

SF2 Image Processing – Final Report

Liu Zihe | zl559 | Group 8

1 Introduction

This final report builds on the previous two interim reports and develops their energy-compaction front-ends into a complete image compression scheme, which is then tested with a 5kB limit on a genuinely unseen image. We first recap the interim conclusions that drive the design, then discuss the pipeline that compares various compression schemes along with their tradeoffs, ultimately settling on the final (*winning*) design.

1.1 Interim Summary

The Laplacian pyramid (Report 1) decomposes an image into a stack of multi-scale highpass residuals plus a tiny lowpass image; it compacts energy well but stores more pixels, coding $\frac{4}{3}N^2$ pixels for an $N \times N$ image, so its best compression ratio is only CR ≈ 1.55 . Report 2 moved to transforms preserving the N^2 image size — the block DCT, the Lapped Bi-orthogonal Transform (LBT = POT + DCT), and the LeGall 5/3 Discrete Wavelet Transform (DWT) — which remove that overhead and compress better (Table 1). All ratios are quoted at matched RMS against direct pixel quantisation at step 17 (per image, RMS $\approx 4.86, 4.92, 4.94$). The rate–distortion curves (bit-RMS tradeoff) and crops are reproduced in Appendix ??.

Front-end Transform	CR at matched RMS			Characteristic artefact
	Lighthouse	Bridge	Flamingo	
Direct pixel quantisation	1.00	1.00	1.00	uniform stepped contours
Laplacian pyramid (eq-MSE, $k=3$)	1.55	1.28	1.41	blur
DCT 8×8	2.94	1.86	2.16	visible block boundaries
DWT (LeGall, 3-level, eq-MSE)	3.05	1.90	2.42	blur
LBT 8×8 ($s=\sqrt{2}$)	3.43	1.98	2.31	heavy ringing
LBT 4×4 ($s=\sqrt{2}$)	3.56	2.00	2.45	light ringing

Table 1: Interim results: matched-RMS compression ratios on all three test images. Most transforms roughly double the pyramid’s CR, and the LBT 4×4 performs the best on every image. The textured Bridge is far harder for all methods (CR barely above 2), whereas the DWT’s multi-scale analysis is also strong on Flamingo. Both LBT rows are coded in 16×16 groups (entropy via `dctbtp(·, 16)`) so the two block sizes are directly comparable.

Several results informed our overall scheme selection. *Equal-MSE quantiser step allocation* for layered schemes — quantising layer i with a step $\Delta_i \propto 1/\sqrt{g_i}$, where g_i is the energy a unit impulse in that layer contributes to the reconstruction — equalises each layer’s MSE contribution and consistently beats a single constant step in compression ratio.¹ A *centre-clipped* quantiser, which widens the zero step to $\text{rise1} = \Delta$, bringing more coefficients to zero and saved $\approx 6\text{--}7\%$ of bits on the transform front-ends, so a zero step spanning 2Δ is a sensible default. A simplified-JPEG *run-length/Huffman* coder gave lossless compression in the scanning and coding stages. Finally, the methods showed a strong image dependence: the heavily textured Bridge reached only CR ≈ 1.9 against Lighthouse’s CR ≈ 3 above, with methods varying in performance across images, which motivates the per-image scheme selection adopted.

1.2 Competition Overview

The interims minimised rate (maximize compression ratio) at a matched RMS, however the competition flips this — it fixes the rate and aims for the best quality. Each image must fit 40,960 bits (5 kB) including the header, and entries are ranked by subjective vote, RMS breaking ties only. For a 256×256 image this is just 0.625 bits/pixel, much tighter than the schemes of Table 1, placing this competition at a much coarser rate (RMS ≥ 10). Visible artefacts (blocking, ringing) now matter more than a small RMS improvement. These motivate a better perceptual metric for selection (§2.2), perceptual quantisation (§2.4), better DC-coefficient coding (§2.5.1), and decoder-side deblocking restoration (§2.6).

2 Compression Scheme and Pipeline

Every candidate scheme we build has the same three-stage pipeline (Fig. 1): an energy-compacting *transform* T , a *quantiser* Q , and a lossless *entropy coder*. Working on the centred image $\mathbf{X}$ — 128, the transform produces coefficients $\mathbf{Y}$, the quantiser maps them to integer step-numbers $\mathbf{Y}_q$, and the entropy coder serialises $\mathbf{Y}_q$ into the variable-length codeword stream (v1c) (counted against the budget). The transform and the entropy coder are exactly invertible, so these are lossless compression while the quantiser is lossy. Then $\text{encode} = \text{serialise} \circ Q \circ T$ and $\text{decode} = T^{-1} \circ Q^{-1} \circ \text{deserialise}$ are exact mirrors — any transform, quantiser or coder can be swapped independently.

After a full sweep across all possible configurations, we choose a small *registry* §2.7 of around 8 schemes we consider at inference time. During inference time, we use SSIM (in the encoder) to select between schemes, using which had the best reconstruction for this specific image. For each scheme, we also reduce the quantiser scale until it fits the 5kB budget.²

¹Here only the quantiser step *ratios* between layers matter, leaving one overall scale free — which we later sweep (§2.4.1).

²Ideally, we’d like to use neural based metrics like DISTS or LPIPS but these required `torch` and I didn’t want to overload the github workflows with the huge 5GB packages.

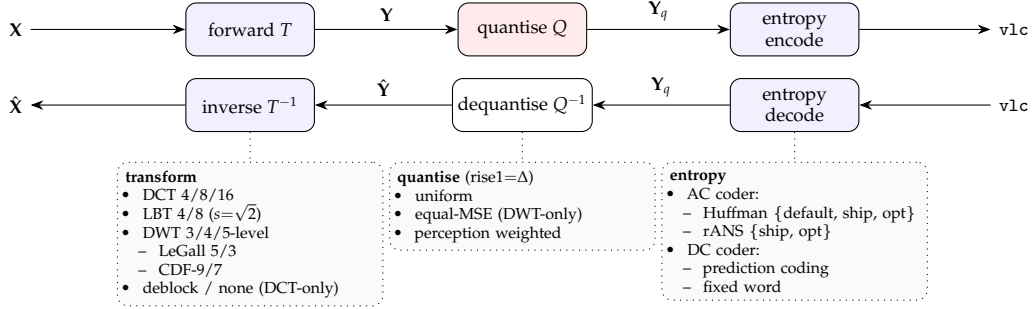


Figure 1: The compression pipeline and the design space `sweep.py` explores (§2.1). *Top:* the centred image is transformed (Y), quantised to integer step-numbers (Y_q), entropy-coded to the `v1c`. *Bottom:* running right-to-left exactly in reverse. Only the quantiser (red) is lossy. The dotted menus list every option crossed at each stage the well performing schemes are settled in §2.7. The dead-zone width $\text{rise1}=\Delta$ and the LBT scaling $s=\sqrt{2}$ are held constant, not swept.

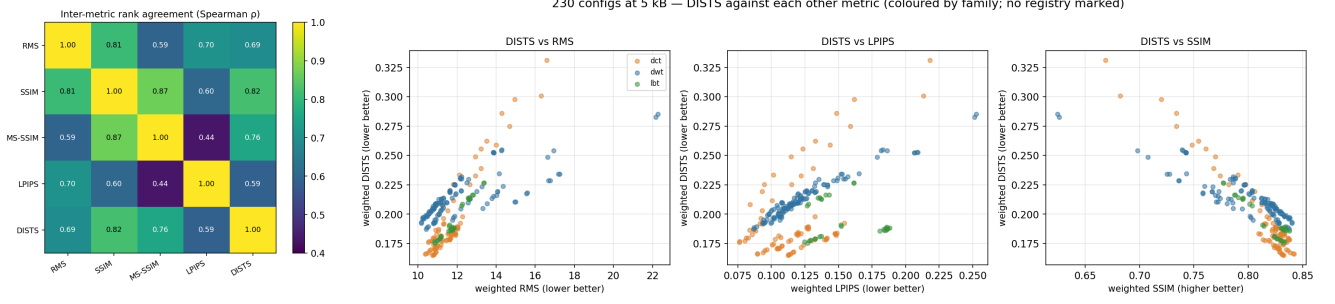


Figure 2: Agreement between five IQA metrics for evaluation on our scheme sweep. *Left:* inter-metric rank agreement (mean per-image Spearman correlation ρ); SSIM/MS-SSIM/DISTS/RMS cluster, LPIPS is the outlier (ρ to 0.44). *Right:* Agreement of performance of various schemes under each metric. DISTs vs RMS, LPIPS and SSIM each. Each point corresponds to one config, coloured by transform family. We use DISTs as the reference metric closest to human visual perception; the other metrics roughly correlate with DISTs, with LPIPS having the least agreement since the point clouds is the biggest.

2.1 Scheme Sweep

This software modularity is what makes a **massive sweep** of all possible combination of hyperparameters tractable. For every configuration of compression scheme³ we do a sweep over all three stages (§2.1). We then investigate each stage carefully, judging which options worked well with others and the best configurations overall. Since RMS is a flawed metric for visual perception quality, §2.2 sets out the perceptual metric the sweep is scored by. The training set for our sweep comprised the eight past-competition images (2018–2025) plus Lighthouse, Bridge and Flamingo — with the 2026 competition image held out.

2.2 Metrics: Image Quality Assessment (IQA)

We score every candidate scheme on their reconstruction Z of our training set images with five full-reference metrics, from pixel-fidelity to perceptual:

- **RMS** = $\text{std}(X - Z)$, **PSNR** = $20 \log_{10}(255/\text{RMS})$: simplest metric scored pixel-wise (lower better)
- **SSIM** [1]: windowed luminance $\times$ contrast $\times$ structure similarity (higher better)
- **MS-SSIM** [2] applies SSIM across an image pyramid (higher better)
- **LPIPS** [3]: distance between deep-CNN features calibrated to human judgements (lower better)
- **DISTS** [4]: deep-CNN features fusing structure and texture (lower better)

The true judge of visual reconstruction quality is by human vote, so every metric here is a proxy, but we use DISTs as the most accurate proxy of human visual perception. Fig. 2 shows SSIM, MS-SSIM, DISTs and RMS form a consensus cluster (correlation $\rho \approx 0.76\text{--}0.87$) while LPIPS is the outlier (ρ down to 0.44). During train time, when selecting the set of 8 schemes to go into our registry, we evaluate the schemes on all 5 metrics. However, during inference time, only SSIM is computationally cheap (via `scikit-image`) and does not require installing `torch`, so we use these as this in encoder to select between schemes which give the best reconstruction. The deep CNN based metrics (via `pyiqa`) are only used to evaluate the sweep configurations during development time.

³Currently, we have 230 possible configurations per image, and 11 images tested, so 2530 (configuration, image) runs swept which took about an hour.

2.3 Transforms

The front-end transforms tested comprise three families and make the biggest difference to visual reconstruction: the block **DCT** [5] at block sizes 4/8/16, the **LBT** (POT + DCT, $s=\sqrt{2}$) at block sizes 4/8, and the **DWT** at 3–5 levels. The interim reports ranked these by compression ratio at matched RMS (Table 1), however at the 5 kB budget we instead judge them on reconstruction quality with various IQA metrics and visually. All four transform front-ends are summarised side by side in Fig. 3. The block transforms (C orthonormal) apply per $N \times N$ block; the LBT inserts a POT pre-filter P_f across the $N/2$ -shifted block boundaries and a post-filter P_r that smooths them; the DWT is a separable multi-scale filter bank whose lifting form, invertibility and recursion are detailed in §2.3.1 below.

DCT [5] ($N \times N$ blocks, coded independently)

$$\mathbf{Y} = \mathbf{C} \mathbf{X} \mathbf{C}^\top, \quad \mathbf{X} = \mathbf{C}^\top \mathbf{Y} \mathbf{C}$$

$$C_{kn} = \sqrt{\frac{2}{N}} \kappa_k \cos \frac{(2n+1)k\pi}{2N}, \quad \kappa_0 = \frac{1}{\sqrt{2}}, \quad \kappa_{k>0} = 1$$

LBT (POT pre-filter $\rightarrow$ DCT, $s=\sqrt{2}$)

$$\mathbf{X}_p = \mathbf{P}_f \mathbf{X} \mathbf{P}_f^\top \quad (N/2\text{-shifted interior}), \quad \mathbf{Y} = \mathbf{C} \mathbf{X}_p \mathbf{C}^\top$$

$$\mathbf{X} = \mathbf{P}_r^\top \mathbf{C}^\top \mathbf{Y} \mathbf{C} \mathbf{P}_r$$

DWT — LeGall 5/3 (even $s_n = x_{2n}$, odd $d_n = x_{2n+1}$, in place)

$$d_n \leftarrow d_n - \frac{1}{2}(s_n + s_{n+1}), \quad s_n \leftarrow s_n + \frac{1}{4}(d_{n-1} + d_n)$$

one predict/update pair: predict yields the $\frac{1}{4}[-1, 2, -1]$ highpass detail, update the $\frac{1}{8}[-1, 2, 6, 2, -1]$ lowpass trend (5/3 taps).

DWT — CDF-9/7 (same split $s_n = x_{2n}$, $d_n = x_{2n+1}$, in place)

$$d_n \leftarrow d_n + \alpha(s_n + s_{n+1}), \quad s_n \leftarrow s_n + \beta(d_{n-1} + d_n),$$

$$d_n \leftarrow d_n + \gamma(s_n + s_{n+1}), \quad s_n \leftarrow s_n + \delta(d_{n-1} + d_n),$$

$$d_n \rightarrow d_n / K, \quad s_n \rightarrow K s_n.$$

$$(\alpha, \beta, \gamma, \delta) = (-1.586, -0.053, 0.883, 0.444), \quad K = 1.150.$$

Figure 3: Forward formulae for the four front-ends; each inverse reverses its forward map — $\mathbf{C}^\top$ for the DCT, $\mathbf{P}_r$ for the LBT, and the lifting steps run backwards with each added term subtracted. $\mathbf{C} = \text{dct_ii}(N)$ is the orthonormal DCT and $\mathbf{P}_f, \mathbf{P}_r$ the POT pre/post-filters straddling the $N/2$ -shifted block boundaries. Both biorthogonal wavelets are written as lifting on the even/odd split (one predict/update pair for 5/3, two pairs for 9/7). §2.3.1 covers how they are applied separately and recursed on the LL band.

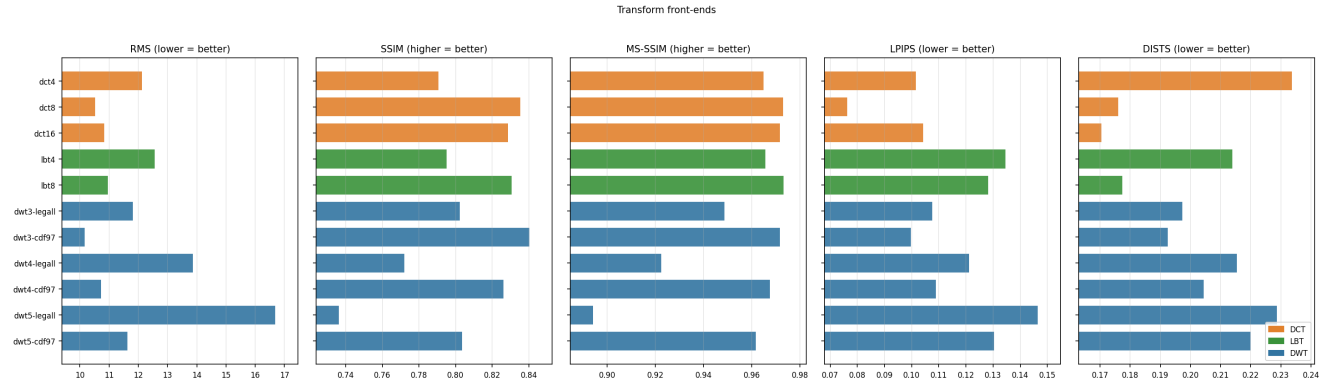


Figure 4: Each front-end’s weighted score per metric at the 5kB budget (controlled settings: uniform quantiser, predicted DC, Huffman shipped AC coder), results averaged across all images. No front-end wins every metric: DWT3-CDF 9/7 wins RMS/SSIM, the block DCT (8/16) wins the perceptual LPIPS/DISTS, while LBT performs in the middle.

Different IQA metrics prefer different transforms. Fig. 4 scores each front-end on all five metrics at the budget rate: DWT-CDF 9/7 wins on RMS and SSIM, but DCT (both 8 and 16) leads on the perceptual CNN-based LPIPS and DISTS. Aggregating across to each transform family (Fig. 5) shows this clearly: RMS picks DWT on 8 of 11 images, yet LPIPS and DISTS pick DCT on 10 of 11. The family-coloured DISTS scatter of Fig. 2 shows the same geometry: DCT configs cluster at low DISTS, DWT configs at low RMS.

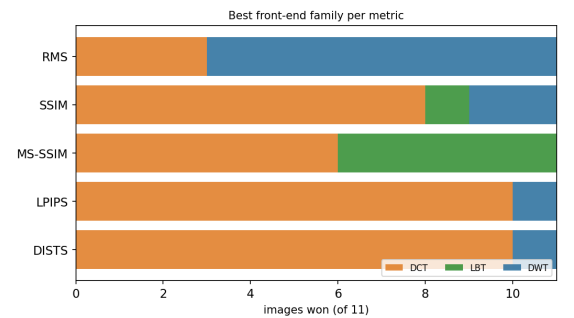


Figure 5: Best front-end *family* per metric over the 11 images: RMS favours DWT (the interim story), the perceptual metrics and the human vote ($\rightarrow \text{dct8-db}$) favour DCT.

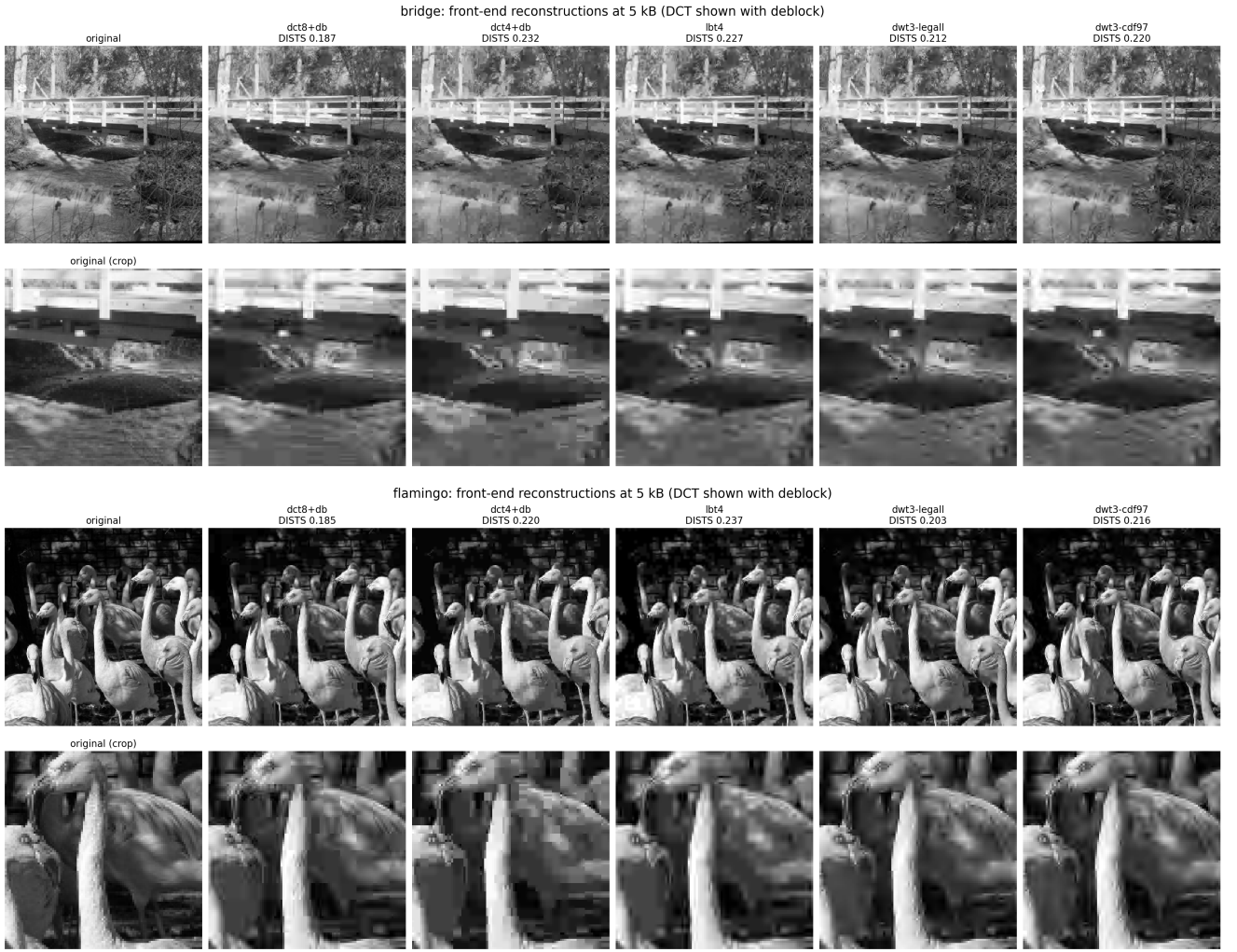


Figure 6: Front-end reconstructions at 5 kB on Bridge and Flamingo (full frame top, zoomed crop bottom; DISTs under each panel). Look for the DCT’s crisp edges, the LBT’s slight wash-out, and the DWT wavelets’ soft ringing — clearest on Flamingo’s smooth gradients, subtle on Bridge’s texture.

Visually (Fig. 6), at 5 kB the DCT gives crisp edges, but blocking effects are still visible despite deblocking turned on. The LBT gives extremely blurry effects and clearly performs the worst here. Both DWT wavelets have slight blur in certain regions, visible on the Flamingo feathers and waves in the Bridge image. However, looking at the Flamingo eyes we can clearly see DWT-CDF 9/7 gives the clearest details on the eyes and faces. This motivates leaning on DCT and DWT in our scheme registry, keeping the LBT only as a low-priority fallback (§2.7).

2.3.1 Wavelet Transforms

For wavelet transforms, a 1-D transform filters the signal with a lowpass analysis filter H_0 (overall trend) and a highpass H_1 (detail), then decimates each output 2:1 so the total sample count is preserved. Applied separably — filtering rows then columns — one 2-D transform splits an image into four quarter-size sub-bands (Fig. 7): a coarse LL (lowpass, lowpass) thumbnail holding almost all the energy, the HL/LH vertical/horizontal edges, and the HH diagonal detail. The transform then *recurses on the LL band alone*, building a pyramid of detail bands above one tiny LL.

Algorithm 1 n -level 2-D DWT (forward)

Require: centred X ($m \times m$), levels n , 1-level DWT₁

```

1: for  $\ell \leftarrow 1$  to  $n$  do
2:    $X[1:m, 1:m] \leftarrow \text{DWT}_1(X[1:m, 1:m])$     ▷ Apply predict and update filters
3:    $m \leftarrow m/2$                                 ▷ recurse on LL
4: end for
5: return  $X$ 

```

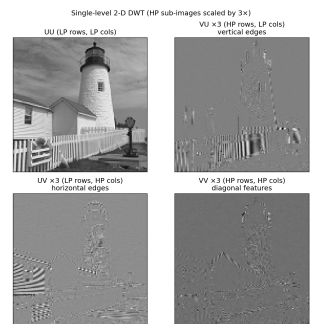


Figure 7: Separable 2-D LeGall 5/3 level on Lighthouse. Four sub-bands — LL approximation (top-left), HL/LH vertical/horizontal edges, HH diagonal

We experiment with 3–5 levels, and the inverse runs the inverse filter bank from the smallest LL outward. We use two biorthogonal, symmetric wavelets: **LeGall 5/3** [6] (JPEG2000’s reversible filter; “bior-2.2”) used in interim reports; **CDF-9/7** [7, 8] (JPEG2000’s lossy default; “bior-4.4”) has longer, smoother filters. Both wavelets use the *lifting scheme* [9] of Fig. 3: a cascade on the even/odd split of a *predict* step (each odd sample from its even neighbours) and an *update* step (the evens corrected to preserve the local mean) — one pair for LeGall 5/3, two pairs for CDF-9/7. For LeGall, eliminating s, d gives the $\frac{1}{8}[-1, 2, 6, 2, -1]$

lowpass exactly, with $\frac{1}{4}[-1, 2, -1]$ as the detail highpass filter.

Energy compaction. The fraction of energy⁴ held by the largest few coefficients measures how well transforms compact energy in its coefficients. CDF-9/7 concentrates energy best, while LeGall 5/3’s shorter filters leave far more in the highpass bands (Table 2). This also explains *why CDF 9/7 wins on RMS* (Fig 4) — it compacts energy best — the largest 1% of its coefficients hold 77–84% of the image energy, against 75–80% for the DCT and only 46–62% for LeGall 5/3. This means after quantisation where many small coefficients are zeroed, the CDF 9/7 would likely preserve RMS (root of energy) the best.

2.4 Quantisers

The quantiser is the only lossy stage. At a quantiser step $s = \Delta \cdot W$ — where we adjust the scalar rate Δ (§2.4.1) until the 5kB budget fits, and a weight map W decides the quantisation scheme. The coefficients $\mathbf{Y}$ are quantised to $\mathbf{Y}_q$ and dequantised to $\hat{\mathbf{Y}}$.

$$\mathbf{Y}_q = \text{sign}(\mathbf{Y}) \cdot \lceil |\mathbf{Y}|/s - 1 \rceil_+, \quad \hat{\mathbf{Y}} = \text{sign}(\mathbf{Y}_q) \cdot s \cdot (|\mathbf{Y}_q| + \tfrac{1}{2}), \quad \lceil x \rceil_+ \equiv \max(0, \lceil x \rceil).$$

The -1 fixes the first rise at $\text{rise1} = 1.0 \cdot s$, a hence the centre-clipped quantiser of width $2s$ as in notebook §11). This rounds many more coefficients to zero, cutting $\approx 6\text{--}7\%$ of bits on the smooth Lighthouse and $\approx 4\text{--}5\%$ on Flamingo, but only $\approx 1\%$ on the textured Bridge (few near-zero coefficients).

The weight map W distinguishes the three variants:

- **uniform** ($W=1$). For the DCT/LBT, their orthonormal basis means a uniform step is already MSE-optimal
- **equal-MSE** ($W=1/\sqrt{g_i}$ per sub-band) for the DWT, so the per layer $\Delta_i \propto 1/\sqrt{g_i}$ as in previous reports
- **perceptual** (W = the standard JPEG luminance quantisation table [5].) Coefficient (i, j) is quantised with step w_{ij}/w_{00} , the weights w_{ij} rising towards high-frequencies to track the human *contrast-sensitivity function* (CSF), which falls off at high spatial frequency. Then detail that the eye barely resolves is quantised coarsely.

Fig. 8 shows how each quantiser performs per metric (note we only apply perceptual quantisation for DCT and LBT, and equal-MSE for DWT only). Equal-MSE beats uniform on all five metrics, so it is the default for DWT. For perceptual weighting, against uniform it raises SSIM and MS-SSIM but lowers RMS, LPIPS and DISTs. Visually (Fig. 9), perceptual weighting loses some details on the Flamingo eye. We include both perceptual and uniform variants in our scheme registry, since on certain images like competition2023 the difference is not noticeable.

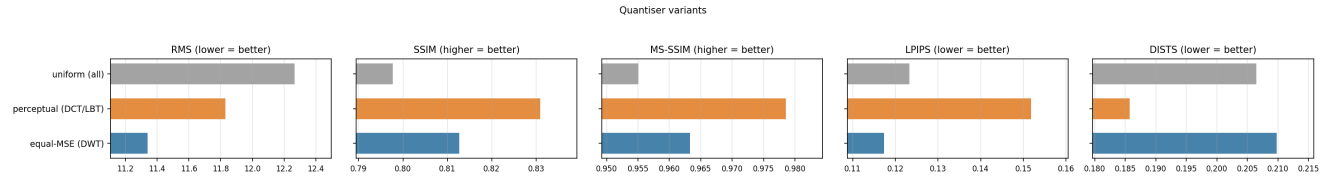


Figure 8: Quantiser variants as the absolute weighted metric over the sweep, each averaged on the front-ends it supports (uniform = all transforms; perceptual = DCT/LBT-8; equal-MSE = DWT). For DWT equal-MSE beats uniform on all five metrics, whereas perceptual (on DCT/LBT) raises SSIM/MS-SSIM but lowers RMS/LPIPS/DISTs.

D — perceptual quant on/off: it raises SSIM/MS-SSIM but lowers LPIPS/DISTs — does the CSF-weighted table actually look better?



Figure 9: Uniform (dct8-db) vs perceptual (dct8-db-perc) reconstructions at 5 kB for DCT 8x8 — zoomed crops on Flamingo and comp2023. Areas around the Flamingo eye are noticeably blurrier with perceptual weighting, but we include both variants in the scheme registry.

⁴The *energy* of coefficients is the squared Frobenius norm $E = \|\mathbf{Y}\|_F^2 = \sum_{ij} Y_{ij}^2$ (sum of squared magnitudes), and similarly for image energy $\|\mathbf{X}\|_F$. The orthonormal DCT preserves image energy exactly, $\|\mathbf{Y}\|_F = \|\mathbf{X}\|_F$ by Parseval (proven in Report 2).

2.4.1 Quantiser Step and Rate Control

Each registered scheme exposes a single free parameter, the overall quantiser scale Δ , drawn from a geometric grid spanning $[1, 256]$ over $2^{\text{scale_bits}}$ indices where we use $\text{scale_bits}=10$. To meet the budget the encoder bisects (searches over the range of) Δ until the variable length code and header bits is $\leq 40,960$. Since this total bit size is monotone in Δ , so searching over Δ is equivalent to searching for the closest total bit size too. The header is a fixed *46 bits* — $(\text{scheme_id}, \text{scale_idx}, h, w) = 4 + 10 + 16 + 16$ — carrying only the scheme id and scale index. Every other per-scheme setting (block size, filter, step ratios, Huffman table) lives in offline code and costs nothing to transmit, as our scheme_id indexes into our scheme registry which contains the required settings.

2.5 Entropy Coders

The quantised $\mathbf{Y}_q$ is serialised losslessly into the v1c , where we serialise each block of $M \times M$ quantised coefficients. Each $M \times M$ block contributes two independent streams: the single **DC** coefficient (the block mean, strongly correlated block-to-block) and the $M^2 - 1$ **AC** coefficients, read in a zig-zag scan that gathers the near-zero high frequencies into a long tail. We generalised the lab's 8×8 -only `jpegenc` to any block geometry — directly for the DCT, a per-block regroup into 16×16 groups for the LBT, and a global `dwtgroup` for the DWT — so one coder serves every front-end transform.

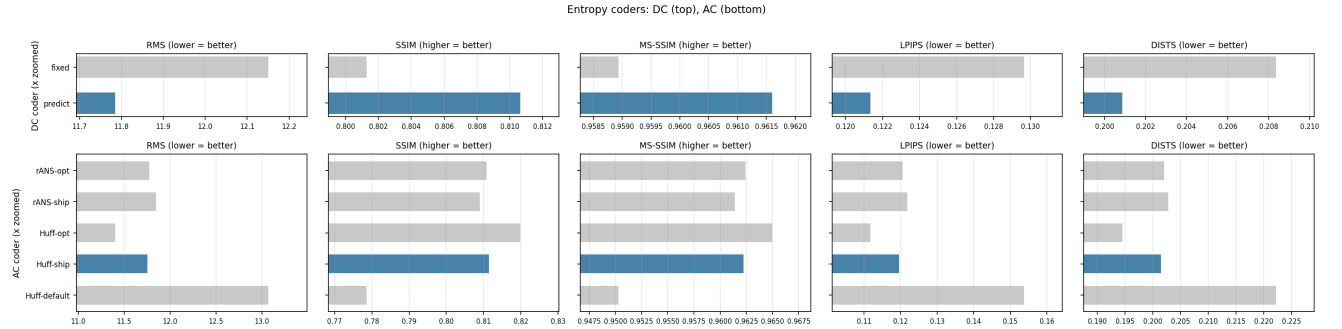


Figure 10: Entropy-coder choices, as the score per metric over the sweep. Chosen choice is blue. *Top:* the DC coder — predicted residual (§2.5.1) beats the fixed word on every metric. *Bottom:* the AC coder — all five coders (default/shipped/opt Huffman, shipped/opt rANS) perform similarly.

2.5.1 DC Coder: Fixed word vs Prediction Coding

Each $M \times M$ block contains one **DC** coefficient — the block mean (DCT/LBT) or LL value (DWT) — which the quantiser maps to a signed integer step-index q . The naive *fixed* option encodes every q in a fixed b -bit word,⁵ which spends b bits on a highly predictable quantity.⁶ We instead follow the LOCO-I/JPEG-LS coder [10] and transmit only a prediction residual. The one-DC-per-block values form a coarse low-resolution image (a $\frac{H}{M} \times \frac{W}{M}$ grid), which is smooth (and so each DC coefficient is well predicted from its neighbours). Coding the blocks in raster order, we first predict the current block's DC from the three already-decoded neighbouring blocks in that grid: d_W (the block to the left, West), d_N (the block above, North) and d_{NW} (above-left, North-West). The predictor combines these using the *median edge detector* (MED):

$$\hat{d} = \text{median}(d_W, d_N, d_W + d_N - d_{NW}) = \begin{cases} \min(d_W, d_N), & d_{NW} \geq \max(d_W, d_N) \\ \max(d_W, d_N), & d_{NW} \leq \min(d_W, d_N) \\ d_W + d_N - d_{NW}, & \text{otherwise (planar gradient),} \end{cases}$$

We then only encode the residuals $e = d - \hat{d}$, which are sharply peaked at zero (a two-sided geometric, i.e. discrete Laplacian [10]), so we code it with an adaptive *Golomb-Rice* code [11]. This uses far fewer bits than the fixed b -bit word to transmit the DC coefficients.

Prediction is strictly better for a given budget: freeing the DC bits allows us a finer overall step and improves all five metrics (Fig. 10, top). Visually, there's almost no difference — reconstructions in Fig. ?? . Algorithms 2 (fixed) and 3 (predicted) give the two coders.

Algorithm 2 Fixed DC Coder.

- 1: **encode:** transmit $q + 2^{b-1}$ in b bits ▷ to unsigned int
- 2: **decode:** $q \leftarrow (\text{read } b) - 2^{b-1}$ ▷ back to signed int

Algorithm 3 Predicted DC (LOCO-I/JPEG-LS).

- 1: $d_W, d_N, d_{NW} \leftarrow G$ ▷ W/N/NW blocks
- 2: $\hat{d} \leftarrow \text{median}(d_W, d_N, d_W + d_N - d_{NW})$; $e \leftarrow q - \hat{d}$
- 3: $u \leftarrow 2e$ if $e \geq 0$ else $2|e| - 1$ ▷ fold
- 4: $k \leftarrow \min\{k : n 2^k \geq A\}$ ▷ adaptive Rice
- 5: transmit $(u \gg k)$ zeros, a 1, then low k bits of u
- 6: $G[i, j] \leftarrow q$; $A += u$; $n += 1$; halve if $n \geq 64$

⁵The signed q is offset by 2^{b-1} into a non-negative b -bit integer and transmitted raw.

⁶This width is also only *provably* sufficient for the plain DCT — the LBT's POT pre-filter and the DWT's LL band can exceed this.

2.5.2 AC Coder: Huffman vs rANS

The zig-zag AC stream is sparse and sharply peaked at zero, so it is run-length coded (Alg. 4). First, RUNAMPL turns the stream into (run of zeros, bit size of coefficient, value of coefficient) events. Then we hash each (run, size) pair to one of 162 symbols (including end-of-block EOB and a pure 16-zero run ZRL). Then a Huffman code maps each of the 162 symbols to a binary encoding, and then the value’s binary encoding is concatenated raw [5]. We compared three Huffman tables — the standard JPEG table, an optimized per-image table transmitted online at +1424 bits (huff-opt), and an offline-trained table shipped in code (trained on all competition images except 2026) which uses zero header (huff-ship). We also compare Huffman encoding against a rANS coder [12]. rANS reaches the entropy with *fractional* bits per symbol: each step multiplies the state x by $\approx 1/p_s$, costing $-\log_2 p_s$ bits, so it breaks Huffman’s one-bit-per-symbol floor exactly when a symbol has $p > 0.5$.

However at 5 kB budget, the AC coders perform very similarly (Fig. 10, bottom). Run-length encoding has already absorbed the dominant $>50\%$ “zero” events into the run codes, so an ideal Huffman code already comes very close to the entropy floor and rANS does not improve by much. huff-ship ties or beats rans-ship on RMS in most scenarios.

Online optimization vs Offline training. The per-image huff-opt table trained online costs 1424 bits to transmit, and that overhead outweighs its coding gain — so at matched quality it uses *more* bits than our zero-header offline table on all 11 images (Fig. 11). Surprisingly, it also generalises quite well. To test the trained table huff-ship fairly we hold each test image out, re-train on the rest, and encode the held-out image as a proxy for the unseen 2026 one.

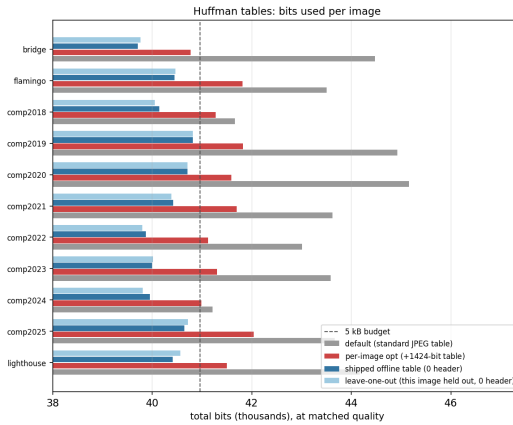


Figure 11: Total bits to code each image under four Huffman tables: default (standard JPEG, grey), per-image huff-opt (its +1424-bit table counted, red), our shipped offline table (zero header, blue), and a leave-one-out table trained with *this* image held out (light blue). Every bar codes the *same* quantised coefficients at the *same* quantiser step (the shipped table’s budget-fit step), so reconstructions are byte-identical. The dashed line is the 40,960-bit budget. Look for the shipped table being the shortest bar on every image — *below* huff-opt and well under the default — so huff-opt’s transmitted table never repays itself. At this step the default table busts the budget on most images while the shipped table stays under it: the zero-header offline table is what lets a finer step survive the cap. Finally, the leave-one-out bar (which never saw the image it codes, a stand-in for the unseen 2026 image) performs very close to the shipped bar — the offline table generalises.

We therefore ship the zero-header offline Huffman table trained on all previous competition images, and do not use rANS coding.

2.6 Decoder-side Restoration (Deblocking)

Because the DCT quantises each 8×8 block independently, at the coarse 5 kB rate adjacent reconstructed blocks no longer agree across their shared boundary, leaving the faint 8-pixel grid of block effects. We suppress it with a decoder-side post-filter applied after the inverse DCT, which costs zero bits (deterministic, needs no header beyond the scheme id) — the cheap, post-loop cousin of the in-loop deblocking filters of H.264/HEVC [13] and AV1’s CDEF [14].

The filter is *edge-gated* (Alg. 5): at each block boundary it measures the step d across the two straddling pixels and pulls them together by $\frac{\alpha}{2}d$ each *only if* $|d| < \tau$ — small enough to be an artefact rather than a real edge. With $\alpha=0.5$ this halves the boundary step. A genuine edge ($|d| \geq \tau=24$ grey levels) is left untouched. Gating on τ allows deblocking to only erases discontinuities the transform should not have produced, not real detail.

Algorithm 4 AC run-length + Huffman, one block

```

1: HUFF ← Huffman table           ▷ 16 symbol ↦ binary codeword
2:  $a \leftarrow$  AC coefficients in zig-zag order
3:  $(r, s, v) \leftarrow \text{RUNAMPL}(a)$    ▷  $r$  consecutive zeros,  $|v|$  in  $s$  bits
4: for each event  $(r, s, v)$  do
5:   while  $r > 15$  do
6:     transmit HUFF(ZRL)
7:      $r \leftarrow r - 16$ 
8:   end while
9:   transmit HUFF( $16r + s$ )
10:  if  $s > 0$  then
11:    transmit  $v$  as  $s$  raw bits
12:  end if
13: end for
14: transmit HUFF(EOB)

```

Algorithm 5 Edge-gated deblock (decoder-side)

```

Require:  $Z$ , block size  $N=8$ ,  $\tau=24$ ,  $\alpha=0.5$ 
1: for each boundary  $c \in \{N, 2N, \dots\}$  (cols, then rows) do
2:    $d \leftarrow Z_c - Z_{c-1}$            ▷ cross-boundary step
3:   if  $|d| < \tau$  then               ▷ artefact, not a real edge
4:      $Z_{c-1} \leftarrow Z_{c-1} + \frac{\alpha}{2}d$ ;  $Z_c \leftarrow Z_c - \frac{\alpha}{2}d$ 
5:   end if
6: end for
7: clip  $Z$  to  $[0, 255]$ 

```

The gain is small but remarkably consistent (Fig. 13). Averaged over the sweep it improves four of the five metrics. Visually (Fig. 12) it softens the block grid in smooth regions (sky, water) while leaving edges and texture crisp (Appendix Fig. 18); at 5 kB it *reduces* rather than removes blocking, consistent with only halving each step. Since the benefit is image-dependent, the registry keeps the un-deblocked DCT variant as a separate candidate.

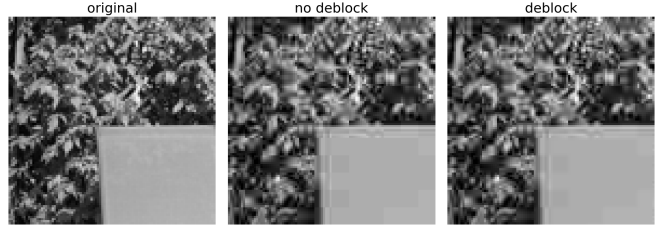


Figure 12: Deblocking on comp2020 (the “bush chair”) at the 5 kB budget — original, no-deblock and deblock crops. Look for the 8-pixel block grid that quantisation leaves in the smooth grey panel (lower-right), softened once deblocking is applied, while the surrounding foliage edges stay sharp (SSIM 0.788 $\rightarrow$ 0.793, DISTS 0.196 $\rightarrow$ 0.185).

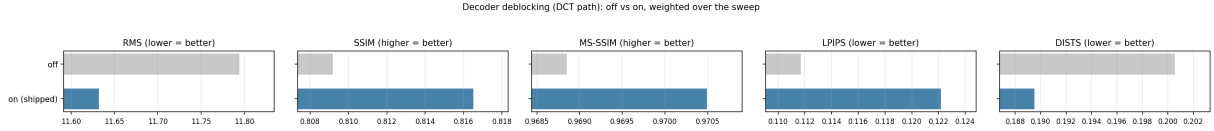


Figure 13: Decoder deblocking with DCT: weighted metric with deblock *off* (grey) vs *on* (blue), averaged over the sweep at the 5 kB budget. Deblocking helps with almost every metric: RMS, SSIM, MS-SSIM and DISTS and losing only on LPIPS.

2.7 Registry and Scheme Selection

We therefore curate a registry of eight schemes that all sit on or near the SSIM–DISTS Pareto front (Fig. 14) and let SSIM choose only among these, at inference (runtime). The eight (all using the shipped Huffman table and predicted DC, §2.5) span the front-ends worth keeping:

- dct8-db and its perceptual twin dct8-db-perc — both with the deblocked 8×8 DCT
- dwt3- $\{\text{legall}, \text{cdf97}\}$ under $\{\text{equal-MSE}, \text{uniform}\}$ quantisation — four wavelets transforms using different quantisation schemes
- 1bt4, 1bt8-perc — LBT retained as a fallback, though it rarely wins at the budget rate

The uniform/perceptual pairs are both kept because their contest is image-dependent (§2.4).

At runtime the encoder runs all eight schemes, searching over the budget-fitting quantisation step, decodes each, and keeps the highest-SSIM reconstruction. The chosen scheme is encoded in the header. Even though SSIM is a weak selector compared to human visual perception or even DISTS, it is still more robust than RMS in choosing a more faithful reconstruction.

3 Final Design and Assessment

3.1 Final Assessment

Table 3 runs the complete pipeline on the three required images. Every image fits the 40,960-bit budget with the 46-bit header included, and the reconstructions (Fig. 19, App. ??) are visually faithful at the budget rate.

Image	Selected scheme	step Δ	v1c bits	total (+46)	RMS	SSIM
Bridge	dct8-db	33.4	40825	40871	11.61	0.787
Flamingo	dct8-db-perc	29.8	40777	40823	11.93	0.854
2026 (unseen)	dct8-db	37.2	40776	40822	11.39	0.841

Table 3: Final scheme on the required set at the 5 kB budget. Each row is the scheme the SSIM selector chose, the quantiser scale Δ the budget bisection landed on, and the resulting bits/quality.

Competition outcome. Of the 13 entries, we *won the subjective vote*, but only placed 4th on 2026 by RMS. The whole field clustered at RMS $\approx$ 11–15 (fairly coarse compression), so the contest turned on artefacts the eye sees, not comparing raw RMS. This validates designing for perceived quality: the deblocked DCT looks cleanest at 5 kB even though wavelet transforms beat it on RMS.

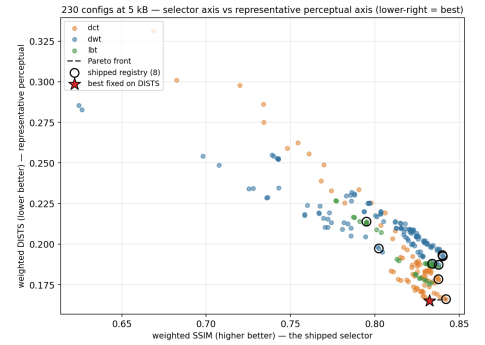


Figure 14: Config quality frontier over the 230-config sweep — weighted SSIM (x , higher better, the shipped selector) vs weighted DISTS (y , lower better, the perceptual reference), one point per config coloured by transform family. The eight circled points are the shipped registry, the star the single best-fixed scheme on DISTS. Look for the registry hugging the lower-right (high-SSIM, low-DISTS) front, so SSIM-selection roughly correlates with DISTS.

3.2 Selection and reconstruction

At encode time the pipeline budget-fits all eight registry schemes to 2026 and keeps the highest-SSIM reconstruction (Table 4). `dct8-db` wins — and uniquely on this image the three families of metric agree: SSIM, DISTs and RMS would all pick it, so the selection is unambiguous.

The 2026 image is heavily textured — measuring its high-frequency energy fraction with the top 1% of DCT coefficients gives (0.099), and is essentially Bridge’s (0.096), double Flamingo’s (0.044). At this budget rate, with a human judge, the crisp-edged deblocked DCT still wins on this detailed image: the DWT’s blur across the image costs more subjectively than the DCT’s residual (deblocked) blocking.

The decoded image and its signed error map (Fig. 15) show the error exists mostly on edges and the heavily textured metal grid — smooth regions are nearly exact.

scheme	RMS	SSIM	DISTS	
<code>dct8-db</code>	11.39	0.841	0.156	←
<code>dct8-db-perc</code>	12.45	0.826	0.166	
<code>lbt8-perc</code>	12.37	0.825	0.178	
<code>dwt3c-uni</code>	11.78	0.824	0.179	
<code>dwt3c-emse</code>	11.77	0.823	0.178	
<code>dwt31-emse</code>	12.09	0.822	0.179	
<code>lbt4</code>	13.32	0.795	0.187	
<code>dwt31-uni</code>	13.72	0.783	0.188	

Table 4: All eight registry schemes budget-fit to the 2026 image (each $\leq 40,960$ bits). Rows sorted by SSIM (the deployed selector); ← marks its pick. On this image SSIM, DISTs and RMS all rank `dct8-db` first — a rare full agreement.

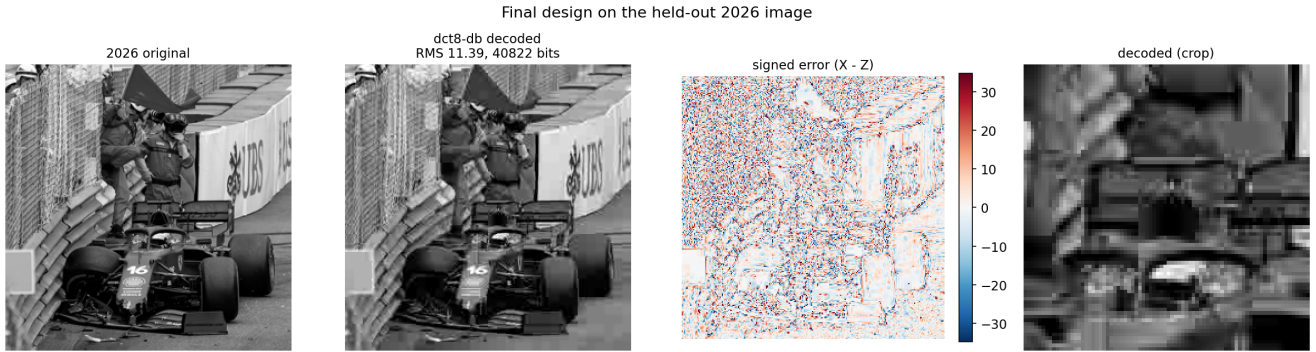


Figure 15: The final `dct8-db` codec on the held-out 2026 image at 40,822 bits (RMS 11.39): original, decoded, signed error $X - Z$ (red/blue, clipped to the 99th percentile), and a decoded crop. Look for the error concentrating on high-contrast edges — the car bodywork, the driver, the track-wall lettering — while the smooth tarmac and background are reconstructed almost exactly; no 8-pixel block grid survives in the flat areas, the deblocking filter having absorbed it.

Comparing the signed error across all eight candidates (Fig. 21, App. ??) makes comparison of texture of errors clear. The DCT variants have hard errors on the edges and leave the flat regions the cleanest of any scheme. The CDF-9/7 wavelet instead smears heavy ringing all along the contours, bleeding error into the smoother zones either side of each edge, and the shorter LeGall 5/3 filters scatter still more residual errors across the whole frame (worst in `dwt31-uni`). The LBT-4 is the most saturated everywhere, its tiny blocks leaving the largest residual. The deblocked DCT without perceptual quantisation has the least error on smallest areas. In other groups images ??, we can see others used DWT and LBT variants which led to blurrier images overall.

4 Conclusion

The three reports trace one design: from the Laplacian pyramid, through to the transforms that roughly doubled the compression ratio (Report 2), to a complete compression pipeline scheme assessed at the competition’s 5 kB budget. The competition flips the goal from minimising rate to maximising *perceived* quality, and our per-image-selected eight-scheme codec won the subjective vote on the unseen 2026 image.

The recurring lesson is that schemes each have their own tradeoffs: at matched RMS the LBT and DWT win, but at the coarse 5 kB rate judged by eye the simplest scheme — a deblocked 8×8 DCT — wins, because quality turns on *where* the error falls, not its RMS. So the cheap zero-bit levers (centre-clipping, equal-MSE/perceptual quantisation, DC prediction, deblocking, per-image selection) paid off while the elaborate rANS coder and per-image tables did not, and selecting on a perceptual proxy did well.

Group contribution. Zach (Zihe) lead the overall pipeline design. Both Zach and Lucas discussed and explored options to try, but Zach focused more on transforms and quantisations, while Lucas on entropy coding.

A Appendix

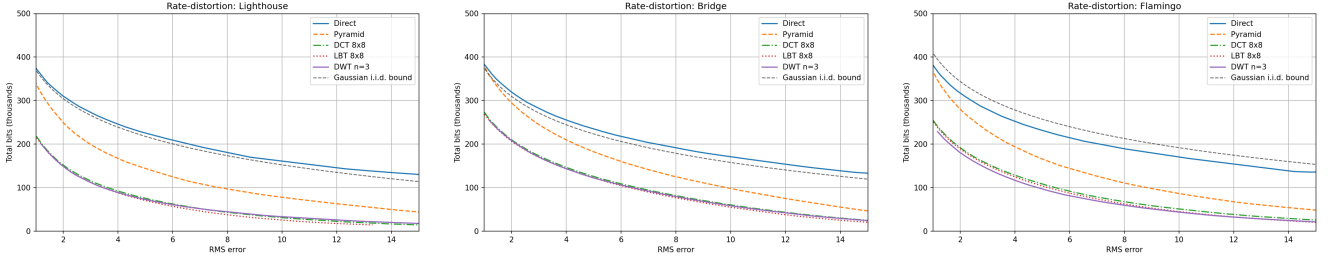


Figure 16: Cross-method rate-distortion from Report 2: total bits versus reconstruction RMS on Lighthouse, Bridge and Flamingo (left to right); lower-left is better. See that (i) the three critically-sampled transforms (DCT, LBT, DWT) collapse onto one curve far below the Laplacian pyramid and direct quantisation at every RMS, confirming the Table 1 ordering, and (ii) every curve lifts as the image gets harder — the fine texture in Bridge forces far more bits for the same RMS than the smoother Lighthouse or Flamingo.

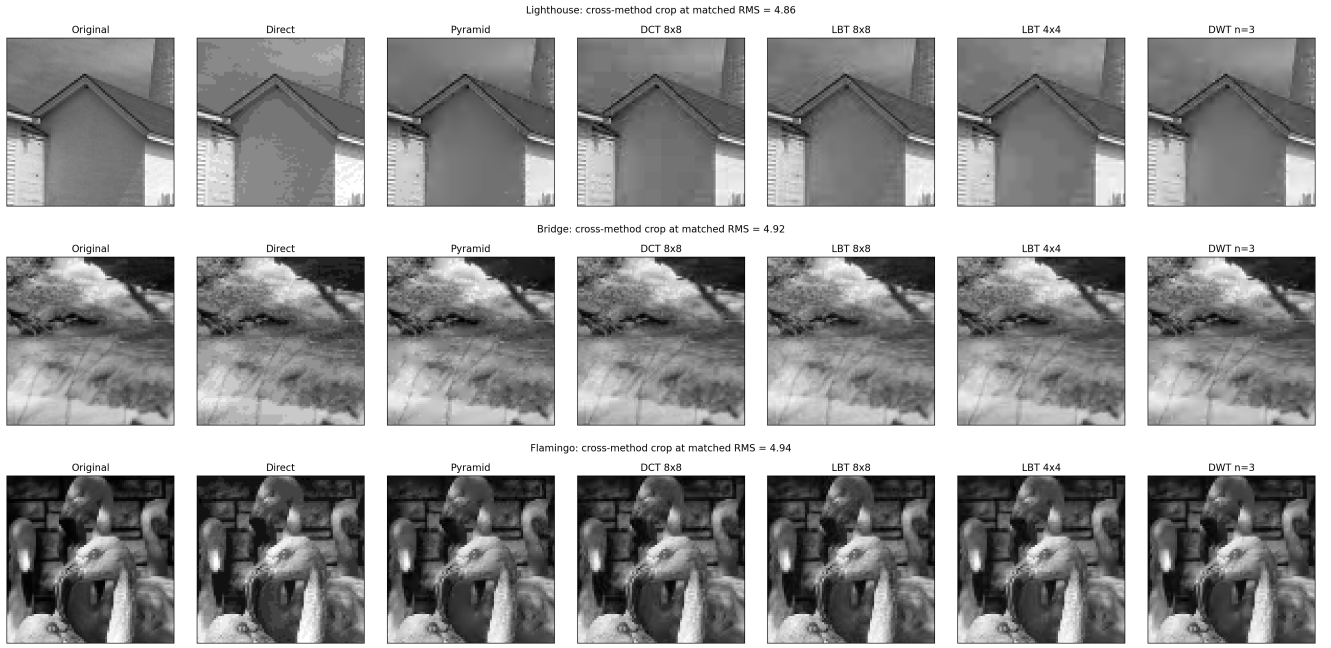


Figure 17: Matched-RMS reconstructions of one crop per test image, each row at its own matched RMS ($\approx 4.86, 4.92, 4.94$); per-panel CR is shown above each reconstruction. Within a row the distortion is fixed, so the panels differ only in how the error appears. Look for the DCT's block boundaries (clearest in Lighthouse's smooth sky), the LBT's removal of these but extra ringing introduced, the DWT's mild ringing/blur on edges.

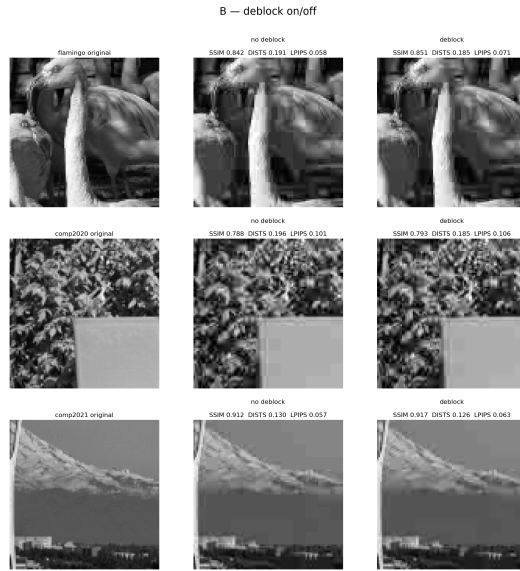


Figure 18: Decoder deblocking on/off at the 5 kB budget (§2.6): zoomed crops on three smooth-region images (original / no-deblock / deblock; SSIM/DISTS/LPIPS under each panel). Look for the faint 8-pixel block grid in the flat regions — the out-of-focus background behind the flamingo, the smooth grey panel in comp2020, the hazy sky in comp2021 — softened in the right column, while genuine edges and texture stay sharp. The change is deliberately gentle ($\alpha=0.5$ halves each boundary step rather than erasing it), so blocking is reduced, not eliminated; SSIM rises on all three.

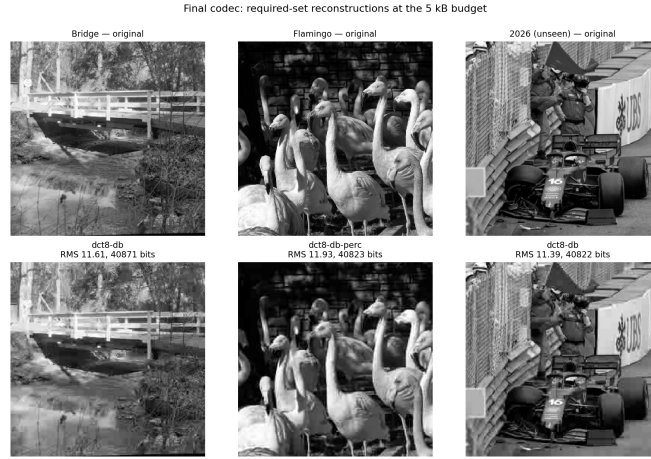


Figure 19: Required-set reconstructions at the 5 kB budget — each image’s original (top) and the reconstruction of its per-image selected scheme (bottom, labelled with scheme, RMS and total bits). The selector takes the perceptual `dct8-db-perc` on the smooth Flamingo and the plain `dct8-db` on the detailed Bridge and 2026.

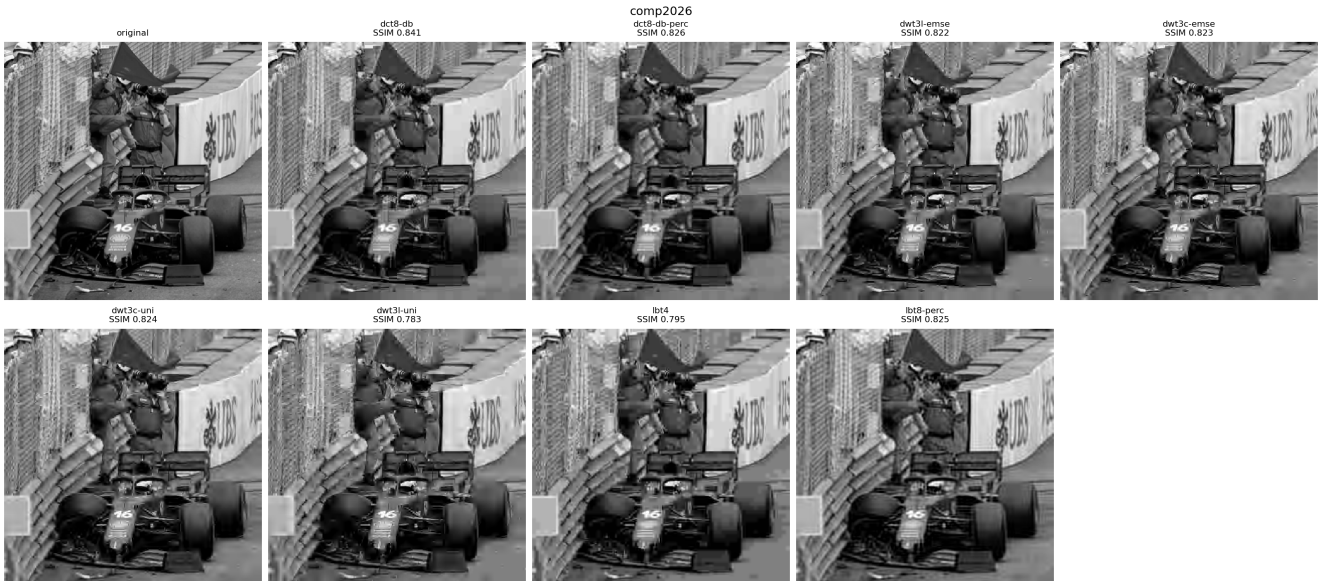


Figure 20: Every registry scheme decoded on the 2026 image at the 5 kB budget (the dashboard grid of experiments/dashboard.py; SSIM under each panel). Look for the deblocked DCT (`dct8-db`, the selected scheme) holding the cleanest edges, the perceptual twin and `lbt8-perc` smoothing texture at a small SSIM cost, and `lbt4`/`dwt31-uni` (lowest SSIM) visibly softer.

2026 signed error per registry scheme (shared scale) — block grid (DCT) vs ringing (DWT/LBT)

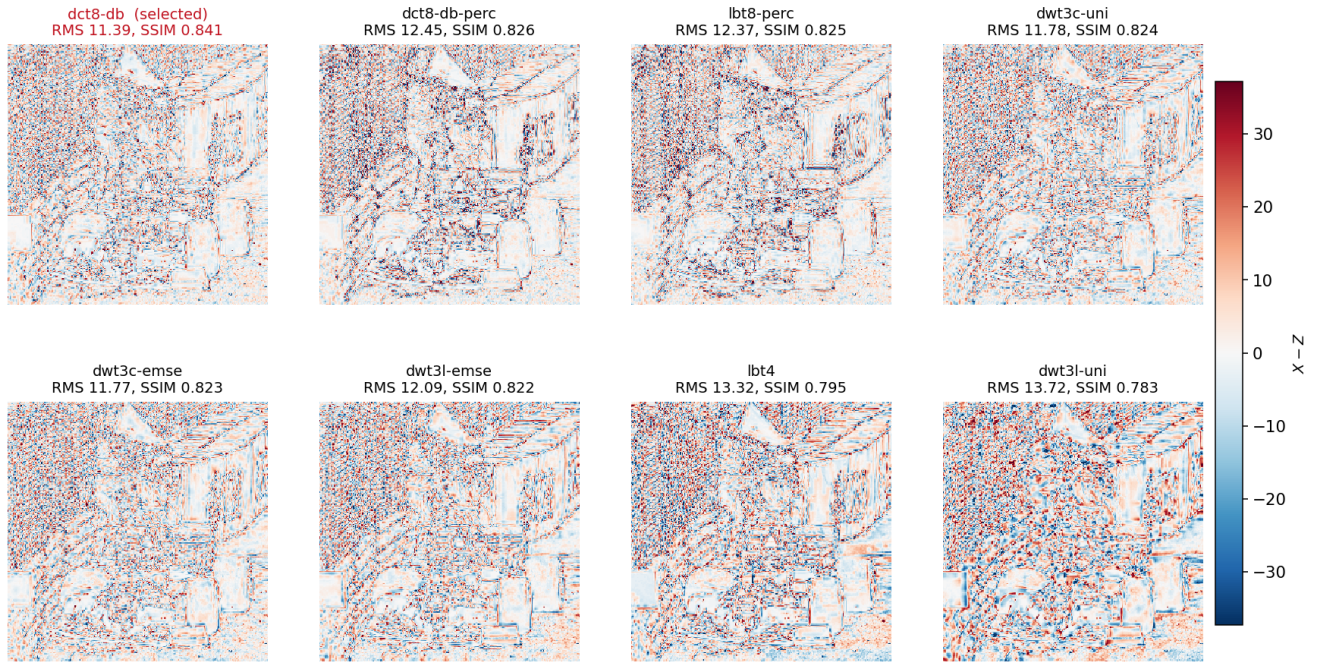


Figure 21: Signed error $X - Z$ for every registry scheme on the held-out 2026 image (red/blue, a single shared scale clipped to the 99th percentile; schemes ordered by SSIM, the selected `dct8-db` in red). Look for the common structure — all error on edges and the wall lettering, flat regions near-zero — but a different error texture per family: the DCT’s tight near-edge speckle (cleanest flats), the CDF-9/7’s thin ringing along contours, the LeGall 5/3’s broader scatter, and the LBT-4’s heavy residual everywhere.

Competition field on the held-out 2026 image (13 entries, ranked by RMS) — ours won the subjective vote



Figure 22: The full competition field on the held-out 2026 image: the baseline reference and all 13 submitted entries (one per group, anonymised by hash), ranked by RMS, downloaded from the public results page. Our entry (`ea53f9`, the `dct8-db` codec) is boxed in red — 4th by RMS (11.39, within 0.2 of the best) yet first on the subjective vote. Look for how little the top entries differ in RMS (11.2–11.8) despite visibly different artefacts, which is exactly why the contest is decided by eye rather than by the RMS column.

References

- [1] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [2] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th Asilomar Conf. on Signals, Systems and Computers (ACSSC)*, vol. 2, pp. 1398–1402, 2003.
- [3] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018.
- [4] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Image quality assessment: Unifying structure and texture similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 5, pp. 2567–2581, 2022.
- [5] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.
- [6] D. Le Gall and A. Tabatabai, "Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 761–764, 1988.
- [7] A. Cohen, I. Daubechies, and J.-C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics*, vol. 45, no. 5, pp. 485–560, 1992.
- [8] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 205–220, 1992.
- [9] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *Journal of Fourier Analysis and Applications*, vol. 4, no. 3, pp. 247–269, 1998.
- [10] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1309–1324, 2000.
- [11] S. W. Golomb, "Run-length encodings," *IEEE Transactions on Information Theory*, vol. 12, no. 3, pp. 399–401, 1966.
- [12] J. Duda, "Asymmetric numeral systems: Entropy coding combining speed of huffman coding with compression rate of arithmetic coding," *arXiv preprint arXiv:1311.2540*, 2014.
- [13] P. List, A. Joch, J. Lainema, G. Bjøntegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 614–619, 2003.
- [14] S. Midtskogen and J.-M. Valin, "The AV1 constrained directional enhancement filter (CDEF)," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 1193–1197, 2018.