a^{-1/2}\,\Psi(\{\lambda - b\}/a). \qquad (3)$$

Чтобы применить дискретное вэйвлет-преобразование к оцифрованному спектру, приняты следующие параметры вэйвлета: $a = 2^j$ и $b = 2^j k$. Отсюда из уравнения (3) можно записать, что

$$\psi_{j,k}(\lambda) = 2^{-j/2}\,\Psi(2^{-j}\,\lambda - k).$$

Здесь переменные j и k – величины растяжения и сдвига соответственно. Разложение $f(\lambda)$ относительно вэйвлет-функций $\{\psi_{j,k}(\lambda)\}$ описывается формулой

$$f(\lambda) = \sum_{j}^{\infty}\sum_{k}^{\infty} c_k^{(j)}\,\psi_{j,k}(\lambda)\,.$$

Из нее можно сделать вывод, что сигнал представляется последовательностью коэффициентов $c_k^{(j)}$.

В быстром вэйвлет-преобразовании эти коэффициенты могут быть вычислены с помощью следующих рекуррентных формул:

$$c_k^{(j)} = \sqrt{2}\,\Sigma_n\, c_n^{(j-1)}\, h_{n-2k}\;;\; d_k^{(j)} = \sqrt{2}\,\Sigma_n\, c_n^{(j-1)}\, g_{n-2k}\;,$$

где n изменяется от $-\infty$ до $+\infty$. Переменные h_k и g_k называются коэффициентами низкочастотного ($G = \{g_k\}$) и высокочастотного ($H = \{hk\}$) фильтров соответственно.

Возможность разложения сигнала для сжатия лежит в способности процедуры вэйвлет-преобразования сконцентрировать большой процент общей энергии сигнала в $c^{(j)}$ на различных уровнях детализации j. Так как коэффициенты $D^{(j)}$ создаются с помощью высокочастотного фильтра G, то эти вэйвлет-коэффициенты отражают высокочастотную информацию. В спектрах высокочастотная составляющая – это, как правило, шум, и ее можно отбросить. Таким образом, для сжатия необходимо сохранять только отобранные вэйвлет-коэффициенты. Отбор, какие коэффициенты оставить, а какие отбросить, производит некоторая пороговая функция. В настоящее время существует несколько различных процедур, реализующих отбор. Одна из них заключается в том, что из всех вэйвлет-коэффициентов сохраняются только те, чье абсолютное значение больше некоторого порога. Очевидно, что выбор порога влияет на эффективность сжатия и качество восстановления спектра. Как правило, больший порог дает лучшее сжатие, но более худшее восстановление спектра.

Практическое использование [4] показало, что этот метод имеет слабое применение для инфракрасных спектров [5], так как они в отличие от ультрафиолетовых и видимых спектров сильно изрезаны и в основном состоят из множества острых пиков. Это приводит к большому числу высокочастотных компонент в вэйвлет-представлении и увеличивает количество значимых элементов. Таким образом, возникает необходимость хранения большого количества величин, что приводит к снижению эффективности сжатия. В данном случае для увеличения силы сжатия можно воспользоваться дополнительными методами, например квантованием сигнала и кодированием Хаффмана [4]. Однако эти преобразования дают незначительный вклад в сжатие, значительно увеличивая время работы алгоритма.

2.2. Математические методы сжатия информации

SVD-преобразование данных

Разложение сингулярных значений (Singular Value Decomposition – SVD) – это хорошо известная методика ортогонального разложения наборов данных [6]. Цель алгоритма состоит в поиске ортогональных матриц U и V таких, чтобы Σ-матрица

$$U^{\mathrm{T}}\,AV = \Sigma$$

была диагональной. Обе эти матрицы получаются как произведения ортогональных матриц, называемых хаусхолдеровыми отражениями; T в индексе означает транспонирование матрицы.

Схема разложения произвольной $m \times n$ матрицы A может быть использована для сжатия произвольных наборов данных [7]. Пусть исходная матрица представляется в виде произведения трех матриц:

$$A = U\,\Sigma\,V^{\mathrm{T}}, \qquad (4)$$

где ортонормальная матрица U имеет размер $m \times n$; Σ – диагональная $m \times n$ матрица; V – $m \times n$ ортонормальная матрица. На диагонали Σ стоят сингулярные значения матрицы A, обычно расположенные по убыванию. Нас интересует ситуация, когда большинство сингулярных значений малы. Предполагая, что из всех сингулярных значений оставим только L наибольших, уравнение (4) перепишется в виде

$$A_{ij} = \left(\sum_{k=1}^{L} U_{ik}\;\sigma_k\;V_{jk}\right) + \left(\sum_{k=L+1}^{n} U_{ik}\;\sigma_k\;V_{jk}\right), \qquad (5)$$

где σ_k – сингулярные значения (диагональные элементы матрицы Σ). Если действительно только первые L сингулярных значений существенны, то матрица A может быть аппроксимирована отбрасыванием второго слагаемого в выражении (5). Это очень эффективно сокращает размеры матриц разложения, так что матрица U становится $m \times L$, Σ – $L \times L$, а V – $n \times L$. Таким образом, точность представления данных будет определяться величиной L. На практике часто удается уменьшить количество собственных векторов $\{U_j\}$ в несколько раз (иногда в десятки раз) при потере точности восстановления исходных величин в несколько процентов.

Используя SVD-подход, матрицу монохроматических коэффициентов поглощения A можно хранить в виде двух матриц – $\hat{A}$ и U:

$$\hat{A} = U^{\mathrm{T}} A = \Sigma V^{\mathrm{T}}.$$

Требуемый объем памяти для этих двух матриц в несколько десятков раз меньше, чем для исходной матрицы A. Для получения исходной матрицы остается только перемножить $\hat{A}$ и U, т.е.

$$\bar{A} = \hat{A}\ U. \qquad (6)$$

Полученная по формуле (6) матрица $\bar{A}$ будет отличаться от точной матрицы A в пределах заданной погрешности, которая, в свою очередь, будет определяться объемом «усечения» матриц SVD-разложения.

Метод сжатия сингулярных значений показал очень хорошие результаты как по точности, так и по эффективности сжатия. Кроме того, достаточно высокая скорость работы делает его одним из основных и предпочтительных методов сжатия.

Преобразование Карунена – Лоэва

Первый этап в преобразовании Карунена – Лоэва заключается в факторном анализе матрицы данных [8–10]. Факторный анализ оперирует матрицей данных с k строками и i столбцами. При этом каждая строка представляет собой нормализованный спектр. Каждый столбец в матрице соответствует конкретной частоте в инфракрасном диапазоне. Задача факторного анализа заключается в уменьшении размерности матрицы с $k \times i$ до $k \times j$, где $j < i$, причем новая матрица описывает исходные данные с установленной точностью [11].

Второй этап в преобразовании Карунена–Лоэва включает линейное отображение исходной матрицы в оптимизированную систему координат с помощью следующего преобразования:

$$T_{k,j} = \sum_{n=1}^{j} D_{k,j}\, E_{i,n}\ ,$$

где $E_{i,n}$ – i-я компонента n-го собственного вектора; $D_{k,j}$ – i-я компонента k-й строки матрицы (т.е. k-го спектра в библиотеке); $T_{k,j}$ – j-я компонента k-й строки преобразованной матрицы, где $j < i$. Это линейное отображение эквивалентно нахождению проекции вектора каждого спектра на каждую ось в новой системе координат. Проекция D_k на наиболее значимый собственный вектор становится первым значением в представлении преобразованного вектора. Последующие величины вычисляются для всех j собственных векторов, использованных для описания новой системы координат. Таким образом, исходное i-мерное векторное представление спектра линейно преобразуется в j-мерный вектор, а так как $j < i$, то достигается сжатие данных.

Экспериментальные проверки авторами [11] показали, что метод Карунена – Лоэва способен сжать инфракрасные спектры эфира и некоторых других химических соединений в 5 раз с допустимой погрешностью.

Аппроксимация кубическим полиномом

Еще одним подходом к сжатию данных могут стать процедуры аппроксимации кубическими полиномами. Поскольку график коэффициента поглощения по давлению и температуре является достаточно гладким, то применение этих процедур может дать существенное сжатие с небольшими погрешностями.

В качестве процедуры аппроксимации можно взять аппроксимацию Чебышева. Суть этого метода заключается в нахождении N коэффициентов полинома для таблично заданной функции, используя которые можно вычислить исходную функцию в произвольной точке. Для фиксированного N уравнение $f(x) \approx \left[\sum_{k=1}^{N} c_k\ T_{k-1}(x)\right] - \frac{1}{2}\ c_1$ является полиномом по x, аппроксимирующим функцию $f(x)$ на интервале [– 1, 1]. Достоинством этого полинома является то, что при его усечении до более *низкой* степени $m \ll N$ полином Чебышева дает наилучшую аппроксимацию для степени m по сравнению с другими полиномиальными схемами. Пусть N достаточно велико, чтобы обеспечить точную аппроксимацию $f(x)$. Тогда усеченная аппроксимация будет иметь вид $f(x) \approx \left[\sum_{k=1}^{m} c_k\ T_{k-1}(x)\right] - \frac{1}{2}\ c_1$ с теми же c_j. Поскольку $T_k(x)$ ограничен между ± 1, то разница между точной и усеченной аппроксимациями будет не больше суммы отброшенных c_k, $k = m + 1, \ldots, N$.

Для кубического полинома требуются четыре коэффициента. Этого количества достаточно, чтобы с высокой точностью восстановить исходный вектор. В результате проведенных экспериментов было выяснено, что с помощью аппроксимации кубическим полиномом средняя величина ошибки не превышает 0,04% при максимальной – менее 0,1%. Степень сжатия зависит от количества элементов в сжимаемом векторе. Так, например, вектор длиной в 10 элементов будет сжат в 2,5 раза.

2.3. Статистические методы сжатия информации

Метод LZW-сжатия данных

Lempel-Ziv-Welch (LZW)-сжатие – известная, повсеместно используемая техника. Этот метод лежит в основе таких популярных программ-архиваторов, как PKZIP, LHA, ARJ. Алгоритм метода очень простой. LZW-сжатие заменяет строки символов некоторыми кодами без какого-либо анализа входного текста. Сжатие происходит, когда код заменяет строку символов. Коды, генерируемые LZW-алгоритмом, могут быть любой длины, но они должны содержать больше бит, чем единичный символ. Первые 256 кодов (когда используются 8-битные символы) по умолчанию соответствуют стандартному набору символов. Остальные коды соответствуют обрабатываемым алгоритмом строкам [12].

Достаточно трудно охарактеризовать результативность какой-либо техники сжатия данных. Степень сжатия определяется различными факторами. LZW-сжатие выделяется среди прочих, когда встречается с потоком данных, содержащим повторяющиеся строки любой структуры. По этой причине он работает весьма эффективно, когда встречает текст. Уровень сжатия может достигать 50% и выше.

Однако при сжатии файлов данных могут возникнуть трудности. В зависимости от исходных данных результат сжатия может быть как хорошим, так и не очень удовлетворительным.

Метод Хаффмана

Метод Хаффмана – статистический метод сжатия, который уменьшает среднюю длину кодового слова для символов алфавита. Код Хаффмана является примером кода, оптимального в случае, когда все вероятности появления символов в сообщении – целые отрицательные степени двойки. Код Хаффмана может быть построен по следующему алгоритму.

1. Выписываем в ряд все символы алфавита в порядке возрастания или убывания вероятности их появления в тексте.

2. Последовательно объединяем два символа с наименьшими вероятностями появления в новый составной символ, вероятность появления которого полагается равной сумме вероятностей составляющих его символов; в конце концов мы построим дерево, каждый узел которого имеет суммарную вероятность всех узлов, находящихся ниже него.

3. Прослеживаем путь к каждому листу дерева, помечая направление к каждому узлу (например, направо – 1, налево – 0).

Для заданного распределения частот символов может существовать несколько кодов Хаффмана. Можно определить «каноническое» дерево Хаффмана, выбрав одно из возможных деревьев. Такое каноническое дерево может быть очень компактным, передавая только длину в битах для каждого кодового слова. Такой метод используется в большинстве архиваторов.

Арифметическое кодирование

Этот метод основан на идее преобразования входного потока в одно число с плавающей запятой. Естественно, что чем длиннее сообщение, тем длиннее получающееся в результате кодирования число. На выходе арифметического компрессора получается число, меньше 1 и больше либо равное 0. Из этого числа можно однозначно восстановить последовательность символов, из которых оно было построено [13].

Эксперименты на различных уровнях показывают, что арифметическое кодирование всегда дает результаты не хуже, чем кодирование Хаффмана. В некоторых случаях выигрыш может быть очень существенным. Однако в силу того, что объем вычислений, необходимых при работе алгоритма арифметического кодирования, значительно выше, чем при кодировании по методу Хаффмана, он работает медленнее. Арифметическое кодирование может быть использовано в тех случаях, когда степень сжатия важнее, чем временные затраты на сжатие информации.

Выводы

На сегодняшний день существует широкий класс разнообразных архиваторов и утилит сжатия информации, но основными критериями выбора алгоритмов сжатия являются, во-первых, скорость работы при декомпрессии данных и, во-вторых, возможность извлечения из сжатого представления произвольного участка исходного набора данных, без декомпрессии всего файла. Этим условиям не удовлетворяют большинство сжимающих алгоритмов. Поэтому внимание было обращено на различные алгоритмы разложения и преобразования данных, основанные на математическом сжатии и спектральном анализе: это разложение сингулярных значений, дискретное вэйвлет-преоб-разование и преобразование Карунена – Лоэва. Все эти алгоритмы работают очень быстро и способны работать с любым участком сжатого представления данных. Кроме того, так как исходные данные изначально получаются в текстовом виде, то, в некотором роде, алгоритмом сжатия является простое преобразование в двоичный формат данных. Это дает значительное дополнительное сокращение объема хранимой информации без потери точности.

Для проверки эффективности работы различных методов сжатия был проведен ряд экспериментов. Результаты проверки приведены в таблице и на рис. 1 и 2.

Как видно из таблицы, сжатие с помощью SVD показало очень хорошие результаты по точности и скорости вычислений. Оно обеспечило пятикратное сжатие данных, опередив популярные архиваторы, обеспечив при этом точность вычислений, сопоставимую с современными методами полинейного счета коэффициента поглощения. К сожалению, несмотря на высокую скорость работы, вэйвлет-преобразование, ввиду малого количества элементов в сжимаемых векторах, дало низкую степень сжатия и большую погрешность восстановления.

Сравнение результатов работы методов сжатия данных

Компрессор	Время расчета, с	Время разжатия, с	Степень сжатия, раз	Средняя погрешность, %	Максимальная погрешность, %
Без сжатия	16,40	0	0	0	0
SVD	16,81	0,44	5	0,001	Менее 0,05
Wavelet	16,31	0,28	1,38	10^{-5} – 0,7	0,008 – 2,94
RAR	17,51	0,47	3,25	0	0
ZIP	17,02	0,21	3,25	0	0
ARJ	17,11	0,23	3,16	0	0

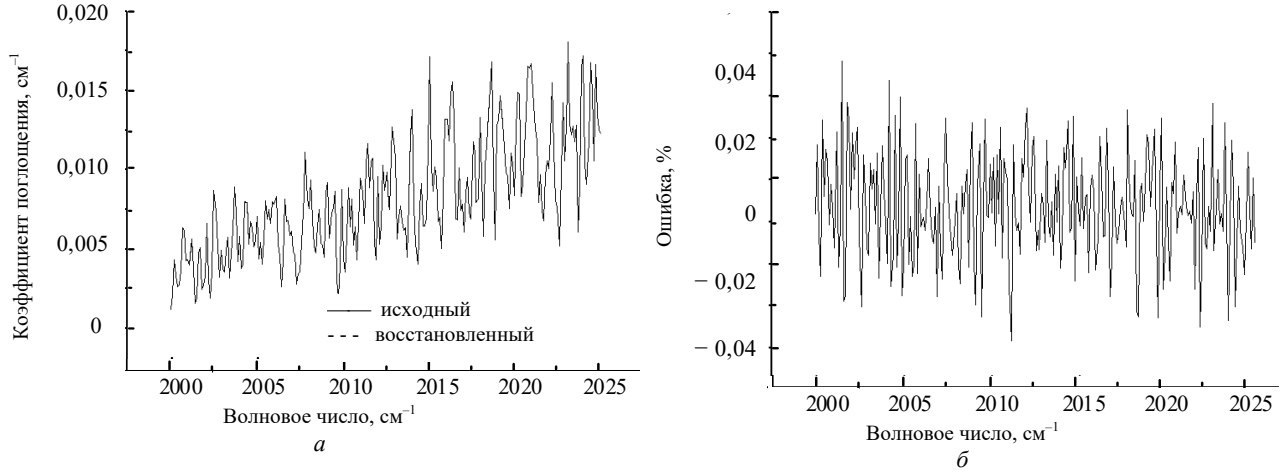

Рис. 1. Исходный и восстановленный коэффициенты поглощения при SVD-сжатии (*а*); ошибки восстановления коэффициентов поглощения при SVD-сжатии (*б*); газ – озон; высота $H = 0$ км; давление 0,91 атм; $T = 278{,}2$ К

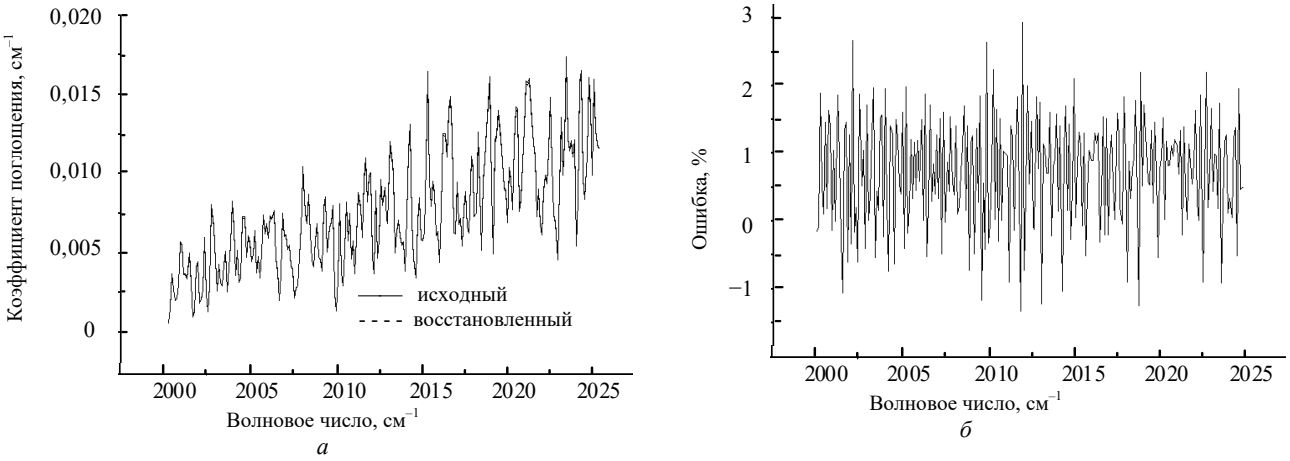

Рис. 2. Исходный и восстановленный коэффициенты поглощения при DWT-сжатии (*а*); ошибки восстановления коэффициентов поглощения при DWT-сжатии (*б*); газ – озон; высота $H = 0$ км; давление 0,91 атм; $T = 278{,}2$ К

По сравнению с SVD архиваторы обеспечили худшую степень сжатия, хотя и не внесли погрешности. В общем случае они не могут быть использованы для сжатия таблиц, так как не позволяют извлекать произвольные данные из архивов, но их можно использовать для дополнительного сжатия либо как альтернативные способы в случае, когда потребуется отсутствие какой-либо погрешности.

Работа выполнена при финансовой поддержке РФФИ (грант № 00-07-90175).

1. *Turner D.S.* Absorption coefficient estimation using a two-dimensional interpolation procedure // J. Quant. Spectrosc. Radiat. Transfer. 1995. V. 53. № 6. P. 633–637.
2. *Strow L.L., Motteler H.E., Beuson R.G., Hannon S.E. and de Souzo-Mackodo S.* Fast computation of monochromatic infrared atmospheric transmittances using compressed look-up tables // JQSRT. 1998. V. 59. P. 481–493.
3. *Разработка* математических методов и алгоритмов предварительной обработки и кодирования сложных изображений: Технический отчет. М.: МИЭТ, 1976. С. 88, 118–127.
4. *Chau F.T., Gao J.B., Shih T.M., and Wang J.* Compres-sion of Infrared Spectral Data Using the Fast Wavelet Transform Method // Appl. Spectrosc. 1997. V. 51. № 5. P. 649–659.
5. *Chau F.T., Shih T.M., Gao J.B., and Chan C.K.* Application of the Fast Wavelet Transform Method to Compress Ultraviolet-Visible Spectra // Appl. Spectrosc. 1996. V. 50. № 3. P. 649–659.
6. *Форсайт Дж., Малькольм М., Моулер К.* Машинные методы математических вычислений / Пер. с англ. М.: Мир, 1980. 280 с.
7. *Harrington P.B., Isenhour T.L.* Compression of IR libraries by Eigenvector projection // Appl. Spectrosc. 1987. V. 41. № 3. P. 449–453.
8. *Funke P.T., Malinowski E.R., Martire D.E., and Pollara L.Z.* // Sep. Sci. 1966. V. 1. P. 661.
9. *Howery D.G.* // Am. Lab. 1976. V. 8. P. 14.
10. *Rummel R.J.* Appl. Factor Analys. Northwest Univ. Press, Evanston. III. 1970.
11. *Hangac G., Weiboldt R.C., Lam R.B., and Isenhour T.L.* Compression of an Infrared Spectral Library by Karhunen – Loeve Transformation // Appl. Spectrosc. 1982. V. 36. № 1. P. 41–47.
12. *Rucker R.* Mind Tools. Houghton Miffilin Company, Boston, Mass. 1987. 234 с.
13. *Witten Ian H., Neal Redford M., Clearly John G.* Arithmetic coding for data compression // Communications of the ACM. 1987. V. 30. № 6. P. 16–21.

A.A. Mitsel, M.Yu. Kataev, I.G. Okladnikov. **Compaction of data bank of the absorption coefficients of atmospheric gases.**

Methods of compaction of table data are considered. The compression of tables of the absorption coefficients by the method of SVD transform of matrices is shown to be optimal. Different methods of the data compression are presented and compared.