Multispectral Image File Format AIX

Version: 1.6, Rev.: 1

1. Introduction

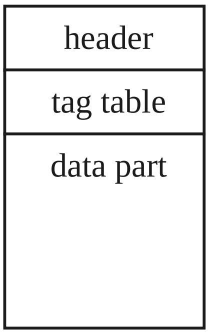

Fig 1: Basic file structure

The multispectral image file format AIX is a generic data container for spectral images in a user-defined spectral range. The image data is stored in a tag-oriented structure. It is possible to store uncompressed spectra as well as spectra which are composed by basis functions resulting from principal component analysis or any other type of decomposition.

The AIX file consists of three main parts: a header, a tag table and a data part. The header contains the basic image information, such as image height and width in pixels and the number of frames. The tag table contains the names and offsets of all tags. The data part contains the tags, which carry the image information. There are tags containing the frame data as well as tags containing calculation rules which define how to decode the spectra from the frame data.

The default way of extracting the spectra from the file is the following (see fig. 2): Due to the format of the raw data a scaling operation can be necessary first. After this, the scaled data is multiplied with the matrix taken from the “Samples2Spectrum” tag, resulting in the reconstructed spectrum. This matrix is the identity matrix in case the frames contain uncompressed spectra. On the other hand, if the data is compressed, the frames will contain the weighting factors and the matrix will contain the basis functions obtained by, e.g., principal component analysis (PCA).

In some cases, a default visualization method of the reconstructed spectra is needed. To allow for this, a tag called “photometric interpretation” is included. The matrix stored in this tag is meant to be applied to the spectra that are reconstructed using the frame data and the reconstruction matrix. In the photometric interpretation matrix, the user can store any linear transform that calculates a visualization of the reconstructed spectra. In the case of spectra inside the visible range, this could be the CIE 1931 standard observer. The transform is not limited to the visible range. Ultraviolet or infrared spectra can be visualized as well by using e.g. pseudocolors.

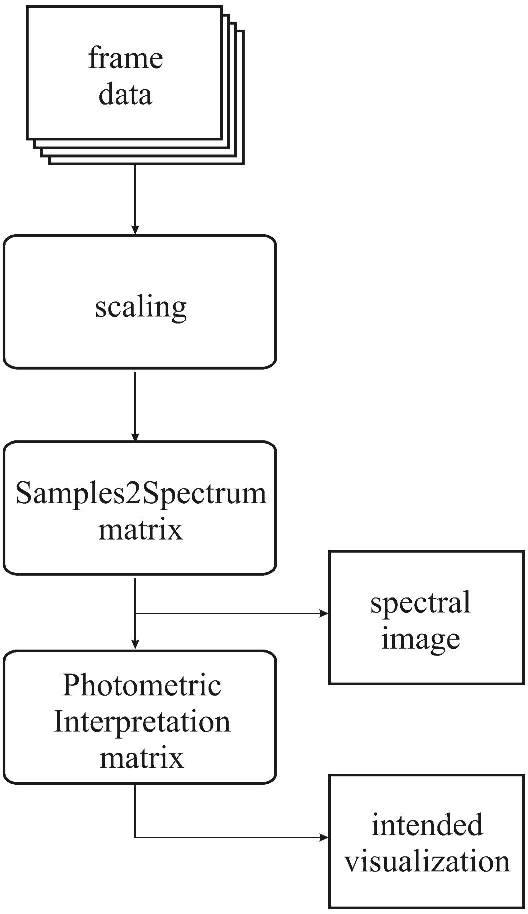

Fig. 2: Data flow for spectral reconstruction and visualization of data stored in the AIX-format. First, the multichannel frame data is scaled. After this, the “Samples2Spectrum” matrix is applied in order to gain the spectral image. If needed, the reconstructed spectra can be visualized by a second matrix, the photometric interpretation matrix.

2. Basic Structure

The file extension is “.aix”. The file consists of a fixed-length header, a tag table, and a subsequent data part containing the tags. The image data as well as the spectral reconstruction matrix and other matrices are stored in the tags.

All data is stored in big endian byte order.

2.1. Header

Offset	Length	Format	Contents	Remarks
0	4	Byte	"AIX "	4 ascii characters identifying the AIX format
4	4	Byte	"0160"	4 ascii characters identifying the version
8	4	Byte	0x00	reserved; padded with zeros
12	2	UInt16		number of frames N
14	2	UInt16	0x00	reserved; padded with zeros
16	4	UInt32		dataWidth in pixels
20	4	UInt32		dataHeight in pixels
24	4	Fixed16.16		ppiX, pixels per inch, horizontal direction
28	4	Fixed16.16		ppiY, pixels per inch, vertical direction
32	28	Byte	0x00	reserved; padded with zeros
60	4	UInt32		no. of tags
64				begin of tag table

2.2. Tag Table

2.2.1. Tag Table

Offset	Length	Format	Contents
0	20	TagTableEntry	tag code, offset from file start, and length of entry
20	20	TagTableEntry	next tag table entry
...			

2.2.2. TagTableEntry

Offset	Length	Format	Contents
0	4	Byte	tag code
4	8	UInt64	offset from file start
12	8	UInt64	length of tag

2.3. Tags

This part of the file contains the image data and at least one matrix for reconstruction purposes. The data is organized in tags.

Tag Code	Tag Name	Remarks
"S2SP"	Samples2Spectrum	spectral reconstruction matrix
"FRxx"	Frame	multispectral data frame. “xx” identifies the channel no., binary encoded as UInt16 number. There can be up to 2^16 "FRxx" tags.
"PHIx"	PhotometricInterpretation	matrix indicating how the reconstructed spectra are meant to be interpreted
"CMTx"	Comment	space for comments and remarks
"XMP "	XMP	Extensible Metadata Platform (see www.adobe.com)

2.3.1. "Samples2Spectrum"

This tag contains a matrix that is multiplied with the frame data in order to achieve the reconstructed spectrum for each pixel successively.

Offset	Length	Format	Contents	Remarks
0	4	Byte	"S2SP"	
4	4	Fixed16.16		lambda_start; first wavelength
8	4	Fixed16.16		lambda_end; last wavelength
12	4	Fixed16.16		delta_lambda; wavelength increment
16	2	UInt16		no. of data frames N
18	2	UInt16		no. of spectral samples M
20	2	UInt16		data type (1=float,2=double)
22	10	Byte	0x00	reserved; padded with zeros
32	N x M x sizeof(data type)			spectral reconstruction matrix

2.3.2. "Frame"

In the frame tag, the image data is stored. Before applying the “Samples2Spectrum” matrix the raw frame data may need to be scaled by dividing by the value of “scale”:

scaled data = raw data / scale.

Then, the spectral reconstruction matrix is applied to the scaled data. The division assures that the data is correctly scaled for the matrix operation (white balance).

Example: The frames contain unmodified raw data of a 12 bit camera. The exposure times have been adjusted in a way that the camera’s white response is approximately 0x8000 (⇒ “scale” = 0x8000).

Note that the field “scale” must have the same length in bytes as the frame data (i.e. “bytes per sample”).

Offset	Length	Format	Contents	Remarks
0	4	Byte	"FRxx"	“xx” identifies the channel no., binary encoded as UInt16 number. There can be up to 2^16 "FRxx" tags.
4	2	UInt16		bytes per sample
6	2	UInt16		bits per sample
8	2	UInt16		type of data compression
10	2	UInt16		compression quality level; for lossy compression only (comprJPG)
12	20	Byte	0x00	reserved; padded with zeros
32	Bytes per Sample			scale
32 + Bytes per sample	Bytes per Sample x dataWidth x dataHeight			frame data; the data is written row by row, starting in the top left corner and ending in the bottom right corner.

Though at first sight they seem to be redundant, both “bytes per sample” and “bits per sample” must be specified. It allows the user to account for different measurement devices: In case the image contains data captured by an 8 bit camera, “bytes per sample” will be 1 and “bits per sample” will be 8. In case a 12 bit camera is used, “bytes per sample” will be 2 and “bits per sample” will be 12. The selection of “bytes per sample” determines the way the data must be interpreted by the computer. Please refer to the table “bytes per sample and frame data interpretation” for an overview of allowed values of “bytes per sample” and the respective data interpretation. In case negative frame values are to be stored (e.g. if a set of orthogonal basis functions is used in the s2sp matrix) the float data format is suitable only.

bytes per sample and frame data interpolation

“bytes per sample”	frame data interpretation
1	UInt8
2	UInt16
4	float

compression type

Format	Number	Remarks
comprNone	0	no compression
comprZIP	1	ZIP algorithm, where a library (zlib) is available
comprJPG	2	12-bit type JPEG compression (see JPEG-lib)

2.3.3. "PhotometricInterpretation"

This tag contains a matrix that must be used if a default visualization of the spectral data is intended. The matrix is applied to the reconstructed spectrum, i.e. after the tag “Samples2Spectrum” has been applied to the channel values. In the photometric interpretation matrix, the user can store a linear transform of the reconstructed spectrum (e.g. XYZ, RGB, ...). Any arbitrary linear transform of the reconstructed spectrum is possible, such that invisible spectra (UV, IR, ...) can be displayed in pseudocolors or grayscale images, as well.

Up to 256 instances of this tag can be present in the file. The short descriptor (16 bytes) and the long descriptor (256 bytes) must be used to describe which transform is stored in the respective photometric interpretation matrix. There is a limited list of valid strings for the short descriptor (see list), while the content of the long descriptor can be chosen freely. It can be used to give more detailed information about the matrix transform, such as light source or type of observer.

Offset	Length	Format	Contents	Remarks
0	4	Byte	"PHIx"	There can be up to 256 instances of this tag type. "x" represents the number of the tag, binary encoded as UInt8 number.
4	16	Byte		short descriptor, 16 ascii characters
20	256	Byte		long descriptor, 256 ascii characters
276	2	UInt16		no. of spectral samples M
278	2	UInt16		no. of channels P in visualization image
280	2	UInt16		data type (1=float, 2=double)
282	230	Byte	0x00	reserved; padded with zeros
512	M x P x sizeof(data type)			photometric interpretation matrix

The first tag of this kind, i.e. "PHIx" where x=0x00, is the default version of the tag. If it exists, the interpreting software should use this tag as the default interpretation of the reconstructed spectrum.

2.3.4. "Comment"

In this tag, the user can store comments and remarks in a 256 byte character string. Up to 256 instances of this tag can exist in the file.

Offset	Length	Format	Contents	Remarks
0	4	Byte	"CMTx"	There can be up to 256 instances of this tag type. "x" represents the number of the comment tag, binary encoded as UInt8 number.
4	256	Byte		Contains the character string. If the string is shorter than 256 characters, zeros (0x00) are padded up to the length of 256 bytes.

valid short descriptor strings

Contents	Remarks
XYZ	transform to XYZ
RGB	transform to RGB
GRAYSCALE	transform to grayscale image
USERDEFINED	user defined transform specified in long descriptor

2.3.5. "XMP"

XMP (Extensible Metadata Platform) provides a standard format for the creation, processing, and interchange of metadata. This tag includes an XMP packet. The XMP data must be encoded as UTF-8. See http://www.adobe.com/products/xmp/ for more information. This tag can be used to store data such as camera settings, exposure times, or any other kind of metadata.

Offset	Length	Format	Contents	Remarks
0	4	Byte	"XMP "	There can be up to one instance of this tag.
4	8	UInt64	data length	Contains the length of the XMP data
12	data length	Byte	XMP data	Contains the XMP data

3. Examples

Example 1: storing a multispectral image

A multispectral image generated by a multispectral camera with 10 channels in the visible range is to be stored in the AIX-format. In this case, the channel signals are saved in the frame tags. The matrix transforming the frame data to spectra is stored in the “Samples2Spectrum” tag. The scientist who took the picture wants the image to be displayed in RGB under illuminant D50 for the 1931 standard observer, so he calculates the respective matrix transform and stores it in the photometric interpretation tag.

Example 2: storing a set of spectral measurements

Another scientist measured a set of 100 solar spectra from 400 nm to 2500 nm in 5 nm steps. He wants to draw the attention of the observer to a special effect in the infrared band. Thus, he stores the raw spectra in the frames which results in (2500 – 400) / 5 + 1 = 421 frames containing 100 measurements respectively. Frame 0 contains all values sampled at 400 nm, frame 1 contains all values sampled at 405 nm and so on. In order to “reconstruct” the spectra from the frame data, the identity matrix is needed and stored in the “Samples2Spectrum” tag.

In order to draw the observer’s attention to the 900 nm – 930 nm band, he generates a matrix which, being applied to the “reconstructed” spectra, adds up all values of the mentioned channels and suppresses the frame values outside this band. The result is a grayscale image visualization of the intended infrared part of the solar spectra.