Verified Perceptual Image Compression: Delivering a Per-File Quality Floor for Professional Photography

Sebastian Haas
Independent Researcher and Wedding Photographer
kontakt@sebastianhaas.eu

***Abstract*-Commercial image optimizers ship fixed settings and hope for quality. Measured with an open perceptual metric, a conventional "quality 85" export spans SSIMULACRA2 scores from 69 to 87 depending on content, and the market-leading proprietary optimizer JPEGmini Pro 5 spans 75.0 to 93.5 across a single wedding shoot. We invert the contract: instead of settings, the system delivers a verified perceptual quality floor per file, measured at full resolution against the source. The architecture combines an open metric ported to both CPU and GPU (agreement within 0.01 points), a tile-guided, bias-corrected search that verifies every delivered byte, a per-image competition between two JPEG encoders with rate–distortion refinement of quantization tables, and a data-derived plan-gating rule that removes 67% of compute at zero measured byte regret. On a complete 535-photograph wedding (45 MP, 10.95 GB), a guaranteed floor of 85-above the incumbent's worst file-yields a gallery 43% smaller than JPEGmini; a texture-adaptive tier is smaller on 100% of files. We report throughput improvements from 15.1 to 1.95 s per photograph, a distilled neural metric (r = 0.994, 209 ms per 45-MP frame on the Neural Engine), an embedded psychophysical calibration of the user's visibility threshold, and the operating points at which the incumbent remains superior: near-lossless quality (+9.7% bytes) and raw throughput (4.6×). All figures are reproducible from per-file data published with the implementation.**

***Index Terms*-image compression, perceptual quality assessment, SSIMULACRA2, JPEG, rate–distortion optimization, knowledge distillation, psychophysics.**

I. Introduction

Professional photographers deliver the same body of work through several channels-a web gallery, social media, a full-resolution client package, and a long-term archive-each with its own size constraint. The state of practice for producing these deliverables is either a fixed encoder setting chosen once and applied to every image, or a proprietary optimizer that selects settings automatically according to an undisclosed quality model.

Both approaches share two deficits. First, delivered quality is uncontrolled: a fixed setting produces widely varying perceptual results because the relationship between encoder quality and perceived quality depends on image content. Second, the quality claim is unverifiable: the photographer receives smaller files and an assurance that they "look the same," with no per-file measurement and, in the proprietary case, no reproducible metric behind the assurance.

This paper describes holdmybyte, a compression system built on the opposite contract. The user specifies a perceptual target; the system searches, per image, for the smallest encoding that provably meets it, and every delivered file is measured at full resolution against its source using an open, reference-validated metric. The contribution is not a new codec-the system uses mozjpeg [6], jpegli [7], libwebp, libavif [12] and libjxl [3]-but an architecture in which an open perceptual metric is the *acceptance authority* rather than an optimization heuristic, and in which the engineering that makes such verification affordable is itself derived from measurement.

We make four claims, each supported by per-file data in Section V: (1) on a complete professional wedding shoot, a verified quality floor above the incumbent's worst file coincides with a 43% smaller gallery; (2) the compute cost of full-resolution verification can be reduced by a factor of 7.7 without changing a single output byte, through profiling-driven engineering rather than algorithmic approximation; (3) a distilled convolutional metric reproduces the reference at full-image level with r = 0.994 while running in 209 ms, but does not accelerate the search, for reasons we quantify; and (4) the incumbent retains measurable advantages at two operating points, which we report rather than omit.

II. Related Work

Perceptual metrics. SSIM [1] established structural similarity as a proxy for perceived quality; its multi-scale descendants and the Butteraugli family informed SSIMULACRA2 [2], developed within the JPEG XL project [3], which combines a six-scale XYB-space structural term with edge-based artifact and detail terms and is calibrated to subjective ratings. We use SSIMULACRA2 unmodified as the arbiter throughout. DISTS [11] unifies structure and texture similarity and is the closest academic reference for the grain-aware verification discussed in Section VII.

Proprietary optimization. JPEGmini is built on a perceptual measure described by Shoham et al. [4] and protected by U.S. Patent 8,908,984 [5] among others. The measure is block-grid focused, combining per-region PSNR, added artifactual edges along coding-block boundaries, texture change and edge preservation. This design explains a failure mode we observe empirically: grain smoothing and in-block ringing create no new block-boundary edges and are therefore under-detected. Our system deliberately does not reproduce this measure or its requantization scheme; it replaces the approach with open measurement and verified search. The claim element concerning weighted averaging of quantization matrices is flagged in Section VI.

Rate–distortion optimization of quantization tables. Ratnakar and Livny's RD-OPT [8] established per-image optimization of DCT quantization tables against a distortion measure. Our refiner (Section III-C) is a greedy variant steered by SSIMULACRA2 rather than mean squared error.

Knowledge distillation. Hinton et al. [9] introduced teacher–student training; we apply it to distil a full-resolution perceptual metric into a small network usable on Apple's Neural Engine (Section III-G).

Psychophysics. The one-up–one-down staircase with catch trials follows Levitt [10]; we embed it in the product to measure the user's own visibility threshold (Section III-F).

III. System Architecture

A. Open Metric as Acceptance Authority

SSIMULACRA2 was implemented twice: a CPU reference in Swift with Accelerate, following the libjxl implementation, and a GPU implementation in Metal. The GPU version encodes all six scales, both images and all fifteen blur passes per scale into a single command buffer; per-pixel arithmetic runs in single precision, threadgroup partial sums of at most 256 values keep accumulation error bounded, and the partials are reduced in double precision on the CPU. Shaders are compiled in safe math mode-fast-math relaxations are excluded because the score constitutes the product's guarantee.

TABLE I

Metric implementations, 24-megapixel frame including decode.

Implementation	Time	Deviation from reference
CPU (Swift/Accelerate)	2.46 s	-
GPU (Metal, safe math)	0.87 s	$\leq$ 0.01 score points

The two implementations agree within 0.01 points (Table I), and a regression test enforces this. The GPU is an accelerator, not a dependency: on any Metal failure the path falls back to the CPU reference. A later optimization (Section IV) splits the GPU computation into a per-image reference session and a per-candidate pass; a test asserts bit-identical scores between the split and monolithic paths.

B. Verified Search

For each image and encoder plan, the search seeks the smallest encoding whose full-frame score meets the target. A naive bisection over encoder quality with one full-frame measurement per round is unaffordable at 45 megapixels. The search therefore proceeds in three stages.

Tile proxy. Four 512×512 tiles are selected by gradient energy-the hardest regions-and their combined score is deliberately pessimistic (half the minimum, half the mean), so that the proxy tends to land high rather than low.

Bias-corrected interpolation. The first full-frame measurement reveals the offset between tile estimate and truth for this image and encoder. Subsequent candidates are chosen by interpolation against the corrected target, replacing blind stepping.

Full-frame verification. Every candidate that may be delivered is measured at full resolution. If the result exceeds the target by more than 0.75 points, a trim loop selects a lower landing from the corrected tile curve and verifies again.

The invariant is that a misestimate costs one round of computation and never quality. This is what makes aggressive targets shippable: the web tier's target of 70 would be reckless under any scheme that infers delivered quality rather than measuring it.

C. Encoder Competition and Table Refinement

For JPEG output, four plans compete: mozjpeg and jpegli, each with 4:2:0 and 4:4:4 chroma subsampling. Chroma resolution receives its own plan rather than being inferred from the target, because subsampling caps the achievable score on chroma-critical content at any bitrate. On the mozjpeg winner, a greedy refiner coarsens individual frequency bands of the quantization tables while the tile proxy still clears the target, then verifies at full resolution; low-frequency bands are never modified. The refiner's safety margin is sized from the search's measured tile-to-full-frame bias rather than a constant.

D. Data-Derived Plan Gating

Running four plans per image is expensive. From 533 fully measured wedding photographs, jpegli-4:2:0 won 92% of comparisons; the remaining plans won only on chroma-heavy content. The gating rule is a single threshold on opponent-colour high-pass energy (2.42): the first wave runs jpegli-4:2:0 alone below it and the full field above it. A boosted-tree model trained on the same data was only marginally better than this constant and was rejected in favour of explainability.

The rule is safe by construction: if the first wave fails to reach the target, the skipped plans run afterwards (*fail-open*). Measured on a held-out set, gating removed 67% of compute at 0.00% byte regret on every file.

E. Adaptive Targets

A *Classic* tier reproduces the incumbent's measured behaviour so that migrating users obtain familiar file sizes. Across the measured material, JPEGmini's delivered score falls linearly with texture density on detailed images (fit: $88 - 0.62 \cdot p50$, where p50 is the median per-block high-pass energy) and collapses to approximately 74 on smooth-dominated images. The tier reproduces this unevenness deliberately; unlike the incumbent, each file reports the score it actually received.

F. Embedded Psychophysics

The product includes a one-up–one-down staircase [10] presenting pairs of the user's own images-hardest crops at native scale-with interleaved catch trials of identical pairs to estimate response reliability. For the author, the visibility threshold converged near score 74. This measurement carried a product decision: web-tier targets of 70, 78 and 85 were delivered as complete folders and compared blind; no difference was visible at 70, which was adopted with the documented residual risk that the staircase placed the 1:1 pixel-peeping threshold at 74.

G. Distilled Metric

To reduce the cost of full-frame measurement, the reference metric was distilled [9] into a 0.67-M-parameter convolutional network. The dataset comprises 38,520 patch pairs from real wedding photographs, split by image to prevent identity leakage; inputs are nine channels (reference, candidate, absolute difference); two resolution branches capture fine artifacts and large-scale structure; a margin ranking loss enforces ordering; the critical score range is weighted twice.

TABLE II
DISTILLED METRIC ON UNSEEN IMAGES.

Measure	Patch level	Full image (aggregated)
Mean absolute error	1.73	0.55
Ordering accuracy	98.6%	100%
Correlation r	-	0.994
Latency, 45 MP (Neural Engine)	-	209 ms

Aggregation decorrelates per-patch error almost completely (Table II). Integrated as a search accelerator, however, the network made the engine 29% slower and outputs 15% larger: a full-frame round is 56% encode and decode, which the network does not replace, and its residual scale offset contaminated the bias correction of the trim loop. The model is retained as an optional component; its intended use is regional quality maps (672 patch scores per pass) for future region-level guarantees.

H. Safety Invariant

All writes pass through a single audited path that never targets the source file, never overwrites, resolves name collisions, and replaces only explicitly via the system trash. Regression tests protect the invariant. The archive tier transcodes JPEG to JPEG XL with byte-exact reconstruction, verified by SHA-256.

IV. THROUGHPUT ENGINEERING

The initial cost of verification was severe: 327 CPU-seconds for five photographs, roughly 21× the incumbent. Nine optimization stages reduced the sustained cost from 15.1 to 1.95 s per 45-MP photograph (Table III). Every stage was evaluated in paired A/B runs; stages marked *byte-identical* produced outputs identical to the previous stage on every file.

TABLE III
OPTIMIZATION STAGES, SECONDS PER 45-MP PHOTOGRAPH.

Stage	Measure	s/photo	Outputs
0	Baseline engine	15.1	-
1–2	Plan gating v1, v2	4.3	byte-identical
3	Batch width = cores/2	3.4	byte-identical
4	Dedicated worker threads	2.4	byte-identical
5	Lazy lossless rescue	−27%	byte-identical
6	GPU reference sessions	±0 wall, −30% GPU	byte-identical
7	Opaque fast paths, shared scans	−4.8%	byte-identical
8	Direct libjpeg-turbo decode	−6.7%, −1.2% bytes	score-verified
9	Entropy-light probe encodes	−4.2% ungated	byte-identical

Two findings are of general interest. First, stage 4 was diagnosed from utilisation alone: 6,335 CPU-seconds over 1,830 wall-seconds indicated 3.5 of 10 cores in use. Elimination experiments (GPU lanes, batch width, subprocess gate) each moved nothing; the decisive test ran the same 19 images as 19 separate processes in 39 s against 61 s in-process, proving that nested Grand Central Dispatch parallelism combined with blocking semaphores starved the cooperative thread pool. Dedicated worker threads restored full utilisation at byte-identical output.

Second, stage 5 was found by a phase profiler in its first line of output: 66% of all CPU time was spent in a lossless re-encode that "rode along" on every image as a safety net-for a candidate mathematically incapable of beating any perceptual result below 75% of the original size. Making it lazy removed 27% of wall time. Conversely, an audit proposal to skip mozjpeg's entropy-only passes in intermediate rounds-provably pixel-neutral, and now guarded by a test-measured slower in both implementations, because the predictor had already reduced searches to one or two full rounds, below the re-encode break-even of 2.6. Only its application to probe encodes, whose bytes are never delivered, survived measurement.

V. EXPERIMENTAL RESULTS

A. Methodology

All comparisons follow three rules: (1) comparisons start from identical pixels; (2) quality is measured per file with the open metric, never inferred; (3) an A/B across engine changes must show byte-identical outputs, or it is documented as a behavioural change through size and score distributions. The primary dataset is a complete wedding shoot: 535 photographs, 45 megapixels, 10.95 GB of fresh Lightroom export. JPEGmini Pro 5 ran at default settings on an APFS clone. Timings were obtained on an Apple M1 Max with 64 GB; absolute times measured under concurrent background load are excluded, and only paired relative measurements from those sessions are reported.

B. Head-to-Head Comparison

TABLE IV
COMPLETE WEDDING, 535 PHOTOGRAPHS, 10.95 GB.

Operating point	Gallery	Savings	Score min / med / max
JPEGmini Pro 5	5.36 GB	−50.0%	75.0 / 89.9 / 93.5
Ours, floor 85 guaranteed	3.05 GB	−72.1%	85.0 / 85.5 / 87.3
Ours, Classic tier	1.35 GB	−87.7%	74.0 / 74.5 / 76.2

At a guaranteed floor of 85-ten points above the incumbent's worst file-the gallery is 43% smaller (Table IV). Per file, the floor-85 tier is smaller than JPEGmini on 94% of images (median −46%); the Classic tier is smaller on 100%. On a mixed set of five camera JPEGs (24–45 MP) compressed to exactly the score JPEGmini delivered per image, the gallery was 28.5% smaller with every file smaller.

C. Delivery Matrix

TABLE V
ONE WEDDING, FOUR DELIVERIES (535 PHOTOGRAPHS).

Delivery	Format	Result	Savings
Web gallery	WebP 2560 px, target 70	159 MB	−98.5%
Social	JPEG 1440 px, target 85, sRGB	155 MB	−98.6%
Client package	JPEG full size, Classic	1.35 GB	−87.7%
Archive	JPEG XL, lossless, byte-exact	8.55 GB	−21.9%

The four deliveries together occupy 10.21 GB-less than the 10.95 GB source export alone (Table V). Sustained processing times were 3.7, 7.1, 17.4 and 11.3 minutes respectively.

D. Where the Incumbent Wins

Forcing the engine to reproduce JPEGmini's delivered scores exactly per file (median ≈ 90) yields galleries 9.7% larger than JPEGmini's: above score ≈ 88, the incumbent's quantization is more efficient than ours. In throughput, JPEGmini processes 0.42 s per photograph against our 1.95 s (4.6×); this is the cost of full-resolution verification of every delivered byte. Both figures are reported in the product's documentation.

E. Reliability Finding

Instrumented harvesting exposed a silent subprocess failure under sustained load: the jpegli helper produced output on only 108 of 535 images, with a timestamp pattern of twenty minutes of success, total failure, and recovery at batch end. After hardening (spawn gate, null-device stdout, drained stderr, bounded retry), jpegli won 92% of comparisons and the full competition delivered 8% fewer bytes than mozjpeg alone. The failure had silently cost approximately 9% per affected image; plan failures are now logged as training rows so that distributional anomalies are visible.

VI. LIMITATIONS

The study covers one observer, one shoot and one camera. The arbiter is our own-open and reference-validated-implementation of the metric. JPEGmini ran at default settings. All calibrations (gating threshold, Classic curve, web target) derive from this material and generalise only after validation on further shoots; the fail-open design protects quality but not the magnitude of savings on unfamiliar content.

SSIMULACRA2 has a known blind spot: statistically equivalent film grain is penalised by up to 18 points while smoothing is rewarded. A grain-synthesis prototype delivering a further 22–25% is therefore parked pending a grain-aware verification.

Before commercial release, patent counsel should review U.S. Patent 8,908,984 [5]: the refiner's retreat step averages a candidate table with the original, arithmetic that resembles one claim element. The system does not implement the patent's block-boundary measure, and infringement requires all claim elements, but the overlap is flagged explicitly.

VII. FUTURE WORK

Three directions follow directly from the limitations. A grain-aware verification-combining the metric's content term with a statistical grain-equivalence term, its pass boundary calibrated by the embedded staircase-would unlock the parked grain synthesis. Regional quality contracts ("face ≥ 85, background may yield") become possible with the distilled metric's per-patch maps, a guarantee a single global score cannot express. Finally, multi-observer calibration through the staircase would produce a just-noticeable-difference dataset from professional photographers on their own material.

VIII. CONCLUSION

An open perceptual metric used as an acceptance authority, combined with full-resolution verification of every delivered file, produced galleries 43% smaller than the market-leading proprietary optimizer at a higher quality floor, and did so with every claim traceable to per-file data. The verification cost, initially prohibitive, was reduced by a factor of 7.7 through measurement-driven engineering with byte-identical outputs at almost every stage. The incumbent remains superior at near-lossless quality and in raw throughput; both are documented. The method's central lesson is procedural rather than algorithmic: measure quality instead of inferring it, verify instead of hoping, and publish the operating points at which the approach loses.

REFERENCES

[1] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[2] J. Sneyers, "SSIMULACRA 2: Structural SIMilarity Unveiling Local And Compression Related Artifacts," Cloudinary, 2023. [Online]. Available: https://github.com/cloudinary/ssimulacra2

[3] J. Alakuijala et al., "JPEG XL next-generation image compression architecture and coding tools," in *Proc. SPIE 11137, Applications of Digital Image Processing XLII*, 2019.

[4] T. Shoham, D. Gill, and S. Carmel, "A novel perceptual image quality measure for block based image compression," in *Proc. SPIE 7867, Image Quality and System Performance VIII*, 2011.

[5] S. Carmel et al., "Apparatus and methods for recompression of digital images," U.S. Patent 8,908,984, Dec. 2014.

[6] Mozilla, "mozjpeg: Improved JPEG encoder," 2014. [Online]. Available: https://github.com/mozilla/mozjpeg

[7] Google, "jpegli: A JPEG encoder and decoder," 2024. [Online]. Available: https://github.com/google/jpegli

[8] V. Ratnakar and M. Livny, "RD-OPT: An efficient algorithm for optimizing DCT quantization tables," in *Proc. Data Compression Conf. (DCC)*, Snowbird, UT, USA, 1995, pp. 332–341.

[9] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," arXiv:1503.02531, 2015.

[10] H. Levitt, "Transformed up-down methods in psychoacoustics," *J. Acoust. Soc. Amer.*, vol. 49, no. 2B, pp. 467–477, 1971.

[11] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Image quality assessment: Unifying structure and texture similarity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2567–2581, May 2022.

[12] Alliance for Open Media, "libaom: AV1 codec library." [Online]. Available: https://aomedia.googlesource.com/aom