Compaction of data bank of absorption coefficients of atmospheric gases

A.A. Mitsel,[1,2] M.Yu. Kataev,[1,3] and I.G. Okladnikov[1]

[1] *Tomsk University of Control Systems and Radioelectronics, Tomsk*
[2] *Institute of Optical Monitoring, Siberian Branch of the Russian Academy of Sciences, Tomsk*
[3] *Institute of Atmospheric Optics, Siberian Branch of the Russian Academy of Sciences, Tomsk*

Received December 20, 2000

Methods of compaction of tabular data are considered. It is shown that compression with SVD transform of matrices is the optimal method for tables of absorption coefficients. Examples of compaction by different methods are presented and compared.

1. Statement of the problem

When designing optical systems for air monitoring, optical communication systems, and other devices based on the idea of measuring the radiation passed through the atmosphere, the necessity for solving optical problems of the gaseous atmosphere arises. The absorption coefficient is the basic characteristic for solving the problems of transfer of IR radiation in absorption bands. This characteristic must be calculated many times for different values of atmospheric pressure and temperature at different frequencies. Calculation of the absorption coefficient can take up to 75% of the total computational time.[1]

To save the computational time, we used an approach based on single calculation of the absorption coefficient at the nodes of some optimized 3D grid in the variables ν, T, and P. The obtained values were then stored in one or several structured files with the possibility of further calculation of k in an arbitrary point (ν, T, P) using interpolation among the closest nodes.[1,2] Such files are called look-up tables (LUT). Since the tables occupy large memory, they should be compacted. Now there exists a wide spectrum of various compaction methods, archivers, and data compression utilities. In this paper, we consider some mathematical algorithms of data expansion and conversion, as well as popular archivers.

2. Methods of data compression and archiving

2.1. Spectral methods of data compression

Discrete Fourier transform

Let N be an arbitrary natural number. Discrete Fourier transformation of a vector

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

has the form

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

To obtain the transformation inverse to transformation (1), the nth equation in Eq. (1) is multiplied by $\exp\left(\frac{2\pi i}{N} nl\right)$, $n = 0, 1, \ldots, N-1$, and the obtained equalities are summed up. Then the coefficients of x_k in the right-hand side of this sum are equal to:

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

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

Therefore, the transformation inverse to Eq. (1) has the form:

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

The feature of this transformation is that for many signals X the spectral characteristics $\{y_n\}$ can concentrate near the origin of coordinates. In other words, several first elements in the Fourier transform bear all the information sufficient for reconstruction of the initial signal with some preset accuracy.[3] Thus, only selected elements should be stored for compression. To determine them, some threshold function is used. Only those elements are chosen to be stored, whose absolute values are higher than some threshold ones. Obviously, the choice of the threshold values affects the degree of compression and the error of reconstruction of the spectrum.

This method is not recommended for compression of high-resolution IR spectra because of their pronounced nonlinearity, but it can be used for compression of smooth UV spectra or low-resolution spectra.

0235-6880/01/01 53-05 $02.00

Discrete wavelet transformation (DWT)

Wavelet transformation includes resolution of a function of a signal or vector (for example, spectrum) into simpler fixed blocks with different scales and positions.[4]

Like the Fourier transform, the wavelet transformation deals with the signal $f(\lambda)$ and transforms it from the space of the signal (in which the wavelength serves as a parameter) to another space. However, in contrast to the Fourier transform, whose frequency space is one-dimensional, the wavelet transformation generates a 2D space with two parameters: the scaling parameter a and the spatial parameter b. This property is an advantage of the wavelet transformation as compared to the Fourier transform. However, the wavelet transformation gives incomplete description of a signal at given frequencies along all the space of wavelengths. On the other hand, while the Fourier transform uses sine and cosine functions as basis functions, for the wavelet transformation there exist a number of methods for choosing the parent wavelet $\Psi(\lambda)$ and the basis functions $\psi_{a,b}(\lambda)$, which can be obtained as follows:

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

To apply the discrete wavelet transformation to a digitized spectrum, the following wavelet parameters were taken: $a = 2^j$ and $b = 2^j k$. From Eq. (3) we can write that

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

Here the variables j and k are the values of spreading and shift, respectively. Expansion of $f(\lambda)$ in terms of wavelet functions $\{\psi_{j,k}(\lambda)\}$ is described by the equation

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

Herefrom we can conclude that the signal can be presented as a series of the coefficients $c_k^{(j)}$.

In the fast wavelet transformation, these coefficients can be calculated with the following recurrence equations:

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

where n varies from $-\infty$ to $+\infty$. The variables h_k and g_k are called the coefficients of low-frequency filter ($H = \{h_k\}$) and high-frequency filter ($G = \{g_k\}$) filters, respectively.

A possibility to expand a signal for its compression is provided by the wavelet transformation procedure capability to concentrate a large fraction of the signal total energy in $c^{(j)}$ at different levels of refining j. Since the wavelet coefficients $d^{(j)}$ are generated with the help of the high-frequency filter G, they reflect the high-frequency information. The high-frequency component in the spectra is, as a rule, noise, and it can be rejected. Thus, only selected wavelet coefficients should be stored for compression. The coefficients are selected using some threshold function. Now a few different selection procedures are available. In one of them, only those wavelet coefficients are selected, whose values are higher than the threshold. Obviously, the choice of the threshold affects both the efficiency of compression and the quality of reconstruction of the spectrum. As a rule, a higher threshold gives better compression, but worse reconstruction.

The practical implementation[4] has shown this method to be hardly applicable to IR spectra,[5] because they represent a lot of sharp peaks, in contrast to the smoother UV and visible spectra. This leads to a great number of high-frequency components in the wavelet representation and increases the number of significant elements. Thus, it becomes necessary to store a great body of data, and the efficiency of the compression decreases. Therefore, some additional methods, for example, signal quantization and Hoffman coding,[4] can be used in this case to improve the compression. However, they improve compression only insignificantly, but strongly increase the operational time of the algorithm.

2.2. Mathematical methods of data compression

SVD transformation of data

Singular value decomposition (SVD) is a well-known technique of orthogonal decomposition of data sets.[6] The purpose of the algorithm is to find orthogonal matrices U and V so that the matrix

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

be diagonal. Both these matrices are products of orthogonal matrices referred to as Householder's reflections; T denotes a transposition.

The scheme of decomposition of an arbitrary $m \times n$ matrix A can be used for compression of arbitrary data sets.[7] Let the initial matrix be represented in the form of a product of three matrices:

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

where the orthonormal matrix U has the size $m \times n$; Σ is a diagonal $m \times n$ matrix; and V is an orthonormal $m \times n$ matrix. The diagonal of Σ consists of singular values of the matrix A, usually, in the decreasing order. We are interesting in the situation where most singular values are low. Assuming that only L largest singular values remain, we can rewrite Eq. (4) as

$$a_{ij} = \left(\sum_{k=1}^{L} u_{ik}\,\sigma_k\,v_{jk}\right) + \left(\sum_{k=L+1}^{n} u_{ik}\,\sigma_k\,v_{jk}\right), \qquad (5)$$

where σ_k are the singular values (diagonal elements of the matrix Σ). If, actually, only first L singular values are significant, then the matrix A can be approximated by rejection of the second term in Eq. (5). This efficiently decreases the size of the decomposition matrices, so that the matrix U acquires the size $m \times L$,

Σ becomes an $L \times L$ matrix, and V is $n \times L$. Thus, the accuracy of data representation is determined by L. In practice, the number of eigenvectors $\{u_j\}$ often can be decreased several times (sometimes, even tens times) with only several percent loss in accuracy of reconstruction of the initial values.

Using the SVD approach, the matrix of monochromatic absorption coefficients A can be stored as two matrices $\hat{A}$ and U:

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

The memory needed for these two matrices is tens times less than that needed for the initial matrix A. To obtain the initial matrix, one has simply to multiply $\hat{A}$ by U, i.e.,

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

The matrix $\bar{A}$ obtained by Eq. (6) differs from the initial matrix A within a given error, which, in its turn, is determined by truncation of the SVD matrices.

The singular value decomposition method has demonstrated very good results in both the accuracy and the efficiency of compression. Sufficiently high speed of operation makes it one of the preferable compression methods.

Karhunen–Loeve transformation

Factor analysis of the data matrix is the first stage in the Karhunen–Loeve transformation.[8–10] The factor analysis operates with the data matrix of k rows and i columns. Each row is a normalized spectrum. Each column corresponds to a specific frequency in the IR region. The aim of the factor analysis is to decrease the size of the matrix from $k \times i$ to $k \times j$, where $j < i$, and the new matrix describes the initial data with a given accuracy.[11]

The second stage in the Karhunen–Loeve transformation includes linear mapping of the initial matrix into some optimized coordinate system with the following conversion:

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

where $E_{i,n}$ is the ith component of the nth eigenvector; $D_{k,j}$ is the ith component of the kth row of the matrix (i.e., kth spectrum in a library); $T_{k,j}$ is the jth component of the kth row of the converted matrix, where $j < i$. The linear mapping is equivalent to determination of the projection of the vector of each spectrum to every axis in the new coordinate system. The projection D_k onto the most significant eigenvector becomes the first value in the representation of the converted vector. The following values are calculated for all j eigenvectors used for description of the new coordinate system. Thus, the initial i-dimension vector representation of the spectrum is linearly converted into a j-dimension vector, and since $j < i$, the data are compressed.

The Karhunen–Loeve method was experimentally tested by Hangac et al.[11] The tests showed that this method compressed IR spectra of the ether and some other compounds five times with an accessible error.

Approximation by cubic polynomial

Procedures of approximation by cubic polynomials can become another approach to the data compression. Since the pressure and temperature dependences of the absorption coefficient are rather smooth, application of these procedures can give good compression with low error.

The Chebyshev approximation can be taken as a procedure of approximation. The essence of this method is in determination of N coefficients of a polynomial for some table function. These coefficients then can be used in calculation of the initial function at an arbitrary point. For fixed N, the equation $f(x) \approx \left[\sum_{k=1}^{N} c_k T_{k-1}(x) \right] - \frac{1}{2} c_0$ is a polynomial in terms of x that approximate the function $f(x)$ in the interval $[-1, 1]$. An advantage of this polynomial is that in its truncation to a *lower* degree $m \ll N$ the Chebyshev polynomial gives the best approximation for the degree m as compared to other polynomial schemes. Let N be large enough to provide the accurate approximation of $f(x)$. Then the truncated approximation has the form $f(x) \approx \left[\sum_{k=1}^{m} c_k T_{k-1}(x) \right] - \frac{1}{2} c_0$ with the same c_j. Since $T_k(x)$ is limited within ± 1, the difference between the accurate and truncated approximations does not exceed the sum of rejected c_k, $k = m + 1, \ldots, N$.

Four coefficients are needed for a cubic polynomial. This is enough to reconstruct the initial vector with high accuracy. The experiments conducted have shown that the average error of approximation by a cubic polynomial does not exceed 0.04% with the maximum error less than 0.1%. The degree of compression depends on the number of elements in the vector to be compressed. Thus, for example, some vector of 10 elements will be compressed by the factor 2.5.

2.3. Statistical methods of data compression

LZW compression method

Lempel-Ziv-Welch (LZW) compression is a well-known widely used technique. This method forms the basis of such popular archivers as PKZIP, LHA, and ARJ. This method has a very simple algorithm. LZW compression replaces character strings with some codes without any analysis of the input text. Compression occurs when the code replaces the character string. The codes generated by the LZW algorithm can be of any length, but they must contain more bits than the unit symbol. By default, the first 256 codes (when 8-bit symbols are used) correspond to the standard set of

symbols. Other codes correspond to strings processed by the algorithm.[12]

It is rather hard to characterize the efficiency of some compression technique. The degree of compression is determined by different factors. The LZW compression stands out of other techniques, when it meets a data flow containing repeated strings of any structure. Therefore, it operates very efficiently with text data. The degree of compression can achieve 50% and higher.

However, some difficulties can arise in the case of compression of data files. Depending on the initial data, the result of compression may be both high and low.

Hoffman method

The Hoffman method is a statistical compression method that decreases the mean length of a code word for alphabet symbols. The Hoffman code is an example of the optimum code in the case that all probabilities of appearance of symbols in a message are integer negative powers of two. The Hoffman code can be constructed by the following algorithm.

1. All alphabet symbols are written out in a series in the increasing or decreasing order according to their probability of appearance in a text;

2. Two symbols with the lowest probabilities are sequentially united in a new compound symbol, whose probability is assumed to be equal to the sum of probabilities of the constituents; finally, we have a tree, whose every node has the probability equal to the sum of probabilities of all lower nodes;

3. The way to every leaf is constructed by marking, for example, the right direction by 1 and the left direction by 0.

Some given distribution of frequencies can be characterized by several Hoffman codes. The "canonical" Hoffman tree can be determined by choosing one of possible trees. This canonical tree can be very compact, carrying only the length in bits of every code word. This method is used in most archivers.

Arithmetic coding

This method is based on the idea of transformation of the input flow into one floating-point number. Naturally, the longer a message, the longer the resulting number. At the output of an arithmetic compressor, we have some number less than 1 and larger than or equal to 0. The initial series of symbols can be unambiguously reconstructed from this number.[13]

Experiments at different levels show that the arithmetic coding always gives results no worse than those of the Hoffman coding. In some cases, the gain can be significant. However, the arithmetic coding algorithm is far more time-consuming comparative to the Hoffman one because of a greater number of the necessary computations. The arithmetic coding can be used in the cases that the degree of compression is more important than the time needed for compression.

3. Conclusions

Now there exist a wide spectrum of various archivers and utilities of data compression, but the main decision criteria are, first, the decompression speed and, second, the possibility to retrieve an arbitrary fragment of the initial data set from the compressed file without its complete decompression. Most of the available algorithms do not meet these conditions. Therefore, the attention was paid to various algorithms of data decomposition and transformation based on mathematical compression and spectral analysis. These algorithms are singular value decomposition, discrete wavelet transformation, and Karhunen–Loeve transformation. All these algorithms are very fast and capable of operating with any fragment of the compressed data. Besides, since the initial data are initially generated in the text form, the compression algorithm is, in some sense, a simple conversion into the binary data format. This additionally decreases the size of the stored information without loss in accuracy.

To test the efficiency of different compression methods, a number of experiments were conducted. The test results are given in the Table and in Figs. 1 and 2.

As is seen from the Table, the SVD compression demonstrates very good results in the accuracy and speed. It gives fivefold compression and passes ahead of the popular archivers, simultaneously providing the accuracy comparable with that given by modern methods of line-by-line calculation of absorption coefficients. Unfortunately, the wavelet transformation, in spite of a high speed, gives low degree of compression and high error of reconstruction because of a small number of elements in the vectors to be compressed.

Table. Comparison of data compression methods

Compressor	Decompression time, s	Degree of compression, times	Mean error, %	Maximum error, %
SVD	0.44	5	0.001	Less than 0.05
Wavelet	0.28	1.38	10^{-5} – 0.7	0.008 – 2.94
RAR	0.47	3.25	0	0
ZIP	0.21	3.25	0	0
ARJ	0.23	3.16	0	0

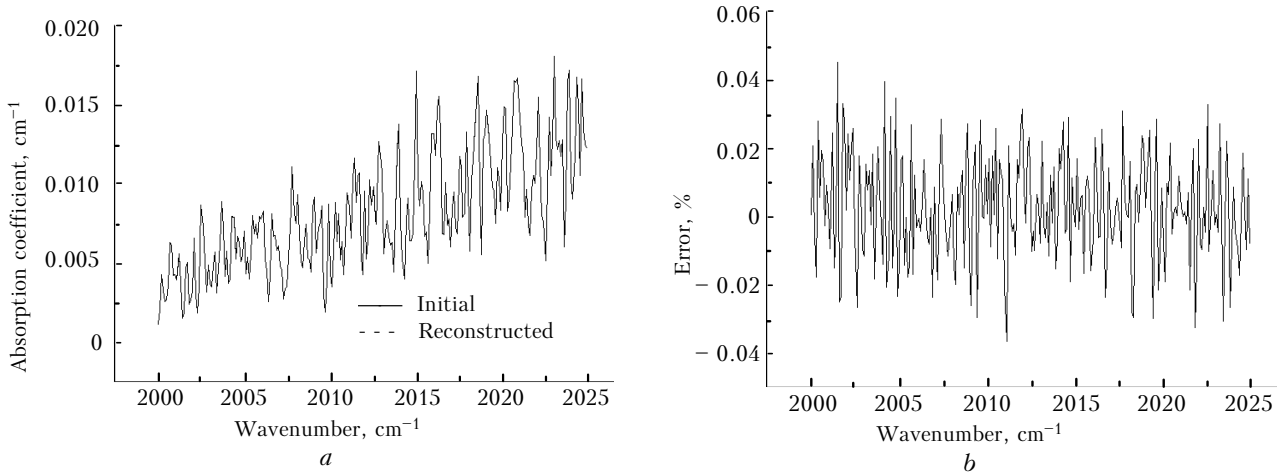

Fig. 1. Initial and reconstructed absorption coefficients at SVD compression (*a*); errors of reconstruction of absorption coefficients at SVD compression (*b*); gas – ozone; height H = 0 km; pressure of 0.91 atm; T = 278.2 K.

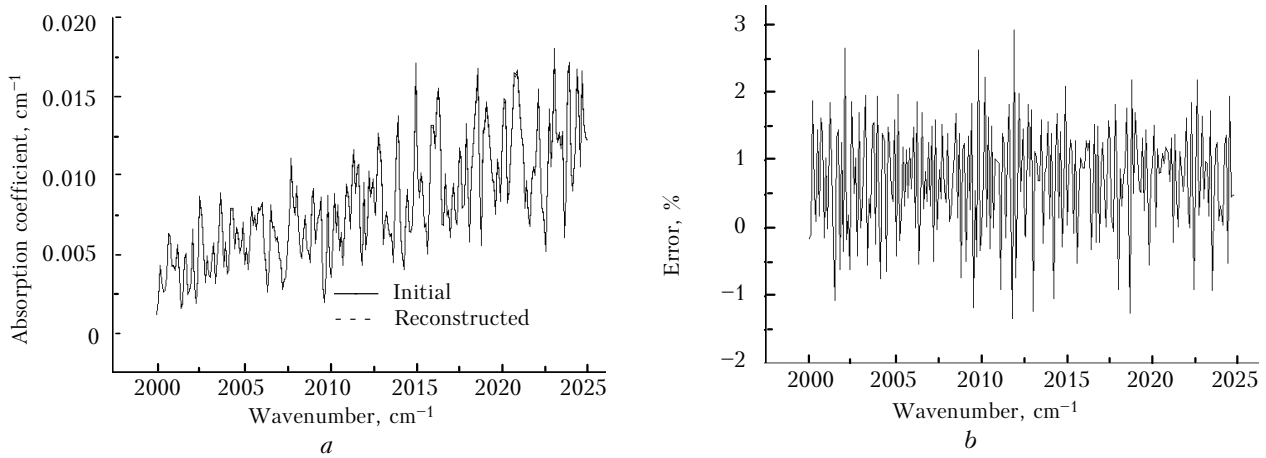

Fig. 2. Initial and reconstructed absorption coefficients at DWT compression (*a*); errors of reconstruction of absorption coefficients at DWT compression (*b*); gas – ozone; height H = 0 km; pressure of 0.91 atm; T = 278.2 K.

As compared to SVD, the archivers provide a lower degree of compression, although they introduce no errors. In the general case, they cannot be used for compression of tables, since they do not allow retrieving arbitrary data from archivers, but they can be used for extra compression or in the case that no error is required.

Acknowledgments

This work was partially supported by the Russian Foundation for Basic Research (Grant No. 00–07–90175).

References

1. D.S. Turner, J. Quant. Spectrosc. Radiat. Transfer **53**, No. 6, 633–637 (1995).
2. L.L. Strow, H.E. Motteler, R.G. Beuson, S.E. Hannon, and S. de Souzo-Mackodo, J. Quant. Spectrosc. Radiat. Transfer **59**, 481–493 (1998).
3. "*Development of mathematical methods and algorithms for pre-processing and coding of complex images*," Technical report (Moscow, 1976), pp. 88, 118–127.
4. F.T. Chau, J.B. Gao, T.M. Shih, and J. Wang, Appl. Spectrosc. **51**, No. 5, 649–659 (1997).
5. F.T. Chau, T.M. Shih, J.B. Gao, and C.K. Chan, Appl. Spectrosc. **50**, No. 3, 649–659 (1996).
6. J. Forsaite, M. Malcolm, and K. Mowler, *Computer Methods of Mathematical Calculations* [Russian translation] (Mir, Moscow, 1980), 280 pp.
7. P.B. Harrington and T.L. Isenhour, Appl. Spectrosc. **41**, No. 3, 449–453 (1987).
8. P.T. Funke, E.R. Malinowski, D.E. Martire, and L.Z. Pollara, Sep. Sci. **1**, 661 (1966).
9. D.G. Howery, Am. Lab. **8**, 14 (1976).
10. R.J. Rummel, *Applied Factor Analysis* (Northwest Univ. Press, Evanston III, 1970).
11. G. Hangac, R.C. Weiboldt, R.B. Lam, and T.L. Isenhour, Appl. Spectrosc. **36**, No. 1, 41–47 (1982).
12. R. Rucker, *Mind Tools* (Houghton Miffilin Company, Boston, Mass., 1987), 234 pp.
13. I.H. Witten, R.M. Neal, and J.G. Clearly, Communications of the ACM **30**, No. 6, 16–21 (1987).