Ghostframe Replay and the GFS2 codec

Replaying annotations after JPEG cropping and recompression

Zakaria Gharzouli
Residual Labs, Moissy-Cramayel, France
ORCID: 0009-0005-7604-4709 • contact@residual-labs.fr
Preprint v1 - November 24, 2026 - CC BY 4.0

Abstract

A screenshot shows the state of an interface, while an issue report often needs a sequence of actions. Ghostframe Replay stores positions, durations, and notes in a visible pixel strip at the bottom of a JPEG. The reader reconstructs these data and places the markers on the received screenshot. GFS2 combines a payload-dependent height, four tiles including one XOR parity tile, CRC-32 checks, and geometry estimated from finder bars. The minimum strip height within this architecture is 40 to 88 pixels for 11 to 243 payload bytes. At equal area, GFS2 succeeds in 98/132 and 96/132 cases on four synthetic screenshots with Lanczos and bicubic resampling, against 30/132 and 32/132 for the best QR or rMQR reference per case. On twelve held-out ScreenSpot-Pro screenshots, a bilinear retry recovers 13 additional cases with the same JPEG. It reaches 240/240 band-present cases, against 190/240 for the best QR, rMQR, or rectangular Data Matrix reference. All twelve band-removal controls are rejected. The paired gain is 19.8 percentage points over 252 cases, with a descriptive interval of [12.7, 25.8] points. The prespecified threshold of ten winning screenshots out of twelve is unmet. The experiments characterize reproducible local transformations.

Keywords JPEG, interface annotations, raster code, XOR parity, bilinear interpolation, reproducibility.

1. Problem and contribution

A standard JPEG export represents a still image [1]. A pipeline that retains the same poster raster from two animations yields identical observations. A reader given that observation alone cannot reconstruct both original sequences. GFS2 writes steps **into the pixels** before processing. An ordinary JPEG reader shows the screenshot and its strip. Ghostframe Replay extracts the strip data and animates markers on the screenshot. Native APNG animation uses a separate mechanism [2].

The contribution is a narrow-strip code with tile-level parity and geometry recovery suited to cropped screenshots. This study derives the strip sizing rule, measures equal-area recovery, and isolates the effect of a second sampling pass. It makes no claim of a new error-correction principle or of recovery after a service removes the strip.

2. Serialization and sizing

Payload. A five-byte header comprises the width W and height H of the screenshot before the strip (network-order uint16) and the number n of steps (uint8). Each step stores x and y (two uint16 values), duration d in units of 100 ms (uint8), the note length u (uint8), and u UTF-8 bytes. Positions are pixel coordinates in an export raster 928 to 1,600 px wide and at most 1,440 px high before the strip. The prototype accepts 1 to 12 steps, notes of up to 96 bytes, and durations from 100 to 10,000 ms in 100 ms increments. A separate limit applies to the combined payload.

(1) $L = 5 + \Sigma_i\ (6 + u_i)$, with $11 \le L \le 243$.

Capacity. Each tile uses 48 data columns, or six bytes per row, and a 15-byte packet header. With r data rows, each tile carries $s = 6r - 15$ shard bytes. Three data shards carry L bytes after zero padding. The fourth tile carries parity. Therefore $L \le 3s$.

(2) $r^*(L) = \lceil (L + 45) / 18 \rceil = \lceil (15 + \lceil L / 3 \rceil) / 6 \rceil$.

The smallest integer satisfying $18r \ge L + 45$ gives (2). For the domain in (1), $4 \le r^* \le 16$. The minimum applies to these fixed parameters (three shards, a 15-byte header, six bytes per row). It establishes no general lower bound for other codes.

Strip. Each tile measures $58 \times (r^* + 6)$ square modules, with four pixels per module side. The four adjacent tiles occupy an active width of 928 px centered on the screenshot. The extension's white background spans the full width W. An export based on a screenshot of height H therefore has $H + h$ rows, and its actual extension is $W \cdot h$ pixels.

(3) $h(L) = 4[r^*(L) + 6]$, and $A(L) = 928 \cdot h(L)$ px^2 for the active area.

Payload L (bytes)	Data rows r*	Bytes per shard s	Capacity 3s	Padding bytes	Height h (px)
11	4	9	27	16	40
84	8	33	99	15	56
156	12	57	171	15	72
243	16	81	243	0	88

Table 1. Sizing of the four synthetic payloads at fixed active width.

Example. Three notes of 18, 19, and 24 bytes give $L = 5 + 3 \cdot 6 + 61 = 84$. Then $r^* = 8$, $h = 56$ px, and the active area is 52,128 px^2. With the same parameters, GFS1 has a fixed height of 88 px. GFS2 reduces the height and active area by 36.4% in this example. For $L = 11$ the reduction is 54.5%. These ratios measure raster geometry. The JPEG file size also depends on the pixels and encoder.

3. Parity, integrity, and readout

3.1 Shards and checks

Padding to 3s bytes yields three data shards D_0, D_1, and D_2. The following operations act independently on each byte.

(4) $P = D_0 \oplus D_1 \oplus D_2$, and $D_k = P \oplus D_i \oplus D_j$ for $\{i, j, k\} = \{0, 1, 2\}$.

For hexadecimal bytes A5, 3C, and 71, parity is E8. If 3C is missing, E8 ⊕ A5 ⊕ 71 recovers 3C. Three valid tiles suffice to recover an arbitrary single erasure. The reader rejects groups with fewer than three tiles. There is no guarantee for two missing tiles.

The 15-byte header stores the ASCII magic GFS2 (4), the payload CRC-32 (4), L (2), a tile index from 0 to 3 (1), and the packet CRC-32 (4). The checksum field is zeroed for the packet checksum calculation. CRC-32 uses reflected polynomial 0xEDB88320, initial state 0xFFFFFFFF, and final XOR 0xFFFFFFFF. For each byte the algorithm XORs the state with the byte and performs eight right shifts, XORing the polynomial when the outgoing bit is 1. The reader verifies both packet and reconstructed payload checksums. CRCs detect ordinary corruption. Authenticating the sender requires a cryptographic signature.



Figure 1. Actual strip of the 84-byte synthetic case. Each tile has its own header. The last tile carries the XOR of the three data shards.

3.2 Geometry and bit decisions

The packet bits are drawn most significant first. Finder bars occupy columns 1–2 and 55–56 and rows 1 through r + 4, with zero-based indices. In the received image let a and b be their horizontal centers and t and z be the mean heights of the upper and lower transitions. The apparent module pitches are a_x and a_y.

(5) $a_x = (b - a) / 54$, $a_y = (z - t) / (r + 4)$, $x_0 = a - 2a_x$, $y_0 = t - a_y$.

(6) $x(c) = x_0 + (c + 5.5)a_x$, $y(j) = y_0 + (j + 3.5)a_y$.

For RGBA pixels the computed luminance is $Y = \lfloor(54R + 183G + 19B)/256\rfloor$. A module is 1 when the luminance at its estimated center is below 128, and 0 otherwise. The ordinary reader samples the pixel containing the estimated point. The experimental reader keeps any ordinary result and retries tile detection using bilinear sampling after a failure. If α and β are the fractional offsets between four neighboring pixel centers with values Y_{00}, Y_{10}, Y_{01}, and Y_{11}, the estimator is

(7) $\hat{Y} = (1 - \alpha)(1 - \beta)Y_{00} + \alpha(1 - \beta)Y_{10}$
$+ (1 - \alpha)\beta Y_{01} + \alpha\beta Y_{11}$.

The reader tries admissible heights and groups verified tiles by magic, length, and payload CRC. Estimated module pitches must fall between 1.5 and 12 px and their horizontal/vertical ratio between 0.66 and 1.5. The reader refines the horizontal scale from the distance between the extreme valid tiles, originally spaced at multiples of 232 px. Finder height provides vertical scale, replaced by horizontal scale when the measured heights agree with uniform resizing. Markers in the received JPEG are placed using

(8) $T(x, y) = (\ell + s_x x, v + s_y y)$.

Here ℓ and v are the estimated offsets of the source screenshot, and s_x and s_y are its scale factors. The browser then replays notes and durations on the received pixels. Cropped-out image regions remain unavailable.

4. Experimental protocol

Synthetic comparison. Four screenshots carry the four payloads of Table 1. Each payload uses the same screenshot and 928 px active strip for GFS2, square QR, and rMQR [3, 4]. The initial JPEG uses quality 92. Each filter produces 33 cases per screenshot, including the original and a 30-case cross of three crops (intact, removal of a left or right tile plus 32 px at the top), five scales (100, 63, 50, 40, and 35%), and two JPEG qualities (65 and 30). Additional controls are a grayscale conversion after crop and reduction to 50% and complete strip removal. Pillow applies Lanczos in one run and bicubic in the other. The two runs share screenshots and some transformations and cannot be treated as independent samples.

Professional-screenshot validation. Two disjoint sets of twelve shots cover the same twelve ScreenSpot-Pro applications [5, 6]. Selection takes the eligible file with the lowest SHA-256 of the seed,

application, and path tuple for each application. Steps are up to three target centers visible after both crops, accompanied by their public text and benchmark-assigned durations. EViews replaces AutoCAD, which had no eligible image in the initial check. The second set excludes the first selected paths, and all 24 file hashes are distinct. OpenCV encodes the initial JPEG at quality 92, applies the crop and INTER_AREA resizing, and recompresses [7]. Each screenshot contributes an 18-case cross of three crops, three scales (63, 50, and 40%), and two JPEG qualities (65 and 30), plus original, grayscale, and strip-removal cases.

References. Every family receives the same bytes and area as GFS2. The benchmark tries 2, 3, or 5 data fragments and one XOR parity fragment. Square QR uses levels L, M, Q, and H and versions 1 to 8 in the synthetic tests or 1 to 15 on ScreenSpot-Pro. rMQR uses levels M and H and 32 versions. The validation adds rectangular Data Matrix variants numbered 25 to 48 in Zint, including DMRE, generated and read with ZXing-C++ [8, 9]. Quiet zones are retained. For each family and case the best tested configuration is retained, and the combined reference succeeds when any reference family succeeds. A separate control provides exact grid geometry to test byte recovery alone. It does not impose a placement criterion.

Primary criterion. All readers receive the bottom 96 rows of the transformed JPEG. The recovered message must match byte for byte. Targets are evaluated in received-image coordinates, using source pixel centers. Let T be the ground-truth transform computed using effective output dimensions and C the set of visible markers. A success must satisfy

$$(9)\quad E = \max \{ \|T(p) - T(p)\|_2 : p \in C \} \le 2 \text{ px}.$$

For the ScreenSpot-Pro splits, d_i is the number of GFS2-only successes minus reference-only successes divided by 21 for screenshot i. The gain is the mean of the twelve d_i. The descriptive interval uses the 2.5% and 97.5% quantiles from 20,000 bootstrap replications of whole screenshots (seed 20261123). The prespecified criterion requires a positive lower bound **and** a gain on ten or more screenshots. The protocol selects the twelve applications. The interval does not quantify uncertainty for a randomly sampled application population.

5. Results

Method	Lanczos, successes / 132	Bicubic, successes / 132
GFS2, ordinary readout	**98**	**96**
Square QR	30	30
rMQR	28	30
Best reference per case	30	32
rMQR, known grid, bytes only	72	71

Table 2. Equal-area synthetic runs. The last row is a byte recovery diagnostic with externally supplied geometry.

GFS2 succeeds on all cases recovered by a reference, plus 68 additional Lanczos cases and 64 additional bicubic cases. The observed margins are 51.5 and 48.5 percentage points. The largest error among GFS2 successes is approximately 0.502 and 0.494 px, respectively. GFS2 uses 4 px export modules. The selected square QR layouts use at most 2 px modules and the rMQR layouts at most 3 px. The experiments compare complete systems and do not identify the causal contribution of each component.

ScreenSpot-Pro measure	Split 1 / 252	Split 2 / 252
GFS2, ordinary readout	226	227
GFS2, ordinary then bilinear	Not evaluated on full split	**240**
Best reference per case	195	190
GFS2-only / reference-only successes	33 / 2	50 / 0
Paired gain, percentage points	12.3	19.8
Descriptive 95% interval, points	[6.0, 18.3]	[12.7, 25.8]
Screenshots won / tied / lost	9 / 2 / 1	9 / 3 / 0

Table 3. Exclusive counts use ordinary readout in Split 1 and two-pass readout in Split 2. Each split includes twelve strip-removal controls.

In Split 1, there was no advantage in success counts at 63% or 50%. At 40%, GFS2 succeeds in 58/72 cases and the reference in 27/72. DaVinci Resolve accounts for both reference-only successes. Out of 14 band-present failures, supplied-grid nearest-pixel sampling produces at least three correct tiles in 3 cases, while bilinear sampling does so in 13. With the reader estimating the grid, a bilinear diagnostic recovers 12 of these 14 cases, including both DaVinci cases. This diagnostic motivated the two-pass reader.

In Split 2, the readout change recovers 13 cases on nine screenshots with identical JPEGs, payloads, and strip areas. It reaches 240/240 band-present successes and rejects all 12 strip-removal controls. The best reference succeeds in 190/240 band-present cases. This is a 20.8 percentage-point difference among band-present cases, compared with 19.8 points over all 252 cases. The maximum measured error is 0.7631 px. The three ties in Figure 2 are at the ceiling of 20/20. Both splits fall short of the ten-screenshot target, with nine wins.

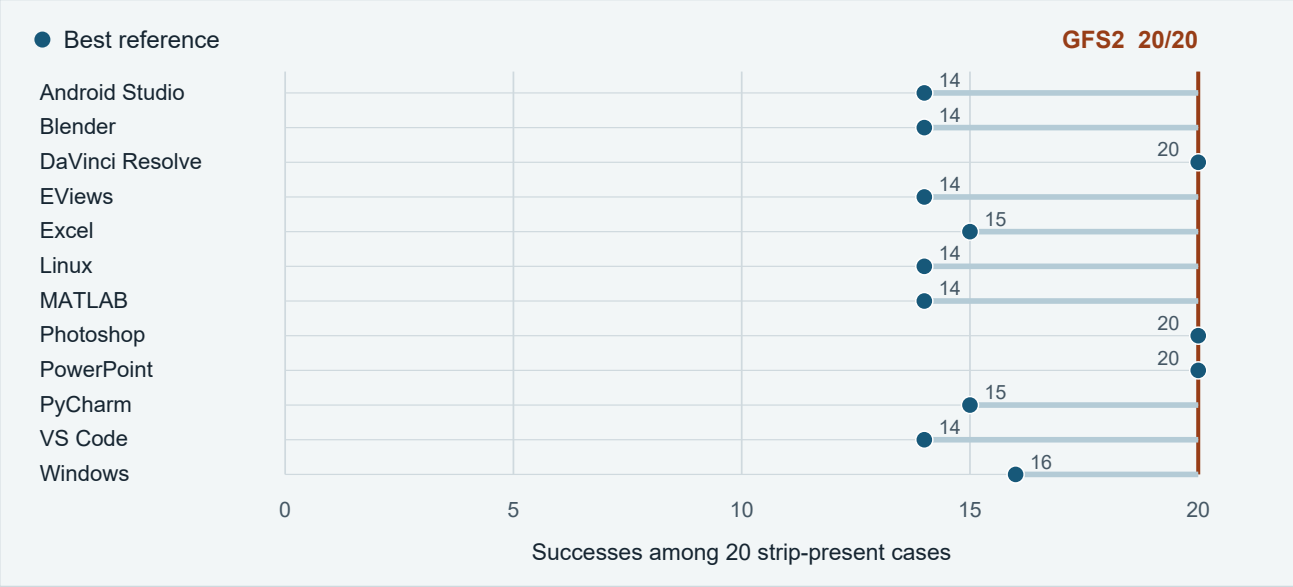

Figure 2. For each application, two-pass GFS2 recovers 20/20 band-present cases. Points show the number of cases recovered by the best reference. The strip-removal control is excluded from this figure.



A. Exported JPEG B. Replayed markers on received JPEG

Figure 3. Measured example. The 1,120 × 706 px JPEG is cropped to remove tile 0, reduced to 50% with Lanczos, and recompressed at JPEG quality 65. In the 396 × 337 px received image, all 84 bytes are exact and the maximum error for the three markers is 0.00975 px. The numbered overlays on the right illustrate replay. The received JPEG is supplied separately without these overlays.

6. Applications to test and limitations

The 243-byte payload suits short sequences and small identifiers. The format is potentially useful when an image circulates separately from text and a Ghostframe reader is available to the recipient. The following are evaluation scenarios without deployment evidence.

Sector	Concrete use	Proposed measure
Support, QA and development	Attach a screenshot with steps for reproducing an issue.	Correct replay rate and time to reproduce.
Training and documentation	Package a short interface walkthrough for offline use.	Task completion accuracy and time per exercise.
CAD, design and media	Point out commands or revisions on a tool screenshot.	Marker error after export and delivery to the recipient.
Operations and maintenance	Send an ordered inspection sequence for a non-sensitive dashboard.	Recovery and false-positive rates through authorized internal channels.
Archiving and accessibility	Retain annotation order with an image and extract it as text.	Extraction rate, reader compatibility, and tests with users.

Table 4. Application hypotheses and proposed field evaluation criteria.

The experiments use axis-aligned crops, retained targets, and local re-encoding. The negative control removes the bottom 88 rows, including the entire strip. It also removes some bottom screenshot pixels when the strip is shorter. All four synthetic screenshots fail at 35% scale. Failures also occur at 40% scale and JPEG quality 30. Rotation, perspective, and production-platform processing are outside the evidence. When the strip is removed, GFS2 cannot recover the payload. The visible code can be read by third parties. A CRC provides neither confidentiality nor integrity against deliberate modification.

The tested bilinear option is exposed by the codec and exercised by the benchmark. The released web app at the reference revision uses ordinary readout. The 240/240 result therefore describes the experimental two-pass reader. Future experiments can compare parity and finder alternatives under the same protocol and measure real delivery channels with independent observers.

7. Reproduction and availability

The public repository contains the code, standalone demo, and benchmark scripts. Run the following from the repository root. Recorded versions are Python 3.12, Node.js 24, Pillow 12.3.0, NumPy 2.5.2, ZXing-C++ 3.1.1, and OpenCV Python headless 5.0.0.93 for ScreenSpot-Pro.

```
git clone https://github.com/zakariagharzouli/project-ghostframe.git
cd project-ghostframe
git checkout f25a018e8ba121e18e19667c63d723e7599844be
node --test web/replay/codec.test.cjs
python -m pip install -r web/replay/requirements-holdout.txt
python web/replay/benchmark.py --resample lanczos --out run/replay/benchmark-equal-area-lanczos.json
python web/replay/benchmark.py --resample bicubic --out run/replay/benchmark-equal-area-bicubic.json
python web/replay/holdout.py --cache run/replay/screenspot-cache --out run/replay/holdout.json
python web/replay/holdout.py --followup-from run/replay/holdout.json --out run/replay/holdout-bilinear.json
```

The preprint archive includes paired logs, source file hashes, the archived selection protocol, and `audit.py`. The script recalculates counts, intervals, and negative controls from individual cases. Rerunning the four benchmarks during preparation produced four byte-identical JSON logs with the same software environment and source files. This is a rerun by the project team. The full trials use ScreenSpot-Pro images from the revision in [6]. The project code is MIT-licensed. The manuscript is licensed CC BY 4.0.

References

[1] ITU-T. T.81, Digital compression and coding of continuous-tone still images. 1992. Recommendation T.81.

[2] W3C. Portable Network Graphics (PNG) Specification, Third Edition. Recommendation of June 24, 2025. PNG specification.

[3] DENSO WAVE. QR Code Standardization. QR Code specifications overview.

[4] DENSO WAVE. rMQR Code. Official rMQR overview.

[5] Li, K., Meng, Z., Lin, H., Luo, Z., Tian, Y., Ma, J., Huang, Z., and Chua, T.-S. ScreenSpot-Pro: GUI Grounding for Professional High-Resolution Computer Use. 2025. arXiv:2504.07981.

[6] Li et al. ScreenSpot-Pro. Revision 210e78d3844251110bff86c95835ebd37a6930fa. Pinned dataset.

[7] OpenCV. Geometric Image Transformations. Resampling documentation.

[8] Zint. Barcode manual, Data Matrix section. DMRE dimensions and variants.

[9] ZXing-C++. Library and Python interface. Python wrapper.

[10] Gharzouli, Z. Project Ghostframe. Reference code and scripts, revision f25a018e8ba121e18e19667c63d723e7599844be. Repository at the reference revision.

Declarations

Author: Zakaria Gharzouli.

Author-declared contributions: design, calculations, and writing assisted by AI tools, with reproduction scripts provided. AI assistance contributed to formalization, code, calculations, and writing. It does not constitute independent review. This manuscript is a preprint and has not been peer reviewed.

The author declares no funding and no conflicts of interest. License: CC BY 4.0. License terms.