

A Fast Finite-Flow Meyer–Bregman Decomposition

Why the scalar hard jump rings, and how to jump the fused state instead

Joshuah Rainstar

joshuah.rainstar@gmail.com

August 2026 revision; historical scope through August 21

Abstract

We study a fixed-cost acceleration of Meyer cartoon–texture decomposition. The quality reference is a 64-pass, warm-interleaved, optimized C++ Meyer–Bregman recurrence. An earlier hard jump was much faster, but it left oscillatory material in the cartoon and generated signed contour halos. We show that these are not separate tuning defects. If H^K is the finite scalar contraction used by that jump and $P_K = I - H^K$, its exact texture error relative to the fused reference is

$$v_H - v_{64} = P_K(u_{64} - s_1) - H^K v_{64} + q_\mu.$$

The first term is a zero-mean structural halo; the second is precisely the uncompleted carrier tail. The numerical residual of this identity is below 1.5×10^{-13} on six surgical controls.

The repair jumps the actual six-field Bregman state $z = (u, w, t_u^x, t_u^y, t_w^x, t_w^y)$. From the exact residual $r = T_f(z) - z$ and a semismooth derivative $A \in \partial T_f(z)$, a depth-two Krylov projection evaluates the finite polynomial $p_m(A)r = (I + A + \dots + A^{m-1})r$. Short ordinary passes refresh the nonlinear disk-projection chart between jumps. No edge detector, texture classifier, source-conditioned threshold, training set, or content law is used. Fixed 19- and 29-pass-cost schedules are closer to fused pass 64 than ordinary fused pass 19 and 29, respectively, on every surgical control and on Barbara, Cameraman, and an analytic crossing. On an 8-thread M4 Mini, the two schedules take 11.52 and 18.16 ms at 256^2 , versus 34.60 ms for fused 64, yielding 3.00x and 1.91x speedups. The public C++ implementation, Python bindings, demos, exact proof arrays, sample comparisons, and complete timing records accompany the paper. **Keywords:** Meyer G -norm, cartoon–texture decomposition, Split Bregman, semismooth flow, Krylov polynomial, spectral fusion.

1 Introduction

Cartoon–texture decomposition asks for $f = \bar{u} + v$, where coherent geometry belongs to the effective cartoon $\bar{u}$ and oscillation belongs to texture v . This is not a frequency split: sharp interfaces can be cartoon, and a slowly varying carrier can be texture. Meyer encoded oscillation as the divergence of a bounded vector field [1]. Aujol et al. developed a BV– G alternating projection method [2]; Gilles and Osher gave a Split-Bregman realization based on two ROF subproblems [5].

Our route began as a speed problem. Transform planning and reuse first made a fast STFT super-resolution experiment possible. A subsequent two-lattice experiment replaced scalar interpolation by transport between related measurements. The work then moved to Meyer decomposition: nested ROF solves were reduced to a warm, fused recurrence; its state was reduced; divergences and spectra were maintained; the lower-triangular spectral pair was fused; and one-axis FACR supplied the arbitrary-shape path. Figure 1 records this path without importing later projects or terminology.

Once the optimized 64-pass result was visually accepted as the decomposition reference, a second question emerged: can the useful trajectory be compiled into less work? The first hard jump appeared attractive because it was nearly one-shot. Synthetic controls made its failure unmistakable: edge halos and texture retained in cartoon. Attempts to repair the output us-

ing content-dependent admission rules were rejected on principle. The present paper instead (i) proves the common source of both defects, (ii) derives a finite jump on the complete fused state, and (iii) validates fixed schedules against both equal-cost ordinary iteration and the 64-pass reference.

The contribution is not a claim that 29 operator passes equal 64 nonlinear passes. It is a measured Pareto result with an explicit error decomposition: the finite-flow method removes the two structural errors of the scalar jump, then exposes what remains as Krylov projection, semismooth chart, and finite tail error.

2 Fused reference recurrence

Let $X = \mathbb{R}^{H \times W}$ with periodic forward gradient $D : X \rightarrow X^2$ and D^* its adjoint. The discrete isotropic variation is

$$J(u) = \text{TV}(u) = \sum_x \|(Du)(x)\|_2. \quad (1)$$

Meyer’s discrete oscillation norm is

$$\|v\|_G = \inf_{v=D^*p} \|p\|_{\infty,2}, \quad \|p\|_{\infty,2} = \max_x \|p(x)\|_2. \quad (2)$$

The associated BV– G formulations and alternating projections are developed in [1, 2, 6]. We use the same [0, 255] intensity convention and fixed parameters $\lambda = .05$, $\mu = 40$.

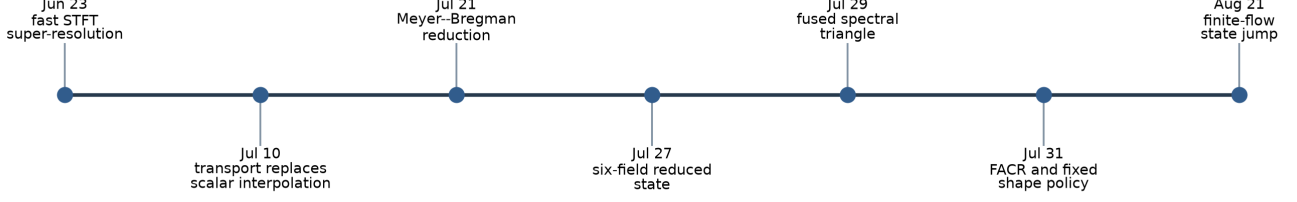


Figure 1: Developmental path used in this paper. The endpoint is the finite flow of the Meyer state; later and unrelated work is outside scope.

The optimized reference retains two primal fields and four accumulated gradient fields:

$$z = (u, w, t_u^x, t_u^y, t_w^x, t_w^y). \quad (3)$$

Here $w = f - u - v$ is the finite-pass survivor. Consequently the public two-product decomposition is

$$v = f - u - w, \quad \bar{u} = f - v = u + w. \quad (4)$$

This convention matters in comparisons: raw u from a finite model is not the effective cartoon $f - v$.

Define the pointwise disk projection and reflected projection

$$\Pi_a(t) = \begin{cases} t, & |t| \leq a, \\ a, t/|t|, & |t| > a, \end{cases} \quad R_a = I - 2\Pi_a, \quad (5)$$

and the screened inverse $S(c, \eta) = (cI + \eta D^* D)^{-1}$. With

$$\begin{aligned} c_u &= \lambda, & \eta_u &= 2\lambda, & a_u &= \eta_u^{-1}, \\ c_w &= \mu^{-1}, & \eta_w &= 10\mu^{-1}, & a_w &= \eta_w^{-1}, \end{aligned} \quad (6)$$

one exact warm fused pass $T_f : z \mapsto z^+$ is

$$u^+ = S(c_u, \eta_u) \{c_u(u + w) - \eta_u D^* R_{a_u}(t_u)\}, \quad (7)$$

$$w^+ = S(c_w, \eta_w) \{c_w(f - u^+) - \eta_w D^* R_{a_w}(t_w)\}, \quad (8)$$

$$t_u^+ = D u^+ + \Pi_{a_u}(t_u), \quad t_w^+ = D w^+ + \Pi_{a_w}(t_w). \quad (9)$$

Proposition 2.1 (Lower-triangular spectral pass). *For periodic boundaries, equations (7)–(8) form a lower-triangular two-by-two system at every Fourier coefficient. Both reflected divergences may be transformed before either inverse transform, and the two primal spectra may be solved in one range traversal.*

Proof. $D^* D$ diagonalizes under the periodic DFT. If σ_u and σ_w are the two screened multipliers and $\hat{d}_u, \hat{d}_w$ the reflected divergences, then

$$\begin{aligned} \hat{u}^+ &= \sigma_u \{c_u(\hat{u} + \hat{w}) - \eta_u \hat{d}_u\}, \\ \hat{w}^+ &= \sigma_w \{c_w(\hat{f} - \hat{u}^+) - \eta_w \hat{d}_w\}. \end{aligned}$$

The second equation depends on the newly computed first variable and no reverse coupling remains. This is exactly the paired spectral loop in the C++ kernel. $\square$

The first pass starts from zero state and has a still simpler source-only triangle. Later passes reuse primal spectra and four reflected-dual fields. This warm interleaving differs computationally from cold nested ROF solves, but it retains the same Meyer–Bregman geometry and provides the high-quality finite trajectory studied here.

3 Exact failure mechanism of the hard jump

The rejected hard construction replaced the state recurrence by two scalar observations. Let $\mathcal{S}$ be its scalar separator, H its contraction, $P_K = I - H^K$, and

$$\begin{aligned} s_0 &= \mathcal{S}(f), & m_0 &= P_K(f - s_0), \\ r_1 &= f - m_0, & s_1 &= \mathcal{S}(r_1). \end{aligned} \quad (10)$$

Before its final G -capacity correction, the texture proposal is $m = P_K(f - s_1)$. Write $v_H = m + q_\mu$, where q_μ is the change caused by that final capacity route.

Theorem 3.1 (Common cause of halo and retained texture). *Let (u_*, v_*) be any exact two-product reference, $f = u_* + v_*$. The second hard observation and its final texture error satisfy*

$$r_1 = u_* + H^K v_* + P_K(s_0 - u_*), \quad (11)$$

$$v_H - v_* = P_K(u_* - s_1) - H^K v_* + q_\mu. \quad (12)$$

Proof. Since $P_K = I - H^K$,

$$\begin{aligned} r_1 &= f - P_K(f - s_0) = s_0 + H^K(f - s_0) \\ &= u_* + H^K v_* + P_K(s_0 - u_*), \end{aligned}$$

which proves (11). Similarly,

$$\begin{aligned} m - v_* &= P_K(f - s_1) - v_* \\ &= P_K(u_* - s_1) + (P_K - I)v_* \\ &= P_K(u_* - s_1) - H^K v_*. \end{aligned}$$

Adding $q_\mu = v_H - m$ proves (12). $\square$

The two visible failures are therefore one representation error. $H^K v_*$ is the uncompleted carrier tail; it remains in cartoon with the opposite sign in texture. If $e = u_* - s_1$ contains a

step mismatch and $H^K e = h_K * e$ for a normalized smoothing kernel h_K , then $P_K e = e - h_K * e$ has the paired positive and negative contour response of a halo. It has zero mean, so a one-sided clamp cannot remove it faithfully. For a Fourier carrier with contraction gain $0 < \rho < 1$, $H^K v_* = \rho^K v_* \neq 0$ at every finite K .

Figure 2 renders the terms independently. Across six 256^2 controls, direct substitution into (11) and (12) leaves at most 1.42×10^{-13} in L_∞ . This is an identity check, not a fitted explanation.

This theorem also rules out the discarded content gate as a fundamental repair. A classifier applied after m is formed can hide samples of the error, but it cannot restore the omitted six-field trajectory and must encode assumptions about image content. We instead change the representation being jumped.

4 Finite flow of the complete state

After a short ordinary prefix, fix the current state z and define

$$r = T_f(z) - z, \quad A \in \partial T_f(z). \quad (13)$$

If T_f were affine on the visited chart, its exact finite displacement after m more passes would be

$$T_f^m(z) - z = p_m(A)r, \quad p_m(A) = I + A + \dots + A^{m-1}. \quad (14)$$

This is the correct target. The stationary Newton displacement $(I - A)^{-1}r$ was tested and rejected: it can overshoot a desired finite state at structural transitions.

The only nonlinear branch derivative comes from the disk projection already present in (5). For $n = t/|t|$,

$$D\Pi_a(t)h = \begin{cases} h, & |t| < a, \\ \frac{a}{|t|}(I - nn^T)h, & |t| > a, \end{cases} \quad (15)$$

and $DR_a(t)h = h - 2D\Pi_a(t)h$. At $|t| = a$ the implementation chooses the interior member of the Clarke derivative. Differentiating the fused triangle gives

$$\begin{aligned} \dot{u}^+ &= S_u\{c_u(\dot{u} + \dot{w}) - \eta_u D^* DR_{a_u}(t_u)\dot{t}_u\}, \\ \dot{w}^+ &= S_w\{-c_w\dot{u}^+ - \eta_w D^* DR_{a_w}(t_w)\dot{t}_w\}, \\ \dot{t}_u^+ &= D\dot{u}^+ + D\Pi_{a_u}(t_u)\dot{t}_u, \\ \dot{t}_w^+ &= D\dot{w}^+ + D\Pi_{a_w}(t_w)\dot{t}_w. \end{aligned} \quad (16)$$

No statistic of f enters (16); only the current optimization state and the original projection geometry do.

4.1 Depth-two nonorthogonal Krylov form

The implementation evaluates r , Ar , and A^2r . Let

$$\begin{aligned} B &= [r, Ar], & G &= B^T B, \\ K &= B^T A B, & \hat{A} &= G^{-1} K. \end{aligned} \quad (17)$$

Because Ar is the second basis vector, the first column of $\hat{A}$ is exactly $(0, 1)^T$. Only the projection of A^2r needs a two-by-two solve. The applied displacement is

$$\delta_m = B p_m(\hat{A}) e_1. \quad (18)$$

This is algebraically the depth-two Arnoldi polynomial without normalization or full-field modified Gram–Schmidt sweeps. A fixed 64-chunk reduction computes the five required inner products in a thread-count-independent summation order.

4.2 What error remains

Write the semismooth expansion at z as

$$T_f(z + d) = T_f(z) + Ad + N_z(d). \quad (19)$$

Let $d_k = T_f^k(z) - z$, $p_0 = 0$, $p_{k+1} = r + A p_k$, and $e_k = d_k - p_k$. Then

$$e_{k+1} = A e_k + N_z(d_k). \quad (20)$$

If $\|A\| \leq \gamma$ and $\|N_z(d)\| \leq L \|d\|^2/2$ on the traversed chart, induction gives

$$\|e_m\| \leq \frac{L}{2} \sum_{k=0}^{m-1} \gamma^{m-1-k} \|d_k\|^2. \quad (21)$$

The implemented error additionally contains the depth-two projection term

$$\|p_m(A)r - B p_m(\hat{A})e_1\|. \quad (22)$$

Ordinary passes after each jump refresh the nonlinear disk branches and shorten the chart over which (21) accumulates. Thus the remaining discrepancy from pass 64 consists of finite target tail, Krylov projection, and semismooth chart error. The terms $P_K(u_* - s_1)$ and $-H^K v_*$ of (12) are structurally absent: this method never forms s_1 , P_K , or a scalar surrogate for z .

4.3 Fixed schedules and the shorter-jump ablation

For prefix p , two tangent products, s ordinary refresh passes, and j jumps, the exact operator-pass cost is

$$C = p + j(1 + 2 + s), \quad (23)$$

where the one counts the exact nonlinear residual evaluation. The public schedules are fixed independently of the source:

	p	horizon	s	j	C
fast	4	18	2	3	19
quality	4	10	2	5	29

We also tested the natural alternative “jump less, refresh once.” A cheap post-jump graph retraction,

$$t \leftarrow Du + \Pi_a(t - Du), \quad (24)$$

restores the dual feasibility $|t - Du| \leq a$ in $O(HW)$ point-wise work and no FFT. Retraction alone prevents instability but loses most of the acceleration. Shorter jumps with one exact refresh won at 128^2 but did not uniformly beat the two-refresh schedules at 256^2 . Reducing only the horizon traded small wins among controls rather than producing dominance. This negative ablation shows that refreshes are nonlinear chart updates, not backtracking overhead. A useful constrained variant is therefore likely to cost about the same total spectral work unless it exploits a stronger bound than (24).

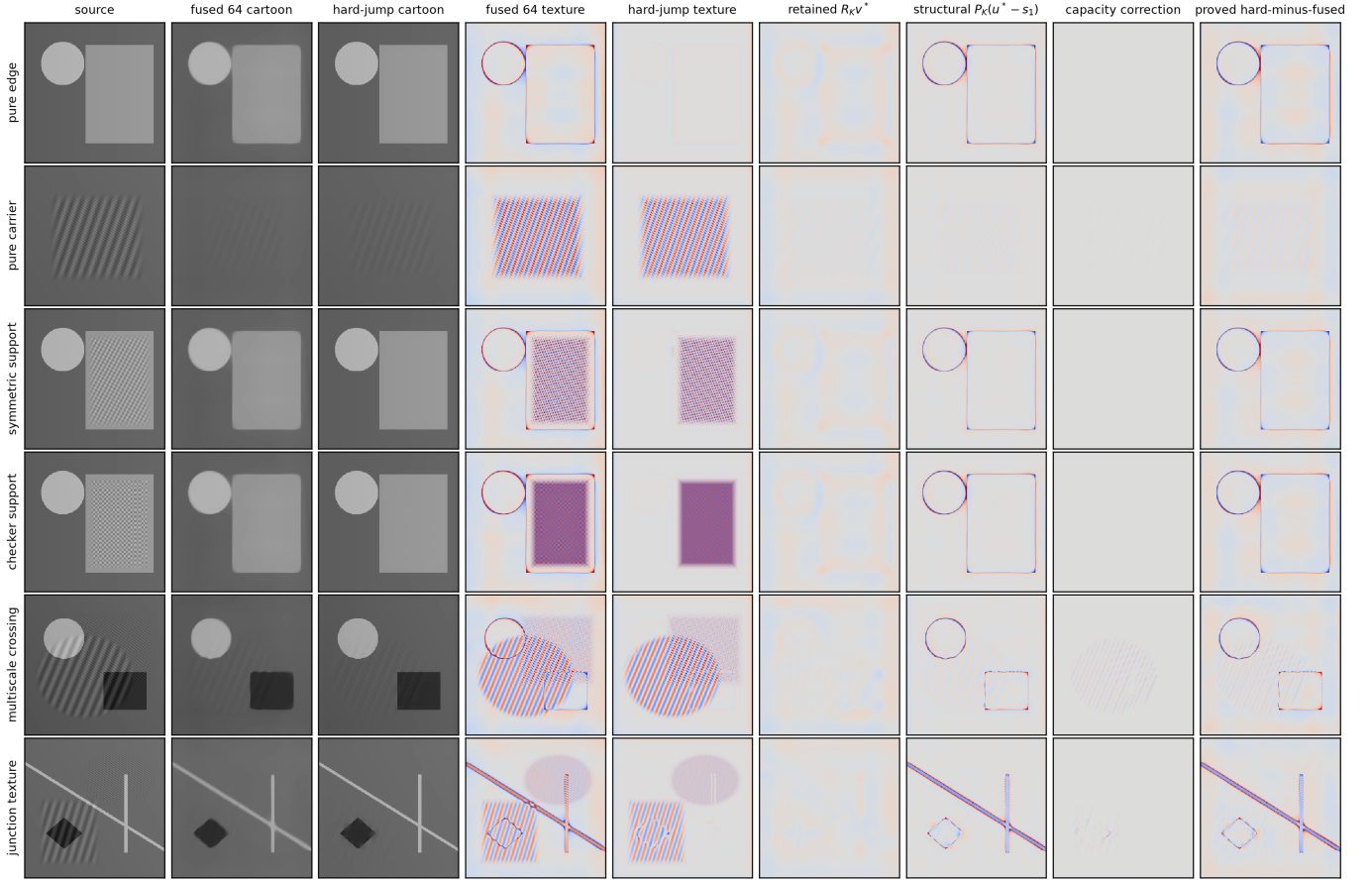


Figure 2: Surgical decomposition of the old hard-jump error. The last column is the sum of the retained-carrier, structural-halo, and capacity-correction columns and agrees with hard texture minus fused-64 texture to roundoff. Pure edge and pure carrier isolate the two terms; compound controls show their superposition.

5 C++ realization

The public full-spectral entry point is `bfft_meyer_split_flow_jump`. Its plan owns the ordinary reduced state and lazily allocates three six-field Krylov vectors plus primal spectra. The implementation performs:

1. a source FFT and ordinary fused prefix;
2. one exact residual pass without overwriting live spectra;
3. two streamed tangent passes using (15);
4. one fixed-chunk five-scalar Gram reduction;
5. the 2×2 polynomial in registers and one state AXPY;
6. fixed ordinary refresh passes, repeated by schedule.

No candidate scan or line search is performed. Texture is emitted as $f - u - w$ and cartoon as $f - v$, guaranteeing exact two-product recomposition.

The Python binding exposes `MeyerPlan.split_flow_jump`; the default full-spectral `split` uses the quality schedule. The effect API and still-image demo expose `quality`, `fast`, `fused64`, and the old hard control. The video demo uses the fixed fast schedule. FACR plans retain their configured ordinary fused recurrence because the current finite-flow tangent is full-spectral; they also fold the finite survivor into effective cartoon.

Native tests compare the C++ output to an independent NumPy implementation of (7)–(18), check exact recomposition, translation equivariance, and bit identity across thread counts, and require both public schedules to beat their equal-cost ordinary controls on the surgical battery.

6 Evidence

6.1 Protocol

All methods use $\lambda = .05$, $\mu = 40$ and are applied unchanged to every scene. Fused 64 is the accepted finite-trajectory reference. The error reported below is

$$E_{64}(v) = \|v - v_{64}\|_2 / \|v_{64}\|_2. \quad (25)$$

For the cold nested Gilles record, its model survivor $f - u - v$ is folded into effective cartoon $f - v$ before display, so every column is an exact two-product pair. Native timings are medians after three warmups and eleven repeats on an 8-thread Apple M4 Mini. Plan construction is excluded. The cold Gilles values are archived NumPy/SciPy wall times, capped at 120 outer iterations; they are historical context, not a same-implementation speed claim.

6.2 Surgical controls

At 256^2 , the old hard jump has mean texture RMS error 3.431 to fused 64 over pure edge, pure carrier, symmetric support, checker support, multiscale crossing, and junction texture. The fast flow lowers that to 1.400, versus 1.496 for ordinary fused 19. The quality flow lowers it to 1.039, versus 1.196 for ordinary fused 29. The flow/equal-cost ratios lie in $[.919, .951]$ for fast and $[.821, .927]$ for quality; every scene improves. Figure 3 shows that the large signed contour pattern and carrier deficit

of the old hard output are replaced by the much smaller ordinary finite-convergence discrepancy.

6.3 Natural and historical comparisons

Figure 4 compares effective cartoons and textures. The hard control visibly retains Barbara’s cloth and Cameraman’s ground texture in cartoon, while its synthetic contours ring. Both finite-flow outputs track the fused-64 allocation. Table 1 reports the corresponding times and errors. At equal operator cost, finite flow is closer than ordinary fused iteration for all three sources and both schedules.

The flow schedules cost more wall time than their equal-pass ordinary controls because tangent projection and deterministic Gram reduction have higher constants than an ordinary fused pass. They nevertheless move farther along the quality frontier. At 256^2 , quality reaches $E_{64} = .092$ on Cameraman in 18.14 ms, whereas ordinary 29 reaches .127 in 15.79 ms and fused 64 reaches zero by definition in 34.86 ms. The choice is an explicit Pareto trade, not an assertion that pass-equivalent cost equals elapsed time.

6.4 Scaling

Table 2 and Figure 6 use a dependency-free multiscale crossing at three sizes. The quality schedule is 2.01x, 1.91x, and 1.64x faster than fused 64; the fast schedule is 3.05x, 3.00x, and 2.55x faster. Absolute 512-square times are 33.92 ms (fast), 52.63 ms (quality), and 86.58 ms (fused 64).

7 Scope and limitations

Fused pass 64 is a finite, visually accepted reference, not a theorem of ground-truth uniqueness. The finite-flow schedules approximate its trajectory; they do not solve every BV- G formulation or equal the cold nested Gilles state. The evidence covers one parameter pair, periodic full-spectral power-of-two shapes, six designed controls, and three comparison images. FACR and Neumann paths remain ordinary fused methods.

The depth-two derivative freezes one semismooth chart. Projection-boundary crossings generate the remainder in (21); refresh passes limit but do not eliminate it. Larger Krylov spaces, certified chart radii, or a useful low-cost retraction could improve the frontier, but the shorter-jump ablation warns against selecting a schedule from one resolution. The old hard control remains public only for reproduction and negative tests.

8 Reproduction

The proof is generated by `experiments/meyer_hard_jump_defect_proof.py`; the independent finite-flow oracle and schedule battery are in `experiments/meyer_semismooth_state_jump.py`. Native scaling is serialized by `experiments/benchmark_meyer_flow_jump.py`. The paper directory contains the archived Gilles arrays, the current natural benchmark, all timing samples, and a renderer that reads records rather than recomputing claims.

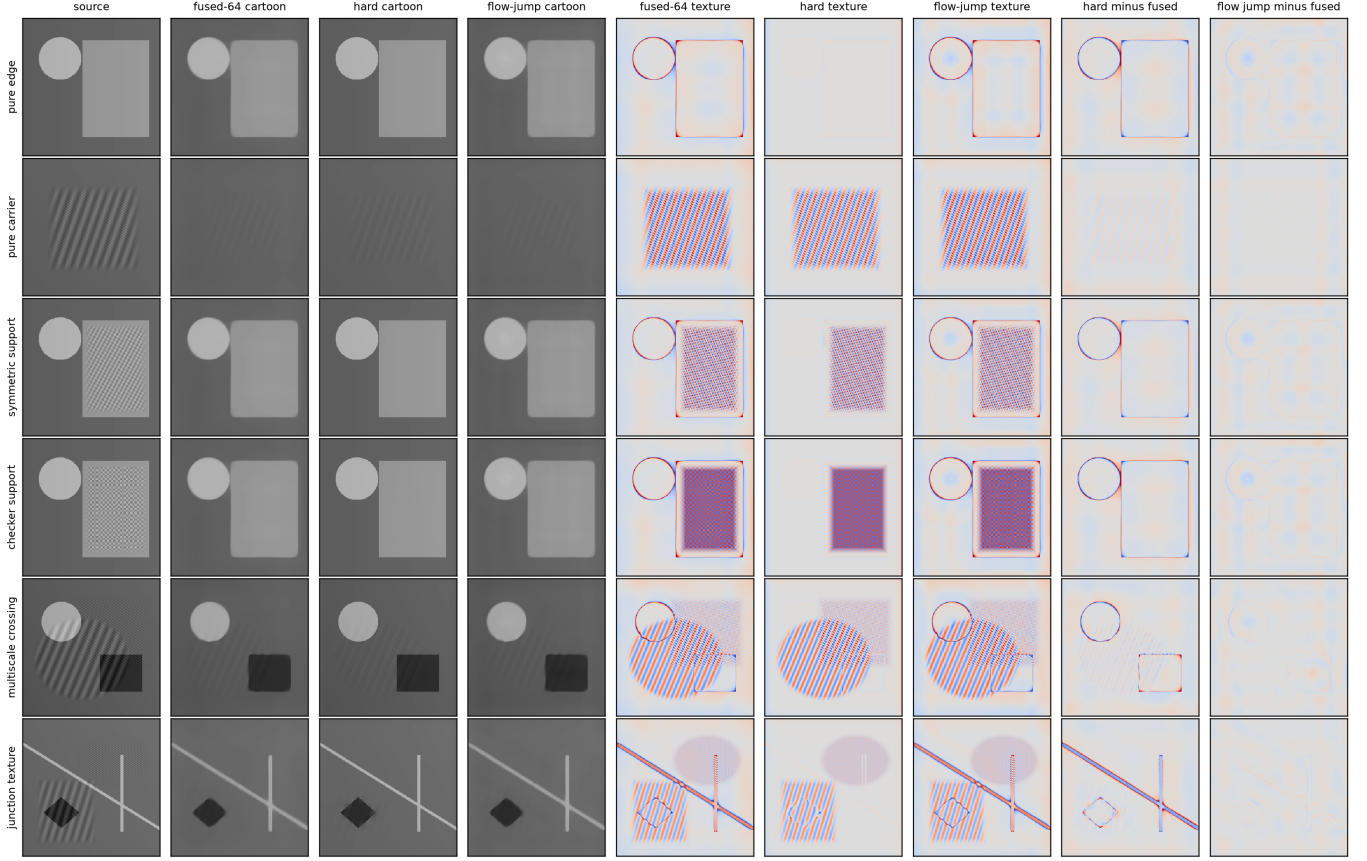


Figure 3: Fused-64 reference, old hard jump, and 29-cost finite-flow quality schedule on six surgical controls. The final two columns use the same diverging scale within each row. The flow discrepancy is faint and follows ordinary finite convergence; the hard discrepancy reproduces the two proven terms of Figure 2.

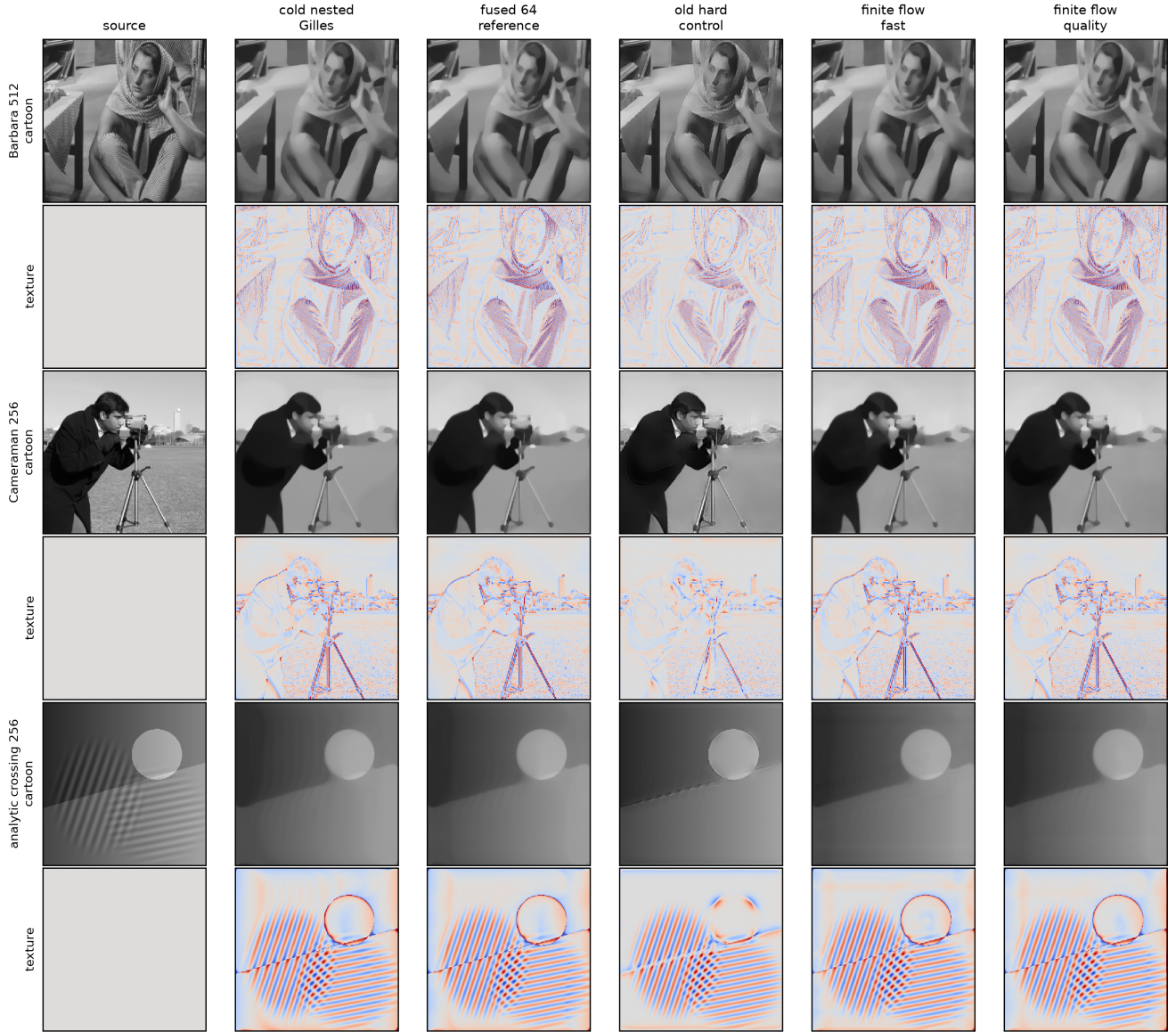


Figure 4: Exact two-product comparison. Each method is displayed as effective cartoon $f - v$ and texture v . The cold nested Gilles output is retained as historical context; fused pass 64 is the trajectory reference. Texture rows share a symmetric scale within each source.

Table 1: M4 native median time in ms and texture error E_{64} . Gilles times (marked †) are archived cold NumPy/SciPy runs and are not native C++ timings.

method	Barbara 512 ²		Cameraman 256 ²		analytic 256 ²	
	ms	E_{64}	ms	E_{64}	ms	E_{64}
old hard control	7.418	.5235	2.821	.6923	2.852	.4200
ordinary fused 19	25.583	.1080	10.347	.1863	10.182	.1889
finite flow fast	33.933	.0923	11.959	.1582	11.869	.1689
ordinary fused 29	39.545	.0710	15.791	.1273	15.849	.1385
finite flow quality	53.083	.0513	18.144	.0917	18.301	.1111
ordinary fused 64	86.988	0	34.864	0	34.732	0
cold nested Gilles†	169582.8	.0744	46167.9	.2095	45659.2	.1584

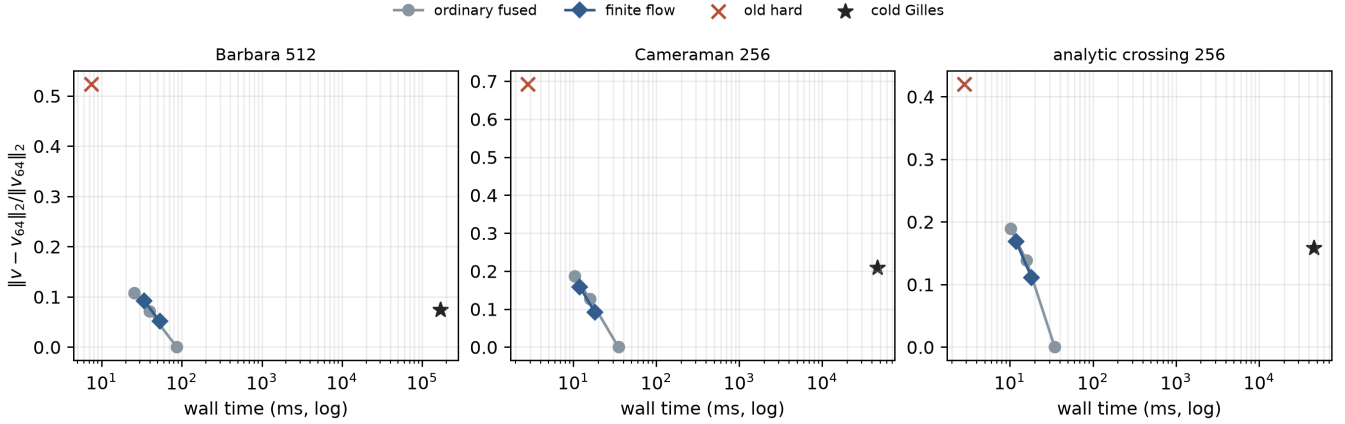


Figure 5: Time–quality plane. Blue diamonds are the two proposed fixed schedules; gray circles are ordinary fused 19, 29, and 64. The hard jump is cheap but far from the accepted decomposition. Cold nested Gilles is shown only to preserve the historical comparison.

Table 2: M4 medians (ms), 8 threads.

size	fast	quality	fused 64	speedups
128	6.671	10.107	20.325	3.05/2.01
256	11.521	18.155	34.598	3.00/1.91
512	33.920	52.633	86.581	2.55/1.64

The ABI, implementation, binding, demo paths, and invariance tests are in the main source tree.

9 Conclusion

The old hard jump failed for a precise reason: it compressed a six-field nonlinear trajectory into two finite scalar observations. Its halo and texture leakage are the two exact terms of that replacement. Once the jump is applied to the actual fused state, those terms disappear from the error algebra. A finite Krylov polynomial then advances the same Meyer–Bregman recurrence, while ordinary refreshes control the nonlinear chart remainder.

The result is a fixed, content-free, reproducible Pareto improvement. It does not obtain speed by deciding what an image contains. It obtains speed by representing more of the work that the accepted decomposition is already doing.

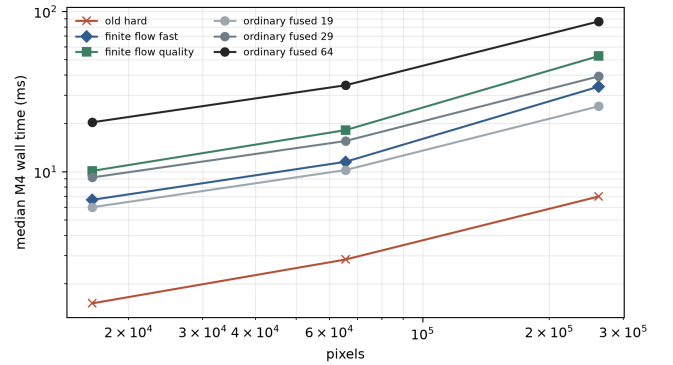


Figure 6: Native scaling, including equal-cost ordinary controls.

AI-assisted research disclosure

Joshuah Rainstar supplied the research direction, visual judgments, and successive falsification criteria. AI systems generated substantial code, algebraic exploration, benchmark scaffolding, and manuscript text under that direction. Claims are tied to executable artifacts and serialized records.

References

- [1] Y. Meyer, *Oscillating Patterns in Image Processing and in Some Nonlinear Evolution Equations*. AMS, 2001.

- [2] J.-F. Aujol, G. Aubert, L. Blanc-Féraud, and A. Chambolle, “Image decomposition into a bounded variation component and an oscillating component,” *J. Math. Imaging Vision*, 22:71–88, 2005.
- [3] A. Chambolle, “An algorithm for total variation minimization and applications,” *J. Math. Imaging Vision*, 20:89–97, 2004.
- [4] T. Goldstein and S. Osher, “The Split Bregman method for L^1 -regularized problems,” *SIAM J. Imaging Sci.*, 2:323–343, 2009.
- [5] J. Gilles and S. Osher, “Bregman implementation of Meyer’s G -norm for cartoon + textures decomposition,” arXiv:2410.22777, 2024.
- [6] J. Gilles and Y. Meyer, “Properties of BV– G decomposition models,” arXiv:2411.04456, 2024.