

Article

Depth-Assisted Sparse Visual Odometry for UAV-Relevant Synthetic RGB-D Evaluation: A Controlled Geometric-Backend Ablation

Andrii Polukhin ^{1,*} , Sergii Stirenko ¹ , Mairo Leier ² , Gert Jervan ² , Oleksandr Rokovyi ¹ , Oleg Alienin ¹ , Nazrul Nazeer ²  and Yuri Gordienko ¹ 

¹ Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, 37 Beresteyskyi Ave., 03056 Kyiv, Ukraine; stirenko@comsys.kpi.ua (S.S.); rokovi@comsys.kpi.ua (O.R.); oleg.alenin@gmail.com (O.A.); gord@comsys.kpi.ua (Y.G.)

² Department of Computer Systems, Tallinn University of Technology, Akadeemia tee 15A, 12618 Tallinn, Estonia; mairo.leier@taltech.ee (M.L.); gert.jervan@taltech.ee (G.J.); nazrul.nazeer@taltech.ee (N.N.)

* Correspondence: polukhin@comsys.kpi.ua

Abstract

Sparse visual odometry (VO) is a core component of lightweight unmanned aerial vehicle (UAV) visual navigation, yet the isolated effect of adding aligned metric depth to a minimal frame-to-frame pipeline is easily obscured in full SLAM systems. This paper presents a UAV-relevant controlled synthetic ablation of RGB-only and RGB-D geometric backends under fixed sparse frontends. ORB matching and KLT tracking are evaluated on a 32-sequence TartanAir validation split of flight-like synthetic RGB-D scenes by routing identical 2D correspondences either to Essential Matrix estimation or, with aligned depth, to PnP with RANSAC. The study reports ATE, Sim(3)-aligned ATE, translational and rotational RPE, robustness under temporal subsampling and RGB degradations, and isolated solver latency on a workstation and Raspberry Pi 4. At stride 1, RGB-D PnP reduces ATE by 61.8% for KLT and 29.1% for ORB, with translational RPE reductions of 61.6% and 41.9%. Rotational RPE reductions are stronger and persist across all tested strides, reaching 85.9% for KLT and 77.3% for ORB at stride 1. Sim(3) analysis shows that only 7–16% of PnP ATE is metric-scale drift. At coarser strides, however, KLT-PnP no longer improves ATE, showing that depth assistance depends on stable frontend tracking and valid depth-supported correspondences. The contribution is a reproducible diagnostic benchmark and failure-mode analysis for UAV-relevant depth-assisted sparse VO under oracle aligned depth, providing component-level evidence rather than full onboard deployment validation.

Keywords: UAV visual navigation; sparse visual odometry; RGB-D sensing; synthetic RGB-D evaluation; geometric backend ablation; Essential Matrix; Perspective-n-Point; ORB/KLT frontends; TartanAir; edge solver latency



Academic Editor: Sunan Huang

Received: 20 May 2026

Revised: 21 June 2026

Accepted: 29 June 2026

Published: 30 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Visual odometry (VO) estimates camera motion from a sequence of images and serves as a basic component for autonomous visual navigation. In lightweight robotic platforms, including unmanned aerial vehicles (UAVs), a minimal VO module may be preferred when the goal is to characterize local ego-motion behavior without deploying a complete Simultaneous Localization and Mapping (SLAM) stack. This UAV-oriented motivation is treated here at the component level: the paper studies sparse visual ego-motion on

repeatable synthetic RGB-D sequences and isolated edge-hardware solver timings, not a complete onboard navigation system. The specific technical issue addressed in this paper is narrower than full robotic deployment: monocular sparse VO recovers translation only up to an unknown metric scale, whereas depth-capable sensing can supply metric three-dimensional structure for pose estimation [1–4]. The mathematical advantage of using depth via Perspective-n-Point (PnP) solvers over monocular Essential Matrix estimation is a well-established geometric fact, since calibrated two-view pose estimation is classically associated with the five-point relative pose problem [5], whereas the PnP-family formulation exploits 3D-to-2D constraints when metric 3D points are available [6]. Therefore, the purpose of this study is not to propose a new VO algorithm or to prove the known geometry, but to quantify how this known backend difference behaves inside minimal sparse VO pipelines when the frontend correspondences are held fixed.

The complete SLAM stacks involving dense mapping, keyframes, loop closure, local bundle adjustment, or global map reuse are intentionally excluded from the present investigation. Such systems, including ORB-SLAM2 [7], SVO [8], and DSO [9], include multiple compensating modules that make it difficult to isolate the contribution of the geometric pose estimation backend itself. The primary objective of this study is therefore limited to a UAV-relevant controlled synthetic ablation: route the same sparse image correspondences through an RGB-only Essential Matrix backend or an RGB-D PnP backend; then measure the resulting accuracy, scale behavior, robustness, failure modes, and isolated solver latency.

The comparative analysis is performed with a fixed sparse two-dimensional (2D) frontend, using either Oriented FAST and Rotated BRIEF (ORB) feature matching or Kanade–Lucas–Tomasi (KLT) optical-flow tracking. The first backend is based on Red–Green–Blue (RGB)-only two-view geometry using the Essential Matrix and RANSAC, thereby generating a scale-ambiguous relative pose. The second backend incorporates RGB-D geometry, where identical image correspondences are supplemented with corresponding depth maps to support three-dimensional (3D) back-projection and metric relative pose estimation via PnP and RANSAC. All experimental evaluations are conducted on the 32-sequence validation split of the simulated TartanAir multimodal dataset [10]. This split provides flight-like synthetic RGB-D scenes and motion regimes relevant to UAV visual navigation while retaining synchronized RGB, aligned dense depth, calibrated intrinsics, and noise-free ground truth. This controlled setting allows the backend effect to be observed without uncontrolled real-sensor confounders. Consequently, the results should be interpreted as a diagnostic upper-bound study under oracle aligned depth, not as evidence of real-flight robustness or complete onboard deployment readiness.

The present investigation is limited to a single-agent frame-to-frame VO pipeline. Multi-agent perception, cooperative map fusion, shared relocalization, obstacle avoidance, and closed-loop flight control fall outside the scope of the current paper. The practical relevance for UAV computer vision is instead that the study identifies when aligned depth improves a minimal sparse backend, when this benefit breaks down because the frontend correspondences or depth-supported inlier set become unreliable, and how large the isolated geometry-solver cost is on a Raspberry Pi-class edge board.

The main contributions of this work are as follows:

- A UAV-relevant controlled RGB-only versus RGB-D backend ablation under fixed sparse frontends, where the geometric pose estimation backend (Essential Matrix versus PnP with RANSAC) is the principal modified component between paired configurations;
- An evaluation across the 32-sequence TartanAir validation split of flight-like synthetic RGB-D scenes under temporal subsampling parameters (strides $k \in \{1, 2, 4, 8\}$) and

five deterministic RGB degradation presets, with an explicit Sim(3)/SE(3) alignment policy for separating monocular scale effects from trajectory-shape error;

- A failure-mode analysis showing that the RGB-D PnP benefit depends on frontend stability: the advantage is strong at small strides but KLT-PnP fails to improve ATE at coarser strides where 2D tracking is degraded;
- A statistical assessment integrating mean $\pm$ std, median with Interquartile Range (IQR), bootstrap 95% confidence intervals, finite-coverage counts, and paired Wilcoxon signed-rank tests with Holm–Bonferroni correction across the family of 16 baseline comparisons;
- An isolated geometry-solver runtime benchmark on a Raspberry Pi 4 Model B edge platform (Broadcom BCM2711 System on a Chip (SoC); quad-core ARM Cortex-A72 at nominal 1.5 GHz; 64-bit Raspberry Pi OS), alongside a workstation reference, explicitly excluding full-pipeline camera, feature extraction, depth handling, and thermal effects;
- Public release of the TartanAir 32-sequence validation split with exploratory data analysis [11], and the open-source evaluation codebase containing the sparse frontends, geometric backends, deterministic RGB degradations, and metric scripts [12].

The structure of the paper is outlined as follows. Section 2 reviews visual ego-motion, multimodal data fusion, the scale and depth trade-off, and sparse correspondence evaluation paradigms. Section 3 describes the experimental protocol and dataset. Section 4 reports the obtained metrics and comparative analysis. Section 5 discusses the limitations of the adopted synthetic setup. Finally, Section 6 summarizes the bounded findings and identifies future validation steps.

2. Related Work

2.1. Multimodal Visual Ego-Motion in Constrained Environments

Operation in Global Navigation Satellite System (GNSS)-denied environments for UAVs is best treated as a combined problem of perception, estimation, and control rather than as a single sensor replacement task [13,14]. GNSS denial or degradation typically arises near structures that block or reflect satellite signals, under foliage, indoors, or in environments with strong radio interference. Recent surveys therefore organize the field around multimodal sensing and fusion, including visual and inertial odometry, RGB, infrared, depth, radar or lidar odometry, ultra-wideband anchors, and cooperative ground-air estimation [15–18].

VO is a core building block for estimating camera motion from image sequences. Full SLAM architectures, such as ORB-SLAM2 and ORB-SLAM3, implement mapping, relocalization, loop closure, local optimization, and bundle adjustment across multiple sensor configurations [7,19]. Semi-direct or direct monocular systems, including Semi-Direct Visual Odometry (SVO) and Direct Sparse Odometry (DSO), provide efficient odometry without relying on a purely descriptor-based frontend [8,9]. More recent learning-based systems, such as DROID-SLAM, DPVO, and Deep Patch Visual SLAM, further improve accuracy and robustness by using learned update operators, differentiable bundle adjustment, or loop closure mechanisms [20–22]. These systems define important practical baselines, but their integrated design also changes multiple components at once. The present study focuses on a narrower question: how trajectory accuracy and failure modes change when aligned depth is incorporated into a fixed sparse correspondence frontend while only the geometric backend is modified.

2.2. The Scale and Depth Trade-Off in Multimodal Visual Odometry

Monocular VO can be considered attractive for small UAVs because of its low payload and power requirements. At the same time, translations are recovered only up to an

unknown scale factor, and degradation can be observed under motion blur, low texture, repetitive patterns, or rapid dynamics. The underlying geometry can be anchored and metric relative motion can be supported by depth measurements, whether sourced from RGB-D cameras, stereo disparity, or other fused sensors, when integrated into the estimation stack [3].

The joint use of synchronized color/depth streams, ground-truth trajectories, and standard trajectory metrics such as ATE and RPE was popularized by the de facto standard TUM RGB-D benchmark [4]. RGB-D SLAM remains an active topic because depth sensing changes both the pose estimation geometry and the system-level trade-offs among accuracy, runtime, map quality, robustness, and sensor requirements [23]. Promising results have also been reported with learned monocular depth and scale-aware multimodal fusion methods, which blur the boundary between classical geometric approaches and learning-based pipelines [24]. In contrast to these directions, their implementation introduces training-data assumptions and runtime computational costs. The basic architectural question is therefore as follows: what quantifiable benefit is provided by adding a depth channel when the sparse frontend is kept unchanged and only the pose estimation geometry is modified?

2.3. Sparse Frontends and Evaluation Paradigms

To expose the impact of the geometric backend, 2D correspondence methods, which can feed either an RGB-only or an RGB-D pipeline, are required for the experimental baselines. The efficient frame-to-frame baseline is provided by KLT tracking on the basis of the Lucas–Kanade formulation [25] and its pyramidal implementation [26]. A fast binary keypoint detector and descriptor, widely used in real-time SLAM, are provided by ORB [27].

Although robust full-scale SLAM architectures are widely available, empirical evaluations of isolated frontends and backends are still useful for dissecting specific failure modes. Much of the literature compares complete SLAM systems under varying benchmarks; fewer works ablate only the geometric backend while locking the visual correspondence layer [3]. Multiple surveys examine full VO or visual-inertial odometry (VIO) performance—including comparisons of monocular pipelines on flying robots [28], evaluations of multimodal odometry techniques in autonomous driving [29], and general monocular VO overviews [30]—but they typically target end-to-end integrated solutions. The present paper narrows this gap through a synthetic, controlled empirical backend evaluation aimed at clarifying how and under which conditions depth mitigates scale drift. The established correspondences are connected by sparse pipelines either to two-view geometry, using Essential Matrix estimation with RANSAC [5,31], or to 3D constraints, using PnP with RANSAC when depth data is available [6]. Thus, the deliberate fixation of the frontend makes the role of depth observable while avoiding conflation with detector or matcher modifications.

Repeatable and statistically reliable evaluation of these multimodal configurations requires a controlled infrastructure for comparative analysis. The TartanAir dataset provides the necessary RGB images, aligned depth maps, camera calibration parameters, and ground-truth poses generated for simulated environments [10]. Within this study, the dataset is used as a controlled instrumentation environment for geometric ablation, not as direct evidence of full autonomous deployment capability. Real-flight and real RGB-D operation introduce range-dependent depth noise, invalid or missing depth, imperfect calibration, rolling-shutter effects, motion blur, synchronization offsets, vibration, exposure changes, and depth-sensor field-of-view limitations; these effects motivate the bounded interpretation of the present oracle depth results and the future perturbation studies outlined later. The classical frontends, namely ORB [27] and KLT [25,32], establish a stable interface

for the potential integration of later learned components. Although learned detectors and matchers, such as SuperPoint [33] and SuperGlue [34], can improve correspondence quality, their practical deployment on edge hardware typically requires model compression and hardware-aware optimization [35]. By maintaining a boundary between feature extraction, matching or tracking, pose estimation, and final trajectory evaluation, this study provides a reproducible baseline for subsequent depth-assisted sparse VO comparisons.

3. Materials and Methods

This section defines the controlled experimental protocol used to compare monocular RGB-only and RGB-D geometric pose estimation backends. The frontend is fixed within each comparative evaluation; as a result, the measured difference is generated by changing the geometric backend and adding aligned depth to the identical sparse 2D correspondences.

3.1. Scope and Comparison Policy

Direct comparison with full SLAM systems such as ORB-SLAM2 [7], SVO [8], or DSO [9] is intentionally excluded from the scope of this study for the following reasons:

- First, these systems combine keyframe selection, local bundle adjustment, loop closure, and global map optimization, which together compensate for and obscure the effects of the per-frame geometric backend considered here as the variable of interest.
- Second, SVO and DSO are direct or semi-direct photometric methods and do not share a sparse 2D correspondence interface with the present ORB- and KLT-based frontends. As a result, any difference in their reported numbers would mix frontend design, photometric optimization, and the isolated change in the geometric backend.
- Third, the main problem investigated is not which open-source pipeline is most accurate, but how much trajectory accuracy is recovered by changing only the geometric backend from the scale-ambiguous Essential Matrix solver to the metric PnP solver on identical 2D correspondences.

The evaluation reports paired comparisons of the four $\{\text{ORB, KLT}\} \times \{\text{Essential, PnP}\}$ configurations on identical input frames, and the reported ATE magnitudes should be interpreted as frame-to-frame integration without keyframing, local bundle adjustment, or loop closure, rather than as a benchmark against full SLAM stacks.

3.2. Multimodal Dataset and Data Preprocessing

The TartanAir dataset [10] was used because it provides synchronized RGB images, aligned dense depth maps, camera poses, and camera calibration parameters in repeatable simulated flight-like scenes. This dataset serves as the controlled instrumentation environment for the geometric ablation and should not be considered direct evidence of real-world UAV deployment performance. Temporal subsampling and deterministic RGB degradations were then applied as described below, and the identical sparse frontend and geometric backends were re-evaluated under each condition.

3.2.1. Temporal Subsampling and Visual Degradation

The integer frame stride $k \geq 1$ was applied so that only every k -th dataset frame is presented to the pipeline. This emulates a lower effective imaging rate relative to the native TartanAir timeline and increases the motion baseline between consecutive views. The ground-truth translations for ATE and RPE were subsampled with the identical stride. Separately, five deterministic RGB-only corruption presets were synthesized prior to feature extraction. The exact parameters used in all reported experiments are listed in

Table 1, making the protocol reproducible from the manuscript alone; the corresponding implementation is released as open-source code [12].

Table 1. Exact parameters of the five deterministic RGB-only degradation presets used in the robustness evaluation. All parameters are applied to the 8-bit, 0–255 BGR image prior to feature extraction; depth maps are not perturbed.

Preset	Kind	Parameters
blur_mild	Gaussian blur	kernel 5×5 , σ from OpenCV default
blur_heavy	Gaussian blur	kernel 11×11 , σ from OpenCV default
gaussian_noise	Additive Gaussian noise	mean 0, $\sigma = 12$ on 0–255 scale, seeded RNG
jpeg_low	JPEG recompression	quality factor 25
low_light	Gamma darkening	$I_{\text{out}} = (I_{\text{in}}/255)^\gamma \times 255$ with $\gamma = 1.8$

The selected parameter values are deliberately moderate proxies for failure modes documented in airborne RGB sensing, rather than calibrated reproductions of any specific sensor. The two Gaussian blur kernels emulate small (5×5) and large (11×11) defocus or vibration-coupled motion blur on the 640×480 image. The additive Gaussian noise with $\sigma = 12$ on the 0–255 8-bit intensity range corresponds to read-noise levels typically observed on small consumer Complementary Metal-Oxide-Semiconductor (CMOS) sensors at moderate gain settings. The Joint Photographic Experts Group (JPEG) quality factor of 25 is positioned in the strongly lossy regime, approximating aggressive onboard recompression used for bandwidth-constrained wireless transmission to a ground station. The gamma value $\gamma = 1.8$ darkens the image by a factor of roughly 2 on average and emulates dusk-like or under-exposed indoor conditions while preserving full dynamic range. These perturbations are artificial and do not model the full set of real UAV camera failure modes, such as rolling-shutter distortion, auto-exposure transients, synchronization error, vibration-coupled rolling-shutter wobble, or missing depth. The depth maps were deliberately left uncorrupted to stress the visual correspondence layer while holding an oracle depth channel fixed. Consequently, the robustness results should be interpreted as synthetic backend ablation results rather than real-world RGB-D sensor robustness predictions.

3.2.2. Exploratory Data Analysis (EDA)

The dataset subset comprises 32 sequences across 18 environments (abandonedfactory, abandonedfactory_night, amusement, carwelding, endofworld, gascola, hospital, japaneseealley, neighborhood, ocean, office, office2, oldtown, seashidetown, seasonsforest, seasonsforest_winter, soulcity, westerndesert). A statistical breakdown by environment and difficulty level is presented in Table 2. The distribution of sequence lengths, shown in Figure 1, includes both short and long trajectories.

Table 2. TartanAir dataset subset summary.

	Value
sequences	32
environments	18
easy sequences	17
hard sequences	15
total frames	26,650
min sequence length	300
max sequence length	3084
mean sequence length	832.81
all sequences aligned	True

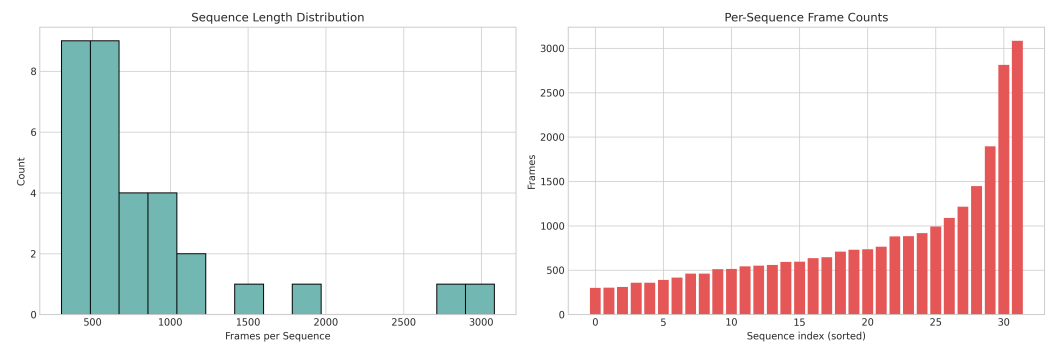


Figure 1. Distribution of the sequence lengths across the dataset.

The official TartanAir Visual SLAM ME/MH challenge split was considered, but ultimately not used for the RGB-D ablation, because aligned depth maps are not part of the monocular challenge archive and the testing split withholds the ground-truth poses to preserve leaderboard integrity. Since this study depends on per-sequence ATE and RPE values, the 32-sequence TartanAir validation split used by DROID-SLAM [20] and DPVO [21], which is derived from the public TartanAir training trajectories, was selected as the de facto standard community validation benchmark instead. For reproducibility, the exact dataset split used in the experiments is published on Kaggle [11]. This subset exposes the pipelines to the structural extremes typical of UAV navigation, including both feature-rich outdoor baselines and indoor environments with textureless walls, repetitive structures, and artificial lighting. The inclusion of both easy (smooth flight) and hard (aggressive rotational motion) sequences exposes specific failure regimes and supports comparison of the trade-offs between RGB-only and RGB-D backends under both regimes.

3.3. Sparse 2D Correspondence Extraction

Two sparse correspondence frontends were used. The KLT optical-flow tracking [25,26] detects up to 1000 Shi–Tomasi corners and tracks them with pyramidal Lucas–Kanade optical flow using a 21×21 window and three pyramid levels. The ORB [27] detects up to 2000 binary keypoints, matches descriptors with Hamming distance, and applies a Lowe ratio threshold of 0.7. For synthetic correspondence evaluation, an inlier ratio of 0.6 was used. These parameters were kept fixed when switching between the Essential Matrix and PnP backends.

3.4. Pose Estimation Backends

The two isolated geometric backends were evaluated using the previously extracted sparse correspondences.

RGB-Only Domain (Scale-Ambiguous)

For the monocular configuration, the 2D-to-2D correspondences established between consecutive frames are passed to Essential Matrix estimation based on the epipolar constraint $\mathbf{x}_2^T \mathbf{E} \mathbf{x}_1 = 0$, where $\mathbf{x}_1$ and $\mathbf{x}_2$ represent normalized image coordinates. In contrast to the PnP branch, no depth gating is applied to these correspondences. RANSAC is used for outlier rejection and recovery of the Essential Matrix $\mathbf{E}$; the relative rotation $\mathbf{R}$ and translation direction $\mathbf{t}$ are then decomposed via singular value decomposition (SVD). Consequently, this backend generates a scale-ambiguous monocular motion estimate, which requires an external scale prior for absolute trajectory reconstruction.

RGB-D Multimodal Domain (Metric)

In contrast to the monocular setup, the multimodal configuration pairs the identical 2D image correspondences directly with the corresponding per-pixel depth maps to recover metric-scale translations. Reference-frame points are back-projected into 3D Cartesian space using the standard transformations $X = (u - c_x)Z / f_x$, $Y = (v - c_y)Z / f_y$, and $Z = D(u, v)$. Only finite depth samples bounded within the interval $[0.1, 100]$ m are accepted by the PnP backend. The camera intrinsics are fixed at $f_x = f_y = 320$, $c_x = 320$, and $c_y = 240$ for all 640×480 images. The metric relative pose is estimated with OpenCV PnP-RANSAC, using the EPNP algorithm, 100 iterations, a 2-pixel reprojection threshold, a confidence level of 0.99, a minimum of 12 inliers, and a maximum accepted translation norm of 5 m.

3.5. Geometric Interpretation and Expected Failure Modes

The central distinction between the compared backends is the observability of metric translation scale. In the RGB-only branch, each normalized correspondence satisfies the epipolar constraint.

$$\mathbf{x}_{2,i}^\top \mathbf{E} \mathbf{x}_{1,i} = 0, \quad \mathbf{E} = [\mathbf{t}]_\times \mathbf{R},$$

where $\mathbf{R}$ is the relative rotation and $\mathbf{t}$ is the translation direction. Scaling the translation vector by any positive scalar leaves the epipolar constraint unchanged. The two-view Essential Matrix backend therefore recovers the translation direction but not its metric magnitude without an external scale prior. This explains why the monocular trajectory is evaluated after a Sim(3) alignment with scale, and why scale drift accumulates during frame-to-frame integration.

The RGB-D PnP branch changes the problem from 2D-to-2D epipolar geometry to 3D-to-2D reprojection geometry. For each accepted depth value, the reference-frame image coordinate is lifted to a metric 3D point,

$$\mathbf{X}_i = Z_i \mathbf{K}^{-1} \tilde{\mathbf{u}}_i,$$

and the relative pose is estimated by minimizing the reprojection residuals,

$$(\mathbf{R}, \mathbf{t}) = \arg \min_{\mathbf{R}, \mathbf{t}} \sum_i \|\pi(\mathbf{K}(\mathbf{R}\mathbf{X}_i + \mathbf{t})) - \mathbf{u}'_i\|_2^2.$$

Depth supplies metric scale before pose estimation. PnP is therefore expected to reduce translational drift when the 2D correspondences, camera intrinsics, depth values, and RGB-D alignment are mutually consistent.

This advantage holds under conditions. Perturbing the back-projected 3D point gives the first-order relation

$$\delta \mathbf{X}_i \approx \mathbf{K}^{-1} \tilde{\mathbf{u}}_i \delta Z_i + Z_i \delta(\mathbf{K}^{-1} \tilde{\mathbf{u}}_i),$$

which shows that depth noise, intrinsic-calibration error, RGB-D registration error, and feature localization error all enter directly into the 3D point consumed by PnP. In an actual flight environment, further effects such as rolling-shutter distortion, vibration, exposure changes, depth invalidation, and RGB-depth time offset can perturb the reprojection residuals. Under these conditions, the metric depth channel may no longer dominate the error budget. A PnP estimate based on corrupted depth-supported inliers can then become less stable than an Essential Matrix estimate that requires only consistent two-view bearing directions.

On this basis, the expected qualitative behavior is as follows. First, PnP should provide the largest benefit at small temporal strides and under visually stable conditions, because

the frontend supplies coherent correspondences and the oracle depth fixes metric scale. Second, the benefit should be stronger for translational trajectory metrics than for pure rotational error, since the main contribution of depth is scale observability. Third, KLT-based PnP should degrade at large temporal strides, when optical-flow tracking violates its small-displacement assumption and transfers poor correspondences into the metric solver. Finally, the isolated solver latency measurements should be read only as backend-call costs, because the end-to-end timing of a real UAV perception stack also depends on sensing, synchronization, rectification, feature extraction, depth handling, and thermal behavior.

3.6. Evaluation Metrics and Alignment Policy

The estimated trajectory $P_{1:n}$ is compared with the ground-truth trajectory $Q_{1:n}$ using the ATE and translational RPE metrics. Since the Essential Matrix backend produces scale-ambiguous translation directions, the ATE for the corresponding Essential trajectories is computed after global Sim(3) Umeyama alignment with scale. For the PnP trajectories, the same Umeyama alignment is applied, but the scale is fixed to one, functioning de facto as an SE(3) alignment. This policy avoids unfair penalization of monocular estimates caused by arbitrary global scale, while preserving the distinction that only the RGB-D PnP framework estimates metric translation during online pose integration.

The ATE is formally reported as the Root Mean Square Error (RMSE) of the aligned translation residuals. The RPE is computed after the same alignment procedure and is represented as the RMSE of the local translation-increment errors, while the rotational RPE captures angular drift over the equivalent increments:

$$\text{RPE}_\Delta = \left(\frac{1}{n - \Delta} \sum_{i=1}^{n-\Delta} \|(Q_{i+\Delta} - Q_i) - (\hat{P}_{i+\Delta} - \hat{P}_i)\|^2 \right)^{1/2},$$

where $\hat{P}$ denotes the aligned estimated trajectory and $\Delta = 1$ throughout all reported tables. Missing or non-finite sequence-level ATE/RPE values, which can arise from degenerate saved trajectories or non-finite alignment outcomes, are excluded from the finite summary statistics.

For each aggregated table entry, the finite sequence-level errors are summarized by providing the mean, standard deviation, median, IQR, and a bootstrap 95% confidence interval over the sequence means. The detailed per-sequence result table is retained as a reproducibility artifact.

The solver-level latency experiment complements the geometric distinction formalized in Section 3.5. The Essential Matrix branch relies on five-point sampling inside RANSAC [5,31], whereas the evaluated RGB-D branch uses the EPnP solver family inside PnP-RANSAC [6]. Although the smaller effective sample requirements characteristic of PnP-family solvers can reduce the overall RANSAC complexity, the empirical runtime is additionally dependent on the OpenCV implementation, the chosen solver, and the total correspondence count.

The algorithmic latency associated with the isolated geometric backends was benchmarked on synthetic correspondences using both a local workstation and a Raspberry Pi 4 Model B edge board. The workstation runs Linux (Windows Subsystem for Linux (WSL2), kernel 6.6) on an AMD Ryzen 9 7900X CPU (Advanced Micro Devices, Inc., Santa Clara, CA, USA; 12 cores, 24 hardware threads) with 64 GB system Random Access Memory (RAM); the reported solver times are CPU-only and bypass Graphics Processing Unit (GPU) acceleration. The edge device is a Raspberry Pi 4 Model B board (Raspberry Pi Holdings plc, Cambridge, UK) running Raspberry Pi OS (64-bit Advanced RISC Machine (ARM) Linux) on the Broadcom BCM2711 SoC (Broadcom Inc., Palo Alto, CA, USA); these measurements

similarly use the Cortex-A72 CPU cores and avoid off-loading the geometric solvers to the VideoCore GPU. The software stack was Python 3.10 with OpenCV 4.13.0.92, NumPy 2.2.6, SciPy 1.15.3, evo 1.36.3, and Matplotlib 3.10.9. The mean execution time and latency variance across varying quantities of correspondences ($N \in \{100, 500, 1000, 2000, 5000\}$) were recorded. These timings measure solely the geometry-solver call and exclude feature extraction, image acquisition, camera bus latency, memory pressure, and the impact of thermal throttling.

3.7. Use of Generative Artificial Intelligence Tools

GenAI tools were not used to generate the dataset, experimental measurements, numerical result tables, figures, or final scientific conclusions reported in this study. Where AI-assisted suggestions were used for manuscript organization, language editing, LaTeX/formatting assistance, or revision drafting, the corresponding text, scripts, numerical outputs, tables, figures, and scientific claims were reviewed and verified by the authors before inclusion.

4. Results and Discussion

Section 4 reports the ATE, ATE under Sim(3) alignment, translational and rotational RPE, robustness trends, qualitative trajectories, and isolated geometry-solver latency for the four sparse VO configurations. The quantitative results must be read within the scope and comparison policy of Section 3.1. They represent a controlled synthetic ablation of the geometric backend on fixed frontend correspondences. They are not a direct comparison with full SLAM systems or an end-to-end UAV deployment benchmark. The configurations integrate relative poses frame-to-frame without keyframes, local bundle adjustment, or loop closure, and therefore accumulate more drift than typical full SLAM pipelines.

4.1. Key Observations

Before presenting the detailed tables, the main empirical observations can be summarized as follows:

- RGB-D PnP gives the clearest translational benefit at small temporal strides. For KLT, ATE decreases by 61.8% at stride 1 and 55.5% at stride 2. For ORB, the reductions are 29.1% and 50.3%.
- The benefit depends on the frontend. KLT-PnP loses its ATE advantage at strides 4–8. ORB-PnP keeps a lower mean ATE at all evaluated strides, though the stride-8 ORB ATE gain is small and not statistically significant after correction.
- Sim(3) re-alignment of PnP trajectories shows that metric-scale drift explains only 7–16% of the absolute PnP ATE. Most remaining error is trajectory-shape error from correspondence quality, frame-to-frame integration, and solver robustness.
- Rotational RPE improves more uniformly than translational ATE under PnP for both frontends. This indicates that the coarse-stride KLT failure is mainly a translation and inlier-quality problem, not a complete rotational collapse.
- The Raspberry Pi 4 Model B timings support only a solver-level conclusion. PnP is faster than Essential Matrix estimation at 500 correspondences and above in this implementation. The measurement excludes acquisition, feature extraction, depth handling, and thermal effects.

4.2. Baseline Trajectory Accuracy

The baseline ATE results acquired without RGB degradation are summarized in Table 3, whereas the corresponding translational RPE values are organized in Table 4.

Table 3. Baseline ATE (m) across frame strides. Bold indicates the lower mean error within each frontend/stride pair.

Frontend Backend Stride	KLT		ORB	
	Essential	PnP	Essential	PnP
1	6.885 ± 4.614	2.633 ± 2.154	7.933 ± 4.680	5.627 ± 3.824
2	6.449 ± 4.093	2.867 ± 2.387	7.498 ± 4.812	3.727 ± 2.922
4	6.443 ± 3.843	6.762 ± 4.471	8.640 ± 4.974	4.732 ± 3.761
8	9.711 ± 5.412	11.274 ± 6.674	10.247 ± 5.610	9.306 ± 5.766

Table 4. Baseline RPE (m/frame) for local stability. Bold indicates the lower mean error within each frontend/stride pair.

Frontend Backend Stride	KLT		ORB	
	Essential	PnP	Essential	PnP
1	0.155 ± 0.054	0.061 ± 0.040	0.203 ± 0.100	0.118 ± 0.075
2	0.283 ± 0.101	0.158 ± 0.101	0.353 ± 0.169	0.178 ± 0.124
4	0.602 ± 0.247	0.582 ± 0.329	0.780 ± 0.364	0.424 ± 0.310
8	1.766 ± 0.804	1.563 ± 0.662	1.790 ± 0.968	1.374 ± 0.679

For KLT operating at stride 1, the RGB-D PnP backend decreases ATE from 6.885 ± 4.614 m to 2.633 ± 2.154 m, corresponding to a 61.8% reduction. The RPE also decreases from 0.155 ± 0.054 m/frame to 0.061 ± 0.040 m/frame, marking an approximate 61.6% reduction. At stride 2, the identical KLT-PnP configuration remains effective, reducing ATE by 55.5% and RPE by 44.1%.

Nevertheless, the observed KLT behavior changes at coarser temporal strides. At stride 4, KLT-PnP produces slightly higher ATE than KLT-Essential (6.762 m versus 6.443 m), while the RPE is only marginally lower (0.582 m/frame versus 0.602 m/frame). At stride 8, KLT-PnP also has worse ATE (11.274 m versus 9.711 m), although it maintains lower RPE (1.563 m/frame versus 1.766 m/frame). These results show a distinct depth-assisted advantage at small inter-frame baselines while contradicting the assumption of a uniform ATE advantage across all temporal subsampling conditions.

For ORB, PnP is more consistently favorable throughout the tested baseline conditions. At stride 1, ORB-PnP reduces ATE from 7.933 ± 4.680 m to 5.627 ± 3.824 m and RPE from 0.203 ± 0.100 m/frame to 0.118 ± 0.075 m/frame. At strides 2 and 4, the measured ATE reductions are 50.3% and 45.2%, respectively, with corresponding RPE reductions of 49.7% and 45.6%. At stride 8, the ATE reduction is smaller (9.2%), whereas the RPE remains lower by 23.2%.

The detailed baseline ATE and translational RPE statistics with median, IQR, 95% bootstrap confidence intervals, and per-cell coverage (finite sequence-level estimates out of 32) are reported in Tables 5 and 6, respectively.

To decouple trajectory-shape error from metric-scale drift, the PnP ATE re-aligned with Sim(3) is reported in Tables 7 and 8. For KLT-PnP, Sim(3) alignment reduces ATE from 2.633 ± 2.154 m to 2.221 ± 1.618 m at stride 1 and from 11.274 ± 6.674 m to 9.931 ± 5.691 m at stride 8; for ORB-PnP, the corresponding reductions are 5.627 ± 3.824 m to 4.821 ± 3.274 m at stride 1 and 9.306 ± 5.766 m to 8.148 ± 4.490 m at stride 8. Across all four strides, metric-scale drift accounts for 7–16% of the absolute PnP ATE, while the dominant remainder is trajectory-shape error that scale alignment cannot remove. Even after this Sim(3) correction, the PnP trajectory-shape error remains 67.7% lower than the (already Sim(3)-aligned by design) KLT-Essential baseline at stride 1 and 39.2% lower than ORB-Essential at the same

stride. These results show that the depth-assisted advantage is not solely a metric-scale phenomenon.

Table 5. Baseline ATE on the 32-sequence TartanAir validation split: absolute mean $\pm$ std (m), normalized mean (% of GT trajectory length), finite-aggregation coverage ($n/32$), median with IQR, and 95% bootstrap confidence interval over finite per-sequence values. **Bold:** within each (frontend, stride) pair, the backend with the lower absolute ATE mean.

Frontend	Backend	Stride	Mean $\pm$ std (m)	ATE% Mean (%)	Coverage	Median [IQR] (m)	95% CI (m)
KLT	Essential	1	6.885 $\pm$ 4.614	5.22 $\pm$ 2.62	32/32	5.010 [5.526]	[5.365, 8.494]
KLT	PnP	1	2.633 $\pm$ 2.154	2.14 $\pm$ 2.10	32/32	2.116 [2.617]	[1.945, 3.352]
KLT	Essential	2	6.449 $\pm$ 4.093	5.03 $\pm$ 2.30	32/32	5.009 [4.485]	[5.156, 7.782]
KLT	PnP	2	2.867 $\pm$ 2.387	2.26 $\pm$ 1.99	32/32	2.104 [2.831]	[2.122, 3.720]
KLT	Essential	4	6.443 $\pm$ 3.843	4.94 $\pm$ 2.02	32/32	5.529 [5.066]	[5.113, 7.817]
KLT	PnP	4	6.762 $\pm$ 4.471	5.07 $\pm$ 3.04	32/32	5.903 [5.209]	[5.176, 8.393]
KLT	Essential	8	9.711 $\pm$ 5.412	7.47 $\pm$ 2.91	32/32	10.015 [7.878]	[7.901, 11.813]
KLT	PnP	8	11.274 $\pm$ 6.674	8.77 $\pm$ 3.78	32/32	11.418 [6.687]	[8.865, 13.619]
ORB	Essential	1	7.933 $\pm$ 4.680	6.05 $\pm$ 2.57	32/32	6.748 [7.678]	[6.443, 9.681]
ORB	PnP	1	5.627 $\pm$ 3.824	4.33 $\pm$ 3.17	32/32	5.214 [7.184]	[4.418, 6.941]
ORB	Essential	2	7.498 $\pm$ 4.812	5.69 $\pm$ 2.60	32/32	6.208 [6.076]	[5.898, 9.128]
ORB	PnP	2	3.727 $\pm$ 2.922	3.04 $\pm$ 2.52	32/32	3.388 [3.723]	[2.728, 4.748]
ORB	Essential	4	8.640 $\pm$ 4.974	6.66 $\pm$ 2.76	32/32	8.162 [7.572]	[7.088, 10.417]
ORB	PnP	4	4.732 $\pm$ 3.761	3.90 $\pm$ 3.17	32/32	3.337 [4.335]	[3.508, 6.018]
ORB	Essential	8	10.247 $\pm$ 5.610	8.01 $\pm$ 3.28	32/32	9.952 [9.604]	[8.416, 12.123]
ORB	PnP	8	9.306 $\pm$ 5.766	7.35 $\pm$ 4.00	32/32	9.113 [4.780]	[7.394, 11.381]

Table 6. Baseline RPE (m/frame) on the 32-sequence TartanAir validation split: mean $\pm$ std, finite-aggregation coverage ($n/32$), median with IQR, and 95% bootstrap confidence interval. **Bold:** within each (frontend, stride) pair, the backend with the lower RPE mean.

Frontend	Backend	Stride	Mean $\pm$ std (m/frame)	Coverage	Median [IQR] (m/frame)	95% CI (m/frame)
KLT	Essential	1	0.155 $\pm$ 0.054	32/32	0.165 [0.077]	[0.136, 0.173]
KLT	PnP	1	0.061 $\pm$ 0.040	32/32	0.045 [0.045]	[0.047, 0.075]
KLT	Essential	2	0.283 $\pm$ 0.101	32/32	0.308 [0.167]	[0.250, 0.316]
KLT	PnP	2	0.158 $\pm$ 0.101	32/32	0.147 [0.101]	[0.125, 0.196]
KLT	Essential	4	0.602 $\pm$ 0.247	32/32	0.586 [0.307]	[0.521, 0.688]
KLT	PnP	4	0.582 $\pm$ 0.329	32/32	0.570 [0.377]	[0.472, 0.696]
KLT	Essential	8	1.766 $\pm$ 0.804	32/32	1.619 [1.004]	[1.488, 2.056]
KLT	PnP	8	1.563 $\pm$ 0.662	32/32	1.525 [0.633]	[1.336, 1.801]
ORB	Essential	1	0.203 $\pm$ 0.100	32/32	0.197 [0.067]	[0.168, 0.241]
ORB	PnP	1	0.118 $\pm$ 0.075	32/32	0.102 [0.087]	[0.094, 0.147]
ORB	Essential	2	0.353 $\pm$ 0.169	32/32	0.343 [0.155]	[0.300, 0.409]
ORB	PnP	2	0.178 $\pm$ 0.124	32/32	0.145 [0.123]	[0.138, 0.219]
ORB	Essential	4	0.780 $\pm$ 0.364	32/32	0.697 [0.340]	[0.670, 0.911]
ORB	PnP	4	0.424 $\pm$ 0.310	32/32	0.354 [0.342]	[0.325, 0.545]
ORB	Essential	8	1.790 $\pm$ 0.968	32/32	1.567 [0.955]	[1.473, 2.131]
ORB	PnP	8	1.374 $\pm$ 0.679	32/32	1.323 [0.922]	[1.144, 1.611]

Table 7. PnP ATE (m) under Sim(3) alignment across frame strides. Sim(3) absorbs metric scale, so the contrast with Table 3 isolates the trajectory-shape component of the PnP advantage from metric-scale drift. Bold indicates the lower mean error within each stride.

Frontend Stride	KLT	ORB
1	2.221 $\pm$ 1.618	4.821 $\pm$ 3.274
2	2.662 $\pm$ 2.315	3.378 $\pm$ 2.662
4	6.034 $\pm$ 3.931	4.230 $\pm$ 3.258
8	9.931 $\pm$ 5.691	8.148 $\pm$ 4.490

The rotational RPE, reported in Table 9 and detailed in Table 10, exhibits a stronger and more uniform PnP advantage than the translational metrics. At stride 1, the rotational RPE is reduced by the depth-assisted PnP from 15.119 $\pm$ 9.788 deg/frame to 2.134 $\pm$ 0.612 deg/frame for KLT (an 85.9% reduction) and from 11.801 $\pm$ 7.175 deg/frame

to 2.674 ± 1.865 deg/frame for ORB (a 77.3% reduction). The PnP advantage is preserved at every tested stride and for both frontends: the KLT-PnP rotational RPE is 61.4%, 26.7%, and 32.3% lower than KLT-Essential at strides 2, 4, and 8, while the ORB-PnP rotational RPE is 51.8%, 42.2%, and 43.8% lower than ORB-Essential at the same strides. In contrast to the KLT-PnP ATE breakdown at strides 4–8, the rotational advantage is not lost at coarser temporal subsampling. This is consistent with the interpretation that the failure mode of KLT-PnP at coarser strides is dominated by translational scale-drift accumulation, not by rotational instability.

Table 8. Baseline PnP ATE under Sim(3) alignment on the 32-sequence TartanAir validation split: mean $\pm$ std (m), finite-aggregation coverage ($n/32$), median with IQR, and 95% bootstrap confidence interval. Reported side-by-side with the SE(3) PnP ATE from Table 5 to isolate trajectory-shape error from metric-scale drift. **Bold:** within each stride, the frontend with the lower Sim(3) ATE mean (i.e., lower trajectory-shape error after metric-scale is absorbed).

Frontend	Backend	Stride	ATE SE(3) (m)	ATE Sim(3) (m)	Coverage	Sim(3) Median [IQR]	Sim(3) 95% CI
KLT	PnP	1	2.633 ± 2.154	2.221 ± 1.618	32/32	1.957 [2.316]	[1.714, 2.862]
ORB	PnP	1	5.627 ± 3.824	4.821 ± 3.274	32/32	4.187 [5.873]	[3.739, 5.893]
KLT	PnP	2	2.867 ± 2.387	2.662 ± 2.315	32/32	1.889 [2.519]	[1.950, 3.479]
ORB	PnP	2	3.727 ± 2.922	3.378 ± 2.662	32/32	2.720 [3.250]	[2.507, 4.348]
KLT	PnP	4	6.762 ± 4.471	6.034 ± 3.931	32/32	5.426 [5.347]	[4.712, 7.316]
ORB	PnP	4	4.732 ± 3.761	4.230 ± 3.258	32/32	3.301 [3.764]	[3.250, 5.411]
KLT	PnP	8	11.274 ± 6.674	9.931 ± 5.691	32/32	9.858 [7.194]	[8.038, 11.926]
ORB	PnP	8	9.306 ± 5.766	8.148 ± 4.490	32/32	8.315 [6.119]	[6.707, 9.687]

Table 9. Baseline rotational RPE (deg/frame) for angular stability. Bold indicates the lower mean error within each frontend/stride pair.

Frontend Backend Stride	KLT		ORB	
	Essential	PnP	Essential	PnP
1	15.119 ± 9.788	2.134 ± 0.612	11.801 ± 7.175	2.674 ± 1.865
2	10.759 ± 7.369	4.150 ± 0.959	9.521 ± 6.662	4.585 ± 1.909
4	10.609 ± 4.315	7.773 ± 1.760	13.896 ± 6.436	8.029 ± 2.210
8	19.110 ± 5.776	12.936 ± 2.548	26.657 ± 9.850	14.992 ± 3.972

Table 10. Baseline rotational RPE (deg/frame) on the 32-sequence TartanAir validation split: mean $\pm$ std, finite-aggregation coverage ($n/32$), median with IQR, and 95% bootstrap confidence interval. **Bold:** within each (frontend, stride) pair, the backend with the lower rotational RPE mean.

Frontend	Backend	Stride	Mean $\pm$ std (deg/frame)	Coverage	Median [IQR] (deg/frame)	95% CI (deg/frame)
KLT	Essential	1	15.119 ± 9.788	32/32	15.618 [15.753]	[11.628, 18.665]
KLT	PnP	1	2.134 ± 0.612	32/32	$2.212 [0.692]$	[1.923, 2.353]
KLT	Essential	2	10.759 ± 7.369	32/32	7.808 [11.545]	[8.337, 13.460]
KLT	PnP	2	4.150 ± 0.959	32/32	$4.405 [1.308]$	[3.812, 4.471]
KLT	Essential	4	10.609 ± 4.315	32/32	9.964 [4.504]	[9.263, 12.175]
KLT	PnP	4	7.773 ± 1.760	32/32	$8.354 [2.500]$	[7.178, 8.347]
KLT	Essential	8	19.110 ± 5.776	32/32	18.380 [6.605]	[17.236, 21.065]
KLT	PnP	8	12.936 ± 2.548	32/32	$13.597 [3.489]$	[12.086, 13.739]
ORB	Essential	1	11.801 ± 7.175	32/32	11.465 [7.719]	[9.402, 14.260]
ORB	PnP	1	2.674 ± 1.865	32/32	$2.256 [0.654]$	[2.149, 3.384]
ORB	Essential	2	9.521 ± 6.662	32/32	7.523 [8.101]	[7.332, 12.012]
ORB	PnP	2	4.585 ± 1.909	32/32	$4.435 [1.645]$	[3.967, 5.282]
ORB	Essential	4	13.896 ± 6.436	32/32	12.407 [8.988]	[11.920, 16.332]
ORB	PnP	4	8.029 ± 2.210	32/32	$8.615 [3.200]$	[7.287, 8.753]
ORB	Essential	8	26.657 ± 9.850	32/32	28.653 [14.188]	[23.321, 29.997]
ORB	PnP	8	14.992 ± 3.972	32/32	$15.743 [4.970]$	[13.597, 16.272]

The normalization of ATE by the ground-truth trajectory length, reported alongside the absolute values in Table 5, places these magnitudes between 2.1% and 8.8% of the integrated trajectory length: the best baseline (KLT-PnP, stride 1) drifts at 2.14% of trajectory

length and the worst case (KLT-PnP, stride 8) at 8.77%. These percentages are consistent with frame-to-frame relative pose integration without keyframes, local bundle adjustment, or loop closure.

The main small-stride effects are supported by the paired Wilcoxon signed-rank tests over the 32 sequences. The raw and Holm–Bonferroni corrected p -values across the family of 16 baseline tests (two frontends $\times$ four strides $\times$ two metrics) are reported in Table 11. For KLT at strides 1–2, the PnP-vs-Essential ATE differences remain significant after correction ($p_{\text{Holm}} = 9.25 \times 10^{-6}$ and $p_{\text{Holm}} = 1.35 \times 10^{-6}$ under the one-sided test), and the RPE differences are also significant after correction. For KLT at stride 4, neither ATE nor RPE gives a significant PnP advantage even at the raw level. For ORB, the one-sided PnP-better hypothesis is significant for ATE and RPE at strides 1–4 ($p_{\text{Holm}} < 10^{-2}$ throughout); at stride 8, the ATE one-sided test is not significant after correction, while the RPE remains significant. At KLT stride 8, the two-sided tests indicate that PnP differs from Essential on both ATE and RPE under Holm correction, but the signs of the median differences disagree: PnP is worse on ATE (median $E-P = -0.69$ m) and better on RPE (median $E-P = +0.13$ m/frame), which is consistent with the frontend-dependent breakdown discussed above.

Table 11. Paired Wilcoxon signed-rank tests across the 16 baseline cells (2 frontends $\times$ 4 strides $\times$ 2 metrics), comparing per-sequence errors of Essential vs. PnP on shared sequences with finite values for both backends. One-sided alternative: error of Essential greater than PnP (i.e., PnP improves accuracy). Raw and Holm–Bonferroni corrected p -values are reported jointly across the family of 16 tests. **Bold:** corrected $p < 0.05$ under the one-sided test.

Metric	Frontend	Stride	n	Median($E-P$)	p_{raw} (1-s.)	p_{Holm} (1-s.)	p_{Holm} (2-s.)
ATE	KLT	1	32	3.356	1.16×10^{-6}	9.25×10^{-6}	1.85×10^{-5}
ATE	KLT	2	32	2.774	1.04×10^{-7}	1.35×10^{-6}	2.71×10^{-6}
ATE	KLT	4	32	0.297	0.6677	1.0000	0.9380
ATE	KLT	8	32	−0.690	0.9990	1.0000	0.0083
ATE	ORB	1	32	2.768	0.0003	0.0018	0.0036
ATE	ORB	2	32	2.752	1.28×10^{-8}	2.05×10^{-7}	4.10×10^{-7}
ATE	ORB	4	32	2.835	6.44×10^{-7}	6.44×10^{-6}	1.29×10^{-5}
ATE	ORB	8	32	0.942	0.1563	0.6253	0.9380
RPE	KLT	1	32	0.109	3.93×10^{-8}	5.90×10^{-7}	1.18×10^{-6}
RPE	KLT	2	32	0.113	3.45×10^{-7}	4.14×10^{-6}	8.29×10^{-6}
RPE	KLT	4	32	0.021	0.2052	0.6253	0.9380
RPE	KLT	8	32	0.127	0.0005	0.0026	0.0053
RPE	ORB	1	32	0.097	1.77×10^{-5}	0.0001	0.0002
RPE	ORB	2	32	0.180	7.15×10^{-8}	1.00×10^{-6}	2.00×10^{-6}
RPE	ORB	4	32	0.375	5.53×10^{-7}	6.08×10^{-6}	1.22×10^{-5}
RPE	ORB	8	32	0.382	8.67×10^{-7}	7.80×10^{-6}	1.56×10^{-5}

4.3. Isolated Geometry-Solver Latency

To quantify the solver-level computational scaling, the Essential Matrix estimation and PnP were measured on synthetic correspondences at $N \in \{100, 500, 1000, 2000, 5000\}$. The isolated-backend latency on the local workstation and on the Raspberry Pi 4 Model B is reported in Tables 12 and 13, respectively, with hardware stated in each table caption. Only the geometric solver call is measured by these timings; feature extraction, image acquisition, memory pressure, and thermal effects are not included.

On the local workstation, PnP was faster than Essential Matrix estimation for all tested correspondence counts, ranging from 1.76 ± 0.02 ms versus 3.44 ± 0.06 ms at 100 correspondences to 6.62 ± 0.07 ms versus 54.99 ± 0.42 ms at 5000 correspondences. On the Raspberry Pi 4 Model B, the pattern is not uniform at the smallest sample size: at 100 correspondences, PnP is slower (13.23 ± 0.65 ms) than Essential Matrix estimation (12.27 ± 0.10 ms). Nevertheless, for 500 correspondences and above, PnP is faster, reaching

25.56 ± 0.19 ms at 5000 correspondences compared with 109.71 ± 1.31 ms for Essential Matrix estimation.

Table 12. Isolated geometry backend latency (mean $\pm$ std., ms) on the local workstation (AMD Ryzen 9 7900X, 12C/24T, 64 GB RAM; Linux under WSL2), 20 warmup and 100 timed iterations per configuration. Bold indicates the lower mean latency within each correspondence-count row.

Correspondences	RGB-Only Essential	RGB-D PnP
100	3.4427 ± 0.0634	1.7604 ± 0.0210
500	9.3774 ± 0.0998	2.7394 ± 0.1012
1000	11.7428 ± 0.0519	2.7258 ± 0.0243
2000	23.1019 ± 0.1902	4.0841 ± 0.0279
5000	54.9851 ± 0.4223	6.6164 ± 0.0658

Table 13. Isolated geometry backend latency (mean $\pm$ std., ms) on a Raspberry Pi 4 Model B (Broadcom BCM2711 SoC; quad-core ARM Cortex-A72 at nominal 1.5 GHz; Raspberry Pi OS, 64-bit Linux; onboard LPDDR4 SDRAM capacity per SKU), 20 warmup and 100 timed iterations per configuration. Bold indicates the lower mean latency within each correspondence-count row.

Correspondences	RGB-Only Essential	RGB-D PnP
100	12.2661 ± 0.0986	13.2281 ± 0.6506
500	21.3037 ± 0.1584	10.2089 ± 0.0611
1000	38.2868 ± 0.1470	12.9623 ± 0.1990
2000	42.7376 ± 0.3211	11.8949 ± 0.1082
5000	109.7124 ± 1.3145	25.5630 ± 0.1927

These timings show that the PnP backend has favorable isolated-solver scaling for moderate and large correspondence sets in this implementation. They do not establish end-to-end real-time UAV feasibility, because the full image-processing and hardware pipeline was not benchmarked.

4.4. Robustness to Effective Frame Rate and RGB Degradation

The protocol described in Section 3.2.1 was executed for every combination of frontend, backend, temporal stride, and RGB degradation preset. The mean ATE values at stride 4 are reported in Table 14, and the trend over temporal stride and degradation is shown in Figure 2.

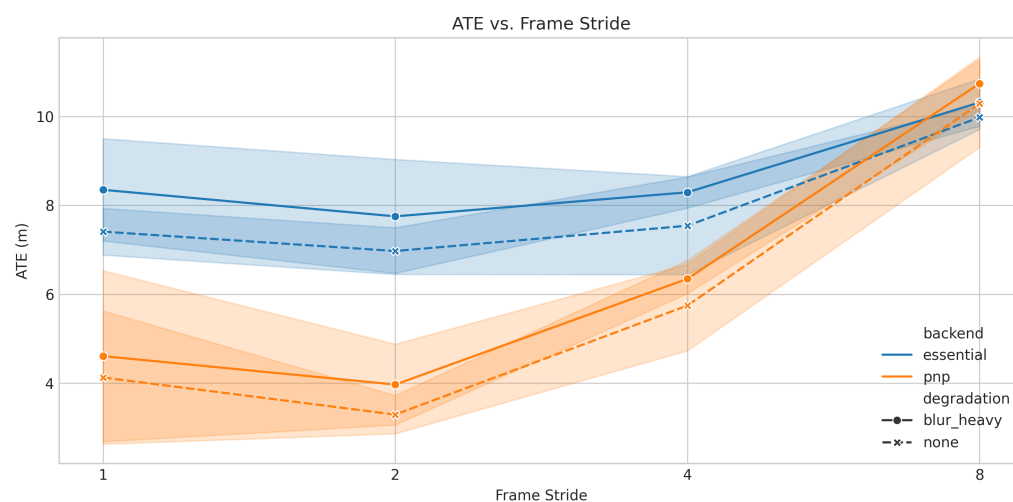


Figure 2. Mean ATE as a function of temporal stride k and RGB degradation preset, averaged over the evaluation sequences. Each subplot fixes one frontend/backend pair.

Table 14. Robustness to visual degradations (ATE in m, stride 4). Bold indicates the lower mean ATE within each frontend/degradation pair.

Frontend Backend Degradation	Essential	PnP	KLT	Essential	PnP	ORB
blur_heavy	7.940 ± 5.192	6.677 ± 4.499		8.646 ± 4.797	6.018 ± 4.352	
blur_mild	6.512 ± 3.760	6.482 ± 4.499		7.763 ± 4.732	4.749 ± 3.789	
gaussian_noise	6.447 ± 3.818	6.445 ± 4.301		7.900 ± 4.232	5.098 ± 4.211	
jpeg_low	7.053 ± 4.130	6.598 ± 4.615		8.393 ± 4.971	5.368 ± 4.084	
low_light	7.366 ± 4.835	6.528 ± 4.347		9.032 ± 5.520	4.921 ± 3.861	
none	6.443 ± 3.843	6.762 ± 4.471		8.640 ± 4.974	4.732 ± 3.761	

The robustness results show that RGB degradation and temporal subsampling increase both error and variance, but do not produce a single monotonic ranking across all frontend/backend pairs. At stride 4, ORB-PnP remains much better than ORB-Essential under all listed degradations. In contrast, KLT-PnP remains close to KLT-Essential and can be slightly worse in the no-degradation condition. This pattern suggests that depth integration is not sufficient by itself when tracked KLT correspondences become less reliable under larger inter-frame motion.

The finite sequence-level ATE and RPE values were produced by all baseline no-degradation configurations. Under degraded ORB configurations, nine non-finite ATE entries were observed, all on the same sequence (ocean_hard_p009) and each affecting one sequence out of 32 in its aggregate. The inspection of the saved trajectories shows that these cases are degenerate zero-motion trajectories caused by insufficient usable ORB correspondences under blur or coarse temporal stride; the affected sequence is excluded from the corresponding finite means.

4.5. Qualitative Trajectory and Depth Examples

The typical difference between the monocular and RGB-D branches under the selected example sequence is illustrated by the qualitative trajectory comparison in Figure 3. Because the monocular Essential backend is evaluated after global Sim(3) alignment, the visualization should be read as the comparison of trajectory shape after post hoc scale correction, not as the demonstration that the metric scale was recovered online by the monocular method.

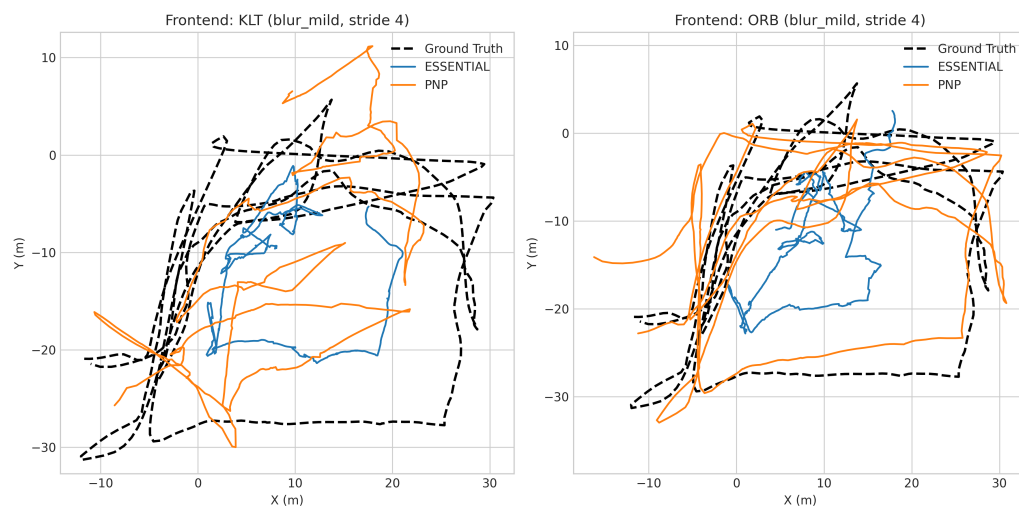


Figure 3. Estimated trajectories versus the TartanAir ground truth. RGB-only Essential Matrix trajectories are globally Sim(3)-aligned for evaluation, while RGB-D PnP trajectories are aligned with fixed scale.

The representative RGB correspondences and aligned depth values are shown in Figure 4. Very large simulator far-depth values may be included in the displayed depth maps, especially in outdoor sky regions. Not all displayed pixels are used by the PnP backend: only finite depth samples in the configured valid range $[0.1, 100]$ m are accepted for 3D back-projection.

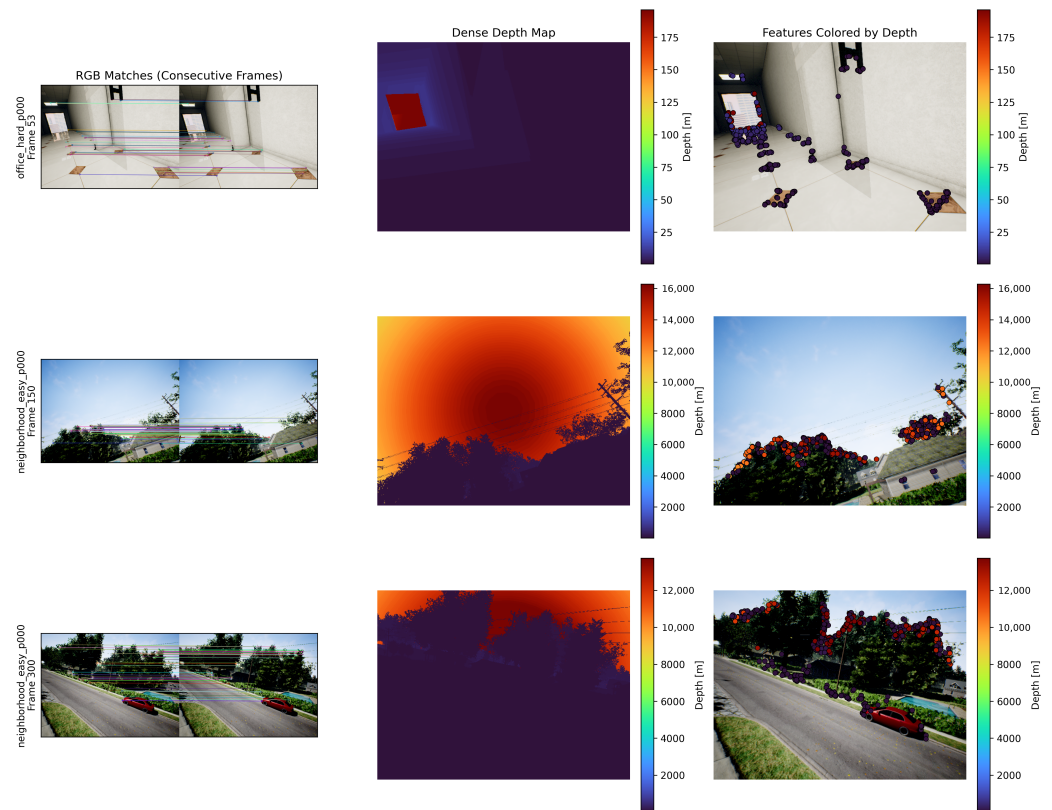


Figure 4. Visual comparison of RGB feature extraction and aligned dense depth information. The **left** image shows RGB matches between consecutive frames. The **middle** image shows the raw aligned depth map for visualization, including far simulator values. The **right** image shows extracted 2D keypoints colored by their corresponding depth values; only valid finite depth samples within the configured range enter the PnP backend.

5. Limitations

The reported evaluation is formulated as a controlled synthetic ablation. Several limitations are stated explicitly to prevent the empirical claims from extending beyond the conducted protocol.

Synthetic data only.

All experimental evaluations are executed on the simulated TartanAir validation split. The dataset is used as a controlled instrumentation environment, featuring synchronized RGB streams, aligned dense depth, calibrated intrinsics, and noise-free ground-truth poses. In contrast to this controlled regime, real UAV imagery introduces rolling-shutter distortion, auto-exposure transients, time-synchronization errors between RGB and depth sensors, and platform-induced vibration coupling. This limitation is a deliberate design choice, because authentic real-world sensor noise would obscure the pure algorithmic differences distinguishing the compared backends. The results should therefore be interpreted as a theoretical upper bound of performance for minimal perception systems, supplying clean data for the design of robust vision-based control loops.

Oracle depth channel.

The implemented RGB-D branch consumes the simulator's dense, aligned depth maps as an oracle channel, and the depth values are not perturbed under any condition. Real RGB-D sensors, in contrast, generate range-dependent noise (typically growing with the square of the metric range for stereo and time-of-flight devices), exhibit missing or invalid depth readings on glossy and far-field surfaces, and produce multi-path artifacts in time-of-flight devices alongside stereo disparity failures in textureless regions. Thus, the presented RGB-D results characterize the theoretical upper bound of the contribution provided by aligned metric depth to the geometric backend, rather than the behavior of a specific real-world depth sensor. A more realistic synthetic protocol—one that incorporates depth-magnitude-dependent noise alongside random per-pixel invalidation applied directly to the simulator depth maps on the basis of stereo-disparity dropouts and time-of-flight outliers—is reserved for future work and is expected to narrow the reported PnP advantage, particularly at the higher end of the standard operating range.

Calibration and synchronization not modeled.

The protocol assumes the RGB image, depth map, intrinsics, and ground-truth pose timestamps are calibrated and time-aligned. This assumption is rarely exact in real UAV flights. Small errors in focal length, principal point, lens distortion, RGB-D extrinsics, timestamping, or depth-to-color registration perturb the back-projected 3D points used by PnP. Such errors can turn a valid depth-supported correspondence into a biased reprojection constraint. Platform vibration and rolling shutter can also break the rigid global-shutter camera model assumed by both backends. These effects lie outside the present ablation by design. They are expected to reduce the observed RGB-D advantage, unless calibration, synchronization, and depth-validity checks are maintained on board.

Classical frontends only.

The evaluated frontend space is deliberately restricted to two classical correspondence pipelines, namely KLT optical-flow tracking and ORB descriptor matching. Advanced learned detectors and matchers, such as SuperPoint [33] and SuperGlue [34], as well as more recent dense matchers, are intentionally excluded from the present evaluation. Since the reported frontend-dependent breakdown of the PnP advantage at coarse temporal strides is a property of the underlying 2D correspondence layer, the conclusions regarding when depth provides a benefit should not be directly extrapolated to learned frontends without re-running the identical backend ablation with those specific correspondences.

Synthetic RGB degradations only.

The application of the five RGB perturbations summarized in Table 1 employs deliberately moderate, fixed parameter values, avoiding the implementation of complex per-sequence sensor-calibrated noise models. These perturbations are intended as a controlled methodological probe of frontend robustness, not as a substitute for modeling the full image signal pipeline of an authentic UAV camera.

No SLAM baseline.

As detailed in Section 3.1, fully integrated SLAM systems, such as ORB-SLAM2, SVO, and DSO, are deliberately excluded from the applied comparison baselines. Such systems routinely incorporate keyframing, local bundle adjustment, loop closure mechanisms, and frequently photometric direct optimization, all of which would compensate for and obscure the per-frame geometric backend effect isolated within this study.

Static backend selection.

Each evaluated system configuration commits to a single geometric backend applied uniformly across the whole sequence: the RGB-only pipelines continuously execute Essential Matrix estimation, and the RGB-D pipelines continuously use PnP, regardless of the fluctuating per-frame correspondence quality. The observed breakdown of KLT-PnP specifically at coarse temporal strides, where the depth-supported inlier set becomes degraded while the classical two-view Essential geometry remains usable, suggests that an adaptive dynamic selector—one capable of switching between PnP and Essential per frame on the basis of the depth-supported inlier count, the residual reprojection statistics, or the quantified validity rate of the depth channel—could recover the superior performance of both backends without modifying the implemented frontend. Since such a selector is neither implemented nor evaluated, the reported quantitative data should be interpreted as the behavior of two fixed backend policies, rather than the maximum achievable envelope of a dynamic backend.

Isolated solver latency only.

The timing results acquired on the Raspberry Pi 4 Model B and the local workstation, documented in Tables 12 and 13, measure the computational latency of the geometric solver call executed on synthetic correspondences. In contrast to this isolated measurement, authentic end-to-end onboard pipeline timing would include camera I/O, image rectification, feature extraction or tracking, depth-stream handling, memory pressure, and thermal throttling, all of which can dominate the computational budget on constrained edge platforms. The stated numbers bound the geometry-solver contribution to the overall cycle budget, but cannot establish the achievable end-to-end real-time VO or SLAM throughput capacity for any specific UAV hardware configuration.

Frame-to-frame integration without keyframes.

All four evaluated algorithmic configurations are designed to integrate relative pose on a frame-to-frame basis and avoid the implementation of keyframe selection, sliding-window bundle adjustment, or loop closure techniques. The measured absolute ATE magnitudes therefore accumulate a higher level of drift than any comparably tuned full SLAM stack subjected to the identically provided data; the obtained percentage drift should be considered as a baseline representing simple frame-to-frame integration, rather than a metric indicative of a full SLAM result.

Single-agent setup.

The presented methodological evaluation models a single-agent ego-motion algorithmic pipeline. For comprehensive perception and mapping executed cooperatively by multiple interconnected robots, this local pipeline must be augmented on the basis of reliable map fusion, relative positioning algorithms, and complex distributed bundle adjustment frameworks. Nevertheless, the development of an optimized, lightweight local odometry module remains a mandatory prerequisite for any functional multi-agent system. The anticipated multi-UAV or advanced swarm-level extensions are discussed in Section 6; the controlled per-agent backend characterization detailed here is intended to serve as a building block for those extensions, rather than as a substitute for them.

6. Conclusions

This paper evaluated the algorithmic trade-offs of minimal sparse VO pipelines through a controlled synthetic ablation. The primary contribution is not a new VO algorithm or a complete UAV navigation system, but a reproducible diagnostic comparison

of RGB-only Essential Matrix and RGB-D PnP geometric backends under fixed ORB and KLT frontends, executed on the 32-sequence TartanAir validation split and accompanied by an embedded-platform characterization of the isolated geometry solver. The limitations of this controlled setup are summarized in Section 5.

The numerical outcomes show that depth-assisted PnP can reduce trajectory error under favorable, small-baseline conditions. Within the baseline stride-1 setting, KLT-PnP reduces the ATE from 6.885 m to 2.633 m (61.8%) and decreases the translational RPE from 0.155 m/frame to 0.061 m/frame ($\sim 61.6\%$); ORB-PnP reduces the ATE from 7.933 m to 5.627 m (29.1%) and suppresses the translational RPE from 0.203 m/frame to 0.118 m/frame (41.9%). The evaluated rotational error exhibits a stronger and more uniform PnP-related advantage: at stride 1, the rotational RPE drops from 15.119 deg/frame to 2.134 deg/frame for KLT (85.9%) and from 11.801 deg/frame to 2.674 deg/frame for ORB (77.3%), while this reduction is preserved across every tested temporal stride for both frontends. The decomposition of the PnP ATE via Sim(3) alignment demonstrates that a small portion (7–16%) of the absolute accumulated error is directly attributable to metric-scale drift, with the remainder constituting trajectory-shape error that standard scale correction cannot remove. The main caveat is equally important: depth assistance is not sufficient by itself. KLT-PnP fails to improve ATE at coarser temporal strides, where the underlying 2D tracking and depth-supported inlier set become less reliable, whereas ORB-PnP retains a measurable ATE advantage across all tested strides. Thus, the observed benefit of aligned depth is conditional on the stability of the frontend correspondences and the validity of the 3D-to-2D constraints.

Practical interpretation.

The Raspberry Pi 4 Model B measurements in Table 13 characterize only the isolated geometry-solver call. They indicate that the PnP backend is not the dominant computational obstacle in this controlled setup, but they do not establish end-to-end onboard real-time performance. A deployed VO or SLAM pipeline would also include image acquisition, feature extraction or tracking, depth-stream handling, memory pressure, synchronization, thermal behavior, and integration with higher-level control. Therefore, the results are best interpreted as backend-level evidence for lightweight VO design rather than as validation of a complete UAV, edge, or swarm perception system.

Future work and links to real-world experiments.

The next research step is to migrate the evaluated protocol from pure simulation onto representative real-world robotic platforms or more realistic sensor models, while maintaining the comparison policy detailed in Section 3.1. Several concrete research directions, aligned with the limitations outlined in Section 5, are as follows:

- The identical controlled backend ablation protocol should be re-run on representative indoor and small Micro Aerial Vehicle (MAV) real-world datasets, including EuRoC MAV [36] and TUM RGB-D [4], using KITTI Odometry [37] as an established outdoor ground-vehicle comparative reference. This would expose the evaluated algorithmic backends to genuine hardware sensor noise, complex rolling-shutter artifacts, and motion regimes that the currently used synthetic TartanAir validation split does not reproduce.
- Before initiating that real-world transition, the underlying synthetic protocol itself should be modified to use a more realistic parameterization. Specifically, adding depth-magnitude-dependent stochastic noise (featuring a variance that grows proportionally with the recorded range to mirror established stereo and time-of-flight hardware characteristics) alongside random per-pixel invalidation applied directly

to the simulator depth maps would allow the identical TartanAir split to probe how much of the reported PnP advantage survives once the depth channel stops acting as an oracle, without losing the controlled instrumentation benefits of simulation.

- The evaluated frontend space should be widened from the classical KLT and ORB implementations towards learned detectors and modern matchers, including SuperPoint [33] and SuperGlue [34] functional pairings, as well as more recent dense matchers. Because the reported translational advantage demonstrated by PnP is dependent upon the frontend formulation and the motion-regime characteristics observed at coarse temporal strides, re-running the identical controlled backend ablation under learned correspondences would clarify whether stronger 2D matches narrow, preserve, or invert the performance gap that the classical frontends presently expose.
- Motivated by the documented frontend-dependent operational breakdown of KLT-PnP at coarse strides—whereas KLT-Essential remains usable—a promising algorithmic extension is a dynamic backend selector that routes each consecutive frame through PnP when the depth-supported inlier count, the calculated depth validity ratio, and the PnP reprojection residuals lie within a statistically confident range, while falling back to the Essential algorithm otherwise. Such a selector would preserve the present comparison’s fixed frontend design and could transform the empirical operational caveat reported here into a concrete scientific contribution.
- The idealized oracle depth assumption should be replaced by realistic RGB-D streams generated by flight-deployable hardware sensors, such as Intel RealSense D435i, Microsoft Azure Kinect DK, and Luxonis OAK-D-class smart cameras. This would allow the algorithmic contribution of depth to be re-quantified directly under range-dependent depth noise, widespread invalid pixels, and systematic field-of-view mismatch. OAK-D-class devices are especially relevant to the embedded processing thread because they couple a dedicated stereo depth pipeline with a specialized on-device Movidius-class Vision Processing Unit (VPU), which can off-load demanding feature detection, description, or matching from the constrained host CPU.
- The presently recorded embedded measurements should be extended from the isolated solver latency evaluation to the end-to-end onboard pipeline throughput analysis executed directly on the Raspberry Pi 4 Model B, NVIDIA Jetson Orin-class high-performance platforms, and specialized OAK-D-style smart-camera edge modules, reporting the accumulated drift measured over total flight time, the algorithmic recovery rate achieved after severe depth dropouts, and the total wall-clock VO / SLAM execution latency—simultaneously including camera Input/Output (I/O), initial feature extraction, and complete depth handling operations, rather than evaluating the isolated geometry solver alone.

Author Contributions: Conceptualization, A.P. and Y.G.; methodology, A.P.; software, A.P.; validation, A.P.; formal analysis, A.P.; investigation, A.P.; resources, A.P., S.S., M.L., G.J., O.R., O.A. and Y.G.; data curation, A.P.; writing—original draft preparation, A.P.; writing—review and editing, A.P., S.S., M.L., G.J., O.R., O.A., N.N. and Y.G.; visualization, A.P.; supervision, Y.G. and S.S.; project administration, Y.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the US National Academy of Sciences (US NAS) and the Office of Naval Research Global (ONRG) (IMPRESS-U initiative, No. STCU-7125) as part of exploratory research on new robust computer vision approaches, and by the NATO Science for Peace and Security Programme under grant id. G6032 as part of the development of UAV-oriented Edge Intelligence solutions.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The experiments use a publicly available TartanAir validation split. The exact 32-sequence dataset split used in this study, together with its exploratory data analysis, is published on Kaggle [11]. The evaluation code, including the sparse frontends, geometric backends, deterministic RGB degradation presets, and metric scripts used to produce the reported results, is released as open-source software on GitHub [12].

Acknowledgments: The authors acknowledge the use of publicly available TartanAir data for repeatable evaluation of visual odometry pipelines. During the preparation and revision of this manuscript, the authors used OpenAI ChatGPT/Codex (GPT-5-based system, accessed in June 2026) for language polishing, grammar correction, manuscript organization, LaTeX/formatting assistance, and author-supervised drafting of revision materials. The authors reviewed and edited the output and take full responsibility for the content of this publication. The GenAI tools were not used as authors and did not independently generate experimental data, perform the final analyses, or determine the scientific conclusions.

Conflicts of Interest: The authors declare no conflicts of interest. The funding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

ATE	Absolute Trajectory Error
GNSS	Global Navigation Satellite System
KLT	Kanade–Lucas–Tomasi
ORB	Oriented FAST and Rotated BRIEF
PnP	Perspective-n-Point
RANSAC	Random Sample Consensus
RGB-D	Red–Green–Blue plus depth
RPE	Relative Pose Error
SLAM	Simultaneous Localization and Mapping
UAV	Unmanned Aerial Vehicle
VO	Visual Odometry

References

1. Choi, S.; Park, J.; Yu, W. Resolving scale ambiguity for monocular visual odometry. In *Proceedings of the 2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*; IEEE: Piscataway, NJ, USA, 2013; pp. 604–608. <https://doi.org/10.1109/URAI.2013.6677403>.
2. Kitt, B.; Rehder, J.; Chambers, A.D.; Schönbein, M.; Lategahn, H.; Singh, S. Monocular visual odometry using a planar road model to solve scale ambiguity. In *Proceedings of the European Conference on Mobile Robots (ECMR)*, Örebro, Sweden, 7–9 September 2011; pp. 1–6. <https://doi.org/10.1184/R1/6555626>.
3. Zhang, J.; Kaess, M.; Singh, S. Real-time Depth Enhanced Monocular Odometry. In *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, IL, USA, 14–18 September 2014; pp. 4973–4980. <https://doi.org/10.1109/IROS.2014.6943269>.
4. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A Benchmark for the Evaluation of RGB-D SLAM Systems. In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Portugal, 7–12 October 2012; pp. 573–580. <https://doi.org/10.1109/IROS.2012.6385773>.
5. Nistér, D. An Efficient Solution to the Five-Point Relative Pose Problem. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 756–770. <https://doi.org/10.1109/TPAMI.2004.17>.
6. Lepetit, V.; Moreno-Noguer, F.; Fua, P. EPnP: An Accurate $O(n)$ Solution to the PnP Problem. *Int. J. Comput. Vis.* **2009**, *81*, 155–166. <https://doi.org/10.1007/s11263-008-0152-6>.
7. Mur-Artal, R.; Tardos, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. <https://doi.org/10.1109/TRO.2017.2705103>.

8. Forster, C.; Pizzoli, M.; Scaramuzza, D. SVO: Fast Semi-Direct Monocular Visual Odometry. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation, Hong Kong, China, 31 May–7 June 2014; pp. 15–22. <https://doi.org/10.1109/ICRA.2014.6906584>.
9. Engel, J.; Koltun, V.; Cremers, D. Direct Sparse Odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 611–625. <https://doi.org/10.1109/TPAMI.2017.2658577>.
10. Wang, W.; Zhu, D.; Wang, X.; Hu, Y.; Qiu, Y.; Wang, C.; Hu, Y.; Kapoor, A.; Scherer, S. TartanAir: A Dataset to Push the Limits of Visual SLAM. In *Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: Piscataway, NJ, USA, 2020; pp. 4909–4916. <https://doi.org/10.1109/IROS45743.2020.9341801>.
11. Polukhin, A.; Stirenko, S.; Leier, M.; Jervan, G.; Rokovyi, O.; Alienin, O.; Nazeer, N.; Gordienko, Y. DASVO TartanAir RGB-D Validation Split. 2026. Available online: <https://www.kaggle.com/datasets/pandrii000/dasvo-tartanair-rgb-d-validation-split> (accessed on 28 June 2026).
12. Polukhin, A.; Stirenko, S.; Leier, M.; Jervan, G.; Rokovyi, O.; Alienin, O.; Nazeer, N.; Gordienko, Y. DASVO: Depth-Assisted Sparse Visual Odometry—Evaluation Code. 2026. Available online: <https://github.com/pandrii000/dasvo> (accessed on 28 June 2026).
13. Jantos, T.; Scheiber, M.; Brommer, C.; Allak, E.; Weiss, S.; Steinbrener, J. AIVIO: Closed-loop, Object-relative Navigation of UAVs with AI-aided Visual Inertial Odometry. *IEEE Robot. Autom. Lett.* **2024**, *9*, 10764–10771. <https://doi.org/10.1109/LRA.2024.3479713>.
14. Al-Jarrah, O.Y.; Shatnawi, A.S.; Shurman, M.M.; Ramadan, O.A.; Muhaidat, S. Exploring Deep Learning-Based Visual Localization Techniques for UAVs in GPS-Denied Environments. *IEEE Access* **2024**, *12*, 113049–113071. <https://doi.org/10.1109/ACCESS.2024.3440064>.
15. Rostum, H.M.; Vásárhelyi, J. A Review of Using Visual Odometry Methods in Autonomous UAV Navigation in GPS-Denied Environment. *Acta Univ. Sapientiae Electr. Mech. Eng.* **2023**, *15*, 14–32. <https://doi.org/10.2478/auseme-2023-0002>.
16. Neyestani, A.; Picariello, F.; Ahmed, I.; Daponte, P.; De Vito, L. From Pixels to Precision: A Survey of Monocular Visual Odometry in Digital Twin Applications. *Sensors* **2024**, *24*, 1274. <https://doi.org/10.3390/s24041274>.
17. Wang, S.; Ahmad, N.S. A comprehensive review on sensor fusion techniques for localization of a dynamic target in GPS-denied environments. *IEEE Access* **2025**, *13*, 2252–2285. <https://doi.org/10.1109/ACCESS.2024.3519874>.
18. Jiang, H.; Cheng, Y.; Dai, W.; Wan, W.; Wang, F. GPS-denied LiDAR-based SLAM—A Survey. *IET Cyber-Syst. Robot.* **2025**, *7*, e70031. <https://doi.org/10.1049/CSY2.70031>.
19. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.M.; Tardós, J.D. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. <https://doi.org/10.1109/TRO.2021.3075644>.
20. Teed, Z.; Deng, J. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–14 December 2021; Volume 34, pp. 7164–7177. <https://doi.org/10.48550/ARXIV.2108.10869>.
21. Teed, Z.; Lipson, L.; Deng, J. Deep Patch Visual Odometry. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 10–16 December 2023; Volume 36, pp. 73562–73574. <https://doi.org/10.48550/ARXIV.2208.04726>.
22. Lipson, L.; Teed, Z.; Deng, J. Deep Patch Visual SLAM. In *Proceedings of the Computer Vision—ECCV 2024; Lecture Notes in Computer Science*; Springer Nature: Cham, Switzerland, 2024; pp. 424–440. https://doi.org/10.1007/978-3-031-72627-9_24.
23. Yan, Y.; Li, J.; Liu, Q.; Lv, S.; Wu, Q. RGB-D SLAM: A Review of Methods and Performance Trade-Offs for Different Requirements. *Sensors* **2026**, *26*, 3513. <https://doi.org/10.3390/s26113513>.
24. Mokssit, S.; Bonilla Licea, D.; Guermah, B.; Ghogho, M. Deep Learning Techniques for Visual SLAM: A Survey. *IEEE Access* **2023**, *11*, 20026–20050. <https://doi.org/10.1109/ACCESS.2023.3249661>.
25. Lucas, B.D.; Kanade, T. An Iterative Image Registration Technique with an Application to Stereo Vision. In Proceedings of the 7th International Joint Conference on Artificial Intelligence, Vancouver, BC, Canada, 24–28 August 1981; pp. 674–679.
26. Bouguet, J.Y. *Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the Algorithm*; Intel Corporation Microprocessor Research Labs: Hillsboro, OR, USA, 2001.
27. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An Efficient Alternative to SIFT or SURF. In Proceedings of the 2011 IEEE International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571. <https://doi.org/10.1109/ICCV.2011.6126544>.
28. Delmerico, J.; Scaramuzza, D. A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots. In *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: Piscataway, NJ, USA, 2018; pp. 2502–2509. <https://doi.org/10.1109/ICRA.2018.8460664>.
29. Agostinho, L.R.; Ricardo, N.M.; Pereira, M.I.; Hiolle, A.; Pinto, A.M. A practical survey on visual odometry for autonomous driving in challenging scenarios and conditions. *IEEE Access* **2022**, *10*, 72228–72260. <https://doi.org/10.1109/access.2022.3188990>.

30. He, M.; Zhu, C.; Huang, Q.; Ren, B.; Liu, J. A review of monocular visual odometry. *Vis. Comput.* **2020**, *36*, 1053–1065. <https://doi.org/10.1007/s00371-019-01714-6>.
31. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. <https://doi.org/10.1145/358669.358692>.
32. Tomasi, C.; Kanade, T. *Detection and Tracking of Point Features*; Technical Report CMU-CS-91-132; Carnegie Mellon University: Pittsburgh, PA, USA, 1991.
33. DeTone, D.; Malisiewicz, T.; Rabinovich, A. SuperPoint: Self-Supervised Interest Point Detection and Description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 224–236. <https://doi.org/10.1109/CVPRW.2018.00060>.
34. Sarlin, P.E.; DeTone, D.; Malisiewicz, T.; Rabinovich, A. SuperGlue: Learning Feature Matching with Graph Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 4938–4947. <https://doi.org/10.1109/CVPR42600.2020.00499>.
35. Polukhin, A.; Gordienko, Y.; Leier, M.; Jervan, G.; Rokoyi, O.; Alienin, O.; Stirenko, S. Edge Intelligence Resource Consumption by UAV-based IR Object Detection. In Proceedings of the 2023 Workshop on UAVs in Multimedia: Capturing the World from a New Perspective, Ottawa, ON, Canada, 29 October–3 November 2023; pp. 57–61. <https://doi.org/10.1145/3607834.3616566>.
36. Burri, M.; Nikolic, J.; Gohl, P.; Schneider, T.; Rehder, J.; Omari, S.; Achtelik, M.W.; Siegwart, R. The EuRoC Micro Aerial Vehicle Datasets. *Int. J. Robot. Res.* **2016**, *35*, 1157–1163. <https://doi.org/10.1177/0278364915620033>.
37. Geiger, A.; Lenz, P.; Urtasun, R. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361. <https://doi.org/10.1109/CVPR.2012.6248074>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.