Adaptive Quality Mask Harvesting for Pixel-Wise Quality-Weighted Deep-Sky Reconstruction

Method definition, implementation contract, diagnostics, and an M31-A validation run

Jeamy Lee[1]

[1]Independent Researcher / tile_compile project

June 11, 2026

Abstract

Adaptive Quality Mask Harvesting (AQMH) is a deterministic pixel-wise quality-weighted reconstruction method for short-exposure deep-sky imaging. It replaces block-constant local quality scores with dense per-frame quality maps and reconstructs each supported output pixel by a weighted mean whose AQMH weight is evaluated at that same pixel. The method is independent of Classic Tile Compile: registration, photometric normalization, common-overlap handling, reporting, and run management may be shared, but AQMH does not use classic local tile metrics, state clustering, or synthetic frames as reconstruction inputs or fallbacks. This paper consolidates the AQMH v0.1.0 definition, states the implementation contract used by tile_compile, and evaluates a concrete M31-A run from 2026-06-11. The run processed 645 frames on a 3924×2310 canvas, wrote 645 dense luminance quality maps at half resolution, and reconstructed the image with 5.699×10^9 finite map samples, zero missing map samples, zero unsupported output pixels, and no fallback to classic tile weights. The measured median FWHM changed from 12.351 px to 5.271 px, corresponding to a 57.327% reduction. The background RMS changed from 4.185 to 0.714, corresponding to an 82.932% decrease. These measurements demonstrate a successful single-run execution of the AQMH path; they do not by themselves establish population-level performance across object classes, instruments, or atmospheric regimes.

Keywords: astronomical image reconstruction, deep-sky imaging, quality maps, pixel-wise weighted mean, short exposure stacking, deterministic reconstruction

1 Introduction

Short-exposure deep-sky imaging collects many noisy frames instead of a small number of long exposures. The data often contain useful signal even when individual frames suffer from local defects: cloud edges, satellite traces, registration remnants, small hot-pixel clusters, or spatially varying seeing. A single global frame score cannot describe these local variations. Coarse tile scores improve locality, but they still assign one scalar quality value to an extended region. This has two structural consequences. First, a small contaminated area can depress the weight of a much larger block. Second, neighboring blocks can receive different constant weights, creating a discontinuous weighting field that is unrelated to the optical image content.

AQMH is documented here as an independent reconstruction path within the tile_compile project. The reference method for Classic Tile Compile is *Tile-Based Quality Reconstruction for DSO*, Version 3.3.9.

AQMH addresses this by computing a dense quality field $Q^{map}_{f,c}(x, y)$ for each frame f and channel or stream c. The reconstruction then uses the frame sample itself and the AQMH weight field at the same pixel:

$$R_c(p) = \frac{\sum_f W_{f,c}^{aqmh}(p) I_{f,c}(p)}{\sum_f W_{f,c}^{aqmh}(p)}, \quad W_{f,c}^{aqmh}(p) \geq 0. \tag{1}$$

After the weights have been computed, the reconstruction core is therefore a deterministic linear weighted mean. AQMH does not hallucinate pixel values, does not synthesize replacement frames, and does not rely on tile clustering. Its purpose is narrower: retain useful local regions from every registered frame while suppressing locally degraded regions through a dense, reproducible quality field.

2 Method overview

2.1 Inputs and independence

AQMH reuses only shared preprocessing outputs:

- calibrated, registered, prewarped, and normalized frames;
- a canvas-valid/common-overlap mask;
- global frame diagnostics used for a frame-level weight;
- run management, artifacts, reports, and UI status.

AQMH is otherwise independent from Classic Tile Compile. In particular, the run studied here reports `classic_tile_weights_used=false` and `fallback_to_classic=false`. The classic phases `STATE_CLUSTERING` and `SYNTHETIC_FRAMES` are not part of AQMH reconstruction.

The distinction matters operationally. Shared preprocessing produces registered, normalized image samples and a valid canvas domain. AQMH then defines its own quality maps and reconstruction weights. Any classic tile grid that appears in reports is therefore a geometry or diagnostic artifact, not the AQMH weight lattice.

2.2 Binding invariants

The AQMH implementation follows five practical invariants:

1. No frame is discarded solely because of its AQMH quality map.
2. Canvas-invalid pixels are excluded from all map and reconstruction statistics.
3. Once weights are known, reconstruction is a linear weighted mean of finite samples.
4. Quality maps and diagnostics are deterministic for identical input data and parameters.
5. AQMH outputs weights, masks, and diagnostics only; it does not predict missing signal.

The M31-A run uses `cherry_pick_enabled=false`; therefore no optional pixel-level frame subset mode is active.

2.3 Quality-map model

For a normalized registered input frame $I_{f,c}(x, y)$, AQMH computes a multi-scale quality field. The specified v0.1.0 default pyramid uses downscale factors $D_s = 4^s$ with a local radius of four pixels in scale space. Scales that are too coarse for the image dimensions are omitted and the geometric-mean exponent is computed from the actually available scale count. At each retained scale s, three local signals are evaluated over valid canvas support:

- a sharpness signal based on local Laplacian-response variance;
- a local SNR signal based on background-centered positive signal over robust scale;
- an artifact anomaly gate based on high-pass outlier fraction.

The sharpness and SNR signals are normalized within a frame by robust z-scores and combined by a sigmoid:

$$\Psi_s(x, y) = \sigma(w_{sharp} z(\Phi_{sharp,s}) + w_{snr} z(\Phi_{snr,s})) \times \Phi_{artifact,s}(x, y). \tag{2}$$

Scale maps are upsampled with valid-support interpolation and fused by geometric mean:

$$Q^{map}_{f,c}(x,y) = \left(\prod_{s \in S_{actual}} \Psi^{up}_{s}(x,y) \right)^{1/|S_{actual}|}. \quad (3)$$

The geometric mean is intentionally conservative: one poor scale suppresses the fused quality at that location, and a zero at any scale remains an exact veto after fusion. Canvas-invalid pixels are set to zero only as an output convention after they have already been excluded from all statistics.

The z-score normalization also fixes the interpretation of Q^{map}. The sigmoid component is a within-frame relative quality field; it indicates which regions of a frame are locally better or worse than other regions of the same frame. Whole-frame discrimination is carried by the global term $G_{f,c}$ and by absolute gates such as the artifact fraction. A narrow frame-to-frame distribution of map means is therefore expected and is not, by itself, evidence that the maps are uninformative.

2.4 Reconstruction

The effective AQMH weight multiplies the frame-level quality term and the spatial quality map:

$$W^{aqmh}_{f,c}(x,y) = G_{f,c} Q^{map}_{f,c}(x,y). \quad (4)$$

For each output pixel, the implementation accumulates finite, canvas-valid samples only. Missing quality maps are not replaced by unweighted mean contributions. A zero quality map is an explicit veto for that frame/pixel contribution.

The studied M31-A run stores one luminance quality map per frame at half canvas resolution (`resolution_divisor=2`) and interpolates it for reconstruction. This is a storage and I/O choice, not a change to the mathematical definition of AQMH as a dense pixel-weighted reconstruction method.

Table 1: M31-A run geometry and frame counts.

Quantity	Value
Input frames loaded	645
Usable frames	645
Canvas size	3924×2310 px
Common-overlap fraction	0.975
Reconstruction fraction	0.982
Adaptive tiles in shared grid	209
Tile size / stride	296 px / 208 px

3 Implementation run

3.1 Data and configuration

The evaluation data are taken from the tile_compile run on the analysis workstation:

```
tile_compile_cpp/runs/M31-A_20260611_193628
```

The paper bundle itself is archived under `docs/AQMH/zenodo-0.1.0` in the project repository. Run artifacts used for the measurements are local analysis outputs located in:

```
tile_compile_cpp/runs/M31-A_20260611_193628/
artifacts
```

The final preview shown in Figure 4 is derived from `outputs/stacked_rgb_hms.fits`.

This paper treats the run as an implementation validation case. It verifies that the expected AQMH execution path is followed and reports the resulting diagnostics. It does not compare AQMH against a separately tuned Classic Tile Compile run on the same data, and it does not claim that the reported percentage changes generalize beyond this run.

3.2 Quality-map diagnostics

The run wrote one luma quality map per frame. The average map mean was 0.496 with standard deviation 0.002. The per-frame map mean ranged from 0.488 to 0.500. The average artifact fraction was 0.051, with a range from 0.049 to 0.060. All 645 diagnostics reported the `scene_dependent_snr` flag. In v0.1.0 this flag marks use of the degraded local-SNR path when

Table 2: AQMH map storage and reconstruction support.

Quantity	Value
AQMH enabled	true
Frames/maps	645 / 645
Stored map size	1962 × 1155 px
Resolution divisor	2
Map dtype	float32
Finite samples	5.699×10^9
Missing samples	0
Unsupported output pixels	0
Zero-veto pixels	0
Classic fallback	false

the background-centered support condition is not fully satisfied. It is deterministic and does not invalidate the reconstruction, but it is a limitation of this run because the local-SNR term may retain more scene-content dependence than the target AQMH definition intends.

3.3 Validation metrics

The validation artifact reports an AQMH method run with both FWHM and background checks passing. The median input seeing estimate was 12.351 px, while the median output FWHM was 5.271 px. This corresponds to a 57.327% reduction of the measured FWHM:

$$100\left(1 - \frac{5.271}{12.351}\right) = 57.327.$$

Background RMS changed from 4.185 to 0.714. The artifact stores this as a −82.932% change relative to the input value; equivalently, the output background RMS is 82.932% lower than the input estimate. These metrics should be interpreted as run-level diagnostics. They verify that the produced stack is sharper and less noisy under the selected measurement procedure, but they are not a substitute for photometric calibration or a controlled multi-run benchmark.

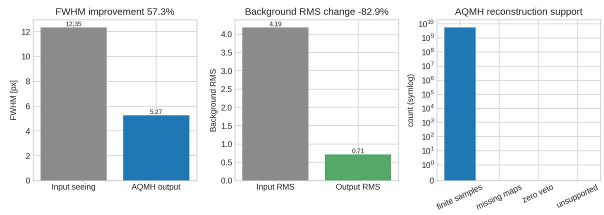

Figure 3: Validation summary for the M31-A AQMH run. FWHM and background RMS both improve, while the reconstruction support counters show no missing maps, unsupported pixels, or zero-veto pixels.

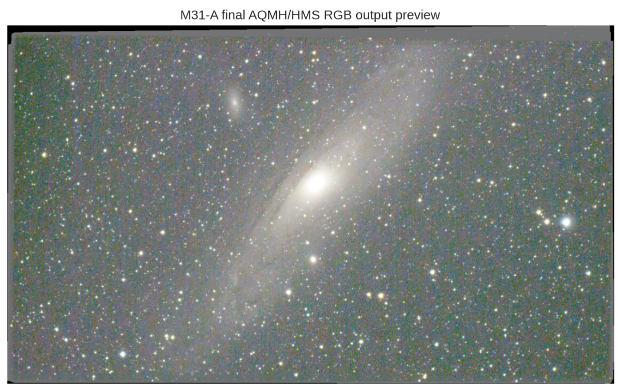

Figure 4: Final RGB preview after AQMH reconstruction, PCC, and HMS. The preview is a visual run product derived from the FITS output, not a quantitative measurement.

4 Discussion

4.1 What the run demonstrates

The M31-A run demonstrates the intended AQMH execution path:

- every usable frame produced a quality map;
- reconstruction consumed AQMH maps without classic tile-weight fallback;
- no quality map samples were missing during reconstruction;
- the output had full valid support under the run's reconstruction mask;
- the measured FWHM and background RMS validation metrics improved.

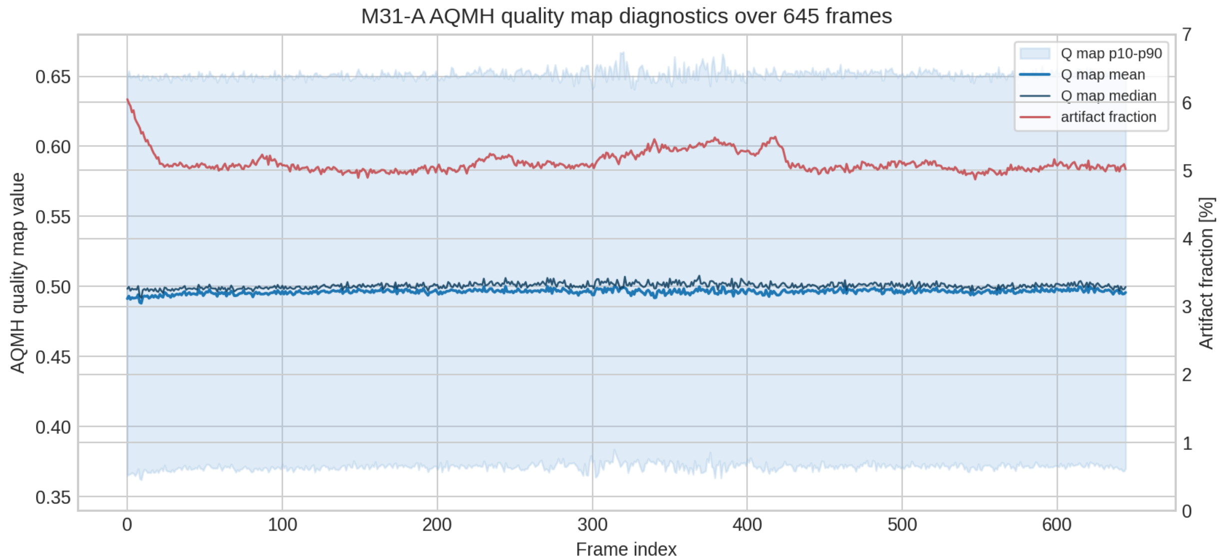

Figure 1: Per-frame AQMH quality-map diagnostics. The shaded band shows the p10-p90 interval of the map values; mean and median remain stable across the 645-frame sequence. The artifact fraction stays near five percent, with a higher early-frame tail.

This is important because AQMH is not a cosmetic replacement for the classic tile phases. It is a separate reconstruction core. The classic shared tile grid remains useful as an artifact and geometry diagnostic, but it is not the AQMH weighting model.

The support counters are particularly important for implementation validation. A successful AQMH-labeled run would be ambiguous if reconstruction silently fell back to missing-map behavior, unweighted means, or classic tile weights. The combination of 645 written maps, zero missing map samples, zero unsupported output pixels, and `fallback_to_classic=false` makes the measured output attributable to the AQMH path rather than to a mixed or compatibility path.

4.2 Interpretation of stable map means

The frame-to-frame mean of Q^{map} is deliberately stable. The sharpness and SNR terms are z-normalized inside each frame, so their sigmoid contribution is mostly a within-frame relative field. Cross-frame quality discrimination comes from the global frame weight $G_{f,c}$ and from absolute veto terms such as the artifact gate. Therefore, a narrow distribution of map means is not a defect; it is a consequence of separating within-frame spatial quality from between-frame global quality. A useful AQMH diagnostic should therefore inspect spatial structure, p10–p90 spread, artifact fraction, support counters, and reconstruction behavior rather than only the mean of each map.

4.3 Limitations

This paper reports one successful M31-A run and should not be read as a full statistical validation across object classes, sensors, exposure lengths, sky brightness levels, and seeing regimes. The diagnostic flag `scene_dependent_snr=true` appears for all frames and deserves further evaluation: the fallback is deterministic, but future work should quantify when background-centered local SNR support is sufficient without the degraded-path flag. The validation met-

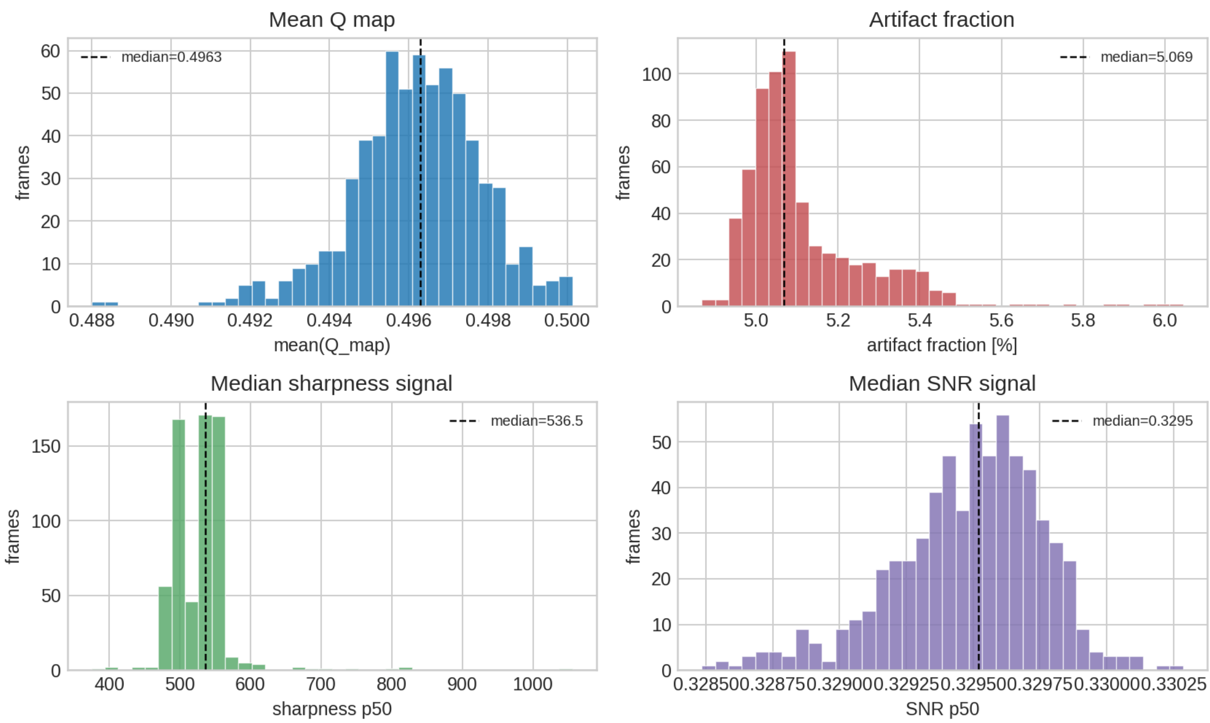

Figure 2: Distributions of AQMH per-frame diagnostic quantities. The narrow quality-map mean distribution reflects the within-frame relative normalization of the sigmoid factor; the artifact gate remains an absolute veto term.

rics are also pipeline diagnostics rather than an externally calibrated astronomical quality assessment. The final preview is not a scientific photometry product; it includes downstream PCC and HMS processing.

Further validation should include repeated runs, independent datasets, explicit comparisons against Classic Tile Compile and conventional stackers, sensitivity tests for pyramid parameters and map resolution, and visual/quantitative checks around strong gradients, saturated stars, nebular structure, and low-overlap frame borders.

5 Conclusion

AQMH provides a deterministic pixel-wise quality-weighted alternative to coarse local tile weighting for short-exposure deep-sky reconstruction. The M31-A run confirms that the implementation can execute the complete AQMH path on a 645-frame dataset, write all dense quality maps, reconstruct without classic fallback, and produce improved run-level validation metrics. The observed FWHM reduction of 57.327%, the background RMS reduction of 82.932%, the absence of missing map samples, and the stable quality-map diagnostics support the practical viability of AQMH as an independent reconstruction method in tile_compile.

Code and Data Availability

Reference method. Classic Tile Compile is described by *Tile-Based Quality Reconstruction for DSO*, Version 3.3.9. The corresponding repository source is the v3.3.9 paper source.

Code. The public project repository is github.com/jeamy/tile_compile. AQMH method details are documented in the AQMH method document.

This paper. The AQMH v0.1.0 paper bundle is archived in docs/AQMH/zenodo-0.1.0. The compact values used for the tables and plots are stored in m31_aqmh_run_summary.json.

Run data. The full source measurement artifacts are local outputs from the M31-A analysis run and are not part of the public repository bundle.