

# AN INTRODUCTION TO JPEG STANDARDS FOR DIGITIZATION AND ARCHIVING APPLICATIONS

Frederik Temmermans (1,2), Tim Bruylants (1,2), Peter Schelkens (1,2), Touradj Ebrahimi (3)  
(1) Vrije Universiteit Brussel, Department of Electronics and Informatics (ETRO), Pleinlaan 2, 1050 Brussels, Belgium. (2) iMinds, Technologiepark 19, 9052 Zwijnaarde, Belgium. (3) Multimedia Signal Processing Group (MMSPG), Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

## RESUM

El nombre d'imatges digitals generades diàriament creix en un percentatge sorprenent. A més, la forma en què aquestes imatges són capturades, emmagatzemades, compartides, mostrades i utilitzades està canviant. Això condueix a múltiples nous reptes relacionats amb la recerca, la gestió de metadades, la privacitat, la seguretat, l'emmagatzematge i l'enviament. D'altra banda, les noves tècniques d'imatge emergents i la demanda d'experiències d'usuari més immersives impulsa cap a la necessitat d'uns formats d'imatge millorats. El Joint Photographic Experts Group (JPEG) és responsable de proporcionar solucions estandaritzades per fer front a aquests reptes. En els últims 30 anys, això ha donat lloc a diversos estàndards reeixits i àmpliament acceptats, com poden ser el JPEG i el JPEG 2000, i més recentment el JPSearch i el JPEG XT. En aquest text es presenta una breu introducció a aquests estàndards que són rellevants per a aplicacions de digitalització i arxiu. A més, es comenten alguns dels esforços que actualment s'estan dedicant a l'estandarització.

## RESUMEN

El número de imágenes digitales generadas diariamente crece en un porcentaje sorprendente. La forma con la que estas imágenes son capturadas, almacenadas, compartidas, mostradas y utilizadas está cambiando. Este hecho conduce a múltiples nuevos retos relacionados con la investigación, la gestión de metadatos, la privacidad, la seguridad, el almacenaje y el envío. Por otra parte, las nuevas técnicas de imagen emergentes y la demanda de experiencias de usuario con más inmersión impulsan hacia la necesidad de unos formatos de imagen mejorados. El Joint Photographic Experts Group (JPEG) es responsable de proporcionar soluciones estandarizadas para hacer frente a estos retos. En los últimos 30 años, esta circunstancia ha dado lugar a unos cuantos estándares exitosos y ampliamente aceptados, como pueden ser el JPEG y el JPEG 2000 y, más recientemente, el JPSearch y el JPEG XT. En este texto se presenta una breve introducción a estos estándares, los cuales son relevantes para aplicaciones de digitalización y archivo. Además, se comentan algunos de los esfuerzos que actualmente se están dedicando a la estandarización.

## Abstract

The number of daily generated digital images grows at a staggering rate. In addition, the way these images are captured, stored, shared, rendered and used is changing. This leads to multiple new challenges related to search, metadata management, privacy and security, storage, and transportation. Moreover, new emerging imaging modalities and the demand for more immersive user experiences push forward the need for enhanced image formats. The Joint Photographic Experts Group (JPEG) is responsible to provide standardized solutions to deal with these challenges. Over the last 30 years, this resulted in several successful

and widely adopted standards such as JPEG and JPEG 2000, and more recently JPSearch and JPEG XT. This paper gives a brief introduction to these standards that are relevant to digitization and archiving applications. In addition, some ongoing standardization efforts are introduced.

## Introduction

Digitization typically results in huge amounts of data being generated. Representing such digitized information in an efficient manner, and also allowing for easy access, retrieval and stable long-term storage,

requires advanced coding technologies that preferably also adhere to international standards. The latter is important with respect to archival and long-term storage as standards ensure that content can be decoded at any time without depending on obsolete or legacy systems. Moreover, standards bring the benefit of interoperability between implementations of different system vendors.

Nevertheless, since different applications imply different requirements, different image file formats have been designed dedicated to specific usages. In addition, the rapid evolution of capturing technology imposes new requirements on image file formats. Increasing resolutions (giga pixel), dynamic range, spectral resolution and dimensionality are just a few examples. As a consequence, image file format development needs to evolve along with innovation in capturing technology. JPEG, a shorthand for the Joint Photographic Experts Group, is a standardization committee responsible for the development of still image standards that provide an answer to industry demand.

JPEG is officially one of two subgroups of WG1 of SC29 of the JTC1. The other subgroup is the JBIG. JTC1 brings together experts from the ISO and the IEC to act as a single standardization body. Thus, JPEG is in practice also a shorthand for WG1 SC29 JTC1 ISO/IEC. To make things even more complicated, JPEG is a joint committee of JTC1 and ITU-T (formerly called CCITT) Study Group 16 (SG16) of the ITU. In ITU-T the activities of JPEG fall in the domain of VCEG. For this reason, many of the published JPEG standards are also integrally published by ITU-T. Thus, many (but not all) JPEG standards carry a double designation, one of ISO/IEC and one of ITU-T. This is why for example the JPEG image coding standard is known as both ISO/IEC 10918-1 and Rec. ITU-T T.81.

In 1983, ISO started working on the development of technology that would allow displaying photo quality graphics on terminals, which at that time were entirely character based. Within ISO, this work was placed under the Technical Committee 97 (TC97), Sub Committee 2 (SC2), Working Group 8 (WG8) – Coding of Audio and Picture Information. At the same time, the CCITT (now ITU-T) had Study Group VIII (SG8) that also focused on image communication. Given the common field of expertise and objectives within ISO and CCITT, it was decided in 1986 to create the Joint (CCITT/ISO) Photographic Experts Group (JPEG), followed by the Joint Bi-level Imaging Group (JBIG) in 1988. And, ISO TC97 merged with IEC TC83 to become ISO/IEC JTC1 in 1986 and CCITT was renamed to ITU-T in 1993.

Conceived in 1986, JPEG started working on its first standard to provide efficient compression of continuous-tone still images, which would be issued as JPEG

in 1992, or more formally as ISO/IEC 10918-1 | Rec. ITU-T T.81 – Information technology - Digital compression and coding of continuous-tone still images. This is the same standard that is still used today to encode, transmit and store millions of pictures each day.

Over the last 30 years, JPEG issued numerous standards. Some are extremely successful and widely used, while others are mainly used in specific markets. Moreover, a complete standard is often comprised of a set, or a suite, of standards, which are then referred to as parts of the standard. Each part is by itself a standard, but typically contains references to other parts of the suite to complement or extend the functionality. In ISO/IEC, a five digit number is assigned to uniquely label the standard, optionally followed by a part number (separated with a dash). In ITU-T, each part of a standard receives its own unique T-number.

JPEG 2000 Part 1, or ISO/IEC 15444-1 | Rec. ITU-T T.800 was released in 2000 and designed not just to provide higher compression efficiency than that of JPEG, but also to support wider set of functionalities such as tiling, resolution and quality scalability, progressive decoding, lossy and lossless compression modes, random access, and more.

JPEG XT is a relatively new standard that specifies a series of backwards compatible extensions to the legacy JPEG standard (ITU Recommendation T.81 – ISO/IEC 10918-1). While JPEG is still the dominant technology for storing digital images, it fails to address several requirements that have become important in recent years. Most importantly, JPEG XT adds support for the compression of images with higher bit depths (9 to 16 bits), high-dynamic-range imaging (HDR), lossless compression, and representation of alpha channels. JPEG XT extends the original JPEG specification in a completely backwards compatible way. Existing tools and software will continue to work with the new code streams, while new features will help move JPEG into the 21st century.

In addition to standards that focus mainly on image coding, JPEG also provides standards related to image access and metadata management. JPSearch provides solutions for embedding and mapping metadata, querying images and image repositories and integration of images within the cloud of linked data.

From the perspective of the user, a key aspect is to select the right standards for the specific use cases and achieve optimal alignment between these adopted file formats, with respect to both image quality and metadata consistency. Moreover, future developments should be taken into account to guarantee the longevity of digitized artefacts.

This paper provides an overview on some of the most prominent suites of standards suitable for digi-

tization purposes: JPEG, JPEG 2000, JPEG XT and JP-Search. In addition, two recent ongoing activities are introduced: JPEG PLENO and JPEG XS. The paper provides technical information on the design of the standards and will as such give the reader better insight on the implications when using the standards in practice.

## JPEG Image coding Standard

The JPEG image coding standard, or ISO/IEC 10918-1 | Rec. ITU-T T.81 [10], was created in 1992 and represents today's most commonly used image coding technology. Without any doubt, it can be stated that JPEG has been one of the most successful multimedia standards defined so far. This section briefly introduces the reader with the inner workings of the classic JPEG image coder [3].

Similar to most other JPEG standards, the specification actually only describes the decoding process. The implementer is free to choose how to encode an image, as long as the final produced code-stream is compliant to the standard. However, this section explains the encoding process. Given that the JPEG encoding-decoding cycle is symmetric, the encoding process can then be considered as the inverse of the decoding process. Figure 1 presents a JPEG encoder block schema. It shows that encoding (and decoding) an image with JPEG can be roughly divided into four steps, which are subsequently described in the following subsections.

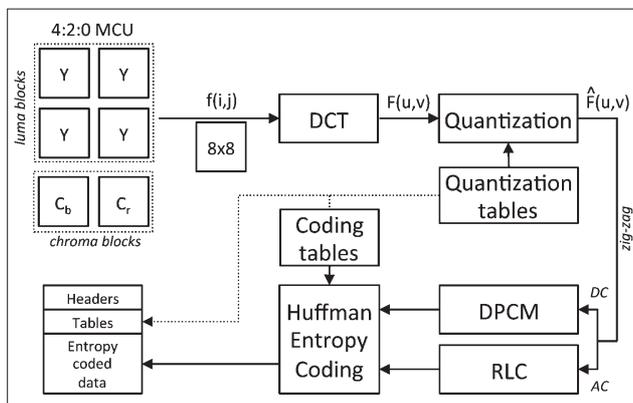


Figure 1: JPEG encoder schematic (showing a 4:2:0 down-sampling mode).

### Preprocessing

The preprocessing step prepares the image samples before they are actually encoded. JPEG is designed to work on continuous-tone color images, with one or more components, each comprised of 8-bit integer samples. Typically, natural photographic images are represented in an RGB color space. The pre-processing

step first converts the RGB samples to a  $YCbCr$  color space, which acts as a first decorrelation step that significantly improves the compression performance. One thing to note here is that only the core coding technology was specified for JPEG and that, as a consequence, the WG1 committee did not specify the precise RGB to  $YCbCr$  color transform. This means that different codec implementations were not necessarily using the same color transform. However, this was later remedied by publication of the JPEG File Interchange Format (JFIF) as ISO/IEC 10918-5 | Rec. ITU-T T.871. Most modern implementations are compliant with JFIF and thus apply the same color transform.

After applying the color transform, the Cb and Cr components are optionally down-sampled by a factor of two. The supported down-sampling ratios of JPEG are labeled as (1) 4:4:4 (no down-sampling), (2) 4:2:2 (reduction by a factor of two in the horizontal dimension), and (3) 4:2:0 (reduction by a factor of two in both the horizontal and vertical directions). Typically, the 4:2:0 down-sampling mode is applied, effectively reducing the spatial resolution of the chrominance components by a factor of four (thus decimating 50% of the data).

Finally, the components are divided in small rectangular blocks of 8x8 samples each, called code-blocks.

### Discrete Cosine Transform (DCT)

Next, each 8x8 code-block is converted to a frequency domain representation by means of a normalized 2D type-II DCT [2], applied to the center shifted samples – i.e. by subtracting 128 from each value. The DCT is given by:

$$F(u, v) = \frac{1}{4} \left[ C(u)C(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right], \quad (1)$$

where  $0 \leq u < 8$  is the horizontal spatial frequency,  $0 \leq v < 8$  is the vertical spatial frequency, and with

$$C(u), C(v) = \begin{cases} 1/\sqrt{2} & \text{for } u, v = 0 \\ 1 & \text{for } u, v \neq 0. \end{cases}$$

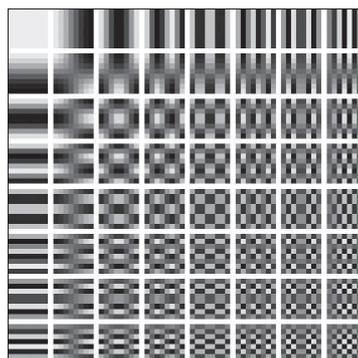


Figure 2: The basis functions of the 8x8 DCT.

The result is an 8x8 block of frequency coefficient values, each corresponding to the respective DCT basis functions, as shown in Figure 2. The top-left value is called the DC coefficient and represents the average for the whole block. The other 63 coefficients are called the AC coefficients and these represent specific high-frequency information of the original block. A property of the 2D DCT is its tendency to aggregate most of the signal energy in the DC and lowest AC coefficients. Consequently, the AC coefficients are scanned using a zig-zag scanning pattern, as shown in Figure 3 (from top-left to bottom-right), according to their respective increasing frequencies. This is useful, as it allows to prioritize the rate allocation depending on the frequencies, present in the block, in order to fit with the human visual system (HVS).

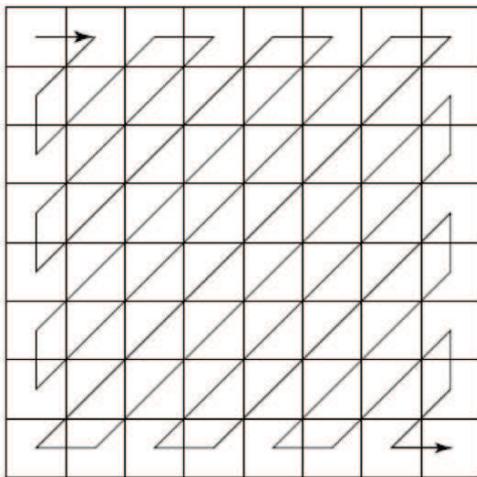


Figure 3: The 8x8 zig-zag scanning pattern. The DC coefficient comes first, followed by 63 AC coefficients.

### Quantization

It is well established that the human eye is good at distinguishing small differences in smooth regions, but fails to notice distortions in high frequency regions. This fact is exploited in the quantization step, by approximating high-frequency coefficients more coarsely than low-frequency coefficients. Thus, after the DCT, JPEG performs a quantization operation on each transformed block by simply dividing each coefficient by a respective quantization step, followed by rounding the result to the nearest integer. As a result, many of the high-frequency coefficients become zero, while many other coefficients become small integer values that take fewer bits to represent than before. The quantization step sizes are represented in an 8x8 quantization matrix, which is signaled in the code-stream. Carefully tuning the quantization matrix allows an encoder to control the rate and quality of the

produced output. Maximum quality can be achieved by using step sizes of 1.

### Entropy coding

Following the quantization step, the encoder entropy-codes the coefficients in the zig-zag scanning order (see figure 3) and packages them in the final code-stream. Entropy coding in JPEG relies on a combination of run-length-coding and Huffman entropy coding [9, 30].

Important to note is the fact that the JPEG specification allows to freely construct the set of Huffman code words. The code-stream provides syntax elements to signal two Huffman code word tables, one for luminance blocks, and another for chrominance blocks. JPEG even allows to define and switch between multiple table sets within a single code-stream. This is actually a very useful feature because it allows an encoder to provide optimized Huffman code words, depending on the actual image content.

### JPEG 2000

JPEG 2000 Part 1, officially named ISO/IEC 15444-1 | Rec. ITU-T T.800 [11] was released in 2000 as the successor to the original JPEG standard [10]. It is a wavelet-based still image coding system that aims to not just to provide higher compression efficiency than that of JPEG, but also to support a much wider set of extra features, such as native tiling support, resolution and quality scalability, progressive decoding, lossless compression, region-of-interest (ROI) coding, error resilience, and true random code-stream access. Moreover, it supports various color formats, such as RGB,  $YCbCr$ , or a generic n-channel format, at bit-depths ranging from 1 to 38 bits per channel. Each channel is represented in the image as an individual image component and JPEG 2000 supports up to 16384 image components in an image.

It is worth noting that the JPEG 2000 Part 1 specification itself is written from a decoding perspective. This grants encoder implementations to tweak and customize the code-stream generation process to application specific use cases and to potentially improve the encoding performance. The standard is only concerned in code-stream compliance by specifying mainly the decoding process. However, for the purpose of explaining the fundamental building blocks of JPEG 2000, this section is written from the encoding point of view. For a more detailed overview of the JPEG 2000 suite of standards, the reader is advised to consult [3, 11, 32, 33].

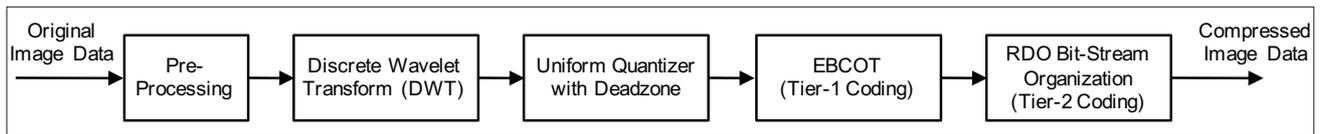


Figure 4: The high-level blocks of the JPEG 2000 encoder.

Figure 4 shows the high-level blocks of the JPEG 2000 encoder, which are described in the following sub-sections [3].

### Preprocessing

The preprocessing phase formats the image samples before actually encoding them. It can be roughly divided into three parts.

The first step partitions the input image into equally sized rectangular and non-overlapping regions, called tiles. The image grid system provides the benefit of allowing post compression image cropping, without the need to go through a complete decode-encode cycle. After tiling of the input image, each tile is then further processed individually, without dependencies on the other tiles. Tiling is particularly useful in application specific use-cases, such as the compression of large images in restricted memory environments, or in order to facilitate parallel processing. However, using tiles influences the overall compression efficiency due to the introduction of hard borders and the restriction on the amount of samples per processed tile. Solutions to reduce tiling artifacts exist [8, 39], though the majority of use cases allow for disabling the tiling functionality altogether.

The 2nd step, the DC level shift, transforms unsigned integer type images to a signed representation to avoid certain implementation problems, such as numerical overflow. However, it does not (significantly) affect the coding efficiency.

The final preprocessing step, decorrelates color information. JPEG 2000 Part 1 defines two component transforms to choose from. Both transforms work on the first three components only that are then implicitly interpreted as RGB components, and requiring that these three components are of equal sample type – same bit-depth and sign format – and size. The first transform is called the ICT and is essentially a commonly used RGB to YCbCr color transform. Due to the floating-point coefficients of the transform, it can only be used in lossy coding. The second alternative transform is called the RCT. It is a reversible integer-to-integer transform that approximates the ICT but can be used for lossless coding as well. It has a lower performance than the ICT, but it allows for exact reconstruction of the original sample values.

### Multi-level 2-D Discrete Wavelet Transform

The Discrete Wavelet Transform lies at the core of JPEG 2000 and it allows decomposing the image information into a set of smaller so-called sub-bands. JPEG 2000 natively supports two built-in wavelet filter banks, labeled 5x3 and 9x7, both originating from the same family of biorthogonal CDF wavelets [6].

- The 5x3 kernel is mainly provided for supporting lossless compression because all of its filters have rational coefficients with the dividends being powers of two. This makes it possible to deliver perfect reversibility, as pure interger calculus suffices and rounding errors can be perfectly controlled.

- The 9x7 kernel has a higher factorization-order than the 5x3 kernel, meaning that it offers an improved energy compaction performance for lossy compression compared to the 5x3 kernel. However, it is irreversible due to the fact that its filter coefficients are irrational numbers.

The JPEG 2000 Part 1 specification contains a very strict definition of a lifting scheme for a one-dimensional DWT that should be followed by all compliant implementations, in order to provide interoperability between different implementations. This 1D-DWT represents the core DWT operation of JPEG 2000. Subsequently, and as a consequence of the wavelet separability property, a two-dimensional DWT can be achieved by applying multiple 1D-DWT along both dimensions. Thus, the one-level 2D-DWT is constructed by applying the 1D-DWT first to each row of the image, followed by applying the same 1D-DWT to each column of the image. A single 2D-DWT step creates four frequency bands, called sub-bands, containing wavelet coefficients that represent filtered and sub-sampled versions of the input image. The total amount of wavelet coefficients is equal to the original amount of samples. Each subsequent resolution level is created by taking the low-frequency sub-band (which is one of the four resulting sub-bands) of the last resolution level as input for the next 2D-DWT operation. This decomposing process may be repeated in order to generate the desired number of resolution levels and to optimally decorrelate the image samples.

Figure 5 (a) illustrates such a multi-level 2D-DWT decomposition structure, commonly known as the Mallat decomposition [28], with R resolution levels.

Each sub-band label indicates the type of filter that was used for respectively the horizontal and vertical dimensions, via the characters L and H, for low-pass and high-pass filtering respectively. It also indicates the decomposition level in sub-script. In this notation,  $LL_0$  represents the original – non-transformed – tile-component. With each further decomposition an extra resolution level is introduced, replacing the current LL band with four smaller sub-bands.

the  $9 \times 7$  wavelet kernel. The outcome is a set of sub-bands, that represent four resolution levels. The lowest resolution level,  $r = 0$ , is represented by just the  $LL_3$  sub-band. The other resolution levels,  $r = N_L - n$ , are represented by their respective sub-band contributions from  $LH_n$ ,  $HL_n$  and  $HH_n$  and the image from resolution  $r - 1$ .

Decomposing the original image via a multi-level 2D-DWT into a set of sub-bands causes the resulting coefficient values to be smaller and closer to zero, which in turn allows for easier and more efficient encoding of the information that is present in the image.

### Quantization

Following the forward multi-level 2D-DWT, the resulting wavelet coefficients of the produced sub-bands are optionally quantized by means of uniform scalar dead-zone quantization. This type of quantization can be easily considered as being a simple division operation that further reduces the nominal value of the coefficients. The quantization factor is selected per sub-band and depends on the applied wavelet kernel and the desired reconstruction quality [11, 32]. It serves two main purposes:

1. It allows for tuning the encoding process for perceptual visual performance, based on the properties of the human visual system, or for low-complexity rate control of the generated output.
2. When using the  $9 \times 7$  floating-point wavelet kernel, it is used to realign the dynamic range of the coefficients at the most significant bit side across all sub-bands. Doing so, enables the rate-distortion optimizer to make full use of the available precision in the floating-point coefficients, thus improving on the final reconstruction quality.

### EBCOT

After quantization, the resulting coefficients are entropy encoded into a binary representation by the Embedded Block-coding with Optimized Truncation (EBCOT) process. This methodology allows JPEG 2000 to efficiently encode the coefficient values in a block-based and bit-plane per bit-plane approach. EBCOT partitions each sub-band into small rectangular blocks, called code-blocks, which are all independently encoded. The dimensions of these code-blocks are selectable at encoding time. The primary advantage of this methodology is that EBCOT generates a set of bit-streams, one for each code-block, which can all be cut off at multiple discrete points, labeled as code-passes (at the cost of losing information). Cutting the bit-stream of a code-block is mathematically equivalent

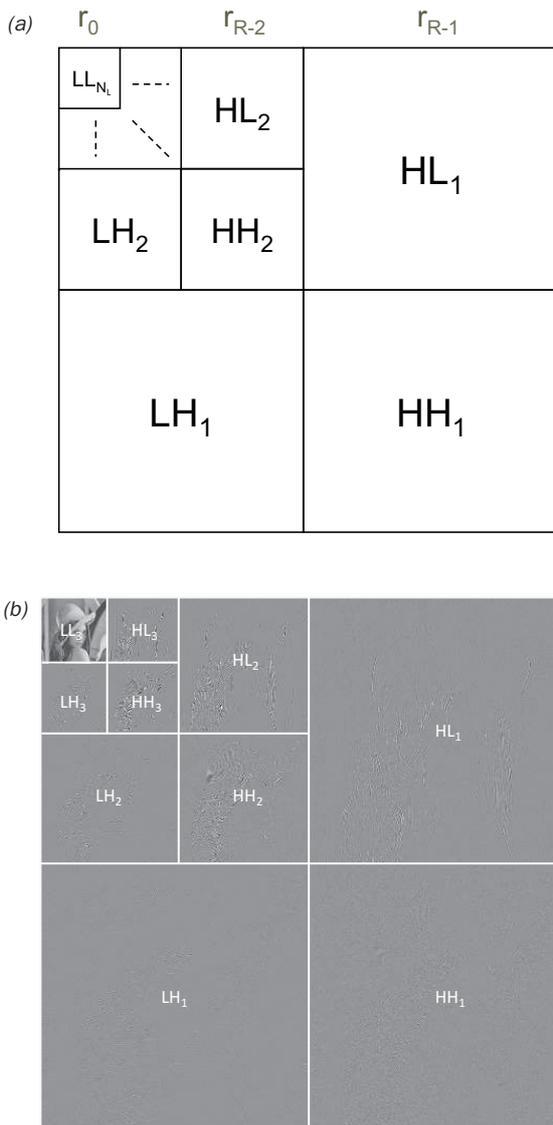


Figure 5: Multi-resolution 2D-DWT Mallat decomposition in JPEG 2000.

(a) Schematic representation of an n-level 2D-DWT Mallat decomposition.

(b) Lena image decomposed with a 3-level  $9 \times 7$  2D-DWT, generating 4 resolution levels.

The lowest resolution level at  $r = 0$  is represented by exclusively the  $LL_0$  sub-band.

As an example, figure 5 (b) shows the result after applying a 3-level 2D-DWT on Lena ( $N_L = 3$ ), using

to a binary quantization and it allows the encoder to provide efficient rate-distortion control. Furthermore, the independent encoding of the code-blocks brings many extra advantages, such as native random accessibility into the image, potential data parallelization, support for cropping of the image (without an entire decode-encode cycle).

### Code-stream organization

As previously stated, the outcome of EBCOT is an individual bit-stream per code-block that can be truncated at multiple byte-aligned positions. These truncation points enable the encoder to select the amount of bytes to allocate for each code-block in the output code-stream. EBCOT encodes the wavelet coefficients in a deterministic order, from the MSB-plane down to the LSB-plane. This means that partially decoding a code-block bit-stream is equivalent with a quantization operation and still provides useful coefficient values for the decoding process (albeit at lower reconstruction quality).

The JPEG 2000 code-stream organization is particularly designed to make use of these embedded bit-stream properties and offers a very flexible and versatile code-stream syntax in order to allow fine grained rate-distortion control and quality and resolution scalability. The syntax allows to break up the code-block bit-streams in small parts, called chunks, and rearrange them according to different progression orders depending on the application use case. The only restriction is that the relative order of chunks within a code-block must be maintained to guarantee correct decoding at any moment.

### Precincts

Foremost, the code-stream syntax relies on the concept of so called precincts. Precincts represent conceptual groups of code-blocks that influence the same spatial region and provide an efficient way to randomly access spatial regions in an image. A precinct is defined as a group of code-blocks from all the sub-bands that represent a given resolution level and a given spatial region in the original image.

Precinct partitioning is, like code-block partitioning, specified per tile and per component and comprises of fixed size rectangles with the width and height being powers of two, anchored at (0,0) in the reference grid. Thus precinct boundaries always coincide with code-block boundaries (the reverse is not true). Figure 6 shows how code-blocks, belonging to three sub-bands that make up a resolution level, are conceptually grouped into a single precinct. The order in which the code-blocks are encoded is denoted by the numbers.

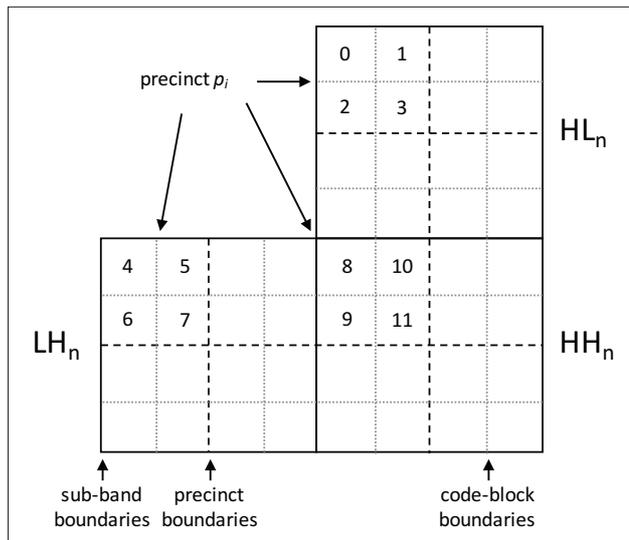


Figure 6: Exemplary illustration of the code-blocks that belong to a single precinct, annotated with the relative order in the precinct.

It essentially follows a raster scan pattern per sub-band, with the sub-bands also ordered raster scan. Choosing the correct precinct sizes is application specific and has two consequences: (1) the amount of overhead depends on the selected precinct sizes and increases with decreasing sizes, and (2) smaller precincts offer better, more precise, spatial random access capabilities.

### Layers and Packets

With precincts defined to represent a set of code-blocks that contribute to a given resolution level, two more conceptual groupings are provided with JPEG 2000: (1) layers and (2) packets.

First of all, the compressed code-block bit-streams are distributed across one or more layers in the final code-stream, with each layer conceptually representing a piece of compressed data. All of the code-blocks from all sub-bands of each tile-component contributes data to each layer. Thus, each layer represents a certain quality increment. Within each layer, the encoder is free to select the amount of code-passes per code-block to include. This selection is typically based on a post compression rate-distortion optimization process. The layer feature offers great flexibility in ordering the data in the code-stream and also enables visual quality scalability.

Secondly, the compressed data belonging to a specific tile-component, resolution, layer and precinct is subsequently aggregated into a packet. These packets represent the smallest accessible data units of a JPEG 2000 code-stream, with each packet containing actual bit-stream chunks of code-blocks. A packet header precedes the packet data and encodes the

required information (such as the length of the packet data, the number of code-passes contributed per code-block, etc) about the content in the packet in order to allow correct decoding. Thus, each packet in the code-stream can be identified by its tile, component, resolution level, layer and precinct, or more formally with (t, c, r, l, p) as its coordinates.

### Rate control

The previous sections 3.4 and 3.5 explain how wavelet coefficients are encoded and how the code-block bit-streams can be distributed over multiple packets to compose a compressed code-stream at the output. However, less attention was spent on the actual rate and quality control that is possible with JPEG 2000. Rate control is the decision process of how much bits to allocate to an image or, more specifically, to regions (spatial, component and bit-depth related) in the image and it is strictly an encoder issue. Generally speaking, two types of rate control exist, targeting (1) a fixed bit-rate or (2) a fixed quality. Depending on the use case, either rate or quality control may be desired. JPEG 2000's code-stream organization is flexible enough to support these two types of rate control, or their combination.

### Region of Interest (ROI) coding

ROI coding refers to encoding certain regions (called the ROI) in the image at a higher quality relative to the rest of the image (called the background). JPEG 2000 provides support for ROI coding by means of two different mechanisms, one being explicit and another being implicit.

The first method involves explicit scaling of the ROI coefficients before entropy coding them to a higher precision than the rest of the coefficients [4]. JPEG 2000 Part 1 allows only a fixed scaling factor that is a power of two and shifts the ROI coefficients by doubling the number of bit-planes. Hence, it is called max shift ROI coding. The max shift method is illustrated in figure 7. Because coefficients are each scaled individually, this ROI coding method supports any shaped ROI. However, it comes at the cost of a slight reduction in compression performance of about 1 to 8% in increased rate (depending on the shape and size of the ROI) [4, 31, 32].

The second method is implicit and involves prioritizing code-block contributions to the ROI over the background regions. Doing so makes sure that the decoder receives data belonging to the ROI before the rest of the data. It has a negligible impact on the overall compression performance, but it is more limited than max

shift ROI in shaping the ROI because it is bound by the code-block granularity.

## JPEG XT

### Introduction

Despite the fact that JPEG is still the dominant technology for storing digital images, it fails to address several requirements that have become important in recent years, such as compression of HDR

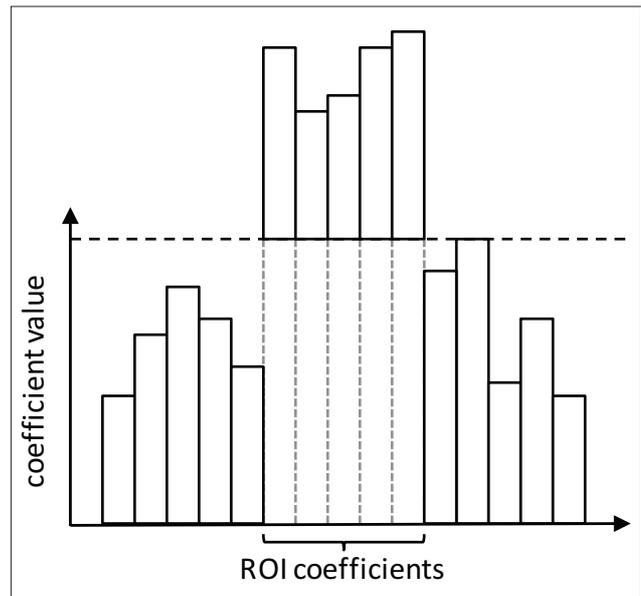


Figure 7: JPEG 2000 max shift ROI method.

images (having bit-depths of 9 to 16 bits), support for lossless compression, and representation of alpha channel information. JPEG XT (ISO/IEC 18477-x) is the latest suite of standards under development by the WG1 committee. It currently consists of nine standards, designed with the purpose to extend the original JPEG specification (ISO/IEC 10918-1 | Rec. ITU-T T.81 [10]) in a completely forward and also backward compatible way. Forward compatibility is an extremely important JPEG XT feature. It ensures that existing tools and software will continue to work with the new code-streams, while new features become available that will help move JPEG into the 21st century. Backward compatibility, on the other hand, allows new JPEG XT software and hardware to process existing JPEG code-streams. It is a native consequence of the JPEG extension based design of JPEG XT [1, 3].

## Suite of standards

JPEG XT is based on JPEG, using the well known discrete cosine transform (DCT) and relying on run-length and Huffman entropy coding at the core. The nine JPEG XT standards are:

- Part 1, Core coding system specifies the core compression technology, and represents in fact JPEG as it is used nowadays. It references ISO/IEC 10918-1 | Rec. ITU-T T.81, 10918-5 and 10918-6, making a normative selection of features from those standards to define the core technology of JPEG XT. Essentially, part 1 defines what is commonly understood as JPEG today.

- Part 2, Coding of high dynamic range images is a backward compatible extension of JPEG towards high-dynamic range photography using a legacy text-based encoding technology for its meta-data [13].

- Part 3, Box file format specifies an extensible boxed-based file format that all JPEG XT extensions are based on [14]. It is designed to be completely compatible to JFIF, ISO/IEC 10918-5, and thus can be read by all existing JPEG implementations (ignoring the content of the extension boxes).

- Part 4, Conformance testing defines the conformance tests of JPEG XT [15].

- Part 5, Reference software provides a reference software implementation of JPEG XT [16].

- Part 6, IDR integer coding extends JPEG XT to support coding of IDR images (like HDR images, but with a limited dynamic range), consisting of integer samples between 9 and 16 bits precision, based on JPEG XT Part 2 and Part 3 [17].

- Part 7, HDR floating-point coding extends JPEG XT to support coding of HDR images, consisting of IEEE floating point samples [18]. It is a super-set of both Part 2 and Part 3 and offers additional coding tools addressing needs of low-complexity or hardware implementations.

- Part 8, Lossless and near-lossless coding defines lossless coding mechanisms for integer and floating point samples [19]. It is an extension of Part 6 and Part 7, allowing for scalable lossy to lossless compression.

- Part 9, Alpha channel coding allows the lossy and lossless representation of alpha channels, thus enabling the coding of transparency information, which is useful for representing arbitrarily shaped images [20].

The explanation of Part 1, the core coding technology, is essentially identical to that of JPEG as described in section 2. More information is also available in [10, 12, 29].

## Forward compatibility and new functionality

Thus, essential to JPEG XT is the ability of always producing forward compatible code-streams that can be decoded by any legacy JPEG decoder. It is evident that such legacy decoders cannot interpret the additional JPEG XT data and require a deterministic way to handle these extended code-streams. Such behavior is constructed by signaling the image data in two segments in the code-stream, while maintaining the syntax of the original JPEG specification, as illustrated in Figure 8.

As with any compliant JPEG code-stream, it starts with the Start of Image (SOI) marker, and ends with the End of Image (EOI) marker. These two markers signal the respective beginning and ending of a single encoded image in a code-stream.

The first segment contains an encoded base layer, that, when decoded, provides a LDR (i.e. 8-bit per component) reconstruction of the image. Although not strictly required by the JPEG XT specification, this base layer typically contains a tone-mapped version of the original image at a specific quality level, as would be expected from decoding any regular JPEG photograph.

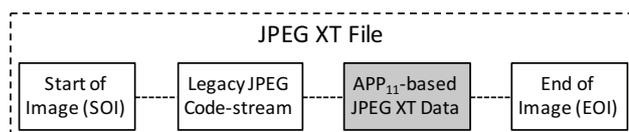


Figure 8: Illustration of embedding JPEG XT extensions in a backward compatible JPEG code-stream by means of the APP11 marker segment.

The base layer is essentially a classic JPEG image, relying only on legacy JPEG technology, and thus readable by any compliant legacy decoder.

The second segment contains JPEG XT extension data, which is embedded in one or more APP11 marker segments in the code-stream. The set of 16 APP marker segments was originally defined to allow embedding of third-party data in JPEG code-streams. For example, EXIF meta-data [5] can be stored inside an APP1 marker segment. JPEG defined such APP markers to be optional and requires decoders to ignore unknown APP markers or APP markers with unknown content1.

Thus, any JPEG XT code-stream is, by means of embedding extension data into one or more APP11 markers, fully compliant to the original JPEG code-stream specifications. While legacy decoders will only see the base layer image data, JPEG XT decoders, on the other hand, can interpret the extra data from the APP11 marker segments.

## JPEG XT Boxes

The APP11 marker segments are in fact containers that can carry any type of byte-oriented content. JPEG XT structures its extension data by means of JPEG XT boxes, which are similar to the boxes that are used by JPEG 2000 (i.e. in JP2, JPX, JPM and the ISO Base Media File Format). A box is a generic data container that has both a type, and a body that carries the actual payload. The type is a four-byte identifier that allows decoders to identify its purpose and the structure of the respective body content. JPEG XT code-streams may carry several boxes of identical type. In such case, each of the identical typed boxes are logically distinct and will differ in the value of the Box Instance Number,  $E_n$ , of the JPEG Extensions marker segment (see Figure 9).

The boxes are embedded into the code-stream format by encapsulating them into one or several JPEG XT marker segments. JPEG XT marker segments are constructed in such a way that they appear as APP11 marker segments to legacy JPEG decoders. Because of this, the maximum length of a single JPEG XT marker segment is, like any other JPEG marker segment, restricted to 65,534 bytes. However, since boxes can grow beyond 64 kiB in size, a single box may extend over multiple JPEG XT marker segments. This means that decoders possibly have to merge multiple marker segments before decoding the box content. JPEG XT marker segments that belong to the same logical box, and thus require merging prior to interpretation, will have identical Box Instance Number fields ( $E_n$ ), but different Packet Sequence Number fields ( $Z$ ). The Packet Sequence Number is a 32-bit field that specifies the order in which payload data shall be merged. Figure 9 shows the complete JPEG XT marker segment structure. The  $L_e$  field represents the marker segment length (16 bits), while the  $L_{Box}$  field represents the total logical box size (32 bits). The Common Identifier (CI) field is always set to 0x4A50 ('JP'). The actual box payload data depends on the box type and is defined in the various JPEG XT specifications.

## Image life cycle and metadata handling: JPSearch

### Introduction

The tremendous growth of generated images does not only require efficient representation formats, it also necessitates tools to assist search and retrieval. Common issues users experience relate to metadata preservation when sharing, exchanging or moving images, metadata consistency and semantics, and inconsistent access via proprietary application interfaces. The JPSearch suite of standards (ISO/IEC 24800) provides methods that help to deal with these issues. In particular, JPSearch defines interfaces and protocols for data exchange between the components of a search and retrieval framework, with minimal restrictions on how these components perform their respective tasks [21].

This section illustrates the approach of JPSearch with respect to metadata handling; and querying of images and image collections. For a more in depth overview of JPSearch the reader is referred to [7, 34–37].

### Core metadata and metadata conversion

Metadata handling is crucial to enable efficient search and data consistency. The JPSearch Core Schema is as root schema to map common metadata fields [22]. The schema is modeled in XML and provides a light-weight and extensible mechanism for annotating image content. It supports 20 main elements, including elements for spatial, region of interest based annotations. The individual elements have been selected according to their coverage by popular metadata formats, including Dublin Core, MPEG-7, and Exif. Furthermore, the JPCore Schema features an extensibility mechanism for integrating parts of external (XML serializable) formats.

In practice many different metadata schemas are used along each other. Therefore, to achieve interoperability, systems require mechanisms for translating metadata from one schema into another. To that end,

<b>APP<sub>11</sub></b> 0xFFE8	$L_e$	CI Common Identifier	$E_n$ Box Instance	$Z$ Packet Seq. Number	$L_{Box}$ Box Length	$T_{Box}$ Box Type	$XL_{Box}$ Box Length Extension	Payload Data
-----------------------------------	-------	-------------------------	-----------------------	---------------------------	-------------------------	-----------------------	------------------------------------	--------------

Figure 9: Organization of the JPEG XT Marker Segment, which is interpreted as an APP11 marker segment by legacy JPEG decoders.

JPSearch defines the Translation Rules Declaration Language (TRDL). It is an XML-based language that provides means for translating metadata to and from the JPSearch Core Schema into equivalent information of an external schema. A transformation model specifies multiple rules with source and target format. One field can be split or multiple fields can be combined during the translation. For more complex conversions regular expressions can be used to describe element or attribute mappings.

JPSearch provides tools for registering and requesting metadata schemas and their translation rules via a registration authority. It serves as a centralized platform where JPSearch compliant systems can find the corresponding translation rules for specific metadata terms [37].

### **Embedding metadata**

JPSearch allows to embed any type of metadata in JPEG and JPEG 2000 images [26]. By embedding the metadata into the image itself, the metadata stays attached to the image, even when it is moved between applications or platforms. The JPSearch file format is backwards compatible with the JPEG and JPEG 2000 image coding standards. Multiple metadata schema instances can be embedded in a single image, even if they have different schemas [37].

### **JPonto**

Many metadata schemas are evolving from tag or key-value based schemas to semantic representations by adopting linked data principles. Therefore, JPEG has introduced a JPEG Ontology for Still Image Descriptions (JPonto) [24, 37, 38]. The main goal is to provide a simple and uniform way of annotating JPEG images with metadata compliant to the linked data principles. JPonto provides a set of classes, properties, and restrictions that can be used to represent and interchange information about still images generated by different systems and under different contexts. It can also be specialized to create new classes and properties to model image information for different applications and domains via the registration authority. In addition, it is specified how RDF metadata annotations should be embedded in JPEG or JPEG 2000 files.

Embedding semantic image descriptions in images allows to link images with objects in the real world. It can be used for many different purposes including:

- Annotating people in images
- Expressing emotions of people in images
- Identifying objects in images such as buildings, artworks, animals, etc.

- Describing the scene
- Returning results of automated recognition algorithms.

### **Collection metadata and exchange**

In addition to metadata at the image level, JPSearch also provides functionalities for handling metadata at the collection level. More specifically, JPSearch Part 5 defines collection metadata and a data interchange format for exchanging image collections and associated metadata between JPSearch-compliant repositories [27]. The interchange format aims at enabling simple and reliable transfer of data between different systems, where both metadata at the level of the images as at the level of the collection is preserved. It serves the following purposes [37]:

- exchanging data between repositories on different devices and platforms,
- consolidating metadata generated on different systems,
- transferring data to a newer system,
- consolidating selected data to a centralized repository, and
- archiving data in a format that will survive current products.

### **Querying**

JPSearch provides two mechanisms for querying images and collections of images: the JPSearch Query Format (JPQF) and the JPSearch API [23]. JPQF focuses on providing an extended and diverse query syntax, while the API provides a simplified and complementary interface and specifies the whole communication chain, including the protocols to be used for sending and retrieving data.

JPQF defines an extensive XML-based query syntax that facilitates searching across repositories. For interoperability reasons, JPQF adopts a subset of the MPEG7 Query Format (MPQF) relevant to the image domain. The query format consists of three parts: the input query format, the output result format and the query management tools. The input query format specifies how to formulate search queries to send to a repository. The output result format on the other hand specifies how to return aggregated results in order to present them to the user or provide them to intermediate systems for further processing. Finally, the query management part defines functionalities for service discovery, service aggregation and service capability description.

Many online platforms such as Facebook, Flickr or Picasa provide access to their services via an Rep-

representational State Transfer (REST) APIs. These APIs provide access to the data in a simple format, typically based upon XML (Extensible Markup Language) or JSON (JavaScript Object Notation) and are easy to use by third party developers. However, most of these APIs are specific to the content provider. Therefore, building applications that aggregate content from several sources requires writing a connector for each content provider separately. With this in mind, JPEG introduced the JPSearch API. It provides means for querying images and collections of images and supports visual search applications. It is designed to complement existing APIs rather than to replace them.

## Ongoing standardization efforts

Since technology continuously evolves new standards need to be developed to answer to market demand. This section briefly introduces two ongoing standardization efforts: JPEG PLENO and JPEG XS. JPEG PLENO focuses on the representation of new image modalities such as light-field, point-cloud and holographic imaging while JPEG XS focuses on low-latency lightweight image coding. More information on the current state and ongoing work related to these activities can be found on the JPEG website [1].

### JPEG PLENO

In recent years, imaging technology has evolved vastly. It is now possible to acquire new and richer forms of data. Images are no longer a representation of a single linear perspective capture but are computed based on various sensor inputs. This fact led to the birth of a new field called "Computational Imaging" that combines the fields of optics and signal processing. Resulting new image modalities require file formats to evolve along. For example, high dynamic range (HDR) and 3D image sensors, burst-mode cameras, light-field sensing devices, holographic microscopes and advanced MEMS (e.g. DMD and SLM) devices, enable new capturing and visualization possibilities. Computational imaging will cause a paradigm shift in the creation and consumption of digital images. The JPEG PLENO group investigates how this evolution can be properly addressed while also taking into account JPEG's legacy formats. As such, it targets the development of a standard framework for representation and exchange of new types of important plenoptic imaging modalities, such as point-cloud, light-field, and holographic imaging [1].

Point-cloud data represents a sparse set of data points in a fixed coordinate system, rather than a set

of pixels in an orthogonal grid. These datasets can be acquired with a 3D scanner or LIDAR and are subsequently processed to represent 3D surfaces. Moreover, point-cloud data can be easily combined with other data modalities, opening a wide range of opportunities for immersive virtual and augmented reality applications.

Light-field data records the amount of light – or "radiance" – at every point in space, in every direction. In practice, the radiance can be captured by means of an array of cameras (resulting in wide baseline light-field data) or by a light-field camera that employs a grid of microlenses to sample each individual ray of light that contribute to the final image (resulting in narrow baseline light-field data).

Holographic imaging represents a larger class of plenoptic image modalities, of which the holographic microscopy modalities are currently the most prevalent in practice. Holographic microscopes are capable of producing interferometric data that can be subsequently rendered with electro-holographic displays, or as computer-generated holographic (CGH) 3D scenes. Moreover, considering the latest developments of the underlying technologies that make macroscopic holographic imaging systems possible, it is expected that also this type of data will be flooding our imaging markets in the near future. In terms of functionality, these holographic data representations will carry even more information than the aforementioned light-field representations to facilitate interactive content consultation.

Another major concern is to provide the ability to manipulate the content of all these new imaging modalities after their acquisition. Such manipulations may have different purposes, notably artistic, task-based and forensic, but are key to their success and usability. For instance, it will be possible for users to change, in real time, the focus, the depth of field and the stereo baseline, as well as the viewer perspective, all of which is not possible with the conventional imaging formats.

Depth or other geometric information about the scene is another very useful component, which can easily be derived from either light-field, point-cloud or holographic data. Such information may simplify image manipulations such as compositing and recoloring. Additionally, accurate 3D scene information is also useful to provide localization within a scene and to provide enhanced capabilities to better detect/recognize objects or actions.

In general, these new forms of imaging allow to overcome some limitations of traditional photography, where all capture parameters are set at the time of acquisition. The creation process is partially shifted to a post-capture manipulation phase, making the shooting less definitive.

With the changing acquisition model, the representation model must also adapt. The aforementioned emerging image modalities require a representation format that simultaneously records intensity and color values at multiple perspectives and distances. The representation of point-clouds will also demand a new format, such that attributes corresponding to each point could be accurately represented. In addition, it is essential to consider interoperability with widely deployed image formats, such as JPEG and JPEG 2000.

### JPEG XS

JPEG XS is a recent activity of the JPEG committee to develop a visually lossless compression standard that targets the professional video equipment markets. Today, in the context of professional environments, image and video sequences are transmitted and stored mostly in uncompressed form. This includes cases, such as professional video links (i.e. 3G/6G/12G-SDI links), IP transport (SMPTE2022 5/6 and proprietary uncompressed RTPs), Ethernet transport (IEEE/AVB), and memory buffers. However, with the upcoming high dynamic range (HDR) content, high frame rate content and 4K and 8K resolutions, the bandwidth of the aforementioned links no longer suffices. Transferring the required amount of data, in real-time, requires a low-latency lightweight image coding system that offers visually lossless quality with reduced amount of resources such as power and bandwidth, yet at a reasonable level. JPEG XS is a standard under development that will offer these features [1]:

- Safeguarding all advantages of an uncompressed stream
  - low power consumption (through lightweight image processing),
  - low-latency in coding and decoding,
  - easy to implement (through low complexity algorithm),
  - small size on chip and fast software running on general purpose CPU with the use of SIMD and GPU.
- Without significant increase in required bandwidth
  - low power consumption (through reasonable bandwidth interfaces),
  - longer cable runs,
  - SRAM size and frequency reduction with a frame buffer compression,
  - more adequate for current infrastructures.

## Conclusions

The JPEG standardization committee has produced several suites of standards that can complement each other. The original JPEG file format was released in 1992 and is nowadays still the most used image file format in the world. It is natively supported on every browser and operating system, both on PCs and mobile devices. Although it offers good compression rates, it lacks features that are supported by other file formats. JPEG 2000 does not only improve in terms of compression efficiency, it is also extremely versatile and comes with excellent rate-distortion control due to wavelet-based compression. Supported features include native support for tiling, resolution and quality scalability, progressive decoding, lossy and lossless compression modes, region-of-interest coding and random access. It can be used in a variety of application specific use-cases, including digitization and archiving, medical imaging, digital cinems and satellite imaging. JPEG XT is a recent suite of standards, that is designed with the intention on extending the very successful image coding standard JPEG, in a forward and backward compatible manner, to add support for high bit-depth images, lossless compression and alpha channel coding. Being able to maintain support for legacy image content – i.e. the data –, but also for legacy image processing chains – i.e. the software and hardware –, is of extreme importance. This aspect was shown to be very relevant many times over in the past when new standards, such as JPEG-LS, JPEG 2000 or JPEG XR, were often poorly adopted in the broader consumer market, despite offering significantly better compression efficiencies and improved functionalities. With JPEG XT, the committee takes another approach by essentially pushing new features into existing applications, piggybacking on the original JPEG standard. In fact, every web browser or image viewer is already capable of decoding the LDR portion (base image) of a JPEG XT image. Furthermore, the committee promotes the further adoption of the other parts of JPEG XT, such as HDR support, lossless compression and alpha channel coding.

In addition to image coding standards, the JPEG committee also provides standards related to image search and access, metadata handling and image portability. The rise of digital imaging led to an enormous amount of archives containing a tremendous amount of digital images. Consumers store images on personal devices, online repositories and social media. Many professional markets, including medical, press agencies and geographic information systems, depend on huge digital image archives for their daily workflow. This growth complicates efficient and

interoperable access to these archives. However, in practice, many of these systems rely on proprietary technologies, tightly coupling components of the retrieval process. This results in a closed ecosystem, preventing interoperability with other systems. Moreover, it severely limits the ability of users to freely export their data and metadata to different systems and devices. JPSearch defines the components of a still image search and retrieval framework to achieve interoperability. More specifically, it defines interfaces and protocols for data exchange between devices and systems.

Since technology quickly evolves, new standards need to be developed in order to maintain interoperability in a continuously changing world. JPEG PLENO and JPEG LS are two examples of ongoing activities, focusing on new image modalities and low-latency respectively. More information on JPEG standards and activities can be found on the JPEG website [1].

## Acknowledgments

This work was partially supported by Europeana Space, iMinds HD2R project and iMinds HI2 Data Science programme. Europeana Space has received funding from the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, under GA nr. 621037.

## References

- [1] JPEG Website. Available at <https://jpeg.org>. [2] N. Ahmed, T. Natarajan, and K. R. Rao. Discrete Cosine Transform. *IEEE Transactions on Computers*, 23(1):90–93, J. 1974.
- [3] T. Bruylants. Advanced Coding Technologies For Medical and Holographic Imaging - Algorithms, Implementations and Standardization. PhD thesis, Vrije Universiteit Brussel, 2015.
- [4] C. Christopoulos, J. Askelof, and M. Larsson. Efficient methods for encoding regions of interest in the upcoming JPEG 2000 still image coding standard. *IEEE Signal Processing Letters*, 2000.
- [5] CIPA. Standard of the Camera and Imaging Products Association - Exchangeable image file format for digital still cameras - Exif Version 2.3. [http://www.cipa.jp/std/documents/e/DC-008-2012\\_E.pdf](http://www.cipa.jp/std/documents/e/DC-008-2012_E.pdf), 2012.
- [6] A. Cohen, I. Daubechies, and J. C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 45(5):485–560, 1992.
- [7] M. Doeller, R. Tous, F. Temmermans, K. Yoon, J. Park, Y. Kim, F. Stegmaier, and J. Delgado. Jpeg's jpsearch standard: Harmonizing image management and search. *IEEE MultiMedia*, PP(99):1–1, 2012.
- [8] M. Hashimoto, K. Matsuo, A. Koike, and Y. Nakajima. Tile Boundary Artifact Reduction Algorithms for Tile Size Conversion of Wavelet Image. In *Proceedings of the Tenth ACM International Conference on Multimedia*, Multimedia '02, pages 97–105, New York, NY, USA, 2002. ACM.
- [9] D. A. Huffman. A Method for the Construction of Minimum-Redundancy Codes. volume 9, pages 1098–1101, September 1952.
- [10] ISO/IEC 10918-1 | Rec. ITU-T T.81. Information technology - Digital compression and coding of continuous-tone still images, 1992.
- [11] ISO/IEC 15444-1 | Rec. ITU-T T.800. Information technology - JPEG 2000 Image Coding System: Core coding system, 2002.
- [12] ISO/IEC 18477-1. Information technology - Scalable Compression and Coding of Continuous-Tone Still Images: Core Coding System Specification, 2015.
- [13] ISO/IEC 18477-2. Information technology - Scalable Compression and Coding of Continuous-Tone Still Images: Extensions for High-Dynamic Range Images, 2015. to be published.
- [14] ISO/IEC 18477-3. Information technology - Scalable Compression and Coding of Continuous-Tone Still Images: Box File Format, 2015. to be published.
- [15] ISO/IEC 18477-4. Information technology - Scalable Compression and Coding of Continuous-Tone Still Images: Conformance Testing, 2016. to be published.
- [16] ISO/IEC 18477-5. Information technology - Scalable Compression and Coding of Continuous-Tone Still Images: Reference Software, 2015. to be published. 23
- [17] ISO/IEC 18477-6. Information technology - Scalable Compression and Coding of Continuous-Tone Still Images: IDR Integer Coding, 2015. to be published.
- [18] ISO/IEC 18477-7. Information technology - Scalable Compression and Coding of Continuous-Tone Still Images: HDR Floating-point Coding, 2015. to be published.
- [19] ISO/IEC 18477-8. Information technology - Scalable Compression and Coding of Continuous-Tone Still Images: Lossless and Near-lossless Coding, 2015. to be published.
- [20] ISO/IEC IS 18477-9. Information technology - Scalable Compression and Coding of Continuous-Tone Still Images: Alpha Channel Coding, 2015. to be published.
- [21] ISO/IEC TR 24800-1:2012: Information technology - JPSearch - Part 1: System framework and components, 2012.
- [22] ISO/IEC 24800-2:2011 Information technology - JPSearch - Part 2: Registration, identification and management of schema and ontology, 2011.
- [23] ISO/IEC 24800-3:2010 Information technology - JPSearch - Part 3: JPSearch Query format, 2010.
- [24] ISO/IEC 24800-2/AMD1 Information technology - JPSearch - Part 2 AMD1: JPEG Ontology for Still Image Descriptions, 2015.
- [25] ISO/IEC 24800-3/AMD1 Information technology - JPSearch - Part 3 AMD1: JPSearch API, 2015.
- [26] ISO/IEC 24800-4:2012 Information technology - JPSearch - Part 4: File format for metadata embedded in image data (JPEG and JPEG 2000), 2012.
- [27] ISO/IEC 24800-5:2011 Information technology - JPSearch - Part 5: Data interchange format between image repositories, 2011.
- [28] S. G. Mallat. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.
- [29] W. B. Pennebaker and J. L. Mitchell. *JPEG Still Image Data Compression Standard*. Kluwer Academic Publishers, 1st edition, 1992.
- [30] D. Salomon. *Data Compression: The Complete Reference*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [31] P. Schelkens, A. Munteanu, J. Barbarien, M. Galca, X. G. i Nieto, and J. Cornelis. Wavelet Coding of Volumetric Medical Datasets. *IEEE Transactions on Medical Imaging*, 22(3):441–458, 2003.

- [32] P. Schelkens, Athanassios Skodras, and Touradj Ebrahimi. The JPEG 2000 Suite. Wiley Publishing, 2009. 24
- [33] D. Taubman and M. W. Marcellin. JPEG2000: image compression fundamentals, standards, and practice. Kluwer Academic Publishers, Boston, 2002.
- [34] F. Temmermans. Visual search in mobile and medical applications: feature extraction and classification, interoperable image search and human-machine interaction. PhD thesis, Vrije Universiteit Brussel, 2014.
- [35] F. Temmermans, M. Doeller, I. Vanhamel, B. Jansen, A. Munteanu, and P. Schelkens. Jpsearch: An answer to the lack of standardization in mobile image retrieval. *Signal Processing: Image Communication*, 28(4):386–401, 2013.
- [36] F. Temmermans, F. Dufaux, and P. Schelkens. Jpsearch: Metadata interoperability during image exchange [standards in a nutshell]. *IEEE Signal Processing Magazine*, 29(5):134–139, 2012.
- [37] Yamada A. Delgado J. Temmermans F., Yoon K. ISO/IEC JTC1/SC29/WG1/N70013 JPSearch Whitepaper, 2015.
- [38] R. Tous, J. Delgado, F. Temmermans, B. Jansen, and P. Schelkens. Towards standardized integration of images in the cloud of linked data. In Y. Demey and H. Panetto, editors, *On the Move to Meaningful Internet Systems: OTM 2013 Workshops*, volume 8186 of *Lecture Notes in Computer Science*, pages 388–397. Springer Berlin Heidelberg, 2013.
- [39] J. Wei, M. R. Pickering, M. R. Frater, J. F. Arnold, J. A. Boman, and W. Zeng. Tile-boundary artifact reduction using odd tile size and the low-pass first convention. *IEEE Transactions on Image Processing*, 14(8):1033–1042, Aug 2005.

*El text en català d'aquesta ponència es pot consultar a:*  
**[http://www.girona.cat/sgdap/cat/jornades\\_actes.php](http://www.girona.cat/sgdap/cat/jornades_actes.php)**