INTERNATIONAL STANDARD

ISO 18381

First edition
2013-06-01

Space data and information transfer systems — Lossless multispectral and hyperspectral image compression

Systèmes de transfert des informations et données spatiales — Compression d'images multispectrales et hyperspectrales sans perte

Reference number
ISO 18381:2013(E)

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2. www.iso.org/directives

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received. www.iso.org/patents

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

ISO 18381 was prepared by the Consultative Committee for Space Data Systems (CCSDS) (as CCSDS 123-0-B-1, May 2012) and was adopted (without modifications except those stated in Clause 2 of this International Standard) by Technical Committee ISO/TC 20, *Aircraft and space vehicles*, Subcommittee SC 13, *Space data and information transfer systems*.

Space data and information transfer systems — Lossless multispectral and hyperspectral image compression

1 Scope

This International Standard establishes a data compression algorithm applied to digital three-dimensional image data from payload instruments, such as multispectral and hyperspectral imagers, and specifies 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;

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

The characteristics of instrument data are specified only to the extent necessary to ensure multi-mission support capabilities. This International Standard does not attempt to quantify the relative bandwidth reduction, the merits of the approaches discussed, or the design requirements for encoders and associated decoders.

This International Standard addresses only lossless compression of three-dimensional data, where the requirement is for a data-rate reduction constrained to allow no distortion to be added in the data compression/decompression process.

The scope and field of application are furthermore detailed in subclause 1.3 of the enclosed CCSDS publication.

2 Requirements

Requirements are the technical recommendations made in the following publication (reproduced on the following pages), which is adopted as an International Standard:

CCSDS 123.0-B-1, May 2012, Lossless multispectral and hyperspectral image compression.

For the purposes of international standardization, the modifications outlined below shall apply to the specific clauses and paragraphs of publication CCSDS 123.0-B-1.

Pages i to vi

This part is information which is relevant to the CCSDS publication only.

Page 1-4

Add the following information to the reference indicated:

[1] Document CCSDS 121.0-B-2, May 2012, is equivalent to ISO 15887:2013.

Page C-1

Add the following information to the reference indicated:

[C2] Document CCSDS 133.0-B-1, September 2003, is equivalent to ISO 22646:2005.

[C3] Document CCSDS 727-0-B-4, January 2007, is equivalent to ISO 17355:2007.

[C4] Document CCSDS 732.0-B-2, July 2006, is equivalent to ISO 22666:2006.

3 Revision of publication CCSDS 123.0-B-1

It has been agreed with the Consultative Committee for Space Data Systems that Subcommittee ISO/TC 20/SC 13 will be consulted in the event of any revision or amendment of publication CCSDS 123.0-B-1. To this end, NASA will act as a liaison body between CCSDS and ISO.

The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

LOSSLESS MULTISPECTRAL & HYPERSPECTRAL IMAGE COMPRESSION

RECOMMENDED STANDARD

CCSDS 123.0-B-1

BLUE BOOK

May 2012

(Blank page)

AUTHORITY

Issue:	Recommended Standard, Issue 1
Date:	May 2012
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*, and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the address below.

This document is published and maintained by:

CCSDS Secretariat
Space Communications and Navigation Office, 7L70
Space Operations Mission Directorate
NASA Headquarters
Washington, DC 20546-0001, USA

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 three 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 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 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-3). 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 addressed to the CCSDS Secretariat at the 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 e.V. (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.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- CSIR Satellite Applications Centre (CSIR)/Republic of South Africa.
- Danish National Space Center (DNSC)/Denmark.
- Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
- 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.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- 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.
- Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.

– 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	Current issue
EC 1	Editorial Change 1	May 2012	Corrects typographical anomalies.

CONTENTS

CONTENTS (continued)

Section Page

Figure

Table

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;

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 [C1].

This Recommended Standard addresses only lossless compression of three-dimensional data, where the requirement is for a data-rate reduction constrained to allow no distortion to be added in the data compression/decompression process.

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 Image Data Compression algorithm described herein may be found in reference [C1].

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 lossless data 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 Protocol Implementation Conformance Statement (PICS) Requirements List (PRL) for implementations of this Recommended Standard.

g) Annex B discusses security, Space Assigned Numbers Authority (SANA), and patent considerations.

h) Annex C lists informative references.

i) Annex D provides tables of symbols used in this document.

j) Annex E 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 $M \bmod n$ is encoded instead of M.

For any integer x and positive integer R, the function $\text{mod}_R^*[x]$ is defined as

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

NOTE – The quantity $\text{mod}_R^*[x]$ 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 $\text{clip}\left(x, \{x_{\min}, x_{\max}\}\right)$ denotes the clipping of the real number x to the range $[x_{\min}, x_{\max}]$, that is,

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

For any real number x, the function $\text{sgn}^+(x)$ is defined as

$$\text{sgn}^+(x) = \begin{cases} 1, & x \ge 0 \\ -1, & x < 0. \end{cases} \tag{6}$$

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, i.e., 2^{N-1}.

In accordance with modern data communications practice, spacecraft data words are often grouped into 8-bit 'words' which 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*. Recommendation for Space Data System Standards, CCSDS 121.0-B-2. Blue Book. Issue 2. Washington, D.C.: CCSDS, May 2012.

2 OVERVIEW

2.1 GENERAL

This Recommended Standard defines a payload lossless 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 [C1].

The input to the compressor is an image, which 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 recovered exactly. Because of variations in image content, the length of compressed images will vary from image to image. That is, the compressed image is variable-length.

A user may choose to partition the output of an imaging instrument into smaller images that are separately compressed, e.g., 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 [C1] presents some examples.

The compressor consists of two functional parts, depicted in figure 2-1: a predictor and an encoder.

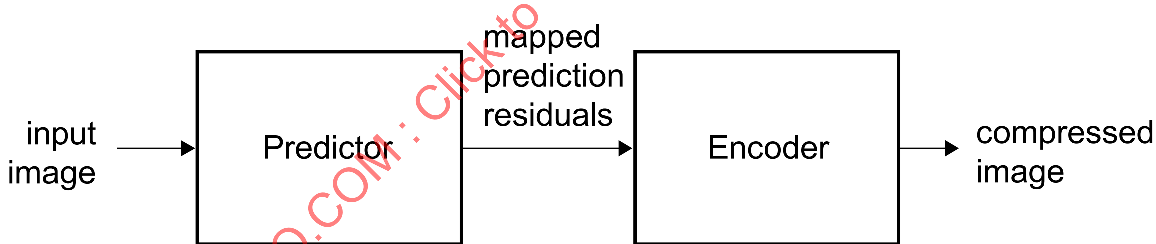

Figure 2-1: Compressor Schematic

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 prediction residual, i.e., the difference between the predicted and actual sample values, is then mapped to an unsigned integer that can be represented using the same number of bits as the input data sample. These mapped prediction residuals make up the predictor output.

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

2.2 DATA TRANSMISSION

The effects of a single bit error can propagate to corrupt reconstructed data to the end of a compressed image (see reference [C1]). Therefore measures should be taken to minimize the number of potential bit errors on the transmission link.

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 header of the next image in the event of a bit error.

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:

– Space Packet Protocol (reference [C2]);

– CCSDS File Delivery Protocol (CFDP) (reference [C3]);

– packet service or bitstream service as provided by the AOS Space Data Link Protocol (reference [C4]).

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 D-1 of annex D.

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 [C1] presents some examples.

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

3.3 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 $2 \le D \le 16$.

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

$$s_{\min} = 0,\ s_{\max} = 2^D - 1,\ s_{\mathrm{mid}} = 2^{D-1} \quad (7)$$

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

$$s_{\min} = -2^{D-1},\ s_{\max} = 2^{D-1} - 1,\ s_{\mathrm{mid}} = 0. \quad (8)$$

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,

$$s_z(t) \equiv s_{z,y,x} \tag{9}$$

$$\delta_z(t) \equiv \delta_{z,y,x} \tag{10}$$

etc., where

$$t = y \cdot N_X + x . \tag{11}$$

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

$$x = t \bmod N_X \tag{12}$$

$$y = \lfloor t / N_X \rfloor . \tag{13}$$

4 PREDICTOR

4.1 OVERVIEW

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

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 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 of samples used for prediction; this neighborhood is suitably truncated when $y = 0$, $x = 0$, $x = N_X - 1$, or $z < P$.

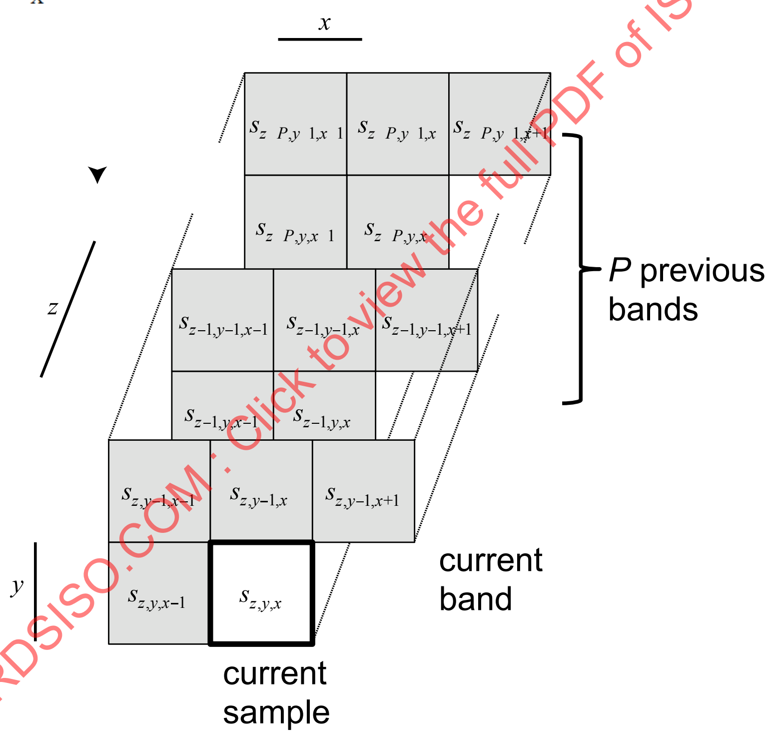

Figure 4-1: Typical Prediction Neighborhood

Within each spectral band, the predictor computes a *local sum* of neighboring sample 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 4.8) 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 mapped to an unsigned integer $\delta_{z,y,x}$, the mapped prediction residual (see 4.9).

The local sum $\sigma_{z,y,x}$ (see 4.4) is a weighted sum of samples in spectral band z that are adjacent to sample $s_{z,y,x}$. Figure 4-2 illustrates the samples used to calculate the local sum. A user may choose to perform prediction using *neighbor-oriented* or *column-oriented* local sums for an image. When neighbor-oriented local sums are used, the local sum is equal to the sum of four neighboring sample values in the spectral band (except when $y = 0$, $x = 0$, or $x = N_X-1$, in which case these four samples 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 value in the previous row (except when $y = 0$, in which case this sample is not available and the local sum calculation is suitably modified as detailed in 4.4).

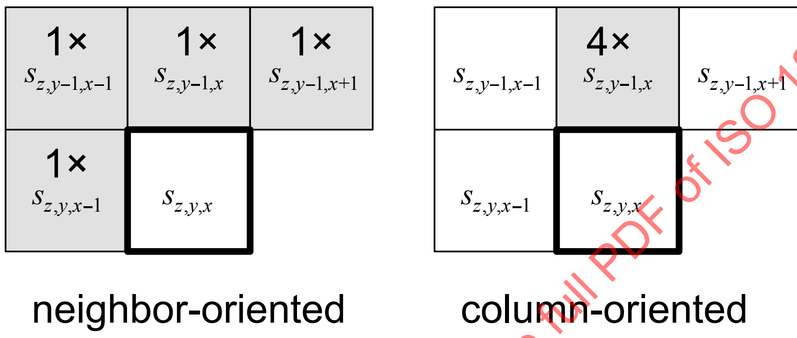

Figure 4-2: Samples Used to Calculate Local Sums

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 value $s_{z,y,x}$ (see 4.5.1). The three *directional local differences*, $d^{N}_{z,y,x}$, $d^{W}_{z,y,x}$, $d^{NW}_{z,y,x}$, 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, i.e., at image edges, as detailed in 4.5.2).

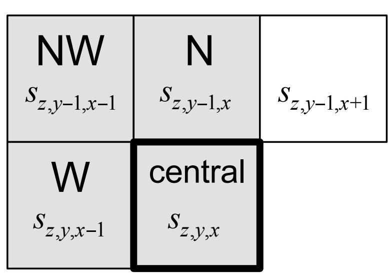

Figure 4-3: Computing Local Differences in a Spectral Band

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 [C1], 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 mapped to an unsigned integer $\delta_{z,y,x}$ (see 4.9). This mapping is invertible, so that the decompressor can exactly reconstruct the sample value $s_{z,y,x}$, and has the property that $\delta_{z,y,x}$ can be represented as a D-bit unsigned integer.

4.2 NUMBER OF BANDS FOR PREDICTION

The user-specified parameter P, which shall be an integer in the range $0 \le P \le 15$, 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

$$P_z^* = \min\{z, P\}. \tag{14}$$

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 1 (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

$$C_z = \begin{cases} P_z^*, & \text{reduced prediction mode} \\ P_z^* + 3, & \text{full prediction mode.} \end{cases} \tag{15}$$

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 values in band z that are neighbors of sample $s_{z,y,x}$. A user may choose to perform prediction using *column-oriented* or *neighbor-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.

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

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

$$\sigma_{z,y,x} = \begin{cases} s_{z,y,x-1} + s_{z,y-1,x-1} + s_{z,y-1,x} + s_{z,y-1,x+1}, & y > 0,\ 0 < x < N_X - 1 \\ 4s_{z,y,x-1}, & y = 0,\ x > 0 \\ 2\left(s_{z,y-1,x} + s_{z,y-1,x+1}\right), & y > 0,\ x = 0 \\ s_{z,y,x-1} + s_{z,y-1,x-1} + 2s_{z,y-1,x}, & y > 0,\ x = N_X - 1, \end{cases} \quad (16)$$

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

$$\sigma_{z,y,x} = \begin{cases} 4s_{z,y-1,x}, & y > 0 \\ 4s_{z,y,x-1}, & y = 0,\ x > 0. \end{cases} \quad (17)$$

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

$$d_{z,y,x} = 4\,s_{z,y,x} - \sigma_{z,y,x}. \quad (18)$$

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

$$d^{N}_{z,y,x} = \begin{cases} 4s_{z,y-1,x} - \sigma_{z,y,x}, & y > 0 \\ 0, & y = 0 \end{cases} \quad (19)$$

$$d_{z,y,x}^{\mathrm{W}} = \begin{cases} 4s_{z,y,x-1} - \sigma_{z,y,x}, & x > 0, y > 0 \\ 4s_{z,y-1,x} - \sigma_{z,y,x}, & x = 0, y > 0 \\ 0, & y = 0 \end{cases} \tag{20}$$

$$d_{z,y,x}^{\mathrm{NW}} = \begin{cases} 4s_{z,y-1,x-1} - \sigma_{z,y,x}, & x > 0, y > 0 \\ 4s_{z,y-1,x} - \sigma_{z,y,x}, & x = 0, y > 0 \\ 0, & y = 0. \end{cases} \tag{21}$$

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

$$\mathbf{U}_z(t) = \begin{bmatrix} d_z^{\mathrm{N}}(t) \\ d_z^{\mathrm{W}}(t) \\ d_z^{\mathrm{NW}}(t) \\ d_{z-1}(t) \\ d_{z-2}(t) \\ \vdots \\ d_{z-P_z^*}(t) \end{bmatrix}, \tag{22}$$

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

$$\mathbf{U}_z(t) = \begin{bmatrix} d_{z-1}(t) \\ d_{z-2}(t) \\ \vdots \\ d_{z-P_z^*}(t) \end{bmatrix}. \tag{23}$$

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 $4 \le \Omega \le 19$.

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

$$\omega_{\min} = -2^{\Omega+2}, \ \omega_{\max} = 2^{\Omega+2} - 1. \tag{24}$$

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 [C1] 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,

$$\mathbf{W}_z(t) = \begin{bmatrix} \omega_z^{\mathrm{N}}(t) \\ \omega_z^{\mathrm{W}}(t) \\ \omega_z^{\mathrm{NW}}(t) \\ \omega_z^{(1)}(t) \\ \omega_z^{(2)}(t) \\ \vdots \\ \omega_z^{(P_z^*)}(t) \end{bmatrix}, \tag{25}$$

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

$$\mathbf{W}_z(t) = \begin{bmatrix} \omega_z^{(1)}(t) \\ \omega_z^{(2)}(t) \\ \vdots \\ \omega_z^{(P_z^*)}(t) \end{bmatrix} \tag{26}$$

where the weight values are calculated as specified in 4.6.3 and 4.8.

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

$$\omega_z^{(1)}(1) = \frac{7}{8}2^{\Omega}, \quad \omega_z^{(i)}(1) = \left\lfloor \frac{1}{8}\omega_z^{(i-1)}(1) \right\rfloor, i = 2,3,\ldots,P_z^*. \tag{27}$$

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

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

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

$$\mathbf{W}_z(1) = 2^{\Omega+3-Q}\Lambda_z + \left\lceil 2^{\Omega+2-Q} - 1 \right\rceil \mathbf{1} \tag{28}$$

where **1** denotes a vector of all 'ones'.

NOTE – In the $(\Omega+3)$-bit two's complement representation of each component of $\mathbf{W}_z(1)$, the Q most significant bits 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 The scaled predicted sample value $\tilde{s}_z(t)$ is an integer defined as

$$\tilde{s}_z(t) = \begin{cases} \operatorname{clip}\left(\left\lfloor \dfrac{\operatorname{mod}_R^*\left[\hat{d}_z(t) + 2^{\Omega}\left(\sigma_z(t) - 4s_{\text{mid}}\right)\right]}{2^{\Omega+1}} \right\rfloor + 2s_{\text{mid}} + 1, \{2s_{\text{min}}, 2s_{\text{max}} + 1\}\right), & t > 0 \\ 2s_{z-1}(t), & t = 0, P > 0, z > 0 \\ 2s_{\text{mid}}, & t = 0 \text{ and } (P = 0 \text{ or } z = 0). \end{cases} \tag{29}$$

In this calculation:

a) 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 $\mathbf{U}_z(t)$:

$$\hat{d}_z(t) = \mathbf{W}_z^{\mathrm{T}}(t)\mathbf{U}_z(t) \tag{30}$$

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

b) 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 scaled predicted sample value. This Recommended Standard does not address the tradeoffs associated with selecting the value of R. Reference [C1] provides some discussion.

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

$$\hat{s}_z(t) = \left\lfloor \frac{\tilde{s}_z(t)}{2} \right\rfloor. \tag{31}$$

4.8 WEIGHT UPDATE

4.8.1 The scaled prediction error $e_z(t)$ is an integer defined as

$$e_z(t) = 2s_z(t) - \tilde{s}_z(t). \tag{32}$$

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

$$\rho(t) = \text{clip}\left(\nu_{\min} + \left\lfloor \frac{t - N_X}{t_{\text{inc}}} \right\rfloor, \{\nu_{\min}, \nu_{\max}\} \right) + D - \Omega, \tag{33}$$

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

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

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

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

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

$$\mathbf{W}_z(t+1) = \text{clip}\left(\mathbf{W}_z(t) + \left\lfloor \frac{1}{2}\left(\text{sgn}^+[e_z(t)] \cdot 2^{-\rho(t)} \cdot \mathbf{U}_z(t) + \mathbf{1} \right) \right\rfloor, \{\omega_{\min}, \omega_{\max}\} \right) \tag{34}$$

where the floor and clip operations are applied to each component of the vector.

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

4.9 MAPPED PREDICTION RESIDUAL

The mapped prediction residual $\delta_z(t)$ is an integer defined as

$$\delta_z(t) = \begin{cases} |\Delta_z(t)| + \theta_z(t), & |\Delta_z(t)| > \theta_z(t) \\ 2|\Delta_z(t)|, & 0 \le (-1)^{\tilde{s}_z(t)} \Delta_z(t) \le \theta_z(t) \\ 2|\Delta_z(t)| - 1, & \text{otherwise} \end{cases} \tag{35}$$

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

$$\Delta_z(t) = s_z(t) - \hat{s}_z(t) \tag{36}$$

and $\theta_z(t)$ is defined as

$$\theta_z(t) = \min\{\hat{s}_z(t) - s_{\min}, s_{\max} - \hat{s}_z(t)\}. \tag{37}$$

NOTE – Each mapped prediction residual $\delta_z(t)$ 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 D-3 of annex D.

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, consists of losslessly encoded mapped prediction residuals $\{\delta_{z,y,x}\}$ from the predictor. The mapped prediction residuals are sequentially encoded in the order selected by the user (see 5.4.2) and indicated in the header. This encoding order need not correspond to the order in which samples are output from the imaging instrument or processed by the predictor.

To encode the mapped prediction residuals for an image, a user may choose to use the *sample-adaptive* entropy coding approach specified in 5.4.3.2 or the *block-adaptive* approach specified in 5.4.3.3; this latter approach relies on the lossless data compressor defined in reference [1]. The sample-adaptive entropy coder typically yields smaller compressed images than the block-adaptive entropy coder. Further examples and comparisons can be found in reference [C1].

Under the sample-adaptive entropy coding approach, each mapped prediction residual is encoded using a variable-length binary codeword. The variable-length codes used are 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 prediction residuals are encoded.

Under the block-adaptive entropy coding approach, the sequence of mapped prediction residuals 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 prediction residuals 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 method 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.

header	body

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 $1 \leq B \leq 8$. Fill bits shall be included in the body (as specified in 5.4.3.2.3.5 and 5.4.3.3.3.2) 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

The header of a compressed image shall consist of the following parts in the following order, as depicted in figure 5-2:

a) Image Metadata—12 bytes (see 5.3.2);

b) Predictor Metadata—variable length (see 5.3.3);

c) Entropy Coder Metadata—variable length (see 5.3.4).

Figure 5-2: Overview of Header Structure

NOTES

1 The length of Predictor Metadata and Entropy Coder Metadata header parts can vary depending on prediction and encoding 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.

5.3.2 IMAGE METADATA

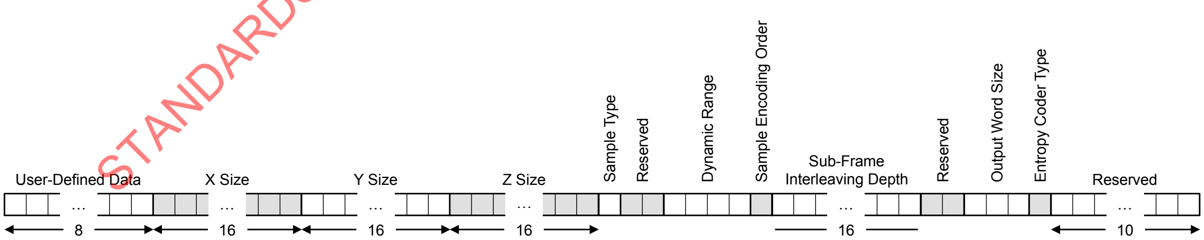

Figure 5-3: Overview of Image Metadata Structure

The Image Metadata header part, depicted in figure 5-3, shall consist of the fields specified in table 5-1, arranged in the order listed.

Table 5-1: Image Metadata

Field	Width (bits)	Description	Reference
User-Defined Data	8	The user may assign the value of this field arbitrarily, e.g., 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	2	This field shall have value '00'.	
Dynamic Range	4	The value D encoded 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 band-sequential 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 band-sequential encoding order is used, this field shall be all 'zeros'.	5.4.2.2
Reserved	2	This field shall have value '00'.	
Output Word Size	3	The value B encoded mod 2^3 as a 3-bit unsigned binary integer.	5.2.2
Entropy Coder Type	1	'0': sample-adaptive entropy coder is used. '1': block-adaptive entropy coder is used.	5.4.3
Reserved	10	This field shall contain all 'zeros'.	

5.3.3 PREDICTOR METADATA

5.3.3.1 General

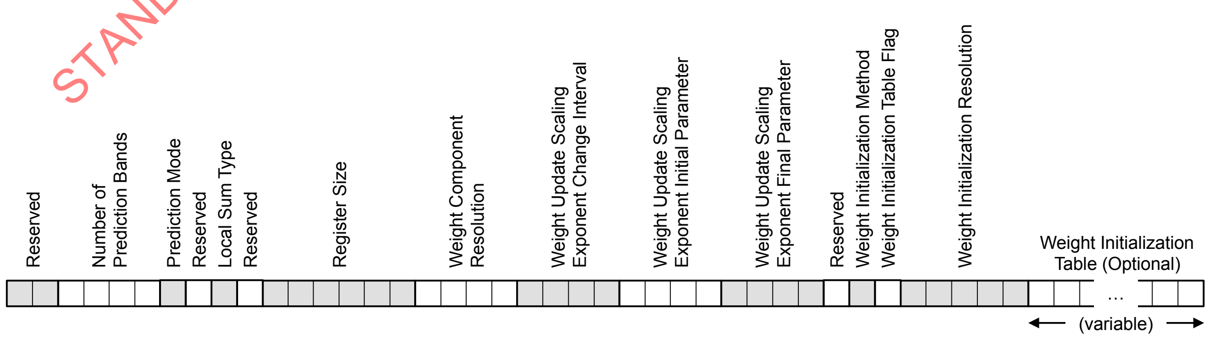

Figure 5-4: Overview of Predictor Metadata Structure

Predictor Metadata, depicted in figure 5-4, shall consist of the fields specified in table 5-2, arranged in the order listed.

Table 5-2: Predictor Metadata

Field	Width (bits)	Description	Reference
Reserved	2	This field shall have value '00'.	
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
Reserved	1	This field shall have value '0'.	
Local Sum Type	1	'0': neighbor-oriented local sums are used. '1': column-oriented local sums are used.	4.4
Reserved	1	This field shall have value '0'.	
Register Size	6	The value R encoded mod 2^6 as a 6-bit unsigned binary integer.	4.7.1
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 $(\log_2 t_{\mathrm{inc}} - 4)$ encoded as a 4-bit unsigned binary integer.	4.8.2
Weight Update Scaling Exponent Initial Parameter	4	The value $(\nu_{\min} + 6)$ encoded as a 4-bit unsigned binary integer.	4.8.2
Weight Update Scaling Exponent Final Parameter	4	The value $(\nu_{\max} + 6)$ encoded as a 4-bit unsigned binary integer.	4.8.2
Reserved	1	This field shall have value '0'.	
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 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
Weight Initialization Table (Optional)	(variable)	(See 5.3.3.2 below.)	4.6.3

5.3.3.2 Weight Initialization Table

5.3.3.2.1 The optional Weight Initialization Table may be included in the Predictor Metadata 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 from the Predictor Metadata. 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.

5.3.3.2.2 When the Weight Initialization Table is included in the Predictor Metadata, the custom weight initialization vectors $\{\Lambda_z\}_{z=0}^{N_Z-1}$ 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 Λ_z.

5.3.3.2.3 Fill bits shall be appended to the Weight Initialization Table as needed to reach the next byte boundary. Fill bits shall be all ‘zeros’.

5.3.4 ENTROPY CODER METADATA

5.3.4.1 General

Entropy Coder Metadata shall follow the structure defined in 5.3.4.2 if the sample-adaptive entropy coder is used, and the structure defined in 5.3.4.3 if the block-adaptive entropy coder is used.

5.3.4.2 Sample-Adaptive Entropy Coder

5.3.4.2.1 General

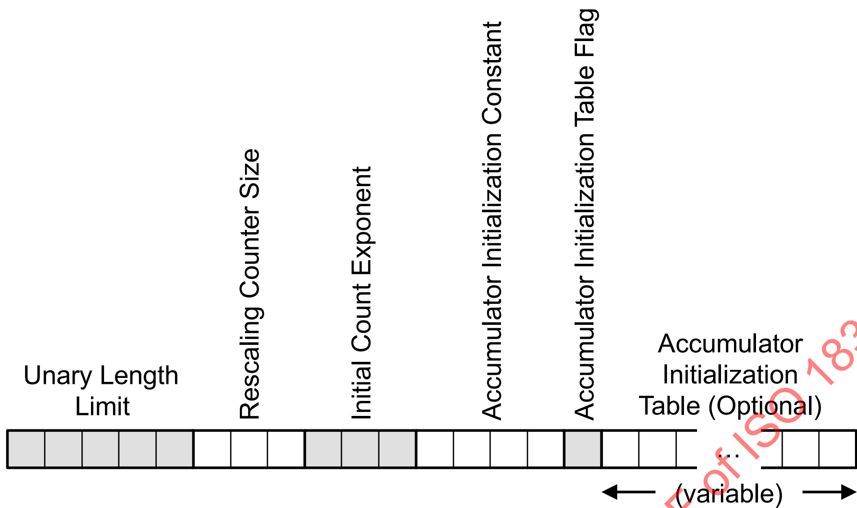

Figure 5-5: Overview of Entropy Coder Metadata Structure When Sample-Adaptive Entropy Coder Is Used

When the sample-adaptive entropy coder is used, the Entropy Coder Metadata, depicted in figure 5-5, shall consist of the fields specified in table 5-3, arranged in the order listed.

Table 5-3: Entropy Coder Metadata 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.	5.4.3.2.3
Rescaling Counter Size	3	The value $(\gamma^* - 4)$ encoded as a 3-bit unsigned binary integer.	5.4.3.2.2.4
Initial Count Exponent	3	The value γ_0 encoded mod 2^3 as a 3-bit unsigned binary integer.	5.4.3.2.2.2
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'.	5.4.3.2.2.3
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.	5.4.3.2.2.3
Accumulator Initialization Table (Optional)	(variable)	(See 5.3.4.2.2 below.)	5.4.3.2.2.3

5.3.4.2.2 Accumulator Initialization Table

5.3.4.2.2.1 The optional Accumulator Initialization Table may be included in the Entropy Coder Metadata 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 from the Entropy Coder Metadata. 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 5.4.3.2.2.3), 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. Fill bits shall be all 'zeros'.

5.3.4.3 Block-Adaptive Entropy Coder

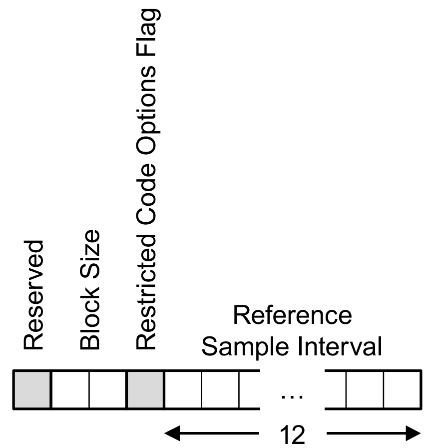

Figure 5-6: Overview of Entropy Coder Metadata Structure When Block-Adaptive Entropy Coder Is Used

When the block-adaptive entropy coder is used, the Entropy Coder Metadata, depicted in figure 5-6, shall consist of the fields specified in table 5-4, arranged in the order listed.

Table 5-4: Entropy Coder Metadata 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 $J = 64$.	5.4.3.3.2.4
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 section 5.1.2 of [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.	5.4.3.3.2.5

5.4 BODY

5.4.1 GENERAL

The compressed image body shall consist of the sequence of losslessly encoded mapped prediction residuals.

NOTE – Subsection 5.4.2 specifies the allowed orders in which mapped prediction residuals may be encoded. Subsection 5.4.3 specifies the two alternative methods of encoding the sequence of mapped prediction residuals.

5.4.2 ENCODING ORDER

5.4.2.1 General

Mapped prediction residuals shall be encoded in Band-Interleaved (BI) order, as defined in 5.4.2.2, or Band-Sequential (BSQ) order, as defined in 5.4.2.3.

NOTES

1 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.

2 The encoding order specifies the order in which the encoded samples are arranged in the compressed image. The encoding order does not necessarily correspond to the order in which samples are produced by an imaging instrument or processed by a predictor or entropy coder implementation.

5.4.2.2 Band-Interleaved Order

5.4.2.2.1 A *frame* F_y is defined as the set of all sample values with the same y coordinate value, that is,

$$F_y = \{s_{z,y,x} : 0 \le x \le N_X - 1,\ 0 \le z \le N_Z - 1\}. \tag{38}$$

Under the BI encoding order, each frame is partitioned along the z axis into one or more *sub-frames*, each including M consecutive spectral bands, except possibly the last sub-frame in each frame, which will have fewer than M spectral bands when N_Z is not a multiple of M. Specifically, for a given y and for $i = 0, 1, \ldots, \lceil N_Z / M \rceil - 1$, the i^{th} sub-frame $f_{y,i}$ is defined as

$$f_{y,i} = \{s_{z,y,x} : 0 \le x \le N_X - 1,\ iM \le z \le \min\{(i+1)M - 1,\ N_Z - 1\}\}. \tag{39}$$

5.4.2.2.2 The *sub-frame interleaving depth* M shall be an integer in the range $1 \le M \le N_Z$.

5.4.2.2.3 Under BI encoding order, samples in an image shall be encoded in the order defined by the nesting of sample index loops as follows:

for $y = 0$ to $N_Y - 1$
    for $i = 0$ to $\lceil N_Z / M \rceil - 1$
        for $x = 0$ to $N_X - 1$
            for $z = iM$ to $\min\{(i+1)M - 1,\ N_Z - 1\}$
                encode $\delta_{z,y,x}$.

NOTES

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

2 Within each sub-frame, samples are encoded in BIP order.

5.4.2.3 Band-Sequential Order

Under BSQ order, samples shall be encoded in the order 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 $N_X - 1$
      encode $\delta_{z,y,x}$.

5.4.3 ENTROPY CODING METHOD

5.4.3.1 General

Mapped prediction residuals shall be encoded using either the sample-adaptive entropy coding approach specified in 5.4.3.2 or the block-adaptive entropy coding approach specified in 5.4.3.3.

5.4.3.2 Sample-Adaptive Entropy Coder

5.4.3.2.1 Overview

Under the sample-adaptive entropy coding option, each mapped prediction residual $\delta_z(t)$ shall be encoded using a variable-length binary codeword. The selection of the code used to encode $\delta_z(t)$ is specified in 5.4.3.2.3. This selection is based on the values of the adaptive code selection statistics specified in 5.4.3.2.2.

5.4.3.2.2 Adaptive Code Selection Statistics

5.4.3.2.2.1 The adaptive code selection statistics 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 prediction residual value in the spectral band. This ratio determines the variable-length code used to encode $\delta_z(t)$.

5.4.3.2.2.2 The initial counter value $\Gamma(1)$ shall be equal to

$$\Gamma(1) = 2^{\gamma_0} \tag{40}$$

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

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

$$\Sigma_z(1) = \left\lfloor \frac{1}{2^7}\left(3 \cdot 2^{k'_z+6} - 49\right)\Gamma(1) \right\rfloor \tag{41}$$

where the user-selected value k'_z shall be an integer in the range $0 \le k'_z \le D-2$. An *accumulator initialization constant* K may be specified, with $0 \le K \le D-2$, in which case $k'_z = K$ for all z.

NOTE – This equation ensures that the initial value of encoding parameter $k_z(t)$ computed for spectral band z (see 5.4.3.2.3.3) will be equal to k'_z.

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

$$\Sigma_z(t) = \begin{cases} \Sigma_z(t-1) + \delta_z(t-1), & \Gamma(t-1) < 2^{\gamma^*} - 1 \\ \left\lfloor \dfrac{\Sigma_z(t-1) + \delta_z(t-1) + 1}{2} \right\rfloor, & \Gamma(t-1) = 2^{\gamma^*} - 1 \end{cases} \tag{42}$$

and the value of the counter is defined as

$$\Gamma(t) = \begin{cases} \Gamma(t-1) + 1, & \Gamma(t-1) < 2^{\gamma^*} - 1 \\ \left\lfloor \dfrac{\Gamma(t-1) + 1}{2} \right\rfloor, & \Gamma(t-1) = 2^{\gamma^*} - 1. \end{cases} \tag{43}$$

The interval at which the counter $\Gamma(t)$ 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 $\max\{4, \gamma_0 + 1\} \le \gamma^* \le 9$.

5.4.3.2.3 Encoding

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

NOTE – The sample-adaptive entropy coding procedure ensures that the codeword for $\delta_z(t)$ is not longer than $U_{\max} + D$ bits.

5.4.3.2.3.2 The first mapped prediction residual in each spectral band z shall be uncoded; i.e., the codeword for $\delta_z(0)$ is simply the D-bit unsigned binary integer representation of $\delta_z(0)$.

5.4.3.2.3.3 For $t > 0$, the codeword for the mapped prediction residual $\delta_z(t)$ depends on the values of $k_z(t)$ and $u_z(t)$, 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

$$\Gamma(t)2^{k_z(t)} \le \Sigma_z(t) + \left\lfloor \frac{49}{2^7}\Gamma(t) \right\rfloor \tag{44}$$

and $u_z(t)$ is computed as

$$u_z(t) = \left\lfloor \delta_z(t) / 2^{k_z(t)} \right\rfloor. \tag{45}$$

5.4.3.2.3.4 For $t > 0$, the codeword for $\delta_z(t)$ shall be determined as follows:

a) If $u_z(t) < U_{\max}$ then the codeword for $\delta_z(t)$ shall consist of $u_z(t)$ 'zeros', followed by a 'one', followed by the $k_z(t)$ least significant bits of the binary representation of $\delta_z(t)$.

b) Otherwise, the codeword for $\delta_z(t)$ shall consist of $U_{\max}$ 'zeros', followed by the D-bit binary representation of $\delta_z(t)$.

5.4.3.2.3.5 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 Block-Adaptive Entropy Coder

5.4.3.3.1 General

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

5.4.3.3.2 Parameters and Options

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

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

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

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

5.4.3.3.2.5 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].

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

5.4.3.3.2.7 The input to the block-adaptive entropy coder shall be the sequence of mapped prediction residuals $\{\delta_{z,y,x}\}$ arranged in the encoding order indicated in the Image Metadata header part and specified in 5.4.2. If the number of mapped prediction residuals, $N_X N_Y N_Z$, is not a multiple of J then 'zeros' shall be appended to the sequence as needed to reach the next multiple of J.

5.4.3.3.3 Body

5.4.3.3.3.1 The compressed image body shall consist of the concatenation of ℓ Coded Data Sets (CDSes), defined in subsection 5.1.4 of reference [1], where

$$\ell = \left\lceil \frac{N_X N_Y N_Z}{J} \right\rceil. \qquad (46)$$

5.4.3.3.3.2 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.

(Blank page)

ANNEX A

PROTOCOL IMPLEMENTATION CONFORMANCE STATEMENT PROFORMA

(NORMATIVE)

A1 INTRODUCTION

A1.1 OVERVIEW

This annex provides the Protocol Implementation Conformance Statement (PICS) Requirements List (PRL) for an implementation of a compressor or decompressor for *Lossless Multispectral & Hyperspectral Image Compression*, CCSDS 123.0-B-1, May 2012. The PICS for an implementation is generated by completing the PRL in accordance with the instructions below. An implementation shall satisfy the mandatory conformance requirements referenced in the PRL.

The PRL in this annex is blank. An implementation's completed PRL is called the PICS. The PICS states which capabilities and options have been implemented. The following can use the PICS:

– the implementer of a compressor or decompressor, as a checklist to reduce the risk of failure to conform to the standard through oversight;

– the supplier and acquirer or potential acquirer of a compressor or decompressor implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;

– the user or potential user of a compressor or decompressor implementation, as a basis for initially checking the possibility of interoperability between compressor and decompressor implementations;

– a compressor or decompressor implementation tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

A1.2 ABBREVIATIONS AND CONVENTIONS

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

<u>Item Column</u>

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

Description Column

The description column contains a brief description of the item. It implicitly means 'is <item description> supported by the implementation?'

Reference Column

The reference column indicates the relevant subsection of *Lossless Multispectral & Hyperspectral Image Compression*, CCSDS 123.0-B-1 (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*, the logic of their selection is defined immediately following the table.
- 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.

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 or values supported column, the support of every item as claimed by the implementer shall be stated by entering the appropriate answer ('Y', 'N', or 'N/A') or the values supported:

- Y yes, item supported by the implementation.
- N no, item not supported by the implementation.

range or list of values supported.

N/A not applicable.

References to Items

The support of an item in the PICS proforma can be referred to by indicating the table and item number separated by a solidus character ‘/’. For example, ‘A-2/8’ refers to the support for the 8th item in table A-2.

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 PRL

An implementer shows the extent of compliance to the Recommended Standard by completing the PRL; that is, the state of compliance with all mandatory requirements and the options supported are shown. The resulting completed PRL is called a PICS. The implementer shall complete the PRL 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 PICS PROFORMA FOR LOSSLESS MULTISPECTRAL & HYPERSPECTRAL IMAGE COMPRESSION

A2.1 GENERAL INFORMATION

A2.1.1 Identification of PICS

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

A2.1.2 Identification of Implementation Under Test (IUT)

Implementation name	
Implementation version	
Function implemented	Compression_____ Decompression_____
Special Configuration	
Other Information	