Dual-Initialization Dense Flow and Scatter-Blur-Decay Boundary Correction for Self-Calibrating Aerial Image Mosaicing

Greg Passmore

March 2026

Abstract

This paper presents a complete pipeline for constructing seamless mosaics from pairs of overlapping aerial photographs captured by uncalibrated consumer-grade drone cameras. The system integrates multi-scale normalized cross-correlation for coarse-to-fine tile registration, six-parameter affine Lucas-Kanade tracking for per-tile local deformation estimation, Brown-Conrady lens distortion fitting via grid-search-initialized weighted least squares, dense Lucas-Kanade optical flow on an 8-pixel grid with iterative dual-initialization refinement, hierarchical Catmull-Rom bicubic upsampling to per-pixel resolution, dynamic-programming seam finding with flow-magnitude-weighted cost, Laplacian pyramid multi-band blending, and a Gaussian-smoothed boundary correction field that eliminates Voronoi discontinuity artifacts. The pipeline introduces several novel methods and combinations: (1) a dual-initialization iterative refinement strategy for Lucas-Kanade flow that competes two independent starting points at each grid node to escape local minima without learned components; (2) error-weighted adaptive spatial smoothing in which Gaussian kernel weights are modulated by inverse photometric warp error rather than image content, so that only high-confidence neighbors influence corrections; (3) hierarchical Catmull-Rom bicubic flow upsampling through three successive doubling stages ($8 \to 4 \to 2 \to 1$), replacing bilinear interpolation to achieve C^1-continuous per-pixel flow fields that eliminate period-2 aliasing artifacts; (4) a flow-magnitude penalty term in the dynamic-programming seam cost that steers seams through regions of minimal geometric correction; (5) a scatter-blur-decay boundary correction field that resolves overlap-to-non-overlap color discontinuities without Poisson solves; and (6) self-calibrating Brown-Conrady distortion fitting from a single image pair using tiled NCC residuals as implicit calibration data, with grid-search initialization to avoid local-minimum traps. The complete pipeline is feature-free, requires no calibration targets, GPS metadata, or external libraries, and is implemented as a self-contained application. The following sections describe the mathematical formulation at each stage, document several approaches that were attempted and subsequently abandoned due to specific failure modes, and present the final process flow that produces artifact-free mosaics from DJI drone imagery of natural terrain. The paper concludes with directions for multi-image extension, wide spectral range matching, and integration into signals intelligence (SIGINT) workflows.

Keywords: aerial mosaicing, optical flow, Laplacian pyramid blending, lens distortion calibration, seam finding

1. Introduction

Aerial image mosaicing, the process of compositing multiple overlapping aerial photographs into a single spatially continuous image, is one of the oldest problems in photogrammetry. Aimé Laussedat conducted the first photogrammetric experiments in 1861, and Nadar captured the first aerial photograph from a balloon over Paris in 1858 [1]. By the 1920s, Sherman Fairchild had produced overlapping-photograph mosaics of Manhattan, and during World War II aerial mosaic maps became indispensable for military intelligence [1]. The advent of digital imagery and computational photogrammetry in the latter half of the 20th century transformed mosaicing from a manual optical bench operation into an algorithmic pipeline amenable to full automation [2].

Modern unmanned aerial systems (UAS) equipped with consumer-grade cameras (e.g., DJI platforms) produce high-resolution imagery at a fraction of the cost of traditional manned platforms. However, consumer lenses introduce

substantial radial and tangential distortion that is not characterized by factory calibration data, and the non-rigid parallax between adjacent exposures, exacerbated by terrain relief, wind-induced attitude variation, and rolling-shutter artifacts, demands registration strategies more flexible than global homography estimation.

This paper describes a complete mosaicing pipeline, implemented as a self-contained application, that addresses these challenges through a cascade of increasingly refined registration stages:

1. Global NCC-based translation estimation with sub-pixel parabolic refinement
2. Two-pass tiled NCC registration (coarse 256 px, fine 96 px) with per-tile 6-DOF affine Lucas-Kanade tracking
3. Robust outlier rejection using the Median Absolute Deviation (MAD)
4. Brown-Conrady lens distortion fitting via coarse-to-fine grid search
5. Dense Lucas-Kanade optical flow at 8-pixel spacing with iterative dual-initialization refinement
6. Hierarchical per-pixel flow upsampling ($8 \to 4 \to 2 \to 1$) using Catmull-Rom bicubic interpolation
7. Dynamic-programming seam finding with color and flow-magnitude cost
8. Laplacian pyramid multi-band blending
9. Gaussian-smoothed boundary correction for seamless overlap-to-non-overlap transitions

The system was tested on drone imagery of dense forest and tree canopy occasionally bisected by roads, using an uncalibrated lens. All computation proceeds at full image resolution, eliminating downsampling as a potential error source.

2. Application and Importance

Aerial image mosaicing serves numerous civilian and defense applications. In precision agriculture, mosaics of crop fields enable monitoring of vegetation health indices (NDVI) across spatial extents exceeding the field of view of any single exposure. In infrastructure inspection, mosaics of bridge decks, pipelines, and solar farms support automated

defect detection. In cartography and land survey, orthomosaics produced from drone imagery have become a low-cost alternative to traditional aerial survey with accuracies approaching sub-decimeter GSD (ground sample distance).

In the defense and intelligence domains, aerial mosaics are foundational to geospatial intelligence (GEOINT) production. The National Geospatial-Intelligence Agency (NGA) integrates imagery intelligence (IMINT) mosaics with signals intelligence (SIGINT), electronic intelligence (ELINT), and communications intelligence (COMINT) for battle damage assessment, target tracking, pattern-of-life analysis, and broad-area situational awareness [3]. The proliferation of small UAS platforms has made wide-area persistent surveillance accessible at the tactical edge, where real-time onboard mosaicing can provide operators with a continuously updating composite image of the area of interest.

A key challenge in all these applications is that the mosaicing pipeline must tolerate imperfect sensor calibration, non-planar terrain, varying illumination, and atmospheric effects. The system described in this paper addresses these factors through a multi-stage registration and blending pipeline that adaptively refines alignment at progressively finer spatial scales.

3. Historical Context of the Methods

3.1 Area-Based Matching and NCC

Area-based image matching via cross-correlation was formalized by Pratt [4] in 1974 for satellite image registration. Lewis [5] introduced fast normalized cross-correlation (NCC) in 1995, demonstrating that the normalization (which makes the metric invariant to linear brightness and contrast changes) can be computed efficiently using integral images. NCC remains the standard metric for template matching in photogrammetric pipelines where feature-based methods (SIFT [6], ORB) are unavailable or inappropriate.

3.2 Lucas-Kanade Optical Flow

Lucas and Kanade [7] introduced gradient-based iterative image registration in 1981, solving a local least-squares system (the 2×2 structure tensor) under the brightness constancy assumption. Bouguet [8] extended the method to a Gaussian image pyramid for handling large displacements. The 6-DOF affine extension of Lucas-Kanade, which models rotation, scale, and shear in addition to translation, is the basis for the per-tile local deformation estimation used here.

3.3 Laplacian Pyramid Blending

Burt and Adelson [9] introduced multi-resolution spline blending in 1983, decomposing images into Laplacian pyramids (band-pass components at multiple octaves), blending each band with a transition zone matched to its spatial frequency, and collapsing the result. This eliminates the ghosting artifacts inherent in linear alpha blending while preserving fine detail. With over 1,800 citations, the method remains the industry standard in panoramic stitching software.

3.4 Seam Finding

Efros and Freeman [10] introduced minimum-error boundary cuts via dynamic programming for texture quilting in 2001. Kwatra et al. [11] generalized the approach to graph-cut optimization for arbitrary-dimensional synthesis. In mosaicing, seam finding selects a connected path through the overlap region where the two images agree most closely, reducing the blending algorithm's burden of masking residual misalignment.

3.5 Lens Distortion Modeling

Conrady [12] first described the mathematics of tangential distortion from imperfect lens element alignment in 1919. Brown [13] extended this to a unified polynomial model combining radial distortion (coefficients k_1, k_2) and tangential/decentering distortion (p_1, p_2). The Brown-Conrady model, formalized in Brown (1971) [14] and later adopted by Tsai (1987) and Zhang (2000), is the basis of OpenCV's camera calibration and virtually all modern photogrammetric toolkits.

3.6 Spatial Interpolation

Shepard [15] introduced inverse distance weighting (IDW) in 1968 for interpolating irregularly-spaced geographic data. Catmull and Rom [16] defined a family of C^1-continuous cubic interpolating splines in 1974 that pass through all control points with automatically computed tangents. Both methods are employed in the present pipeline: IDW for mesh-warp blending of tiled registration vectors, and Catmull-Rom splines for the final stage of flow field upsampling.

4. Implementation: Approaches Attempted and Abandoned

The development of the pipeline proceeded through several iterations in which specific algorithmic choices were evaluated and, when found deficient, replaced. This section documents the principal failure modes encountered and the reasoning behind each design change.

4.1 Bilinear Upsampling at Step=1 (Abandoned)

Initial per-pixel flow upsampling used bilinear interpolation directly from the 8-pixel flow grid to individual pixels. This produced paired-delta artifacts: at every other pixel along each axis, the interpolated flow exhibited a period-2 oscillation (stripe pattern) arising from the even/odd aliasing inherent in bilinear sampling with integer-spaced output and non-centered source coordinates. The symptom was fine vertical and horizontal striping visible in the mosaic, particularly in gradient-blend mode.

Resolution: Replaced with hierarchical upsampling ($8 \to 4 \to 2 \to 1$) where each octave doubles the resolution using Catmull-Rom bicubic interpolation. At the final level ($\text{step} = 2 \to 1$), Catmull-Rom's C^1 continuity eliminates the grid-boundary kinks that caused the period-2 artifacts.

4.2 BFS Nearest-Neighbour Boundary Correction (Abandoned)

To correct color discontinuities at the boundary between the blended overlap zone and the single-image zone, the initial approach propagated the nearest boundary pixel's color correction via breadth-first search (BFS). Each non-overlap pixel received the correction vector from whichever boundary pixel the BFS reached first. This created Voronoi-cell tessellations in the correction field: where two boundary pixels with different correction vectors met, a sharp discontinuity appeared as a visible vertical or horizontal stripe in the mosaic.

Resolution: Replaced with a Gaussian-smoothed correction grid. Boundary corrections are scattered onto a coarse grid (8 px cells), separably Gaussian-blurred, then bilinearly sampled at each non-overlap pixel with an exponential distance-decay envelope. The BFS is retained solely for computing the distance field used in the decay function.

4.3 Lucas-Kanade Micro-Refinement at Step=1 (Abandoned)

An attempt was made to run additional Lucas-Kanade refinement iterations at the per-pixel level ($\text{step} = 1$) after hierarchical upsampling. At this fine scale, the optimizer frequently fell into local minima in the high-frequency texture of forest canopy, introducing noise that was worse than the interpolated flow. The refinement was removed; the per-pixel flow is now obtained purely by interpolation from the 8-pixel-grid Lucas-Kanade solution.

4.4 Simple Flow Smoothing Without Root-Cause Fix (Abandoned)

Before the boundary correction mechanism was understood, extensive Gaussian smoothing was applied to the flow field itself in an attempt to suppress vertical stripe artifacts. While this reduced stripe visibility, it also degraded alignment quality in the overlap region. The smoothing was ineffective because the stripes originated in the boundary correction field, not in the flow field. Once the Gaussian-smoothed correction grid was implemented, no additional flow smoothing was required.

5. Final Process Flow and Detailed Mathematics

This section presents the complete pipeline as implemented in version 5 of the system. Each stage is described with its mathematical formulation.

5.1 Global Translation Estimation

Both images are downsampled to a maximum dimension of 400 pixels. A coarse exhaustive search evaluates the NCC at every 5-pixel offset within a radius of half the smaller image dimension, followed by a fine search at every integer pixel in a ± 5-pixel neighborhood around the coarse optimum.

The NCC between the reference image R and target image T at integer offset (d_x, d_y) over their overlap Ω is:

$$\mathrm{NCC}(d_x, d_y) = \frac{\sum_{(x,y)\in\Omega} \hat{R}(x,y)\,\hat{T}(x,y)}{\sqrt{\sum_{(x,y)\in\Omega} \hat{R}^2(x,y) \cdot \sum_{(x,y)\in\Omega} \hat{T}^2(x,y)}}$$

where $\hat{R} = R(x,y) - \mu_R$ and $\hat{T} = T(x - d_x,\, y - d_y) - \mu_T$ are the zero-mean brightness values, and all sums are over the overlap region Ω. Evaluation uses a stride of 2 pixels for efficiency.

Sub-pixel refinement. A 1D parabolic fit is applied independently in x and y [17]. Given the NCC values at offsets $(d-1,\, d,\, d+1)$ denoted $c_-,\, c_0,\, c_+$, the sub-pixel correction is:

$$\delta = \frac{c_- - c_+}{2\,(c_- - 2c_0 + c_+)}$$

The correction is accepted only if $|\delta| < 1$ and the denominator exceeds 10^{-3}.

5.2 Two-Pass Tiled Registration

The overlap region is divided into a grid of tiles, first at coarse resolution (8×8 grid, 256-pixel half-width, pass 1) and then at fine resolution (12×12 grid, 96-pixel half-width, pass 2) targeting regions with high residual error. Each tile is matched by:

(a) Texture gate: The patch variance is computed from brightness samples at 4-pixel stride. Tiles with variance below 80 are rejected as low-texture.

(b) Coarse NCC search: Exhaustive search at 4-pixel stride within a ± 30-pixel radius.

(c) Fine NCC search: Integer-pixel refinement in a ± 4-pixel neighborhood around the coarse peak, evaluated at 2-pixel stride.

(d) Sub-pixel parabolic refinement: Identical to the global registration sub-pixel step.

(e) Affine Lucas-Kanade: Starting from the NCC translation, a 6-DOF affine model is iteratively refined using the Lucas-Kanade framework. The warp is parameterized as:

$$W(\mathbf{x};\,\mathbf{p}) = \begin{bmatrix} 1 + a_{00} & a_{01} \\ a_{10} & 1 + a_{11} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

where $\mathbf{p} = (a_{00},\, a_{01},\, a_{10},\, a_{11},\, t_x,\, t_y)$. The parameter update is obtained by solving the 6×6 normal equations derived from the Gauss-Newton approximation:

$$\mathbf{H}\,\Delta\mathbf{p} = \mathbf{J}^\top \mathbf{e}$$

where $\mathbf{H} = \mathbf{J}^\top \mathbf{J}$ is the Hessian approximation, $\mathbf{J}$ is the Jacobian of the warped image brightness with respect to the affine parameters, and $\mathbf{e}$ is the brightness error vector. The Jacobian for each pixel (x, y) with image gradients (I_x, I_y) has columns:

$$\mathbf{J}_{\text{row}} = \begin{bmatrix} I_x \cdot x & I_x \cdot y & I_y \cdot x & I_y \cdot y & I_x & I_y \end{bmatrix}$$

Iteration proceeds for up to 30 steps or until the translation update $\|\Delta \mathbf{t}\| < 0.005$ pixels. After convergence, the affine parameters are decomposed to extract local rotation $\theta = \operatorname{atan2}(a_{10},\, a_{00})$ and scale $s = \sqrt{\det(\mathbf{A})}$. Tiles with NCC confidence below 0.3 or exhibiting excessive local scale deformation ($|s - 1| > 0.1$) are rejected.

5.3 Global Offset Refinement from Tile Medians

After coarse tile registration, the median of the valid tiles' residual displacements in x and y is computed. This median is added to the global offset, and the residuals are re-centered. This step separates the true lens distortion field (which is radially symmetric about the optical center) from a registration bias that would otherwise contaminate the distortion fit.

5.4 Robust Outlier Rejection via MAD

Outlier rejection uses the Median Absolute Deviation (MAD) [18, 19] applied independently to the x and y residual components. The normalized MAD estimator is:

$$\text{MAD}_n = 1.4826 \cdot \text{median}_i \left| r_i - \text{median}_j(r_j) \right|$$

where the factor 1.4826 makes MAD_n a consistent estimator of the standard deviation under Gaussian noise. Tiles whose residual exceeds $3.5 \cdot \text{MAD}_n$ from the median are rejected. A secondary spatial consistency check rejects tiles whose residual deviates from the mean of their neighbors (within 15% of the image diagonal) by more than $\max(3.0,\ 2.5 \cdot \text{MAD})$.

5.5 Brown-Conrady Lens Distortion Fitting

The lens distortion is modeled using the four-parameter Brown-Conrady model [13, 14] with radial coefficients k_1, k_2 and tangential coefficients p_1, p_2. For a point at normalized coordinates $(x_n,\ y_n) = \left(\frac{x - c_x}{s},\ \frac{y - c_y}{s}\right)$, where (c_x, c_y) is the optical center and $s = \max(W, H)/2$, the distortion displacement is:

$$\Delta x = x_n(k_1 r^2 + k_2 r^4) + p_1(r^2 + 2x_n^2) + 2p_2 x_n y_n$$

$$\Delta y = y_n(k_1 r^2 + k_2 r^4) + 2p_1 x_n y_n + p_2(r^2 + 2y_n^2)$$

where $r^2 = x_n^2 + y_n^2$. The distortion center (c_x, c_y) is found by a two-stage grid search: a coarse sweep over $[0.3,\ 0.7] \times [0.3,\ 0.7]$ in normalized image coordinates at 0.05 steps, followed by a fine sweep at 0.01 steps within ± 0.04 of the coarse optimum. At each candidate center, the four distortion coefficients are solved via weighted least squares:

$$(\mathbf{A}^\top \mathbf{W} \mathbf{A})\, \mathbf{p} = \mathbf{A}^\top \mathbf{W} \mathbf{b}$$

where each valid tile contributes two rows (one for x, one for y), the weight $w_i = \text{confidence}_i^2$, and the design matrix columns correspond to the four partial derivatives of the distortion model with respect to $(k_1,\ k_2,\ p_1,\ p_2)$. The system is solved by Gaussian elimination with partial pivoting.

5.6 Dense Lucas-Kanade Optical Flow

Dense optical flow is computed at every 8th pixel ($\text{flowStep} = 8$) across the overlap region. At each grid point (x, y) in the reference image, the corresponding point in the target image is located at $(x - d_x + f_x,\ y - d_y + f_y)$, where (d_x, d_y) is the global offset and (f_x, f_y) is the local flow displacement.

The flow at each point is estimated by the translational Lucas-Kanade algorithm over an 11×11-pixel window. The 2×2 structure tensor is:

$$\mathbf{M} = \sum_w \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

and the displacement update is $\Delta\mathbf{d} = \mathbf{M}^{-1}\mathbf{b}$, where $\mathbf{b} = -\sum_w \begin{bmatrix} I_x\,\delta I \\ I_y\,\delta I \end{bmatrix}$ and δI is the brightness difference between the reference and warped target patches. Iteration proceeds for up to 20 steps or until $\|\Delta\mathbf{d}\| < 0.01$ pixels.

5.6.1 Iterative Dual-Initialization Refinement

The flow field undergoes 4 refinement passes ($\text{flowRefineIters} = 4$). In each pass, grid points whose RGB error exceeds a threshold ($\text{flowErrorThresh} = 25.0$) are re-estimated using two independent initializations: (a) the current flow value, and (b) the average of 8-connected neighbors' flow. The initialization that produces lower error after LK iteration is accepted. The error at each grid point is the mean absolute RGB difference between the reference and flow-warped target over the window:

$$e = \frac{1}{N} \sum \frac{|R_r - T_r| + |R_g - T_g| + |R_b - T_b|}{3}$$

5.6.2 Adaptive Spatial Smoothing

After each refinement pass, flow vectors with error above the threshold are replaced by a Gaussian-weighted average of their neighbors. The Gaussian weights are modulated by each neighbor's inverse error, so that well-matched neighbors contribute more to the correction:

$$\mathbf{f}_{\text{smooth}}(x, y) = \frac{\sum_j G(\Delta x, \Delta y;\, \sigma) \cdot \dfrac{1}{e_j} \cdot \mathbf{f}_j}{\sum_j G(\Delta x, \Delta y;\, \sigma) \cdot \dfrac{1}{e_j}}$$

where the smoothing radius is 2 grid cells and only neighbors with error below threshold contribute.

5.7 Hierarchical Per-Pixel Flow Upsampling

The 8-pixel flow grid is upsampled to per-pixel resolution through three successive doubling stages: $8 \to 4$, $4 \to 2$, $2 \to 1$. At each level, the coarser grid is interpolated to twice its resolution using Catmull-Rom bicubic interpolation.

The Catmull-Rom kernel (with parameter $a = -0.5$) is a piecewise cubic function evaluated on the 4×4 neighborhood of each output point [16]:

$$h(t) = \begin{cases} (1.5|t| - 2.5)\,|t|^2 + 1, & |t| \le 1 \\ (-0.5|t| + 2.5)\,|t|^2 - 4|t| + 2, & 1 < |t| \le 2 \end{cases}$$

The 2D interpolation weight at fractional position (f_x, f_y) is the separable product $w_{ij} = h(f_x - i) \cdot h(f_y - j)$ for each of the 16 neighbors $(i, j) \in \{-1, 0, 1, 2\}^2$. The interpolated flow is:

$$\mathbf{f}(x, y) = \sum_{i,j} w_{ij} \cdot \mathbf{f}_{\text{grid}}(i, j)$$

At boundary points where the 4×4 neighborhood extends outside the grid, the interpolator falls back to bilinear interpolation over the available 2×2 neighborhood. This fallback uses the standard bilinear weights: $w = (1 - f_x)(1 - f_y),\ f_x(1 - f_y),\ (1 - f_x)f_y,\ f_x f_y$.

5.8 Flow Extrapolation Beyond the Overlap

Outside the flow grid, the flow is extrapolated by clamping the query point to the nearest boundary grid point and applying exponential decay:

$$\mathbf{f}_{\text{extrap}}(x, y) = \mathbf{f}_{\text{boundary}} \cdot \exp\left(\frac{-3\, d}{d_{\max}}\right)$$

where d is the Euclidean distance from the query point to the flow grid boundary in pixels, and $d_{\max} = 400$ pixels is the maximum extrapolation range (`boundaryDecayPx`). Points beyond $d_{\max}$ receive no flow correction. The factor -3 in the exponent ensures that the correction decays to approximately 5% of the boundary value at full distance ($e^{-3} \approx 0.05$).

5.9 Dynamic-Programming Seam Finding

The seam orientation is chosen automatically: if $|\text{offset}_y| > |\text{offset}_x|$, the seam runs horizontally (left $\rightarrow$ right); otherwise vertically (top $\rightarrow$ bottom). The overlap region is downsampled by a factor of 4 ($\text{dsStep} = 4$) for efficiency.

At each downsampled pixel, the cost combines the color difference between the reference and flow-warped target with a flow-magnitude penalty:

$$\text{cost}(x, y) = \|R(x, y) - T_w(x, y)\|^2 + 2\,\|\mathbf{f}(x, y)\|^2$$

where R is the reference color, T_w is the flow-warped target color, and $\mathbf{f}$ is the flow vector. The flow penalty steers the seam through regions of low flow magnitude where the warp is most trustworthy.

A center bias is added to keep the seam near the middle of the overlap where both images have valid data. The bias uses a combined quadratic-quartic bowl:

$$\text{bias}(c) = (\hat{c}^2 + \hat{c}^4) \cdot \alpha$$

where $\hat{c} = (c - c_{\text{mid}})/c_{\text{mid}}$ is the normalized distance from the center of the cross-dimension and $\alpha = \max(50,\ 0.3 \cdot \overline{\text{cost}})$.

The DP propagates along the sweep dimension with ± 1-step connectivity in the cross-dimension (3-connected). The optimal seam is traced back from the minimum-cost endpoint.

The downsampled path is upsampled to full resolution by linear interpolation between adjacent downsampled points, then smoothed by a box filter with a radius of 16 pixels to eliminate jagged staircase artifacts.

5.10 Mosaic Composition and Blending

The mosaic canvas is sized to span both images and is populated in two passes: the reference image is placed at its natural coordinates, and the target image is placed at the global offset with per-pixel flow correction applied. Flow correction uses the per-pixel arrays (`pixFlowDX`, `pixFlowDY`) when available, with fallback to bicubic interpolation from the 8-pixel grid via `getFlowAtExtrap()` for pixels outside the per-pixel coverage.

5.10.1 Blend Mask Construction

In PYRAMID mode (the default), a binary seam mask is constructed: pixels on the reference side of the seam receive $\text{mask} = 0$, and pixels on the target side receive $\text{mask} = 1$. The Laplacian pyramid blending then smooths the transition across frequency bands.

In GRADIENT mode (toggle with G key), a distance-ratio mask provides smooth linear feathering. For each pixel where both images overlap, the blending weight toward the target is:

$$\text{mask} = \frac{w_T}{w_R + w_T}$$

where $w_R = \min(d_{\text{left}}, d_{\text{right}}, d_{\text{top}}, d_{\text{bottom}})$ is the minimum distance from the pixel to any edge of the reference image (and analogously for w_T in target coordinates). This produces a linear cross-dissolve weighted by proximity to image boundaries. The mask is then Gaussian-blurred with radius $= \text{seamFeather}/4 = 30$ pixels.

5.10.2 Source Fill for Black-Bleed Prevention

Before blending, pixels covered by only one source are filled with the other source's color. This prevents black (zero) values from bleeding into the blend at the periphery of the overlap, which would create dark halos around the mosaic edges.

5.11 Laplacian Pyramid Multi-Band Blending

The Laplacian pyramid blend [9] operates on each color channel independently using $L = 5$ pyramid levels. The procedure is:

(1) Gaussian pyramids: Construct Gaussian pyramids $\{G_A^l\}$, $\{G_B^l\}$, and $\{G_M^l\}$ for both source images and the mask. Each level is obtained by convolving with a 5×5 binomial kernel $[1,\ 4,\ 6,\ 4,\ 1]/16$ (applied separably) and downsampling by factor 2.

(2) Laplacian pyramids: Compute $L_A^l = G_A^l - \text{Upsample}(G_A^{l+1})$ for each level, with the coarsest level retained as-is. Upsampling uses bilinear interpolation with proper pixel-center coordinate mapping: $s_x = (w_{\text{src}} - 1)/(w_{\text{dst}} - 1)$.

(3) Blended pyramid:

$$B^l = (1 - G_M^l) \cdot L_A^l + G_M^l \cdot L_B^l$$

(4) Collapse: Starting from the coarsest blended level, iteratively upsample and add the next finer blended band:

$$\text{result} = \text{Upsample}(\text{result}) + B^l$$

The multi-band blend allows low-frequency exposure differences to transition gradually (over hundreds of pixels at the coarsest level) while high-frequency texture detail transitions sharply (within a few pixels at the finest level), producing perceptually seamless results.

5.12 Gaussian-Smoothed Boundary Correction

The boundary between the blended overlap zone and the single-image zone can exhibit a color discontinuity because the blended overlap is a weighted mixture of two exposures while the adjacent single-image zone contains only one. The correction procedure is:

(1) Boundary detection: Identify overlap pixels whose 4-connected neighborhood includes a non-overlap pixel.

(2) Correction measurement: At each boundary pixel, measure the mean color on the overlap side and the non-overlap side within a 7×7 kernel. The correction vector is the difference:

$$\Delta \mathbf{c} = \bar{\mathbf{c}}_{\text{overlap}} - \bar{\mathbf{c}}_{\text{non-overlap}}$$

per channel (R, G, B).

(3) Grid scatter: Scatter the boundary corrections onto a coarse grid with cell size 8 pixels. Multiple boundary pixels mapping to the same cell are averaged.

(4) Gaussian blur: The grid is separably Gaussian-blurred with $\sigma = \max\left(2,\ \frac{d_{\max}}{8 \cdot 3}\right)$ grid cells, using normalized weighted-only convolution (cells with no scattered data are excluded from the kernel normalization). This produces a smooth, continuous correction field that eliminates the Voronoi discontinuities of the prior nearest-neighbor approach.

(5) Distance field: An 8-connected BFS computes the Euclidean distance from each non-overlap pixel to the nearest boundary pixel, using cardinal distance 1 and diagonal distance $\sqrt{2} \approx 1.414$.

(6) Correction application: At each non-overlap pixel within $d_{\max}$, the correction is bilinearly sampled from the blurred grid and attenuated by exponential decay:

$$\mathbf{c}_{\text{corrected}} = \mathbf{c}_{\text{original}} + \mathbf{c}_{\text{grid}}(x, y) \cdot \exp\left(\frac{-3\,d}{d_{\max}}\right)$$

This produces a smooth color transition that fades to zero correction over 400 pixels (the `boundaryDecayPx` parameter).

5.13 Mesh Warp Mosaic via IDW (Legacy Mode)

An alternative mosaic mode (toggle with W key) uses inverse distance weighting [15] to interpolate the per-tile affine corrections across the image. For each output pixel, the warp correction is a weighted sum over all valid tiles:

$$\Delta \mathbf{f}(x, y) = \frac{\sum_i w_i \left[(\mathbf{A}_i - \mathbf{I})\, \Delta \mathbf{r}_i + \mathbf{d}_i \right]}{\sum_i w_i}$$

where $\Delta \mathbf{r}_i = (x - x_i,\ y - y_i)$ is the displacement from tile center, $\mathbf{A}_i$ is the local affine matrix, $\mathbf{d}_i$ is the tile's residual displacement, and $w_i = \text{confidence}_i / (d + 0.5)^3$ is the IDW weight with exponent $p = 3$ (sharper than Shepard's original $p = 2$). Tiles beyond an influence radius of half the image diagonal are excluded.

5.14 Image Distortion Correction

When calibration parameters are available (toggle with U key), both images are undistorted by inverse-mapping each output pixel through the Brown-Conrady model: for output pixel (x, y), compute the normalized coordinates, evaluate the distortion displacement $(\Delta x,\ \Delta y)$ as in Section 5.5, and sample the input image at $(x + \Delta x \cdot s,\ y + \Delta y \cdot s)$ using bilinear interpolation.

5.15 Bilinear Sampling

All sub-pixel image lookups use bilinear interpolation. For a query at continuous coordinates (x, y), the integer floor is (x_0, y_0) and the fractional part is (f_x, f_y). The interpolated value is:

$$I(x,y) = I_{00}(1-f_x)(1-f_y) + I_{10}\,f_x(1-f_y) + I_{01}(1-f_x)f_y + I_{11}\,f_x f_y$$

where $I_{ij} = I(x_0+i,\ y_0+j)$. This is applied to all three color channels independently. At image boundaries, coordinates are clamped to the valid range and nearest-neighbor sampling is used as a fallback.

5.16 Linear System Solver

All linear systems (6×6 for affine LK, 4×4 for distortion fitting) are solved by Gaussian elimination with partial pivoting. The augmented matrix $[\mathbf{A}|\mathbf{b}]$ is row-reduced; if the maximum pivot element falls below 10^{-14}, the system is declared singular and the tile/fit is rejected. Back-substitution recovers the solution vector.

5.17 Unified Notation and the Master Equation

This section consolidates the per-stage formulations of Sections 5.1–5.16 into a single end-to-end expression that maps every output pixel in the final mosaic to the input pixels of the reference image R and the target image T. The master equation is developed in stages, first defining the coordinate transformations, then the per-pixel color contributions, and finally the complete output including boundary correction.

5.17.1 Coordinate Conventions

Let the mosaic canvas have dimensions $W_m \times H_m$ and pixel coordinates $(x_m, y_m) \in \{0, \ldots, W_m - 1\} \times \{0, \ldots, H_m - 1\}$. The reference image R of size $W_R \times H_R$ is placed at canvas origin (r_x, r_y), so the reference-local coordinate of canvas pixel (x_m, y_m) is:

$$(u_R,\ v_R) = (x_m - r_x,\ y_m - r_y)$$

The target image T of size $W_T \times H_T$ is displaced by the global offset (d_x, d_y) found in Section 5.1 and refined in Section 5.3. In the absence of any flow correction, the target-local coordinate is:

$$(u_T^{(0)},\ v_T^{(0)}) = (u_R - d_x,\ v_R - d_y)$$

5.17.2 Per-Pixel Flow Correction

The dense optical flow field $\mathbf{f}$ assigns a displacement correction to each reference-coordinate pixel. The flow is computed on an 8-pixel grid and upsampled hierarchically (Section 5.7). Let $\mathbf{f}^{(s)}$ denote the flow field at grid spacing $s \in \{8, 4, 2, 1\}$, with $\mathbf{f}^{(8)}$ being the LK-estimated field after iterative dual-initialization refinement (Section 5.6.1). Each successive level is obtained by Catmull-Rom bicubic interpolation from the previous:

$$\mathbf{f}^{(s/2)}(x,y) = \sum_{i=-1}^{2}\sum_{j=-1}^{2} h\Big(\frac{x}{s} - \Big\lfloor\frac{x}{s}\Big\rfloor - i\Big)\ h\Big(\frac{y}{s} - \Big\lfloor\frac{y}{s}\Big\rfloor - j\Big)\ \mathbf{f}^{(s)}\Big(\Big\lfloor\frac{x}{s}\Big\rfloor + i,\ \Big\lfloor\frac{y}{s}\Big\rfloor + j\Big)$$

for $s \in \{8, 4, 2\}$, where $h(t)$ is the Catmull-Rom kernel defined in Section 5.7. The final per-pixel flow is $\mathbf{f}^{(1)}(u_R, v_R) = (f_x, f_y)$.

The flow-corrected target-local coordinate is therefore:

$$(u_T,\, v_T) = (u_R - d_x + f_x(u_R, v_R),\; v_R - d_y + f_y(u_R, v_R))$$

For pixels outside the flow grid, the flow is extrapolated (Section 5.8):

$$\mathbf{f}_{\text{ext}}(u_R, v_R) = \mathbf{f}^{(1)}(u_R^*, v_R^*) \cdot \exp\!\left(\frac{-3\,\|(u_R, v_R) - (u_R^*, v_R^*)\|}{d_{\max}}\right)$$

where (u_R^*, v_R^*) is the nearest boundary point of the flow grid, and $d_{\max} = 400$ px.

5.17.3 Source Color Sampling

Both source images are sampled at continuous coordinates using bilinear interpolation (Section 5.15). For any continuous coordinate (u, v) with integer floor (u_0, v_0) and fractional part (α, β):

$$\mathcal{B}[I](u, v) = \sum_{i=0}^{1}\sum_{j=0}^{1} I(u_0 + i,\, v_0 + j) \;\cdot\; |1 - i - \alpha|_+ \;\cdot\; |1 - j - \beta|_+$$

where $|\cdot|_+$ denotes the positive part (i.e., the standard bilinear weight: $(1-\alpha)(1-\beta)$, $\alpha(1-\beta)$, $(1-\alpha)\beta$, $\alpha\beta$). Applied per-channel, the reference and target color values at canvas pixel (x_m, y_m) are:

$$\mathbf{c}_R(x_m, y_m) = \mathcal{B}[R](u_R,\, v_R), \qquad \mathbf{c}_T(x_m, y_m) = \mathcal{B}[T](u_T,\, v_T)$$

where each $\mathbf{c} = (c^r, c^g, c^b)^\top$ is a 3-vector. A source is valid at (x_m, y_m) if and only if its local coordinates fall within the image bounds:

$$1_R(x_m, y_m) = \begin{cases} 1 & \text{if } 0 \le u_R < W_R \text{ and } 0 \le v_R < H_R \\ 0 & \text{otherwise} \end{cases}$$

and analogously, 1_T for the flow-corrected target coordinates.

5.17.4 Seam Mask Construction

The dynamic-programming seam (Section 5.9) partitions the overlap region. Let the seam path be $\mathcal{S}$, parameterized as $\mathcal{S}(t)$ along the sweep dimension. Define the seam indicator for each overlap pixel (u_R, v_R) in overlap-local coordinates $(l_x, l_y) = (u_R - o_x, v_R - o_y)$ as:

$$\sigma(l_x, l_y) = \begin{cases} 1 & \text{if } l_y > \mathcal{S}(l_x) \;\; \text{(horizontal seam: pixel is on target side)} \\ 0 & \text{otherwise} \end{cases}$$

(with the analogous definition for vertical seams). For pixels outside the overlap, the mask defaults to whichever single source covers the pixel:

$$m_0(x_m, y_m) = \begin{cases} \sigma(l_x, l_y) & \text{if } 1_R = 1 \text{ and } 1_T = 1 \\ 1 & \text{if } 1_T = 1 \text{ and } 1_R = 0 \\ 0 & \text{otherwise} \end{cases}$$

5.17.5 Laplacian Pyramid Blending Operator

The Laplacian pyramid blend (Section 5.11) is a linear operator on the two source images and the mask. Let $\mathcal{G}_\sigma$ denote the Gaussian convolution operator with the 5×5 binomial kernel applied separably, and let $\downarrow_2$ and $\uparrow_2$ denote downsampling and upsampling by factor 2 (with bilinear interpolation). Define the Gaussian pyramid at level l recursively:

$$G^0[I] = I, \qquad G^l[I] = \mathcal{G}_\sigma \downarrow_2 \left(G^{l-1}[I]\right) \quad \text{for } l = 1, \ldots, L$$

The Laplacian pyramid bands are:

$$\Lambda^l[I] = \begin{cases} G^l[I] - \uparrow_2 G^{l+1}[I] & l = 0, \ldots, L-1 \\ G^L[I] & l = L \end{cases}$$

The blended pyramid is formed by mixing each band using the Gaussian pyramid of the mask:

$$B^l = (1 - G^l[m_0]) \cdot \Lambda^l[\mathbf{c}_R] \; + \; G^l[m_0] \cdot \Lambda^l[\mathbf{c}_T]$$

The blended image is recovered by collapsing the pyramid:

$$\mathbf{c}_{\text{blend}} = \sum_{l=0}^{L-1} \uparrow_2^{(l)} B^l \; + \; \uparrow_2^{(L)} B^L$$

where $\uparrow_2^{(l)}$ denotes l successive upsample-by-2 operations. Expanding all operators, the blended color at pixel (x_m, y_m) for a single channel is:

$$c_{\text{blend}}(x_m, y_m) = \sum_{l=0}^{L} \uparrow_2^{(l)} \left[\, (1 - G^l[m_0]) \cdot \Lambda^l[c_R] \; + \; G^l[m_0] \cdot \Lambda^l[c_T] \, \right] (x_m, y_m)$$

This is a linear combination of c_R and c_T with spatially varying, frequency-dependent weights determined by the mask pyramid $\{G^l[m_0]\}$.

5.17.6 Source Fill (Black-Bleed Prevention)

Before blending, any pixel covered by only one source is filled with the other source's color (Section 5.10.2). Define the pre-blend source colors:

$$\tilde{\mathbf{c}}_R = \begin{cases} \mathbf{c}_R & \text{if } 1_R = 1 \\ \mathbf{c}_T & \text{if } 1_R = 0,\ 1_T = 1 \end{cases} \qquad \tilde{\mathbf{c}}_T = \begin{cases} \mathbf{c}_T & \text{if } 1_T = 1 \\ \mathbf{c}_R & \text{if } 1_T = 0,\ 1_R = 1 \end{cases}$$

The Laplacian blend then operates on $\tilde{\mathbf{c}}_R$ and $\tilde{\mathbf{c}}_T$.

5.17.7 Boundary Correction Field

The blended mosaic $\mathbf{c}_{\text{blend}}$ may exhibit a color discontinuity at the boundary $\partial\Omega$ between the overlap region Ω and the single-image region. The correction field (Section 5.12) is constructed in four stages.

Stage 1: Boundary measurement. At each boundary pixel $\mathbf{p} \in \partial\Omega$, the correction vector is measured over a $(2K+1) \times (2K+1)$ kernel ($K = 3$):

$$\Delta\mathbf{c}(\mathbf{p}) = \frac{\sum_{\mathbf{q}\in\mathcal{N}_K(\mathbf{p})\cap\Omega} \mathbf{c}_{\text{blend}}(\mathbf{q})}{|\mathcal{N}_K(\mathbf{p})\cap\Omega|} - \frac{\sum_{\mathbf{q}\in\mathcal{N}_K(\mathbf{p})\setminus\Omega} \mathbf{c}_{\text{blend}}(\mathbf{q})}{|\mathcal{N}_K(\mathbf{p})\setminus\Omega|}$$

where $\mathcal{N}_K(\mathbf{p})$ is the set of pixels within the K-radius square neighborhood.

Stage 2: Grid scatter. The boundary corrections are accumulated onto a coarse grid with cell size $\gamma = 8$ px. For grid cell (g_x, g_y), let $\mathcal{P}_{g_x,g_y} = \{\mathbf{p} \in \partial\Omega : \lfloor p_x/\gamma \rfloor = g_x,\ \lfloor p_y/\gamma \rfloor = g_y\}$. The cell correction is:

$$\Delta\mathbf{c}_{\text{grid}}(g_x, g_y) = \frac{1}{|\mathcal{P}_{g_x,g_y}|} \sum_{\mathbf{p}\in\mathcal{P}_{g_x,g_y}} \Delta\mathbf{c}(\mathbf{p})$$

and the cell is marked as populated ($w_{g_x,g_y} = |\mathcal{P}_{g_x,g_y}| > 0$).

Stage 3: Gaussian blur. The populated grid is separably Gaussian-blurred with $\sigma_g = \max(2,\ d_{\max}/(\gamma \cdot 3))$ grid cells. The blur uses weighted-only normalization (unpopulated cells excluded):

$$\hat{\Delta}\mathbf{c}(g_x, g_y) = \frac{\sum_{i=-R}^{R}\sum_{j=-R}^{R} 1_w(g_x+i,\ g_y+j)\ \cdot\ G(i,j;\,\sigma_g)\ \cdot\ \Delta\mathbf{c}_{\text{grid}}(g_x+i,\ g_y+j)}{\sum_{i=-R}^{R}\sum_{j=-R}^{R} 1_w(g_x+i,\ g_y+j)\ \cdot\ G(i,j;\,\sigma_g)}$$

where $R = \lceil 3\sigma_g \rceil$, $G(i,j;\sigma_g) = \exp\!\left(-\frac{i^2+j^2}{2\sigma_g^2}\right)$, and 1_w indicates whether a cell has scattered data. (In practice, the blur is applied separably: horizontal then vertical, each with a 1D kernel.)

Stage 4: Sampling and decay. At each non-overlap pixel $(x_m, y_m) \notin \Omega$ within distance $d_{\max}$ of the boundary, the correction is bilinearly sampled from the blurred grid and attenuated by exponential decay. Let $d(x_m, y_m) = \text{BFS}_{\partial\Omega}(x_m, y_m)$ be the 8-connected Euclidean distance to the nearest boundary pixel. Then:

$$\mathbf{C}(x_m, y_m) = \mathcal{B}\Big[\hat{\Delta}\mathbf{c}\Big]\left(\frac{x_m}{\gamma} - \tfrac{1}{2},\ \frac{y_m}{\gamma} - \tfrac{1}{2}\right) \cdot \exp\!\left(\frac{-3\,d(x_m, y_m)}{d_{\max}}\right)$$

where $\mathcal{B}$ is the bilinear interpolation operator (Section 5.15), applied in grid coordinates.

5.17.8 The Master Equation

Assembling all stages, the final output color at canvas pixel (x_m, y_m) for each channel $k \in \{r, g, b\}$ is:

$$\mathbf{c}^k_{\text{out}}(x_m, y_m) = \underbrace{\sum_{l=0}^{L} \uparrow_2^{(l)} \left[\ (1 - G^l[m_0]) \cdot \Lambda^l[\tilde{c}^k_R]\ +\ G^l[m_0] \cdot \Lambda^l[\tilde{c}^k_T]\ \right](x_m, y_m)}_{\text{Laplacian pyramid multi-band blend}} + \underbrace{(1 - 1_\Omega)\ \cdot\ \mathbf{C}^k(x_m, y_m)}_{\text{boundary correction}}$$

where each constituent is itself a composition of all upstream stages. Expanding the source terms fully:

Reference source:

$$\tilde{c}_R^k(x_m, y_m) = \begin{cases} \displaystyle\sum_{i=0}^{1}\sum_{j=0}^{1} R^k(u_{R,0}+i,\, v_{R,0}+j)\,(1-i-\alpha_R)(1-j-\beta_R) & \text{if } 1_R = 1 \\ \tilde{c}_T^k(x_m, y_m) & \text{if } 1_R = 0,\ 1_T = 1 \end{cases}$$

where $(u_{R,0}, v_{R,0}) = (\lfloor u_R \rfloor, \lfloor v_R \rfloor)$ and $(\alpha_R, \beta_R) = (u_R - u_{R,0},\, v_R - v_{R,0})$.

Target source (flow-corrected):

$$\tilde{c}_T^k(x_m, y_m) = \begin{cases} \displaystyle\sum_{i=0}^{1}\sum_{j=0}^{1} T^k(u_{T,0}+i,\, v_{T,0}+j)\,(1-i-\alpha_T)(1-j-\beta_T) & \text{if } 1_T = 1 \\ \tilde{c}_R^k(x_m, y_m) & \text{if } 1_T = 0,\ 1_R = 1 \end{cases}$$

where the target-local coordinates incorporate the full flow chain:

$$u_T = (x_m - r_x) - d_x + f_x^{(1)}(x_m - r_x,\, y_m - r_y)$$

$$v_T = (y_m - r_y) - d_y + f_y^{(1)}(x_m - r_x,\, y_m - r_y)$$

and the per-pixel flow $\mathbf{f}^{(1)}$ is the result of three hierarchical Catmull-Rom upsampling stages from the LK-estimated field $\mathbf{f}^{(8)}$:

$$f_x^{(1)}(u_R, v_R) = \sum_{i_1=-1}^{2}\sum_{j_1=-1}^{2} h(\phi_x^{(2)} - i_1)\, h(\phi_y^{(2)} - j_1) \left[\sum_{i_2=-1}^{2}\sum_{j_2=-1}^{2} h(\phi_x^{(4)} - i_2)\, h(\phi_y^{(4)} - j_2) \left[\sum_{i_3=-1}^{2}\sum_{j_3=-1}^{2} h(\phi_x^{(8)} - i_3)\, h(\phi_y^{(8)} - j_3)\, f_x^{(8)}(g_{i_3}, g_{j_3}) \right] \right]$$

where for each level $s \in \{8, 4, 2\}$:

$$\phi_x^{(s)} = \frac{u_R}{s} - \left\lfloor \frac{u_R}{s} \right\rfloor, \qquad g_i = \left\lfloor \frac{u_R}{s} \right\rfloor + i$$

and $h(t)$ is the Catmull-Rom kernel of Section 5.7. An identical expression holds for $f_y^{(1)}$.

The flow field itself at the coarsest level is the converged LK estimate after $P = 4$ refinement passes. At grid point $\mathbf{g} = (g_x, g_y)$ on the 8-pixel grid, define the error-evaluated dual-init operator $\mathcal{D}$ (Section 5.6.1). At each pass $p \in \{1, \ldots, P\}$:

$$\mathbf{f}^{(8,p)}(\mathbf{g}) = \begin{cases} \arg\min_{\mathbf{f}^* \in \{\mathbf{f}_a,\, \mathbf{f}_b\}} e(\mathbf{g},\, \mathbf{f}^*) & \text{if } e(\mathbf{g},\, \mathbf{f}^{(8,p-1)}) > \tau_e \\ \mathbf{f}^{(8,p-1)}(\mathbf{g}) & \text{otherwise} \end{cases}$$

where:
– $\mathbf{f}_a = \mathrm{LK}(\mathbf{g},\, \mathbf{f}^{(8,p-1)}(\mathbf{g}))$ is the LK result initialized from the current flow
– $\mathbf{f}_b = \mathrm{LK}\left(\mathbf{g},\, \frac{1}{|\mathcal{N}_8|} \displaystyle\sum_{\mathbf{g}' \in \mathcal{N}_8(\mathbf{g})} \mathbf{f}^{(8,p-1)}(\mathbf{g}')\right)$ is the LK result initialized from the 8-connected neighbor mean

– $\tau_e = 25.0$ is the RGB error threshold
– $e(\mathbf{g}, \mathbf{f})$ is the mean absolute RGB error:

$$e(\mathbf{g}, \mathbf{f}) = \frac{1}{|W|} \sum_{(x,y) \in W(\mathbf{g})} \frac{|R^r(x,y) - T_w^r| + |R^g(x,y) - T_w^g| + |R^b(x,y) - T_w^b|}{3}$$

with $T_w^k = \mathcal{B}[T^k](x - d_x + f_x,\, y - d_y + f_y)$ and $W(\mathbf{g})$ the 11×11 window.

After each dual-init pass, the adaptive spatial smoothing operator (Section 5.6.2) is applied to points still above threshold:

$$\mathbf{f}^{(8,p)}_{\text{smooth}}(\mathbf{g}) = \frac{\displaystyle\sum_{\mathbf{g}' \in \mathcal{N}_2(\mathbf{g})} 1_{e<\tau_e}(\mathbf{g}') \cdot \exp\!\left(-\frac{\|\mathbf{g} - \mathbf{g}'\|^2}{2\sigma_s^2}\right) \cdot \frac{1}{e(\mathbf{g}',\, \mathbf{f}^{(8,p)})} \cdot \mathbf{f}^{(8,p)}(\mathbf{g}')}{\displaystyle\sum_{\mathbf{g}' \in \mathcal{N}_2(\mathbf{g})} 1_{e<\tau_e}(\mathbf{g}') \cdot \exp\!\left(-\frac{\|\mathbf{g} - \mathbf{g}'\|^2}{2\sigma_s^2}\right) \cdot \frac{1}{e(\mathbf{g}',\, \mathbf{f}^{(8,p)})}}$$

where $\mathcal{N}_2$ is the 2-cell-radius neighborhood and $\sigma_s = 2$ grid cells.

The seam mask m_0 is determined by the DP seam path $\mathcal{S}$, which is the minimum-cost path through the overlap cost surface:

$$\mathcal{S} = \arg\min_{\text{path}} \sum_t \left[\sum_{k \in \{r,g,b\}} \left(\tilde{c}_R^k(x_t, y_t) - \tilde{c}_T^k(x_t, y_t)\right)^2 + 2\left(f_x^2(x_t, y_t) + f_y^2(x_t, y_t)\right) + \left(\hat{c}_t^2 + \hat{c}_t^4\right) \alpha \right]$$

where $\hat{c}_t = (c_t - c_{\text{mid}})/c_{\text{mid}}$ is the normalized cross-dimension distance and $\alpha = \max(50,\ 0.3 \cdot \bar{E})$ with $\bar{E}$ the mean pixel cost.

The boundary correction $\mathbf{C}^k$ propagates measured color jumps across the overlap boundary $\partial\Omega$ into the single-image zone, as defined in Section 5.17.7.

5.17.9 Summary: From Input Pair to Output Pixel

Combining all of the above, the complete transformation from two raw input images R and T to the output color at any canvas pixel (x_m, y_m) can be stated as a single (if elaborate) functional composition. For each color channel k:

$$\mathbf{c}^k_{\text{out}}(x_m, y_m) = \mathcal{L}\Big[\, \tilde{\mathcal{B}}[R^k] \circ \pi_R,\ \ \tilde{\mathcal{B}}[T^k] \circ \pi_T \circ \Phi,\ \ m_{\mathcal{S}} \,\Big]\,(x_m, y_m) \ +\ (1 - 1_\Omega) \cdot \mathcal{E}^k_{\partial\Omega}(x_m, y_m)$$

where:

Symbol	Definition
π_R	Reference projection: $(x_m, y_m) \mapsto (x_m - r_x,\ y_m - r_y)$
π_T	Target projection: $(u_R, v_R) \mapsto (u_R - d_x,\ v_R - d_y)$
Φ	Per-pixel flow warp: $(u, v) \mapsto (u + f_x^{(1)},\, v + f_y^{(1)})$, with $\mathbf{f}^{(1)}$ from 3-stage Catmull-Rom upsample of $\mathbf{f}^{(8)}$

$\mathbf{f}^{(8)}$	Dense LK flow after P dual-init refinement passes with error-weighted smoothing
$\tilde{\mathcal{B}}$	Bilinear sampler with source-fill fallback (Section 5.17.6)
$m_{\mathcal{S}}$	Binary seam mask from DP with flow-magnitude + center-bias cost
$\mathcal{L}$	Laplacian pyramid blend: $\sum_{l=0}^{L} \uparrow_2^{(l)} \left[(1 - G^l[m]) \cdot \Lambda^l[\cdot] + G^l[m] \cdot \Lambda^l[\cdot]\right]$
$\mathcal{E}_{\partial\Omega}^{k}$	Boundary correction: scatter $\rightarrow$ Gaussian blur $\rightarrow$ bilinear sample $\rightarrow$ exponential decay
1_{Ω}	Overlap indicator

This master equation encodes the complete data path: every pixel of the output mosaic is a multi-scale, seam-guided, frequency-decomposed blend of bilinearly-sampled source colors, where the target source coordinates are warped by a per-pixel flow field obtained through iterative dual-initialization Lucas-Kanade on an 8-pixel grid, hierarchically upsampled via three stages of Catmull-Rom bicubic interpolation, and where the boundary between the blended overlap zone and the single-image zone is healed by a Gaussian-smoothed, exponentially decaying correction field.

6. Statement of Novelty

The pipeline described in this paper is, taken as a whole, a novel process for aerial image mosaicing. While individual components draw on established techniques, NCC [5], Lucas-Kanade optical flow [7], Brown-Conrady distortion modeling [13, 14], Laplacian pyramid blending [9], and dynamic-programming seam finding [10], the system introduces six specific innovations and combines them into a self-contained, self-calibrating architecture that has no direct precedent in the literature. This section identifies each novelty, states what conventional limitation it corrects, and summarizes the resulting advantage. Section 7 then provides the full comparative mathematics for each.

6.1 The Novel Process

Conventional aerial mosaicing pipelines follow a well-established sequence: detect sparse keypoints (SIFT, ORB), match them across images, estimate a global homography via RANSAC, and blend with a Voronoi or distance-weighted mask. This paradigm assumes that a single projective transformation adequately models the geometric relationship between images. In practice, uncalibrated consumer drone imagery violates this assumption due to lens distortion, terrain relief, wind-induced attitude changes, and rolling-shutter effects. The result is ghosting, misalignment in the overlap zone, visible seams, and abrupt color discontinuities at the overlap boundary.

The process introduced here replaces the global-homography paradigm with a dense, per-pixel registration cascade that is entirely feature-free, requires no calibration targets or GPS metadata, uses no external libraries or pretrained models, and runs on CPU in a single-file application. The pipeline progressively refines alignment through nine stages, from coarse global NCC through tiled affine tracking, self-calibrating distortion correction, dense optical flow with dual-initialization refinement, hierarchical Catmull-Rom upsampling, flow-aware seam finding, multi-band Laplacian blending, and scatter-blur-decay boundary correction, with each stage consuming quality signals produced by its predecessors.

6.2 Summary of Novel Contributions

The six specific innovations introduced by this pipeline, each of which corrects a concrete limitation of conventional methods, are:

Contribution 1: Dual-Initialization Iterative Flow Refinement

Problem corrected: Standard Lucas-Kanade flow refinement uses a single initialization per grid point. If the initial estimate sits in a poor local minimum of the brightness constancy cost surface, re-running LK from the same starting

point returns the same minimum. In highly textured scenes such as forest canopy, a substantial fraction of grid points can become trapped.

Innovation: At each refinement pass, every high-error grid point is re-estimated from two independent initializations, the current flow and the 8-connected neighbor mean, and the result with lower photometric error is kept. This provides a per-point, per-pass mechanism for escaping local minima through competitive local search, without multi-start optimization or learned components.

Advantage: Reduces the number of outlier flow vectors in textured scenes without increasing computational cost by more than $2\times$ per affected point. Unlike deep-network approaches (IRR, RAFT), this requires no training data and operates at the classical LK level.

Contribution 2: Error-Weighted Adaptive Spatial Smoothing

Problem corrected: Conventional flow smoothing (bilateral filtering, Horn-Schunck regularization) weights neighbors by image content, pixel intensity or gradient similarity. This preserves image edges but has no knowledge of whether a neighbor's flow is actually correct. A neighbor with low photometric error and a neighbor with high photometric error contribute equally if their pixel intensities are similar.

Innovation: The smoothing kernel weights each neighbor by the inverse of its registration error $(1/e)$ rather than by image similarity, and a hard gate excludes neighbors above the error threshold entirely. Smoothing is applied selectively, only to points still above the error threshold after dual-init refinement.

Advantage: The smoothing kernel is shaped by registration quality rather than image content. Well-aligned neighbors dominate; poorly aligned neighbors contribute negligibly regardless of texture similarity. This prevents bad flow from propagating through the field.

Contribution 3: Hierarchical Catmull-Rom Bicubic Flow Upsampling

Problem corrected: The standard approach to upsampling a sparse flow field (computed on an 8-pixel grid) to per-pixel resolution is single-step bilinear interpolation. This produces only C^0 continuity at grid boundaries, the interpolated flow has derivative discontinuities (kinks) at every 8th pixel, which manifest as visible period-2 stripe artifacts in the mosaic.

Innovation: The flow field is upsampled through three successive doubling stages ($8 \to 4 \to 2 \to 1$), with each stage using Catmull-Rom bicubic interpolation over a 4×4 neighborhood. The Catmull-Rom kernel provides C^1 continuity, smooth first derivatives, at every grid boundary.

Advantage: Eliminates the period-2 stripe artifacts caused by bilinear upsampling. The hierarchical structure ensures each doubling stage operates on an already-smooth field, preventing accumulation of interpolation error. No training data or GPU is required, unlike learned upsampling masks (RAFT, UPFlow).

Contribution 4: Flow-Magnitude-Weighted Seam Cost

Problem corrected: Conventional dynamic-programming seam finding uses color difference as the sole cost term. This steers the seam toward regions where the two images happen to look similar, but provides no information about whether the geometric alignment is trustworthy at that location. In uniform-color regions (shadows, sky, dark canopy), the color cost can be low even where the flow field is large and unreliable.

Innovation: A flow-magnitude penalty term $2(f_x^2 + f_y^2)$ is added to the DP cost function. Since flow magnitude directly measures how much geometric correction was needed at each pixel, large flow indicates parallax, terrain relief, rolling-shutter distortion, or residual misalignment, all cases where blending artifacts are most likely.

Advantage: The seam is steered through regions of minimal geometric correction, where the warp is most trustworthy, rather than merely through regions of similar color. This provides a registration-confidence signal that is entirely absent from conventional seam costs.

Contribution 5: Scatter-Blur-Decay Boundary Correction

Problem corrected: After blending the overlap zone, a color discontinuity typically remains at the boundary between the blended region (a weighted mixture of two exposures) and the adjacent single-image zone (one exposure only). Conventional solutions are either Poisson image editing (which requires solving a large sparse linear system) or extended feathering (which degrades sharpness). A simpler BFS nearest-neighbor approach was attempted but produced Voronoi-cell tessellations in the correction field, visible as sharp vertical stripe artifacts.

Innovation: Boundary color corrections are measured at each boundary pixel, scattered onto a coarse grid, Gaussian-blurred with weighted-only normalization, bilinearly sampled back to pixel resolution, and attenuated by exponential distance decay. This four-stage pipeline (scatter $\rightarrow$ blur $\rightarrow$ sample $\rightarrow$ decay) produces a smooth, continuous correction field without any linear solver.

Advantage: Eliminates the Voronoi discontinuities of nearest-neighbor propagation and the computational cost of Poisson solves. The Gaussian blur averages all nearby boundary corrections (rather than assigning the single nearest one), and the exponential decay provides a natural fade-to-zero envelope. The correction field is C^1-continuous and costs only $O(N/\gamma^2)$ for a grid of cell size γ.

Contribution 6: Self-Calibrating Distortion Fit from a Single Image Pair

Problem corrected: Brown-Conrady lens distortion calibration conventionally requires a known calibration target (checkerboard) photographed in multiple views, or bundle adjustment over many overlapping images. Neither is available when processing a single pair of drone photographs in the field without calibration infrastructure.

Innovation: The tiled NCC registration residuals, the displacement errors remaining after subtracting the global offset, are treated as implicit calibration data. Because the Brown-Conrady model is linear in the distortion coefficients (k_1, k_2, p_1, p_2) for a fixed optical center, the fit reduces to a single weighted least-squares solve. The optical center itself is found by exhaustive grid search over a coarse-then-fine grid, avoiding local-minimum traps in the non-convex center-distortion landscape.

Advantage: Calibrates the lens from a single image pair without calibration targets, GPS, or iterative nonlinear solvers. The grid search guarantees the global optimum for center location, and each WLS solve is $O(N_{\text{tiles}})$, negligible compared to the tile registration itself.

6.3 The Pipeline as a Novel Architecture

Beyond the six individual contributions, the pipeline architecture itself is novel in two respects:

Quality-signal cascading. Each stage produces not only a refined alignment but also a quality metric that the next stage consumes. The flow error drives the adaptive smoothing kernel; the flow magnitude drives the seam cost; the seam mask drives the Laplacian blend; the blend residual at the boundary drives the correction field. This cascaded quality-signal architecture differs fundamentally from the conventional detect-match-homography-blend pipeline, in which stages are largely independent.

Complete self-containment. The entire system, registration, calibration, flow estimation, seam finding, blending, and boundary correction, is implemented in a single application with no external libraries, no GPU dependencies, and no pretrained models. This makes the pipeline reproducible, auditable, and deployable in constrained environments (e.g., tactical edge computing) where library dependencies and GPU hardware may be unavailable.

7. Novel Contributions and Comparative Mathematics

While each individual technique in the pipeline draws on established methods, NCC [5], Lucas-Kanade [7], Brown-Conrady distortion [13, 14], Laplacian pyramid blending [9], dynamic-programming seam finding [10], several elements of this system are, to the authors' knowledge, novel in their specific formulation or in the way they are combined. This section identifies each novel contribution, states the conventional formulation it replaces, derives the replacement in the unified notation of Section 5.17, and explains why the substitution improves the result.

7.1 Dual-Initialization Iterative Flow Refinement

Conventional approach. Standard Lucas-Kanade optical flow estimation uses a single initialization per grid point, either zero or coarse-to-fine propagation from a pyramid level above [8]. The conventional iterative refinement at pass p is:

$$\mathbf{f}^{(8,p)}_{\text{conv}}(\mathbf{g}) = \text{LK}\Big(\mathbf{g},\ \mathbf{f}^{(8,p-1)}(\mathbf{g})\Big)$$

This re-runs LK from the previous estimate. If the previous estimate sits in a poor local minimum of the brightness constancy objective, the refinement returns the same minimum, there is no mechanism for escape.

Replacement here. In the master equation (Section 5.17.8), the flow at each grid point is instead determined by the dual-initialization operator:

$$\mathbf{f}^{(8,p)}(\mathbf{g}) = \begin{cases} \arg\min\limits_{\mathbf{f}^* \in \{\mathbf{f}_a,\, \mathbf{f}_b\}} e(\mathbf{g},\, \mathbf{f}^*) & \text{if } e(\mathbf{g},\, \mathbf{f}^{(8,p-1)}) > \tau_e \\ \mathbf{f}^{(8,p-1)}(\mathbf{g}) & \text{otherwise} \end{cases}$$

where:

$$\mathbf{f}_a = \text{LK}\Big(\mathbf{g},\ \mathbf{f}^{(8,p-1)}(\mathbf{g})\Big), \qquad \mathbf{f}_b = \text{LK}\left(\mathbf{g},\ \frac{1}{|\mathcal{N}_8|}\sum_{\mathbf{g}' \in \mathcal{N}_8(\mathbf{g})} \mathbf{f}^{(8,p-1)}(\mathbf{g}')\right)$$

The key structural difference is the $\arg\min$ over a candidate set of cardinality 2. The second candidate $\mathbf{f}_b$ is initialized from the 8-connected neighbor mean, which lies in a different basin of the LK cost surface whenever the current estimate is an outlier relative to its spatial context. This provides a per-pass, per-point mechanism for escaping local minima, effectively a competitive local search, without the cost of a full multi-start optimization.

Existing iterative refinement schemes in the literature, such as Iterative Residual Refinement (IRR) [Hur & Roth, 2019] and RAFT's recurrent update operator [Teed & Deng, 2020], operate on learned residual corrections within deep networks and do not employ an explicit dual-initialization competition at the classical LK level. The approach here is purely classical and requires no training data.

7.2 Error-Weighted Adaptive Spatial Smoothing

Conventional approach. Standard flow smoothing applies a spatially uniform or bilateral Gaussian kernel whose weights depend on image content (pixel intensity or gradient similarity). The Horn-Schunck formulation [Horn & Schunck, 1981] uses a global Laplacian regularizer, and bilateral smoothing uses:

$$\mathbf{f}_{\text{bilateral}}(\mathbf{g}) = \frac{\sum_{\mathbf{g}'} G(\|\mathbf{g}-\mathbf{g}'\|;\,\sigma_s)\;\cdot\;G(\|I(\mathbf{g})-I(\mathbf{g}')\|;\,\sigma_r)\;\cdot\;\mathbf{f}(\mathbf{g}')}{\sum_{\mathbf{g}'} G(\|\mathbf{g}-\mathbf{g}'\|;\,\sigma_s)\;\cdot\;G(\|I(\mathbf{g})-I(\mathbf{g}')\|;\,\sigma_r)}$$

The range kernel $G(\|I - I'\|;\sigma_r)$ preserves edges defined by image intensity, but has no knowledge of whether the flow at each neighbor is actually correct.

Replacement here. In the master equation, the smoothing kernel is instead weighted by the inverse of each neighbor's registration error (Section 5.17.8):

$$\mathbf{f}^{(8,p)}_{\text{smooth}}(\mathbf{g}) = \frac{\sum_{\mathbf{g}'\in\mathcal{N}_2(\mathbf{g})} 1_{e<\tau_e}(\mathbf{g}')\;\cdot\;G(\|\mathbf{g}-\mathbf{g}'\|;\,\sigma_s)\;\cdot\;\dfrac{1}{e(\mathbf{g}',\,\mathbf{f}^{(8,p)})}\;\cdot\;\mathbf{f}^{(8,p)}(\mathbf{g}')}{\sum_{\mathbf{g}'\in\mathcal{N}_2(\mathbf{g})} 1_{e<\tau_e}(\mathbf{g}')\;\cdot\;G(\|\mathbf{g}-\mathbf{g}'\|;\,\sigma_s)\;\cdot\;\dfrac{1}{e(\mathbf{g}',\,\mathbf{f}^{(8,p)})}}$$

Comparing the two formulations term by term:

Factor	Bilateral (conventional)	Ours (novel)
Spatial weight	$G(\|\mathbf{g}-\mathbf{g}'\|;\sigma_s)$	$G(\|\mathbf{g}-\mathbf{g}'\|;\sigma_s)$, identical
Range/quality weight	$G(\|I(\mathbf{g})-I(\mathbf{g}')\|;\sigma_r)$	$1_{e<\tau_e}(\mathbf{g}')\cdot e(\mathbf{g}',\mathbf{f})^{-1}$
What it preserves	Image edges (intensity boundaries)	Registration quality (low-error flow vectors)
Scope	All points globally	Only points with $e > \tau_e$ (selective)

The critical substitution is replacing the intensity-based range kernel with the inverse warp error $1/e$. This means the smoothing kernel is shaped by how well each neighbor's flow actually aligns the images, not by whether the underlying image content is similar. A neighbor with perfect alignment ($e \to 0$) dominates; a neighbor that is itself poorly matched contributes negligibly, regardless of image similarity. The hard gate $1_{e<\tau_e}$ further excludes neighbors above the error threshold, preventing bad flow from propagating.

7.3 Hierarchical Catmull-Rom Flow Upsampling

Conventional approach. The standard method for upsampling a sparse flow field to per-pixel resolution is direct bilinear interpolation from the coarsest grid:

$$\mathbf{f}^{(1)}_{\text{bilinear}}(x,y) = \sum_{i=0}^{1}\sum_{j=0}^{1} \mathbf{f}^{(8)}\left(\left\lfloor\frac{x}{8}\right\rfloor + i,\;\left\lfloor\frac{y}{8}\right\rfloor + j\right)\;\cdot\;w^{\text{blin}}_{ij}$$

where $w^{\text{blin}}_{ij} = (1-\alpha)(1-\beta),\;\alpha(1-\beta),\;(1-\alpha)\beta,\;\alpha\beta$ with $\alpha = x/8 - \lfloor x/8\rfloor$. This single-step $8\times$ upsampling produces only C^0 continuity at grid boundaries: the interpolated flow has kinks at every 8th pixel, manifesting as period-2 stripe artifacts in the mosaic (Section 4.1).

In learning-based systems (RAFT, UPFlow [Luo et al., 2021]), upsampling uses convex masks predicted by a neural network, which requires training data and GPU inference.

Replacement here. In the master equation (Section 5.17.8), the per-pixel flow is instead the result of three nested Catmull-Rom bicubic interpolations:

$$f_x^{(1)}(u_R, v_R) = \sum_{i_1=-1}^{2} \sum_{j_1=-1}^{2} h(\phi_x^{(2)} - i_1)\, h(\phi_y^{(2)} - j_1) \left[\sum_{i_2=-1}^{2} \sum_{j_2=-1}^{2} h(\phi_x^{(4)} - i_2)\, h(\phi_y^{(4)} - j_2) \left[\sum_{i_3=-1}^{2} \sum_{j_3=-1}^{2} h(\phi_x^{(8)} - i_3)\, h(\phi_y^{(8)} - j_3)\, f_x^{(8)}(g_{i_3}, g_{j_3}) \right] \right]$$

Comparing the two:

Property	Bilinear (conventional)	Hierarchical Catmull-Rom (ours)
Upsampling stages	1 ($8 \to 1$)	3 ($8 \to 4 \to 2 \to 1$)
Kernel support per stage	2×2	4×4
Continuity class	C^0 (kinks at grid lines)	C^1 (smooth first derivatives)
Interpolating?	Yes	Yes (Catmull-Rom passes through control points)
Training data required	No	No
Effective support at finest level	4 grid nodes	$4^3 = 64$ grid nodes (nested)

The C^1 continuity of the Catmull-Rom kernel eliminates the grid-boundary kinks that cause the period-2 artifacts. The hierarchical structure means each doubling stage sees a 4×4 neighborhood that is already smooth from the previous stage, preventing the accumulation of interpolation error that occurs when jumping directly from $8\times$ spacing to per-pixel.

7.4 Flow-Magnitude-Weighted Seam Cost

Conventional approach. Dynamic-programming seam finding for mosaicing [10, 11] typically uses color difference as the sole cost term, sometimes augmented by image gradient magnitude:

$$\text{cost}_{\text{conv}}(x, y) = \sum_{k \in \{r,g,b\}} \left(R^k(x, y) - T_w^k(x, y) \right)^2$$

This steers the seam toward regions where the two images happen to look similar, but it has no knowledge of whether the geometric alignment is trustworthy at that location. In regions of uniform color (e.g., a shadowed area under tree canopy), the color cost can be low even where the flow field is large and unreliable.

Replacement here. In the master equation (Section 5.17.8), the seam path $\mathcal{S}$ minimizes a cost that includes the flow-magnitude penalty:

$$\mathcal{S} = \arg\min_{\text{path}} \sum_t \left[\underbrace{\sum_{k \in \{r,g,b\}} \left(\tilde{c}_R^k(x_t, y_t) - \tilde{c}_T^k(x_t, y_t) \right)^2}_{\text{color cost (standard)}} + \underbrace{2 \left(f_x^2(x_t, y_t) + f_y^2(x_t, y_t) \right)}_{\text{flow-magnitude penalty (novel)}} + \underbrace{\left(\hat{c}_t^2 + \hat{c}_t^4 \right) \alpha}_{\text{center bias}} \right]$$

The novel second term, $2\|\mathbf{f}\|^2$, penalizes regions where the dense optical flow is large. Since flow magnitude is a direct measure of how much geometric correction was needed to align the images at that point, a large flow indicates either parallax, terrain relief, rolling-shutter distortion, or residual misalignment, all cases where the warp is least trustworthy and blending artifacts are most likely. This provides a registration-confidence signal that is entirely absent from conventional color-only seam costs.

Comparing:

Term	Conventional	Ours
Color difference	$\|R - T_w\|^2$	$\|\tilde{c}_R - \tilde{c}_T\|^2$, same, but uses source-filled colors
Geometric confidence	None	$2(f_x^2 + f_y^2)$, penalizes large flow corrections
Spatial prior	None or gradient-based	$(\hat{c}^2 + \hat{c}^4)\alpha$, center bias

7.5 Gaussian-Smoothed Boundary Correction Field

Conventional approach. The boundary between a blended overlap zone and a single-image zone is typically handled by one of two methods. The first is Poisson image editing [Pérez et al., 2003], which solves a large sparse linear system:

$$\nabla^2 \mathbf{c}_{\text{corrected}} = \nabla^2 \mathbf{c}_{\text{original}} \quad \text{subject to} \quad \mathbf{c}_{\text{corrected}}\big|_{\partial\Omega} = \mathbf{c}_{\text{blend}}\big|_{\partial\Omega}$$

The second is simply extending the feather zone, which degrades sharpness. The initial approach (Section 4.2) used BFS nearest-neighbor propagation, which assigns each non-overlap pixel the correction of the nearest boundary pixel:

$$\mathbf{C}_{\text{BFS}}(x_m, y_m) = \Delta\mathbf{c}\left(\arg\min_{\mathbf{p} \in \partial\Omega} \|\mathbf{p} - (x_m, y_m)\|\right)$$

This creates piecewise-constant Voronoi cells in the correction field. Where two boundary pixels with different corrections meet, a sharp discontinuity appears, manifesting as the vertical stripe artifacts documented in Section 4.2.

Replacement here. In the master equation (Section 5.17.7–5.17.8), the boundary correction $\mathbf{C}^k$ is a four-stage pipeline that replaces the nearest-neighbor assignment with a continuous field:

$$\mathbf{C}^k(x_m, y_m) = \underbrace{\mathcal{B}\Big[\hat{\Delta}\mathbf{c}^k\Big]\left(\frac{x_m}{\gamma} - \tfrac{1}{2},\ \frac{y_m}{\gamma} - \tfrac{1}{2}\right)}_{\text{bilinear sample from Gaussian-blurred grid}} \cdot \underbrace{\exp\left(\frac{-3\, d(x_m, y_m)}{d_{\max}}\right)}_{\text{exponential distance decay}}$$

where the blurred grid correction is:

$$\hat{\Delta}\mathbf{c}^k(g_x, g_y) = \frac{\displaystyle\sum_{i=-R}^{R}\sum_{j=-R}^{R} 1_w(g_x + i, g_y + j) \cdot G(i, j; \sigma_g) \cdot \Delta\mathbf{c}^k_{\text{grid}}(g_x + i, g_y + j)}{\displaystyle\sum_{i=-R}^{R}\sum_{j=-R}^{R} 1_w(g_x + i, g_y + j) \cdot G(i, j; \sigma_g)}$$

Comparing the three approaches:

Property	BFS nearest-neighbor (failed)	Poisson (conventional)	Scatter-blur-decay (ours)
Correction field continuity	C^{-1} (discontinuous)	C^2 (harmonic)	C^1 (Gaussian-smoothed)
Computational cost	$O(N)$ BFS	$O(N \log N)$ Poisson solve	$O(N/\gamma^2)$ grid blur
Requires linear solver	No	Yes (large sparse system)	No

Decay envelope	None (abrupt)	Dirichlet boundary	Exponential $e^{-3d/d_{\max}}$
Artifact mode	Voronoi stripes	None	None

The critical insight is that the BFS assigns a single boundary pixel's correction to each non-overlap pixel ($\arg\min$ over $\partial\Omega$), producing Voronoi cells. The replacement here averages all nearby boundary corrections through the Gaussian blur ($\sum$ weighted by G), producing a smooth field. The Voronoi operator is a degenerate limit of the Gaussian operator as $\sigma_g \to 0$: this approach generalizes nearest-neighbor propagation to continuous propagation.

7.6 Self-Calibrating Distortion Fit via Grid-Search-Initialized WLS

Conventional approach. Brown-Conrady distortion calibration [13, 14] normally requires a known calibration target. Given N correspondences between image points $\mathbf{x}_i$ and world points $\mathbf{X}_i$ on the target, the distortion parameters (k_1, k_2, p_1, p_2) and camera intrinsics are jointly optimized via Levenberg-Marquardt:

$$\min_{k_1,k_2,p_1,p_2,\mathbf{K},\mathbf{R},\mathbf{t}} \sum_{i=1}^{N} \left\| \mathbf{x}_i - \pi(\mathbf{K}, k_1, k_2, p_1, p_2, \mathbf{R}, \mathbf{t}, \mathbf{X}_i) \right\|^2$$

where π is the projection function and $\mathbf{K}$ is the intrinsic matrix. This requires the calibration target, multiple views, and an iterative nonlinear solver with good initialization.

Self-calibration via bundle adjustment [20] eliminates the target but requires multiple overlapping images and jointly solves for all camera poses, which is a much larger optimization problem.

Replacement here. The pipeline instead uses the tiled NCC registration residuals as implicit calibration data. At each valid tile i centered at image position (x_i, y_i), the residual displacement (r_x^i, r_y^i) after subtracting the global offset median is modeled as the differential Brown-Conrady distortion:

$$\begin{bmatrix} r_x^i \\ r_y^i \end{bmatrix} = \begin{bmatrix} x_n^i (k_1 r_i^2 + k_2 r_i^4) + p_1 (r_i^2 + 2(x_n^i)^2) + 2p_2 x_n^i y_n^i \\ y_n^i (k_1 r_i^2 + k_2 r_i^4) + 2p_1 x_n^i y_n^i + p_2 (r_i^2 + 2(y_n^i)^2) \end{bmatrix}$$

where (x_n^i, y_n^i) are the tile coordinates normalized to the candidate optical center. This is linear in the unknowns (k_1, k_2, p_1, p_2), so the fit is a single weighted least-squares solve (Section 5.5):

$$(\mathbf{A}^\top \mathbf{W} \mathbf{A})\, \mathbf{p} = \mathbf{A}^\top \mathbf{W} \mathbf{b}$$

The optical center (c_x, c_y) is found by exhaustive grid search (coarse 0.05, fine 0.01 steps) rather than gradient-based optimization, avoiding local-minimum traps in the non-convex center-distortion landscape.

Comparing:

Property	**Target-based (conventional)**	**Bundle adjustment [20]**	**Tile-residual WLS (ours)**
Calibration data	Known target pattern	Multiple overlapping images	Single image pair
Correspondences	Target $\leftrightarrow$ image	Feature $\leftrightarrow$ feature	Tile NCC residuals
Optimization	Nonlinear (LM)	Nonlinear (LM)	Linear WLS + grid search
Center estimation	Joint with LM	Joint with LM	Exhaustive grid search

GPS/IMU required	No	Helpful	No
Failure mode	None (controlled setup)	Drift, local minima	Insufficient tile coverage

The key advantage is that a single WLS solve at each candidate center costs $O(N_{\text{tiles}})$ and is guaranteed to find the global optimum for that center. The grid search over centers costs $O(M^2)$ WLS solves (where $M \approx 17$ for the coarse grid), but each is trivial. The total cost is negligible compared to the tile registration itself.

7.7 The Pipeline as a Whole

Perhaps the most significant contribution is the specific combination of stages into an end-to-end pipeline that is entirely self-contained (no external libraries, no GPU, no pretrained models) and self-calibrating (no calibration targets, no control points, no GPS metadata). Expressed in the master equation notation (Section 5.17.9), the output at each pixel is:

$$\mathbf{c}_{\text{out}}^{k}(x_m, y_m) = \mathcal{L}\Big[\, \tilde{\mathcal{B}}[R^k] \circ \pi_R,\;\; \tilde{\mathcal{B}}[T^k] \circ \pi_T \circ \Phi,\;\; m_{\mathcal{S}} \,\Big]\,(x_m, y_m) \;+\; (1 - 1_\Omega) \;\cdot\; \mathcal{E}_{\partial\Omega}^{k}(x_m, y_m)$$

Each operator in this composition feeds its successor with progressively refined information, and, critically, several operators consume quality signals produced by earlier stages:

– π_T uses the global offset (d_x, d_y) from NCC (Section 5.1), refined by tile median (Section 5.3)
– Φ applies the per-pixel flow $\mathbf{f}^{(1)}$, which is the hierarchical Catmull-Rom upsample of $\mathbf{f}^{(8)}$, which is the output of dual-init refinement with error-weighted smoothing
– $m_{\mathcal{S}}$ is determined by a DP seam whose cost depends on the flow magnitude $\|\mathbf{f}\|^2$, a quality signal from the flow estimation stage
– $\mathcal{L}$ blends the two sources across frequency bands guided by the seam mask
– $\mathcal{E}_{\partial\Omega}$ corrects the residual color jump that the blend cannot reach

This cascaded architecture, where each stage's output is both a refinement of alignment and a quality signal for the next stage, differs from the dominant paradigm in modern mosaicing, which relies on feature detection (SIFT/ORB), RANSAC homography estimation, and global bundle adjustment [Brown & Lowe, 2007]. The conventional pipeline can be written as:

$$\mathbf{c}_{\text{conv}}^{k}(x_m, y_m) = \mathcal{L}\big[\, R^k \circ \pi_R,\;\; T^k \circ \mathbf{H},\;\; m_{\text{Voronoi}} \,\big]\,(x_m, y_m)$$

where $\mathbf{H}$ is a single global homography estimated from SIFT/RANSAC, the seam mask m_{Voronoi} is a Voronoi partition with no confidence weighting, and no boundary correction is applied. The master equation replaces:

Conventional operator	Replacement here	Gain
$\mathbf{H}$ (global homography)	$\pi_T \circ \Phi$ (global offset + per-pixel flow)	Handles non-rigid deformation, parallax, rolling shutter
SIFT/RANSAC features	NCC tiles $\rightarrow$ affine LK $\rightarrow$ dense LK	No feature detection; robust to repetitive texture
Single-step bilinear upsample	3-stage Catmull-Rom hierarchy	C^1 flow field; no period-2 artifacts
Color-only seam cost	Color + $2\|\mathbf{f}\|^2$ + center bias	Seam avoids regions of unreliable warp

No boundary correction	Scatter-blur-decay $\mathcal{E}_{\partial\Omega}$	Seamless overlap-to-non-overlap transition
External calibration	Tile-residual WLS + grid search	Self-calibrating from single pair

8. Future Work

8.1 Multi-Image Mosaicing ($N > 2$)

The current system operates on exactly two images. Extension to $N > 2$ images requires: (a) pairwise registration of all overlapping pairs, (b) global bundle adjustment [20] to jointly optimize all camera parameters and minimize cumulative drift, (c) a global seam-finding strategy (e.g., Voronoi-based label assignment with graph-cut optimization [11]) to partition the output canvas among N sources, and (d) multi-image Laplacian pyramid blending with per-pixel source selection. The Brown-Conrady distortion calibration generalizes naturally to multi-image scenarios: the same distortion parameters apply to all images from the same camera, providing a strong cross-image constraint. Bundle adjustment could employ Levenberg-Marquardt optimization over the joint set of global offsets, per-pair flow fields, and the shared distortion model.

8.2 Wide Spectral Range Matching

Matching images from different spectral bands (e.g., visible RGB against near-infrared, thermal infrared, or synthetic aperture radar) requires replacing the NCC metric, which assumes correlated brightness, with a modality-invariant similarity measure. Mutual information (MI), which measures the statistical dependence between intensity distributions, has been shown effective for multi-modal registration [21]. Alternatively, structural descriptors such as the Modality-Independent Neighborhood Descriptor (MIND) or learned feature embeddings from cross-spectral matching networks could serve as the matching cost in both the tile-registration and dense-flow stages. The Laplacian pyramid blending would require adaptation to handle different dynamic ranges and radiometric responses across bands.

8.3 Integration into SIGINT Workflows

Aerial image mosaics provide the spatial reference layer onto which signals intelligence products are geo-registered. Integration into a SIGINT workflow would involve: (a) geotagging each mosaic pixel using the drone's GPS/IMU telemetry and the computed registration transforms, producing an orthorectified georeferenced mosaic; (b) fusing the mosaic with RF sensor geolocation data (e.g., TDoA emitter positions from onboard COMINT/ELINT payloads [3]) by projecting emitter coordinates onto the mosaic canvas; (c) enabling temporal mosaicing for change detection and pattern-of-life analysis; and (d) implementing real-time incremental mosaicing for onboard processing on edge computing platforms, potentially leveraging GPU-accelerated flow computation. The pipeline's self-calibrating nature (no external calibration targets required) makes it suitable for field deployment where calibration infrastructure is unavailable.

Further integration could include automated object detection and geotagging (applying convolutional neural networks to the mosaic for vehicle, structure, or anomaly detection), mosaic-based GMTI (ground moving target indication) by differencing temporally adjacent mosaics, and multi-INT fusion where IMINT mosaics, SIGINT geolocation ellipses, and HUMINT report locations are composited into a unified common operating picture.

9. Final Notes

The most instructive lesson from the development process was that visible artifacts (vertical stripe patterns) often have root causes far removed from where they manifest. The stripes appeared in the final mosaic and were initially attributed to the flow field, the pyramid blending, or the seam finding. Extensive diagnostic instrumentation, including per-stage pipeline image exports, flow field visualizations, and selective disabling of pipeline components, eventually traced the cause to the boundary correction field and its Voronoi-cell discontinuity structure. The fix was straightforward

(Gaussian blur of a scattered correction grid), but identifying the root cause required systematic elimination of hypotheses across a 10-stage pipeline.

The combination of multi-scale NCC registration, iterative affine Lucas-Kanade refinement, dense optical flow with hierarchical upsampling, seam-guided Laplacian pyramid blending, and Gaussian-smoothed boundary correction produces artifact-free mosaics from uncalibrated consumer drone imagery without requiring external calibration targets, control points, or GPS metadata.

10. References

[1] Polidori, L. (2020). "On Laussedat's Contribution to the Emergence of Photogrammetry." *ISPRS Archives*, XLIII-B2-2020, pp. 893–898. https://isprs-archives.copernicus.org/articles/XLIII-B2-2020/893/2020/

[2] Szeliski, R. (2006). "Image Alignment and Stitching: A Tutorial." *Foundations and Trends in Computer Graphics and Vision*, vol. 2, no. 1, pp. 1–104.

[3] Defense Systems Information Analysis Center (DSIAC). "Unmanned Aerial Systems (UAS) for Intelligence, Surveillance and Reconnaissance (ISR)." DTIC Technical Report. https://dsiac.dtic.mil/wp-content/uploads/2018/05/UNMANNED-AERIAL-SYSTEMS-UAS-FOR-INTELLIGENCE-SURVEILLANCE-AND-RECONNAISSANCE-ISR.pdf

[4] Pratt, W.K. (1974). "Correlation Techniques of Image Registration." *IEEE Trans. Aerospace and Electronic Systems*, vol. 10, no. 3, pp. 353–358.

[5] Lewis, J.P. (1995). "Fast Normalized Cross-Correlation." *Vision Interface*, pp. 120–123. https://scribblethink.org/Work/nvisionInterface/nip.pdf

[6] Lowe, D.G. (2004). "Distinctive Image Features from Scale-Invariant Keypoints." *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110.

[7] Lucas, B.D. and Kanade, T. (1981). "An Iterative Image Registration Technique with an Application to Stereo Vision." *Proc. 7th IJCAI*, pp. 674–679.

[8] Bouguet, J.-Y. (2001). "Pyramidal Implementation of the Lucas Kanade Feature Tracker." Intel Corporation Technical Report. http://robots.stanford.edu/cs223b04/algo_tracking.pdf

[9] Burt, P.J. and Adelson, E.H. (1983). "A Multiresolution Spline with Application to Image Mosaics." *ACM Trans. Graphics*, vol. 2, no. 4, pp. 217–236.

[10] Efros, A.A. and Freeman, W.T. (2001). "Image Quilting for Texture Synthesis and Transfer." *Proc. SIGGRAPH 2001*, pp. 341–346.

[11] Kwatra, V., Schödl, A., Essa, I., Turk, G., and Bobick, A. (2003). "Graphcut Textures: Image and Video Synthesis Using Graph Cuts." *ACM Trans. Graphics*, vol. 22, no. 3, pp. 277–286.

[12] Conrady, A.E. (1919). "Decentred Lens-Systems." *Monthly Notices of the Royal Astronomical Society*, vol. 79, pp. 384–390.

[13] Brown, D.C. (1966). "Decentering Distortion of Lenses." *Photogrammetric Engineering*, vol. 32, no. 3, pp. 444–462.

[14] Brown, D.C. (1971). "Close Range Camera Calibration." *Photogrammetric Engineering*, vol. 37, no. 8, pp. 855–866.

[15] Shepard, D. (1968). "A Two-Dimensional Interpolation Function for Irregularly-Spaced Data." *Proc. 23rd ACM National Conference*, pp. 517–524.

[16] Catmull, E. and Rom, R. (1974). "A Class of Local Interpolating Splines." In *Computer Aided Geometric Design*, Academic Press, pp. 317–326.

[17] Fisher, R.B. and Naidu, D.K. (1996). "A Comparison of Algorithms for Subpixel Peak Detection." In *Image Technology: Advances in Image Processing, Multimedia and Machine Vision*, Springer, pp. 385–404.

[18] Hampel, F.R. (1974). "The Influence Curve and Its Role in Robust Estimation." *Journal of the American Statistical Association*, vol. 69, no. 346, pp. 383–393.

[19] Rousseeuw, P.J. and Croux, C. (1993). "Alternatives to the Median Absolute Deviation." *Journal of the American Statistical Association*, vol. 88, no. 424, pp. 1273–1283.

[20] Brown, M. and Lowe, D.G. (2007). "Automatic Panoramic Image Stitching using Invariant Features." *International Journal of Computer Vision*, vol. 74, no. 1, pp. 59–73. https://www.cs.ubc.ca/~lowe/papers/07brown.pdf

[21] Viola, P. and Wells, W.M. (1997). "Alignment by Maximization of Mutual Information." *International Journal of Computer Vision*, vol. 24, no. 2, pp. 137–154.