ISO/IEC JTC 1/SC 29/WG1 N100400

JPEG White Paper:
JPEG XL Image Coding System

Version 2.0
January 2023

Editors:

Jyrki Alakuijala – Google, Switzerland
Jon Sneyers – Cloudinary, Belgium
Luca Versari – Google, Switzerland
Jan Wassenberg – Google, Switzerland

Contact:

ISO/IEC JTC 1/SC 29/WG 1 Convener – Prof. Touradj Ebrahimi

EPFL/STI/IEL/GR-EB, Station 11, CH-1015 Lausanne, Switzerland

Tel: +41 21 693 2606, Fax: +41 21 693 7600, E-mail: convenor@jpeg.org

Table of Contents

Executive Summary

JPEG XL is a new image coding standard targeting capture, authoring, storage, archival, transmission, and distribution of photographic images as well as graphics, illustrations, mixed contents (e.g. screenshots) and animations. JPEG XL fills the specific needs for responsive web, wide colour gamut, and high dynamic range applications. It provides legacy transition features, and effective and reliable compression at high visual quality. This document provides an overview of the JPEG XL architecture, coding tools and features.

Introduction

The JPEG XL image coding system (ISO/IEC 18181) has a rich set of features compared to other codecs. It is designed to meet the needs of image delivery on the web, as well as professional photography, digital art, printing, medical and scientific applications. It supports wide colour gamut as well as high dynamic range and high bit depth images. JPEG XL further includes features such as animation, alpha channels, layers, thumbnails, lossless and progressive coding to support a wide range of use cases[1]. It is particularly optimised for responsive web environments, so that content renders well on a wide range of devices.

JPEG XL offers significantly better image quality and compression ratios compared to legacy JPEG. Compared to contemporary alternatives, JPEG XL has a shorter specification and allows a simpler implementation, while still offering similar or better compression performance. It is designed for computationally efficient encoding and decoding using software implementations without the need for additional hardware acceleration, even on mobile devices.

Moreover, JPEG XL includes several features that help transition from the legacy JPEG coding format. Existing JPEG files can be losslessly transcoded to JPEG XL files, significantly reducing their size (Fig. 1). These can be reconstructed to the exact same JPEG file, ensuring backward compatibility with legacy applications. Both transcoding and reconstruction are computationally efficient. Migrating to JPEG XL reduces storage costs because servers can store a single JPEG XL file to serve both JPEG and JPEG XL clients. This provides a smooth transition path from legacy JPEG platforms to the modern JPEG XL.

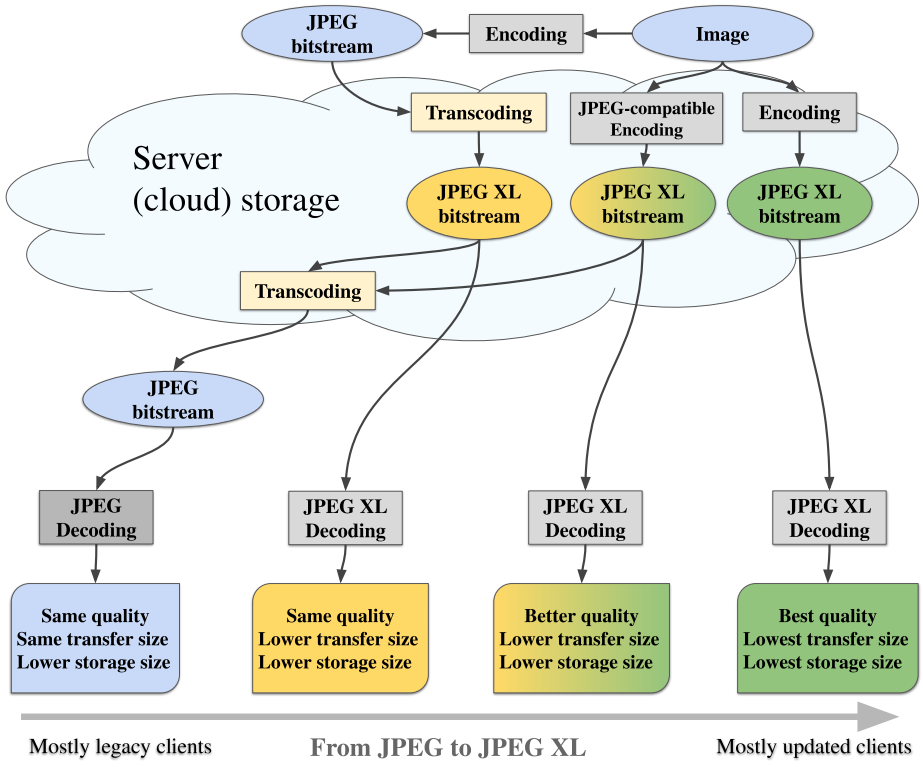


Fig. 1: Example usage scenario

[1] ISO/IEC JTC1/SC29/WG1, "JPEG XL Use Cases and Requirements" output document wg1n83043, Geneva, Switzerland (March 2019), http://ds.jpeg.org/documents/wg1n83043-REQ-JPEG_XL_Use_Cases_and_Requirements.pdf. Accessed: 2023-01-14.

Key Features

The JPEG XL codec is designed to allow users to balance three primary criteria for their application:

- high fidelity to the source image (closely matching human perception),
- encoding and/or decoding speed,
- compression ratio (typically 20:1 to 50:1).

Key features of the JPEG XL codec are:

1. improved functionality and efficiency compared to traditional image formats (e.g. JPEG, GIF and PNG),
2. progressive decoding (by resolution and precision),
3. lossless JPEG transcoding,
4. support for both photographic and synthetic imagery,
5. graceful quality degradation across a large range of bitrates,
6. perceptually optimized reference encoder,
7. designed for wide gamut and HDR,
8. support for animated content,
9. efficient encoding and decoding without requiring specialized hardware.

In terms of compression performance, key results are:

- Lossless JPEG transcoding reduces JPEG size by around **16% to 22%**.
- Lossless compression (from pixels) is around **35%** smaller than (optimized) PNG for 8-bit images, and the improvement is even larger for images with a higher bit depth (e.g. 10-bit or 12-bit).
- JPEG XL is visually lossless at about half the bitrate required by JPEG, i.e. at compression ratios of around 20:1 (1.2 bpp) where JPEG would compress only around 10:1 (2.4 bpp).
- Compared to other image codecs, JPEG XL provides superior compression at a given visual quality.

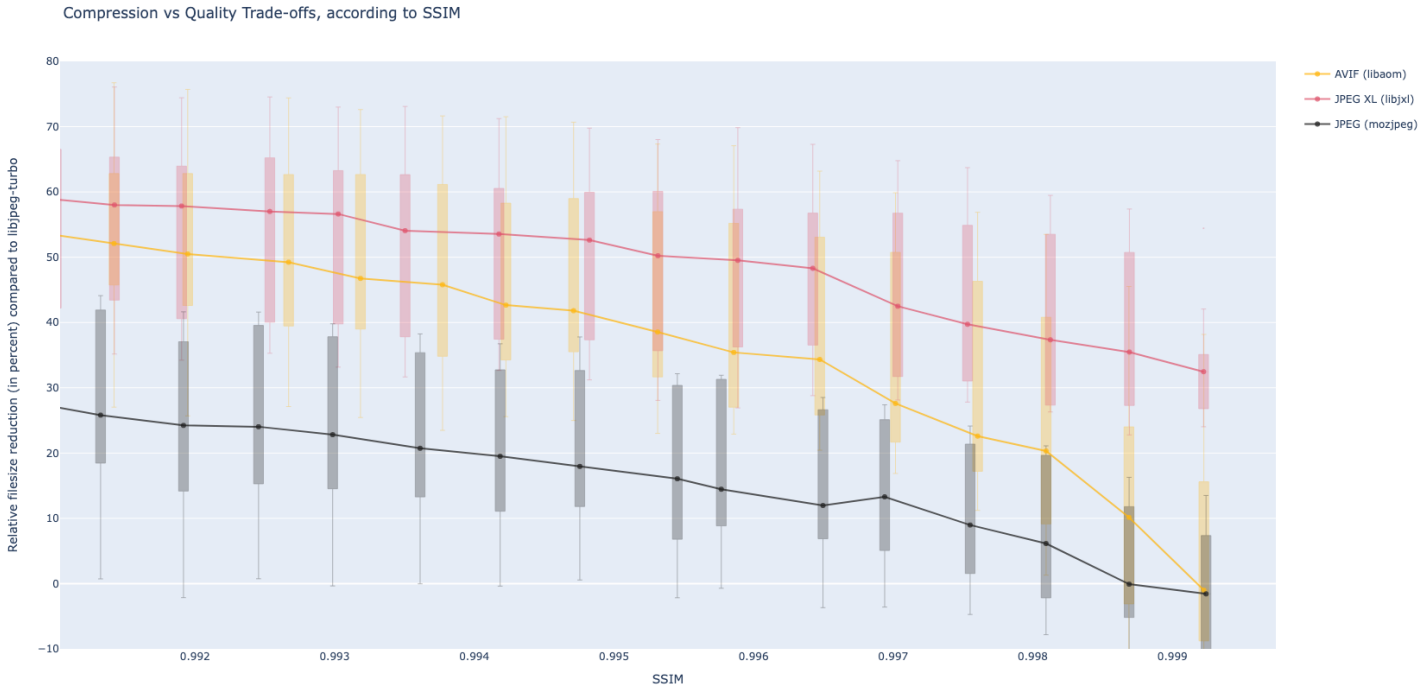
Compression performance

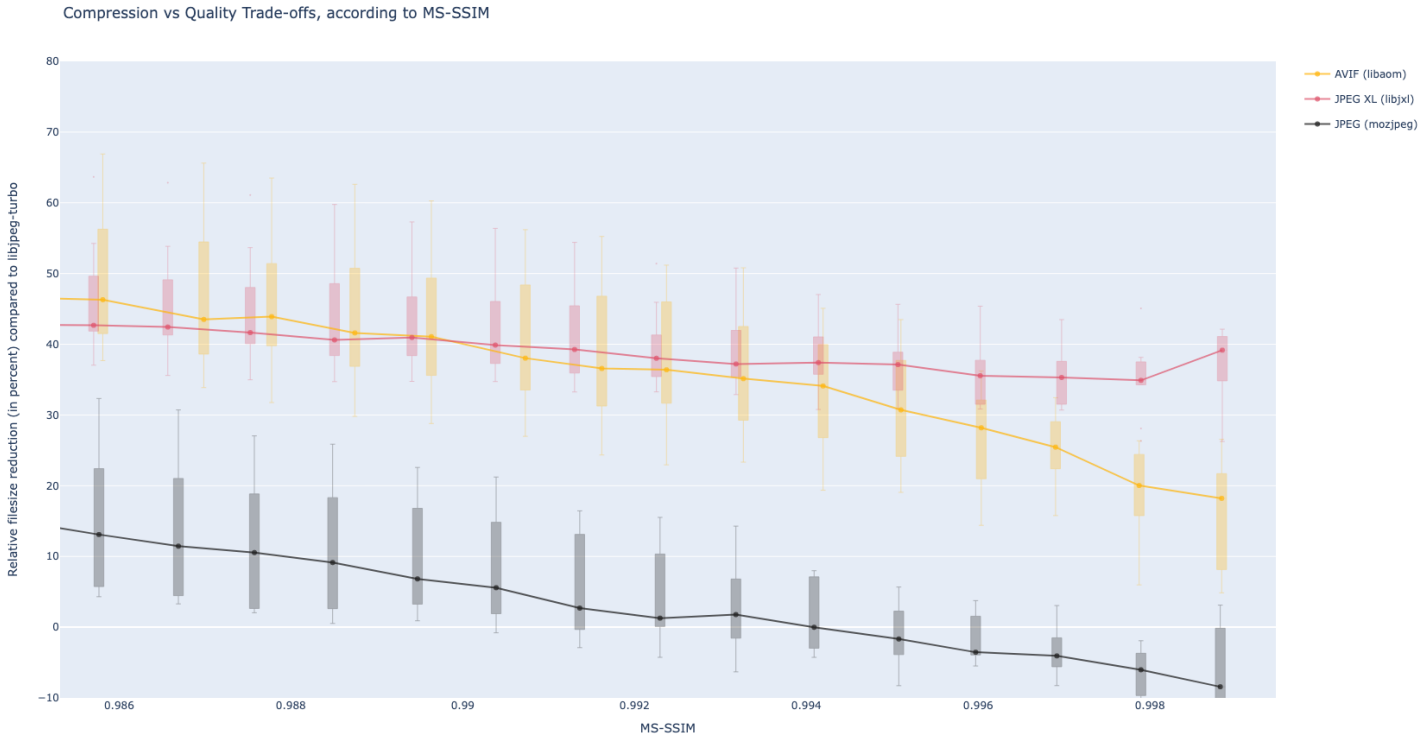
The following plots show the relative filesize reduction (in percent) compared to JPEG (libjpeg-turbo) at a given quality as estimated by the SSIM and MS-SSIM objective metrics. The source images are the ones from the JPEG AIC CTC[2] dataset. The following encoders and command lines were used:

- MozJPEG (version 4.1.2): `cjpeg -quality {10..100} -sample 1x1 input.ppm > output.jpg`
- JPEG XL (libjxl 0.8): `cjxl input.ppm output.jxl -q {5..100} -e 6 --num_threads=0`
- AVIF (libavif 0.11.1, libaom 3.5.0): `avifenc -c aom -y 444 --min 0 --max 63 -s 6 -j 1 -a end-usage=q -a tune=ssim -a cq-level={1..63}`

[2] ISO/IEC JTC1/SC29/WG1, "JPEG AIC Common Test Conditions for subjective quality assessment" output document wg1n100334, October 2022. https://ds.jpeg.org/documents/jpegaic/wg1n100334-097-ICQ-JPEG_AIC_CTC_for_subjective_quality_assessment.pdf Accessed: 2023-01-17.

The CPU type used was Intel® Xeon® Platinum 8259CL CPU @ 2.50GHz. With the above encoder settings, this led to the following relative encode speeds: in the time AVIF encodes 1 megapixel, JPEG XL and MozJPEG encode 4 megapixels. The metric score ranges shown in the plots below start at the average metric score for images encoded with libjpeg-turbo at quality 30. Both metrics are as implemented in VMAF 2.3.1.





Codec Architecture

JPEG XL has two coding modes, shown in the encoder diagram in Figure 3:

- VarDCT (green): lossy encoding of photos with an emphasis on the human visual system; can also be used for lossless transcoding of existing JPEG images,
- Modular (red): mathematically lossless encoding or generic lossy encoding.

VarDCT mode uses Modular mode to store sub-images such as LF (8x8 subsampled image, identical to the DC of JPEG in case only 8x8 blocks are used), control fields (e.g. for colour correlation and filtering), and extracted image features ("patches") as explained further below. For lossless transcoding of JPEG bitstreams (to reconstruct not just the image data, but also the bit-exact file), optional JPEG bitstream reconstruction data can be stored (shown in yellow).

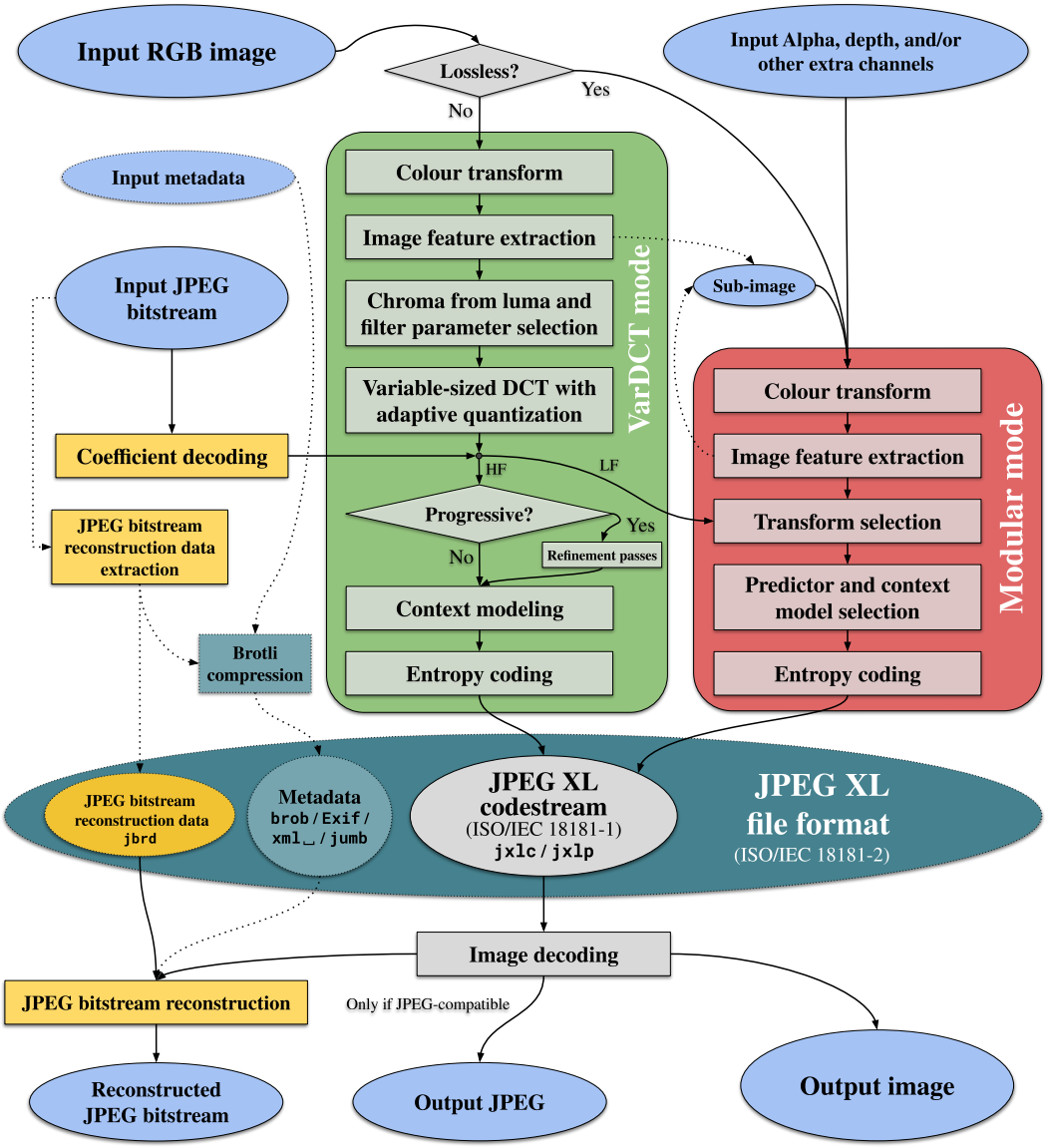


Fig. 3: Overview of the encoder architecture and associated functionalities. An input image is encoded as a compressed JPEG XL codestream in a lossless or lossy way, and decoded to an identical or perceptually similar output image. An input JPEG bitstream can be encoded as a JPEG XL file, containing a codestream that uses a subset of the VarDCT mode coding tools (to represent the image data) as well as JPEG bitstream reconstruction data, which can be decoded to an identical reconstructed JPEG bitstream.

Coding Tools

A JPEG XL file can contain one or more frames, e.g. for animation or overlays. Frames may be subtracted from prior frames without motion compensation and can have smaller bounding boxes, blend modes and dispose modes. They may also reference regions from prior frames marked as reference frames, which can serve as a simple **inter-frame** coding tool and as a **screen content** coding tool for still images.

Each frame is split into groups for **parallel encoding and decoding** and efficient **region-of-interest** (cropped) decoding. The group size for VarDCT mode is 256x256 pixels, and 128x128 to 1024x1024 in Modular mode. Besides thread-parallelism, the coding tools have also been designed for SIMD (single instruction, multiple data), making the JPEG XL codec suitable for efficient CPU encoding and decoding on current desktop and mobile processors.

Special attention has been devoted to **reducing header and metadata overhead**, including compression of colour profiles and Exif/XMP metadata, which is especially important for web delivery of smaller images.

A brief description of other JPEG XL coding tools follows:

- **XYB colour space**[3] is an LMS-based colour model inspired by the human visual system, facilitating perceptually uniform quantization. It uses a gamma of 3 for computationally efficient decoding.
- **Variable-sized DCT** (square or rectangular from 2x2 to 256x256) is used to model the high frequency information of images. An encoder can use smaller blocks where needed and larger, more efficient blocks where possible for optimal compression. DCT coefficients may be sent in arbitrary order.
- **Image features** are rendered on top of the decoded image, for precise and dense representations of repeated elements (“patches”), curvilinear image features (“splines”), and adaptive, intensity dependent synthetic noise modelling.
- **Chroma from luma** allows local colour decorrelation using signalled multipliers.
- An **edge-preserving filter** with fine-grained (8x8) strength signalling helps to suppress artifacts while preserving fine detail.
- The **Gabor-like transform** effectively provides the advantages of a lapped transform without the disadvantages of that approach. Both the edge-preserving filter and the Gabor-like transform are applied on the whole image, so also across group borders.
- **Adaptive quantization** uses per-block quantization step sizes which also guide the restoration filter.
- **Prediction** is run using a pixel-by-pixel decorrelator without side information, including a parametrized self-correcting weighted ensemble of predictors.
- **Context modeling** includes specialised static models and powerful meta-adaptive models that take local error into account, with a signalled tree structure and predictor selection per context.

---

[3] Starting with RGB samples with the sRGB primaries and a *linear* transfer curve (with range 0..1 for colors in the sRGB gamut), the default forward XYB transform is as follows:

$$bias = 0.0037930732552754493$$
$$Lmix = 0.3\,R + 0.622\,G + 0.078\,B + bias$$
$$Mmix = 0.23\,R + 0.692\,G + 0.078\,B + bias$$
$$Smix = 0.24342268924547819\,R + 0.20476744424496821\,G + 0.55180986650955360\,B + bias$$
$$Lgamma = \sqrt[3]{Lmix} - \sqrt[3]{bias}$$
$$Mgamma = \sqrt[3]{Mmix} - \sqrt[3]{bias}$$
$$Sgamma = \sqrt[3]{Smix} - \sqrt[3]{bias}$$
$$X = (Lgamma - Mgamma) / 2$$
$$Y = (Lgamma + Mgamma) / 2$$
$$B = Sgamma$$

- **Entropy coding** is LZ77-enabled and can use both Asymmetric Numeral Systems and Huffman (for low complexity encoders or for reducing overhead of short streams); integers of arbitrary length are supported through a configurable splitting mechanism of integers into symbols and raw bits.
- **Modular transform** chains can be signaled for an entire frame or per-group, and include reversible colour transforms (including a generalised delta palette transform) and a modified nonlinear Haar wavelet, which enables efficient and progressive decoding.

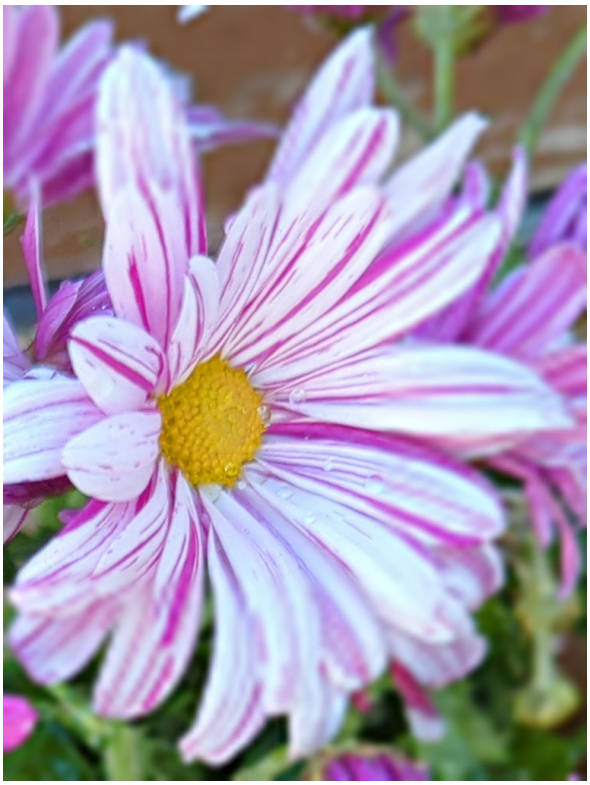
Progressive Decoding

JPEG XL has several features related to bitstream ordering to allow **progressive refinement** during streaming decoding, which is relevant for responsive web image delivery:

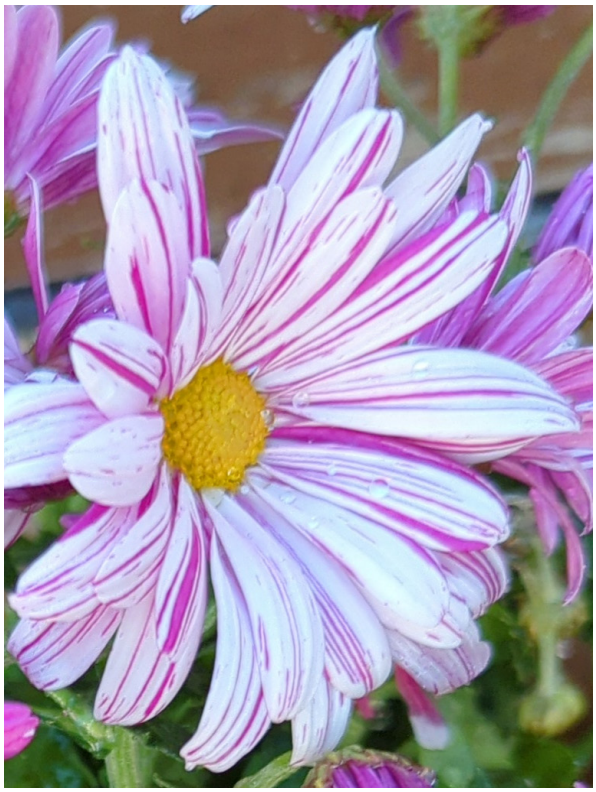
- **LF first.** In VarDCT mode, the 1:8 low frequency information is always encoded first in the bitstream, which allows rendering a good approximation of the final image (cf. Figure 4a).
- **Group permutation.** The image is divided into groups of 256x256 pixels. This will lead to more than 100 tiles for a 4k-image. The format allows to store the passes (or single pass) for those groups in an arbitrary order in the bitstream. This can for example be used to start with the most salient parts of the image, spiralling outwards in squares around the initial tile. Figure 4 illustrates group ordering on a small example.

(a) 1:8 LF image, rendered with smoothing

(b) A prescribed order for the groups



(c) After decoding the first four groups



(d) Final image

Fig. 4: Example of (a) the 8x upsampling used to display a progressive preview when only the 1:8 LF information is available; (b) an example group permutation; (c) a progressive preview when the first few groups are decoded; (d) the final image.

- **Recursively progressive LF.** The 1:8 image can optionally be encoded as a separate frame (cf. Figure 5), and this can be done recursively to create a LF pyramid of up to 4 levels, corresponding to 1:8, 1:64, 1:512 and 1:4096 previews for gigapixel-class images.

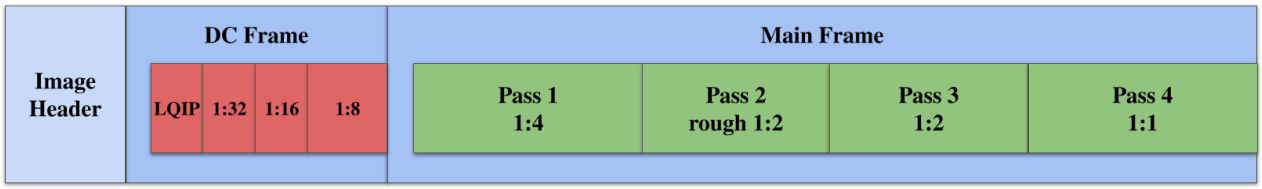


Fig. 5: Example bitstream structure for progressive refinement with multiple passes and LF encoded as a DC frame

- **HF passes.** The high-frequency (HF) information can be split into several passes, which can be used for spectral selection and successive approximation passes (as in progressive JPEG), but also for saliency-based passes since the same coefficients can be sent in multiple passes and are added together.
- **Multi-layer encoding.** Zero-duration frames can be blended together in order to encode a composite still image. Layers of type kRegularFrame are shown progressively as they are loaded, while layers of type kSkipProgressive can be used to hide layer progressions.

Encoder Coding Tool Selection Guideline

When designing or configuring a JPEG XL encoder for a specific use case, the following table can provide some guidelines on which coding tools to select. Each column in this table corresponds to a set of use cases, including but not limited to the following:

- Lossy photo: consumer-grade digital cameras, photos on websites or social media, …
- Lossy synthetic: casual viewing of rendered images, illustrations, digital art, comics, …
- Near lossless: prosumer-grade digital cameras, high-quality renditions
- Low bpp: lower quality, bandwidth-optimised photos, e.g. web previews
- Lossless photo: professional photography, medical or scientific imagery, archival, cultural preservation
- Lossless synthetic: master renders, authoring workflows for illustrations etc.
- Lossless JPEG transcoding: archival of existing JPEG-encoded images, transition solutions
- Fast lossy encoding: hardware implementations, e.g. in a smartphone or digital camera
- Fast lossless encoding: screenshots, authoring workflows
- Fast decoding (encoding with the aim of allowing fast decoding): web delivery, low-power devices
- Progressive: web delivery

From the encoder point of view, nearly all coding tools in JPEG XL are optional. A very simple encoder can be essentially as simple as a JPEG encoder (using VarDCT with only 8x8 DCT, no filters or image features, Huffman only, etc.) or similar to a PNG encoder (using Modular without transforms, Huffman + LZ77 only).

		Lossy				Lossless			fast encoding			
	Coding tool	photo	synth.	near lossless	low bpp	photo	synth.	JPEG	lossy	lossless	fast dec	progressive
Entropy coding & Context model	ANS entropy coding	■	■	■	■	■	■	■	■	■	■	■
	Huffman entropy coding	■	■	■	■	■	■	■	■	■	■	■
	2D-aware LZ77		■	■		■	■	■		■	■	■
	Custom hybrid int config	■	■	■	■	■	■	■			■	■
	Static context model	■	■	■	■	■	■	■	■	■	■	■
	Context clustering	■	■	■	■	■	■	■			■	■
	MA context model	(LF)	(LF)	■	(LF)	■	■	(DC)			■	■
Modular	Palette transform		■			■	■			■	■	
	Delta palette		■	■	(LF)							
	Squeeze transform	(LF)	(LF)	■	(LF)	■	■					■
	RCT (reversible colour transform)			■		■	■			■	■	■
	Self-correcting predictor					■				■	■	
VarDCT	XYB colour space	■	■	■	■				■		■	■
	Chroma from luma	■	■	■	■			(444)	■		■	■
	Variable-sized DCT	■	■	■	■			(8x8)	■		■	■
	Adaptive quantization	■	■	■	■				■		■	■
	Adaptive LF smoothing	■	■	■	■			■	■		■	■
Bitstream structure	Multi-layer encoding	■	■	■	■	■	■					■
	LF frame	■	■	■	■			■			■	■
	Passes	■	■	■	■	■	■	■				■
	Group permutation	■	■	■	■	■	■	■	■	■	■	■
Restoration filters	Chroma upsampling							(420)	■		■	■
	Gabor-like transform	■	■	■	■				■		■	■
	Edge-preserving filter	■	■		■				■			■
Image features	Non-sep. upsampling	■	■		■						■	■
	Patches	■	■	■	■		■					■
	Splines	■	■	■	■							■
	Noise	■	■		■				■			■

Reference Software

The reference software **`libjxl`** is available at http://gitlab.com/wg1/jpeg-xl. The complete JPEG XL reference software is available under a **free and open source, royalty-free** license. The contributors have also declared to ISO/IEC that according to the Common Patent Policy they are offering a Type 1 Free of Charge patent grant. The software is available for a very wide variety of architectures, platforms and operating systems, including SIMD optimizations for x86 and ARM, and a web assembly (wasm) build.

In the reference software, encoder configuration is done by perceptual target, reliably reaching a desired visual quality, unlike the usual approach of configuring by means of bitrate or quantization settings, which results in a highly image-dependent visual quality. As a result, the JPEG XL (libjxl) encoder produces results that are visually consistent.

Conclusion

JPEG XL achieves efficient, reliable and high-quality image compression and responsive delivery. It is designed to be a practical and modern replacement for existing usages of JPEG, PNG and GIF formats, and compares favorably to other modern codecs like JPEG 2000 and AVIF.

Additional information is available at https://jpeg.org/jpegxl/.

Acknowledgements

We gratefully acknowledge contributions from: Jyrki Alakuijala, João Ascenso, Sami Boukortt, Martin Bruse, Iulia-Maria Comsa, Touradj Ebrahimi, Moritz Firsching, Thomas Fischbacher, Sebastian Gomez, Walt Husak, Renata Khasanova, Evgenii Kliuchnikov, Jeffrey Lim, Robert Obryk, Fernando Pereira, António Pinheiro, Krzysztof Potempa, Alexander Rhatushnyak, Thomas Richter, Jon Sneyers, Zoltan Szabadka, Evgeniy Upenik, Ruud van Asseldonk, Lode Vandevenne, Luca Versari, and Jan Wassenberg.