Using a Residual Image to Extend the Color Gamut and Dynamic Range of an sRGB Image

Kevin E. Spaulding, Geoffrey J. Woolfe, and Rajan L. Joshi
Eastman Kodak Company
Rochester, New York

Abstract

Digital camera captures and photographic negatives typically have a larger dynamic range and color gamut than can be reproduced on a photographic print or a video display. This extra information is lost during the tone/color rendering process. In conventional photographic systems, most of this extra information is archived on the photographic negative, and can be accessed by adjusting the way the negative is printed. However, most digital imaging systems archive only a rendered video RGB image. Consequently, it is not possible to make the same sort of image manipulations that are possible with conventional photographic systems. This suggests that there would be advantages to archiving images using an extended dynamic range/color gamut color encoding. However, because of file compatibility issues, digital imaging systems that store images using a color encoding other than a standard video RGB representation (e.g., *sRGB*) would be significantly disadvantaged in the marketplace. This paper describes a solution that has been developed to maintain compatibility with existing file formats and software applications, while simultaneously retaining the extended dynamic range and color gamut information associated with the original scenes. Tests on a population of 950 real customer images have demonstrated that the extended dynamic range scene information can be stored with an average file size overhead of about 8%, compared to the *sRGB* images alone.

Introduction

Digital images are stored in a multitude of different digital file formats (e.g., *JPEG*, *TIFF*, *GIF*, etc.), using a large variety of different color encodings to specify the image colors. Images can be encoded in terms of device-independent color spaces, such as *CIE XYZ* tristimulus values, *CIELAB,* or the Kodak *PhotoYCC* color interchange space, or in terms of device-dependent color spaces, such as video RGB or printer CMY(K). Additionally, the digital images that are being encoded can be in a variety of different *image states*.[1] For example, the color values in a digital image may represent the color of an original scene or the color of a desired output picture.

The amount of flexibility that exists for manipulating and "repurposing" a digital image for different applications depends on both the image state and the color encoding used to represent the image. Images in a *scene-referred* image state typically have a much larger dynamic range than a rendered *output-referred* image and, therefore, allow a wider range of image manipulations to be used without encountering artifacts because of dynamic range clipping.

In addition to dynamic range differences, there are also color gamut differences between different image encodings. For example, the color gamuts of video devices are significantly different from those of color printers. Therefore, color spaces, such as *sRGB* that are based on video devices, cannot represent all of the colors that can be produced on color printers. Consequently, this can limit the quality of a print made from an image stored in a video RGB color space.

The most commonly used color encoding for digital images is *sRGB*,[2] which is becoming a *defacto* standard color interchange space for desktop digital imaging applications. The use of the *sRGB* color encoding provides several benefits including eliminating much of the ambiguity associated with the interchange of digital image data, and providing an image ready for direct display on a CRT. However, these benefits come at the expense of a limited color gamut and dynamic range relative to many other digital input and output devices. These issues limit the usefulness of *sRGB* for certain applications and restrict the capabilities of applications and output devices that can utilize the extended color gamut and dynamic range information that may have existed in an original input image.

Therefore, from the point of view of maximizing flexibility, it is desirable to use a digital image encoding that maintains the full dynamic range and color gamut of the image capture device until an optimal rendering can be selected by the user and the final output device is identified. *ERIMM RGB*[1] and the Kodak *PhotoYCC* color interchange space[3] are examples of color encodings that were designed to retain the large dynamic range and color gamut associated with original scenes. However, the primary disadvantage of encoding digital images using an extended-range, scene-referred color encoding is that they must be rendered before they can be displayed or printed on any particular output device. This rendering requirement is inconsistent with many desktop digital imaging applications that have been designed assuming that the images are already fully rendered and are ready for direct display on the monitor.

This results in a serious dilemma as the imaging industry transitions into the age of digital imaging. In order to maintain compatibility with the wide range of digital imaging applications and systems, it is necessary to deliver images from digital cameras and film scanners in the *sRGB* color space using standard file formats. However, this is inconsistent with the goal of providing the highest level of image quality and providing innovative new features and algorithms to customers.

This paper addresses this dilemma by providing a means to represent images with the full dynamic range and color gamut associated with extended-gamut, scene-referred color encodings, while simultaneously maintaining compatibility with imaging devices and applications that are configured to interchange *sRGB* image files in a standard image format.[4-6]

Residual Image Concept

The process of rendering an image for a particular output device involves "discarding" a significant amount of information present in the original digital camera capture or negative. The discarded information relates to both the extended dynamic range and color gamut of the original scene/image. To see where this information is lost, consider a typical imaging chain for converting from a scene-referred *ERIMM RGB* image to an output-referred *sRGB* image shown in Fig. 1.

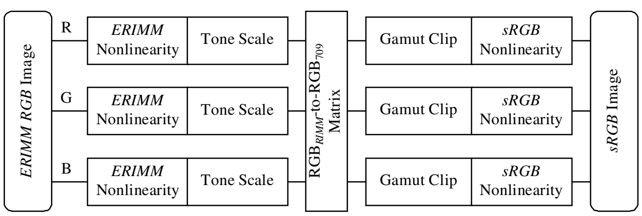


Figure 1. Typical imaging chain to convert from an ERIMM RGB scene-referred image to an output-referred sRGB image

The largest amount of information is lost during the *tone scale* step. This step maps the large dynamic range of the scene onto the limited dynamic range of the rendered picture, using some sort of "s-shaped" tone rendering function. Information is also discarded in the *sRGB gamut-clipping* step, which discards any information about picture color values outside of the *sRGB* gamut. A small amount of additional information is also lost due to quantization errors that occur in the final step of applying the *sRGB* nonlinearity.

The most significant information losses occur in the extreme highlight and shadow regions of the scene and in areas of very saturated color, while large portions of the image suffer only a minimal information loss attributable to the introduction of the 8-bit quantization errors. The *residual image* concept presented here makes use of this by computing a difference image between the original extended-gamut image and the final *sRGB* image in such a way that the difference signal for the in-gamut regions of the image is essentially zero. The resulting residual image is, therefore, highly compressible because most images contain large areas where the image pixels are in gamut. Next, the compressed residual image is stored as metadata within the normal *sRGB* image file. Other useful metadata, such as information about the original rendering transformation, that may be useful to image processing applications can also be stored in the file as metadata. At a later time, the residual image information can be combined with the *sRGB* image data to reconstruct the extended color gamut image.

The residual image approach provides several important benefits relative to other alternatives for storing extended-gamut images. The primary advantage is that the image file will be compatible with the large population of applications that are designed to accept images stored in a video RGB metric. One of the main disadvantages associated with the direct use of extended-gamut color spaces has been that special file readers are required to display/print the image on a particular output device. With the residual image approach, the image file will look just like any other *sRGB* image file to an application that does not know how to interpret the special metadata tags. However, applications that are able to interpret the metadata tags can reconstruct the original extended-gamut image and thereby produce images that are advantaged relative to the applications that only use the *sRGB* image data. A related advantage to the residual image approach is that the basic *sRGB* image stored in the image file can provide a fast monitor preview of the image, whereas other extended-gamut color spaces would require the image to be transformed before being displayed on a CRT. Finally, the file size required using the technique is only slightly larger than that of the *sRGB* image alone. This is because the residual image metadata will be highly compressible for most typical images. Preliminary results indicate that the average amount of metadata information should be less than 10% of the compressed *sRGB* image size.

Residual Image Encoding

Extended-Gamut Color Encoding Selection

As discussed above, scenes typically have a substantially larger dynamic range and color gamut than can be encoded in a rendered output-referred color encoding such as *sRGB*. It is the ability to capture and retain the extended-gamut information that has provided conventional photography with much of its robustness. This same extended-gamut information is needed to enable many of the new concepts and algorithms that are being developed for digital imaging systems.[7] Therefore, it is desirable to design the residual image encoding such that it is possible to recover all of the original scene information in a digital camera capture or a photographic negative. The large color gamut and dynamic range of the *ERIMM RGB* color encoding[1] makes it an appropriate choice for use as the reference color space.

Residual Image Creation Path

The fundamental residual image concept shown in Fig. 2 involves computing a residual image, representing the difference between the original extended-gamut *ERIMM RGB* image and the final rendered *sRGB* image. Two factors complicate the formation of this residual image. First, the extended-gamut image and the *sRGB* image are in two different color spaces. Second, the colorimetry of even the in-gamut portions of the image has been modified by the tone/color rendering process that was used to form the *sRGB* image. Therefore, subtracting the *sRGB* image from the *ERIMM RGB* would result in nonzero residual image values, even for the regions in the image where no information was lost in the color rendering process. This would defeat the main advantage of the residual image concept because the resulting residual image would not be very compressible.

This problem can be overcome by bringing the extended-gamut and *sRGB* images to a common color space and image state before computing the difference image. This can be accomplished in several ways, but the most desirable alternative is shown in Fig. 2. The upper branch of this diagram shows the same basic color rendering process for forming an *sRGB* image from an *ERIMM RGB* image that was shown in Fig. 1. The formation of the residual image requires both the extended-gamut input image and the rendered output image to first be brought to a common color space. This involves applying the inverse of the *sRGB* nonlinearity to get back to linear picture radiance values and applying the inverse of the primary conversion used in the color-rendering path. (This is equivalent to transforming the *sRGB* values to linear *ROMM RGB* color values.[1])

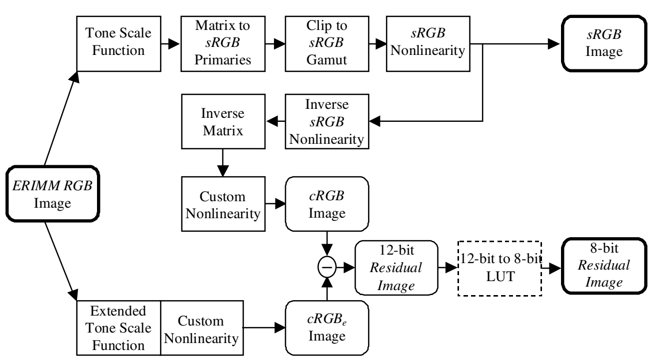


Figure 2. Residual image creation path

In order to get the two images into a common rendering state, the *ERIMM RGB* image is processed through an extended version of the rendering tone scale used to create the original *sRGB* image. The extended tone scale is designed to match the original rendering tone scale through the mid-tones, but deviates at the light and dark ends to avoid the severe tone compression associated with the rendering tone scale. This creates an extended-range rendered image that will be equivalent to the original rendered image throughout most of the gamut of the original rendered image.

The last step is to encode the extended-gamut image and the limited-gamut image using the same nonlinear encoding function. A 12-bit logarithmic encoding analogous to that used for the *ERIMM RGB* color encoding[1] is used for this purpose, in order to efficiently encode the large dynamic range of the extended-gamut image. In Fig. 2, the resulting limited-gamut image is labeled *cRGB* (for "custom" RGB), and the corresponding extended-gamut image is labeled $cRGB_e$.

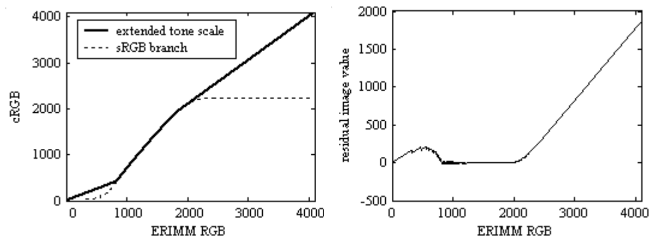
An example of an extended tone scale LUT is shown as the solid curve in Fig. 3(a). A dashed curve illustrating the cascade of all the LUTs on the upper branch of the recommended residual image creation path is shown for comparison. This would correspond to the result that would be obtained by mapping neutral *ERIMM RGB* scene color values through to *sRGB* values and, finally, to the corresponding *cRGB* values. It can be seen that the extended tone scale LUT matches the result from the *sRGB* path throughout much of the tone scale but it preserves the highlight and shadow information that was clipped in the *sRGB* image. At the top end of the tone scale, this is accomplished by extending the tone scale function bypassing the highlight compression associated with the rendering tone scale. At the bottom end of the tone scale, a different approach is used incorporating a slope limit. This option avoids some of the complexities that would be associated with the use of negative $cRGB_e$ values. (In this example, a slope limit of 1/2 is used, ensuring that, at most, two consecutive *ERIMM RGB* values are mapped to the same $cRGB_e$ value.)

With the extended-gamut and the limited-gamut images in the same color space and image state, a residual image with the desired characteristics can be determined using a simple difference operation:

$$\Delta = cRGB_e - cRGB \quad . \tag{1}$$

Because the *cRGB* and $cRGB_e$ images are both 12–bit encodings, the difference value can, in theory, span the range from –4095 to +4095. However, in practice, only a fraction of this range is required. In order to support the formation of a reconstructed *ERIMM RGB* image with the full precision of the original, a 12–bit residual encoding is necessary. However, an 8–bit residual image can be formed that has sufficient quality for most applications and is more convenient in many workflows.

To visualize the characteristics of the residual image signal, consider the ramp of neutral scene colors shown in Fig. 3(a). The corresponding neutral residual image values, calculated using Eq. (1), are shown in Fig. 3(b). It can be seen that throughout a wide range of scene exposure values between *ERIMM RGB* values of 800 and 2000, where most of the properly exposed scene information is located, the residual image signal is very small. For larger exposures, the residual image signal grows continuously to encode the information that is clipped in the *sRGB* image. For deep shadows, there is also a significant residual image signal corresponding to the exposures where the slope limit was imposed on the extended tone scale LUT to ensure reversibility. A series of serrations can be observed at the low exposure end of the tone scale. These correspond to the quantization errors introduced in encoding the *sRGB* image. A serration is formed each time the *sRGB* value increments by one code value. Note that positive residual image values are obtained in both the shadow and highlight regions of the tone scale. This is because the $cRGB_e$ values in Fig. 3(a) are larger than the *cRGB* values at both ends of the tone scale.

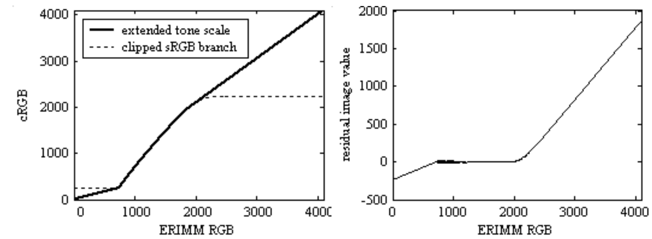


(a) (b)

Figure. 3. (a) Extended tone scale LUT compared to the cascade of the LUTs for the sRGB branch; (b) residual image values generated for neutral ERIMM RGB values

The residual image signal given in Fig. 3(b) produces reconstructed *ERIMM RGB* values that are accurate to within ±1 code value when recombined with the corresponding *sRGB* image. However, in most cases, both the *sRGB* image and the residual image will be compressed for storage purposes. Not only do the serrations in the residual image signal cause it to be less compressible, but it was also found that objectionable artifacts could be introduced in the reconstructed *ERIMM RGB* images for even small compression errors. Consequently, the encoding path was modified slightly to address this problem. It was found that a simple solution was to introduce a clipping step into the calculation of the *cRGB* values. In this clipping step, *cRGB* values that are less than a given threshold are set equal to the threshold value.[8]

A threshold value of 240 was sufficient to eliminate the problem regions in the reconstructed *ERIMM RGB* image, while not moving an excessive amount of shadow image signal into the residual image. For implementation purposes, this clipping step can be incorporated into the custom nonlinearity LUT so that there is no additional computational complexity. The resulting clipped *cRGB* values for neutral *ERIMM* RGB values are shown in Fig. 4(a), and the corresponding residual image signal is shown in Fig. 4(b). In comparison to the case shown in Fig. 3(b), it can be seen that the large serrations have been eliminated from the residual image signal. Therefore, the residual image will be significantly more compressible.



(a) (b)

Figure 4. (a) Extended tone scale LUT with slope limit, compared to the cascade of the LUTs for the sRGB branch where a clipping function has been introduced into the custom nonlinearity LUT; (b) residual image values generated for neutral ERIMM RGB values using clipped cRGB value.

For the case where bit-depth of the residual image is not a critical issue, the distribution of residual image values for real images suggests that the residual image values can be encoded as 12-bit numbers with no significant loss of information. A baseline unsigned 12-bit encoding was defined to be:

$$\Delta_{12} = \Delta + 1000 = cRGB_e - cRGB + 1000\,, \qquad (2)$$

where the Δ_{12} values should be clipped to the range 0 to 4095 for storage as a 12–bit integer.

8-bit Residual Image Encoding

While a 12–bit residual image encoding is required to maximize the precision of the reconstructed *ERIMM RGB* image, there are some significant barriers to its use. Foremost among these barriers is the compression of the residual images. Many of the widespread image compression software libraries are only available in 8–bit versions or must be switched between different precision levels at compile time. Because a conventional 8–bit compression algorithm would be required for the *sRGB* image, and a 12-bit compression algorithm required for the residual images, the compile-time switch does not represent a practical solution. To overcome this difficulty an 8–bit residual image encoding was developed with the understanding that it represented a small image quality compromise.

The 8–bit encoding was defined to be a simple transformation of the 12–bit residual image values that could be implemented as a LUT as shown in Fig. 2. Conversion of the 12–bit residual image values to 8–bits requires information to be discarded. This can be accomplished by either quantizing the residual image values more coarsely or by limiting the range of the residual image signal. A careful analysis of error propagation through the reconstruction path determined that it generally required a change of four code values in the 12–bit residual image signal to produce a one code value change in a corresponding *sRGB* image. This suggests that for most purposes, it should be possible to quantize the residual image values by a factor of 4×

without risking significant quantization artifacts in the final image relative to those associated with the conventional *sRGB* image.

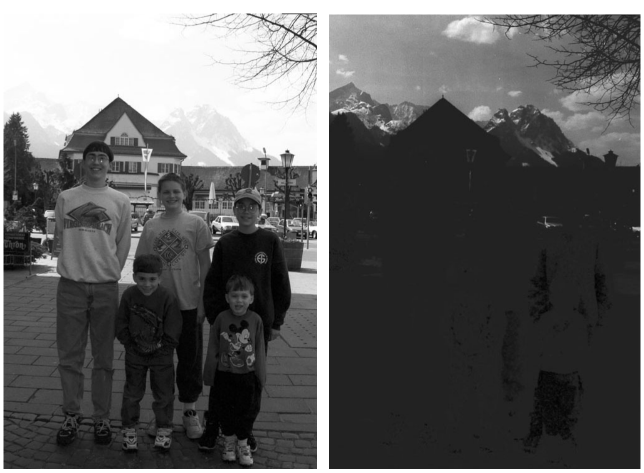
If a quantization factor of 4× is used, this implies that a range of $256 \times 4 = 1024$ 12–bit residual image values can be selected for the 8–bit encoding. The largest negative residual image value that can be expected for neutral *ERIMM RGB* values is –240. Any residual image values more negative than this would result from gamut clipping of saturated colors. Because most of the benefit of the residual image concept lies in the ability to encode a larger dynamic range, rather than the ability to encode a large color gamut, it makes sense to use the maximum amount of range for the positive residual image values. Based on these considerations, a subset of values starting at –240 was determined to be a reasonable choice. An 8–bit residual image encoding can be defined as follows:

$$\begin{aligned} \Delta_8 &= round\left[(\Delta + 240)/4\right] \\ &= round\left[(\Delta_{12} - 760)/4\right] \end{aligned}, \qquad (3)$$

where the Δ_8 values are clipped to be in the range of 0 to 255 before storing them as unsigned byte values.

This implies that all of the neutral *ERIMM RGB* values above approximately 3000 will be clipped to residual image values of 255. (A neutral *ERIMM RGB* value of 3000 corresponds to a scene radiance value about 10× that of a properly exposed perfect diffuse reflector.) Therefore, it will be impossible to reconstruct *ERIMM RGB* values larger than this exposure value. However, this will only affect a small percentage of images from sources with a very high dynamic range.

An example of an *sRGB* image and the corresponding 8–bit residual image are shown in Fig. 5. It can be seen that the *sRGB* image contains large highlight areas in the background that have been clipped to white. The residual image retains the information that was lost during the *sRGB* rendering process.



(a) (b)

Figure 5. Example images generated using the method of Fig. 2: (a) sRGB image and (b) corresponding 8–bit residual image

Reconstruction of *ERIMM RGB* Images

The residual image, together with the *sRGB* image, can be used to reconstruct an *ERIMM RGB* image using the imaging chain shown in Fig. 6. The first step is to process the *sRGB* image to determine a *cRGB* image using the same steps that were discussed earlier. Namely, the inverse *sRGB* nonlinearity is applied to determine the linear rendered-image intensity values, the inverse matrix is used to convert the intensity values to the *ROMM RGB* primaries, and finally, the custom nonlinearity (with clipping) is applied to determine the 12–bit *cRGB* values. Next, the residual image values are added to the *cRGB* values to form a reconstructed extended-range, $cRGB_e$ image. Recalling that the 12–bit residual image values were stored with an offset of 1000 to make them all positive, the appropriate transformation can be obtained by rearranging Eq. (2) to solve for $cRGB_e$:

$$cRGB_e = cRGB + \Delta = cRGB + \Delta_{12} - 1000. \qquad (4)$$

Note that if an 8–bit residual image encoding was used, a conversion first needs to be applied to the 8–bit residual image values to determine the corresponding 12–bit values. If the default 8–bit encoding described above in Eq. (3) was used, this conversion will be given by:

$$\Delta_{12} = 4\Delta_8 + 760. \qquad (5)$$

Finally, the $cRGB_e$ values are mapped to the corresponding reconstructed *ERIMM RGB* values by applying the inverse custom nonlinearity followed by the inverse extended tone scale.

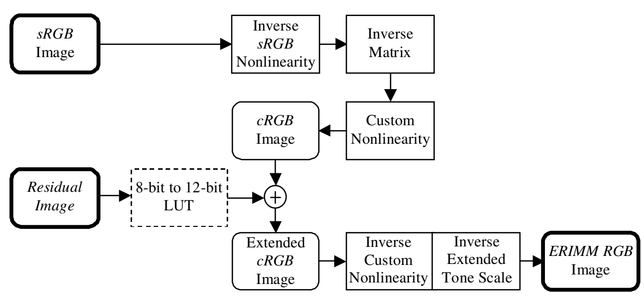


Figure 6. Reconstruction of ERIMM RGB image

Compression of the Residual Images

Another important consideration was the form of image compression that should be used for the residual image. Preliminary investigations concluded that the use of non-standard compression algorithms could potentially provide a benefit because they could be designed to take advantage of added knowledge about the characteristics of the residual image. For example, they could make use of the fact that the residual image signal should be negligible for regions in the image where the *sRGB* image values are not near the gamut boundary. If it were anticipated that the residual image would be used exclusively by proprietary applications, the use of a non-standard compression algorithm would be feasible. However, it was concluded that the capability of implementing residual image decoding software using off-the-shelf compression routines was critical to simplify implementation issues and to preserve the option of easily enabling other software applications to access the extended-gamut image data.

Therefore, the conclusion was to use conventional *JPEG* compression for the residual image data as well as the *sRGB* image data. As noted earlier, one limitation of many of the available *JPEG* compression libraries is that it is not possible to use the 12–bit version and the 8–bit version of the *JPEG* algorithm simultaneously. Hence, in the first implementation of this technology, it was necessary to use the 8–bit residual image encoding method.

Choice of the Q-table for JPEG Compression

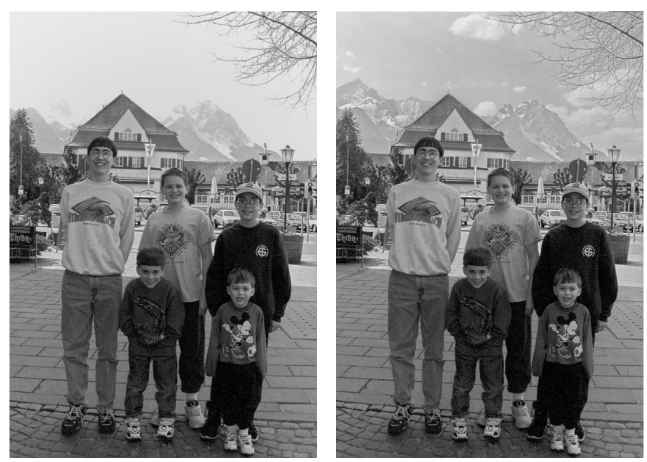
Another important consideration is what level of com-pression should be used for the residual image data. In the *JPEG* algorithm, the quantization table (Q-table) controls the level of compression. In the results presented here, the Q-table used for the compression of *sRGB* images was identical to the Q-table currently used in a number of typical consumer products. We will refer to the level of com-pression as 1×. The "1" indicates that all the entries in the baseline Q-table are multiplied by 1. Similarly, "*N*×" indi-cates that all the entries in the baseline Q-table are multi-plied by *N*. The choice of *N* will determine the file size and the corresponding image quality for the residual image. Higher values of *N* produce lower file sizes as well as lower image quality. We considered the 1×, 2×, 3×, and 4× compression levels for the compression of the residual image.

The effect of compression level on the reconstruction error for the extended-gamut image and on the file size increase required for the residual image data was examined using a normal distribution of 950 images. These images were obtained from film scans of approximately 40 real customer orders. The size of the images used for this test was 1024 × 1536 pixels. Visual evaluation of the set of 950 images indicated that 20% had significant extended dynamic range information, while another 25% had a moderate amount of extended dynamic range information. Therefore, about 55% of the images fell into the category of having little or no extended dynamic range information.

It was found that using a 2× quantization factor as the baseline compression level for 8–bit residual images maintained reasonable image quality, while keeping the average residual image size at an acceptable level. This choice can be easily adjusted on an application-by-application basis. For the 950 images that were tested, the average size of the compressed *sRGB* image was 363 Kbytes, while the average size of the compressed residual image was 28 Kbytes. Thus, the added overhead associated with storing the residual image as metadata in the *sRGB* image file averaged less than 8%.

Use of Residual Image Data

The preceding sections have discussed how to create residual images, and use the residual images to form reconstructed *ERIMM RGB* images. However, the real value of the residual image concept lies in what you can do with the reconstructed extended range image data.[9] There are many types of image processing algorithms that can be used to manipulate the extended range image data, ranging from simple exposure adjustment to sophisticated adaptive tone scale manipulation techniques. Many of these algorithms can be applied to the image in the *ERIMM RGB* color space and will produce a modified *ERIMM RGB* image. For example, Fig. 7 shows an improved image obtained by applying a semi-automatic digital "dodge-and-burn" algorithm to the image shown in Fig. 5. It can be seen that the improved image determined from the reconstructed *ERIMM RGB* image is able to bring back much of the detail that was lost in the original *sRGB* image. An analogous "improved" image created using just the *sRGB* image data is shown for comparison. While the overall density of the foreground portion of the image can be improved somewhat, it can be seen that there is nothing that can be done to restore the details in the overexposed background regions of the image.



(a) (b)

Figure 7. Comparison of using a digital "dodge-and-burn" algorithm to manipulate (a) the original sRGB image shown in Fig. 5; and (b) a reconstructed ERIMM RGB image created using the residual image

Implementation in Digital Cameras

The first product implementation of the residual image technique is in several models of Kodak's high-end professional digital cameras (Kodak Professional DCS 720, and DCS 760X digital cameras, and Kodak Professional DCS Pro Back 645, and DCS Pro Back Plus), where it is known as *Extended Range Imaging Technology* (*ERI*).[10] There has been a long-standing dilemma with professional digital camera users between the desire to retain the maximum quality and flexibility offered by storing "raw" unrendered image files and the conflicting desire for ease-of-use and interoperability. Customers have had to make a choice at the time of image capture which of these requirements is most important to them. If they chose to store images in Kodak's proprietary raw format, it was necessary to use special software, such as an Adobe Photoshop plug-in, to process the image on the host computer. This made it impossible to conveniently use these images in many workflows. Additionally, the size of the raw images was quite large, limiting the number of images that could be stored on the camera's memory cards. On the other hand, if the customer chose to store the images in a finished *JPEG* file format, they were forced to live with the limited color gamut/flexibility of the *sRGB* color encoding, which was unacceptable for many applications.

The *ERI* solution provides a means to simultaneously satisfy both of these customer requirements. Cameras equipped with *ERI* produce *Exif* 2.1-compliant *JPEG* image files, which are compatible with any standard image-capable software application. However, the residual image information stored as metadata in the image file provides the capability of forming a reconstructed *ERIMM RGB* image with virtually all of the benefits of the original raw image file. The user can access and manipulate the reconstructed extended-range image data using specially developed Adobe Photoshop software File Format Module (FFM). The first version of this software allows all of the same image manipulations, such as exposure/white-balance corrections and tone scale adjustments, which have been available with the raw image files. It is anticipated that forth-coming versions of this software will provide additional image processing options. Other software applications may also be enabled to decode and use the residual image data at some point in the future.

Conclusions

A method for encoding extended dynamic range/color gamut digital images has been developed that maintains compatibility with existing file formats and software applications. The method works by forming a residual image that can be stored using proprietary metadata tags in the image file. The residual image represents the difference between the original extended dynamic range image and the final rendered *sRGB* image and is determined in a manner such that the pixel values are negligible for in-gamut portions of the image. As a result, the residual image will be highly compressible and will not increase the file size significantly for most images. Applications that are appropriately enabled can decode the residual image metadata, and use it to reconstruct the original extended dynamic range image. Applications that are not enabled to decode the residual image data will still have access to the standard *sRGB* image in the usual way. This approach preserves the advantages of high dynamic range image capture mechanisms, such as conventional photographic film or high-quality digital cameras, without sacrificing the critical issue of compatibility with existing software and output devices.

The ability to reconstruct the extended dynamic range/color gamut image allows the same sort of image manipulations that have historically been possible with conventional photographic systems (e.g., exposure and color balance adjustments). Additionally, it enables many of the new advanced image processing algorithms that are being developed to improve the quality of images, such as digital "dodge-and-burn" techniques that rely on having the extended scene dynamic range information. The method was designed to encode images that originate in the *ERIMM RGB* scene-referred color encoding, but it can be generalized to also work with other large-gamut color spaces. Extensive testing has shown that the residual image metadata requires an average file size overhead of about 8% using a compression level that has no significant affect on the image quality under normal viewing conditions.

References

1. K. E. Spaulding, G. J. Woolfe and E. J. Giorgianni , "Optimized Extended Gamut Color Encodings for Scene-Referred and Output-Referred Image States," J. Imaging Sci. Technol., 45, 418-426 (2001).
2. "Multimedia systems and equipment-Color measurement and management-Part 2-1: Color management-Default RGB color space-sRGB," IEC 61966-2-1.
3. E. J. Giorgianni and T. E. Madden, Digital Color Management: Encoding Solutions, Addison-Wesley, Reading, MA (1998).
4. A. L. McCarthy, K. E. Spaulding, and E. J. Giorgianni, "Using a residual image to represent an extended color gamut digital image," U. S. Patent 6,282,311 (28 Aug 2001).
5. A. L. McCarthy, K. E. Spaulding, and E. J. Giorgianni, "Using a set of residual images to represent an extended color gamut digital image," U. S. Patent 6,282,313 (28 Aug 2001).
6. A. L. McCarthy, K. E. Spaulding, and E. J. Giorgianni, "A system using one or more residual images to represent an extended color gamut digital image," U. S. Patent 6,282,312 (28 Aug 2001).
7. H.-C. Lee, "Automatic tone adjustment by contrast gain-control on edges," U.S. Patent 6,285,798 (2001).
8. K. E. Spaulding, R. L. Joshi, and G. J. Woolfe, "Using a residual image to represent an extended color gamut digital image including a clipping step," U. S. Patent 6,301,393 (9 Oct 2001).
9. K. E. Spaulding, A. L. McCarthy, and E. J. Giorgianni, "Method of applying manipulations to an extended color gamut digital image," U. S. Patent 6,285,784 (4 Sep 2001).
10. "KODAK PROFESSIONAL Extended Range Imaging Technology," white paper posted on the Kodak website, <http://www.kodak.com/global/en/professional/products/cameras/erit/erit.jhtml>

Biography

Kevin Spaulding received a BS in Imaging Science from Rochester Institute of Technology in 1983 and MS and Ph.D. degrees in Optical Engineering from the University of Rochester in 1988 and 1992, respectively. He has been with Eastman Kodak Company since 1983 where he is currently a Senior Principal Scientist in the Imaging Science Division. He is also technical secretary for the CIE TC8-05 Committee, which is tasked with defining standards for the unambiguous communication of color information in images, and project leader/editor for the ISO 22028-1 standard. His research interests include digital color encoding, color reproduction, digital halftoning, image quality metrics, and image processing algorithms for digital camera and printers.