Technical report · IF-TR-2026-01 · version 1.0

Model validation of a sensor-rig planner: Vantage v0.8.1 checked against independent references

Sherin Joseph Roy 10 November 2026 Vantage v0.8.1 P3 · Rigs and calibration

Funded entirely by BluRabbit Mobility Private Limited.

Abstract

Vantage is a desktop application that plans the cameras, lidars, radars, GNSS antennas and IMUs of a data-collection vehicle and predicts in simulation what the rig will record before any hardware is fitted. Its figures rest on a validation suite that checks each model against a reference sharing none of its code: OpenCV, closed forms, exact ray casting, the KITTI devkit, ITU-R P.838-3, the NREL Solar Position Algorithm example, a published radar plate measurement and Monte Carlo solves. This report describes the suite by model family and sets the author's run at release v0.8.1 beside the report committed to the repository, generated by an earlier version on a different software stack. Every check passed in both runs. Four checks report different errors in the two runs, and the two run times are not comparable. The release notes state that the system has been validated in simulation and on synthetic captures, not on a fitted rig's captures.

1 Problem

Vantage is a desktop application for designing the sensor rigs of data-collection vehicles, the cars that record the camera data self-driving models are trained on [1]. Before such a vehicle is fitted, it is hard to know what its cameras and sensors will record: where the body, the rig and the other sensors block them, how many pixels land on objects, and how sensitive the layout is to mounting errors. Vantage models this in simulation and then checks the installed rig against the plan.

A planning tool of this kind is only as useful as the models behind its figures: a coverage map, a perception range or a calibration uncertainty that is systematically wrong produces a rig that is fitted, driven and then refitted. Since release v0.3.0 the repository has carried a model validation suite that checks each model against a reference that shares none of its code, with a technical reference documenting each model’s equations, assumptions and limits. This report examines the suite at release v0.8.1: what each check compares against, what the author’s run of it gave, and where that run differs from the report committed to the repository.

2 System design

Application

Vantage is a Python application with a PyQt6 window and an OpenGL viewport. Its engine (`vantage/core/`) holds the lens models, occlusion, coverage analysis, tolerance study, optimiser, sensor studies, calibration and exporters; around it sit the interface, a CARLA 0.9.16 bridge (`vantage/sim/`), an optional assistant, a Model Context Protocol server and the validation suite (`vantage/validation/`). Positions are given in the ISO 8855 vehicle frame: x forward, y left, z up, in metres, with the origin at the rear axle. Cameras, taken from the library’s camera modules, are modelled through rectilinear, fisheye and Kannala–Brandt lens models [3]. The suite runs on the built-in sample rig, which carries six fisheye cameras, one lidar and five radars.

Validation suite

The suite is a subpackage of nine check modules grouped by domain (optics, ranging, environment, vehicle, rig, calibration, formats, experiments, recording), a module of references, a runner and a report writer. Each check states its quantity, reference, error, tolerance and verdict. A run writes JSON, CSV and HTML reports with SVG figures and exits with code 6 when a check fails, so a continuous-integration job can gate on it. Options select models, a project other than the built-in sample, the seed of every random draw, and a quick smoke run with fewer Monte Carlo trials.

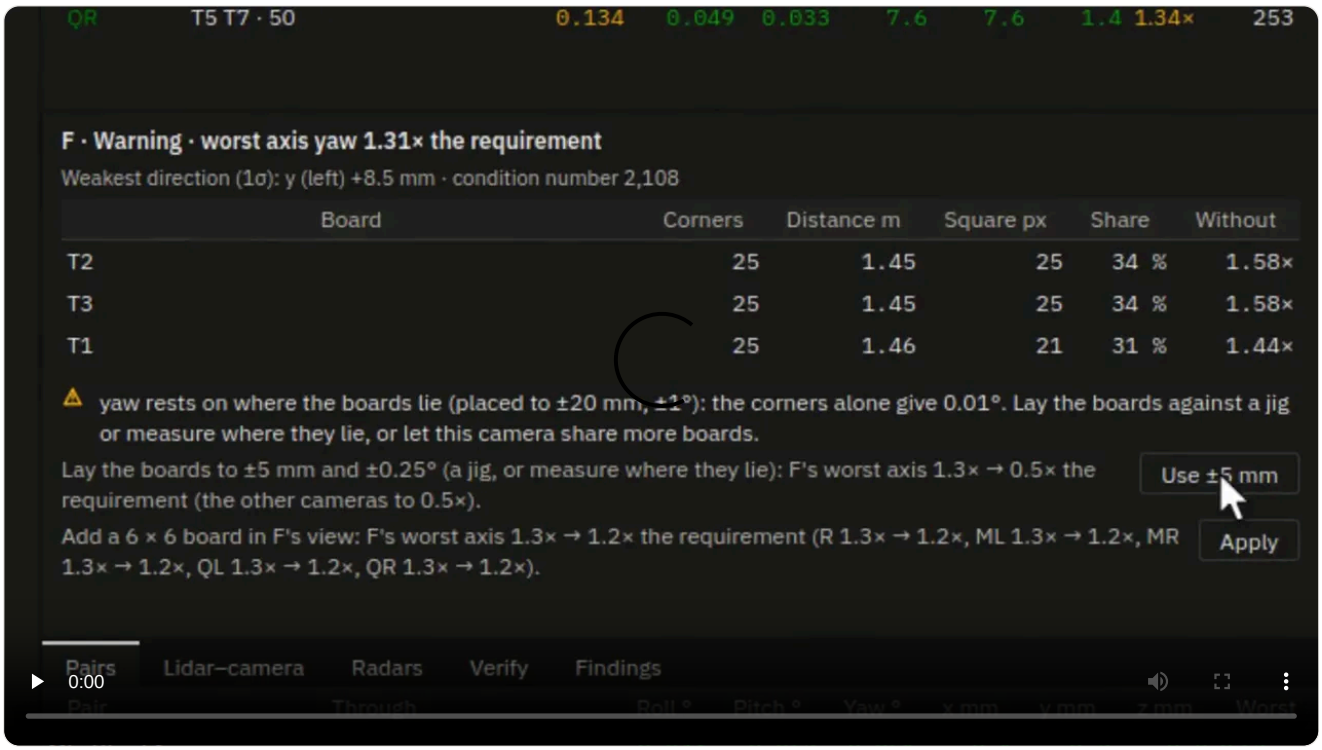


Figure 1. Calibration observability on a six-camera data-collection rig: for each camera, the 1σ pose uncertainty the planned boards give, against the requirement, and the effect of a suggested change to board placement. Screen recording from the repository (site/media/observability.mp4), trimmed and muted; the values on screen are the application's output for that project, not results of this report.

3 Method

The author ran the full suite (not the quick variant) of tag v0.8.1 on CPU with Qt's offscreen platform, installed as given under Reproducibility. The run used the built-in sample project, variant "6-cam · v3", and the suite's default seed 20261027. Per-check values were taken from the `validation.csv` the run wrote.

The comparison is the report committed in the same checkout under `site/validation/`. Its environment block states that it was generated by Vantage 0.7.0, on other interpreter and library versions (listed under Reproducibility) and unnamed hardware, with the same project, variant and seed. Where the two reports disagree, both values are given.

The validation run was made while the machine was shared with unrelated CPU workloads, including a training job; the load during the run was not recorded. The suite's run time is therefore not a performance figure. One lens check is withheld from this report for confidentiality.

4 Validation

In the author's run `71 checks` passed, with none failed and none skipped; the committed report records `71 checks`, also with none failed. The suite's checks fall into families as follows, counted from the CSV the author's run wrote.

Family	Checks	Reference
Lens	seven (one withheld here)	OpenCV's pinhole and fisheye projection and undistortion; identities; finite differences
Coverage	three	analytic footprints on a plate
Occlusion	three	exact ray–triangle intersection
Perception, critical targets	three, two	the pinhole relation; the lens's local magnification
Lidar	four	beam geometry; every beam of a revolution cast
Radar, glass	six, two	the radar equation; thin-film closed forms; a published plate measurement; the plate formula

Family	Checks	Reference
Weather	eight	ITU-R P.838-3, P.840, P.1814; Koschmieder's law; UN R112 test points
Sun	two	the NREL SPA worked example; Meeus's worked examples
Suspension, terrain	three, four	hand calculations; closed forms; brute-force line-of-sight march; the tangent to a crest
Timing, GNSS	three, two	sweep geometry; rolling-shutter relation; solid angle; Monte Carlo
Calibration, joint solve	one, two	the simulated unit's own lens; an independent Gauss–Newton solve; the injected poses
Observability, lidar–camera	four, two	Monte Carlo solves; Fisher information; an injected mounting error
Formats	four	the nuScenes and KITTI devkit conventions, ROS `camera_info`, Kalibr's model equations
Experiments, recording	four, two	analytic Ishigami indices; scipy's Wilson interval; faults written into a sequence

Cameras, coverage and occlusion

The fisheye lens model is compared with OpenCV's `cv2.fisheye.projectPoints` [2] for every built-in Kannala–Brandt lens, over directions out to a wide angle off the axis. The largest pixel distance was `1.85069 × 10⁻¹² px` in the author's run and `1.28622 × 10⁻¹² px` in the committed report; the two differ, and both pass. Occlusion cube maps of the body and rig are compared with exact ray casting against the same triangles [4]. The share of random directions on which the two disagree was `0.00195833 share of directions` in the author's run and `0.00195833 share of directions` in the committed report.

Radar and materials

The modelled 77 GHz one-way transmission of a painted, talc-filled polypropylene plate is compared with the range of two published measurements of that plate [5]. The distance from the measured range was `0.182332 dB` in the author's run and `0.182332 dB` in the committed report.

Weather and sun

The rain attenuation coefficients are compared with Table 5 of ITU-R P.838-3 as printed [6], in units of half the last printed digit. The largest difference was `0.999793 half last digits` in the author's run and `0.999793 half last digits` in the committed report. Read against its unit, this states agreement with the printed table to its rounding and no closer. The sun position is compared with the worked example of the NREL Solar Position Algorithm [7]: `0.00298214 degrees` in the author's run, `0.00298214 degrees` committed.

Terrain

The terrain check compares where the sample's front camera loses sight of the road beyond a crest with the analytic tangent from the lens to the crest's vertical curve. The difference was `0.111242 m` in the author's run and `0.111242 m` in the committed report.

Calibration and observability

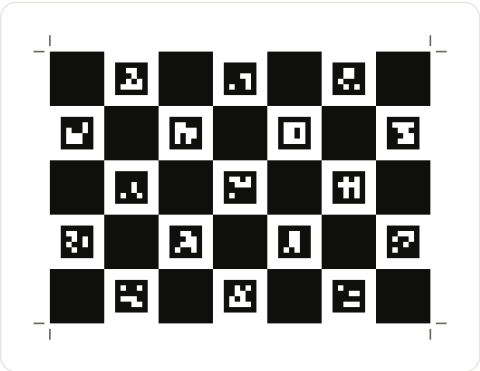


Figure 2. A ChArUco-style target, 7 × 5 squares: chessboard corners give sub-pixel points and the markers identify each corner, so one target ties every camera on a rig to the same frame. Schematic; the marker patterns are illustrative.

These checks run Vantage's solvers on synthetic captures and compare the result with what the captures were made from. The unit lens calibration fits a lens from sixteen synthetic board views of a unit set off the library focal length; its RMS pixel error over the field was `0.701022 px` in the author's run and `0.697518 px` in the committed report. From noise-free corners of cameras mounted off plan, the joint pose solve recovered the poses to `1.06089 × 10⁻¹² degrees or mm` in the author's run and `6.51608 × 10⁻¹⁰ degrees or mm` committed. The lidar–camera calibration recovered an injected lidar mounting error to `0.701492 mm` in the author's run and `0.83487 mm` committed. The observability study's predicted camera pose uncertainty was compared with the spread of the joint solve over two hundred simulated captures of the sample's six cameras: the largest absolute log ratio was `0.163033 log ratio` in the author's run and `0.163033 log ratio` committed.

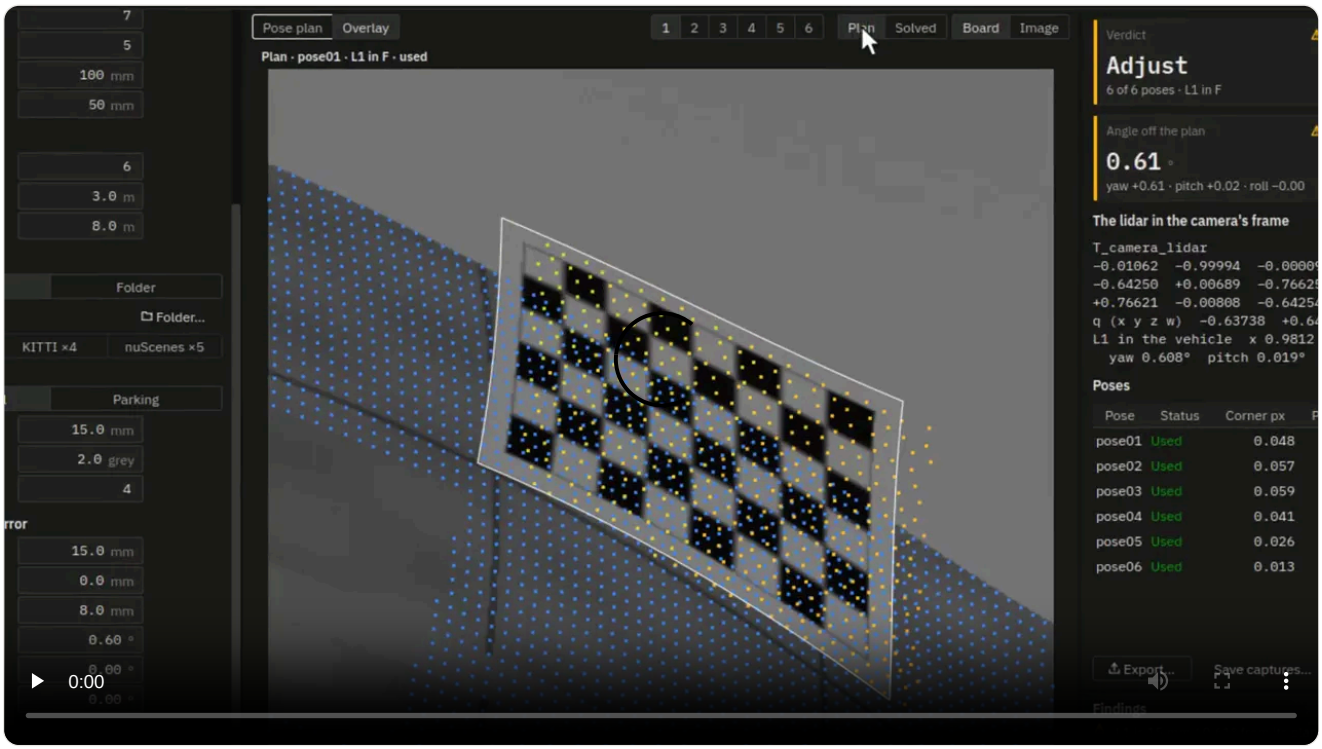


Figure 3. Lidar–camera calibration on a synthetic capture with an injected mounting error: the board seen by both sensors, the solved pose against the plan, and the verdict panel with the recovered extrinsics and their 1σ uncertainty. Screen recording from the repository (site/media/lidarcalib.mp4), trimmed and muted.

Hand-off formats, experiments and recording

For KITTI's `calib_cam_to_cam` and `calib_cams_to_vehicle` files, read as the KITTI raw devkit reads them [8], the largest pixel distance from Vantage's lens model was `0.000640193 px` in the author's run and `0.000640193 px` committed; a sibling check covers nuScenes `calibrated_sensor` rows [9]. The experiments module's Sobol estimator [10] was compared with the analytic first-order and total indices of the Ishigami function [11]: `0.00048921 index` in the author's run, `0.00048921 index` committed. Rig health's capture-time skew, against a trigger stagger written into the capture times of three cameras, differed by at most `0.199724 ms` in the author's run and `0.199724 ms` committed.

Quantity	Value	Conditions	Basis
Fisheye lens model vs OpenCV cv2.fisheye.projectPoints: largest pixel distance	1.85069×10^{-12} px	validation check lens.opencv_fisheye: every built-in Kannala–Brandt lens, directions up to 88° off the axis; tolerance 1e-06 px; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run Differs from another source; both are listed.	Reproduced 2026-11-10
Fisheye lens model vs OpenCV cv2.fisheye.projectPoints: largest pixel distance	1.28622×10^{-12} px	validation check lens.opencv_fisheye: every built-in Kannala–Brandt lens, directions up to 88° off the axis; tolerance 1e-06 px; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run Differs from another source; both are listed.	Reported 2026-10-31
Occlusion cube map vs exact ray casting: share of directions that disagree	0.00195833 share of directions	validation check occlusion.cube_vs_rays: random directions from each camera of the project, 256 × 256 texels per cube face, every disagreement within 1° of the exact silhouette; tolerance 0.005 share of directions; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reproduced 2026-11-10
Occlusion cube map vs exact ray casting: share of directions that disagree	0.00195833 share of directions	validation check occlusion.cube_vs_rays: random directions from each camera of the project, 256 × 256 texels per cube face, every disagreement within 1° of the exact silhouette; tolerance 0.005 share of directions; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reported 2026-10-31
77 GHz painted bumper plate transmission vs a published measurement: distance from the measured range	0.182332 dB	validation check radar.painted_plate: one-way |S21| of a 3.416 mm talc-filled PP plate with primer, metallic basecoat and 40 µm clear coat, substrate ε 2.9–3.1; tolerance 0.4 dB; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reproduced 2026-11-10
77 GHz painted bumper plate transmission vs a published measurement: distance from the measured range	0.182332 dB	validation check radar.painted_plate: one-way |S21| of a 3.416 mm talc-filled PP plate with primer, metallic basecoat and 40 µm clear coat, substrate ε 2.9–3.1; tolerance 0.4 dB; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reported 2026-10-31
Rain attenuation coefficients vs ITU-R P.838-3 Table 5: largest difference	0.999793 half last digits	validation check weather.p838_table: k_H, α_H, k_V, α_V at 16 frequencies from 1 to 1000 GHz, in units of half the last printed digit; tolerance 1 half last digits; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reproduced 2026-11-10
Rain attenuation coefficients vs ITU-R P.838-3 Table 5: largest difference	0.999793 half last digits	validation check weather.p838_table: k_H, α_H, k_V, α_V at 16 frequencies from 1 to 1000 GHz, in units of half the last printed digit; tolerance 1 half last digits; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reported 2026-10-31
Sun position vs the NREL Solar Position Algorithm worked example: largest angle difference	0.00298214 degrees	validation check sun.spa_example: topocentric zenith angle and azimuth at the paper's worked-example time and site; tolerance 0.01 degrees; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reproduced 2026-11-10
Sun position vs the NREL Solar Position Algorithm worked example: largest angle difference	0.00298214 degrees	validation check sun.spa_example: topocentric zenith angle and azimuth at the paper's worked-example time and site; tolerance 0.01 degrees; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reported 2026-10-31
Road hidden beyond a crest vs the analytic tangent: difference in sight-loss distance	0.111242 m	validation check terrain.crest_tangent: sample's front camera, crest radius 250 m falling to −10 %, 6 m ahead of the bumper; tolerance 0.3 m; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reproduced 2026-11-10

Quantity	Value	Conditions	Basis
Road hidden beyond a crest vs the analytic tangent: difference in sight-loss distance	0.111242 m	validation check terrain.crest_tangent: sample's front camera, crest radius 250 m falling to −10 %, 6 m ahead of the bumper; tolerance 0.3 m; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reported 2026-10-31
Unit lens calibration recovers a simulated lens: RMS pixel error	0.701022 px	validation check calib.unit_recovery: 16 synthetic board views, unit 1.5 % off the library focal length with an offset principal point, 2 grey levels of noise, half resolution, field out to 65° off the axis; tolerance 1 px; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run Differs from another source; both are listed.	Reproduced 2026-11-10
Unit lens calibration recovers a simulated lens: RMS pixel error	0.697518 px	validation check calib.unit_recovery: 16 synthetic board views, unit 1.5 % off the library focal length with an offset principal point, 2 grey levels of noise, half resolution, field out to 65° off the axis; tolerance 1 px; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run Differs from another source; both are listed.	Reported 2026-10-31
Joint pose solve recovers known camera poses from exact corners: largest error	1.06089×10^{-12} degrees or mm	validation check joint.exact_recovery: noise-free corners of the sample's boards, every camera mounted off plan by up to 1° and 10 mm; tolerance 0.001 degrees or mm; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run Differs from another source; both are listed.	Reproduced 2026-11-10
Joint pose solve recovers known camera poses from exact corners: largest error	6.51608×10^{-10} degrees or mm	validation check joint.exact_recovery: noise-free corners of the sample's boards, every camera mounted off plan by up to 1° and 10 mm; tolerance 0.001 degrees or mm; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run Differs from another source; both are listed.	Reported 2026-10-31
Lidar–camera calibration recovers an injected lidar mounting error: position error	0.701492 mm	validation check lidar_calib.recovery: synthetic captures of six planned board poses, lidar mounted 15, −10, 8 mm and 0.6°, −0.4°, 0.3° off plan, lidar range noise; tolerance 3 mm; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run Differs from another source; both are listed.	Reproduced 2026-11-10
Lidar–camera calibration recovers an injected lidar mounting error: position error	0.83487 mm	validation check lidar_calib.recovery: synthetic captures of six planned board poses, lidar mounted 15, −10, 8 mm and 0.6°, −0.4°, 0.3° off plan, lidar range noise; tolerance 3 mm; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run Differs from another source; both are listed.	Reported 2026-10-31
Predicted camera pose σ vs Monte Carlo spread of the joint solve: largest |log ratio|	0.163033 log ratio	validation check observability.cameras: 200 simulated captures, the sample's six cameras, boards exact, 0.4 px corner noise; tolerance is 4.5 standard errors; tolerance 0.225565 log ratio; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reproduced 2026-11-10
Predicted camera pose σ vs Monte Carlo spread of the joint solve: largest |log ratio|	0.163033 log ratio	validation check observability.cameras: 200 simulated captures, the sample's six cameras, boards exact, 0.4 px corner noise; tolerance is 4.5 standard errors; tolerance 0.225565 log ratio; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reported 2026-10-31
KITTI calibration export vs the KITTI devkit reading: largest pixel distance	0.000640193 px	validation check formats.kitti: calib_cam_to_cam and calib_cams_to_vehicle files; tolerance 0.05 px; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reproduced 2026-11-10

Quantity	Value	Conditions	Basis
KITTI calibration export vs the KITTI devkit reading: largest pixel distance	`0.000640193 px`	validation check formats.kitti: calib_cam_to_cam and calib_cams_to_vehicle files; tolerance 0.05 px; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reported 2026-10-31
Sobol index estimator vs analytic Ishigami indices: largest difference	`0.00048921 index`	validation check experiments.sobol_ishigami: first-order and total indices, 8192 base samples of a scrambled Sobol sequence, a = 7, b = 0.1; tolerance 0.02 index; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reproduced 2026-11-10
Sobol index estimator vs analytic Ishigami indices: largest difference	`0.00048921 index`	validation check experiments.sobol_ishigami: first-order and total indices, 8192 base samples of a scrambled Sobol sequence, a = 7, b = 0.1; tolerance 0.02 index; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reported 2026-10-31
Rig health's capture-time skew vs the skew written into the capture times: largest difference	`0.199724 ms`	validation check recording.capture_skew: three cameras of a lidar-synchronised rig fired 25 and 75 ms after the front camera, 0.4 ms jitter; tolerance 0.5 ms; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reproduced 2026-11-10
Rig health's capture-time skew vs the skew written into the capture times: largest difference	`0.199724 ms`	validation check recording.capture_skew: three cameras of a lidar-synchronised rig fired 25 and 75 ms after the front camera, 0.4 ms jitter; tolerance 0.5 ms; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reported 2026-10-31

Table 1. Selected validation checks, each with the author's run of v0.8.1 on CPU (reproduced by the author) directly above the report committed to the repository, generated by Vantage 0.7.0 (reported in the repository). Built-in sample project, variant 6-cam · v3, seed 20261027, full run; every check passed in both. “Reproduced” rows were run by the author; “Reported” rows are quoted from the repository at the commit shown.

Where the committed report and the author’s run differ

Four checks report different errors in the two runs: the fisheye projection against OpenCV, the unit lens calibration, the joint solve’s exact recovery and the lidar–camera calibration. Both runs used the same project, variant and seed, and every one of these checks passed in both. The likely sources are the different interpreter and numerical libraries of the two runs and the code changes between Vantage 0.7.0 and 0.8.1; the author did not isolate which is responsible. The two calibration recoveries, which carry image and range noise through an iterative solve, are the checks where the difference is visible at the scale of their units.



Figure 4. Unit lens calibration from sixteen synthetic board views: RMS pixel error against the simulated unit's lens. Same project and seed; committed report reported in the repository, lab run reproduced on CPU.



Figure 5. Lidar–camera calibration: position error of the recovered lidar mounting against the injected error. Same project and seed; committed report reported in the repository, lab run reproduced on CPU.

Suite run time

The author's run recorded `288.97 s`; the committed report records `64.49 s`. The two differ. They come from different hardware and software, and the author's run shared the machine with unrelated workloads at an unrecorded load, so neither is a performance claim and they are not comparable.

5 Limitations

No fitted rig. The v0.1.0 release notes give the reason for the 0.1 version number: everything had been validated in simulation and on synthetic captures, not yet on a fitted rig's real captures. The known limits of v0.8.1 repeat this, add that the first fitted rig will tell how close the planned poses, the installation check and the calibrations come in practice, and state that the validation suite does not replace measuring a fitted rig. The committed report's own limits say the suite does not show that a model describes a real vehicle, lens or sensor.

Synthetic inputs throughout. The calibration, joint-solve and lidar–camera checks run Vantage's solvers on captures rendered through Vantage's own lens model, so a lens that departs from its model is not exercised. The Monte Carlo checks are statistical, judged at a fixed seed against a tolerance set in standard errors of the estimated spread.

Nominal and typical inputs. The release notes state that the library's lidar, radar, GNSS and IMU figures are nominal datasheet values, and the material, weather and headlamp figures typical values, not measurements of a given vehicle or unit. The optimiser, tolerance study, perception range, installation check and critical targets still assume flat ground.

Conflicts and withheld items. Four check errors differ between the committed report and the author's run, as set out under Validation. The committed report was generated by Vantage 0.7.0 and so does not itself document release v0.8.1.

Claims not reproduced. The release notes state CARLA figures, needing a GPU, that have no committed measurement log and that the author did not run: a six-second drive of the built-in sample rig recorded in `28 s`, and every camera within `0.25 degrees` of its plan on that drive, an upper bound rather than a measured maximum. The notes also state an installation-check joint solve time of `0.15 s` after the v0.2.0 fix, on hardware they do not state.

6 Reproducibility

The report describes Vantage release v0.8.1, commit 16a6ee7 [1]. The author's validation run used:

```
git clone https://github.com/Sherin-SEF-AI/Vantage && cd Vantage && git checkout v0.8.1
uv venv --python 3.12 && uv pip install -e .[test]
QT_QPA_PLATFORM=offscreen .venv/bin/vantage --validate validation-out
```

The committed report is regenerated with `python -m vantage.validation site/validation --methods site/methods.html`, as documented in `vantage/validation/__main__.py`. The runtime was Python 3.12.13 on Linux 7.0.0-42 (Ubuntu 26.04.1), an AMD Ryzen 9 9900X (12 cores, 24 threads) with 30 GB of RAM, CPU only, with numpy 2.5.4, scipy 1.18.1, OpenCV 5.0.0 and Qt 6.12.0; the machine was shared with unrelated CPU workloads during the run. The committed report states Python 3.14.4, numpy 2.3.5, scipy 1.16.3, OpenCV 4.13.0 and Qt 6.11.0 on unstated hardware.

The repository's pytest suite was also started by the author on the same machine. Neither attempt completed: the machine was shared with unrelated jobs, and the second run had reached about one test in sixteen when it was stopped. No test totals are reported; the partial log is kept with the raw evidence.

The author did not run what needs hardware or external services: CARLA drives (a CARLA server and a GPU), live rig health (camera streams), log replay (PyTorch, practical only on a CUDA GPU) and the assistant's providers (an API key).

References

1. Sherin Joseph Roy. Vantage, release v0.8.1. Source code, GPL-3.0. github.com/Sherin-SEF-AI/Vantage/tree/v0.8.1
2. OpenCV. Camera Calibration and 3D Reconstruction (calib3d) and its fisheye module (cv::fisheye). Documentation. docs.opencv.org/
3. J. Kannala and S. S. Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. IEEE Trans. Pattern Analysis and Machine Intelligence 28(8), 1335–1340, 2006. doi:10.1109/TPAMI.2006.153
4. T. Möller and B. Trumbore. Fast, minimum storage ray–triangle intersection. Journal of Graphics Tools 2(1), 21–28, 1997. doi:10.1080/10867651.1997.10487468
5. E. Emilsson. Radar Transparency and Paint Compatibility: A Study of Automobile Bumper and Bumper-Skin Complex Permittivities for 77 GHz Microwaves. Master's thesis, Chalmers University of Technology, 2017. odr.chalmers.se/handle/20.500.12380/254360
6. Recommendation ITU-R P.838-3. Specific attenuation model for rain for use in prediction methods. ITU, 2005. www.itu.int/rec/R-REC-P.838-3-200503-I/en
7. I. Reda and A. Andreas. Solar position algorithm for solar radiation applications. Solar Energy 76(5), 577–589, 2004. doi:10.1016/j.solener.2003.12.003

8. A. Geiger, P. Lenz, C. Stiller and R. Urtasun. Vision meets robotics: The KITTI dataset. International Journal of Robotics Research 32(11), 1231–1237, 2013. doi:10.1177/0278364913491297

9. H. Caesar et al. nuScenes: A multimodal dataset for autonomous driving. CVPR 2020. doi:10.1109/CVPR42600.2020.01164

10. A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto and S. Tarantola. Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. Computer Physics Communications 181(2), 259–270, 2010. doi:10.1016/j.cpc.2009.09.018

11. T. Ishigami and T. Homma. An importance quantification technique in uncertainty analysis for computer models. Proc. ISUMA '90, 398–403, 1990. doi:10.1109/ISUMA.1990.151285

Evidence

Quantity	Value	Conditions	Basis	Source and reproduction
Model validation checks passed	71 checks	of 71 checks run, 0 failed, none skipped; 21 models; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reproduced 2026-11-10	command: vantage --validate commit 16a6ee7 (v0.8.1) · AMD Ryzen 9 9900X (12 cores / 24 threads), 30 GB RAM, Linux 7.0.0-42 (Ubuntu 26.04.1), Python 3.12.13, CPU only (Qt offscreen) `git clone https://github.com/Sherin-SEF-AI/Vantage && cd Vantage && git checkout v0.8.1 && uv venv --python 3.12 && uv pip install -e .[test] && QT_QPA_PLATFORM=offscreen .venv/bin/vantage --validate validation-out`
Model validation checks passed	71 checks	of 71 checks run, 0 failed, none skipped; 21 models; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reported 2026-10-31	site/validation/validation.json commit f8a7cfd · not stated (report environment: Linux x86_64, Python 3.14.4, numpy 2.3.5, scipy 1.16.3, OpenCV 4.13.0, Qt 6.11.0; generated by Vantage 0.7.0) `python -m vantage.validation site/validation --methods site/methods.html (as documented in vantage/validation/__main__.py)`
Fisheye lens model vs OpenCV cv2.fisheye.projectPoints: largest pixel distance	1.85069×10^{-12} px	validation check lens.opencv_fisheye: every built-in Kannala–Brandt lens, directions up to 88° off the axis; tolerance 1e-06 px; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run Differs from another source; both are listed.	Reproduced 2026-11-10	command: vantage --validate (value from the validation.csv the run wrote) commit 16a6ee7 (v0.8.1) · AMD Ryzen 9 9900X (12 cores / 24 threads), 30 GB RAM, Linux 7.0.0-42 (Ubuntu 26.04.1), Python 3.12.13, CPU only (Qt offscreen) `git clone https://github.com/Sherin-SEF-AI/Vantage && cd Vantage && git checkout v0.8.1 && uv venv --python 3.12 && uv pip install -e .[test] && QT_QPA_PLATFORM=offscreen .venv/bin/vantage --validate validation-out`
Fisheye lens model vs OpenCV cv2.fisheye.projectPoints: largest pixel distance	1.28622×10^{-12} px	validation check lens.opencv_fisheye: every built-in Kannala–Brandt lens, directions up to 88° off the axis; tolerance 1e-06 px; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run Differs from another source; both are listed.	Reported 2026-10-31	site/validation/validation.csv (summary and environment: site/validation/validation.json) commit f8a7cfd · not stated (report environment: Linux x86_64, Python 3.14.4, numpy 2.3.5, scipy 1.16.3, OpenCV 4.13.0, Qt 6.11.0; generated by Vantage 0.7.0) `python -m vantage.validation site/validation --methods site/methods.html (as documented in vantage/validation/__main__.py)`
Occlusion cube map vs exact ray casting: share of directions that disagree	0.00195833 share of directions	validation check occlusion.cube_vs_rays: random directions from each camera of the project, 256 × 256 texels per cube face, every disagreement within 1° of the exact silhouette; tolerance 0.005 share of directions; passed: yes; built-in sample project, variant "6-cam · v3", seed	Reproduced 2026-11-10	command: vantage --validate (value from the validation.csv the run wrote) commit 16a6ee7 (v0.8.1) · AMD Ryzen 9 9900X (12 cores / 24 threads), 30 GB RAM, Linux 7.0.0-42 (Ubuntu 26.04.1), Python 3.12.13, CPU only (Qt offscreen) `git clone https://github.com/Sherin-SEF-AI/Vantage && cd Vantage && git checkout v0.8.1 && uv venv --python 3.12 && uv pip install -e .[test] && QT_QPA_PLATFORM=offscreen`

Quantity	Value	Conditions	Basis	Source and reproduction
		20261027, full (not quick) run		.venv/bin/vantage --validate validation-out
Occlusion cube map vs exact ray casting: share of directions that disagree	0.00195833 share of directions	validation check occlusion.cube_vs_rays: random directions from each camera of the project, 256 × 256 texels per cube face, every disagreement within 1° of the exact silhouette; tolerance 0.005 share of directions; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reported 2026-10-31	site/validation/validation.csv (summary and environment: site/validation/validation.json) commit f8a7cfd · not stated (report environment: Linux x86_64, Python 3.14.4, numpy 2.3.5, scipy 1.16.3, OpenCV 4.13.0, Qt 6.11.0; generated by Vantage 0.7.0) python -m vantage.validation site/validation --methods site/methods.html (as documented in vantage/validation/__main__.py)
77 GHz painted bumper plate transmission vs a published measurement: distance from the measured range	0.182332 dB	validation check radar.painted_plate: one-way |S21| of a 3.416 mm talc-filled PP plate with primer, metallic basecoat and 40 µm clear coat, substrate ε 2.9–3.1; tolerance 0.4 dB; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reproduced 2026-11-10	command: vantage --validate (value from the validation.csv the run wrote) commit 16a6ee7 (v0.8.1) · AMD Ryzen 9 9900X (12 cores / 24 threads), 30 GB RAM, Linux 7.0.0-42 (Ubuntu 26.04.1), Python 3.12.13, CPU only (Qt offscreen) git clone https://github.com/Sherin-SEF-AI/Vantage && cd Vantage && git checkout v0.8.1 && uv venv --python 3.12 && uv pip install -e .[test] && QT_QPA_PLATFORM=offscreen .venv/bin/vantage --validate validation-out
77 GHz painted bumper plate transmission vs a published measurement: distance from the measured range	0.182332 dB	validation check radar.painted_plate: one-way |S21| of a 3.416 mm talc-filled PP plate with primer, metallic basecoat and 40 µm clear coat, substrate ε 2.9–3.1; tolerance 0.4 dB; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reported 2026-10-31	site/validation/validation.csv (summary and environment: site/validation/validation.json) commit f8a7cfd · not stated (report environment: Linux x86_64, Python 3.14.4, numpy 2.3.5, scipy 1.16.3, OpenCV 4.13.0, Qt 6.11.0; generated by Vantage 0.7.0) python -m vantage.validation site/validation --methods site/methods.html (as documented in vantage/validation/__main__.py)
Rain attenuation coefficients vs ITU-R P.838-3 Table 5: largest difference	0.999793 half last digits	validation check weather.p838_table: k_H, α_H, k_V, α_V at 16 frequencies from 1 to 1000 GHz, in units of half the last printed digit; tolerance 1 half last digits; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reproduced 2026-11-10	command: vantage --validate (value from the validation.csv the run wrote) commit 16a6ee7 (v0.8.1) · AMD Ryzen 9 9900X (12 cores / 24 threads), 30 GB RAM, Linux 7.0.0-42 (Ubuntu 26.04.1), Python 3.12.13, CPU only (Qt offscreen) git clone https://github.com/Sherin-SEF-AI/Vantage && cd Vantage && git checkout v0.8.1 && uv venv --python 3.12 && uv pip install -e .[test] && QT_QPA_PLATFORM=offscreen .venv/bin/vantage --validate validation-out
Rain attenuation coefficients vs ITU-R P.838-3 Table 5: largest difference	0.999793 half last digits	validation check weather.p838_table: k_H, α_H, k_V, α_V at 16 frequencies from 1 to 1000 GHz, in units of half the last printed digit; tolerance 1 half last digits; passed: yes;	Reported 2026-10-31	site/validation/validation.csv (summary and environment: site/validation/validation.json) commit f8a7cfd · not stated (report environment: Linux x86_64, Python 3.14.4, numpy 2.3.5, scipy 1.16.3,

Quantity	Value	Conditions	Basis	Source and reproduction
		built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run		OpenCV 4.13.0, Qt 6.11.0; generated by Vantage 0.7.0) `python -m vantage.validation site/validation --methods site/methods.html (as documented in vantage/validation/__main__.py)`
Sun position vs the NREL Solar Position Algorithm worked example: largest angle difference	`0.00298214 degrees`	validation check sun.spa_example: topocentric zenith angle and azimuth at the paper's worked-example time and site; tolerance 0.01 degrees; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reproduced 2026-11-10	command: vantage --validate (value from the validation.csv the run wrote) commit 16a6ee7 (v0.8.1) · AMD Ryzen 9 9900X (12 cores / 24 threads), 30 GB RAM, Linux 7.0.0-42 (Ubuntu 26.04.1), Python 3.12.13, CPU only (Qt offscreen) `git clone https://github.com/Sherin-SEF-AI/Vantage && cd Vantage && git checkout v0.8.1 && uv venv --python 3.12 && uv pip install -e .[test] && QT_QPA_PLATFORM=offscreen .venv/bin/vantage --validate validation-out`
Sun position vs the NREL Solar Position Algorithm worked example: largest angle difference	`0.00298214 degrees`	validation check sun.spa_example: topocentric zenith angle and azimuth at the paper's worked-example time and site; tolerance 0.01 degrees; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reported 2026-10-31	site/validation/validation.csv (summary and environment: site/validation/validation.json) commit f8a7cfd · not stated (report environment: Linux x86_64, Python 3.14.4, numpy 2.3.5, scipy 1.16.3, OpenCV 4.13.0, Qt 6.11.0; generated by Vantage 0.7.0) `python -m vantage.validation site/validation --methods site/methods.html (as documented in vantage/validation/__main__.py)`
Road hidden beyond a crest vs the analytic tangent: difference in sight-loss distance	`0.111242 m`	validation check terrain.crest_tangent: sample's front camera, crest radius 250 m falling to −10 %, 6 m ahead of the bumper; tolerance 0.3 m; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reproduced 2026-11-10	command: vantage --validate (value from the validation.csv the run wrote) commit 16a6ee7 (v0.8.1) · AMD Ryzen 9 9900X (12 cores / 24 threads), 30 GB RAM, Linux 7.0.0-42 (Ubuntu 26.04.1), Python 3.12.13, CPU only (Qt offscreen) `git clone https://github.com/Sherin-SEF-AI/Vantage && cd Vantage && git checkout v0.8.1 && uv venv --python 3.12 && uv pip install -e .[test] && QT_QPA_PLATFORM=offscreen .venv/bin/vantage --validate validation-out`
Road hidden beyond a crest vs the analytic tangent: difference in sight-loss distance	`0.111242 m`	validation check terrain.crest_tangent: sample's front camera, crest radius 250 m falling to −10 %, 6 m ahead of the bumper; tolerance 0.3 m; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reported 2026-10-31	site/validation/validation.csv (summary and environment: site/validation/validation.json) commit f8a7cfd · not stated (report environment: Linux x86_64, Python 3.14.4, numpy 2.3.5, scipy 1.16.3, OpenCV 4.13.0, Qt 6.11.0; generated by Vantage 0.7.0) `python -m vantage.validation site/validation --methods site/methods.html (as documented in vantage/validation/__main__.py)`
Unit lens calibration recovers a simulated lens: RMS pixel error	`0.701022 px`	validation check calib.unit_recovery: 16 synthetic board views, unit 1.5 % off the library focal length with an offset	Reproduced 2026-11-10	command: vantage --validate (value from the validation.csv the run wrote) commit 16a6ee7 (v0.8.1) · AMD Ryzen 9 9900X (12 cores / 24 threads), 30 GB

Quantity	Value	Conditions	Basis	Source and reproduction
		principal point, 2 grey levels of noise, half resolution, field out to 65° off the axis; tolerance 1 px; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run Differs from another source; both are listed.		RAM, Linux 7.0.0-42 (Ubuntu 26.04.1), Python 3.12.13, CPU only (Qt offscreen) `git clone https://github.com/Sherin-SEF-AI/Vantage && cd Vantage && git checkout v0.8.1 && uv venv --python 3.12 && uv pip install -e .[test] && QT_QPA_PLATFORM=offscreen .venv/bin/vantage --validate validation-out`
Unit lens calibration recovers a simulated lens: RMS pixel error	`0.697518 px`	validation check calib.unit_recovery: 16 synthetic board views, unit 1.5 % off the library focal length with an offset principal point, 2 grey levels of noise, half resolution, field out to 65° off the axis; tolerance 1 px; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run Differs from another source; both are listed.	Reported 2026-10-31	site/validation/validation.csv (summary and environment: site/validation/validation.json) commit f8a7cfd · not stated (report environment: Linux x86_64, Python 3.14.4, numpy 2.3.5, scipy 1.16.3, OpenCV 4.13.0, Qt 6.11.0; generated by Vantage 0.7.0) `python -m vantage.validation site/validation --methods site/methods.html (as documented in vantage/validation/__main__.py)`
Joint pose solve recovers known camera poses from exact corners: largest error	`1.06089 × 10⁻¹² degrees or mm`	validation check joint.exact_recovery: noise-free corners of the sample's boards, every camera mounted off plan by up to 1° and 10 mm; tolerance 0.001 degrees or mm; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run Differs from another source; both are listed.	Reproduced 2026-11-10	command: vantage --validate (value from the validation.csv the run wrote) commit 16a6ee7 (v0.8.1) · AMD Ryzen 9 9900X (12 cores / 24 threads), 30 GB RAM, Linux 7.0.0-42 (Ubuntu 26.04.1), Python 3.12.13, CPU only (Qt offscreen) `git clone https://github.com/Sherin-SEF-AI/Vantage && cd Vantage && git checkout v0.8.1 && uv venv --python 3.12 && uv pip install -e .[test] && QT_QPA_PLATFORM=offscreen .venv/bin/vantage --validate validation-out`
Joint pose solve recovers known camera poses from exact corners: largest error	`6.51608 × 10⁻¹⁰ degrees or mm`	validation check joint.exact_recovery: noise-free corners of the sample's boards, every camera mounted off plan by up to 1° and 10 mm; tolerance 0.001 degrees or mm; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run Differs from another source; both are listed.	Reported 2026-10-31	site/validation/validation.csv (summary and environment: site/validation/validation.json) commit f8a7cfd · not stated (report environment: Linux x86_64, Python 3.14.4, numpy 2.3.5, scipy 1.16.3, OpenCV 4.13.0, Qt 6.11.0; generated by Vantage 0.7.0) `python -m vantage.validation site/validation --methods site/methods.html (as documented in vantage/validation/__main__.py)`

Quantity	Value	Conditions	Basis	Source and reproduction
Predicted camera pose σ vs Monte Carlo spread of the joint solve: largest |log ratio|	0.163033 log ratio	validation check observability.cameras: 200 simulated captures, the sample's six cameras, boards exact, 0.4 px corner noise; tolerance is 4.5 standard errors; tolerance 0.225565 log ratio; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reproduced 2026-11-10	command: vantage --validate (value from the validation.csv the run wrote) commit 16a6ee7 (v0.8.1) · AMD Ryzen 9 9900X (12 cores / 24 threads), 30 GB RAM, Linux 7.0.0-42 (Ubuntu 26.04.1), Python 3.12.13, CPU only (Qt offscreen) `git clone https://github.com/Sherin-SEF-AI/Vantage && cd Vantage && git checkout v0.8.1 && uv venv --python 3.12 && uv pip install -e .[test] && QT_QPA_PLATFORM=offscreen .venv/bin/vantage --validate validation-out`
Predicted camera pose σ vs Monte Carlo spread of the joint solve: largest |log ratio|	0.163033 log ratio	validation check observability.cameras: 200 simulated captures, the sample's six cameras, boards exact, 0.4 px corner noise; tolerance is 4.5 standard errors; tolerance 0.225565 log ratio; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reported 2026-10-31	site/validation/validation.csv (summary and environment: site/validation/validation.json) commit f8a7cfd · not stated (report environment: Linux x86_64, Python 3.14.4, numpy 2.3.5, scipy 1.16.3, OpenCV 4.13.0, Qt 6.11.0; generated by Vantage 0.7.0) `python -m vantage.validation site/validation --methods site/methods.html (as documented in vantage/validation/__main__.py)`
Lidar–camera calibration recovers an injected lidar mounting error: position error	0.701492 mm	validation check lidar_calib.recovery: synthetic captures of six planned board poses, lidar mounted 15, −10, 8 mm and 0.6°, −0.4°, 0.3° off plan, lidar range noise; tolerance 3 mm; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run Differs from another source; both are listed.	Reproduced 2026-11-10	command: vantage --validate (value from the validation.csv the run wrote) commit 16a6ee7 (v0.8.1) · AMD Ryzen 9 9900X (12 cores / 24 threads), 30 GB RAM, Linux 7.0.0-42 (Ubuntu 26.04.1), Python 3.12.13, CPU only (Qt offscreen) `git clone https://github.com/Sherin-SEF-AI/Vantage && cd Vantage && git checkout v0.8.1 && uv venv --python 3.12 && uv pip install -e .[test] && QT_QPA_PLATFORM=offscreen .venv/bin/vantage --validate validation-out`
Lidar–camera calibration recovers an injected lidar mounting error: position error	0.83487 mm	validation check lidar_calib.recovery: synthetic captures of six planned board poses, lidar mounted 15, −10, 8 mm and 0.6°, −0.4°, 0.3° off plan, lidar range noise; tolerance 3 mm; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run Differs from another source; both are listed.	Reported 2026-10-31	site/validation/validation.csv (summary and environment: site/validation/validation.json) commit f8a7cfd · not stated (report environment: Linux x86_64, Python 3.14.4, numpy 2.3.5, scipy 1.16.3, OpenCV 4.13.0, Qt 6.11.0; generated by Vantage 0.7.0) `python -m vantage.validation site/validation --methods site/methods.html (as documented in vantage/validation/__main__.py)`
KITTI calibration export vs the KITTI devkit reading: largest pixel distance	0.000640193 px	validation check formats.kitti: calib_cam_to_cam and calib_cams_to_vehicle files; tolerance 0.05 px; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reproduced 2026-11-10	command: vantage --validate (value from the validation.csv the run wrote) commit 16a6ee7 (v0.8.1) · AMD Ryzen 9 9900X (12 cores / 24 threads), 30 GB RAM, Linux 7.0.0-42 (Ubuntu 26.04.1), Python 3.12.13, CPU only (Qt offscreen) `git clone https://github.com/Sherin-SEF-AI/Vantage && cd Vantage && git checkout v0.8.1 && uv venv --python`

Quantity	Value	Conditions	Basis	Source and reproduction
				`3.12 && uv pip install -e .[test] && QT_QPA_PLATFORM=offscreen .venv/bin/vantage --validate validation-out`
KITTI calibration export vs the KITTI devkit reading: largest pixel distance	`0.000640193 px`	validation check formats.kitti: calib_cam_to_cam and calib_cams_to_vehicle files; tolerance 0.05 px; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reported 2026-10-31	site/validation/validation.csv (summary and environment: site/validation/validation.json) commit f8a7cfd · not stated (report environment: Linux x86_64, Python 3.14.4, numpy 2.3.5, scipy 1.16.3, OpenCV 4.13.0, Qt 6.11.0; generated by Vantage 0.7.0) `python -m vantage.validation site/validation --methods site/methods.html (as documented in vantage/validation/__main__.py)`
Sobol index estimator vs analytic Ishigami indices: largest difference	`0.00048921 index`	validation check experiments.sobol_ishigami: first-order and total indices, 8192 base samples of a scrambled Sobol sequence, a = 7, b = 0.1; tolerance 0.02 index; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reproduced 2026-11-10	command: vantage --validate (value from the validation.csv the run wrote) commit 16a6ee7 (v0.8.1) · AMD Ryzen 9 9900X (12 cores / 24 threads), 30 GB RAM, Linux 7.0.0-42 (Ubuntu 26.04.1), Python 3.12.13, CPU only (Qt offscreen) `git clone https://github.com/Sherin-SEF-AI/Vantage && cd Vantage && git checkout v0.8.1 && uv venv --python 3.12 && uv pip install -e .[test] && QT_QPA_PLATFORM=offscreen .venv/bin/vantage --validate validation-out`
Sobol index estimator vs analytic Ishigami indices: largest difference	`0.00048921 index`	validation check experiments.sobol_ishigami: first-order and total indices, 8192 base samples of a scrambled Sobol sequence, a = 7, b = 0.1; tolerance 0.02 index; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reported 2026-10-31	site/validation/validation.csv (summary and environment: site/validation/validation.json) commit f8a7cfd · not stated (report environment: Linux x86_64, Python 3.14.4, numpy 2.3.5, scipy 1.16.3, OpenCV 4.13.0, Qt 6.11.0; generated by Vantage 0.7.0) `python -m vantage.validation site/validation --methods site/methods.html (as documented in vantage/validation/__main__.py)`
Rig health's capture-time skew vs the skew written into the capture times: largest difference	`0.199724 ms`	validation check recording.capture_skew: three cameras of a lidar-synchronised rig fired 25 and 75 ms after the front camera, 0.4 ms jitter; tolerance 0.5 ms; passed: yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run	Reproduced 2026-11-10	command: vantage --validate (value from the validation.csv the run wrote) commit 16a6ee7 (v0.8.1) · AMD Ryzen 9 9900X (12 cores / 24 threads), 30 GB RAM, Linux 7.0.0-42 (Ubuntu 26.04.1), Python 3.12.13, CPU only (Qt offscreen) `git clone https://github.com/Sherin-SEF-AI/Vantage && cd Vantage && git checkout v0.8.1 && uv venv --python 3.12 && uv pip install -e .[test] && QT_QPA_PLATFORM=offscreen .venv/bin/vantage --validate validation-out`
Rig health's capture-time skew vs the skew written into the capture times: largest difference	`0.199724 ms`	validation check recording.capture_skew: three cameras of a lidar-synchronised rig fired 25 and 75 ms after the front camera, 0.4 ms jitter; tolerance 0.5 ms; passed:	Reported 2026-10-31	site/validation/validation.csv (summary and environment: site/validation/validation.json) commit f8a7cfd · not stated (report environment: Linux x86_64, Python 3.14.4, numpy 2.3.5, scipy 1.16.3,

Quantity	Value	Conditions	Basis	Source and reproduction
		yes; built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run		OpenCV 4.13.0, Qt 6.11.0; generated by Vantage 0.7.0) `python -m vantage.validation site/validation --methods site/methods.html (as documented in vantage/validation/__main__.py)`
Model validation suite run time (as the suite reports it)	288.97 s	all 71 checks, built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run; process wall time including start-up and report writing was 290.424309976 s (date before/after) Differs from another source; both are listed.	Reproduced 2026-11-10	command: vantage --validate commit 16a6ee7 (v0.8.1) · AMD Ryzen 9 9900X (12 cores / 24 threads), 30 GB RAM, Linux 7.0.0-42 (Ubuntu 26.04.1), Python 3.12.13, CPU only (Qt offscreen) `git clone https://github.com/Sherin-SEF-AI/Vantage && cd Vantage && git checkout v0.8.1 && uv venv --python 3.12 && uv pip install -e .[test] && QT_QPA_PLATFORM=offscreen .venv/bin/vantage --validate validation-out`
Model validation suite run time (as the suite reports it)	64.49 s	all 71 checks, built-in sample project, variant "6-cam · v3", seed 20261027, full (not quick) run; Vantage 0.7.0, the hardware is not stated Differs from another source; both are listed.	Reported 2026-10-31	site/validation/validation.json commit f8a7cfd · not stated (report environment: Linux x86_64, Python 3.14.4, numpy 2.3.5, scipy 1.16.3, OpenCV 4.13.0, Qt 6.11.0; generated by Vantage 0.7.0) `python -m vantage.validation site/validation --methods site/methods.html (as documented in vantage/validation/__main__.py)`
Time to record a simulated drive of the sample rig in CARLA	28 s	6 s of drive with the sample's six fisheye cameras, one lidar and five radars in CARLA 0.9.16, server headless (0.3.0 release notes)	Reported 2026-11-03	CHANGELOG.md commit 16a6ee7 · NVIDIA RTX 5080 (GPU stated; rest of the machine not stated) `vantage project.vantage --carla-run drive/ (needs a running CARLA server; README)`
Drive-log check: largest camera orientation error against the plan on a CARLA drive	0.25 degrees	stated as 'every camera within 0.25° of its plan' for the sample rig's 6 s CARLA drive (0.3.0 release notes); an upper bound, not a measured maximum	Reported 2026-11-03	CHANGELOG.md commit 16a6ee7 · NVIDIA RTX 5080 (GPU stated; rest of the machine not stated) `vantage project.vantage --carla-run drive/ then the drive-log check (needs CARLA); not documented as one command`
Installation check's joint solve time	0.15 s	after the 0.2.0 fix; 3.4 s before it; the release notes also state it previously stopped short of its optimum in 37 of 40 simulated checks	Reported 2026-11-03	CHANGELOG.md commit 16a6ee7 · not stated `not documented`

Table 2. Every measured quantity in this report, with its source, commit, hardware and the command that reproduces it. “Reproduced” rows were run by the author; “Reported” rows are quoted from the repository at the commit shown.

Cite this report

APA

Roy, S. J. (2026). Model validation of a sensor-rig planner: Vantage v0.8.1 checked against independent references (Technical Report IF-TR-2026-01, Version 1.0). IsoFrames. https://isoframes.io/reports/if-tr-2026-01/

BibTeX

```
@techreport{isoframes_if_tr_2026_01,
  author      = {Roy, Sherin Joseph},
  title       = {Model validation of a sensor-rig planner: Vantage v0.8.1 checked against independent references},
  institution = {IsoFrames, BluRabbit Mobility},
  number      = {IF-TR-2026-01},
  type        = {Technical Report},
  year        = {2026},
  month       = nov,
  note        = {Version 1.0},
  url         = {https://isoframes.io/reports/if-tr-2026-01/}
}
```