Supplementary Material for Geometry-Aware Super-Resolution Fusion Calibration for Binocular Structured Light 3D Reconstruction

Hongyan Cao,[1,2,3] Dayong Qiao,[1,2,*] Mengya Han,[4] Wangke Yu,[3] Benquan Wang,[3,†] and Yijie Shen[3,5,‡]

[1]*Ministry of Education Key Laboratory of Micro/Nano Systems for Aerospace, Key Laboratory of Micro- and Nano-Electro-Mechanical Systems of Shaanxi Province, School of Mechanical Engineering, Northwestern Polytechnical University, 127 Youyi West Road, Xi'an, 710072, China.*
[2]*Ningbo Institute, Northwestern Polytechnical University, Ningbo, Zhejiang, 315103, China.*
[3]*Centre for Disruptive Photonic Technologies, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637371, Republic of Singapore*
[4]*National Engineering Research Center for Multimedia Software, School of Computer Science, Wuhan University, and Hubei Luojia Laboratory, Wuhan, 430071, China.*
[5]*School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore*

(Dated: December 30, 2025)

Supplementary Material Outline

S1. Coordinate Systems and Notation

We briefly summarize the coordinate systems and symbols used in the main paper.

World coordinate system $\{W\}$. A 3D point in the scene or on the checkerboard plane is denoted by $\mathbf{P}_W = [X_W, Y_W, Z_W]^\top$. When the checkerboard is placed on the plane $Z_W = 0$, all internal corners and cell centers lie on a regular 2D grid in $\{W\}$.

Camera coordinate systems $\{C^L\}$ *and* $\{C^R\}$. For each camera $s \in \{L, R\}$, we denote camera-centered coordinates by $\mathbf{P}_C^s = [X_C^s, Y_C^s, Z_C^s]^\top$. The rigid transformation from world to camera coordinates is

$$\mathbf{P}_C^s = \mathbf{R}^s \mathbf{P}_W + \mathbf{t}^s, \tag{1}$$

where $\mathbf{R}^s \in \mathrm{SO}(3)$ and $\mathbf{t}^s \in \mathbb{R}^3$ are the extrinsic parameters of camera s.

Normalized image coordinate system $\{r^s\}$. We use a virtual imaging plane at $Z_C^s = 1$ to represent undistorted perspective projection. The normalized image coordinates are defined as

$$\mathbf{P}_r^s = \begin{bmatrix} x^s \\ y^s \end{bmatrix} = \begin{bmatrix} X_C^s / Z_C^s \\ Y_C^s / Z_C^s \end{bmatrix}. \tag{2}$$

Pixel coordinate system $\{v^s\}$. The discrete pixel coordinates on the sensor are denoted by $\mathbf{P}_v^s = [u^s, v^s]^\top$. After applying radial and tangential distortion $\mathcal{D}(\cdot; \boldsymbol{k}^s)$ to the normalized coordinates (x^s, y^s), the mapping to pixel coordinates is

$$\begin{bmatrix} u^s \\ v^s \\ 1 \end{bmatrix} = \mathbf{K}^s \begin{bmatrix} x_d^s \\ y_d^s \\ 1 \end{bmatrix}, \quad (x_d^s, y_d^s) = \mathcal{D}(x^s, y^s; \boldsymbol{k}^s), \tag{3}$$

where $\mathbf{K}^s$ is the intrinsic matrix and $\boldsymbol{k}^s$ denotes the distortion parameters.

Checkerboard corners. Let $\mathbf{Q}_W^{(i,j)}$ denote the 3D coordinates of the (i, j)-th internal checkerboard corner on the plane $Z_W = 0$. In the n-th image of camera s, its sub-pixel pixel coordinate is written as

$$\hat{\mathbf{q}}_v^{s,(n,i,j)} = \left[u^{s,(n,i,j)},\, v^{s,(n,i,j)}\right]^\top. \tag{4}$$

Checkerboard cell centroids. We treat each checkerboard cell as a geometric primitive with an unbiased centroid. The world-space center of the k-th cell is denoted by $\mathbf{C}_W^{(k)} = [X_W^{(k)}, Y_W^{(k)}, 0]^\top$. In the n-th image of camera s, the geometric centroid computed from its four sub-pixel corners is denoted by

$$\hat{\mathbf{c}}_v^{s,(n,k)} = \left[u_c^{s,(n,k)},\, v_c^{s,(n,k)}\right]^\top, \tag{5}$$

while the corresponding projected centroid predicted by the calibration model is

$$\tilde{\mathbf{c}}_v^{s,(n,k)} = \Pi\big(\mathbf{K}^s, \boldsymbol{k}^s, \mathbf{R}^s, \mathbf{t}^s, \mathbf{C}_W^{(k)}\big), \tag{6}$$

where $\Pi(\cdot)$ denotes the full projection and distortion mapping from $\{W\}$ to $\{v^s\}$.

For completeness, we also summarize the notation used later for the reprojection and centroid errors in the two-stage calibration, but omit them here for brevity in the main paper.

S2. Implementation Details of Geometry-Aware Checkerboard Super-Resolution

This section provides additional details including the preprocessing pipeline, homography estimation, mask generation, and the architecture and losses of the super-resolution (SR) module.

S2.1 Harris-Based Corner Seeding and Preprocessing

Given a raw calibration image, we first apply lightweight preprocessing to improve robustness under varying illumination and noise. Specifically, we use a bilateral filter for denoising, followed by contrast normalization in the checkerboard region using CLAHE. The preprocessed intensity image is denoted by $I(x, y)$.

We then compute the Harris corner response[?] . For each pixel, the second-moment matrix over a local window $w(x, y)$ is defined as

$$\mathbf{M}(x,y) = \sum_{(u,v)\in\Omega(x,y)} w(u,v) \begin{bmatrix} I_x^2(u,v) & I_x(u,v)I_y(u,v) \\ I_x(u,v)I_y(u,v) & I_y^2(u,v) \end{bmatrix}, \tag{7}$$

where I_x, I_y are image gradients and $\Omega(x, y)$ denotes a local neighborhood. The Harris response is

$$R(x,y) = \det(\mathbf{M}) - \kappa \,\mathrm{trace}^2(\mathbf{M}), \tag{8}$$

with κ typically set in $[0.04, 0.06]$. We keep pixels with $R(x, y)$ above a fraction τ_R of the global maximum and apply non-maximum suppression in a $N_{\mathrm{NMS}} \times N_{\mathrm{NMS}}$ window to obtain a sparse set of robust corner hypotheses.

These Harris detections are not used directly for calibration, but serve as geometric seeds to (i) estimate the checkerboard homography and (ii) derive the global ROI and per-cell masks. All parameters (κ, window sizes, thresholds) are empirically chosen and fixed across all experiments.

S2.2 Homography Estimation Between Ideal Grid and Image

Let $\mathbf{p}_g = [i, j, 1]^\top$ denote the homogeneous coordinate of an ideal internal grid point at integer row–column index (i, j) on the checkerboard, with $i \in \{0, \ldots, W-1\}$ and $j \in \{0, \ldots, H-1\}$. The corresponding pixel coordinate in the image plane is denoted by $\mathbf{p}_v = [u, v, 1]^\top$ in the pixel coordinate system $\{v^s\}$.

We assume a planar checkerboard and model the mapping between the ideal grid and the distorted image by a homography

$$\mathbf{p}_v \sim \mathbf{H}\,\mathbf{p}_g, \tag{9}$$

where $\mathbf{H} \in \mathbb{R}^{3\times3}$ and $\sim$ denotes equality up to scale. To estimate $\mathbf{H}$, we proceed as follows:

1. Cluster the Harris detections into a dominant lattice using RANSAC-based line fitting and estimate the main orientation of the checkerboard.
2. Project the detections onto two dominant directions to obtain approximate row and column indices $(\tilde{i}, \tilde{j})$ via 1D peak detection and spacing analysis.
3. Establish tentative correspondences between detected points and integer grid positions (i, j) by a nearest-neighbor assignment in the 2D index space.
4. Use the resulting correspondences $\{(\mathbf{p}_g^{(n)}, \mathbf{p}_v^{(n)})\}$ to estimate $\mathbf{H}$ with a DLT solver, refined by non-linear least squares under a Sampson error, while rejecting outliers with RANSAC.

The estimated homography $\mathbf{H}$ is subsequently used to (i) infer the full set of grid vertices, (ii) obtain a tight global ROI, and (iii) generate per-cell polygon masks.

S2.3 Global ROI and Per-Cell Mask Generation

Given the homography $\mathbf{H}$ in Eq. (9), we can project all grid vertices into the image:

$$\mathbf{p}_v^{(i,j)} \sim \mathbf{H} \begin{bmatrix} i \\ j \\ 1 \end{bmatrix}, \quad i \in \{0, \ldots, W\},\ j \in \{0, \ldots, H\}. \tag{10}$$

Let $(u^{(i,j)}, v^{(i,j)})$ denote the inhomogeneous pixel coordinates after normalization.

Global checkerboard ROI mask. We form the convex hull of all projected vertices $\{\mathbf{p}_v^{(i,j)}\}$ and rasterize it to obtain a binary mask $M_{\mathrm{ROI}}(u, v)$. To account for small homography errors and blurring, we apply a morphological dilation with a radius of 1–2 pixels.

Internal corner mask. For each internal grid intersection (i, j), $i \in \{1, \ldots, W-1\}$, $j \in \{1, \ldots, H-1\}$, we create a small disk or square neighborhood around $\mathbf{p}_v^{(i,j)}$, forming an internal corner mask $M_{\mathrm{corner}}(u, v)$. This mask emphasizes regions where saddle-point corner fitting will be performed.

Per-cell polygon mask. For each checkerboard cell indexed by k with four vertices (i, j), $(i+1, j)$, $(i+1, j+1)$, and $(i, j+1)$, we obtain the corresponding quadrilateral in the image:

$$\mathcal{Q}_k = \left\{\mathbf{p}_v^{(i,j)}, \mathbf{p}_v^{(i+1,j)}, \mathbf{p}_v^{(i+1,j+1)}, \mathbf{p}_v^{(i,j+1)}\right\}. \tag{11}$$

We rasterize each quadrilateral into a binary mask $M_{\mathrm{cell}}^{(k)}(u, v)$ using standard polygon filling. These per-cell masks are later used to compute geometric centroids and to inject cell-level priors into the SR network.

For binocular data, we repeat the same procedure independently for the left and right images, and optionally enforce a weak consistency by checking that the projected

grid layout in both views is compatible with the stereo geometry. This step is purely for robustness and does not change the formulation.

S2.4 Super-Resolution Network Architecture

We adopt an IPG-style reconstruction network[?] as our SR backbone, which is conceptually similar to residual CNN-based SR models. In all experiments, we use a fixed upsampling factor of $\times 4$.

Inputs and outputs. For each image, we crop the bounding box of the global ROI mask M_{ROI} and feed a multi-channel tensor to the network:

$$\mathcal{X} = \big[I_{\mathrm{crop}},\ M_{\mathrm{ROI}},\ M_{\mathrm{corner}},\ \sum_k M_{\mathrm{cell}}^{(k)}\big], \tag{12}$$

where I_{crop} is the cropped intensity or RGB patch and the masks are resized to the same resolution. All masks are normalized to $[0, 1]$. The network outputs a super-resolved patch $\hat{I}_{\mathrm{SR}}$ with resolution $\times 4$ in each dimension.

Network structure. The SR backbone consists of an initial convolution, a stack of residual blocks with skip connections, and a pixel-shuffle upsampling module. The mask channels are concatenated with the image at the input layer, and optionally concatenated again at intermediate stages as a form of spatial prior. We do not introduce any checkerboard-specific operators; the geometry awareness is entirely encoded by the mask channels and the geometric loss described below.

Specific convolution kernel sizes, number of channels, and block counts are fixed across all experiments and are omitted here for brevity.

S2.5 Loss Functions and Geometric Consistency Term

The SR network is trained using paired low-resolution and pseudo ground-truth high-resolution checkerboard patches. The overall loss function is

$$\mathcal{L} = \lambda_{\mathrm{rec}}\,\mathcal{L}_{\mathrm{rec}} + \lambda_{\mathrm{geo}}\,\mathcal{L}_{\mathrm{geo}}, \tag{13}$$

where λ_{rec} and λ_{geo} control the trade-off between reconstruction fidelity and geometric consistency.

Reconstruction loss. We adopt a standard pixel-wise reconstruction loss between the super-resolved output $\hat{I}_{\mathrm{SR}}$ and the reference high-resolution patch I_{HR}:

$$\mathcal{L}_{\mathrm{rec}} = \big\|\hat{I}_{\mathrm{SR}} - I_{\mathrm{HR}}\big\|_1. \tag{14}$$

In some experiments we also include a perceptual loss on features extracted by a shallow CNN, but we observe that the simple ℓ_1 loss is already sufficient for calibration.

Geometric consistency loss. To encourage the SR output to align with the ideal checkerboard geometry, we introduce a lightweight geometric consistency term based on the homography $\mathbf{H}$. For each grid line in the ideal domain (horizontal or vertical), we sample a set of points $\{\mathbf{p}_g^{(\ell)}\}$ and project them into the SR image using Eq. (9), obtaining $\{\mathbf{p}_v^{(\ell)}\}$ at the high-resolution scale. We then penalize deviations between these projected lines and the intensity gradients of $\hat{I}_{\mathrm{SR}}$:

$$\mathcal{L}_{\mathrm{geo}} = \frac{1}{|\mathcal{L}|}\sum_{\ell\in\mathcal{L}}\Big(1 - \big\|\nabla_{\perp}\hat{I}_{\mathrm{SR}}(\mathbf{p}_v^{(\ell)})\big\|_2\Big)_+, \tag{15}$$

where $\nabla_{\perp}$ denotes the image gradient component along the normal direction of the ideal grid line at $\mathbf{p}_v^{(\ell)}$, $(\cdot)_+$ is the hinge function, and $\mathcal{L}$ is the set of sampled points. Intuitively, this loss encourages sharp intensity transitions orthogonal to the projected grid lines, leading to straighter and better localized checkerboard edges in the SR output.

In practice, we approximate $\nabla_{\perp}\hat{I}_{\mathrm{SR}}$ using simple Sobel filters and evaluate the loss only inside the global ROI mask to reduce computation. The geometric consistency loss is kept relatively small ($\lambda_{\mathrm{geo}} \ll \lambda_{\mathrm{rec}}$) so that it acts as a gentle regularizer rather than dominating the reconstruction.

Overall, the geometry-aware SR module enhances the contrast and sharpness of checkerboard edges and corners while preserving their global projective layout, which is crucial for the subsequent saddle-point-based corner localization and unbiased cell centroid estimation described in the main paper.

S3 Supplementary Derivations for Saddle-Point Corners and Unbiased Cell Centroids

This section provides detailed derivations for the saddle-point corner refinement and the unbiased cell centroid construction used in the main paper. For clarity, we temporarily omit the camera index $s \in \{L, R\}$, image index n and cell index k when there is no ambiguity.

S3.1 Second-Order Saddle-Point Corner Model

We work on a super-resolved checkerboard image patch $\hat{I}_{\mathrm{SR}}$ in the pixel coordinate system $\{v^s\}$. For each internal corner, an initial sub-pixel estimate

$$\hat{\mathbf{q}}_{v,0} = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \tag{16}$$

is obtained using a standard detector (Harris + sub-pixel refinement). Around this point we define a local coordinate system

$$u = u_0 + x, \quad v = v_0 + y, \tag{17}$$

and approximate the local intensity by a quadratic function

$$\hat{I}_{\mathrm{SR}}(x, y) \approx ax^2 + bxy + cy^2 + dx + ey + f, \tag{18}$$

where the coefficients $\{a, b, c, d, e, f\}$ are estimated by least squares over a small window centered at $(0, 0)$.

The gradient and Hessian of (18) are

$$\nabla \hat{I}_{\mathrm{SR}}(x, y) = \begin{bmatrix} \partial_x \hat{I}_{\mathrm{SR}} \\ \partial_y \hat{I}_{\mathrm{SR}} \end{bmatrix} = \begin{bmatrix} 2ax + by + d \\ bx + 2cy + e \end{bmatrix}, \tag{19}$$

$$\mathbf{H} = \nabla^2 \hat{I}_{\mathrm{SR}} = \begin{bmatrix} 2a & b \\ b & 2c \end{bmatrix}. \tag{20}$$

A checkerboard corner is modeled as a saddle point of this quadratic surface: the gradient vanishes and the Hessian has eigenvalues of opposite signs. The saddle point (x^*, y^*) satisfies

$$\nabla \hat{I}_{\mathrm{SR}}(x^*, y^*) = \mathbf{0}, \quad \lambda_1(\mathbf{H}) \cdot \lambda_2(\mathbf{H}) < 0. \tag{21}$$

The stationary point is obtained by solving the linear system

$$\mathbf{H} \begin{bmatrix} x^* \\ y^* \end{bmatrix} = - \begin{bmatrix} d \\ e \end{bmatrix}, \tag{22}$$

which yields

$$\begin{bmatrix} x^* \\ y^* \end{bmatrix} = - \mathbf{H}^{-1} \begin{bmatrix} d \\ e \end{bmatrix} = \frac{1}{4ac - b^2} \begin{bmatrix} be - 2cd \\ bd - 2ae \end{bmatrix}, \tag{23}$$

provided that $4ac - b^2 \neq 0$. The discriminant of the Hessian is

$$\det(\mathbf{H}) = 4ac - b^2, \tag{24}$$

and the eigenvalues are

$$\lambda_{1,2}(\mathbf{H}) = (a + c) \pm \sqrt{(a - c)^2 + b^2}. \tag{25}$$

The saddle-point condition in (21) is equivalent to

$$\det(\mathbf{H}) < 0 \quad \Longleftrightarrow \quad 4ac - b^2 < 0, \tag{26}$$

ensuring that one eigenvalue is positive and the other is negative.

The refined corner in the pixel coordinate system $\{v^s\}$ is thus

$$\hat{\mathbf{q}}_v = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u_0 + x^* \\ v_0 + y^* \end{bmatrix}, \tag{27}$$

In practice, we reject candidates for which the fitted Hessian does not satisfy the saddle condition or the local quadratic fit has insufficient support.

S3.2 Polygon Moments of Undistorted Quadrilaterals

We now derive the polygon moments of a checkerboard cell in the undistorted normalized plane $\{r^s\}$, which will be used to construct an unbiased centroid under radial distortion.

Quadrilateral in the normalized plane. Consider a single cell on the normalized plane, bounded by four vertices

$$\mathbf{v}_\ell = (x_\ell, y_\ell)^\top, \quad \ell = 1, \ldots, 4, \tag{28}$$

in counter-clockwise order, with $\mathbf{v}_5 = \mathbf{v}_1$. These vertices are obtained by mapping the refined pixel corners $\hat{\mathbf{q}}_v$ from $\{v^s\}$ to the normalized plane $\{r^s\}$ using the camera intrinsics and current distortion parameters.

We define the oriented edge term

$$\Delta_\ell = x_\ell y_{\ell+1} - x_{\ell+1} y_\ell, \quad \ell = 1, \ldots, 4. \tag{29}$$

The signed area of the quadrilateral A_n is

$$|A_n| = \frac{1}{2} \sum_{\ell=1}^{4} \Delta_\ell, \tag{30}$$

where the sign encodes the orientation (we assume counter-clockwise so that $|A_n| > 0$). Using the standard polygon formulas, the first-order moments are

$$\int_{A_n} x \, dA = \frac{1}{6} \sum_{\ell=1}^{4} (x_\ell + x_{\ell+1}) \, \Delta_\ell, \tag{31}$$

$$\int_{A_n} y\,dA = \frac{1}{6}\sum_{\ell=1}^{4}(y_\ell + y_{\ell+1})\,\Delta_\ell. \tag{32}$$

The centroid of the undistorted quadrilateral on the normalized plane is therefore

$$\bar{x}_n = \frac{1}{6|A_n|}\sum_{\ell=1}^{4}(x_\ell+x_{\ell+1})\,\Delta_\ell, \quad \bar{y}_n = \frac{1}{6|A_n|}\sum_{\ell=1}^{4}(y_\ell+y_{\ell+1})\,\Delta_\ell. \tag{33}$$

Higher-order monomial moments. For the radial distortion model, we need to integrate monomials $x^a y^b$ over the quadrilateral. We define the *normalized* monomial moments

$$M_n^{a,b} = \frac{1}{|A_n|}\int_{A_n} x^a y^b\,dA. \tag{34}$$

Using Green's theorem, integrals of the form $\int_{A_n} x^a y^b dA$ can be reduced to sums over the polygon edges:

$$\int_{A_n} x^a y^b\,dA = \frac{1}{a+b+2}\sum_{\ell=1}^{4}\left(x_{\ell+1}^{a+1}y_{\ell+1}^{b+1} - x_\ell^{a+1}y_\ell^{b+1}\right), \tag{35}$$

for non-negative integers a, b.[1] Dividing by $|A_n|$ yields $M_n^{a,b}$ as in (34).

In our implementation, we precompute and cache $M_n^{a,b}$ for all (a, b) required by the chosen polynomial distortion degree; the explicit formulas are straightforward but lengthy, and are omitted here for brevity.

S3.3 Moment-Based Centroid Under Polynomial Radial Distortion

We now derive the centroid of the distorted region $A_d = \mathcal{D}(A_n)$ in the normalized plane, under the polynomial radial distortion model used in the main paper.

Radial distortion mapping and Jacobian. We work in the normalized coordinate system $\{r^s\}$, with

$$\mathbf{p}_r = (x, y)^\top, \quad \mathbf{p}_d = (x_d, y_d)^\top. \tag{36}$$

The radial distortion is modeled as

$$\mathbf{p}_d = k(s)\,\mathbf{p}_r, \quad s = x^2 + y^2, \quad k(s) = \sum_{i=0}^{n_d} d_i s^i, \tag{37}$$

where d_i are distortion coefficients. The Jacobian of $\mathbf{p}_d$ with respect to $\mathbf{p}_r$ is

$$\mathbf{J}_\mathcal{D}(\mathbf{p}_r) = \frac{\partial \mathbf{p}_d}{\partial \mathbf{p}_r} = k(s)\,\mathbf{I}_2 + 2k'(s)\begin{bmatrix} x^2 & xy \\ xy & y^2 \end{bmatrix}, \tag{38}$$

where $k'(s)$ is the derivative of $k(s)$ with respect to s. Its determinant is

$$\det \mathbf{J}_\mathcal{D}(\mathbf{p}_r) = k(s)\big(k(s) + 2s\,k'(s)\big), \quad s = x^2 + y^2. \tag{39}$$

Area and centroid of the distorted region. Let A_n be the undistorted quadrilateral on the normalized plane and $A_d = \mathcal{D}(A_n)$ its distorted image. Using the change-of-variables formula, the area of A_d is

$$\begin{aligned} |A_d| &= \int_{A_d} dA_d = \int_{A_n} \det \mathbf{J}_\mathcal{D}(\mathbf{p}_r)\,dA \\ &= \int_{A_n} k(s)\big(k(s) + 2s\,k'(s)\big)\,dA. \end{aligned} \tag{40}$$

The centroid of A_d is defined as

$$\bar{\mathbf{p}}_d = \frac{1}{|A_d|}\int_{A_d} \mathbf{p}_d\,dA_d. \tag{41}$$

Using the mapping $\mathbf{p}_d = k(s)\mathbf{p}_r$ and the Jacobian determinant (39), we obtain

$$\begin{aligned} \int_{A_d} \mathbf{p}_d\,dA_d &= \int_{A_n} k(s)\,\mathbf{p}_r\,\det \mathbf{J}_\mathcal{D}(\mathbf{p}_r)\,dA \\ &= \int_{A_n} k(s)^2\big(k(s) + 2s\,k'(s)\big)\,\mathbf{p}_r\,dA. \end{aligned} \tag{42}$$

Hence

$$\bar{\mathbf{p}}_d = \frac{\displaystyle\int_{A_n} k(s)^2\big(k(s) + 2s\,k'(s)\big)\,\mathbf{p}_r\,dA}{\displaystyle\int_{A_n} k(s)\big(k(s) + 2s\,k'(s)\big)\,dA}. \tag{43}$$

Expansion in terms of polygon moments. Because $k(s)$ is a polynomial in $s = x^2 + y^2$, and $(x^2 + y^2)^r$ can be expanded as a finite sum of monomials $x^a y^b$, both numerator and denominator of (43) can be expressed as finite linear combinations of the moments $\int_{A_n} x^a y^b dA$, and thus of the normalized moments $M_n^{a,b}$ in (34).

More concretely, let

$$k(s) = \sum_{i=0}^{n_d} d_i s^i, \quad k(s) + 2s\,k'(s) = \sum_{i=0}^{n_d} \tilde{d}_i s^i, \tag{44}$$

where $\tilde{d}_i$ are coefficients determined by $\{d_i\}$. The product $k(s)\big(k(s) + 2s\,k'(s)\big)$ and $k(s)^2\big(k(s) + 2s\,k'(s)\big)$ are polynomials in s of finite degree. Expanding $s^r = (x^2 + y^2)^r$ as

$$s^r = \sum_{a+b=2r} c_{a,b}^{(r)} x^a y^b, \tag{45}$$

we can write

$$|A_d| = |A_n| \sum_{a,b} w_{0,ab} M_n^{a,b}, \tag{46}$$

$$\begin{aligned} \int_{A_d} x_d\,dA_d &= |A_n| \sum_{a,b} w_{x,ab}\,M_n^{a,b}, \\ \int_{A_d} y_d\,dA_d &= |A_n| \sum_{a,b} w_{y,ab}\,M_n^{a,b}. \end{aligned} \tag{47}$$

for some distortion-dependent coefficients $w_{0,ab}, w_{x,ab}, w_{y,ab}$ that can be precomputed once the distortion degree n_d is fixed. Substituting into (43), we obtain the rational form

$$\boxed{\bar{x}_d = \frac{\sum_{a,b} w_{x,ab} M_n^{a,b}}{\sum_{a,b} w_{0,ab} M_n^{a,b}}, \quad \bar{y}_d = \frac{\sum_{a,b} w_{y,ab} M_n^{a,b}}{\sum_{a,b} w_{0,ab} M_n^{a,b}}.} \tag{48}$$

Under the assumed polynomial radial distortion model, the centroid $\bar{\mathbf{p}}_d = (\bar{x}_d, \bar{y}_d)^\top$ given by (48) is an *unbiased* centroid of the distorted cell in the normalized plane. Finally, mapping back to the pixel coordinate system $\{v^s\}$ via the intrinsic matrix $\mathbf{K}^s$ yields the theoretical pixel centroid

$$\tilde{\mathbf{c}}_v = \mathbf{K}^s \begin{bmatrix} \bar{x}_d \\ \bar{y}_d \\ 1 \end{bmatrix}, \tag{49}$$

Discussion and relation to dot-based methods. The derivation above follows the same principle as the moment-based conic method for circular dots[?] , but replaces ellipses by quadrilaterals defined by four refined checkerboard corners. In our case, the polygon geometry is determined by (i) second-stage saddle-point corners and (ii) the global checkerboard grid (including cross-ratio invariants used later in the calibration stage), which leads to more accurate and stable polygon moments than directly segmenting circular blobs in challenging conditions. This, in turn, improves the accuracy of the unbiased centroid used as a soft constraint in our two-stage calibration.

S4. Optimization Details for Centroid-Assisted Two-Stage Binocular Calibration

This section provides additional details, including parameterization, residual definitions, Jacobian structure, and the progressive two-stage optimization scheme: world frame $\{W\}$, left/right camera frames $\{C^L\}, \{C^R\}$, normalized planes $\{r^L\}, \{r^R\}$, and pixel planes $\{v^L\}, \{v^R\}$.

S4.1 Parameterization and Notation

We consider N_{img} checkerboard images observed by a binocular system. For camera $s \in \{L, R\}$, we denote:

- Intrinsic matrix: $\mathbf{K}^s = \begin{bmatrix} f_x^s & \eta^s & c_x^s \\ 0 & f_y^s & c_y^s \\ 0 & 0 & 1 \end{bmatrix}$.
- Radial distortion coefficients: $\boldsymbol{k}^s = [d_0^s, \ldots, d_{n_d}^s]^\top$.
- Per-image extrinsics: rotation and translation from $\{W\}$ to $\{C^s\}$, $\mathbf{R}^{s,(n)} \in SO(3)$, $\mathbf{t}^{s,(n)} \in \mathbb{R}^3$, $n = 1, \ldots, N_{\text{img}}$.

We collect all parameters into a single vector

$$\boldsymbol{\theta} = \Big[\underbrace{\boldsymbol{\theta}_{\text{int}}^L}_{\mathbf{K}^L, \boldsymbol{k}^L}, \underbrace{\boldsymbol{\theta}_{\text{int}}^R}_{\mathbf{K}^R, \boldsymbol{k}^R}, \underbrace{\boldsymbol{\theta}_{\text{ext}}^L}_{\{\mathbf{R}^{L,(n)}, \mathbf{t}^{L,(n)}\}}, \underbrace{\boldsymbol{\theta}_{\text{ext}}^R}_{\{\mathbf{R}^{R,(n)}, \mathbf{t}^{R,(n)}\}} \Big]^\top, \tag{50}$$

where rotations are parameterized by axis–angle vectors or Rodrigues vectors (minimal 3-parameter representation). The relative pose between $\{C^L\}$ and $\{C^R\}$ is implicitly encoded by these extrinsics; if desired, one may reparameterize the right camera pose as a fixed transform w.r.t. $\{C^L\}$ plus per-image relative motions, but this is not required by our method.

S4.2 Stage 1: Corner-Only Binocular Initialization

Corner projection. For world corner $\mathbf{Q}_W^{(i)}$ on the checkerboard plane $Z_W = 0$ and image index n, the forward projection to the pixel plane $\{v^s\}$ proceeds as

$$\mathbf{P}_C^{s,(n,i)} = \mathbf{R}^{s,(n)} \mathbf{Q}_W^{(i)} + \mathbf{t}^{s,(n)} \in \{C^s\}, \tag{51}$$

$$\mathbf{P}_r^{s,(n,i)} = \begin{bmatrix} X_C^{s,(n,i)} / Z_C^{s,(n,i)} \\ Y_C^{s,(n,i)} / Z_C^{s,(n,i)} \end{bmatrix} \in \{r^s\}, \tag{52}$$

$$\mathbf{P}_d^{s,(n,i)} = \mathcal{D}\big(\mathbf{P}_r^{s,(n,i)}; \boldsymbol{k}^s\big) = k^s(s)\, \mathbf{P}_r^{s,(n,i)}, \tag{53}$$

$$\tilde{\mathbf{q}}_v^{s,(n,i)} = \Pi\big(\mathbf{K}^s, \mathbf{P}_d^{s,(n,i)}\big) = \begin{bmatrix} f_x^s x_d^{s,(n,i)} + \eta^s y_d^{s,(n,i)} + c_x^s \\ f_y^s y_d^{s,(n,i)} + c_y^s \end{bmatrix}, \tag{54}$$

where $s = x_r^2 + y_r^2$ is the squared radius on the normalized plane.

Corner residual and loss. Given the refined saddle-point corner $\hat{\mathbf{q}}_v^{s,(n,i)} = [u^{s,(n,i)}, v^{s,(n,i)}]^\top$ The Stage 1 corner residual is

$$\mathbf{e}_{\text{corner}}^{s,(n,i)} = \hat{\mathbf{q}}_v^{s,(n,i)} - \tilde{\mathbf{q}}_v^{s,(n,i)}, \tag{55}$$

and the Stage 1 objective is

$$\mathcal{L}_{\text{corner}} = \sum_{s \in \{L,R\}} \sum_{n=1}^{N_{\text{img}}} \sum_i \big\| \mathbf{e}_{\text{corner}}^{s,(n,i)} \big\|_2^2. \tag{56}$$

Jacobian structure. The Jacobian of $\mathbf{e}_{\text{corner}}^{s,(n,i)}$ w.r.t. parameters in $\boldsymbol{\theta}$ is obtained by chain rule:

$$\frac{\partial \mathbf{e}_{\text{corner}}^{s,(n,i)}}{\partial \boldsymbol{\theta}} = - \frac{\partial \tilde{\mathbf{q}}_v^{s,(n,i)}}{\partial \boldsymbol{\theta}}. \tag{57}$$

For a given camera s and image n, the relevant blocks are:

$$\frac{\partial \tilde{\mathbf{q}}_v^{s,(n,i)}}{\partial \mathbf{K}^s} = \frac{\partial \Pi(\mathbf{K}^s, \mathbf{P}_d)}{\partial \mathbf{K}^s}, \tag{58}$$

$$\frac{\partial \tilde{\mathbf{q}}_v^{s,(n,i)}}{\partial \boldsymbol{k}^s} = \frac{\partial \Pi(\mathbf{K}^s, \mathbf{P}_d)}{\partial \mathbf{P}_d} \frac{\partial \mathcal{D}(\mathbf{P}_r; \boldsymbol{k}^s)}{\partial \boldsymbol{k}^s}, \tag{59}$$

$$\frac{\partial \tilde{\mathbf{q}}_v^{s,(n,i)}}{\partial \mathbf{R}^{s,(n)}} = \frac{\partial \Pi(\mathbf{K}^s, \mathbf{P}_d)}{\partial \mathbf{P}_d} \frac{\partial \mathcal{D}(\mathbf{P}_r; \boldsymbol{k}^s)}{\partial \mathbf{P}_r} \frac{\partial \mathbf{P}_r}{\partial \mathbf{P}_C} \frac{\partial \mathbf{P}_C}{\partial \mathbf{R}^{s,(n)}}, \tag{60}$$

$$\frac{\partial \tilde{\mathbf{q}}_v^{s,(n,i)}}{\partial \mathbf{t}^{s,(n)}} = \frac{\partial \Pi(\mathbf{K}^s, \mathbf{P}_d)}{\partial \mathbf{P}_d} \frac{\partial \mathcal{D}(\mathbf{P}_r; \boldsymbol{k}^s)}{\partial \mathbf{P}_r} \frac{\partial \mathbf{P}_r}{\partial \mathbf{P}_C} \frac{\partial \mathbf{P}_C}{\partial \mathbf{t}^{s,(n)}}. \tag{61}$$

Each factor is standard in bundle adjustment:

- $\partial \Pi / \partial \mathbf{K}^s$ is linear in (x_d, y_d);
- $\partial \mathcal{D} / \partial \boldsymbol{k}^s$ depends on derivatives of $k^s(s)$ w.r.t. d_i^s;
- $\partial \mathbf{P}_r / \partial \mathbf{P}_C$ encodes the division by Z_C;
- $\partial \mathbf{P}_C / \partial \mathbf{R}^{s,(n)}, \partial \mathbf{P}_C / \partial \mathbf{t}^{s,(n)}$ follow from the exponential map or Rodrigues parameterization of $SO(3)$.

In practice, we use automatic differentiation or analytically coded Jacobians, and solve (56) using a damped Gauss-Newton (Levenberg–Marquardt) optimizer.

S4.3 Stage 2: Centroid-Assisted Refinement

Stage 2 refines $\boldsymbol{\theta}$ starting from the Stage 1 optimum by incorporating centroid residuals and an optional binocular epipolar regularizer, while preserving corner accuracy.

Centroid Residuals and Jacobians For the k-th cell in image n of camera s, let:

- $\hat{\mathbf{c}}_v^{s,(n,k)} \in \{v^s\}$ be the observed geometric centroid, computed from the four refined corners in $\{v^s\}$ using the polygon-centroid formula.

- $\tilde{\mathbf{c}}_v^{s,(n,k)} \in \{v^s\}$ be the theoretical unbiased centroid, obtained as

$$\bar{\mathbf{p}}_d^{s,(n,k)} = \mathbf{F}_{\text{centroid}}\big(\{M_n^{a,b}(s,n,k)\}, \boldsymbol{k}^s\big) \in \{r^s\}, \quad (62)$$

followed by projection with $\mathbf{K}^s$:

$$\tilde{\mathbf{c}}_v^{s,(n,k)} = \mathbf{K}^s \begin{bmatrix} \bar{x}_d^{s,(n,k)} \\ \bar{y}_d^{s,(n,k)} \\ 1 \end{bmatrix}. \quad (63)$$

Here $\{M_n^{a,b}(s,n,k)\}$ are normalized polygon moments of the undistorted quadrilateral on $\{r^s\}$, and $\mathbf{F}_{\text{centroid}}$ is the rational mapping from moments and distortion parameters to the distorted centroid (see Eq. (48) in Sec. S3).

The centroid residual is

$$\mathbf{e}_{\text{centroid}}^{s,(n,k)} = \tilde{\mathbf{c}}_v^{s,(n,k)} - \hat{\mathbf{c}}_v^{s,(n,k)}, \quad (64)$$

and the centroid loss is

$$\mathcal{L}_{\text{centroid}} = \sum_{s\in\{L,R\}} \sum_{n=1}^{N_{\text{img}}} \sum_{k} \big\|\mathbf{e}_{\text{centroid}}^{s,(n,k)}\big\|_2^2. \quad (65)$$

The Jacobian $\partial \mathbf{e}_{\text{centroid}}^{s,(n,k)}/\partial\boldsymbol{\theta}$ involves:

$$\frac{\partial \mathbf{e}_{\text{centroid}}^{s,(n,k)}}{\partial\boldsymbol{\theta}} = \frac{\partial \tilde{\mathbf{c}}_v^{s,(n,k)}}{\partial\boldsymbol{\theta}} - \frac{\partial \hat{\mathbf{c}}_v^{s,(n,k)}}{\partial\boldsymbol{\theta}}. \quad (66)$$

Derivative of the observed centroid. The observed centroid $\hat{\mathbf{c}}_v^{s,(n,k)}$ depends on the four refined corners in $\{v^s\}$, which in turn depend on the saddle-point fit and thus on image intensities, not directly on calibration parameters. In our optimization, we treat $\hat{\mathbf{c}}_v^{s,(n,k)}$ as fixed observations, i.e.,

$$\frac{\partial \hat{\mathbf{c}}_v^{s,(n,k)}}{\partial\boldsymbol{\theta}} \approx \mathbf{0}. \quad (67)$$

This is consistent with the usual treatment of detected feature points in bundle adjustment.

Derivative of the theoretical centroid. The theoretical centroid $\tilde{\mathbf{c}}_v^{s,(n,k)}$ depends on: (i) the undistorted quadrilateral on $\{r^s\}$ via polygon moments $M_n^{a,b}(s,n,k)$, which are determined by the undistorted corner positions; and (ii) the distortion parameters $\boldsymbol{k}^s$ and intrinsics $\mathbf{K}^s$. Applying chain rule to Eq. (63) and the mapping in Eq. (48), we obtain

$$\frac{\partial \tilde{\mathbf{c}}_v^{s,(n,k)}}{\partial \mathbf{K}^s} = \frac{\partial \mathbf{K}^s \bar{\mathbf{p}}_d^{s,(n,k)}}{\partial \mathbf{K}^s}, \quad (68)$$

$$\frac{\partial \tilde{\mathbf{c}}_v^{s,(n,k)}}{\partial \boldsymbol{k}^s} = \mathbf{K}^s \frac{\partial \bar{\mathbf{p}}_d^{s,(n,k)}}{\partial \boldsymbol{k}^s}, \quad (69)$$

$$\frac{\partial \tilde{\mathbf{c}}_v^{s,(n,k)}}{\partial \mathbf{R}^{s,(n)}} = \mathbf{K}^s \frac{\partial \bar{\mathbf{p}}_d^{s,(n,k)}}{\partial M_n^{a,b}} \frac{\partial M_n^{a,b}}{\partial \hat{\mathbf{p}}_r^{s,(n,k,\ell)}} \frac{\partial \hat{\mathbf{p}}_r^{s,(n,k,\ell)}}{\partial \mathbf{R}^{s,(n)}}, \quad (70)$$

$$\frac{\partial \tilde{\mathbf{c}}_v^{s,(n,k)}}{\partial \mathbf{t}^{s,(n)}} = \mathbf{K}^s \frac{\partial \bar{\mathbf{p}}_d^{s,(n,k)}}{\partial M_n^{a,b}} \frac{\partial M_n^{a,b}}{\partial \hat{\mathbf{p}}_r^{s,(n,k,\ell)}} \frac{\partial \hat{\mathbf{p}}_r^{s,(n,k,\ell)}}{\partial \mathbf{t}^{s,(n)}}, \quad (71)$$

where summation over (a,b) and corner index ℓ is implied. The factors:

- $\partial \bar{\mathbf{p}}_d/\partial \boldsymbol{k}^s$ and $\partial \bar{\mathbf{p}}_d/\partial M_n^{a,b}$ follow from the rational expression in Eq. (48);
- $\partial M_n^{a,b}/\partial \hat{\mathbf{p}}_r^{s,(n,k,\ell)}$ follows from the Green-theorem-based formulas in Eq. (35);
- $\partial \hat{\mathbf{p}}_r/\partial \mathbf{R}^{s,(n)}, \partial \hat{\mathbf{p}}_r/\partial \mathbf{t}^{s,(n)}$ follow from the undistortion and normalization steps.

In our implementation, these derivatives are either coded analytically or approximated by automatic differentiation on $\mathbf{F}_{\text{centroid}}$ and the undistortion pipeline.

Epipolar Regularization Given the relative pose between $\{C^L\}$ and $\{C^R\}$, $(\mathbf{R}^{LR}, \mathbf{t}^{LR})$, the essential matrix is

$$\mathbf{E} = [\mathbf{t}^{LR}]_\times \mathbf{R}^{LR}, \quad (72)$$

where $[\cdot]_\times$ denotes the skew-symmetric matrix. For corresponding normalized (undistorted) corner points $\hat{\mathbf{q}}_r^{L,(n,i)}, \hat{\mathbf{q}}_r^{R,(n,i)} \in \{r^L\}, \{r^R\}$, the epipolar constraint requires

$$\hat{\mathbf{q}}_r^{R,(n,i)\top} \mathbf{E}\, \hat{\mathbf{q}}_r^{L,(n,i)} \approx 0. \quad (73)$$

We define a scalar epipolar residual

$$e_{\text{epi}}^{(n,i)} = \hat{\mathbf{q}}_r^{R,(n,i)\top} \mathbf{E}\, \hat{\mathbf{q}}_r^{L,(n,i)}, \quad (74)$$

and the epipolar loss

$$\mathcal{L}_{\text{epi}} = \sum_{n=1}^{N_{\text{img}}} \sum_{i} \big(e_{\text{epi}}^{(n,i)}\big)^2, \quad (75)$$

The Jacobian of $e_{\text{epi}}^{(n,i)}$ w.r.t. $\boldsymbol{\theta}$ follows from the chain rule on $\mathbf{E}(\mathbf{R}^{LR}, \mathbf{t}^{LR})$ and the normalized coordinates; details are standard and omitted for brevity.

S4.4 Two-Stage Optimization Scheme

We now summarize the complete optimization procedure.

Stage 1: Corner-Only Initialization The Stage 1 optimization can be summarized as the following procedure.

Algorithm (Stage 1).

1. **Input:** saddle-point corners $\hat{\mathbf{q}}_v^{s,(n,i)}$, world corners $\mathbf{Q}_W^{(i)}$.
2. Initialize $\mathbf{K}^s$, $\boldsymbol{k}^s$, $\mathbf{R}^{s,(n)}$, $\mathbf{t}^{s,(n)}$ using a Zhang-style method.
3. Compute corner predictions $\tilde{\mathbf{q}}_v^{s,(n,i)}$ and residuals $\mathbf{e}_{\text{corner}}^{s,(n,i)}$ in Eq. (55).
4. Assemble the Jacobian $\partial \mathbf{e}_{\text{corner}}^{s,(n,i)}/\partial\boldsymbol{\theta}$.

5. Take Gauss–Newton / LM steps on $\boldsymbol{\theta}$ to minimize $\mathcal{L}_{\text{corner}}$ in Eq. (56).

6. Iterate steps (iii)–(v) until convergence.

The resulting parameters are denoted as $\boldsymbol{\theta}^{(1)}$ and used as the initialization for Stage 2.

Stage 2: Centroid-Assisted Progressive Refinement Starting from $\boldsymbol{\theta}^{(1)}$, Stage 2 minimizes the combined objective

$$\mathcal{L}_{\text{stage2}} = \mathcal{L}_{\text{corner}} + \lambda_{\text{c}}\,\mathcal{L}_{\text{centroid}} + \lambda_{\text{e}}\,\mathcal{L}_{\text{epi}}, \tag{76}$$

with a progressive schedule on λ_{c} as described in the main paper.

Algorithm (Stage 2).

1. **Input:** Stage 1 parameters $\boldsymbol{\theta}^{(1)}$, centroid observations $\hat{\mathbf{c}}_v^{s,(n,k)}$.

2. Set $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{(1)}$ and compute the baseline corner RMS error $E_{\text{corner}}^{(1)}$ from $\mathcal{L}_{\text{corner}}$.

3. Initialize $\lambda_{\text{c}} \leftarrow \lambda_{\text{c}}^{\min}$ and fix λ_{e}.

4. For outer iterations $t = 1, \ldots, T$:

   (a) Compute theoretical centroids $\tilde{\mathbf{c}}_v^{s,(n,k)}$ and residuals $\mathbf{e}_{\text{centroid}}^{s,(n,k)}$ in Eq. (66).

   (b) (If used) compute epipolar residuals $e_{\text{epi}}^{(n,i)}$ and $\mathcal{L}_{\text{epi}}$.

   (c) Assemble the residual vector and Jacobian for $\mathcal{L}_{\text{stage2}}$ with current $\lambda_{\text{c}}, \lambda_{\text{e}}$, and take one or several LM steps on $\boldsymbol{\theta}$.

   (d) Recompute the corner RMS error $E_{\text{corner}}^{(t)}$.

   (e) If $E_{\text{corner}}^{(t)} > E_{\text{corner}}^{(1)} + \epsilon$, stop increasing λ_{c} (or revert the last update) to protect corner accuracy.

   (f) Otherwise, increase λ_{c} towards $\lambda_{\text{c}}^{\max}$ (e.g., linearly or geometrically) and continue.

The final parameters after Stage 2 are denoted as $\boldsymbol{\theta}^{\star}$. Here ϵ is a small tolerance (0.01 pixels) controlling how much the corner RMS is allowed to increase. In our experiments, we typically choose a small $\lambda_{\text{c}}^{\min}$ so that early iterations are dominated by $\mathcal{L}_{\text{corner}}$ and $\mathcal{L}_{\text{epi}}$, and a moderate $\lambda_{\text{c}}^{\max}$ such that centroids significantly improve bias without overfitting to local centroid noise.

This two-stage strategy realizes the design principle: corners are hard constraints that define the primary binocular geometry, while unbiased cell centroids act as soft but informative regularizers that further reduce distortion and extrinsic bias, ultimately benefiting downstream structured-light 3D reconstruction.

S5 Additional Implementation Details

This section complements the main paper with concrete implementation details, hyper-parameters and practical choices for all modules. We follow the coordinate systems: world frame $\{W\}$, camera frames $\{C^L\}, \{C^R\}$, normalized planes $\{r^L\}, \{r^R\}$, and pixel planes $\{v^L\}, \{v^R\}$.

S5.1 Geometry-Aware Checkerboard Super-Resolution

Network architecture. The checkerboard super-resolution (SR) module is implemented as a lightweight encoder–decoder network:

- **Input.** Grayscale checkerboard ROI cropped in the pixel plane $\{v^s\}$ and normalized to $[0, 1]$.
- **Encoder.** Three 3×3 convolutional layers with feature dimensions $32 \to 64 \to 64$, stride 1 and ReLU activations.
- **Residual body.** $N_{\text{res}} = 5$ residual blocks; each block contains two 3×3 convolutions (64 channels) with a local skip connection.
- **Upsampling.** A pixel-shuffle layer with up-scale factor $\times 2$, followed by a 3×3 convolution to generate the SR output.

This design keeps the model compact and efficient while providing sufficient capacity to sharpen checkerboard edges and corners.

Training data and augmentation. The SR network is trained offline from a mixture of synthetic and real data:

- *Synthetic data.* Ideal high-resolution checkerboards are rendered in the world plane $\{W\}$, then degraded by bicubic down-sampling, Gaussian blur, exposure noise, and mild vignetting to simulate realistic optics. The clean renders serve as ground-truth HR targets.
- *Real data.* We capture calibration sequences with our binocular system under different focus, exposure and noise conditions. Checkerboard ROIs are cropped and, when possible, approximate HR references are obtained by temporal averaging of multiple frames or by down-scaling higher-resolution captures.

Data augmentation includes random rotation ($\pm 5°$), small perspective warps, contrast jitter, and additive Gaussian noise. All experiments use a fixed scale factor of $\times 2$.

Loss and optimization. The SR model is trained with a combination of intensity and gradient losses:

$$\mathcal{L}_{\text{SR}} = \|\hat{I}_{\text{SR}} - I_{\text{HR}}\|_1 + \lambda_\nabla \Big(\|\nabla_x \hat{I}_{\text{SR}} - \nabla_x I_{\text{HR}}\|_1 + \|\nabla_y \hat{I}_{\text{SR}} - \nabla_y I_{\text{HR}}\|_1 \Big). \tag{77}$$

where λ_∇ is set to 0.1. We use Adam with an initial learning rate of 10^{-4}, cosine decay, batch size 32, and train for 300 epochs. The trained network is then fixed for all calibration experiments.

S5.2 Corner and Centroid Extraction

Initial corner detection. For each calibration image, we perform:

1. **Pre-processing.** Convert to grayscale, apply a small Gaussian blur ($\sigma \approx 1$) and contrast normalization inside the checkerboard ROI.
2. **Checkerboard localization.** Run a Harris detector to obtain candidate corners, followed by a grid-fitting step (RANSAC + least squares) to estimate the board layout and reject spurious responses.
3. **Sub-pixel refinement.** Apply OpenCV's `cornerSubPix` in a 7×7 window to obtain initial LR sub-pixel corners; then map them to SR coordinates by scaling and cropping.

Saddle-point refinement. On the SR image, we extract a 19×19 patch around each initial corner and fit the quadratic surface in Eq. (18). Coefficients are solved by linear least squares. The stationary point is obtained by Eq. (23); if the Hessian fails the saddle condition in Eq. (21) or the patch has insufficient contrast, we keep the initial sub-pixel corner. Otherwise, we update the corner to the saddle-point-refined location $\hat{\mathbf{q}}_v^{s,(n,i)}$.

Cell selection and moment preparation. Using the refined corners, we construct the regular grid of cells (squares) for each view:

- Cells touching the ROI boundary or partially occluded are discarded.
- For each valid cell, we form a quadrilateral in the pixel plane $\{v^s\}$ from its four refined corners, compute the geometric centroid via the polygon formula, and treat it as the observed centroid $\hat{\mathbf{c}}_v^{s,(n,k)}$.
- The four corners are then mapped to the normalized plane $\{r^s\}$ using the current intrinsics and distortion parameters. From these four vertices we compute polygon moments $M_n^{a,b}(s, n, k)$ using Eq. (35), which are fed into the unbiased centroid model (Eq. (48)) in Stage 2.

S5.3 Two-Stage Calibration Hyper-Parameters

Stage 1 (corner-only). Stage 1 follows the Zhang-style initialization and bundle adjustment described in Sec. . Typical settings are:

- **Initialization.** We run OpenCV stereo calibration with refined corners to obtain initial $\mathbf{K}^s$, $\boldsymbol{k}^s$ and per-image extrinsics. When the baseline between cameras is known, we fix its magnitude during optimization.
- **Optimizer.** Levenberg–Marquardt (damped Gauss–Newton) with initial damping $\lambda_0 = 10^{-3}$, maximum 50 iterations.
- **Stopping criteria.** Relative decrease of $\mathcal{L}_{\text{corner}}$ below 10^{-8} or parameter increment norm below 10^{-8}.
- **Parameter bounds.** Focal lengths are constrained to a narrow interval around the initial estimate; distortion coefficients are bounded (e.g. $|d_i^s| < 1$) to avoid degenerate solutions.

Stage 2 (centroid-assisted). Stage 2 minimizes the combined loss $\mathcal{L}_{\text{stage2}} = \mathcal{L}_{\text{corner}} + \lambda_{\text{c}}\,\mathcal{L}_{\text{centroid}} + \lambda_{\text{e}}\,\mathcal{L}_{\text{epi}}$ starting from the Stage 1 optimum:

- **Weights.** We use $\lambda_{\text{c}}^{\min} = 0.01$, $\lambda_{\text{c}}^{\max} = 0.5$. The centroid weight starts from $\lambda_{\text{c}}^{\min}$ and is multiplied by 2 after every 5 outer iterations until it reaches $\lambda_{\text{c}}^{\max}$ or the corner safeguard is activated. The epipolar weight is fixed to $\lambda_{\text{e}} = 0.1$.
- **Corner safeguard.** Let $E_{\text{corner}}^{(1)}$ be the Stage 1 RMS corner error. During Stage 2, if the current RMS exceeds $E_{\text{corner}}^{(1)} + \epsilon$ with $\epsilon = 0.01$ pixels, we stop increasing λ_{c} (and optionally revert the last update). This enforces the "corner-as-hard-constraint" principle.
- **Inner iterations.** For each outer step (fixed $\lambda_{\text{c}}, \lambda_{\text{e}}$), we run up to 10 LM iterations on $\mathcal{L}_{\text{stage2}}$, using the Jacobians described in Sec. and Sec. .

In practice, $\mathcal{L}_{\text{centroid}}$ decreases rapidly in the first few outer iterations, yielding noticeable improvements in distortion and extrinsics, while the corner RMS remains almost unchanged thanks to the safeguard.

S5.4 Runtime and Memory

All experiments are conducted on a workstation with an Intel Xeon CPU and a single NVIDIA RTX-class GPU, using single-precision (FP32) computation for the SR network and double precision for optimization:

- For a typical 1920×1200 image, checkerboard SR inference on a 256×256 ROI takes approximately 5–8 ms per camera.
- Corner detection, saddle-point refinement and centroid computation together require around 10–20 ms per image.
- Stage 1 and Stage 2 calibration with up to $N_{\text{img}} = 20$ image pairs complete within one second on CPU.
- The peak GPU memory usage for SR is below 1 GB, and the memory footprint of the optimization is dominated by Jacobian storage, which stays well within a few hundred MB.

Therefore, the proposed pipeline can be integrated into existing structured-light 3D reconstruction systems with negligible computational overhead while providing significant gains in calibration accuracy.

S6 Structured-Light 3D Reconstruction

Our stereo structured-light 3D reconstruction system is a highly integrated experimental platform designed to perform high-precision 3D measurements. The entire system consists of two core subsystems that work in synergy to reconstruct the 3D geometric information of an object.

- Imaging System: We utilize a pair of grayscale stereo cameras with identical parameters and a high resolution of 1024×1280 pixels. These grayscale cameras are capable of capturing high-contrast stripe patterns, avoiding potential errors caused by color channel crosstalk in color cameras. The cameras' wide field of view (approx. 55° horizontal, 44° vertical) ensures a sufficiently large overlapping region, providing rich information for stereo matching.

- Projection System: The heart of the projector is a high-performance MEMS (Micro-Electro-Mechanical System) single-axis micromirror. Unlike traditional DLP projectors, the MEMS micromirror can project structured-light patterns at extremely high speeds and with remarkable precision. The coding strategy we employ combines Gray code and phase-shifting code. Gray code is used for the unique identification of each pixel, solving the spatial encoding problem, while phase-shifting code provides sub-pixel-level precise phase information, significantly enhancing the accuracy of depth measurements.

Reconstruction Workflow: In each experimental scene, the projection system rapidly projects a sequence of 18 coded patterns. The imaging system, equipped with a global shutter, synchronously captures these 18 images. Subsequently, using our pre-calibrated camera parameters and the precise phase information decoded from the stripe images, the system employs the principle of triangulation to compute the 3D coordinates of every visible point in the scene, ultimately achieving high-accuracy 3D reconstruction. This entire process is highly automated, ensuring both efficiency and accuracy in data acquisition and reconstruction.

S7 Implementation Details of Experiments

S7.1 Dataset Generation

Simulated Data To provide a controlled experimental environment with precise ground truth, we generated the first set of calibration photos using Blender simulation software. By keeping the stereo cameras stationary and using a Python script to randomly change the virtual calibration board's pose (translation and rotation), we were able to automatically simulate a wide range of relative pose variations. This approach ensures high data diversity and allows for precise control over all variables, including camera parameters, noise types, and distortion levels. This makes the simulated data ideal for quantitative analysis and ablation studies, as well as for model training and evaluation. For Fig. 4 in the article, we took 18 photos, with poses as shown in Fig. 2.

Real-World Data To validate the simulation results in a real-world setting, we collected a second set of photos using a physical experimental system, as shown in Fig. 1. Data was acquired using two methods: Manual Collection: The stereo cameras were fixed, and a calibration board was manually moved to capture a variety of poses. Robot-Assisted Collection: The calibration board was attached to the end of a six-axis robotic arm. This method allowed for precise and repeatable control over the board's pose, resulting in high-accuracy calibration data. During real-world data collection, we observed that the calibration accuracy did not significantly improve when the number of photos exceeded 30. Therefore, to achieve an optimal balance between calibration accuracy and collection efficiency, each photo set was limited to 30 images.

Camera calibration is a prerequisite for accurate metric reconstruction in stereo and structured-light systems, because it establishes the mapping from image measurements to physical 3D geometry via the intrinsic parameters and the lens distortion model. As shown in Fig. 4(a1–c3), real captures often suffer from severe photometric degradation across under/normal/over-exposure conditions, resulting in low-SNR regions, intensity compression, and saturation-induced clipping. These effects directly destabilize feature extraction (e.g., checkerboard corners) and can bias the estimated intrinsics if not properly accounted for.

Geometric distortion introduces an additional, purely spatial source of error. Fig. 3(a) illustrates how an ideal imaging point is displaced by lens distortion, which can be decomposed into radial and tangential components, and Fig. 3(b–d) visualizes the corresponding warping of a regular grid. Even small residual distortions can translate into systematic reprojection errors and, in stereo settings, into imperfect epipolar alignment.

Finally, defocus and motion blur further undermine calibration robustness by attenuating high-frequency edges and smearing corner structures, effectively reducing the localization precision and increasing ambiguity in correspondence. Taken together, noise (Fig. 4), distortion (Fig. 3), and blur constitute three dominant degradations that challenge reliable calibration in practical acquisitions; hence, a robust calibration pipeline must jointly tolerate low SNR and saturation, accurately model and compensate lens distortion, and maintain subpixel feature localization accuracy under blurred observations.

S7.2 Experimental Supplement

To keep the main manuscript concise, we provide additional qualitative and quantitative results that further validate the proposed GeoRobustCalib framework.

First, Supplementary Fig. 5 expands the analysis of Module 1 in Fig. E3 (main article) by visualizing the intermediate outputs of the geometry-aware checkerboard super-resolution pipeline, including the original calibration image, Harris-based corner segmentation, homography-driven grid reconstruction, and the final super-resolved checkerboard with background suppression. These results confirm that the SR module selectively enhances only the checkerboard region, producing sharper and more geometrically consistent edges while leaving the background largely unchanged, which directly benefits downstream corner and centroid estimation.

Second, Supplementary Fig. **??** provides a focused experiment corresponding to Fig. 3 (main article), where the Stage 1 calibration already achieves nearly optimal corner accuracy for a given image set. In this setting, Stage 2 is configured to preserve the Stage 1 corner reprojection RMS (0.465 pixels for the left camera) and only refine the cell centroids. The centroid reprojection RMS is further reduced from 0.385 pixels to 0.376 pixels without sacrificing corner accuracy, illustrating that the centroid-assisted refinement acts as a gentle regularizer that reduces systematic bias in distortion and extrinsics while strictly enforcing the "corners-as-hard-constraints, centroids-as-soft-priors" design principle.

Third, Supplementary Table 1 provides a detailed numerical counterpart to Fig. 4 (main article) by reporting the estimated intrinsic parameters on synthetic stereo data with known ground truth under controlled Gaussian noise, blur, and additional radial distortion. For each degradation type and three interference levels, we report the mean and standard deviation over 30 independent trials for both the MATLAB toolbox and the proposed method. Across all conditions, our estimates of focal lengths, principal point, and first radial distortion coefficient remain consistently closer to the ground truth and exhibit smaller variance than the MATLAB baseline, especially at higher interference levels, which quantitatively corroborates the robustness trends observed in Fig. 4 (main article).

TABLE I: The details in Figure 4 of the main article. Stereo calibration on synthetic data with known ground truth (GT). For each degradation type (Gaussian noise, blur, and additional radial distortion), we test three interference levels and report the mean ± standard deviation of the estimated intrinsics over 30 trials. Best results (closer to GT) are in **bold**.

Condition	Level	Method	f_x	f_y	c_x	c_y	k_1
GT (true)	—	—	694.5399	694.5197	250.2381	250.4674	−0.0018
Noise	NS=0.01	MATLAB	693.5±0.40	695.5±0.42	251.2±0.40	249.5±0.45	−0.0030±0.0005
		Ours	**694.4±0.10**	**694.6±0.11**	**250.3±0.10**	**250.4±0.12**	**−0.0019±0.0002**
	NS=0.03	MATLAB	693.3±0.50	695.8±0.52	251.5±0.50	249.2±0.55	−0.0033±0.0007
		Ours	**694.3±0.12**	**694.7±0.13**	**250.4±0.12**	**250.3±0.14**	**−0.0020±0.0002**
	NS=0.05	MATLAB	693.0±0.60	696.0±0.65	251.8±0.60	248.9±0.65	−0.0036±0.0009
		Ours	**694.2±0.15**	**694.8±0.16**	**250.4±0.15**	**250.3±0.17**	**−0.0021±0.0003**
	NS=0.07	MATLAB	692.8±0.70	696.2±0.75	252.0±0.70	248.7±0.75	−0.0039±0.0011
		Ours	**694.1±0.18**	**694.8±0.19**	**250.5±0.18**	**250.2±0.20**	**−0.0022±0.0004**
	NS=0.1	MATLAB	692.6±0.85	696.4±0.88	252.2±0.85	248.5±0.88	−0.0042±0.0014
		Ours	**694.0±0.22**	**694.9±0.23**	**250.6±0.21**	**250.1±0.23**	**−0.0023±0.0005**
	NS=0.15	MATLAB	692.4±0.95	696.5±1.00	252.4±0.95	248.3±1.00	−0.0045±0.0017
		Ours	**693.9±0.26**	**695.0±0.28**	**250.7±0.25**	**250.0±0.27**	**−0.0024±0.0006**
Blur	BS=0.5	MATLAB	695.6±0.45	693.4±0.48	249.2±0.42	251.5±0.50	−0.0031±0.0006
		Ours	**694.6±0.11**	**694.4±0.12**	**250.1±0.11**	**250.6±0.13**	**−0.0019±0.0002**
	BS=1.0	MATLAB	695.9±0.55	693.1±0.60	248.9±0.55	251.8±0.60	−0.0034±0.0008
		Ours	**694.7±0.14**	**694.3±0.15**	**250.0±0.14**	**250.7±0.16**	**−0.0020±0.0003**
	BS=1.5	MATLAB	696.1±0.68	692.8±0.72	248.6±0.65	252.1±0.75	−0.0037±0.0010
		Ours	**694.8±0.18**	**694.2±0.19**	**249.9±0.18**	**250.8±0.20**	**−0.0021±0.0004**
	BS=2.0	MATLAB	696.3±0.80	692.6±0.85	248.4±0.78	252.4±0.88	−0.0040±0.0013
		Ours	**694.9±0.22**	**694.1±0.23**	**249.8±0.22**	**250.9±0.25**	**−0.0022±0.0005**
	BS=2.5	MATLAB	696.5±0.92	692.4±0.98	248.2±0.90	252.6±1.00	−0.0043±0.0016
		Ours	**695.0±0.26**	**694.0±0.28**	**249.7±0.26**	**251.0±0.30**	**−0.0023±0.0006**
	BS=3.0	MATLAB	696.6±1.05	692.3±1.10	248.0±1.05	252.8±1.15	−0.0046±0.0019
		Ours	**695.1±0.30**	**693.9±0.33**	**249.6±0.30**	**251.1±0.35**	**−0.0024±0.0007**
Distortion	$k_1 = -0.15$	MATLAB	693.4±0.50	695.7±0.55	251.4±0.48	249.3±0.52	−0.0032±0.0007
		Ours	**694.4±0.12**	**694.6±0.14**	**250.3±0.12**	**250.4±0.13**	**−0.0019±0.0002**
	$k_1 = -0.1$	MATLAB	693.2±0.62	696.0±0.65	251.7±0.58	249.0±0.62	−0.0035±0.0009
		Ours	**694.3±0.16**	**694.7±0.18**	**250.4±0.16**	**250.3±0.17**	**−0.0020±0.0003**
	$k_1 = -0.05$	MATLAB	693.0±0.75	696.3±0.78	252.0±0.70	248.7±0.75	−0.0038±0.0011
		Ours	**694.2±0.20**	**694.8±0.22**	**250.5±0.20**	**250.2±0.22**	**−0.0021±0.0004**
	$k_1 = 0.05$	MATLAB	696.0±0.78	693.1±0.80	249.0±0.72	251.8±0.80	−0.0037±0.0011
		Ours	**694.8±0.21**	**694.2±0.23**	**250.0±0.21**	**250.7±0.24**	**−0.0021±0.0004**
	$k_1 = 0.1$	MATLAB	696.3±0.88	692.8±0.92	248.7±0.85	252.1±0.92	−0.0040±0.0014
		Ours	**694.9±0.25**	**694.1±0.27**	**249.9±0.25**	**250.8±0.29**	**−0.0022±0.0005**
	$k_1 = 0.15$	MATLAB	696.6±1.00	692.5±1.05	248.4±0.95	252.4±1.05	−0.0043±0.0017
		Ours	**695.0±0.30**	**694.0±0.32**	**249.8±0.30**	**250.9±0.34**	**−0.0024±0.0006**

* Electronic address: dyqiao@nwpu.edu.cn
† Electronic address: benquan001@e.ntu.edu.sg
‡ Electronic address: yijie.shen@ntu.edu.sg

[1] The exact expression depends on the choice of potential function in Green's theorem; here we show a generic form for illustration. In practice, we use closed-form expressions derived once and reused for all cells.

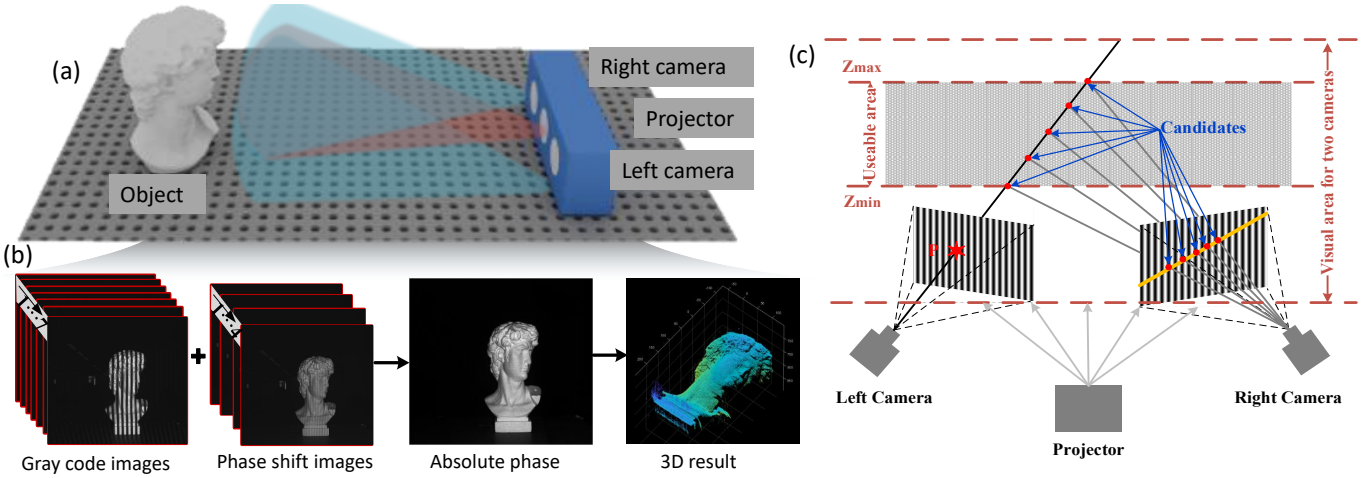

FIG. 1: Dataset acquisition system and structured-light 3D reconstruction pipeline. (a) Experimental setup of the binocular structured-light system, consisting of a left camera, a right camera, and a projector observing the common measurement volume. (b) Typical projection–capture sequence, including Gray-code images and phase-shifting fringe patterns, together with the decoded depth/absolute-phase maps. (c) Final 3D reconstruction result obtained from the absolute phase, illustrating the complete processing chain used throughout our experiments.

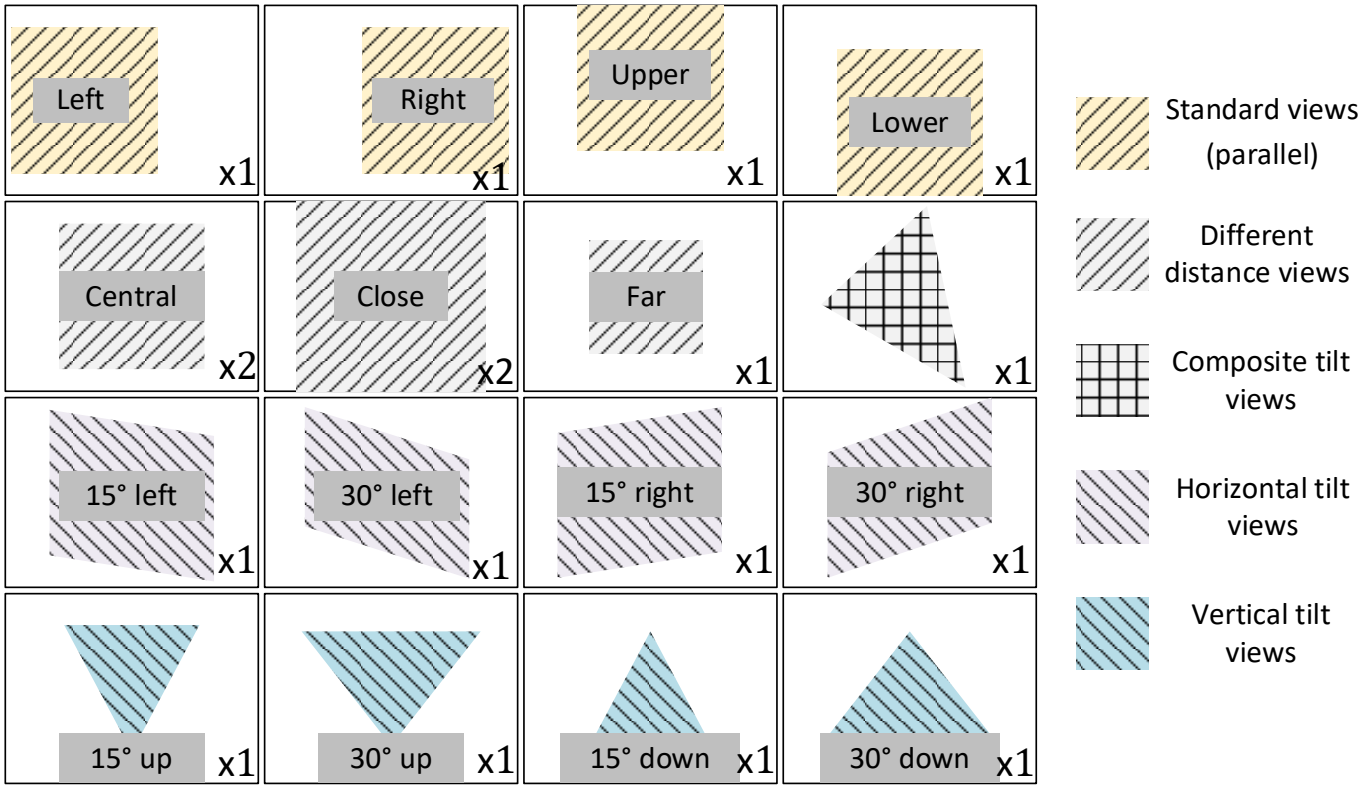

FIG. 2: Pose distribution of the calibration board used in Fig. 4 of the main article. The calibration images are grouped into five representative view types: standard (parallel) views, different-distance views, horizontal-tilt views, vertical-tilt views, and composite-tilt views. Each tile corresponds to a specific board pose (e.g., left/right/upper/lower/central, near/far, $\pm 15^\circ/\pm 30^\circ$ tilts), showing that the dataset covers diverse translations and rotations of the checkerboard in front of the stereo rig.

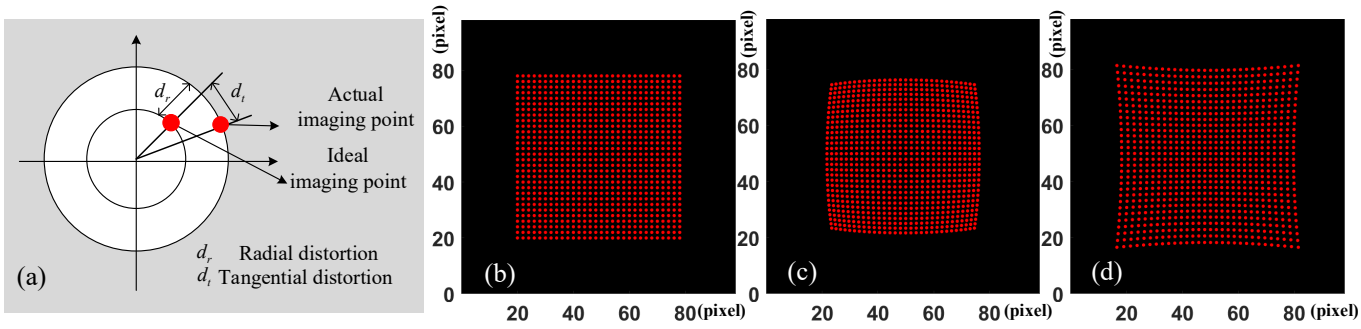

FIG. 3: **Illustration of lens distortion and its effects on a grid pattern.** (a) Schematic of an ideal imaging point (black) and its distorted observation (red), decomposed into radial distortion d_r (along the radial direction) and tangential distortion d_t (orthogonal component caused by decentering/tilt). (b–d) Example grid projections under different distortion conditions: (b) undistorted reference grid, (c) predominantly radial distortion producing barrel/pincushion-like bending, and (d) combined radial and tangential distortion leading to asymmetric warping of the grid.

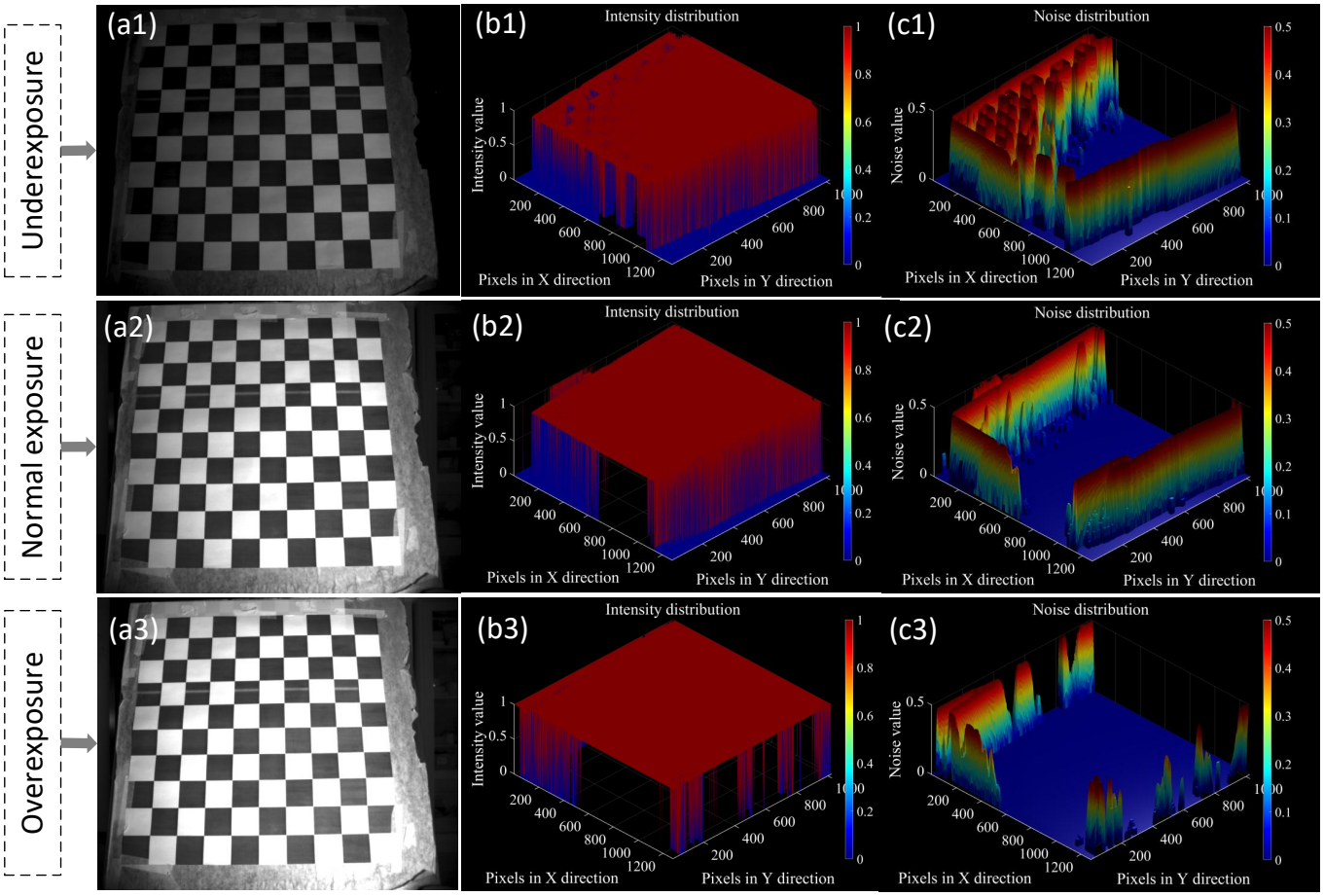

FIG. 4: **Photometric degradation across exposures and its effect on signal/noise distributions.** (a1–a3) Representative checkerboard captures under underexposure, normal exposure, and overexposure, respectively. (b1–b3) Corresponding 3D visualisations of the normalised intensity distribution over the image plane (pixel coordinates in x and y), highlighting signal compression in dark regions (underexposure) and saturation/clipping in bright regions (overexposure). (c1–c3) Estimated noise (or residual) magnitude distribution over the same images, showing elevated noise in low-signal areas for underexposure and distorted/noisy responses around saturated regions for overexposure. These results motivate robust calibration and reconstruction under high-dynamic-range acquisition conditions.

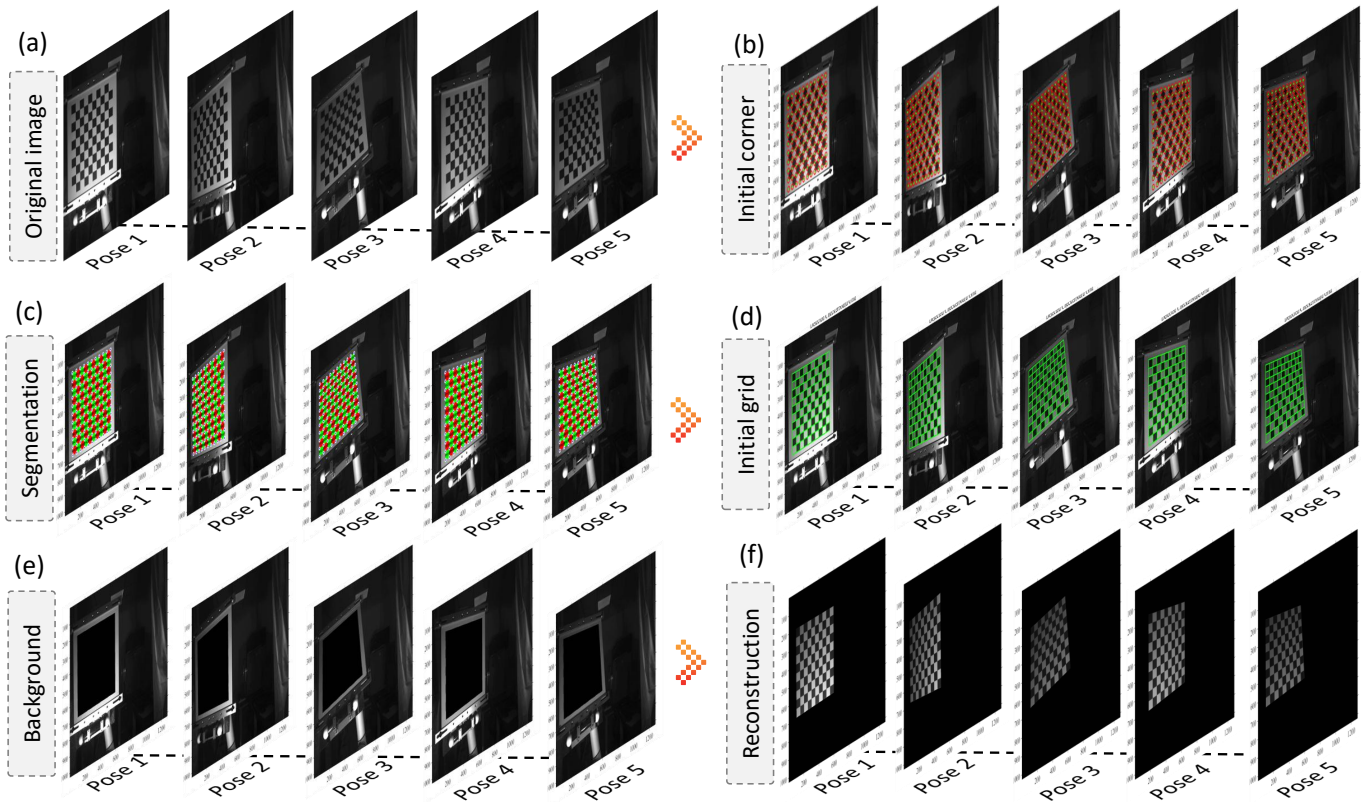

FIG. 5: Intermediate results of Module 1: geometry-aware checkerboard super-resolution (corresponding to Module 1 in Fig. E3 of the main article). (a) Original calibration image. (b) Initial corner response and coarse corner segmentation. (c) Estimated checkerboard region mask. (d) Homography-based initial grid reconstruction. (e) Background image with the checkerboard suppressed. (f) Super-resolved checkerboard after geometry-aware SR. These visualizations confirm that Module 1 sharpens the grid structure while effectively removing background clutter.

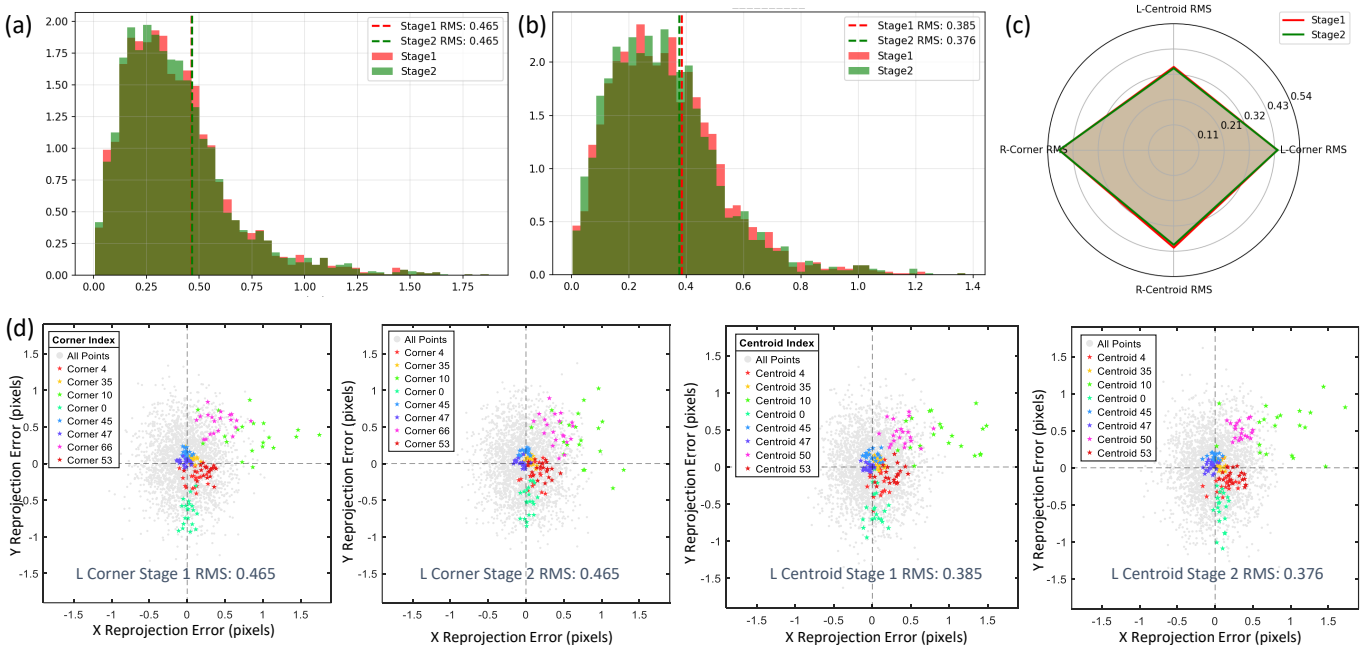

FIG. 6: Supplementary analysis for Fig. 3 in the main article. Reprojection error histograms in the x- and y-directions for the left camera over all calibration views, comparing Stage 1 and Stage 2. The corner reprojection RMS remains unchanged between the two stages (0.465 pixels), while the centroid reprojection RMS is slightly reduced (from 0.385 pixels to 0.376 pixels). This verifies that Stage 2 refines only the centroids and preserves the high corner accuracy obtained in Stage 1, so that corners keep their dominant role in the optimization.