Beacon-based Scale Estimation for Monocular Structure from Motion in Robotics Systems

Leonardo Ospizio[1], Francesco Riccardo Tassone[1]

[1]*Sapienza, University of Rome*

Abstract

Accurate scale estimation in monocular Structure from Motion (SfM) pipelines plays a crucial role in robotics systems. In this paper, we present a novel method to address the scale ambiguity challenge by strategically leveraging beacons placed at known positions within the environment. Our objective is to propose a new methodology that integrates existing optimization techniques to effectively tackle this problem. We demonstrate the efficacy of our approach by conducting benchmarking experiments and comparing our system with state-of-the-art methods. The results indicate that, in some cases, our method achieves comparable accuracy in scale estimation, highlighting its potential for improving the reliability and precision of SfM-based robotics applications or providing a viable alternative.

Keywords

Scale Estimation, Structure from Motion, Simultaneous Localization and Mapping, Visual Odometry, Scene Reconstruction, Iterative Closest Point Optimization

1. Highlights

In this work we present:

- Extensive research on absolute scale ambiguity and a summary of state-of-the-art recovery approaches.
- A novel approach for estimating the absolute scale in monocular SfM based on known objects in the environment.
- A highly modular pipeline that provides the possibility to adapt the method to various use cases and needs.
- Validation of the system using synthetic and simulated data.
- Comparable results with existing state-of-the-art approaches in scale drift correction.

2. Introduction

Vision-based Structure from Motion (SfM), Visual Odometry (VO), and Simultaneous Localization and Mapping (SLAM) have been employed in many applications, including autonomous driving, scene reconstruction, and object measurement. Compared to other similar methods, vision-based systems offer high performance with low costs and requirements. Indeed, the camera sensor is relatively cheap and can provide useful information that can also be employed for other objectives such as semantic, geometric, and color analysis. However, it is known that these techniques suffer from scale ambiguity when the scene reconstruction is performed using only a single camera. Consequently, we can only recover the structure of the scene and/or the camera motion up to a scale without the possibility of real-world measuring based on the acquired data.

Many techniques have been proposed to overcome this problem, usually relying on other sensors or some domain-dependent constraint. The first process is known as sensor data fusion [1]. Typically, Visual Odometry and Visual SLAM applications use images in combination with wheel odometry [1], GPS [2], inertial measurement unit (IMU) [3], laser sensor [4], sonar [5], LiDAR or directly use a depth camera in place of a monocular camera [6, 7]. Most of these works have shown promising results in robot localization, achieving high accuracy in indoor and outdoor environments. Nevertheless, the use of these sensors leads to higher costs, power consumption, and space requirements that can limit their usage [8, 3]. The second approach adds prior scene knowledge to complete the missing information and add a geometric or physical constraint. These methods solve the problems given by adding sensors while maintaining accurate results. However, the lack of a fixed baseline or external measurement usually makes these methods more sensitive to a phenomenon called scale drift.

Scale drift is an accumulative error present in almost every SfM, VO or SLAM system. Indeed, the data camera trajectory and the map are incrementally reconstructed. In each step of any VO or SLAM pipeline, small errors are generated and they can accumulate over time. This drift may not be a problem in short operations where the system needs to be active for a relatively brief amount of time. However, to achieve large-scale operations the phenomenon must be corrected. To mitigate scale drift, many techniques have been proposed. A common ap-

CEUR Workshop Proceedings (CEUR-WS.org)

proach is to decrease the error given by each step of the pipeline by optimizing each module. Alternatively, it is possible to adopt strategies explicitly designed to reduce the overall error such as Bundle Adjustment (BA) or loop closure [3]. Usually, these procedures are very effective, although they both have some limitations. The Bundle Adjustment is indeed effective in the short/medium term but it still suffers in long-distance scenarios. Also, the loop closure has a major drawback: in many scenarios, the robot may not visit the same place twice creating a loopless trajectory.

In this paper, we propose an approach for recovering the absolute scale using only a monocular camera and a receiver in combination with identifiable transmitters sparse in the environment. In the following, we will refer to these transmitters as beacons. By beacon, we mean every object in the scene that can be uniquely identified in the environment and is capable of communicating its position to the robot. Suppose that a robot is set up as described above and is navigating the environment with a map known up to a scale factor. Our goal is to exploit the position information received by the beacons in order to recover the scale. The communication step between the robot and beacons can be performed in any possible way, provided that the beacons are distinguishable in the image captured by the robot and that they provide accurate position information or at least noise-bounded data. This system was tested using the living room scene from the Imperial College London and the National University of Ireland Maynooth (ICL-NUIM) dataset [9], which is a synthetic dataset accurately designed for evaluating Visual Odometry, 3D reconstruction, and SLAM algorithms.

This paper is organized as follows; first, a summary of the existing scale estimation algorithm is presented in Section 3. Section 4 introduces our pipeline and illustrates each building block. In the end, the obtained results are commented on and compared with the state-of-the-art scale estimation methods. Finally, a discussion about possible future works is presented.

3. Related Works

The estimation of the absolute scale for SfM, VO, and SLAM has been widely explored by the community. In this section, we briefly discuss the latest works on scale estimation for monocular camera systems. In particular, we cover two common approaches that have been used.

Prior-knowledge based methods use additional information about the scene to estimate the scale. These approaches rely on different known constraints such as the height of the camera, the size of objects, or particular physical properties. In the first case, the camera height is estimated by detecting the ground plane in the scene. This can be achieved by fitting the ground plane from a group of reconstructed 3D points [10], decomposing the 2D homography matrix [11], or matching the dense ROI between two frames [12, 13]. In the second case, the dimensions of known elements are used to infer the scale of the scene [14]. For instance, Knorr et al. [15] proposed using the size of human faces and comparing the average dimensions of a face with the detected one. In the last case, the problem can be solved by considering some physical properties of the system [16]. For example, Scaramuzza et al. [17] used the nonholonomic constraint given by a wheeled vehicle to reduce the DOFs of the camera trajectory when the vehicle turns. Another solution is given by the work of Mishima et al. [18] in which they use a camera that can zoom and focus on each shot independently. In this way, it is possible to couple the scene structure with the focused distance. The main disadvantage of these methods is that the assumptions may only be valid in a strict domain or may not be very accurate. For instance, camera height approaches can only be applied if it is possible to detect a ground plane, which can be challenging for noisy frames, crowded scenes, or even impossible if the camera is moving in free space. Similarly, the object dimensions methods usually rely on statistical information that can lead to inaccurate estimates.

Multi-sensor based methods use additional data sources, such as LiDARs, sonar, or GPS, to recover the absolute scale. In general, these methods can achieve great performance, both in terms of accuracy and precision. However, many multi-sensor solutions suffer from high hardware costs, power consumption, or space requirements, and many works aim to overcome these issues. In [4], the classic LiDAR has been replaced with a single-ray Laser Range Finder to drastically reduce hardware costs and space. Another example is the work of Nikolov et al. [2], who used GPS to correct the scale drift of UAVs and outdoor environments. The multi-sensor approach can also be useful in domains where it is difficult to have prior knowledge. For instance, in underwater environments, it is not possible to use any ground plane method or object size, but a possible solution is given by Yang et al [5]. using a 2D imaging sonar to retrieve the 3D geometrical constraints.

In this paper, we propose a prior-knowledge approach to compute the absolute scale from a monocular camera. Our method shows that it is feasible to recover the scale using points in the environment where the 3D location is known. By using beacon objects, we can pinpoint the location of the detected object in the scene and match it with the ambient map. The advantage is that it can be implemented in many indoor and outdoor environments without the need for expensive hardware or particular constraints.

4. Implementation

The model architecture uses three different components (see Figure 1). The SfM module produces a map of the scene and localizes the camera given the input images. The Detection and Classification module uses the images to identify the beacon location, which will then be used by the Scale Estimator module. We kept the three components separate for two main reasons. First, this separation allows us to easily develop and test each block. Second, we aim to create a modular pipeline in which the performance of different SfM and Detection algorithms can be compared.

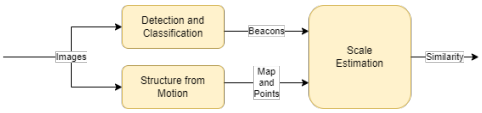

Figure 1: An overview of the model architecture is provided, showcasing the three building blocks along with their corresponding input and output.

Starting from the images taken by the monocular camera and the camera intrinsic parameters, the SfM module constructs a preliminary map of the scene. The map contains the camera poses and the three-dimensional points that were relevant in the map construction. The set of camera extrinsic parameters and map points is collected and will be used later for recovering the scale. During the Detection and Classification stage, the same images used for mapping are segmented to extract Regions of Interest (ROIs) where beacons may be present. Figure 2 shows the segmented areas and key points. The key points will then be used to perform the association with the three-dimensional beacon positions. Once the beacons have been identified, their positions are matched with the three-dimensional points inside the map. The data association is trivial since both the three-dimensional points extracted by the SfM module and the three-dimensional sensor information given by the beacons are associated one-to-one with the key points in the image. In other words, to perform the association is sufficient to take the three-dimensional map points with their two-dimensional image projection inside the ROIs.

Finally, the Scale Estimator takes the three-dimensional points in the map with their true three-dimensional locations in world coordinates, and the Iterative Closest Point optimization is performed.

4.1. Scene Mapping

One of the main components of our architecture is the model used for performing Structure from Motion. For the initial estimation of the scene map and camera poses, we decided to use COLMAP [19], an open-source SfM

Figure 2: The output of the segmentation module. The images have been thoroughly segmented by the model. In the first case (left), only two objects (i.e. painting, door) will be considered in the following steps. While in the second case (right), the pipeline will consider more object (i.e. vase, cabinet, chair, ..).

software written in C++ and designed to provide a complete reconstruction pipeline from unordered images. Although this model is recommended for unordered images, its robustness and flexibility make it suitable for our case as well.

SfM pipelines typically involve a correspondence search phase, which consists of three main steps: Feature Extraction, Feature Matching, and Geometric Verification. During the Feature Extraction stage, the input images are used to extract a collection of local features that describe the points of interest in the images (key points). Feature Matching determines the common parts of the scene in different images by utilizing the features and key points extracted in the previous step. Finally, Geometric Verification is performed to validate the matches, taking into account the geometric constraints of the environment in addition to the feature appearance and similarity [20].

We utilized the code provided by Paul-Edouard Sarlin et al. [21] in the Hierarchical Localization work, which integrates the Python binding of COLMAP with various Feature Descriptors and Feature Matchers for 3D Reconstruction. We customized the reconstruction options by setting the pinhole camera model and using the camera intrinsics provided in our dataset. Additionally, we fixed the camera parameters refinement flags to prevent the Bundle Adjustment (BA) procedure from modifying the focal length, principal points, and other intrinsic camera parameters during optimization. The correspondence search phase is performed using two architectures: Superpoint [22] for Feature Extraction and Superglue [23] for the Matching phase. Superpoint is a fully-convolutional neural network that produces interest point detections accompanied by fixed-length descriptors from full-sized images. The model architecture consists of an encoder that reduces the input image dimension, shared by two parallel decoders trained for interest point detection and description, respectively. The second module, Superglue, is specifically designed for Feature Matching. It matches local features across images by finding correspondences and rejecting non-matchable points. It comprises two

major blocks: an Attentional Graph Neural Network that computes matching descriptors based on an initial set of local features, and an Optimal matching layer that produces a partial assignment matrix using the computed matching descriptors.

4.2. Beacon Detection

The detection and identification of beacons within the scene play a crucial role in our system for performing scale estimation. It is essential to recognize the beacons in the image to establish correspondences between their positions in the world and the keypoint locations on the map. We opted to utilize a segmentation model due to its ability to provide higher precision compared to an object tracker, despite its relatively slower performance.

In this study, we employed the Masked-attention Mask Transformer model (Mask2Former), which is a universal image segmentation architecture introduced in [24]. This model proposes several modifications to an existing architecture [25], which consists of a backbone feature extractor, a pixel decoder, and a transformer decoder. The authors have appropriately enhanced these three building blocks to achieve superior results compared to previous state-of-the-art models in various segmentation tasks. The main enhancements of this architecture include the utilization of masked attention in the Transformer decoder, the incorporation of multi-scale high-resolution features, and certain optimization improvements that enhance the model's accuracy without increasing computational requirements.

The segmentation module was configurated to segment all parts of the input image, although only relevant regions are considered during scale optimization. Specifically, we exclusively utilize objects within the scene that remain fixed during the localization phase (e.g., furniture, windows, doors, etc.). However, the number and type of objects can be properly defined during the model initialization.

4.3. Scale Estimation

The last component of our system is the module designed to solve the scale ambiguity of the estimated reconstruction. The Iterative Closest Point (ICP) optimization is one of the main techniques used for point cloud registration in various applications in the robotics field and 3D reconstruction [26]. The literature presents several variants of ICP that aim to handle outliers, improve convergence speed, or minimize different error functions [27, 28].

We have adopted a classical least squares optimization approach that utilizes the positions of the beacons in the scene and their corresponding estimated positions to recover a similarity transformation that aligns the two point clouds. Drawing inspiration from the ICP optimization techniques described in [29], we have implemented two main ICP schemes. The first scheme performs point registration using two three-dimensional point clouds to recover a similarity transformation involving scale, rotation, and translation. The second scheme recovers a rigid transformation comprising rotation and translation. Both algorithms utilize the Gauss-Newton iterative method to minimize the sum of the L2 Omega norm function. More formally, the method seeks to minimize the following function:

$$f = \sum_i e_i^T \Omega e_i \tag{1}$$

with

$$e_i = \|p_i - \hat{p}_i\|_\Omega^2 \tag{2}$$

being the squared norm of the difference between the prediction p_i and the measurement $\hat{p}_i$, weighted by the Omega matrix Ω, which is the identity matrix in our case. It is important to note that different error functions can be employed for the minimization process. The primary distinction between the two implementations lies in the number of estimated parameters. For the rigid registration problem, six parameters are estimated: three parameters for the translation vector (t_x, t_y, t_z) and three parameters for the rotation matrix (α, β, γ). They represent the minimal representation of a rigid transformation, up to a singularity for the rotation. Regarding the similarity registration, an additional parameter is needed to define the similarity transformation, which is the scale factor s. Therefore, a total of seven parameters are estimated: $(t_x, t_y, t_z, \alpha, \beta, \gamma, s)$.

ICP schemes are typically sensitive to initialization. An initial guess has to be provided to the algorithm in order to start the optimization. We have chosen to use the closed-form solution to the problem [30] to have a good initial guess to start the optimization.

5. Results

We have individually tested the functionality of each module in the pipeline to assess the impact of each building block on the final error. The evaluation of scene reconstruction was done using the Root Mean Square Error (RMSE), with the Absolute Translation Error (ATE) specifically used to measure the translation difference between the ground-truth position and our estimation [30].

Root Mean Square Error. Let $P \subseteq \mathbb{R}^3$ be a three-dimensional point cloud and $\hat{P} \subseteq \mathbb{R}^3$ be the estimated point cloud, then the Root Mean Square Error is defined as

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \|p_i - \hat{p}_i\|^2} \tag{3}$$

where $p_i \in P$ and $\hat{p}_i \in \hat{P}$.

Absolute Translation Error. Let $\hat{X}_i$ and X_i be the i-th ground-truth pose and the i-th pose estimation expressed as homogeneous transformations. And let $R_i, \hat{R}_i$ be the rotation matrices and $p_i, \hat{p}_i$ be the translation vectors of the respective poses, then

$$ATE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\Delta p_i\|^2} \tag{4}$$

where $\Delta p_i = p_i - \Delta R_i \hat{p}_i$ and $\Delta R_i = R_i(\hat{R}_i)^T$.

Validation on generated point clouds. we first validated the scale estimation algorithm using simulation data. We developed a simulator to generate point clouds of various shapes (e.g. sphere, toroid, cuboid) and point cloud measurements with different noise levels. The results of the ICP optimization are shown in the Figure 3, while the Table 1 summarizes the RMSE values. The algorithm performs very well when the correspondences are known and the noise is limited.

Table 1
Results of the Point Clouds Registration with progressively increasing noise levels. The Root Mean Square Error (RMSE) is evaluated post the optimization of the Similarity Iterative Closest Point (SICP) algorithm. A few iterations of the Rigid Iterative Closest Point (RICP) method allow for further refinement of the SICP solution.

Gaussian Noise (μ, σ)	SICP	SICP+RICP
0, 0.01	6.48	0.01
0, 0.1	6.49	0.16
0, 1	6.77	1.73
0, 5	8.67	6.50

Validation on Ground truth trajectories. As an additional evaluation method, we chose to test the scale estimation module by using ground-truth camera poses instead of beacon positions. This serves two purposes: firstly, to demonstrate the effectiveness of the ICP method when camera locations are available, and secondly, to assess the accuracy of the trajectory estimated by the SfM model. Figure 4 illustrates how the method can directly utilize position information (e.g., GPS measurements) for performing ICP optimization. It can be observed that the error is minimized, and the pose registration is influenced by the reconstruction error. It should be noted that we assume the camera trajectory is known up to a similarity transformation, and if the SfM module provides incorrect camera poses, there is little we can do. The results of the optimization are presented in Table 2.

Our results. After validation, we continued the analysis of the entire pipeline using the beacon locations.

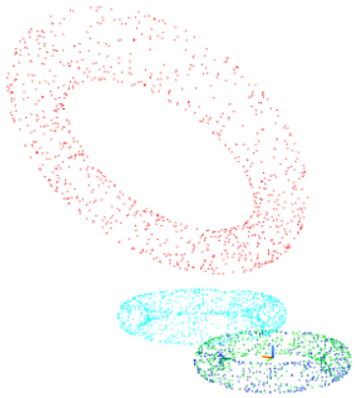

Figure 3: Optimization results for a toroidal point cloud: a torus is generated (depicted in blue), and corresponding random measurements (red) are produced using white Gaussian noise with μ=0 and σ=0.1. Initial SICP optimization recovers the scale (shown in cyan), followed by a few iterations of RICP optimization to refine the solution (represented in green).

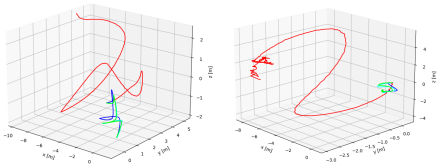

Figure 4: A comparison between the trajectory before and after optimization is shown. The estimated trajectory is represented in red, while the ground truth trajectory is displayed in blue. After the scale correction, the estimated trajectory is rescaled using only a similarity transformation, shown in cyan, and both the similarity and the rigid transformation depicted in green.

We simulated the signals received by the beacons using the depth information obtained from the RGB-D camera. Since the dataset is synthetic, we have access to noise-free depth data. The results generated by the pipelines are presented in Table 3. It is evident that the SICP optimization provides a good estimation of the scene scale. Subsequent iterations of RICP further reduce the Absolute Translation Error in most cases. To illustrate the convergence of our system, we plotted the ATE (Figure 5) and scale (Figure 5) for test #1. This behavior is consistent across all the tests we conducted. Additionally, we included a histogram (Figure 6) showing the number of extracted key points, segmented points, and those used for optimization. This emphasizes that our system utilizes a small amount of data compared to what is collected during the segmentation procedure, streamlining the optimization process.

Table 2
The results of the ICP optimization using the ground truth poses are presented. The performance is assessed based on the translation error percentage and the Absolute Trajectory Error in terms of mean and standard deviation. The errors are computed after the SICP optimization (column 3) and further improved through a few iterations of RICP optimization (columns 4 and 5). Additionally, the trajectory length is included for comprehensive analysis.

Test [#]	Sequence	SICP (μ, σ) [m]	SICP + RICP (μ, σ) [m]	T(%)	Path length [m]
1	001	0.54, 0.39	0.09, 0.04	9.83	0.95
2	002	0.32, 0.18	0.32, 0.18	9.04	3.54
3	000-299	0.50, 0.19	0.5, 0.18	8.56	5.83
4	300-599	0.24, 0.11	0.32, 0.09	14.67	2.20

Table 3
Results of our test in terms of mean and standard deviation of Absolute Trajectory error and translation error percentage with respect to the trajectory length. The errors are computed after SICP optimization (column 3), and after SICP and few iteration of RICP optimization (column 4, 5).

Test [#]	Sequence	SICP (μ, σ) [m]	SICP + RICP (μ, σ) [m]	ATE [%]	Path length [m]
1	001	0.1509, 0.0987	0.0907, 0.0683	9.56	0.95
2	002	0.4457, 0.1709	0.4230, 0.1846	11.95	3.54
3	000-299	0.5833, 0.2683	0.5173, 0.2486	8.87	5.83
4	300-599	0.8427, 0.3818	0.6627, 0.2595	30.17	2.20

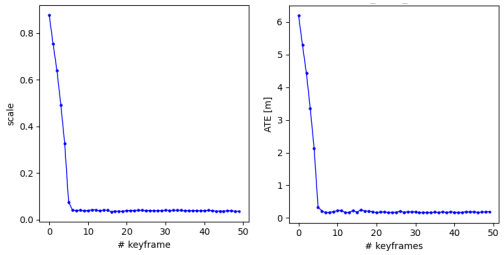

Figure 5: The scale factor value (on the left) and the Absolute Translation Error (on the right) are presented across the keyframes. Both values converge within fewer than 10 keyframes and subsequently stabilize.

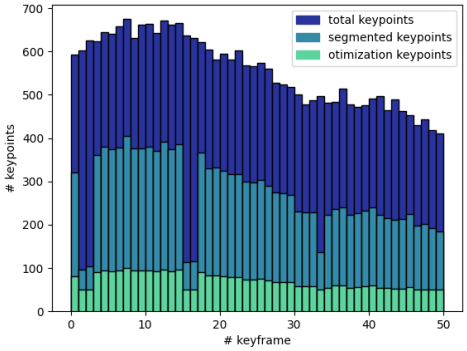

Figure 6: The number of keypoints extracted from the the keyframes, those that are been segmented and those that are used by the scale estimator are reported in blue, light blue and light green respectively. As it is clear from the picture, a minimum of 50 key points are sampled from the segmented areas and then used for optimization.

Comparison. We compared our system with other state-of-the-art (SOTA) SfM pipelines. The systems we considered were evaluated using the Kitti dataset, which is widely used for mobile robotics and autonomous driving [31]. However, we were unable to benchmark our system using Kitti due to the absence of beacons in the dataset and the inability to simulate beacons due to noisy LiDAR measurements in real-world scenarios. Moreover, it is important to note that our dataset is indoor while theirs is outdoor. Additionally, their trajectories are much longer, around 1000m, compared to our 20m trajectory. However, their trajectories are more stable as they are recorded in road driving scenarios and primarily involve two coordinates, whereas ours is more complex and evolves in all coordinates. Consequently, we provided both the Translation Error Percentage and the Absolute Trajectory Error, but the comparison primarily focuses on the Translation Error Percentage.

Our system achieves comparable results with [32, 33] and [34] (10.43%). E. Sucar et al. [32] (9.75%) applies an online algorithm for estimating the scale correction, to the output of a monocular SLAM system. They use a deep-learning-based object detector and a prior knowledge on the evolution of the scale drift with a predefined prior of the heights of the detected object class. B. Lee at al. instead, recover the true scene scale by estimating the

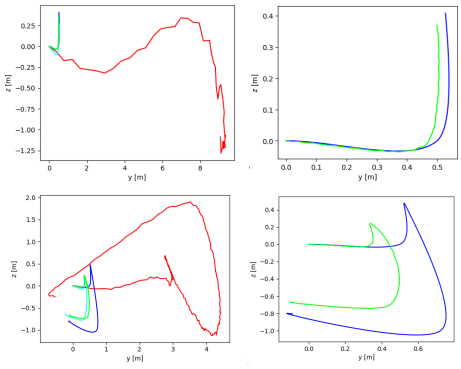

Figure 7: The results of the scale drift correction on sequence 001 and sequence 002 are depicted. The SfM trajectory estimation and the ground-truth trajectory are represented by red and blue lines, respectively. The estimation undergoes rescaling after the SICP optimization (indicated in cyan) and is subsequently refined through iterations of RICP (illustrated in green).

Table 4
Comparison of the SOTA results with respect to ours. The Absolute Translation Error (ATE) with the Translation Error Percentage are reported.

Methods	T(%)	ATE [m]
B. Lee at al. [33]	6.86	n.a
A. Geiger et al. [34]	10.43	n.a
S. Song et al.[36, 12]	2.03	n.a
N. Fanani et al. [37]	1.54	n.a
X. Wang et al. [38]	1.25	6.17
X. Yin et al. [39]	2.68	n.a
E. Sucar et al. [32]	9.75	n.a
S. Yang et al.[40]	1.78	8.91
Ming Fan et al. [35]	1.31	2.75
OURS	8.56	0.51

ground surface through an image classifier [33] (6.86%).

It is worth mentioning that the approach proposed by Ming Fan et al. is similar to ours [35]. They exploit a geometric assumption on the normal vector of the ground plane combined with road segmentation to correct scale drift. This method is very indicative for comparison since we also extract semantic information from the scene using segmentation. They test two segmentation modules, SEGM and GCSEMK, and achieve excellent results with both (1.54%, 1.31%).

Discussion. We argue that the scale estimation algorithm is not the main source of system error. As demonstrated in the validation section, the ICP optimization should be capable of recovering scale drift even in the presence of noise or incorrect correspondences. We believe that there are two primary reasons contributing to the error: the beacon locations obtained from the depth sensor and the scene reconstruction module. As mentioned earlier, we utilize the depth camera to simulate beacon signals for position information. While we use noise-free RGB-D data, the sensor readings accumulate drift over time. Although sensor drift is typically handled by RGB-D SLAM systems during the optimization process, this is beyond the scope of our work. We have determined that a portion of the error can be attributed to sensor drift. Additionally, regarding the Structure from Motion module, we have observed that the camera trajectory estimated by the system is not very accurate. In some cases, there is a significant mismatch between the reconstruction and the ground truth trajectory.

6. Future Works

During the development of the project, we followed a precise direction towards the resolution of the problem. However, other more sophisticated details could be added to make the system more robust to uncertainty, noise, outliers, and other problems that could arise in real-world scenarios. We drafted a list of improvements and modifications to the existing work.

Weighted Least Squares Optimization. The use of segmentation to identify and classify portions of the scene is not an error-free step in the pipeline. Typically, two main situations may occur at that stage. Firstly, the segmented area may be classified wrongly, and/or the border of the segmented object may not be well defined, resulting in imprecision. This could cause two problems. One problem is that the position of another beacon is assigned to the given area, but this issue is not too relevant because a more robust identification procedure can be defined for the beacons. The other possibility is that some map points are not considered as part of a segmented beacon inside the image. We believe that the second problem can be more problematic. By construction, the visual SfM pipeline exploits the object's border in the images as key points, which is a natural consequence of how the feature extractor is constructed. This makes the probability with which a map point is obtained from the projection of the object's edge in the images quite high. Consequently, we need to take care of object borders and try to avoid this type of situation. A possible solution to this problem could be to expand the border of the segmented area by a few pixels. This way, we hope that at least the majority of the point will be contained in the segmented area. However, we cannot assume that this region perfectly matches the true beacon location in the image. For this reason, a suitable option for the scale estimation step would be to provide a weighted version of the ICP optimization algorithm that assigns less weight to the points contained in that region. An example of what we mean

by weighted ICP can be found in [41], and other variants of the ICP optimization are mentioned in [42].

Object Detection and Tracking. Using a segmentation model leads to precise results in terms of beacon identification compared to other detection architectures. However, segmentation models are typically slower than object detectors (e.g., YOLO), which produce bounding boxes instead of segmentation masks. Using an Object Tracker instead of segmentation would improve performance in terms of speed, but it may potentially decrease the overall system accuracy.

Beacons point clouds. What we find the most interesting idea is to use beacons that transmit signals containing point clouds instead of a single point location. Initially, we assumed that the beacons' signals contained only a single three-dimensional location of the beacons in the environment. However, we have relaxed this constraint by considering multiple points within the segmented region in the image to recover scale. Utilizing a dense point cloud corresponding to the shape of the beacon would be a suitable direction to strike a balance between the amount of information, execution time, and accuracy. It's important to note that the system will still process relatively small amounts of data compared to the existing SLAM pipeline.

7. Conclusion

Our proposed system has demonstrated its effectiveness in addressing the scale estimation problem. Although our achieved accuracy is comparable to some existing approaches, it does not reach the state-of-the-art results in general. However, it is important to emphasize that our system utilizes significantly less data compared to complete SLAM systems, which typically rely on multiple sensors to mitigate the scale problem. Unlike similar methods, our approach does not rely on hand-crafted information such as camera heights or assumptions about object dimensions. Instead, we leverage beacon position information, which can be easily integrated into the environment, particularly in industrial scenarios but not limited to them.

The limitation of our system's accuracy appears to be associated with the Structure from Motion (SfM) module used. Exploring alternative SLAM models could be a key area for further development, facilitated by the modularity of our pipeline. Our primary objective was to design the system in a way that allows for adaptability, including the incorporation of new datasets, benchmarking with different metrics, and substituting the SfM module, along with the detection and classification components, with other state-of-the-art models.

Moving forward, the next phase of our work will involve the creation of a reliable dataset to evaluate the system on real-world data, providing beacon position measurements from transmitters. This will introduce new challenges regarding data correspondence, beacon detection, and identification.

In conclusion, our system presents a promising approach to address scale estimation in monocular SfM pipelines, and future efforts will focus on further improving accuracy through the exploration of alternative SLAM models and the development of robust benchmark datasets for real-world evaluations.

References

[1] A. Ligocki, A. Jelínek, Fusing the rgbd slam with wheel odometry, IFAC-PapersOnLine 52 (2019) 7–12. URL: https://www.sciencedirect.com/science/article/pii/S2405896319326734. doi:https://doi.org/10.1016/j.ifacol.2019.12.724, 16th IFAC Conference on Programmable Devices and Embedded Systems PDES 2019.

[2] I. A. Nikolov, C. B. Madsen, Performance characterization of absolute scale computation for 3d structure from motion reconstruction, in: A. Trémeau, G. M. Farinella, J. Braz (Eds.), Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP 2019, Volume 5: VISAPP, Prague, Czech Republic, February 25-27, 2019, SciTePress, 2019, pp. 884–891. URL: https://doi.org/10.5220/0007444208840891. doi:10.5220/0007444208840891.

[3] I. Abaspur Kazerouni, L. Fitzgerald, G. Dooly, D. Toal, A survey of state-of-the-art on visual slam, Expert Systems with Applications 205 (2022) 117734. URL: https://www.sciencedirect.com/science/article/pii/S0957417422010156. doi:https://doi.org/10.1016/j.eswa.2022.117734.

[4] Z. Zhang, R. Zhao, E. Liu, K. Yan, Y. Ma, Scale estimation and correction of the monocular simultaneous localization and mapping (slam) based on fusion of 1d laser range finder and vision data, Sensors 18 (2018). URL: https://www.mdpi.com/1424-8220/18/6/1948. doi:10.3390/s18061948.

[5] D. Yang, H. Ai, J. Liu, B. He, Absolute scale estimation for underwater monocular visual odometry based on 2-d imaging sonar, Measurement 190 (2022) 110665. URL: https://www.sciencedirect.com/science/article/pii/S0263224121015293. doi:https://doi.org/10.1016/j.measurement.2021.110665.

[6] C. Kerl, J. Sturm, D. Cremers, Dense visual slam for rgb-d cameras, in: Proc. of the Int. Conf. on Intelligent Robot Systems (IROS), 2013.

[7] C. Kerl, J. Sturm, D. Cremers, Robust odometry

estimation for rgb-d cameras, in: International Conference on Robotics and Automation (ICRA), 2013.

[8] M. Aqel, M. H. Marhaban, M. I. Saripan, N. Ismail, Review of visual odometry: types, approaches, challenges, and applications, SpringerPlus 5 (2016). doi:10.1186/s40064-016-3573-7.

[9] A. Handa, T. Whelan, J. McDonald, A. Davison, A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM, in: IEEE Intl. Conf. on Robotics and Automation, ICRA, Hong Kong, China, 2014.

[10] M. Fan, S.-W. Kim, S.-T. Kim, J.-Y. Sun, S.-J. Ko, Simple but effective scale estimation for monocular visual odometry in road driving scenarios, IEEE Access 8 (2020) 175891–175903. doi:10.1109/ACCESS.2020.3026347.

[11] D. Zhou, Y. Dai, H. Li, Reliable scale estimation and correction for monocular visual odometry, in: 2016 IEEE Intelligent Vehicles Symposium (IV), 2016, pp. 490–495. doi:10.1109/IVS.2016.7535431.

[12] S. Song, M. Chandraker, Robust scale estimation in real-time monocular SFM for autonomous driving, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014, IEEE Computer Society, 2014, pp. 1566–1573. URL: https://doi.org/10.1109/CVPR.2014.203. doi:10.1109/CVPR.2014.203.

[13] D. Zhou, Y. Dai, H. Li, Ground plane based absolute scale estimation for monocular visual odometry, 2019. arXiv:1903.00912.

[14] D. Frost, V. Prisacariu, D. Murray, Recovering stable scale in monocular slam using object-supplemented bundle adjustment, IEEE Transactions on Robotics 34 (2018) 736–747. doi:10.1109/TRO.2018.2820722.

[15] S. B. Knorr, D. Kurz, Leveraging the user's face for absolute scale estimation in handheld monocular SLAM, in: W. Broll, H. Saito, J. E. S. II (Eds.), 2016 IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2016, Merida, Yucatan, Mexico, September 19-23, 2016, IEEE Computer Society, 2016, pp. 11–17. URL: https://doi.org/10.1109/ISMAR.2016.20. doi:10.1109/ISMAR.2016.20.

[16] L. Li, H. Lan, Recovering absolute scale for structure from motion using the law of free fall, Optics Laser Technology 112 (2019) 514–523. URL: https://www.sciencedirect.com/science/article/pii/S003039921830224X. doi:https://doi.org/10.1016/j.optlastec.2018.11.045.

[17] D. Scaramuzza, F. Fraundorfer, M. Pollefeys, R. Siegwart, Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints, in: IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009, IEEE Computer Society, 2009, pp. 1413–1419. URL: https://doi.org/10.1109/ICCV.2009.5459294. doi:10.1109/ICCV.2009.5459294.

[18] N. Mishima, A. Seki, S. Hiura, Absolute scale from varifocal monocular camera through sfm and defocus combined, in: 32nd British Machine Vision Conference 2021, BMVC 2021, Online, November 22-25, 2021, BMVA Press, 2021, p. 28. URL: https://www.bmvc2021-virtualconference.com/assets/papers/0287.pdf.

[19] J. L. Schönberger, J.-M. Frahm, Structure-from-Motion Revisited, in: Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[20] S. Bianco, G. Ciocca, D. Marelli, Evaluating the performance of structure from motion pipelines, Journal of Imaging 4 (2018). URL: https://www.mdpi.com/2313-433X/4/8/98. doi:10.3390/jimaging4080098.

[21] P.-E. Sarlin, C. Cadena, R. Siegwart, M. Dymczyk, From coarse to fine: Robust hierarchical localization at large scale, 2019. arXiv:1812.03506.

[22] D. DeTone, T. Malisiewicz, A. Rabinovich, Superpoint: Self-supervised interest point detection and description, 2018. arXiv:1712.07629.

[23] P.-E. Sarlin, D. DeTone, T. Malisiewicz, A. Rabinovich, SuperGlue: Learning feature matching with graph neural networks, in: CVPR, 2020.

[24] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, R. Girdhar, Masked-attention mask transformer for universal image segmentation, CoRR abs/2112.01527 (2021). URL: https://arxiv.org/abs/2112.01527. arXiv:2112.01527.

[25] B. Cheng, A. Schwing, A. Kirillov, Per-pixel classification is not all you need for semantic segmentation, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan (Eds.), Advances in Neural Information Processing Systems, volume 34, Curran Associates, Inc., 2021, pp. 17864–17875. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/950a4152c2b4aa3ad78bdd6b366cc179-Paper.pdf.

[26] J. Zhang, Y. Yao, B. Deng, Fast and robust iterative closest point, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (2020) 3450–3466.

[27] K.-L. Low, Linear least-squares optimization for point-to-plane icp surface registration, 2004.

[28] S. Du, Y. Xu, T. Wan, H. Hu, S. Zhang, G. Xu, X. Zhang, Robust iterative closest point algorithm based on global reference point for rotation invariant registration, PLOS ONE 12 (2017) 1–14. URL: https://doi.org/10.1371/journal.pone.0188039. doi:10.1371/journal.pone.0188039.

[29] G. Grisetti, T. Guadagnino, I. Aloise, M. Colosi, B. D. Corte, D. Schlegel, Least squares optimization: from

theory to practice, 2020. arXiv:2002.11051.
[30] Z. Zhang, D. Scaramuzza, A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 7244–7251. doi:10.1109/IROS.2018.8593941.
[31] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: The kitti dataset, The International Journal of Robotics Research 32 (2013) 1231–1237. URL: https://doi.org/10.1177/0278364913491297. doi:10.1177/0278364913491297. arXiv:https://doi.org/10.1177/027836491349129[illegible]elective_artificial_intelligence_1
[32] E. Sucar, J.-B. Hayet, Bayesian scale estimation for monocular slam based on generic object detection for correcting scale drift, 2018 IEEE International Conference on Robotics and Automation (ICRA) (2017) 1–7.
[33] B. Lee, K. Daniilidis, D. Lee, Online self-supervised monocular visual odometry for ground vehicles, Proceedings - IEEE International Conference on Robotics and Automation 2015 (2015) 5232–5238. doi:10.1109/ICRA.2015.7139928.
[34] A. Geiger, J. Ziegler, C. Stiller, Stereoscan: Dense 3d reconstruction in real-time, 2011, pp. 963 – 968. doi:10.1109/IVS.2011.5940405.
[35] M.-Y. Fan, S.-W. Kim, S.-T. Kim, J.-Y. Sun, S.-J. Ko, Simple but effective scale estimation for monocular visual odometry in road driving scenarios, IEEE Access 8 (2020) 175891–175903.
[36] S. Song, M. Chandraker, C. Guest, High accuracy monocular sfm and scale correction for autonomous driving, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (2015) 1–1. doi:10.1109/TPAMI.2015.2469274.
[37] N. Fanani, A. Sturck, M. Barnada, R. Mester, Multimodal scale estimation for monocular visual odometry, 2017 IEEE Intelligent Vehicles Symposium (IV) (2017) 1714–1721.
[38] X. Wang, H. Zhang, X. Yin, M. Du, Q. Chen, Monocular visual odometry scale recovery using geometrical constraint, 2018 IEEE International Conference on Robotics and Automation (ICRA) (2018) 988–995.
[39] X. Yin, X. Wang, X. Du, Q. Chen, Scale recovery for monocular visual odometry using depth estimated with deep convolutional neural fields, 2017 IEEE International Conference on Computer Vision (ICCV) (2017) 5871–5879.
[40] S. Yang, S. A. Scherer, Cubeslam: Monocular 3-d object slam, IEEE Transactions on Robotics 35 (2018) 925–938.
[41] Y. Guo, L. Zhao, Y. Shi, X. Zhang, S. Du, F. Wang, Adaptive weighted robust iterative closest point, Neurocomputing 508 (2022) 225–241. URL: https://www.sciencedirect.com/science/article/pii/S0925231222010323. doi:https://doi.org/10.1016/j.neucom.2022.08.047.
[42] S. Rusinkiewicz, M. Levoy, Efficient variants of the icp algorithm, Proceedings Third International Conference on 3-D Digital Imaging and Modeling (2001) 145–152.

8. Online Resources

The source code of this work is available at: