Australian National University

SLAM on SLAM: Benchmarking Monocular Systems

Ryan Pike

Honours Thesis

Supervised by

Associate Professor Jochen Trumpf

of the

College of Engineering & Computer Science

Australian National University

Acknowledgements

I would like to thank my primary supervisor Jochen for the illuminating discussions over the year, there were times when I thought I would not be able to complete certain tasks but your reassurance and encouragement made it all possible. Thanks also goes to my secondary supervisor Pieter for sharing information specific to the problem, as well as providing me with valuable feedback. It has been an excellent introduction into the field of research and it was made enjoyable by both of you.

Gratitude also extends to my family for providing me with an environment to study effectively, without distractions and worries. This really did make a significant difference.

Abstract

Simultaneous localisation and mapping (SLAM) is finding its way into the consumer ready market. From the increasing availability of monocular cameras SLAM is an attractive choice to generate accurate representations of the pose and the surrounding environment. These algorithms are being implemented on low-power architectures for large-scale production. For this next stage of delivery there needs to be consistent frameworks that enable effective regulation. Ensuring safety in autonomous driving is just one example where the co-design of SLAM algorithms will be essential.

This report develops a V-stage multi-objective pipeline that transforms a given application of SLAM into an informed decision about which solver to use. Stage I abstracts the application into a condition table that characterises the given application. Stage II provides selection critera for algorithms, datasets and metrics along with the proposed sequence classification matrix (SCM) which admits a partial ordering on features of the dataset for application-based evaluation. Stage III transforms the condition table into a specific protocol that selects which SCM and which metrics to use. Stage IV provides the evaluation guidelines following a robust set of principles intended to delineate cases of over-fitting. In the final stage the results from the benchmark are cross-checked and compared to determine the outcome for solver choice.

The general framework was instantiated through three examples concerning a household consumer robot, unsafe mine exploration and a race track environment. Following the pipeline proposed three recommendations for which solver to use along with conditions on ensuring working performance. The metric for map consistency was also extended from the planar case to the visual SLAM problem with limitations concerning coordinate parameterisation.

Under the SCM characterisation a profound result is the systematic discovery of the following three properties: 1. If loop closures do not occur on large scenes then scale drift will occur. 2. The sliding window optimisation gives a performance boost on scenes with greater rotational velocity and 3. From the sequence ordering scenes with higher average motion characteristics will deteriorate the performance of the solvers. Although these are established in the literature the approach presented created an efficient window into these properties.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Overview

Simultaneous localisation and mapping (SLAM) is an important feature for several sensor modalities in the modern world. In scientific domains it has helped place the rover on mars [71] and given new insights into underwater reef mapping [92]. As we continue to develop the algorithms we find its application instrumental for safe autonomous driving and even applications for cleaning your home [53]. This next stage of consumer delivery will come with it a host of challenges like effective embedding practices, efficient power constraints and most importantly the framework for benchmarking and testing various SLAM algorithms.

To enable a robust future the way in which benchmarking is carried out will be critical. The performance measures, testing conditions and available algorithms will all contribute to the calculus of a decision. A decision that will answer the question *which solver is most applicable for the application at hand*. Currently in the literature researchers tend to err on the side of robustness and take a publish on performance approach. This accepted evaluation procedure is running the risk of over-fitting the sequence and producing results that do not represent the behaviour on similar sequences to the one published. The ineffective co-design between validating the benchmarks and developing the algorithms can cause a vicious cycle where only solvers with better performance are published, as is the case in the computer vision community [72]. In order to restrict this cycle of research we need to look at principled frameworks for benchmarking that if followed makes cherry picking a sequence impermissible. Through a rigid framework the performance of an algorithm will be more representative of real world operation and not an optimised parameter set that scores very well on one sequence.

This report intends to develop a principled framework for benchmarking monocular SLAM. The price, size and availability of these cameras makes it a candid choice for the consumer delivery stage. The developments for the monocular case has also been shown to translate to both stereo and visual-intertial implementations [95]. The systematic approach will have important implications in the regulatory procedures as well as providing greater cohesion among researchers. The algorithms and datasets have been selected to highlight this framework in action, and under this approach it was found that limitations and features of the solvers could be efficiently uncovered.

1.2 Report Structure

The background section §2 develops the problem, notation and language used within the report. The typical formulation is given along with a historical overview §2.1.2 to see how SLAM has developed in a sensor agnostic framework. The monocular case in §2.1.3 is provided to show how the categories for an *online* or *full* solution are adapated for a camera modality. Both the front end §2.2.1 and back end §2.2.2 are provided to illuminate the algorithmic architecture so that the results can be technically explored. The solution §2.3 concerns ways in which we measure the trajectory §2.3.1, the map §2.3.2 and how scene features can describe robustness §2.3.3. The project scope looks at the available datasets §2.4.1 that monocular SLAM can be tested on as well as the limitations and advantages of different sequences. The co-design paradigm §2.4.2 showcases examples where the current evaluation procedure is not ready for the consumer market and lastly the benchmarking section §2.4.3. explores how we benchmark and how the next stage will need frameworks to allow comprehensive evaluation.

The systematic approach §3 section outlines the proposed methodology for fairly benchmarking SLAM systems given the context of the application. The set of principles §3.1 outlined underpin the entire operation. Application and acquisition §3.2 contain the first two stages of the bencmarking pipeline and it summarises the inputs into the benchmark. A selection criteria for metrics and algorithms is also provided, aswell as the condition table for transforming the application into functional requirements. Selection and evaluation §3.3 introduces the sequence classification matrix, how the ordering works and a decision tree for transforming the condition table into a benchmarking protocol. Examples and selection criteria are also provided. This section contains stages 3 and 4 of the pipeline. Finally section §3.4 finalises the process and turns the evaluated results into a useful decision.

The results is split up into two sections. The first section §4.1 concerns the application where a visual odometry solver is used. This section also explores the steps for validating the intended functionality to ensure meaningful results. The SLAM results §4.2 deals with the other two applications and makes recommendations on both trajectory and map performance. In both results section the recommendation is given followed by an exploration of the results based on the rules of thumb provided in §3.3.

Lastly the conclusion section §5 outlines the main results, links the developed framework to the research question and gives its relation in the broader context. Limitations of the approach are also discussed along with recommendations for future work.

1.3 Summary of Contributions

- Developed a V-stage multi-objective framework for fair benchmarking
- Allow a user friendly process which can take clients functional requirements, test a set of solvers and make an informed decision about which solver is best for the chosen application
- Developed a novel way of characterising sequences from both visual and motion features.
- Extended Mazurans work on map consistency to the 3D visual SLAM problem, along with limitations of the approach
- 3 VO Systems Benchmarked (DSO, ORB and SVO)
- 2 SLAM Systems benchmarked
- Following the proposed framework can elucidate profound solver properties: 1. The effect of scale drift on scenes without loop closures. 2. How the sliding window optimisation can give a performance boost on scenes with greater rotational velocity. Although these properties have been established in the literature this approach allows a systematic arrival.

2 Background

The following three subsections give a view of what visual SLAM is, how we solve it and lastly what the solution actually means. The first section *The Problem* outlines its development, from its early infancy in geosciences to the state of the art monocular solutions. Following this is a section on *The Solver* which breaks down the typical pipeline which each SLAM/VO method uses. Lastly, *The Solution* attempts to characterise quality and its tight coupling to context. The section on *Project Scope* situates the benchmarking problem and why there needs to be a careful re-development. This chapter will provide the necessary language and tools to describe the cutting edge solvers we see today, it will also give insight into why SLAM is important in modern robotics and why benchmarking is such a complex but necessary tool in order to continually improve our understanding of the problem and how to effectively solve the problem.

2.1 The Problem

2.1.1 Formulation

Simultaneous localisation and mapping, otherwise known as SLAM is one of the most fundamental challenges in robotics. The problem arises when neither the map nor the pose of the robot is known a priori. Adding to this complexity is an attempt to estimate both states in the face of noise. We use onboard proprioceptive sensors of a robot to predict its position in the environment. This prediction is compared to observations from the exteroceptive sensors in such a way to update its position in the environment and update the environment itself.

> "One may separate the problem of physical realization into two stages: computations of the "best approximation" $\hat{x}(t_1)$ of the state from knowledge of $y(t)$ for $t \leq t_1$ and computation of $u(t_1)$ given $\hat{x}(t_1)$"
>
> – R. E. Kalman, "Contributions to the Theory of Optimal Control," 1960

Kalman can be attributed with the seminal technique for solving SLAM however the solution can be predated to the calculation of planetary orbits. At an abstract level we can break the SLAM problem into two categories [82] which will be explored more deeply in §**??**. The *online SLAM problem* which involves estimating the posterior over a pose in an incremental fashion. This is typically written as;

$$p(x_t, m | z_{1:t}, u_{1:t}) \tag{1}$$

Here z_i denotes the vector of measurements and time i and similarly for u_i representing control inputs. For the case of 3D SLAM, $x_t \in SE(3)$ is the pose at time t. Refer to the definition section when we use the term pose as it can be intepreted in a host of different ways. For our purposes we will always consider the $SE(3)$ transformation in a global reference frame that describes both the orientation and translation of the robot written in the basis of that reference frame. Robotics is a dynamic phenomena and you should think of these transformations as literal re-locations of objects in space. The second type of SLAM problem is known as the *full SLAM problem* where we attempt to determine the full posterior over the entire path $x_{1:t}$, this has also been adapted to batch solvers where instead of the entire path it is simply the path $x_{k:k'}$ where $k' - k > 1$. Both problems are directly related from the following [82]

$$p(x_t, m|z_{1:t}, u_{1:t}) = \int \int \cdots \int p(x_{1:t}, m|z_{1:t}, u_{1:t}) dx_1 dx_2 \dots dx_{t-1} \tag{2}$$

This difference can be observed graphically in Figure 1. x_0 represents the prior on the system which is the initial location and orientation of the robot. This represntation highlights the simultaneous nature of the problem in that the robot acquires a map whilst trying to localise itself in that map. The solution to the full SLAM problem is then given

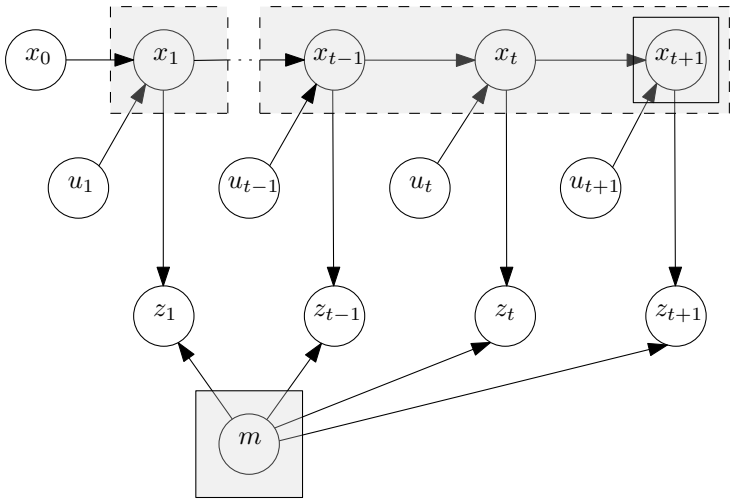


Figure 1: Graphical representation of the SLAM problem. The areas in grey are what we are estimating. For the full SLAM problem we estimate all filled in regions, whilst in the in online problem we only estimate the regions with the solid bounding box.

by;

$$p(x_{1:t}, m_{1:t}|z_{1:t}, u_{1:t}) = \eta p(x_0, m_0) \prod_t p(x_t|x_{t-1}, u_t) \prod_t p(z_t|x_t, m_t) \tag{3}$$

Withe η a normalising factor. From this setup you can recognise the difficulties surrounding the problem. The three discussed will be the high dimensionality in the continuous parameter space, the correspondence problem and the representation and propagation of measurements; Combined they make SLAM an active and challenging research field.

2.1.2 Historical Overview

There is an excellent exposition of the history of the SLAM problem given in [28] however the authors of this paper released some of the seminal works in the field [27], whilst the claim is true that SLAM (in the robotic sense) was born at the 1986 IEEE Robotics and Automation Conference in San Francisco, California. It is also worth mentioning that the problem can be reduced to geographical surveying and under this characterisation it has been around since the calculation of planetary orbits by Gauss (1809). Tools like the method of least squares which is used extensively in modern SLAM is also traced to this period. The difference between both is that issues like data correspondence are easy for a human surveyor but very difficult for a robot. The robotic SLAM - which will be referred to as simply SLAM - problem was formulated in a probabilistic sense in [78, 27], where the authors explain how to deal with uncertainty in the geometry and relative coordinate frames. Around the same time 3-D representations of an environment from a passive sensor were being explored [6]. These descriptions were called *visual maps* which required geometric primitives (here points, lines, and planes), aswell as a characterization of the uncertainty on the parameters of these primitives, caused by noisy measurements. The authors also linearised the measurements to apply the extended kalman filter for constructing the maps. In a similar light the EKF was being used in sonar navigation. These works provided the probabilistic representation of 3D maps, the linearisation of the measurement equation aswell as dealing with uncertainty in coordinate frames to produce the first framework for solving the SLAM problem [79]. It develops the *stochastic map* and formalises how to read and incrementally update spatial arrangements given noisy measurements. It also recognised the correlation between all variables (landmarks and poses) forcing the solution to contain the entire state. With the framework in place the community began analysing and implementing EKF-SLAM on feature based measurements (using artificial beacons) [79], exploring effective data association and proving fundamental convergence results [83]. Occupancy grid mapping, and particle filters were also introduced as alternate solutions [82]. Over the years different sensor modalities were used as measurement inputs for the SLAM problem but it wasn't until the *structure from motion* (SFM) - recovering relative camera poses and three-dimensional (3-D) structure from a set of camera images - was established that saw a camera to be used as a sensor. Considering single cameras can only provide bearing measurements the first bearing-only SLAM was developed. These works provided estimations of features that were invariant to the robot pose, which decoupled the pose and map error. The SLAM problem with camera input forced techniques surrounding place recognition, sensor fusion and dense reconstruction of urban environments (2000 - 2003) [28]. It is important to recognise that SLAM was developed to be sensor agnostic and operate in real-time whilst SFM only deals with cameras as the input and was largely developed to operate offline. A significant linkage between the two fields is visual odometry whose genesis concerned a particular case of SFM but now identifies strongly as

a reduced version of visual SLAM. The first implementation of EKF-based SLAM on a single (monocular) camera was developed by Davison in 2004 [23].

2.1.3 The Monocular Case

SLAM will now refer to the visual SLAM problem where a single camera is used as the only sensor modality. The appeal of this setup is the ongoing reduction in size and price of monocular cameras, making it readily available. It is also a suitable design choice for GPS-denied environments. An effective solution should be able to transform a sequence of images into a trajectory of the cameras optical centre, as well as a representation of the map. The feature-based solver is represented in Figure 2.

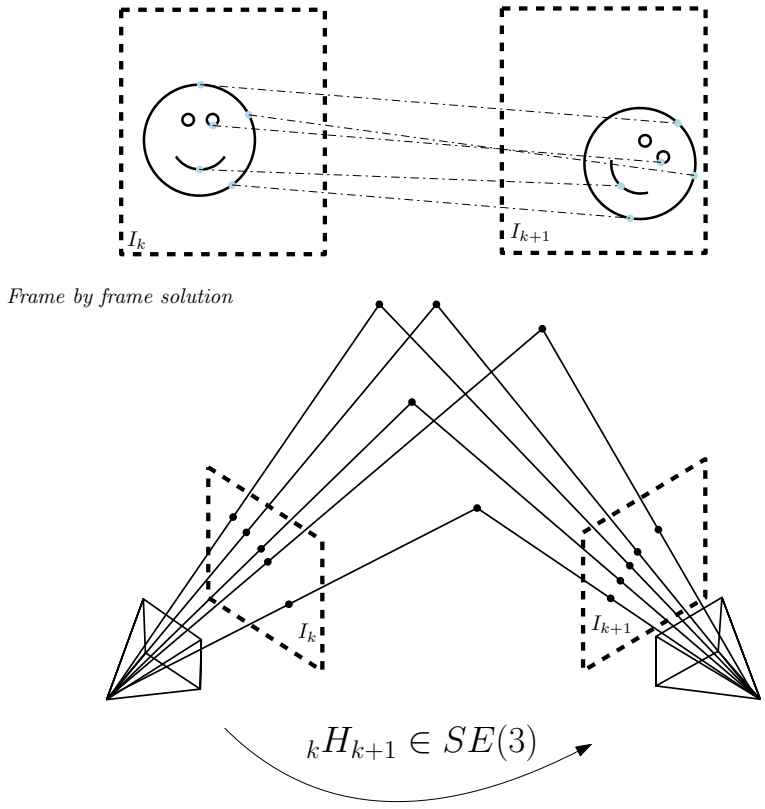


Figure 2: From adjacent image frames I_k, I_{k+1} the SLAM solver attempts to estimate the $SE(3)$ transformation that describes the relative motion of the cameras optical centre, the top planes show the matching of features for correct motion

We can classify these solvers according to §2.1.1. For the *full SLAM problem* the primary idea is bundle adjustment (BA) which minimises a cost function over a large set of variables. Typical cost functions include the reprojection error as used in [49, 67], or for direct methods the photometric error [30, 68]. The cutting edge implementations are LDSO [29], ORB-SLAM2 [67], PTAM [49] and DTAM [68]. Each solver performs an online (BA)

problem at an interactive rate. This is achievable through advanced linear algebra and sparse graph techniques. [24]. DTAM and LSD-SLAM are both direct methods which do not require feature extraction or the corresponding map artifacts. They also tend be more robust to low-texture environments and blur [59]. The photometric consistency limits the baseline making it less robust to sweeping camera motions or swift rotations. This is a limitation of the direct methods compared to the feature based methods like ORB-SLAM2 or PTAM. The dense methods do not take into account mapping performance only the localisation of the robot [30, 68]. Lastly ORB-SLAM2 has been shown to outperform PTAM especially with the number of outliers contained in the map [67]. This is also instructive since ORB-SLAM2 builds on the work of PTAM.

The most common formulation of the *online SLAM problem* is with a state-space model with additive Gaussian noise. This naturally leads to the use of the extended Kalman filter (EKF) since the state-space model is typically non-linear, seminal works from Mono-SLAM represented image patches as landmarks in the map. The EKF provides a recursive estimate of both the pose and landmarks with the computation time being quadratic with respect to the number of landmarks [83]. EKF can be viewed as a generalised Bayes filter and it is one of the most well-known algorithms in the SLAM community. A response to the quadratic dependence of the EKF observer is to use a particle filter. Specifically it was FASTSLAM that is able to provide a cheaper factored solution to the SLAM problem. It has time complexity $O(M \log(K))$ where K is the number of landmarks, and M is the number of particles [83]. Since the development of these algorithms there has been increasing attention towards developing non-linear observers that act on the state-space in a natural way, without the need for linearisation, which has shown to be problematic from poor Jacobian estimates [42], or poor scalability from computational complexity [17]. The non-linear community has placed continued efforts developing models for pose estimation and attitude estimation [89, 64, 14, 88, 90, 8]. Bonnabel et al. showed that the SLAM formulation admits a natural invariance with respect to a reference frame pose change. This allowed an invariant EKF to be designed [13]. This idea seeded the development and analysis of the invariant EKF model [10, 9]. The works of Mahony and Hamel developed a geometric nonlinear observer for SLAM [63]. The authors assert the SLAM state-space as a quotient manifold whose equivalence class is defined by a change of reference frame. This development had several benefits over standard approaches. Most notably taking into account the known invariance, allowing the scene to be dynamic and the natural symmetry admitting robustness. The work following this model is outlined in [39] which continues from the last paper and develops a new symmetry action that is consistent with bearing and range measurements where the prior paper still required linearisation of the output map.

2.2 The Solver

The community has presented several different ways to transform successive image frames into a trajectory and map estimate. From dense-direct to sparse-indirect the way in which solvers function is clearly different, however the overarching pipeline can be separated into a front-end and back-end as is shown in Figure 3. We can break down these solvers into

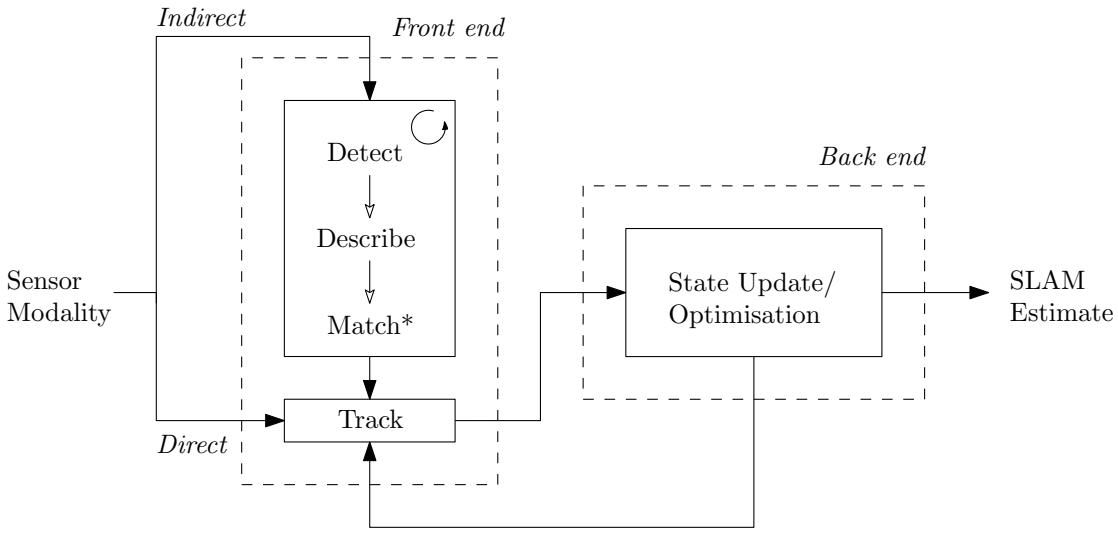


Figure 3: Solvers algorithmic pipeline, the sensor modality in this case is a monocular image and the SLAM estimate is the resulting state estimate.

the categories mentioned in

2.2.1 Front End

Given successive image frames the front end is responsible for data processing and data association. Looking at the flow within the front end we have four primary stages. I_i is fed into the algorihtm and a filtering method is applied to *detect* a set of features from the entire frame. Common methods are FAST, SURF or SIFT [73, 12, 60]. These features are then abstracted into a binary descriptor space such as BRIEF, BRISK or the retina inspired FREAK [18, 55, 3]. Now that the detected features are *described* in their binary space we can attempt to *match* pairs of features from frame I_{i-1}. This is done through comparing two sets of binary descriptors in an attempt to find correspondences between frames. If we did not use binary descriptors the standard way of matching two SIFT features $v_1, v_2 \in \mathbb{R}^3$ would be to compute the euclidean norm;

$$d_E(v_1, v_2) = \|v_1 - v_2\|_2^2 \tag{4}$$

For two BRIEF descriptors $v_1, v_2 \in \{0,1\}^{256}$ we can use the hamming norm which is the number of positions in which bit-vectors differ. It is equivalent to the ℓ_1 norm.

$$d_H(v_1, v_2) = \|v_1 - v_2\|_1 \tag{5}$$

The speed from this method exploits the low level hardware instructions in that only the XOR and sum operator is required. We consider matched points if the hamming distance is lower than a certain threshold. Before moving onto tracking it is very important to understand the different roles matching plays on SLAM systems. Just discussed was short-term data association or feature matching whilst the second long-term data association is known as loop closures and is a necessary feature for place recognition. If at some point in the sequence the camera has visited a previous location, say in frame I_l we have to match feature descriptors to a database of global features which hopefully confirm that I_l is indeed in the same place as I_k with $k << l$. The primary tool for making this robust is DBoW and its updated versions [96]. If the feature matching is stable and successful we consider the solver in a tracking state. The next step concerns operating on matched feature points to arrive at an estimate for both the motion of the camera, as well as the map of the environment. The motion estimate typically employs a constant velocity motion model if tracking has started, or methods like the *eight-point* and *five-point* algorithm will be used. It is possible however to bypass this process and use all the information available. Direct methods aim to estimate camera motion directly from the input images.

2.2.2 Back End

The backend of SLAM is always responsible for providing an estimate of the state. For an incremental solver such as EKF, the backend provides an estimate of the state and no furter optimisation. Batch solvers however will optimise the updated state through a tool known as bundle adjustment (for a modern synthesis see [86]). To explore this properly is out of the scope of the report however there are some well known libraries like GTSAM, g2o and HOGman worth mentioning [24, 51, 40]. A comparison is also shown here [62]. For a high-level description each library uses sparse linear algebra techniques and advanced graph theory to solve a factor graph. It is a reduction from a Bayesian network that describes the entire state of the SLAM problem. When we enforce the assumption that the state-space model of SLAM is modelled via additive Gaussian noise we can solve the factor graph as a non-linear least squares estimate. we can define the motion of the camera as [25];

$$x_i = f_i(x_{i-1}, u_i) + w_i \iff P(x_i|x_{i-1}, u_i) \propto \exp -\frac{1}{2} \|f_i(x_{i-1}, u_i) - x_i\|^2_{\Lambda_i} \tag{6}$$

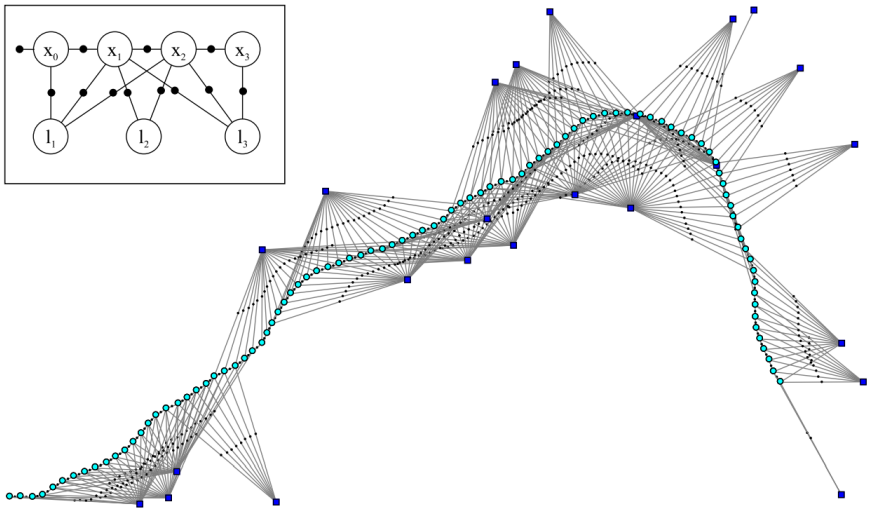


Figure 4: Factor graph for the full SLAM problem. The unknown poses and landmarks correspond to the circular and square variable nodes, respectively, while each measurement corresponds to a factor node (filled black circles) [25]

where f_i is the process model, w_i is normally distributed zero-mean noise with covariance matrix Λ_i. For the measurement model; [25];

$$z_k = h_k(x_{i_k}, l_{j_k}) + v_k \iff P(z_k|x_{i_k}, l_{j_k}) \propto \exp -\frac{1}{2} \|h_k(x_{i_k}, l_{j_k}) - v_k\|^2_{\Sigma_k} \tag{7}$$

where h_k is a measurement equation, v_k and Σ_k are defined as above. Here we are using the notation that $\|e\|^2_\Sigma = e^\top \Sigma^{-1} e$ which is the squared *Mahalanobis distance*. To give two examples, consider an indirect solver applied to a monocular sequence. The Gaussian measuerment noise corresponds to the distance between the back projected feature point and the corresponding feature on the image plane. For a direct solver the gaussian noise is modelled through the difference in intensity of the back projected feature point onto the image plane against the corresponding feature intensity.

2.3 The Solution

The following section will define the metrics used throughout the benchmarking process, aswell as choices on why certain metrics were left out. It will also explain the importance and heuristic motivation of each performance measure. The table outlines the chosen metrics applied to either a SLAM or VO system; Upon initialisation the scale, orientation

	ATE	RPE	Alignment Error	Map Consistency
SLAM metric	Yes	Yes	No	Yes
VO metric	Yes	Yes	Yes	No

Table 1: Metrics considered in the benchmarking process

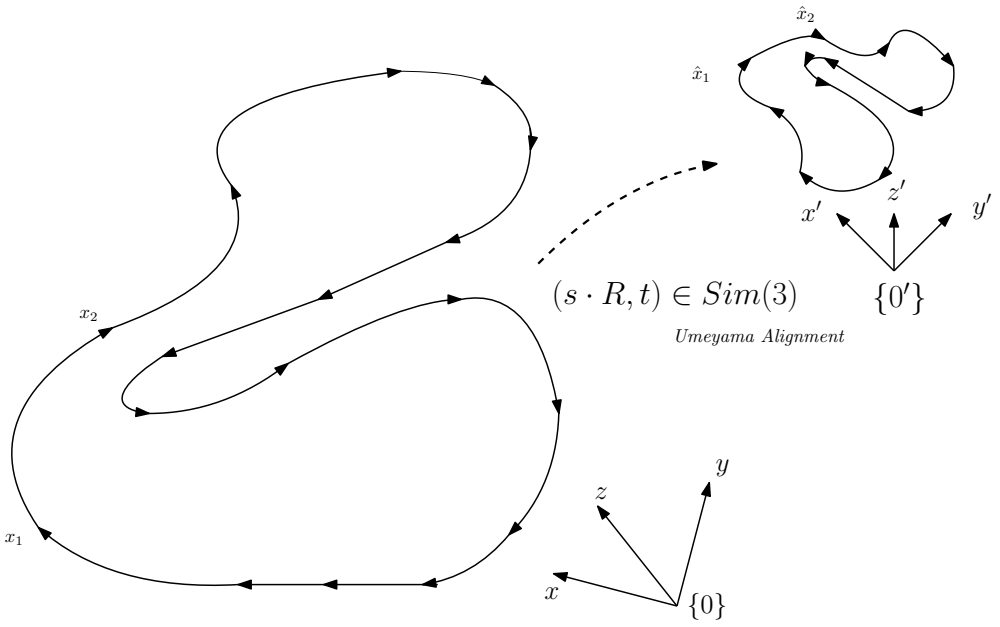


Figure 5: Umeyama method to find the solution to the absolute orientation problem

and translation of the map is set in an arbitrary but consistent manner. This creates solutions where only the shape of the trajectory is the same (See Figure 5). When evaluating the performance of a solution there are a number of pre-processing steps that need to be taken to ensure a solution is of its *best representation*. Most implementations will not store the pose at every time index however ground truth files typically do. This places a restriction on the number of poses that can be compared. We get around this issue by selecting a set of poses that are close in timestamp to the ground truth poses and then apply methods of Umeyama to find a $Sim(3)$ transformation that aligns the solution with the ground-truth [87]. Ideally we want to minimise the euclidean distance between sets of paired points in $\mathbb{R}^3$. The question of pairing poses $x_i \Longleftrightarrow \hat{x}_{i'}$ plays an important role in data association. We have;

$$\min \sum_{i=1}^{k} \left\| \hat{x}_{f(i)} - (sRx_i + t) \right\|_2 \qquad x, \hat{x} \in \mathbb{R}^3 \tag{8}$$

This describes the mean squared error of two point patterns. The solution to this problem is an element of $Sim(3)$ parametrised by [87], for the implementation see Algorithm **??**;

$$H = \begin{bmatrix} s \cdot R & t \\ \mathbf{0}^\top & 1 \end{bmatrix} \qquad R \in SO(3),\ t \in \mathbb{R}^3, s \in \mathbb{R} \tag{9}$$

2.3.1 Localisation Metrics

When evaluating the performance of the trajectory two well established metrics are used within the literature. In both cases we want to optimise the error of the estimated solution with respect to the ground truth which finds the best $S \in Sim(3)$ transformation that minimises the least square error. The absolute trajectory error (ATE) and the relative pose error (RPE). The former can be considered an indicator for global consistency, whilst the latter measures the drift or local performance of the trajectory over a fixed time interval. Given estimated poses in a global reference frame $\{H_1, H_2, \ldots, H_n \in SE(3)\}$ and ground truth poses $\{\hat{H}_1, \hat{H}_2, \ldots, \hat{H}_n \in SE(3)\}$ The ATE F_i at timestep i is [80]:

$$F_i = \hat{H}_i^{-1} S H_i \tag{10}$$

Where S is transforms the estimated poses to the same coordinate frame as the ground truth, such that the distance between points is minimised. The RMSE over all timesteps of the translation component is then computed as;

$$RMSE(F_{1:n}) := \sqrt{\frac{1}{m}\sum_{i=1}^{m} \|trans(F_i)\|^2} \tag{11}$$

For the relative pose error we now look at relative transformations. For the estimated motion $\{{}_0H_1, {}_1H_2, \ldots, {}_{n-1}H_n \in SE(3)\}$, and the groundtruth motion $\{{}_0\hat{H}_1, {}_1\hat{H}_2, \ldots, {}_{n-1}\hat{H}_n \in SE(3)\}$ with a time interval Δ, the RPE E_i at time step i [80]:

$$E_i = \left({}_i\hat{H}_{i+\Delta}\right)^{-1} \times S \times {}_iH_{i+\Delta} \tag{12}$$

For n poses, $m = n - \Delta$ errors are obtained. The RMSE over all time steps of the translation component is then computed:

$$RMSE(E_{1:n}, \Delta) := \sqrt{\frac{1}{m}\sum_{i=1}^{m} \|trans(E_i)\|^2} \tag{13}$$

For certain sequences where external ground truth is not provided it is still possible to get a measure of *drift* as long as the camera begins and ends in the same location, as is done in [31]. In this case the authors have provided the *alignment error* which is evaluated as follows. Let $\{p_1, \ldots, p_n \in \mathbb{R}^3\}$ be the tracked positions of frames 1 to n. Let $S \subset [1; n]$ and $E \subset [1; n]$ be the frame indices for the start and end segments for which the aligned ground truth positions $\hat{p} \in \mathbb{R}^3$ are provided. The first step is aligning the tracked trajectory with the start and end segments independently. This provides two relative transformations;

$$T_s^{gt} := \underset{T \in Sim(3)}{\operatorname{argmin}} \sum_{i \in S} (T p_i - \hat{p}_i)^2 \tag{14}$$

$$T_e^{gt} := \underset{T \in Sim(3)}{\operatorname{argmin}} \sum_{i \in E} (T p_i - \hat{p}_i)^2 \tag{15}$$

We now define the *alignment error*, which is an indicator for the scale, rotation and translation drift over the full sequence;

$$e_{\text{align}} := \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left\| T_s^{gt} p_i - T_e^{gt} p_i \right\|^2} \tag{16}$$

2.3.2 Mapping Metrics

Typical datasets do not provide the ground truth of the environment. Even when one is provided as is done in the ICL-NUIM dataset [41], there is still the issue of finding correspondence between the feature point cloud generated by the algorithm and the given ground truth. We wish to find an affine transformation that minimises two sets of 3D point clouds. The problem formulation is similar to (8) with a slight difference;

$$\min \sum_{i=1}^{k} \left\| \hat{p}_{f(i)} - (R p_i + t) \right\|_2 \qquad p, \hat{p} \in \mathbb{R}^3 \tag{17}$$

Solutions to this problem have been based on the singular value decomposition, quaternions and also iterative methods [5, 33, 57]. This difficult, and often expensive task makes mapping accuracy less important than trajectory error. It is possible however to use mapping metrics that are independent of the ground truth. This direction is still in its infancy however there is work being done on statistical measures for map consistency. Mazuran et al. produced a paper in 2014 that develops a pairwise inconsistency measure between observable viewing cones in poses [66]. The author uses 2D range scans to create observable boundaries. These visibility polygons are then used to compute inconsistency distances. After normalising this measure and computing a matrix that compares all range scans over the entire map the author tests all pairs of scans through an inverse CDF inequality with a user-set confidence. If all tests are succeeded we consider the map *globally consistent*. In this paper we aim to extend the 2D polygon boundary to a 3D convex hull and see the resulting outcome. This method employs an approximate time complexity of $\mathcal{O}(N^2 + K^2)$ where N is the number of camera poses and K is the size of the point cloud seen from that camera pose. The logarithm dependency of computing the convex hull and finding the closest distance to the plane has been omitted since it is dominated by large N and K. The inconsistency distance is measured as follows;

$$d_i(p_j^k) = \begin{cases} dist(p_j^k) & if p_j^k \in \mathcal{V}_i \\ 0 & \text{Otherwise} \end{cases} \tag{18}$$

Here $\mathcal{V}_i$ is the convex hull generated from the point cloud seen from frame i as well as the optical centre of the camera. p_j^k are the boundary points of $\mathcal{V}_j$. We define the inconsistency measure to be;

$$M_{ij} = \sum_k d_i(p_j^k) + \sum_{k'} d_j(p_i^k) \tag{19}$$

Given N poses we can generate the following $N \times N$ matrix;

$$\mathbf{\Psi} = \left[\frac{M_{ij} - \mu_i n_{ij}}{\sigma_i \sqrt{n_{ij}}} \right]_{ij} \tag{20}$$

Here n_{ij} is the number of inconsistent points in frame i with respect to frame j which will make entries in (20) zero if the viewing cones do not overlap. μ_i is the mean distance from the optical centre of the camera to the landmark, and σ_i is the associated variance. Suppose $F^{-1}(p)$ is the inverse CDF of the normal distribution, we conclude consistency from frame i to frame j if the following inequality holds;

$$\mathbf{\Psi}_{ij} \leq F^{-1}(1-\alpha) \tag{21}$$

For *global consistency* we model the outcome of a pairwise hypothesis test over a Bernoulli-distributed random variable parametrised by α. Since (21) has a type I error probability of $1-(1-\alpha)^r$ we have to compute the maximum number $\hat{\xi}$ of tests that can fail for a confidence level $1-\alpha'$ as;

$$\hat{\xi} = \min_{0 \leq \xi \leq r} \left\{ \xi \middle| \sum_{i=\xi+1}^{r} \binom{r}{i} \alpha^i (1-\alpha)^{r-i} \leq \alpha' \right\} \tag{22}$$

Computing $\hat{\xi}$ is numerically unstable and instead the author proposes [66];

$$\binom{r}{i} \alpha^i (1-\alpha)^{r-i}$$

$$= \exp\left(i \log \alpha + (r-i)\log(1-\alpha) + \log\Gamma(r+1) - \log\Gamma(i+1) - \log\Gamma(r-i+1)\right) \tag{23}$$

Where $\Gamma(\cdot)$ is the gamma function. This allows the computation of a cascaded hypothesis test. We first perform all pairwise hypothesis tests. Then, if the number of failed tests is smaller than $\hat{\xi}$, the overall consistency test is positive.

2.3.3 Characterising Robustness

The metrics mentioned give a sense of performance for an individual sequence. Taking this a level higher we wish to understand how the performance changes based on properties of the sequence. These can be broken down into **intrinsic** (what is inherent to the visual feed), **extrinsic** (elements that give the sequence context) and **event-based** (specific actions in the sequence) conditions. The proposed features are outlined as follows;

Intrinsic	**Description**
Readout method	Whether the camera uses rolling or global shutter
Sequence resolution	The output resolution of the video feed
Light exposure	Image plane illuminance times the exposure time (determined by the aperture)
Extrinsic	
Motion	The linear and translational velocity of the camera
Environment	Whether the scene is indoors, outdoors or a combination of both
Texture	Low texture or repeating patterns
Event-based	
Occlusion	Whether an object you are tracking is hidden by another object
Dynamic objects	Whether the scene has dynamic components
Motion blur	Apparent streaking of moving objects

Table 2: Different properties in a monocular sequence

All of these properties have direct ramifications on the performance of the solver. When the linear and angular displacement increases the constant velocity motion model begins to break down, if there are dynamic objects the back projection function will incorrectly align with the image plane. If the readout method is rolling shutter unless explicitly modelled, direct solvers will suffer. Low textured environment can create the *aperture problem* which causes degenerate directions for tracking. These properties are responsible for the different solvers in the community. The very difficulty produces different techniques and solver types. Occlusion and rolling shutter are depicted in Figure 6.

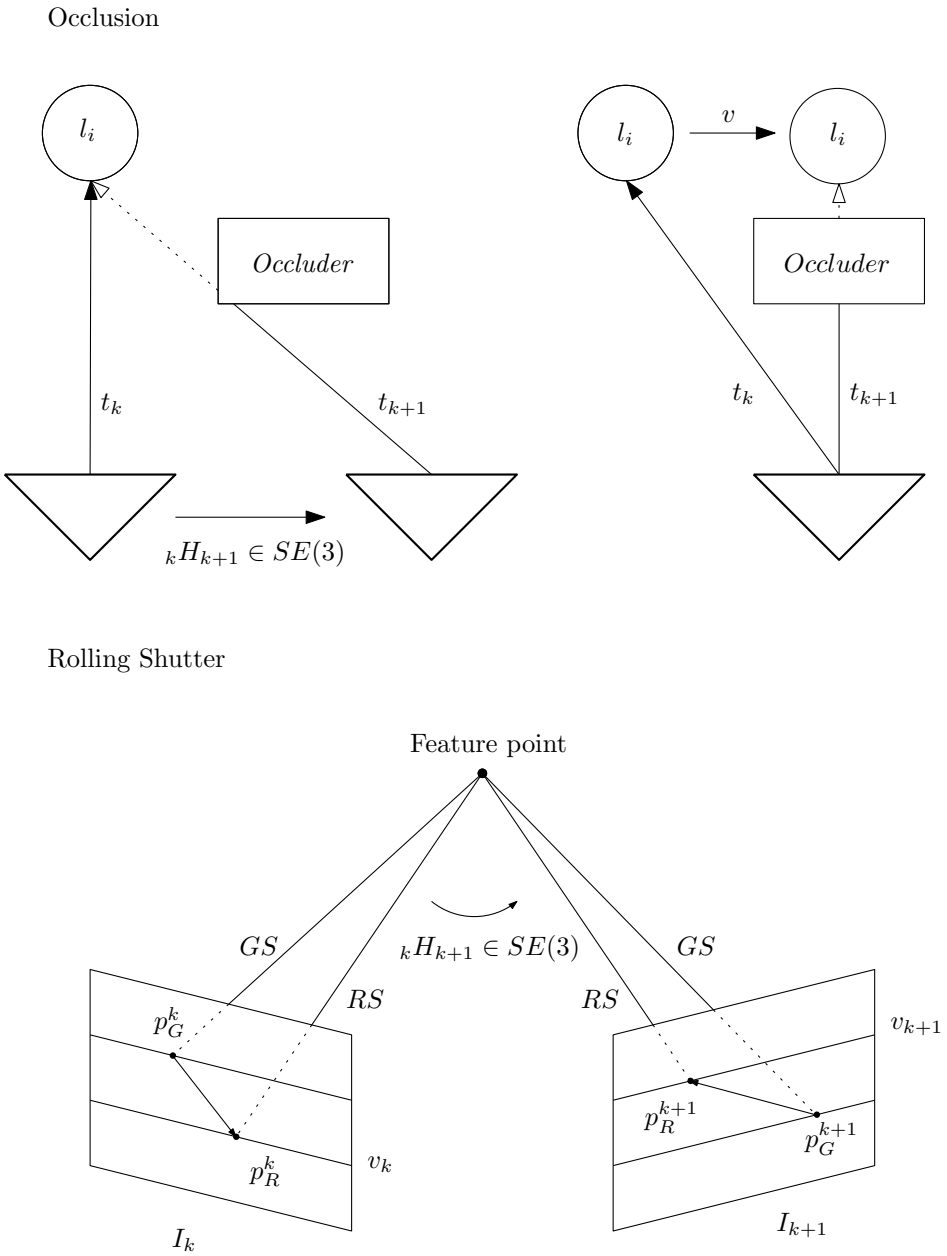


Figure 6: Two scenarios where occlusion (top-left) and dynamic objects (top-right) disrupt the line of tracking. (bottom) The effects of rolling shutter tend to represent features on the image plane in a distorted manner, often shifting the location. This is because the sensors on an RS camera occur sequentially when capturing an image.

The brief introduction to the visual SLAM problem provides the necessary language to explore the recent directions in the community. The understanding of the solution and how we characterise performance and robustness can give context to the benchmarking tools that have been developed.

2.4 Project Scope

This section looks at the rapid development of the datasets in the community as well as the benchmarking methods developed, it will highlight a particular case where overfitting has been presented as performance and hopefully illuminate the absence of principles surrounding benchmarking.

2.4.1 Testing Environments

Datasets provide a means of evaluating a solution. Typically, the dataset will contain a reference file or groundtruth that will show the performance of your solution. They provide necessary feedback and can highlight certain characteristics of your approach. From a literature search there are 310 sequences available to test your SLAM algorithm [95] (as of May 2019). The vast collection contains real world datasets focusing on mapping [69, 19, 61], localisation [47, 20, 91] and odometry [80, 31, 46], although It was Kitti's urban driving experience [38], The EuRoC drone dataset [16] and the VI collection from TUM [77] that championed VO/VI SLAM. Underwater reef mapping [92], mine exploration [85] and low-cost consumer robotics [53] are some examples of how SLAM is being deployed. The datasets had to respond by increasing the available testing conditions. Subterranean environments including mines in Chile [54] and underwater datasets [34, 65] that use a submersed UAV to navigate. The turbidity and light refraction alone constitutes brand new conditions, which was explored in [65]. Further difficult visual conditions are explored through dense fog [22] and the presence of smoke and dust [70]. With the former motivating solvers that could operate in smoke environments [50, 2]. With respect to micro aerial vehicles the UZH-FPV drone [44] sequence was released in October 2019 with indoor flight speeds of 23.9 [m/s]. The sustained improvement at both a hardware and software level permits solutions to this highly aggressive environment.

InteriorNet[56] and ICL-NUIM[41] are recent examples of simulated datasets. ICL-NUIM contains two indoor scenes. Kerl et al. extended the dataset to model a rolling shutter camera [48]. In the low textured scene DSO managed to outperform ORB2 highlighting that a low textured scene could be more damaging to a feature based solver than rolling shutter is to a direct solver [94]. Investigating the effects of rolling shutter is one of the major challenges in monocular VO [94] and this was made possible through simulated environment. There is also vehicular scenes in the VKITTI suite. These datasets are attractive due to environment customisation and perfect groundtruths however it is still not the real world. Early experiments have been conducted to understand consistency between real and simulated environments [7], which concludes similar tracking and mapping results for a slow moving indoor ground robot. It has been confirmed that the difference between simulated and real conditions does not affect scale estimation in monocular SLAM [74].

Recent work more akin to this report investigates methods of characterising scene difficulty. Ye et al. uses a decision tree on five variables like duration and motion and classifies difficulty between *easy, medium and hard*. All inputs and response variables have at most 3 different values [95]. The difficulty labels were either provided by the original dataset or determined from the reported tracking outcomes using DSO and ORB2. The second order *Wasserstein* metric has been employed to characterise difficulty in a continuous sense [75]. The metric captures motion, structure and appearance qualities and managed to show a positive relationship between the metric and drift (RPE). This was tested on an EKF-solver on both ICL-NUIM and the TUM-RGBD dataset.

2.4.2 Co-Design Paradigm

The co-design paradigm is intended to develop solutions from the involvement of all stakeholders. This approach will be necessary for the next stage of consumer delivery however the current landscape in SLAM will not accept this framework if the accepted practise of evaluation continues. Technologists in monocular SLAM are developing novel and elegant approaches to tackle the problem. From direct image alignment to double windowed optimization the field is being postured by advanced knowledge and rigorous theory. To gauge this acceleration, the community agrees on common indicators that are used to inform on performance. If an algorithm or method produces results with promising indicators then the community can determine whether further investigation is required. This approach gives a sense of cohesion among researchers in that potentially promising directions can be observed by all. From the various assumptions in different approaches - brightness constancy assumption in direct methods and constant velocity motion in feature methods - comes different benchmarks to support and explore those assumptions. The ICL-NUIM dataset has sequences with generated rolling and motion blur [41]. The TUMmono dataset is photometrically calibrated for improved performance on direct solvers [31]. It is okay to provide conditions where a solver excels but it is approaching a point where algorithms are developed and datasets are carefully selected to showcase performance. This is creating a divide between the presented indicators and actual performance. A classic example is in the $ETH : MH01$ dataset in Figure 24, where the authors for DSO start the solver near the 50 second mark to remove the portion of the scene where the MAV is static (20 - 40 [s]) [29]. In direct methods a constant scene can cause indeterminate convergence of the homography matrix, and with this portion of the scene removed the solver has a reduction of the RMSE ATE of almost 50%. Another example of favoured performance is the TUMmono dataset. The photometric calibration, whilst improving performance for direct solvers actually damages the performance of feature based solvers. It increases the spectrum of intensity which causes dark scenes to become darker and this can cause tracking to fail. ORB2 for instance lost tracking on 3 scenes in [94] where without calibration it was steady. It also increased the alignment error e_{align} on all scenes.

What we see here is alterations, and over parametrisations to fit the data to the solver and unfortunately this forces solvers to not be robust in conditions that are similar to the results that are published. There is a whole range of benchmarks available online, with an extensive list of metrics and evaluation tools however there are no guidelines or principles in how to use them. There needs to be a focus on delineating the process to avoid over parametrisation. This will provide a better, more honest representation of the performance. This is not a new idea even for the robotics community; Researchers should all uphold a baseline of experimental conduct, and for the robotic mapping problem (2002-2007) [81] reproducibility rarely occured [4], despite the standards in place to reduce this [45]. In response, a paper was published that proposed several recommendations, two of which are directly applicable in this context; *The behavior of the mapping system for different values of the parameters should be shown*, and *experiments in which the mapping system does not perform well should be shown.*[4] The first principle provides insight into robustness whilst the latter promotes a deeper understanding from other researchers.

2.4.3 Benchmarking For Consumer Delivery

There are three activities that achieve the benchmarking of systems [52]. In predefined problems where robots compete against each other (i.e. navigating through a desert [84] or FPV racing [44]). The comparison of performance indicators on publicly available datasets, and related publications that introduce scoring metrics on different methods. Focusing on the comparison of solutions a set of performance indicators is benchmarked across a range of solvers, or a benchmark is presented that represents how the solution will be deployed. For performance only there is a comparison of several ROS-based SLAM solutions in an indoor environment [43]. The solvers use stereo, monocular, or depth sensor modalities. A comprehensive comparison of monocular SLAM algorithms uses several publicly available datasets [71], with the addition of testing the algorithms on 8 new datasets. This is due to the often unattainable performance given in the original SLAM paper [71]. There is an extensive evaluation on the EuRoC dataset with strictly VIO solutions [26]. Whilst the first two comparisons investigated error metrics under different sensors, scenes and solvers this paper wanted to recognise time and memory constraints for different hardware implementations. Which found that accuracy and robustness is very dependent on the implementation. A response to this is to use low powered, highly efficient architectures like FPGA. Early works explored the system architecture required for 3D reconstruction [37] followed by feature extraction with ORB features [32]. These works provided the groundwork for a parallel implementation of FastSLAM2.0 on an GPGPU/FPGA [1]. Due to the increasing applications of low power consumer electronics [53] efforts need to be taken to ensure effective embedding practises. The compilation and tuning techniques for reaching the consumer stage are explored here [76]. One aspect of realising this future is the requirement of having frameworks that allow effective and rapid benchmarking.

Comparing algorithms can be a difficult task with the different software architectures of each solution. the diversity of software interfaces for the different datasets and algorithms complicates comprehensive evaluation. This makes bench marking multiple solvers, with multiple metrics a drawn out task. In response researchers from Kings College London and NWPU have developed SLAMBench3.0 [15] and GSLAM[97] respectively. It provides a library of common algorithms used in the SLAM pipeline along with an API to develop your solution. Multiple datasets, algorithms and metrics can be comprehensively evaluated on these systems however there is no framework surrounding the benchmarking process. A response to this, and the main contribution of this report is the development of a principled approach at comparing SLAM algorithms.

3 A Systematic Approach

A note on terminology

It will be very important to clarify the terminology for this framework. For the proposed benchmark we will use the following: The *pipeline* is the general procedure that one takes from transforming an application to an informed decision (see Figure 7). The *parameters* will refer to the internal changes in the pipeline, such as which SCM to use or which metrics to evaluate. When these parameters are selected as in §3.3.2, it will be considered a *protocol* for that application. Concerning solver types; visual-intertial navigation is SLAM: VIN, and VO can be considered a reduced SLAM system, in which the loop closure (or place recognition) module is disabled. This claim was made in [17]

In order to benchmark different SLAM solvers it useful to look from a top down view at what benchmarking is for. There are several papers that explain the datasets, equipment used and accompanying metrics yet there is a lack of clarity about how to transform the results into a useful result. This chapter aims to look at the principles that will be followed for each experiment which will allow the experimenter to make an informed decision on;

$$\textit{Which solver is most applicable for the chosen application?} \tag{24}$$

In order to come to this decision a framework will be developed such that, when followed you can make that justified choice. The principles developed along the way consider the connection between the solver, the scene and the solution so that a holistic decision can be made. The first part summarises the datasets visual and technical properties which will allow a rigid classification. This will then lead into a specific protocol which will be used for evaluation. This will provide a pathway to answer the question in (24), which as a flow diagram is;

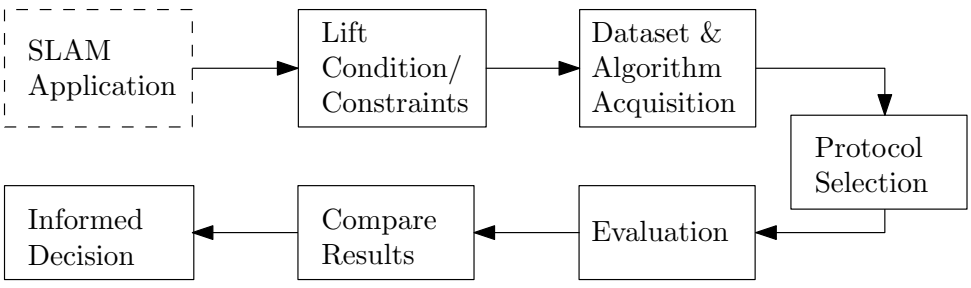


Figure 7: Proposed benchmarking pipeline that will be used in the report

3.1 Principles

It is integral that we consider the context as much as the SLAM approach. There is no merit to the community or the decision in making a benchmark that tests all datasets as there is no SLAM algorithm that will be able to perform best on all. This balance between creating a universal structure and allowing for the individual application makes forming a set of principles difficult. The following set, with justifications will be followed throughout the benchmarking pipeline.

1. No internal changes will be made to the solvers except for logging purposes.

Reason This paper provides a general framework and treats each SLAM implementation as a black box.

2. The same quality of information will be used when running the simulation, the different types are described by;

   2.1 Camera intrinsics

   2.2 Standard image formatting (Zip or png is typical depending on the solver)

   2.3 For photometrically calibrated datasets only scenes with the vignette, transfer function and exposure provided will be used.

Reason To ensure a fair comparison for the given solvers.

3. The following attributes are required for each application

   3.1 The readout method (global shutter (GS) or rolling shutter (RS))

   3.2 The type of scene (indoor vs outdoor vs mixture)

   3.3 The types of motion that the camera will be exposed to.

   3.4 Scale of the scene

   3.5 Tolerance for error on the trajectory

   3.6 Loop closure capability

   3.7 Whether the map is required, and whether it is required to be consistent.

Reason These requirements can be applied to each monocular application yet the specificity still allows reasonable context.

4. The same computer will handle each simulation which will operate with no background tasks.

Reason Processing speed and performance can affect the estimated solution. We attempt to recreate equivalent working conditions.

5. The sequence will be fed into the solver from the beginning frame with no adjustments made for any solver.

Reason To ensure each solver experiences the same visual conditions.

6. No parameter optimisation will take place between sequences, however for different datasets the parameters will be changed to the authors predefined parameters.

Reason This restriction is to avoid over parameterizing a sequence and attaining unrealistic working performance for the application. It also shows uniform conditions for each dataset.

3.2 Application and Acquisition

3.2.1 Condition Table

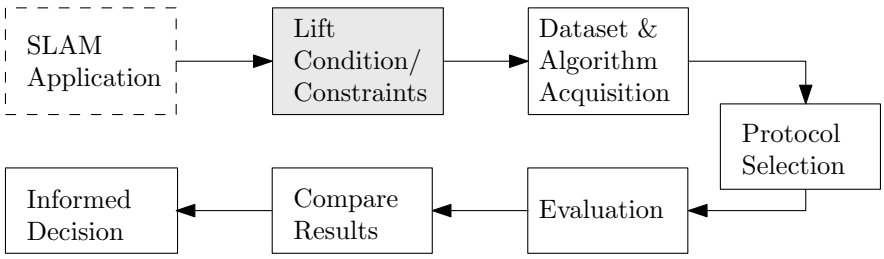


Figure 8: Pipeline Stage I

In order to use the benchmark the applications requirements have to be lifted to the following table.

Condition/Constraint	**Description**	**Range**
Scene type	Indoor, outdoor or mixed	-
Scale	Roughly order of magnitude of metric trajectory	10m - 10km
Motion type	Linear, rotational or the full spectrum of motion	-
Readout method	Global or rolling shutter (based on CCD or CMOS)	GS/RS
Trajectory tolerance	How close you require the trajectory to be to the ground truth	1cm - 10m
Loop closures	place recognition in the solver	Yes/No
Map consistency	Does the map generation need to be consistent	Yes/No

Table 3: Condition table for each SLAM application, cells in bold will be used for protocol selection

Loop closures, readout method and map consistency determine which SCM and metrics to use. The remaining fields act as clearances, that if broken will not permit the use of that specific solver. The trajectory tolerance is how close you want the estimated solution to be the ground truth solution. When evaluating the SCM for either ATE or RPE it gives a RMSE value which is an average representation of performance. If this value gets within a small distance to the tolerance (See Table 4) then that sequence should be investigated further, by plotting the entire ATE.

Scale	**Closeness of RMSE**
10m	1%
100m	2.5%
1km	2.5%
10km	5%

Table 4: Closeness values to determine whether further investigation is required. These examples are given for context, in reality the tolerances would be based on the application.

3.2.2 Datasets and SLAM Algorithms

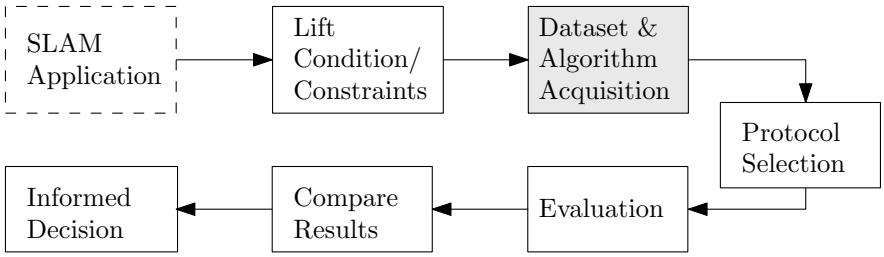


Figure 9: Pipeline Stage II

Selection Criteria for Each Dataset

This criteria is motivated from §2.3.3.

1. Range of trajectory scales

Reason In order to understand how solvers perform for varying scale. This will highlight the performance against scalability.

2. Different camera dynamics (linear vs angular velocity)

Reason Often certain solvers have an edge over linear velocity or rotational velocity and determining these conclusions will assist in deciding on a solver for that application

3. Different readout methods

Reason Direct and indirect solver assumptions can breakdown depending on the readout method. This is to ensure certain solvers are not subjected to a readout method that will result in failure.

4. Allow different metrics to be evaluated based on the data available in the ground truth

Reason Having multiple metrics provide a characterisation of performance and not an isolated indicator.

5. Range of environments

Reason Understanding the solvers performance on different environments can determine whether it can be used for the given SLAM application.

6. Short and long sequence duration

Reason To further understand scalability with respect to time; Note: This is different to trajectory scale since a robot could be moving at a high speed over a large trajectory however temporally the scene could be short (i.e. KIT04).

7. Different motion modalities

Reason Although motion modalities still give rise to different motion variables the characteristics of the motion will be very different (i.e. fixed camera in a car vs. camera from a drone)

The following table is a brief summary of the four datasets. It includes properties of each dataset along with the metrics that can be evaluated. The selection criteria is evident.

Dataset	ETH	KIT	TUM	TUR
# Sequences	10	11	50	19
Scale*	10m	1km	100m	10m
Robot	Drone	Car	Hand	Ground/Hand
Environment	Indoor	Outdoor	Indoor/Outdoor	Indoor
Readout Method	Global	Global	Global	Rolling
Metrics measurable from ground truth provided				
ATE	Yes	Yes	No	Yes
RPE	Yes	Yes	No	Yes
Alignment Error	No	No	Yes	No
Map Consistency	Yes	Yes	Yes	Yes
Map Accuracy	No	No	No	No

*Table 5: Datasets (*Approximate length of the metric trajectory)*

ETH: EuRoC MAV

The EuRoC mav dataset was released in 2016 by researchers from ETH zurich. In this report it will be referred to as ETH. It consists of 10 sequences taken on the The AscTec "Firefly" hex-rotor helicopter (see Figure 10). The datasets contain a stereo camera as well as an IMU for acceleration readings. The authors split the sequences into two main environments, the vicon room and the machine hall with difficulties ranging from easy to difficult. The vicon room has full 6DOF pose capture whilst the machine hall sequences use an MS50 Leica for position capture. For the motion characteristics the slowest and fastest average linear velocity occurs in $ETH : V201$ and $ETH : MH03$ at speeds of $0.33[m/s]$ and $0.99[m/s]$. The highest average roational velocity (ARV) occurs in $ETH : V203$ at $0.66[rad/s]$. The average length of the trajectory is $81.2[m]$ with the longest trajectory at $130.9[m]$ for $ETH : MH03$. These sequences span on average 2 minutes.

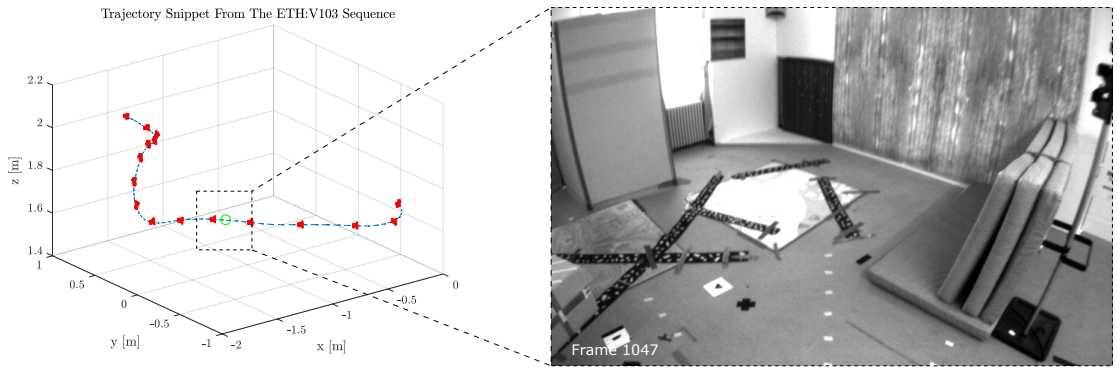


Figure 10: (Left) Trajectory snippet from the ETHV103 sequence. (Right) Still frame from the mav in motion

KIT: Visual Odometry

The kitti vision benchmark suite is a joint project of Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago. The odometry dataset (KIT) was released in 2012 that uses an automobile along 21 different sequences. The rig contains a stereo camera, with a velodyne HDL-64E laserscanner. Sequences range anywhere from 1km to 10km in suburban areas. Only the first 11 sequences will be considered as they contain pose information whilst the remaining are used for validation. For the motion characteristics the slowest and fastest average linear velocity occurs in $KIT07$ and $KIT04$ at speeds of $6.07[m/s]$ and $140[m/s]$. The highest average roational velocity (ARV) occurs in $KIT06$ at $0.13[rad/s]$. The average length of the trajectory is $2016[m]$ with the longest trajectory at $3724[m]$ for $KIT00$. These sequences span on average 3.5 minutes.

Figure 11: (left) Snippet from the second Kitti sequence, (right) still frame showing the cameras input

TUR: RGB-D SLAM

Using both a handheld camera as well as a pioneer rig the TUM research team capture several indoor environments from a kinect sensor (TUR). Each sequence contains the full 6DOF pose estimate with vicon equipment. The dataset does contains colour and depth estimates for each frame as well as accelerometer data. The sequences come with and without loop closures. For the motion characteristics the slowest and fastest average linear velocity occurs in $TUR2rpy$ and $TUR1desk2$ at speeds of $0.01[m/s]$ and $0.43[m/s]$. The highest average rotational velocity occurs in $TUR1360$ at $0.73[rad/s]$. The average length of the trajectory is $14.5[m]$ with the longest trajectory at $43.08[m]$ for $TUR2PS3$. These sequences span on average 3.5 minutes.

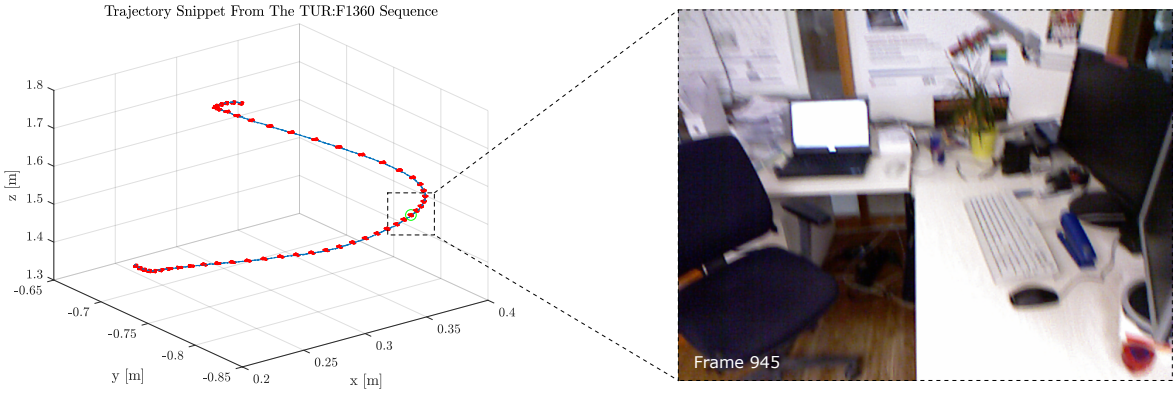


Figure 12: (left) Snippet from the TUM-RGBDFr1360 sequence, showing part of the loop. (right) Still frame from the trajectory show

TUM: Monocular Visual Odometry

Brightness variations due to vignette, gamma correction and exposure time can be eliminated by a complete photometric calibration [29].

$$I(x) = G(tV(x)B(x)) \tag{25}$$

where the measured brightness I depends on the irradiance B, the vignette V, the exposure time t and the camera response function G (gamma function). G and V can be calibrated beforehand, t can be read out from the camera. The research team from the Technische Universität München (TUM) provide 50 photometrically calibrated sequences for the evalutation of visual odometry (TUM). The dataset is shot with two separate cameras (see Figure 13) that traverse several parts of the campus. Each sequence has the same initial and terminating spot which is why it is used to measure drift from VO solutions. Of the 50 sequences only 2 (TUM50, 34) contain both indoor and outdoor environments.

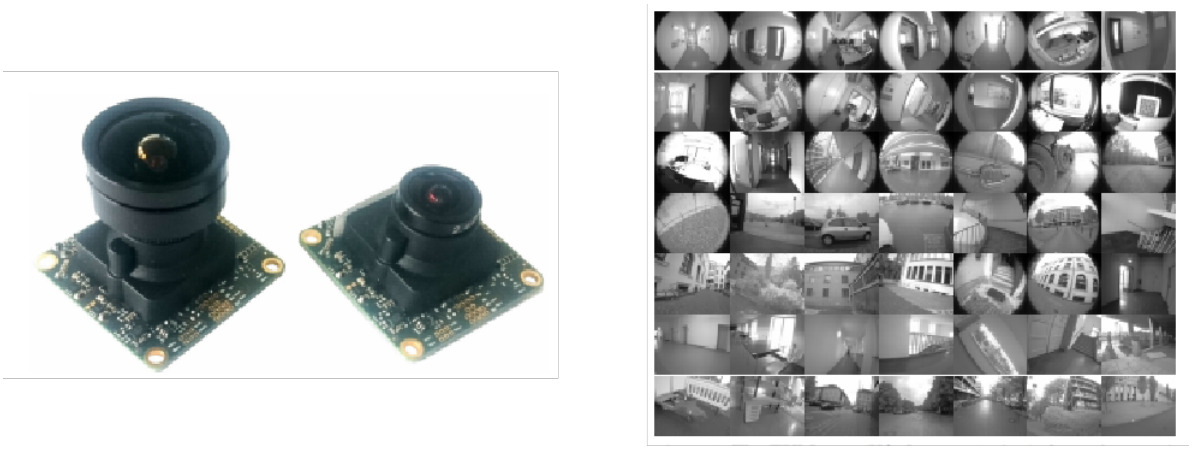


Figure 13: (left) narrow and wide lens for the sequences. (right) snippett of all 50 sequnces used in the making of the TUMmono dataset

SLAM Algorithms

The following is a list of the implementations used in the benchmarking pipeline;

ORB-SLAM2: [**67**] feature-based monocular SLAM system that operates in real time, in small and large, indoor and outdoor environments. The system is robust to severe motion clutter, allows wide baseline loop closing and relocalization, and includes full automatic initialization. Building on the works of PTAM from Klein and Murray ORB-SLAM2 (ORB2) was developed by Raul Mur-Artal. It has low latent interactive bundle adjustment as well as a strict outlier rejection protocol. These together permit long operation and elimination bad data association [67].

DSO: [**29**] A direct sparse visual odometry formulation. Developed by Engel, et al. It combines a fully direct probabilistic model (minimizing a photometric error) with consistent, joint optimization of all model parameters, including geometry – represented as inverse depth in a reference frame – and camera motion. This is achieved in real time by omitting the smoothness prior used in other direct methods and instead sampling pixels evenly throughout the images.

SVO: [**35**]A semi-direct monocular visual odometry algorithm. The approach eliminates the need of costly feature extraction and robust matching techniques for motion estimation. Our algorithm operates directly on pixel intensities, which results in subpixel precision at high frame-rates. A probabilistic mapping method that explicitly models outlier measurements is used to estimate 3D points, which results in fewer outliers and more reliable points [21].

LDSO: [**36**] an extension of Direct Sparse Odometry (DSO) to a monocular visual SLAM system with loop closure detection and pose-graph optimization (LDSO). Loop closure candidates are verified geometrically and Sim(3) relative pose constraints are estimated by jointly minimizing 2D and 3D geometric error terms. These constraints are fused with a co-visibility graph of relative poses extracted from DSO's sliding window optimization [29].

3.3 Selection and Evaluation

3.3.1 Sequence Classification Matrix

From using a criteria for dataset collection we require a way of ordering them in a sensible way. The proposed idea is to form a sequence classification matrix (SCM) that categorises sequences based on a choice of *quantitative* variables. The construction of the SCM must satisfy the following;

1. The SCM must be a square matrix.
2. Atleast one common metric should be measurable within the groundtruth to enable evaluation on the entire matrix.
3. There has to be a pairwise ordering between the matrix cells along the direction of the variable

It would be valuable if the variables chosen would highlight performance and robustness of an algorithm. The variables selected is the average linear velocity, average rotational velocity, proportion of time spent indoors/outdoors and duration of the sequence. These are adopted from the selection critera in 3.2.2 with the benefits of being simple yet informative and can easily be applied to the datasets considered. Three examples SCM's are shown in Figure 15.

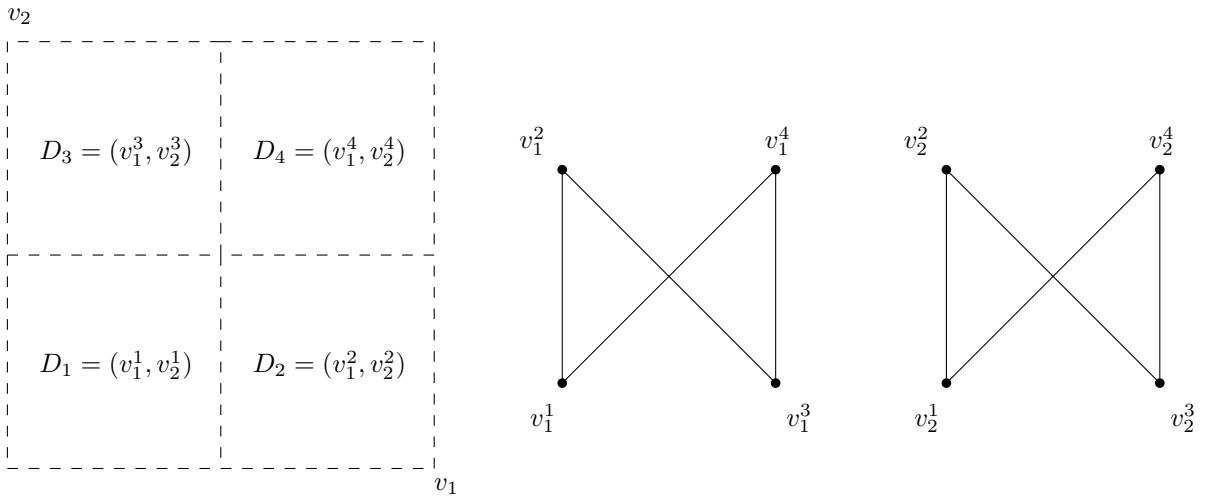


Figure 14: Construction of an SCM $n = 2$, with a Hasse diagram (using $\leq$) to show the partial ordering of datasets in SCM.

You can patch four basic SCM's to form a coordinate style SCM. Each quadrant uses the ordering proposed except the operator switches to $\geq$ below the x-axis, the arrows shown in the example indicate the ordering.

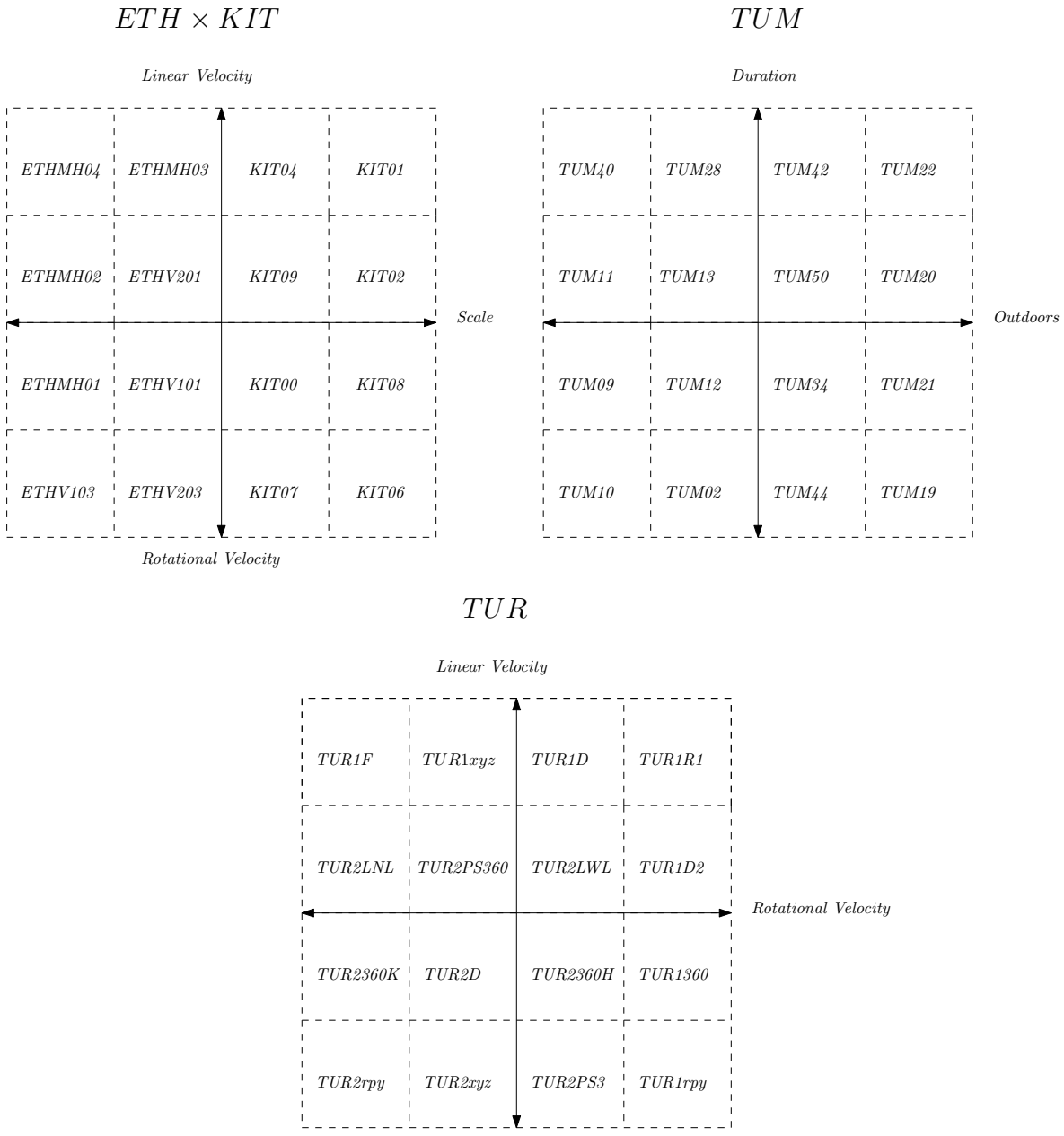


Figure 15: Selected variables and sequences that give three SCM's to be used in the benchmarking process

From the rules provided it is possible to build the matrix by selection however this can be tedious. For this report the variables of interest were plotted centred around the unit square. You then select sequences that best fall into the 16 bins on the SCM ensuring proper ordering. Generating datasets of different size comes with a different experience. A 2×2 SCM may be necessary if less datasets are provided, the only issue with using a smaller SCM is a weaker characterisation over the chosen variables. In the other direction you could have $n = 5$ which would give you 25 sequences to evaluate. If the choice of variables only have small changes between cells then a more refined characterisation would take place. A condensed version of each SCM (See Figure 15), along with metrics and chosen variables is shown in Table 6.

SCM	$ETH \times KIT$	TUR	TUM
Variables	Scale vs. motion	Motion	Environment vs. duration
SLAM/VO	Both	Both	VO
ATE	Yes	Yes	No
RPE	Yes	Yes	No
Drift	No	No	Yes
Map Consistency	Yes	Yes	Yes

Table 6: The variables and metrics for the three proposed sequence classification matrices

Issues noticed when generating the SCM is the inhomogeneous distribution of sequences when plotted. If benchmarks were motivated by a characterisation such as the SCM, it could create a smoother pool of sequences to choose from, which would provide a uniform performance evaluation. For example, the TUM dataset only contains two sequences (34, and 50) that go both indoors and outdoors. The rest of the sequences are either one or the other. For the $ETH \times KIT$ SCM the four sequences in the fourth quadrant do not highlight significant differences in increasing rotational velocity. This is because it is difficult to increase rotational velocity when metric scale increases. A way around this would be to have circular trajectories of constant radii, like in KIT18, however this sequence is only used for validation and does not provide a ground truth.

3.3.2 Selecting The Protocol

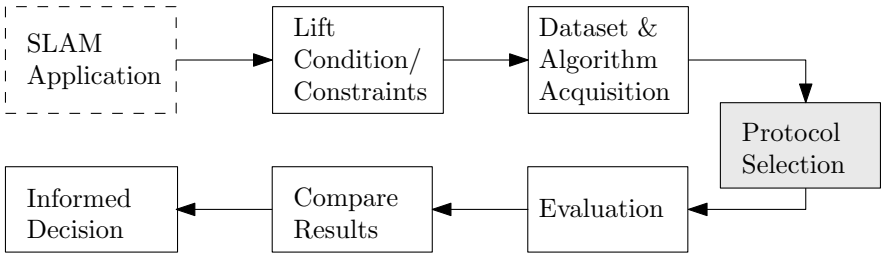


Figure 16: Pipeline Stage III

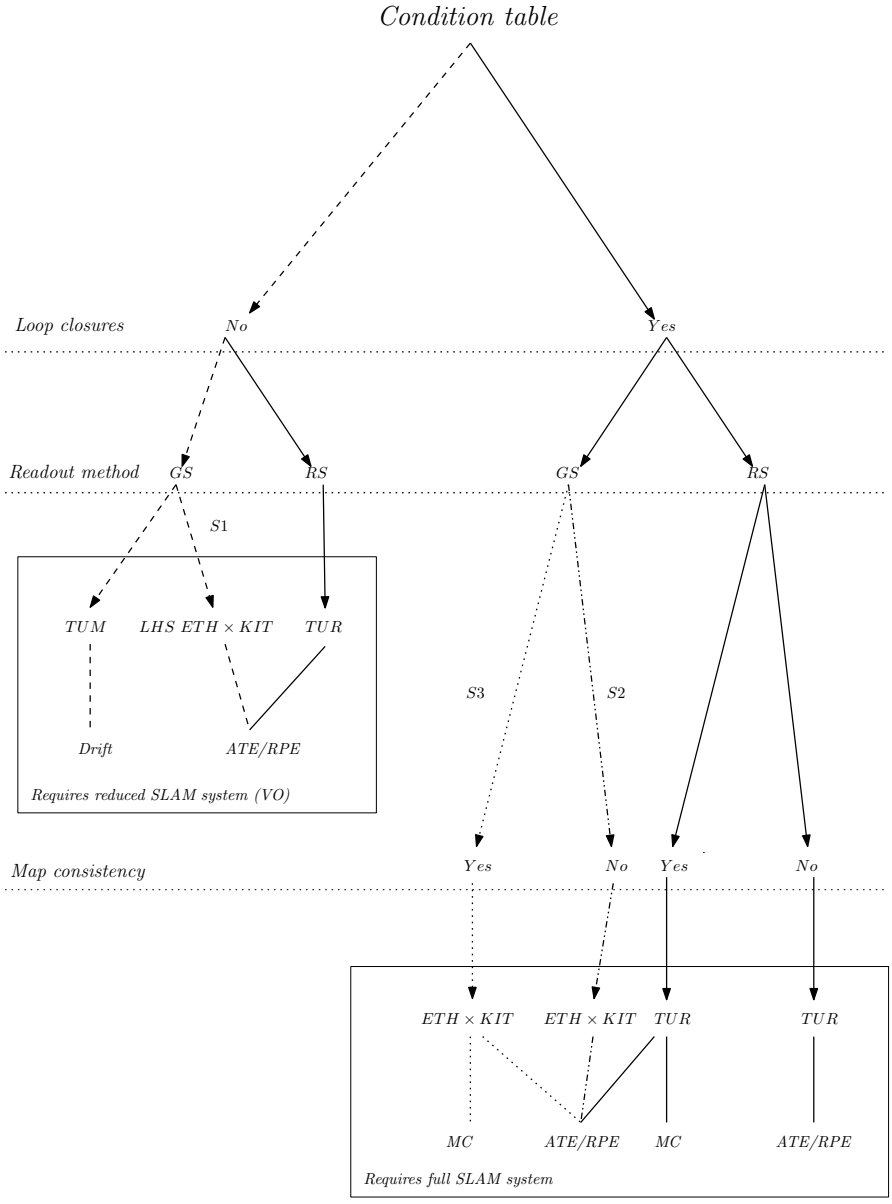


Figure 17: Three stage decision tree for selecting the benchmarking protocol. Highlighted are the three applications (S1,S2 and S3) explored in §4, aswell as bounding boxes for the different versions of SLAM algorithms most suitable for that application

3.3.3 Evaluation and Alignment

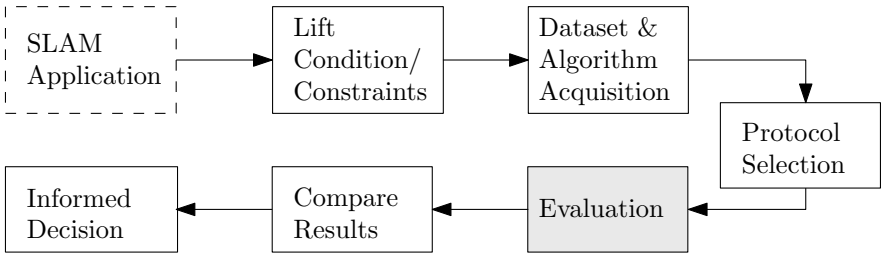


Figure 18: Pipeline Stage IV

After the algorithm[s] have been selected, the solvers parameters can be changed if there is a change of dataset. Within a dataset all solver parameters will remain fixed. For evaluation refer to the principles section in §3.1. Specifically points 2, 4, 5 and 6. The following setup was used when benchmarking.

System Configuration

Category	**System Specification**
Operating System	Ubuntu 18.04.3 LTS
CPU	Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz
VM Configuration	1 physical processor; 2 cores; 2 threads
RAM	12192200 KiB
Motherboard	1.2/VirtualBox (Oracle Corporation)
Graphics	Intel(R) HD Graphics 620

Table 7: System specifications when benchmarking

Methods of Umeyama

The following pseudo algorithm was used on all trajectories (except TUM) in order to evaluate the metrics described in §**??**. As mentioned in §2.3 the algorithm determines $(s \cdot R, t) \in Sim(3)$ that minimises the following;

$$\min \sum_{i=1}^{k} \left\| \hat{x}_{f(i)} - (sRx_i + t) \right\|_2 \qquad x, \hat{x} \in \mathbb{R}^3 \tag{26}$$

Where x and $\hat{x}$ are the estimated and groundtruth positions of the camera respectively. The algorithm assumes both input vectors are ordered through correspondence.

Algorithm 1: Umeyama Alignment to obtain $Sim(3)$ parameters [87]

Input: $\{x_i\}_{i=1}^n, \{y_i\}_{i=1}^n, x_i, y_i \in \mathbb{R}^3$

Two sets of paired 3D points

Output: $(s \cdot R, t) \in Sim(3)$ $s, t \in \mathbb{R}$ $R \in SO(3)$

/* Variable initialization */

$\mu_x \leftarrow \frac{1}{n}\sum_{i=1}^n x_i$ /* mean vector of {x} */

$\mu_y \leftarrow \frac{1}{n}\sum_{i=1}^n y_i$ /* mean vector of {y} */

$\sigma_x^2 \leftarrow \frac{1}{n}\sum_{i=1}^n \|x_i - \mu_x\|^2$ /* variance around mean {x} */

$\sigma_y^2 \leftarrow \frac{1}{n}\sum_{i=1}^n \|y_i - \mu_y\|^2$ /* variance around mean {y} */

$\Sigma_{xy} \leftarrow \frac{1}{n}\sum_{i=1}^n (y_i - \mu_y)(x_i - \mu_x)^\top$ /* Covariance matrix */

/* Singular value decomposition of the covariance matrix */

$UDV^\top \leftarrow \mathrm{SVD}(\Sigma_{xy})$

if $\mathrm{rank}(\Sigma_{xy}) > m - 1$ **then**

  if $\det(\Sigma_{xy}) \geq 0$ **then**

    $S \leftarrow I$

    $(R, t, s) \leftarrow \left(USV^\top, \mu_y - c \cdot \mu_x, \sigma_x^{-2} \cdot \mathrm{Tr}(DS)\right)$

    return (R, t, s)

  else if $\det(\Sigma_{xy}) < 0$ **then**

    $S \leftarrow \mathrm{diag}(1, 1, \ldots, 1, -1)$

    $(R, t, s) \leftarrow \left(USV^\top, \mu_y - c \cdot \mu_x, \sigma_x^{-2} \cdot \mathrm{Tr}(DS)\right)$

    return (R, t, s)

  end

else if $\mathrm{rank}(\Sigma_{xy}) = m - 1$ **then**

  if $\det(U) \cdot \det(V) = 1$ **then**

    $S \leftarrow I$

    $(R, t, s) \leftarrow \left(USV^\top, \mu_y - c \cdot \mu_x, \sigma_x^{-2} \cdot \mathrm{Tr}(DS)\right)$

    return (R, t, s)

  else if $\det(U) \cdot \det(V) = -1$ **then**

    $S \leftarrow \mathrm{diag}(1, 1, \ldots, 1, -1)$

    $(R, t, s) \leftarrow \left(USV^\top, \mu_y - c \cdot \mu_x, \sigma_x^{-2} \cdot \mathrm{Tr}(DS)\right)$

    return (R, t, s)

  end

end

3.4 Making an Informed Decision

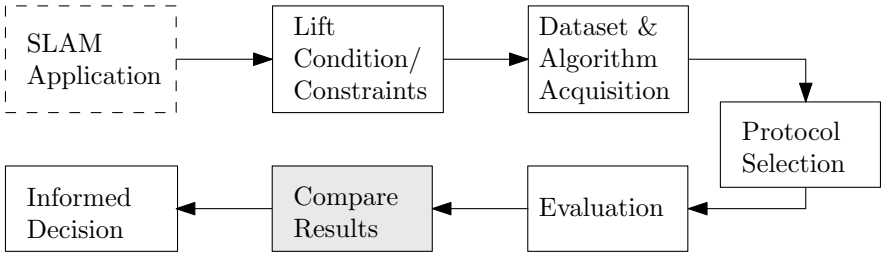


Figure 19: Pipeline Stage V

After lifting the requirements to a condition table you select the protocol according to the decision tree in Figure 17. From there you evaluate and align the recommended sequence classification matrices for the chosen metrics. The matrix then identifies whether there is a clear choice for the given application. It is also necessary to check the RMSE ATE values against the trajectory tolerance. If this value falls within a threshold (See Table 4) of the tolerance then that sequence should be investigated further showing confidence intervals for the trajectory error.

Rules of Thumb

1. Explore the entire ATE and RPE result for the best and worst solution
2. Investigate sequences that have significant performance differences
3. Select sequences on the boundary of the SCM that is relevant to your application

A benefit from using this classification approach is that you can quickly identify good and bad performance, and it acts as a window into which sequences should be analysed further. In this benchmarking process some of the more complex visualisations, and interesting results are explored in Figure 32 and 33. These sequences were chosen for further analysis when following the rules of thumb provided. The main objective in this pipeline is to provide an honest characterisation of performance against variables of interest to allow an informed decision.

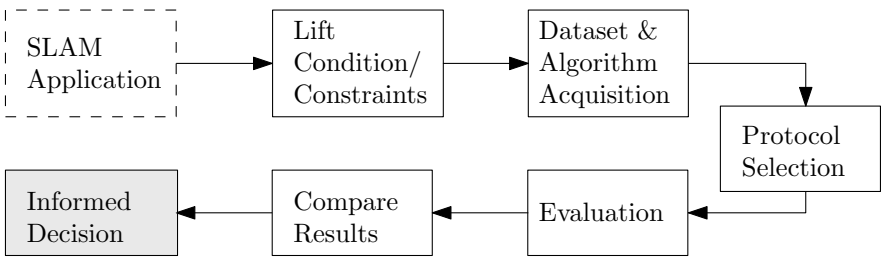


Figure 20: Informed Decision

4 Benchmarking Results

Consider three separate applications where visual SLAM is a viable solution. Following the principles layed out in §3.1 we will take a methodocial approach in making justified recommendations to each scenario. The following cases are;

(S1) A light-weight indoor application for flying a drone with an attached gimbal. The user is interested in installing small rings and obstacles within the environment so that he/she can look at the trajectory later and see whether any improvements in the flight controls can be made.

(S2) A Mining company would like to trial a rover in exploring underground tunnels. These are outdated due to the width and height violating current regulations regarding structural integrity and size (2m by 2m). The tunnels do not contain loops and were used for human operation in the 1970's. The company would like to get an accurate sense of the trajectory with a consistent map generation in low light conditions.

(S3) A professional go-kart firm is looking at finding optimal routes on racetracks. It would like to explore the different path characteristics to discover improved trajectories on the track. In order to understand driver fatigue over time it should be able to perform global loop closures over several rounds on the track. The track also has GPS-denied zones.

Lifting these constraints into abstract functional requirements produces the following tolerance table; Following the principles and methodology developed in §3 we will arrive

Application Parameters	S1	S2	S3
Loop closures	No	Yes	Yes
Readout method	GS	GS	GS
Map consistency	No	No	Yes
Scene	Indoor	Indoor	Mixed
Scale	10m	100m	10km
Motion type	Full spectrum	Linear	Full spectrum
Trajectory Tolerance	0.5m	1m	25m
Recommendation*	**ORB2 (VO)**	**ORB2**	**ORB2**

Table 8: Condition table for three different applications. (the recommendations do have some caveats)

at the recommendations stated in Table 8.

Following Figure 17 we identify three separate benchmarking protocols. For S1 we will be evaluating drift on TUM and ATE/RPE on a submatrix of $ETH \times KIT$. From the decision tree the solution admits a reduced VO system. For S2 we will be testing ATE/RPE on the $ETH \times KIT$ SCM. For S3 we will be testing ATE/RPE as well as map consistency on $ETH \times KIT$.

4.1 Visual Odometry Results

S1 does not require loop closure capability. The reduced VO solvers that will be used are ORB-SLAM2 (VO), DSO and SVO. Looking at the scale of the setup it would not be informative to evaluate each solver on the entire SCM. We will select a submatrix consisting of all sequences on the left hand side (LHS) of the SCM which is shown in 27. A shaded cell indicates no test is performed.

4.1.1 Solver Validation

Both ORB2 (VO) and DSO have a frame acceptance protocol which makes the RPE defined in (12) not suitable, as a time step Δ for both solvers will be different lengths. This can be verified from the pose acquisition graph in Figure 25. For this version of SVO it only attains tracking for 3 out of the 8 scenes so we will observe the performance of all three solvers on sequences where tracking is unanimous. This is done for validation purposes before we begin the decision process. For the *MH_01_easy* sequence we have the following trajectory path

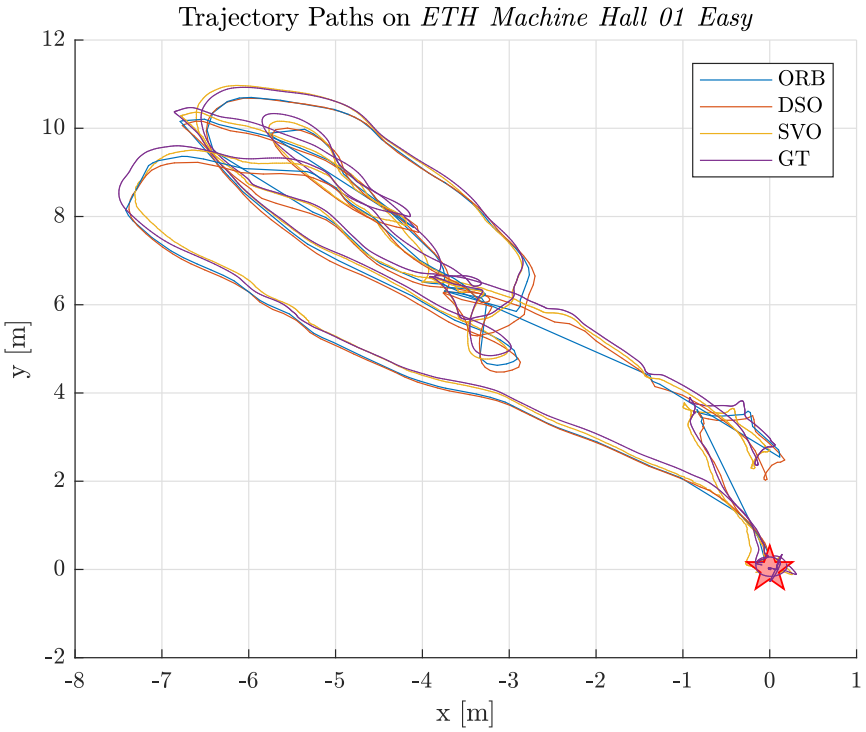


Figure 21: Trajectories in x-y plane for ETHMH01 from all three solvers. Groundtruth is also plotted (the star indicates start and finish point)

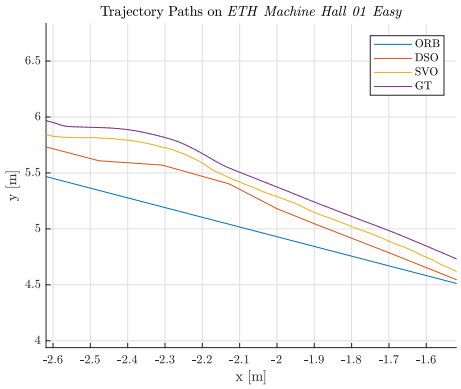


Figure 22: Close-up of linear motion of ETHMH01 on all three solvers

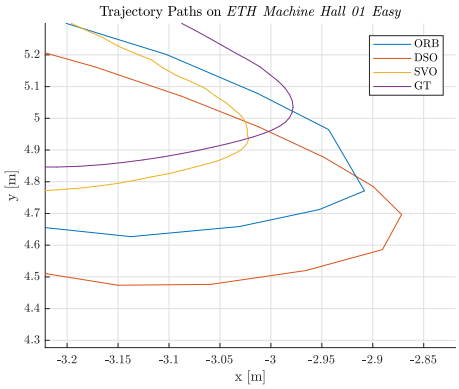


Figure 23: Close-up of arc motion of ETHMH01 on all three solvers

Figure 21 validates the intended functionality of all three solvers when comparing to the ground truth. The $Sim(3)$ transformation (discussed in §3.3.3) correctly aligned the solution from each solver to the given ground truth. From the exigent frame culling procedure in ORB2 (VO) the close-up taken on Figure 22 only shows a single $SE(3)$ transformation along that trajectory snippet. However when inspecting SVO we see every frame being accounted for. DSO does include a culling procedure although less pressing than ORB. For the arc motion in Figure 23 we see similar frame increments for both ORB2 (VO) and DSO, whilst SVO computes every frame. With respect to the ground truth we can compute ATE for all three solvers. The RPE would only serve as a weak indicator considering the Leica MS50 used in the MH sequences only measures position and not the

entire 6-DOF pose as is done in the Vicon sequences. From Figure 24 both ORB2 (VO)

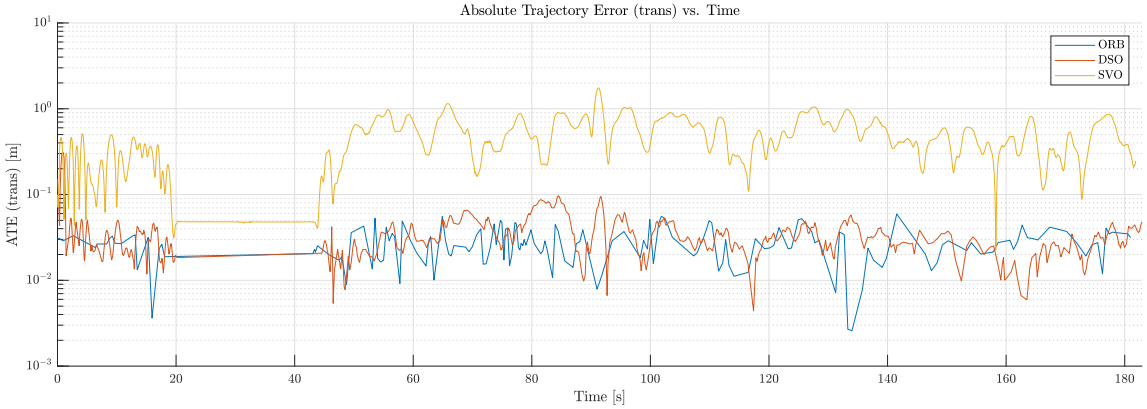


Figure 24: Absolute trajectory error (trans) on MH_01_easy

and DSO outperform SVO for ATE. Looking at Figure 25 there is a linear growth in the number of poses vs. time for SVO, whilst DSO and ORB2 (VO) on average attain a new pose every 4 and 20 frames respectively, this behaviour is also reported in [67]. Looking at the RMSE of both performance measures (See Figure 26) we see DSO and ORB2 (VO) attain very similar results whilst SVO is approximately a magnitude larger in ATE.

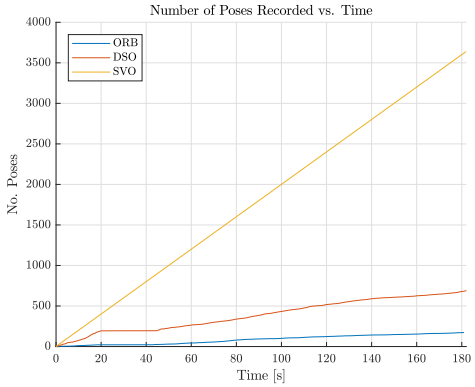


Figure 25: Pose acquisition vs time for ETHMH01

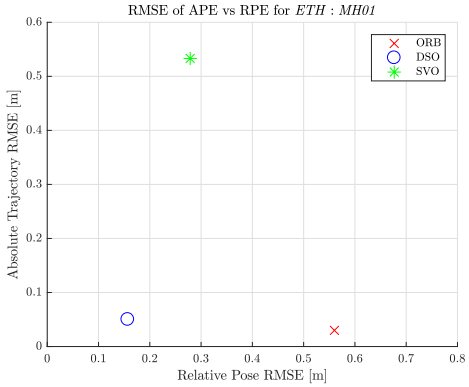


Figure 26: ATE vs RPE (RMSE) on ETHMH01

It is clear from the sequence presented the better option for [S1] is to use DSO or ORB2 (VO). Considering SVO could not retain tracking it has been removed from the decision process. The performance of the remaining solvers is displayed on the SCM in Figure 27. Looking at each cell in Figure 27 we see that all but two scenes fall within the desired range for ORB2 (VO), whilst DSO fails in 3 scenes. Since ORB2 (VO) outperforms DSO on all scenes except $ETHV103$ the clear choice is ORB2 (VO), however both scenes which fell within the desired tolerance will be explored. The variables $(\dot{\theta}, l)$ for each sequence are as follows. $ETHV103 = (0.62\ [rad/s], 36.5\ [m])$ and $ETHV203 = (0.66\ [rad/s], 86.1\ [m])$. The average rotational velocity for both scenes is almost 50% higher than the sequences directly above on the SCM.

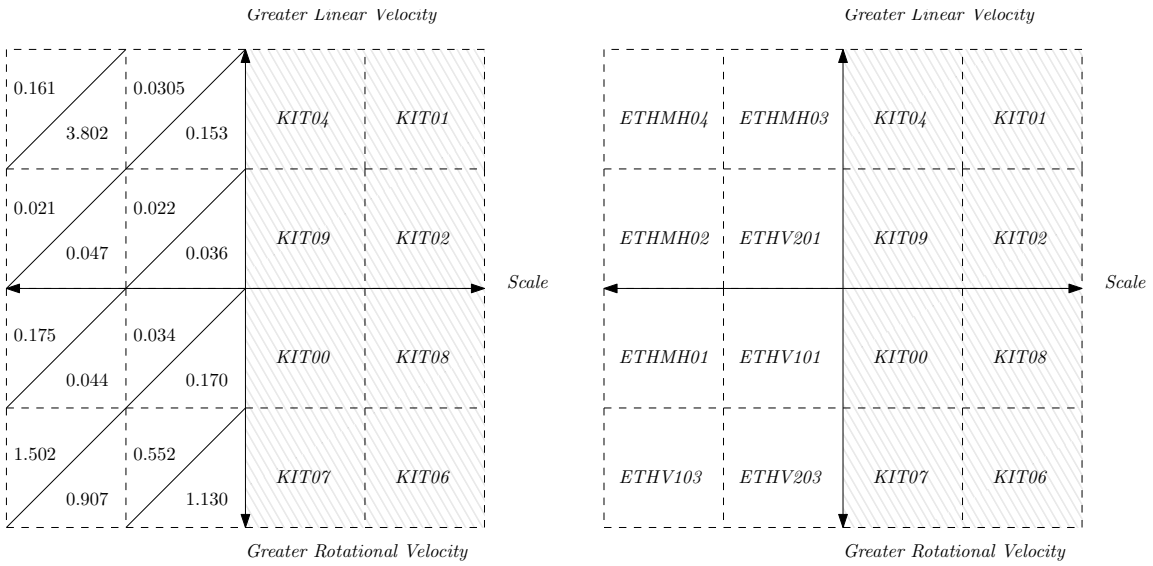


Figure 27: Evaluated SCM from S1 for the RMSE absolute trajectory error [m] for ORB2 (VO) (Top) and DSO (Bottom). The shaded cell indicates that dataset is exempt from the benchmark

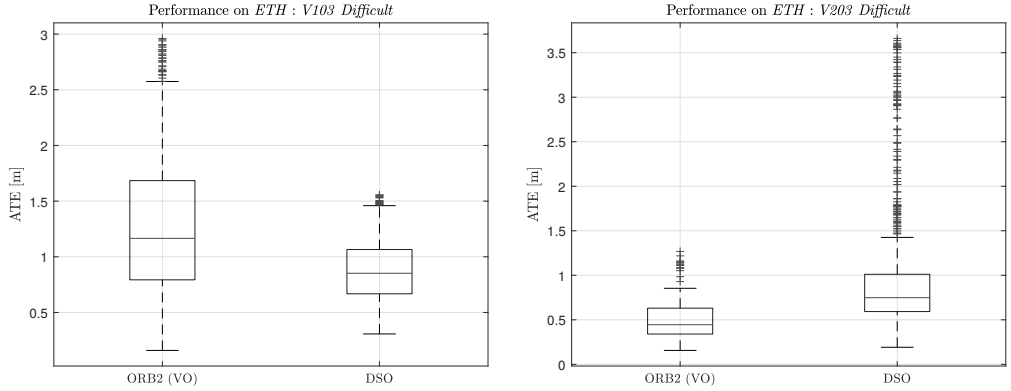


Figure 28: Boxplot displaying the ATE metric on the datasets which violated the tolerance condition (left: ETHV103, right ETH:V203)

S1 Recommendation

ORB2 (VO) with the rotational velocity not exceeding 0.3 [rad/s]. For improved performance in faster environments a different solver or further tuning would be required

Following the recommendation is an exploration into the results based on the rules of thumb §3.4; It is evident from Figure 27 that scenes with increasing rotational or translational velocity will deteriorate performance. The scenes centred around the middle all achieve close performance (within 20cm) however as we look at scenes with increased motion characteristics the solvers can break down. As does DSO on $ETH : MH04$. Under these

benchmarking conditions ORB2 (VO) managed to outperform DSO on 6 out of the 8 scenes. Of the two cases where DSO performed better, $ETH : V103$ produced an RMSE absolute trajectory error of 1.5m for ORB2 (VO) and nearly 1m for DSO. This highlights that both solvers did not perform to a usable level considering the ETH setup is only a 5x5 metre room. Both times where DSO did outperform ORB2 (VO) was on scenes with a higher rotational velocity which stems from the sliding window optimisation that DSO has [29]. It considers the most recent frames which gives it an edge for tracking motion with high rotational velocity. From the scenes tested scale does not appear to have a noticeable influence on the solvers, which is probably because all scenes in the ETH dataset travel a maximum distance of 130m ($MH03$) whilst the average length is 81.3m. Considering the RMSE ATE for the 2nd and 3rd row the solvers produce similar results under easier motion environments. If S1 only considered this spectrum of motion then a further SCM evaluation could be undertaken on a separate SCM such as the TUM dataset to cement a decision (results for this are shown §B.)

4.2 SLAM Results

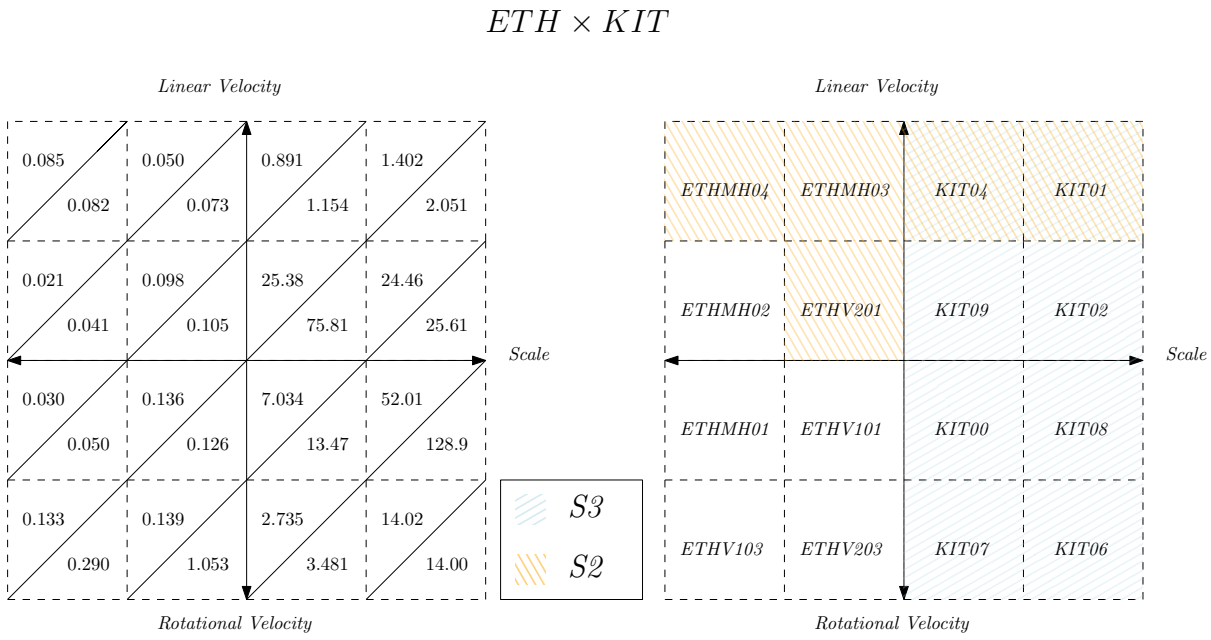


Figure 29: LDSO vs ORB2 on the ETH × KIT SCM. Displayed is the RMSE of the absolute trajectory error (ATE) in [m]

4.2.1 Trajectory Analysis on ETH×KIT

For S2 and S3 we require loop closure functionality which will use a full SLAM system (From the decision tree in Figure 17). The results in the SCM on Figure 29 indicate an average reduction of 27.7% (0.08m) when comparing to ORB2 (VO) and a reduction of 55.8% (0.23m) for DSO, which indicates the added power from the map, aswell as validating its utility. For both applications the increase in scale from the condition table has justified the usage of the entire SCM. Based on S2 the mining tunnels are primarily straight lines with minimal turning. Because of this the top submatrix has been highlighted to better represent the range of conditions the application demands, $ETH : V201$ has also been included as it is good practise to ensure atleast one sequence has a full 6DOF ground truth. The KIT sequences do contain this however $ETHV201$ is significantly shorter. Looking at the hatched sequences we recognise only $KIT01$ violates the applications requirements for $ORB2$. Sequences in this quadrant use scale and velocity (s, v) as the indicators. For $KIT01 = (2453m, 21.5m/s)$ and $KIT04 = (393m, 30.9m/s)$. Since the trajectory scale will be exposed to at most $1km$ this edge case will unlikely be experienced. Under the scale requirements ORB2 falls under the trajectory limit. With further parameter tuning both solvers could operate sucessfully (only a 0.2m reduction for LDSO) however from these results alone

S2 Recommendation

ORB2 ensuring the operating trajectory remains within the proposed conditions of $1km$**.**

For S3 we want to understand the performance over the entire spectrum of motion. Again the choice to use the right hand side for S3 is reflected in the applications trajectory. Sequences which violate the trajectory tolerance is $KIT02 = (5067m, 10.5m/s)$ and $KIT08 = (3223m, 7.62m/s)$. There is no clear restriction to be placed on the recommendation as $KIT02$ has a lower trajectory error yet it has higher variables. Looking at

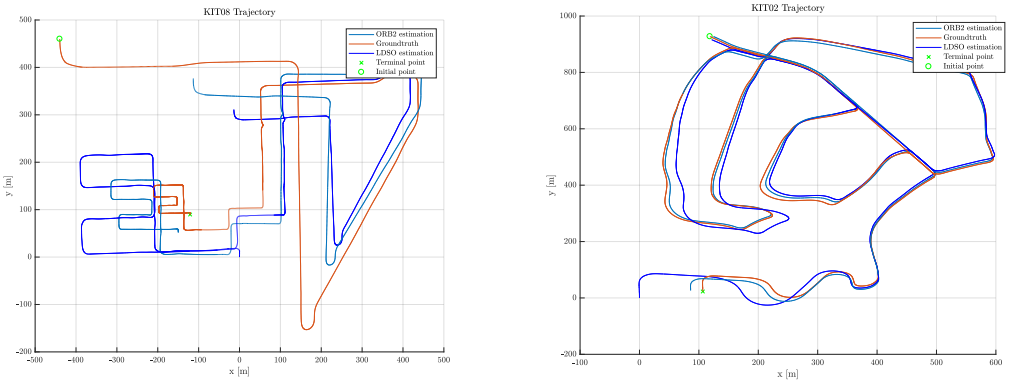


Figure 30: KIT08 and KIT02, highlighting scenes where scale drift is large (left) and how scale drift can be avoided on longer sequences if loop closures occur.

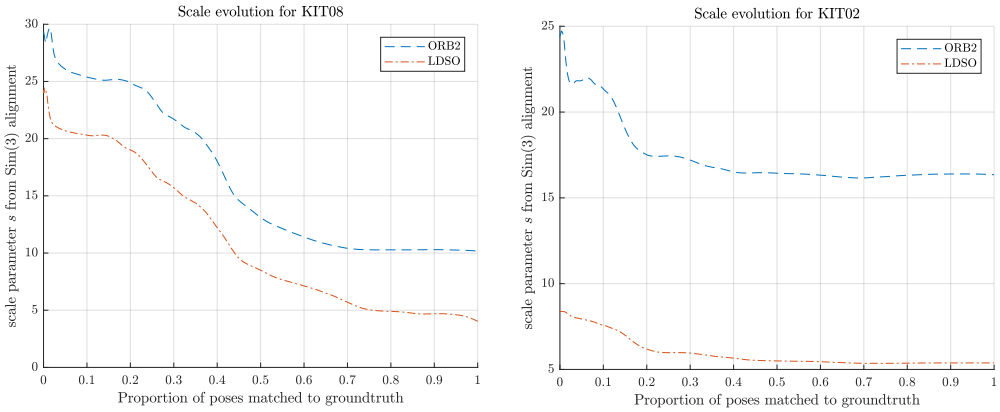


Figure 31: Observing how the scale parameter s in the $Sim(3)$ alignment evolves when more poses are matched to the ground truth. When scale drift causes the solver to be ineffective the parameter will not converge to a value. For KIT02 (right) we see an immediate convergence from multiple loop closure corrections

the trajectory for both sequences shows that the map itself does not have loop closures. Considering the application performs multiple loops

S3 Recommendation Part 1

ORB2 ensuring there are sufficient loop closures to avoid scale drift.

Without the inter camera distance from a stereo setup the algorithm is prone to incorrectly constructing the trajectory or the map points at each frame [74]. This compounding error can affect the scale of the problem and unless corrected via a loop closure or an optimisation technique the solution can quickly diverge. From Figure 31 the scale parameter does not converge on the $KTI02$ sequence as more poses are added to the *Umeyama* alignment. When the scale parameter does converge scale drift does not appear to deteriorate the solution, as shown to the right of Figure 31. The effects of this drift are very damaging. You can see that both solvers gradually make the map and trajectory decrease in size. Scenes such as this, identified by the SCM play an important role in validation and testing a chosen application. Scale is not an issue for these solvers if certain conditions are satisfied. $KIT08$ for instance is approximately 2km (5067 [m]) longer than $KIT02$ (3223 [m]), yet with apparent loop closures the system can maintain working operation (based on S3 requirements). The ordering of the SCM is able to reveal interesting behaviour. For the scale drift the sequences variables was not indicative of the performance difference, which forced a look at the trajectory. A concern immediately presented under the selected ordering is the large difference between scales on either side of the abcissa. The shortest trajectory from the KIT dataset is 400m which is almost twice the length of the longest trajectory from the ETH dataset. This characterisation for presenting the datasets is motivated with the intention to promote the collection and acquisition of a homogeneous pool of sequences in the community. In these 16 sequences on the SCM ORB2 achieves a lower RMSE ATE in 12 sequences. Focusing on the S2 submatrix we notice similar

performance for all sequences. ORB2 slightly outperforms each solver. To get a better characterisation of the performance we will explore the performance measure against a motion variable. Figure 32 shows the frame filtering from each solver, especially from ORB2. Both solvers from $10-20$ seconds do not compute any poses for $ETH:V203$. This is because the visual feed from the monocular camera was static, and this pre-processing stage allows lifelong operation. What is clear from the distribution of points is that scenes with higher rotational and translational velocity will accept more frames. In 90 seconds LDSO acquired 875 frames in $ETH:V203$ whilst for the $ETH:MH01$ sequence it had 450. ORB2 on the other hand remained within 10% of 100 frames for both. This indicates that ORB2 is more scalable than LDSO. This is always a limitation for solvers that perform bundle adjustment. Although LDSO only performs BA along the trajectory and map points in the sliding window it will still add to the processing time. ORB2 does a larger BA taking into account the entire covisibility graph. This strength from ORB2 stems from the ability to perform interactive bundle adjustment by focusing on scalable rejection protocols. Both S2 and S3 are an order of magnitude larger than the first scenario demanding longer operation. For S2 and S3 the magnitude of the scale will usually require

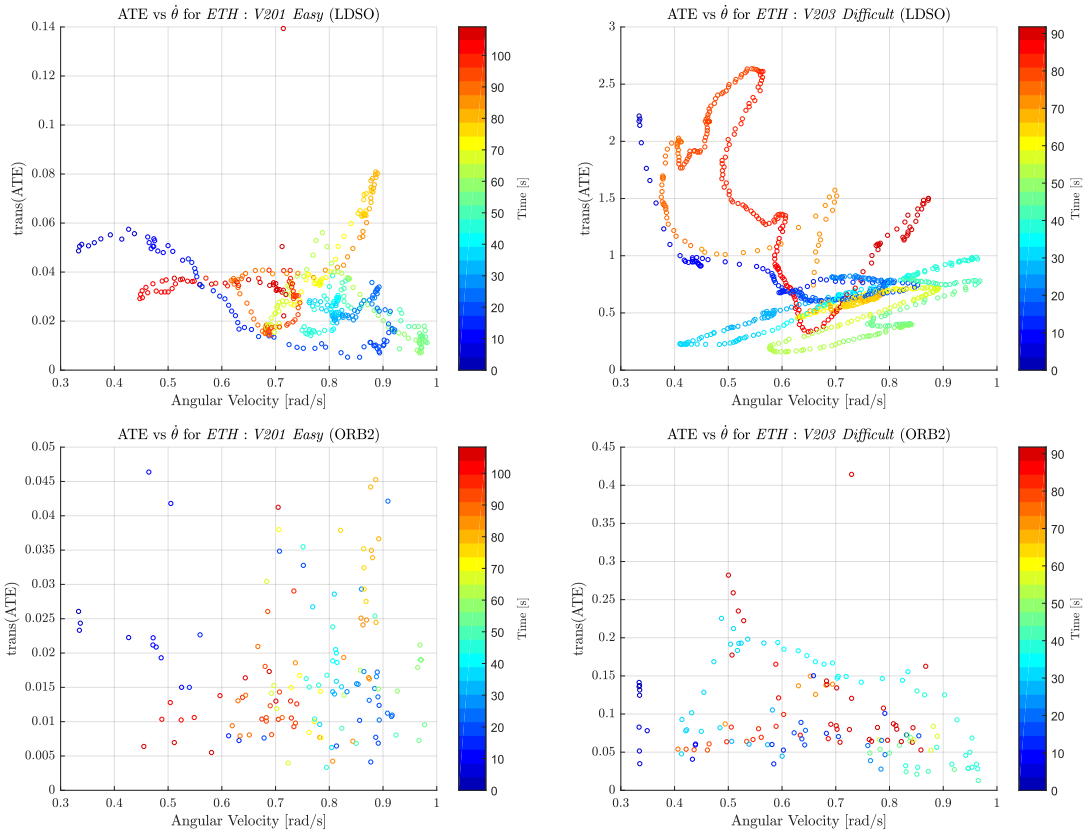


Figure 32: Looking at two vicon sequences from the ETH dataset to highlight the difficult transient effects as well as the exigency in ORB2's pose acquisition. (Best viewed in colour)

global bundle adjustment. For S3 we will investigate the performance on the submatrix in Figure 29. Looking at the ATE measure we see a convincing winner from ORB2. It

outperforms LDSO on 7 from 8 solvers and achieves equal performance on $KIT06$. From the rules of thumb it is informative to look at the edge cells of the SCM, as well as focus on any clear outliers. We can investigate the solvers performance on $KIT07$ to tease out any concerns dealing with the application of S3. From Figure 33 we notice peaks and troughs from both solvers. What is significant is that both peaks occur at corner points of the sequence. This indicates that points of increasing rotational velocity are harder to track, which is a similar conclusion we had in §4.1.1. It also appears that LDSO is lagging behind ORB2 which could be an indication for a slower solver Based on the trajectory alone ORB2 will be carried forward to determine map consistency.

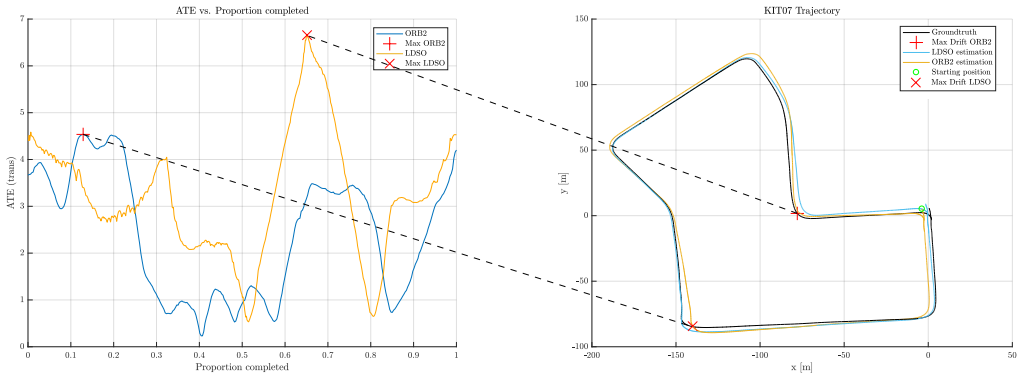
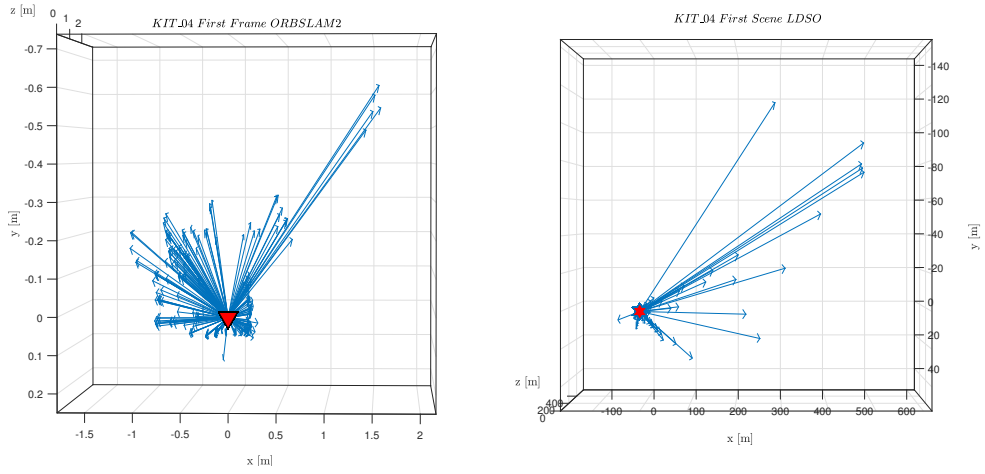


Figure 33: Plotting the ATE of ORB2 and LDSO on the KIT07 sequence. Both markers refer to frames in the scene with maximum ATE. These have been mapped to the trajectory to indicate position.

4.2.2 Map Consistency



(a) ORBSLAM2 *(b) LDSO*

Figure 34: Rays from the optical centre of the camera to the map points for the first frame in the KIT sequence 04. (Note the scale difference)

In order to validate global consistency in (S3) it is important to note a serious restriction when comparing this method of map consistency between two implementations. Firstly,

the number of features that the SLAM implementation uses per keyframe will significantly affect computation time, and the paramaterisation of feature points can invalidate this method in certain sequences. Take for example $KIT04$, and the first frame of the sequence (See Figure 34). Considering the convex hulls between respective frames can be significantly different in LDSO (from the inverse depth parametrisation) the map consistency measure breaks down §2.3.2 for outdoor environments. One could place a saturation value on the map points generated by LDSO but in §3.1 it was made clear that no filters or specialised adjustments would be made to the internal pipeline. Removing the very map points that give accurate rotational anchors would affect the integrity of the implementation. Therefore for the definition extended from the works of [66] LDSO will most likely not achieve consistency in outdoor environments or environments with windows. In the proposed paper *global consistency* is achieved if the cascading hypothesis test (outlined in §2.3.2 is passed for all pairs of poses. This produces an SCM with each cell producing a binary value (shown in Figure 35). All sequences were considered globally consistent

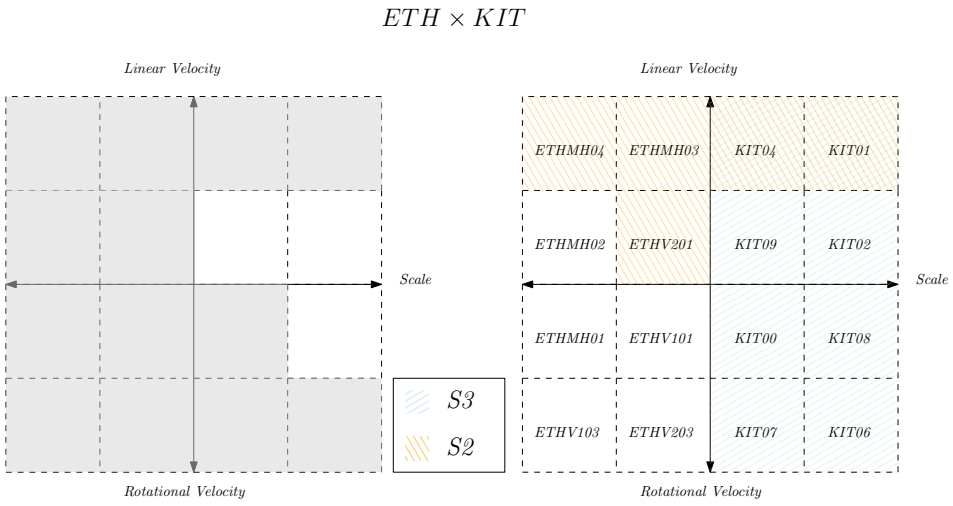


Figure 35: Map consistency on $ETH \times KIT$ for ORB2. The filled in cells indicate global consistency has been reached with a confidence of 0.95*. This is a user set parameter.*

except for KIT02,09 and 08. It does appear that scenes with higher ATE produce less consistent maps. The scene that experiences scale drift is intuitively not consistent due to the growing and shrinking that occurs to the viewing cone over time. The recommendation for map consistency is to ensure loop closures and be careful when the operating trajectory exceeds 3000m, as both KIT02 and KIT08 fall into this category.

S3 Recommendation Part 2

For consistent map generation in S3 it is recommended to follow Part 1 as well as an operating restriction that the length of the trajectory does not exceed 3000m.

Limitations concerning this metric is the unsuitability for the inverse depth representation. The authors definition of *consistency*[66] is not immediately clear. A better definition given in [93] is that it should contain no artifacts and doublets except all perceivable structures. The idea of computing the polygonal boundaries from sparse points has been explored here [58], however the author also uses ORB2 and does not consider an inverse depth representation. A further limitation in our approach is the way in which variance

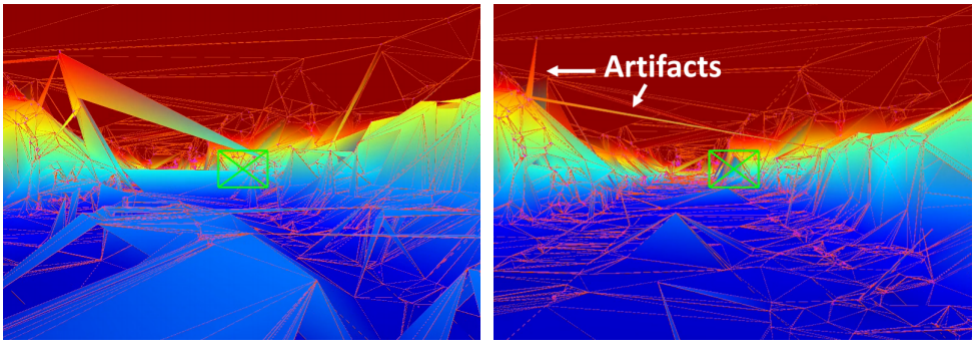


Figure 36: Upon a loop closure on KIT01 the previous sparse map cloud contains artifacts, however after the loop closure the algorithm deletes vertexes and constructs a consistent map [58]

was computed, for both solvers the first frame of the sequence was run 100 times and the difference between the average length of the optical ray was assigned the variance for that sequence. Although this represents the variability between consecutive runs and captures the non deterministic effects it is still not rigorous in the original assumption of the paper [93]. For an RGB-D sensor it is possible to generate variance in the position of the landmark as done here [11]. However this benchmark was for monocular SLAM which defeats the intention of the report.

5 Conclusions and Future Work

A V-stage multi-objective pipeline has been developed to transform a given SLAM application into an informed decision about which solver to use. To arrive at the decision a condition table was developed to abstract the SLAM application into functional requirements. A decision tree was used to select the benchmarking protocol and a selection and ordering criteria concerning metrics, algorithms and datasets allowed a consistent and thorough evaluation. Three separate applications were considered and three recommendations were made with accompanying operating conditions. In the broader context the set of principles proposed allow a robust and fair framework for comparing SLAM implementations. The procedure, when followed disrupts the pattern of over parameterisation and delineates over-fitting.

The condition table contained 6 variables that were general yet informative for the given context. This characterisation of an application could be applied in any setting and this is evident in the diversity of the three example applications. The estimated solutions for each application were validated through comparisons to the trajectory, ensuring the frame acceptance protocol was operational and observing the increased performance of adding loop closure functionality. This was cross referenced with the literature to ensure the solvers achieved intended functionality. The validation provides strength in the recommendations. Each recommendation also provided insightful conditions surrounding scale drift and velocity constraints.

This systematic approach allowed an efficient identification of three intrinsic limitations: First. If loop closures do not occur on large scenes then scale drift will occur, second. The sliding window optimisation gives a performance boost on scenes with greater rotational velocity and three. From the sequence ordering scenes with higher average motion characteristics will deteriorate the performance of the solvers. Although these properties have been established this approach to ordering and characterising the datasets allowed an efficient discovery. The map consistency from [66] was extended to the visual SLAM problem and the limitations were discussed, it was recognised that map consistency should be able to deal with solvers that use an inverse depth representation

The black box approach to testing the SLAM implementations had the advantage of clarifying that the benchmarking framework was the objective of the report, however it also meant that the same three recommendations for each application was ORB2. With parameter tweaking LDSO could very well achieve S2. The variability in recommendations would have provided a better outcome that a different solver could be selected under this framework. However if the blackbox approach was not taken this could have resulted in the testing of a lot more solver variants and that was not the point of this report. With respect to trajectory metrics the choices provided an understanding of global accuracy of the map however the solvers chosen did not allow the relative pose error to be correctly measured.

The need to develop independent mapping metrics is apparent, it is particularly important to deal with SLAM applications that use different coordinate representations. The community should place continued efforts in ensuring a set of principles is followed when benchmarking SLAM implementations for commercial use. As SLAM becomes more applicable in industry the regulatory framework will need to be clear to allow a virtuous co-design cycle. Under the proposed principles it is difficult to show performance that is achieved by over fitting sequences, which would be very advantageous in this next stage, especially if the principles can be embedded in an evaluation framework like GSLAM [97] or SlamBench3.0 [15].

References

[1] Abouzahir, Mohamed et al. "Embedding SLAM algorithms: Has it come of age?" In: *Robotics and Autonomous Systems* 100 (2018), pp. 14–26. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2017.10.019. URL: http://www.sciencedirect.com/science/article/pii/S0921889017301963.

[2] Agarwal, Aditya, Maturana, Daniel, and Scherer, Sebastian A. "Visual Odometry in Smoke Occluded Environments". In: 2015.

[3] Alahi, Alexandre, Ortiz, Raphael, and Vandergheynst, Pierre. "FREAK: Fast Retina Keypoint". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition* (2012), pp. 510–517.

[4] Amigoni, F., Gasparini, S., and Gini, M. "Good Experimental Methodologies for Robotic Mapping: A Proposal". In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. Apr. 2007, pp. 4176–4181. DOI: 10.1109/ROBOT.2007.364121.

[5] Arun, K. S., Huang, T. S., and Blostein, S. D. "Least-Squares Fitting of Two 3-D Point Sets". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 9.5 (May 1987), pp. 698–700. ISSN: 0162-8828. DOI: 10.1109/TPAMI.1987.4767965. URL: https://doi.org/10.1109/TPAMI.1987.4767965.

[6] Ayache, Nicholas and Faugeras, Olivier D. "Building, Registrating, and Fusing Noisy Visual Maps". In: *The International Journal of Robotics Research* 7.6 (1988), pp. 45–65. DOI: 10.1177/027836498800700605. eprint: https://doi.org/10.1177/027836498800700605. URL: https://doi.org/10.1177/027836498800700605.

[7] Balaguer, Benjamin, Carpin, Stefano, and Balakirsky, Stephen. "Towards Quantitative Comparisons of Robot Algorithms : Experiences with SLAM in Simulation and Real World Systems". In: 2007.

[8] Baldwin, Grant, Mahony, Robert, and Trumpf, Jochen. "A nonlinear observer for 6 DOF pose estimation from inertial and bearing measurements". In: June 2009, pp. 2237–2242. DOI: 10.1109/ROBOT.2009.5152242.

[9] Barrau, A. and Bonnabel, S. "The Invariant Extended Kalman Filter as a Stable Observer". In: *IEEE Transactions on Automatic Control* 62.4 (Apr. 2017), pp. 1797–1812. ISSN: 0018-9286. DOI: 10.1109/TAC.2016.2594085.

[10] Barrau, Axel and Bonnabel, Silvere. "An EKF-SLAM algorithm with consistency properties". In: *CoRR* abs/1510.06263 (2015). arXiv: 1510.06263. URL: http://arxiv.org/abs/1510.06263.

[11] Barron, J. T. and Malik, J. "Intrinsic Scene Properties from a Single RGB-D Image". In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. June 2013, pp. 17–24. DOI: 10.1109/CVPR.2013.10.

[12] Bay, Herbert et al. "Speeded-Up Robust Features (SURF)". In: *Computer Vision and Image Understanding* 110.3 (2008). Similarity Matching in Computer Vision and Multimedia, pp. 346–359. ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2007.09.014. URL: http://www.sciencedirect.com/science/article/pii/S1077314207001555.

[13] Bonnabel, Silvere. "Symmetries in observer design: Review of some recent results and applications to ekf-based slam". In: *Robot Motion and Control 2011*. Springer, 2012, pp. 3–15.

[14] Bonnabel, Silvere, Martin, Ph, and Rouchon, Pierre. "Symmetry-preserving observers". In: *arXiv preprint math/0612193* (2006).

[15] Bujanca, M. et al. "SLAMBench 3.0: Systematic Automated Reproducible Evaluation of SLAM Systems for Robot Vision Challenges and Scene Understanding". In: *2019 International Conference on Robotics and Automation (ICRA)*. May 2019, pp. 6351–6358. DOI: 10.1109/ICRA.2019.8794369.

[16] Burri, Michael et al. "The EuRoC micro aerial vehicle datasets". In: *The International Journal of Robotics Research* (2016). DOI: 10.1177/0278364915620033. eprint: http://ijr.sagepub.com/content/early/2016/01/21. URL: http://ijr.sagepub.com/content/early/2016/01/21.

[17] Cadena, Cesar et al. "Simultaneous Localization And Mapping: Present, Future, and the Robust-Perception Age". In: *CoRR* abs/1606.05830 (2016). arXiv: 1606.05830. URL: http://arxiv.org/abs/1606.05830.

[18] Calonder, Michael et al. "BRIEF: Binary Robust Independent Elementary Features". In: *Computer Vision – ECCV 2010*. Ed. by Daniilidis, Kostas, Maragos, Petros, and Paragios, Nikos. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 778–792. ISBN: 978-3-642-15561-1.

[19] Carlevaris-Bianco, Nicholas, Ushani, Arash K., and Eustice, Ryan M. "University of Michigan North Campus long-term vision and lidar dataset". In: *International Journal of Robotics Research* 35.9 (2015), pp. 1023–1035.

[20] Chen, D. M. et al. "City-scale landmark identification on mobile devices". In: *CVPR 2011*. June 2011, pp. 737–744. DOI: 10.1109/CVPR.2011.5995610.

[21] Civera, J., Davison, A. J., and Montiel, J. M. M. "Inverse Depth Parametrization for Monocular SLAM". In: *IEEE Transactions on Robotics* 24.5 (Oct. 2008), pp. 932–945. ISSN: 1552-3098. DOI: 10.1109/TRO.2008.2003276.

[22] Cordts, Marius et al. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[23] Davison, A. J. et al. "MonoSLAM: Real-Time Single Camera SLAM". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.6 (June 2007), pp. 1052–1067. DOI: 10.1109/TPAMI.2007.1049.

[24] Dellaert, Frank and Kaess, Michael. "Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing". In: *The International Journal of Robotics Research* 25.12 (2006), pp. 1181–1203. DOI: 10.1177/0278364906072768. eprint: https://doi.org/10.1177/0278364906072768. URL: https://doi.org/10.1177/0278364906072768.

[25] Dellaert, Frank and Kaess, Michael. "Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing". In: *The International Journal of Robotics Research* 25.12 (2006), pp. 1181–1203. DOI: 10.1177/0278364906072768. eprint: https://doi.org/10.1177/0278364906072768. URL: https://doi.org/10.1177/0278364906072768.

[26] Delmerico, J. and Scaramuzza, D. "A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. May 2018, pp. 2502–2509. DOI: 10.1109/ICRA.2018.8460664.

[27] Durrant-Whyte, H. F. "Uncertain geometry in robotics". In: *IEEE Journal on Robotics and Automation* 4.1 (Feb. 1988), pp. 23–31. DOI: 10.1109/56.768.

[28] Durrant-Whyte, H. and Bailey, T. "Simultaneous localization and mapping: part I". In: *IEEE Robotics Automation Magazine* 13.2 (June 2006), pp. 99–110. DOI: 10.1109/MRA.2006.1638022.

[29] Engel, J., Koltun, V., and Cremers, D. "Direct Sparse Odometry". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Mar. 2018).

[30] Engel, J., Schöps, T., and Cremers, D. "LSD-SLAM: Large-Scale Direct Monocular SLAM". In: *European Conference on Computer Vision (ECCV)*. Sept. 2014.

[31] Engel, J., Usenko, V., and Cremers, D. "A Photometrically Calibrated Benchmark For Monocular Visual Odometry". In: *arXiv:1607.02555*. July 2016.

[32] Fang, W. et al. "FPGA-based ORB feature extraction for real-time visual SLAM". In: *2017 International Conference on Field Programmable Technology (ICFPT)*. Dec. 2017, pp. 275–278. DOI: 10.1109/FPT.2017.8280159.

[33] Faugeras, O. D. and Hebert, M. "A 3-D Recognition and Positioning Algorithm Using Geometrical Matching Between Primitive Surfaces". In: *Proceedings of the Eighth International Joint Conference on Artificial Intelligence - Volume 2*. IJCAI'83. Karlsruhe, West Germany: Morgan Kaufmann Publishers Inc., 1983, pp. 996–1002. URL: http://dl.acm.org/citation.cfm?id=1623516.1623603.

[34] Ferrera, Maxime et al. "AQUALOC: An underwater dataset for visual–inertial–pressure localization". In: *The International Journal of Robotics Research* 0.0 (0), p. 0278364919883346. DOI: 10.1177/0278364919883346. eprint: https://doi.org/10.1177/0278364919883346. URL: https://doi.org/10.1177/0278364919883346.

[35] Forster, Christian, Pizzoli, Matia, and Scaramuzza, Davide. "SVO: Fast semi-direct monocular visual odometry". In: *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2014, pp. 15–22.

[36] Gao, Xiang et al. "LDSO: Direct Sparse Odometry with Loop Closure". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2018), pp. 2198–2204.

[37] Gautier, Quentin, Althoff, Alric, and Kastner, Ryan. "FPGA Architectures for Real-time Dense SLAM". In: ().

[38] Geiger, Andreas et al. "Vision meets Robotics: The KITTI Dataset". In: *International Journal of Robotics Research (IJRR)* (2013).

[39] Goor, Pieter van et al. "An Equivariant Observer Design for Visual Localisation and Mapping". In: *CoRR* abs/1904.02452 (2019). arXiv: 1904.02452. URL: http://arxiv.org/abs/1904.02452.

[40] Grisetti, G. et al. "Hierarchical optimization on manifolds for online 2D and 3D mapping". In: *2010 IEEE International Conference on Robotics and Automation*. May 2010, pp. 273–278. DOI: 10.1109/ROBOT.2010.5509407.

[41] Handa, A. et al. "A Benchmark for RGB-D Visual Odometry, 3D Reconstruction and SLAM". In: *IEEE Intl. Conf. on Robotics and Automation, ICRA*. Hong Kong, China, May 2014.

[42] Huang, Shoudong et al. "Iterated D-SLAM map joining: Evaluating its performance in terms of consistency, accuracy and efficiency". In: *Auton. Robots* 27 (Nov. 2009), pp. 409–429. DOI: 10.1007/s10514-009-9153-8.

[43] Ibragimov, I. Z. and Afanasyev, I. M. "Comparison of ROS-based visual SLAM methods in homogeneous indoor environment". In: *2017 14th Workshop on Positioning, Navigation and Communications (WPNC)*. Oct. 2017, pp. 1–6. DOI: 10.1109/WPNC.2017.8250081.

[44] J. Delmerico, T et al. "Are We Ready for Autonomous Drone Racing The UZH-FPV Drone Racing Dataset". In: *IEEE International Conference on Robotics and Automation* (2019).

[45] Jacoff, Adam et al. "Test arenas and performance metrics for urban search and rescue robots". In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*. Vol. 4. IEEE. 2003, pp. 3396–3403.

[46] Jeong, Jinyong et al. "Complex urban dataset with multi-level sensors from highly diverse urban environments". In: *The International Journal of Robotics Research* (2019), p. 0278364919843996.

[47] Kendall, Alex, Grimes, Matthew, and Cipolla, Roberto. "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization". In: (2015).

[48] Kerl, C., Stückler, J., and Cremers, D. "Dense Continuous-Time Tracking and Mapping with Rolling Shutter RGB-D Cameras". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015, pp. 2264–2272. DOI: 10.1109/ICCV.2015.261.

[49] Klein, Georg and Murray, David. "Parallel Tracking and Mapping for Small AR Workspaces". In: *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*. Nara, Japan, Nov. 2007.

[50] Kleinschmidt, S. P. and Wagner, B. "Visual Multimodal Odometry: Robust Visual Odometry in Harsh Environments". In: *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. Aug. 2018, pp. 1–8. DOI: 10.1109/SSRR.2018.8468653.

[51] Kümmerle, Rainer et al. "g 2 o: A general framework for graph optimization". In: *2011 IEEE International Conference on Robotics and Automation*. IEEE. 2011, pp. 3607–3613.

[52] Kümmerle, Rainer et al. "On measuring the accuracy of SLAM algorithms". In: *Autonomous Robots* 27.4 (Sept. 2009), p. 387. ISSN: 1573-7527. DOI: 10.1007/s10514-009-9155-6. URL: https://doi.org/10.1007/s10514-009-9155-6.

[53] Lee, S. and Lee, S. "Embedded Visual SLAM: Applications for Low-Cost Consumer Robots". In: *IEEE Robotics Automation Magazine* 20.4 (Dec. 2013), pp. 83–95. DOI: 10.1109/MRA.2013.2283642.

[54] Leung, Keith et al. "Chilean underground mine dataset". In: *The International Journal of Robotics Research* 36.1 (2017), pp. 16–23. DOI: 10.1177/0278364916679497. eprint: https://doi.org/10.1177/0278364916679497. URL: https://doi.org/10.1177/0278364916679497.

[55] Leutenegger, Stefan, Chli, Margarita, and Siegwart, Roland. "BRISK: Binary Robust invariant scalable keypoints". In: Nov. 2011, pp. 2548–2555. DOI: 10.1109/ICCV.2011.6126542.

[56] Li, Wenbin et al. "InteriorNet: Mega-scale Multi-sensor Photo-realistic Indoor Scenes Dataset". English. In: 29th British Machine Vision Conference 2018, BMVC 2018 ; Conference date: 03-09-2018 Through 06-09-2018. Sept. 2018. URL: http://bmvc2018.org/.

[57] Lin, Zse Cherng et al. "MOTION ESTIMATION FROM 3-D POINT SETS WITH AND WITHOUT CORRESPONDENCES." In: *Unknown Host Publication Title*. IEEE, 1986, pp. 194–201.

[58] Ling, Y. and Shen, S. "Building maps for autonomous navigation using sparse visual SLAM features". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sept. 2017, pp. 1374–1381. DOI: 10.1109/IROS.2017.8202316.

[59] Lovegrove, Steven, Davison, Andrew J, and Ibanez-Guzmán, Javier. "Accurate visual odometry from a rear parking camera". In: *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2011, pp. 788–793.

[60] Lowe, David G. "Distinctive Image Features from Scale-Invariant Keypoints". In: *Int. J. Comput. Vision* 60.2 (Nov. 2004), pp. 91–110. ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000029664.99615.94. URL: https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[61] Maddern, Will et al. "1 Year, 1000km: The Oxford RobotCar Dataset". In: *The International Journal of Robotics Research (IJRR)* 36.1 (2017), pp. 3–15. DOI: 10.1177/0278364916679498. eprint: http://ijr.sagepub.com/content/early/2016/11/28/0278364916679498.full.pdf+html. URL: http://dx.doi.org/10.1177/0278364916679498.

[62] Mahmoud, Doaa et al. "Comparison of Optimization Techniques for 3D Graph-based SLAM". In: Oct. 2013.

[63] Mahony, R. and Hamel, T. "A geometric nonlinear observer for simultaneous localisation and mapping". In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. Dec. 2017, pp. 2408–2415. DOI: 10.1109/CDC.2017.8264002.

[64] Mahony, R., Hamel, T., and Pflimlin, J. "Nonlinear Complementary Filters on the Special Orthogonal Group". In: *IEEE Transactions on Automatic Control* 53.5 (June 2008), pp. 1203–1218. ISSN: 0018-9286. DOI: 10.1109/TAC.2008.923738.

[65] Mallios, Angelos et al. "Underwater caves sonar data set". In: *The International Journal of Robotics Research* 36.12 (2017), pp. 1247–1251. DOI: 10.1177/0278364917732838. eprint: https://doi.org/10.1177/0278364917732838. URL: https://doi.org/10.1177/0278364917732838.

[66] Mazuran, M. et al. "A statistical measure for map consistency in SLAM". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. May 2014, pp. 3650–3655. DOI: 10.1109/ICRA.2014.6907387.

[67] Mur-Artal, Raul, Montiel, J. M. M., and Tardós, Juan D. "ORB-SLAM: a Versatile and Accurate Monocular SLAM System". In: *CoRR* abs/1502.00956 (2015). arXiv: 1502.00956. URL: http://arxiv.org/abs/1502.00956.

[68] Newcombe, R. A., Lovegrove, S. J., and Davison, A. J. "DTAM: Dense tracking and mapping in real-time". In: *2011 International Conference on Computer Vision*. Nov. 2011, pp. 2320–2327. DOI: 10.1109/ICCV.2011.6126513.

[69] Pandey, Gaurav, McBride, James R., and Eustice, Ryan M. "Ford campus vision and lidar data set". In: *International Journal of Robotics Research* 30.13 (2011), pp. 1543–1552.

[70] Peynot, T., Scheding, S., and Terho, S. "The Marulan Data Sets: Multi-Sensor Perception in Natural Environment with Challenging Conditions". In: *International Journal of Robotics Research* 29.13 (Nov. 2010), pp. 1602–1607.

[71] Quattrini Li, Alberto et al. "Experimental Comparison of Open Source Vision-Based State Estimation Algorithms". In: *2016 International Symposium on Experimental Robotics*. Ed. by Kulić, Dana et al. Cham: Springer International Publishing, 2017, pp. 775–786. ISBN: 978-3-319-50115-4.

[72] Recht, Benjamin et al. "Do ImageNet Classifiers Generalize to ImageNet?" In: *CoRR* abs/1902.10811 (2019). arXiv: 1902.10811. URL: http://arxiv.org/abs/1902.10811.

[73] Rosten, E., Porter, R., and Drummond, T. "Faster and Better: A Machine Learning Approach to Corner Detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.1 (Jan. 2010), pp. 105–119. DOI: 10.1109/TPAMI.2008.275.

[74] Rukhovich, Danila et al. *Estimation of Absolute Scale in Monocular SLAM Using Synthetic Data*. 2019. arXiv: 1909.00713 [cs.CV].

[75] Saeedi, Sajad et al. "Characterizing Visual Localization and Mapping Datasets". English. In: *IEEE/RSJ International Conference on Robotics and Automation (ICRA)*. Proceedings - International Conference on Robotics and Automation. IEEE, June 2019.

[76] Saeedi, S. et al. "Navigating the Landscape for Real-Time Localization and Mapping for Robotics and Virtual and Augmented Reality". In: *Proceedings of the IEEE* 106.11 (Nov. 2018), pp. 2020–2039. DOI: 10.1109/JPROC.2018.2856739.

[77] Schubert, D. et al. "The TUM VI Benchmark for Evaluating Visual-Inertial Odometry". In: *International Conference on Intelligent Robots and Systems (IROS)*. Oct. 2018.

[78] Smith, Randall C. and Cheeseman, Peter. "On the Representation and Estimation of Spatial Uncertainty". In: *The International Journal of Robotics Research* 5.4 (1986), pp. 56–68. DOI: 10.1177/027836498600500404. eprint: https://doi.org/10.1177/027836498600500404. URL: https://doi.org/10.1177/027836498600500404.

[79] Smith, Randall, Self, Matthew, and Cheeseman, Peter. "Estimating Uncertain Spatial Relationships in Robotics* *The research reported in this paper was supported by the National Science Foundation under Grant ECS-8200615, the Air Force Office of Scientific Research under Contract F49620-84-K-0007, and by General Motors Research Laboratories." In: *Uncertainty in Artificial Intelligence*. Ed. by LEMMER, John F. and KANAL, Laveen N. Vol. 5. Machine Intelligence and Pattern Recognition. North-Holland, 1988, pp. 435–461. DOI: https://doi.org/10.1016/B978-0-444-70396-5.50042-X. URL: http://www.sciencedirect.com/science/article/pii/B978044470396550042X.

[80] Sturm, J. et al. "A benchmark for the evaluation of RGB-D SLAM systems". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Oct. 2012, pp. 573–580. DOI: 10.1109/IROS.2012.6385773.

[81] Thrun, Sebastian. "Exploring Artificial Intelligence in the New Millennium". In: ed. by Lakemeyer, Gerhard and Nebel, Bernhard. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003. Chap. Robotic Mapping: A Survey, pp. 1–35. ISBN: 1-55860-811-7. URL: http://dl.acm.org/citation.cfm?id=779343.779345.

[82] Thrun, Sebastian, Burgard, Wolfram, and Fox, Dieter. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN: 0262201623.

[83] Thrun, Sebastian, Burgard, Wolfram, and Fox, Dieter. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN: 0262201623.

[84] Thrun, Sebastian et al. "Stanley: The robot that won the DARPA Grand Challenge". In: *J. Field Robotics* 23 (2006), pp. 661–692.

[85] Thrun, S. et al. "Autonomous exploration and mapping of abandoned mines". In: *IEEE Robotics Automation Magazine* 11.4 (Dec. 2004), pp. 79–91. DOI: 10.1109/MRA.2004.1371614.

[86] Triggs, Bill et al. "Bundle Adjustment - A Modern Synthesis". In: *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*. ICCV '99. London, UK, UK: Springer-Verlag, 2000, pp. 298–372. ISBN: 3-540-67973-1. URL: http://dl.acm.org/citation.cfm?id=646271.685629.

[87] Umeyama, S. "Least-squares estimation of transformation parameters between two point patterns". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13.4 (Apr. 1991), pp. 376–380. DOI: 10.1109/34.88573.

[88] Vasconcelos, José. "A Nonlinear Observer for Rigid Body Attitude Estimation Using Vector Observations". In: July 2008, pp. 8599–8604. ISBN: 9783902661005. DOI: 10.3182/20080706-5-KR-1001.01454.

[89] Vasconcelos, José et al. "A nonlinear position and attitude observer on SE(3) using landmark measurements". In: *Systems Control Letters* 59 (Mar. 2010), pp. 155–166. DOI: 10.1016/j.sysconle.2009.11.008.

[90] Vik, B. and Fossen, T. I. "A nonlinear observer for GPS and INS integration". In: *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No.01CH37228)*. Vol. 3. Dec. 2001, 2956–2961 vol.3. DOI: 10.1109/CDC.2001.980726.

[91] Wasenmüller, O., Meyer, M., and Stricker, D. "CoRBS: Comprehensive RGB-D benchmark for SLAM using Kinect v2". In: *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Mar. 2016, pp. 1–7. DOI: 10.1109/WACV.2016.7477636.

[92] Williams, Stefan B. and Mahon, Ian. "Design of an Unmanned Underwater Vehicle for Reef Surveying". In: *IFAC Proceedings Volumes* 37.14 (2004). 3rd IFAC Symposium on Mechatronic Systems 2004, Sydney, Australia, 6-8 September, 2004, pp. 175–180. ISSN: 1474-6670. DOI: https://doi.org/10.1016/S1474-6670(17)31100-X. URL: http://www.sciencedirect.com/science/article/pii/S147466701731100X.

[93] Wulf, Oliver et al. "Benchmarking urban six-degree-of-freedom simultaneous localization and mapping". In: *J. Field Robotics* 25 (2008), pp. 148–163.

[94] Yang, Nan, Wang, Rui, and Cremers, Daniel. "Feature-based or Direct: An Evaluation of Monocular Visual Odometry". In: *CoRR* abs/1705.04300 (2017). arXiv: 1705.04300. URL: http://arxiv.org/abs/1705.04300.

[95] Ye, Wenkai, Zhao, Yipu, and Vela, Patricio A. "Characterizing SLAM Benchmarks and Methods for the Robust Perception Age". In: *ArXiv* abs/1905.07808 (2019).

[96] Zeng, F., Huang, Z., and Ji, Y. "Discriminative Bag-of-Words-Based Adaptive Appearance Model for Robust Visual Tracking". In: *IEEE Signal Processing Letters* 24.6 (June 2017), pp. 907–911. DOI: 10.1109/LSP.2017.2698140.

[97] Zhao, Yong et al. "GSLAM: A General SLAM Framework and Benchmark". In: *CoRR* abs/1902.07995 (2019). arXiv: 1902.07995. URL: http://arxiv.org/abs/1902.07995.

A Coordinate Representation

The homogeneous coordinates of a landmark with respect to a given frame of reference is defined as;

$$\bar{p} \in \mathbb{E}^3, \qquad \bar{p} = \begin{bmatrix} p \\ 1 \end{bmatrix} \tag{27}$$

Where $p = (p_1, p_2, p_3)^\top \in \mathbb{R}^3$. We should think of these points as physical points in space without the additive vector space structure of $\mathbb{R}^3$. The issue with this representation, in a *monocular* setting is that low parallax measurements make it very difficult to deduce whether the feature has a depth of 10 units or of a magnitude greater. We can devise methods to circumvent this problem by only choosing features that are close to the camera relative to its latest translation (previous image) but this could mean we lose valuable information. Furthermore, paying too much attention to outlier rejection can be computationally demanding. There is also the horizon issue or dealing with points at infinity, especially in outdoor scenes. These features should exhibit no parallax and have no influence on camera translation, but they are perfect for gaining information about rotation. These issues with the homogeneous representation motivated what is known as the *inverse depth representation* which is exposited for the monocular case in [21]. The primary results conclude that numerically, this representation is far superior, it is able to deal with points of low parallax and points at infinity. It is important though how it formulated, the inverse depth is *relative to the positions from which the landmarks were first observed*. This does give a computational drawback as the inverse depth scheme requires six paramaters rather than 3. This can be shown in Figure 37

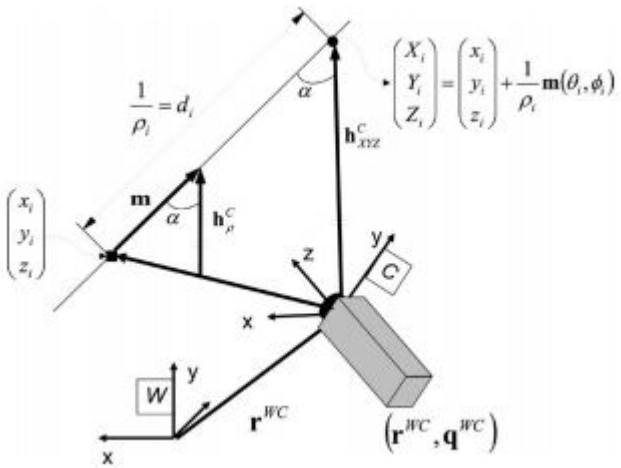


Figure 37: Inverse depth representation [21]

For this paramaterisation a point in space is defined as;

$$p_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \frac{1}{\rho_i}\mathbf{m}(\theta_i, \phi_i) \tag{28}$$

Where $\mathbf{m} = (\cos\phi_i \sin\theta_i, -\sin\phi_i, \cos\phi_i \cos\theta_i)^\top$. We also need to remember that now we define the landmark taking into account the optical centre of the camera (x_i, y_i, z_i) at the frame with which it was seen. Whats notable about this paper is the authors result on the linearity index of both the homogeneous case and the inverse depth case. When a feature is initialised the inverse depth allows the feature to have infinite depth, but cannot include zero depth. It turns out that the linearity index remains stable for low parallax α and high parallax whilst the homogeneous case breaks down.

B Drift performance on TUM

From Figure 15, the top right SCM looks at 16 sequences from the TUM dataset, whereby the duration of the footage and the proportion of time spent indoors are the two characterising variables. The heuristic here was to determine whether scalability will accelerate the affects of drift. It also serves as another meaure to differentiate performance. Recall (16) which describes the scale, rotation and translation drift over the entire sequence. We will evaluate this metric on all 16 sequences for ORB2 and DSO. The metric is defined as the translational RMSE of the tracked trajectory, when aligned (a) to the start segment and (b) to the end segment. the feature matching.

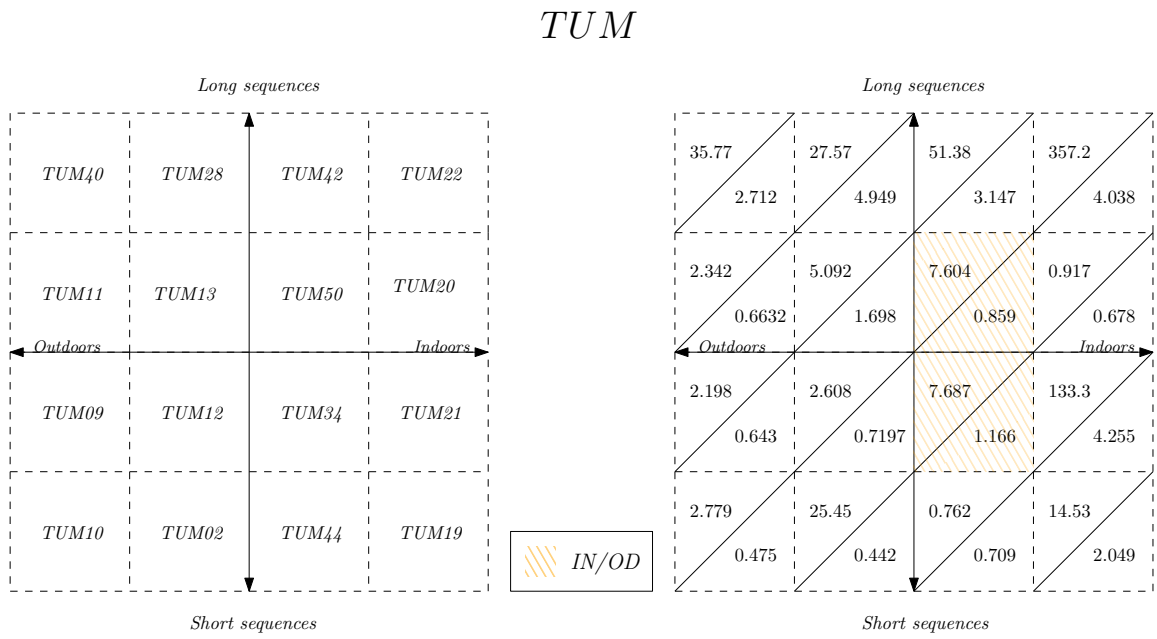


Figure 38: Evaluating the scale alignment error which describes the drift of the VO system with respect to scale, translation and rotation. (ORB top, DSO bottom)

C Sequence Properties

Sequence	Duration [s]	Length [m]	Avg. Linear Velocity [msˆ-1]	Rotational Velocity Avg.
KIT00	471	3724	7.91	0.085
KIT01	114	2453	21.5	0.055
KIT02	483	5067	10.5	0.067
KIT03	73	561	6.79	0.036
KIT04	28	393	31.8	0.01
KIT05	288	2206	7.67	0.058
KIT06	114	1233	10.8	0.13
KIT07	114	695	6.07	0.12
KIT08	423	3223	7.62	0.096
KIT09	164	1705	10.33	0.094
KIT10	124	920	7.39	0.064
ETHMH01	182	80.6	0.44	0.22
ETHMH02	150	73.5	0.49	0.21
ETHMH03	132	130.9	0.99	0.29
ETHMH04	99	91.7	0.93	0.24
ETHMH05	111	97.6	0.88	0.21
ETHV101	144	58.6	0.41	0.28
ETHV102	83.5	75.9	0.91	0.56
ETHV103	105	79	0.75	0.62
ETHV201	112	36.5	0.33	0.28
ETHV202	115	83.2	0.72	0.59
ETHV203	115	86.1	0.75	0.66
TUR1xzy	30	9.16	0.24	0.16
TUR1rpy	28	2.64	0.06	0.88
TUR2xyz	123	9.7	0.06	0.03
TUR2rpy	110	4.49	0.01	0.10
TUR1360	29	7.49	0.21	0.73
TUR1floor	50	14.08	0.26	0.26
TUR1desk	23	10.56	0.41	0.41
TUR1desk2	25	11.46	0.43	0.51
TUR1room	49	17.48	0.33	0.52
TUR2360H	91	17.17	0.16	0.36
TUR2360K	48	15.16	0.3	0.23
TUR2D	99	20.34	0.19	0.11
TUR2LNL	112	10.92	0.24	0.26
TUR2LWL	173	15.13	0.23	0.30
TUR2P360	73	17.28	0.23	0.21
TUR2PS	156	43.08	0.26	0.23
TUR2PS3	112	20.48	0.16	0.22