HDR Image Encodings

03

An important consideration for any digital image is how to store it. This is especially true for HDR images, which record a much wider gamut than standard 24-bit RGB and therefore require an efficient encoding to avoid taking an excess of disk space and network bandwidth. Fortunately, several HDR file encodings and formats have already been developed by the graphics community. A few of these formats have been in use for decades, whereas others have just recently been introduced to the public. Our discussion includes these existing encodings, as well as encodings that lie on the horizon.

An *encoding* is defined as the raw bit representation of a pixel value, whereas a *format* includes whatever wrapper goes around these pixels to compose a complete image. The quality of the results is largely determined by the encoding, rather than the format, making encodings the focus of this chapter. File formats that include some type of "lossy" compression are the exceptions to this rule, and must be considered and evaluated as a whole. Lossy HDR formats are only starting to appear at the time of writing, making any comparisons premature. We simply introduce the basic concepts.

3.1 LDR VERSUS HDR ENCODINGS

There is more than bit depth to defining the difference between HDR and LDR encodings. Specifically, a 24-bit RGB image is usually classified as an *output-referred* standard, because its colors are associated with some target output device. In contrast, most HDR images are *scene-referred*, in that their pixels have a direct relation to

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 2

radiance in some scene, either real or virtual. This is logical, because most output devices are low in dynamic range, whereas most scenes are high in dynamic range. One cannot refer a color encoding to scene values if those values are beyond what can be represented, and thus LDR images are inappropriate for scene-referred data. On the other hand, an HDR encoding *could* be used to hold output-referred data, but there would be little sense in it because scene-referred data can always be mapped to a particular output device, but not the reverse. A scene-referred to output-referred transformation is a one-way street for the simple reason that no output device can reproduce all we see in the real world. This transformation is called *tone mapping*, a topic we return to frequently in this book. (See Chapters 6 through 8.)

Having just introduced this standard term, it is important to realize that *scene-referred* is really a misnomer, because no image format ever attempts to record *all* of the light projected from a scene. In most cases, there would be little sense in recording infrared and ultraviolet wavelengths, or even completely sampling the visible spectrum, because the eye is *trichromatic*. As explained in Chapter 2, this means that it is sufficient to record three color channels in order to reproduce every color visible to a human observer. These may be defined by the CIE XYZ tristimulus space or any equivalent three-primary space (e.g., RGB, YC_BC_R, CIELUV, and so on). Because we are really interested in what people see, as opposed to what is available, it would be better to use the term *human-referred* or *perceptual* for HDR encodings.

Nonetheless, the term *scene-referred* is still preferred, because sometimes we do wish to record more than the eye can see. Example applications for extrasensory data include the following.

- Satellite imagery, in which the different wavelengths may be analyzed and visualized in false color
- Physically-based rendering, in which lighting and texture maps interact to produce new colors in combination
- Scientific and medical visualization, in which (abstract) data is collected and visualized

In such applications, we need to record more than we could see of a scene with our naked eye, and HDR formats are a necessary means in accomplishing this. Further applications of HDR imagery are outlined in the following section.

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 3

3.2 APPLICATIONS OF HDR IMAGES

The demands placed on an HDR encoding vary substantially from one application to another. In an Internet application, file size might be the deciding factor. In an image database, it might be decoding efficiency. In an image-compositing system, accuracy might be most critical. The following describe some of the many applications for HDR, along with a discussion of their requirements.

Physically-based rendering (global illumination): Perhaps the first application to use HDR images, physically-based rendering, and lighting simulation programs must store the absolute radiometric quantities for further analysis and for perceptually based tone mapping [142,143]. In some cases, it is important to record more than what is visible to the human eye, as interactions between source and surface spectra multiply together. Additional accuracy may also be required of the encoding to avoid accumulated errors, and alpha and depth channels may also be desirable. A wide dynamic range is necessary for *image-based lighting*, especially in environments that include daylight. (See Chapter 9.)

Remote sensing: As mentioned in the previous section, satellite imagery often contains much more than is visible to the naked eye [75]. HDR is important for these images, as is multispectral recording and the ability to annotate using image metadata. Accuracy requirements may vary with the type of data being recorded, and flexibility is the key.

Digital photography: Camera makers are already heading in the direction of scene-referred data with their various RAW formats, but these are cumbersome and inconvenient compared to the standard encodings described in this chapter.[1] It is only a matter of time before cameras that directly write HDR images begin to appear on the market. File size is clearly critical to this application. Software compatibility is also important, although this aspect is largely neglected by camera RAW formats. Adobe's Digital Negative specification and software works toward alleviating this problem [3].

1 Each camera manufacturer employs its own proprietary format, which is usually not compatible with other manufacturers' RAW formats or even with popular image-editing software. These formats are collectively called RAW, because the camera's firmware applies only minimal processing to the data that are read from the sensor.

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 4

Image editing: Image-editing applications with support for HDR image data are now available. Photoshop CS 2 incorporates reading and writing of 32-bit pixel data, as does Photogenics (*www.idruna.com*), and a free open-source application called Cinepaint (*www.cinepaint.org*). A vast number of image-editing operations are possible on HDR data that are either difficult or impossible using standard output-referred data, such as adding and subtracting pixels without running under or over range, extreme color and contrast changes, and white balancing that works. Accuracy will be an important requirement here, again to avoid accumulating errors, but users will also expect support for all existing HDR image formats.

Digital cinema (and video): Digital cinema is an important and fast-moving application for HDR imagery. Currently, the trend is heading in the direction of a medium-dynamic-range output-referred standard for digital film distribution. Film editing and production, however, will be done in some HDR format that is either scene-referred or has some intermediate reference, such as movie film stock. For intermediate work, resolution and color accuracy are critical, but file size is also a consideration in that there are over 200,000 frames in a two-hour movie, and each of these may be composited from dozens of intermediate layers. Rendering a digital movie in HDR also permits HDR projection. (See Chapter 5, on display devices.) Looking further ahead, an exciting possibility is that HDR video may eventually reach the small screen. At least one HDR MPEG extension has already been proposed, which we discuss at the end of the next section.

Virtual reality: Many web experiences require the efficient transmission of images, which are usually encoded as JPEG or some other lossy representation. In cases in which a user is attempting to view or move around a virtual space, image exposure is often a problem. If there were a version of QuicktimeVR that worked in HDR, these problems could be solved. Establishing standards for lossy HDR compression is therefore a high priority for virtual reality on the Web.

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 5

Format	Encoding(s)	Compression	Metadata	Support/Licensing
HDR	RGBE	Run-length	Calibration, color space, +user-defined	Open source software (*Radiance*)
	XYZE	Run-length		Quick implementation
TIFF	IEEE RGB	None	Calibration, color space, +registered, +user-defined	Public domain library (*libtiff*)
	LogLuv24	None		
	LogLuv32	Run-length		
EXR	Half RGB	Wavelet, ZIP	Calibration, color space, +windowing, +user-defined	Open source library (*OpenEXR*)

TABLE 3.1 *Established HDR image file formats.*

Each of these applications, and HDR applications not yet conceived, carries its own particular requirements for image storage. The following section lists and compares the established HDR formats and discusses upcoming formats.

3.3 HDR IMAGE FORMATS

Table 3.1 lists three existing HDR image formats and compares some of their key attributes. The encodings within these formats are broken out in Table 3.2, where the basic parameters are given. In some cases, one format may support multiple encodings (e.g., TIFF). In other cases, we list encodings that have not yet appeared in any format but are the subject of published standards (e.g., scRGB). The standard 24-bit RGB (sRGB) encoding is also included in Table 3.2, as a point of comparison.

Encoding	Color Space	Bits/pixel	Dynamic Range ($\log_{10}$)	Relative Step
sRGB	RGB in [0,1] range	24	1.6 orders	Variable
RGBE	Positive RGB	32	76 orders	1.0%
XYZE	(CIE) XYZ	32	76 orders	1.0%
IEEE RGB	RGB	96	79 orders	0.000003%
LogLuv24	Log Y + (u',v')	24	4.8 orders	1.1%
LogLuv32	Log Y + (u',v')	32	38 orders	0.3%
Half RGB	RGB	48	10.7 orders	0.1%
scRGB48	RGB	48	3.5 orders	Variable
scRGB-nl	RGB	36	3.2 orders	Variable
scYCC-nl	YC_BC_R	36	3.2 orders	Variable

TABLE 3.2 *HDR pixel encodings, in order of introduction.*

Formats based on logarithmic encodings, LogLuv24 and LogLuv32, maintain a constant relative error over their entire range.[2] For the most part, the floating-point encodings RGBE, XYZE, IEEE RGB, and Half RGB also maintain a constant relative error. The dynamic ranges quoted for the encodings sRGB, scRGB48, scRGB-nl, and scYCC-nl are based on the point at which their relative steps pass 5%. Above 5%, adjacent steps in the encoding are easily distinguished. If one were to view an sRGB image on an HDR display, regions below 0.025 of the maximum would exhibit visible banding, similar to that shown in Figure 3.1.[3] For luminance quantization to be completely invisible, the relative step size must be held under 1% [149]. This

2 Relative step size is the difference between adjacent values divided by the value. The relative error is generally held to half the relative step size, and is the difference between the correct value and the representation divided by the correct value.

3 Thus, the dynamic range of sRGB is 0.025:1, which is the same ratio as $1{:}10^{1.6}$. In Table 3.2, we just report the number of orders (powers of 10).

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 7

FIGURE 3.1 *Banding due to quantization at the 5% level.*

is the goal of most HDR encodings, and some have relative step sizes considerably below this level. Pixel encodings with variable quantization steps are difficult to characterize in terms of their maximum dynamic range and are ill suited for HDR applications in which the display brightness scaling is not predetermined.

3.3.1 THE HDR FORMAT

The HDR format, originally known as the *Radiance* picture format (*.hdr*, *.pic*), was first introduced as part of the Radiance lighting simulation and rendering system in 1989 [144], and has since found widespread use in the graphics community, particularly for HDR photography and image-based lighting [17,18]. (See Chapters 4 and 9.) The file wrapper consists of a short ASCII header, followed by a resolution string that defines the image size and orientation, followed by the run-length en-

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 8

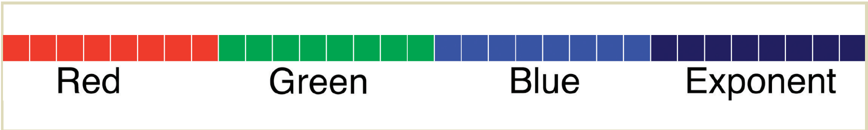

FIGURE 3.2 *Bit breakdown for the 32-bit/pixel RGBE (and XYZE) encodings.*

coded pixel data. The pixel data comes in two flavors: a 4-byte RGBE encoding [138] and a CIE variant, XYZE. The bit breakdown is shown in Figure 3.2.

The RGBE components R_M, G_M, and B_M are converted from the scene-referred color (R_W, G_W, B_W) via the following formula.

$$E = \left\lceil \log_2 \left(\max \left(R_W, G_W, B_W\right)\right) + 128 \right\rceil$$

$$R_M = \left\lfloor \frac{256\, R_W}{2^{E-128}} \right\rfloor$$

$$G_M = \left\lfloor \frac{256\, G_W}{2^{E-128}} \right\rfloor$$

$$B_M = \left\lfloor \frac{256\, B_W}{2^{E-128}} \right\rfloor$$

There is also a special case for an input in which $\max(R_W, G_W, B_W)$ is less than 10^{-38}, which is written out as $(0, 0, 0, 0)$. This gets translated to $(0, 0, 0)$ on the reverse conversion. The reverse conversion for the normal case is as follows.

$$R_W = \frac{R_M + 0.5}{256}\, 2^{E-128}$$

$$G_W = \frac{G_M + 0.5}{256}\, 2^{E-128}$$

$$B_W = \frac{B_M + 0.5}{256}\, 2^{E-128}$$

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 9

The conversions for XYZE are precisely the same, with the exception that CIE X, Y, and Z are substituted for R, G, and B, respectively. Because the encoding does not support negative values, using XYZE instead of RGBE extends the range to cover the entire visible gamut. (See Chapter 2 for details on the CIE XYZ space.) The dynamic range for these encodings is quite large (over 76 orders of magnitude), and the accuracy is sufficient for most applications. Run-length encoding achieves an average of 25% compression (1:1.3), making the image files about as big as uncompressed 24-bit RGB.

3.3.2 THE TIFF FLOAT AND LOGLUV FORMATS

For over a decade, the Tagged Image File Format (.tif, .tiff) has included a 32-bit/component IEEE floating-point RGB encoding [2]. This standard encoding is in some ways the ultimate in HDR image representations, covering nearly 79 orders of magnitude in miniscule steps. The flip side to this is that it takes up more space than any other HDR encoding — over three times the space of the Radiance format (described in the preceding section). The TIFF library does not even attempt to compress this encoding, because floating-point data generally do not compress very well. Where one might get 30% compression from run-length encoding of RGBE data, 10% is the most one can hope for using advanced entropy compression (e.g., ZIP) on the same data stored as IEEE floats. This is because the last 12 bits or more of each 24-bit mantissa will contain random noise from whatever camera or global illumination renderer generated them. There simply are no image sources with 7 decimal digits of accuracy, unless they are completely synthetic (e.g., a smooth gradient produced by a pattern generator).

Nevertheless, 96-bit/pixel RGB floats have a place, and that is as a lossless intermediate representation. TIFF float is the perfect encoding for quickly writing out the content of a floating-point frame buffer and reading it later without loss. Similarly, raw floats are a suitable means of sending image data to a compositor over a high-bandwidth local connection. They can also serve as a "gold standard" for evaluating different HDR representations, as shown in Section 3.4. However, most programmers and users are looking for a more compact representation, and within TIFF there are two: 24-bit and 32-bit LogLuv.

The LogLuv encoding was introduced as a perceptually based color encoding for scene-referred images [71]. Like the IEEE float encoding just described,

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 10

LogLuv is implemented as part of the popular public domain TIFF library. (See *www.remotesensing.org/libtiff*. Appropriate examples are also included on the companion DVD-ROM.) The concept is the same for the 24-bit and 32-bit/pixel variants, but they achieve a different range and accuracy. In both cases, the scene-referred data is converted to separate luminance (Y) and CIE (u, v) channels. (Review Chapter 2 for the conversions between CIE and RGB color spaces.) The logarithm of luminance is then taken, and the result is quantized into a specific range, which is different for the two encodings, although both reserve the 0 code for $Y = 0$ (black). In the case of the 24-bit encoding, only 10 bits are available for the log luminance value. Quantization and recovery are computed as follows.

$$L_{10} = \left\lfloor 64 \left(\log_2 Y_{\mathrm{W}} + 12\right)\right\rfloor$$

$$Y_{\mathrm{W}} = 2^{\dfrac{L_{10} + 0.5}{64} - 12}$$

This encoding covers a world luminance (Y_{W}) range of 0.00025:15.9, or 4.8 orders of magnitude in uniform (1.1%) steps. In cases in which the world luminance is skewed above or below this range, we can divide the scene luminances by a constant and store this calibration factor in the TIFF STONITS tag.[4] When decoding the file, applications that care about absolute values consult this tag and multiply the extracted luminances accordingly.

The remaining 14 bits of the 24-bit LogLuv encoding are used to represent chromaticity, based on a lookup of CIE (u, v) values, as diagrammed in the lower portion of Figure 3.3. A zero lookup value corresponds to the smallest v in the visible gamut, and subsequent table entries are built up left to right, then bottom to top, in the diagram. The uniform step size for u and v is 0.0035, which is just large enough to cover the entire visible gamut in 2^{14} codes. The idea is that employing a *perceptually uniform* color space, in which equal steps correspond to equal differences in color, keeps quantization errors below the visible threshold. Unfortunately, both the (u, v) step size and the luminance step size for the 24-bit encoding are slightly larger than the ideal. This quantization was chosen to cover the full gamut over a reasonable luminance range in a 24-bit file, and the TIFF library applies dithering

4 STONITS stands for "sample-to-nits." Recall from Chapter 2 that the term *nits* is shorthand for candelas/meter^2.

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 11

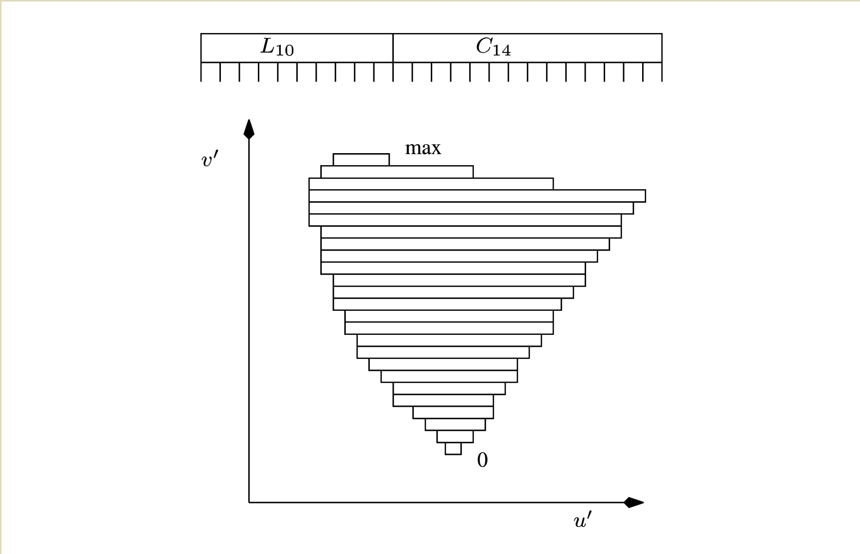

FIGURE 3.3 *Bit breakdown for 24-bit LogLuv encoding and method used for CIE (u, v) lookup.*

by default to hide steps where they might otherwise be visible.[5] Because there is no compression for the 24-bit LogLuv encoding, there is no penalty in dithering.

The 32-bit LogLuv TIFF encoding is similar to the 24-bit LogLuv variant, but allows a greater range and precision. The conversion for luminance is as follows.

$$L_15 = \lfloor 256 \left(\log_2 Y_W + 64 \right) \rfloor$$

$$Y_W = 2^{\frac{L_{15} + 0.5}{256} - 64}$$

5 Dithering is accomplished during encoding by adding a random variable in the (–0.5, 0.5) range immediately before integer truncation.

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 12

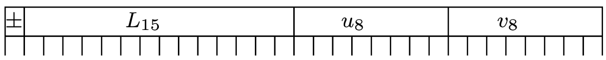

FIGURE 3.4 *Bit breakdown for 32-bit LogLuv encoding. Upper- and lower-order bytes are separated per scan line during run-length compression to reduce file size.*

This 15-bit encoding of luminance covers a range of $5.5 \times 10^{-20} : 1.8 \times 10^{19}$, or 38 orders of magnitude in 0.3% steps. The bit breakdown for this encoding is shown in Figure 3.4. The leftmost bit indicates the sign of luminance, permitting negative values to be represented.[6] The CIE u and v coordinates are encoded in 8 bits each, which allows for sufficiently small step sizes without requiring a lookup. The conversion for chromaticity is simply

$$
\begin{aligned}
u_8 &= \lfloor 410\, u' \rfloor \\
v_8 &= \lfloor 410\, v' \rfloor \\
u' &= \frac{u_8 + 0.5}{410} \\
v' &= \frac{v_8 + 0.5}{410}.
\end{aligned}
$$

Again, dithering may be applied by the TIFF library to avoid any evidence of quantization, but it is not used for 32-bit LogLuv by default because the step sizes are below the visible threshold and run-length compression would be adversely affected. The compression achieved by the library for undithered output is 10 to 70%. Average compression is 40% (1:1.7).

Most applications will never see the actual encoded LogLuv pixel values, in that the TIFF library provides conversion to and from floating-point XYZ scan lines. However, it is possible through the use of lookup on the raw encoding to combine

6 This is useful for certain image-processing operations, such as compositing and error visualizations.

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 13

the reading of a LogLuv file with a global tone-mapping operator, thus avoiding floating-point calculations and providing for rapid display [70]. The TIFF library provides raw data access for this purpose.

3.3.3 THE OPENEXR FORMAT

The EXtended Range format (*.exr*) was made available as an open-source C++ library in 2002 by Industrial Light and Magic (see *www.openexr.com*) [62]. It is based on a 16-bit *half* floating-point type, similar to IEEE float with fewer bits. Each RGB pixel occupies a total of 48 bits, broken into 16-bit words (as shown in Figure 3.5). The Half data type is also referred to as *S5E10*, for "sign, five exponent, ten mantissa." The OpenEXR library also supports full 32-bit/channel (96-bit/pixel) floats and a new 24-bit/channel (72-bit/pixel) float type introduced by Pixar. We have already discussed the 32-bit/channel IEEE representation in the context of the TIFF format, and we have no further information on the 24-bit/channel type at this time. We will therefore restrict our discussion to the 16-bit/channel Half encoding.

The formula for converting from an encoded Half value follows. Here, S is the sign bit, E the exponent (0 to 31), and M the mantissa (0 to 1,023).

$$h = \begin{cases} (-1)^S \, 2^{E-15} \left(1 + \dfrac{M}{1024}\right) & 1 \le E \le 30 \\ (-1)^S \, 2^{-14} \, \dfrac{M}{1024} & E = 30 \end{cases}$$

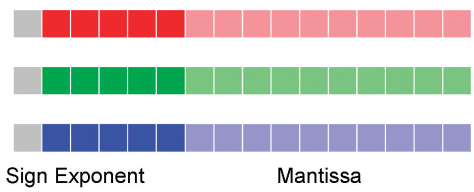

FIGURE 3.5 *Bit breakdown for the OpenEXR Half pixel encoding.*

If the exponent E is 31, the value is infinity if $M = 0$ and NaN (not a number) otherwise. Zero is represented by all zero bits. The largest representable value in this encoding is 65,504, and the smallest normalized (i.e., full accuracy) value is 0.000061. This basic dynamic range of 9 orders is enhanced by the "denormalized" values below 0.000061, which have a relative error below 5% down to 0.000001, for a total dynamic range of 10.7 orders of magnitude. Over most of this range, the quantization step size is under 0.1%, which is far below the visible threshold. This permits extensive image manipulations before artifacts become evident, which is one of the principal strengths of this encoding. Another advantage of the Half encoding is that the selfsame 16-bit representation is specified in NVidia's Cg language [79]. This will ultimately make transfer to and from graphics hardware straightforward and is promising for future hardware standardization as well.

The OpenEXR library contains C++ classes for reading and writing EXR image files, with support for lossless compression, tiling, and mip mapping. Compression is accomplished using the ZIP deflate library as one alternative, or Industrial Light and Magic's (ILM) more efficient PIZ lossless wavelet compression. From our experiments, PIZ achieves a 60% reduction on average compared to uncompressed 48-bit/pixel RGB. OpenEXR also supports arbitrary data channels, including alpha, depth, and user-defined image data. Similar to the TIFF format, standard attributes are provided for color space, luminance calibration, pixel density, capture date, camera settings, and so on. User-defined attributes are also supported, and unique to OpenEXR is the notion of a "display window" to indicate the active region of an image. This is particularly useful for special effects compositing, wherein the notion of what is on-screen and what is off-screen may evolve over the course of a project.

3.3.4 OTHER ENCODINGS

There are a few other encodings that have been used or are being used to represent medium-dynamic-range image data (i.e., between 2 and 4 orders of magnitude). The first is the Pixar log encoding, which is available in the standard TIFF library along with LogLuv and IEEE floating point. This 33-bit/pixel encoding assigns each of 11 bits to red, green, and blue using a logarithmic mapping designed to fit the range of movie film. The implementation covers about 3.8 orders of magnitude in 0.4% steps, making it ideal for film work but marginal for HDR work. Few people have used this encoding outside of Pixar, and they have themselves moved to a

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 15

higher-precision format. Another image standard that is even more specific to film is the Cineon format, which usually records logarithmic density in 10 bits/channel over a 2.0 range (*www.cineon.com*). Although these 2.0 orders of magnitude may correspond to slightly more range once the film response curve has been applied, it does not qualify as an HDR encoding, and it is not scene-referred. Mechanically, the Cineon format will handle greater bit depths, but the meaning of such an extension has never been defined.

More recently, the IEC has published a standard that defines the scRGB48, scRGB-nl, and scYCC-nl encodings, listed in Table 3.2 [56]. As shown in Table 3.2, these encodings also encompass a relatively small dynamic range, and we are not aware of any software product or file format that currently uses them. We will therefore leave this standard out of this discussion, although an analysis may be found at *www.anyhere.com/gward/hdrenc* as well as on the companion DVD-ROM.

3.3.5 EMERGING "LOSSY" HDR FORMATS

All of the HDR image formats we have discussed so far, and indeed all of the HDR standards introduced to date, are *lossless* insofar as once the original scene values have been converted into the encoding, no *further* loss takes place during storage or subsequent retrieval. This is a desirable quality in many contexts, especially when an image is expected to go through multiple storage and retrieval steps (with possible manipulations) before reaching its final state. However, there are some applications for which a *lossy* format is preferred, particularly when the storage costs are onerous or further editing operations are anticipated or desired. Two such applications lie just around the corner, and they will need suitable lossy standards to meet their needs: HDR photography and HDR video. At the time of writing, two lossy encoding methods have been introduced for HDR: one for still images and one for video.

HDR Disguised as JPEG: The Sub-band Encoding Method Ward and Simmons developed a still image format that is backward compatible with the 8-bit JPEG standard [136]. This *sub-band encoding* method stores a tone-mapped image as a JPEG/JFIF file, packing restorative information in a separate 64-Kbyte marker. Naïve applications ignore this marker as extraneous, but newer software can recover the

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 16

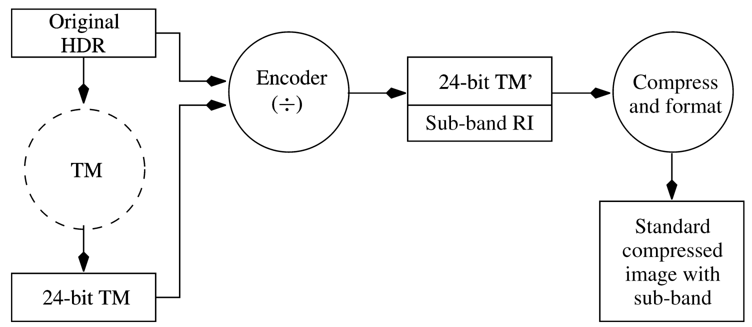

FIGURE 3.6 *The sub-band encoding pipeline. (Reprinted from [136].)*

full HDR data by recombining the encoded information with the tone-mapped image. In other words, 64 Kbytes is enough to create a scene-referred original from an output-referred JPEG. Since most JPEG images produced by today's digital cameras are over 1 Mbyte, this is only a 5% increase in file size. By comparison, the most compact lossless HDR format requires 16 times as much storage space as JPEG.

Figure 3.6 shows the encoding pipeline, including the to-be-specified tone-mapping operator (*TM*). In principle, any tone-mapping operator can work, but we found that the photographic operator [109] (see Section 7.3.6) and the bilateral filter operator [23] (see Section 8.1.2) worked the best with this method. Once the tone-mapped image is derived, its pixels are divided into the original to obtain a grayscale "ratio image," which is then compressed and incorporated in the JPEG file as a sub-band marker.

Figure 3.7 shows an HDR image of a church that has been decomposed into a tone-mapped image and the corresponding ratio image. The ratio image is down-sampled and log encoded before being passed to the JPEG compressor to squeeze it into 64 Kbytes. This size is the upper limit for a JFIF marker, and the ratio image size and compression quality are optimized to fit within a single marker, although multiple markers might be used in some cases. Loss of detail is prevented either by

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 17

FIGURE 3.7 *An HDR image of a church divided into a tone-mapped version and the downsampled ratio image that is stored as a sub-band.*

enhancing edges in the tone-mapped image to compensate for the downsampled ratio image or by synthesizing high frequencies in the ratio image during upsampling, depending on the application and user preference. The dynamic range of the format is unrestricted in the sense that the log encoding for the ratio image is optimized to cover the input range with the smallest step size possible.

Figure 3.8 illustrates the decode process. A naïve application extracts the tone-mapped pixels and treats them as a standard output-referred image. An HDR application, however, recognizes the sub-band and decompresses both this ratio image and the tone-mapped version, multiplying them together to recover the original scene-referred data.

Two clear benefits arise from this strategy. First, a tone-mapped version of the HDR image is immediately available—for naïve applications that cannot handle

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 18

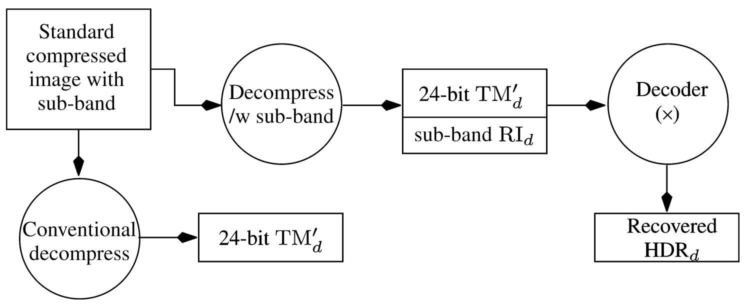

FIGURE 3.8 *The sub-band decoding pipeline. The lower left shows the backward-compatible path for naïve applications, and the upper right shows the path for an HDR decoder.*

anything more and for HDR applications that may be able to perform their own tone-mapping given time but wish to provide the user with immediate feedback. Second, by making the format *backward compatible* with the most commonly used image type for digital cameras an important barrier to adoption has been removed for the consumer, and hence for camera manufacturers.

Ward and Simmons tested the sub-band encoding method on 15 different HDR images, passing them through a single encoding-decoding cycle and comparing them to the original using Daly's visible differences predictor (VDP) [15]. With the exception of a single problematic image, VDP showed that fewer than 0.2% of the pixels had visible differences at the maximum quality setting in any one image, which went up to an average of 0.6% at a 90% quality setting.

The problem image was a huge (5,462 × 4,436) Radiance rendering of a car in a tunnel, with stochastic samples whose local variance spanned about 5 orders of magnitude. This proved too much for the downsampled ratio image to cope with, leading to the conclusion that for some large renderings it may be necessary to break the 64-Kbyte barrier and record multiple markers to improve accuracy. In most cases, however, the authors found that 64 Kbytes was ample for the sub-

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 19

band, and the overall image size was therefore comparable to a JPEG file of the same resolution. This makes a strong argument for lossy HDR compression when file size is critical. In addition to digital photography, Internet sharing of HDR imagery is a prime candidate for such a format.

An HDR Extension to MPEG Mantiuk et al. have introduced an HDR encoding method built on the open-source XviD library and the MPEG-4 video standard [78]. The diagram in Figure 3.9 shows the standard MPEG compression pipeline in black. Extensions to this pipeline for HDR are shown in blue. The modified MPEG encoding pipeline proceeds as follows for each frame.

1. A 32-bit/channel XYZ is taken on input rather than 8-bit/channel RGB.
2. XYZ is converted into CIE (u, v) coordinates of 8 bits each and an 11-bit perceptual luminance encoding, L_{p}.
3. This 11/8/8 bit encoding is passed through a modified discrete cosine transform (DCT), which extracts high-contrast edges from the luminance channel for separate run-length encoding.
4. The DCT blocks are quantized using a modified table and passed through a variable-length coder.
5. The edge blocks are joined with the DCT blocks in an HDR-MPEG bit stream.

The decoding process is essentially the reverse of this, recombining the edge blocks at the DCT reconstruction stage to get back L_{p} (u, v) color values for each pixel. These may then be decoded further, into CIE XYZ floats, or passed more efficiently through appropriate lookup tables for real-time display (e.g., tone mapping).

One of the key optimizations in this technique is the observation that the entire visible range of luminances, 12 orders of magnitude, can be represented in only 11 bits using invisible quantization steps. By taking advantage of the human contrast versus intensity (CVI) curve, it is possible to find a varying step size from the minimum perceivable luminance to the maximum, avoiding wasted codes in the darker regions (where the eye is less sensitive) [35].[7] This implicitly assumes that the encoded information has some reasonable calibration, and that the ultimate

7 The CVI curve is equal to the threshold versus intensity (TVI) curve divided by adaption luminance. The TVI function is discussed in detail in Chapter 6.

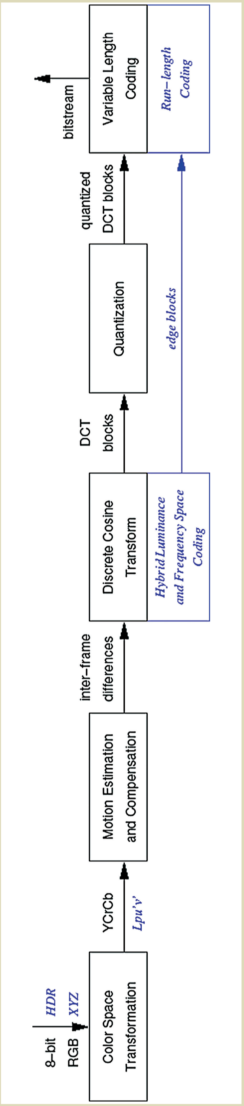

FIGURE 3.9 MPEG encoding pipeline (black) with HDR extensions (blue). (Reprinted from [78].)

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 21

consumer is a human observer as opposed to a rendering algorithm in need of HDR input. These are defensible assumptions for compressed video. Even if the absolute calibration of the incoming luminances is unknown, a suitable multiplier could be found to take the maximum input value to the maximum L_p representation, thus avoiding clipping. In that an observer would not be able to see more than 12 orders of magnitude below any safe maximum, such a scaling would provide full input visibility. The only exception to this is if the input contains an unreasonably bright source in the field of view, such as the sun.

Figure 3.10 shows the human CVI curve compared to the quantization errors for the encodings we have discussed. The blue line shows the error associated with Mantiuk et al.'s L_p encoding, which mirrors human contrast sensitivity while staying comfortably below the visible threshold. When HDR video displays enter the

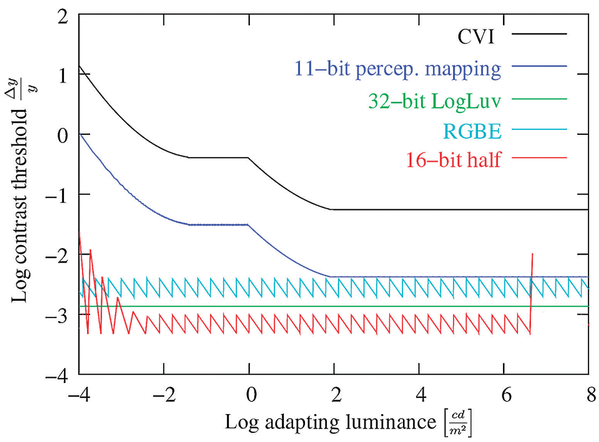

FIGURE 3.10 *Quantization error of HDR encodings as a function of adapting luminance. (The sawtooth form of the floating point encodings has been exaggerated for clarity.)*

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 22

market, an extended video standard will be needed, and this research is an important step toward such a standard.

3.4 HDR ENCODING COMPARISON

To compare HDR image formats, we need a driving application. Without an application, there are no criteria, just speculation. The application determines the context and sets the requirements.

For our example comparison, we have chosen a central application for HDR: scene-referred image archival. Specifically, we wish to store HDR images from any source to be displayed at some future date on an unknown device at the highest quality it supports. Assuming this display has not been invented, there is no basis for writing to an output-referred color space, and hence a scene-referred encoding is the only logical representation. Thus, the need for HDR.

A reasonable assumption is that a full spectral representation is not necessary, because humans perceive only three color dimensions. (Refer to Chapter 2.) Further, we assume that it is not necessary to record more than the visible gamut, although it is not safe to assume we can store less. Likewise, the quantization steps must be kept below the visible threshold, but because we plan no further manipulations prior to display, extra accuracy only means extra storage. The requirements for image archiving are therefore as follows.

- Cover the visible gamut with a tristimulus color space (XYZ, RGB, and so on)
- Cover the full range of perceivable luminances
- Have quantization steps below the visible threshold at all levels

Furthermore, it is desirable for the format to:

- Minimize storage costs (Mbytes/Mpixel)
- Encode and decode quickly

Considering the previous requirements list, we can rule out the use of the RGBE encoding (which does not cover the visible gamut) and the 24-bit LogLuv encoding, which does not cover the full luminance range. This leaves us with the XYZE encoding (*.hdr*), the IEEE floating-point and 32-bit LogLuv encodings (*.tif*), and the Half encoding (*.exr*). Of these, the IEEE float representation will clearly lose in terms of storage costs, but the remaining choices merit serious consideration. These are as follows.

- The 32-bit Radiance XYZE encoding
- The 32-bit LogLuv encoding
- The 48-bit OpenEXR Half encoding

On the surface, it may appear that the XYZE and LogLuv encodings have a slight edge in terms of storage costs, but the OpenEXR format includes a superior compression engine. In addition, the extra bits in the Half encoding may be worthwhile for some archiving applications that need or desire accuracy beyond normal human perception. To evaluate the candidate formats, the following test was conducted on a series of IEEE floating-point images, some captured and some rendered.

1. The data is encoded into the test format, noting the CPU time and disk space requirements.
2. The data is then decoded, noting the CPU time required.
3. The decoded data can then be compared to the original using CIE ΔE^* 1994 perceptual color difference metric.

CIE ΔE^* 1994 is an updated version of the perceptual difference metric presented in Chapter 2 [80]. Using this metric, an encoded pixel color can be compared to the original source pixel by computing the visible difference. However, we must first modify the difference metric to consider local adaptation in the context of HDR imagery. To do this, the brightest Y value within a fixed region about the current pixel is found, and this value is used as the reference white. This simulates the effect of a viewer adapting their vision (or display) locally, as we would expect them to do with an HDR image. The only question is how large a region to use, and for this a reasonable choice is to use a radius of 50 pixels, as this tends to be the size of interesting objects in our test image set.

Among the test images, we included a synthetic pattern that covered the full visible gamut and dynamic range with sufficient density to sample quantization errors at all levels. This pattern, a spiral slice through the visible gamut from 0.01

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 24

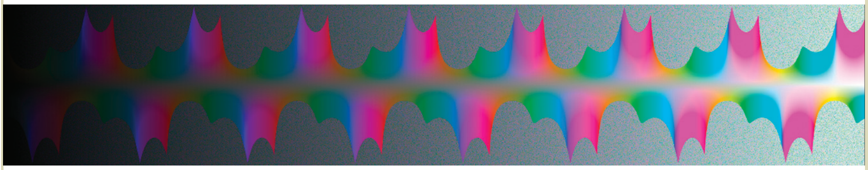

FIGURE 3.11 *The gamut test pattern, spanning eight orders of magnitude. This image was tone mapped with the histogram adjustment technique for the purpose of reproduction in this book (see Section 7.2.8).*

to 1,000,000 cd/m^2, is shown in Figure 3.11. (This image is included on the companion DVD-ROM as an IEEE floating-point TIFF.) Each peak represents one revolution through the visible color gamut, and each revolution spans one decade (factor of 10) in luminance. The gray-looking regions above and below the slice actually contain random colors at each luminance level, which provide an even more thorough testing of the total space. Obviously, tone mapping has been used to severely compress the original dynamic range and colors in order to print this otherwise undisplayable image.

Figure 3.12 shows the CIE ΔE^* encoding errors associated with a 24-bit sRGB file, demonstrating how ill suited LDR image formats are for archiving real-world colors. Only a narrow region covering under two orders of magnitude with an incomplete gamut is below the visible difference threshold (2.0 in ΔE^*). In contrast, the three HDR encodings we have chosen for this application do quite well on this test pattern, as shown in Figure 3.13. Errors are held below the visible threshold in each encoding over all eight orders, except for a few highly saturated colors near the top of the EXR Half range. The average ΔE^* values for Radiance XYZE, 32-bit LogLuv, and EXR Half were 0.2, 0.3, and 0.06, respectively.

Figure 3.14 shows the two encodings we rejected on the basis that they did not cover the full dynamic range and gamut, and indeed we see they do not. As expected, the Radiance RGBE encoding is unable to represent highly saturated colors, although it easily spans the dynamic range. The 24-bit LogLuv encoding, on the

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 25

other hand, covers the visible gamut, but only spans 4.8 orders of magnitude. Although they may not be well suited to our proposed application, there are other applications to which these encodings are perfectly suited. In some applications, for example, there is no need to represent colors outside those that can be displayed on an RGB monitor. Radiance RGBE has slightly better resolution than XYZE in the same number of bits and does not require color transformations. For other applications, 4.8 orders of magnitude may be sufficient because they only need to cover the human *simultaneous* range; that is, the range over which an observer can comfortably adapt without the use of blinders. Because 24-bit LogLuv covers the full gamut in this range as well, applications that need to fit the pixel data into a 24-bit buffer for historical reasons may prefer it to the 32-bit alternatives. It was used in a proprietary hardware application in which a prepared 24-bit lookup translates scene-referred colors to device space via a 16-million entry table. Such a lookup would be impossible with a 32-bit encoding, which would require 4 billion entries.

In addition to color gamut and dynamic range, we are also interested in the statistical behavior of these formats on real images, especially with regard to file size and compression times. Figure 3.15 shows a test set of 34 images. Of these, 19 are HDR photographs of real scenes and 15 are computer generated, and sizes range from 0.2 to 36 Mpixels, with 2.4 Mpixels being average.

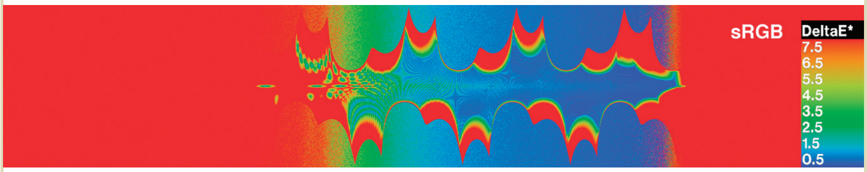

FIGURE 3.12 *This false color plot shows the visible error behavior of the 24-bit sRGB encoding on the test pattern shown in Figure 3.11. (CIE ΔE^* values above 2 are potentially visible, and above 5 are evident.)*

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 26

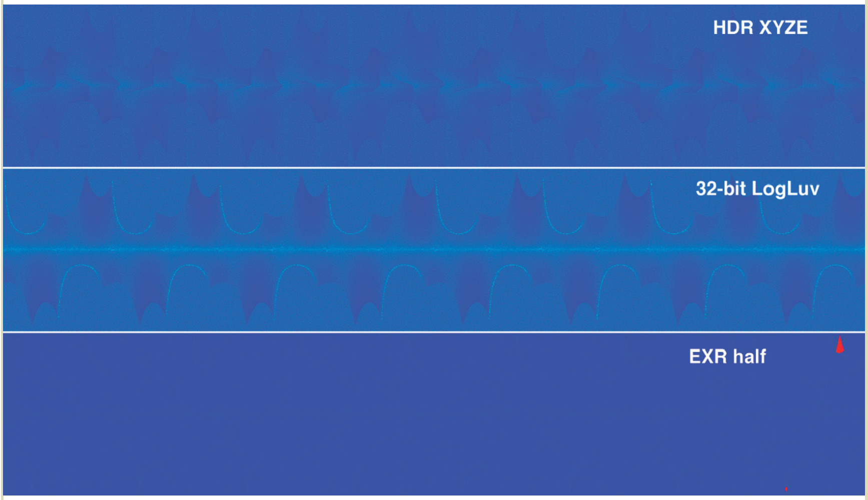

FIGURE 3.13 *Error levels for the chosen HDR encodings applied to the gamut test pattern from Figure 3.11, using the same scale as shown in Figure 3.12.*

Figure 3.16 charts the read/write performance and file size efficiency for each of the three selected formats. This figure shows that the Radiance HDR format has the fastest I/O performance, but creates larger files. The OpenEXR library is considerably slower with its I/O, but creates smaller files than Radiance, despite the 48 bits of the Half pixel encoding. The 32-bit LogLuv TIFF format has intermediate I/O performance, and produces the smallest files.

The average CIE ΔE^* error performance of the three encodings is the same over the entire image set as we reported for the gamut test alone, with the following exceptions. One of the test images from ILM, "Desk," contained pixel values that are completely outside the visible gamut and could not be reproduced with either Radiance XYZE or 32-bit LogLuv. Because we do not expect a need for archiving

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 27

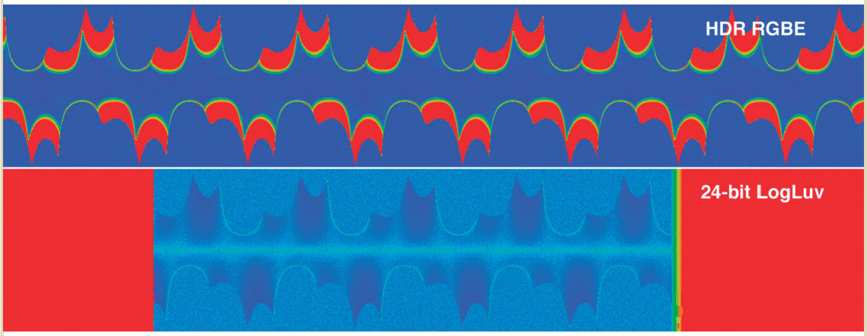

FIGURE 3.14 *The CIE ΔE^* associated with the gamut test pattern in Figure 3.11 for the Radiance RGBE and 24-bit LogLuv encodings, using the same scale as Figure 3.12.*

colors that cannot be seen or reproduced, this should not count against these two encodings for this application. A few of the renderings had pixels outside the representable dynamic range of EXR's Half data type. In those cases, we did not resort to scaling the images to fit within the 10^{-6}:10^4 range as we might have.

In summary, we found that the XYZE and LogLuv encodings are restricted to the visible gamut, and the Half encoding has a slightly smaller dynamic range. Neither of these considerations is particularly bothersome, and thus we conclude that all three encodings perform well for HDR image archiving.

3.5 CONCLUSIONS

The principal benefit of using scene-referred HDR images is their independence from the display process. A properly designed HDR format covers the full range and sensitivity of human vision, and is thus prepared for *any* future display technology intended for humans. Many HDR formats offer the further benefit, through additional range and accuracy, of permitting complex image operations without

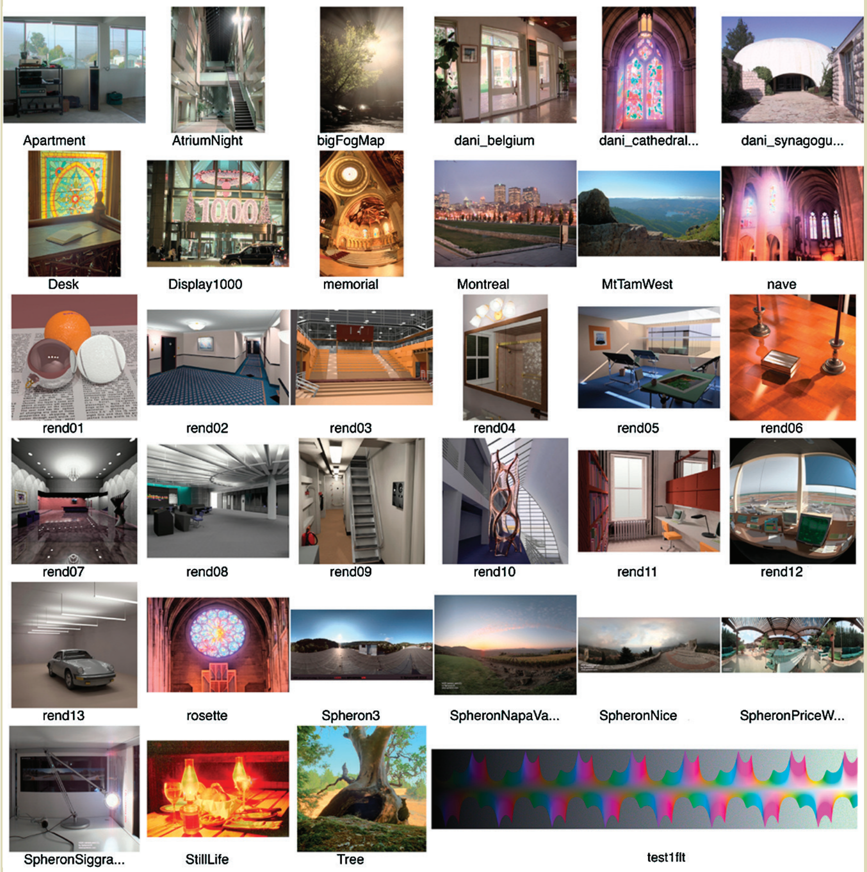

FIGURE 3.15 *The test set of 34 HDR images.*

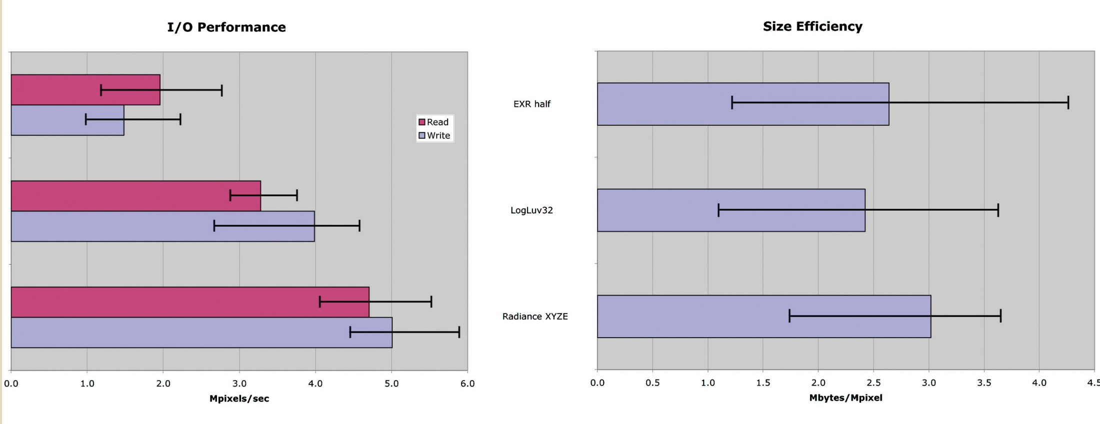

FIGURE 3.16 *I/O performance and file size for each of our three candidate formats. For I/O, longer bars mean faster reading and writing. Times were measured on a 1.8-GHz G5 processor. For size efficiency, shorter bars mean smaller files. Lines show the range of values (extrema) we saw in our tests.*

reinhard v.2005/03/22 Prn:13/06/2005; 14:43 F:reinhard03.tex; VTEX/SL p. 30

exposing quantization and range errors typical of more conventional LDR formats. The cost of this additional range and accuracy is modest — similar to including an extra alpha channel in an LDR format. This burden can be further reduced in cases in which accuracy is less critical (i.e., when multiple image read/edit/write cycles are not expected).

All of the existing HDR file formats are "lossless" in the sense that they do not lose information after the initial encoding, and repeated reading and writing of the files does not result in further degradation. However, it seems likely that "lossy" HDR formats will soon be introduced that offer much better compression, on a par with existing JPEG images. This will remove an important barrier to HDR adoption in markets such as digital photography and video and in web-based applications such as virtual reality tours.

The Resources section of the DVD-ROM includes complete demonstration software and images employing the JPEG-HDR lossy compression format described as preliminary work in this chapter.