Jpeg-xl vs Avif vs Heic vs BCN vs Astc vs Webp vs Qoi vs Png vs Jpeg vs Jpeg2000 vs Exr vs Hdr vs Tiff

Multi Image Codec Introduction/Comparison

Larry Lin, Zhouyang He, Martin Yang

1. What's the problem? why is it important

- Compression rate

  As digital storage continues to expand exponentially while being more cost-effective, the typical consumer might be less concerned about image file size. But increasing the compression rate still remains impactful for AI researchers, professionals in graphic, large-scale storage clusters, and the load of overall network traffic.

- Decoding speed

  The speed of decoding may appear sufficient for the average user, but loading a single PNG might slow down the MCU in an embedded system. Even for modern PCs, the decoding rate can be unacceptable, as newer image codecs increase complexity by a factor of 10-100 or more. Also, most importantly, texture decoding has been a huge problem that requires GPU hardware and dedicated decoder.

- Compression Algorithm

  Many traditional algorithms did not consider the use of progressive or block encoding and were not designed with parallelism in mind. Concepts like resolution and different scales are not considered. Additionally, insufficient metadata design during standardization has caused significant problems for color management and future decoding.

- Field specific support

  Typical image codecs do not contain the information for field specific requirements（dynamic range, norm etc）, so that for 3d/vfx/photo/hdr, displaying, post processing, layer editing, and render passing is not possible.

2. What has been done? Why are they not sufficient? including any of your previous, other and ongoing projects too
   - With the advent of newer formats such as AVIF and HEVC, the encoding rate has notably increased. However, the decoding time has also seen a 10x increase, and 20-50x for encoding time .
   - While jxl manages to reach the encoding rate of AVIF and HEVC and keep the de/encoding at faster speed, it still requires sufficient power to convince google.
   - Dedicated format has been standardized for different purpose such as exr/hdr for the field of vfx/3d, bc6/bc7/astc/etc for gpu achieving on the fly fast decompression, but field specific codec introduced problem like architectural limitation, unacceptable file size (single image over 2Gb)
   - Improving algorithms, adding Simd or native specific optimization can largely improve performance in different magnitudes. But simultaneously increase the complexity of porting the codec to alternative platforms and architectures, as well as implementing it at register transfer level.

—-------------------------------

Research result:

Jpeg-xl:

universal, fast enc/dec, support for lossy/lossless/frames/hdr

Encode steps:

Image to adaptive chunks for var-dct(8x8-64x64 into 4x4-8x4) (lossy)

encode in layers of quality for progressive rendering

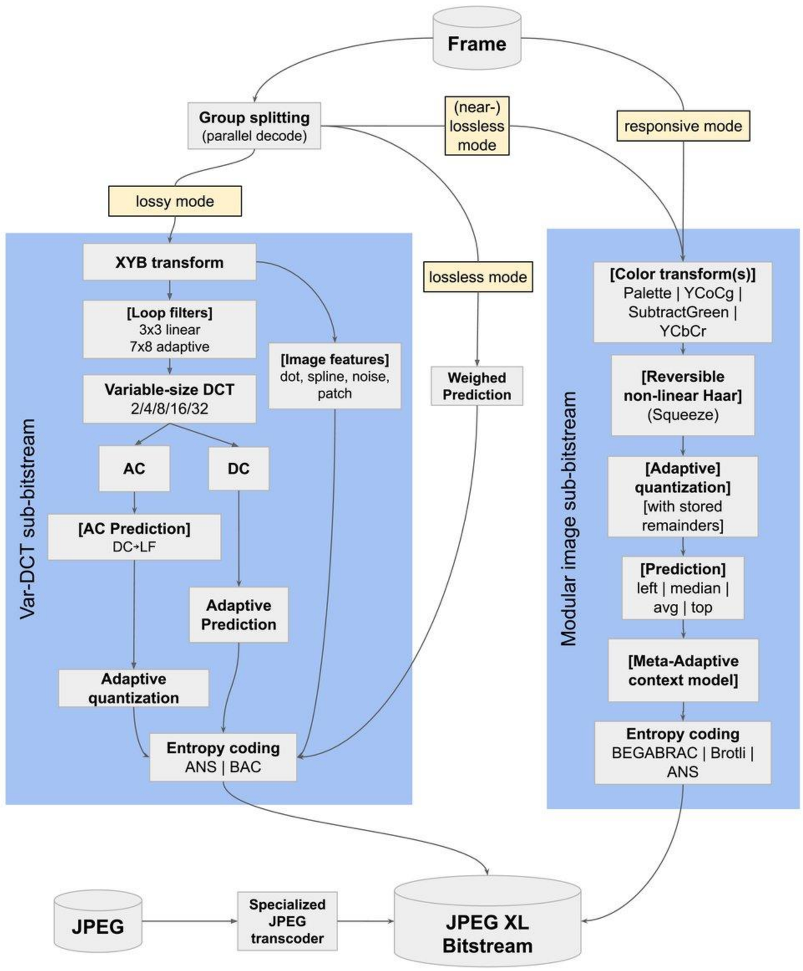
Luma S LM difference color space transform(XYB)

Locally adjusted adaptive quantization

Block, lines, noise in compressed representation for reference

Entropy encoding with Asymmetric Numeral System + lz77(optional)

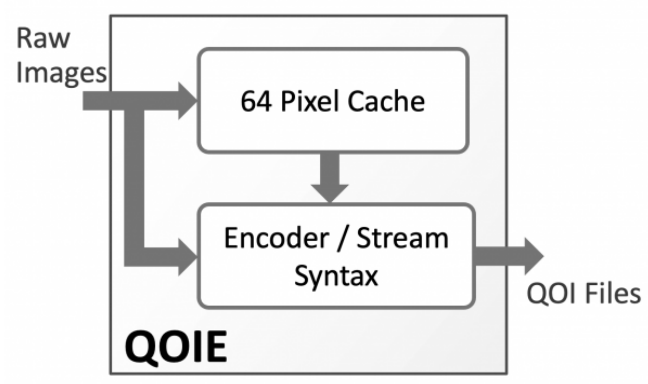
Weighted self correcting predictor adjusted per context with modeling (lossless)



Avif Heic:

Avif and Heic are part of the Av1/Hevc encoder where the frame based compression are used for storing individual image but achieving almost double the compression ratio of Png

Qoi Png：dc rate, dc speed, usage, improvement, limitation

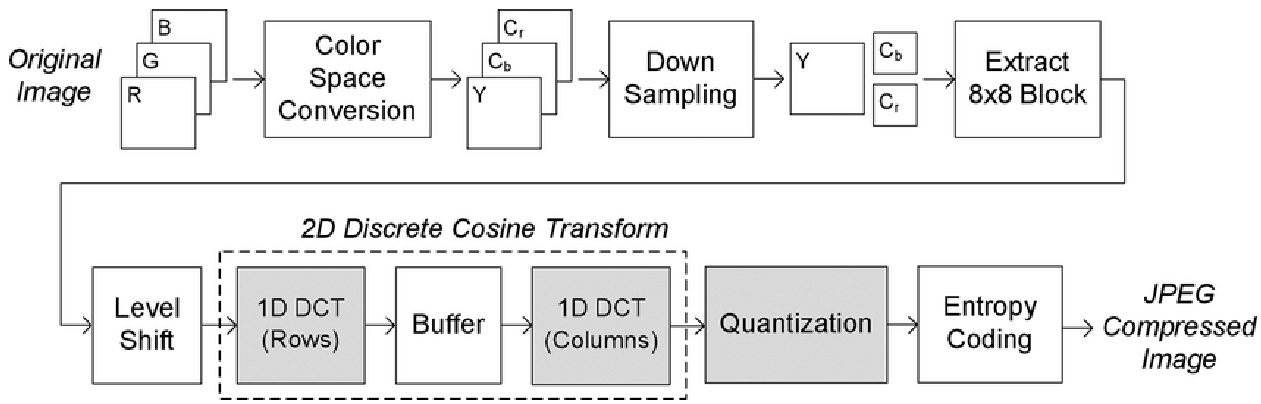


The QOI algorithm compresses RGB or RGBA images with 8 bits per color without any loss.

PNG employs the DEFLATE compression algorithm, a non-patented, lossless data compression technique. DEFLATE is a combination of two main components: LZ77 (Lempel-Ziv 1977) and Huffman coding. The LZ77 algorithm, also known as sliding window compression, identifies repeated sequences of data and replaces them with references to a single copy. This mechanism proves especially effective for compressing images with areas of uniform color or patterns. Huffman coding, on the other hand, is a variable-length encoding method that assigns shorter codes to more frequently occurring values, optimizing the representation of the data.

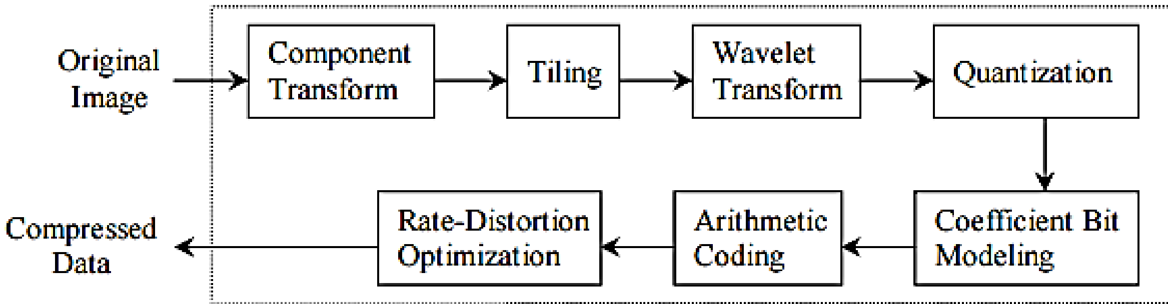
DEFLATE operates by first applying LZ77 compression, followed by Huffman coding. During LZ77 compression, the algorithm searches for repeated substrings within the data, creating a dictionary of these substrings and their corresponding references. The Huffman coding step then assigns variable-length codes to the dictionary entries based on their frequency of occurrence. The result is a compressed representation of the original data, where redundancy has been reduced without any loss of information.

Jpeg Jpeg2000 Webp：usage, improvement, limitation

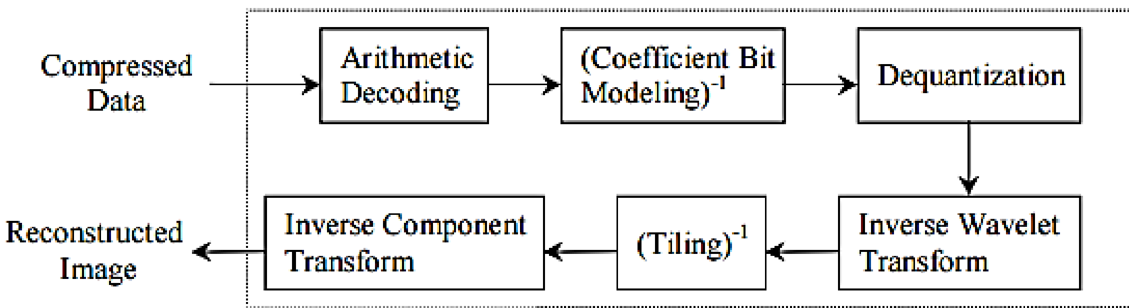


When JPEG first receives the raw data, it employs a color space transformation from RGB (Red, Green, Blue) to YUV (Luma, Chrominance). This transformation is based on the fact that the human visual system is more sensitive to changes in brightness (luminance or Y component) than to changes in color (chrominance or U and V components). By separating luminance and chrominance information, JPEG can take advantage of the human eye's characteristics to achieve more efficient compression. Following that, JPEG divides the YUV image into smaller blocks, typically 8x8 pixels. This block-based approach facilitates the application of compression techniques to individual units of the image. After that, JPEG employs a mathematical transformation known as the discrete cosine transform (DCT) to convert the image data from the spatial domain to the frequency domain. The DCT identifies patterns and variations in intensity within each block, allowing for the separation of image information into different frequency components. High-frequency components, representing fine details and rapid changes in intensity, are then quantized, meaning that their precision is reduced. This quantization step introduces some loss of information, contributing to the characteristic "lossy" nature of JPEG compression. Finally, JPEG uses entropy coding, specifically Huffman coding, to further compress the data. Huffman coding assigns variable-length codes to different symbols based on their frequency of occurrence. This coding scheme is particularly effective at representing more common values with shorter codes, resulting in a more compact representation of the image data.

While the compression achieved by JPEG is significant and allows for the efficient storage and transmission of images, the process inherently introduces some degree of image degradation due to the quantization of high-frequency components. The balance between compression ratio and image quality can be controlled by adjusting compression settings, allowing users to make trade-offs based on their specific needs. JPEG compression is widely used for photographic images on the web and in various digital media, where the compromise in image quality is generally acceptable for the benefits of reduced file size and faster transmission.



Encoder Processing Steps



Decoder Processing Steps

JPEG 2000 represents a significant advancement in image compression technology, utilizing wavelet transforms to achieve efficient encoding and compression. Unlike the original JPEG format, which relies on discrete cosine transforms (DCT), JPEG 2000 employs wavelet transforms, specifically the discrete wavelet transform (DWT). The wavelet transform allows for a more adaptive representation of image details across different scales and resolutions.

One notable feature of JPEG 2000 is its ability to provide superior image quality at lower bit rates compared to traditional JPEG. This is due in part to the efficient representation of image information through the wavelet transform, allowing for a more accurate capture of both high and low-frequency components. The use of wavelets enables the creation of a multi-resolution representation of the image, facilitating more effective compression while preserving essential details.

JPEG 2000 supports both lossless and lossy compression, providing flexibility to users based on their specific requirements. In lossy compression, users can adjust the compression ratios to achieve a balance between file size reduction and acceptable image quality, making it suitable for a wide range of applications. In scenarios where preservation of every detail is critical, JPEG 2000's lossless compression option ensures that no information is sacrificed during the compression process.

WebP is a versatile image format designed explicitly for web-based applications, combining elements of both lossless and lossy compression to optimize image delivery. In its lossy compression mode, WebP utilizes the VP8 methodology for frame prediction, a technique that efficiently predicts subsequent frames based on preceding ones, reducing redundancy in the image data. This predictive approach allows for significant compression while maintaining perceptual image quality, making it ideal for scenarios where a balance between file size and visual fidelity is crucial, such as web graphics and multimedia content.

On the other hand, WebP's lossless compression mode employs a series of transformative operations applied to the image data without sacrificing any visual information. These operations include techniques like color indexing, entropy encoding, and spatial prediction. By carefully managing the representation of pixel values and patterns within the image, lossless WebP compression achieves compression ratios comparable to other lossless formats, making it suitable for applications where preserving every detail is paramount, such as graphics with text and logos.

BCn Astc:

Texture compression format features fast decompression for on the fly frame and block specific decompression in multi resolution for the purpose of rendering, the algorithm also considers implementation in hardware but architecture might be limited (astc on arm only)

Exr Hdr Tiff:

The OpenEXR(exr) is a high dynamic range image compression that is often used in film and vfx. It is designed to store and accurately represent images with a wide range of arbitrary image channels. These image channels can include RGBA, luminance and sub-sampled chroma channels; depth, surface normal directions, motion vectors ..etc

High Dynamic Range(hdr) image is a format used in film, photography and other industries, and hdr image files contain a wide range of luminance values allowing for a scene with varying levels of brightness and contrast. It uses floating point representation to store a wide range of values including very large and very small ones. Due to it is file size, compress algorithms are necessary to efficiently store and transmit images such as tone mapping algorithms.

Tagged Image File Format (TIFF) is widely utilized within the realms of design, photography, and desktop publishing industries. TIFF files are characterized by their capacity to retain detailed data, primarily employing lossless compression methods such as Lempel-Ziv-Welch compression. However, it is worth noting that due to their substantial file size, TIFF images are not considered optimal for implementation in web design or applications prefer short loading times. Alternatives like JPEG or PNG formats are typically favored in these contexts.

TIFF format is capable of storing bilevel, grayscale, palette-color, and full-color images in multiple color spaces. TIFF supports both lossy and lossless compression methods, providing the flexibility to balance between storage space and processing time based on the needs of the applications utilizing the format. TIFF is not limited to processors, operating systems, or file systems.

TIFF is not only an image file extension and it is also an image container. Each subfile within TIFF uses a data structure called an image file directory (IFD) to handle image entries in the subfile. Each IFD contains one or several entries and is managed by its tag. The tags are arbitrary 16-bit numbers and define its dimensions, color information, compression method, and other details.

TIFF IFD typically includes:

- Image dimensions specify the width and height of the image in pixels;
- Color information is about the color space used in the image, such as RGB, CMYK, grayscale, etc;

- Compression information defines the compression method used to reduce file size (e.g., LZW compression);
- Bits per Sample indicates the number of bits used to represent each color component.
- Image Data stores a value pointing to the actual image data in the file;
- And Various other entries can include information like date and time of creation, software used, authorship, etc.

TIFF images mainly use lossless compression method and are also capable of using lossy compression like JPEG. TIFF supports a number of compression methods and Lempel–Ziv–Welch (LZW) compression is one of them. LZW compression is a dictionary based compression scheme. LZW compression commonly uses a table with 4096 entries and the first 256 entries in the table represent single bytes for the input. The table starts with a set of individual characters as entries and is typically initialized with the unique symbols present in the input data.

The encoding process begins with:

1. Read a input character A
2. If A exist in the table:
    a. If yes, read next input character B, and join A and B and assign it to A, and then go to step 2
    b. If no, create a new entry with joining A and B. Then output the code of A and assign B to A and go to step 1
3. Repeat until reaching the end of the input.

Decoding is similar to encoding and the full dictionary table does not need to be provided from the encoder. The encoder provides the table with the initial symbols or informs the decoder about the table they used like ASCII.The algorithm builds the original file with the table and uses revealed characters to reconstruct the dictionary table with the encoder algorithm in order to restore the original file. However, if the encountered code is not yet recorded in the dictionary during decoding, the algorithm will use the first single symbol of the current string and append it to the end of the string and make a new entry into the dictionary.

The LZW compression is not the best compression method to use in very small files or files with minimal redundancy. The dictionary overhead could exceed the size of the original files. The dictionary size could also be a problem with a large file. It could require large memory space during encoding and decoding. These two problems could add up together. If a large file with minimal redundancy is compressed with LZW, it will lead to a huge dictionary table and the compressed file will be much larger than the original.

Statistic:

Physical capabilities:

Jpeg2000: 4294x4294 mp + 16384 channel

Png: 2147x2147 mp + 4 channel

Jpeg-xl: 1073x1073 mp + 4099 channel

Jpeg: 65x65 kp + 4? channel

webp: 16x16 kp + 4 channel

Avif: 8x4 kp + 5 channel

Heic: 8x4 kp + 5 channel

Uncovered graphical format:

Common or used in specific field:

gif, ico, xmb, dds, bmp, pnm, flif, rast, tga, s3tc, bc45, etc, pvrtc, eac, dxtc, svg, raws/container(psd/clip/ai/pdf/epsxcf/etc)

uncommon:

Bgp, fits, pcx, jpeg-xr, sgi, pik, ecw, jpeg-xt, …. +100 more

The Future of Image Format:

JPEG, PNG

—-------------------------------

3. What's your approach? Why can it do better or differently?

   Future Approach 1:

   Generative encoding：

1000x compression ratio, represent 10MB image in 50-500B

Approach 2:

Lossy encoding:

Decent quality and good encoding ratio for all kind of texture(include extreme case)

Lossless encoding:

Compress the file more efficiently by applying multi-level encoding

4. Expected deliverables and a rough biweekly time schedule

   Oct. 21: Background research

   Nov. 4: Midterm update

   Nov.20: Demo and adjustment

   Dec: Final report and presentation

5. website url; any resources needed, references if any, etc

   https://shaoshanfan.wixsite.com/my-site-1

   References:

   https://jpegxl.info