Manual Calibration: Circle-band objective (contrast with background) and differential-evolution optimization

This note documents the objective used by `musclex/ui/ManualCalibrationDialog.py` in `ManualCalibrationDialog._circle_band_objective(center, radius, Q)` (the "circle band" optimizer).

1 What it computes

Given:

- circle center $c = (c_x, c_y)$
- radius r
- band scale $Q > 0$ (roughly a ring-width scale, in pixels)
- `objective_alpha` α
- `objective_bg_k` k
- `objective_nphi` N_ϕ (number of angular samples)

Let angles be uniformly sampled:

$$\phi_j = \frac{2\pi j}{N_\phi}, \quad j = 0, \ldots, N_\phi - 1$$

Let the bilinear-sampled image intensity at a point be $I(x, y)$.

1.1 Signal bands

The "signal" is sampled on 5 concentric circles near the candidate ring, using offsets

$$o \in \{-Q,\ -0.5Q,\ 0,\ 0.5Q,\ Q\}.$$

1.2 Background bands

The "background" is sampled on 2 concentric circles farther away (symmetric inside/outside), using offsets

$$o \in \{-kQ,\, +kQ\}.$$

1.3 Per-angle averaging

For each angular sample ϕ_j the signal and background intensities are averaged over the respective radial offsets:

$$s_j = \frac{1}{5} \sum_{o \in \mathrm{sig}} I(c_x + (r+o)\cos\phi_j,\ c_y + (r+o)\sin\phi_j)\,, \qquad b_j = \frac{1}{2} \sum_{o \in \mathrm{bg}} I(c_x + (r+o)\cos\phi_j,\ c_y + (r+o$$

1.4 MAD-based outlier rejection

Before computing the final means, outlier angles are rejected using Median Absolute Deviation (MAD) filtering on the per-angle signal values $\{s_j\}$:

1. Compute the median: $m = \mathrm{median}(s_j)$.
2. Compute the MAD: $\mathrm{MAD} = \mathrm{median}(|s_j - m|)$.
3. Define the "good" set $\mathcal{G} = \{j : |s_j - m| \le 3\,\mathrm{MAD}\}$. (If $\mathrm{MAD} < 10^{-10}$, all angles are kept.)
4. Apply the same angular mask to both signal and background for consistency.

1.5 Objective value (maximize)

The returned objective is computed over the surviving angles:

$$J(c, r; Q) = \frac{1}{|\mathcal{G}|} \sum_{j \in \mathcal{G}} s_j \;-\; \alpha\, \frac{1}{|\mathcal{G}|} \sum_{j \in \mathcal{G}} b_j.$$

2 Why subtract a symmetric background

2.1 The problem: "maximize the integration" is biased by radial background

If you maximize a raw ring integral/mean, you are effectively optimizing

$$\int (S(r,\phi) + B(r,\phi))\, d\phi,$$

where S is the ring/peak contribution you care about and B is baseline/background intensity.

In real diffraction images, B commonly varies strongly with radius (beam halo, small-angle scatter, detector response, broad diffuse scatter). That means the optimizer can increase the integral by moving the circle to a location where B is larger—even if the ring alignment is worse.

Concretely, small changes to center/radius can "ride" the background gradient and produce a larger sum/mean without actually matching the ring. This can pull the solution toward bright halos, broad low-q scatter, nearby rings with higher baseline, or detector shading artifacts.

2.2 What the background bands do

Subtracting $\alpha\,\mu_{\text{bg}}$ makes the objective approximate *local contrast*: "how much brighter is the ring neighborhood compared to nearby off-ring samples".

Sampling background at both $r - kQ$ and $r + kQ$ is important:

- **Symmetry cancels first-order radial slope:** if $B(r)$ is roughly smooth, then averaging inside/outside approximates the local baseline at r and reduces bias from dB/dr.
- **Stays local but avoids the peak:** using $\pm kQ$ ties the background separation to the ring-width scale Q, so the off-ring samples remain "nearby" across different rings/images while being far enough not to sit on the peak itself.

2.3 Why MAD outlier rejection?

Real diffraction images often contain beam stops, detector gaps, parasitic scatter, or saturated pixels that produce extreme intensity values at particular angles. Without filtering, a handful of such angles can dominate the mean and steer the optimizer toward solutions that maximize these artifacts rather than the ring contrast.

MAD filtering is chosen because:

- It is *robust*: the median and MAD are not affected by a minority of extreme values, unlike the standard deviation.
- It is *cheap*: only two passes over the per-angle array are needed (median, then median of absolute deviations).

- The threshold 3 MAD is a conventional choice for moderate outlier rejection without discarding too many valid samples.

The same angular mask is applied to both signal and background arrays so that the contrast estimate remains paired (each surviving angle contributes to both the numerator and the background penalty).

2.4 Why not subtract a full radial profile?

A full radial background model can work, but it requires additional assumptions (masking peaks, robust fits, handling anisotropy). The two-band symmetric sampling is a lightweight, optimization-friendly proxy that is local (responds to local baseline), robust to global illumination changes, and cheap to compute.

2.5 Assumptions and practical caveats

The interpretation above relies on two practical assumptions:

- **Background smoothness:** the baseline varies reasonably smoothly with radius so that sampling at $r \pm kQ$ approximates the local baseline at r.
- **Background samples are off-peak:** the offsets $\pm kQ$ are far enough from the ring so they do not land on the ring itself or a nearby ring/feature.

If rings are thick (e.g., $\sim$20 pixels wide) or nearby features exist, increase Q and/or k so the background bands are truly outside the ring, or set $\alpha = 0$ to disable the background term.

3 Optimization strategy: differential evolution

The optimizer in `ManualCalibrationDialog._optimize_center_radius(center, radius, Q)` refines the initial circle fit (from user-selected points) by maximizing the circle-band objective $J(c, r; Q)$ over three parameters: center (c_x, c_y) and radius r.

3.1 Search bounds

The search region is defined relative to the initial fit:

$$c_x \in [\hat{c}_x - \Delta c,\ \hat{c}_x + \Delta c], \qquad c_y \in [\hat{c}_y - \Delta c,\ \hat{c}_y + \Delta c], \qquad r \in [\hat{r}(1-\epsilon),\ \hat{r}(1+\epsilon)],$$

where $\Delta c = 100$ pixels and $\epsilon = 0.10$ (10%).

3.2 Solver

The optimization uses `scipy.optimize.differential_evolution` with the following settings:

- **Population size:** 100 (per parameter dimension).
- **Tolerance:** 10^{-4} (convergence criterion on objective improvement).
- **Polish:** enabled—after convergence, a local L-BFGS-B refinement is applied to the best solution.
- **Updating:** `deferred` (full-generation updates; better for smooth, non-noisy landscapes).
- **Seed:** 42 (reproducible results).

Because the objective is negated for the minimizer, the solver effectively *maximizes* J.

3.3 Why differential evolution?

The previous implementation used an alternating coordinate-descent approach (refine center $\rightarrow$ refine radius $\rightarrow$ repeat) with a grid-search over 8-connected neighbors and shrinking step sizes.

Differential evolution replaces this with a *global* optimizer that:

- Explores the full 3-D search region simultaneously, avoiding local optima that depend on the initial point quality.
- Does not require manual tuning of step sizes, shrink factors, or alternation schedules.
- Terminates with a local polish step that achieves sub-pixel precision.

The trade-off is higher computational cost per call (hundreds to thousands of objective evaluations), but for a one-shot calibration step this is acceptable—typical runs complete in under a second on modern hardware.

4 Parameter intuition

- Q**:** sets the "thickness" scale of the signal sampling band; also scales how far away background is measured.
- k **(`objective_bg_k`):** how far from the ring to sample background (in units of Q). Larger k reduces contamination from the ring peak but can make the background less "local".
- α **(`objective_alpha`):** weight of the background penalty. Higher α emphasizes contrast; lower α behaves more like raw intensity maximization.
- N_ϕ **(`objective_nphi`):** angular sampling density; higher values reduce noise at higher compute cost.