A Spectral Extension to the Tagged Image File Format

Alexander Wilkie, Robert F. Tobler and Werner Purgathofer

Institute of Computer Graphics
Vienna University of Technology
Karlsplatz 13/186/2
A-1040 Wien
Austria
e-mail: {wilkie,rft,wp}@cg.tuwien.ac.at

ABSTRACT

In this paper we present a simple, consistent and yet powerful spectral extension to the Tagged Image File Format (commonly referred to as TIFF). This extension would presumably turn TIFF, which is already one of the most frequently used formats for "normal" image storage tasks, into a standard also for spectral image storage.

Keywords: spectral imaging, image file format

1 SPECTRAL IMAGES

Since the inception of digital imaging some decades ago, an almost uncountable number of conventional image file formats has been defined. What is still lacking is a standardised image file format for spectral image data.

1.1 Definition

For the purposes of this paper, we assume the term *spectral image file format* to refer to image file formats that store *spectral data* for each pixel, as opposed to the far more common *colour values*. Generally, spectral data consists of measurements of physical quantities (the intensity of electromagnetic radiation), measured in physical units without any regard for what the human visual system can discern, while colour values (the quantification of a visual sensation, usually as a function of some basis colours like *e.g.* red, green and blue) are intrinsically limited to what humans can actually *see*.

The spectral data which is stored for each pixel is more specifically the intensity of electromagnetic radiation *per wavelength*, so that one can reconstruct a visual sensation from it. This can be done since the relationship between spectra as a visual stimulus and their corresponding colour sensations is well understood (for a thorough coverage of this topic, see [2]; for encyclopædic reference on it, see [7]).

1.2 Scope

In contrast to "normal" image formats, the contents of a spectral image need not be something that human vision can directly perceive at all. The human eye is only sensitive to wavelengths between (at the utmost) 380 and 830 nanometers; if one wants to visualise other wavelengths one has to resort to false colour techniques of some kind.

The more obvious candidates that will have use for such a format are natural sciences like geology (remote sensing), astronomy (multispectral imaging), and of course computer graphics (especially for correct texture mapping in spectral rendering and tone mapping approaches such as [4]).

2 THE TAG IMAGE FILE FORMAT

TIFF was developed by Aldus in the mid-1980s. The current revision is 6.0, dating from June 1992 [1]. Since then several enhancements have been proposed, most notably the logLuv extension by Greg Ward–Larson[6], but no new revision has been published so far.

The purpose of TIFF is to store raster image data, and to this end it provides a rich set of features and predefined image formats. Though TIFF is a complex format, it can easily be used for simple purposes because the set of required fields (opposed to the *possible* fields) is small.

A high priority of the developers was to give TIFF an open structure, so that future extensions can easily be added. This is a fundamental difference from most other image formats, and one of the main reasons why we chose it to be the target of our extension.

The following paragraphs describe the basic format of a TIF. The content is an excerpt from the TIFF 6.0 specification [1].

2.1 Structure

A TIFF file begins with an 8-byte image file header that points to an image file directory (IFD). An image file directory contains information about the image, as well as pointers to the actual image data.

A TIFF field is a logical entity consisting of TIFF tag and its value. This logical concept is implemented as an IFD Entry, plus the actual value if it doesn't fit into the value/offset part, the last 4 bytes of the IFD Entry.

There is a number of predefined image types in the 6.0 spec, such as *e.g.* Bilevel, Grayscale, RGB Full Color, CMYK Images, and CIE L*a*b* Images. For each image type, there is a number of fields that are required by the specification; additional ones, like *e.g. Artist, ImageDescription, GrayResponseCurve, Copyright* aso. can be used as needed.

The key feature of TIFF with respect to our extension is the possibility to add new tags that describe new properties *without interfering* with the specification of all other formats. One can and, has to, request numbers for these private tags from the TIFF administrator, the Aldus (now Adobe) Developers Desk.

3 PROPOSED EXTENSION

The key reason why we want to extend an existing image format is that we feel little inclination to re–invent the wheel. There are basic design and implementation issues connected with the definition of an entirely new image format, that do not pertain to our topic of interest at all. Also, for TIFF an excellent public domain library exists that can serve as a code basis for extension. The mechanisms of the TIFF library already cover all the (for our purpose) uninteresting groundwork like *e.g.* storing image attributes and raw data in the correct byte order.

What we need is an extension that enables us to store spectral data in TIFF images in a meaningful way. This task is split into three parts:

1. **Defining a new image type**: defining a new value for the `PhotometricInterpretation` tag (and hence a new image type) called *Spectral*.
2. **Modifications to the image header**: we have to store information about the representation and structure of the spectral samples that make up the data.
3. **Storing the actual spectral image data**: storage and retrieval of image data can be done using generic TIFF read/write routines if the SamplesPerPixel and BytesPerSample tags are set correctly, since the actual values of the spectral representations will be arrays of integers or floating point numbers, for which provisions already exist.

Items 1 and 3 require little and no work, respectively. Item 2 alone requires some further consideration. It has to be noted that the tag mechanisms of TIFF allow us to specify all of the following constructs in a way that is consistent with the rest of the format.

Pixel spectrum representation

We define a representation of a spectrum Λ to be a set of n coefficients (of either integer or floating point type) Λ_i of specified basis functions. The choice of coefficient type (int or float) depends on the needs of the user. The weighted sum of these *basis functions* reconstructs the spectrum for a pixel.

The basis functions are the same for each pixel in the image, and they are specified once in the image header. This approach lets us store pixel spectra as sets of float or integer values, for which the normal image storage mechanisms (with n coefficients instead of 3 RGB samples per pixel) can be utilised.

3.1 Basis Functions

For each of the n samples Λ_i in a pixel spectrum, the image header defines a basis function b_i that determines its contribution to the overall pixel spectrum. We propose that these functions be piecewise linear (with possible extensions to higher orders in later revisions of the format).

Why the restriction to linear functions? It is desirable that addition and inner product of the basis functions should be computable exactly; this suggests piecewise polynomial functions as bases. However, the complexity (both programmatic and and with respect to execution time) of these operations rises significantly for the non–linear cases. In conjunction with the fact that one can specify even complicated bases reasonably well with a suitably densely sampled linear function, keeping implementations of the extension simple and efficient weighed more heavily for us than the potential loss of accuracy.

The shape of the basis functions depends entirely on the application at hand; for results of digital image synthesis, usually the simple case of mutually exclusive box basis functions is a good choice, since spectral renderers tend to work with sets of n spectral bands as "colour representation". In [5], Peercy suggested an alternative to box samples for rendering purposes. Results of astronomical imaging, on the other hand, might (*e.g.* in the case of a bandpass filter image) be stored with a single value per pixel, with the transmission function of the filter as basis function. Refer to image 1 for two examples of basis functions.

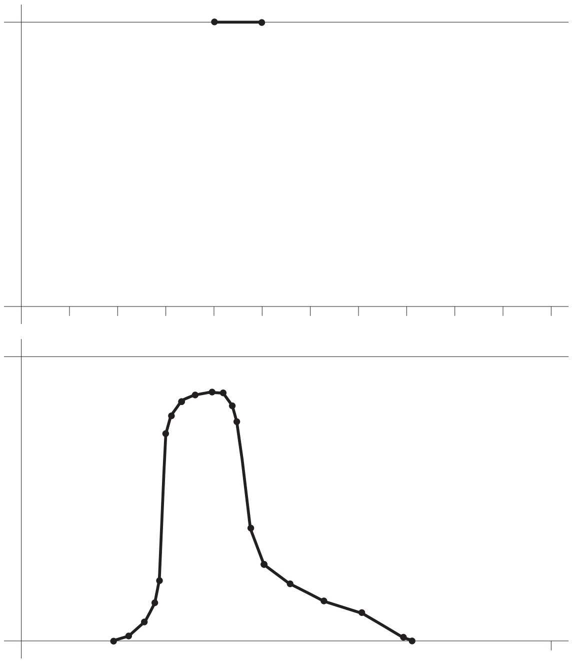

Figure 1: **Basis function examples**. The first function is specified by only two points and is one of mutually exclusive regularly spaced colour bands. The second one is an approximation of an (imaginary) response curve of a sensor, with many unevenly spaced samples. Note that outside the specified range, the basis function is implicitly 0, so one need not specify the slopes of a box filter.

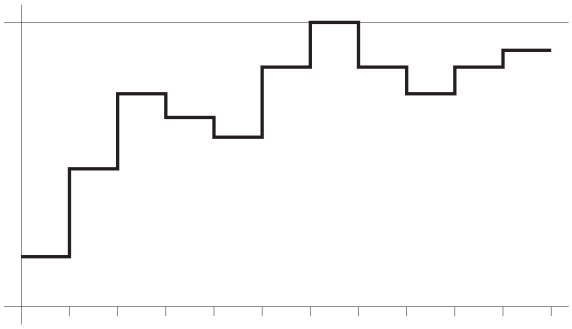

Figure 2: **Reconstructed spectrum**. A box sampled spectrum can be reconstructed from mutually exclusive box samples. Note that the number of samples of the pixel spectrum are *independent* from the properties of the basis functions.

Reconstruction

When decoding the image, the spectrum of each sample is determined by adding all the n basis functions together, weighted by the n spectral samples that are stored for each pixel.

$$\Lambda = \sum_{0}^{n} \Lambda_i b_i$$

The result of this process has then to be converted to the spectral representation of the application that reads the image, which may involve re–sampling.

3.2 Premultiplication

In order to increase the dynamic range of the format, it may be desirable to add a premultiplication factor to each pixel. This amounts to one additional channel, for which no basis function is stored; the encoding can either be linear or logarithmic. The first sample of a spectrum is the premultiplication factor, it now contains $n+1$ samples, and the reconstruction is (for the linear and logarithmic case, respectively) defined as

$$\Lambda_{lin} = \Lambda_0 \sum_{1}^{n+1} \Lambda_i b_i \qquad \Lambda_{log} = e^{\Lambda_0} \sum_{1}^{n+1} \Lambda_i b_i$$

3.3 Compression

Another bonus of using existing TIFF mechanisms is that all in–built lossless compression mechanisms (*e.g.* LZW compression) are directly available for the n-coefficient pixel values. Lossy compression techniques like JPEG may introduce artifacts that influence the resulting visual output of image data in ways which remain to be understood; we refrain from using them at the moment.

3.4 Implementation

In order to validate our design assumptions, we will extend the public domain

TIFF library by Sam Leffler of Silicon Graphics Inc. [3] to meet our specifications. Once this work is completed we will contact the author for experimental inclusion of the new features (in the spirit of the experimental inclusion of Greg Ward–Larson's logLuv format) into the library and Adobe for allocation of definite tags for the extensions (we will use locally assigned temporary values in the meantime).

Information concerning the current state of our implementation, sample images and code can be found on the Web at `http://www.cg.tuwien.ac.at/research/rendering/spectralTIFF`.

4 FUTURE WORK

What remains to be seen are whether efficient lossy compression schemes that provide users with (preferably controllable) minimum information loss, while maintaining an acceptable and predictable compression rate, can be devised for the spectral case. Something that might be worth investigating might be the application of JPEG algorithms to the individual sample planes, although the errors introduced by this might be hard to fathom for the general case, since the error characteristics of JPEG image compression are optimised for tristimulus values and not spectra.

Acknowledgements

The authors would like to thank Greg Ward–Larson of SGI Inc. for his valuable input and fruitful discussions. The authors would also like to thank the anonymous reviewers for their constructive criticism and input. This work has been supported by the "Hochschuljubiläumsfonds der Österreichischen Nationalbank" under project number 7301.

REFERENCES

[1] Adobe Inc. Tiff 6.0 specification, 1992. Available from `ftp://ftp.adobe.com/pub/adobe/devrelations/devtechnotes/pdffiles/tiff6.pdf`.

[2] R. M. Boynton. *Human Color Vision*. Holt Rinehart and Winston, 1979.

[3] S. Leffler. Tiff library version 3.4 beta 037. Available from `ftp://ftp.sgi.com/graphics/tiff/tiff-v3.4beta037.tar.gz`.

[4] S. N. Pattanaik, J. A. Ferweda, M. D. Fairchild, and D. P. Greenberg. A multiscale model of adaptation and spatial vision for realistic image display. In *Computer Graphics Proceedings*, Annual Conference Series, pages 287–298. ACM SIGGRAPH, 1998.

[5] M. S. Peercy. Linear Color Representations for Full Spectral Rendering. In *Computer Graphics Proceedings*, Annual Conference Series, pages 287–298. ACM SIGGRAPH, 1993.

[6] G. Ward-Larson. LogLuv encoding for TIFF images. Available at `http://www.sgi.com/Technology/pixformat/tiffluv.html`.

[7] G. Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. Wiley, 1982.