The JPEG XT Suite of Standards: Status and Future Plans

Thomas Richter[1], Tim Bruylants[2,3], Peter Schelkens[2,3], Touradj Ebrahimi[4]

[1]IZUS/TIK Computing Center, Universität Stuttgart, 70550 Stuttgart, Germany
[2]Vrije Universiteit Brussel, Pleinlaan 9, 1050 Brussels, Belgium
[3]iMinds, Gaston Crommenlaan 8 (box 102), 9050 Ghent, Belgium
[4]EPFL Multimedia Signal Processing Group, CH-1015 Lausanne, Switzerland

ABSTRACT

The JPEG standard has known an enormous market adoption. Daily, billions of pictures are created, stored and exchanged in this format. The JPEG committee acknowledges this success and spends continued efforts in maintaining and expanding the standard specifications. JPEG XT is a standardization effort targeting the extension of the JPEG features by enabling support for high dynamic range imaging, lossless and near-lossless coding, and alpha channel coding, while also guaranteeing backward and forward compatibility with the JPEG legacy format. This paper gives an overview of the current status of the JPEG XT standards suite. It discusses the JPEG legacy specification, and details how higher dynamic range support is facilitated both for integer and floating-point color representations. The paper shows how JPEG XT's support for lossless and near-lossless coding of low and high dynamic range images is achieved in combination with backward compatibility to JPEG legacy. In addition, the extensible boxed-based JPEG XT file format on which all following and future extensions of JPEG will be based is introduced. This paper also details how the lossy and lossless representations of alpha channels are supported to allow coding transparency information and arbitrarily shaped images. Finally, we conclude by giving prospects on upcoming JPEG standardization initiative JPEG Privacy & Security, and a number of other possible extensions in JPEG XT.

1. INTRODUCTION

High Dynamic Range (HDR) imaging is becoming increasingly popular in photography, computer graphics and digital cinema applications. Besides JPEG 2000[4] and JPEG XR[5] no other image or video coding solutions are currently providing support for HDR content. In this context, lack of support for HDR by the omnipresent JPEG legacy standard[1–3] can become an issue in its further adoption. Enabling coding of HDR content by a straightforward increase in bit-depth support to 16-bit integer (IDR) or floating-point (HDR) would break backward compatibility with JPEG legacy implementations. Hence, the JPEG XT standard[9,17] enables HDR support by splitting the image data in a Low Dynamic Range (LDR) JPEG legacy compliant codestream and a residual stream — allowing to scale-up to an HDR decoded image — which is smartly embedded in the JPEG legacy codestream such that it does not jeopardize backward or forward compatibility. Mapping the HDR image onto an LDR image is performed by a tone-mapping algorithm, which is signaled in the codestream as look-up tables (LUT) or tone-mapping curves to allow the decoder to perform an approximate tone-mapping operation.

2. STRUCTURE OF THE STANDARD

The JPEG XT standard is structured into nine parts, forming a hierarchy of extensions of JPEG as it is used today. However, the omnipresent image file format diverts somewhat from the original ISO document, formally ISO/IEC 10918-1 | Recommendation ITU-T T.81, which defines additional coding modes that have been rarely put into use, namely the arithmetic coding mode, a 12-bit lossy coding mode, a lossless coding mode, and a hierarchical coding mode. Of all these modes, only the lossless mode has found some adoption in the medical market. Arithmetic coding is supported by some implementations but lacks serious market penetration, and the hierarchical mode — combining other JPEG coding modes with pyramid coding — is mostly unknown, even to many experts in the field.

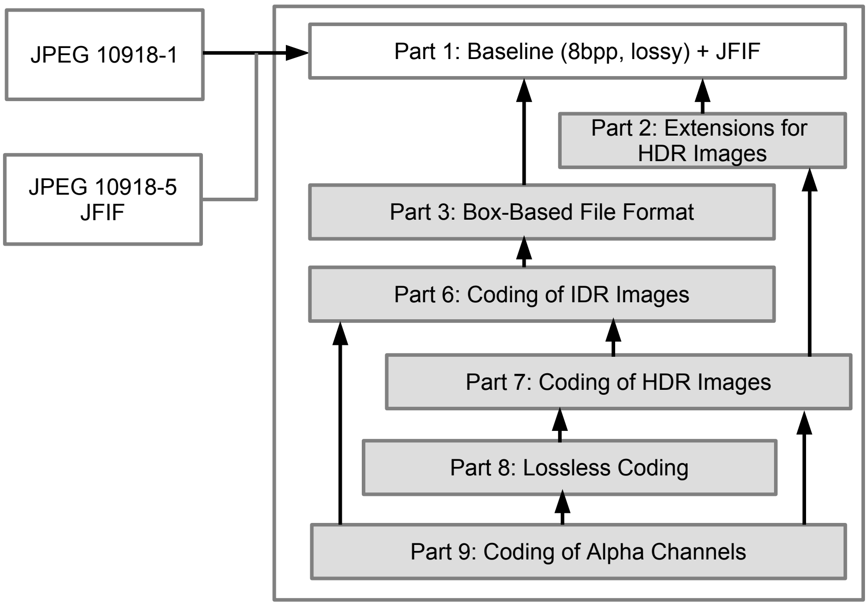

Figure 1. Overview on the parts of JPEG XT and their relation. Parts 4 and 5 define conformance testing and the reference software and are not included in the picture.

Even though may seem that the "living JPEG" is purely a subset of the original ISO standard, it also relies on a set of extensions. In fact, the most prominent and widely accepted JPEG extension is JFIF,[6,7] a file format extension for JPEG that adds two important features; first, it defines a colorspace within which JPEG images are coded, using Recommendation ITU-R BT.601, and a decorrelation from this colorspace into an internal opponent colorspace known from color TV; second, normative upsampling of the chroma components onto a common sampling grid. These two fundamental operations are missing from the original ISO standard because JPEG was initially never considered to describe all aspects of image coding. It was rather intended as the core coding engine of a container-based image file format and as such constrained itself to the transport mechanism of (otherwise meaningless) integer sample values of multiple components of unknown origin. Although the JPEG committee later designed such a container as the SPIFF file format,[8] this initiative arrived too late in the market when the much simpler JFIF extension already dominated the application domain.

Part 1 of the JPEG XT standard,[9] and by that also the core of the JPEG XT standard, restricts the legacy JPEG coding modes to those that were accepted in practice, namely baseline, sequential and progressive Huffman coded modes, and extends them with the JFIF definitions on how to reconstruct image color pixels from raw JPEG sample values by means of upsampling and transforming from YCbCr to RGB. It is by that in essence identical to what is, as of today, understood as "JPEG", even though it diverts slightly from the intent of the original ISO/IEC and ITU-T specifications.

Part 2[10] describes a backward compatible coding mechanism for HDR/floating point images; even though the entropy coding mechanism and the coding principles are identical to Profile A of part 7, its transport mechanism differs from the latter to support early implementations of the standard in the market. For its technical discussion, we refer to part 7.

Part 3[11] offers a generic extension mechanism that allows embedding side-channels and additional meta-data into JPEG codestreams in such a way that legacy implementations will just skip over the additional data. Thus, part 3 is not a description of an image decoder, but rather defines the embedding of syntax elements, called “boxes”, into the legacy JPEG syntax. Section 4 will discuss the details of the transport mechanism.

Part 6[13] uses the foundation of part 1 and the extension mechanism of part 3 to enhance the sample depth of 8-bits per component up to 16 bit integer samples. The intended use case of part 6 is to describe an “intermediate dynamic range (IDR)” image, i.e. an image whose dynamic range is somewhat larger than that of traditional JPEG images. Its 8-bit integer mode constraints the dynamic range of its samples to little more than two magnitudes ($255 \approx 10^2$). While this small range is usually sufficient for reproduction on CRTs or consumer-level LCD monitors, it is only a fraction of the capabilities of modern digital camera sensors whose internal precision is usually above ten bits per sample. Part 6 hence supports encoding of raw sensor data created by a single scene exposure, and allows users to transport such data in a vendor-neutral way. Compression of the dynamic range to 8 bits per sample, a process commonly known as “tone mapping” can then happen in a later part of the image production chain, e.g. when viewing the images on a standard consumer monitor. This allows a work-flow that is similar to analog film where the “exposure process” from a negative to a print also reduces the dynamic range of the analog film to that of the paper print. Part 6 encoding is almost identical to part 7 profile C encoding, except that it lacks a conversion step from integer to floating-point that is present in the latter.

Part 7[14] is a further extension of part 6 and is able to describe images of much larger dynamic range requiring floating-point representations. Such images are usually acquired by multiple exposures of the same scene while varying the aperture from shot to shot, and then merging the individual shots computationally into one single scene description. This process is also known under the name “HDR photography”.[20] Part 7 can describe both relative radiance images and absolute radiance images. In the latter case, image samples are calibrated to absolute physical radiance given in a suitable unit encoded in the image codestream, for example by calibrating the camera equipment by a radiometer. Relative radiance only describes the physical radiance up to a proportionality factor. Part 7 is defined along three profiles that constrain the parameter selection of one common but complex decoding logic depicted in Fig. 2. One of the profiles is a straightforward extension of part 6. Part 7 is discussed in section 5.

While parts 6 and 7 only define lossy coding mechanisms, part 8[15] is a lossless extension of both parts 6 and 7. It allows lossless encoding of images of integer or floating-point samples of up to 16 bit resolution. Its performance[27] is comparable to that of PNG,[19] though it remains backward compatible to JPEG XT part 1, i.e. JPEG legacy decoders will always be able to reconstruct such streams to an 8-bit per sample image, though will then create loss. A full part 8 implementation is required to ensure lossless reconstruction. We will discuss part 8 in section 6.

Part 9[16] is completely orthogonal to parts 6, 7 and 8 and fills one important gap in JPEG legacy specification, namely, the coding of opacity information. That is, part 9 adds to JPEG legacy the ability to include an alpha channel along with the image data. The most important application for such a feature is the design of web-pages where text, graphics and visual objects should flow free-form around images. Traditional JPEG only allows rectangular image bounds, hence forces web designers to represent images with much larger PNG files — or use other CSS-based trickery. Section 7 will introduce this extension.

Part 5[12] of JPEG XT specifies a reference software[24] that facilitates implementers’ access to JPEG XT technology. Part 4 defines the mechanisms for reference testing and suitable error bounds for conforming implementations.

JPEG will continue to extend the JPEG XT standard and by that its JPEG family of standards. Some future directions, such as the privacy protection of images, are currently under discussion in the JPEG committee. We will cover such extensions in section 9.

3. ARCHITECTURE OF THE JPEG XT STANDARD

Even though JPEG XT currently extends over nine parts, all parts derive from a single common decoder architecture depicted in Fig. 2. The fundamental component is the JPEG legacy decoder, represented by

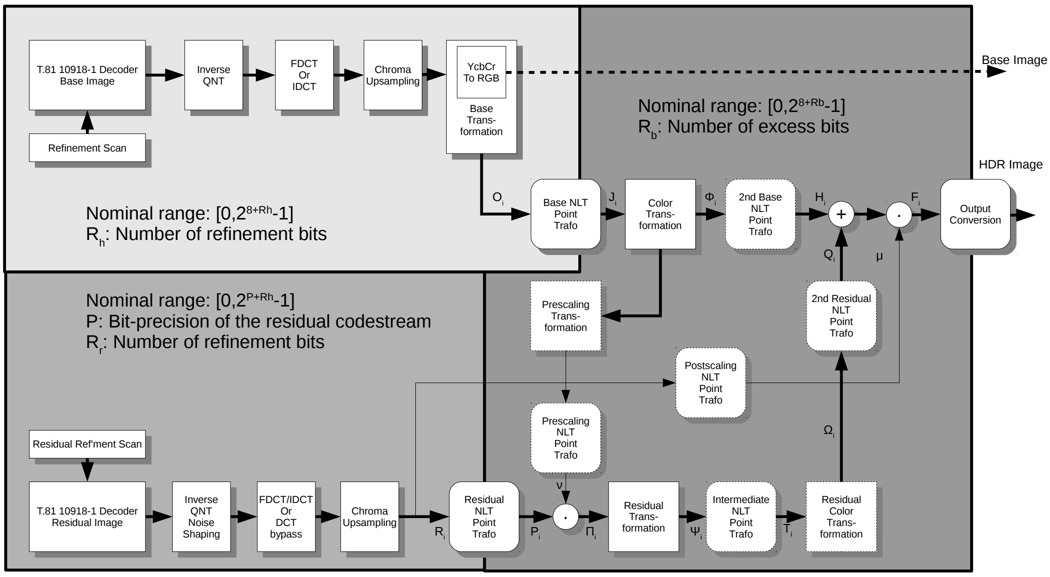

Figure 2. Decoder design of JPEG XT, merging the functionality of all parts except part 9. Thick lines represent vectorial data (i.e. three components), thin lines scalar data (i.e. a single number). The gray areas represent domains of constant nominal range. Non-linear transformations use an internal scale of $[0, 1]$ and upscale or downscale their inputs and outputs accordingly to match the domain range they operate in.

the left-top box labeled "T.81 10918-1 Decoder", and the inverse DCT and inverse quantization to its right. These three boxes together — the entropy decoder, inverse quantization and inverse DCT — make up the JPEG legacy decoder as specified in ISO/IEC 10918-1 | Recommendation ITU-T T.81. The two boxes to the right — chroma upsampling and conversion from YCbCr to RGB – were later added by the widely accepted JFIF,[6] the JPEG Interchange File Format, which became an ISO standard in 2011.[7]

This basic decoder is now extended in the following directions: The bit precision in the DCT domain can be extended by encoding additional least-significant bits, hidden from legacy decoders, for enlarging the sample precision in the DCT domain. This extension is called *Refinement Coding*; it is comparable to the JPEG 2000 *Refinement Coding Pass*, which also adds least significant bits, but unlike its JPEG 2000 counterpart, it is based on the successive approximation scan pattern of the progressive mode of JPEG.

A second JPEG codestream, the *Residual Codestream*, extends the dynamic range in the spatial domain by providing a *correction* or *extension* signal. Decoding the residual codestream is depicted by the lower left three boxes of Fig. 2. Similar to the legacy codestream, raw entropy coded data is decoded, then inversely quantized and inversely DCT transformed. The residual signal can also be extended by refinement scans, similar to the base signal.

All boxes on the right of the chroma upsampling steps in Fig. 2 merge the base image and residual image together to form an IDR or HDR output image. While the depicted set of operations in this figure is complete, each part or each profile of a part uses only a subset of all operations. The motivation and design of each part will be discussed in section 5 in more details, yet it is important to note that the full decoder is constructed from *four elementary operations* performed on a pixel-by-pixel basis:

- Scalar *non-linear transformations* perform either a table look-up using the input value as index into the table, or scale the input to the range $[0, 1]$, apply a selection of parametrized functions such as linear ramps or gamma corrections, and then scale the output of such functions back to the output range. In Fig. 2, such scalar non-linear transformations are depicted by boxes with rounded edges.

- The second type of transformations are multiplications by 3×3 matrices that operate again on a pixel-by-pixel basis, but across components or channels. These transformations either perform inverse component decorrelation transformations such as the transformation from YCbCr to RGB, or change the colorspace from one to another. They are depicted in Fig. 2 by square boxes.
- A *vector addition* indicated by a circled plus-sign, combining base and residual images.
- A *scalar multiplication* that optionally scales all components by a common factor, represented by a circled dot.

All parts and profiles of JPEG XT are constructed from these four operations, and they are always applied in the order given in Fig. 2. Which of the operations is applied, however, and which are bypassed, is a matter of the specification for a given profile in the standard. A bypassed non-linear transformation will just scale the bit-range of the input to the bit-range of the output, a bypassed linear transformation performs no operation, i.e. uses the identity matrix. Hence, all parts of JPEG XT can be implemented by one common code, replacing those boxes that are not present by their corresponding default operations.

It is not by accident that some of the data flows are similar to ICC profiles[23] that also convert between differing colorspaces. A *Matrix Based ICC Profile* also performs first a non-linear transformation, then a matrix multiplication, and then another non-linear transformation. This combination of operations can be clearly identified both in the base as well as in the residual decoding path of Fig. 2.

The standard currently does not allow to deviate from the order of operations, i.e the data flow from the decoded and inversely DCT transformed data to the output is always identical, and always as in Fig. 2, for all parts and all profiles. It may become necessary for future applications to include additional elements in the common decoder architecture, and hence to include a description of the decoder chain within the JPEG XT chain.

4. TRANSPORT AND ENCODING

JPEG XT is designed as a backward compatible extension to the JPEG legacy standard.[3] Thus, all additional data has to be encoded non-intrusively for legacy implementations. Luckily, the JPEG standard already defines a mechanism for including extension information by so-called *application markers*. Sixteen of such markers are available, and each marker segment can carry up to 64K of payload data. JPEG XT employs application marker 11 (APP_{11}) to both carry the entropy coded data of refinement and residual scans, and the meta-data that defines all the parameters for merging base and residual images together.

With the exception of JPEG XT part 2, which uses a text-based syntax, all parts rely on a generic extension derived from the ISO-based media format that is jointly defined by JPEG and MPEG. Part 7 profile A then replicates Part 2 in this common syntax.

The ISO-based media format, also applied for example in the JPEG 2000 file format, is structured around a common syntax element called a *box*. Boxes have both a length and a type field such that decoders that are unaware of a particular box type and its purpose can skip over them. Boxes contain either structured data, with the structure defined by the standard, entropy coded data, or other boxes. The latter boxes are also referred to as *super boxes*.

Unlike JPEG 2000 and MPEG file formats, which encapsulate entropy coded image or video data into one or more particular boxes, JPEG XT has — for legacy reasons — to use a different transport mechanism. Within JPEG XT, the legacy codestream carries application markers which again carry the box information. Due to the size constraint of application markers, a single box may here, however, extend over several application markers of the same type, and the box syntax has been extended by indicators that tell the decoder how to assemble one or several application markers back together into one single box.

5. JPEG XT HDR PROFILES AND ENCODING STRATEGIES

Even though all parts and profiles share the same common decoder architecture, the underlying mathematical models for representing images are quite different. For the sake of brevity, this section will only discuss HDR coding, i.e. part 7[14] of JPEG XT, and we will mention in how far the fundamentals carry over to other parts when necessary. For the discussion, recall that a fundamental design principle of JPEG XT is the backward compatibility with the JPEG legacy standard, i.e. a JPEG XT stream necessarily includes an 8-bit per sample LDR codestream called the *base image*.

Part 7 of JPEG XT defines a decoder for an HDR image, i.e. an image whose dynamic range is too large for allowing a sample representation based on integers. Typically, sample values are here encoded in floating-point, where the sample value is often assumed to be proportional to the physical radiance at a specific point in the scene. If the proportionality factor is known, the image is called to be in *absolute radiance*, otherwise in *relative radiance*.

Such images are typically acquired by taking several pictures of the same scene and varying the exposure time from image to image. A merging algorithm then attempts to approximate the camera answer function, i.e. the relation between the sample values as returned by the camera, and the radiance in the scene. The discussion of such algorithms is beyond the scope of this paper and the reader is referred to the literature.[20]

Part 7 defines three profiles, each of which requires a subset of the functional blocks of the common decoder design of Fig. 2. For the sake of simplicity, this explanation assumes that the colorspace of the base image is identical to the colorspace of the HDR image, i.e. the coordinates of the color primaries of both colorspaces are considered identical.

5.1 Profile A

The decoding algorithm of profile A of part 7 is identical to that of part 2, except that the syntax representing meta-data is different. The HDR image is here reconstructed as

$$\mathrm{HDR} = \mu\left(\Phi(\mathrm{LDR}) + \chi\right) \tag{1}$$

where HDR is the reconstructed HDR image, LDR is the base image in the legacy codestream, Φ is an inverse gamma-correction, χ a color-residual and μ a position-dependent scale factor.

The guideline of this algorithm proposes that the LDR image is typically encoded in a gamma-corrected space, i.e. sample values are not proportional to radiance, but proportional to a power function of the radiance. The power function, actually stemming from analog TV standards, can be explained as a simple form of background luminance adaption of the human eye and can be easily derived from the Naka-Rushton equation.[20] The Φ function linearizes the non-linear LDR image and hence obtains a signal proportional to physical radiance. Because the dynamic range of the HDR image is, however, several magnitudes larger than the dynamic range of the LDR image, it requires scaling by the factor μ that varies spatially from pixel to pixel. Finally, χ is a color residual that adapts between the chrominance of the LDR image and the chroma signals of the HDR image and takes care of small chroma deviations between them.

In the common decoder design, Φ is represented by the *Base NLT point trafo*, and μ is computed through an exponential ramp from the luminance of the residual image RES_0, i.e.

$$\mu = \exp\left(a\mathrm{RES}_0 + b\right) \tag{2}$$

This exponential function is represented by the *Postscaling NLT point trafo* in Fig. 2.

The chroma residual, finally, is encoded in the chroma signal of the residual image and is scaled into the luminance range of the linearized base image by means of the pre-scaling factor ν:

$$\chi = \nu R \, \mathrm{RES}_{1,2} \tag{3}$$

Here $\text{RES}_{1,2}$ denotes the chroma signal of the residual image and R is an affine transformation computing an RGB signal from the YCbCr signal encoded by the residual codestream. It is represented by the box denoted *Residual Transformation* in Fig. 2.

The factor ν is now a function of the luminance of the linearized base image:

$$\nu = N_f + S\Phi\left(\text{LDR}\right) \tag{4}$$

As above, Φ is the inverse gamma correction, and S the projection onto the luminance component of its argument, i.e. it is the first component of the RGB to YCbCr transformation of its input signal. Finally, N_f is a small value called the *noise floor* that avoids a divergence in the encoder for small luminance signals.

Within the standard, S is represented as the *Prescaling transformation*, and the addition of N_f by the *Prescaling NLT transformation*. Secondary NLT transformations and the output conversion are not required for this decoder. The reader may now verify with Fig. 2 that, indeed, the decoding algorithm of Profile A is representable with all functional elements of the standard.

Inverting this decoder, i.e. finding the corresponding encoder, is somewhat more complex. The input of the encoder is an LDR, HDR image pair from which a suitable residual image must be computed. The LDR image is always directly represented by the legacy codestream (top left box in Fig. 2). For that, first note that $R\,\text{RES}$ is, by construction, a pure chrominance signal and does not include any luminance information. Thus, if we denote by P_l the projection onto luminance space, one finds that

$$P_L\Big(\Phi\big(\text{LDR}\big) + \nu R\,\text{RES}\Big) = P_L C\Phi\big(\text{LDR}\big) \tag{5}$$

This allows the encoder to determine first $\mu(\text{RES}_0\big)$ and then, by inversion of $\mu(x) = \exp(ax+b)$, compute RES_0 as the quotient of reconstructed base image luminance and original luminance image:

$$\mu\big(\text{RES}_0\big) = \frac{P_L\Phi\big(\text{LDR}\big)}{P_L\text{HDR}} \tag{6}$$

In the last step, the chroma residual χ and by that the chroma components of the residual image are derived by:

$$R\,\text{RES} = \frac{1}{\nu\big(S\text{LDR}\big)}\left(\frac{\text{HDR}}{\mu\big(\text{RES}_0\big)} - \Phi\big(\text{LDR}\big)\right) \tag{7}$$

One can observe that the right-hand side lies indeed in the chroma-subspace of the color space, i.e. the luma signal is zero (or close to zero, due to numerical inaccuracies). It accommodates the logarithm of the luminance scale factor instead. It should now also become clear why ν may not become zero, or why the addition of N_f is necessary to stabilize the encoder.

5.2 Profile C

At this point, it is more convenient to discuss profile C first and proceed to profile B later: While profile A scales all components by one single factor μ, profile C can be understood as a component wise scaling. That is,

$$\text{HDR}_i = \Phi\left(\text{LDR}\right)_i \text{RES}_i \quad (i = 0, 1, 2) \tag{8}$$

Profile C however expresses this relation in the logarithmic space:

$$\text{HDR}_i = \exp\left(\log\left(\Phi\left(\text{LDR}_i\right)\right) + \left(\log\left(\text{RES}_i\right)\right)\right) \tag{9}$$

One can observe that it is possible to merge $\log\Phi$ into a single non-linear function $\hat{\Phi}$ that is represented as *Base NLT Transformation* in the common decoder architecture, and since the residual image is never observed directly, one can encode its logarithm directly, i.e. there is no need to replicate the logarithm in the decoder. Last but not least, exponential and logarithmic function are approximated by piece-wise linear functions that are exactly invertible. This gives the following simple reconstruction algorithm:

$$\mathrm{HDR}_i = \psi\exp\left(\hat{\Phi}\left(\mathrm{LDR}_i\right) + \mathrm{RES}_i - O\right) \tag{10}$$

Here, $\psi\exp$ is the pseudo-exponential, a piece-wise linear approximation of the exponential function which is represented by the *Output Conversion* in Fig. 2, and O is an offset that ensures that the residual image is non-negative. Post or pre-scaling non-linearities are not required in this profile.

Encoding in this profile is trivial: Given $\hat{\Phi}$ is known, one only has to compute the difference between the (pseudo)-logarithm of the HDR input image and $\hat{\Phi}$ (LDR). Even though one could pick $\hat{\Phi}$ as introduced above, i.e. as the composition of an inverse gamma transformation followed by a approximate logarithm, profile C encoders typically try to find an approximate global inverse of the tone-mapping function between HDR and LDR image, thus trying to minimize the residual as part of the problem for determining the function $\hat{\Phi}$.

Part 6 of JPEG XT is related to this profile, its reconstruction algorithm is identical to that of Eqn. 10 except that the data is not encoded in the logarithmic domain, i.e. the pseudo-exponential function is dropped and the output consists always of integer samples. Here the IDR output image is the sum (and not the product) of the base image and the extension layer.

5.3 Profile B

Profile B operates similar to Profile C except that it expresses the HDR image as a component-wise quotient instead of a component-wise product:

$$\mathrm{HDR}_i = \sigma\frac{\Phi\left(\mathrm{LDR}_i\right)}{\Psi\left(\mathrm{RES}_i\right) + \epsilon} \quad (i = 0, 1, 2) \tag{11}$$

Similar to the working of profile C, the standard operates in a logarithmic representation, using the exact logarithm and not a piece-wise linear approximation:

$$\mathrm{HDR}_i = \sigma\exp\left(\log\left(\Phi\left(\mathrm{HDR}_i\right)\right) - \log\left(\Psi\left(\mathrm{Res}_i\right)\right)\right) \tag{12}$$

The Φ function is as in profile A an inverse gamma correction and represented by the *Base NLT Transformation*. The outmost exponential function together with the scale σ is defined by the *Output Conversion*, and the logarithms within the exponential by the *Secondary Base* and *Secondary Residual NLT Trafo*. Finally, Ψ is the *Intermediate Residual NLT*. It is typically selected to be a gamma correction whose exponent is image dependent and derived by the encoder.

Similar to profile C, profile B does not require the pre- and post-scaling maps, but it is the only profile that depends on the *Secondary Base* and *Secondary Residual NLT Trafo*. A practical decoder implementation would, of course, never go through the exponential and logarithmic functions but would evaluate the quotient directly.

While the profile B algorithm allows for arbitrary LDR images and hence for arbitrary tone-mapping of the HDR content, it is motivated by a particular choice of the Φ map and the LDR image, namely:

$$\mathrm{LDR}_i = \min\left(\Phi^{-1}\left(\sigma^{-1}\mathrm{HDR}\right)_i, 1\right) \tag{13}$$

where Φ^{-1} is the gamma correction from HDR to LDR image. Hence, the HDR image is first inversely scaled with σ, then gamma-corrected and clamped to the range $[0, 1]$. One observes that this has the following consequences for the residual: As long as the sample value of the HDR image is below σ, no clamping takes place and the quotient defining the residual image

$$\Psi\left(\rho\left(R\,\mathrm{RES}_i\right)\right) = \sigma\frac{\Phi\left(\Phi^{-1}\left(\sigma^{-1}\mathrm{HDR}\right)\right)_i}{\mathrm{HDR}_i} = 1 \tag{14}$$

becomes maximal, i.e. +1. For larger HDR sample values, the LDR image is saturated, and the residual carries all image information on the overexposed image. The purpose of σ becomes obvious now: It determines which image luminances go into the LDR image, and which become part of the residual image, i.e. it controls the *exposure* of the HDR image into an LDR image by means of gamma correction and clamping. Its inverse, σ^{-1}, is therefore called the *Exposure Value*. One should still note, however, that this represents only one particular choice of the coding parameters and JPEG XT allows much more flexible arrangements for the non-linearities.

6. LOSSLESS ENCODING WITH PART 8

The part 8 decoder is derived from profile C of part 7. For that, recall that the part 7 profile C reconstruction algorithm reads

$$\mathrm{HDR}_i = \psi \exp\left(\hat{\Phi}\left(\mathrm{LDR}_i\right) + \mathrm{RES}_i - O\right) \tag{15}$$

where $\hat{\Phi}$ is implemented as a look-up table. Hence, except for the final output conversion by $\psi \exp$, the piece-wise linear approximation of the exponential function, all operations are integer-based. The former function is, however, simply the casting of the bit-pattern of a 16-bit binary integer representation to a 16-bit IEEE half-float representation, and has, hence, an exact inverse $\log \exp$. With the applied DCT to the legacy codestream fully specified, the encoder can now exactly predict the expression $\hat{\Phi}\left(\mathrm{LDR}_i\right)$ that a compliant decoder would use for reconstruction. This allows the encoder to compute the error term RES_i necessary for lossless representation.

Part 8 specifies, for this end, a DCT implementation the decoder has to follow for the base image reconstruction. To avoid any loss in the coding of the residual signal RES_i, the DCT is here bypassed and replaced by the identity, i.e. the residual signal is encoded by the JPEG-type zig-zag Huffman scan in the spatial domain. To encode integer samples without loss, the $\psi \exp$ map in (15) is bypassed.

7. CODING OF OPACITY INFORMATION

As mentioned before, the coding of alpha channel information is specified in JPEG XT part 9, which specifically extends part 3 to provide the capability for lossy and lossless storage of continuous opacity information for the associated image. Such additional channels are commonly known as alpha channels and allow compositing the actual image content with other image content. An alpha value of zero represents maximal transparency – i.e. no opacity – and effectively hides the image content at its respective spatial location. A maximal alpha value, on the other hand, represents maximal opacity – i.e. no transparency – at its respective spatial location.

Additionally, part 9 allows for the encoded image content to be pre-multiplied with the alpha value or pre-multiplied and blended with an arbitrary selected background color, M. The first option might improve the compression performance, as sample values are adjusted before encoding to reflect their respective contributions to the reconstructed image. In fact, pre-multiplication will even zero out completely transparent samples. Moreover, the second option allows for improved legacy decoding fall-back behavior, as the legacy LDR image is in this case a blended representation on a matte background with color M instead of black. Thus, decoding of the base image with a legacy decoder will create the appearance that the reconstructed image is composed on a matte background with color M. On the other hand, compliant JPEG XT decoders can undo the blending by removing the shading color, M, and subsequently composite the image on any background. Formally, let A be the original image, and A' the actual encoded image content in the JPEG XT code-stream, then

$$A' = \alpha \times A \text{ for pre-multiplication,}$$
$$A' = \alpha \times A + (1 - \alpha) \times M \text{ for pre-multiplication with shading.}$$

Figure 3. Two of the images from the test image set: Left “WillyDesk”, right “BloomingGorse2”. Here only the tone-mapped LDR versions are shown to facilitate printing.

Then, blending the reconstructed image on a background image, B, will result in the composed output image, C. This process is defined by

$$C = \alpha \times A + (1 - \alpha) \times B \text{ for regular content,}$$
$$C = A + (1 - \alpha) \times B \text{ for pre-multiplied content, and}$$
$$C = A + (1 - \alpha) \times (B - M) \text{ for pre-multiplied content with shade removal.}$$

Part 9 provides the facilities to encode alpha values for each spatial location, with or without loss, either as a continuous scale of integers with bit-depths between 8 and 16 bits, or as 16-bit precision floating-point values between zero and one. The standard relies exclusively on coding technology from the other parts of JPEG XT to encode opacity information. Furthermore, it can be used in combination with any of the other parts.

8. PERFORMANCE OF JPEG XT

For the sake of brevity, we will only discuss the performance of JPEG XT part 7 in this paper. JPEG internal tests have shown that part 8, the lossless extension, performs approximately as well as PNG, but slightly worse than JPEG 2000 or JPEG LS, though the differences are minor and all lossless image compression algorithms we investigated provided an approximate 2:1 bit-rate reduction.[27]

Evaluating part 7 is in so far a challenging problem as the subjective and objective evaluation of HDR image fidelity is still at its infancy. Current recommendations for subjective quality evaluation only cover standard dynamic range and would require, if blindly carried over to HDR, viewing conditions that are unpractical and unrealistic, e.g. back-light luminances that are much too high. For objective assessment, the situation is even worse, because the peak signal to noise ratio (PSNR) – which is already known to be only a very rough approximation of subjective image fidelity for standard dynamic range – becomes outright unusable for high dynamic range images. This is because the human visual system is non-linear, and the non-linearity cannot be reasonably approximated by a linear function over a large dynamic range input. Several objective quality indices have been proposed in the past, including measuring the relative error,[26] or first mapping the HDR sample values to a perceptually uniform space and then applying known indices such as PSNR or SSIM[25] in this space. In this work, we adopt HDR-VDP-2[28] as an intrinsic HDR metric that includes a full (non-linear) model of the first processing stages of the human visual system (HVS). Correlation tests with subjective experiments performed internally by the JPEG group have shown that HDR-VDP-2 is superior to other indices quoted above.

Due to the non-linearity of the HVS, proper calibration of the input images is here of vital importance. HDR-VDP-2 requires absolute radiance information, even though most HDR images obtained by multiple-exposure plus merging[20] provide only relative radiance information. The input images used in this work

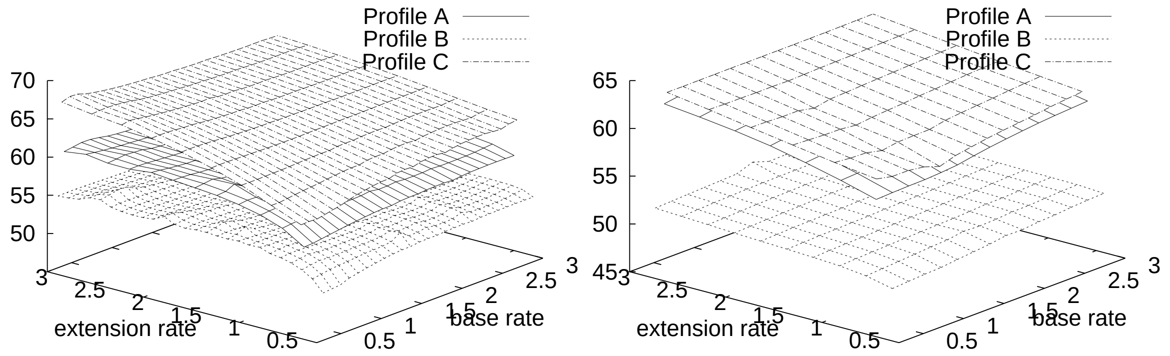

Figure 4. Rate-distortion performance of JPEG XT as HDR-VDP-2 quality over the base and extension layers rates, on two example images with unconstrained base quality. Left the WillyDesk image of an indoor scene, right the BloomingGorse2 image, which is structurally very rich and challenging.

were obtained from Mark Fairchild's Photographic Survey,[21] and were then calibrated to an estimated peak radiance of 10000 nits by the pfstools package.[22] To reduce computational complexity when performing HDR-VDP-2, images were down-scaled by a factor of four in each dimension before compression. Two typical test images are shown in Fig. 3.

Despite the challenges of finding a reliable fidelity index, JPEG XT is by design hard to evaluate. For that, recall that a JPEG XT encoder requires an image pair, LDR and HDR images, as input, and encodes the images as two layers: (1) a base layer representing the LDR input, reconstructible by any JPEG legacy compliant decoder, and (2) an extension layer providing the data necessary to extend the dynamic range to the full range. The parameter space for the image quality is hence (at least) two-dimensional, one parameter for the base and another parameter for the extension layer. Even more so, the LDR image can be selected purely by artistic intent* as the relation between LDR and HDR images is not constrained by the standard. Typically, the LDR image is, however, a tone-mapped version of the HDR image, where both global and local operators are possible. Details on tone-mapping and possible tone-mapping operators are discussed in the literature.[20]

For evaluations, we restricted ourselves to a single tone-mapping operator, namely, one built into Adobe Photoshop, with parameters adjusted to generate a visually pleasing LDR image. Base and extension layers quality were changed between two extremes 0 and 100 in steps of two, and bit-rates of base and extension layers were measured for each LDR, HDR quality pair. Hence, the overall data-set consists of the HDR-VDP-2 Q value (estimated quality) as a two-dimensional function of the base and extension layers rates, resulting in plots as depicted in Fig. 4.

This two-dimensional data-set was used to generate a rate-distortion curve as follows: For the first experiment, an algorithm identified the best possible combination of base rate, i.e. LDR image quality, and extension layer rate for a given target bit-rate. In this specific configuration, see Fig. 5, the quality of the LDR image is allowed to vary unconstrained, and may become unacceptably low for some combinations. In the second experiment, the base quality was constrained to be at least 75 or better, ensuring an acceptable to good LDR quality. Otherwise, the same algorithm was used to find the ideal LDR, HDR quality combination. Results are in Fig. 6, again showing HDR-VDP-2 Q values over the bit-rate. Both plots also include JPEG 2000 and JPEG XR as single-layer HDR compression algorithms, where the former uses a floating-point extension standardized in AMD 3 of ISO/IEC 15444-2. Interestingly, JPEG XT can both outperform JPEG 2000 and JPEG XR for some images, when HDR-VDP-2 is used as metric to predict quality.

Finally, one may wonder how much overhead the additional HDR content represents in a JPEG XT codestream, when compared to its LDR. The ratio of base layer rate to total bit-rate is plotted in Fig. 7,

*In extreme cases, the LDR image can even look like an entirely different image.

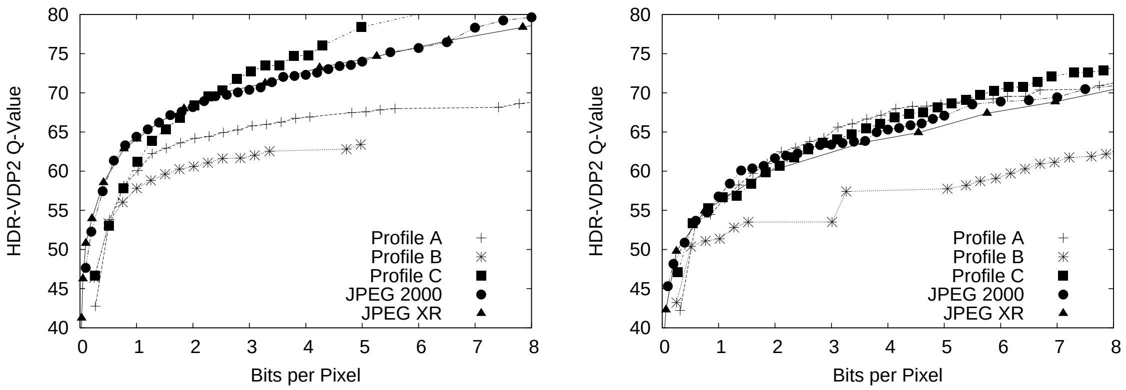

Figure 5. Rate-distortion performance of JPEG XT on WillyDesk (left) and BloomingGorse2 (right) when selecting the best possible base and extension layer configuration to reach the desired bit-rate from the two-parameter space shown in Fig. 4.

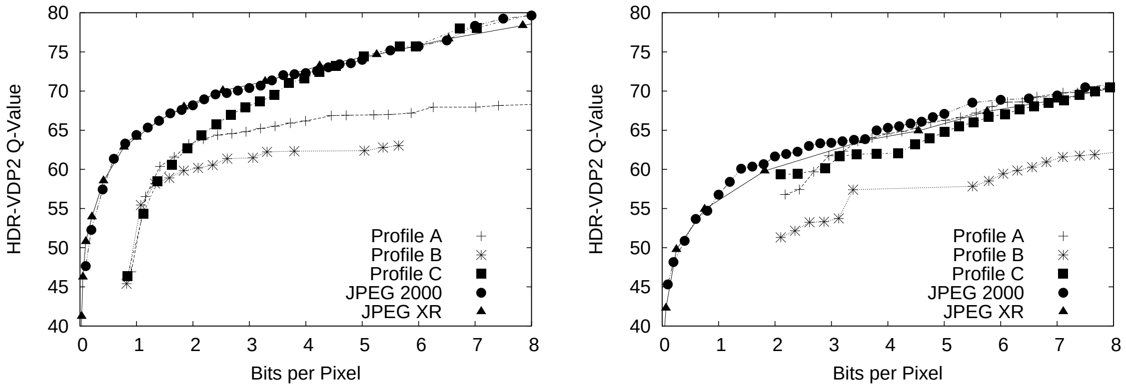

Figure 6. Rate-distortion performance of JPEG XT on two example images from the HDR image test set with base quality constrained to be above 75. Left WillyDesk, right BloomingGorse2.

again with a 75 base quality constraint. While the overhead may be as large as 80% for low bit-rates and hence low qualities, it becomes negligible for higher bit-rates. The overall bit-rate dependency is quite smooth for profile C, but shows irregularities for the two other profiles. This is an artifact of the method generating the plots and not a defect of the profiles. It is due to the almost horizontal rate-distortion surfaces of profiles A and B as observed in Fig. 4 which causes the maximum-detector to become unstable. By allowing the quality to decrease slightly, it would be possible to select a more regular base/extension rate curve in the two-dimensional configuration parameter set and hence regularize the overhead as a function of the overall-rate.

9. FUTURE PLANS AND POTENTIAL EXTENSIONS

In the next section, we briefly discuss a new standardization activity called JPEG Privacy & Security[33] which can be considered as an extension of the JPEG XT standard and fits well into the existing framework. The following sections then give an outlook into potential future JPEG initiatives. While the former is already a concrete activity of JPEG, the ideas described in sections 9.2 and on are currently purely academic in nature and not yet part of any official JPEG standardization trajectory. For more information and recent updates on these activities we refer to `www.jpeg.org`.

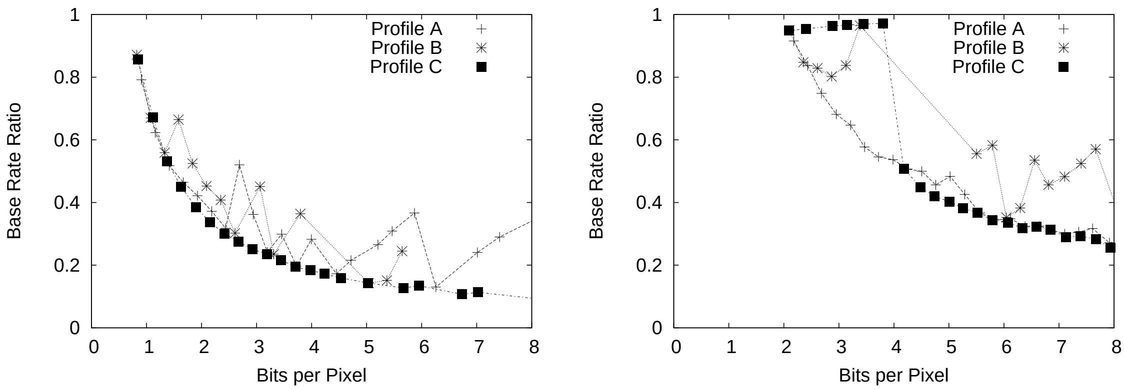

Figure 7. Ratio of the base layer rate to the total bit-rate for the three JPEG XT profiles, left for WillyDesk, right BloomingGorse2, again with the base quality constrained to be above 75.

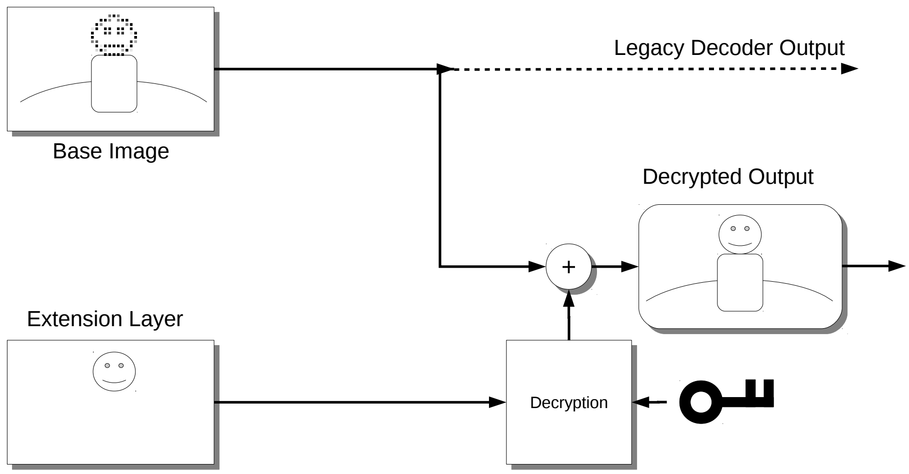

Figure 8. Envisioned JPEG privacy decoder based on the two-layer design of JPEG XT. The base layer contains only a pixelated version of the face, the encrypted extension layer contains the correction information to allow displaying the full data.

9.1 JPEG Privacy & Security

Privacy and security support for image data is becoming steadily more important given the fact that image collections are increasingly stored in distributed ways and on cloud rather than in private repositories. Moreover, social media — inherently impacting the privacy of the associated persons — are currently offering insufficient means to secure the privacy-sensitive information carried by the picture and/or associated metadata. Observing that on a daily basis billions of pictures are shared in JPEG legacy format, it is evident that embedding additional security and privacy functionalities that would contribute to better safeguard this type of information would benefit a significant user base.

While we can encrypt arbitrary data – text and images – such that only authorized persons can get access to the content, it is also desirable to enable a finer graded control especially on image content. As said, typical applications for such extensions would be social networks where, for example, the general public would only receive images with scrambled or blurred faces, while the entire image would be accessible by only a restricted audience with appropriate credentials. A second application would be that of image databases offering a lower quality, lower resolution preview of images for free, while by buying an additional authorization token,

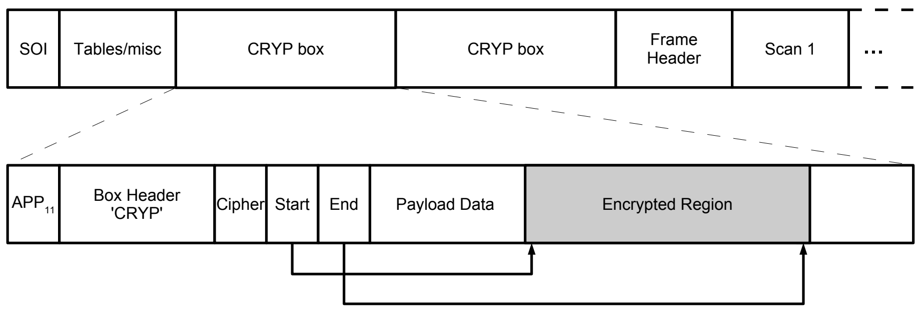

Figure 9. Illustration of a potential JPEG codestream syntax equipped with JPEG Privacy & Security functionality.

the full image could become available without requiring the client to download the image a second time.

In the context of JPEG 2000, a significant part of the required functionality for privacy and security support was already embedded in Part 8 of this standard: ISO/IEC 15444-8 | Recommendation ITU-T T.808.[29] However, due to the limited adoption of JPEG 2000 by the consumer market, this part never got the track it deserved.

The particular focus of the JPEG Privacy & Security activity is to provide codestream and file format syntax support to extend security and privacy functionalities both similar to JPEG 2000, for all JPEG standards. Although a particular focus is put on JPEG legacy, also other standards such as JPEG XT, JPSearch and JPEG Systems are within the scope of this activity.

The foundations for implementation of necessary extensions to cope with Privacy & Security in JPEG legacy are to a large degree already present in JPEG XT. For that, recall that JPEG XT separates the image into a base layer available to all legacy decoders and an extension layer only visible to applications conforming to the new standard (see Fig. 2). The two layers are recombined by a simple addition, thus any defect in the base image can be compensated by a residual signal in the extension layer. In a simple extension of JPEG XT, it would be only necessary to encrypt the extension layer. In the simplest possible realization, as base and extension layer are 8 bits per component images, both encoded in JPEG — such an arrangement is already covered by JPEG XT part 6. Image regions could be pixelated or of lower quality in the base layer, while the extension layer would carry the differential signal between intended full-quality image and the degraded base image, see Fig. 8.

As the extension layer is encapsulated in markers carrying boxes along with all other meta-data, the JPEG privacy extension would only need to specify a method to encrypt selected markers or subsets of markers with a suitable cipher; at the very same time additional meta-data, e.g. EXIF or IPCT data, could be encrypted along with the image data. Examples of extensions that enhance JPEG legacy with privacy and security features have been reported in our works.[36–38]

9.2 JPIP: Interactive Image Browsing and Delivery

As the sensor resolution of digital cameras still increase, image dimensions grow larger and larger by year. While it is useful to keep images unaltered in their original size on disk and in databases, browsing through a large collection of images would then require significant bandwidth while lower resolution, smaller images are usually sufficient as previews. While servers could either downscale images themselves or store an additional image index consisting of smaller thumbnails, such approaches either cost storage or computing capacity. A client-server architecture allowing clients to specify exactly the necessary image regions and servers to transmit only the minimal data to satisfy such requests is, however, a more elegant approach to address this problem.

Both proprietary and standardized solutions for this application already exist: "Google Maps" and "Google Earth" are two proprietary implementations of such a protocol, JPIP — formally known as ISO/IEC 15444-9[30] — a standard based on JPEG 2000 and a `http` request syntax.

A possible extension of JPIP would be to allow similar interactive image browsing on the basis of the older but much more popular JPEG legacy format and the more recent JPEG XT format. Down-scaling and cropping can be addressed by only transmitting the necessary frequencies or by selecting the requested DCT blocks and trans-coding the image on the server. In the context of JPEG XT, mechanisms allowing the selection of base and residual layers could be incorporated. Moreover, compliance with the JPEG Privacy & Security framework should be envisioned as well.

Modifications to the current JPIP standard would be minimal: JPIP is already a very generic format by design, though the data transport layer would require extension to be applicable to DCT transformed data. The specification of image regions and image resolutions as well as the request syntax is already independent of the actual encoding and would not require any change.

9.3 Recording Image Editing Operations

It is not uncommon that images undergo multiple editing steps before they are published, printed or transmitted, especially in professional applications. At the same time, it is also desirable to record the editing history, the individual outcome of each step, hence allowing the possibility to revert each individual step.[35] The envisioned system may very well fit into the JPEG Privacy & Security initiative as it would allow an image producer to transparently signal the entire image generation from acquisition, editing to reproduction. For that, the entire image production chain would need to be secured and certified by means of tools provided by JPEG Privacy & Security.

Image manipulation programs define as of today proprietary formats to store image edits which include the original image along with all the manipulation steps to generate the end-result. At the same time, the image processing tools offered by such applications (e.g. "sharpen", "emboss", "deskew") are usually proprietary as well. To enable an application independent standard, it is hence not sufficient to store the meta-data describing the editing operations.

The two-layer nature of JPEG XT could offer a remedy again: To illustrate the idea (cf. Fig 10), let us first consider only the case of a single editing step: While the precise implementation of an image processing algorithm is probably unknown, the intended JPEG extension would include definitions of various "prototype" operations. While the standard would fully specify these operations, their outcome would likely differ from the outcome of a proprietary implementation of the same or a similar operation. The envisioned JPEG standard would now store the meta-data describing these operations along with a differential image that encodes the difference between the outcome of the standardized algorithm and the output of a specific implementation, see Fig. 10. An example of extension of JPEG legacy format to allow for the above mentioned functionality is provided in our references.[38]

Trivial cascading of this design allows to support multiple editing steps without changing the design principles.

10. CONCLUSIONS

In this paper, we provided an overview on JPEG XT that targets the expansion of the JPEG functionality by enabling support for high dynamic range imaging, lossless and near-lossless coding, and alpha channel coding, while also guaranteeing backward and forward compatibility with JPEG legacy decoders. Moreover, principles behind a recently initiated standardization effort called JPEG Privacy & Security[33] have been briefly described and illustrated. Finally, potential further extensions of the JPIP framework to handle JPEG XT images is provided, along with a proposal to signal image editing operations in the JPEG and JPEG XT files.

Interested parties may get an overview on the current JPEG work-items and initiatives at `www.jpeg.org`. We especially point readers to the JPEG XT reference software at `www.jpeg.org/jpegxt/software.html` to test and validate our latest standard.

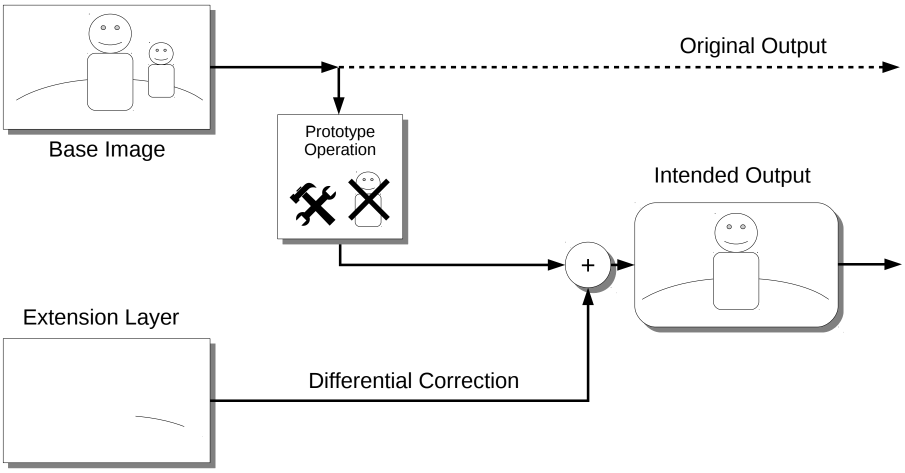

Figure 10. Emulating a single editing operation by a prototype and a differential image.

ACKNOWLEDGMENTS

The authors wants to thank the Computing Center of the University of Stuttgart and the iMinds project Bahamas for funding this work. Peter Schelkens has received funding from the European Research Council under the European Unions Seventh Framework Programme (FP7/2007-2013)/ERC Grant Agreement n.617779 (INTERFERE). Touradj Ebrahimi has received funding from the European Union Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 643072, Network QoE-Net.

REFERENCES

[1] G.K. Wallace: "The JPEG still picture compression standard", IEEE Trans. on Consumer Electronics, Vol. 38, Issue 1, pp. xviii - xxxiv, (1992)

[2] William B. Pennebaker, Joan L. Mitchell: "JPEG Still Image Data Compression Standard" *Van Nostrand Reinhold, New York* (1992)

[3] ISO/ITU: "Information Technology — Digital Compression and Coding of Continuous-Tone Still Images — Requirements and Guidelines", published by ISO as 10918-1 and ITU as T.81, (1992)

[4] D. Taubman: " High performance scalable image compression with EBCOT", *IEEE Transactions on Image Processing*, Vol. 9 No. 7, pp. 1151-1170, (2000)

[5] S. Srinivasan, C. Tu, Zhi Zhou, D. Ray, S. Regunathan, G. Sullivan: "An Introduction to the HDPhoto Technical Design", JPEG document WG1N4183, (2007)

[6] E. Hamilton: "JPEG Interchange File Format", `http://www.w3.org/Graphics/JPEG/jfif3.pdf`, retrieved 2015

[7] B. Brower, A. Hinds, E. Hamilton, D. Lee, G. Sullivan: "Information Technology: Digital compression and coding of continuous-tone still images: JPEG File Interchange Format (JFIF)", published as ISO/IEC 10918-5 | ITU-T T.871, ITU (2011)

[8] ISO | ITU: "Information Technology: Digital compression and coding of continuous-tone still images: Extensions", ISO/IEC 10918-3 | ITU-T T.84 (1996)

[9] W. Husak, Th. Richter: "Information Technology: Scalable Compression and Coding of Continuous-Tone Still Images, Core Coding System Specification", ISO/IEC 18477-1, ISO/IEC (2014)

[10] Th. Richter, W. Husak, A. Niman, A. Ten, P. Korshunov, T. Ebrahimi, A. Atrusi, M. Agostinelli: "Information Technology: Scalable Compression and Coding of Continuous-Tone Still Images, Extensions for High-Dynamic Range Images", ISO/IEC 18477-2, ISO/IEC (2014)

[11] Th. Richter, P. Schelkens, T. Ishikawa: "Information Technology: Scalable Compression and Coding of Continuous-Tone Still Images, Box File Format", ISO/IEC 18477-3 (under publication), ISO/IEC (2015)

[12] Th. Richter, A. Ten, A. Artusi: "Information Technology: Scalable Compression and Coding of Continuous-Tone Still Images, Reference Software", ISO/IEC 18477-5 (under preparation)

[13] Th. Richter, S. Ogawa: "Information Technology: Scalable Compression and Coding of Continuous-Tone Still Images, IDR Integer Coding", ISO/IEC 18477-6 (under preparation)

[14] Th. Richter, A. Artusi,M. Agostinelli: "Information Technology: Scalable Compression and Coding of Continuous-Tone Still Images, HDR Floating-Point Coding", ISO/IEC 18477-7 (under preparation)

[15] Th. Richter, S. Ogawa: "Information Technology: Scalable Compression and Coding of Continuous-Tone Still Images, Lossless and Near-Lossless Coding", ISO/IEC 18477-8 (under preparation)

[16] Th. Richter, T. Bruylants, "Information Technology: Scalable Compression and Coding of Continuous-Tone Still Images, Encoding of Alpha Channels", ISO/IEC 18477-9 (under preparation)

[17] Th. Richter: "Backwards Compatible Coding of High Dynamic Range Images with JPEG", Proc. of Data Compression Conference 2013 (DCC), pp. 153-160, (2013)

[18] Th. Richter: "Lossless Coding Extensions for JPEG", Proc. of Data Compression Conference 2015 (DCC), pp. 143-152, (2015)

[19] T. Boutell: "PNG (Portable Network Graphics) Specification Version 1.0", Network Working Group RFC 2083, available online at `http://tools.ietf.org/html/rfc2083` (retrieved November 2014).

[20] E. Reinhard, W. Heidrich, P. Debevec, S. Pattanaik, G. Ward, K. Myszkowski: "High Dynamic Range Imaging, Acquisition, Display and Image-Based Lighting" (2^{nd} Ed.), Morgan Kaufman (2010)

[21] M. Fairchild: "The HDR Photographic Survey", available online at `http://rit-mcsl.org/fairchild/HDR.html` (retrieved July 2015)

[22] R. Mantiuk: "pfstools: High Dynamic Range Images and Video", available online at `http://pfstools.sourceforge.net/` (retrieved July 2015)

[23] International Color Consortium: "Image technology colour management — Architecture, profile format, and data structure", published as ICC.1:2010, profile version 4.3.0.0 and ISO 15076-1 (2010)

[24] Th. Richter: "JPEG XT Reference Codec 1.31 (ISO License)", available online at `http://www.jpeg.org/jpegxt/software.html`, (retrieved July 2015)

[25] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli: "Image quality assessment: From error visibility to structural similarity", *IEEE Trans. on Image Proc.*, Vol. 13, No. 4, pp. 600-612, (2004)

[26] Th. Richter: "Evaluation of floating point image compression", Proc. of Quality of Multimedia Experience (QOMEX 2009), Intl. Workshop on, pp. 222-227, (2009)

[27] Th. Richter: "Lossless Coding Extensions for JPEG", Proc. of Data Compression Conference (DCC), pp. 143-152, (2015)

[28] R. Mantiuk, K.J. Kim, A.G. Rempel, W. Heidrich: "HDR-VDP-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions", ACM Trans. on Graphics (Proc. of SIGGRAPH'11), 30(4), article no. 40, (2011)

[29] V. Conan, T. Ebrahimi, C. Rollin, S. Wee: "Information Technology — JPEG 2000 image coding system: Secure JPEG 2000", published as ISO/IEC 15444-8, ISO, (2007)

[30] R. Prandolini, G. Colyer, S. Houchin: "Information Technology — JPEG 2000 image coding system: Interactivity tools, APIs and protocols", ISO/IEC 15444-9, ISO, (2005)

[31] P. Schelkens, F. Pereira, T. Ebrahimi, J. Kleinert, A. de Polo, M. Domanski (Eds.): "JPEG PLENO Workshop Proceedings", ISO/IEC JTCS/SC29 document WG1N69022 Warsaw, Poland, (2015)

[32] P. Schelkens (Ed.): "JPEG PLENO — Scope, use cases and requirements Ver.0.1", ISO/IEC JTC1/SC29 document WG1N69021 Warsaw, Poland (2015)

[33] T. Ishikawa, J. Delgado, A. Natu, J. Kleinert, P. Schelkens, A. de Polo: "Draft requirements for JPEG Privacy & Security", ISO/IEC JTC1/SC29 document WG1n69020 Warsaw, Poland (2015)

[34] G. Rouvroy, A. Descampe (Eds.): "Requirements document for a low-latency lightweight image coding system", ISO/IEC JTC1/SC29 document WG1N69027 Warsaw, Poland (2015)

[35] S. Kim, X. Qu, D. Atnafu, H.J. Kim: "Storing Unedited Image as Metadata" ISO/IEC JTC1/SC29 Requirements input document, Warsaw, Poland (2015)

[36] L. Yuan, P. Korshunov, T. Ebrahimi: "Secure JPEG Scrambling enabling Privacy in Photo Sharing" Workshop on De-identification for privacy protection in multimedia, Ljubljana, Slovenia, (2015).

[37] L. Yuan, P. Korshunov, T. Ebrahimi: "Privacy-Preserving Photo Sharing based on a Secure JPEG" The Third International Workshop on Security and Privacy in Big Data (BigSecurity 2015), Hong Kong, (2015).

[38] L. Yuan, T. Ebrahimi: "Image transmorphing with JPEG" IEEE International Conference on Image Processing (ICIP 2015), Canada, (2015).