JPEG 2000

EE-569 - Image & Video Coding

Frédéric Piguet

Table of Contents

1. Dataset
   1. Requirements
   2. Constraints
   3. Our Dataset
2. JPEG 2000 – Summary
   1. How it works
   2. What it was built to solve
3. Implementations
   1. Kakadu
4. Results

Dataset - Requirements

- An image dataset for image compression:
  - Built in accordance with the target **use cases of the codecs**
  - **Variety** to cover a wide range of real-world scenarios:
    - Different *resolutions*
    - Different *colors*
    - Different *scenes*
    - Different *acquisition* & *generation* methods

Dataset – Constraints

Bit Depth
& Specialized Content

- For **fair comparison**, all codecs must be able to process the dataset:
  - This constrains the **bit depth** of the images
  - ➢ Using images with unsupported bit depths would lead to an unfair comparison

- Avoid **specialized content** requiring domain-specific knowledge:
  - Medical images
  - Astronomical images

  ⋮

Dataset – Constraints *Licenses*

- Images must be **freely usable** for research and academic purposes
- Licenses must allow:
  - Compression experiments
  - Quantitative evaluation
  - Visualization and presentation of results
- Preference for **minimally restrictive licenses**:
  - To ensure reproducibility and publication
  - Avoid licenses that may **restrict reuse or publication of results**

Dataset – Our Dataset

Original datasets

- Dataset composed of **15 images** from:
  - JPEG AIC-3 Dataset *5 images*
  - CLIC 2022 *10 images*

00003_945x840.png
945x840 px
8-bit depth

crosswalk_backlight_2048_1360.png
2048x1360 px
8-bit depth

Dataset – Our Dataset *Choices*

- Variety of images representative of common use cases:
  - Landscapes
  - Faces
  - Animals
  - Low-light and night scenes
  - Bright and overexposed scenes
- Bit depth: 8 bits per channel
- Image sizes ranging from *840px* to *2592px*
- Permissive licenses (compatible with research and publication)

Dataset – Our Dataset *JPEG AIC-3*

- **Created by:** JPEG Committee, MMSPG EPFL
- **Purpose:** Dataset for subjective image quality assessment
- **Content:**
  - Natural and computer-generated images
- **Size:** 10 uncompressed images
- License: *Creative Commons CC0*
  - Publicly available for research and evaluation purposes

Dataset – Our Dataset *CLIC*

- CLIC: Challenge on Learned Image Compression
- **Purpose:** benchmark dataset for image compression research and challenges
- **Content:**
  - Natural images from real-world photography
  - Diverse scenes, textures, and lighting conditions
- **Size:** several thousand high-resolution images
- License: Permissive license (*Unplash License)*
  - Download and use freely for commercial and non-commercial use

JPEG 2000 *Standard Components*

- Pre-processing and color transform
- Discrete Wavelet Transform (DWT)
- Quantization
- Block-Based Adaptive Binary Arithmetic Coder
- Codestream organization & scalability

- Optional tools (ROI, tiling, progression modes)

JPEG 2000 *What it solves*

- Design goals of JPEG 2000:
  - Higher compression efficiency than JPEG 1
  - Graceful degradation at low bitrates
  - Progressive (quality & resolution)
  - Region-of-interest coding/decoding (ROI)
  - Robustness to errors

JPEG 2000 *Real applications*

- Digital cinema (DCP)
- Medical imaging
- Remote sensing & satellite imagery
- Cultural heritage & digital archives
- Broadcasting

Implementations

- JPEG 2000 has various implementations
  - They can be found on the JPEG website:
    - OpenJPEG (openjpeg.org)
    - JasPer (ece.uvic.ca)
    - JJ2000 (code.google.com)

- Chosen implementation: **Kakadu**
  - kakadusoftware.com

Implementations *Kakadu*

- Proprietary implementation of JPEG 2000
  - Made available for demonstration purposes
- High-performance JPEG 2000 encoder/decoder
- Highly optimized encoder
- Widely used in industry:
  - Apple Inc. Quick Time
  - Google Earth
  - Medical imaging (DICOM - Digital Imaging and Communications in Medicine)

    ⋮

Implementations *Kakadu*

- JPEG 2000 is made of different *parts* (17 parts)
- Kakadu supports:
  - **Part 1**, ISO/IEC 15444-1 – **Core coding system**
    - Baseline JPEG 2000 image coding

    Only part used in this study
  - Part 2 (coding and file format extensions)
    - Extended transforms
  - Part 3 (motion JPEG 2000)
  - Part 9 (JPIP)
  - Part 15 (HTJ2K)

Implementations *Encoding parameters*

- Main encoding parameter:
  - **Target bitrate (bpp)**
    - Direct control of compression rate
    - Enables fair comparison across codecs
- Other important parameters (fixed):
  - Wavelet transform (5/3 or 9/7)
  - Number of decomposition levels
  - Code-block and tile sizes
  - Number of quality layers

Implementations *Influence of parameters*

Parameter	***Typical values / format***	***Controls***	***Influence on quality & bpp***
Target bitrate (bpp)	Continuous (float)	Total bit budget	Primary rate–distortion tradeoff, saturates
Quality	Continuous (0 to 100)	Tunes quantization w.r.t rate-distortion slopes	Controls rate-distortion tradeoff
Wavelet transform	5/3 (lossless), 9/7 (lossy)	Energy compaction	Compression efficiency
Decomposition levels	Integer	Multi-resolution depth	Low-rate quality, scalability
Code-block size	32×32, 64×64, …	Size of block decomposition	Rate-distortion efficiency vs overhead
Quality layers	Integer (e.g. 1–10)	Bitstream truncation points	Rate control

Implementations *Other parameters*

- Additional Kakadu features
  - Quality factor
    - Controls the rate based on desired quality
    - Redundant with the (more precise) rate command
    - Useful to reach higher bpp
  - Region of Interest (ROI)
    - Selective quality refinement
    - Not used to ensure fairness
  - High Throughput mode (HTJ2K)
    - Faster entropy coding
    - Outside the scope of this study

Results

Codecs

- All 15 images were compressed using 5 different codecs:
  - **JPEG 2000**
  - JPEG XL
  - HEIF
  - AVIF
  - WebP

Results

Bit Per Pixel Rates

- Four target bitrates were defined for each image:
  - *Lower bound*: JPEG 60% quality
  - *Upper bound*: 90% of the minimum lossless bpp achieved among all codecs
  - Using a linear interpolation for two intermediate target bpp

Linearly increasing target bpp →

Images	JPEG 60% quality bpp	interpolation 1	interpolation 2	0.9min lossless bpp
00002_853x945.png	0.60	2.33	4.06	5.78
00003_945x840.png	0.53	2.32	4.11	5.90
00007_1600x1200.png	0.77	2.64	4.50	6.36
00009_2048x1536.png	0.62	2.33	4.03	5.74
00010_2592x1946.png	0.73	2.36	4.00	5.63
apple_tree_1365_2048.png	0.46	2.00	3.55	5.10
bridge_1848_1224.png	1.15	2.53	3.92	5.30
celebration_2048_1365.png	0.53	2.00	3.47	4.94
crosswalk_backlight_2048_1360.png	1.43	4.10	6.77	9.43
neon_2048_1365.png	1.31	4.21	7.11	10.01
night_event_1463_2048.png	1.05	3.82	6.59	9.36
portrait_veil_1391_2048.png	0.86	2.61	4.37	6.12
rapeseed_field_2048_1365.png	0.98	3.32	5.66	8.00
street_dusk_2048_1135.png	0.87	3.42	5.97	8.52
video_game_2048_1152.png	0.49	2.07	3.66	5.24

Results *Bit Per Pixel Rates*

- The achieved bpp using the *rate* command saturates before the maximum target:
  - *Rate* control truncates the final bitstream to achieve the target bpp
  - Limits its bpp range, unlike *quality* command which leverages the quantization steps
- Replacing it with the quality control allows reaching the maximum target
- Comparison at the highest bitrate must be interpreted with care, as different control modes are used

Results *Metrics*

- For codec comparison, we used 4 metrics:
  - PSNR — Peak Signal to Noise Ratio
  - MS-SSIM — Multi-Scale Structural SIMilarity
  - VIF — Visual Information Fidelity
  - BRISQUE — Blind/Referenceless Image Spatial Quality Evaluator

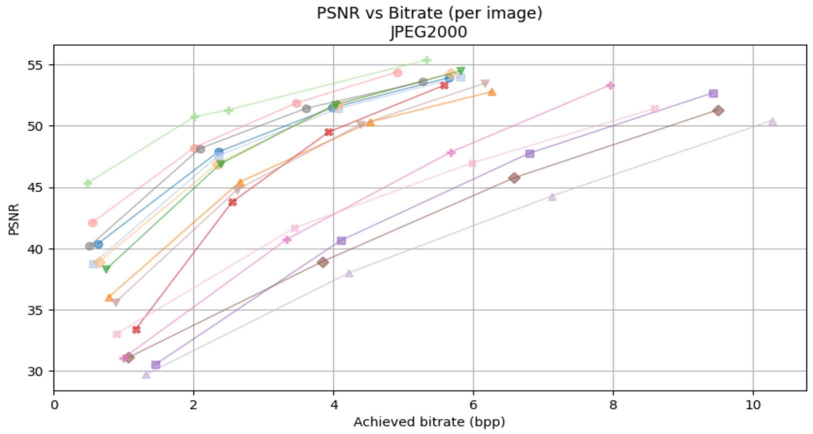
- BPP: Bits Per Pixel
  - Quantifies the number of bits necessary to code a pixel

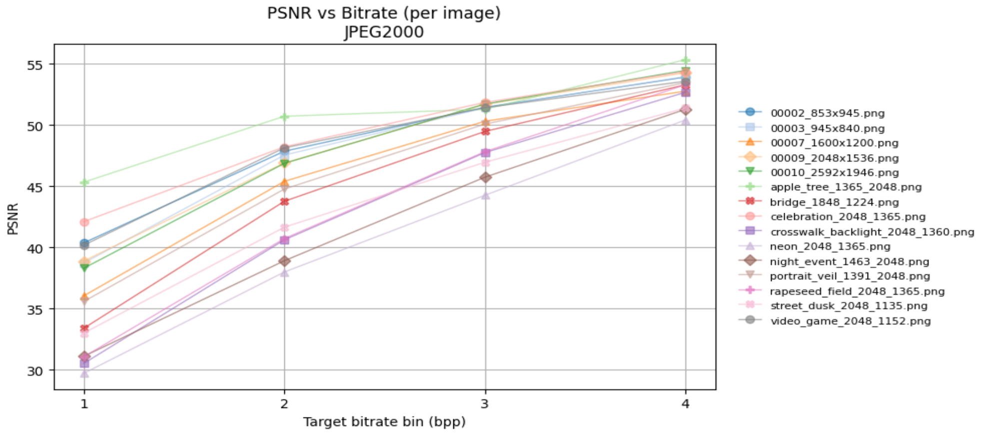
$$BPP = \frac{\text{Image file size [bits]}}{\text{Image size [pixels]}}$$

Results

PSNR

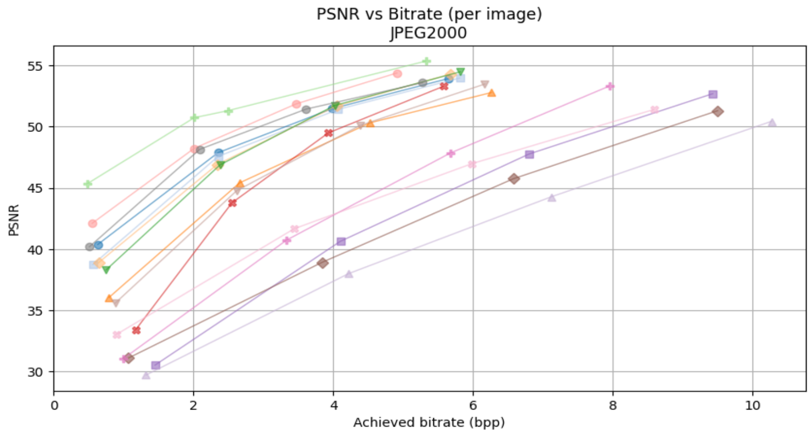
- Let's compare how PSNR evolves for each picture over every bpp
- Both graphs show:
  - Achieved PSNR for *each image* at *each bitrate*
  - Left graph: Real Achieved bpp vs. Achieved PSNR
  - Right graph: bpp bin vs. Achieved PSNR

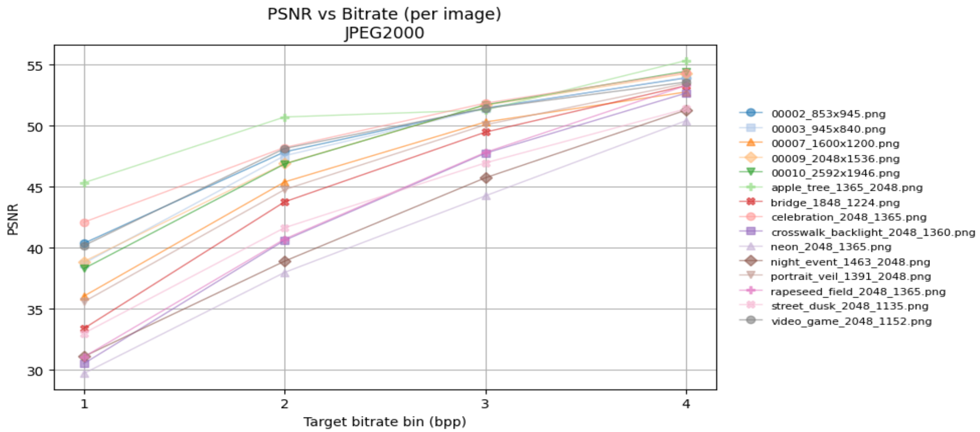




Results *PSNR*

- Let’s compare how PSNR evolves for each picture over every bpp
- Observations:
  - Expected positive evolution
  - Images with lower maximum bpp suffer less noise distortion at low coding rates





Results

PSNR

- PSNR is very sensitive to noise
- This might not be relevant to all applications for the HVS
- Here's the worst case: *neon_2048_1365*

Results

PSNR

- PSNR is very sensitive to noise
- This might not be relevant to all applications for the HVS
- Here's the worst case: *neon_2048_1365*

bpp = 24 bits/px
Uncompressed

bpp = 1,32 bits/px
Lowest bpp compression

Results

PSNR

- PSNR mainly reflects pixel-wise noise and does not account for perceptual structure

bpp = 24 bits/px
Uncompressed

bpp = 1,32 bits/px
Lowest bpp compression

Results

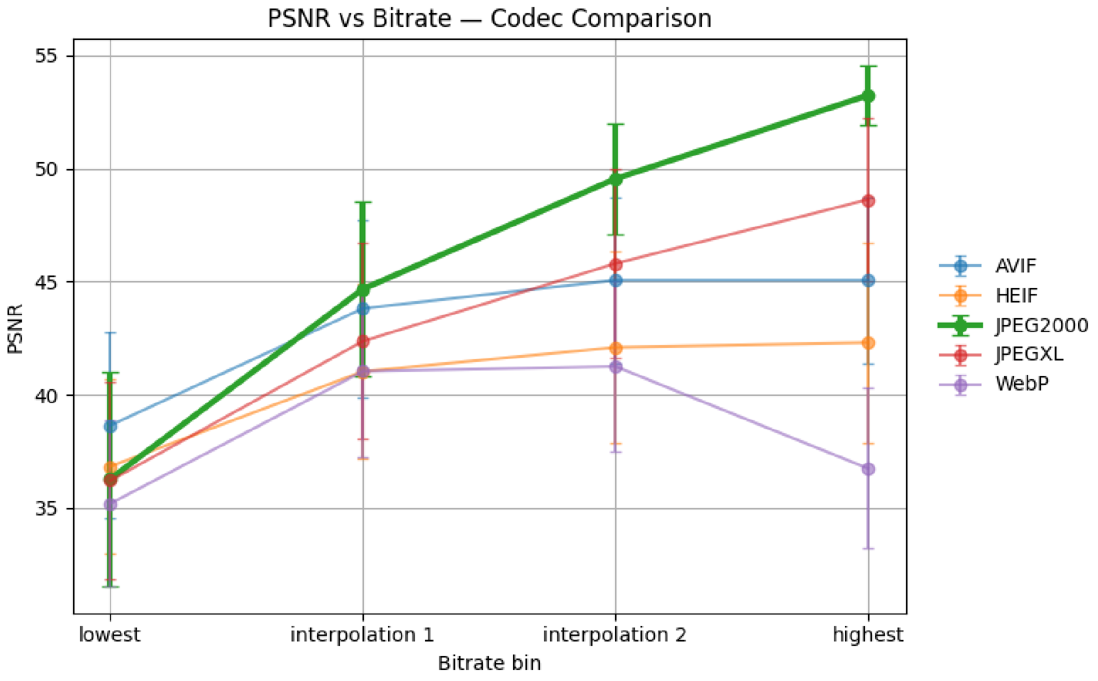
PSNR across codecs

- Taking the average of PSNR
- For each bpp target bin
- For each codec

- Error bars: Standard deviation σ

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1}(v_i - \mu)^2}$$

- σ : standard deviation
- N : number of samples
- v_i : sample i's values
- μ : population mean (average)

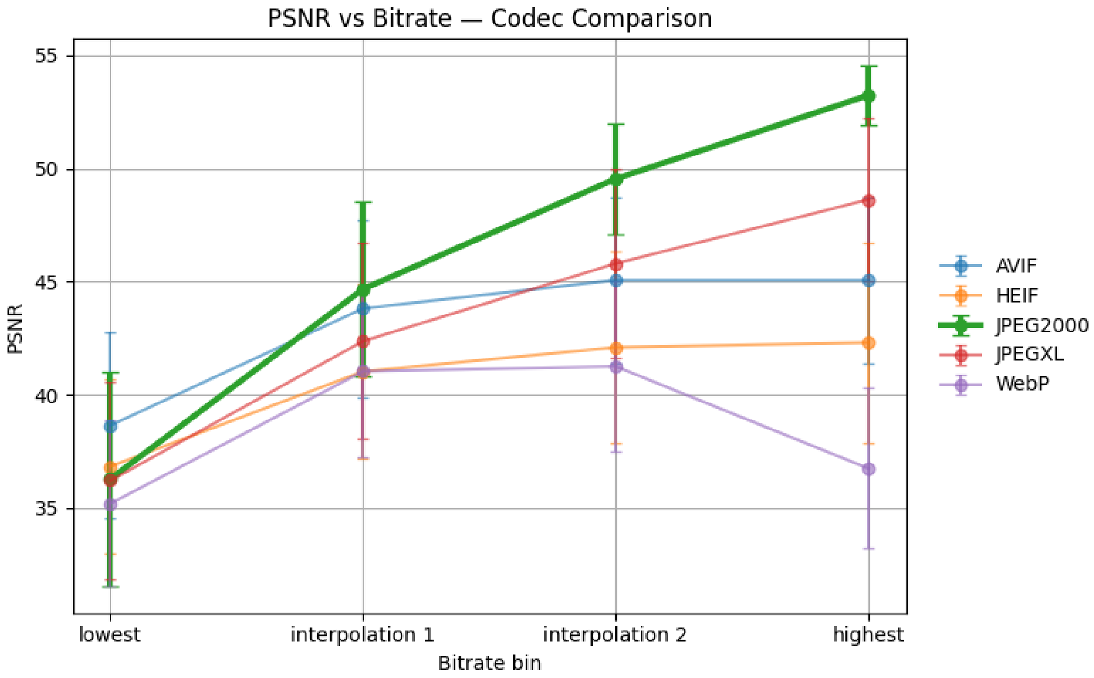


Results

PSNR across codecs

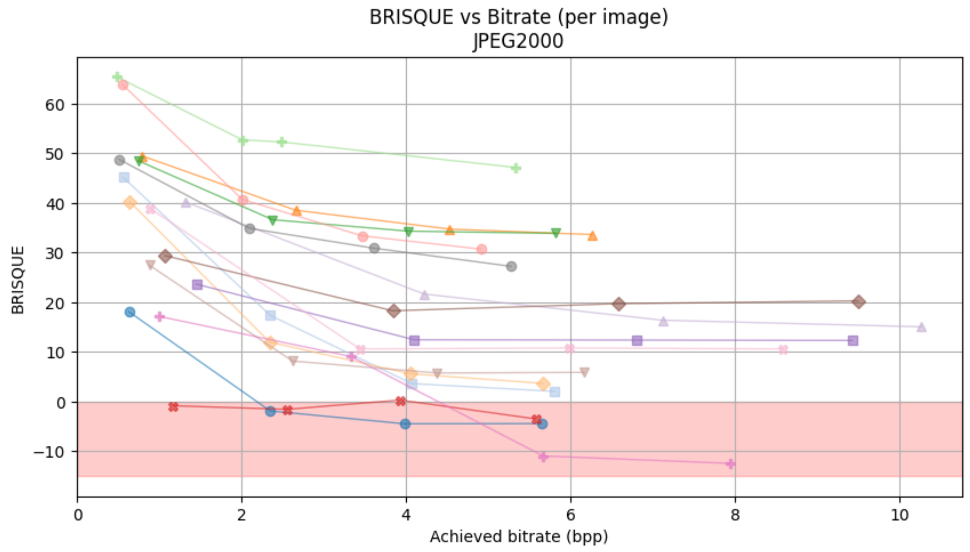
- At higher bitrates:
  - JPEG 1 standards perform better
  - JPEG 2000 achieves higher PSNR

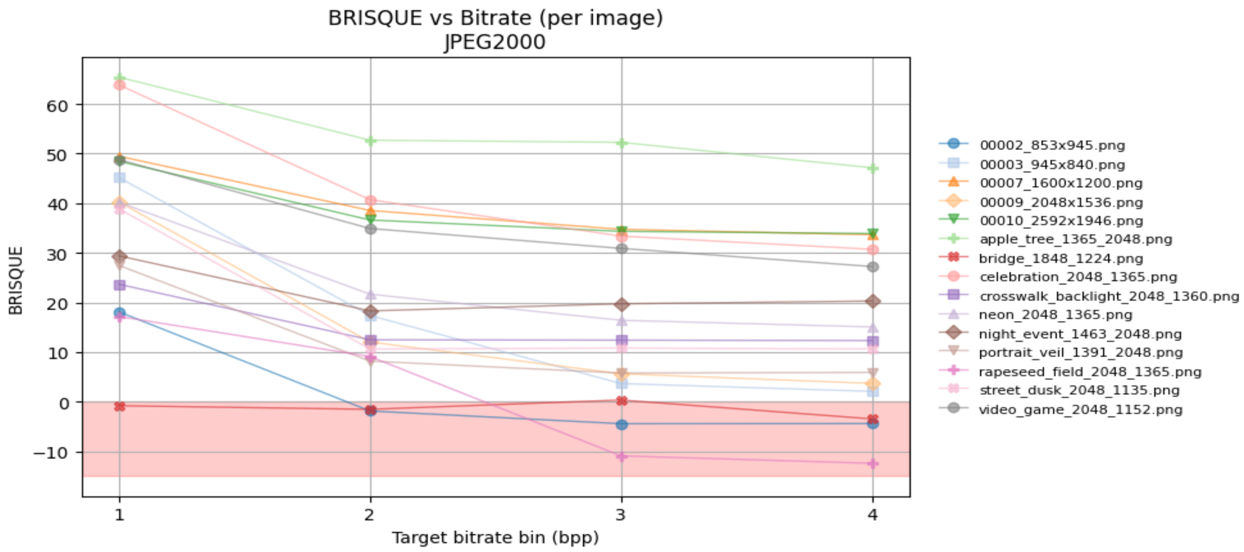
- At lower bitrates:
  - JPEG 2000 drops



Results *BRISQUE*

- Gives a global quality trend
- Instability at high bitrates
- Some values fall outside BRISQUE’s meaningful operating range

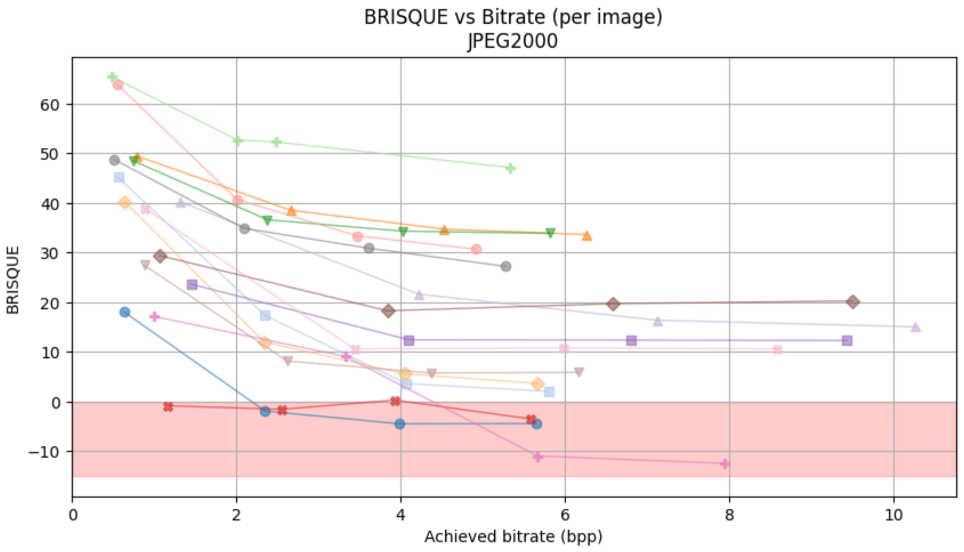


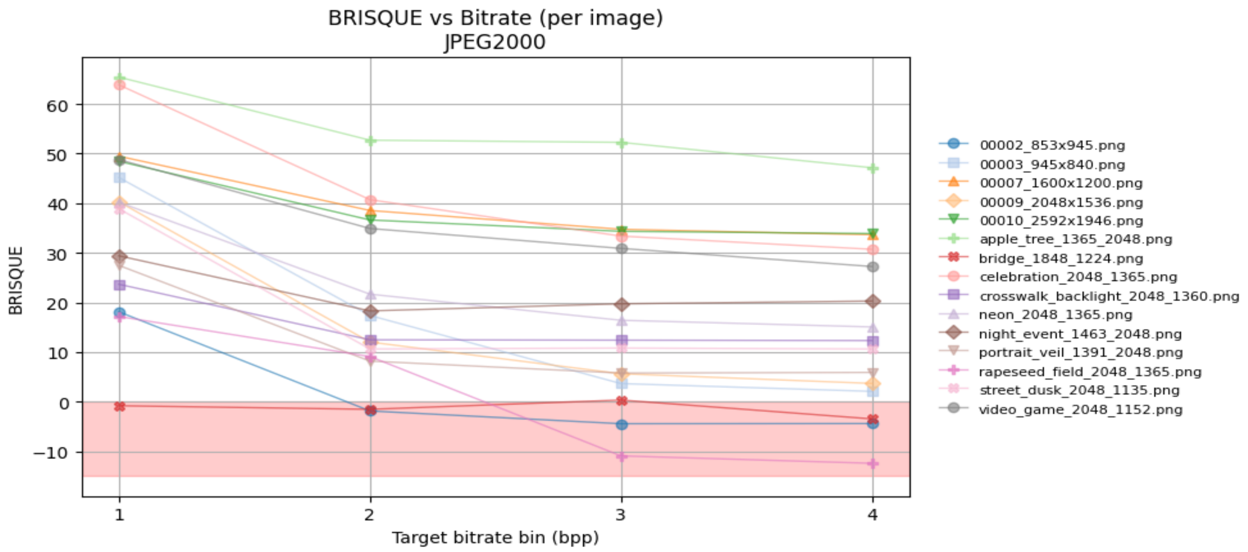


Results

BRISQUE

- Very large variability between images
- Non-monotonic behavior at high bitrates (upward trends)
- Weak correlation with PSNR



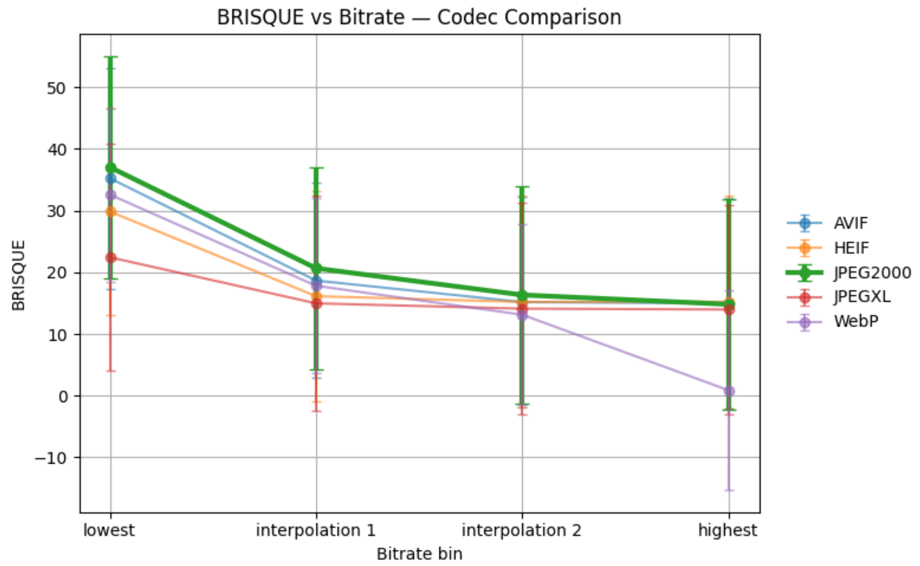


Results

BRISQUE

- Comparing all codecs over BRISQUE is imprecise (see standard deviation bars)
  - Large range of values for different images

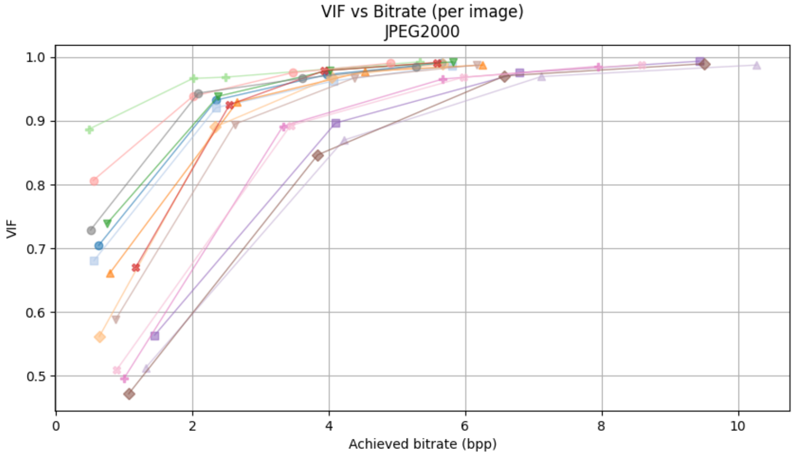
- But shows coherent trends
- From this plot:
  - JPEG 2000 tends to score lower under BRISQUE
  - JPEG XL gives the best referenceless results
    - (WebP is not considered best due to outlying values)

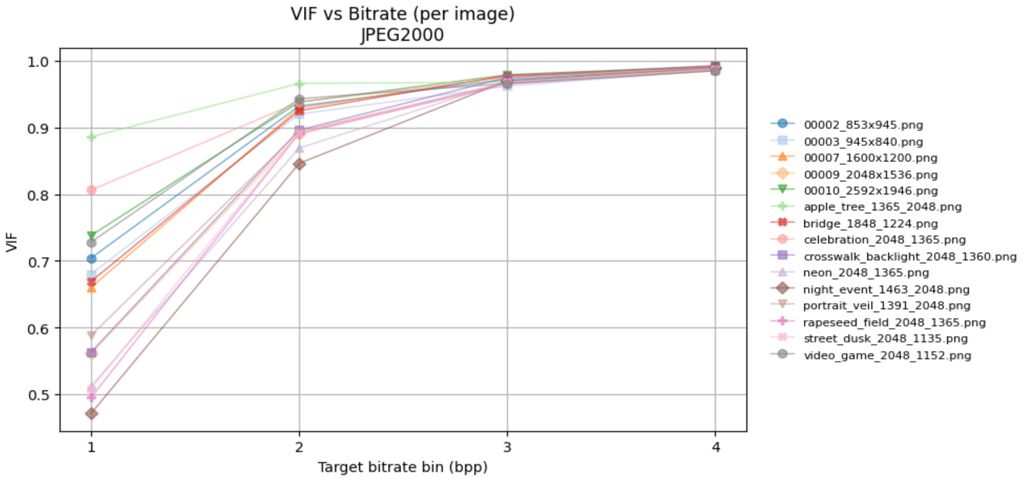


Results

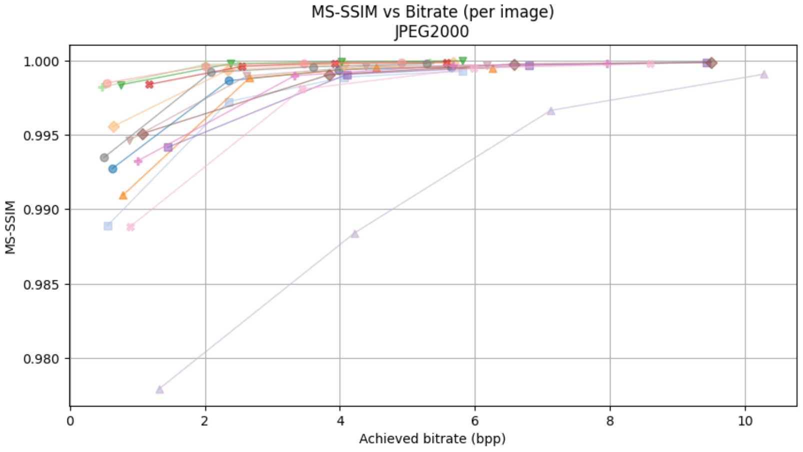
VIF & MS-SSIM

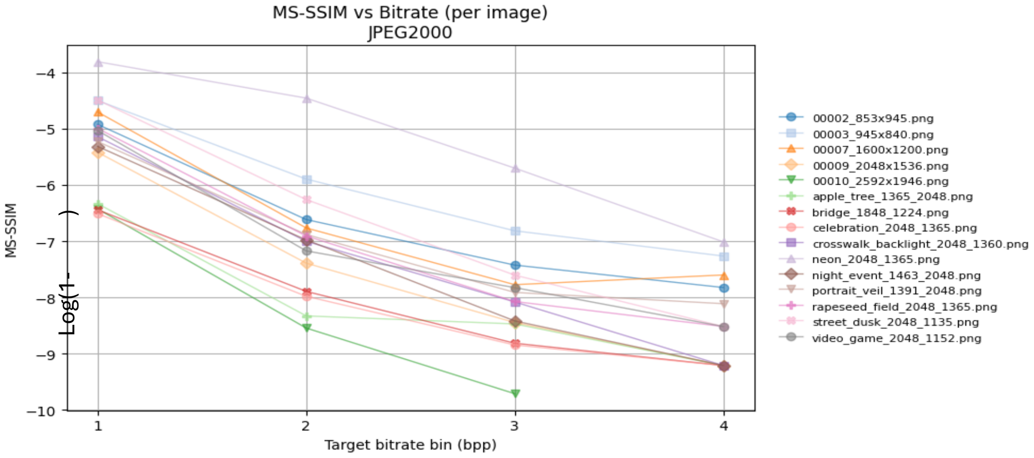
VIF





MS-SSIM



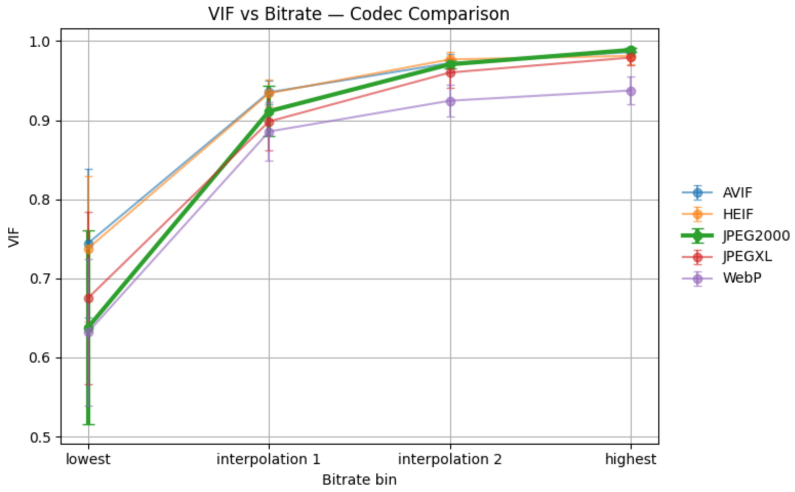


Results

VIF & MS-SSIM

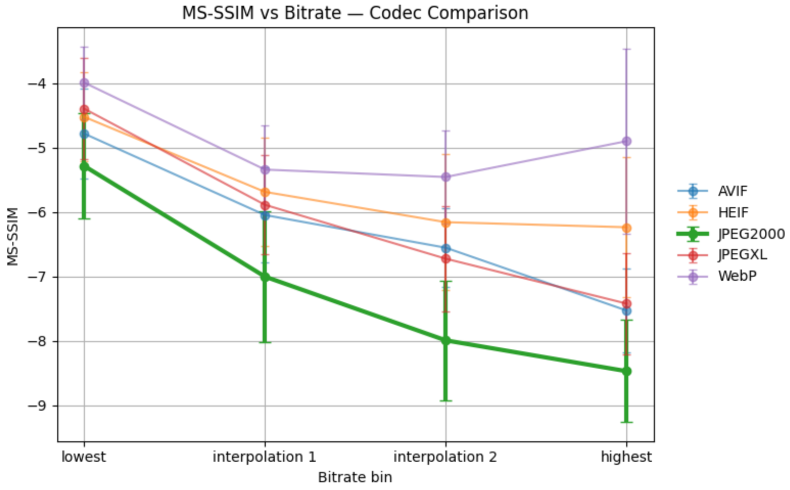
VIF

- VIF asserts the common trend of codecs
- JPEG 2000 starts lower
- At higher bitrates, JPEG 2000 reaches comparable or higher value
- Lower bitrates have higher variability



MS-SSIM

- AVIF, JPEG 2000 and JPEG XL show steady evolutions of MS-SSIM
- These three codecs show similar performance and improvements



Conclusion

- A diverse and fair dataset was built to enable reproducible codec comparison
- JPEG 2000 remains a robust and mature standard, offering:
  - High PSNR and strong performance at high bitrates
  - Scalability, progressive decoding, and industrial adoption
  - Rate control saturation highlights limitations of embedded bitstream truncation

Conclusion

- Objective metrics lead to different rankings:
  - MS-SSIM and VIF show comparable performance between modern codecs
  - BRISQUE is less reliable for compression evaluation

- This study showed the difficulty of quantifying image quality for human perception

- Overall, JPEG 2000 is still competitive, even though modern codecs are better aligned with perceptual quality objectives

Ressources

- Joint Photographic Experts Group (JPEG), *JPEG 2000 Overview*, available at: https://jpeg.org/jpeg2000/index.html
- Kakadu Software Pty Ltd., *Kakadu JPEG 2000 Codec*, available at: http://www.kakadusoftware.com
- Multimedia Signal Processing Group (MMSPG), École Polytechnique Fédérale de Lausanne (EPFL), *JPEG-AI C3 Dataset*, available at: https://www.epfl.ch/labs/mmspg/downloads/jpeg-aic3-dataset/
- Data Compression Conference (DCC), *2022 Benchmark Tasks*, available at: https://archive.compression.cc/2022/tasks/index.html
- Photosynthesis Team, *PIQ: Perceptual Image Quality Metrics*, GitHub repository, available at: https://github.com/photosynthesis-team/piq
- Ebrahimi Touradj, “Image compression Standards – I & II.”, EE-569 - Image and Video Coding, Class lectures, EPFL