Aperture: Robust Vision-Based Active Optical Alignment - An Evidence-Aware Architecture, Evaluation Plan, and Integration Roadmap

Aaron Singh

Abstract **- Aperture studies closed-loop optical alignment from camera-space observations. It compares classical optimization, Bayesian search, and learned policies under sensor noise, optical tolerance variation, stage backlash, and settling constraints. The initial work uses the relign simulator and makes no claim about physical optical hardware. This paper defines the system model, evidence contract, risks, and staged validation plan. No physical-system or performance result is claimed without a committed reproducibility artifact.**

Index Terms - active alignment, optical systems, computer vision, reinforcement learning, sim-to-real

I. INTRODUCTION

Active alignment adjusts optical elements while observing a sensor-derived quality signal. A useful controller must converge from diverse initial conditions, minimize motion, and tolerate imaging and actuator imperfections. Aperture builds a shared benchmark in which controller quality and physical assumptions are recorded separately.

Research contribution

The contribution is not a performance claim. It is a reproducible reference architecture, a falsifiable evaluation protocol, and an evidence schema that records inputs, decisions, outputs, failure modes, and claim boundaries.

II. TECHNICAL CONTEXT

Active optical alignment is a sequential decision problem: a controller observes an image-derived objective, requests bounded stage motion, waits for settling, and observes again. The comparison is meaningful only when every controller receives the same initial poses, tolerance distributions, noise model, and motion budget.

III. RESEARCH QUESTIONS

1 Which controller reaches the best image quality with the fewest stage moves?

2 How well do learned policies generalize to unseen optical tolerances?

3 Which modeled errors dominate the gap between simulation and later hardware tests?

IV. SYSTEM MODEL AND ARCHITECTURE

A camera observation enters an image-quality estimator and controller. The controller emits constrained axis commands to Kinetic's stage model. After modeled motion and settling, the optical simulator generates the next observation. An evidence logger preserves frames, commands, objective values, and termination reasons.

Choose bounded motion u_t to maximize image quality q(I(x_t)) while minimizing travel, evaluations, and constraint violations.

The model is a design and test abstraction. It establishes invariants and expected-value checks; it does not convert simulated variables into measured hardware behavior.

V. DETAILED OPERATION AND RATIONALE

1 Reproduce a documented relign alignment environment.

2 Establish coordinate-descent, Nelder-Mead, and Bayesian-optimization baselines.

3 Train PPO or SAC policies with domain randomization.

4 Evaluate every controller on identical held-out starts and tolerance distributions.

VI. DATA AND EVIDENCE CONTRACT

The primary data are simulated sensor frames and alignment trajectories generated by relign. A later hardware phase may add calibration-target captures, but those measurements will be reported separately from simulation.

Each run must preserve a machine-readable configuration, random seed where applicable, source revision, environment description, raw result, summary method, and known limitations. Derived plots must point back to the exact records used to produce them.

VII. IMPLEMENTATION BLUEPRINT

Optical simulator. Maps element pose and tolerances to sensor observations. Its acceptance artifact is seeded scene and tolerance manifest.

Quality estimator. Computes focus and alignment objectives. Its acceptance artifact is frame-level scores and calibration.

Controller suite. Runs classical, Bayesian, and learned policies. Its acceptance artifact is action and objective trajectory.

Kinetic stage adapter. Applies backlash, limits, and settling behavior. Its acceptance artifact is commanded versus realized pose trace.

VIII. EXPERIMENTAL METHODOLOGY

Evaluation proceeds in four gates. Gate 1 checks deterministic behavior and invariants. Gate 2 applies controlled noise, tolerance, and injected-failure sweeps. Gate 3 repeats measurements and reports distributions rather than a favorable single run. Gate 4 introduces an external data source or bounded lab adapter only after the first three gates are reviewable.

Primary metrics are reported with sample counts, aggregation rules, uncertainty, and failure cases.

IX. REPRODUCIBILITY AND ACCEPTANCE CRITERIA

A result is accepted only when another clean environment can reproduce the reference behavior from a named revision. Every comparison uses identical budgets and held-out conditions. Failed runs remain part of the evidence record. Unknown hardware, calibration, and production conditions are labeled unknown rather than assigned synthetic values.

X. FAULT INJECTION AND NEGATIVE TESTING

Negative testing is part of the main evaluation rather than an appendix. Each risk is converted into an explicit scenario with an expected detection, rejection, abstention, or safe-state outcome. A silent success

under missing or contradictory state is treated as a failed test.

Simulator bias. The failure consequence is controller exploits unrealistic optics. The required control is held-out tolerance families.

Objective mismatch. The failure consequence is sharpness improves while alignment worsens. The required control is multiple optical metrics.

Stage non-idealities. The failure consequence is policy fails after direction changes. The required control is backlash and settling sweeps.

Unsafe motion. The failure consequence is travel beyond a physical envelope. The required control is hard limits and independent stop.

XI. ANALYSIS AND INTERPRETATION

Analysis reports distributions, failure counts, and representative traces rather than a single favorable score. Baselines share the same inputs, budgets, stopping rules, and held-out conditions. When multiple configurations are explored, the selection procedure is recorded so the test set is not used as a tuning signal.

The primary metric families are Final alignment error, Image-quality score, Stage-movement count, Time to convergence, Success rate under noise. Each is reported with its unit, aggregation rule, sample count, spread, and failed or undefined scenarios. An improvement is meaningful only when reference correctness is preserved and the result repeats across declared evaluation seeds or groups.

XII. INTEGRATION INTERFACES

Integration occurs through versioned records rather than hidden shared state. Inputs declare units, coordinate frames or topology identifiers, timestamps, validity windows, and provenance. Outputs include a status, result payload, limitation flags, and a stable link to the run configuration.

Adapters are introduced in read-only or simulated form first. A live adapter cannot inherit the trust assigned to the reference model; it requires separate authentication, calibration, timeout, retry, and safe-state evidence.

XIII. EXTERNAL VALIDITY AND RELEASE CRITERIA

The initial conclusions are bounded by the declared simulator, public data, protocol mocks, and sampled conditions. They do not automatically generalize to other equipment, optical assemblies, wafers, stages, power systems, operating ranges, or production environments. External validity is evaluated as a new experiment, not assumed from architectural similarity.

A portfolio release requires the paper source, generated PDF, configuration schema, reproducible entry point, tests, raw or referenced evidence, and a concise limitations record. If one of these artifacts is missing, the project may remain a research design but cannot be presented as an evidence-backed implementation.

XIV. CLAIM BOUNDARIES, SAFETY, AND RISKS

The first phase is simulation-only. It does not establish real lens alignment accuracy, camera calibration quality, stage repeatability, or sim-to-real transfer. Any hardware experiment requires travel limits and an independent stop mechanism.

The primary research risk is a category error: treating simulation, public benchmark data, or protocol mocks as proof of physical-system performance. The mitigation is provenance at every boundary, fail-closed behavior for missing state, and an explicit review before any live interface is enabled.

XV. GOALS, MILESTONES, AND INTEGRATION

1 Reproduce the two-lens relign baseline.

2 Add classical controller comparisons.

3 Integrate the Kinetic non-ideal stage model.

4 Train and evaluate domain-randomized policies.

5 Validate a bounded camera-stage experiment with explicit interlocks.

XVI. CONCLUSION

Aperture now has a research plan comparable in rigor and presentation to the established portfolio papers. Its architecture is inspectable, its evaluation is falsifiable, and its limitations are explicit. Credibility will come from completing the staged evidence artifacts, not from promoting unverified numbers.

REFERENCES

[1] HS Kempten, 'relign: Python software for simulating the active alignment of optical systems,' GitHub.

[2] Aaron Singh, 'Aperture research specification,' local portfolio repository, 2026.

TABLE I
SYSTEM COMPONENTS AND EVIDENCE RESPONSIBILITIES

Component	Responsibility	Required evidence
Optical simulator	Maps element pose and tolerances to sensor observations	Seeded scene and tolerance manifest
Quality estimator	Computes focus and alignment objectives	Frame-level scores and calibration
Controller suite	Runs classical, Bayesian, and learned policies	Action and objective trajectory
Kinetic stage adapter	Applies backlash, limits, and settling behavior	Commanded versus realized pose trace

TABLE II
PLANNED EVALUATION MATRIX

Metric family	Acceptance record
Final alignment error	Repeated result with configuration, seed, revision, and limitations
Image-quality score	Repeated result with configuration, seed, revision, and limitations
Stage-movement count	Repeated result with configuration, seed, revision, and limitations
Time to convergence	Repeated result with configuration, seed, revision, and limitations
Success rate under noise	Repeated result with configuration, seed, revision, and limitations

TABLE III
RISK REGISTER AND CONTROLS

Risk	Failure consequence	Control
Simulator bias	Controller exploits unrealistic optics	Held-out tolerance families
Objective mismatch	Sharpness improves while alignment worsens	Multiple optical metrics
Stage non-idealities	Policy fails after direction changes	Backlash and settling sweeps
Unsafe motion	Travel beyond a physical envelope	Hard limits and independent stop

TABLE IV
PROJECT GOALS AND REQUIRED PROOF

ID	Goal	Completion evidence
G1	Reproduce the two-lens relign baseline.	Reviewable source, test, trace, or measurement artifact
G2	Add classical controller comparisons.	Reviewable source, test, trace, or measurement artifact
G3	Integrate the Kinetic non-ideal stage model.	Reviewable source, test, trace, or measurement artifact
G4	Train and evaluate domain-randomized policies.	Reviewable source, test, trace, or measurement artifact
G5	Validate a bounded camera-stage experiment with explicit interlocks.	Reviewable source, test, trace, or measurement artifact