

Draft Recommendation for  
Space Data System Standards

|  |
| --- |
| LOSSLESS MULTISPEctral & hyperspectral IMAGE COMPRESSION |

Draft Recommended Standard

CCSDS 123.0-R-0

DRAFT Red Book

February 23, 2011

AUTHORITY

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  | Issue: | Red Book, Issue 0 |  |
|  | Date: | (tba) |  |
|  | Location: | Not Applicable |  |
|  | | | |

**(WHEN THIS RECOMMENDED STANDARD IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF AUTHORITY:)**

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 the *Procedures Manual 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

FOREWORD

This Recommendation specifies a method for lossless compression of multispectral and hyperspectral image data and a format for storing the compressed data.

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 the *Procedures Manual for the Consultative Committee for Space Data Systems*. 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.
* British National Space Centre (BNSC)/United Kingdom.
* 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.

Observer Agencies

* Austrian Space Agency (ASA)/Austria.
* Belgian Federal Science Policy Office (BFSPO)/Belgium.
* Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
* Centro Tecnico Aeroespacial (CTA)/Brazil.
* Chinese Academy of Sciences (CAS)/China.
* Chinese Academy of Space Technology (CAST)/China.
* Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
* Danish National Space Center (DNSC)/Denmark.
* European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
* European Telecommunications Satellite Organization (EUTELSAT)/Europe.
* 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.
* MIKOMTEK: CSIR (CSIR)/Republic of South Africa.
* Ministry of Communications (MOC)/Israel.
* National Institute of Information and Communications Technology (NICT)/Japan.
* National Oceanic and Atmospheric Administration (NOAA)/USA.
* National Space Organization (NSPO)/Chinese Taipei.
* Naval Center for Space Technology (NCST)/USA.
* Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
* Swedish Space Corporation (SSC)/Sweden.
* United States Geological Survey (USGS)/USA.

PREFACE

This document is a draft CCSDS Recommended Standard. Its ‘Red Book’ status indicates that the CCSDS believes the document to be technically mature and has released it for formal review by appropriate technical organizations. As such, its technical contents are not stable, and several iterations of it may occur in response to comments received during the review process.

Implementers are cautioned **not** to fabricate any final equipment in accordance with this document’s technical content.

DOCUMENT CONTROL

|  |  |  |  |
| --- | --- | --- | --- |
| **Document** | **Title and Issue** | **Date** | **Status** |
| CCSDS 123.0-R-0 | Lossless Multispectral & Hyperspectral Image Compression, Draft Recommended Standard, Issue 0 | February 23, 2011 | Current draft |
|  |  |  |  |
|  |  |  |  |

CONTENTS

Section Page

# Introduction

## 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 how this compressed data shall be formatted to enable decompression.

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

1. reduction of transmission channel bandwidth;
2. reduction of the buffering and storage requirement;
3. reduction of data-transmission time at a given rate.

## 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 [A1].

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. See reference [A1] for an outline of an implementation.

## 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.

## Rationale

The concept and rationale for the Image Data Compression algorithm described herein may be found in reference [A1].

## Document Structure

This document is organized as follows:

1. 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. It also identifies terminology that is used in this document but is defined elsewhere.
2. Section 2 provides an overview of the lossless data compressor.
3. Section 3 defines parameters and notation pertaining to an input image to be compressed.
4. Section 4 specifies the predictor stage of the compressor.
5. Section 5 specifies the entropy coding stage of the compressor and the format of a compressed image.
6. ANNEX A lists informative references.
7. ANNEX B provides tables of symbols used in this document.
8. ANNEX C discusses security.

## Definitions

### Mathematical notation and Definitions

In this document, for any real number *x*, the largest integer *n* such that *n*<*x* is denoted by

,



and correspondingly, the smallest integer *n* such that *n*>*x* by

.



The modulus *m* of a number *M* with respect to a divisor *n* is denoted by

*m* = *M* mod *n*.

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

*m* = *M* mod *n*

is encoded instead of *M*.

For any integer *x* and positive integer *R*, the function is defined as

.

The notation denotes the clipping of the real number to the range , that is,



.



For any real number , the function  is defined as



### Nomenclature

The following conventions apply throughout this Recommended Standard:

1. the words ‘shall’ and ‘must’ imply a binding and verifiable specification;
2. the word ‘should’ implies an optional, but desirable, specification;
3. the word ‘may’ implies an optional specification;
4. the words ‘is’, ‘are’, and ‘will’ imply statements of fact.

In the normative sections of this document (sections 3, 4, and 5), informative (i.e., non- normative) text is differentiated from normative text through the following convention:

1. notes introduce brief informative statements;
2. the following headings introduce one or more paragraphs of informative text:
   * Overview
   * Rationale

### Convention

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) shall correspond to the highest power of two, i.e. 2N-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 recommendation, the following nomenclature is used to describe this grouping:



8-Bit Word = ‘Byte’

## Reference

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

[1] *Lossless Data Compression*. Recommendation for Space Data System Standards, CCSDS 121.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, May 1997.

# Overview

## 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, which is partially addressed in [A1].

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. This Recommended Standard does not address the tradeoffs associated with selecting the size of images produced under such partitioning. Reference [A1] presents some examples.

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

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 comprise the predictor output.

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

## Data transmission

The effects of a single bit error can propagate to corrupt reconstructed data to the end of a compressed image (see reference [A1]). 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 [A2];
* CCSDS File Delivery Protocol (CFDP) [A3];
* packet service or bitstream service as provided by the AOS Space Data Link Protocol [A4].

# Image

## Overview

This section defines parameters and notation pertaining to an image. Quantities defined in this section are summarized in Table B-1 of ANNEX B.

## Dimensions

An *image* is a three-dimensional array of signed or unsigned integer sample values , 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. Re-arranging the order of spectral bands can affect compression performance. This Recommended Standard does not address the tradeoffs associated with such a band reordering. Reference [A1] presents some examples.

Indices *x*, *y*, and *z* take on integer values in the ranges , , , where each image dimension , , and shall have a value of at least 1 and at most .



## Dynamic Range

Data samples shall have a fixed-size dynamic range of *D* bits, where *D* shall be an integer in the range bits.

The lower and upper sample value limits are denoted  and  respectively. When samples are unsigned integers, the values of  and  are defined as

and when samples are signed integers, the values of  and  are defined as

.

## 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., , , etc.), or by the pair of indices *t*, *z*, (e.g., , , etc.). That is,

etc., where

.



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

.

# Predictor

## Overview

This section specifies the calculation of the set of *predicted sample values* and *mapped prediction residuals* from the input image samples . Quantities defined in this section are summarized in Table B-2 of ANNEX B.



Prediction can be performed causally in a single pass through the image. Prediction at sample , that is, the calculation of and , 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 , , , or .





Figure 4-1: Typical Prediction Neighborhood

Within each spectral band, the predictor computes a *local sum* of neighboring sample values (see 4.4). The local sum is used to compute one or more *local differences* within the spectral band (see 4.5). The predicted sample value is calculated using a weighted sum of these 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. The prediction residual, that is the difference between the sample value  and the predicted sample value , is mapped to an unsigned integer  (see 4.9) to produce the predictor output.

The local sum (see 4.4) is a weighted sum of samples in spectral band  that are adjacent to sample . 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 , , or , 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 , 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*, , is equal to the difference between the local sum and four times the current sample value  (see 4.5.1). The three *directional local differences*, , , , are each equal to the difference between 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 [A1], 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 pushbroom 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  and the predicted sample value , is mapped to an unsigned integer  (see 4.9). This mapping is invertible, so that the decompressor can exactly reconstruct the sample value , and has the property that  can be represented as a *D*-bit unsigned integer.

## Number of Bands for Prediction

The user specified parameter *P*, which shall be an integer in the range , determines the number of spectral bands used for prediction. Specifically, prediction in spectral band *z* depends on central local differences, defined in 4.5.1, computed in bands , , …, , where



.



## Full and Reduced Prediction Modes

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

Under reduced mode, prediction in spectral band *z* makes use of central local differences from the preceding  spectral bands. Under full prediction mode, prediction in spectral band *z* also 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 , is



## Local sum

The *local sum* is an integer equal to a weighted sum of previous sample values in band *z* that are neighbors of sample . 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., ), in which case column-oriented local sums shall be used.



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

When neighbor-oriented local sums are used, is defined as





and when column-oriented local sums are used, is defined as



.

NOTE – The value of  is not defined, as it is not needed.

## Local Differences

### Central Local Difference

When and are not both zero (i.e., when ), the central local difference is defined as



.



### Directional Local Differences

When and are not both zero (i.e., when ), the three directional local differences are defined as









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

### Local Difference Vector

For , the local difference vector is a vector of the local difference values used to calculate the predicted sample value . Under full prediction mode, is defined as



and under reduced prediction mode, for , is defined as



.



NOTE – Under reduced mode, is not defined as it is not needed.

## Weights

### Weight Values and Weight Resolution

In the prediction calculation (see 4.7), for each component of the local difference vector is multiplied by a corresponding integer *weight value*.



The resolution of the weight values is controlled by the user-specified parameter , which shall be an integer in the range . Each weight value is a signed integer quantity that can be represented using  bits. Thus, each weight value has minimum and maximum possible values and respectively, where

, .

NOTE – Increasing the number of bits used to represent weight values (i.e., using a larger value of ) provides increased resolution in the prediction calculation. This Recommended Standard does not address the tradeoffs associated with selecting the value of . Reference [A1] presents some examples.

### Weight Vector

The weight vector is a vector of the weight values used in prediction. Under full prediction mode,



and under reduced prediction mode, for



.



NOTE – Under reduced mode, for , is not defined as it is not needed.



### Initialization

A user may choose to use either *default* or *custom* weight initialization methods, defined below, to select the initial weight vector  for each spectral band *z*. The same weight initialization method shall be used for all spectral bands.

**Default Weight Initialization**. When default weight initialization is used, for each spectral band *z*, initial weight vector components , , … , shall be assigned values



.



With this option, under full prediction mode the remaining components of shall be assigned values



.



**Custom Weight Initialization**. When custom weight initialization is used, for each spectral band *z*, the initial weight vector shall be assigned using a user-specified *weight initialization vector* , consisting of signed *Q*-bit integer components. The weight initialization resolution *Q* shall be a user-specified integer in the range bits. The initial weight vector shall be calculated from by



where denotes a vector of all ones. The weight initialization vector may be encoded in the header as described in 5.3.



NOTES

1. Under this option, the first *Q* bits of each component of are equal to the binary representation of the corresponding component of .



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



## Prediction Calculation

The scaled predicted sample value is an integer defined as



In this calculation:

1. For the predicted central local difference is equal to the inner product of vectors and :



except for under reduced mode, in which case .



1. The user-selected register size parameter *R* shall be an integer in the range .

NOTE – Increasing the register size 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 . Reference [A1] provides some discussion.

The predicted sample value is defined as



.



## Weight Update

The scaled prediction error is an integer defined as

.

The weight update scaling exponent is an integer defined as

where user-specified integer parameters , , and  are constrained as follows. The values of  and  shall be integers in the range . The weight update factor change interval shall be an integer power of 2 in the range .

NOTE – These parameters control the rate at which weights adapt to image data statistics. Initially , and at regular intervals determined by the value of , is incremented by one until reaching a final value . Smaller values of produce larger weight increments, yielding faster adaptation to source statistics but worse steady-state compression performance.

For , following the calculation of , the next weight vector in the spectral band, , is defined as



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

NOTE – The quantity  is equivalent to  but is not in general equivalent to .

## Mapped Prediction Residual

The mapped prediction residual shall be equal to



where the prediction residual is the difference between the predicted and actual sample values,



and is defined as



.

NOTE – Each mapped prediction residual can be represented as a -bit unsigned integer.

# Entropy Coder

## Overview

This section specifies the entropy coding stage of the compressor and the format of a compressed image. Quantities defined in this section are summarized in Table B-3 of ANNEX B.

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 prediction residuals  from the predictor. The mapped prediction residuals are sequentially encoded in the order selected by the user (see 5.4.1) and indicated in the header. The order in which the mapped prediction residuals are encoded in the compressed image 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* encoding approach specified in 5.4.2.1 or the *block-adaptive* approach specified in 5.4.2.2; this latter approach relies on the lossless data compressor defined in reference [1]. The sample-adaptive encoder typically yields smaller compressed images than the block-adaptive encoder. Further examples and comparisons can be found in reference [A1].

Under the sample-adaptive encoding 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 encoding 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.

## General

A compressed image shall consist of variable-length *header*, defined in 5.3, followed by a variable-length *body*, defined in 5.4. Figure 5-1 depicts the structure of a compressed image.



**Figure 5-1: Compressed Image Structure**

The user-selected *output word size*, measured in bytes, shall be an integer *B* in the range . Fill bits shall be included in the body (as specified in 5.4.2.1.3.5 and 5.4.2.2.3.2) when needed to ensure that the size of the compressed image is a multiple of the output word size.

## Header

### General

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

1. Image Metadata (12 bytes, mandatory – see 5.3.2)
2. Predictor Metadata (variable length, optional – see 5.3.3)
3. Entropy Coder Metadata (variable length, optional – see 5.3.4)



**Figure 5-2: Overview of Header Structure when All Header Parts are Included**

The length of Predictor Metadata and Entropy Coder Metadata header parts can vary depending on prediction and encoding options selected by the user. Each header part is an integer number of bytes.

NOTE – The header length is not necessarily a multiple of the output word size.

Optional header parts may be omitted when the parameters described in a part can be determined without that part, e.g., when those parameters are set to known fixed values for an entire mission.

By CCSDS convention, all reserved bits in each header part shall be set to ‘zero’.

### 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 encoded mod as a 16-bit unsigned binary integer | 3.2 |
| Y Size | 16 | The value encoded mod as a 16-bit unsigned binary integer | 3.2 |
| Z Size | 16 | The value encoded mod 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 encoded mod as a 4-bit unsigned binary integer | 3.3 |
| Sample Encoding Order | 1 | ‘0’: samples are encoded in BI order  ‘1’: samples are encoded in BSQ order | 5.4.1 |
| Sub-Frame Interleaving Depth | 16 | When BI encoding order is used, this field shall contain the value encoded mod as a 16-bit unsigned binary integer.  When BSQ encoding order is used, this field shall be all zeros. | 5.4.1.1 |
| Reserved | 2 | This field shall have value ‘00’ |  |
| Output Word Size | 3 | The value encoded mod as a 3-bit unsigned binary integer. | 5.2.2 |
| Entropy Coder Type | 1 | ‘0’: sample-adaptive encoder is used  ‘1’: block-adaptive encoder is used | 5.4.2 |
| Predictor Metadata Flag | 1 | ‘0’: Predictor Metadata is not included in the header  ‘1’: Predictor Metadata is included in the header | 5.3.3 |
| Entropy Coder Metadata Flag | 1 | ‘0’: Entropy Coder Metadata is not included in the header  ‘1’: Entropy Coder Metadata is included in the header | 5.3.4 |
| Reserved | 8 | This field shall contain all zeros |  |

### Predictor Metadata

#### General



**Figure 5-4: Overview of Predictor Metadata Structure**

When the Predictor Metadata Flag field in the Image Metadata is set to ‘1’, Predictor Metadata, depicted in figure 5-4, shall be included in the header and 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 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 Mode | 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 encoded mod as a 6-bit unsigned binary integer | 0 |
| Weight Component Resolution | 4 | The value encoded as a 4-bit unsigned binary integer | 4.6.1 |
| Weight Update Scaling Exponent Change Interval | 4 | The value encoded as a 4-bit unsigned binary integer | 4.8.2 |
| Weight Update Scaling Exponent Initial Parameter | 4 | The value encoded as a 4-bit unsigned binary integer | 4.8.2 |
| Weight Update Scaling Exponent Final Parameter | 4 | The value 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 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 |

#### Weight Initialization Table

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.

When the Weight Initialization Table is included in the Predictor Metadata, the custom weight initialization vectors 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  to 

for to

encode *j*th component of .

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

### Entropy Coder Metadata

When the Entropy Coder Metadata Flag field of the Image Metadata header part is set to ‘1’, Entropy Coder Metadata shall be included in the header. Entropy Coder Metadata shall follow the structure defined in 5.3.4.1 if the sample-adaptive encoder is used, and the structure defined in 5.3.4.2 if the block-adaptive encoder is used.

#### Sample-Adaptive Encoder

##### General

Wolverine:Users:kiely:Desktop:CCSDS:CCSDS-Hyperspectral:2011-May:draft_standard:figs-pdf:fig5-5.pdf

**Figure 5-5: Overview of Entropy Coder Metadata Structure when Sample-Adaptive Encoder is Used**

When the sample-adaptive encoder 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 Encoder is Used

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Width (bits)** | **Description** | **Reference** |
| Unary Length Limit | 5 | The value encoded mod as a 5-bit unsigned binary integer | 5.4.2.1.3 |
| Rescaling Counter Size | 3 | The value encoded mod as a 3-bit unsigned binary integer | 5.4.2.1.2.4 |
| Initial Count Exponent | 3 | The value encoded mod as a 3-bit unsigned binary integer | 5.4.2.1.2.2 |
| 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 | 5.4.2.1.2.3 |
| Accumulator Initialization Constant | 4 | When the accumulator initialization constant is defined, this field encodes the value of as a 4-bit unsigned binary integer. Otherwise, this field shall be all zeros. | 5.4.2.1.2.3 |
| Accumulator Initialization Table (Optional) | (variable) | See 5.3.4.1.2 below | 5.4.2.1.2.3 |

##### Accumulator Initialization Table

The optional Accumulator Initialization Table shall be included in the Entropy Coder Metadata when the sample-adaptive encoder is used and the Accumulator Initialization Table Flag field is set 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 accumulator initialization values to be used throughout a mission, and elect to not encode these values with each image.

The Accumulator Initialization Table shall consist of the concatenated sequence of values, , (defined in 5.4.2.1.2.3) each encoded as a 4-bit binary unsigned integer.

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

#### Block-Adaptive Encoder

Wolverine:Users:kiely:Desktop:CCSDS:CCSDS-Hyperspectral:2011-May:draft_standard:figs-pdf:fig5-6.pdf

**Figure 5-6: Overview of Entropy Coder Metadata Structure When Block-Adaptive Encoder is Used**

When the block-adaptive encoder 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 Encoder is Used

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Width (bits)** | **Description** | **Reference** |
| Reserved | 1 | This field shall have value ‘0’ |  |
| Block Size[[1]](#footnote-1) | 2 | ‘00’: Block size  ‘01’: Block size | 5.4.2.2.2.4 |
| Reserved | 1 | This field shall have value ‘0’ |  |
| Reference Sample Interval | 12 | Value of encoded mod as a 12-bit unsigned binary integer | 5.4.2.2.2.5 |

## Body

The compressed image body shall consist of the sequence of losslessly encoded mapped prediction residuals. Subsection 5.4.1 specifies the order in which mapped prediction residuals may be encoded. Subsection 5.4.2 specifies the two alternative methods of encoding the sequence of mapped prediction residuals.

### Encoding Order

Mapped prediction residuals shall be encoded in band-interleaved (BI) order, as defined in 5.4.1.1, or band sequential (BSQ) order, as defined in 5.4.1.2.

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.

#### Band-Interleaved (BI) Order

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

.

Under the Band Interleaving 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 is not a multiple of *M*. Specifically, for a given  and for , the *i*th sub-frame  is defined as

.

The *sub-frame interleaving depth* *M* shall be an integer in the range .

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  to 

for  to 

for  to 

for  to 

encode 

NOTES

1. Under BI encoding order, when , the encoding order corresponds to band-interleaved-by-line (BIL), and when  the encoding order corresponds to band-interleaved-by-pixel (BIP).
2. Within each sub-frame, samples are encoded in BIP order.

#### Band-Sequential (BSQ) Order

Under BSQ order, samples shall be encoded in the order defined by the nesting of sample index loops as follows:

for  to 

for  to 

for  to 

encode 

### Encoding Method

Mapped prediction residuals shall be encoded using either the sample-adaptive encoding approach specified in 5.4.2.1 or the block-adaptive encoding approach specified in 5.4.2.2.

#### Sample-Adaptive Encoder

##### General

Under the sample-adaptive encoding option, each mapped prediction residual is encoded using a variable length binary codeword. The selection of the code used to encode  is specified in 5.4.2.1.3. This selection is based on the values of the adaptive code selection statistics specified in 5.4.2.1.2.

##### Adaptive Code Selection Statistics

The adaptive code selection statistics consist of an *accumulator* and a *counter* that are adaptively updated during the encoding process.



NOTE – The ratio  provides an estimate of the mean mapped prediction residual value in the spectral band. This ratio effectively determines the variable length code used to encode .

For each spectral band *z*, the initial counter value  shall be equal to

where the user-supplied value of the initial count exponent shall be an integer in the range .

For each spectral band *z*, the initial accumulator value  shall be equal to



where the user-selected value of shall be an integer in the range . If for all *z*, then *K* is referred to as the *accumulator initialization constant*.

NOTE – This equation ensures that initial value of encoding parameter  computed for spectral band *z* (see 5.4.2.1.3.3) will be equal to .

For ,  and . For , the value of the accumulator for spectral band *z* is defined as



and the value of the counter is defined as

The interval at which the counter and the accumulator are rescaled is controlled by the user-defined rescaling counter size parameter , which shall be an integer in the range .

##### Encoding

The user-supplied unary length limit  shall be an integer in the range .

NOTE – The sample-adaptive encoding procedure ensures that the codeword for  is not longer than  bits.

The first mapped prediction residual in the image shall be uncoded, i.e., the codeword for  is simply the *D*-bit unsigned binary integer representation of .

When  and  are not both zero, the codeword for the mapped prediction residual depends on the values of and , where is the largest nonnegative integer  such that



and is computed as



.



The codeword for shall be determined as follows:



1. If then the codeword for shall consist of zeros, followed by a one, followed by the least significant bits of the binary representation of .



1. Otherwise, the codeword for shall consist of zeros, followed by the *D*-bit binary representation of .



Following the last codeword in the compressed image, fill bits shall be appended as needed to reach the next output word boundary. Fill bits shall be all zeros.

#### Block-Adaptive Encoder

##### General

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

##### Parameters and Options

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

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

The input to the block-adaptive encoder shall be the sequence of mapped prediction residuals arranged in the encoding order indicated in the Image Metadata header part and specified in 5.4.1.

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

The *block size* parameter, *J*, defined in subsection 3.1 of reference [1], shall be equal to 8 or 16.[[2]](#footnote-2)

The *reference sample interval* parameter, *r*, defined in subsection 4.3 of reference [1], shall be a positive integer not larger than 256.[[3]](#footnote-3)

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

##### Body

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

.

Fill bits shall be appended after the last CDS as needed to reach the next output word boundary. Fill bits shall be all zeros. Fill bits shall not be inserted between CDSs.

1. References  
     
   (Informative)

[A1] *Lossless Multispectral & Hyperspectral Image Compression*, Report concerning Space Data Systems Standards, CCSDS ###.#-G-1, Green Book. [At time of publication, this document was under development.]

[A2] *Space Packet Protocol*. Recommendation for Space Data Systems Standards. CCSDS 133.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, September 2003.

[A3] *CCSDS File Delivery Protocol (CFDP)*. Recommendation for Space Data System Standards, CCSDS 727.0-B-3. Blue Book. Issue 3. Washington, D.C.: CCSDS, June 2005.

[A4] *AOS Space Data Link Protocol*. Recommendation for Space Data System Standards, CCSDS 732.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, September 2003.

1. Tables of Symbols Used  
     
   (Informative)

This annex tabulates symbols used in this Recommended Standard.

Table B-1: Coordinate Indices & Image Quantities

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Meaning** | **Reference** |
| , , | image coordinate indices | 3.2.1 |
|  | alternate image coordinate index | 3.4 |
| , | image data sample | 3.2.1 |
|  | image dynamic range in bits (user-specified) | 3.3.1 |
| , | sample value limits | 3.3.2 |
| , , | image dimensions (user-specified) | 3.2.2 |

Table B-2: Predictor Quantities

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Meaning** | **Reference** |
|  | number of spectral bands used for prediction (user-specified) | 4.2 |
|  | number of previous spectral bands used for prediction in band | 4.2 |
|  | number of local difference values used for prediction in band | 4.3.2 |
|  | local sum | 4.4 |
|  | central local difference | 4.5.1 |
| , ,  , , | directional local differences | 4.5.2 |
|  | local difference vector | 4.5.3 |
|  | weight resolution (user-specified) | 4.6.1 |
| , | minimum and maximum weight values | 4.6.1 |
|  | weight vector | 4.6.2 |
| , , , | weight values | 4.6.2 |
|  | weight initialization vector (optional, user-specified) | 4.6.3.2 |
|  | initial weight value resolution (optional, user-specified) | 4.6.3.2 |
|  | scaled predicted sample value | 4.7.1 |
|  | predicted central local difference | 4.7.1 |
|  | register size, in bits, used in prediction calculation | 0 |
| , | predicted sample value | b) |
|  | scaled prediction error | 4.8.1 |
|  | weight update scaling exponent | 4.8.2 |
| , | initial and final weight update scaling exponent parameters (user-specified) | 4.8.2 |
|  | weight update scaling exponent change interval (user-specified) | 4.8.2 |
| , | mapped prediction residual | 4.9 |
|  | prediction residual | 4.9 |
|  | difference between and nearest endpoint , | 4.9 |

Table B-3: Entropy Coder Quantities

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Meaning** | **Reference** |
|  | output word size in bytes | 5.2.2 |
|  | frame | 5.4.1.1.1 |
|  | sub-frame | 5.4.1.1.1 |
|  | sub-frame interleaving depth | 5.4.1.1.2 |
| **Sample-Adaptive Entropy Coder** | | |
|  | accumulator | 5.4.2.1.2 |
|  | counter | 5.4.2.1.2 |
|  | initial count exponent (user-specified) | 5.4.2.1.2.2 |
|  | accumulator initialization parameters (user-specified) | 5.4.2.1.2.3 |
|  | accumulator initialization constant (optional, user-specified) | 5.4.2.1.2.3 |
|  | rescaling counter size (user-specified) | 5.4.2.1.2.4 |
|  | unary length limit (user-specified parameter) | 5.4.2.1.3.1 |
|  | variable length code parameter | 5.4.2.1.3.3 |
|  | unary codeword length | 5.4.2.1.3.3 |
| **Block-Adaptive Entropy Coder** | | |
|  | resolution | 5.4.2.2.2.3 |
|  | block size | 5.4.2.2.2.4 |
|  | reference sample interval | 5.4.2.2.2.5 |
|  | number of coded data sets in compressed image body | 5.4.2.2.3.1 |

1. Security  
     
   (Informative)
   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.

* 1. 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.

* 1. POTENTIAL THREATS AND ATTACK SCENARIOS

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

* 1. 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.

1. Note for Tom Gannett: in the Block Size field, we’d eventually like for ‘10’ and ‘11’ to indicate block sizes J=32 and J=64 respectively. But at the moment, these values of J are not allowed by the current edition of [1]. I’m bringing this to your attention in case there is a “right” and “wrong” way to write a standard that anticipates the change of another standard. [↑](#footnote-ref-1)
2. Note for Tom Gannett: ultimately we want to allow J to be 8, 16, 32, or 64, but this assumes a revision of [1] to allow it. (Currently only J=8 and J=16 are allowed.) [↑](#footnote-ref-2)
3. Note for Tom Gannett: ultimately we want to allow *r* to be as large as 4096, but this assumes a revision of [1] to allow it. (Currently the maximum is 256.) [↑](#footnote-ref-3)