Toward Verifiable Sensor Data for Scientific and AI Systems

A 3D Reconstruction Case Study and a Framework for Conditional Assurance

Richard Goodman and Stephanie Goodman
The Institute For Applied Ontological Mathematics
Research technical report | 2 November 2026

Abstract

Scientific and AI systems increasingly consume reconstructed geometry and learned representations whose visual plausibility can conceal errors in acquisition, calibration or computation. We describe an IAOM research program that connects an operational 3D capture workflow to a proposed framework for conditional assurance. A six-position room-capture case study produced panoramic tours, a fused point cloud and a navigable but visually soft Gaussian-splat model. The engineering record reports a discrepancy of approximately 116 degrees between an assumed camera-rig orientation model and image-derived relative rotations. Re-estimation improved internal consistency, while cross-position image-pose recovery and a controlled rendering comparison remain unfinished. A companion Apiary implementation supplies finite exact-rational contracts for calibration, sensor response, record-bound processing and aggregation. Together, these projects motivate explicit contracts for sensor data, calibration, coordinate transforms, numerical semantics, learned operators and downstream acceptance. We distinguish demonstrated engineering observations from reported formal results in a separate computation program and from the planned integration of those results into the camera pipeline. Four elementary propositions specify conditional composition, calibrated range enclosures, geometric error propagation and the effect of additional linearized measurements. We explain how Agda source proofs, MAlonzo extraction, Lean developments and selected ONNX operator semantics can contribute without removing compilation, hardware, physical-model or statistical assumptions. The proposed ingestion boundary accepts a claim only within its declared scope and preserves refusal or unverified status otherwise. The resulting research agenda connects additional instrumentation to testable improvements in uncertainty, observability and replayable evidence, rather than treating formal computation as proof of physical truth.

Keywords: scientific sensing; formal verification; 3D reconstruction; measurement uncertainty; AI ingestion; Agda; Lean; provenance; machine learning assurance.

1 Introduction

A measurement passes through several interpretations before an AI system acts on it. A sensor produces encoded values; software assigns units and coordinate frames; calibration supplies parameters; reconstruction combines observations; and a learned model produces a prediction or representation. Each transition can preserve some facts while introducing assumptions. A convincing image does not reveal whether those assumptions were satisfied. Conversely, a

checked theorem about a transformation does not establish that the transformation received an accurate account of the physical scene.

Our objective is to make these transitions explicit and checkable. We use the term ingestion node for the boundary at which a downstream scientific or AI system receives data together with the information needed to judge a specified claim about it. Such a node may supply geometry, imagery, a derived feature or a learned prediction. Its useful output includes the value, its interpretation, its supporting evidence and any remaining conditions. A valid refusal is also an output: the consumer must be able to distinguish an unavailable or unsupported claim from an accepted one.

Two complementary applications motivate this boundary. The camera work exposes errors in geometric interpretation. Apiary, a companion IAOM project, supplies finite constructive contracts for thermal response, mass calibration, record binding and aggregation [1]. The latter provides a concrete software foundation for preserving measurement meaning across stages. Extending that foundation into sensor-side processing, vision and learned models is the shared program developed here.

The immediate experimental basis is an owner-operated Matterport MC250 capture workflow developed at IAOM. Its Python software controls acquisition, decodes images, constructs and registers point clouds, and produces panoramic and Gaussian-splat representations. A separate IAOM program develops constructive mathematics and executable specifications using Agda and Lean. The internal engineering record describes results in that program, including selected neural-network semantics and exact-arithmetic execution examples, but identifies their integration into the camera pipeline as planned [2]. This distinction governs every assurance claim in this report.

The paper makes three contributions. First, it presents a preliminary reconstruction failure analysis that separates within-sweep rig consistency from the unresolved recovery of camera poses across tripod positions. Second, it specifies a framework for composing physical assumptions, computation contracts and artifact evidence at a sensor-to-model boundary. Third, it gives an evaluation program in which each additional hardware capability resolves a named experimental question. The mathematical propositions clarify these contracts; they are elementary analytical derivations, not claims of new general mathematics or completed mechanization.

The intended applications include scientific reconstruction, industrial inspection, reproducible AI datasets and accountable sensor processing in demanding environments. The contribution is an approach to producing assessable evidence. It is not a claim of regulatory certification, universal perception accuracy, or a completed formally verified instrument.

2 Related work and the meaning of assurance

Multi-view geometry provides the established foundations for projection, homographies, triangulation and bundle adjustment [3]. Incremental Structure-from-Motion, as implemented in COLMAP, offers a mature approach to camera registration and reconstruction [4]. Gaussian splatting optimizes a scene representation for view synthesis, and gsplat supplies an open-source implementation framework [5, 6]. These methods provide relevant computational machinery; their use does not by itself establish metric accuracy or the correctness of an implementation.

Certification already has specific meanings in geometric estimation. SE-Sync and TEASER provide certifiable estimation or registration results under stated mathematical conditions [7, 8]. An optimality certificate addresses a defined objective and model. It does not independently validate sensor calibration, authenticate the original observations, or prove that an arbitrary compiled program implements the certified algorithm. The present work therefore does not claim to originate certified perception. Its emphasis is the explicit composition of several different forms of evidence.

Proof-carrying code and proof-carrying data established approaches in which an artifact is accompanied by evidence checked against a policy or statement [9, 10]. Our proposed ingestion record is inspired by this separation of producer and checker. It is not an implementation of a particular cryptographic proof-carrying-data protocol, and a log containing hashes is not automatically a proof of execution. Structured assurance cases similarly distinguish claims, assumptions and supporting evidence [11]; an assurance argument may incorporate both empirical results and formal proofs without confusing their roles.

Metrology supplies another essential boundary. A calibrated value is an estimate under a measurement model, accompanied by an evaluation of uncertainty. Traceability concerns a measurement result and requires a documented chain to a reference, with the calibrations contributing to uncertainty [12, 13, 14, 15]. The digital meaning of a number and the physical credibility of that number must consequently remain separate obligations.

Categorical accounts of learning and backpropagation provide relevant mathematical foundations [16, 17]. Verification of a neural network may instead concern an input-output property over a specified domain, as in Reluplex [18]. Neither the algebraic organization of a network nor correct evaluation of its forward map establishes the scientific adequacy of its predictions. ONNX provides a versioned representation of model graphs and operators, which is useful for defining a supported import boundary [19, 20].

Existing instrumentation standards also matter. GenICam defines device interfaces and feature conventions, while GenDC provides a transport-independent container spanning imagery, multidimensional measurements and metadata [21]. EMVA 1288 provides camera and sensor characterization methods [22]. A useful assurance layer should bind claims to these representations where appropriate, rather than replace them with undocumented conventions. No conformance to either standard is claimed for the present prototype.

W3C PROV-DM represents entities, activities, agents and derivation relationships. OGC SensorThings specifies observation structure, including sensors, datastreams, observed properties and features of interest [23, 24]. These are useful interoperability foundations. The proposed layer binds a particular predicate and its assumptions to that structure; metadata or interface conformance alone does not establish computational correctness or physical validity. No implementation conformance to these standards is claimed here.

3 The capture system and its current evidence

3.1 Instrument and processing workflow

The instrument is an IAOM-owned Matterport MC250, identified by the manufacturer as the Pro2. The manufacturer describes Pro2 depth sensing as structured light [25]. We operate the device through its local-network interface using owner-side software. This report concerns acquisition and processing; it does not claim manufacturer endorsement or specify a new device protocol.

The engineering record describes a sweep lasting approximately 25 seconds, with 18 RGB frames at 2560 × 1920 pixels arranged across six yaw directions and three elevation bands. These are observed characteristics of the reported workflow, not a replacement for the manufacturer's specifications. Six tripod positions supplied the room case study. A complete run under that arrangement would contain 108 RGB frames; a released capture manifest would be needed to establish the exact retained frame set [2].

The software separates connection supervision, capture storage, image decoding, geometry, registration, scale estimation and coverage reasoning. Connection supervision reports measured reachability and device state. Capture records associate raw material with stable identifiers. Geometry maps depth samples into points; registration combines several positions using coarse alignment, Iterative Closest Point and pose-graph optimization. These functions are implemented in the conventional processing pipeline. Their camera-specific Agda counterparts are future work [2].

The reported products are interactive panoramic tours, a fused point cloud with a target-referenced scale, and a navigable Gaussian-splat model. Panoramas were the clearest visual product in this experiment. The splat pipeline ran on the program's ARM-based GPU workstation using gsplat, but the reconstruction remained soft. Near-plane and radius culling addressed a reported rendering instability associated with Gaussians close to the camera. The record does not provide controlled image-quality, geometric-accuracy or throughput benchmarks [2].

3.2 Geometric and calibration model

For a depth sample, let c be the encoded nonnegative range value, s a scale in metres per count, and u a unit bearing determined by intrinsics, distortion and sensor orientation. Let R and t map that sensor observation into the selected world frame. A simplified point model is

$$r = s\ c, \qquad p = R(r\ u) + t. \tag{1}$$

A real rotating rig requires sensor-specific optical-center offsets, head-angle conventions, timing and frame transforms. Equation (1) represents their composed effect only when R and t are correctly supplied. The direction model and the assumption that a single s is sufficient are empirical modeling choices. Neither shape nor angle is exact merely because the equations are exact.

The reported workflow estimated s = 0.2445 mm per count using a target of known distance [2]. We call this a target-referenced scale estimate. The engineering record does not include the target's calibration certificate or dimensional uncertainty, the endpoint observations, the residual distribution, or held-out distances. Accordingly, the value is not presented as a traceable metrological certification. Its useful methodological feature is that the software requires an external scale reference before asserting an absolute dimension.

For positive unscaled target separations D_i, reference lengths L_i and fixed nonnegative weights w_i, a scalar least-squares model has estimate

$$\hat{s} = \frac{\sum_i w_i\ D_i\ L_i}{\sum_i w_i\ D_i^2}, \quad \text{provided } \sum_i w_i\ D_i^2 > 0. \tag{2}$$

Equation (2) specifies one candidate calibration procedure; it is not a claim that this exact estimator was used in the reported run. With uncertainty in D_i as well as L_i, the statistical model requires additional care. Independent targets, varied ranges and sensor-specific residuals should test whether a shared scale is adequate before a stronger measurement claim is accepted.

3.3 Evidence status

Table 1 uses demonstrated to mean reported as built and exercised in the engineering record, in progress to mean partially completed with an identified remaining task, and planned to mean a proposed integration. It separately identifies evidence scope. A narrative report of an internal theorem or experiment is not the same evidence as a public proof artifact or an independently replayed run.

Table 1. Status and scope of the principal capabilities

Capability	Reported status	Scope of evidence
Local capture, image decoding and panoramic tours	Demonstrated	Operational observations in the engineering record
Depth reconstruction, registration and scale estimation	Demonstrated	Six-position case; independent metric validation absent

Capability	Reported status	Scope of evidence
Gaussian-splat training and navigation	Demonstrated	Working pipeline; visual quality limited
Rig-model revision from image relationships	Demonstrated	Reported internal consistency improvement
Cross-position image poses and sharp splat	In progress	Incremental SfM proposed; outcome unreported
Agda extraction and selected Lean or ML results	Demonstrated in separate program	Author-reported results; theorem and run artifacts not included
Apiary finite measurement contracts	Implemented software foundation	Named declarations, local reference replay and mutation results reported in companion manuscript
Direct sensor execution and proof binding	In progress	Current extension of the measurement-contract program; hardware acceptance results not reported here
FPGA scientific-data processing example	Demonstrated in separate program	Pinned-data proof of concept; camera integration absent
Formal camera geometry and supported vision ML	Planned	Contracts and integration targets
Integrated sensor-to-model assurance	Planned	Research architecture and evaluation program

4 A camera orientation inconsistency as a case study

4.1 Failure and diagnostic reasoning

Gaussian-splat optimization relies on the relationship between scene points and input camera poses. The case study combined a recognizably room-shaped depth cloud with a splat that could be navigated but did not resolve scene appearance clearly. The team investigated the photographic pose model and reported disagreement of approximately 90–117 degrees between its relative rotations and image-derived estimates, with an approximately 116-degree example [2]. This is a large internal inconsistency, sufficient to motivate correcting the pose model. It is not an independently measured error against a surveyed orientation reference.

The diagnostic separated two estimation problems. Within a sweep, the repeated head motion should produce a consistent relationship between photo indices. Across sweeps, each tripod position has an independent pose relative to the room. The team pooled selected within-sweep image relationships across six positions and robustly averaged their relative-rotation estimates. Agreement across positions supported the existence of a reusable rig relationship, while leaving the global placement of the positions unresolved.

For ideal images formed from a common optical center under pure rotation, the calibrated homography relation is $H \sim K_2 R_{21} K_1^{-1}$ [3]. A common tripod position does not establish that

condition: different sensors and a rotating head may have displaced optical centers. Planar scene structure can also produce a homography for other motion. The reported analysis should therefore be interpreted as a rotation-based diagnostic whose applicability depends on intrinsics, distortion, scene structure and residual translation. A subsequent validation must test those conditions rather than treating the estimated rotations as exact ground truth.

4.2 Reported observations and their limits

Table 2 summarizes the diagnostic quantities reported for the room case.

Table 2. Diagnostic quantities reported for the room case

Quantity	Reported observation	Interpretation
Tripod positions	6	Single room case, not a multi-scene benchmark
Model versus estimated relative rotation	Approximately 90–117 degrees; example 116 degrees	Internal model disagreement
Selected rotation-cycle residuals	Approximately 6 degrees	Consistency diagnostic; sample distribution unavailable
Relative-rotation agreement across sweeps	Approximately 3 degrees; best pairs below 1 degree	Repeatability, not absolute orientation accuracy
Filtered image correspondences	Approximately 31,000	Pairwise correspondence count, not an established count of independent tracks
Reported position separations	0.9–5.0 m; median approximately 2 m	Viewpoint diversity; positional uncertainty unreported

These observations come from the engineering record [2]. The record does not specify the angular-distance convention, pair and cycle counts, matching thresholds, full residual distributions or a release of the estimation code. It would be incorrect to turn the agreement figures into accuracy bars, or to infer that match volume alone establishes a well-conditioned global reconstruction. Spatial coverage, repeated texture, track connectivity, occlusion, intrinsics and initialization all remain relevant.

The revised within-sweep rig reportedly reduced disagreement with the image-derived estimates to a few degrees. The available record does not include a controlled comparison of splats before and after the revision with all other settings fixed. The evidence supports a substantial pose-model defect and a plausible source of blur; it does not establish that this defect was the only cause or that photorealistic reconstruction is one guaranteed step away.

4.3 The next reconstruction experiment

The remaining task is to recover image-camera poses across tripod positions and evaluate their effect on reconstruction. The team's pairwise initialization and global optimization attempts did not produce a satisfactory solution. Incremental SfM is a well-motivated next experiment:

initialize a suitable pair, register additional images from 2D–3D correspondences, and refine poses and structure through bundle adjustment [4]. A rig-aware configuration should preserve validated relationships without forcing an inaccurate rigid model.

An available x86 Linux host would offer a convenient environment for this experiment because the team reported local package difficulties on its ARM workstation [2]. This is an environment and throughput rationale, not a claim that COLMAP intrinsically requires x86 or that another workstation guarantees geometric success. Alternative initialization strategies and direct use of registered depth information should be compared when their assumptions can be made explicit.

A controlled follow-up should hold images, train/test split and optimization budget fixed while comparing the original pose model, revised rig initialization and final cross-position solution. It should report registered-camera fraction, reprojection distributions, independent geometry checks and held-out rendering quality. Those measurements would establish how much the revision explains, and what defects remain.

5 A proposed boundary for verifiable ingestion

5.1 Separate physical validity from computation and identity

The proposed architecture associates each claim with three kinds of obligation. Physical obligations describe the sensor, calibration, environment, timing and domain in which the measurement model is applicable. Computational obligations describe the transformation, numerical representation, theorem or test evidence, and supported inputs. Identity obligations bind those claims to the actual raw data, parameters, executable and output. All three are necessary when a consumer needs to rely on a result.

A hash can identify the bytes covered by a claim under a collision-resistance assumption. It does not establish that those bytes originated at the stated sensor. A signature can establish a statement by a key holder, subject to key-management and identity assumptions; it does not make the measurement accurate. Similarly, proof checking can establish a formal proposition while leaving open whether that proposition captures the intended physical or operational requirement. These distinctions should remain visible to the consumer. Figure 1 summarizes the proposed architecture.

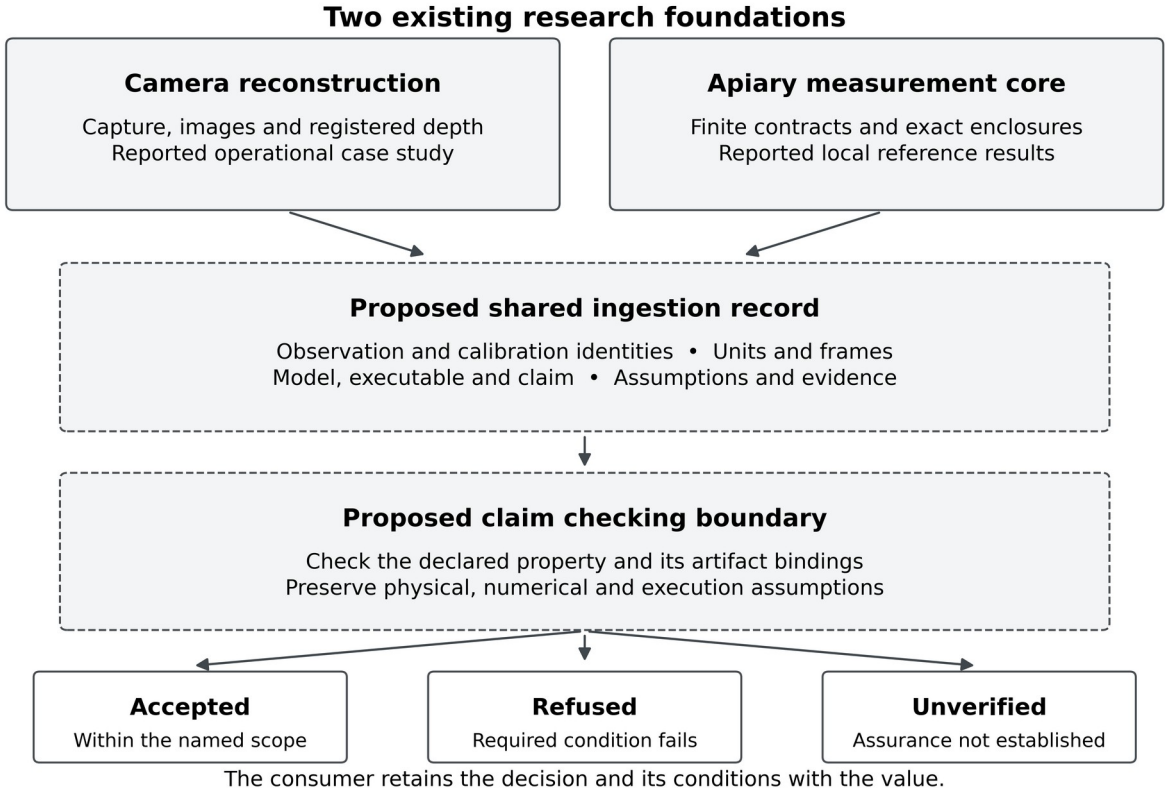


Figure 1. Proposed ingestion architecture. The current capture workflow supplies the experimental starting point. The evidence record, checking boundary and integrated guarantees are research targets; arrows do not assert completed formal refinement.

The initial threat and failure model includes missing or reordered frames, stale or substituted calibration, coordinate-convention mistakes, unsupported operators, model or weight substitution, numerical-domain violations, and replay of an output under a different input identifier. It also includes accidental defects such as the reported pose inconsistency. The proposed software boundary alone does not establish integrity of camera firmware, sensor electronics, clocks or an adversarially manipulated physical scene. Those require their own evidence and trust assumptions.

5.2 The proposed result record

Table 3 sets out the information required to assess a claimed result.

Table 3. Information required to assess a claimed result

Record component	Required content
Observation identity	Capture identifier; sensor identifier; raw-data digests; sequence and time information
Measurement meaning	Units; coordinate convention; validity domain; calibration identifier; uncertainty representation
Transformation identity	Source and executable revisions; parameters; model and weights digests; numerical semantics
Claim and evidence	Exact predicate or theorem identifier; assumptions; checker version; proof or test artifacts
Execution context	Compiler and backend versions; relevant runtime and hardware configuration; input/output bindings
Decision	Accepted, refused or unverified for a named claim; reason and usable scope

The same data may support an accepted claim about file completeness and an unverified claim about physical distance. Status is therefore attached to a claim, not just to a file. A consumer seeking metric geometry must not infer that geometry is certified because a checksum check succeeded. The record should expose the predicates that were checked, the evidence used and the conditions that were not evaluated.

Accepted means that the declared acceptance policy has been satisfied for the named claim. Refused means that a required condition failed or could not be established under that policy. Unverified means that the value is available without the specified assurance. The latter may still be useful for exploration, but downstream software must not silently promote it. Refusal should preserve enough diagnostics to permit repair while avoiding invented estimates.

5.3 Proposition 1 on conditional composition

Let $X_0, \ldots, X_n$ be stage types, $f_i : X_{i-1} \to X_i$ total functions, and P_i predicates on the respective types. Assume each stage satisfies the contract below for all of its inputs.

$$P_{i-1}(x) \Rightarrow P_i\big(f_i(x)\big), \quad \text{for } i = 1, \ldots, n. \tag{3}$$

Proposition 1. Under these assumptions, $P_0(x)$ implies $P_n((f_n \circ \ldots \circ f_1)(x))$.

Proof. For zero stages, the claim is the identity implication. If the predicate holds after the first n − 1 stages, the contract for stage n establishes the final predicate. Induction completes the proof.

For processing that may refuse, the proposition applies to a fully accepted path through a result type containing either a value with evidence or a refusal reason. The formal interpretation must specify how each evidence object establishes the corresponding predicate. Correct artifact

binding and faithful execution of each f_i remain separate obligations. The proposition transfers stated predicates; it does not invent physical evidence for P_0 or prove the soundness of an unspecified checker.

All four propositions in this report are analytical arguments written to expose the required contracts. Their mechanization in the camera system is planned. They should not be cited as completed Agda theorems or as the novel invention of the underlying mathematical facts.

6 Measurement models and numerical obligations

6.1 Proposition 2 on scale and range enclosures

Assume a reference length L lies in $[L_-, L_+]$ with $0 < L_- \le L_+$, and a reconstructed count-distance D lies in $[D_-, D_+]$ with $0 < D_- \le D_+$. Under the scalar model $s = L/D$, the admissible scale is enclosed by

$$s \in \left[\frac{L_-}{D_+}, \frac{L_+}{D_-}\right]. \tag{4}$$

For a count c in $[c_-, c_+]$ with $0 \le c_- \le c_+$, Proposition 2 states that the associated range $r = sc$ is enclosed by

$$r \in \left[\frac{L_-}{D_+}c_-, \frac{L_+}{D_-}c_+\right]. \tag{5}$$

Proof. Positive division is increasing in its numerator and decreasing in its denominator. Multiplication of nonnegative quantities is increasing in both arguments. Applying these monotonicity properties gives (4) and (5). A numerical implementation must use exact arithmetic or outward rounding so that its computed endpoints preserve the enclosure.

The proposition assumes that a single scale actually describes the observations. Channel-dependent bias, distortion, range-dependent error or incorrect feature correspondences can invalidate that model. Intersecting scale intervals from additional target observations with jointly valid bounds provides a useful consistency check: an empty intersection rejects the shared-scale model under the declared bounds. A nonempty intersection alone does not prove the bounds or establish traceability.

6.2 Proposition 3 on geometric error propagation

Let $p = R(ru) + t$ and $\hat{p} = \hat{R}(\hat{r}\hat{u}) + \hat{t}$, where R and $\hat{R}$ are rotations, u and $\hat{u}$ are unit directions, and ranges are nonnegative. Suppose the range, direction, rotation and translation discrepancies satisfy the following bounds, with the matrix norm induced by the Euclidean norm.

$$|r - \hat{r}| \le \varepsilon_r, \quad \|u - \hat{u}\| \le \varepsilon_u, \quad \|R - \hat{R}\|_2 \le \varepsilon_R, \quad \|t - \hat{t}\| \le \varepsilon_t. \tag{6}$$

Proposition 3. The resulting point discrepancy is bounded by

$$\|p - \hat{p}\| \leq \varepsilon_t + \varepsilon_r + \hat{r}\ \varepsilon_u + \hat{r}\ \varepsilon_R. \tag{7}$$

Proof. Expand the difference as R((r − r̂)u + r̂(u − û)) + (R − R̂)r̂û + (t − t̂). Apply the triangle inequality, norm preservation by R, the unit lengths of u and û, and the induced matrix-norm bound. This yields (7).

The expression makes the range dependence of bearing and orientation errors explicit. Improving raw range resolution may have little practical effect if pose error dominates. The proposition does not supply any of the ε values: those require justified calibration bounds or explicitly probabilistic statements. A computation using imperfectly orthogonal stored matrices needs the corresponding matrix-norm factor, and an implementation error bound must be added for numerical evaluation. Probabilistic coverage intervals must not be presented as deterministic guarantees without their probability assumptions.

6.3 Exact arithmetic and finite precision

Exact rational arithmetic can represent useful calibration quantities and algebraic fragments, but it does not make general trigonometric geometry rational or inexpensive. Floating-point implementations introduce rounding, exceptional values, overflow, underflow and platform-dependent evaluation choices. Work such as Flocq demonstrates the importance of specifying the relevant arithmetic model when proving numerical properties [26].

Three implementation routes are worth comparing: exact arithmetic for a small reference core; interval or bounded-error implementations for geometric calculations; and conventional accelerated computation followed by a sound checker for a precisely stated property. These routes offer different costs and guarantees. A checker must validate a meaningful property independently of the producer's asserted answer. Agreeing with a second implementation on selected cases is valuable test evidence, but it is not a universal semantic-preservation theorem.

6.4 Proposition 4 on additional measurements

Consider a fixed linearized scene model with measurement operators A and B. Define Cx = (Ax, Bx), the stacked observations. Proposition 4 states

$$\ker C = \ker A \cap \ker B \subseteq \ker A. \tag{8}$$

Proof. Cx = 0 holds exactly when both Ax = 0 and Bx = 0. The intersection identity and inclusion follow. The inclusion is strict only when B detects at least one direction that A cannot observe.

For a nonlinear sensor, A and B may be Jacobians at a declared scene and calibration state. The conclusion is then local; occlusion, correspondence changes and global ambiguity remain outside it. Additional viewpoints or modalities can reduce unobservable directions, but redundant measurements need not do so. Even in the unweighted linear setting, an increased smallest

singular value does not imply a better condition number, because the largest singular value can increase faster.

This proposition gives a precise equipment rationale. Before adding a sensor, identify the previously unresolved direction or uncertainty contribution it is expected to address, then measure whether it does so. A quotient by a linear nullspace captures deterministic indistinguishability; it is not automatically a statistical minimal sufficient statistic. Statistical sufficiency requires a specified probability family and parameter of interest.

7 From checked specifications to executable systems

7.1 The Agda and Lean boundaries

The broader IAOM program reports constructive specifications in Agda and extraction through the MAlonzo Haskell backend, followed by native compilation with GHC [2]. Agda's safe mode excludes several mechanisms that would allow unchecked assumptions or definitions and imposes restrictions on imports [27]. The compiler documentation describes the translation route [28]. These facts establish a practical path from checked source programs to execution, but do not establish that the complete compilation chain is itself formally verified.

A source-level correctness argument has at least three obligations: the specification captures the intended property; the checking implementation accepts a valid derivation under the declared assumptions; and the executed program preserves the relevant source semantics. Compiler-verification work makes the last obligation explicit [29]. In the present pipeline, MAlonzo, GHC, runtime behavior, foreign interfaces, libraries, operating system and hardware must be recorded in the execution trust base unless a separate preservation result or sound validation method covers them. Proof terms may be erased during compilation; a binary need not literally contain the proof.

Lean supplies a distinct checking environment. The engineering record reports approximately 7,100 lines across more than 50 modules in its categorical-learning development [2]. Those counts describe development scope, not proof strength. Public claims of axiom-free results should identify exact declarations, dependency revisions and transitive axiom reports, together with reproducible proof checking. Lean's own validation guidance emphasizes inspection of both statements and dependencies [30]. Token searches for particular words are insufficient, and constructs such as classical reasoning are not automatically unsound.

7.2 Evidence that tests the checker

A credible workflow binds checks to the exact artifact used. A build receipt should identify source, dependencies, configuration and executable; a run receipt should identify input, parameters and output. The evidence type must remain explicit: a theorem certificate, an execution attestation, a differential test and an ordinary log establish different things.

The broader program reports a previous failure in which a gate accepted a trivial relation that merely echoed expected data, followed by a repair using actual reduction and independent comparison paths [2]. The transferable lesson is to test whether a checker rejects a meaningful false claim. Mutations should include a changed input, stale calibration, altered weights, a missing derivative factor and an unsupported operator. A finite mutation suite cannot prove checker soundness, but it can expose vacuous checks and incorrect artifact binding.

A camera-facing implementation should initially formalize a small set of high-value contracts: typed units and frames, nondegenerate scale estimation, rigid-transform composition, bounds on supported numerical operations and preservation of explicit refusal. Expanding the certified boundary stage by stage produces assessable claims earlier than attempting an undifferentiated proof of the entire vision stack.

7.3 Implemented measurement contracts in Apiary

The companion Apiary manuscript makes the measurement foundation more specific [1]. It describes an exact-rational core checked with Agda 2.8.0 in safe mode with `--without-K`, a vendored standard library at version 2.3, and executable reference paths through MAlonzo and GHC 9.12.2. Named declarations connect numerical results to realizations of declared measurement models, rather than merely asserting that an interval is internally well formed. These are reported results of that software project, not new experiments conducted for this camera report. Table 4 summarizes the declarations and their proposed relevance to sensing.

Table 4. Apiary contracts and their relevance to the sensor program

Reported declaration	Established scope in the companion account	Transfer target
`thermalStepSound`	Encloses a discrete thermal-response step under explicit premises	Sensor dynamics and sampling-domain contracts
`massEnclosureSound`	Encloses an affine calibration realization with bounded errors	Calibrated raw-count interpretation
`chainSound`	Binds processing-stage evidence to identified input and result records	Camera frames, calibration versions and transform chains
`weightedMeanSound`	Encloses weighted aggregation of admitted contributor values	Time-window or multi-observation summaries
`windowDifferenceSound`	Encloses a difference between compatible aggregate intervals	Change estimates with preserved units and evidence

The companion reports replay of 12 native records across two reference configurations and adversarial checks involving changed raw counts, changed installations and a vacuous thermal-model substitute. Its complete reference temperature cases yield the same [4/5, 6/5] kelvin-difference enclosure from Celsius and kelvin representations, while an expired-calibration case with missing coverage yields insufficient evidence. These are reference-vector outcomes, not

measured hive or greenhouse temperature changes. They demonstrate a useful interface pattern: the result depends on the record, its interpretation and the requested purpose [1].

The same manuscript records that local software replay passed, while final authoritative-host acceptance was not established: 23 of 66 scheduled Agda checks were accepted before workspace state prevented further checking. That operational limitation does not invalidate the analytical statements, but it bounds what that validation record establishes. Empirical instrument characterization also remained incomplete. The current extension toward direct code and proof binding in sensors must consequently identify its target hardware, executable semantics, numerical limits, timing and acquisition trust boundary before claiming sensor-resident assurance.

This foundation changes the next step from designing a generic evidence vocabulary to adapting concrete contracts to a new measurement family. Raw camera data can be bound to sensor and installation identities; calibration and transform results can carry specified realizations; aggregates can retain contributors and exclusions; and a downstream task can state the evidence it requires. Each adaptation still needs a proof for its own model. Thermal and mass contracts do not become camera geometry theorems by analogy.

8 What can be proved about learned components

8.1 Distinct guarantee classes

Table 5 distinguishes the guarantee classes considered here.

Table 5. Separate claims for a learned component

Claim	What establishes it	What does not follow
Graph well-formedness	Supported shapes, datatypes and operator versions	Accurate predictions
Forward-map implementation	Refinement proof or sound result checker for declared semantics	Robustness or generalization
Differentiation	Derivative theorem with domain and nonsmooth conventions	Training convergence
Input-output property	Proof for a specified input set and output predicate	Validity outside that set
Scientific task validity	Representative empirical evaluation and a justified measurement model	Universal correctness on all scenes

These distinctions permit strong claims where evidence exists without making an unlimited claim about AI. A verified implementation can compute precisely the specified wrong prediction. A proved gradient can optimize a poorly chosen loss. A local robustness theorem can hold around inputs that do not represent the deployment environment. Formal reasoning and empirical evaluation answer connected but different questions [18, 31].

8.2 Reported foundations in the separate program

The engineering record describes neural layers as parameterized maps $P \times X \to Y$, with mechanized constructions for composition, shared parameters, recurrence and differentiation. It reports examples across seven architecture families, including sheaf graph networks, equivariant networks, state-space models, mixtures of experts, neural differential equations, diffusion and meta-learning [2]. These are reported developments on established categorical foundations [16, 17]; they do not establish coverage of every concrete architecture within those families.

The record further reports forward and backward computations represented as reductions of addition-and-multiplication interaction nets. Selected exact-arithmetic examples reportedly produced byte-identical normal forms on CPU, CUDA and GB10 execution paths, with results related to Lean statements. It also describes an exact-rational Kolmogorov–Arnold network example and selected results for tied parameters [2]. These are useful motivations for integration, but the present report does not include the theorem declarations, graph instances, encodings, execution logs or backend implementations needed to independently assess their full scope.

For integer-valued examples, a derivative statement must specify whether it means a formal polynomial derivative evaluated at integers or the derivative of a real polynomial at integer inputs. Ordinary differentiability is not a property of a discrete integer-domain function without further structure. Similarly, an exact gradient-descent result for one constructed example does not establish general convergence. Backend agreement on examples is evidence about those executions, not a theorem of universal GPU correctness.

8.3 A bounded ONNX integration path

The reported Agda work provides semantics for selected ONNX operators, including convolution and global average pooling, and tooling to locate affine model fragments for replacement and comparison [2]. This is an initial compatibility surface. ONNX identifies models and operators using distinct IR versions, domains and operator-set versions [19, 20]; a sound import claim must state the supported combination, along with tensor layout, datatypes, attributes, padding, broadcasting and exceptional-value behavior.

A practical first experiment is a small fixed vision graph whose supported operator subset is fully enumerated. Bind the model and weights to a digest, compare the extracted semantics with a reference runtime on specified cases, and establish a compositional property for the supported graph. Unsupported custom operators or numerical modes must cause refusal of the corresponding assurance claim. Replacing one operator with a proved fragment does not automatically certify the surrounding network; exact equivalence or an explicit accumulated error bound is needed.

Softmax, exponentials, division, comparisons, quantization, sampling, numerical differential-equation solvers and parallel reductions introduce obligations beyond a simple addition-and-multiplication fragment. The appropriate research question is which subset can be checked at

useful cost and under which semantics. Applying that result to learned feature matching, depth estimation, segmentation or splat optimization remains planned. Physical and statistical validity must still be evaluated on representative data after computational properties are established.

9 Acquisition integrity and extensions beyond imaging

The separate FPGA program supplies a complementary research direction. The engineering record describes a configuration-success observation that did not establish functional integrity, and a proof of concept binding processing of a pinned GW150914 scientific-data segment to a formal statement [2]. The latter concerns processing and replay of an external dataset. It does not certify the original observatory acquisition or show that camera sensor outputs are already captured by a verified FPGA front end.

A camera-facing experiment should first define the interface boundary: which samples, frame identifiers, timestamps and metadata are observed, and which failures the receiver can detect. Sequence continuity, buffer behavior, payload binding and clock relationships are distinct properties. A receiver cannot prove the content of an opaque sensor's internal processing merely by receiving well-formed packets. A claim that the stored data equal the transmitted data therefore requires a defined starting point and a threat model.

Moving checked computation closer to a sensor admits several designs. A device may execute code whose source properties were proved before deployment, while a host checks the resulting records. Alternatively, a small device-side checker may validate a bounded claim at acquisition time. Neither design requires a full proof assistant to run inside every sensor. Each requires an explicit connection between the deployed artifact, the executed computation and the evidence delivered to the consumer. The existing host-oriented MAlonzo/Haskell route does not by itself establish an implementation for an arbitrary microcontroller or FPGA.

The first embedded target should consequently be small: a calibrated integer or rational interval operation, a sequence-consistency check, or an identity-bound summary. Finite storage, overflow behavior, execution time and external I/O belong in its specification. A proof about mathematical integers must be refined to the target arithmetic, or the allowed input range must be bounded so the representation is adequate. Comparing that implementation with the exact host reference is a useful initial test; a universal implementation claim requires a suitable refinement argument or sound checking boundary.

The same reasoning extends to thermal, spectral, acoustic, vibration and other scientific measurements. Each modality changes the measurand, calibration, synchronization and model assumptions. A spectral or polarization measurement can add information about a scene under suitable physical models; an extra filter does not automatically yield material stress or chemical identity. The transferable contribution is the contract and evidence discipline, while each physical inference requires its own validation.

10 Evaluation program and experimental resources

10.1 A sequence of falsifiable milestones

The next evaluation should establish progressively stronger claims. Each milestone in Table 6 has an observable success condition, a relevant negative control and an explicit limit. None of the proposed measurements is presented as an already completed result.

Table 6. Proposed evaluation milestones

Milestone	Evidence to collect	Failure to expose
Reproduce the pose diagnosis	Pinned image pairs, intrinsics, conventions, residuals and before/after pose estimates	Index swaps; frame-convention errors; unmodeled optical-center translation
Validate metric geometry	Independent targets across ranges, held-out lengths, repeatability and uncertainty budget	Scale bias; range dependence; sensor-dependent residuals
Check a geometry core	Named source theorems, pinned builds and extracted reference outputs	Wrong units; invalid rotations; zero calibration denominator
Bind data to claims	Input/output manifests and changed-artifact rejection results	Stale calibration; substituted frames; replayed receipts
Integrate a small learned graph	Supported-operator manifest, arithmetic semantics and whole-graph evidence	Unsupported opsets; altered weights; rounding-domain mismatch
Characterize system cost	Checker time, throughput, memory and refusal frequency	Unsustainable overhead; silent bypass under load

Independent geometry should be evaluated against a reference whose uncertainty is appropriate to the claim. Report absolute errors and their distribution, scene coverage, registration failures and the declared operating domain. For repeated runs, specify the number of independent captures, environmental changes and aggregation rule. A single room is not evidence of generality across materials, illumination or spatial scales.

Rendering quality should be evaluated separately from metric geometry. Held-out images may support PSNR, SSIM or another explicitly defined comparison, with exposure handling and train/test separation recorded. Geometry requires reference distances or surfaces. Neither category should substitute for the other, and the available notes supply no values for these proposed metrics.

For the checking layer, measure acceptance of valid cases and rejection of declared mutations. Record false refusals as well as false acceptances. An injected failure must break the actual predicate being tested; a visually dramatic corruption that is irrelevant to the claim is a weak control. Checker runtime should be separated from producer runtime so the benefit and cost of reuse can be assessed.

10.2 Why additional equipment changes the experiment

A certified dimensional target and its measurement report would support calibration uncertainty and held-out length tests. A synchronized pair of accessible industrial cameras would permit explicit timing and coordinate-frame experiments. An independent depth modality would permit comparison of failure modes under a shared reference. These additions answer different questions; none should be justified solely by a larger sensor count.

Additional compute would support reproducible builds, repeated optimization and concurrent reference checks. A compatible workstation can remove local software-environment obstacles and shorten an experiment, but it does not reduce physical uncertainty by itself. Storage supports preservation of raw captures, calibration records, model versions and evidence artifacts, enabling later replay. FPGA and timing instrumentation can make sequence and timing properties observable at a defined digital boundary.

The resource-to-result relationship should be recorded before acquisition. For example, a calibration target should enable a bounded dimensional claim; a trigger-capable camera pair should enable a measured synchronization claim; and an accelerated execution platform should enable a cost-versus-guarantee comparison. These experiments produce reusable methods and engineering evidence that can benefit both instrumentation providers and independent research users.

11 Discussion and limitations

The principal limitation is the separation between the working capture system and the proposed assurance integration. The present camera pipeline is conventional software. The broader formal-computation results are described by the authors' engineering record, and their artifacts are not included with this report. No end-to-end camera theorem, physical accuracy certificate or universal ML guarantee is established here.

The reconstruction case is informative because it reveals a plausible failure that visual output alone did not diagnose. Its evidential strength remains preliminary: a single room, selected consistency statistics, incomplete cross-position pose recovery and no controlled final rendering ablation. Independent targets and explicit optical-center modeling are particularly important before interpreting the revised rig as metrically accurate.

For an AI ingestion system, the central benefit is a more precise relationship between a claim and its evidence. A downstream model could distinguish a dimension with a documented uncertainty model from a visually inferred dimension, or a checked operator evaluation from an unconstrained prediction. This could support scientific dataset curation, digital-twin inspection and human review. The extent of any reduction in validation effort remains a hypothesis to measure.

For demanding public-sector or defense research, the relevant early applications are instrument evaluation, infrastructure inspection, maintenance evidence and accountable data processing. The

same contracts may help organize test, evaluation, verification and validation evidence. They do not establish operational suitability, accreditation, system safety or authorization for a particular deployment. NIST's AI risk framework similarly treats validity, reliability and contextual evaluation as broader system concerns [31].

A vendor-neutral framework is useful only if its claims remain portable and inspectable. Equipment-specific adapters may change, while the meaning of units, coordinate frames, supported operators and refusal must stay explicit. Sharing a narrowly scoped reference implementation, its evidence and its known failure cases is therefore a stronger next result than announcing a broad guarantee that cannot yet be replayed.

12 Conclusion

The IAOM reconstruction program provides a working experimental basis for studying how physical observations can become assessable inputs to scientific and AI systems. Its reported camera-pose inconsistency illustrates why plausible output and successful execution are insufficient evidence of correct interpretation. The proposed response is to connect calibrated measurement assumptions, specified computation, artifact identity and claim-specific acceptance within a single documented boundary.

The immediate research objective is a small, replayable instance of that boundary: one calibrated geometric claim, one clearly specified processing path, one explicit compilation trust base and a set of meaningful rejection cases. Integration of supported learned operators and FPGA acquisition can then proceed with the same discipline. Additional instruments are scientifically valuable when they resolve a named uncertainty or observability limitation and permit that improvement to be demonstrated.

Data and code availability

The implementation-status descriptions and preliminary camera observations are documented in the IAOM engineering overview dated 2 November 2026 [2]. The Apiary companion manuscript identifies declaration names, project-relative evidence locations and recorded validation results for its separate software scope [1]. A public release of the raw camera capture, calibration records, complete code revisions, proof artifacts and execution receipts is not identified in this version. Independent reproduction of the reported results is therefore not established by the materials accompanying this report. The analytical propositions and their assumptions are stated in full. External documentation and software identified in the references are not substitutes for the IAOM experiment artifacts.

The Apiary manuscript describes a separate software ownership and licensing arrangement involving Equation Capital LLC/Apoth3osis [1]. No blanket open-source license, public redistribution permission or absence of competing interests is asserted here. A release of software or data requires the applicable rights-holder terms. This report does not assign a public repository, DOI or license to unreleased materials.

Appendix A Minimum evidence package for a stronger result

A reproducible camera package should contain an exact capture manifest; selected or complete releasable RGB and depth data; device and software versions; coordinate and timestamp conventions; intrinsics and sensor offsets; calibration-target evidence; original and revised rig models; pair-selection and matching settings; and the scripts that reproduce the diagnostic quantities. Privacy-preserving selection of a scene subset is possible, but the subset must still support the claimed result.

A formal package should contain named theorem statements, source revisions, dependencies, checking commands, transitive assumption reports and the executable extraction boundary. A runtime package should add actual input/output bindings, numerical encodings, model weights or their permitted references, backend versions and mutation outcomes. Each receipt should state whether it contains a proof, an attestation or test evidence. Publishing these materials would make the next version materially stronger without changing the core research question.

References

[1] R. Goodman and S. Goodman. From Sensor Readings to Scientific Models: Apiary and the Preservation of Physical Meaning through Constructive Measurement Contracts. The Institute For Applied Ontological Mathematics, revised research manuscript, 1 November 2026. Author-supplied companion manuscript; no public release identifier supplied.

[2] IAOM Engineering Team. Toward a Provably-Correct 3D Scientific Capture System: Technical Overview of the IAOM Matterport Reconstruction Program and its Certified-Computation Roadmap. Internal technical note, 2 November 2026. Unpublished engineering record supplied by the authors.

[3] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. 2nd ed., Cambridge University Press, 2004. doi: 10.1017/CBO9780511811685.

[4] J. L. Schönberger and J.-M. Frahm. Structure-from-Motion Revisited. CVPR, pp. 4104–4113, 2016. doi: 10.1109/CVPR.2016.445.

[5] B. Kerbl, G. Kopanas, T. Leimkühler and G. Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42(4), Article 139, 2023. doi: 10.1145/3592433.

[6] V. Ye et al. gsplat: An Open-Source Library for Gaussian Splatting. arXiv:2409.06765, 2024. https://arxiv.org/abs/2409.06765.

[7] D. M. Rosen, L. Carlone, A. S. Bandeira and J. J. Leonard. SE-Sync: A Certifiably Correct Algorithm for Synchronization over the Special Euclidean Group. *International Journal of Robotics Research* 38(2–3), pp. 95–125, 2019. doi: 10.1177/0278364918784361.

[8] H. Yang, J. Shi and L. Carlone. TEASER: Fast and Certifiable Point Cloud Registration. *IEEE Transactions on Robotics* 37(2), pp. 314–333, 2021. doi: 10.1109/TRO.2020.3033695.

[9] G. C. Necula. Proof-Carrying Code. POPL, pp. 106–119, 1997. doi: 10.1145/263699.263712.

[10] A. Chiesa and E. Tromer. Proof-Carrying Data and Hearsay Arguments from Signature Cards. Innovations in Computer Science, pp. 310–331, 2010. https://projects.csail.mit.edu/pcd/.

[11] Safety-Critical Systems Club Assurance Case Working Group. Goal Structuring Notation Community Standard. Version 3, SCSC-141C, 2021. https://scsc.uk/gsn-standard.

[12] Joint Committee for Guides in Metrology. Evaluation of Measurement Data: Guide to the Expression of Uncertainty in Measurement. JCGM 100:2008. doi: 10.59161/JCGM100-2008E.

[13] Joint Committee for Guides in Metrology. International Vocabulary of Metrology: Basic and General Concepts and Associated Terms. JCGM 200:2012. doi: 10.59161/JCGM200-2012.

[14] National Institute of Standards and Technology. Metrological Traceability: Frequently Asked Questions and NIST Policy. Accessed 2 November 2026. https://www.nist.gov/metrology/metrological-traceability.

[15] Joint Committee for Guides in Metrology. Guide to the Expression of Uncertainty in Measurement, Part 6: Developing and Using Measurement Models. JCGM GUM-6:2020. doi: 10.59161/JCGMGUM-6-2020.

[16] B. Fong, D. I. Spivak and R. Tuyéras. Backprop as Functor: A Compositional Perspective on Supervised Learning. *Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science*, pp. 1–13, 2019. https://arxiv.org/abs/1711.10455.

[17] B. Gavranović et al. Position: Categorical Deep Learning Is an Algebraic Theory of All Architectures. ICML, PMLR 235, pp. 15209–15241, 2024. https://proceedings.mlr.press/v235/gavranovic24a.html.

[18] G. Katz, C. Barrett, D. L. Dill, K. Julian and M. J. Kochenderfer. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. *Computer Aided Verification*, Lecture Notes in Computer Science 10426, pp. 97–117, 2017. doi: 10.1007/978-3-319-63387-9_5; https://arxiv.org/abs/1702.01135.

[19] ONNX Project. Open Neural Network Exchange Intermediate Representation Specification. Official documentation, accessed 2 November 2026. https://onnx.ai/onnx/repo-docs/IR.html.

[20] ONNX Project. ONNX Versioning. Official documentation, accessed 2 November 2026. https://onnx.ai/onnx/repo-docs/Versioning.html.

[21] European Machine Vision Association. GenICam Introduction and Current Standard Documents. Accessed 2 November 2026. https://www.emva.org/standards-technology/genicam/introduction-new/.

[22] European Machine Vision Association. EMVA Standard 1288, Release 4.0 General: Standard for Characterization of Image Sensors and Cameras. 2021. https://www.emva.org/standards-technology/emva-1288/emva-standard-1288-downloads-2/.

[23] L. Moreau and P. Missier, eds. PROV-DM: The PROV Data Model. W3C Recommendation, 30 April 2013. https://www.w3.org/TR/2013/REC-prov-dm-20130430/.

[24] S. Liang, T. Khalafbeigi and H. van der Schaaf, eds. OGC SensorThings API Part 1: Sensing Version 1.1. OGC 18-088, 4 August 2021. https://docs.ogc.org/is/18-088/18-088.html.

[25] Matterport. Compare Cameras; FAQ: Matterport Pro2 3D Camera. Manufacturer documentation, accessed 2 November 2026. https://matterport.com/cameras; https://support.matterport.com/s/article/FAQ-Matterport-Pro2-3D-Camera?language=en_US.

[26] S. Boldo and G. Melquiond. Flocq: A Unified Library for Proving Floating-point Algorithms in Coq. IEEE Symposium on Computer Arithmetic, pp. 243–252, 2011. doi: 10.1109/ARITH.2011.40.

[27] Agda Development Team. Safe Agda. Official documentation, accessed 2 November 2026. https://agda.readthedocs.io/en/stable/language/safe-agda.html.

[28] Agda Development Team. Compilers. Official documentation, accessed 2 November 2026. https://agda.readthedocs.io/en/stable/tools/compilers.html.

[29] X. Leroy. Formal Verification of a Realistic Compiler. *Communications of the ACM* 52(7), pp. 107–115, 2009. https://xavierleroy.org/publi/compcert-CACM.pdf.

[30] Lean Development Team. Validating a Lean Proof. Lean reference manual, accessed 2 November 2026. https://lean-lang.org/doc/reference/latest/ValidatingProofs/.

[31] National Institute of Standards and Technology. Artificial Intelligence Risk Management Framework 1.0. NIST AI 100-1, 2023. https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-1.pdf.