TTIC 31040, Spring 2021
Final Project: Monocular and Stereo Visual Odometry

Chris Penny
David Huang
June 3, 2021

Summary: Visual Odometry (VO) is a commonly utilized process for ego-motion estimation in which successive frames are leveraged to derive a transformation of a given camera center.[1] Here, we describe exploratory implementations of both monocular and stereo VO. Data was sourced from the KITTI Odometry Dataset, including video captured from a stereo camera setup mounted on a moving vehicle, GPS-recorded ground truth, and the camera projection matrices for a number of capture sessions ('sequences').[2] In brief, Monocular VO is implemented via estimation of the essential matrix using key-points in successive frames from the same camera. This essential matrix is decomposed into a rotation and translation that describe the transformation of the camera center. Scale is borrowed from ground truth. Stereo VO is implemented using *open3d*'s (*o3d)* global registration[3] and ICP registration[4] routines to derive the translation between two 3D point clouds obtained using the calibrated stereo setup. Results indicate that both methods of VO are reasonably robust over limited frame intervals; however, the compounded nature of errors - particularly those involving translation around street corners - leads to substantial divergence of the course of a full KITTI sequence.

Implementation: Detailed implementation steps are described below.

Data Import: Grayscale KITTI image sequences along with ground truth poses and calibration information were downloaded as publicly available.[5] Once downloaded, the import was conducted using the *pykitti* module specifically written for this purpose.[6]

Monocular VO ('Monocular'): Monocular VO was implemented according to the following procedure. Note that open-cv is used extensively in the mechanics of the implementation, though results have been compared favorably with a by-hand implementation (see Appendix notebooks).

- Calculate SIFT descriptors for the purpose of identifying key points in frames *t* and *t+1* (`cv2.SIFT_create`). From experiments, ORB descriptors appeared to derive far worse trajectory results than SIFT descriptors.
- Find key point matches across frames *t* and *t+1* using KNN and filtering using Lowe's ratio with a cutoff of ~0.8 (`cv2.BFMatcher`).
- Estimate the essential matrix inside of a RANSAC loop (`cv2.findEssentialMat`). Two hyperparameters for this step appeared to have a significant effect on the resulting essential matrix: correctness probability (`prob`) and error tolerance (`threshold`). Increasing `prob` from the default setting of 0.99 to 0.9999 and reducing `threshold` from the default 1.0 to 0.25 ultimately yielded much more stable and consistent results.
- Decompose the essential matrix into relative rotation and translation (`cv2.recoverPose`) across frames *t* and *t+1*. Open-cv's implementation is notable for performing the SVD decomposition while also choosing the solution with the correct sign, i.e. a cheirality check that selects the "most likely" transformation out of four possible transformations that ensures the greatest number of visible points. Translation is estimated up to scale.
- Borrow ground truth scale to apply units to translation. This was done by using the difference of ground-truth translation norms between frames *t* and *t+1* as scaling for the estimated translation.
- Update camera pose with the transformation 'currentPose @ [rotation | translation]'.

Stereo VO: We experimented with and implemented stereo in a few ways. Across all approaches, we rely on the camera intrinsics/extrinsics provided by *pykitti* calibration data - namely intrinsic camera matrix *K* (identical cameras) and the real-world stereo camera baseline of ~half a meter.

- Compute stereo disparity at every frame *t* and *t+1* with block matching (`cv2.StereoSGBM_create`) and use the resulting disparity along a projection matrix derived from camera intrinsic/extrinsics (focal length *f*, image dimensions, and baseline scaling) to produce scaled point clouds (`cv2.reprojectImageTo3D`). We found that the hyperparameters for the disparity estimation have a fairly significant effect on the subsequent steps. Downstream results appeared to be best when we suppressed the noise with larger block sizes and set a minimum disparity cap to a "reasonable" 16 pixels. Point clouds do not include points with the minimum recorded disparity but are taken from each pixel on the depth map, yielding approximately 350 thousand points per cloud.
- With two sets of point clouds corresponding to frames *t* and *t+1*, we proceeded to use open3d's point cloud 'registration' methods to derive alignment and yield estimates on rotation and translation. We experimented with the following approaches of varying degrees of complexity and robustness.
- Naive Monocular Hot Start ('Stereo Hybrid Full'):
  - The *o3d* point-2-point 'ICP Registration' algorithm (`o3d.pipelines.registration.registration_icp(..., ...TransformationEstimationPointToPoint())`) is documented to be a refinement step that generally requires one to provide a 'reasonable' initial alignment transformation between two sets of point clouds. In a simple approach, we deployed the monocular VO strategy outlined above (without scaling) to use the as the initial transformation. We matched large point clouds in their entirety with this strategy, a computationally expensive approach taking about 10-20 seconds per frame. As the initial transformation supplied to the routine was observed to be fairly

[1] https://link.springer.com/article/10.1186/s40064-016-3573-7

[2] See http://www.cvlibs.net/datasets/kitti/eval_odometry.php

[3] See http://www.open3d.org/docs/release/tutorial/pipelines/global_registration.html#RANSAC

[4] See http://www.open3d.org/docs/0.7.0/tutorial/Basic/icp_registration.html#point-to-point-icp

[5] See http://www.cvlibs.net/datasets/kitti/eval_odometry.php

[6] See https://github.com/utiasSTARS/pykitti

accurate (even without scaling applied), a `threshold` parameter of 0.02 was used, serving to restrict the search space for a potential correspondence.
    - The 3D-3D point cloud transformation from ICP registration is then used to update the current pose.
- Global Registration (RANSAC) + Subsampling ('Stereo Global 10k, 20k, and 50k'):
    - 'Global Registration' seeks to produce a rough alignment between point clouds. It adds significant overhead to the VO pipeline as a whole, so subsampling of point clouds was conducted to produce a point cloud with fewer points. Samplings of 10, 20, and 50 thousand points were considered (approx. fraction of total, respectively: 1/35, 2/35, 1/7).
    - In this approach, the *o3d* 'Global Registration' pipeline[7] was used to find a globally optimal point cloud transformation before solving for a locally optimal solution with ICP Registration. The implemented steps of this pipeline are described in brief:
    - Downsampling: Point clouds were subject to voxel downsampling using a 'voxel size' of 0.2 meters using the (`.voxel_down_sample`) method. This fairly substantial downsample reduces runtime significantly. For example, tests with lesser voxel downsampling of size 0.05m, yielded runtimes on the order of ~50 seconds per frame with clouds of 50k points.
    - Normal Estimation and Feature Extraction: Point cloud normals are estimated using a KDTree to determine nearest neighbors within a radius of twice the voxel size. Normals are then used in computation of Fast Point Feature Histogram descriptors, again derived from a KDTree determined nearest neighbors but with a search radius of five times the voxel size.
    - RANSAC-based Global Registration: A RANSAC feature matching routine (`registration.registration_ransac_based_on_feature_matching`) performs global registration to find a rough point cloud transformation that yields an optimal number of inliers while maintaining goodness of fit. The routine was set to consider correspondence based on edge length and distance. A RANSAC convergence criteria of 10 million iterations/confidence of 0.999 was set, with the high number of iterations coming at relatively low computational expense in comparison to feature derivation. To our knowledge, we did not converge based on confidence.
    - The transformation derived from global registration is then used to initialize the ICP registration routine described above. One notable difference exists in the `threshold` parameter, which was increased to 0.1 as the initial transformation here is taken to be a rougher starting point than in the hot start scenario. The output of ICP registration is subsequently used to update pose.

Results: Results were collected across a number of implementation sensitivities on KITTI sequence '00'. Solutions were evaluated using the *evo* library for comparison to ground truth poses and rigidly transformed post-hoc for alignment to account for undefined and arbitrary coordinate frames. It is worth noting that stereo implementations on our local hardware averaged around 5 seconds per frame depending on the number of points sampled for the point clouds. Given sequence '00' contains about 4,500 frames, the runtime costs are evident, and a deep exploration of hyperparameters was sacrificed in favor of an investigation of overall robustness and differentiating factors between monocular and stereo performance. Note that APE values are much larger than those provided for ORB-SLAM. A summary of absolute path error can be found in **Table I**. Plots of *evo*-aligned paths are displayed in **Figures 1-5**.

APE Metrics w/ Alignment (No Scaling)					
	Mono	Stereo Global 10k	Stereo Global 20k	Stereo Global 50k	Stereo Hybrid Full
max	299.9	472.7	407.0	303.5	274.1
mean	91.6	219.6	144.8	105.4	106.5
median	83.6	192.2	122.5	79.6	98.7
min	25.6	60.2	29.0	36.0	15.6
rmse	102.9	241.4	160.1	122.8	120.3
sse	48,037,244.9	264,623,548.3	116,389,242.4	68,522,515.4	65,677,107.1
std	46.7	100.2	68.2	63.0	55.8

Table I: APE Metrics across implementation sensitivities.

Monocular Case: The monocular implementation was intended as a starting point for more complex implementations. Still, monocular VO proves remarkably robust across limited frame sequences, in part by virtue of the relative simplicity of 2D-2D matching as opposed to 3D-3D matching in the stereo case. Although, it is important to remember that the monocular case is scaled with borrowed ground truth poses. Notably, the monocular case exhibits the lowest mean absolute path error (APE) and smallest level of dispersion, suggesting that the 2D-2D transformation is potentially more stable than the 3D-3D transformation used in stereo, though apples-to-apples is difficult as a result of borrowed ground-truth scaling. Additionally, a finer tuning of stereo hyperparameters could potentially reveal different results.

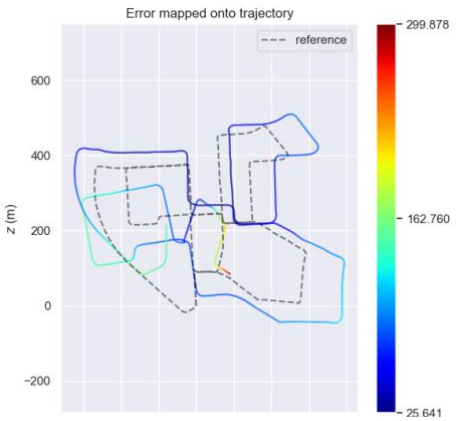


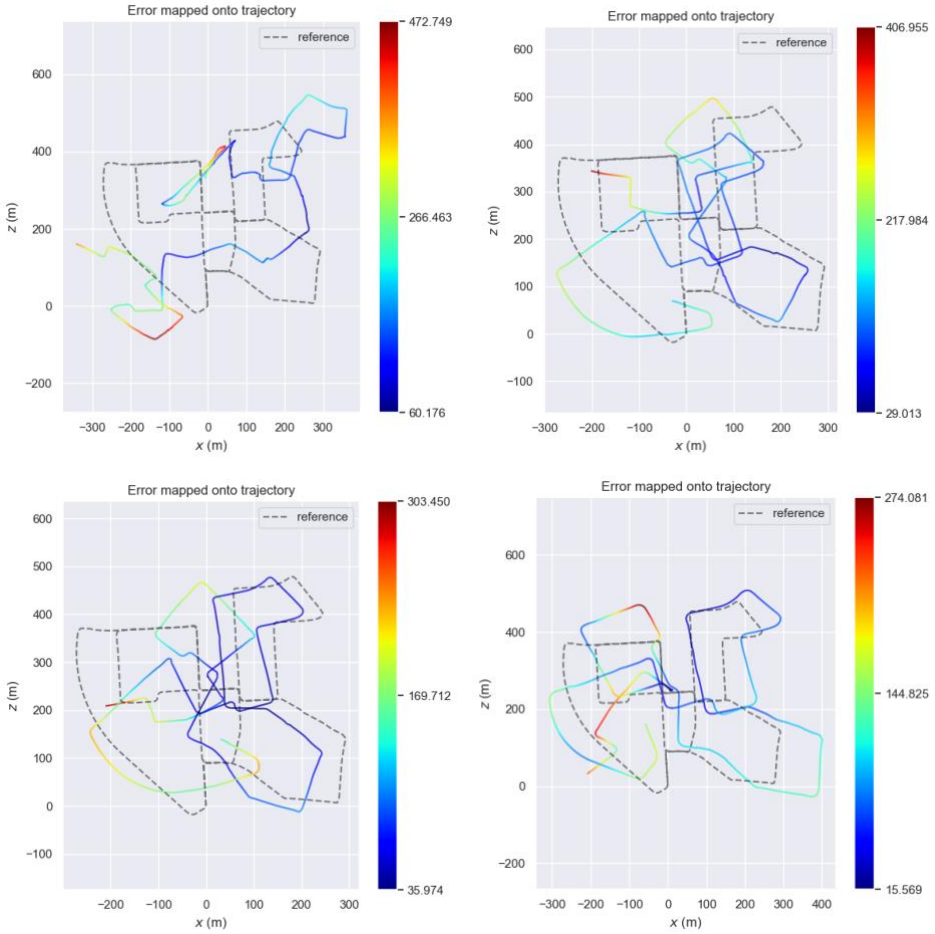
Figure 1: Monocular VO result.

The plot shown to the right (**Figure 1**) bears some consideration. It is clear that, up to a point, the monocular estimation of ego-motion closely mirrors ground truth; however, a significant error is recorded at an intersection roughly 800 frames into the sequence. While the vehicle is paused at the intersection, key points in successive frames are not perfectly static, leading to erroneous estimation of the camera center

[7] http://www.open3d.org/docs/release/tutorial/pipelines/global_registration.html

transformation and a significant deviance from ground truth. Recorded frames compiled in video with estimated motion demonstrate the nature of failure (intersection in question is reached at timestamp 2:13).[8] The successive nature of pose estimation ensures that we carry that error forward, though post-hoc alignment via *evo* makes results significantly easier to interpret and obfuscates errors. Because of the optimization alignment step, we can see the path from start up till the intersection is skewed, such that our path does not start at *(0, 0)* and exhibits the largest deviation from ground-truth.

Stereo Cases: Results for stereo implementation sensitivities, as outlined in **Implementation**, are compiled below. Briefly, we see that the Stereo Hybrid Full sensitivity provides a less desirable set of APE metrics than pure monocular VO but the best metrics among the stereo implementations, despite the hot start transformation being unscaled. The Stereo Global 50k APE set is comparable to Stereo Hybrid Full, with smaller point samplings providing noticeably poorer APE metrics. Given that they arise from totally different methods of transformation estimation, the parity between Stereo Global 50k and Stereo Hybrid Full is notable.



Figures 2 - 5: Evo-aligned paths. (Clockwise) Fig 2.) Stereo Global 10k; Fig 3.) Stereo Global 20k; Fig 4.) Stereo Global 50k; Fig 5.) Stereo Hybrid Full.

Examination of the above plots reveals reasonably consistent performance as described in **Table I**. The Stereo Global 50k and Stereo Hybrid Full capture clearly corresponding paths, albeit with errors around street corners causing substantial deviation from ground truth. Stereo Global 10k performs the worst, and the 20k sensitivity exhibits relatively truer shape with still substantial error. It is finally worth noting, as described in detail below, that the Stereo Global 50k sensitivity makes it through the intersection noted above with a far better estimation than the Stereo Hybrid Full, which falls prey to the same issues as monocular (see **Figure 6** below).

Design Choices: The final implementations derive from a number of design choices that undoubtedly bear relevance on experimental results. In terms of the final monocular approach, a decision was made to lock in at 600 key points, relax the RANSAC threshold, and increase the strictness of the RANSAC convergence criteria. In submitted materials, developmental approaches span the creation of fewer key points, different Lowe's cutoffs, and even the absence of RANSAC entirely. Overall, the submitted parameters and RANSAC approach provide a robust set of results in comparison to other tested settings, but a reasonable ego-motion estimation can be performed at different settings and without RANSAC. In experiments, we briefly compared the performance of SIFT-detected features with ORB-detection; and empirically, we found SIFT to be more robust (although slower).

In the monocular approach, *open-cv*'s method `cv2.recoverPose` for deriving the relative rotation and translation of the second camera to the first. So the resulting rotation and translation appear to be "reversed" with respect to the consecutive frames of *t* and *t+1*. A simple fix here was to simply feed the reversed order of points in both steps of recovering the essential matrix and its

[8] See https://www.youtube.com/watch?v=7iAKHMUh_v8.

decomposition. Alternatively, one could invert the matrices and reverse the translation, but we avoided this alternative due to the risk of accumulating numerical errors from matrix inversions across frames.

Within stereo implementations, there is somewhat more room for flexibility. Careful inspection of the stereo process described in **Implementation** reveals that we elected to diverge slightly from the suggested procedure as given in the assignment. Rather than generate a 3D point cloud for just key points, we made the decision to convert the full depth map into a point cloud (randomly subsampled in global registration implementations) for a given stereo frame pair at time *t*. The motivation for this decision is found in the instability of *o3d* registration methods when using a limited number of key points. Only when using far denser point clouds were we able to calculate reasonable translation, although whether results on two less-dense point clouds constructed from matching key point sets could be correctly transformed with well-selected parameters remains an open question.

Regarding such parameters, when performing ICP registration, the Stereo Hybrid Full case uses a relatively conservative threshold for finding correspondences at 0.02 compared to 0.1 for all global registration methods. As alluded to, this is because we expect the hot start from the monocular case to be a better initial guess for the transformation than the comparatively rough process of global registration - we need more freedom to find correspondences. As for global registration itself, a parameter that proved highly influential in terms of resulting runtime was the degree of voxel downsampling performed when preprocessing the already-subsampled point cloud. Our voxel size parameter of 0.2m is a significant reduction in the number of points - when performing global registration using the downsampled cloud we obtain on the order of 80 correspondences versus up to tens of thousands when performing ICP registration. Without implementation of downsampling, global registration would yield intractable runtimes. Finally, regarding RANSAC parameterization in global registration, it was observed that we reached a maximum number of iterations as opposed to the confidence threshold. Despite this, as feature derivation for the point clouds is so costly (even after subsampling and downsampling), the max iteration limit was set high (10 million) to get the most out of derived features.

Qualitative Performance and Fundamental Limitations: Qualitative evaluation of performance suggests that both pure stereo implementations and the hybrid pipeline possess respective strengths, while also illuminating a number of limitations to the methods as implemented. From the material in **Results**, it can be seen that the Stereo Hybrid Full sensitivity produces a somewhat more coherent path shape overall than Stereo Global 50k, which is further supported by APE metric. This suggests that global registration is subpar to the monocular process in terms of providing ICP registration with a globally optimal area to begin its local optimization; however, this is not the full story. If we evaluate these two stereo implementations on the basis of their ability to successfully traverse the first 1,000 frames of the sequence, as shown in **Figure 6**, we are left with the impression that global registration is more robust to the aforementioned street intersection that causes substantive error in the monocular case. As we avoid key point detection and use the depth map image, that Stereo Global 50k proves more performant in observation of the street intersection is consistent with the approach, suggesting that further improvement may be obtained by following a more sophisticated hybrid approach. Finally, it should be noted that our VO solutions are far in accuracy from ORB-SLAM, as we allow errors to accumulate.

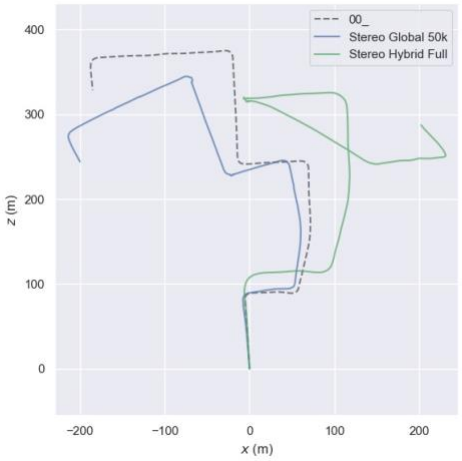


Figure 6: Stereo Global 50k is more robust than Stereo Hybrid Full for the first 1000 frames.

An aspect of the given implementations that is difficult to ignore is slow speed of execution. Both monocular and stereo implementations fall far short of the required target required for realtime VO in terms of time spent processing a given timestep. This area is where the Stereo Hybrid Full implementation is remarkably poor (due to matching the full point cloud in ICP registration), but even the monocular implementation takes at least one second per frame. We observe a relationship between the size of the point cloud in the stereo case and the runtime, but are unable to reduce runtime in this respect given that *o3d*'s library provided stable results with dense clouds only. Other libraries or more intelligent subsampling strategies (e.g. sample points from key patches) could serve to alleviate these issues. A 3D-2D correspondence to estimate the transformation between poses could also prove more efficient.

Another potential improvement that could reduce both noise and computational complexity would involve cataloging features across a small window of frames. In creating a "database" of features, one can more easily detect moments when the camera is at a standstill, e.g. a possible "standstill" detection heuristic: 99+% of features are visible between frames. In "standstill" frames, we won't need to recompute point clouds or subsequent matching, which could subdue the introduction of spurious transformations (rotations and translations) and reduce unnecessary computation. Further, as alluded previously, we did not spend as significant of time tuning the hyperparameters compared to the time spent developing and implementing various visual odometry SLAM strategies. It remains to be explored whether further fine-tuning of hyperparameters (especially for stereo) could boost overall performances. Lastly, despite being relatively quicker, the greater variance in errors from sub-sampling could potentially be reduced via an error-catching strategy; detecting "probably-incorrect" transformations could trigger a re-sampling and recomputation.

Conclusions: As alluded to above, time constraints and significant runtimes in the stereo case enacted limits on our ability to perform an exhaustive parameter search. Nevertheless, the catalogued observations suggest the presented monocular and stereo VO implementations are successful in capturing aspects of the true ego-motion. Accumulated errors lead to notable deviations from ground truth in terms of APE and path profile, but visual odometry can be seen to capture length frame sequences of accurate movement while displaying a remarkable ability to reproduce scale in the stereo case.