SCYLLARUS™

Beyond Vision

HSZ Data Structure and File Format

For HSZ Version 0.9

Publication Date 5 May 2014

The HSZ File Format, the Scyven Application, the Scyllarus system, the Scyllarus Matlab Toolbox and the Scyllarus C++ API are owned and managed by National ICT Australia Limited (NICTA). For more information go to www.scyllarus.com.

Table of Contents

Introduction

The Scyllarus Hyperspectral Zipped (HSZ) format is a compressed format for storing Hyperspectral images.

This format is used both in the HSZ file and the HSZ data structure in the Matlab environment.

Document Scope

This document describes the Hyperspectral Zipped (HSZ) format as used in both, the Scyllarus HSZ file and the Scyllarus Matlab toolbox.

This document also describes in sufficient detail the Scyllarus processing pipeline to allow understanding of the HSZ format.

Intended Audience

This document is intended for researchers and software engineers who wish to read and understand the Scyllarus HSZ file format of the Scyllarus Matlab Toolbox data structure. It assumes a workable knowledge of Hyperspectral imagery and the methods generally used to process it and store it.

Using this Guide

This guide is intended as a reference. Readers wishing to understand the Scyllarus processing pipeline and the theory behind it should refer to www.scyllarus.com

Reporting Problems

If you find a problem in this document, you can report it to NICTA using the Scyven web page. See http://feedback.scyven.com.

Note that NICTA cannot commit to fixing all reported problems and this document is provided as is.

References

[1] Link to the SLZ document goes here

[2] C. P. Huynh, A. Robles-Kelly, *A Solution of the Dichromatic Model for Multispectral Photometric Invariance*, International Journal of Computer Vision, 90(1), pp. 1-27, 2010.

[3] E. Angelopoulou, R. Molana, K. Daniilidis, *Multispectral Skin Color Modeling*, CVPR, pp. 635-642, 2001.

[4] https://www.exelisvis.com/ProductsServices/ENVI/ENVI.aspx

[5] http://www.hdfgroup.org/HDF5/

[6] G. D. Finlayson, G Schaefer, *Solving for colour constancy using a constrained dichromatic reflection model*. International Journal of Computer Vision, 42(3), pp. 127–144, 2001.

HSZ Format Overview

The HSZ Structure is used by Scyllarus in two ways.

- The Hyperspectral Zipped (HSZ) file format is used by Scyllarus to store compressed Hyperspectral images to disk using the HDF5 file format [5].
- HSZ is a data structure used in the Scyllarus Matlab toolbox.

The file format and the data structure have an identical hierarchical structure.

This common structure has two advantages. Firstly, it allows for the Scyllarus Matlab toolbox, C++ API and Scyven to be compatible in the manner they handle Hyperspectral data. Secondly, HSZ files can be opened and saved on a wide variety of platforms and operating systems using the libraries provided by the HDF group [5].

Theoretical Basis

The dichromatic model states that, at a given wavelength λ and pixel *u*, the image radiance is expressed by

Equation 1: $$I(u,\lambda) = L(\lambda)(g(u)S(u,\lambda) + k(u))$$

where

$L(\lambda)$ is the illuminant power spectrum,

g(u) is the shading factor,

k(u) is the specular coefficient and

$S(u,\lambda)$ is the reflectance spectrum

However, in an HSZ file, an additional wavelength dependent field $K(u,\lambda)$ is also included to model highlights and specular reflectance more flexibly. The HSZ file also used a pixelwise illuminant power spectrum. This yields the following modified Dichromatic model which is used by the current version of Scyllarus and supported by the HSZ structure.

Equation 2: $$I(u,\lambda) = L(\mathrm{u},\lambda)(g(u)S(u,\lambda) + k(u)K(u,\lambda))$$

Hierarchical Structure

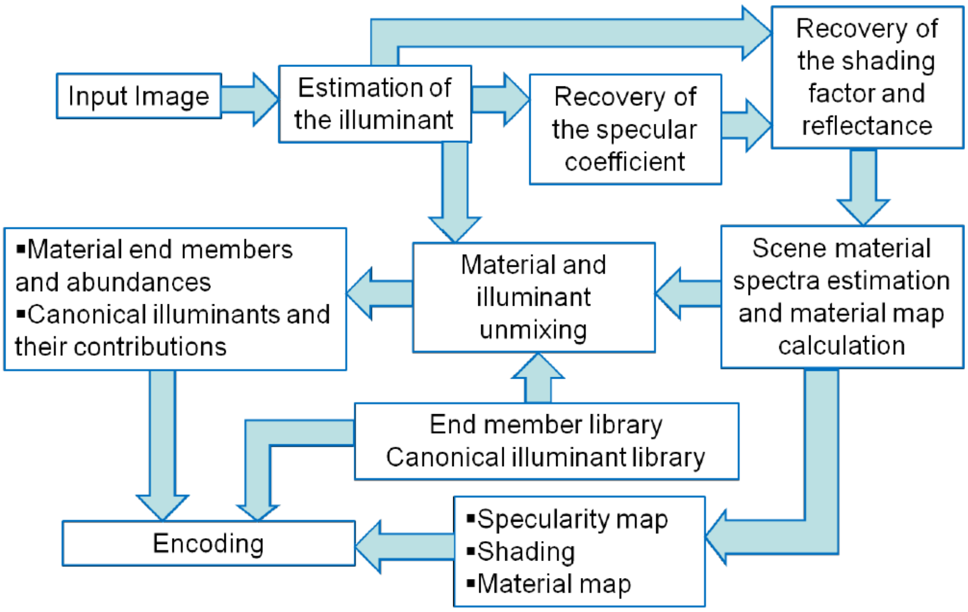

Figure 1: Diagrammatic structure of the Scyllarus processing pipeline

The HSZ Hierarchical structure arises naturally from the Scyllarus processing pipeline shown in Figure 12.

As an image is processed by Scyllarus, the illuminant power spectrum, specular highlights, shading and reflectance are computed and, if required, separated into linear combinations of materials, canonical illuminants, end members, etc. As a result, the HSZ structure consists of fields containing physically meaningful information.

At the top level of the HSZ hierarchy, these fields are:

HDR	A field containing ENVI compatible header fields and other tags specific to the HSZ structure.
HDREndmembersL	Header field for the end member library. This is identical to the HDR field on the SLZ where the library was loaded from. This field is only present when the **Illuminant Spectrum** has been unmixed with respect to a set of end member spectra. Refer also to reference **Error! Reference source not found.** for more information about the SLZ structure.
HDREndmembersS	Header field for the canonical illuminant library. Note that this field is only present when the **Reflectance Spectra** have been unmixed with respect to a set of canonical illuminant power spectra loaded from an SLZ file. Refer also to reference **Error! Reference source not found.** for more information about the SLZ structure.
L	A structure containing the illuminant information.
S	A field containing the reflectance data.

K	A structure containing the specular highlights.

Using HDF5 Fields

For the HSZ file format, the HDF5 file format is used to store the fields and the subfields that are described in this document.

String Fields and Subfields

If these fields or subfields are strings, they are stored using HDF5 attributes

Numerical and Binary Fields and Subfields

For numerical and binary subfields, they are stored as unsigned integers using HDF datasets.

Each of these subfields consist of DATA, MAX and MIN datasets such that the final value of the field is given by

$$x = \frac{DATA}{r}(MAX - MIN) + MIN$$

where r is the maximum value of the unsigned char type being used for saving the dataset. $DATA$ can be an array, whereas MIN and MAX are always single values.

When a sub-field has an array of dimensionality greater than 2, the last dimension is reserved for the non-spatial domain of the variable with respect to the image. That is, for the **HDR.wavelength** (the wavelength subfield of **HDR**), the second dimension corresponds to the wavelength domain. Similarly, for the **ElementAbundances** sub-field, the first two dimensions correspond to the spatial domain of the image.

HDR Field

The header (HDR) field in the HSZ format is based upon ENVI Standard tags. The ENVI Standard file header is compatible with several software vendors, such as ITT and ESRI. This also allows for the field to be populated in a straightforward manner from the input data for a flat file (FLA) as taken from its header file. For more information on the tags in ENVI Standard headers, see Appendix 1.

The general Scyllarus specific tags in the **HDR** are:

Method String attribute – Denotes the decomposition model used for the image irradiance. Currently, the only valid value for this field is **Dichromatic**. This is since the current version of Scyllarus is based on the dichromatic model of Finlayson and Schaefer [6]. This field is provided for forward compatibility.

The Scyllarus specific tags in the **HDR** which specify the encoding used for the illuminant in the **L** field are as follows.

EncodingL String attribute – Specifies the encoding scheme for the data in the Elements and Endmember sub-fields of the L field. It is always one of the following strings: **RAW**, **NURBS (for the representation in [2])** or **GMM (for the Gaussian mixture descriptor in [3])**. Note that the C++ API and the Scyven application up to version 0.8 only supports **NURBS** for illuminant spectra encoding.

degreeNURBSL Sub-field, numeric – The degree of the polynomial used for the B-Spline function used to represent the Illuminant spectra. It must be an integer greater than zero. If not present and the **EncodingL** attribute contains the string **NURBS**, use the default value of 2.

numGMMsL Sub-field, numeric – The number of Gaussians comprising the mixture used to represent the Illuminant spectra. This sub-field is required when the **EncodingL** attribute is set to **GMM**.

IndexedL Sub-field, Binary flag – If True (i.e. 1), it indicates that the pixel-wise illuminant is indexed to a set of illuminants which are representative of the lights in the scene and Unmixed Spectrum Decomposition is being used. See section *Unmixed* on page 15.

numElementsL Sub-field, numeric – The number of representative lights extracted from the scene. This sub-field is required when the **IndexedL** sub-field has the value True (i.e. 1).

EndmemberIndexedL Sub-field, Binary flag – If True (i.e. 1), it indicates that the representative lights in the scene have been unmixed into a combination of power spectra

	of canonical illuminants. See section *Canonical Endmember Set Stored in SLZ File* on page 16.
numEndmembersL	Sub-field, numeric – The number of canonical illuminants comprising each of the representative lights in the scene. This sub-field is required when the **EndmemberIndexedL** sub-field has the value True (i.e. 1).

The Scyllarus specific tags in the **HDR** which specify the encoding used for the Reflectance spectra in the **S** field are as follows. They are analogous to those for the Illuminant **L** sub-field above.

EncodingS	String attribute – Specifies the encoding scheme for the data in the Elements and Endmember sub-fields of the L field. It is always one of the following strings **RAW**, **NURBS (for the representation in [2])** or **GMM (for the Gaussian mixture descriptor in [3])**.. Note that the C++ API and the Scyven application up to version 0.8 only supports **NURBS** for Reflectance Spectra encoding.
degreeNURBSS	Sub-field, numeric – The degree of the polynomial used for the B-Spline function used to represent the Reflectance spectra. It must be an integer greater than zero. If not present and the **EncodingS** attribute contains the string **NURBS**, use the default value of 2.
numGMMsS	Sub-field, numeric – The number of Gaussians comprising the mixture used to represent the Reflectance spectra. This attribute is not supported in the current C++ API and the Scyven application, as they do not support GMM encoding for Reflectance spectra.
IndexedS	Sub-field, Binary flag – If True (i.e. 1), it indicates that the Reflectance spectra have been decomposed into the materials in the scene. See section *Unmixed* on page 15.
numElementsS	Sub-field, numeric – The number of materials comprising the scene reflectance. This sub-field is required when the **IndexedS** sub-field has the value True (i.e. 1).
EndmemberIndexedS	Sub-field, Binary flag – If True (i.e. 1), it indicates that the material spectra have been unmixed into a combination of spectra from an end member library. This sub-field must not be True unless **IndexedS** is also True. See section *Canonical Endmember Set Stored in SLZ File* on page 16.
numEndmembersS	Sub-field, numeric – The number of end members comprising each of the materials in the scene. This sub-field is required when the **EndmemberIndexedS** sub-field has been set to True (i.e. 1).

The Scyllarus specific tags in the **HDR** which specify the encoding used for the Highlight information in the **K** field are as follows. Note that the Endmember related fields which apply to both the Illuminant and Reflectance spectra do not apply to the **K** sub-field.

EncodingK	String attribute – Specifies the encoding scheme for the data in the Elements and Endmember sub-fields of the **K** field. It is always one of the following strings **RAW**, **NURBS (for the representation in [2])** or **GMM (for**

the Gaussian mixture descriptor in [3]).. Note that the C++ API and the Scyven application **up to** version 0.8 only supports **NURBS** for Highlight spectra encoding.

degreeNURBSK Sub-field, numeric – The degree of the polynomial used for the B-Spline function used to represent the Highlight spectra. It must be an integer greater than zero. If not present and the **EncodingK** attribute contains the string **NURBS**, the default value is 2.

numGMMsK Sub-field, numeric – The number of Gaussians comprising the mixture used to represent the Highlight spectra. This attribute is not supported in the current C++ API and the Scyven application, as they do not support GMM encoding for Highlight spectra.

IndexedK Sub-field, Binary flag – If True (i.e. 1), it indicates that the Highlight spectra have been decomposed into a set of prototype spectra representative of the specularities in the scene.

numElementsK Sub-field, numeric – The number of spectral prototypes used for the indexation of the Element subfield on **K**. This sub-field is required when the **IndexedK** sub-field has the value True (i.e. 1).

S, L and K Fields

The S (Reflectance Spectra), L (Illuminant power spectrum) and K (spectral refection factor) fields each independently store data which is based on the wavelength. Thus, for $L(u,\lambda)$ $\mathrm{S}(\mathrm{u},\lambda)$ and $\mathrm{K}(\mathrm{u},\lambda)$, ultimately one spectrum is stored for each pixel in the image. However, for each of these cases, the stored spectrum can be constructed as a linear combination of multiple element or end member spectra.

There are two aspects to representation of these spectra. In the following table, the rows list the three encoding methods for a single spectrum. The columns list the three ways that one or more spectra can be used to define a single spectrum from the Dichromatic model. The table contents indicate which of the Dichromatic model spectra can be represented with that combination of methods in the HSZ structure.

	Spectra Decomposition Methods		
Spectra Encoding Methods	**Direct representation**	**Unmixed**	**Canonical Endmember Set Stored in SLZ File**
RAW	L, S, K	L, S,K	L, S
NURBS	L, S, K	L, S, K	L, S *
Gaussian Mixture	L, S, K	L, S, K	L, S *

** Note that the SLZ Spectral Library files always store spectra in RAW form. However, when they are copied into the HSZ file as Endmembers, they may be represented using NURBS or Gaussian Mixture descriptors.*

The following sections describe how these methods are used and the HSZ fields which are relevant to each method.

Spectra Encoding Methods

RAW Encoding

To indicate that this encoding method is being used, the relevant **EncodingL**, **EncodingS** or **EncodingK** attribute in the **HDR** field is set to the string **RAW**. The same encoding method must be used for EndMembers and Elements if they are present.

If present, **degreeNURBSL/S/K** and **numGMMsL/S/K** sub-fields in the **HDR** field should be ignored when RAW Encoding is being used.

The following table shows the sub-fields which will be present in either the **L**, **S** or **K** field and their meanings. The symbols in the size column take the following meanings.

N The number of Spectra being encoded. This number depends on the Spectra Decomposition Method in use. For representing **S** and **K**, this symbol may represent two dimensions, being the width and height of the image.

B The number of wavelength bands in the image, from the ENVI tag **bands** sub-field. The wavelengths for these bands are found in the ENVI tag **wavelength** sub-field of the **HDR** field.

Sub-field Name	Sub-field Size	Description
Elements	B x N	The wavelength indexed values of the spectraunder consideration..
Endmembers	B x N	Present only if decomposition to Endmembers has been done. The wavelength indexed values of the endemember spectra are stored here.

NURBS Encoding

To indicate that this encoding method is being used, the relevant **EncodingL/S/K** attribute of the **HDR** field is set to the string **NURBS**.

The **degreeNURBSL/S/K** sub-field must be present in the **HDR** field.

If present, the **numGMMsL/S/K** sub-field in the **HDR** field should be ignored.

B-Spline line interpolation is used in the XY plane, where the X axis is the wavelength and the Y axis is the wavelength domain.

The Elements and Endmember spectra are computed independently. Depending on the Decomposition method being used, the Endmember sub-fields may not be present.

The following table shows the sub-fields which will be present in either the **L**, **S** or **K** field and their meanings. The symbols in the size column are given as follows:

D The Degree of the B-Spline interpolation from the **degreeNURBSL/S/K** sub-field in the **HDR** field.

N The number of Spectra being encoded. This will depend on the Decomposition method being used.

C, C’ The number of control points used to represent each spectrum. This value is determined by reading the size of the Elements or Endmembers sub-field. The number of B-Spline control points can be different for Elements and Endmembers sub-fields.

Sub-field Name	Sub-field Size	Description
Elements	C x N	The X coordinates of the control points.

ElementsCP	C x N	The Y coordinates of the control points.
ElementsKnots	(C + D + 1) x N	The knot vector for the B-Spline calculation.
Endmembers	C' x N	Present only if decomposition to Endmembers has been done. The X coordinates of the control points. See section *Canonical Endmember Set Stored in SLZ File* on page 16.
EndmembersCP	C' x N	Present only if decomposition to Endmembers has been done. The Y coordinates of the control points. See section *Canonical Endmember Set Stored in SLZ File* on page 16.
EndmembersKnots	(C' + D + 1) x N	Present only if decomposition to Endmembers has been done. The knot vector for the B-Spline calculation. See section *Canonical Endmember Set Stored in SLZ File* on page 16.

Gaussian Mixture Encoding

To indicate that this encoding method is being used, the relevant **EncodingL/S/K** attribute of the **HDR** field is set to the string **GMM**.

The **numGMMsL/S/K** sub-field must be present in the **HDR** field. It indicates how many Gaussian mixture components will be used to represent each spectrum.

If present, the **degreeNURBSL/S/K** sub-field in the **HDR** field should be ignored.

Each of the Elements and Endmember spectra are computed independently. Depending on the Decomposition method being used, the **Endmember*** sub-fields may not be present.

The following table shows the sub-fields which will be present in either the **L**, **S** or **K** field and their meanings. The symbols in the size column are the following:

D — The number of Gaussians used to represent a spectrum, from the **numGMMsL/S/K** sub-field in the **HDR** field.

N, N' — The number of Spectra being encoded. This will depend on the Decomposition method being used. This number will be different for Elements and Endmembers.

C, C' — The number of Gaussian curves used to represent each spectrum. This value is determined by reading the size of the Elements or Endmembers sub-field. The number of Gaussian curves can be different for Elements and Endmembers sub-fields.

Sub-field Name	Sub-field Size	Description
Elements	C x N	The mixture weight vectors.
ElementMean	C x N	The vectors of mean values for the Gaussiansin the mixture.
ElementStd	C x N	The vectors of standard deviations values for the Gaussians in the mixture.

Endmembers	C' x N'	Present only if decomposition to Endmembers has been done. The mixture weight vectors. See section *Canonical Endmember Set Stored in SLZ File* on page 16.
EndmembersMean	C' x N'	Present only if decomposition to Endmembers has been done. The vectors of mean values for the Gaussian curves in the mixture. See section *Canonical Endmember Set Stored in SLZ File* on page 16.
EndmembersStd	C' x N'	Present only if decomposition to Endmembers has been done. The vectors of standard deviations values for the Gaussian curves in the mixture. See section *Canonical Endmember Set Stored in SLZ File* on page 16.

With the notation above, the intended spectrum is constructed as the weighted sum of Gaussian curves.

$$Spectrum(n) = \sum_{i=1}^{C} Elements(i,n) * Gaussian(ElementMean(i,n), ElementStd(i,n))$$

The process is similar for Endmembers, if they are present.

Spectra Decomposition Methods

Direct Representation

This refers to the method of storing the Spectra directly, without decomposition, for each pixel or for the illuminant. This method does not yield effective compression, but can be used if exact representation is required.

The following table shows the sub-fields which will be present in either the **L**, **S** or **K** field and their meanings. The symbols in the size column take the following meanings.

i The width of the image in pixels.

j The height of the image in pixels.

Sub-field Name	Sub-field Size	Description
Elements, **ElementsCP**, **ElementsKnots**, **ElementsMean**, **ElementsSTD**	Depends on Spectra Encoding Method	These sub-fields store the spectra represented accordingly to the encoding scheme chosen, whereby the **Elements** sub-field is used to store the spectra under consideration. See the description of the Spectra Encoding Methods above for more information about the other sub-fields will be present.
Factor	i x j	Present for **S** and **K** only. For **S** it is $\mathrm{g(u)}$and for **K** is it $\mathrm{k(u)}$ as shown in *Equation 2* on page 5.

Unmixed

Each spectrum has been decomposed into a limited number of spectra which must be re-combined in linear combinations to reconstruct the original spectra. The physical meaning of these decomposed spectra varies by field type.

L The Illuminant has been decomposed into multiple scene Illuminants.

S The Reflectance has been decomposed into multiple material spectra.

K The Highlight spectra have been decomposed into multiple characteristic highlights.

The following table shows the sub-fields which will be present in either the **L**, **S** or **K** field and their meanings. The symbols in the size column take the following meanings.

S The sizes of the various **Elements*** sub-fields as required by the Spectra Encoding method. It is not always the same for each of the Elements sub-fields. Refer to the Spectra Encoding methods described above in the Spectra Encoding Methods section.

N The number of spectra used for decomposition as specified in the **numElementsL/S/K** sub-field of the **HDR** field.

M The number of decomposed spectra, selected from the spectra specified in the **Elements*** sub-fields, which will be combined to construct each spectrum. This number must not exceed N.

i x j The image width and height.

C The number of spectra to be recombined. In general, this field will hold a 3D array.

Sub-field Name	**Sub-field Size**	**Description**
Elements, **ElementsCP**, **ElementsKnots**, **ElementsMean**, **ElementsSTD**	S x N	These sub-fields store the decomposed spectra. Elements will always be present. See the description of the Spectra Encoding Methods above for more information about which of the other sub-fields will be required depending on the chosen encoding scheme.
ElementAbundanceIndexes	C x M	For each spectrum to be reconstructed, this matrix holds a set of M indexes into the Elements array. Each index is in the range 1...N.
ElementAbundances	C x M	For each spectrum to be reconstructed, this matrix holds a set of weights, which are used when combined the spectra identified by the **ElementAbundanceIndexes** sub-field.
Factor	i x j	Present for **S** and **K** only. For **S** it is $\mathrm{g(u)}$and for **K** is it $\mathrm{k(u)}$ as shown in *Equation 2* on page 5.

By way of example, when constructing $S(u,\lambda)$ for pixel u with coordinate (i,j) in the image lattice, for N materials per pixel, the Reflectance is given by

$$S(i,j,\lambda) = \sum_{k=1}^{N} \{S.ElementAbundances(i,j,k) * S.Elements(S.ElementAbundanceIndexes(i,j,k),\lambda)\}$$

where $S.Elements(\cdot)$ indicates that **Elements** is a subfield of **S**.

For **RAW** Spectrum Encoding, λ represents a simple index into the **Elements** sub-field matrix. For other Encoding Methods, $S.Elements(m,\lambda)$ must be computed for each λ based on the other **Elements*** sub-fields for the m^{th} spectrum and the encoding method.

Canonical Endmember Set Stored in SLZ File

This method applies only to the S and L sub-fields. It is not available for Highlight spectra.

This Decomposition Method adds further fields to the Unmixed method. However, when reconstructing the image, the method explained in the Unmixed description above should be employed. The additional fields are not required for simple image reconstruction. Instead, they provide further information about the breakdown of the Elements Spectra into their likely components.

These additional fields can be used to reconstruct the Elements spectra from the Endmembers which have been identified in the Spectral Library Zipped (SLZ) file. The accuracy of this reconstruction will depend on the appropriateness of the spectra in that library and, if no spectra in that library specifically relate to the spectra in this image, the reconstruction may be quite inaccurate.

When this decomposition method is used for Illuminant decomposition, the **HDREndmembersL** field must be present at the HSZ structure's top level. Similarly, when this decomposition method is used for Reflectance decomposition, the **HDREndmembersS** field must be present. These two fields are copied from the related Spectral Library SLZ file's **HDR** field.

The following table shows the sub-fields which will be present in either the **L**, **S** field and their meanings. The symbols in the Sub-field Size column take the following meanings.

S — The sizes of the various **Endmembers*** sub-fields as required by the Spectra Encoding method. It is not always the same for each of the **Endmembers*** sub-fields. Refer to the relevant Spectra Encoding methods described above in the *Spectra Encoding Methods* section.

N — The number of end member spectra used for decomposition as specified in the **numEndmembersL/S/K** sub-field of the **HDR** field.

M — The number of decomposed spectra, selected from the spectra specified in the **Endmembers*** sub-fields, which will be combined to re-construct each spectrum. This number must not exceed N.

C The number of spectra to be recombined as specified in the **numElementsL/S/K** sub-field of the **HDR** field.

Sub-field Name	Sub-field Size	Description
Endmembers, **EndmembersCP**, **EndmembersKnots**, **EndmembersMean**, **EndmembersSTD**	S x N	These sub-fields store the decomposed spectra. Endmembers will always be present. See the description of the Spectra Encoding Methods above for more information about which of the other sub-fields are required depending on the encoding scheme used. These sub-fields have been constructed by copying the spectra found in this image from the Spectrum Library and encoding them accordingly. For NURBS encoding, the number of Control Points may vary from that used for the **Elements*** sub-fields. The B-Spline degree will be the same.
EndmemberAbundanceIndexes	C x M	For each spectrum to be reconstructed, this matrix holds a set of M indexes into the Elements array. Each index is in the range 1...N. This is a matrix that contains the indexes corresponding to the spectra in Elements that correspond to the pixel under consideration.
EndmemberAbundances	C x M	For each spectrum to be reconstructed, this matrix holds a set of weights, which are used when combined the spectra identified by the **ElementAbundanceIndexes** sub-field. This is a matrix containing the abundances of the Endmembers in the unmixed spectra.

By way of example, to re-construct the i^{th} spectrum from the Elements array we can use$S.Elements(i,\lambda) = \sum_{k=1}^{M}\{S.EndmemberAbundances(i,k) * S.Endmembers(S.EndmemberAbundanceIndexes(i,k),\lambda)\}$

where, as before, $S.Elements(\cdot)$ indicates that **Elements** is a subfield of **S**.

For **RAW** Spectrum Encoding, λ represents a simple index into the relevant sub-field matrix. For other Encoding Methods, $S.Endmembers(m,\lambda)$ must be computed for each λ based on the other **Endmembers*** sub-fields for the m^{th} spectrum and the encoding method.

Document History

Release	Date	Author	Description
0.1 draft		Antonio Robles-Kelly	First draft.
0.2 draft	28 April 2014	Peter Roberts	Draft ready for Technical Review.
0.3 draft	1 May 2014	Antonio Robles-Kelly	Technical Review of Draft 0.2.

Appendix 1. ENVI Header Structure

General Rules

Curly brace usage: Used when a keyword contains more than one value.

Entry format: Entries have to use one of the two following formats (entry ENVI excepted):

Keyword = value, or

Keyword = {value1, value2, …, value n}

Nesting: This is allowed for entries.

Libraries: Spectral library files in ENVI are widely used. These contain the additional keyword: spectra names. This additional keyword contains a comma-separated list of ASCII names enclosed in curly brackets.

Additional keywords: Additional keywords for specific required information. For example, classification results files will contain the additional keywords: classes, which defines the number of classes, including the unclassified, class lookup, which lists RGB color definitions for each respective class, and class names. These tags are to be used for the pipeline output header files.

Coordinates: In ENVI, pixel values always refer to the upper-left corner of the pixel. Map coordinates also typically refer to the upper-left corner of the pixel. However, if you entered "magic pixel" coordinates in the ENVI header, the map coordinates would refer to the x,y coordinates entered (e.g., x=1.5, y=1.5 would make the map coordinates refer to the center of the pixel).

Comments: Comments can be added to ENVI header files by inserting a line with a semicolon as the first character. These lines are ignored when the header file is parsed. Comments can appear anywhere within a header file but must be an entire line (and the semicolon must be the first character of that line). Comments cannot follow a keyword/value pair.

Required Header Entries

ENVI No value

description This is a character string describing the image or processing performed. It may contain the following tags

time and date These are in the format hh:mm:ss, weekday, month dd, yyyy. The time is given in 24hrs formal, with the month and the weekday denoted in full in words.

gain Floating point value in the interval [0,1] (if there's no information on the gain, it should be set to "1").

exposure time This is a vector of exposure times per band in a vectorial form. The units are given in parenthesis next to the tag.

exposure time units Text string indicating the unit of measure for the exposure time.

GPS data Time, date, latitude, longitude, heading (or bearing), satellites in view and fix quality. This is for a mobile/vehicle mounted camera system.

headeroffset Number of bytes of imbedded header information present in the file (for example 128 bytes for ERDAS 7.5 `.lan` files). These bytes are skipped when the ENVI file is read.

samples Number of columns in the image.

lines Number of rows. The spatial resolution in the image is, hence, given by samples x lines

bands Number of wavelength indexed bands.

data type Data representation type, where 1=8 bit byte; 2=16-bit signed integer; 3=32-bit signed long integer; 4=32-bit floating point; 5=64-bit double precision floating point; 6=2x32-bit complex, real-imaginary pair of double precision; 9=2x64-bit double precision complex, real-imaginary pair of double precision; 12=16-bit unsigned integer; 13=32-bit unsigned long integer; 14=64-bit signed long integer; and 15=64-bit unsigned long integer.

interleave Denotes the interleaving on the data. These may be band sequential (bsq), band interleaved by pixel (bip), or band interleaved by line (bil)

sensor type Type of sensor used to acquire the data (if there's no information on the sensor, it should be "unknown").

byte order Describes the order of the bytes in integer, long integer, 64-bit integer, unsigned 64-bit integer, floating point, double precision, and complex data types; Byte order=0 is Least Significant Byte First (LSF) data (DEC and MS-DOS systems) and byte order=1 is Most Significant Byte First (MSF) data (all others - SUN, SGI, IBM, HP, DG).

wavelength Lists the center wavelength values of each band in an image. Units should be the same as those used for the **fwhm** parameter (see below) and set in the wavelength units parameter

Optional Header Entries

wavelength units Units for the wavelength indexed band centers

projection info This keyword can be nested and is added to the ENVI header file if a user-defined projection is used instead of a standard projection.

default bands If set, indicates which band numbers to automatically use to construct an RGB pseudo-color image. If only 1 band number is used, then a gray scale image is

	constructed.
reflectance scale factor	The value that scales the image cube to be between zero and unity.
file type	ENVI defined file type. The file type ASCII string must match verbatim, including case. This file is an ASCII file that lists the specific file types that are used in the ENVI header to identify special files. These file types include ENVI specific files such as meta files, classification results, virtual mosaics, spectral libraries, and fft results. The file types also include data specific formats such as ADRG and AVHRR. Files such as TIFF and BMP are also recognized as well as ERDAS 8.x and PCI files. The file type field allows these files to have an ENVI header but still exist in their native formats. These types can be edited or even be user defined.
map info	Time, date, latitude, longitude, heading (or bearing), satellites in view and fix quality. This is for a mobile/vehicle mounted camera system.
x start and **y start**	Image coordinates for the upper-left hand pixel in the image. The values in the header file are specified in "file coordinates," which is a zero-based number.
fwhm	Full-width-half-max values of each band in an image. Units should be the same as those used for wavelength and set in the wavelength units parameter.
z plot range	Default minimum and maximum values for Z plots.
z plot average	Number of pixels in the X and Y directions to average for Z plots.
z plot titles	X and Y axis titles for Z plots.
pixel size	Indicates width and length pixel size in meters for non-georeferenced files
default stretch	Determines what type of stretch (% linear, linear range, Gaussian, equalize, square root) is used when the image displays
band names	List of names for each band.
bbl	List of the bad band multiplier values for each band in an image, typically zero for bad bands and one for good bands.
data gain values	List of individual gain values per band. This has to be multiplied by the value in the file description gain parameter.
data offset values	List of individual offset values per band. These are the number of bytes to be skipped when the ENVI FLA file is read.

Typical ENVI standard headers

This is the typical header for the ENVI standard file. This corresponds to an image acquired on March 3rd 2006 with an OKSI camera in the visible range. The image has resolution 1392 x 1040 and spans 30 wavelength indexed bands

```
ENVI
```

```
description = {
   17:06:56, Friday, March 03, 2006
gain = 1.000
exposure time units = ms
exposuretime = {
 200.0,200.0,200.0, 200.0, 200.0, 200.0, 200.0, 200.0, 200.0, 200.0, 200.0, 200.0,
200.0, 200.0, 200.0, 200.0, 200.0, 200.0, 200.0, 200.0, 200.0, 200.0, 200.0, 200.0,
200.0, 200.0, 200.0, 200.0, 200.0, 200.0 } }
offset = 0
samples = 1392
lines   = 1040
bands   = 30
header offset = 0
filetype = ENVI Standard
data type = 12
interleave = bsq
sensor type = OKSI
byte order = 0
wavelength units = nm
wavelength = {
   430,440, 450, 460, 470, 480, 490, 500, 510, 520, 530, 540,550, 560, 570, 580,
590, 600, 610, 620, 630, 640, 650, 660, 670, 680, 690, 700, 710, 720 }
```

The following header file corresponds to an image acquired on Dec 20th 1995 by Landsat TM. The image has resolution 709 x 946 and spans 7 wavelength indexed bands and is geoindexed.

```
ENVI
description = {
Registration Result. Method: 1st degree Polynomial w/ nearest
neighbor [Wed Dec 20 23:59:19 1995] }
samples = 709
lines   = 946
bands   = 7
header offset = 0
file type = ENVI Standard
data type = 1
interleave = bsq
sensor type = Landsat TM
byte order = 0
map info = {UTM, 1, 1, 295380.000, 4763640.000, 30.000000,
30.000000, 13, North}
z plot range = {0.00, 255.00}
z plot titles = {Wavelength, Reflectance}
pixel size = {30.000000, 30.000000}
default stretch = 5.0% linear
band names = {
 Warp (Band 1:rs_tm.img), Warp (Band 2:rs_tm.img), Warp (Band
3:rs_tm.img), Warp (Band 4:rs_tm.img), Warp (Band 5:rs_tm.img),
Warp (Band 6:rs_tm.img), Warp (Band 7:rs_tm.img)}
wavelength = {
  0.485000,  0.560000,  0.660000,  0.830000,  1.650000, 11.400000,
2.215000}
fwhm = {
 0.070000, 0.080000, 0.060000, 0.140000, 0.200000, 2.100000,
0.270000}
```