Computational Geosciences (2019) 23:969–979
https://doi.org/10.1007/s10596-019-09855-2

ORIGINAL PAPER

Hybrid hyperspectral image compression technique for non-iterative factorized tensor decomposition and principal component analysis: application for NASA's AVIRIS data

Jeyakumar S[1] · **Sudha S**[2]

Received: 9 July 2018 / Accepted: 24 June 2019 / Published online: 20 July 2019

Abstract

Remote sensing data has shown tremendous potential for applications in various fields like land use mapping and detection, geologic mapping, water resource applications, wetland mapping, urban and regional planning, environment inventory, natural disaster assessment, archaeological applications, and others. Every day, thousands of gigabytes of memory are involved in capturing the hyperspectral remote sensing datasets. The compelling information present in these hyperspectral images (HSIs) is very minimal due to redundancy. Spatial and spectral correlations monopolize the acquired HSI data sets. Therefore, an algorithm that exploits these correlations and compresses the HSI tensors is proposed in this paper. First, the acquired HSI image (reflectance data) is subjected to the removal of geometric and radiometric errors. Second, spectral bands of interest affiliated to the underlying application are exclusively processed for principal component analysis (PCA). Results of this PCA are scrutinized to identify the absolute dependent components. Further, these components are exposed to a non-iterative factorized compression technique. As a result, HSI 3D tensors are disintegrated into 1D tensors. This tensor breakdown leads to a compression ratio as high as 3747:1 while the total encoding time observed is 332 s and RMSE is as low as 0.0017. Later, the original HSI is reconstructed back by the product of decomposed individual tensors and its PSNR is 53.03 dB. The proposed compression method targets the tucker decomposition-based HSI compression technique which is computationally complex and time consuming, and hence, a breakthrough is achieved with the technique introduced.

Keywords HSI · PCA · 3D · PSNR · Wavelet · RMSE

1 Introduction

Hyperspectral image (HSI) applications are widespread but extensively employed in remote sensing. For example, vegetation indices derived from remote sensing imagery are used to quantify crop growth and yield variations as in [1], land-cover types classification from features extracted from hyperspectral image pixels in [2], robust mineralogical investigation over extraterrestrial bodies in hyperspectral datasets from Mars and Moon as in [3], poultry fecal detection from multispectral images in [4], and discrimination and recognition of sugarcane plants infected with mosaic virus through aerial imaging as in [5] were all innovations in the field of computational geoscience and also involve tremendous processing and storage resources for their operation.

The HSIs are frequently captured by special imaging sensors of aircrafts [6]. A team of NASA scientists is using a high-altitude aircraft and a sophisticated imaging spectrometer built by NASA's Jet Propulsion Laboratory in Pasadena, CA, to study environmental impacts called AVIRIS. Even NASA boasts utilizing 76 GB of data over few flight runs per day [7, 8]. Therefore, an obligation to compress these data sets has been raised. First, compression is achieved by filtering out the redundancies in both spatial and spectral resolutions [9] and [10]. The size of the samples is also shortened by eliminating the spatial redundancy along their adjacent spectral bands [11–14]. However, this

✉ Jeyakumar S
jeyakumars1989@gmail.com

Sudha S
sudha.s@srmeaswari.ac.in

[1] Research Scholar, Anna University, Chennai, India

[2] Professor Department of ECE, SRM Easwari Engineering College, Chennai, India

dimensionality reduction results in either lossy or lossless compression. Compression by dimensionality reduction is also proposed, but their methods hardly bought a considerable difference [15]. Independent component analysis (ICA) is a preferred technique in [16] and [17], but it has no legacy in considering the spatial correlation in between the bands. SPIHT and EZW algorithm are also used in [18, 19] and [20], which takes into concern only the relationship between spatial and spectral correlation. Hybrid algorithms employing PCA and JPEG2000 are also selected to compress HSIs in [21, 22], which utilizes transforms to analyze the spectral details and eliminate their redundancies. Singular value decomposition (SVD) is the most employed technique to decompose 3D data sets into 2D matrices in spite of it exploiting greater computation resource due to its iterative behavior [23]. Tucker decomposition, another compression technique based on SVD, achieves near lossless compression with the help of rank matrices, but it has relentless computational complexity [24–27]. Its encoding time measured is 94 min. Thus, a prominent near-lossless compression technique devoid of processing complexity is necessary and one such technique is proposed in this paper which consumes 6 min on an i3-5005U processor clocked at 2.0 GHz.

The rest of the paper is organized as follows. Section 1 briefs about the prime objective of the proposed algorithm. Sections 2.1, 2.2, and 2.3 elaborate about the different compression methods, discrete wavelet transform (DWT), PCA, and the proposed factorized decomposition technique. The process of compression is elaborated through steps 1 to 13 with necessary equations and diagrams in Section 3 (Fig. 1). Section 4 has the end results tabulated for performance analysis, and the final conclusion is portrayed in Section 5.

1.1 Objective of the proposal

The main objective is to develop a linear non-iterative compression algorithm that operates on the spectral bands of interest and utilizes principal component analysis to select the significant components for redundancy minimization.

2 Principal techniques

2.1 Wavelet transform

Wavelet transform is chosen because of its property to decompose signals into sub-bands of smaller bandwidths. The continuous wavelet transform (CWT) is defined as in Eq. 1:

$$\mathrm{CWT}(f(ab)) = \int f(t)\varphi_{a,b}(t)dt \tag{1}$$

where

$$\varphi_{a,b}(t) = (1 \div \sqrt{a})\varphi\left(\frac{t-b}{a}\right), a \in R^{+}, b \in R \tag{2}$$

Here, $\varphi_{a,b}(t)$ is the wavelet window function while a and b are scaling and location parameter resp. taking constant values e.g. 1, 2, etc. It is represented by

$$\mathrm{CWT}_{m,n}(f) = a_o^{-m/2} \int f(t)\varphi(a_o^{-m}t - nb_o)\mathrm{dt} \tag{3}$$

Since the application of wavelet transform is on digital images in our paper, we chose discrete wavelet transform (DWT). It is represented by Eqs. 4 and 5.

$$F(ab) = \int_{-\infty}^{\infty} f(x)\varphi^{*}_{(a,b)}(x)dx \tag{4}$$

Approximation coefficients and detail coefficients being $CA_1$and CD_1 resp. are computed while the input signal is represented by s.

$$f(n) = \frac{1}{\sqrt{M}}\sum\nolimits_k W_{\emptyset}(j_0,k)\emptyset_{j0,k}(n) + \frac{1}{\sqrt{M}}\sum\nolimits_{j=j_0}^{\infty}\sum\nolimits_k W_{\varphi} \tag{5}$$

$$W_{\emptyset}(j_0,k) = \frac{1}{\sqrt{M}}\sum\nolimits_n f(n)\emptyset_{j_0,k}(n) \tag{6}$$

$$W_{\varphi}(jk) = \frac{1}{\sqrt{M}}\sum\nolimits_n f(n)\varphi_{j,k}(n),\ j \geq j_0 \tag{7}$$

Equations 6 and 7 are approximation anddetail coefficients resp., and $f(n), \emptyset_{j_0,k}(n), \varphi_{j,k}(n)$are discrete wavelet functions [28, 29].

2.2 Principal component analysis

The fundamental idea of principal component analysis (PCA) in our paper is to reduce the dimensionality of a data set in z direction, which incorporates a number of interrelated variables, while retaining the deviations present in the data set. It is addressed by exhibiting linear orthogonal transformation that revamps the input data to a comprehensively new coordinate system, which in turn has maximal variances characterized in the first few coordinates of the signal [30–34].

Let x refer to the input signal containing n input vectors of length K,

$$x = [x_1, x_2, x_3 \ldots ..x_n]^T \tag{8}$$

then PCA refers to transforming $x$to vector y by the relation

$$y = A(x - m_x) \tag{9}$$

where m_x is the mean of all variables, defined by

$$m_x = \frac{1}{K}\sum\nolimits_{k=1}^{K} x_k = E\{x\} \tag{10}$$

and A is determined by the covariance matrix C_x of dimension $n \times n$.It is possible by the relation

$$C_x = E(x - m_x)(x - m_x)^T = \frac{1}{K}\sum_{k=1}^{K} x_k x_k^T - m_k m_k^T \tag{11}$$

The main diagonal elements of C_x, i.e.,$C_x(ii)$ are the variances or covariance of x

$$C_x(ii) = E(x_i - m_i)^2 \tag{12}$$

and $C_x(ij)$ are the cross covariance between inputs x_i and x_j

$$C_x(ij) = E(x_i - m_i)(x_j - m_j)\} \tag{13}$$

Further, to increase variance within the variables, matrix A can be diagonalized by choosing a new orthogonal coordinate system, i.e., by eigenvectors and eigenvalues through singular value decomposition [35].

2.3 Factorized compression

The novel compression strategy proposed refers to decomposing a $N \times N$ matrix into two $1 \times N$ matrices for the purpose of data transmission as well as storage.

Consider a matrix $x$of dimension 4×4.

Let $x_1, x_2, x_3 \ldots\ldots x_{16}$ be the appropriately scaled pixel values of the 2D image data.

Then, the multiplication of decomposed R_1, R_2, R_3, R_4row vectors and C_1, C_2, C_3, C_4 column vectors produces the original or near-estimated 4×4 image pixel matrix in Eq. 14.

$$x_{4x4} = \sum_{i=1}^{N}\sum_{j=1}^{N} C_i R_j \tag{14}$$

This algorithm is defined and developed with the working principles involved in least mean-square error estimations. Based on the analysis in [36] and [37], this strategy proved to ascertain better compression results and therefore was chosen as a primitive compression methodology in this paper.

3 Description of the process

3.1 Data set

A hyperspectral image data set is procured from [38]. The size of the reflectance files chosen was 1021 × 1339. The peak transmission wavelength varies between 400 and 720 nm, sampled at 10-nm intervals, comprising 33 adjacent spectral bands. Each set of scene is preprocessed for obtaining reflectance values. Each data set comprises 45,114,927 pixel values represented using double datatype (8 byte). Therefore, the total size of each sample data set is 344 MB.

Step 1: Preprocessing of data sets refers to removal of geometric and radiometric errors associated with the motion of the aircraft, correction of atmospheric effects, removal of dataless bands from reflectance data, etc. The before-mentioned

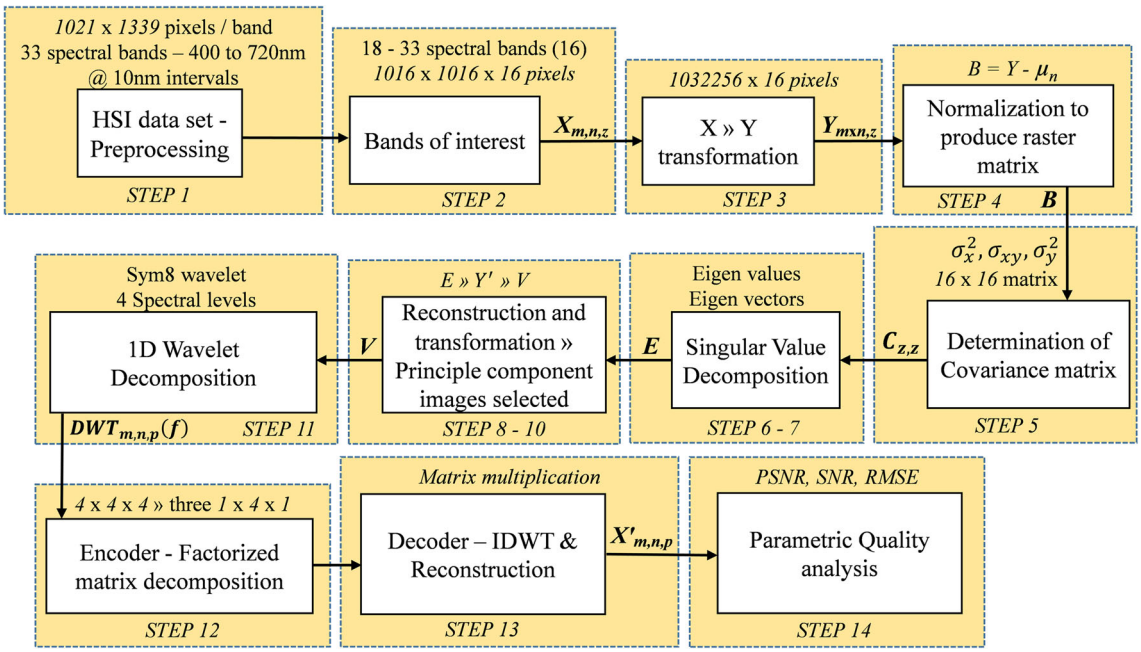


Fig. 1 Block diagram—conceptual work flow

result-driving factors are persuaded at the source [38].

Step 2: Let X be the input HSI cube of dimension 1016 × 1016 × 16 ($m \times n \times z$). In this paper, PCA is applied only on the desired bands of interest (BOI), i.e., 18 to 33 bands in spite of all 33 available spectral bands because of their high contrast and least noise disturbance property. Therefore, the size of the HSI cube is reduced to 126 MB. Figures 2 and 3 represent scene 1—pixels of band 21 and 33 respectively showing varying details. Scene 1 depicts the aerial view of an urban environment with a little vegetation and a lot of detailed information.

Step 3: The matrix cube X is transformed into Y whose dimensions are 1,032,256 × 16. Ref Fig. 4.

Step 4: Standardization or normalization of the input signal is performed on matrix $Y$to make its mean μ_Y and variance σ_Y^2 as 0 and 1 resp. In this paper, normalization is achieved by subtracting the empirical mean $\mu_n$from each column data of matrix Y. Let B be the normalized raster matrix,

$$B = \begin{bmatrix} x_{11} - \mu_1 & y_{11} - \mu_1 & \cdots & z_{11} - \mu_1 \\ x_{12} - \mu_2 & y_{12} - \mu_2 & \cdots & z_{12} - \mu_2 \\ x_{13} - \mu_3 & y_{13} - \mu_3 & \cdots & z_{13} - \mu_3 \\ \vdots & \vdots & \ddots & \vdots \\ x_{1n} - \mu_n & y_{1n} - \mu_n & \cdots & z_{1n} - \mu_n \\ \vdots & \vdots & \ddots & \vdots \\ x_{nn} - \mu_{nn} & y_{nn} - \mu_{nn} & \cdots & z_{nn} - \mu_{nn} \end{bmatrix} \tag{15}$$

Fig. 2 Pixels of band 21

Fig. 3 Pixels of band 33

Step 5: According to Eq. 16, covariance matrix $C_{z,z}$is calculated. $C_{z,z}$ is a real, symmetric, positive semi-definite square matrix. Its dimension, 16 × 16, is dependent on the number of spectral bands subjected to consideration in step 2, and the dimension of this covariance matrix determines the number of principal components produced in step 9.

$$C_{z,z} = \frac{1}{n} B B^T = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2n} \\ \sigma_{31} & \sigma_{32} & \cdots & \sigma_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_n^2 \end{pmatrix} \tag{16}$$

An ideal covariance matrix for PCA is a diagonal matrix $C\prime$, i.e., all cross covariance values in $C_{z,z}$is equal to zero (17). This fact ensures maximum variances among variables being represented in the vectors.

$$C' = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n^2 \end{pmatrix} \tag{17}$$

Step 6: singular-value decomposition is applied on the matrix $C_{z,z}$ [39] and [40]. According to Eq. 18, eigenvalues are calculated using the formulae $C_{z,z} - \lambda I = 0$ and its eigenvectors are determined by reducing the $C_{z,z} - \lambda I$ matrix. The roots of the characteristic polynomial of E will also yield

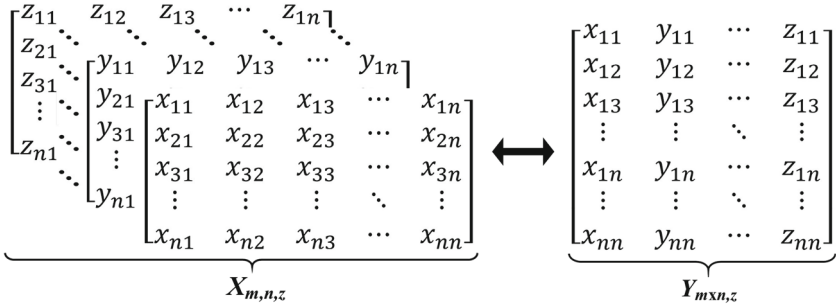


Fig. 4 Transformation of matrix X into raster matrix Y

the eigenvalues. The final matrix containing the eigenvectors is denoted by U.

$$E = \begin{vmatrix} \sigma_1^2 - \lambda & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 - \lambda & \cdots & \sigma_{2n} \\ \sigma_{31} & \sigma_{32} & \cdots & \sigma_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_n^2 - \lambda \end{vmatrix} \tag{18}$$

Step 7: Eigenvalues are arranged in the order of descending magnitude, and the values with very feeble magnitudes are neglected before reconstruction as it contributes less or no significance to the input signal. The process of reconstruction involves matrix multiplication of $Y$and U (19). The resultant matrix is of dimension 1,032,256 × 16, represented as Y' (20).

$$Y' = Y \times U \tag{19}$$

$$Y' = \begin{bmatrix} x_{11} & y_{11} & \cdots & z_{11} \\ x_{12} & y_{12} & \cdots & z_{12} \\ x_{13} & y_{13} & \cdots & z_{13} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1n} & y_{1n} & \cdots & z_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{nn} & y_{nn} & \cdots & z_{nn} \end{bmatrix} \text{x} \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \tag{20}$$

Step 8: Matrix Y' is transformed back into matrices of cubical dimensions 1016 × 1016 × 16, i.e., the input signal is projected into a new feature phase. However, in Y', many spectral bands do not have functional values, i.e., null images, because of the eigenvalue approximation in step 7. Such spectral components do not contain any significant information and hence can be neglected for analysis, storage, and transmission. Thus, the phenomenon of compression is introduced in this step through spectral component reduction.

Step 9: Based on our analysis on several data sets, if the no. of spectral bands z is >32, then the no. of principal components p that contains significant information is restricted to 8. For any other $z \leq 32$, the first four principal components carry the maximum variance, i.e., information. Therefore, the first four principal components are considered Fig. 5, instead of 16 in the HSI data set which reduces the data set size by 75%, i.e., to 31 MB. The compressed data set is represented by V. Those 12 principal components indeed look similar to Fig. 5d PCA 4 with no details to interpret or analyze.

Step 10: Since the data sets are of high resolution (4,129,024 pixel values), huge physical memory (over 8 GB) and high processing potential, i.e., a multicore processor with parallel processing ability [43], are required for faster implementation of the proposed compression algorithm in

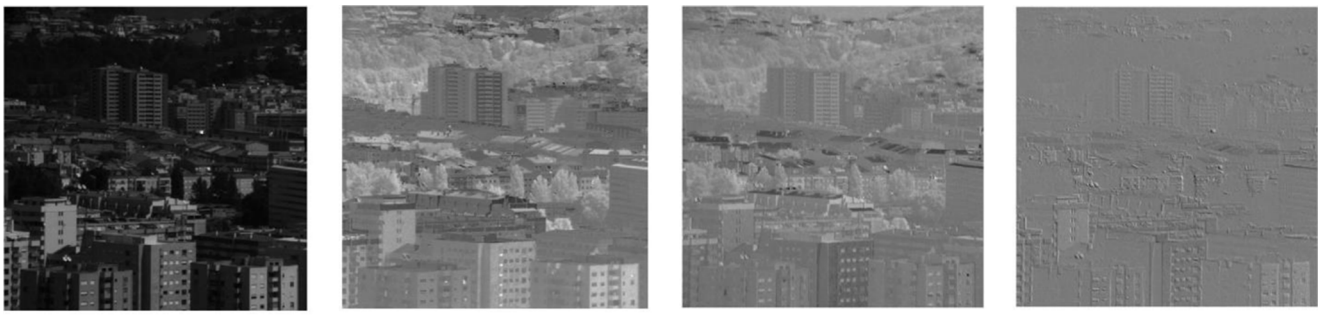
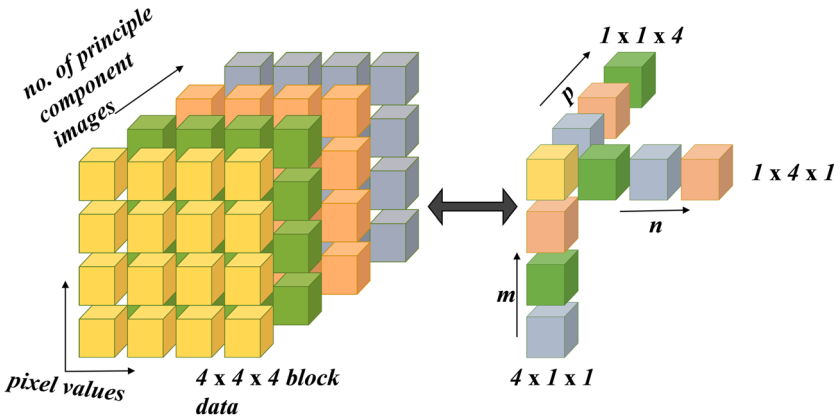


Fig. 5 Principal components—**a** PCA 1, **b** PCA 2, **c** PCA 3, and **d** PCA 4

Fig. 6 Tensor decomposition



Section 2.3. In addition, to reduce this processing complexity, the entire data set is divided into 4 × 4 × 4 cubical blocks and then formulized. This technique is called block processing [41, 42]. The PCA component data set is further compressed by using a combination of 1D discrete wavelet transform and factorized compression technique as elaborated in Sections 2.2 and 2.3 resp. First, matrix V is subjected to 1D symmetric wavelet transform. As a result, approximate $W_{\emptyset}\left(j_0, k\right)$ and detail coefficients $W_{\varphi}\left(j, k\right)$ are obtained.

Step 11: Results of step 10 are subjected to factorized decomposition which replaces all $n \times n \times n$ cubes with three $1 \times n \times 1$ row vector values while n can take values 4 (method 1) and 8 (method 2). Thus, if $n = 4$, Fig. 6 for every cubical data set, 12 values are transmitted or encoded instead of 64, and therefore, size is further reduced by 81%. Similarly, 24 values for $n = 8$ and, therefore, size is compressed by 95%. Now, the actual size that is required on the storage disk to back up or process these factorized values is just 94 KB and 1.4 MB resp. Therefore, compression ratio as high as 3747:1 and 245:1 resp. is obtained.

Step 12: At the decoder, multiplication of the individual tensors is performed without exploiting any bigger processing resource since it is mere matrix multiplication. This method refers to the reconstruction of the original data after experiencing near lossless compression. Scene 1 original and reconstructed images for $n = 4,8$ are displayed in Fig. 7.

Step 13: The accuracy of the reconstructed data set is analyzed with the help of several performance parameters like SNR, PSNR, RMSE, etc. Let $x(n)$ be the original HSI reflectance data and $y(n)$ be the estimated decompressed data.

$$\text{SNR} = 10log_{10}\left[\frac{\sum_{n=0}^{N}|x(n)|}{\frac{\sum_{n=0}^{N}[x(n)-y(n)]^2}{N}}\right]\text{dB} \quad (21)$$

$$\text{PSNR} = 10log_{10}\left[\frac{[\max(x(n))]^2}{\frac{\sum_{n=0}^{N}[x(n)-y(n)]^2}{N}}\right]\text{dB} \quad (22)$$

$$\text{RMSE} = \sqrt{\frac{\sum_{n=0}^{N}[x(n)-y(n)]^2}{N}} \quad (23)$$

(a) (b) (c)

Fig. 7 Scene 1—**a** original (band 21), **b** compressed RMSE = 0.0105 ($n = 4$), **c** compressed RMSE = 0.0168 ($n = 8$)

Table 1 Comparison of different method performances

Block size/compression factor	Samples	Scene 1	Scene 2	Scene 3	Scene 4
Method 1	Encoding time (s)	222.39	197.34	332.53	230.65
n =4	Decoding time (s)	0.94	0.83	1.23	1.07
4 × 4 × 4-sized blocks	PSNR (dB)	50.42	50.46	48.06	53.03
Total: 64516 blocks	RMSE	0.0105	0.0039	0.0017	0.0212
5.33					
Method 2	Encoding time (s)	72.95	87.25	68.07	71.28
$n = 8$	Decoding time (s)	1.15	1.09	0.98	1.73
8 × 8 × 8-sized blocks	PSNR (dB)	46.7	47.57	44.92	50.38
Total: 8065 blocks	RMSE	0.0168	0.0054	0.0025	0.0288
21.33					

4 Performance analysis—results

Table 1 shows encoding, decoding time, PSNR, and RMSE results of both the methods performed for 4 individual data sets [38]. It is evident from the images in Fig. 7 that there is no significant difference in spatial resolution or details. Therefore, 4× spatially scaled images are displayed in Fig. 8 which portrays the distorted block-like portions in the edge of the objects inside the image [43]. These patterns occur due to block processing employed in the factorized decomposition ($n = 4,8$), and the blocks in Fig. 8c are more disturbing than that of scene 1 ($n = 4$) and their PSNR values are 46.7 dB and 50.42 dB resp. RMSE of 0.0105 proves the absence of distortions in Fig. 8b compared to 0.0168. The entire process of disintegrating the HSI image data set explained in Section 3.1 (steps 1 through 12) takes a duration of 222 s ($n = 4$) while for $n = 8$, total encoding time is 72.9 s. This difference is due to the fact that the former processes 64,516 blocks while the latter targets only 8065 blocks which in turn reduces the computational time. The steps of decoding, which is done at a farther end, are just matrix multiplication. The decoding time of the former and latter is 0.94 s and 1.15 s, respectively, in spite of the fact that the former has a greater number of blocks. This lesser time of the former is due to the multicore processors used which can parallelize the process of decoding, i.e., matrix multiplication.

Data sets of different environments (scenes 1 to 4) are chosen so as to check the integrity of the proposed technique on all global data conditions. Scene 2 is an HSI of wooden planks exhibiting high contrast and vibrancy. Its compressed and scaled results are displayed in Figs. 9 and 10. The reconstructed image Fig. 10c is the worst performer of all because of the block processing that destroyed majority of the details. Therefore, Fig. 10b can be considered as the effective replacement for the original image.

Scene 3 contains branches of a rose plant, and it indeed has a lot of foreground object and background depth details (Figs. 11 and 12). The 3.5× scaled result is more distorted in Fig. 12c (RMSE = 0.0025), and its performance is similar to that of scene 2.

Figures 13 and 14 are compressed versions of scene 4 representing a wooden house delivering a varying level of textures. On observing Fig. 14, the scaled versions have no significant difference and, therefore, Fig. 14c can be used instead of the original image for processing. It is also evident from the difference in magnitudes of individual PSNRs, i.e., 2.65 dB. Scene 4—compressed image by method 2 consumes 71 s, lesser than that of method 1 which is due to the block size of 8 × 8 × 8.

(a) (b) (c)

Fig. 8 Scene 1 (4× scaled)—**a** original (band 21), **b** compressed RMSE = 0.0105 ($n = 4$), **c** compressed RMSE = 0.0168 ($n = 8$)

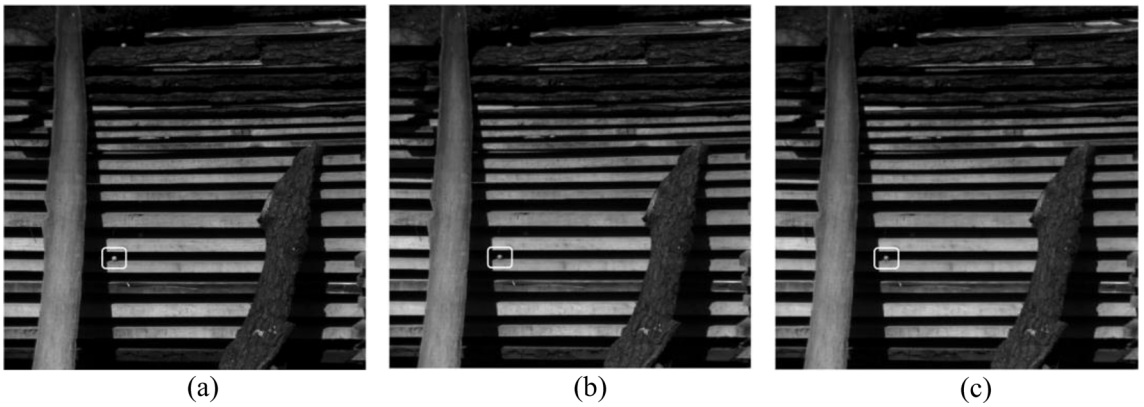


Fig. 9 Scene 2—**a** original (band 21), **b** compressed RMSE = 0.0039 ($n = 4$), **c** compressed RMSE = 0.0054 ($n = 8$)

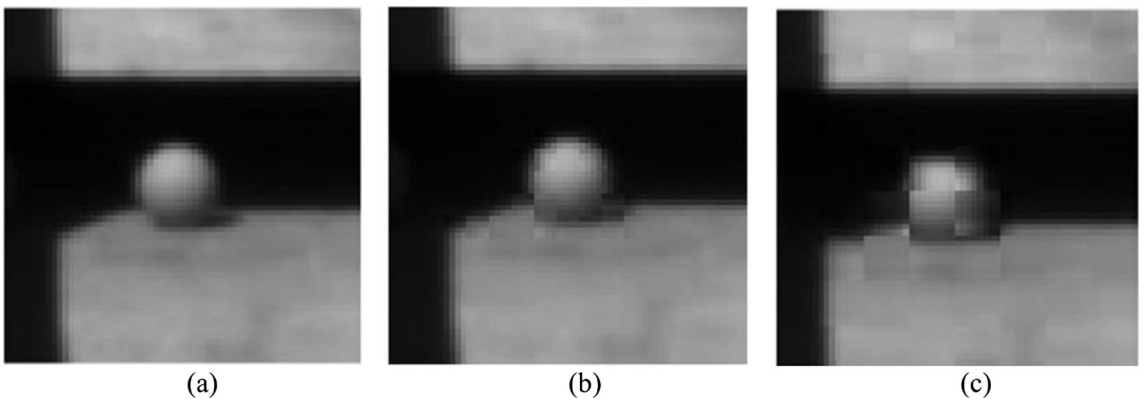


Fig. 10 Scene 2 (4× scaled)—**a** original (band 21), **b** compressed RMSE = 0.0039 ($n = 4$), **c** compressed RMSE = 0.0054 ($n = 8$)

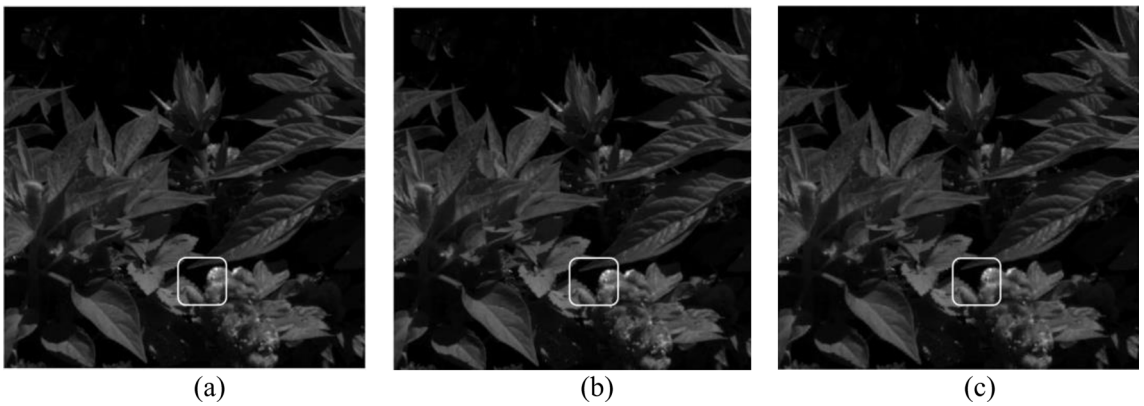


Fig. 11 Scene 3—**a** original (band 21), **b** compressed RMSE = 0.0017 ($n = 4$), **c** compressed RMSE = 0.0025 ($n = 8$)

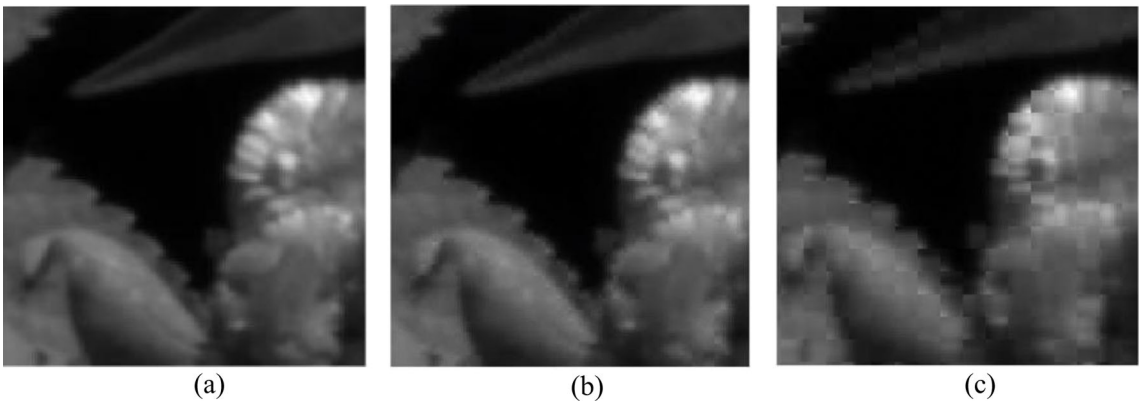


Fig. 12 Scene 3 (4× scaled)—**a** original (band 21), **b** compressed RMSE = 0.0017 ($n = 4$), **c** compressed RMSE = 0.0025 ($n = 8$)

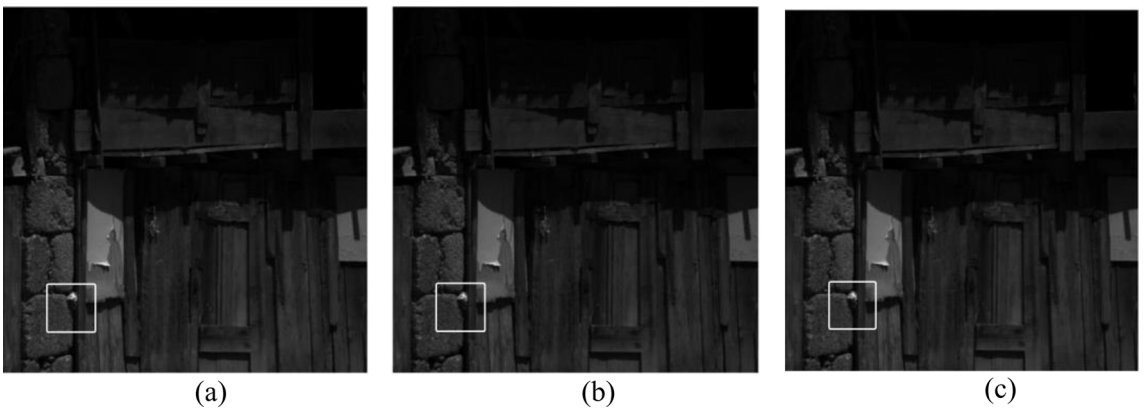


Fig. 13 Scene 4—**a** original (band 21), **b** compressed RMSE = 0.0212 ($n = 4$), **c** compressed RMSE = 0.0288 ($n = 8$)

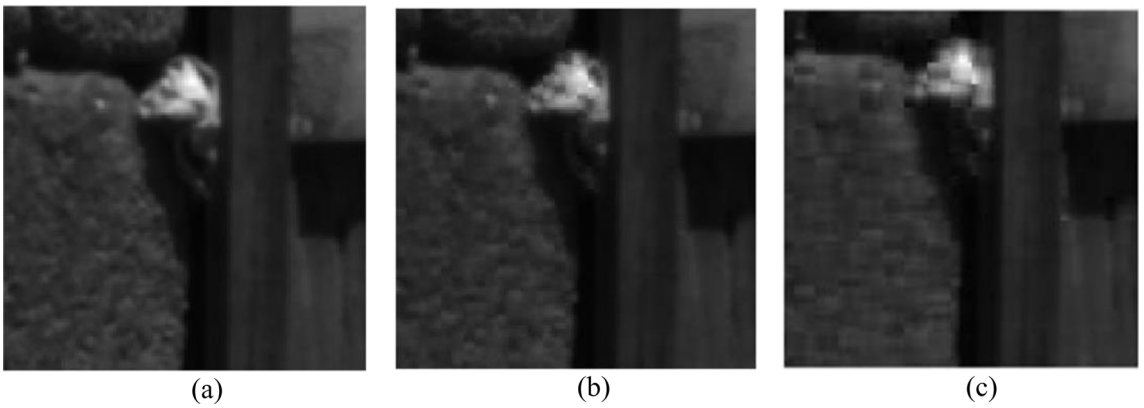


Fig. 14 Scene 4 (4× scaled)—**a** original (band 21), **b** compressed RMSE = 0.0212 ($n = 4$), **c** compressed RMSE = 0.0288 ($n = 8$)

5 General discussion and conclusion

The proposed method is applied over 4 sample data sets of diversified feeds. It's quantitative and qualitative performance is investigated with the help of parameters like PSNR, RMSE, and processing time and tabulated in comparison Table 1. The absolute challenge in compressing a HSI data is to venture the compression algorithms onto selected bands of interest and exploit the redundancy in the spectral resolution without employing time-consuming iterations. Therefore, such a novel algorithm is proposed and analyzed in this paper. Method 1 and method 2 have their own significance in different data sets, and they are indeed experimental and cannot be formulated. Method 2 reconstructed images in Fig. 14c scene 4 have negligible spatial difference than that of method 1, and so, the former can replace the original. With reference to the comparison Table 1, we would like to sum up that the proposed algorithm compresses the image data set at the ratio of 3747:1 for $n = 4$ and 245:1 for $n = 8$. It also completes the process of compression in 222 s (scene 1) which is 10 times less than that of the time consumed by the Tucker decomposition technique [36]. Tucker decomposition is a time inefficient method since it employs several steps of perpetual iterations and involves the determination of a singular rank matrix for decomposing the 3D tensors in spite of its familiarity in remote sensing. However, our proposed algorithm has surpassed all the shortcomings of Tucker decomposition. A dedicated memory cum processing chip can also further aid to decrease the encoding time at the storage or transmitter end. At the receiver or decoder end, the process of regeneration or reconstruction from the row tensors consumes less than 1 s (scene 1) which is similar to the decoding time of the present generation 2D image in JPEG formats. However, the compression ratio can also be increased further by transmitting significant decomposed pixel values instead of entire row vectors.

Acknowledgments The authors wish to acknowledge with thanks the help rendered by Thiru Ashok Kumar Das, IPS., ADGP (Technical Services), Tamil Nadu and Tmt N.Z Asiammal, IPS., DIG (Technical Services), Tamil Nadu Dr. C. V. Jayakumar, Principal, Sri Sairam Engg. College, Chennai, and Prof. A. R. Rajini, Head of the ECE Department, and all others in Sri Sairam Engg. College, Chennai, and Dr. N. Nithiyanandam, Professor, ECE Department, B. S. Abdur Rahman University, Chennai.

References

1. Yang, C., Everitt, J.H.: Precision Agric **13**, 62 (2012). https://doi.org/10.1007/s11119-011-9248-z
2. Cao, X., Li, R., Wen, L., Feng, J., Jiao, L.: Deep multiple feature fusion for hyperspectral image classification. IEEE J. Selected Topics Appl. Earth Observ. Remote Sens. **11**(10), 3880–3891 (2018)
3. Marinoni, A., Clenet, H.: Higher order nonlinear hyperspectral unmixing for mineralogical analysis over extraterrestrial bodies. IEEE J. Selected Topics Appl. Earth Observ. Remote Sens. **10**(8), 3722–3733 (2017)
4. Park, B., Yoon, S.C., Windham, W.R., et al.: Sens. Instrumen. Food Qual. **5**, 25 (2011). https://doi.org/10.1007/s11694-011-9107-7
5. Moriya, É.A.S., Imai, N.N., Tommaselli, A.M.G., Miyoshi, G.T.: Mapping mosaic virus in sugarcane based on hyperspectral images. IEEE J. Selected Topics Appl. Earth Observ. Remote Sens. **10**(2), 740–748 (2017)
6. Aviris.jpl.nasa.gov.: AVIRIS - airborne visible / infrared imaging spectrometer. [online] Available at: https://aviris.jpl.nasa.gov/index.html (2018)
7. Aviris.jpl.nasa.gov.: AVIRIS - airborne visible / infrared imaging spectrometer - concept. [online] Available at: https://aviris.jpl.nasa.gov/aviris/concept.html (2018)
8. Lee, C.M. et al.: An introduction to the NASA Hyperspectral InfraRed Imager (HyspIRI) mission and preparatory activities. Remote Sens. Environ. **167**, 6–19 (2015)
9. Huo, C., Zhang, R., Peng, T.: Lossless compression of hyperspectral images based on searching optimal multibands for prediction. IEEE Geosci. Remote Sens. Lett. **6**(2), 339–343 (2009)
10. Licciardi, G.A., Chanussot, J., Piscini, A.: Spectral compression of hyperspectral images by means of nonlinear principal component analysis decorrelation. In: 2014 IEEE International Conference on Image Processing (ICIP), pp. 5092–5096, Paris (2014). https://doi.org/10.1109/ICIP.2014.7026031
11. Ertürk, A., Ertürk, S.: Seam carving for hyperspectral image size reduction and unmixing. In: 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 1752–1755, Fort Worth (2017)
12. Ryan, M.J., Arnold, J.F.: Lossy compression of hyperspectral data using vector quantization. Remote Sens. Environ. **61**(3), 419–436 (1997)
13. Settle, J.: On the dimensionality of multi-view hyperspectral measurements of vegetation. Remote Sens. Environ. **90**(2), 235–242 (2004)
14. Warner, T.A., Shank, M.C.: Spatial autocorrelation analysis of hyperspectral imagery for feature selection. Remote Sens. Environ. **60**(1), 58–70 (1997)
15. Senay, S., Erives, H.: Low complexity dimensionality reduction for hyperspectral images. In: 2014 48th Asilomar Conference on Signals, Systems and Computers, pp. 1551–1554, Pacific Grove (2014). https://doi.org/10.1109/ACSSC.2014.7094724
16. Falco, N., Bruzzone, L., Benediktsson, J.A.: A comparative study of different ICA algorithms for hyperspectral image analysis. In: 2013 5th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), pp. 1–4, Gainesville (2013)
17. Falco, N., Benediktsson, J.A., Bruzzone, L.: Spectral and spatial classification of hyperspectral images based on ICA and reduced morphological attribute profiles. IEEE Trans. Geosci. Remote Sens. **53**(11), 6223–6240 (2015)
18. Hou, Y., Liu, G.: Lossy-to-lossless compression of hyperspectral image using the improved AT-3D SPIHT algorithm. In: 2008 International Conference on Computer Science and Software Engineering, pp. 963–966, Wuhan (2008)
19. Christophe, E., Mailhes, C., Duhamel, P.: Hyperspectral image compression: Adapting SPIHT and EZW to anisotropic 3-D wavelet coding. IEEE Trans. Image Process. **17**(12), 2334–2346 (2008)

20. Lee, S., Lee, E., Choi, H., Lee, C.: Compression of hyperspectral images with 2D wavelet transform using adjacent information and SPIHT slgorithm, Proceedings. In: 2005 IEEE International Geoscience and Remote Sensing Symposium, 2005. IGARSS '05, p. 3 (2005). https://doi.org/10.1109/IGARSS.2005.1526118
21. Du, Q., Fowler, J.E.: Hyperspectral image compression using JPEG2000 and principal component analysis. IEEE Geosci. Remote Sensing Lett. **4**(2), 201–205 (April 2007)
22. Zemliachenko, A., Lukin, V., Vozel, B.: Lossy compression of hyperspectral images based on JPEG2000. In: 2017 4th International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC SandT), pp. 600–603, Kharkov (2017)
23. Menon, V., Du, Q., Fowler, J.E.: Fast SVD with random Hadamard projection for hyperspectral dimensionality reduction. IEEE Geosci. Remote Sensing Lett. **13**(9), 1275–1279 (2016)
24. Bai, X., Xu, F., Zhou, L., Xing, Y., Bai, L., Zhou, J.: Nonlocal similarity based nonnegative Tucker decomposition for hyperspectral image denoising. IEEE J. Selected Topics Appl. Earth Observ. Remote Sens. **11**(3), 701–712 (2018)
25. Wang, Y., Lin, L., Zhao, Q., Yue, T., Meng, D., Leung, Y.: Compressive sensing of hyperspectral images via joint tensor tucker decomposition and weighted total variation regularization. IEEE Geosci. Remote Sens. Lett. **14**(12), 2457–2461 (2017)
26. Karami, A., Yazdi, M., Mercier, G.: Compression of hyperspectral images using discerete wavelet transform and Tucker decomposition. IEEE J. Selected Topics Appl. Earth Observ. Remote Sens. **5**(2), 444–450 (2012)
27. Karami, A., Yazdi, M., Asli, A.Z.: Hyperspectral image compression based on Tucker decomposition and discrete cosine transform. In: 2010 2nd International Conference on Image Processing Theory, Tools and Applications, pp. 122–125, Paris (2010)
28. Bentley, P.M., McDonnell, J.T.E.: Wavelet transforms: an introduction. Electron. Commun. Eng. J. **6**(4), 175–186 (1994)
29. Chun-Lin, L.: A Tutorial of the Wavelet Transform. Taipei (2010)
30. Lerma, M.A.: Principal Components Analysis in 2D. Evanston (2017)
31. Jolliffe, I.T.: Principal Component Analysis, 2nd edn. Springer (2002)
32. Mudrová, M., Procházka, A.: Principal component analysis in image processing. In: Proceedings of International Conference Technical Computing, Prague (2005)
33. Jauregui, J.: Principal Component Analysis with Linear Algebra. Penn Arts and Sciences, Philadelphia (2012)
34. Richardson, M.: Principal component analysis, Special topic essay, M.Sc. Mathematical Modelling and Scientific Computing, University of Oxford (2009)
35. Golub, G.H., Van Loan, C.F.: The Singular Value Decomposition and Unitary Matrices. § 2.5.3 and 2.5.6 in Matrix Computations. 3rd edn., pp. 70–71 and 73. Johns Hopkins University Press, Baltimore (1996)
36. Jeyakumar, S., Sudha, S.: A unique method to decompose hyperspectral images with transformed pixel data. Int. J. Appl. Eng. Res., **10**(17). ISSN 0973–4562 (2015)
37. Jeyakumar, S., Sudha, S.: Tensor decomposition based compression and analysis for 2D image data. Int. J. Appl. Eng. Res., **10**(87). ISSN 0973–4562 (2015)
38. Foster, D.H., Amano, K., Nascimento, S.M.C., Foster, M.J.: Frequency of metamerism in natural scenes. J. Optical So. Am. A **23**, 2359–2372 (2006)
39. Abdullatif, H.: You don't know SVD (singular value decomposition), Towards Data Science, 21-Feb-2019. [Online]. Available: https://towardsdatascience.com/svd-8c2f72e264f
40. A gentle introduction to singular-value decomposition (SVD) for machine learning, Machine Learning Mastery, 19-Apr-2019. [Online]. Available: https://machinelearningmastery.com/singular-value-decomposition-for-machine-learning/
41. Fernández, J.A., Moreno, M.D.: A block size optimization algorithm for parallel image processing. In: 2014 International Conference on Computer Vision Theory and Applications (VISAPP), pp. 138–144, Lisbon (2014)
42. Raju, U.S.N., Kumar, K.S., Mehta, V., Sharma, R., Kuli, S.: Cluster based block processing for gigantic images: dimension and size. In: 2017 Fourth International Conference on Image Information Processing (ICIIP), pp. 1–5, Shimla (2017)
43. In.mathworks.com.: Distinct block processing- MATLAB and Simulink- MathWorks India. [online] Available at: https://in.mathworks.com/help/images/distinct-block-processing.html (2018)