# Optimization of computational complexity of lossy compression algorithms for hyperspectral images

**L I Lebedev**[1], **A O Shakhlan**[2]

[1]Lobachevsky State University of Nizhny Novgorod, Ulyanov st., 10, Nizhny Novgorod, Russia, 603005
[2]Bauman Moscow State Technical University, 2nd Baumanskaya st., 5/1, Moscow, Russia, 105005

e-mail: lebedev@pmk.unn.ru

**Abstract.** In this paper, we consider the solution of the problem of increasing the speed of the algorithm for hyperspectral images (HSI) compression, based on recognition methods. Two methods are proposed to reduce the computational complexity of a lossy compression algorithm. The first method is based on the use of compression results obtained with other parameters, including those of the recognition method. The second method is based on adaptive partitioning of hyperspectral image pixels into clusters and calculating the estimates of similarity only with the templates of one of the subsets. Theoretical and practical estimates of the increase in the speed of the compression algorithm are obtained.

## 1. Introduction

The main idea of compression methods using recognition methods is to replace the original description of a pixel (signature) $\mathbf{y} = (y_1, y_2, ..., y_n)$ in a constant length $\Upsilon$ record with the values of parameters obtained during recognition. On average, as a rule, there are quite many pixels that are similar to $\mathbf{y}$, and the description $\mathbf{y}$ is much larger than the description $\Upsilon$, which ensures the efficiency of the HSI compression. We shall consider the methods of comparison with templates as the main recognition methods. Recognition based on comparison with templates consists in establishing the similarity (proximity) of a certain pixel $\mathbf{y}$ to the equivalence class $K(\mathbf{y}^{\mathbf{e}}, A)$ generated by the template $\mathbf{y}^{\mathbf{e}}$ and given by the transformation operator $A$. As an estimate of the proximity $\varepsilon_m(A)$ of the pixel $\mathbf{y}$ to the template $\mathbf{y}^{\mathbf{e}}$, the minimum mean square error $\varepsilon(A)$ (NSE) is used with respect to the parameters of the operator $A$:

$$\varepsilon_m(A) = \min_A \varepsilon(A) = \frac{1}{n} \cdot \left\| \mathbf{y} - A(\mathbf{y}^{\mathbf{e}}) \right\|^2 = \frac{1}{n} \cdot \sum_{i=1}^{n} \left( y_i - A(y_i^{\mathrm{e}}) \right)^2 . \qquad (1)$$

The HSI compression algorithm is based on recognition by the methods of comparison with templates. This type of recognition is used in the self-learning mode, when a set of templates is formed on the basis of unrecognized signatures. With different transformation operators $A$, it is possible to obtain several criteria for similarity estimates by solving the optimization problem (1). On their basis, several recognition methods will be obtained.

## 2. Criteria for assessing the proximity of signatures

Let us consider some criteria for calculating similarity estimates:

$$\begin{aligned}
&A(\mathbf{y^e}) = \mathbf{y^e} : \quad \varepsilon_m^{nt} = \mathrm{M}(\mathbf{y^e})^2 + \mathrm{M}\,\mathbf{y}^2 - 2\cdot\mathrm{M}\,\mathbf{y^e}\mathbf{y} \;;\\
&A(\mathbf{y^e}) = \mathbf{y^e} + \Delta y^{\mathrm{e}} : \quad \varepsilon_m^{ot} = \mathrm{D}\,\mathbf{y^e} + \mathrm{D}\,\mathbf{y} - 2\cdot\mathrm{cov}(\mathbf{y^e},\mathbf{y}) \;, \quad \Delta y^{\mathrm{e}} = \mathrm{M}\,\mathbf{y} - \mathrm{M}\,\mathbf{y^e} \;;\\
&A(\mathbf{y^e}) = k_m\cdot\mathbf{y^e} : \quad \varepsilon_m^{st} = \mathrm{M}\,\mathbf{y}^2 - \mathrm{M}^2(\mathbf{y^e}\mathbf{y})\big/\mathrm{M}(\mathbf{y^e})^2 \;, \quad k_m = \mathrm{M}(\mathbf{y^e}\mathbf{y})\big/\mathrm{M}(\mathbf{y^e})^2 \;;\\
&A(\mathbf{y^e}) = k_m\cdot\mathbf{y^e} + \Delta y^{\mathrm{e}} : \quad \varepsilon_m^{at} = \mathrm{D}\,\mathbf{y} - \mathrm{cov}^2(\mathbf{y^e},\mathbf{y})\big/\mathrm{D}\,\mathbf{y^e} \;, \quad k_m = \mathrm{cov}(\mathbf{y^e},\mathbf{y})\big/\mathrm{D}\,\mathbf{y^e} \;, \quad \Delta y^{\mathrm{e}} = \mathrm{M}\,\mathbf{y} - k_m\cdot\mathrm{M}\,\mathbf{y^e} \;.
\end{aligned} \tag{2}$$

In the above formulas, $\mathrm{M}\mathbf{y}$ , $\mathrm{M}\mathbf{y^e}, \mathrm{M}\mathbf{y}^2, \mathrm{M}(\mathbf{y^e})^2$ denote the initial moments of the first and second orders of the current signature and template; $\mathrm{M}\mathbf{y^e}\mathbf{y}$ is the mixed covariance moment; $\mathrm{D}\mathbf{y}$ , $\mathrm{D}\mathbf{y^e}$ denote the dispersion of the signature and the template; $\mathrm{cov}(\mathbf{y^e},\mathbf{y})$ is the correlation moment; $\Delta y^{\mathrm{e}}$ , $k_m$ are the bias and the scale factor [1-3].

## 3. Recognition algorithm with self-learning

Based on the proximity estimates (2), in accordance with the recognition algorithm, for each template pixel $\mathbf{y^e}$ and the current signature $\mathbf{y}$ , the value $\varepsilon_m(\mathbf{y},\mathbf{y^e})$ is calculated; among these values, the smallest one, $\varepsilon_{\min}(\mathbf{y})$ is found. If this estimate is greater than a given threshold $\varepsilon_{\min}(\mathbf{y}) > \delta$ , then a new template is formed on the basis of the signature description $\mathbf{y}$ . When $\varepsilon_{\min}(\mathbf{y}) < \delta$ , the pixel is considered to be recognized and using the parameters obtained, a corresponding record in the output file is made. For all the similarity estimates (2), a template number must be assigned, which is recorded in a 2-byte field (or a 4-byte field if the number of templates exceeds 65536). The values $\Delta y^{\mathrm{e}}$ and $k_m$ are given by real variables, which are recorded in 4-byte fields. Therefore, the length of the structures for writing the parameters depending on the similarity estimate used will be equal, respectively, to 2, 6, 6 and 10 bytes (or 4, 8, 8 and 12 bytes)..

## 4. Optimization of computations in the HSI compression algorithm

Let us consider two possibilities for increasing the speed of the compression algorithm under consideration. It follows from the recognition algorithm that the number of obtained templates depends on the threshold value $\delta$, which in turn is determined by the error limited to a certain percentage $\sigma$ of the current signature. Then the recognition and, therefore, the compression will be performed using an adaptive threshold $\delta = (0.01\cdot\sigma)^2\cdot\mathrm{M}\mathbf{y}^2$. From the analysis of formula (2), it follows that when compressing a HSI with a similarity estimate $\varepsilon_m^{at}$, the number of templates will be the smallest among all compression algorithms using the remaining proximity estimates. In turn, compression algorithms based on proximity estimates $\varepsilon_m^{st}$ and $\varepsilon_m^{ot}$ have some advantages in terms of speed over the algorithm using the proximity estimate $\varepsilon_m^{nt}$. Similarly, the greater the value $\sigma$, the less templates will be obtained in the recognition process and thus the shorter the HSI compression time. Therefore, in order to reduce compression time required by more time-consuming algorithms, it is necessary to use the results of compression provided by faster compression algorithms. The modification of the compression algorithm in this variant consists in calculating the estimate of similarity only with the templates generated by the template of the high-speed algorithm. The second modification of the compression algorithm is based on the current clustering of signatures being recognized and forming own templates in each cluster. Therefore, if the current signature is assigned to a cluster, the similarity estimate will be formed on the basis of the templates of this cluster. The clustering is carried out in the space of the features, which can be easily obtained by calculating the initial moments of the first and second orders of the current pixels.

Let us estimate the computational complexity of recognition methods based on the proposed similarity estimates. Let the function describing the dependence of the number of templates used on the current number of the pixel being recognized be given by the formula $v = \left[\, f(u) \right]$, with

$m = f(N)$, $f(0) = 0$, where $N$ is the number of pixels of the image, and $m$ is the number of templates obtained during image compression. The function $v = \left[ f(u) \right]$ is piecewise constant and, therefore, an increase in the number of templates per unit occurs during the compression of pixels with the numbers $u = N_i$. Hence, $f(N_i) - f(N_{i-1}) = 1$, and from this expression one can find the numbers of pixels $N_i$. Then the computational complexity of the HSI compression algorithm will be determined by the formula:

$$C = \sum_{i=1}^{m} i \cdot (N_i - N_{i-1}) \cdot O(n) \quad , \qquad (3)$$

where $O(n)$ is the complexity of obtaining the estimate of similarity of a pixel with the template. If we assume that $f(u)$ is a linear function of (L), then $C(L) = [N \cdot (m+1)/2] \cdot O(n)$, since $N_i - N_{i-1} = N/m$. However, as a rule, the number of templates increases faster at the initial stage of compression and slows down closer to the completion of the HSI pixel recognition process. If we take the parabola (P) $u = N \cdot v^2/m^2$ as a function approximating the plot of the number of templates depending on the number of pixels processed, then it follows from the expression $m \cdot \sqrt{N_i/N} - m \cdot \sqrt{N_{i-1}/N} = 1$ that $N_i - N_{i-1} = N/m^2 \cdot (2 \cdot i - 1)$. Hence, the computational complexity of the HSI compression algorithm in this case can be estimated by the value $C(P) = [N \cdot (m+1) \cdot (4 - 1/m)/6] \cdot O(n)$, and therefore, $C(P)/C(L) \approx 4/3$. In the case of the cubic dependence (K) $u = N \cdot v^3/m^3$, the computational complexity is equal to $C(K) = [N \cdot (m+1) \cdot (3 - 1/m)/4] \cdot O(n)$, and $C(K)/C(L) \approx 3/2$. For the proposed approximation functions, the computational complexity of the recognition algorithm is proportional to $N(m+1)$ and, therefore, if the number of templates is $M > m$, then while maintaining the same distribution law, the HSI compression time will increase by a factor of $M/m$. Note that an increase in the HSI compression time does not depend on the distribution law.

## 5. Experimental research

Experiments on solving recognition problems were performed on the f100520t01p00-12 HSI file of the AVIRIS spectrometer based on 224 frequencies with wavelengths in the range from 365.9298 to 2496.236 nm. The original HSI was represented by a matrix of 813×2109 pixels with a resolution of 17.3 m. A two-byte format for representing the spectrum amplitude values was used. At the beginning, filtering (correction of values) of spectral images of the original HSI was performed. Its purpose was to replace the negative values of the signature with the data obtained by piecewise linear approximation with the closest positive values. Signatures with completely negative or positive component values were not adjusted. The number of pixels with completely negative values was 207055 of the total number 1714617 (≈12%).

By using compression with an adaptive threshold, calculated for the value $\sigma = 2$ for the similarity estimates listed in (2), we obtained 2582, 1841, 976 and 785 templates, respectively. Since it can be expected that the computational complexity of compression is proportional to the number of templates obtained in the course of recognition, the running time of the algorithm may vary up to 3.3 times when using different similarity estimates. Figure 1 shows the graphs of the number of templates used depending on the current number of a HSI pixel being recognized for various similarity estimates. Figure 2 shows the graphs of various functions a) - c) approximating the dependence of the number of templates used on the current number d). Figure 3 shows a histogram of the number of signatures recognized by each template when using the estimate $\varepsilon_m^{st}$. It can be concluded from these graphs that the bulk of the pixels is encoded on the basis of the first third of the templates. However, the remaining pixels will use in the coding almost the entire set of templates, which leads to an increase in the running time of the algorithm. When using a modification of the compression algorithm utilizing the results obtained in recognition with the estimate $\varepsilon_m^{st}$, each signature will be compared on average with only three templates.

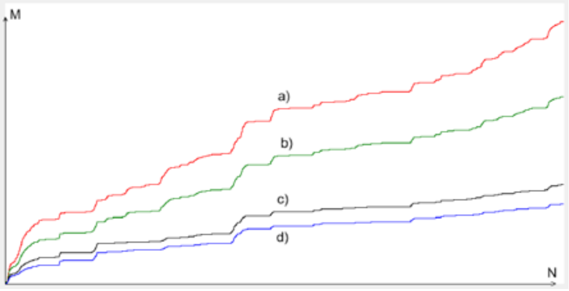

**Figure 1.** The number of templates used vs. the current number of the HSI pixel when compressed with estimates: a) $-\varepsilon_m^{nt}$; b) $-\varepsilon_m^{ot}$; c) $-\varepsilon_m^{st}$; d) $-\varepsilon_m^{at}$.

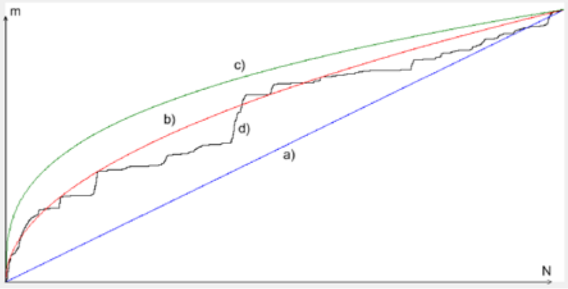

**Figure 2.** Approximation of the initial dependence $m = f(N)$ obtained for compression with the estimate $\varepsilon_m^{st}$ - plot d), functions: a) – linear; b) $m = k \cdot \sqrt{N}$; c) $m = k \cdot \sqrt[3]{N}$.

As a result, compression with the estimate $\varepsilon_m^{nt}$ was obtained tens of times faster than with the use of the original algorithm. The use of a modified algorithm for compressing HSI using the same estimate $\varepsilon_m^{st}$ with the parameter $\sigma = 1$ was even more efficient since the number of templates in this case increases to 12232.

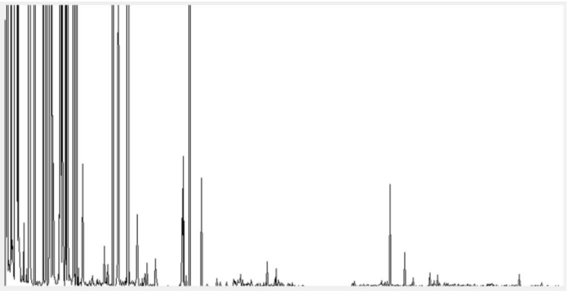

**Figure 3.** Histogram of the number of recognized HSI pixels vs. the template number.

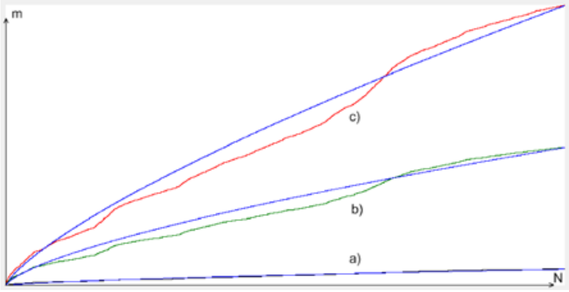

**Figure 4.** The number of templates vs. the pixel number for different thresholds: a) $\sigma = 10$; b) $\sigma = 6$; c) $\sigma = 5$ and their approximation by the functions $m = k \cdot N^{\mu}$.

The second modification was based on identifying clusters in the space of features, which were obtained on the basis of groups of pixels. There are many examples of the application of clustering and segmentation methods for solving practical problems [4-6]. However, for the purpose of modification, clustering algorithms must have computational complexity that is an order of magnitude smaller than that of the original compression methods. Therefore, to solve the clustering problem, one can use the original compression algorithms for large values of the parameter $\sigma$, which will provide the necessary speed. As a result, for $\sigma = 5$, about three dozen clusters with different sets of templates were obtained to get the similarity estimate $\varepsilon_m^{at}$ for the given hyperspectral image, thus making it possible to reduce by an order of magnitude the execution time of the compression algorithm. Let us consider the efficiency of solving the problem of computational complexity optimization by the proposed compression algorithms for more variable data, both in terms of the representation of various types of underlying surfaces on the HSI, and the values of spectral scatter. This MoffettField HSI of the same AVIRIS spectrometer was represented by a matrix of 753 × 1924 pixels with a resolution of 17.0 m. 68335 out of 1448772 pixels represent fragments that have not been covered by the scan band. The templates obtained by compressing the HSI for $\sigma = 10$ with the similarity estimate $\varepsilon_m^{st}$ were used as cluster representatives. As a result, 764 templates were obtained within the time ≈ 10 minutes. The computational complexity was 675863990 operations for computing similarity estimates. For $\sigma = 5$, the number of operations amounted to 10300082029 and the original compression algorithm produced

12891 templates for the time ≈ 160 minutes. The execution time of the proposed compression algorithm for $\sigma = 5$ with the account of the time for obtaining clusters was ≈ 13 minutes, which corresponds to theoretical estimates. Figure 4 shows the plots of the number of templates used to obtain the same similarity estimate depending on the current HSI pixel number for various compression thresholds and their approximation by analytical functions.

## 6. Conclusion

Experiments performed on hyperspectral images have confirmed the estimates of the expected increase in the speed of the modified recognition algorithms based on self-learning recognition. It was established that the speed of the modified compression algorithms is more than an order of magnitude higher than their original counterparts. A further increase in speed can be achieved by parallelizing the computations at each step of the two-step compression algorithm proposed.

## 7. References

[1] Lebedev L I 2018 Recognition and classification of objects in a hyperspectral image *Proceedings of the 5th International Scientific Conference «Regional problems of remote sensing of the earth»* (Krasnoyarsk: Siberian Federal University) 138-143

[2] Lebedev L I and Shakhlan A O 2018 Spatial analysis of hyperspectral images *Proceedings of the 25th International Scientific Conference* (Tomsk: Tomsk Polytechnic University) 150-153

[3] Lebedev L I 2013 *Correlation - extreme contour recognition methods. Theoretical foundations: A study manual* (Nizhny Novgorod: Nizhny Novgorod State University) p 113

[4] Tatarnikov V V, Pestunov I A and Berikov V B 2017 Centroid averaging algorithm for a clustering ensemble *Computer Optics* **41(5)** 712-718 DOI: 10.18287/2412-6179-2017-41-5-712-718

[5] Myasnikov E V 2017 Hyperspectral image segmentation using dimensionality reduction and classical segmentation approaches *Computer Optics* **41(4)** 564-572 DOI: 10.18287/2412-179-2017-41-4-564-572

[6] Varlamova A A, Denisova A Y and Sergeev V V 2018 Earth remote sensing data processing technology for obtaining vegetation types maps *Computer Optics* **42(5)** 864-876 DOI: 10.18287/2412-6179-2018-42-5-864-876

## Acknowledgements

This work was supported by the Russian Science Foundation, project No. 16-11-00068.