Comparative Analysis of Disparity-Filtered vs. Unrestricted Feature Matching in Visual Odometry

Author: Shreyas Yellenki

Advisors for 16-597: Dr. David Wettergreen and Tushaar Jain

1 Motivation

Visual odometry systems are based on establishing correspondences between feature points across consecutive frames to estimate camera motion. The current pipeline for Moonranger implements the following strategy: *feature detection on Frame A → disparity filter (Frame A) → feature matching with Frame B → disparity filter (Frame B).* This approach applies disparity filtering at two critical junctures: immediately after feature detection in the first frame and again after establishing tentative matches with the second frame. Although this strategy guaranties that all retained features possess valid depth information, it introduces a fundamental limitation: any feature point lacking a corresponding disparity value is discarded before the matching algorithm can evaluate its potential utility.

This pre-filtering approach constrains the search space available for feature correspondence. Environments with sparse depth measurements, such as textureless regions (Ex: the sand testing bed of Moonranger), the disparity filter can eliminate a substantial proportion of detected features, low feature counts can lead to a brittle motion estimation system. An alternative approach presents itself: *feature detection (Frame A) → feature matching with Frame B → RANSAC-based outlier rejection.* This shift eliminates the disparity filtering stage, allowing the matching algorithm to consider the complete set of detected features regardless of depth availability. An initial border filter is used only to remove features near image boundaries, where matching would be unreliable.

The fundamental insight motivating this alternative lies in recognizing that feature correspondences without associated depth measurements are not valueless, they can still be used to constrain motion. This epipolar constraint provides information about the camera's rotation and the *direction* of translation, though the magnitude (scale) remains ambiguous without depth. The following analysis was done on the argus high slip dataset.

2 Methodology

2.1 RANSAC-Based 2D Feature Correspondence

The proposed flow leverages the Random Sample Consensus (RANSAC) to distinguish valid feature correspondences from poor-quality matches. Unlike the original flow, which employed a 3D clique-finding algorithm requiring valid depth measurements for all points, the new approach operates purely in the 2D plane. An advantage to this approach is it imposes no depth requirement, enabling utilization of a larger pool of features.

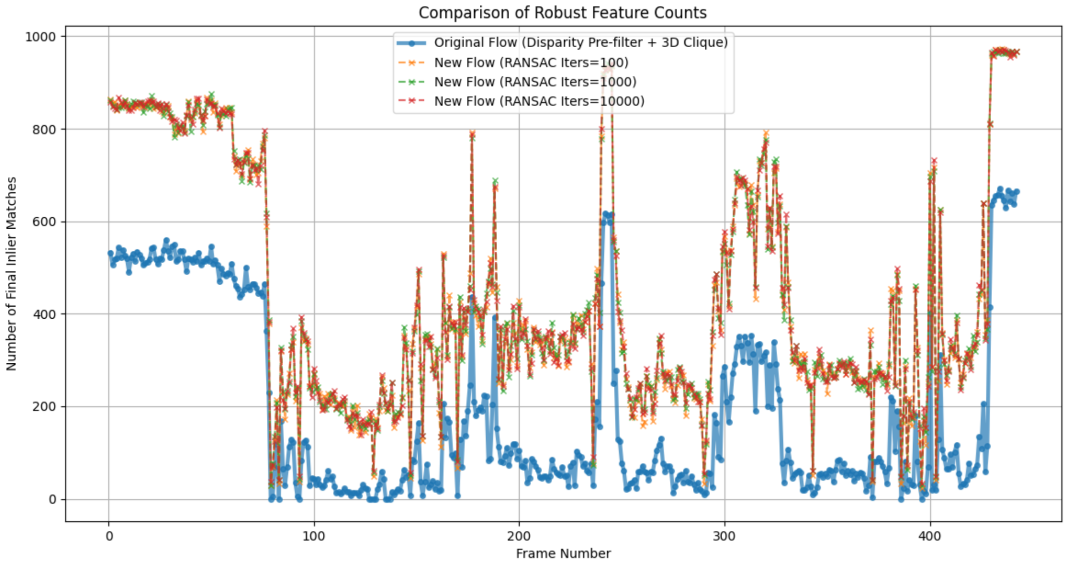


Figure 1: Comparison of feature points identified by the original method versus the proposed approach across the argus high slip dataset. The new flow consistently recovers significantly more correspondences.

2.2 Computational Performance Analysis

Eliminating disparity pre-filtering substantially increases the input for feature matching, which leads to a more expensive algorithm when comparing time complexity. We also compared the difference in time between the different outlier rejection methods (3D Clique algorithm in the current flow, and 2D RANSAC in the proposed flow). Particular attention was devoted to the RANSAC iteration parameter (`max_iters`), to see if there was a note-worthy difference in the time it took the algorithm to run compared to the number of feature points it can discover. To accurately test the parameter, the confidence parameter of 2D Ransac was set to .9999 to force the algorithm to run to close as possible for max iterations.

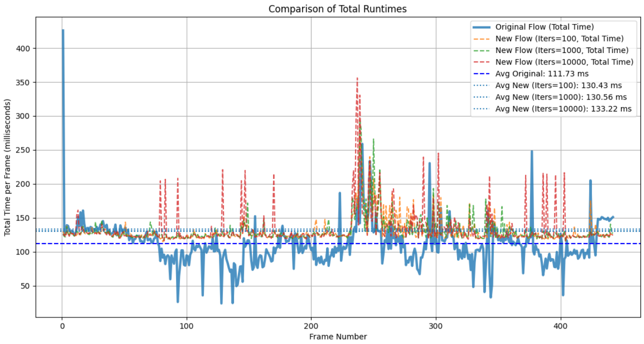


Figure 2: Computational cost breakdown comparing the different runtimes of the algorithm with varying `max_iters` parameter. We declared the benefit of increasing the parameter was negligible, as the number of feature points found was nearly identical even though there was an increase in runtime. For upcoming graphs, the results of runs where `max_iters` is 1000 are shown. As expected the runtime of the new flow was greater for nearly every frame in the dataset when compared to the new flow.

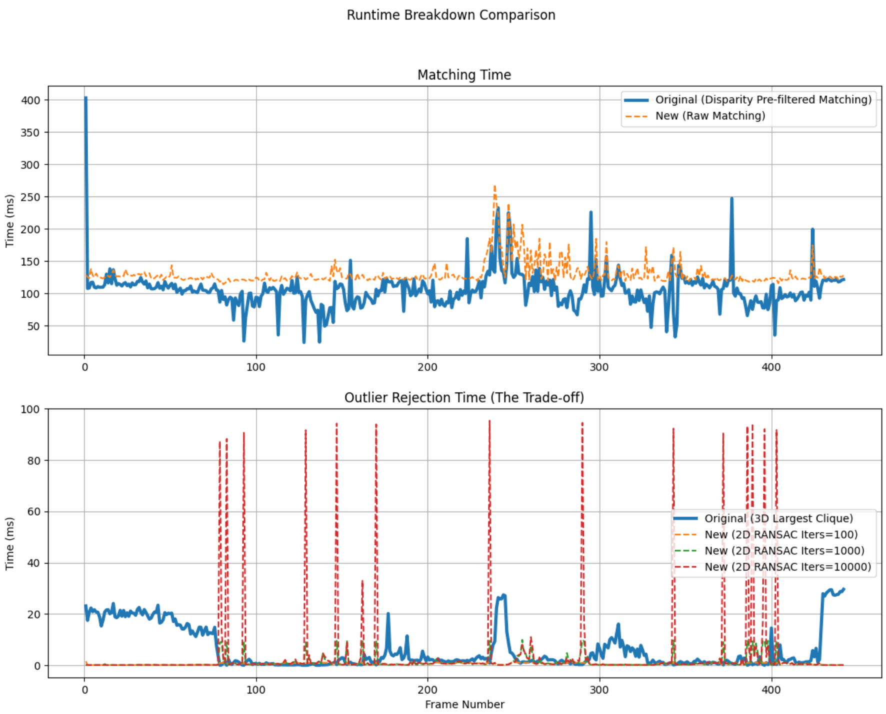


Figure 3: A breakdown on the primary categories of how the runtime is split between the two algorithms: feature matching and outlier rejection

2.3 Motion Estimation and Geometric Comparison

The proposed approach yields motion estimates through the following sequence:

1. **Fundamental Matrix Estimation:** RANSAC identifies the fundamental matrix $\mathbf{F}$ that best explains the inlier correspondence set.

2. **Essential Matrix Recovery:** The fundamental matrix operates in pixel coordinates, while motion estimation requires working in normalized camera coordinates. Given the camera intrinsic matrix $\mathbf{K}$, the essential matrix is recovered via:

$$\mathbf{E} = \mathbf{K}^T \mathbf{F} \mathbf{K}$$

3. **Pose Decomposition:** The essential matrix admits four possible decompositions into rotation $\mathbf{R}$ and translation $\mathbf{t}$. OpenCV's `cv2.recoverPose` “recovers” the 3D motion from the essential matrix by performing a chirality check. For each of the 4 possible solutions it triangulates the 2D inlier points into 3D and counts how many of those 3D points end up in front of both cameras. The solution (the $\mathbf{R}_{new}$ and $\mathbf{t}_{new}$) that results in the most points being ”in front” is declared the winner and returned. $\mathbf{t}_{new}$ emerges as a unit vector indicating direction but not magnitude.

The original pipeline, operating on 3D point correspondences with known depth, produces a translation vector $\mathbf{t}_{orig}$ with meaningful scale information derived directly from metric 3D geometry. To enable meaningful comparison, $\mathbf{t}_{orig}$ was normalized to unit length, reducing both estimates to pure directional information:

$$\hat{\mathbf{t}}_{orig\_norm} = \frac{\mathbf{t}_{orig}}{\|\mathbf{t}_{orig}\|}$$

The angular deviation between these unit vectors as well as there components were measured.

$$\theta_t = \arccos(\hat{\mathbf{t}}_{orig\_norm}^T \hat{\mathbf{t}}_{new})$$

2.4 Metric Scale Recovery

While the RANSAC-based approach yields only directional translation information, metric scale can be recovered when a subset of feature correspondences possesses valid depth measurements. We extend the feature points we discovered in our new flow with depth and use them to calculate scale with the following derivation:

Goal: Find scale s
Starting from the relationship between coordinate frames:

$$^0P = {}^0R_1 \cdot {}^1P + {}^0t_1 \tag{1}$$

We want to minimize the reprojection error:

$$\min \left\|{}^0R_1 \cdot {}^1P + {}^0t_1 - {}^0P\right\|^2 \tag{2}$$

Substituting ${}^0t_1 = s \cdot \hat{t}$ where $\hat{t}$ is the unit translation vector:

$$\min \left\|{}^0R_1 \cdot {}^1P + s\hat{t} - {}^0P\right\|^2 \tag{3}$$

Rearranging terms:

$$\min_s \left\|({}^0R_1 \cdot {}^1P - {}^0P) + s\hat{t}\right\|^2 \tag{4}$$

Let $\mathbf{a}_i = ({}^0R_1 \cdot {}^1P_i - {}^0P_i)$, then we can view this as a least squares problem:

$$\min_s \sum_i \left\|\mathbf{a}_i + s\hat{t}\right\|^2 \tag{5}$$

Expanding the squared norm:

$$\sum_i \left\|\mathbf{a}_i + s\hat{t}\right\|^2 = \sum_i \left[(\mathbf{a}_i \cdot \mathbf{a}_i) + 2s(\mathbf{a}_i \cdot \hat{t}) + s^2(\hat{t} \cdot \hat{t})\right] \tag{6}$$

Taking the derivative with respect to s and setting to zero:

$$\frac{d}{ds} \sum_i \left[(\mathbf{a}_i \cdot \mathbf{a}_i) + 2s(\mathbf{a}_i \cdot \hat{t}) + s^2(\hat{t} \cdot \hat{t})\right] = 0 \tag{7}$$

$$\sum_i \left[2(\mathbf{a}_i \cdot \hat{t}) + 2s(\hat{t} \cdot \hat{t})\right] = 0 \tag{8}$$

Since $\hat{t}$ is a unit vector, $\hat{t} \cdot \hat{t} = 1$, therefore $\sum_i (\hat{t} \cdot \hat{t}) = N$ where N is the number of points:

$$s \cdot N = -\sum_i (\mathbf{a}_i \cdot \hat{t}) \tag{9}$$

Solving for scale:

$$s = -\frac{1}{N} \sum_i (\mathbf{a}_i \cdot \hat{t}) \tag{10}$$

where $\mathbf{a}_i = ({}^0R_1 \cdot {}^1P_i - {}^0P_i)$ and N is the total number of point correspondences.

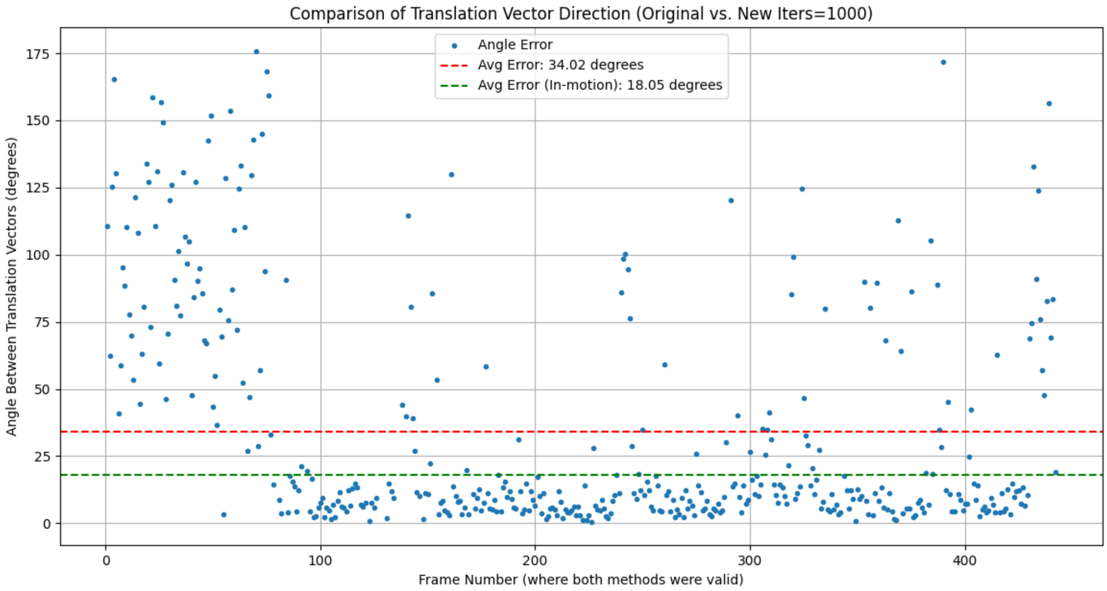


Figure 4: In the beginning of the dataset, the rover is stationary leading to outliers in the angle difference between translation vectors as the zero vector does not have a well defined angle. The average was calculated for the whole dataset, and again for when the rover is in motion to better understand the relationship between both flows.

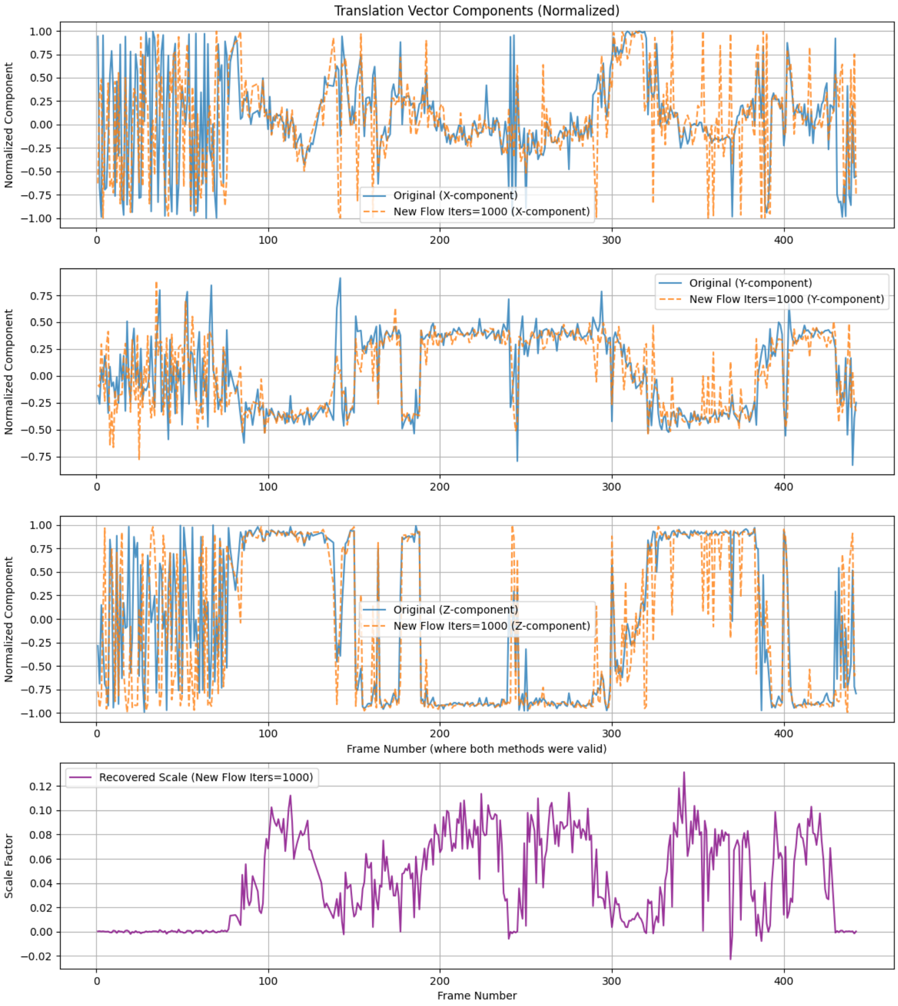


Figure 5: Components of both translation vectors (as unit vectors) as well as the recovered scale from the new VO flow. As mentioned before, the initial frames in the dataset the rover is stationary leading the translation vector to essentially be a random vector close to **(0,0,0)** due to noise.

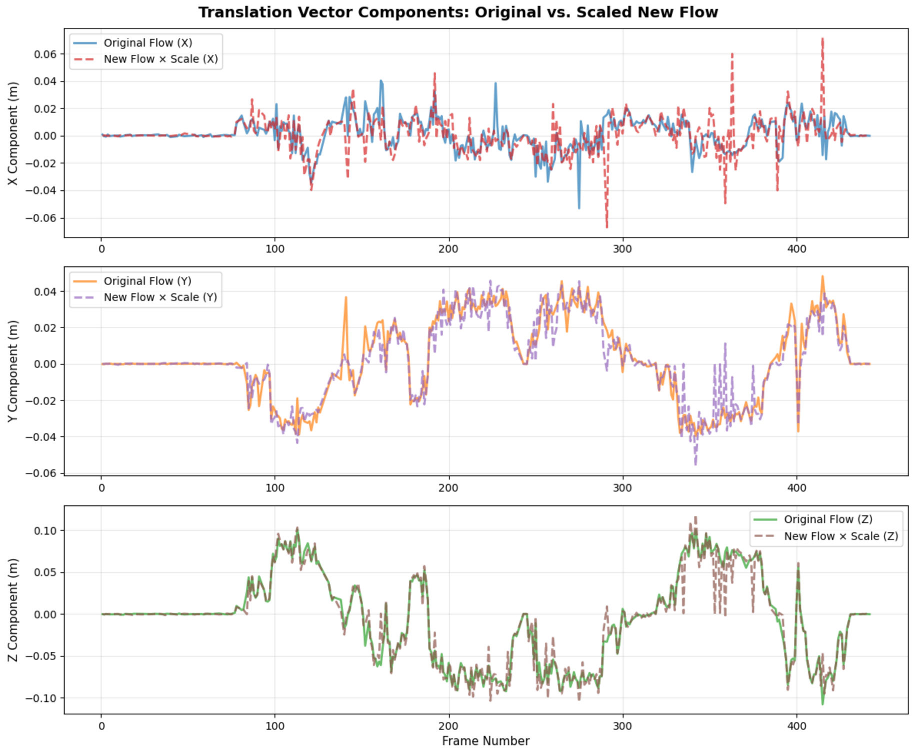


Figure 6: The components of the original translation vector compared the components of the unit translation vector from the new flow multiplied by scale. Further analysis is required to validate both flows to determine which provides a more accurate measurement.

2.4.1 Rotation Error Quantification

Both pipelines produce rotation matrices: $\mathbf{R}_{orig}$ from the original approach and $\mathbf{R}_{new}$. The relative rotation error can be expressed as a correction matrix $\mathbf{C}$.

$$\mathbf{R}_{orig} = \mathbf{R}_{new} \cdot \mathbf{C} \implies \mathbf{C} = \mathbf{R}_{new}^T \cdot \mathbf{R}_{orig}$$

While $\mathbf{C}$ characterizes the rotational difference, a 3×3 matrix proves difficult to interpret intuitively so we used an axis-angle interpretation which yields a scalar angle θ_r, providing a digestible metric for analysis.

It should be noted that the observed differences between the two methods do not necessarily indicate inferiority of the new flow. The original pipeline may itself have errors arising from depth measurement noise, feature uncertainty, or algorithmic approximations in the 3D clique solver. These comparisons serve purely to quantify the divergence between the two methodologies, establishing expectations rather than correctness.

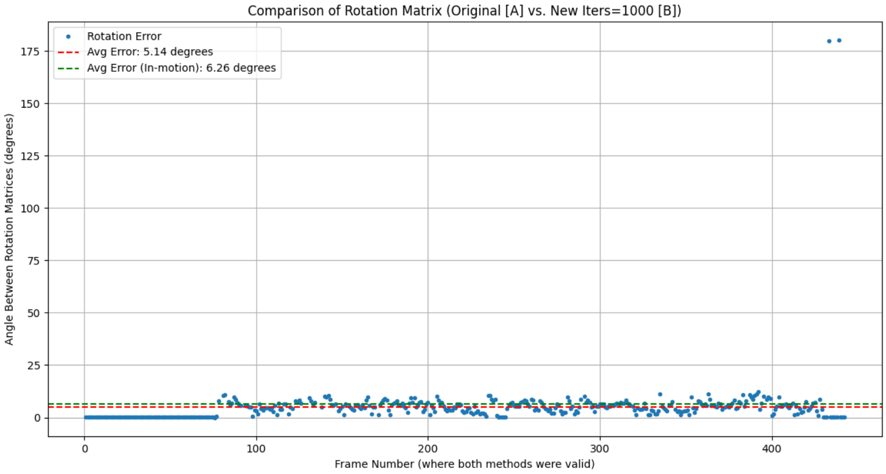
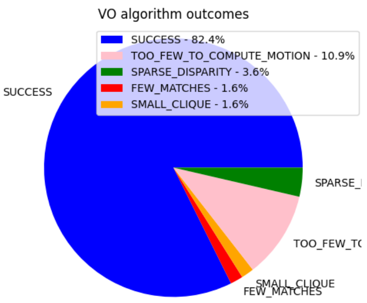


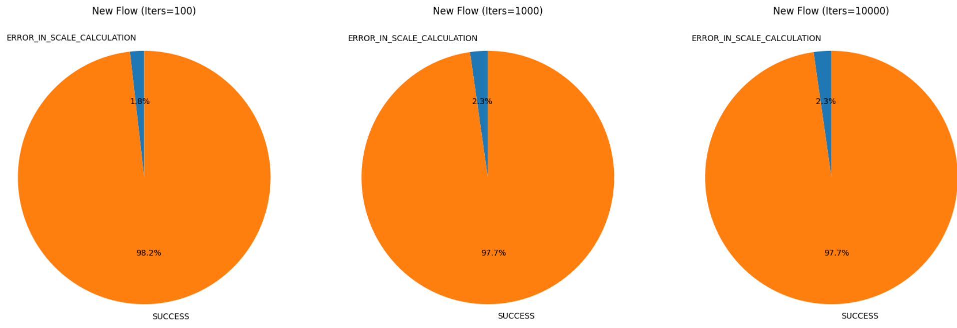
Figure 7: Similar to the graph depicting the difference in angle between the translation vectors, the average was calculated twice for this graph one with the whole dataset and one with the rover in motion.

3 Results and Discussion

3.1 Failure Mode Analysis

As a part of the current flow's testing, a pie chart was generated categorizing the percentage the algorithm ran to success as well as various cases it failed on certain iterations. A similar chart was built for the new flow with categories such as **Error in Scale Calculation, Too Few Points To Compute Motion, RANSAC FAILURE** and the following results were discovered. For validation purposes, similar categories from the old flow such as **SMALL CLIQUE** (even though that is not relevant to the new algorithm) were checked for better comparison.





3.2 Limitations and Future Directions

The comparative analysis demonstrates substantial improvements in feature correspondence counts and the geometric comparisons presented here establish that the proposed method produces motion estimates somewhat consistent with the original approach, but full correctness is not guaranteed. The difference in the angle of translation vectors is particularly an area of concern, and further work is required to test validity of both flows. A step in this direction would be to do further testing on more diverse datasets.

We also discussed how feature detection employs a tile-based strategy to ensure spatial distribution across the image frame. The current implementation divides each frame into a 40×40 grid, extracting features from each tile to prevent clustering. If future analysis reveals that the new depth-free correspondence approach consistently produces more valid matches, the grid size can be altered to 20×20 to lower computation.