CCSDS
The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

LOW-COMPLEXITY LOSSLESS AND NEAR-LOSSLESS MULTISPECTRAL AND HYPERSPECTRAL IMAGE COMPRESSION

RECOMMENDED STANDARD

CCSDS 123.0-B-2

BLUE BOOK
February 2019

The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

LOW-COMPLEXITY LOSSLESS AND NEAR-LOSSLESS MULTISPECTRAL AND HYPERSPECTRAL IMAGE COMPRESSION

RECOMMENDED STANDARD

CCSDS 123.0-B-2

BLUE BOOK

February 2019

AUTHORITY

Issue:	Recommended Standard, Issue 2
Date:	February 2019
Location:	Washington, DC, USA

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and represents the consensus technical agreement of the participating CCSDS Member Agencies. The procedure for review and authorization of CCSDS documents is detailed in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4), and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the e-mail address below.

This document is published and maintained by:

CCSDS Secretariat
National Aeronautics and Space Administration
Washington, DC, USA
E-mail: secretariat@mailman.ccsds.org

STATEMENT OF INTENT

The Consultative Committee for Space Data Systems (CCSDS) is an organization officially established by the management of its members. The Committee meets periodically to address data systems problems that are common to all participants, and to formulate sound technical solutions to these problems. Inasmuch as participation in the CCSDS is completely voluntary, the results of Committee actions are termed **Recommended Standards** and are not considered binding on any Agency.

This **Recommended Standard** is issued by, and represents the consensus of, the CCSDS members. Endorsement of this **Recommendation** is entirely voluntary. Endorsement, however, indicates the following understandings:

o Whenever a member establishes a CCSDS-related **standard**, this **standard** will be in accord with the relevant **Recommended Standard**. Establishing such a **standard** does not preclude other provisions which a member may develop.

o Whenever a member establishes a CCSDS-related **standard**, that member will provide other CCSDS members with the following information:

-- The **standard** itself.

-- The anticipated date of initial operational capability.

-- The anticipated duration of operational service.

o Specific service arrangements shall be made via memoranda of agreement. Neither this **Recommended Standard** nor any ensuing **standard** is a substitute for a memorandum of agreement.

No later than five years from its date of issuance, this **Recommended Standard** will be reviewed by the CCSDS to determine whether it should: (1) remain in effect without change; (2) be changed to reflect the impact of new technologies, new requirements, or new directions; or (3) be retired or canceled.

In those instances when a new version of a **Recommended Standard** is issued, existing CCSDS-related member standards and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each member to determine when such standards or implementations are to be modified. Each member is, however, strongly encouraged to direct planning for its new standards and implementations towards the later version of the Recommended Standard.

FOREWORD

This Recommended Standard specifies a method for lossless and near-lossless compression of multispectral and hyperspectral image data and a format for storing the compressed data.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CCSDS has processes for identifying patent issues and for securing from the patent holder agreement that all licensing policies are reasonable and non-discriminatory. However, CCSDS does not have a patent law staff, and CCSDS shall not be held responsible for identifying any or all such patent rights.

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Standard is therefore subject to CCSDS document management and change control procedures, which are defined in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4). Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

Questions relating to the contents or status of this document should be sent to the CCSDS Secretariat at the e-mail address indicated on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- China National Space Administration (CNSA)/People's Republic of China.
- Deutsches Zentrum für Luft- und Raumfahrt (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Federal Space Agency (FSA)/Russian Federation.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.
- UK Space Agency/United Kingdom.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian Federal Science Policy Office (BFSPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
- Chinese Academy of Sciences (CAS)/China.
- China Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Danish National Space Center (DNSC)/Denmark.
- Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
- Electronics and Telecommunications Research Institute (ETRI)/Korea.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
- Hellenic National Space Committee (HNSC)/Greece.
- Hellenic Space Agency (HSA)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- Mohammed Bin Rashid Space Centre (MBRSC)/United Arab Emirates.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
- National Space Organization (NSPO)/Chinese Taipei.
- Naval Center for Space Technology (NCST)/USA.
- Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
- South African National Space Agency (SANSA)/Republic of South Africa.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- Swiss Space Office (SSO)/Switzerland.
- United States Geological Survey (USGS)/USA.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 123.0-B-1	Lossless Multispectral & Hyperspectral Image Compression, Recommended Standard, Issue 1	May 2012	Original issue, superseded
CCSDS 123.0-B-2	Low-Complexity Lossless and Near-Lossless Multispectral and Hyperspectral Image Compression, Recommended Standard, Issue 2	February 2019	Current issue

CONTENTS

Section Page

CONTENTS (continued)

Section Page

Figure

Table

CONTENTS (continued)

Table Page

CONTENTS (continued)

1 INTRODUCTION

1.1 PURPOSE

The purpose of this document is to establish a Recommended Standard for a data compression algorithm applied to digital three-dimensional image data from payload instruments, such as multispectral and hyperspectral imagers, and to specify the compressed data format.

Data compression is used to reduce the volume of digital data to achieve benefits in areas including, but not limited to,

a) reduction of transmission channel bandwidth;

b) reduction of the buffering and storage requirement; and

c) reduction of data-transmission time at a given rate.

1.2 SCOPE

The characteristics of instrument data are specified only to the extent necessary to ensure multi-mission support capabilities. The specification does not attempt to quantify the relative bandwidth reduction, the merits of the approaches discussed, or the design requirements for encoders and associated decoders. Some performance information is included in reference [D1].

This Recommended Standard addresses lossless and near-lossless compression of three-dimensional data and provides a compression method that ensures that the distortion in the reconstructed image does not exceed user-specified limits.

1.3 APPLICABILITY

This Recommended Standard applies to data compression applications of space missions anticipating packetized telemetry cross support. In addition, it serves as a guideline for the development of compatible CCSDS Agency standards in this field, based on good engineering practice.

1.4 RATIONALE

The concept and rationale for the Low-Complexity Lossless and Near-Lossless Multispectral and Hyperspectral Image Compression algorithm described herein may be found in reference [D1].

1.5 DOCUMENT STRUCTURE

This document is organized as follows:

a) Section 1 provides the purpose, scope, applicability, and rationale of this Recommended Standard and identifies the conventions and references used throughout the document. This section also describes how this document is organized. A brief description is provided for each section and annex so that the reader will have an idea of where information can be found in the document.

b) Section 2 provides an overview of the compressor.

c) Section 3 defines parameters and notation pertaining to an input image to be compressed.

d) Section 4 specifies the predictor stage of the compressor.

e) Section 5 specifies the entropy coding stage of the compressor and the format of a compressed image.

f) Annex A provides the Implementation Conformance Statement Requirements List for implementations of this Recommended Standard.

g) Annex B lists code tables used by one of the entropy coding options.

h) Annex C discusses security, Space Assigned Numbers Authority (SANA), and patent considerations.

i) Annex D lists informative references.

j) Annex E provides tables of symbols used in this document.

k) Annex F expands abbreviations and acronyms used in this document.

1.6 CONVENTIONS AND DEFINITIONS

1.6.1 MATHEMATICAL NOTATION AND DEFINITIONS

In this document, for any real number x, the largest integer n such that $n \leq x$ is denoted by

$$n = \lfloor x \rfloor, \tag{1}$$

and correspondingly, the smallest integer n such that $n \geq x$ by

$$n = \lceil x \rceil, \tag{2}$$

The modulus of an integer *M* with respect to a positive integer divisor *n*, denoted *M* mod *n*, is defined to be

$$M \bmod n = M - n\lfloor M / n \rfloor . \tag{3}$$

When it is stated that a value *M* is encoded modulo *n*, this means the number *M* mod *n* is encoded instead of *M*.

For any integer *x* and positive integer *R*, the function $\bmod_R^*[x]$ is defined as

$$\bmod_R^*[x] = \left(\left(x + 2^{R-1}\right) \bmod 2^R\right) - 2^{R-1} . \tag{4}$$

NOTE – The quantity $\bmod_R^*[x]$ is the *R*-bit two's complement integer that is congruent to *x* modulo 2^R. This is a natural result of storing a signed integer *x* in an *R*-bit register in two's complement form when overflow might occur.

The notation $\mathrm{clip}\left(x, \left\{x_{\min}, x_{\max}\right\}\right)$ denotes the clipping of the real number *x* to the range $\left[x_{\min}, x_{\max}\right]$, that is,

$$\mathrm{clip}\left(x, \left\{x_{\min}, x_{\max}\right\}\right) = \begin{cases} x_{\min}, & x < x_{\min} \\ x, & x_{\min} \le x \le x_{\max} \\ x_{\max}, & x > x_{\max} \end{cases} . \tag{5}$$

For any real number *x*, the function sgn(*x*) is defined as

$$\mathrm{sgn}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases} . \tag{6}$$

and the function $\mathrm{sgn}^+(x)$ is defined as

$$\mathrm{sgn}^+(x) = \begin{cases} 1, & x \ge 0 \\ -1, & x < 0 \end{cases} . \tag{7}$$

1.6.2 NOMENCLATURE

1.6.2.1 Normative Text

The following conventions apply for the normative specifications in this Recommended Standard:

a) the words 'shall' and 'must' imply a binding and verifiable specification;

b) the word ‘should’ implies an optional, but desirable, specification;

c) the word ‘may’ implies an optional specification;

d) the words ‘is’, ‘are’, and ‘will’ imply statements of fact.

NOTE – These conventions do not imply constraints on diction in text that is clearly informative in nature.

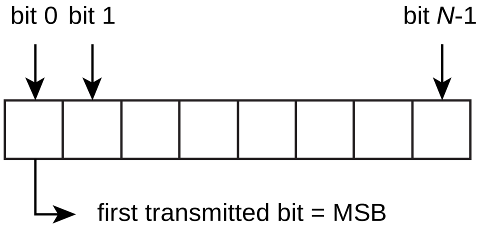
1.6.2.2 Informative Text

In the normative sections of this document (sections 3-5), informative text is set off from the normative specifications either in notes or under one of the following subsection headings:

- Overview;
- Background;
- Rationale;
- Discussion.

1.6.3 CONVENTIONS

In this document, the following convention is used to identify each bit in an *N*-bit word. The first bit in the word to be transmitted (i.e., the most left justified when drawing a figure) is defined to be ‘bit 0’, the following bit is defined to be ‘bit 1’, and so on up to ‘bit *N*-1’. When the word is used to express an unsigned binary value (such as a counter), the Most Significant Bit (MSB), bit 0, shall correspond to the highest power of two, that is, 2^{N-1}.



In accordance with modern data communications practice, spacecraft data words are often grouped into 8-bit ‘words’ that conform to the above convention. Throughout this Recommended Standard, the following nomenclature is used to describe this grouping:

8-bit word = ‘Byte’

1.7 REFERENCE

The following publication contains provisions which, through reference in this text, constitute provisions of this document. At the time of publication, the edition indicated was valid. All publications are subject to revision, and users of this document are encouraged to investigate the possibility of applying the most recent edition of the publication indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS publications.

[1] *Lossless Data Compression*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 121.0-B-2. Washington, D.C.: CCSDS, May 2012.

NOTE – Non-normative references are contained in annex D.

2 OVERVIEW

2.1 GENERAL

This Recommended Standard defines a payload data compressor that has applicability to multispectral and hyperspectral imagers and sounders. This Recommended Standard does not attempt to explain the theory underlying the compression algorithm; that theory is partially addressed in reference [D1].

This Issue 2 revision extends the CCSDS Lossless Multispectral & Hyperspectral Image Compression standard (reference [D2]) to provide an effective method of performing either lossless or near-lossless compression of three-dimensional image data. Here, 'near-lossless' refers to the ability to perform compression in a way that the maximum error in the reconstructed image can be limited to a user-specified bound. Key changes introduced in this revision include the incorporation of a closed-loop quantization scheme to provide near-lossless compression and the extension of an entropy coding method of reference [D2] to provide better compression of low-entropy data.

The input to the compressor is an image, which for the purposes of this Recommended Standard is a three-dimensional array of integer sample values, as specified in section 3. The compressed image output from the compressor is an encoded bitstream from which the input image can be exactly or approximately reconstructed.

The compression method is capable of producing a reconstructed image meeting a fidelity constraint specified by the user during compression, including lossless compression. A user may vary fidelity settings from band to band and change these settings periodically within an image.

For a given set of compression parameters, the length of the compressed image will vary depending on image content. That is, the compressed image is variable-length. A user could exploit the ability to adaptively vary fidelity settings within an image in an effort to meet a constraint on compressed image data volume; techniques for performing this optimization are outside the scope of this document. Reference [D1] presents some examples.

A user may choose to partition the output of an imaging instrument into multiple images that are separately compressed, for example, to limit the impact of data loss or corruption on the communications channel, or to limit the maximum possible size of a compressed image. This Recommended Standard does not address such partitioning or the tradeoffs associated with selecting the size of images produced under such partitioning. Reference [D1] presents some examples.

Figure 2-1 depicts the components of the compressor defined in this Recommended Standard. The compressor consists of a predictor followed by an encoder.

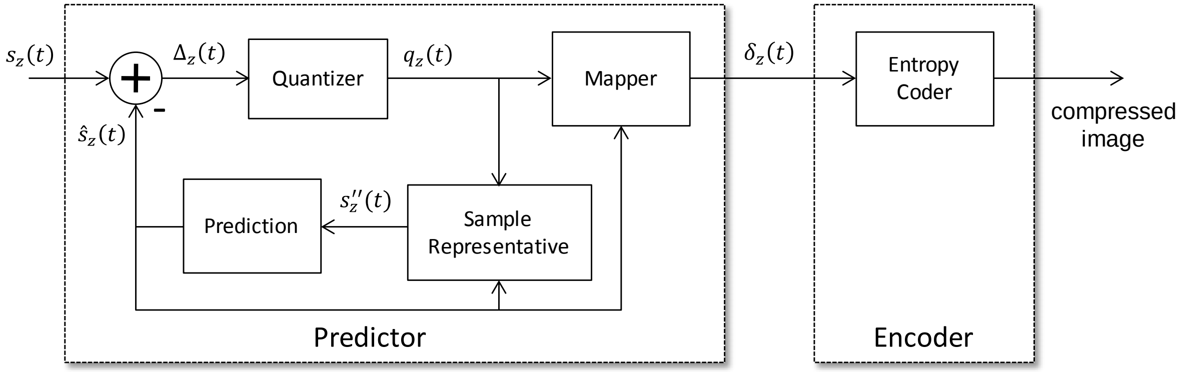


Figure 2-1: Compressor Schematic

The predictor, specified in section 4, uses an adaptive linear prediction method to predict the value of each image sample based on the values of nearby samples in a small three-dimensional neighborhood. Prediction is performed sequentially in a single pass.

The predictor makes use of an adaptively weighted prediction algorithm similar to the one in reference [D2]. Compared to reference [D2], this Recommended Standard has some minor changes in calculation of the prediction weights. More significantly, here the predictor cannot in general utilize the exact values of the original sample values $s_z(t)$ because these values will not be available to the decompressor at the time of reconstruction when compression is not lossless. Instead, prediction calculations are performed using a *sample representative* $s''_z(t)$ in place of each original sample value $s_z(t)$. The calculation of the sample representative is specified in 4.9.

The predictor in the present Recommended Standard also differs from that of reference [D2] in that each prediction residual $\Delta_z(t)$, that is, the difference between the predicted and actual sample values, is quantized using a uniform quantizer. The quantizer step size can be controlled via an *absolute error limit* (so that samples can be reconstructed with a user-specified error bound) and/or a *relative error limit* (so that samples predicted to have smaller magnitude can be reconstructed with lower error). Lossless compression in a band is obtained simply by setting the absolute error limit to zero. The quantized prediction residual $q_z(t)$ is mapped to an unsigned integer *mapped quantizer index* $\delta_z(t)$, similar to the calculation of mapped prediction residuals in reference [D2]. These mapped quantizer indices make up the output of the predictor.

The compressed image, specified in section 5, consists of a header that encodes image and compression parameters followed by a body that is produced by an entropy coder, which losslessly encodes the mapped quantizer indices. Entropy coder parameters are adaptively adjusted during this process to adapt to changes in the statistics of the mapped quantizer indices.

2.2 LOSSLESS COMPRESSION

Some simplification of the predictor arises when lossless compression is selected for the quantizer fidelity control method. Specifically, the quantization calculation (4.8) becomes trivial: $q_z(t) = \Delta_z(t)$.

In addition, under lossless compression, in the sample representative calculation (4.9), the offset parameter ψ_z has no effect and is defined as zero; and if a user chooses to set the damping parameter to zero, $\phi_z = 0$, then the sample representatives are equal to the original sample values, $s''_z(t) = s_z(t)$. This simplification can facilitate the ability to perform pipelining in a hardware implementation of the compressor.

It should be noted, however, that lossless compression performance may be improved by using a nonzero value for the damping parameter ϕ_z in the sample representative calculation.

Reference [D1] includes additional discussion.

2.3 BACKWARDS COMPATIBILITY

The features and compressed image header structure of the present Recommended Standard have been developed to ensure backwards compatibility with issue 1 of this Recommended Standard specified in reference [D2], which provided only lossless compression.

Specifically, reference [D2] can be viewed as a restricted case of the present Recommended Standard; a decompressor supporting all features of the present Recommended Standard would be able to decompress a compressed image that is compliant with issue 1. However, it should be noted that the additional features added in this issue are not limited to near-lossless compression capabilities. Thus, for example, a losslessly compressed image that is compliant with the present Recommended Standard might not be decompressible with a decompressor that is compliant with issue 1.

Table 2-1 enumerates the constraints that would need to be imposed on an implementation of this Recommended Standard to produce a compressor that is compliant with reference [D2].

Table 2-1: Backwards Compatibility with CCSDS-123.0-B-1

Reference	Constraint
3.3.1	Limit dynamic range to $D \leq 16$ bits.
3.5	Do not use supplementary information tables (set $\tau = 0$).
4.4	Do not use narrow local sums.
4.8	Set the quantizer fidelity control method to be lossless (4.8.2.1.1).

Reference	Constraint
4.9, 5.3.3	Set sample representative parameters to $\phi_z = \psi_z = 0$ for all z. Each sample representative $s''_z(t)$ will be equal to $s_z(t)$. Set $\Theta = 0$ so that the Predictor Metadata header part does not include the Sample Representative subpart.
4.10.3, 5.3.3.2	Set all weight exponent offsets $\varsigma_z^{(i)}$, ς_z^* to 0. In the Primary subpart of the Predictor Metadata header part, set the Weight Exponent Offset Flag to ‘0’.
5.4.3.2.3.4	If using the sample-adaptive entropy coder, do not use a rescaling counter size parameter γ^* value larger than 9.
5.4.3.3	Do not use the hybrid entropy coder.

2.4 DATA TRANSMISSION

The effects of a small error or data loss event can propagate to corrupt an entire compressed image (see reference [D1] for an example). Therefore, measures should be taken to minimize errors and data loss in the compressed image.

This Recommended Standard does not incorporate sync markers or other mechanisms to flag the header of an image; it is assumed that the transport mechanism used for the delivery of the encoded bitstream will provide the ability to locate the beginning and end of a compressed image and, in the event of data corruption, the header of the next image.

In case the encoded bitstream is to be transmitted over a CCSDS space link, several protocols can be used to transfer a compressed image, including but not limited to

– Space Packet Protocol (reference [D3]);

– CCSDS File Delivery Protocol (CFDP) (reference [D4]); and

– packet service as provided by the AOS Space Data Link Protocol (reference [D5]), TM Space Data Link Protocol (reference [D6]), and Unified Space Data Link Protocol (reference [D7]).

When transmission over a CCSDS space link occurs, application of one of the set of Channel Coding and Synchronization Recommended Standards will significantly reduce the loss of portions of transmitted data caused by data corruption.

Limits on the maximum size data unit that can be transmitted may be imposed by the protocol used or by other practical implementation considerations. The user is expected to take such limits into account when using this Recommended Standard.

3 IMAGE

3.1 OVERVIEW

This section defines parameters and notation pertaining to an image. Quantities defined in this section are summarized in table E-1 of annex E.

3.2 DIMENSIONS

3.2.1 An *image* is a three-dimensional array of signed or unsigned integer sample values $s_{z,y,x}$, where x and y are indices in the spatial dimensions, and the index z indicates the spectral band.

NOTES

1 When spatially adjacent data samples are produced by different instrument detector elements, changing values of the x index should correspond to changing detector elements. Thus, for a typical push-broom imager, the x and y dimensions would correspond to cross-track and along-track directions, respectively.

2 The spectral bands of the image need not be arranged in order of increasing or decreasing wavelength. Rearranging the order of spectral bands can affect compression performance. This Recommended Standard does not address the tradeoffs associated with such a band reordering. Reference [D1] includes some discussion of this topic.

3.2.2 Indices x, y, and z take on integer values in the ranges $0 \le x \le N_X - 1$, $0 \le y \le N_Y - 1$, and $0 \le z \le N_Z - 1$, where each image dimension N_X, N_Y, and N_Z shall have a value of at least 1 and at most 2^{16}.

3.2.3 A *frame* F_y is defined as the sub-array of all image sample values with the same y coordinate value; that is,

$$F_y(z, x) = s_{z,y,x} \text{ for any } 0 \le x \le N_X - 1, 0 \le z \le N_Z - 1. \qquad (8)$$

3.3 DYNAMIC RANGE

3.3.1 Data samples shall have a fixed-size dynamic range of D bits, where D shall be an integer in the range $2 \le D \le 32$.

3.3.2 The quantities $s_{\min}$, $s_{\max}$, and s_{mid} denote the lower sample value limit, the upper sample value limit, and a mid-range sample value, respectively. When samples are unsigned integers, the values of $s_{\min}$, $s_{\max}$, and s_{mid} are defined as

$$s_{\min} = 0, s_{\max} = 2^D - 1, s_{\text{mid}} = 2^{D-1}, \qquad (9)$$

and when samples are signed integers, the values of s_{min}, s_{max}, and s_{mid} are defined as

$$s_{\mathrm{min}} = -2^{D-1},\ s_{\mathrm{max}} = 2^{D-1} - 1,\ s_{\mathrm{mid}} = 0. \tag{10}$$

3.4 SAMPLE COORDINATE INDICES

For notational simplicity, data samples and associated quantities may be identified either by reference to the three indices x, y, z (e.g., $s_{z,y,x}$, $\delta_{z,y,x}$, etc.), or by the pair of indices t, z (e.g., $s_z(t)$, $\delta_z(t)$, etc.); that is,

$$s_z(t) \equiv s_{z,y,x} \tag{11}$$

$$\delta_z(t) \equiv \delta_{z,y,x} \tag{12}$$

etc., where

$$t = y \cdot N_{\mathrm{X}} + x. \tag{13}$$

NOTES

1 The value of t corresponds to the index of a sample within its spectral band when samples in the band are arranged in raster-scan order, starting with index t=0.

2 Given t, the values of x and y can be computed as

$$x = t \bmod N_{\mathrm{X}} \tag{14}$$

$$y = \lfloor t / N_{\mathrm{X}} \rfloor. \tag{15}$$

3.5 SUPPLEMENTARY INFORMATION TABLES

3.5.1 OVERVIEW

A user can choose to include up to 15 *supplementary information tables* to be encoded as part of the compressed image. Each such table is a zero-dimensional (a single element), one-dimensional (one element for each band z), or two-dimensional (one element for each (z, x) pair, or each (y, x) pair) table of floating-point, signed integer, or unsigned integer value(s). Such tables can be used to provide auxiliary image information to an end user, for example, the wavelength associated with each spectral band, a band-dependent scaling factor to convert reconstructed sample values to meaningful physical units, or a table identifying defective elements of a detector array. When used, such tables are encoded in the image header, as specified in 5.3.2.3.

3.5.2 SPECIFICATION

3.5.2.1 If supplementary information tables are used, the number of such tables, τ, shall be at most 15.

3.5.2.2 For each supplementary information table, the user shall identify a *purpose* for the table according to table 3-1. The 'reserved' purpose values are reserved for future use and shall not be used.

Table 3-1: Supplementary Information Table Purpose

Purpose	Interpretation
0	scale
1	offset
2	wavelength
3	full width at half maximum
4	defect indicator
5–9	reserved
10–15	user-defined

NOTE – The purpose is intended to indicate how a decompressor or end user might interpret the information in a supplementary information table. This does not impose any requirements on post-processing operations to be performed following decompression of an image.

3.5.2.3 Each supplementary information table *type* shall be unsigned integer, signed integer, or float.

3.5.2.3.1 For an unsigned integer table, the user-specified table bit depth D_I shall be an integer in the range $1 \le D_\mathrm{I} \le 32$, and each element of the table shall be an integer i in the range $0 \le i \le 2^{D_\mathrm{I}} - 1$.

3.5.2.3.2 For a signed integer table, the user-specified table bit depth D_I shall be an integer in the range $1 \le D_\mathrm{I} \le 32$, and each element of the table shall be an integer i in the range $-2^{D_\mathrm{I}-1} \le i \le 2^{D_\mathrm{I}-1} - 1$.

3.5.2.3.3 For a float table, user-specified significand and exponent bit depths D_F and D_E shall be integers in the range $1 \le D_\mathrm{F} \le 23$, $2 \le D_\mathrm{E} \le 8$, and the user-specified exponent bias β shall be an integer in the range $0 \le \beta \le 2^{D_\mathrm{E}} - 1$. Each element of the table shall consist of a sign bit b that is either 0 or 1, an exponent α that is an integer in the range $0 \le \alpha \le 2^{D_\mathrm{E}} - 1$, and a significand j that is an integer in the range $0 \le j \le 2^{D_\mathrm{F}} - 1$. If the exponent α is 0, the value represented is

$$(-1)^b \cdot j \cdot 2^{1-\beta-D_\mathrm{F}} . \tag{16}$$

If the exponent α is $2^{D_\mathrm{E}} - 1$, the value represented is non-numeric:

– $+\infty$ if $b = 0$ and $j = 0$;

– $-\infty$ if $b = 1$ and $j = 0$;

– NaN (‘not a number’, an undefined or unrepresentable value) if $j \neq 0$.

Otherwise, $0 < \alpha < 2^{D_\mathrm{E}} - 1$, and the value represented is

$$(-1)^b \left(2^{D_\mathrm{F}} + j\right) 2^{\alpha - \beta - D_\mathrm{F}} . \tag{17}$$

NOTE – When $D_\mathrm{E} = 8$, $D_\mathrm{F} = 23$, and $\beta = 127$, the float table representation of values is the same as the IEEE 754 single-precision binary floating-point format (binary32). When $D_\mathrm{E} = 5$, $D_\mathrm{F} = 10$, and $\beta = 15$, the float table representation of values is the same as the IEEE 754 half-precision binary floating-point format (binary16).

3.5.2.4 Each supplementary information table *structure* shall be zero-dimensional, one-dimensional, two-dimensional-*zx*, or two-dimensional-*yx*.

3.5.2.4.1 A zero-dimensional signed or unsigned integer supplementary information table consists of a single integer *i*.

3.5.2.4.2 A one-dimensional signed or unsigned integer supplementary information table consists of N_Z integers i_z, for $z = 0,\ldots, N_\mathrm{Z} - 1$.

3.5.2.4.3 A two-dimensional-*zx* signed or unsigned integer supplementary information table consists of $N_\mathrm{Z} \cdot N_\mathrm{X}$ integers $i_{z,x}$, for $z = 0,\ldots, N_\mathrm{Z} - 1$, $x = 0,\ldots, N_\mathrm{X} - 1$.

3.5.2.4.4 A two-dimensional-*yx* signed or unsigned integer supplementary information table consists of $N_\mathrm{Y} \cdot N_\mathrm{X}$ integers $i_{y,x}$, for $y = 0,\ldots, N_\mathrm{Y} - 1$, $x = 0,\ldots, N_\mathrm{X} - 1$.

3.5.2.4.5 A zero-dimensional float supplementary information table consists of a single element, defined by sign bit *b*, significand *j*, and exponent α.

3.5.2.4.6 A one-dimensional float supplementary information table consists of N_Z elements, each defined by sign bit b_z, significand j_z, and exponent α_z, for $z = 0,\ldots, N_\mathrm{Z} - 1$.

3.5.2.4.7 A two-dimensional-*zx* float supplementary information table consists of $N_\mathrm{Z} \cdot N_\mathrm{X}$ elements, each defined by sign bit $b_{z,x}$, significand $j_{z,x}$, and exponent $\alpha_{z,x}$, for $z = 0,\ldots, N_\mathrm{Z}-1$, $x = 0,\ldots, N_\mathrm{X}-1$.

3.5.2.4.8 A two-dimensional-*yx* float supplementary information table consists of $N_\mathrm{Y} \cdot N_\mathrm{X}$ elements, each defined by sign bit $b_{y,x}$, significand $j_{y,x}$, and exponent $\alpha_{y,x}$, for $y = 0,\ldots, N_\mathrm{Y} - 1$, $x = 0,\ldots, N_\mathrm{X} - 1$.

4 PREDICTOR

4.1 OVERVIEW

This section specifies the calculation of the *predicted sample values* $\hat{s}_{z,y,x}$ and *mapped quantizer indices* $\delta_{z,y,x}$ from the input image samples $s_{z,y,x}$. Quantities defined in this section are summarized in table E-2 of annex E.

This Recommended Standard makes use of the same adaptively weighted predictor as reference [D2], but to accommodate near-lossless compression, prediction calculations are performed using *sample representatives* $s''_{z,y,x}$, defined in 4.9, in place of the original sample values $s_{z,y,x}$. This is necessary so that the decompressor can duplicate the prediction calculation.

Prediction can be performed causally in a single pass through the image. Prediction at sample $s_{z,y,x}$, that is, the calculation of $\hat{s}_{z,y,x}$ and $\delta_{z,y,x}$, generally depends on the values of sample representatives for nearby samples in the current spectral band and P preceding (i.e., lower-indexed) spectral bands, where P is a user-specified parameter (see 4.2). Figure 4-1 illustrates the typical neighborhood used for prediction; this neighborhood is suitably truncated when $y = 0$, $x = 0$, $x = N_X - 1$, or $z < P$.

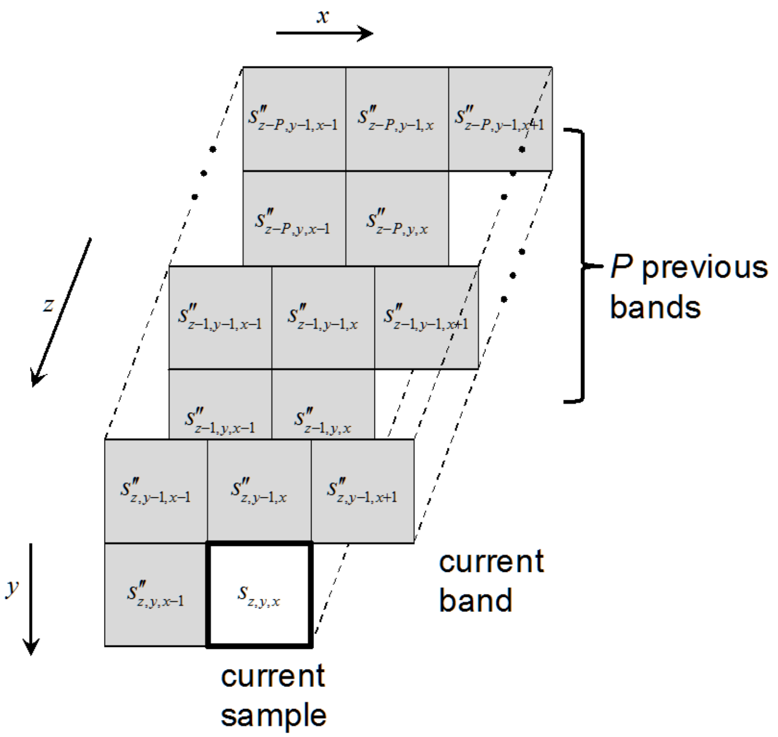


Figure 4-1: Typical Prediction Neighborhood

Within each spectral band, the predictor computes a *local sum* of neighboring sample representative values (see 4.4). Each such local sum is used to compute a *local difference* (see 4.5). Predicted sample values are calculated using the local sum in the current spectral band and a weighted sum of local difference values from the current and previous spectral bands (see 4.7). The *weights* (see 4.6) used in this calculation are adaptively updated (see 4.10) following the calculation of each predicted sample value. Each prediction residual, that is, the difference between a given sample value $s_{z,y,x}$ and the corresponding predicted sample value $\hat{s}_{z,y,x}$, is quantized (see 4.8) and then mapped to an unsigned integer $\delta_{z,y,x}$, the mapped quantizer index (see 4.11). The quantized value of sample $s_{z,y,x}$ is used to calculate a corresponding *sample representative* value $s''_{z,y,x}$ (see 4.9).

The local sum $\sigma_{z,y,x}$ (see 4.4) is a weighted sum of sample representatives in spectral band z that are adjacent to sample $s_{z,y,x}$. Figure 4-2 illustrates the sample representatives used to calculate the local sum. A user may choose to perform prediction using *neighbor-oriented* or *column-oriented* local sums for an image, and local sums may be *wide* or *narrow*. When neighbor-oriented local sums are used, the local sum is equal to a combination of up to four neighboring sample representative values in the spectral band (except when $y = 0$, $x = 0$, or $x = N_X - 1$, in which case these four values are not all available, and the local sum calculation is suitably modified, as detailed in 4.4). When column-oriented local sums are used, the local sum is equal to four times the neighboring sample representative value in the previous row (except when $y = 0$, in which case this value is not available and the local sum calculation is suitably modified as detailed in 4.4). Narrow local sums are defined to eliminate the dependency on sample representative $s''_{z,y,x-1}$ when calculating $\sigma_{z,y,x}$, which may facilitate pipelining in a hardware implementation.

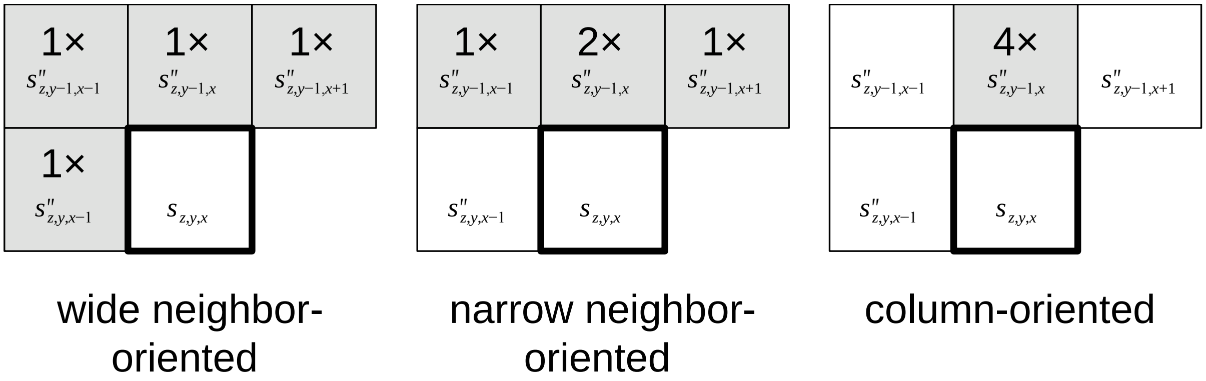


Figure 4-2: Samples Used to Calculate Local Sums

The local sums are used to calculate local difference values. In each spectral band, the *central local difference*, $d_{z,y,x}$, is equal to the difference between the local sum $\sigma_{z,y,x}$ and four times the sample representative value $s''_{z,y,x}$ (see 4.5.1). The three *directional local differences*, $d^{\mathrm{N}}_{z,y,x}$, $d^{\mathrm{W}}_{z,y,x}$, and $d^{\mathrm{NW}}_{z,y,x}$, are each equal to the difference between $\sigma_{z,y,x}$ and four times a sample value labeled as 'N', 'W', or 'NW' in figure 4-3 (except when this sample value is not available, that is, at image edges, as detailed in 4.5.2).

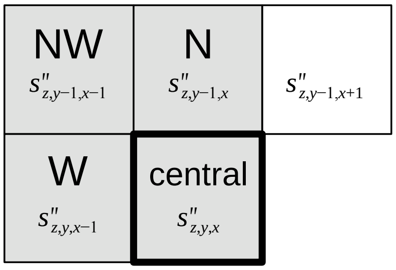


Figure 4-3: Computing Local Differences in a Spectral Band

A user may choose to perform prediction for an image in *full* or *reduced* mode (see 4.3). Under reduced mode, prediction depends on a weighted sum of the central local differences computed in preceding bands; the directional local differences are not used, and thus need not be calculated, under reduced mode. Under full mode, prediction depends on a weighted sum of the central local differences computed in preceding bands and the three directional local differences computed in the current band.

As described in reference [D1], the use of reduced mode in combination with column-oriented local sums tends to yield smaller compressed image data volumes for raw (uncalibrated) input images from push-broom imagers that exhibit significant along-track streaking artifacts. The use of full mode in combination with neighbor-oriented local sums tends to yield smaller compressed image data volumes for whiskbroom imagers, frame imagers, and calibrated imagery.

The prediction residual, the difference between the sample value $s_{z,y,x}$ and the predicted sample value $\hat{s}_{z,y,x}$, is quantized (see 4.8), and the quantizer index is mapped to an unsigned integer $\delta_{z,y,x}$ (see 4.11). This mapping is invertible, so that the decompressor can exactly reconstruct the quantizer index. User-specified *absolute* and/or *relative error limit* values (see 4.8) control the *maximum error* value $m_z(t)$ for each sample. Reconstruction of sample $s_{z,y,x}$ with at most $m_z(t)$ units of error can be achieved by a decompressor. However, the Recommended Standard imposes no specific requirements on reconstructing a sample by a decompressor, and minimizing the maximum reconstruction error of each sample may not minimize other distortion metrics (see reference [D1] for an example).

4.2 NUMBER OF BANDS FOR PREDICTION

The user-specified parameter P, which shall be an integer in the range $0 \leq P \leq 15$, determines the number of preceding spectral bands used for prediction. Specifically, prediction in spectral band z depends on central local differences, defined in 4.5.1, computed in bands $z-1, z-2, \ldots, z-P_z^*$, where

$$P_z^* = \min\{z, P\}. \tag{18}$$

4.3 FULL AND REDUCED PREDICTION MODES

4.3.1 A user may choose to perform prediction using *full* or *reduced* mode for an image, except when the image has width of one (i.e., $N_X = 1$), in which case reduced mode shall be used.

4.3.2 Under both full and reduced modes, prediction in spectral band z makes use of central local differences from the preceding P_z^* spectral bands. Under full prediction mode, prediction in spectral band z additionally makes use of three directional local differences, defined in 4.5.2, computed in the current spectral band z. Thus the number of local difference values used for prediction at each sample in band z, denoted C_z, is

$$C_z = \begin{cases} P_z^*, & \text{reduced prediction mode} \\ P_z^* + 3, & \text{full prediction mode} \end{cases} . \tag{19}$$

4.4 LOCAL SUM

4.4.1 The *local sum* $\sigma_{z,y,x}$ is an integer equal to a weighted sum of previous sample representative values in band z that are neighbors of sample $s_{z,y,x}$. A user may choose to perform prediction using *neighbor-oriented* or *column-oriented* local sums for an image, except when the image has width 1 (i.e., $N_X = 1$), in which case column-oriented local sums shall be used. In either case, a user may choose to use *wide* or *narrow* local sums.

NOTE – Column-oriented local sums are not suggested under full prediction mode.

4.4.2 When wide neighbor-oriented local sums are used, $\sigma_{z,y,x}$ is defined as

$$\sigma_{z,y,x} = \begin{cases} s''_{z,y,x-1} + s''_{z,y-1,x-1} + s''_{z,y-1,x} + s''_{z,y-1,x+1}, & y > 0,\ 0 < x < N_X - 1 \\ 4s''_{z,y,x-1}, & y = 0,\ x > 0 \\ 2\left(s''_{z,y-1,x} + s''_{z,y-1,x+1}\right), & y > 0,\ x = 0 \\ s''_{z,y,x-1} + s''_{z,y-1,x-1} + 2s''_{z,y-1,x}, & y > 0,\ x = N_X - 1 \end{cases} ; \tag{20}$$

when narrow neighbor-oriented local sums are used, $\sigma_{z,y,x}$ is defined as

$$\sigma_{z,y,x} = \begin{cases} s''_{z,y-1,x-1} + 2s''_{z,y-1,x} + s''_{z,y-1,x+1}, & y > 0,\ 0 < x < N_X - 1 \\ 4s''_{z-1,y,x-1}, & y = 0, x > 0,\ z > 0 \\ 2\left(s''_{z,y-1,x} + s''_{z,y-1,x+1}\right), & y > 0,\ x = 0 \\ 2\left(s''_{z,y-1,x-1} + s''_{z,y-1,x}\right), & y > 0,\ x = N_X - 1 \\ 4s_{\text{mid}}, & y = 0, x > 0, z = 0 \end{cases} ; \tag{21}$$

when wide column-oriented local sums are used, $\sigma_{z,y,x}$ is defined as

$$\sigma_{z,y,x} = \begin{cases} 4s''_{z,y-1,x}, & y > 0 \\ 4s''_{z,y,x-1}, & y = 0,\ x > 0 \end{cases}; \tag{22}$$

and when narrow column-oriented local sums are used, $\sigma_{z,y,x}$ is defined as

$$\sigma_{z,y,x} = \begin{cases} 4s''_{z,y-1,x}, & y > 0 \\ 4s''_{z-1,y,x-1}, & y = 0,\ x > 0,\ z > 0 \\ 4s_{mid}, & y = 0,\ x > 0,\ z = 0 \end{cases}, \tag{23}$$

where sample representative values $s''_{z,y,x}$ are defined in 4.9.

NOTE – The value of $\sigma_{z,0,0}$ is not defined, as it is not needed.

4.5 LOCAL DIFFERENCES

4.5.1 CENTRAL LOCAL DIFFERENCE

When x and y are not both zero (i.e., when $t > 0$), the central local difference $d_{z,y,x}$ is defined as

$$d_{z,y,x} = 4\,s''_{z,y,x} - \sigma_{z,y,x}\,. \tag{24}$$

4.5.2 DIRECTIONAL LOCAL DIFFERENCES

When x and y are not both zero (i.e., when $t > 0$), the three directional local differences are defined as

$$d^{\mathrm{N}}_{z,y,x} = \begin{cases} 4s''_{z,y-1,x} - \sigma_{z,y,x}, & y > 0 \\ 0, & y = 0 \end{cases}, \tag{25}$$

$$d^{\mathrm{W}}_{z,y,x} = \begin{cases} 4s''_{z,y,x-1} - \sigma_{z,y,x}, & x > 0,\ y > 0 \\ 4s''_{z,y-1,x} - \sigma_{z,y,x}, & x = 0,\ y > 0 \\ 0, & y = 0 \end{cases}, \text{ and} \tag{26}$$

$$d^{\mathrm{NW}}_{z,y,x} = \begin{cases} 4s''_{z,y-1,x-1} - \sigma_{z,y,x}, & x > 0,\ y > 0 \\ 4s''_{z,y-1,x} - \sigma_{z,y,x}, & x = 0,\ y > 0 \\ 0, & y = 0 \end{cases}. \tag{27}$$

NOTE – Directional local differences are not used under reduced prediction mode.

4.5.3 LOCAL DIFFERENCE VECTOR

For $t > 0$, the local difference vector $\mathbf{U}_z(t)$ is a vector of the C_z local difference values used to calculate the predicted sample value $\hat{s}_z(t)$. Under full prediction mode, $\mathbf{U}_z(t)$ is defined as

$$\mathbf{U}_z(t) = \begin{bmatrix} d_z^{\mathrm{N}}(t) \\ d_z^{\mathrm{W}}(t) \\ d_z^{\mathrm{NW}}(t) \\ d_{z-1}(t) \\ d_{z-2}(t) \\ \vdots \\ d_{z-P_z^*}(t) \end{bmatrix}, \tag{28}$$

and under reduced prediction mode, for $z > 0$, $\mathbf{U}_z(t)$ is defined as

$$\mathbf{U}_z(t) = \begin{bmatrix} d_{z-1}(t) \\ d_{z-2}(t) \\ \vdots \\ d_{z-P_z^*}(t) \end{bmatrix}. \tag{29}$$

NOTE – Under reduced mode, $\mathbf{U}_0(t)$ is not defined, as it is not needed.

4.6 WEIGHTS

4.6.1 WEIGHT VALUES AND WEIGHT RESOLUTION

4.6.1.1 In the prediction calculation (see 4.7), for $t > 0$, each component of the local difference vector $\mathbf{U}_z(t)$ is multiplied by a corresponding integer *weight value*.

4.6.1.2 The resolution of the weight values is controlled by the user-specified parameter Ω, which shall be an integer in the range $4 \leq \Omega \leq 19$.

4.6.1.3 Each weight value is a signed integer quantity that can be represented using $\Omega + 3$ bits. Thus each weight value has minimum and maximum possible values $\omega_{\min}$ and $\omega_{\max}$, respectively, where

$$\omega_{\min} = -2^{\Omega+2}, \ \omega_{\max} = 2^{\Omega+2} - 1. \tag{30}$$

NOTE – Increasing the number of bits used to represent weight values (i.e., using a larger value of Ω) provides increased resolution in the prediction calculation. This Recommended Standard does not address the tradeoffs associated with selecting the value of Ω. Reference [D1] presents some examples.

4.6.2 WEIGHT VECTOR

The weight vector $\mathbf{W}_z(t)$ is a vector of the C_z weight values used in prediction. Under full prediction mode,

$$\mathbf{W}_z(t) = \begin{bmatrix} \omega_z^{\mathrm{N}}(t) \\ \omega_z^{\mathrm{W}}(t) \\ \omega_z^{\mathrm{NW}}(t) \\ \omega_z^{(1)}(t) \\ \omega_z^{(2)}(t) \\ \vdots \\ \omega_z^{(P_z^*)}(t) \end{bmatrix}, \tag{31}$$

and under reduced prediction mode, for $z > 0$,

$$\mathbf{W}_z(t) = \begin{bmatrix} \omega_z^{(1)}(t) \\ \omega_z^{(2)}(t) \\ \vdots \\ \omega_z^{(P_z^*)}(t) \end{bmatrix}, \tag{32}$$

where the weight values are calculated as specified in 4.6.3 and 4.10.

NOTE – Under reduced mode, $\mathbf{W}_0(t)$ is not defined as it is not needed.

4.6.3 INITIALIZATION

4.6.3.1 General

A user may choose to use either *default* or *custom* weight initialization, defined below, to select the initial weight vector $\mathbf{W}_z(1)$ for each spectral band z. The same weight initialization method shall be used for all spectral bands.

4.6.3.2 Default Weight Initialization

4.6.3.2.1 When default weight initialization is used, for each spectral band *z*, initial weight vector components $\omega_z^{(1)}(1)$, $\omega_z^{(2)}(1)$, …, $\omega_z^{(P_z^*)}(1)$, shall be assigned values

$$\omega_z^{(1)}(1) = \frac{7}{8}2^{\Omega}, \quad \omega_z^{(i)}(1) = \left\lfloor \frac{1}{8}\omega_z^{(i-1)}(1) \right\rfloor, i = 2, 3, \ldots, P_z^* . \tag{33}$$

4.6.3.2.2 With this option, under full prediction mode the remaining components of $\mathbf{W}_z(1)$ shall be assigned values

$$\omega_z^{\mathrm{N}}(1) = \omega_z^{\mathrm{W}}(1) = \omega_z^{\mathrm{NW}}(1) = 0 . \tag{34}$$

4.6.3.3 Custom Weight Initialization

4.6.3.3.1 When custom weight initialization is used, for each spectral band *z*, the initial weight vector $\mathbf{W}_z(1)$ shall be assigned using a user-specified *weight initialization vector* Λ_z, consisting of C_z signed *Q*-bit integer components.

NOTES

1 The weight initialization vector Λ_z may be encoded in the header as described in 5.3.

2 A weight initialization vector Λ_z might be selected based on instrument characteristics or training data, or might be selected based on a weight vector from a previous compressed image.

4.6.3.3.2 The weight initialization resolution *Q* shall be a user-specified integer in the range $3 \le Q \le \Omega + 3$ bits.

4.6.3.3.3 The initial weight vector $\mathbf{W}_z(1)$ shall be calculated from Λ_z by

$$\mathbf{W}_z(1) = 2^{\Omega+3-Q}\Lambda_z + \left\lceil 2^{\Omega+2-Q} - 1 \right\rceil \mathbf{1} , \tag{35}$$

where **1** denotes a vector of all ‘ones’.

NOTE – In the (Ω + 3)-bit two’s complement representation of each component of $\mathbf{W}_z(1)$, the *Q* MSBs are equal to the binary representation of the corresponding component of Λ_z. The remaining bits, if any, are made up of a ‘0’ bit followed by ‘1’ bits in the remaining positions.

4.7 PREDICTION CALCULATION

4.7.1 For $t > 0$, the predicted central local difference $\hat{d}_z(t)$ is equal to the inner product of vectors $\mathbf{W}_z(t)$ and $\mathbf{U}_z(t)$:

$$\hat{d}_z(t) = \mathbf{W}_z^{\mathrm{T}}(t)\mathbf{U}_z(t), \tag{36}$$

except for $z = 0$ under reduced mode, in which case $\hat{d}_z(t) = 0$.

4.7.2 The high-resolution predicted sample value, $\breve{s}_z(t)$, is calculated as

$$\breve{s}_z(t) = \mathrm{clip}\left(\mathrm{mod}_R^*\left[\hat{d}_z(t) + 2^{\Omega}\left(\sigma_z(t) - 4s_{\mathrm{mid}}\right)\right] + 2^{\Omega+2}s_{\mathrm{mid}} + 2^{\Omega+1}, \left\{2^{\Omega+2}s_{\mathrm{min}}, 2^{\Omega+2}s_{\mathrm{max}} + 2^{\Omega+1}\right\}\right), \tag{37}$$

where the user-selected register size parameter R shall be an integer in the range $\max\{32, D + \Omega + 2\} \le R \le 64$.

NOTE – Increasing the register size R reduces the chance of an overflow occurring in the calculation of a high-resolution predicted sample value. This Recommended Standard does not address the tradeoffs associated with selecting the value of R. Reference [D1] provides some discussion.

4.7.3 The double-resolution predicted sample value is

$$\tilde{s}_z(t) = \begin{cases} \left\lfloor \dfrac{\breve{s}_z(t)}{2^{\Omega+1}} \right\rfloor, & t > 0 \\ 2s_{z-1}(t), & t = 0, P > 0, z > 0 \\ 2s_{\mathrm{mid}}, & t = 0 \text{ and } (P = 0 \text{ or } z = 0) \end{cases}. \tag{38}$$

4.7.4 The predicted sample value $\hat{s}_z(t)$ is defined as

$$\hat{s}_z(t) = \left\lfloor \frac{\tilde{s}_z(t)}{2} \right\rfloor. \tag{39}$$

4.8 QUANTIZATION

4.8.1 QUANTIZER OUTPUT

The prediction residual $\Delta_z(t)$ is the difference between the predicted and actual sample values,

$$\Delta_z(t) = s_z(t) - \hat{s}_z(t)\,. \tag{40}$$

The prediction residual shall be quantized using a uniform quantizer with step size $2m_z(t) + 1$, producing as quantizer output the signed integer *quantizer index* $q_z(t)$, defined as

$$q_z(t) = \begin{cases} \Delta_z(0), & t = 0 \\ \operatorname{sgn}(\Delta_z(t)) \left\lfloor \dfrac{|\Delta_z(t)| + m_z(t)}{2m_z(t)+1} \right\rfloor, & t > 0 \end{cases}, \tag{41}$$

where the *maximum error* value $m_z(t)$ is determined via user-specified quantizer fidelity settings as specified in 4.8.2.

NOTE – Given $q_z(t)$, reconstruction of sample $s_z(t)$ with no more than $m_z(t)$ units of error is possible. Thus lossless compression is achieved for this sample when $m_z(t) = 0$.

4.8.2 FIDELITY CONTROL

4.8.2.1 Controlling Maximum Error

4.8.2.1.1 For a given image, a user may choose the quantizer fidelity control method to be *lossless*, in which case

$$m_z(t) = 0 \tag{42}$$

for all z and t. Otherwise, the user may control the maximum error value $m_z(t)$ by specifying an *absolute error limit* a_z for each z, a *relative error limit* r_z for each z, or both.

NOTE – Restrictions on allowed error limit values are specified in 4.8.2.2.

4.8.2.1.2 When only absolute error limits are used, the maximum error shall be computed as

$$m_z(t) = a_z \tag{43}$$

for all z and t; when only relative error limits are used,

$$m_z(t) = \left\lfloor \frac{r_z\,|\hat{s}_z(t)|}{2^D} \right\rfloor \tag{44}$$

for all *z* and *t*; and when both absolute and relative error limits are used,

$$m_z(t) = \min\left(a_z, \left\lfloor \frac{r_z \left| \hat{s}_z(t) \right|}{2^D} \right\rfloor \right) \tag{45}$$

for all *z* and *t*.

4.8.2.2 Allowed Error Limit Values

4.8.2.2.1 If absolute error limits are used, then for each spectral band *z*, the value of a_z shall be an integer in the range $0 \le a_z \le 2^{D_\mathrm{A}} - 1$, where the user-specified *absolute error limit bit depth* D_A shall be an integer in the range $1 \le D_\mathrm{A} \le \min\{D - 1, 16\}$.

4.8.2.2.2 If relative error limits are used, then for each spectral band *z*, the value of r_z shall be an integer in the range $0 \le r_z \le 2^{D_\mathrm{R}} - 1$, where the user-specified *relative error limit bit depth* D_R shall be an integer in the range $1 \le D_\mathrm{R} \le \min\{D - 1, 16\}$.

4.8.2.3 Error Limit Assignment Methods

4.8.2.3.1 If used, absolute error limits shall be either (a) *band-dependent*, in which case the user shall specify a set of absolute error limit values $\{a_z\}_{z=0}^{N_Z-1}$, or (b) *band-independent*, in which case $a_z = A^*$ for each spectral band *z*, where A^* shall be the user-specified integer *absolute error limit constant*, satisfying $0 \le A^* \le 2^{D_\mathrm{A}} - 1$.

4.8.2.3.2 If used, relative error limits shall be either (a) *band-dependent*, in which case the user shall specify a set of relative error limit values $\{r_z\}_{z=0}^{N_Z-1}$, or (b) *band-independent*, in which case $r_z = R^*$ for each spectral band *z*, where R^* shall be the user-specified integer *relative error limit constant*, satisfying $0 \le R^* \le 2^{D_\mathrm{R}} - 1$.

NOTE – When both absolute and relative error limits are used for an image, the choice of assignment methods for relative and absolute error limits need not be the same. That is, band-independent absolute error limits may be used in combination with band-dependent relative error limits, and vice-versa.

4.8.2.4 Periodic Error Limit Updating

4.8.2.4.1 When used, error limit values may be fixed for an entire image, or the user may choose to use *periodic error limit updating*, in which case error limit values are periodically updated.

4.8.2.4.2 When periodic error limit updating is used, the user shall provide error limit values every 2^u frames, where the user-specified *error limit update period exponent u* shall be an integer in the range $0 \le u \le 9$.

4.8.2.4.3 All other quantizer fidelity settings (choice to use absolute and/or relative error limits, choice between band-dependent and band-independent assignment methods for the error limit method[s] in use, and error limit bit depth[s]) shall be fixed for the entire image.

4.8.2.4.4 Periodic error limit updating shall not be used with Band-SeQuential (BSQ) input order (defined in 5.4.2.3).

4.9 SAMPLE REPRESENTATIVES

4.9.1 Sample representatives are calculated using user-specified *resolution* parameter Θ, which shall be an integer in the range $0 \le \Theta \le 4$, and for each spectral band z, parameters *damping*, ϕ_z, and *offset*, ψ_z.

4.9.1.1 Each ϕ_z shall be a user-specified integer in the range $0 \le \phi_z \le 2^\Theta - 1$.

4.9.1.2 Each ψ_z shall be a user-specified integer in the range $0 \le \psi_z \le 2^\Theta - 1$, unless lossless fidelity control is used, in which case $\psi_z = 0$.

4.9.2 The sample representative $s''_z(t)$, which has the same resolution as the original samples, shall be calculated as

$$s''_z(t) = \begin{cases} s_z(0), & t = 0 \\ \left\lfloor \dfrac{\tilde{s}''_z(t) + 1}{2} \right\rfloor, & t > 0 \end{cases} \tag{46}$$

from the double-resolution sample representative

$$\tilde{s}''_z(t) = \left\lfloor \frac{4\left(2^\Theta - \phi_z\right) \cdot \left(s'_z(t) \cdot 2^\Omega - \operatorname{sgn}(q_z(t)) \cdot m_z(t) \cdot \psi_z \cdot 2^{\Omega - \Theta}\right) + \phi_z \cdot \breve{s}_z(t) - \phi_z \cdot 2^{\Omega+1}}{2^{\Omega+\Theta+1}} \right\rfloor, \tag{47}$$

where $\breve{s}_z(t)$ is the high-resolution predicted sample value defined in 4.7.2, and

$$s'_z(t) = \mathrm{clip}\left(\hat{s}_z(t) + q_z(t)\left(2m_z(t)+1\right), \{s_{\min}, s_{\max}\}\right) \tag{48}$$

is a clipped version of the quantizer bin center.

NOTES

1 Reconstructing sample $s_z(t)$ with value $s'_z(t)$ by the decompressor ensures that reconstruction error will be at most $m_z(t)$. If $m_z(t) = 0$ then $s'_z(t) = s_z(t)$.

2 Setting $\phi_z = \psi_z = 0$ causes the sample representative $s''_z(t)$ to be equal to $s'_z(t)$.

3 The difference between the sample representative $s''_z(t)$ and the predicted sample value $\hat{s}_z(t)$ may exceed $m_z(t)$.

4.10 WEIGHT UPDATE

4.10.1 The double-resolution prediction error $e_z(t)$ is an integer defined as

$$e_z(t) = 2s'_z(t) - \tilde{s}_z(t). \tag{49}$$

4.10.2 For $t > 0$, the weight update scaling exponent $\rho(t)$ is an integer defined as

$$\rho(t) = \mathrm{clip}\left(v_{\min} + \left\lfloor \frac{t - N_X}{t_{\mathrm{inc}}} \right\rfloor, \ \left\{v_{\min}, v_{\max}\right\}\right) + D - \Omega, \tag{50}$$

where user-specified integer parameters $v_{\min}$, $v_{\max}$, and t_{inc} are constrained as follows:

a) The values of $v_{\min}$ and $v_{\max}$ shall be integers in the range $-6 \le v_{\min} \le v_{\max} \le 9$.

b) The weight update scaling exponent change interval t_{inc} shall be a power of 2 in the range $2^4 \le t_{\mathrm{inc}} \le 2^{11}$.

NOTE – These parameters control the rate at which weights adapt to image data statistics. The initial weight update scaling exponent is $\rho(1) = v_{\min} + D - \Omega$, and at regular intervals determined by the value of t_{inc}, $\rho(t)$ is incremented by one until reaching a final value $v_{\max} + D - \Omega$. Smaller values of $\rho(t)$ produce larger weight increments, yielding faster adaptation to source statistics but worse steady-state compression performance.

4.10.3 For $t > 0$, following the calculation of $\tilde{s}_z(t)$, components of the next weight vector in the spectral band, $\mathbf{W}_z(t+1)$, are defined as

$$\omega_z^{(i)}(t+1) = \operatorname{clip}\left(\omega_z^{(i)}(t) + \left\lfloor \frac{1}{2}\left(\operatorname{sgn}^+\left[e_z(t)\right] \cdot 2^{-\left(\rho(t)+\varsigma_z^{(i)}\right)} \cdot d_{z-i}(t) + 1 \right) \right\rfloor, \left\{\omega_{\min}, \omega_{\max}\right\} \right), \tag{51}$$

and, when full prediction mode is used, for the directional components,

$$\omega_z^{\mathrm{N}}(t+1) = \operatorname{clip}\left(\omega_z^{\mathrm{N}}(t) + \left\lfloor \frac{1}{2}\left(\operatorname{sgn}^+\left[e_z(t)\right] \cdot 2^{-\left(\rho(t)+\varsigma_z^*\right)} \cdot d_z^{\mathrm{N}}(t) + 1 \right) \right\rfloor, \left\{\omega_{\min}, \omega_{\max}\right\} \right), \tag{52}$$

$$\omega_z^{\mathrm{W}}(t+1) = \operatorname{clip}\left(\omega_z^{\mathrm{W}}(t) + \left\lfloor \frac{1}{2}\left(\operatorname{sgn}^+\left[e_z(t)\right] \cdot 2^{-\left(\rho(t)+\varsigma_z^*\right)} \cdot d_z^{\mathrm{W}}(t) + 1 \right) \right\rfloor, \left\{\omega_{\min}, \omega_{\max}\right\} \right), \tag{53}$$

$$\omega_z^{\mathrm{NW}}(t+1) = \operatorname{clip}\left(\omega_z^{\mathrm{NW}}(t) + \left\lfloor \frac{1}{2}\left(\operatorname{sgn}^+\left[e_z(t)\right] \cdot 2^{-\left(\rho(t)+\varsigma_z^*\right)} \cdot d_z^{\mathrm{NW}}(t) + 1 \right) \right\rfloor, \left\{\omega_{\min}, \omega_{\max}\right\} \right). \tag{54}$$

4.10.4 The inter-band weight exponent offsets $\varsigma_z^{(i)}$, for $z = 0,\ldots, N_{\mathrm{Z}} - 1$ and $i = 1,\ldots,P_z^*$, and intra-band weight exponent offsets ς_z^* shall be user-specified integers in the range $-6 \le \varsigma_z^{(i)} \le 5$ and $-6 \le \varsigma_z^* \le 5$, respectively.

NOTE – The quantity $\left\lfloor \frac{1}{2}\left(\operatorname{sgn}^+\left[e_z(t)\right] \cdot 2^{-(\rho(t)+\zeta)} \cdot d + 1 \right) \right\rfloor$ is equivalent to $\left\lfloor \frac{1}{2}\left(\left\lfloor \operatorname{sgn}^+\left[e_z(t)\right] \cdot 2^{-(\rho(t)+\zeta)} \cdot d \right\rfloor + 1 \right) \right\rfloor$ but is not in general equivalent to $\left\lfloor \frac{1}{2}\left(\operatorname{sgn}^+\left[e_z(t)\right] \cdot \left\lfloor 2^{-(\rho(t)+\zeta)} \cdot d \right\rfloor + 1 \right) \right\rfloor$.

4.11 MAPPED QUANTIZER INDEX

The signed quantizer index $q_z(t)$ is converted to an unsigned *mapped quantizer index* $\delta_z(t)$ defined as

$$\delta_z(t) = \begin{cases} \left|q_z(t)\right| + \theta_z(t), & \left|q_z(t)\right| > \theta_z(t) \\ 2\left|q_z(t)\right|, & 0 \le (-1)^{\tilde{s}_z(t)} q_z(t) \le \theta_z(t) \\ 2\left|q_z(t)\right| - 1, & \text{otherwise} \end{cases}, \tag{55}$$

where

$$\theta_z(t) = \begin{cases} \min\left\{ \hat{s}_z(0) - s_{\min}, s_{\max} - \hat{s}_z(0) \right\} & t = 0 \\ \min\left\{ \left\lfloor \dfrac{\hat{s}_z(t) - s_{\min} + m_z(t)}{2m_z(t)+1} \right\rfloor, \left\lfloor \dfrac{s_{\max} - \hat{s}_z(t) + m_z(t)}{2m_z(t)+1} \right\rfloor \right\}, & t > 0 \end{cases}. \tag{56}$$

NOTE – Each mapped quantizer index $\delta_z(t)$ can be represented as a *D*-bit unsigned integer.

5 ENCODER

5.1 OVERVIEW

This section specifies the encoding stage of the compressor and the format of a compressed image. Quantities defined in this section are summarized in table E-3 of annex E.

A compressed image consists of a *header* followed by a *body*.

The variable-length header, defined in 5.3, encodes image and compression parameters.

The body, defined in 5.4, losslessly encodes mapped quantizer indices $\delta_{z,y,x}$ from the predictor. If the periodic error limit updating option is used (see 4.8.2.4), then error limit values are also periodically encoded as part of the body.

A user can choose to perform encoding using the *sample-adaptive* entropy coding approach specified in 5.4.3.2, the *hybrid* approach specified in 5.4.3.3, or the *block-adaptive* approach specified in 5.4.3.4.

The sample-adaptive and block-adaptive entropy coding approaches are the same as the ones specified in reference [D2]. They are generally effective for lossless compression, but the ability to use near-lossless compression under the present Recommended Standard tends to yield mapped quantizer indices having a lower-entropy distribution. The hybrid encoding approach tends to provide more effective encoding for lower-entropy distributions. Examples and comparisons can be found in reference [D1].

Under the sample-adaptive entropy coding approach, each mapped quantizer index is encoded using a variable-length binary codeword from a family of codes. Which member of this family is used is adaptively selected based on statistics that are updated after each sample is encoded; separate statistics are maintained for each spectral band, and the compressed image size does not depend on the order in which mapped quantizer indices are encoded.

Like the sample-adaptive coder, the hybrid entropy coding approach uses similar adaptive code selection statistics. It includes codes equivalent to those used by the sample-adaptive encoder, but augmented with an additional 16 variable-to-variable length ‘low-entropy’ codes. A single output codeword from a low-entropy code may encode multiple mapped quantizer indices, allowing lower compressed data rates than can be achieved by the ‘high-entropy’ codes. The order in which mapped quantizer indices are encoded has virtually no impact on compressed image size.

The block-adaptive entropy coding approach relies on the lossless data compressor defined in reference [1]. Under this approach, the sequence of mapped quantizer indices is partitioned into short blocks, and the encoding method used is independently and adaptively selected for each block. Depending on the encoding order, the mapped quantizer indices in a block may be from the same or different spectral bands, and thus the compressed image size depends on the encoding order when this approach is used.

5.2 GENERAL

5.2.1 A compressed image shall consist of a variable-length *header*, defined in 5.3, followed by a variable-length *body*, defined in 5.4.

NOTE – Figure 5-1 depicts the structure of a compressed image.

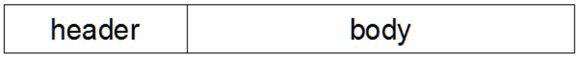


Figure 5-1: Compressed Image Structure

5.2.2 The user-selected *output word size*, measured in bytes, shall be an integer B in the range $1 \leq B \leq 8$.

NOTE – Fill bits are included in the body (as specified in 5.4.3.2.4.4, 5.4.3.4.3.2, and 5.4.3.3.5.4.5) when needed to ensure that the size of the compressed image is a multiple of the output word size.

5.3 HEADER

5.3.1 GENERAL

5.3.1.1 The header of a compressed image shall have the structure specified in table 5-1.

Table 5-1: Header Structure

Part	Status	Size	Reference
Image Metadata	Mandatory	Variable	5.3.2
Predictor Metadata	Mandatory	Variable	5.3.3
Entropy Coder Metadata	Mandatory	Variable	5.3.4

NOTES

1 The length of each header part varies depending on compression options selected by the user.

2 Each header part consists of an integer number of bytes. The header length is not necessarily a multiple of the output word size.

5.3.1.2 Whenever fill bits are included in a header element, fill bits shall be all zeros.

5.3.2 IMAGE METADATA

5.3.2.1 Header

The Image Metadata header part shall have the structure specified in table 5-2.

Table 5-2: Image Metadata Structure

Subpart	Status	Size (Bytes)	Reference
Essential	Mandatory	12	5.3.2.2
Supplementary Information Tables	Optional	Variable	5.3.2.3

5.3.2.2 Essential

The Essential subpart shall have the structure specified in table 5-3.

Table 5-3: Essential Subpart Structure

Field	Width (bits)	Description	Reference
User-Defined Data	8	The user may assign the value of this field arbitrarily, for example, to indicate the value of a user-defined index of the image within a sequence of images.	
X Size	16	The value N_X encoded mod 2^{16} as a 16-bit unsigned binary integer.	3.2
Y Size	16	The value N_Y encoded mod 2^{16} as a 16-bit unsigned binary integer.	3.2
Z Size	16	The value N_Z encoded mod 2^{16} as a 16-bit unsigned binary integer.	3.2
Sample Type	1	‘0’: image sample values are unsigned integers. ‘1’: image sample values are signed integers.	3.2.1
Reserved	1	This field shall have value ‘0’.	
Large Dynamic Range Flag	1	‘0’: dynamic range satisfies $D \leq 16$. ‘1’: dynamic range satisfies $D > 16$.	3.3
Dynamic Range	4	The value D mod 2^4 as a 4-bit unsigned binary integer.	3.3
Sample Encoding Order	1	‘0’: samples are encoded in band-interleaved order. ‘1’: samples are encoded in BSQ order.	5.4.2
Sub-Frame Interleaving Depth	16	When band-interleaved encoding order is used, this field shall contain the value M encoded mod 2^{16} as a 16-bit unsigned binary integer. When BSQ encoding order is used, this field shall be all ‘zeros’.	5.4.2.2
Reserved	2	This field shall have value ‘00’.	

Field	Width (bits)	Description	Reference
Output Word Size	3	The value B encoded mod 2^3 as a 3-bit unsigned binary integer.	5.2.2
Entropy Coder Type	2	‘00’: sample-adaptive entropy coder is used. ‘01’: hybrid entropy coder is used. ‘10’: block-adaptive entropy coder is used.	5.4.3
Reserved	1	This field shall have value ‘0’.	
Quantizer Fidelity Control Method	2	‘00’: lossless. ‘01’: absolute error limit only. ‘10’: relative error limit only. ‘11’: both absolute and relative error limits.	4.8.2.1
Reserved	2	This field shall contain all ‘zeros’.	
Supplementary Information Table Count	4	The value τ, encoded as a 4-bit unsigned integer.	3.5.2.1

5.3.2.3 Supplementary Information Tables

5.3.2.3.1 General

5.3.2.3.1.1 The Supplementary Information Tables subpart shall be present when the number of supplementary information tables, τ, is nonzero and shall be omitted otherwise.

5.3.2.3.1.2 When present, the Supplementary Information Tables subpart shall consist of a sequence of τ Supplementary Information Tables, each having the structure specified in table 5-4.

Table 5-4: Supplementary Information Table Structure

Field	Width (bits)	Description	Reference
Table Type	2	‘00’: unsigned integer. ‘01’: signed integer. ‘10’: float.	3.5.2.3
Reserved	2	This field shall have value ‘00’.	
Table Purpose	4	Table purpose value encoded as a 4-bit unsigned integer (see table 3-1).	3.5.2.2
Reserved	1	This field shall have value ‘0’.	
Table Structure	2	‘00’: zero-dimensional. ‘01’: one-dimensional. ‘10’: two-dimensional-*zx*. ‘11’: two-dimensional-*yx*.	3.5.2.4
Reserved	1	This field shall have value ‘0’.	
Supplementary User-Defined Data	4	The user may assign the value of this field arbitrarily.	
Table Data Subblock	(variable)	(See 5.3.2.3.2 below.)	3.5

5.3.2.3.2 Table Data Subblock

5.3.2.3.2.1 A Table Data subblock shall have the structure specified in 5.3.2.3.2.2 when the table type is unsigned or signed integer, and the structure specified in 5.3.2.3.2.3 when the table type is float.

5.3.2.3.2.2 If the Table Type is signed or unsigned integer, then the Table Data subblock shall consist of

a) the value of the table bit depth D_I encoded modulo 2^5 as a 5-bit unsigned integer (5 bits);

b) the sequence of table elements (D_I bits each):

   1) if the table structure is zero-dimensional, then the single value i is encoded;

   2) if the table structure is one-dimensional, then each i_z is encoded in order of increasing index z;

   3) if the table structure is two-dimensional-zx, then each $i_{z,x}$ is encoded in the order defined by the nesting of loops as follows:

```
for $z$ = 0 to $N_Z - 1$
    for $x$ = 0 to $N_X - 1$
        encode $i_{z,x}$;
```

   4) if the table structure is two-dimensional-yx, then each $i_{y,x}$ is encoded in the order defined by the nesting of loops as follows:

```
for $y$ = 0 to $N_Y - 1$
    for $x$ = 0 to $N_X - 1$
        encode $i_{y,x}$;
```

   5) each table element, i, i_z, $i_{z,x}$, or $i_{y,x}$, is encoded as a D_I-bit unsigned binary integer, or in two's complement representation for table types unsigned integer and signed integer, respectively; and

c) fill bits appended as needed to reach the next byte boundary.

5.3.2.3.2.3 If the Table Type is float, then the Table Data subblock shall consist of

a) the value of the significand bit depth D_F encoded as a 5-bit unsigned integer (5 bits);

b) the value of the exponent bit depth D_E encoded mod 2^3 as a 3-bit unsigned integer (3 bits);

c) the value of the exponent bias β encoded as an D_E-bit unsigned integer (D_E bits);

d) the sequence of table elements ($1 + D_F + D_E$ bits each):

1) if the table structure is zero-dimensional, then the single table element $\{b, \alpha, j\}$ is encoded;

2) if the table structure is one-dimensional, then table elements $\{b_z, \alpha_z, j_z\}$ are encoded in the following order:

```
for $z = 0$ to $N_Z - 1$
        encode $\{b_z, \alpha_z, j_z\}$;
```

3) if the table structure is two-dimensional-*zx*, then table elements $\{b_{z,x}, \alpha_{z,x}, j_{z,x}\}$ are encoded in the order defined by the nesting of loops as follows:

```
for $z = 0$ to $N_Z - 1$
        for $x = 0$ to $N_X - 1$
                encode $\{b_{z,x}, \alpha_{z,x}, j_{z,x}\}$;
```

4) if the table structure is two-dimensional-*yx*, then table elements $\{b_{y,x}, \alpha_{y,x}, j_{y,x}\}$ are encoded in the order defined by the nesting of loops as follows:

```
for $y = 0$ to $N_Y - 1$
        for $x = 0$ to $N_X - 1$
                encode $\{b_{y,x}, \alpha_{y,x}, j_{y,x}\}$;
```

5) for each table element $\{b, \alpha, j\}$, $\{b_z, \alpha_z, j_z\}$, $\{b_{z,x}, \alpha_{z,x}, j_{z,x}\}$, or $\{b_{y,x}, \alpha_{y,x}, j_{y,x}\}$, the following are encoded:

   i) the value of the sign bit (1 bit);

   ii) the value of the exponent, encoded as a D_E-bit unsigned integer (D_E bits);

   iii) the value of the significand, encoded as an D_F-bit unsigned integer (D_F bits); and

e) fill bits appended as needed to reach the next byte boundary.

5.3.3 PREDICTOR METADATA

5.3.3.1 Header

The Predictor Metadata header part shall have the structure specified in table 5-5.

Table 5-5: Predictor Metadata Structure

Subpart	Status	Size (Bytes)	Reference
Primary	Mandatory	5	5.3.3.2
Weight Tables	Optional	Variable	5.3.3.3
Quantization	Conditional	Variable	5.3.3.4
Sample Representative	Conditional	Variable	5.3.3.5

5.3.3.2 Primary

The Primary subpart shall have the structure specified in table 5-6.

Table 5-6: Primary Structure

Field	Width (bits)	Description	Reference
Reserved	1	This field shall have value ‘0’.	
Sample Representative Flag	1	‘0’: Sample Representative subpart is not included in Predictor Metadata header part; sample representatives use $\phi_z = \psi_z = 0$ for all spectral bands z. ‘1’: Sample Representative subpart is included in Predictor Metadata header part.	4.9
Number of Prediction Bands	4	The value P encoded as a 4-bit unsigned binary integer.	4.2
Prediction Mode	1	‘0’: full prediction mode is used. ‘1’: reduced prediction mode is used.	4.3
Weight Exponent Offset Flag	1	‘0’: all $\varsigma_z^{(i)}$ and ς_z^{*} values are zero. ‘1’: some $\varsigma_z^{(i)}$ and ς_z^{*} values may be nonzero.	4.10.3
Local Sum Type	2	‘00’: wide neighbor-oriented local sums are used. ‘01’: narrow neighbor-oriented local sums are used. ‘10’: wide column-oriented local sums are used. ‘11’: narrow column-oriented local sums are used.	4.4
Register Size	6	The value R encoded mod 2^6 as a 6-bit unsigned binary integer.	4.7.2
Weight Component Resolution	4	The value $(\Omega - 4)$ encoded as a 4-bit unsigned binary integer.	4.6.1
Weight Update Scaling Exponent Change Interval	4	The value $(\log_2 t_{\text{inc}} - 4)$ encoded as a 4-bit unsigned binary integer.	4.10.2
Weight Update Scaling Exponent Initial Parameter	4	The value $(\nu_{\min} + 6)$ encoded as a 4-bit unsigned binary integer.	4.10.2
Weight Update Scaling Exponent Final Parameter	4	The value $(\nu_{\max} + 6)$ encoded as a 4-bit unsigned binary integer.	4.10.2
Weight Exponent Offset Table Flag	1	‘0’: Weight Exponent Offset Table is not included in Predictor Metadata. ‘1’: Weight Exponent Offset Table is included in Weight Tables subpart of Predictor Metadata.	4.10.3
Weight Initialization Method	1	‘0’: default weight initialization is used. ‘1’: custom weight initialization is used.	4.6.3
Weight Initialization Table Flag	1	‘0’: Weight Initialization Table is not included in Predictor Metadata. ‘1’: Weight Initialization Table is included in Weight Tables subpart of Predictor Metadata.	4.6.3
Weight Initialization Resolution	5	When the default weight initialization is used, this field shall have value ‘00000’. Otherwise, this field shall contain the value Q encoded as a 5-bit unsigned binary integer.	4.6.3

5.3.3.3 Weight Tables

5.3.3.3.1 General

5.3.3.3.1.1 The Weight Tables subpart of the Predictor Metadata header part shall be present if the Weight Initialization Table Flag or Weight Exponent Offset Table Flag is set to ‘1’ and be omitted otherwise.

5.3.3.3.1.2 When present, the Weight Tables subpart shall have the structure specified in table 5-7.

Table 5-7: Weight Tables Subpart Structure

Block	Status	Size	Reference
Weight Initialization Table	Optional	Variable	5.3.3.3.2
Weight Exponent Offset Table	Optional	Variable	5.3.3.3.3

5.3.3.3.2 Weight Initialization Table

5.3.3.3.2.1 The optional Weight Initialization Table may be included only when the custom weight initialization method is selected. The presence of the Weight Initialization Table shall be indicated by setting the Weight Initialization Table Flag field to ‘1’.

NOTE – Even when the custom weight initialization option is used, the Weight Initialization Table may be omitted. For example, a mission might design a fixed set of custom weight initialization vectors for an instrument to be used throughout a mission and elect to not encode these vectors with each image.

5.3.3.3.2.2 When the Weight Initialization Table is included, the custom weight initialization vectors $\left\{\Lambda_z\right\}_{z=0}^{N_Z-1}$ shall be encoded, component-by-component, with each component encoded as a Q-bit signed two’s complement binary integer, in the order defined by the nesting of loops as follows:

```
for $z = 0$ to $N_Z - 1$
    for $j = 0$ to $C_z - 1$
        encode component $j$ of $\Lambda_z$.
```

5.3.3.3.2.3 Fill bits shall be appended to the Weight Initialization Table as needed to reach the next byte boundary.

5.3.3.3.3 Weight Exponent Offset Table

5.3.3.3.3.1 The optional Weight Exponent Offset Table may be included only when the Weight Exponent Offset Flag field is ‘1’. The presence of the optional Weight Exponent Offset Table shall be indicated by setting the Weight Exponent Offset Table Flag field to ‘1’.

NOTE – Even when nonzero values of $\varsigma_z^{(i)}$ and ς_z^* are used, the Weight Exponent Offset Table might be omitted. For example, a mission might design a fixed set of custom weight exponent offsets for an instrument to be used throughout a mission and elect to not encode these vectors with each image.

5.3.3.3.3.2 When the Weight Exponent Offset Table is included, the inter-band weight exponent offsets $\varsigma_z^{(i)}$ and inter-band weight exponent offsets ς_z^*, for $z = 0,\ldots, N_Z - 1$ and $i = 1,\ldots, P_z^*$, shall be encoded, component-by-component, with each component encoded as a 4-bit signed two’s complement binary integer, in the order defined by the nesting of loops as follows:

```
for $z$ = 0 to $N_Z - 1$
    if full prediction mode is used
        encode $\varsigma_z^*$
    for $i$ = 1 to $P_z^*$
        encode $\varsigma_z^{(i)}$.
```

5.3.3.3.3.3 Fill bits shall be appended to the Weight Exponent Offset Table as needed to reach the next byte boundary.

5.3.3.4 Quantization

5.3.3.4.1 General

5.3.3.4.1.1 The Quantization subpart shall be included unless the quantizer fidelity control method is lossless (see 4.8.2.1), in which case it shall be omitted.

5.3.3.4.1.2 When present, the Quantization subpart shall have the structure specified in table 5-8.

Table 5-8: Quantization Subpart Structure

Block	Status	Size (Bytes)	Reference
Error Limit Update Period	Conditional	1	5.3.3.4.2
Absolute Error Limit	Conditional	Variable	5.3.3.4.3
Relative Error Limit	Conditional	Variable	5.3.3.4.4

5.3.3.4.2 Error Limit Update Period

5.3.3.4.2.1 When the Quantization subpart is present, it shall include the Error Limit Update Period block unless BSQ encoding is used, in which case it shall be omitted.

5.3.3.4.2.2 When present, the Error Limit Update Period block shall have the structure specified in table 5-9.

Table 5-9: Error Limit Update Period Block Structure

Field	Width (bits)	Description	Reference
Reserved	1	This field shall have value ‘0’.	
Periodic Updating Flag	1	‘0’: periodic error limit updating is not used. ‘1’: periodic error limit updating is used.	4.8.2.4
Reserved	2	This field shall contain all ‘zeros’.	
Update Period Exponent	4	When periodic error limit updating is used, this field shall contain the value u encoded as a 4-bit unsigned binary integer. Otherwise, this field shall contain all ‘zeros’.	4.8.2.4

5.3.3.4.3 Absolute Error Limit

5.3.3.4.3.1 General

5.3.3.4.3.1.1 The Absolute Error Limit block shall be included when absolute error limits are used and be omitted otherwise.

5.3.3.4.3.1.2 When present, the Absolute Error Limit block shall have the structure specified in table 5-10.

Table 5-10: Absolute Error Limit Block Structure

Field	Width (bits)	Description	Reference
Reserved	1	This field shall have value ‘0’.	
Absolute Error Limit Assignment Method	1	‘0’: band-independent absolute error limit assignment. ‘1’: band-dependent absolute error limit assignment.	4.8.2.3.1
Reserved	2	This field shall have value ‘00’.	
Absolute Error Limit Bit Depth	4	The value D_{A} encoded mod 2^4 as a 4-bit unsigned integer.	4.8.2.2.1
Absolute Error Limit Values Subblock (conditional)	(variable)	(See 5.3.3.4.3.2 below.)	4.8.2

5.3.3.4.3.2 Absolute Error Limit Values Subblock

5.3.3.4.3.2.1 When the Absolute Error Limit block is present, it shall include the Absolute Error Limit Values Subblock unless periodic error limit updating is used, in which case it shall be omitted.

5.3.3.4.3.2.2 If band-independent absolute error limits are used, then the Absolute Error Limit Values Subblock consists of the value A^* encoded as a D_A-bit unsigned binary integer, followed by fill bits as needed to reach the next byte boundary.

5.3.3.4.3.2.3 If band-dependent absolute error limits are used, then the Absolute Error Limit Values Subblock shall consist of (a) the sequence of a_z values, in order of increasing band index z, each encoded as a D_A-bit unsigned binary integer, followed by (b) fill bits as needed to reach the next byte boundary.

5.3.3.4.4 Relative Error Limit

5.3.3.4.4.1 General

5.3.3.4.4.1.1 The Relative Error Limit block shall be included when relative error limits are used and shall be omitted otherwise.

5.3.3.4.4.1.2 When present, the Relative Error Limit block shall have the structure specified in table 5-11.

Table 5-11: Relative Error Limit Block Structure

Field	Width (bits)	Description	Reference
Reserved	1	This field shall have value ‘0’.	
Relative Error Limit Assignment Method	1	‘0’: band-independent relative error limit assignment. ‘1’: band-dependent relative error limit assignment.	4.8.2.3.2
Reserved	2	This field shall have value ‘00’.	
Relative Error Limit Bit Depth	4	The value D_R encoded mod 2^4 as a 4-bit unsigned integer.	4.8.2.2.2
Relative Error Limit Values Subblock (conditional)	(variable)	(See 5.3.3.4.4.2 below.)	

5.3.3.4.4.2 Relative Error Limit Values Subblock

5.3.3.4.4.2.1 When the Relative Error Limit block is present, it shall include the Relative Error Limit Values Subblock unless periodic error limit updating is used, in which case it shall be omitted.

5.3.3.4.4.2.2 If band-independent relative error limits are used, then the Relative Error Limit Values Subblock shall consist of the value R^* encoded as a D_R-bit unsigned binary integer, followed by fill bits as needed to reach the next byte boundary.

5.3.3.4.4.2.3 If band-dependent relative error limits are used, then the Relative Error Limit Values Subblock shall consist of (a) the sequence of r_z values, in order of increasing band index z, each encoded as a D_R-bit unsigned binary integer, followed by (b) fill bits as needed to reach the next byte boundary.

5.3.3.5 Sample Representative

5.3.3.5.1 General

5.3.3.5.1.1 The Sample Representative subpart may only be included when $\Theta > 0$. The inclusion of the Sample Representative subpart shall be indicated by setting the Sample Representative Flag field bit to ‘1’.

5.3.3.5.1.2 When present, the Sample Representative subpart shall have the structure specified in table 5-12.

Table 5-12: Sample Representative Subpart Structure

Field	Width (bits)	Description	Reference
Reserved	5	This field shall contain all ‘zeros’.	
Sample Representative Resolution	3	Value of Θ encoded as a 3-bit unsigned binary integer.	4.9.1
Reserved	1	This field shall have value ‘0’.	
Band-Varying Damping Flag	1	‘0’: all bands use the same value of ϕ_z. ‘1’: the value ϕ_z of may vary from band to band.	4.9.1
Damping Table Flag	1	‘0’: the Damping Table subblock is not included in the Sample Representative subpart. ‘1’: the Damping Table subblock is included in the Sample Representative subpart.	4.9.1
Reserved	1	This field shall have value ‘0’.	

Field	Width (bits)	Description	Reference
Fixed Damping Value	4	If the Band-Varying Damping Flag field is ‘0’, then this field encodes the value of ϕ_z to use for all bands as a 4-bit unsigned integer. Otherwise, this field shall be all ‘zeros’.	4.9.1
Reserved	1	This field shall have value ‘0’.	
Band-Varying Offset Flag	1	‘0’: all bands use the same value of ψ_z. ‘1’: the value of ψ_z may vary from band to band.	4.9.1
Offset Table Flag	1	‘0’: the Offset Table subblock is not included in the Sample Representative subpart. ‘1’: the Offset Table subblock is included in the Sample Representative subpart.	4.9.1
Reserved	1	This field shall have value ‘0’.	
Fixed Offset Value	4	If the Band-Varying Offset Field Flag field is ‘0’, then this field encodes the value of ψ_z to use for all bands as a 4-bit unsigned integer. Otherwise, this field shall be all ‘zeros’.	4.9.1
Damping Table Subblock (optional)	(variable)	(See 5.3.3.5.2 below.)	4.9.1
Offset Table Subblock (optional)	(variable)	(See 5.3.3.5.3 below.)	4.9.1

5.3.3.5.2 Damping Table Subblock

5.3.3.5.2.1 The optional Damping Table Subblock may only be included when the Band-Varying Damping Flag field is ‘1’. The inclusion of the Damping Table Subblock shall be indicated by setting the Damping Table Flag field to ‘1’.

NOTE – Even when the damping value ϕ_z varies from band to band, the Damping Table Subblock might be omitted. For example, a mission might design a fixed set of damping values to be used throughout a mission and elect to not encode these values with each image.

5.3.3.5.2.2 When present, the Damping Table Subblock shall consist of (a) the sequence of ϕ_z values, in order of increasing band index z, each encoded as a Θ-bit unsigned binary integer, followed by (b) fill bits as needed to reach the next byte boundary.

5.3.3.5.3 Offset Table Subblock

5.3.3.5.3.1 The optional Offset Table Subblock may only be included when the Band-Varying Offset Flag field is ‘1’. The inclusion of the Offset Table Subblock shall be indicated by setting the Offset Table Flag field to ‘1’.

NOTE – Even when the offset value ψ_z varies from band to band, the Offset Table Subblock might be omitted. For example, a mission might design a fixed set of offset values to be used throughout a mission and elect to not encode these values with each image.

5.3.3.5.3.2 When present, the Offset Table Subblock shall consist of (a) the sequence of ψ_z values, in order of increasing band index z, each encoded as a Θ-bit unsigned binary integer, followed by (b) fill bits as needed to reach the next byte boundary.

5.3.4 ENTROPY CODER METADATA

5.3.4.1 General

The Entropy Coder Metadata header part shall follow the structure defined in 5.3.4.2 if the sample-adaptive entropy coder is used, the structure defined in 5.3.4.3 if the hybrid entropy coder is used, or the structure defined in 5.3.4.4 if the block-adaptive entropy coder is used.

5.3.4.2 Sample-Adaptive Entropy Coder

5.3.4.2.1 Header

When the sample-adaptive entropy coder is used, the Entropy Coder Metadata header part shall have the structure specified in table 5-13.

Table 5-13: Entropy Coder Metadata Structure When Sample Adaptive Entropy Coder Is Used

Field	Width (bits)	Description	Reference
Unary Length Limit	5	The value $U_{\max}$ encoded mod 2^5 as a 5-bit unsigned binary integer.	5.4.3.2.2
Rescaling Counter Size	3	The value $(\gamma^* - 4)$ encoded as a 3-bit unsigned binary integer.	5.4.3.2.3.4
Initial Count Exponent	3	The value γ_0 encoded mod 2^3 as a 3-bit unsigned binary integer.	5.4.3.2.3.2
Accumulator Initialization Constant	4	When an accumulator initialization constant K is specified, this field encodes the value of K as a 4-bit unsigned binary integer. Otherwise, this field shall be all 'ones'.	5.4.3.2.3.3
Accumulator Initialization Table Flag	1	'0': Accumulator Initialization Table is not included in Entropy Coder Metadata. '1': Accumulator Initialization Table is included in Entropy Coder Metadata.	5.4.3.2.3.3
Accumulator Initialization Table (Optional)	(variable)	(See 5.3.4.2.2 below.)	5.4.3.2.3.3

5.3.4.2.2 Accumulator Initialization Table

5.3.4.2.2.1 The optional Accumulator Initialization Table may be included when an accumulator initialization constant is not specified. The presence of an accumulator initialization table shall be indicated by setting the Accumulator Initialization Table Flag field to '1'.

NOTE – Even when an accumulator initialization constant is not used, the Accumulator Initialization Table may be omitted. For example, a mission might design a fixed set of accumulator initialization values to be used throughout a mission and elect to not encode these values with each image.

5.3.4.2.2.2 The Accumulator Initialization Table shall consist of the concatenated sequence of k_z'' values, $k_0'', k_1'', \ldots, k_{N_z-1}''$ (defined in 5.4.3.2.3.3), each encoded as a 4-bit binary unsigned integer.

5.3.4.2.2.3 Fill bits shall be appended to the Accumulator Initialization Table as needed to reach the next byte boundary.

5.3.4.3 Hybrid Entropy Coder

When the hybrid entropy coder is used, the Entropy Coder Metadata header part shall have the structure specified in table 5-14.

Table 5-14: Entropy Coder Metadata Structure When Hybrid Entropy Coder Is Used

Field	Width (bits)	Description	Reference
Unary Length Limit	5	The value $U_{\max}$ encoded mod 2^5 as a 5-bit unsigned binary integer.	5.4.3.3.3.2.2
Rescaling Counter Size	3	The value $(\gamma^* - 4)$ encoded as a 3-bit unsigned binary integer.	5.4.3.3.4.4
Initial Count Exponent	3	The value γ_0 encoded mod 2^3 as a 3-bit unsigned binary integer.	5.4.3.3.4.2
Reserved	5	This field shall have value ‘00000’.	

5.3.4.4 Block-Adaptive Entropy Coder

When the block-adaptive entropy coder is used, the Entropy Coder Metadata header part shall have the structure specified in table 5-15.

Table 5-15: Entropy Coder Metadata Structure When Block Adaptive Entropy Coder Is Used

Field	Width (bits)	Description	Reference
Reserved	1	This field shall have value ‘0’.	
Block Size	2	‘00’: Block size $J = 8$. ‘01’: Block size $J = 16$. ‘10’: Block size $J = 32$. ‘11’: Block size $J = 64$.	5.4.3.4.2.4
Restricted Code Options Flag	1	This field shall have value ‘1’ when $D \leq 4$ and the Restricted set of code options (as defined in subsection 5.1.2 of reference [1]) are used. Otherwise, this field shall have value ‘0’.	
Reference Sample Interval	12	Value of r encoded mod 2^{12} as a 12-bit unsigned binary integer.	5.4.3.4.2.5

5.4 BODY

5.4.1 OVERVIEW

The *entropy coder input sequence* consists of the mapped quantizer indices, and, when periodic error limit updating is used (see 4.8.2.4), quantizer error limit values. This input sequence is arranged in one of the allowed orders specified in 5.4.2. The compressed image body losslessly encodes this sequence using one of the three entropy coding methods specified in 5.4.3.

5.4.2 INPUT ORDER

5.4.2.1 General

The entropy coder input sequence shall be arranged in Band-Interleaved (BI) order, as defined in 5.4.2.2, or BSQ order, as defined in 5.4.2.3.

NOTES

1 The input order specifies the order in which the entropy coder input sequence values are input to the entropy coder.

2 The commonly used Band-Interleaved-by-Pixel (BIP) and Band-Interleaved-by-Line (BIL) orders are each special cases of the more general BI encoding order.

3 The entropy coder input sequence order does not necessarily correspond to the order in which samples are produced by an imaging instrument or processed by a predictor implementation.

5.4.2.2 Band-Interleaved Order

5.4.2.2.1 The user-specified *sub-frame interleaving depth M* shall be an integer in the range $1 \leq M \leq N_Z$.

5.4.2.2.2 Under BI input order, the entropy coder input sequence order is defined by the nesting of sample index loops as follows:

```
for $y = 0$ to $N_Y - 1$
    if $y \bmod 2^u = 0$ and periodic error limit updating is used
        if absolute error limits are used
            if absolute error limits are band-independent
                input $A^*$ to the entropy coder
            else
                input $a_0, a_1, ..., a_{N_Z-1}$ to the entropy coder
        if relative error limits are used
            if relative error limits are band-independent
                input $R^*$ to the entropy coder
            else
                input $r_0, r_1, ..., r_{N_Z - 1}$ to the entropy coder
    for $i = 0$ to $\lceil N_Z / M \rceil - 1$
        for $x = 0$ to $N_X - 1$
            for $z = iM$ to $\min\{(i+1)M - 1,\ N_Z - 1\}$
                input $\delta_{z,y,x}$ to the entropy coder.
```

NOTES

1 Under BI encoding order, when $M = 1$, the input order corresponds to BIL, and when $M = N_Z$ the input order corresponds to BIP.

2 When periodic error limit updates are not used, error limit values are not part of the entropy coder input sequence and instead are encoded in the header as specified in 5.3.3.4.

5.4.2.3 Band-Sequential Order

Under BSQ input order, the entropy coder input sequence order is defined by the nesting of sample index loops as follows:

```
for $z = 0$ to $N_Z - 1$
    for $y = 0$ to $N_Y - 1$
        for $x = 0$ to $N_X - 1$
            input $\delta_{z,y,x}$ to the entropy coder.
```

NOTE – As specified in 4.8.2.4, periodic error limit updates are not permitted when BSQ encoding order is used.

5.4.3 ENTROPY CODING METHOD

5.4.3.1 General

The entropy coder input sequence shall be encoded using either the sample-adaptive entropy coding approach specified in 5.4.3.2, the hybrid entropy coding approach specified in 5.4.3.3, or the block-adaptive entropy coding approach specified in 5.4.3.4.

5.4.3.2 Sample-Adaptive Entropy Coder

5.4.3.2.1 General

Under the sample-adaptive entropy coding option, each mapped quantizer index $\delta_z(t)$ shall be encoded using a variable-length binary codeword.

NOTE – The family of variable-length codes used is defined in 5.4.3.2.2, and the adaptive code selection statistics used to select the codeword for each mapped quantizer index are specified in 5.4.3.2.3. The procedure for selecting the codeword for each mapped quantizer index and encoding error limit values is specified in 5.4.3.2.4.

5.4.3.2.2 Length-Limited Golomb-Power-of-2 Codewords

5.4.3.2.2.1 The length-limited Golomb-power-of-2 (GPO2) codeword for unsigned integer j and unsigned integer code index k, denoted $\Re_k(j)$, is a variable-length binary codeword defined as follows:

a) if $\lfloor j/2^k \rfloor < U_{\max}$ then $\Re_k(j)$ consists of $\lfloor j/2^k \rfloor$ ‘zeros’, followed by a ‘one’, followed by the k least significant bits of the binary representation of j;

b) otherwise, $\Re_k(j)$ consists of $U_{\max}$ ‘zeros’ followed by the D-bit binary representation of j.

5.4.3.2.2.2 The user-specified unary length limit $U_{\max}$ shall be an integer in the range $8 \le U_{\max} \le 32$.

NOTE – The definition ensures that each codeword $\Re_k(j)$ is not longer than $U_{\max} + D$ bits.

5.4.3.2.3 Adaptive Code Selection Statistics

5.4.3.2.3.1 The adaptive code selection statistics shall consist of an *accumulator* $\Sigma_z(t)$ and a *counter* $\Gamma(t)$ that are adaptively updated during the encoding process.

NOTE – The ratio $\Sigma_z(t) / \Gamma(t)$ provides an estimate of the mean mapped quantizer index value in the spectral band. This ratio determines the variable-length code used to encode $\delta_z(t)$.

5.4.3.2.3.2 The initial counter value Γ(1) shall be equal to

$$\Gamma(1) = 2^{\gamma_0}, \tag{57}$$

where the user-specified value of the initial count exponent γ_0 shall be an integer in the range $1 \le \gamma_0 \le 8$.

5.4.3.2.3.3 For each spectral band *z*, the initial accumulator value $\Sigma_z(1)$ shall be equal to

$$\Sigma_z(1) = \left\lfloor \frac{1}{2^7}\left(3 \cdot 2^{k'_z+6} - 49\right)\Gamma(1) \right\rfloor, \tag{58}$$

where

$$k_z' = \begin{cases} k_z'', & k_z'' \le 30 - D \\ 2k_z'' + D - 30, & k_z'' > 30 - D \end{cases}, \tag{59}$$

and the user-selected value k_z'' shall be an integer in the range $0 \le k_z''' \le \min(D-2,14)$. An *accumulator initialization constant* *K* may be specified, with $0 \le K \le \min(D-2,14)$, in which case k_z'' = *K* for all *z*.

NOTE – This calculation ensures that initial value of encoding parameter $k_z(t)$ computed for spectral band *z* (see 5.4.3.2.4.3) will be equal to k'_z.

5.4.3.2.3.4 For *t* >1, the value of the accumulator for spectral band *z* is defined as

$$\Sigma_z(t) = \begin{cases} \Sigma_z(t-1) + \delta_z(t-1), & \Gamma(t-1) < 2^{\gamma^*} - 1 \\ \left\lfloor \dfrac{\Sigma_z(t-1) + \delta_z(t-1) + 1}{2} \right\rfloor, & \Gamma(t-1) = 2^{\gamma^*} - 1 \end{cases}, \tag{60}$$

and the value of the counter is defined as

$$\Gamma(t) = \begin{cases} \Gamma(t-1) + 1, & \Gamma(t-1) < 2^{\gamma^*} - 1 \\ \left\lfloor \dfrac{\Gamma(t-1) + 1}{2} \right\rfloor, & \Gamma(t-1) = 2^{\gamma^*} - 1 \end{cases}. \tag{61}$$

The interval at which the counter Γ(*t*) and the accumulator $\Sigma_z(t)$ are rescaled is controlled by the user-defined rescaling counter size parameter γ^*, which shall be an integer in the range $\max\{4, \gamma_0 + 1\} \le \gamma^* \le 11$.

5.4.3.2.4 Coding Procedure

5.4.3.2.4.1 Each absolute error limit value shall be encoded as a D_A-bit unsigned binary integer, and each relative error limit value shall be encoded as a D_R-bit unsigned binary integer.

NOTE – The adaptive code selection statistics are unaffected by the encoding of error limit values.

5.4.3.2.4.2 The first mapped quantizer index in each spectral band z shall be uncoded; that is, the codeword for $\delta_z(0)$ is simply the D-bit unsigned binary integer representation of $\delta_z(0)$.

5.4.3.2.4.3 For $t > 0$, the codeword for the mapped quantizer index $\delta_z(t)$ is $\Re_{k_z(t)}(\delta_z(t))$, where $k_z(t) = 0$ if $2\Gamma(t) > \Sigma_z(t) + \left\lfloor \frac{49}{2^7}\Gamma(t) \right\rfloor$; otherwise, $k_z(t)$ is the largest positive integer $k_z(t) \le D - 2$, such that

$$\Gamma(t)2^{k_z(t)} \le \Sigma_z(t) + \left\lfloor \frac{49}{2^7}\Gamma(t) \right\rfloor. \tag{62}$$

5.4.3.2.4.4 Following the last codeword in the compressed image, fill bits shall be appended as needed to reach the next output word boundary, so that the compressed image size is a multiple of the output word size. Fill bits shall be all ‘zeros’.

5.4.3.3 Hybrid Entropy Coder

5.4.3.3.1 Overview

Under the hybrid entropy coding option, adaptive code selection statistics are used to assign each mapped quantizer index to either a ‘high-entropy’ or ‘low-entropy’ coding method. Each high-entropy mapped quantizer index is encoded using a variable-length binary codeword from a family of codes. For each low-entropy mapped quantizer index, one of 16 variable-to-variable length codes is used. A single output codeword from a low-entropy code can encode multiple input-mapped quantizer indices, which allows lower compressed data rates than can be achieved by the high-entropy codes. Each high-entropy mapped quantizer index immediately produces an output codeword that is written to the compressed bitstream, while each low-entropy code waits until enough data has arrived to determine the next output codeword.

The decoder can accommodate the varying latency between the arrival of a low-entropy mapped quantizer index and its ultimate encoding by decoding the compressed image body in reverse order. This is possible because (1) the output codewords from the high- and low-entropy codes are suffix-free rather than prefix-free, (2) the compressed image body ends with a compressed image ‘tail’ (see 5.4.3.3.5.4) that encodes the final state of each low-entropy code and the final high-resolution accumulator value for each band, and (3) each time the adaptive code selection statistics are rescaled (see 5.4.3.3.4.4), an additional bit is output (see 5.4.3.3.5.1.2) so that the decoder can invert this rescaling operation. Because decoding proceeds in reverse, users need to provide a mechanism by which the decoder can locate the end of the compressed image body (see 2.4).

5.4.3.3.2 General

The high-entropy and low-entropy encoding methods used to encode mapped quantizer indices are specified in 5.4.3.3.3. The coding method selection depends on the adaptive code selection statistics specified in 5.4.3.3.4. Following the processing of the entropy coder input sequence using the procedure specified in 5.4.3.3.5, the compressed image body concludes with the compressed image tail, specified in 5.4.3.3.5.4.

5.4.3.3.3 Encoding Methods

5.4.3.3.3.1 Overview

A mapped quantizer index is encoded using one of several reversed length-limited GPO2 codes specified in 5.4.3.3.3.2, or using one of 16 low-entropy codes specified in 5.4.3.3.3.3.

5.4.3.3.3.2 Reversed Length-Limited Golomb-Power-of-2 Codewords

5.4.3.3.3.2.1 The reversed length-limited GPO2 codeword for unsigned integer j and unsigned integer code index k, denoted $\Re'_k(j)$, is a variable-length binary codeword defined as follows:

a) if $\lfloor j/2^k \rfloor < U_{\max}$ then $\Re'_k(j)$ consists of the k least significant bits of the binary representation of j, followed by a ‘one’, followed by $\lfloor j/2^k \rfloor$ ‘zeros’;

b) otherwise, $\Re'_k(j)$ consists of the D-bit binary representation of j followed by $U_{\max}$ ‘zeros’.

NOTE – The codewords $\Re'_k(j)$ and $\Re_k(j)$ are equivalent, but with the bits arranged in a different order. Also, $\Re'_k(j)$ is not in general the reverse of $\Re_k(j)$.

5.4.3.3.3.2.2 The user-specified unary length limit $U_{\max}$ shall be an integer in the range $8 \leq U_{\max} \leq 32$.

5.4.3.3.3.3 Low-Entropy Codes

5.4.3.3.3.3.1 The low-entropy codes are a set of 16 non-binary-input, binary-output, variable-to-variable length codes. Each low-entropy shall consist of

a) a threshold value T_i and input symbol limit L_i, values for both of which shall be those given in table 5-16;

b) a code defined by a prefix-free set of non-binary variable-length *input codewords* with a mapping onto a set of variable-length binary *output codewords*; and

c) a flush table that gives a mapping from the set of all proper prefixes of input codewords onto a set of output flush words.

NOTE – The code table and flush table for each low-entropy code are specified in annex B.

Table 5-16: Low-Entropy Code Input Symbol Limit and Threshold

Code Index, i	Input Symbol Limit, L_i	Threshold, T_i
0	12	303336
1	10	225404
2	8	166979
3	6	128672
4	6	95597
5	4	69670
6	4	50678
7	4	34898
8	2	23331
9	2	14935
10	2	9282
11	2	5510
12	2	3195
13	2	1928
14	2	1112
15	0	408

5.4.3.3.3.3.2 During encoding, each low-entropy code has an *active prefix*, which is a sequence of input symbols. Initially, the active prefix for each low-entropy code shall be equal to the null (empty) sequence.

5.4.3.3.4 Adaptive Code Selection Statistics

5.4.3.3.4.1 The adaptive code selection statistics for the hybrid entropy coder shall consist of a *high-resolution accumulator* $\tilde{\Sigma}_z(t)$ and a *counter* $\Gamma(t)$, which are adaptively updated during the encoding process.

NOTE – The ratio $\tilde{\Sigma}_z(t)/\Gamma(t)$ provides a scaled estimate of the mean mapped quantizer index value in the spectral band. This ratio determines how $\delta_z(t)$ is encoded.

5.4.3.3.4.2 The initial counter value $\Gamma(0)$ shall be equal to

$$\Gamma(0) = 2^{\gamma_0}, \tag{63}$$

where the user-specified value of the initial count exponent γ_0 shall be an integer in the range $1 \le \gamma_0 \le 8$.

5.4.3.3.4.3 For each spectral band z, the initial high-resolution accumulator value $\tilde{\Sigma}_z(0)$ shall be a user-specified integer in the range $0 \le \tilde{\Sigma}_z(0) < 2^{D+\gamma_0}$.

NOTES

1 The value of $\tilde{\Sigma}_z(0)$ is not directly encoded in the header or bitstream.

2 If an estimate $\hat{\delta}_z$ (such an estimate might arise from a preceding compressed image) of the mean mapped quantizer index for spectral band z is available, then a reasonable rule-of-thumb is to initialize $\tilde{\Sigma}_z(0)$ to be approximately equal to $4\Gamma(0)\hat{\delta}_z$.

5.4.3.3.4.4 For $t \ge 1$, the value of the high-resolution accumulator for spectral band z is defined as

$$\tilde{\Sigma}_z(t) = \begin{cases} \tilde{\Sigma}_z(t-1) + 4\delta_z(t), & \Gamma(t-1) < 2^{\gamma^*} - 1 \\ \left\lfloor \dfrac{\tilde{\Sigma}_z(t-1) + 4\delta_z(t) + 1}{2} \right\rfloor, & \Gamma(t-1) = 2^{\gamma^*} - 1 \end{cases}, \tag{64}$$

and the value of the counter is defined as

$$\Gamma(t) = \begin{cases} \Gamma(t-1)+1, & \Gamma(t-1) < 2^{\gamma^*} - 1 \\ \left\lfloor \dfrac{\Gamma(t-1)+1}{2} \right\rfloor, & \Gamma(t-1) = 2^{\gamma^*} - 1 \end{cases}. \qquad (65)$$

5.4.3.3.4.5 The interval at which the counter $\Gamma(t)$ and the high-resolution accumulator $\tilde{\Sigma}_z(t)$ are rescaled is controlled by the user-defined rescaling counter size parameter γ^*, which shall be an integer in the range $\max\{4, \gamma_0 + 1\} \leq \gamma^* \leq 11$.

5.4.3.3.5 Coding Procedure

5.4.3.3.5.1 General

5.4.3.3.5.1.1 Each absolute error limit value shall be encoded as a D_A-bit unsigned binary integer, and each relative error limit value shall be encoded as a D_R-bit unsigned binary integer.

NOTE – The adaptive code selection statistics are unaffected by the encoding of error limit values.

5.4.3.3.5.1.2 When $\Gamma(t-1) = 2^{\gamma^*} - 1$ (i.e., when code selection statistics are rescaled, as described in 5.4.3.2.3.4), the least-significant bit of $\tilde{\Sigma}_z(t-1)$ shall be encoded in the bitstream (i.e., a single ‘1’ bit when this quantity is odd and a ‘0’ bit when it is even) immediately before any bits output as a result of the processing steps for $\delta_z(t)$ specified below.

NOTE – This bit allows the decoder to reconstruct the sequence of high-resolution accumulator values.

5.4.3.3.5.1.3 The first mapped quantizer index in each spectral band z shall be uncoded; that is, the D-bit unsigned binary integer representation of $\delta_z(0)$ is output to the compressed bitstream.

5.4.3.3.5.1.4 For $t > 0$, if $\tilde{\Sigma}_z(t) \cdot 2^{14} \geq T_0 \cdot \Gamma(t)$, then $\delta_z(t)$ is said to be a ‘high-entropy’ mapped quantized index and shall be encoded using a reversed length-limited GPO2 code as described below in 5.4.3.3.5.2. Otherwise, $\delta_z(t)$ is said to be a ‘low-entropy’ mapped quantized index and shall be processed as described below in 5.4.3.3.5.3.

NOTE – When D=2, the condition for using the high-entropy coding method is never met; all mapped quantizer indices are low-entropy.

5.4.3.3.5.2 High-Entropy Processing

If $\delta_z(t)$ is a high-entropy mapped quantizer index, then it shall be encoded by appending codeword $\Re'_{k_z(t)}(\delta_z(t))$ to the compressed bitstream, where $k_z(t)$ is the largest positive integer $k_z(t) \leq \max\{D - 2, 2\}$, such that

$$\Gamma(t)2^{k_z(t)+2} \leq \tilde{\Sigma}_z(t) + \left\lfloor \frac{49}{2^5}\Gamma(t) \right\rfloor. \tag{66}$$

NOTE – For high-entropy samples, it can be shown that $k_z(t) \geq 2$.

5.4.3.3.5.3 Low-Entropy Processing

5.4.3.3.5.3.1 If $\delta_z(t)$ is a low-entropy mapped quantizer index, then it shall be encoded using the low-entropy code with largest code index i satisfying $\tilde{\Sigma}_z(t) \cdot 2^{14} < \Gamma(t) \cdot T_i$.

5.4.3.3.5.3.2 The *input symbol* to the low-entropy code is

$$\iota_z(t) = \begin{cases} \delta_z(t), & \delta_z(t) \leq L_i \\ \mathrm{X}, & \delta_z(t) > L_i \end{cases}, \tag{67}$$

where L_i is the input symbol limit for the code, and ‘X’ denotes the ‘escape’ symbol.

5.4.3.3.5.3.3 If $\iota_z(t) = \mathrm{X}$, then the residual value $\delta_z(t) - L_i - 1$ shall be encoded by appending codeword $\Re'_0(\delta_z(t) - L_i - 1)$ to the compressed bitstream.

5.4.3.3.5.3.4 The active prefix for the ith low-entropy code is updated by appending the input symbol $\iota_z(t)$ to that active prefix.

5.4.3.3.5.3.5 If after updating the active prefix it is equal to a complete input codeword, as specified in the code table for that code, then

a) the corresponding output codeword listed in the table shall be appended to the compressed bitstream; and

b) the active prefix for the low-entropy code shall be reset to the null sequence.

NOTE – The low-entropy code designs ensure that the active prefix is always equal to a complete input codeword whenever the input symbol is the escape symbol.

5.4.3.3.5.4 Compressed Image Tail

5.4.3.3.5.4.1 Following the processing of the entropy coder input sequence as specified in 5.4.3.3.5.1.1–5.4.3.3.5.1.4, the compressed image tail shall be produced by using the low-entropy code flush tables to encode the active prefix of each low-entropy code as described in 5.4.3.3.5.4.2, encoding the final high-resolution accumulator value in each band as described in 5.4.3.3.5.4.3, appending an additional '1' bit as described in 5.4.3.3.5.4.4, and appending fill bits (if needed) as described in 5.4.3.3.5.4.5.

5.4.3.3.5.4.2 For each low-entropy code, in order of increasing code index, the active prefix for the low-entropy code shall be encoded by writing the corresponding flush codeword (given in the code's flush table in annex B) to the compressed bitstream.

NOTE – The flush code tables define an output codeword for each possible active prefix, including the null sequence. Thus a flush codeword is output for each of the 16 low-entropy codes.

5.4.3.3.5.4.3 For each spectral band z, in order of increasing band index, the final high-resolution accumulator value $\tilde{\Sigma}_z(N_X \cdot N_Y - 1)$ shall be encoded directly as an unsigned integer using $2 + D + \gamma^*$ bits.

5.4.3.3.5.4.4 Following the encoding of final high-resolution accumulator values, a single '1' bit shall be appended.

NOTE – This '1' bit allows the decoder to identify fill bits encoded in 5.4.3.3.5.4.5.

5.4.3.3.5.4.5 Fill bits shall be appended as needed to reach the next output word boundary, so that the compressed image size is a multiple of the output word size. Fill bits shall be all 'zeros'.

5.4.3.4 Block-Adaptive Entropy Coder

5.4.3.4.1 General

When the block-adaptive entropy coding method is used, the entropy coder input sequence shall be encoded using the adaptive entropy coder specified in reference [1].

5.4.3.4.2 Parameters and Options

5.4.3.4.2.1 When the block-adaptive entropy coding method is used, the following options and parameters shall apply.

5.4.3.4.2.2 The preprocessor function defined in section 4 of reference [1] shall not be used. The option to bypass the preprocessor shall be used.

5.4.3.4.2.3 The *resolution* parameter, *n*, defined in subsection 3.1 of reference [1], shall be equal to the image dynamic range *D*.

5.4.3.4.2.4 The *block size* parameter, *J*, defined in subsection 3.1 of reference [1], shall be equal to 8, 16, 32, or 64.

5.4.3.4.2.5 The *reference sample interval* parameter, *r*, defined in subsection 4.3 of reference [1], shall be a positive integer not larger than 4096.

NOTE – Because the preprocessor is bypassed, reference samples are not included in the compressed image body. The reference sample interval serves only to define an interval of input data sample blocks that will be further segmented in the 'zero-block' encoding option defined in reference [1].

5.4.3.4.2.6 Either the Basic or Restricted set of code options, as defined in subsection 5.1.2 of reference [1], may be used.

5.4.3.4.2.7 The input to the adaptive entropy coder specified in reference [1] shall be the entropy coder input sequence, as specified in 5.4.2, with 'zeros' appended as needed so that the length is a multiple of *J*.

5.4.3.4.3 Body

5.4.3.4.3.1 The compressed image body shall consist of the concatenation of the Coded Data Sets (CDSes), defined in subsection 5.1.4 of reference [1], produced by the encoder.

5.4.3.4.3.2 Fill bits shall be appended after the last CDS as needed to reach the next output word boundary, so that the compressed image size is a multiple of the output word size. Fill bits shall be all 'zeros'. Fill bits shall not be inserted between CDSes.

ANNEX A

IMPLEMENTATION CONFORMANCE STATEMENT (ICS) PROFORMA

(NORMATIVE)

A1 INTRODUCTION

A1.1 OVERVIEW

This annex provides the Implementation Conformance Statement (ICS) Requirements List (RL) for an implementation of *Low-Complexity Lossless and Near-Lossless Multispectral and Hyperspectral Image Compression*, CCSDS 123.0-B-2, February 2019. The ICS for an implementation is generated by completing the RL in accordance with the instructions below. An implementation claiming conformance must satisfy the mandatory requirements referenced in the RL.

A1.2 ABBREVIATIONS AND CONVENTIONS

The RL consists of information in tabular form. The status of features is indicated using the abbreviations and conventions described below.

Item Column

The label in the item column identifies the item in the table.

The use of nested item labels indicates subordination of conditional items. For example, an item with label L*i.j* is not applicable unless the parent item L*i* is supported.

Description Column

The description column contains a brief description of the item. It implicitly means “Is this item supported by the implementation?”

Reference Column

The reference column indicates the relevant subsection of *Low-Complexity Lossless and Near-Lossless Multispectral and Hyperspectral Image Compression*, CCSDS 123.0-B-2 (this document).

Status Column

The status column uses the following notations:

M	mandatory.

O optional.

N/A not applicable.

O.*i* qualified optional—for a group of related optional items labeled by the same numeral *i*, it is mandatory to support at least one of the items.

C.*j* conditional—the requirement on the capability (‘M’, ‘O’, or ‘N/A’) depends on the support of another optional item. The numeral *j* identifies a unique conditional status expression defined immediately following the table.

C:<status> indicates that the status applies for the given subordinate item when the parent item is supported, and is not applicable otherwise.

<condition>:<status> indicates that the status applies only when the given condition is met, and is not applicable otherwise. For example, ‘(Q2 or Q3):M’ indicates that support for the item is mandatory if item Q2 or item Q3 are supported and not applicable otherwise.

Values Allowed Column

The values allowed column contains the list or range of values allowed. The following notations are used:

range of values:	<min value> .. <max value>
example:	2 .. 16
list of values:	<value1>, <value2>, ..., <valueN>
example:	3, 6, 9, …, 21
N/A	not applicable

Item Support or Values Supported Column

In the item support column, the support of every item as claimed by the implementer shall be stated by entering the appropriate answer:

Y yes, item supported by the implementation;

N no, item not supported by the implementation;

N/A not applicable.

In the values supported column, the implementer shall enter the values supported.

Prerequisite Line

A prerequisite line takes the form: Prerequisite: <predicate>. A prerequisite line at the top of a table indicates that the table need not be completed if the predicate is FALSE.

A1.3 INSTRUCTIONS FOR COMPLETING THE RL

An implementer shows the extent of compliance to the Recommended Standard by completing the RL; that is, the state of compliance with all mandatory requirements and the options supported are shown. The resulting completed RL is called an ICS. The implementer shall complete the RL by entering appropriate responses in the support or values supported column, using the notation described in A1.2. If a conditional requirement is inapplicable, N/A should be used. If a mandatory requirement is not satisfied, exception information must be supplied by entering a reference X*i*, where *i* is a unique identifier, to an accompanying rationale for the noncompliance.

A2 ICS PROFORMA FOR LOW-COMPLEXITY LOSSLESS AND NEAR-LOSSLESS MULTISPECTRAL AND HYPERSPECTRAL IMAGE COMPRESSION

A2.1 GENERAL INFORMATION

A2.1.1 Identification of ICS

Date of Statement (DD/MM/YYYY)	
ICS serial number	
System Conformance statement cross-reference	

A2.1.2 Identification of Implementation Under Test

Implementation Name	
Implementation Version	
Function Implemented	Compression_____ Decompression_____
Special Configuration	
Other Information	

A2.1.3 Identification of Supplier

Supplier	
Contact Point for Queries	
Implementation Name(s) and Versions	
Other information necessary for full identification, for example, name(s) and version(s) for machines and/or operating systems; System Name(s)	

A2.1.4 Identification of Specification

CCSDS 123.0-B-2	
Have any exceptions been required? NOTE – A YES answer means that the implementation does not conform to the Recommended Standard. Non-supported mandatory capabilities are to be identified in the ICS, with an explanation of why the implementation is non-conforming.	Yes [ ] No [ ]

A2.2 REQUIREMENTS LIST

A1.1.1 Image

Table A-1: Image Properties

Item	Description	Reference	Status	Values Allowed	Item Support or Values Supported
I1	Signed Samples	3.2.1	O.1	N/A	
I2	Unsigned Samples	3.2.1	O.1	N/A	
I3	X Size, N_{X}	3.2.2	M	1 .. 2^{16}	
I4	Y Size, N_{Y}	3.2.2	M	1 .. 2^{16}	
I5	Z Size, N_{Z}	3.2.2	M	1 .. 2^{16}	
I6	Dynamic Range, D	3.3.1	M	2 .. 32	
I7	Supplementary Information Tables	3.5	O	N/A	

Table A-2: Supplementary Information Table Features

Prerequisite: I7 – Supplementary Information Tables supported

Item	Description	Reference	Status	Values Allowed	Item Support or Values Supported
S1	Number of Supplementary Information Tables, τ	3.5.2.1	M	1 .. 15	
S2	Table Purpose	3.5.2.2	M	0 .. 4, 10 .. 15	
S3	Unsigned Integer Tables	3.5.2.3	O.2	N/A	
S3.1	Unsigned Integer Table Bit Depth, D_{I}	3.5.2.3.1	C:M	1 .. 32	
S3.2	Unsigned Integer Table Structure	3.5.2.4	C:M	0D, 1D, 2DZX, 2DYX	
S4	Signed Integer Tables	3.5.2.3	O.2	N/A	
S4.1	Signed Integer Table Bit Depth, D_{I}	3.5.2.3.2	C:M	1 .. 32	
S4.2	Signed Integer Table Structure	3.5.2.4	C:M	0D, 1D, 2DZX, 2DYX	
S5	Float Tables	3.5.2.3	O.2	N/A	
S5.1	Float Table Significand Bit Depth, D_{F}	3.5.2.3.3	C:M	1 .. 24	
S5.2	Float Table Exponent Bit Depth, D_{E}	3.5.2.3.3	C:M	1 .. 8	
S5.3	Exponent Bias, β	3.5.2.3.3	C:M	0 .. $2^{D_{\mathrm{E}}} -1$	
S5.4	Float Table Structure	3.5.2.4	C:M	0D, 1D, 2DZX, 2DYX	

A1.1.2 Predictor

Table A-3: Prediction Calculation Features

Item	Description	Reference	Status	Values Allowed	Item Support or Values Supported
P1	Number of Prediction Bands, P	4.2	M	0 .. 15	
P2	Full Prediction Mode	4.3	C.1	N/A	
P3	Reduced Prediction Mode	4.3	C.1	N/A	
P4	Wide Neighbor-Oriented Local Sums	4.4	C.2	N/A	
P5	Narrow Neighbor-Oriented Local Sums	4.4	C.2	N/A	
P6	Wide Column-Oriented Local Sums	4.4	C.2	N/A	
P7	Narrow Column-Oriented Local Sums	4.4	C.2	N/A	
P8	Register Size, R	4.7.2	M	$\max\{32, D+\Omega+2\}$ .. 64	
P9	Sample Representative Resolution, Θ	4.9.1	M	0 .. 4	
P10	Band-Varying Damping	5.3.3.4	O	N/A	
P11	Sample Representative Damping,	4.9.1	M	0 .. $2^{\Theta}-1$	
P12	Band-Varying Offset	5.3.3.4	O	N/A	
P13	Sample Representative Offset, ψ_z	4.9.1	M	0 .. $2^{\Theta}-1$	

C.1: When $N_X = 1$, support is mandatory for Reduced Prediction Mode and not applicable for Full Prediction Mode. Otherwise, it is mandatory to support at least one of these items.

C.2: When $N_X = 1$, support is mandatory for Column-Oriented Local Sums and not applicable for Neighbor-Oriented Local Sums. Otherwise, it is mandatory to support at least one of these items.

Table A-4: Weight Initialization and Update Features

Item	Description	Reference	Status	Values Allowed	Item Support or Values Supported
W1	Weight Component Resolution, Ω	4.6.1	M	4 .. 19	
W2	Default Weight Initialization	4.6.3.2	O.3	N/A	
W3	Custom Weight Initialization	4.6.3.3	O.3	N/A	
W3.1	Weight Initialization Resolution, Q	4.6.3.3	C:M	3 .. Ω+3	
W4	Weight Update Scaling Exponent Initial Parameter, v_{min}	4.10.2	M	−6 .. v_{max}	
W5	Weight Update Scaling Exponent Final Parameter, v_{max}	4.10.2	M	v_{min} .. 9	
W6	Weight Update Scaling Exponent Change Interval, t_{inc}	4.10.2	M	2^4, 2^5, …, 2^{11}	
W7	Intra-Band Weight Exponent Offsets, $\varsigma_z^{(i)}$	4.10.3	M	–6 .. 5	
W8	Inter-Band Weight Exponent Offset, ς_z^*	4.10.3	M	–6 .. 5	

Table A-5: Quantization Features

Item	Description	Reference	Status	Values Allowed	Item Support or Values Supported
Q1	Lossless Fidelity Control Method	4.8.2	O.4	N/A	
Q2	Absolute Error Fidelity Control Method	4.8.2	O.4	N/A	
Q2.1	Absolute Error Limit Bit Depth, D_{A}	4.8.2.2.1	C:M	0 .. $\min\{D-1,15\}$	
Q2.2	Band-Dependent Absolute Error Limits	4.8.2.3.1	C:O.5	N/A	
Q2.2.1	Absolute Error Limits, a_z	4.8.2	C:M	0 .. $2^{D_{\mathrm{A}}}-1$	
Q2.3	Band-Independent Absolute Error Limits	4.8.2.3.1	C:O.5	N/A	
Q2.3.1	Absolute Error Limit Constant, A^*	4.8.2.3.1	C:M	0 .. $2^{D_{\mathrm{A}}}-1$	
Q3	Relative Error Fidelity Control Method	4.8.2	O.4	N/A	
Q3.1	Relative Error Limit Bit Depth, D_{R}	4.8.2.2.2	C:M	0 .. $\min\{D-1,15\}$	
Q3.2	Band-Dependent Relative Error Limits	4.8.2.3.2	C:O.6	N/A	
Q3.2.1	Relative Error Limits, r_z	4.8.2	C:M	0 .. $2^{D_{\mathrm{R}}}-1$	
Q3.3	Band-Independent Relative Error Limits	4.8.2.3.2	C:O.6	N/A	
Q3.3.1	Relative Error Limit Constant, R^*	4.8.2.3.2	C:M	0 .. $2^{D_{\mathrm{R}}}-1$	
Q4	Periodic Error Limit Updating	4.8.2.4	(Q2 or Q3):O	N/A	
Q4.1	Error Limit Update Period Exponent, u	4.8.2.4	C:M	0 .. 8	

A1.1.3 Encoder

Table A-6: Encoder Features

Item	Description	Reference	Status	Values Allowed	Item Support or Values Supported
E1	Output Word Size, B	5.2.2	M	1 .. 8	
E2	BI Encoding Order	5.4.2.2	O.7	N/A	
E2.1	Sub-Frame Interleaving Depth, M	5.4.2.2	C:M	1 .. N_Z	
E3	BSQ Encoding Order	5.4.2.3	O.7	N/A	
E4	Sample-Adaptive Entropy Coder	5.4.3.2	O.8	N/A	
E5	Hybrid Entropy Coder	5.4.3.3	O.8	N/A	
E6	Block-Adaptive Entropy Coder	5.4.3.4	O.8	N/A	

Table A-7: Header Elements

Item	Description	Reference	Status	Item Support
Image Metadata Part				
H1	Essential Subpart	5.3.2.2	M	
H2	Supplementary Information Tables Subpart	5.3.2.3	I7:M	
Predictor Metadata Part				
H3	Primary Subpart	5.3.3.2	M	
H4	Weight Tables Subpart	5.3.3.3	W3:O	
H4.1	Weight Initialization Table Block	5.3.3.3.2	O	
H4.2	Weight Exponent Offset Table Block	5.3.3.3.3	O	
H5	Quantization Subpart	5.3.3.4	(Q2 or Q3):M	
H5.1	Error Limit Update Period Block	5.3.3.4.2	E2:M	
H5.2	Absolute Error Limit Block	5.3.3.4.3	Q2:M	
H5.3	Relative Error Limit Block	5.3.3.4.4	Q3:M	
H6	Sample Representative Subpart	5.3.3.5	C.3	
Entropy Coder Metadata Part				
H7	Sample-Adaptive Entropy Coder Metadata Part	5.3.4.2	E4:M	
H7.1	Accumulator Initialization Table Block	5.3.4.2.2	C:O	
H8	Hybrid Entropy Coder Metadata Part	5.3.4.3	E5:M	
H9	Block-Adaptive Entropy Coder Metadata Part	5.3.4.4	E6:M	

C.3: If the implementation supports nonzero values for any of the sample representative parameters (items P9, P11, P13), then support for this item is mandatory; otherwise, it is not applicable.

Table A-8: Sample-Adaptive Entropy Coder Features

Prerequisite: E4 – Sample-Adaptive Entropy Coder supported					
Item	**Description**	**Reference**	**Status**	**Values Allowed**	**Item Support or Values Supported**
ES1	Initial Count Exponent, γ_0	5.4.3.2.3.2	M	1 .. 8	
ES2	Accumulator Initialization Table, k_z''	5.4.3.2.3.3	O.9	0 .. $\max\{D-2,14\}$	
ES3	Accumulator Initialization Constant, K	5.4.3.2.3.3	O.9	0 .. $\max\{D-2,14\}$	
ES4	Rescaling Counter Size, γ^*	5.4.3.2.3.4	M	$\max\{4, \gamma_0 + 1\}$ .. 11	
ES5	Unary Length Limit, $U_{\max}$	5.4.3.2.2	M	8 .. 32	

Table A-9: Hybrid Entropy Coder Features

Prerequisite: E5 – Hybrid Entropy Coder supported					
Item	**Description**	**Reference**	**Status**	**Values Allowed**	**Item Support or Values Supported**
EH1	Initial Count Exponent, γ_0	5.4.3.3.4.2	M	1 .. 8	
EH2	Initial High-Resolution Accumulator Value, $\tilde{\Sigma}_z(0)$	5.4.3.3.4.3	M	0 .. $2^{D+\gamma_0}$	
EH3	Rescaling Counter Size, γ^*	5.4.3.3.4.4	M	$\max\{4, \gamma_0 + 1\}$ .. 11	
EH4	Unary Length Limit, $U_{\max}$	5.4.3.3.3.2.2	M	8 .. 32	

Table A-10: Block-Adaptive Entropy Coder Features

Prerequisite: E6 – Block-Adaptive Entropy Coder supported					
Item	**Description**	**Reference**	**Status**	**Values Allowed**	**Item Support or Values Supported**
EB1	Block Size, *J*	5.4.3.4.2.4	M	8, 16, 32, 64	
EB2	Reference Sample Interval, *r*	5.4.3.4.2.5	M	1 .. 4096	
EB3	Basic Code Options	5.4.3.4.2.6	O.10	N/A	
EB4	Restricted Code Options	5.4.3.4.2.6	O.10	N/A	

ANNEX B

LOW-ENTROPY CODE TABLES

(NORMATIVE)

Tables B-1–B-32 list the code tables and flush tables that define the encoding of low-entropy samples, as described in 5.4.3.3.3.3. The following conventions are used in the tables:

- In flush tables, (null) indicates an empty active prefix.
- The notation 0^{i} is used to denote i consecutive occurrences of input symbol 0. By convention, 0^{0} is interpreted as the empty sequence.
- The notation n'hX denotes an output codeword having length n bits, where X is the hexadecimal representation of that codeword. Thus, for example, 7'h1F denotes the codeword 0011111.
- The notation n'h(X+2r) or n'h(X+r) denotes an output codeword having length n bits, where the quantity in parentheses evaluates to the hexadecimal representation of that codeword. It should be noted that the quantity X is written in hexadecimal. Angle brackets <> indicates the reversal of a binary codeword. Thus, for example, <7'h1F> denotes the codeword 1111100.

For example, an entry in table B-27 for low entropy code 13 indicates that input codeword 0^{r}1, $1 \leq r \leq 2$ produces output codeword <7'h(3F+2r)>. Thus, for this code, input codeword 001 is encoded via output codeword <7'h(3F+4)> = <7'h43> = <1000011> = 1100001.

Table B-1: Code Table for Low-Entropy Code 0

Input Codeword	Output Codeword	Input Codeword	Output Codeword	Input Codeword	Output Codeword
00	5'h19	09	8’h3B	5	4'h6
010	8'h57	0A	8’hBB	6	4'hE
011	8'hD7	0B	9’h00F	70	7'h0B
012	8'h37	0C	9’h10F	71	7'h4B
013	9'h0AF	0X	8’h7B	72	7'h2B
014	9'h1AF	1	3’h0	73	8'h47
015	9'h06F	2	3’h4	74	8'hC7
016	9'h16F	3	3’h2	75	8'h27
017	10'h03F	40	6’h1D	76	8'hA7
018	10'h23F	41	6’h3D	77	9'h0CF
019	11'h17F	42	6’h03	78	9'h1CF
01A	11'h57F	43	6’h23	79	10'h15F
01B	12'h1FF	440	9’h09F	7A	10'h35F
01C	12'h9FF	441	9’h19F	7B	11'h07F
01X	11'h37F	442	9’h05F	7C	11'h47F
020	8'hB7	443	10’h0BF	7X	10'h0DF
021	8'h77	444	10’h2BF	80	7'h6B
022	8'hF7	445	10’h1BF	81	7'h1B
023	9'h0EF	446	10’h3BF	82	7'h5B
024	9'h1EF	447	11’h2FF	83	8'h67
025	9'h01F	448	11’h6FF	84	8'hE7
026	9'h11F	449	12’h3FF	85	8'h17
027	10'h13F	44A	12’hBFF	86	8'h97
028	10'h33F	44B	13’h0FFF	87	9'h02F
029	11'h77F	44C	13’h1FFF	88	9'h12F
02A	11'h0FF	44X	12’h7FF	89	10'h2DF
02B	12'h5FF	45	7’h33	8A	10'h1DF
02C	12'hDFF	46	7’h73	8B	11'h27F
02X	11'h4FF	47	8’hFB	8C	11'h67F
03	6'h15	48	8’h07	8X	10'h3DF
04	6'h35	49	9’h08F	9	5'h01
05	6'h0D	4A	9’h18F	A	5'h11
06	6'h2D	4B	9’h04F	B	6'h05
07	7'h13	4C	9’h14F	C	6'h25
08	7'h53	4X	8’h87	X	5'h09

Table B-2: Flush Table for Low-Entropy Code 0

Active Prefix	Flush Word	Active Prefix	Flush Word	Active Prefix	Flush Word
(null)	1'h0	02	6’h1F	7	5'h07
0	2'h1	4	3’h3	8	5'h17
01	5'h0F	44	6’h3F		

Table B-3: Code Table for Low-Entropy Code 1

Input Codeword	Output Codeword	Input Codeword	Output Codeword	Input Codeword	Output Codeword
000	7'h73	130	8’h87	233	9'h0AF
001	7'h0B	131	8’h47	234	9'h1AF
002	7'h4B	132	8’hC7	235	10'h2DF
003	8'h9B	133	9’h08F	236	10'h1DF
004	8'h5B	134	9’h18F	237	11'h07F
005	9'h137	135	10’h19F	238	11'h47F
006	9'h0B7	136	10’h39F	239	12'h6FF
007	10'h11F	137	11’h6BF	23A	12'hEFF
008	10'h31F	138	11’h1BF	23X	12'h1FF
009	11'h63F	139	12’h77F	24	6'h23
00A	11'h13F	13A	12’hF7F	250	9'h06F
00X	11'h53F	13X	12’h0FF	251	9'h16F
01	5'h0E	14	6’h03	252	9'h0EF
02	5'h1E	150	9’h04F	253	10'h3DF
030	8'hDB	151	9’h14F	254	10'h03F
031	8'h3B	152	9’h0CF	255	11'h27F
032	8'hBB	153	10’h05F	256	11'h67F
033	9'h1B7	154	10’h25F	257	12'h9FF
034	9'h077	155	10’h15F	258	12'h5FF
035	9'h177	156	11’h5BF	259	13'h17FF
036	9'h0F7	157	12’h8FF	25A	13'h0FFF
037	10'h09F	158	12’h4FF	25X	13'h1FFF
038	11'h33F	159	13’h03FF	26	7'h33
039	12'h57F	15A	13’h13FF	27	8'hEB
03A	12'hD7F	15X	12’hCFF	28	8'h1B
03X	11'h73F	160	9’h1CF	29	9'h197
040	8'h7B	161	9’h02F	2A	9'h057
041	8'hFB	162	9’h12F	2X	9'h157
042	8'h07	163	10’h35F	3	3'h0
043	9'h1F7	164	10’h0DF	4	3'h4
044	9'h00F	165	11’h3BF	5	4'h2
045	9'h10F	166	11’h7BF	6	4'hA
046	10'h29F	167	12’h2FF	7	5'h06
047	11'h0BF	168	12’hAFF	8	5'h16
048	11'h4BF	169	13’h0BFF	9	6'h0D
049	12'h37F	16A	13’h1BFF	A0	9'h0D7
04A	12'hB7F	16X	13’h07FF	A1	9'h1D7
04X	11'h2BF	17	8’hAB	A2	9'h037
05	6'h1D	18	8’h6B	A3	10'h1EF
06	6'h3D	19	9’h017	A4	10'h3EF
07	7'h13	1A	9’h117	A5	10'h01F
08	7'h53	1X	9’h097	A6	10'h21F
09	9'h0E7	20	5’h19	A7	11'h23F
0A	9'h1E7	21	5’h05	A8	12'h17F
0X	8'h2B	22	5’h15	A9	13'h0DFF
10	5'h01	230	8’h27	AA	13'h1DFF
11	5'h11	231	8’hA7	AX	12'h97F
12	5'h09	232	8’h67	X	6'h2D

Table B-4: Flush Table for Low-Entropy Code 1

Active Prefix	Flush Word	Active Prefix	Flush Word	Active Prefix	Flush Word
(null)	1'h0	1	3’h5	23	6'h2F
0	3'h1	13	6’h0F	25	7'h7F
00	5'h07	15	7’h5F	A	7'h1F
03	6'h17	16	7’h3F		
04	6'h37	2	3’h3		

Table B-5: Code Table for Low-Entropy Code 2

Input Codeword	Output Codeword	Input Codeword	Output Codeword	Input Codeword	Output Codeword
0	2'h0	148	12’h1FF	242	8'h77
10	4'h1	14X	12’h9FF	243	9'h05F
110	7'h7D	15	7’h35	244	9'h15F
111	7'h03	16	7’h75	245	10'h13F
112	7'h43	17	8’h6B	246	10'h33F
113	8'h3B	18	9’h0F7	247	12'hBFF
114	8'hBB	1X	9’h1F7	248	12'h7FF
115	9'h0CF	200	6’h15	24X	12'hFFF
116	9'h1CF	201	7’h33	25	7'h0D
117	11'h3BF	202	7’h73	26	7'h4D
118	11'h7BF	203	8’hA7	27	8'hEB
11X	11'h07F	204	8’h67	28	9'h00F
120	7'h23	205	9’h0EF	2X	9'h10F
121	7'h63	206	9’h1EF	3	3'h2
122	7'h13	207	11’h17F	4	3'h6
123	8'h7B	208	11’h57F	50	6'h25
124	8'hFB	20X	11’h37F	51	7'h2D
125	9'h02F	210	7’h0B	52	7'h6D
126	9'h12F	211	7’h4B	53	8'h1B
127	11'h47F	212	7’h2B	54	8'h9B
128	11'h27F	213	8’hE7	55	9'h08F
12X	11'h67F	214	8’h17	56	9'h18F
130	7'h53	215	9’h01F	57	11'h0BF
131	8'h07	216	9’h11F	58	11'h4BF
132	8'h87	217	11’h77F	5X	11'h2BF
133	9'h0AF	218	11’h0FF	60	7'h1D
134	9'h1AF	21X	11’h4FF	61	7'h5D
135	10'h0DF	22	5’h09	62	7'h3D
136	10'h2DF	230	8’h97	63	8'h5B
137	12'h2FF	231	8’h57	64	8'hDB
138	12'hAFF	232	8’hD7	65	9'h04F
13X	12'h6FF	233	9’h09F	66	9'h14F
140	8'h47	234	9’h19F	67	11'h6BF
141	8'hC7	235	10’h03F	68	11'h1BF
142	8'h27	236	10’h23F	6X	11'h5BF
143	9'h06F	237	12’h5FF	7	6'h19
144	9'h16F	238	12’hDFF	8	6'h39
145	10'h1DF	23X	12’h3FF	X	6'h05
146	10'h3DF	240	8’h37		
147	12'hEFF	241	8’hB7		

Table B-6: Flush Table for Low-Entropy Code 2

Active Prefix	Flush Word	Active Prefix	Flush Word	Active Prefix	Flush Word
(null)	1'h0	14	6’h2F	24	6'h3F
1	3'h1	2	3’h5	5	5'h03
11	5'h0B	20	5’h07	6	5'h13
12	5'h1B	21	5’h17		
13	6'h0F	23	6’h1F		

Table B-7: Code Table for Low-Entropy Code 3

Input Codeword	Output Codeword	Input Codeword	Output Codeword	Input Codeword	Output Codeword
000	5'h19	212	7’h63	316	11'h47F
0010	7'h4B	213	8’h3B	31X	12'h6FF
0011	8'h17	214	8’hBB	320	7'h73
0012	8'h97	215	10’h2DF	321	8'h27
0013	9'h0EF	216	10’h1DF	322	8'hA7
0014	9'h1EF	21X	11’h1BF	323	9'h0AF
0015	11'h37F	220	6’h0D	324	9'h1AF
0016	11'h77F	2210	8’hB7	325	11'h27F
001X	12'h9FF	2211	9’h05F	326	11'h67F
0020	7'h2B	2212	9’h15F	32X	12'hEFF
0021	8'h57	2213	10’h33F	33	7'h1D
0022	8'hD7	2214	10’h0BF	34	7'h5D
0023	9'h01F	2215	12’hBFF	35	9'h077
0024	9'h11F	2216	12’h7FF	36	9'h177
0025	11'h0FF	221X	13’h1FFF	3X	9'h0F7
0026	11'h4FF	222	7’h13	40	5'h09
002X	12'h5FF	223	8’h7B	410	7'h0B
003	7'h43	224	8’hFB	411	8'h67
004	7'h23	225	10’h3DF	412	8'hE7
005	9'h18F	226	10’h03F	413	9'h06F
006	9'h04F	22X	11’h5BF	414	9'h16F
00X	9'h14F	230	7’h53	415	11'h17F
01	4'h2	231	8’h07	416	11'h57F
02	4'hA	232	8’h87	41X	12'h1FF
03	5'h1E	233	9’h0CF	42	6'h15
04	5'h01	234	9’h1CF	43	7'h3D
05	7'h2D	235	11’h3BF	44	7'h7D
06	7'h6D	236	11’h7BF	45	9'h1F7
0X	8'h6B	23X	12’hAFF	46	9'h00F
1	2'h0	24	6’h25	4X	10'h0DF
20	4'h6	25	8’hEB	5	5'h0E
210	6'h35	26	8’h1B	60	7'h03
2110	8'h37	2X	8’h9B	61	8'h5B
2111	9'h09F	30	5’h11	62	8'hDB
2112	9'h19F	310	7’h33	63	9'h10F
2113	10'h23F	311	8’h47	64	9'h08F
2114	10'h13F	312	8’hC7	65	11'h2BF
2115	12'hDFF	313	9’h02F	66	11'h6BF
2116	12'h3FF	314	9’h12F	6X	12'h2FF
211X	13'h0FFF	315	11’h07F	X	6'h05

Table B-8: Flush Table for Low-Entropy Code 3

Active Prefix	Flush Word	Active Prefix	Flush Word	Active Prefix	Flush Word
(null)	1'h0	21	6'h17	31	7'h2F
0	3'h1	211	8'h7F	32	7'h6F
00	4'hB	22	6'h37	4	5'h07
001	7'h5F	221	8'hFF	41	7'h1F
002	7'h3F	23	7'h4F	6	7'h0F
2	3'h5	3	4'h3		

Table B-9: Code Table for Low-Entropy Code 4

Input Codeword	Output Codeword	Input Codeword	Output Codeword	Input Codeword	Output Codeword
000	4'h5	0120	7'h7B	024	8'h57
0010	6'h03	0121	8'h8F	025	10'h1DF
0011	7'h2B	0122	8'h4F	026	11'h57F
0012	7'h6B	0123	10'h0BF	02X	11'h37F
0013	9'h11F	0124	10'h2BF	030	7'h0B
0014	9'h09F	0125	12'h1FF	031	8'hD7
0015	12'h8FF	0126	13'h13FF	032	8'h37
0016	12'h4FF	012X	13'h0BFF	033	9'h01F
001X	12'hCFF	013	8'hE7	034	10'h3DF
002	5'h0D	014	8'h17	035	12'h77F
003	7'h33	015	10'h0DF	036	12'hF7F
004	7'h73	016	10'h2DF	03X	13'h05FF
005	9'h1EF	01X	11'h17F	040	7'h4B
006	10'h15F	020	5'h1D	041	8'hB7
00X	10'h35F	0210	7'h07	042	8'h77
0100	6'h23	0211	8'hCF	043	10'h03F
0101	7'h1B	0212	8'h2F	044	10'h23F
0102	7'h5B	0213	10'h1BF	045	12'h0FF
0103	9'h19F	0214	10'h3BF	046	13'h15FF
0104	9'h05F	0215	12'h9FF	04X	13'h0DFF
0105	12'h2FF	0216	13'h1BFF	05	8'hA7
0106	12'hAFF	021X	13'h07FF	06	8'h67
010X	12'h6FF	0220	7'h47	0X	9'h0EF
0110	7'h3B	0221	8'hAF	1	2'h0
0111	8'hF7	0222	8'h6F	2	2'h2
0112	8'h0F	0223	10'h07F	3	4'h1
0113	10'h13F	0224	10'h27F	4	4'h9
0114	10'h33F	0225	13'h17FF	5	7'h13
0115	12'hEFF	0226	13'h0FFF	6	7'h53
0116	13'h1DFF	022X	13'h1FFF	X	8'h27
011X	13'h03FF	023	8'h97		

Table B-10: Flush Table for Low-Entropy Code 4

Active Prefix	Flush Word	Active Prefix	Flush Word	Active Prefix	Flush Word
(null)	1'h0	010	6'h0F	022	7'h7F
0	2'h1	011	6'h2F	03	6'h17
00	4'h3	012	7'h5F	04	7'h1F
001	6'h37	02	5'h07		
01	4'hB	021	7'h3F		

Table B-11: Code Table for Low-Entropy Code 5

Input Codeword	Output Codeword	Input Codeword	Output Codeword	Input Codeword	Output Codeword
0	1'h0	20200	8'hD7	2123	12'hE7F
1	2'h1	20201	9'h0AF	2124	12'h17F
2000	5'h0B	20202	9'h1AF	212X	14'h1BFF
200100	9'h01F	20203	12'h8FF	213	10'h39F
200101	10'h1DF	20204	12'h4FF	214	10'h05F
200102	10'h3DF	2020X	14'h37FF	21X	11'h0BF
200103	13'h05FF	2021	8'h87	22000	8'h37
200104	13'h15FF	2022	8'h47	22001	9'h0EF
20010X	14'h1FFF	2023	11'h1BF	22002	9'h1EF
20011	9'h04F	2024	11'h5BF	22003	12'hAFF
20012	9'h14F	202X	13'h11FF	22004	12'h6FF
20013	12'hD7F	203	8'h7B	2200X	14'h2FFF
20014	12'h37F	204	9'h1B7	2201	8'hE7
2001X	14'h07FF	20X	10'h19F	2202	8'h17
20020	8'h57	210000	9'h09F	2203	11'h07F
20021	9'h0CF	210001	10'h13F	2204	11'h47F
20022	9'h1CF	210002	10'h33F	220X	13'h19FF
20023	12'hB7F	210003	13'h03FF	2210	8'h97
20024	12'h77F	210004	13'h13FF	2211	9'h08F
2002X	14'h27FF	21000X	15'h7FFF	2212	9'h18F
2003	9'h177	21001	9'h06F	2213	12'h97F
2004	10'h2DF	21002	9'h16F	2214	12'h57F
200X	11'h4BF	21003	12'hCFF	221X	14'h3BFF
201000	9'h11F	21004	12'h2FF	222	7'h3B
201001	10'h03F	2100X	14'h0FFF	223	10'h25F
201002	10'h23F	2101	8'hC7	224	10'h15F
201003	13'h0DFF	2102	8'h27	22X	12'h27F
201004	13'h1DFF	2103	11'h3BF	23	7'h5B
20100X	15'h3FFF	2104	11'h7BF	240	9'h077
20101	9'h02F	210X	13'h09FF	241	10'h35F
20102	9'h12F	2110	8'hA7	242	10'h0DF
20103	12'hF7F	2111	9'h0F7	243	13'h0EFF
20104	12'h0FF	2112	9'h1F7	244	13'h1EFF
2010X	14'h17FF	2113	12'hA7F	24X	14'h0BFF
2011	8'hFB	2114	12'h67F	2X	9'h0B7
2012	8'h07	211X	14'h2BFF	3	5'h03
2013	11'h2BF	2120	8'h67	4	5'h13
2014	11'h6BF	2121	9'h00F	X	7'h1B
201X	13'h01FF	2122	9'h10F		

Table B-12: Flush Table for Low-Entropy Code 5

Active Prefix	Flush Word	Active Prefix	Flush Word	Active Prefix	Flush Word
(null)	1'h0	2010	7'h6F	211	7'h37
2	3'h1	20100	8'h7F	212	7'h77
20	3'h5	202	6'h3B	22	5'h13
200	5'h0B	2020	7'h1F	220	6'h27
2001	7'h4F	21	5'h03	2200	7'h5F
20010	8'hBF	210	6'h07	221	7'h0F
2002	7'h2F	2100	6'h17	24	8'h3F
201	6'h1B	21000	8'hFF		

Table B-13: Code Table for Low-Entropy Code 6

Input Codeword	Output Codeword	Input Codeword	Output Codeword	Input Codeword	Output Codeword
0000	3'h2	0013	10'h2BF	12	5'h09
000100	6'h13	0014	11'h07F	13	9'h0EF
000101	8'h4F	001X	13'h1DFF	14	9'h1EF
000102	8'hCF	002	4'h6	1X	11'h1BF
000103	12'h6FF	003	8'h57	200	4'h1
000104	12'hEFF	004	8'hD7	20100	7'h17
00010X	14'h2FFF	00X	10'h03F	20101	9'h0DF
00011	7'h3B	01	3'h0	20102	9'h1DF
00012	7'h7B	020	4'hE	20103	13'h1BFF
00013	11'h47F	0210	6'h03	20104	13'h07FF
00014	11'h27F	0211	8'h37	2010X	15'h7FFF
0001X	13'h03FF	0212	8'hB7	2011	8'h0F
000200	6'h33	0213	12'hF7F	2012	8'h8F
000201	8'h2F	0214	12'h0FF	2013	12'hCFF
000202	8'hAF	021X	14'h17FF	2014	12'h2FF
000203	12'h1FF	022	6'h0D	201X	14'h0FFF
000204	12'h9FF	023	10'h23F	202	6'h2D
00020X	14'h1FFF	024	10'h13F	203	10'h33F
00021	7'h07	02X	12'h37F	204	10'h0BF
00022	7'h47	03	7'h4B	20X	12'h77F
00023	11'h67F	04	7'h2B	21	5'h19
00024	12'hAFF	0X	9'h16F	22	5'h05
0002X	13'h13FF	10	3'h4	23	9'h01F
0003	9'h05F	1100	6'h23	24	9'h11F
0004	9'h15F	1101	8'h77	2X	11'h5BF
000X	11'h7BF	1102	8'hF7	30	7'h6B
00100	5'h15	1103	12'h8FF	31	9'h09F
00101	7'h27	1104	12'h4FF	32	9'h19F
00102	7'h67	110X	14'h37FF	33	13'h05FF
00103	11'h17F	111	7'h1B	34	13'h15FF
00104	11'h57F	112	7'h5B	3X	15'h3FFF
0010X	13'h0BFF	113	11'h3BF	4	7'h0B
0011	6'h1D	114	12'hB7F	X	9'h06F
0012	6'h3D	11X	13'h0DFF		

Table B-14: Flush Table for Low-Entropy Code 6

Active Prefix	Flush Word	Active Prefix	Flush Word	Active Prefix	Flush Word
(null)	2'h0	00020	7'h3F	110	7'h6F
0	2'h2	001	6'h27	2	4'hD
00	3'h1	0010	6'h0F	20	5'h1B
000	4'h3	02	5'h0B	201	7'h1F
0001	6'h17	021	7'h2F	2010	8'hFF
00010	7'h5F	1	4'h5	3	8'h7F
0002	6'h37	11	6'h07		

Table B-15: Code Table for Low-Entropy Code 7

Input Codeword	Output Codeword	Input Codeword	Output Codeword	Input Codeword	Output Codeword
00	1'h0	0223	15'h27FF	103	12'h6FF
0100	4'hD	0224	15'h67FF	104	12'hEFF
01010	7'h77	022X	16'h5FFF	10X	14'h0BFF
01011	9'h15F	023	12'h2FF	110	6'h17
01012	9'h0DF	024	12'hAFF	111	8'h6F
01013	15'h17FF	02X	14'h33FF	1120	9'h05F
01014	15'h57FF	03	9'h1EF	1121	11'h77F
0101X	17'h07FFF	04	9'h01F	1122	11'h0FF
01020	7'h0F	0X	11'h27F	1123	16'hDFFF
01021	9'h1DF	1000	4'h3	1124	17'h1BFFF
01022	10'h3BF	10010	7'h4F	112X	18'h3FFFF
01023	15'h37FF	10011	9'h03F	113	14'h2BFF
01024	15'h77FF	10012	9'h13F	114	14'h1BFF
0102X	17'h17FFF	10013	15'h0FFF	11X	16'h9FFF
0103	12'h1FF	10014	15'h4FFF	12	6'h0B
0104	12'h9FF	1001X	16'h3FFF	13	11'h67F
010X	14'h3BFF	10020	7'h2F	14	11'h17F
011	6'h2B	10021	9'h0BF	1X	13'h03FF
012	6'h1B	10022	10'h07F	2	3'h1
013	12'h4FF	10023	15'h2FFF	30	9'h11F
014	12'hCFF	10024	15'h6FFF	31	11'h57F
01X	14'h13FF	1002X	17'h0FFFF	32	11'h37F
020	4'h5	1003	12'h5FF	33	16'h1FFF
021	6'h3B	1004	12'hDFF	34	17'h0BFFF
0220	7'h37	100X	14'h07FF	3X	18'h1FFFF
0221	9'h09F	101	6'h07	4	9'h0EF
0222	9'h19F	102	6'h27	X	10'h1BF

Table B-16: Flush Table for Low-Entropy Code 7

Active Prefix	Flush Word	Active Prefix	Flush Word	Active Prefix	Flush Word
(null)	1'h0	02	5'h1B	1002	8'h7F
0	2'h1	022	8'h5F	11	7'h1F
01	5'h0B	1	4'h3	112	9'h1FF
010	5'h17	10	5'h07	3	9'h0FF
0101	8'hDF	100	5'h0F		
0102	8'h3F	1001	8'hBF		

Table B-17: Code Table for Low-Entropy Code 8

Input Codeword	Output Codeword	Input Codeword	Output Codeword	Input Codeword	Output Codeword
0^{10}	3'h0	0^{6}21	9'h0EF	00X	10'h03F
0^{9}1	6'h0B	0^{6}22	9'h1EF	010	4'hA
0^{9}200	7'h37	0^{6}2X	14'h17FF	011	7'h3B
0^{9}201	10'h27F	0^{6}X	11'h4FF	012	7'h7B
0^{9}202	10'h17F	000001	5'h09	01X	13'h03FF
0^{9}20X	15'h3FFF	000002	5'h19	02	4'h4
0^{9}21	10'h3BF	00000X	11'h37F	0X	9'h16F
0^{9}22	10'h07F	000010	5'h05	100	4'h6
0^{9}2X	15'h5FFF	000011	8'hF7	101	7'h07
0^{9}X	12'hDFF	000012	8'h0F	102	7'h47
0^{8}1	6'h03	00001X	13'h1BFF	10X	12'h5FF
0^{8}2	6'h23	000020	5'h15	110	7'h27
0^{8}X	11'h6FF	000021	8'h8F	111	10'h23F
0^{7}100	6'h2B	000022	8'h4F	112	10'h13F
0^{7}101	9'h0DF	00002X	14'h07FF	11X	15'h2FFF
0^{7}102	9'h1DF	0000X	10'h1BF	12	7'h1B
0^{7}10X	15'h1FFF	0001	4'h1	1X	12'h1FF
0^{7}11	9'h01F	000200	5'h0D	200	4'hE
0^{7}12	9'h11F	000201	8'hCF	201	7'h67
0^{7}1X	14'h37FF	000202	8'h2F	202	7'h17
0^{7}20	6'h13	00020X	14'h27FF	20X	13'h13FF
0^{7}21	9'h09F	000210	8'hAF	210	7'h57
0^{7}22	9'h19F	000211	11'h77F	211	10'h33F
0^{7}2X	15'h6FFF	000212	11'h0FF	212	10'h0BF
0^{7}X	11'h2FF	00021X	16'hFFFF	21X	16'h7FFF
0^{6}1	5'h1D	00022	8'h77	22	7'h5B
0^{6}200	6'h33	0002X	13'h0BFF	2X	12'h9FF
0^{6}201	9'h05F	000X	10'h2BF	X	9'h06F
0^{6}202	9'h15F	001	4'hC		
0^{6}20X	14'h0FFF	002	4'h2		

Table B-18: Flush Table for Low-Entropy Code 8

Active Prefix	Flush Word	Active Prefix	Flush Word	Active Prefix	Flush Word
(null)	2'h0	0^{9}2	9'h17F	00020	7'h6F
0	3'h2	0^{9}20	9'h0FF	00021	10'h3FF
00	3'h6	0^{7}1	8'h5F	01	6'h17
000	3'h1	0^{7}10	8'hBF	1	6'h07
0000	4'h5	0^{7}2	8'hDF	10	6'h37
00000	4'hD	0^{6}2	8'h9F	11	9'h07F
0^{6}	5'h03	0^{6}20	8'h3F	2	6'h27
0^{7}	5'h13	00001	7'h2F	20	7'h0F
0^{8}	5'h0B	00002	8'h1F	21	10'h1FF
0^{9}	5'h1B	0002	7'h4F		

Table B-19: Code Table for Low-Entropy Code 9

Input Codeword	Output Codeword	Input Codeword	Output Codeword	Input Codeword	Output Codeword
0^{11}	2'h0	0^{7}21	9'h01F	000X	11'h2FF
0^{10}10	6'h3B	0^{7}22	10'h15F	001000	5'h0D
0^{10}11	10'h1BF	0^{7}2X	16'hEFFF	001001	9'h16F
0^{10}12	10'h3BF	0^{7}X	12'h5FF	001002	9'h0EF
0^{10}1X	17'h17FFF	0^{6}1	5'h1D	00100X	15'h4FFF
0^{10}2	6'h1B	0^{6}2000	6'h2B	00101	8'h77
0^{10}X	12'hBFF	0^{6}2001	10'h23F	00102	9'h04F
0^{9}100	6'h07	0^{6}2002	10'h13F	0010X	15'h57FF
0^{9}101	10'h07F	0^{6}200X	16'hDFFF	0011	8'h97
0^{9}102	10'h27F	0^{6}201	9'h11F	0012	8'h57
0^{9}10X	17'h0FFFF	0^{6}202	10'h35F	001X	15'h47FF
0^{9}11	10'h33F	0^{6}20X	16'h1FFF	00200	5'h01
0^{9}12	10'h0BF	0^{6}21	9'h1EF	00201	9'h14F
0^{9}1X	16'h3FFF	0^{6}22	10'h25F	00202	9'h0CF
0^{9}2	6'h13	0^{6}2X	16'h6FFF	0020X	16'hAFFF
0^{9}X	12'h3FF	0^{6}X	12'h9FF	0021	8'hD7
0^{8}1000	6'h27	000001	5'h09	0022	9'h1F7
0^{8}1001	10'h17F	000002	5'h19	002X	15'h27FF
0^{8}1002	10'h37F	00000X	12'h1FF	00X	11'h4FF
0^{8}100X	17'h1FFFF	000010	5'h05	01	4'h6
0^{8}101	9'h19F	000011	9'h12F	02000	5'h11
0^{8}102	10'h2BF	000012	9'h0AF	02001	9'h1CF
0^{8}10X	16'hBFFF	00001X	15'h77FF	02002	9'h02F
0^{8}11	9'h09F	00002	5'h0E	0200X	15'h37FF
0^{8}12	10'h0DF	0000X	11'h6FF	0201	8'h37
0^{8}1X	16'h9FFF	000100	5'h15	0202	9'h00F
0^{8}20	6'h33	000101	9'h1AF	020X	15'h67FF
0^{8}21	10'h2DF	000102	9'h06F	021	8'h17
0^{8}22	10'h1DF	00010X	15'h0FFF	022	9'h0F7
0^{8}2X	17'h07FFF	00011	8'hB7	02X	15'h07FF
0^{8}X	12'hDFF	00012	9'h10F	0X	11'h0FF
0^{7}1	5'h03	0001X	15'h17FF	1	4'h2
0^{7}200	6'h0B	00020	5'h1E	2	4'hA
0^{7}201	10'h3DF	00021	9'h08F	X	10'h05F
0^{7}202	10'h03F	00022	9'h18F		
0^{7}20X	16'h5FFF	0002X	16'h2FFF		

Table B-20: Flush Table for Low-Entropy Code 9

Active Prefix	Flush Word	Active Prefix	Flush Word
0^{r}, 0≤r≤3	<3'h(0+r)>	0^{r}10, 2≤r≤3	<7'h(71+2r)>
0^{r}, 4≤r≤8	<4'h(4+r)>	0^{8}10	8'hBF
0^{r}, 9≤r≤10	<5'h(11+r)>	0^{9}10	9'h1FF
001	7'h47	020	7'h67
0^{r}1, 3≤r≤4	<7'h(6E+2r)>	0020	8'hCF
0^{8}1	8'h1F	0^{6}20	8'hEF
0^{9}1	8'h3F	0^{7}20	8'h5F
0^{10}1	9'h0FF	00100	7'h0F
0^{r}2, 1≤r≤2	<7'h(6E+2r)>	0^{8}100	8'h7F
0^{r}2, 6≤r≤7	<8'h(EF+r)>	0200	8'h2F
0^{8}2	8'h9F	0^{6}200	8'hDF

Table B-21: Code Table for Low-Entropy Code 10

Input Codeword	Output Codeword	Input Codeword	Output Codeword
0^{9}	1'h0	0^{7}102	10'h13F
0001	5'h19	0^{8}102	11'h4FF
0^{r}1, 4≤r≤6	<5'h(14+r)>	0^{r}10X, 0≤r≤1	<17'h(1FFF2+2r)>
0^{r}2, 0≤r≤2	<5'h(10+r)>	0^{r}10X, 7≤r≤8	<18'h(3FFEC+2r)>
0002	5'h05	0^{6}200	6'h1B
0^{r}X, 0≤r≤1	<12'h(FFA+r)>	0^{7}200	6'h17
0^{r}X, 2≤r≤8	<13'h(1FF6+r)>	0000201	10'h31F
0010	5'h15	00000201	10'h19F
0^{r}11, 0≤r≤1	<9'h(1E8+r)>	0^{6}201	10'h35F
0011	10'h26F	0^{7}201	10'h33F
0^{7}11	10'h05F	0000202	10'h09F
0^{8}11	10'h1DF	00000202	11'h47F
0^{r}12, 0≤r≤1	<10'h(3D6+r)>	0^{6}202	11'h17F
0012	10'h16F	0^{7}202	11'h0FF
0^{7}12	10'h25F	0^{r}20X, 4≤r≤5	<18'h(3FFE7+2r)>
0^{8}12	10'h3DF	0^{6}20X	18'h2BFFF
1X	17'h01FFF	0^{7}20X	18'h37FFF
0^{r}1X, 1≤r≤2	<17'h(1FFEF+2r)>	1000	5'h1D
0^{7}1X	18'h33FFF	0^{8}1000	6'h37
0^{8}1X	18'h07FFF	1001	10'h2EF
0^{8}20	6'h27	0^{r}1001, 7≤r≤8	<10'h(3EE+r)>
0^{r}21, 4≤r≤5	<10'h(3D7+2r)>	1002	10'h1EF
0^{6}21	10'h29F	0^{r}1002, 7≤r≤8	<11'h(7F3+r)>
0^{7}21	10'h15F	100X	17'h15FFF
0^{8}21	10'h03F	0^{r}100X, 7≤r≤8	<18'h(3FFF6+r)>
0^{r}22, 4≤r≤5	<10'h(3D8+2r)>	000002000	6'h3B
0^{6}22	11'h07F	00002001	10'h39F
0^{7}22	11'h67F	000002001	10'h0DF
0^{8}22	11'h77F	00002002	11'h27F
00002X	17'h0DFFF	000002002	11'h57F
0^{r}2X, 5≤r≤6	<18'h(3FFE4+2r)>	0000200X	18'h13FFF
0^{7}2X	18'h0BFFF	00000200X	18'h1BFFF
0^{8}2X	18'h27FFF	0^{7}10000	6'h0F
0100	5'h0D	0^{7}10001	10'h3BF
101	9'h0AF	0^{7}10002	11'h1FF
0101	10'h36F	0^{7}1000X	18'h3FFFF
0^{7}101	10'h23F	000020000	6'h07
0^{8}101	10'h0BF	000020001	10'h2DF
102	10'h06F	000020002	11'h37F
0102	10'h0EF	00002000X	18'h3BFFF

Table B-22: Flush Table for Low-Entropy Code 10

Active Prefix	Flush Word	Active Prefix	Flush Word
0^{r}, 0≤r≤5	<3'h(0+r)>	0^{r}10, 7≤r≤8	<9'h(1EC+2r)>
0^{r}, 6≤r≤8	<4'h(6+r)>	0^{r}20, 4≤r≤5	<9'h(1E7+2r)>
1	8'h0F	0^{6}20	9'h15F
0^{r}1, 1≤r≤2	<8'h(EF+2r)>	0^{7}20	9'h1BF
0^{7}1	9'h19F	100	8'hAF
0^{8}1	9'h03F	0^{r}100, 7≤r≤8	<9'h(1F6+r)>
00002	8'h6F	0000200	9'h09F
0^{r}2, 5≤r≤6	<9'h(1E4+2r)>	00000200	9'h0DF
0^{7}2	9'h05F	0^{7}1000	9'h1FF
0^{8}2	9'h13F	00002000	9'h1DF
0^{r}10, 0≤r≤1	<8'h(F2+2r)>		

Table B-23: Code Table for Low-Entropy Code 11

Input Codeword	Output Codeword	Input Codeword	Output Codeword
0^{16}	1'h0	000101	11'h13F
1	5'h01	0000101	11'h1BF
0^{7}1	6'h0D	00000101	11'h37F
0^{r}1, 8≤r≤10	<6'h(21+2r)>	0102	11'h6DF
0^{r}1, 11≤r≤15	<6'h(2C+r)>	00102	11'h7DF
002	6'h11	000102	11'h53F
0^{r}2, 3≤r≤6	<6'h(22+r)>	0000102	11'h5BF
0^{7}2	6'h2D	00000102	11'h77F
0^{r}2, 8≤r≤10	<6'h(22+2r)>	0^{r}10X, 1≤r≤2	<19'h(7FFF0+2r)>
0^{14}2	7'h4F	00010X	20'h0FFFF
0^{15}2	7'h1F	000010X	20'h2FFFF
0^{r}X, 0≤r≤6	<14'h(3FF4+r)>	0000010X	20'h9FFFF
0^{r}X, 7≤r≤15	<15'h(7FEF+r)>	200	6'h09
0^{6}10	6'h1D	201	11'h35F
011	11'h25F	0^{11}201	12'hEFF
0011	11'h0DF	202	11'h75F
00011	11'h1DF	0^{11}202	12'h1FF
000011	11'h23F	20X	20'h77FFF
0000011	11'h2BF	0^{11}20X	20'hBFFFF
0^{6}11	11'h17F	0001000	6'h15
012	11'h65F	01001	11'h03F
0012	11'h4DF	001001	11'h33F
00012	11'h5DF	0001001	11'h3BF
000012	11'h63F	01002	11'h43F
0000012	11'h6BF	001002	11'h73F
0^{6}12	11'h57F	0001002	11'h7BF
01X	19'h07FFF	0100X	19'h57FFF
0^{r}1X, 2≤r≤3	<19'h(7FFED+2r)>	00100X	20'h8FFFF
00001X	20'hF7FFF	000100X	20'hAFFFF
000001X	20'hCFFFF	0^{11}2000	7'h6F
0^{6}1X	20'h1FFFF	0^{11}2001	12'hDFF
020	6'h31	0^{11}2002	12'h3FF
0^{r}20, 12≤r≤13	<7'h(60+2r)>	0^{11}200X	20'hFFFFF
21	11'h05F	0010000	6'h35
021	11'h15F	010001	11'h0BF
0^{r}21, 11≤r≤12	<12'h(FDD+2r)>	0010001	11'h07F
0^{13}21	12'h9FF	010002	11'h4BF
22	11'h45F	0010002	11'h47F
022	11'h55F	01000X	20'h4FFFF
0^{r}22, 11≤r≤12	<12'h(FDE+2r)>	001000X	20'h6FFFF
0^{13}22	12'h5FF	0100001	11'h27F
0^{r}2X, 0≤r≤1	<20'h(FFFEC+r)>	0100002	11'h67F
0^{11}2X	20'hDFFFF	010000X	20'hEFFFF
0^{r}2X, 12≤r≤13	<20'h(FFFE4+2r)>	01000000	6'h03
0000100	6'h25	01000001	11'h0FF
00000100	6'h3D	01000002	12'h4FF
0101	11'h2DF	0100000X	20'h5FFFF
00101	11'h3DF		

Table B-24: Flush Table for Low-Entropy Code 11

Active Prefix	Flush Word	Active Prefix	Flush Word
0^{r}, 0≤r≤14	<4'h(0+r)>	000010	10'h0BF
0^{15}	5'h0F	0000010	10'h27F
01	9'h01F	20	10'h1DF
0^{r}1, 2≤r≤3	<9'h(1ED+2r)>	0^{11}20	10'h2FF
00001	10'h3DF	0100	9'h15F
000001	10'h33F	00100	10'h23F
0^{6}1	10'h07F	000100	10'h2BF
0^{r}2, 0≤r≤1	<10'h(3EC+r)>	0^{11}200	10'h3FF
0^{11}2	10'h37F	01000	10'h13F
0^{r}2, 12≤r≤13	<10'h(3E4+2r)>	001000	10'h1BF
0^{r}10, 1≤r≤2	<9'h(1F0+2r)>	010000	10'h3BF
00010	10'h03F	0100000	10'h17F

Table B-25: Code Table for Low-Entropy Code 12

Input Codeword	Output Codeword	Input Codeword	Output Codeword
0^{27}	1'h0	021	12'h0FF
1	6'h01	0^{r}21, 2≤r≤3	<12'h(FED+2r)>
0^{r}1, 1≤r≤8	<6'h(21+r)>	022	13'h09FF
0^{11}1	7'h3D	0^{r}22, 2≤r≤3	<13'h(1FEF+2r)>
0^{r}1, 12≤r≤26	<7'h(4A+2r)>	02X	20'h1FFFF
2	6'h21	0^{r}2X, 2≤r≤3	<20'h(FFFF5+2r)>
00002	7'h15	0^{9}100	7'h43
0^{r}2, 5≤r≤7	<7'h(53+r)>	0^{9}101	12'h1FF
0^{11}2	7'h7D	0^{9}102	13'h0BFF
0^{r}2, 12≤r≤25	<7'h(4B+2r)>	0^{9}10X	20'h7FFFF
0^{26}2	8'h7F	00200	7'h35
0^{r}X, 0≤r≤12	<15'h(7FEC+r)>	0^{r}201, 1≤r≤2	<12'h(FF0+2r)>
0^{r}X, 13≤r≤25	<16'h(FFE5+r)>	0^{r}202, 1≤r≤2	<13'h(1FF2+2r)>
0^{26}X	17'h0FFFF	020X	20'h5FFFF
0^{10}10	7'h03	0020X	21'h0FFFFF
0^{r}11, 9≤r≤10	<12'h(FED+r)>	02000	7'h75
0^{r}12, 9≤r≤10	<13'h(1FEF+r)>	02001	12'hAFF
0^{r}1X, 9≤r≤10	<20'h(FFFF3+r)>	02002	13'h1DFF
00020	7'h55	0200X	21'h1FFFFF

Table B-26: Flush Table for Low-Entropy Code 12

Active Prefix	Flush Word	Active Prefix	Flush Word
0^{r}, 0≤r≤4	<4'h(0+r)>	0002	10'h1FF
0^{r}, 5≤r≤25	<5'h(5+r)>	0^{9}10	9'h0FF
0^{26}	6'h1F	020	9'h0BF
0^{r}1, 9≤r≤10	<9'h(1F3+r)>	0020	10'h3FF
0^{r}2, 1≤r≤2	<9'h(1F7+r)>	0200	9'h1BF

Table B-27: Code Table for Low-Entropy Code 13

Input Codeword	Output Codeword	Input Codeword	Output Codeword
0^{46}	1'h0	0^{r}X, 0≤r≤17	<15'h(7FE0+r)>
0^{r}1, 1≤r≤2	<7'h(3F+2r)>	0^{r}X, 18≤r≤44	<16'h(FFD2+r)>
0^{r}1, 3≤r≤8	<7'h(40+2r)>	0^{45}X	17'h0FFFF
0^{r}1, 9≤r≤26	<7'h(49+r)>	11	12'h1FF
0^{r}1, 27≤r≤45	<8'h(A4+2r)>	12	12'h9FF
0^{r}2, 0≤r≤2	<7'h(40+2r)>	1X	18'h1FFFF
0^{r}2, 3≤r≤8	<7'h(41+2r)>	100	7'h51
0^{r}2, 9≤r≤25	<8'h(BF+r)>	101	12'h5FF
0^{r}2, 26≤r≤44	<8'h(A5+2r)>	102	12'hDFF
0^{45}2	9'h0FF	10X	18'h3FFFF

Table B-28: Flush Table for Low-Entropy Code 13

Active Prefix	Flush Word	Active Prefix	Flush Word
0^{r}, 0≤r≤17	<5'h(0+r)>	1	8'h7F
0^{r}, 18≤r≤44	<6'h(12+r)>	10	8'hFF
0^{45}	7'h3F		

Table B-29: Code Table for Low-Entropy Code 14

Input Codeword	Output Codeword	Input Codeword	Output Codeword
0^{85}	1'h0	0^{r}2, 21≤r≤64	<9'h(197+r)>
0^{r}1, 0≤r≤20	<8'h(80+2r)>	0^{r}2, 65≤r≤83	<9'h(157+2r)>
0^{r}1, 21≤r≤64	<8'h(95+r)>	0^{84}2	10'h1FF
0^{r}1, 65≤r≤84	<9'h(156+2r)>	0^{r}X, 0≤r≤42	<16'h(FFC0+r)>
0^{r}2, 0≤r≤20	<8'h(81+2r)>	0^{r}X, 43≤r≤84	<17'h(1FFAB+r)>

Table B-30: Flush Table for Low-Entropy Code 14

Active Prefix	Flush Word
0^{r}, 0≤r≤42	<6'h(0+r)>
0^{r}, 43≤r≤84	<7'h(2B+r)>

Table B-31: Code Table for Low-Entropy Code 15

Input Codeword	Output Codeword
0^{256}	1'h0
0^{r}X, 0≤r≤255	<9'h(100+r)>

Table B-32: Flush Table for Low-Entropy Code 15

Active Prefix	Flush Word
0^{r}, 0≤r≤255	<8'h(0+r)>

ANNEX C

SECURITY, SANA, AND PATENT CONSIDERATIONS

(INFORMATIVE)

C1 SECURITY CONSIDERATIONS

C1.1 SECURITY BACKGROUND

It is assumed that security is provided by encryption, authentication methods, and access control to be performed at the application and/or transport layers. Mission and service providers are expected to select from recommended security methods suitable to the specific application profile. Specification of these security methods and other security provisions is outside the scope of this Recommended Standard.

C1.2 SECURITY CONCERNS

Security concerns in the areas of data privacy, integrity, authentication, access control, availability of resources, and auditing are to be addressed in the appropriate layers and are not related to this Recommended Standard. The use of lossless data compression does not affect the proper functioning of methods used to achieve such protection.

The use of lossless data compression slightly improves data integrity because the alteration of even a single bit of compressed data is likely to cause conspicuous and easily detectible corruption of the reconstructed data, thus making it more likely that malicious data alteration will be detected.

C1.3 POTENTIAL THREATS AND ATTACK SCENARIOS

An eavesdropper will not be able to decompress compressed data if proper encryption is performed at a lower layer.

C1.4 CONSEQUENCES OF NOT APPLYING SECURITY

There are no specific security measures prescribed for compressed data. Therefore, consequences of not applying security are only imputable to the lack of proper security measures in other layers.

C2 SANA CONSIDERATIONS

The recommendations of this document do not require any action from SANA.

C3 PATENT CONSIDERATIONS

At time of publication, the specifications of this Recommended Standard are not known to be the subject of patent rights.

ANNEX D

REFERENCES

(INFORMATIVE)

[D1] *Lossless Multispectral and Hyperspectral Image Compression*. Issue 2. Report Concerning Space Data System Standards (Green Book), CCSDS 120.2-G-2. Forthcoming.

[D2] *Lossless Multispectral & Hyperspectral Image Compression*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 123.0-B-1. Washington, D.C.: CCSDS, May 2012.

[D3] *Space Packet Protocol*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 133.0-B-1. Washington, D.C.: CCSDS, September 2003.

[D4] *CCSDS File Delivery Protocol (CFDP)*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 727.0-B-4. Washington, D.C.: CCSDS, January 2007.

[D5] *AOS Space Data Link Protocol*. Issue 3. Recommendation for Space Data System Standards (Blue Book), CCSDS 732.0-B-3. Washington, D.C.: CCSDS, September 2015.

[D6] *TM Space Data Link Protocol*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 132.0-B-2. Washington, D.C.: CCSDS, September 2015.

[D7] *Unified Space Data Link Protocol*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 732.1-B-1. Washington, D.C.: CCSDS, October 2018.

ANNEX E

TABLES OF SYMBOLS USED

(INFORMATIVE)

This annex tabulates symbols used in this Recommended Standard.

Table E-1: Coordinate Indices and Image Quantities

Symbol	Meaning	Reference
x, y, z	image coordinate indices	3.2.1
t	alternate image coordinate index	3.4
$s_{z,y,x}$, $s_z(t)$	image data sample	3.2.1
N_X , N_Y , N_Z	image dimensions (user-specified)	3.2.2
F_y	frame	3.2.3
D	image dynamic range in bits (user-specified)	3.3.1
s_min , s_max	lower and upper sample value limits	3.3.2
s_mid	mid-range sample value	3.3.2
τ	number of supplementary information tables (user-specified)	3.5.2.1
i_z , $i_{z,x}$	integer supplementary information table data value (optional, user-specified)	3.5.2.3
D_I	integer supplementary information table bit depth (optional, user-specified)	3.5.2.3
b_z , $b_{z,x}$	float supplementary information table sign bit (optional, user-specified)	3.5.2.3
j_z , $j_{z,x}$	float supplementary information table significand (optional, user-specified)	3.5.2.3
α_z , $\alpha_{z,x}$	float supplementary information table exponent (optional, user-specified)	3.5.2.3
β	float supplementary information table exponent bias (optional, user-specified)	3.5.2.3.3
D_F	float supplementary information table significand bit depth (optional, user-specified)	3.5.2.3.3
D_E	float supplementary information table exponent bit depth (optional, user-specified)	3.5.2.3.3

Table E-2: Predictor Quantities

Symbol	Meaning	Reference
P	number of spectral bands used for prediction (user-specified)	4.2
P_z^*	number of previous spectral bands used for prediction in band z	4.2
C_z	number of local difference values used for prediction in band z	4.3.2
$\sigma_{z,y,x}$	local sum	4.4

Symbol	Meaning	Reference
$d_{z,y,x}$, $d_z(t)$	central local difference	4.5.1
$d^{\mathrm{N}}_{z,y,x}$, $d^{\mathrm{W}}_{z,y,x}$, $d^{\mathrm{NW}}_{z,y,x}$, $d^{\mathrm{N}}_{z}(t)$, $d^{\mathrm{W}}_{z}(t)$, $d^{\mathrm{NW}}_{z}(t)$	directional local differences	4.5.2
$\mathbf{U}_z(t)$	local difference vector	4.5.3
Ω	weight resolution (user-specified)	4.6.1
$\omega_{\min}$, $\omega_{\max}$	minimum and maximum weight values	4.6.1.3
$\mathbf{W}_z(t)$	weight vector	4.6.2
$\omega^{\mathrm{N}}_{z}(t)$, $\omega^{\mathrm{W}}_{z}(t)$, $\omega^{\mathrm{NW}}_{z}(t)$, $\omega^{(i)}_{z}(t)$	weight values	4.6.2
Λ_z	weight initialization vector (optional, user-specified)	4.6.3.3
Q	weight initialization resolution (optional, user-specified)	4.6.3.3
$\hat{d}_z(t)$	predicted central local difference	4.7.1
$\breve{s}_z(t)$	high-resolution predicted sample value	4.7.2
R	register size, in bits, used in prediction calculation (user-specified)	4.7.2
$\tilde{s}_z(t)$	double-resolution predicted sample value	4.7.3
$\hat{s}_z(t)$, $\hat{s}_{z,y,x}$	predicted sample value	4.7.4
$\Delta_z(t)$	prediction residual	4.8.1
$m_z(t)$	maximum error	4.8.2.1
$q_z(t)$	quantizer index	4.8.1
a_z	absolute error limit (optional, user-specified)	4.8.2
r_z	relative error limit (optional, user-specified)	4.8.2
D_{A}	absolute error limit bit depth (optional, user-specified)	4.8.2.2.1
D_{R}	relative error limit bit depth (optional, user-specified)	4.8.2.2.2
A^*	absolute error limit constant (optional, user-specified)	4.8.2.3.1
R^*	relative error limit constant (optional, user-specified)	4.8.2.3.1
u	error limit update period exponent (optional, user-specified)	4.8.2.4
Θ	sample representative resolution (user-specified)	4.9.1
ϕ_z	sample representative damping (user-specified)	4.9.1
ψ_z	sample representative offset (user-specified)	4.9.1
$s''_z(t)$	sample representative	4.9.2

Symbol	Meaning	Reference
$\tilde{s}''_z(t)$	double-resolution sample representative	4.9.2
$s'_z(t)$	clipped quantizer bin center	4.9.2
$e_z(t)$	double-resolution prediction error	4.10.1
$\rho(t)$	weight update scaling exponent	4.10.2
$v_{\min}$, $v_{\max}$	initial and final weight update scaling exponent parameters (user-specified)	4.10.2
t_{inc}	weight update scaling exponent change interval (user-specified)	4.10.2
$\varsigma_z^{(i)}$, ς_z^{*}	weight update scaling exponent offsets (user-specified)	4.10.3
$\delta_z(t)$, $\delta_{z,y,x}$	mapped quantizer index	4.11
$\theta_z(t)$	scaled difference between $\hat{s}_z(t)$ and nearest endpoint $s_{\min}$, $s_{\max}$	4.11

Table E-3: Encoder Quantities

Symbol	Meaning	Reference
B	output word size in bytes (user-specified)	5.2.2
M	sub-frame interleaving depth (optional, user-specified)	5.4.2.2
Sample-Adaptive Entropy Coder		
$\Re_k(j)$	length-limited GPO2 codeword	5.4.3.2.2.1
$U_{\max}$	unary length limit (user-specified)	5.4.3.2.2.2
$\Sigma_z(t)$	accumulator	5.4.3.2.3
$\Gamma(t)$	counter	5.4.3.2.3
γ_0	initial count exponent (user-specified)	5.4.3.2.3.2
k_z' , k_z''	accumulator initialization parameters (user-specified)	5.4.3.2.3.3
K	accumulator initialization constant (optional, user-specified)	5.4.3.2.3.3
γ^*	rescaling counter size (user-specified)	5.4.3.2.3.4
$k_z(t)$	variable length code parameter	5.4.3.2.4.3
Hybrid Entropy Coder		
$\Re'_k(j)$	reversed length-limited GPO2 codeword	5.4.3.3.3.2.1
$U_{\max}$	unary length limit (user-specified)	5.4.3.3.3.2
T_i	low-entropy code threshold value	5.4.3.3.3.3.1
L_i	low-entropy code input symbol limit	5.4.3.3.3.3.1
$\tilde{\Sigma}_z(t)$	high-resolution accumulator	5.4.3.3.4.1
$\Gamma(t)$	counter	5.4.3.3.4.1
γ_0	initial count exponent (user-specified)	5.4.3.3.4.2

γ^*	rescaling counter size (user-specified)	5.4.3.3.4.4
$k_z(t)$	high-entropy variable length code parameter	5.4.3.3.5.2
$\iota_z(t)$	low-entropy code input symbol	5.4.3.3.5.3
Block-Adaptive Entropy Coder		
n	resolution	5.4.3.4.2.3
J	block size (user-specified)	5.4.3.4.2.4
r	reference sample interval (user-specified)	5.4.3.4.2.5

ANNEX F

ABBREVIATIONS AND ACRONYMS

(INFORMATIVE)

AOS	Advanced Orbiting Systems
BI	band-interleavedBIL band-interleaved-by-line
BIP	band-interleaved-by-pixel
BSQ	band-sequentialCDS coded data set
CFDP	CCSDS File Delivery Protocol
GPO2	Golomb-power-of-2
ICS	implementation conformance statement
MSB	most significant bit
RL	requirements list
SANA	Space Assigned Numbers Authority