Computational Visual Media
https://doi.org/10.1007/s41095-022-0329-x Vol. 10, No. 2, April 2024, 215–241

Research Article

Benchmarking visual SLAM methods in mirror environments

Peter Herbert[1], Jing Wu[1] (✉), Ze Ji[2], and Yu-Kun Lai[1]

Abstract Visual simultaneous localisation and mapping (vSLAM) finds applications for indoor and outdoor navigation that routinely subjects it to visual complexities, particularly mirror reflections. The effect of mirror presence (time visible and its average size in the frame) was hypothesised to impact localisation and mapping performance, with systems using direct techniques expected to perform worse. Thus, a dataset, MirrEnv, of image sequences recorded in mirror environments, was collected, and used to evaluate the performance of existing representative methods. RGBD ORB-SLAM3 and BundleFusion appear to show moderate degradation of absolute trajectory error with increasing mirror duration, whilst the remaining results did not show significantly degraded localisation performance. The mesh maps generated proved to be very inaccurate, with real and virtual reflections colliding in the reconstructions. A discussion is given of the likely sources of error and robustness in mirror environments, outlining future directions for validating and improving vSLAM performance in the presence of planar mirrors. The MirrEnv dataset is available at https://doi.org/10.17035/d.2023.0292477898.

Keywords visual simultaneous localisation and mapping (vSLAM); mirror; localisation; mapping; reflection; dataset

1 Introduction

Simultaneous localisation and mapping (SLAM) is widely used in autonomous navigation tasks: it uses sensors to build a map of an environment, whilst also determining the sensor's position within that environment. SLAM algorithms can make use of various sensors: for example, visual SLAM (vSLAM) uses one or more cameras to provide image input [1]. Other SLAM algorithms may use different sensors, such as LiDAR, sonar, and IMUs (inertial measurement units), in combination with cameras, for, e.g., visual-inertial SLAM [2–4] and visual-LiDAR SLAM [5, 6]. Since cameras are cheap and consume little power, yet also provide high-quality information [7], visual SLAM remains an active area of research. The literature covers various camera arrangements (monocular, stereo, RGB-depth) and processing methods (e.g., dense versus sparse, direct versus indirect, use of machine learning), as discussed in Refs. [8, 9]. Estimation of camera pose from images can be hindered by a range of environmental circumstances [8, 10]. Many visual complexities have been considered in the literature, and methods proposed to overcome specific difficulties: motion blur [11–13], illumination change [14, 15], dynamic scenes [16, 17], textures [18–20], indoor/outdoor transitions, and specular highlights [18, 21]. General approaches to tackle complexities have recently been proposed [18, 22], but do so indiscriminately of the source of errors.

One problem that has been largely overlooked is the presence of *mirror reflections* in real-world environments. One of the reasons that planar mirrors cause difficulties in computer vision is that reflections are typically indistinguishable from their real counterparts without context. Mirrors provide alternative viewpoints for an environment, allowing light to be redirected around corners or behind occlusions. Reflections are ubiquitous in many domestic and industrial settings, and are often utilised by the human visual system to help

1 School of Computer Science and Informatics, Cardiff University, Abacws Building, Senghennydd Rd, Cardiff CF24 4AG, UK. E-mail: P. Herbert, Herbertp1@cardiff.ac.uk; J. Wu, WuJ11@cardiff.ac.uk (✉); Y.-K. Lai, LaiY4@cardiff.ac.uk.

2 School of Engineering, Cardiff University, Queen's Buildings, The Parade, Cardiff CF24 3AA, UK. E-mail: JiZ1@cardiff.ac.uk.

Manuscript received: 2022-07-27; accepted: 2022-12-18

people understand their surroundings. On the other hand, computer vision systems have struggled to recognise reflective surfaces and correctly understand the environment's geometry—this has only recently begun to be addressed [23, 24]. Filtering out reflections means discarding information that has the potential to extend a camera's coverage of an environment or object. Ideally, this information should be extracted and utilised to improve the reconstruction output of SLAM methods.

In some navigation tasks, collisions with mirrors have been avoided by depending on additional sensors, such as sonar or multi-echo LiDAR [25, 26]. However, consider Fig. 1, which shows a map created by an autonomous vacuum cleaner using camera and LiDAR sensors, deployed in a room with a large domestic mirror. The occupancy grid mistakenly presents the view through the mirror as a real space to be entered and cleaned. Whilst additional sensors may be an engineering solution, determining which types of sensors, in what arrangement, and how their data streams are fused to provide the needed robustness requires further research. Meanwhile, using more sensors does not address how computer vision algorithms should interpret mirror environments in camera data. Since this is the main focus of our work, we concentrate on SLAM systems that only use camera sensors. Improving the quality of data extraction and understanding can then help with downstream tasks such as sensor fusion.

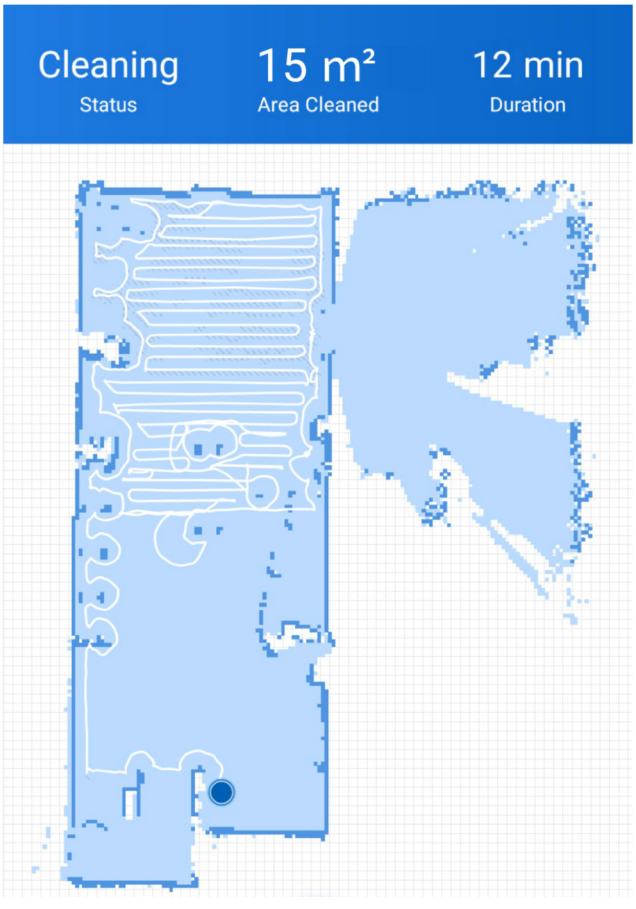


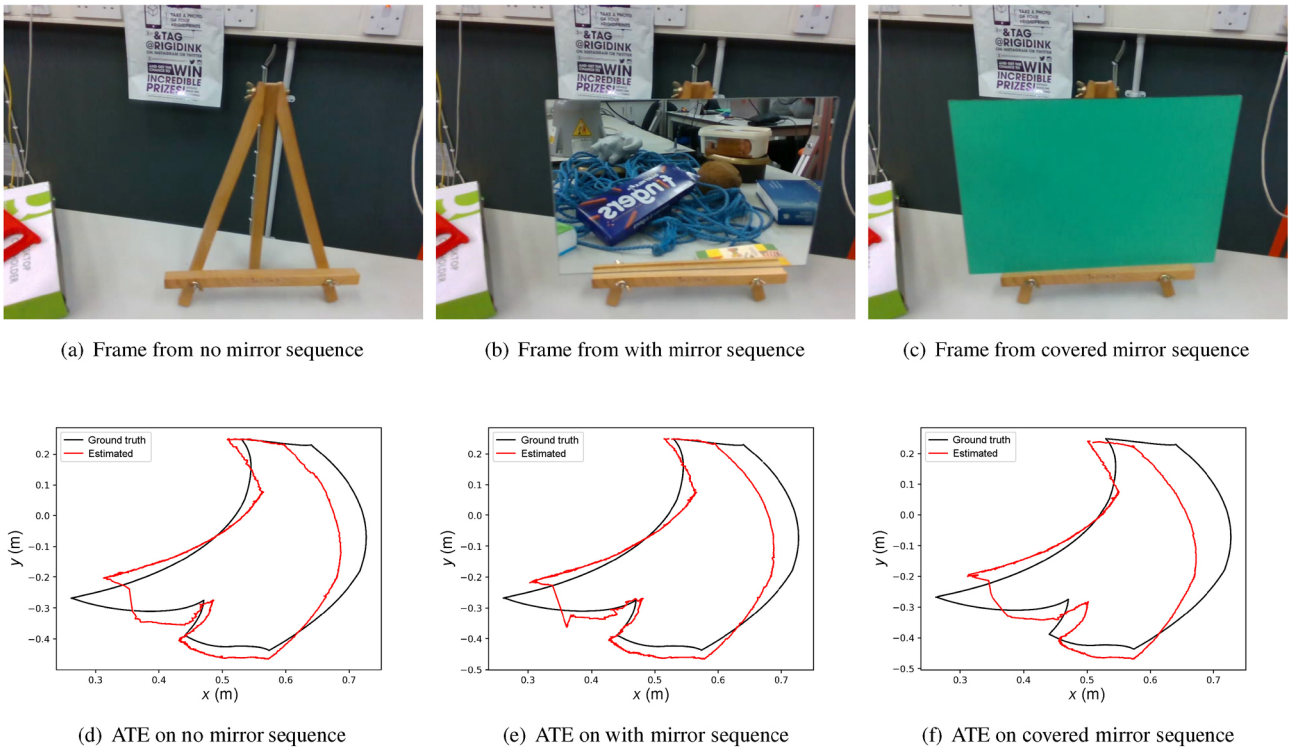
Fig. 1 Screenshot of map produced by an autonomous cleaner with combined camera and LiDAR sensors. Image used with permission from `https://twitter.com/qrs/status/1358450163216490498`, © Trammell Hudson (`https://trmm.net`) 2021.

A *reflection aware vSLAM* (RA-vSLAM) algorithm that is aware of mirrors in environments, and can even make use of the reflections, would thus be of interest for both domestic and industrial applications. A prerequisite to the development of such algorithms is to understand the performance of different vSLAM algorithms in environments with mirrors. However, one of the main barriers to this is a lack of suitable vSLAM datasets containing sufficient mirrors with sharp specular reflections.

In this paper we present our initial attempt at RA-vSLAM by systematically evaluating existing visual SLAM algorithms in environments with mirrors. The main idea is illustrated in Fig. 2. We collected an RGBD dataset, MirrEnv, with 7 unique trajectories, each recorded with one of 3 differently-sized mirrors, as well as control sequences with no mirror and with the mirrors covered. These sequences were then used to evaluate the performance of several representative vSLAM algorithms: monocular ORB-SLAM2 [27], RGBD ORB-SLAM2 [28], and BundleFusion [29], in order to determine the influence of planar mirror reflections.

In summary, the contributions of our work include:

- The MirrEnv dataset, which contains image sequences captured in environments with mirror reflections. With ground-truth poses and ground-truth mirror-label masks, this dataset provides a benchmark to promote research into robustness of vSLAM algorithms to visual complexities.
- Quantitative and qualitative evaluations of vSLAM algorithms representative of the state-of-the-art, in environments with mirror reflections.
- A discussion of the advantages and disadvantages to vSLAM of mirrors in environments, with potentials and directions for further research into this visual complexity.



(a) Frame from no mirror sequence (b) Frame from with mirror sequence (c) Frame from covered mirror sequence

(d) ATE on no mirror sequence (e) ATE on with mirror sequence (f) ATE on covered mirror sequence

Fig. 2 Image frames from trajectory `OutLoop3`, showing the sequences where (a) there is no mirror, (b) the medium mirror is visible, and (c) the medium mirror is covered. Ground-truth (black) and estimated (red) trajectories from RGBD ORB-SLAM2 are shown for these sequences in (d)–(f), respectively.

2 Related work

2.1 Visual SLAM methods

There are several ways to categorise visual SLAM algorithms. As indicated in Ref. [9], a classical categorisation is based on whether they utilise information by extracting features from frames (indirect) or using pixel intensities themselves (direct). Semi-direct [30] methods try to balance the benefits and costs of these approaches.

Indirect methods, often using monocular commodity cameras [31, 32], were some of the earliest visual SLAM methods. By focusing on prominent features, the number of pixels processed, across a whole image sequence, could be kept low, reducing computational costs. Later, ORB-SLAM [27] introduced a framework for real-time sparse SLAM, using multithreading to extract ORB features, track them, and maintain a global pose-graph. An improved version, ORB-SLAM2 [28], became one of the most established sparse indirect methods, with high localisation accuracy; it has been extended to RGBD and stereo input. ORB-SLAM2 is often used as a basis upon which other methods are built or compared, such as IV-SLAM [18] and GCNv2-SLAM [33, 34]. Whilst feature extraction has been used to make vSLAM robust to some types of visual complexities, it was believed that mirrors might impede the feature matching and tracking processes, incorrectly positioning a feature of a real object at the perceived location of the virtual object.

In comparison, direct methods calculate photometric and geometric errors at every pixel rather than extracted sparse feature points [35–37]. However, these methods tend to suffer when images contain noise. They usually require more expensive global shutter cameras and additional calibration techniques that need to be carried out precisely. As a consequence, there has been less development of purely direct methods. Large-scale direct SLAM (LSD) [38] was one of the first methods, using direct image alignment before estimating depth values for a filtered set of pixels. Direct sparse odometry (DSO) [35] took this further to only calculate photometric errors to reduce bias of geometric constraints when the image input is well calibrated. DSO was updated to include a loop-closure step to

provide global consistency [39]. An RGBD method called bundle adjusted direct SLAM (BAD-SLAM) [37] addressed the problem of applying bundle adjustment when every pixel of a frame represents a feature that needs to be included as a parameter in the optimisation. In the presence of mirrors, direct methods might struggle with the moving perspective of the virtual objects, illumination changes caused by non-Lambertian surfaces, and invalidated assumptions (such as those in Ref. [10]) used in correspondence search. These could all contribute to additional accumulated error.

Semi-direct methods use a mixture of direct and indirect techniques, balancing their advantages and disadvantages to try and improve overall performance. Semi-direct visual odometry (VO) [30] only used indirect feature extraction from monocular input when keyframes were created, and otherwise used the direct approach over small patches. A later attempt at semi-direct VO improved real-time performance by using direct frame alignment only on areas of an image with high intensity gradients (edges and corners) [40]. Pose estimation from indirect features and refined by downsampled-direct matching was used in BundleFusion [29], providing sufficient information to integrate frames into dense mesh maps. BundleFusion was considered representative of these semi-direct methods, which also provided dense mesh maps for qualitative analysis. Since it was expected that both direct and indirect methods would experience worse localisation performance, such a semi-direct method was also expected to suffer from reduced accuracy, although it was unclear if the combination of techniques would marginally improve or worsen this compared to pure direct or indirect methods. Additionally, it was expected that the dense meshes output by BundleFusion would show physically unrealisable representations, e.g., visible objects behind solid opaque walls, and real and virtual objects simultaneously occupying the same space.

RGBD sensors have been popular in visual SLAM, requiring less setup, calibration, and processing to accurately calculate a depth image compared to typical stereo arrangements [8]. A number of methods [30, 38, 41, 42] specifically take advantage of RGB and depth images as input. These methods offer high accuracy with dense mappings for use in applications. Hardware improvements allowed some methods, including Kintinuous [43], ElasticFusion, and BundleFusion to use GPU processing to generate dense maps. Some components of vSLAM have been replaced by machine learning components. Hybrid methods like CNN-SLAM [44] and D3VO [22] use convolutional neural networks (CNNs) to replace significant front-end parts of the system, whilst codeSLAM [45] and DeepFactors [46] use auto-encoders.

Background on the development of these areas in visual SLAM can be found in several detailed reviews including Refs. [8, 47, 48]. Specific attention is given to recent machine learning methods in Refs. [49–51].

2.2 Visual SLAM datasets

Most vSLAM methods are evaluated against data collected from real-world environments. These datasets are usually made up of image sequences from either a stereo or RGBD camera, with one set of the RGB/monochrome images usable as input for monocular systems too.

Such datasets include the TUM RGBD dataset [52], which initially contained 5 handheld sequences in an office, as well as 10 sequences in a warehouse (6 handheld, 4 mounted on a mobile ground robot), many containing loop closures. Additional sequences have since been added to provide particular challenges like low texture environments. These sequences were recorded using a rolling shutter depth camera (640 × 480, 30 Hz) with ground truth calculated from motion capture equipment. The ground robot sequences keep the camera at a fixed pose relative to the robot, and at the same height, limiting the motion to panning left and right. The ETH3D dataset [37] uses the same dataset format and contains 61 training and 35 test sequences (with monocular, stereo, RGBD, and IMU sensors on a calibrated rig in a motion capture environment) for optimising model parameters, originally used to improve the scene representation to reduce artefacts and distortions.

The KITTI dataset [53] of stereo images (1392 × 512, 10 Hz) was recorded from a car driving in a suburban area. The dataset contains 22 sequences, covering approximately 40 km of outdoor environments. Ground truth positions and environment point clouds were collected using GPS, IMU, and a LiDAR scanner. As well as loop closures, these sequences contain dynamic objects (pedestrians and cars), illumination changes, and shadows.

EuRoC [54] is another prominent dataset collected with monochrome stereo global-shutter cameras (752×480, 20 Hz) attached to a micro aerial vehicle (MAV). Of the 11 sequences collected, 5 were from a large room with industrial equipment, and 6 from an office environment. In the industrial setting, ground truth was collected using a laser tracking system, whilst the office utilised a motion capture arrangement. Difficulty of sequences was determined by the illumination, sharpness of motion, and amount of texture in the environment.

Other datasets for visual SLAM exist, such as the photometrically calibrated monocular dataset TUM monoVO described in Ref. [36]; it is more suitable for direct methods such as DSO. Some datasets are accompanied by information for other computer vision tasks. For example, ScanNet [55] and NYU-Dv2 [56] have RGBD images with semantic segmentation labels, and are commonly used to train deep-learning systems to integrate semantic information into navigation. Whilst NYU-Dv2 lacks the ground-truth camera poses needed for vSLAM benchmarking, ScanNet does have this ground-truth data and has started seeing usage for evaluation purposes, especially for neural implicit SLAM [57, 58].

Synthetic datasets and simulation environments have also grown in popularity for testing visual SLAM systems. These include ICL-NUIM [59], Replica [60], and TartanAir [61]. They allow 3D reconstructions to be compared to an accurate ground truth, with ray-tracing and photogrammetry being used to make them look more realistic. However, they are limited by the degree of physical realism, in terms of ambient lighting, modelling of non-Lambertian surfaces, camera artefacts, and the scales at which photo-realism can be maintained.

However, none of these datasets have specifically included real-world specular reflections from planar mirrors, and many of them avoid or minimise time where mirrors may be present. As a result, this visual complexity is insufficiently represented in existing vSLAM datasets to be able to analyse the effect it might have on a system's performance.

2.3 Mirror detection

Removal of specularities, appearing as bright spots on glass, metal, or glossy surfaces has been well studied in the literature [62–64]. These ideas have appeared in a limited number of visual SLAM methods, such as ElasticFusion and Introspective Vision (IV) SLAM [18]. ElasticFusion tries to identify bright illumination spots (specular highlights) on surfaces, as a means of finding light sources and improving model construction. IV SLAM uses a machine learning component to create a mask that identifies regions of a frame likely to introduce context-dependent noise. The mask is then used to exclude these regions from pose estimation. From the examples shown in their paper, the method seems capable of filtering out specular highlights, shadows, and lens flare. However, it is unclear if IV-SLAM can isolate mirror reflections, meaning that there are no known visual SLAM methods capable of explicitly taking mirror environments into account.

Removing mirror reflections is less common, especially for a planar mirror. In recent years, machine learning approaches have been used to detect mirrors in single images, such as MirrorNet [65], progressive mirror detection (PMD) [66], and systems described in Refs. [23, 24, 67]. Datasets containing ground-truth mirror masks have also been provided in Refs. [23, 65–67] for training these mirror segmentation networks. However, these datasets consist of individual images without ground truth poses, so are unsuitable for evaluating visual SLAM systems.

Another use of mirror reflections in robotics is for autonomous self-recognition. A system might have a fiducial marker on itself [68], or need to identify the reflection of its own dynamic movements using techniques such as those from ClusterSLAM [16]. Other literature discusses how a robot can identify itself from its own movements [69, 70].

3 Data collection

3.1 Equipment and calibration

Our MirrEnv (mirror environments) dataset was collected using an Intel RealSense D435i RGBD camera connected to an HP EliteBook 840 G3 laptop running Ubuntu 18.04. The camera was attached to the end effector of a KUKA iiwa 14R820 robotic manipulator arm, controlled using MATLAB and the KUKA Sunrise Toolbox (KST) [71, 72] to provide ground truth Euclidean pose for the end-effector (EEF), as shown in Fig. 3.

The RGBD camera has its own proprietary calibration software and checkerboard pattern for

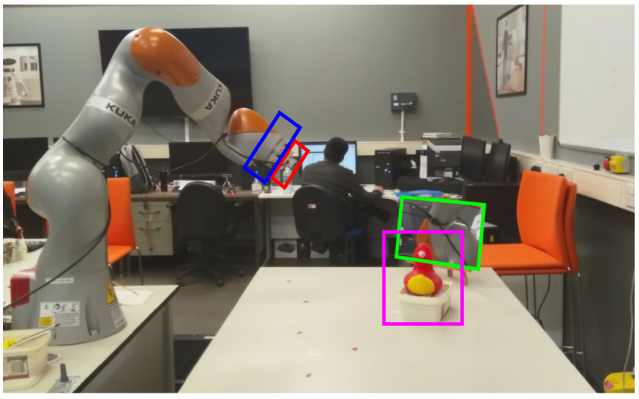


Fig. 3 RealSense camera (red box) attached to a KUKA robot arm end-effector (blue box), viewing a mirror (green box) and objects in an environment (pink box) during initial testing.

determining intrinsic parameters. The joints of the manipulator arm are determined using a method internal to the KUKA control box, which counteracts drift from physically calibrated positions.

Obtaining the transformation $\boldsymbol{X}$ between the manipulator arm's end-effector (EEF) and the RGBD camera requires a process known as hand-eye calibration. This involves recording a collection of end-effector poses $\{\boldsymbol{A}_1, \cdots, \boldsymbol{A}_n\}$ relative to the robot arm base, and checkerboard poses $\{\boldsymbol{B}_1, \cdots, \boldsymbol{B}_n\}$ relative to the camera, as explained in Ref. [73] and depicted in Fig. 4. Here, $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ are obtained by querying the KUKA control box and using the camera intrinsic parameters mentioned previously. Hand–eye calibration can then be used to obtain $\boldsymbol{X}$ by solving

$$\boldsymbol{A}_{ij}\boldsymbol{X} = \boldsymbol{X}\boldsymbol{B}_{ij}, \quad \forall\, 1 \leqslant i < j \leqslant n \tag{1}$$

where $\boldsymbol{A}_{ij} = \boldsymbol{A}_i^{-1}\boldsymbol{A}_j$ and $\boldsymbol{B}_{ij} = \boldsymbol{B}_i\boldsymbol{B}_j^{-1}$ are the relative transformations between the ith and jth pair of poses.

This can be interpreted in terms of the equivalence

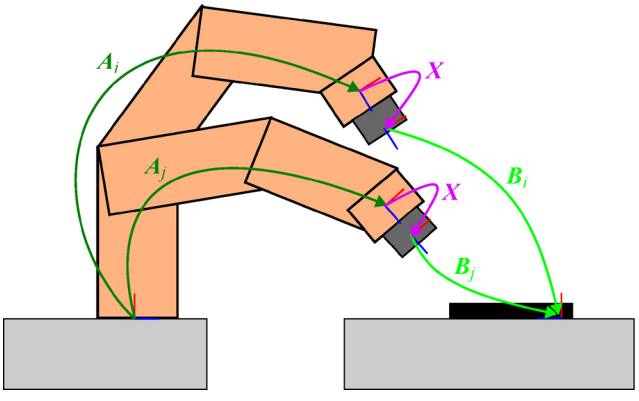


Fig. 4 Transformations involved in the hand–eye calibration process.

of different orders of changing the frame of reference: $\boldsymbol{A}_{ij}\boldsymbol{X}$ represents transforming from the ith to the jth EEF pose, then transforming from EEF to camera; $\boldsymbol{X}\boldsymbol{B}_{ij}$ represents transforming between the ith EEF and camera poses, then transforming from ith to jth camera poses. A total of 46 pairs of images and ground truth poses were captured, from which these transformations were determined.

A hand–eye calibration survey [73] notes that there are several ways to solve for $\boldsymbol{X}$: separately, simultaneously, or iteratively, depending on how the rotational component and the translational component of $\boldsymbol{X}$ are determined. We evaluated four hand–eye calibration methods [74–77] utilising code from Ref. [73]. The EEF to camera transformation $\boldsymbol{X}$ was finally calculated using a separable method [75], which had the lowest translational and rotational error. However, without bespoke and expensive calibration equipment, such as that used for motion capture arrangements, systematic errors may still occur.

3.2 Data capture

By manually controlling the manipulator arm, specific poses were found that allowed the camera to have a particular view of the surrounding environment from a desired location. These poses were then used as waypoints when planning the motion path of the manipulator through joint space in MATLAB. The planned path was then relayed to the manipulator arm in real time for smooth motion between waypoints. During movement of the robot arm, the camera provided a 3×8 bit RGB image and a 16 bit depth image, both at 640×480 resolution, aligned using the internal registration process on the camera. Following the dataset structure used in the TUM RGBD benchmark, Python and FFMPEG were used to compress and store the RGB and depth images into video frames in real time, using lossy MJPG for RGB images and lossless HEVC for depth. FFMPEG was later used to extract the individual frames after recording had finished. RGB and depth images were paired using the association script from the TUM RGBD benchmark.

Since the manipulator returned poses at approximately 333 Hz, and the camera only returned frames at 30 Hz, the association script was used again to pair them. The camera would begin recording shortly before and after the manipulator

was moving, so the OpenCV implementation of feature-based optical flow was used to identify the first image with significant motion and determine the temporal offset between frame and manipulator timestamps.

3.3 Image sequences

The image sequences collected are summarized in Table 1. Three sizes of planar mirror (13 cm × 18 cm, 21 cm × 30 cm, and a rounded mirror within 40 cm × 50 cm) (see Fig. 5) were combined with 7 unique trajectories, giving a total of 49 sequences recorded, including sequences with no mirror at all, and sequences with the mirror covered by a plain green card. Removing the mirror allows features behind the mirror plane to be visible; covering the mirror hides the features behind it, but introduces features at the occlusion boundary.

(a) Small mirror (b) Medium mirror (c) Large mirror

Fig. 5 Mirrors used, shown from trajectory `InLoop2`.

Table 1 Image sequences captured for the MirrEnv dataset. Values in brackets indicate sequences with a covered mirror. durMirr estimates the proportion of frames in that trajectory showing at least part of the mirror. avgMirr estimates the overall proportion of pixels belonging to the mirror, averaged over the whole trajectory

Trajectory	Mirror	Frames	Trajectory length (m)	durMirr (%)	avgMirr (%)
InLoop1	None	1878	4.989	—	—
	Small	1892 (1881)	4.988 (4.989)	77.895	2.753
	Medium	1883 (1887)	4.990 (4.989)	80.423	6.306
	Large	1886 (1866)	4.989 (4.989)	80.952	9.797
InLoop2	None	1915	2.719	—	—
	Small	1892 (1903)	2.718 (2.718)	91.579	3.653
	Medium	1937 (1893)	2.718 (2.719)	91.753	9.752
	Large	1915 (1862)	2.719 (2.717)	91.667	20.367
InLoop3	None	1937	3.659	—	—
	Small	1879 (1885)	3.658 (3.659)	100	3.944
	Medium	1875 (1865)	3.657 (3.657)	100	11.130
	Large	1912 (1877)	3.658 (3.657)	100	30.126
OutLoop1	None	2375	4.099	—	—
	Small	2362 (2353)	4.099 (4.100)	64.979	5.257
	Medium	2384 (2365)	4.101 (4.100)	68.201	13.026
	Large	2394 (2352)	4.101 (4.100)	68.333	23.324
OutLoop2	None	2378	2.915	—	—
	Small	2366 (2368)	2.916 (2.916)	46.414	1.920
	Medium	2368 (2362)	2.916 (2.916)	51.899	5.529
	Large	2363 (2362)	2.916 (2.916)	63.713	14.727
OutLoop3	None	2362	3.013	—	—
	Small	2368 (2356)	3.014 (3.015)	30.380	1.732
	Medium	2368 (2351)	3.014 (3.013)	39.662	4.854
	Large	2383 (2353)	3.014 (3.014)	44.770	11.651
InfBehind	None	2347	2.786	—	—
	Small	2336 (2352)	2.788 (2.788)	71.368	2.716
	Medium	2332 (2344)	2.788 (2.787)	71.795	8.531
	Large	2357 (2345)	2.787 (2.787)	71.186	22.692

Each image sequence was given a formatted label "`Trj`_X_1_X_2_X_3_X_4", where X_1 is a numerical index across all sequences from 1 to 49, X_2 is one of the 7 trajectories, X_3 indicates the size of the mirror (small, medium, large), and X_4 indicates whether the mirror was visible (W) or covered (C). If the sequence had no mirror, then X_3 and X_4 are replaced by "`No_Mirror`".

For each sequence with a mirror, every 10th frame was manually annotated to create a binary mask identifying the mirror region in that frame. These masks were used to estimate the proportion of annotated frames in the sequence that contained at least some mirror: the mirror duration (durMirr). They were also used to estimate the average proportion of each frame by covered by the mirror: the mirror coverage (avgMirr). Graphs illustrating the amount of mirror visible over each sequence, grouped by unique trajectory, are given in Fig. 15 in the Appendix.

The trajectories are grouped as `InLoop`, `OutLoop`, and `InfBehind`; example video clips are provided in the Electronic Supplementary Material (ESM). The `InLoop` trajectories follow a looping trajectory with the camera looking inward on the loop with fixed attention on the environment. Conversely, the `OutLoop` sequences follow a loop with the camera looking somewhat outward from the looped path, thereby lacking fixed attention. These sequences are intended to evaluate the effect that the presence of mirrors has on long term data association, i.e., loop closure, as well as on short term tracking. Trajectories with the same numerical suffix have the mirror positioned in the same part of the table (1: right, 2: left, 3: centre). Finally, `InfBehind` sequences have objects both in front of and behind a mirror, and the camera moves from in front of the mirror to behind it, then back again. Virtual objects present in reflections are often represented as existing in the space behind the mirror plane or conflicting with objects actually behind the mirror plane.

The trajectory path, size of mirror, and whether the mirror was covered were controlled as categorical, independent variables top determine the recorded image sequences. These choices influenced the extracted continuous quantities durMirr and avgMirr, calculated from the mirror region masks of every 10th frame. As Fig. 6 shows, the trajectory taken significantly affected the quantity durMirr, whilst

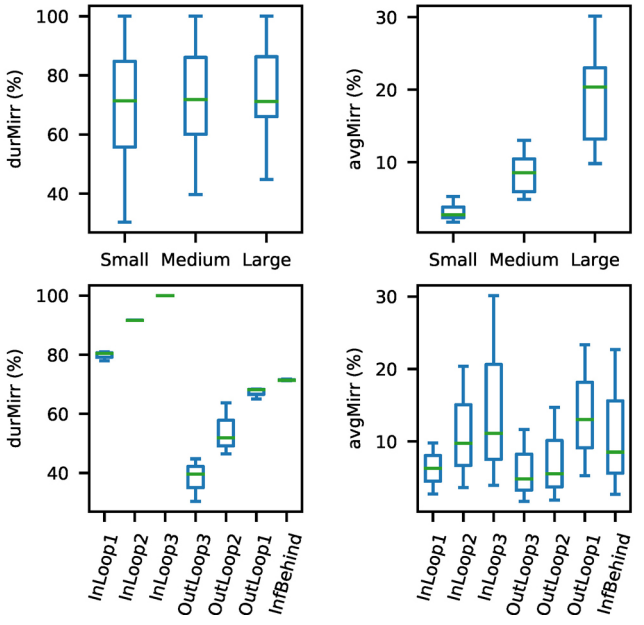


Fig. 6 Boxplots showing relationships between the calculated values durMirr and avgMirr, and the trajectory/mirror categories.

only having a limited influence on avgMirr, which was more dependent on the size of mirror used.

In keeping with the analysis of the mirror segmentation dataset (MSD) done in MirrorNet, the mirror region masks from all sequences were combined into a heatmap (Fig. 7). This demonstrates the variety of mirror locations within the frames across all sequences. Due to the real-world position of the mirror and the limited reach of the manipulator arm, the mirror occupied the lower left corner of frames less frequently than other parts of frames. This is also fairly consistent with the results in MirrorNet. The minimum value of the heatmap is 15, so every pixel is at some point within the mirror region. Due to the fixed attention sequences, and positions chosen

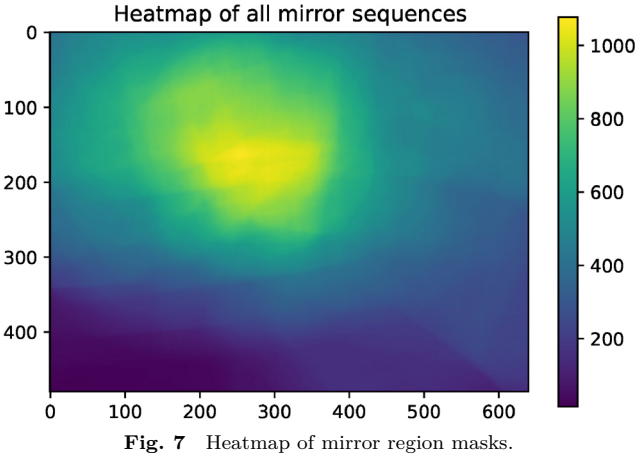


Fig. 7 Heatmap of mirror region masks.

for the mirror crossing from side-to-side, the centre of the frames is the most frequent location for the mirror.

One of the main advantages of the MirrEnv dataset over other vSLAM datasets is the availability of control sequences for the presence of mirrors. This allows for a comparative analysis of how vSLAM systems work in environments with mirrors versus environments without reflections.

4 Experimental setting

4.1 General questions

Prior to evaluating the representative methods on the collected data, there were some questions of particular interest to be answered. The initial hypotheses to be tested included:

- Does the amount of time that the mirror is visible affect vSLAM performance? A longer duration was expected to cause individual errors to accumulate, thereby increasing overall error.
- Does the coverage of the mirror affect vSLAM performance? An increase in the size of the mirror region was expected to increase error.
- How are different types of methods affected by the presence of mirrors? It was considered that performance might degrade if there was confusion matching real objects and virtual objects across frames, with direct methods possibly suffering more, as feature-based methods are more resilient to photometric and geometric errors in raw data.

To test these hypotheses, we carefully selected representative methods and evaluation metrics, and carried out experiments to quantitatively and qualitatively evaluate the results.

4.2 Representative methods

Of the methods described in Section 2.1, it was necessary to select some representative of different techniques and approaches. Particular focus was given to methods previously evaluated on the similarly structured TUM RGBD dataset, as these methods were more likely to produce reasonable pose and map estimates on this type of data. Another hurdle to overcome was finding methods that were open-source and readily deployable, an issue raised in Ref. [78]. The methods chosen were implemented on a Ubuntu 18.04 computer with an NVIDIA GeForce RTX 3070 GPU and CUDA 11.4.

When choosing a feature-based method, both ORB-SLAM2 and ORB-SLAM3 [79] were considered. ORB-SLAM2 is often used as the basis upon which other methods have been built, and as such forms a very meaningful baseline for comparison. ORB-SLAM3 has additional functionality over ORB-SLAM2 including adapted map initialisation and merging strategies, an improved loop closure process, and integration of sensors with other modalities. For these reasons, both ORB-SLAM2 and ORB-SLAM3 were used, each operating in the monocular and RGBD modalities.

A direct method was also sought. LDSO was initially considered as it was the version of DSO including a loop-closure mechanism, and DSO had been evaluated on the commodity camera data of TUM RGBD dataset. However, the poor performance of direct methods on data collected by rolling shutter cameras became clear, as all mirror environment sequences terminated almost immediately due to tracking problems.

Despite this setback, semi-direct methods show some greater robustness given data from commodity cameras, and provide a dense map as output. BundleFusion uses SIFT features for initial alignment and long-term data association, thereby being more resilient to data from rolling shutter cameras. However, it improves on the inter-frame alignment using dense correspondences determined by minimising photometric and geometric error terms: the technique used in direct methods. BundleFusion additionally produces a 3D mesh of the surfaces being mapped, which could then be qualitatively assessed for accuracy, and for sources of visual confusion. An implementation of BundleFusion [80] ported to Ubuntu was used.

To summarise, the representative methods chosen were the feature-based methods ORB-SLAM2 and ORB-SLAM3 (both monocular and RGBD modalities), as well as the semi-direct method BundleFusion.

4.3 Evaluation metrics

Across the various datasets discussed in Section 2.2, and by authors of different methods, several performance metrics have been proposed. *Alignment error* was proposed in TUM monoVO to capture errors in translation, rotation, and scale. A similar, proprietary metric is also defined in KITTI, which

separates the translational and rotational errors to better characterise the sources of error in the system.

In the TUM RGBD benchmark, two trajectory error metrics for a sequence of pose estimates $\boldsymbol{P}_i \in \mathrm{SE}(3)$ and ground truth poses $\boldsymbol{Q}_i \in \mathrm{SE}(3)$ were introduced. The *absolute trajectory error* (ATE) is given by

$$\boldsymbol{F}_i = \boldsymbol{Q}_i^{-1}\boldsymbol{S}\boldsymbol{P}_i \tag{2}$$

where $\boldsymbol{S}$ is a rigid transformation aligning the two trajectories. The performance for a whole sequence is reported as the root mean square error (RMSE) of all $\boldsymbol{F}_i$. ATE measures the difference between ground truth and estimated poses at a particular time, and used by the TUM RGBD benchmark specifically for SLAM systems to be able to measure overall consistency of the estimated poses.

The second metric is the *relative pose error* (RPE) which measures the drift of pose estimates away from the ground truth over different size time-steps. It is calculated as

$$\boldsymbol{E}_i = (\boldsymbol{Q}_i^{-1}\boldsymbol{Q}_{i+\Delta})^{-1}(\boldsymbol{P}_i^{-1}\boldsymbol{P}_{i+\Delta}) \tag{3}$$

where Δ can be adjusted to compare changes in pose over a certain time span, and $\boldsymbol{E}_i$ can be split into translational and rotational components, denoted here as $\mathrm{RPE_t}$ and $\mathrm{RPE_r}$ respectively.

ATE has seen widespread use as the primary and often sole performance metric for visual SLAM throughout the literature, with RPE being used with visual odometry systems like DSO. Moreover, it is recognised that ATE and RPE can be heavily correlated [52, 81]. For clarity and conciseness, ATE has been chosen as the primary metric for reporting our results in Table 6 in the Appendix, with $\mathrm{RPE_t}$ values being provided for completeness in Table 7.

5 Results: visual SLAM with mirror presence

5.1 Overview

In monocular operation, ORB-SLAM2 and ORB-SLAM3 produce a sparse collection of pose estimates for keyframes. In RGBD operation, these ORB-based methods and BundleFusion provide pose estimates for almost every frame in a sequence. From these pose estimates, we quantitatively analyse the localisation performance of these methods. BundleFusion also reconstructs a 3D mesh as a map of the environment, which we also qualitatively analyse.

5.2 Pose estimation failures

5.2.1 Need

In some instances, pose estimation may fail, skewing the resulting error calculations and misrepresenting localisation quality. It was therefore necessary to identify and remove such results before analysing localisation performance.

5.2.2 Loss of tracking

Whilst a loss of tracking did not stop these methods (unlike LDSO), large gaps in pose estimations could skew trajectory error calculations. Re-localisation only occurred once enough previously seen features had been recognised (typically from a similar camera pose). Loss of tracking could be inferred by the number of keyframes associated with ground truth poses and seen more explicitly by visualising the per-frame error to identify significant gaps. In this way, it was found that RGBD ORB-SLAM3, did not lose tracking on any sequence, while RGBD ORB-SLAM2 maintained tracking for all but 1 sequence. Monocular ORB-SLAM2 had significant gaps for 6 sequences, while monocular ORB-SLAM3 had such gaps for 3 sequences. BundleFusion failed to track adequately in 18 sequences, many occurring within specific trajectories. The specific sequences affected are given in Table 2, and are represented by missing values in Table 6 in the Appendix.

Several details within the MirrEnv dataset sequences were recognised as particularly challenging, with the potential to cause vSLAM systems to deteriorate in performance. Covered mirror sequences frequently caused prolonged tracking problems, probably because the green covering was almost featureless. Monocular ORB-SLAM2 lost tracking for `OutLoop2_Large_C`, `OutLoop3_Large_C`, and `InfBehind_Large_C`. In each of these cases, the tracking problems appear to be due to the large mirror covered by the textureless green card, leading to a discontinuity in tracked features. Other than the large covered mirror sequence from the `OutLoop1` trajectory, RGBD ORB-SLAM2 did not lose tracking for a significant period, probably because it utilises depth information to detect and track features. In fact, all vSLAM systems lost tracking when panning across the covered area of the large mirror of `OutLoop1`, but only RGBD ORB-SLAM3 was able to recover, probably because it created a new map and merged the maps upon identifying a similar

Table 2 Image sequences for each representative method in which tracking was substantially lost, resulting in missing values in Table 6 in the Appendix

Monocular ORB-SLAM2	RGBD ORB-SLAM2	Monocular ORB-SLAM3	BundleFusion
OutLoop3_Large_C OutLoop2_Large_C OutLoop1_No_Mirror OutLoop1_Large_W OutLoop1_Large_C InfBehind_Large_C	OutLoop1_Large_C	OutLoop2_Large_W OutLoop1_Large_C InfBehind_Large_W	InLoop2_No_Mirror, InLoop2_Small_W InLoop2_Small_C, InLoop2_Medium_C OutLoop3_No_Mirror, OutLoop3_Small_W OutLoop3_Small_C, OutLoop3_Medium_W OutLoop3_Medium_C, OutLoop3_Large_W OutLoop3_Large_C, OutLoop1_No_Mirror OutLoop1_Small_W, OutLoop1_Small_C OutLoop1_Medium_W, OutLoop1_Medium_C OutLoop1_Large_W, OutLoop1_Large_C

pose before and after the interruption in tracking.

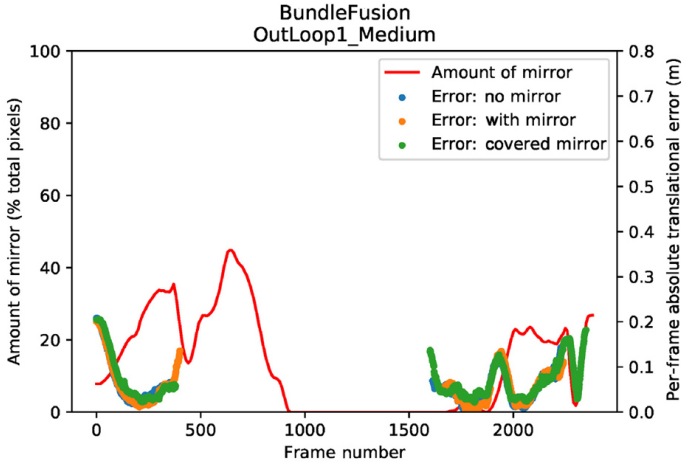
However, mirror sequences may also create features for the vSLAM systems to detect. Figure 8 shows the example of `OutLoop1`, where BundleFusion and monocular ORB-SLAM2 lost tracking in the no mirror sequence. Figure 8(a) shows the ATE for BundleFusion during the `OutLoop1` medium covered and uncovered mirror sequences, as well as the no mirror sequence. Tracking is lost in the no mirror sequence at frame 370, seemingly because a rapid rotation causes the few remaining tracked features from a few frames before (Fig. 8(b)) to move out of view, and it is unable to replace these features from a less feature rich background (Fig. 8(c)). In the mirror sequence, frame 390 (Fig. 8(d)) shows that before losing tracking at frame 420, there were still sufficient foreground features visible, in the mirror region. Once tracking was lost (Fig. 8(e)), few of the previously tracked foreground features would have been left, and the background was featureless and would not have provided new points stable enough to track.

5.2.3 Inverted motion and coincident pose estimation

Another cause of pose estimation failure is when the calculated pose contradicts the camera motion observable from the image sequences. This was first noticed when evaluating ORB-SLAM3, and confirmed to occur to a lesser degree in ORB-SLAM2 too. In particular, it was observed that (i) the estimated camera motion could be inverted compared to the image sequence, and (ii) the scale of the translation component in a pose transformation could change significantly, with pose estimates appearing to coincide at a single point in space. These issues were considered to be systematic, possibly due to poor initialisation, handling complex rotation movements, and monocular depth estimation. Such sequences were identified by watching the mapping visualisation of each run, and could be inferred by plotting the estimated poses to see if poses coincided for a significant proportion of their evaluation. In this way, it was found that only monocular ORB-SLAM2 and ORB-SLAM3 appeared to be affected by these failures. Monocular ORB-SLAM2 repeatedly gave inverted motion estimates for 1 sequence, while Monocular ORB-SLAM3 did so for 3 sequences. Monocular ORB-SLAM3 also had coincident pose estimates (coinciding at a single point in space) for another 3 sequences. All sequences resulting in such pose estimation failures are given in Table 3 and result in missing values in Table 6 in the Appendix.

Some of the trajectories involved panning across featureless regions of the surrounding environment (walls, floors, tables), and the use of a 7-axis robot arm meant that these movements could involve such tight rotations that vSLAM feature tracking sometimes struggles. These factors appeared to cause inverted motion estimation and coincident pose estimation for the monocular methods.

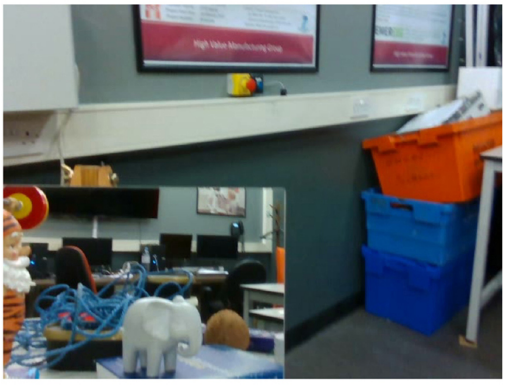
In summary, it was noted that the RGBD modality of ORB-SLAM2 and ORB-SLAM3 showed general robustness across the different trajectories, usually only struggling when there were extremely few features available to track. The monocular modality of both ORB-based methods encountered additional situations which prevented adequate pose estimates to be calculated. BundleFusion encountered tracking problems whenever the complexities of the environment or trajectory motion became too difficult, resulting in many unusable results. If loss of tracking affected more than half of an image sequence, then the sequence was omitted from Table 6. Similarly, if inverted motion and coincident pose estimation affected the majority of a sequence, it was omitted.



(a) BundleFusion per-frame ATE for medium-sized mirror.

(b) Frame 340, no mirror

(c) Frame 370, no mirror



(d) Frame 390, with mirror

(e) Frame 420, with mirror

Fig. 8 Example showing loss of tracking: `OutLoop1`, medium-sized mirror. (a) Per-frame ATE for BundleFusion. (b, c) Sample frames before and after loss of tracking, without mirror. (d, e) Sample frames before and after loss of tracking, with mirror.

Table 3 Image sequences for which representative methods suffered pose estimation failures, resulting in missing values in Table 6 in the Appendix. Rows indicate the primary cause of failure

	Monocular ORB-SLAM2	Monocular ORB-SLAM3

Inverted motion estimation	`InLoop3_No_Mirror`	`InLoop2_Small_C`
		`InLoop2_Medium_C`
		`InLoop2_Large_W`

Coincident pose estimation	—	`InLoop3_Small_C`, `InLoop3_Medium_W`, `InLoop3_Medium_C`

Table 4 RMSE ATE for each vSLAM system, taken over sequences with a given kind of mirror presence

Method	Mirror	Covered	No mirror
Monocular ORB-SLAM2	0.102	0.106	0.113
RGBD ORB-SLAM2	0.097	0.080	0.077
Monocular ORB-SLAM3	0.117	0.132	0.150
RGBD ORB-SLAM3	0.086	0.080	0.078
BundleFusion	0.132	0.143	0.122

5.3 Localisation

5.3.1 Approach

Our analysis of localisation errors used the control sequences that either had no mirror or a covered mirror to establish baseline results along the trajectories. The robustness of vSLAM systems on the sequences with uncovered mirrors could then be compared to these baselines. Since all chosen representative vSLAM methods use multi-threading and random sample consensus (RANSAC), there will be variations when running the methods multiple times. Therefore we follow the protocol used for ORB-SLAM2: run each vSLAM method on the same sequence 5 times, and compute the median ATE of the 5 runs. By considering (i) localisation error by sequence type, (ii) localisation error versus mirror duration/coverage, and (iii) per-frame localisation error, we analyze that the results are at increasing levels of granularity whilst exploring their relationship to the presence of the mirror.

5.3.2 Localisation error by sequence type

The differences between the poses estimated by the three vSLAM methods and the associated ground truth poses was calculated to give the overall localisation error, primarily measured using ATE. Sequences with significant loss of tracking and other pose estimation problems (more than half of trajectory lacking an appropriate estimate) were discounted as erroneous. The results are summarised, grouped by mirror presence, in Table 4, whilst complete results are provided in Table 6 in the Appendix. For completeness, accompanying $\mathrm{RPE_t}$ results are provided separately in Table 7 in the Appendix.

Table 4 shows that, as expected, both RGBD ORB-SLAM2 and ORB-SLAM3 have noticeably higher average error on the mirror sequences, with the lowest average error being on the no mirror sequences. However, the opposite appears to be true for both the monocular ORB-SLAM methods. Table 6 in the Appendix, shows this opposite trend to be especially significant for OutLoop sequences. This may be because the OutLoop sequences have longer distances, and the depth maps can become noisy and less reliable over the unbroken views of the background over longer distances; this may be made worse by some tight rotational motion at the same time. BundleFusion saw increased ATE on sequences with mirrors and with covered mirrors, the latter having the highest error. From Table 6 in the Appendix, this appears to be due to higher errors on sequences with a large covered mirror. In these sequences, the large covered mirror renders a significant portion of the frame featureless. In cases where BundleFusion was able to extract sufficient SIFT features to retain tracking, these may have been too sparse or unreliable when used to calculate pose estimates. Additionally, the direct techniques used might not have adjusted the localisation enough given a large area of seemingly uniform intensity.

5.3.3 Localisation error and mirror duration/coverage

It is of interest to determine if a correlation exists between ATE, and durMirr and avgMirr determined from mask images for mirror sequences. Since these values for the control sequences were imputed, it was reasonable to stratify the analysis and focus on sequences with mirror. Spearman's rank correlation coefficients were calculated, along with their associated p-values; these are shown in Table 5. For both RGBD ORB-SLAM3 and BundleFusion, the correlation between ATE and the mirror duration durMirr fall within the 95%

Table 5 Spearman's rank correlation coefficients between mirror quantities and error metrics for sequences with a mirror, and associated p-values. Statistically significant results (within a 95% confidence interval) are highlighted in bold

		Mono. ORB. 2	RGBD ORB. 2	Mono. ORB. 3	RGBD ORB. 3	BundleFusion
avgMirr	Correlation coefficient	0.079	0.251	0.422	0.295	−0.015
	p-value	0.748	0.273	0.092	0.195	0.958
durMirr	Correlation coefficient	−0.155	0.278	0.07	0.458	0.567
	p-value	0.527	0.222	0.79	**0.037**	**0.034**

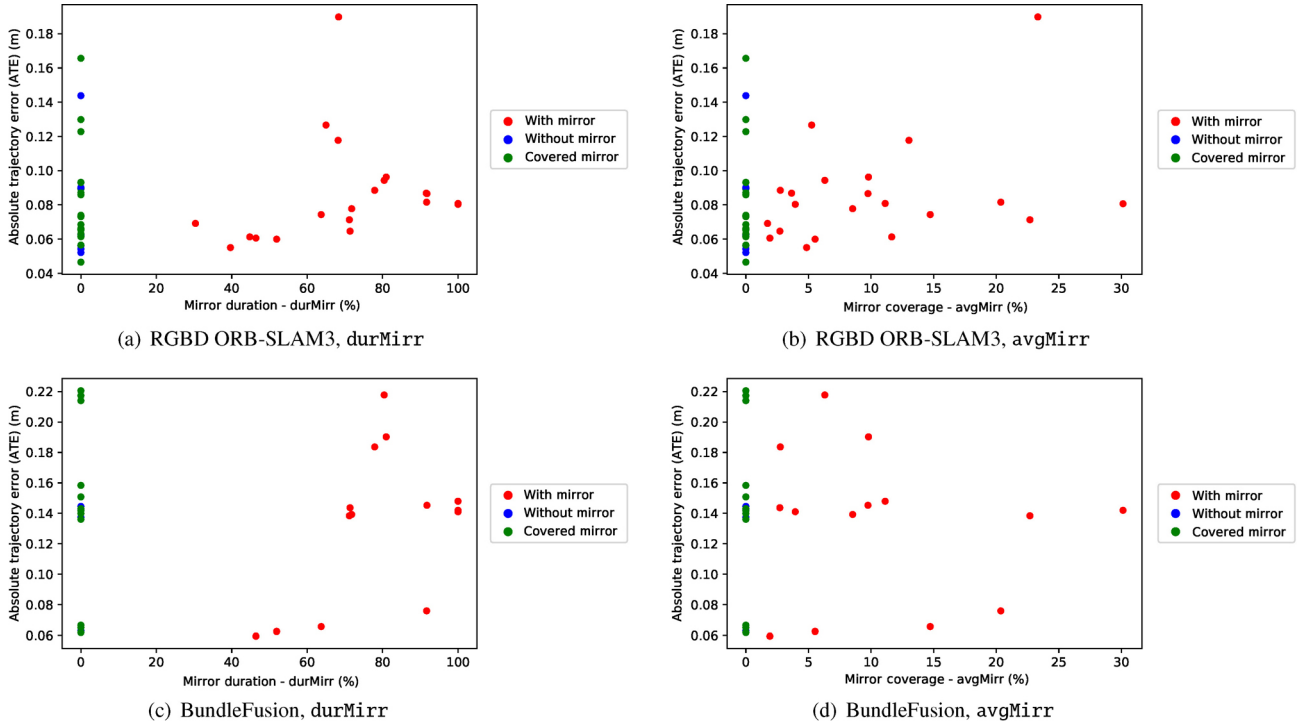
confidence interval of statistical significance, with correlation coefficients of 0.458 and 0.567 respectively. These are moderate positive correlations, indicating that increased duration of mirror presence added to the accumulated errors.

This correlation can be seen in Fig. 9, which shows the ATE for both RGBD ORB-SLAM3 and BundleFusion as durMirr and avgMirr vary for different sequences (scatter graphs for the other representative methods can be found in Fig. 16). In Figs. 9(b) and 9(d), the dispersed clusters of red markers correspond to weak correlation with mirror coverage avgMirr, but that the markers in Figs. 9(a) and 9(c) do show a moderately positive correlation. It can be further seen that the range of ATE values for the sequences with mirrors significantly overlaps the ATE range for the control sequences without mirrors (blue) or covered mirrors (green), suggesting that mirror duration and coverage do not significantly impact the ATE values overall.

In the case of RGBD ORB-SLAM3, one particular result with mirror duration of approximately 70% has a very high ATE, which probably increases the overall correlation. Similarly, it is worth highlighting that BundleFusion failed to produce meaningful results on the `OutLoop1` and `OutLoop3` trajectories, sequences with low mirror duration, and this may have weakened the observed correlation: BundleFusion may have performed poorly and yielded high ATE for those sequences. As mentioned in Section 3.3, the durMirr metric is closely related to specific trajectories, leading to the stratified nature of the control sequences plotted in Figs. 9(c) and 9(d). Therefore, the trajectories taken represent a confounding variable, so causation is not guaranteed, and should be explored further through the per-frame analysis.

5.3.4 Per-frame localisation error

As well as considering performance over the entire image sequence, an analysis of per-frame error versus mirror quantities was performed to seek fine-grained patterns in vSLAM performance, and to understand the sources of errors that affected the correlation coefficients and scatter graphs. We thus look at per-frame graphs for ORB-SLAM3 and BundleFusion on those trajectories for which they performed worst: `OutLoop1` for ORB-SLAM3, and `InLoop1` for BundleFusion.



(a) RGBD ORB-SLAM3, `durMirr` (b) RGBD ORB-SLAM3, `avgMirr`

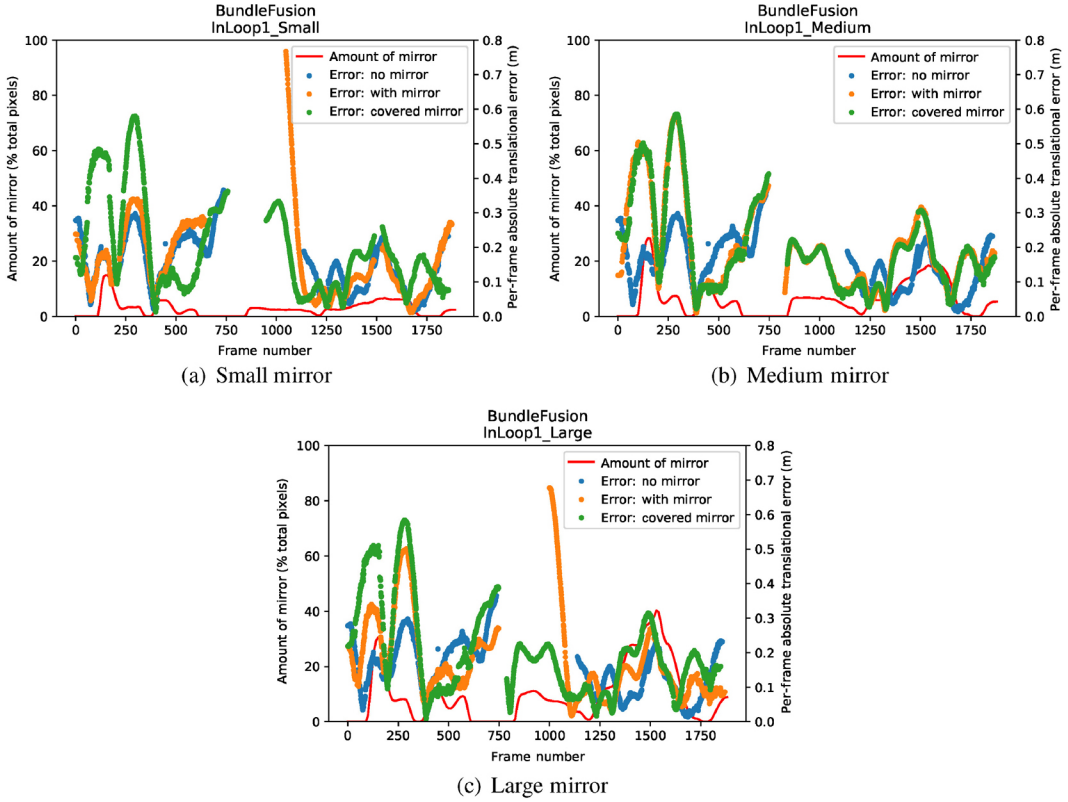
(c) BundleFusion, `durMirr` (d) BundleFusion, `avgMirr`

Fig. 9 Scatter graphs visualising the correlation between (a) RGBD ORB-SLAM3 ATE and durMirr; (b) RGBD ORB-SLAM3 ATE and avgMirr; (c) BundleFusion ATE and durMirr; (d) BundleFusion ATE and avgMirr. ATE of sequences without a mirror (removed or covered) are included to provide a baseline distribution for the sequences with a mirror.

Figure 10 shows that on `InLoop1`, BundleFusion performed almost uniformly on all covered mirror sequences and the medium mirror sequence, with only small variation around a brief period of lost tracking and re-localisation. This agrees with the very close ATE values in Table 6. However, the small and large mirror sequences begin with low errors similar to those for the no mirror sequences, and then spike after re-localisation occurs. While the ATE for the large mirror sequences appears to be correlated with the amount of mirror in each frame (red curve), similar results are also obtained when there is less mirror visible (in the small mirror case). Also, the peaks in ATE at the beginning of most covered and uncovered sequences could indicate that BundleFusion has difficulty in initialising for this trajectory. Overall, this appears to indicate that the high ATE values for BundleFusion on `InLoop1` were largely due to trajectory specific issues, such as difficulty of initialisation and the particular camera motions experienced, rather than the presence of the mirrors.

Figure 11 shows that for the small and medium mirror sequences for `OutLoop1`, the performances of RGBD ORB-SLAM3 were fairly consistent with each other and the control sequences. However, the large mirror sequence appears to result in periods of higher ATE compared to its control sequence, with the final peaks in error seeming to match the rise in the amount of mirror occurring at the same time. Indeed, the large mirror sequence is the one that caused RGBD ORB-SLAM3 to significantly influence the correlation coefficient in Table 6. However, without similar patterns noticeable in the small and medium mirror results, it is not possible to conclude that the amount of mirror is itself influencing the behaviour of RGBD ORB-SLAM3.

In summary, most comparisons did not show significant effects due to the presence of mirrors. In the case of RGBD ORB-SLAM3 and BundleFusion, the ATE did not overall appear to increase with the amount of mirror. The most frequent conclusion for the different situations examined was that trajectory-specific factors had a large influence on the ATE. Although quantitative analysis indicates that mirror presence could be a subtle contributing factor to vSLAM localisation error, it remains difficult to reliably identify the effect of mirrors on visual SLAM localisation accuracy without more sequences from



(a) Small mirror (b) Medium mirror

(c) Large mirror

Fig. 10 BundleFusion per-frame ATE versus proportion of frame within the mirror region, for the `InLoop1` trajectory, for varying mirror sizes.

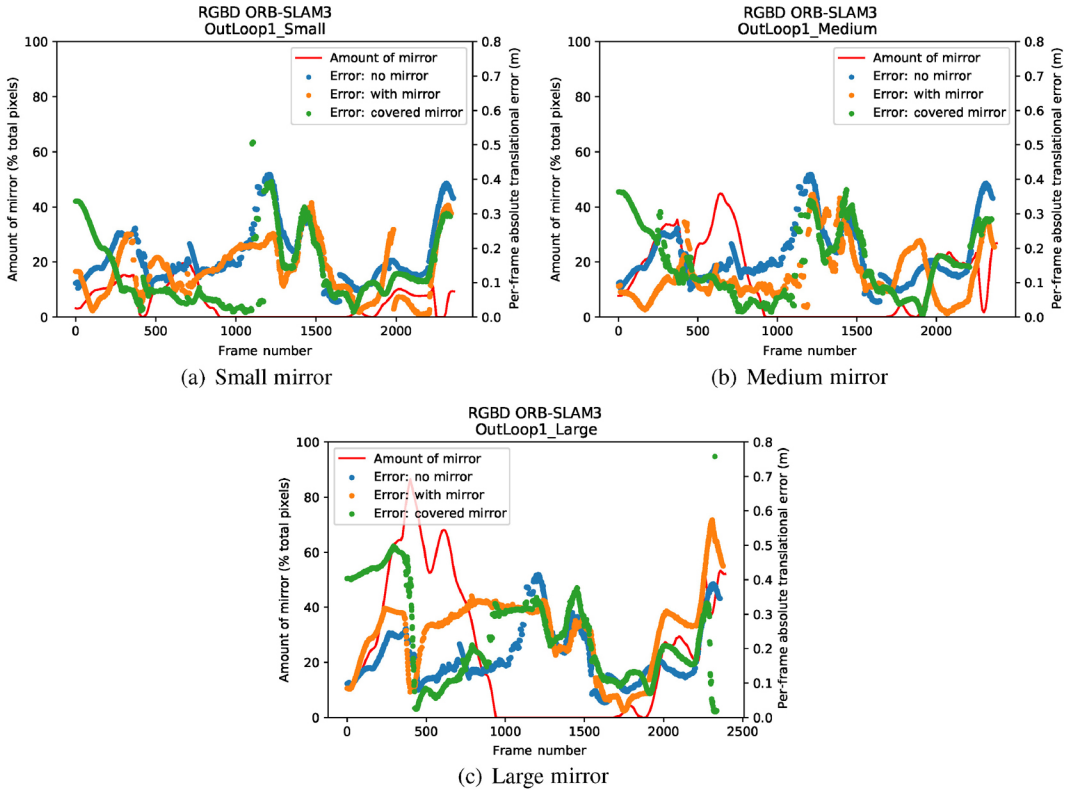


Fig. 11 RGBD ORB-SLAM3 per-frame ATE versus proportion of frame that is within the mirror region, for the `OutLoop1` trajectory, for varying mirror sizes.

trajectories with overlapping amounts of mirror and a variety of difficulties of motion. For the representative methods where no correlation between ATE and either of the mirror metrics was statistically significant, the overall evidence does not support the proposed hypotheses, rather than the null hypothesis. Many of the same reasons why vSLAM systems can continue tracking in the presence of mirrors could also explain why the localisation error did not degrade under the same conditions (tracking features reflected from the mirror region, or formed at the intersection of the mirror boundary with the background).

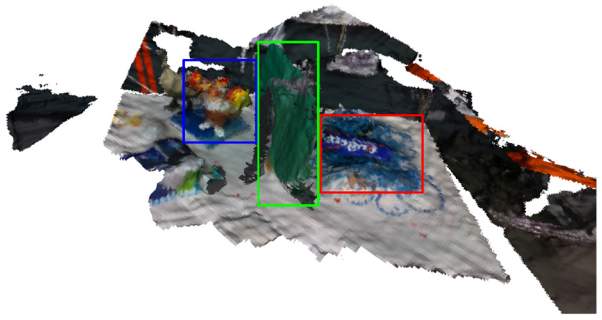
5.4 Mapping

Monocular and RGBD ORB-SLAM2 and ORB-SLAM3 produced feature maps, but their sparsity makes static visualisations difficult to interpret. It is notable that RGBD modalities generally produced denser feature maps than monocular modalities. Conversely, BundleFusion produced dense surface meshes, with colour inferred from the RGB input.

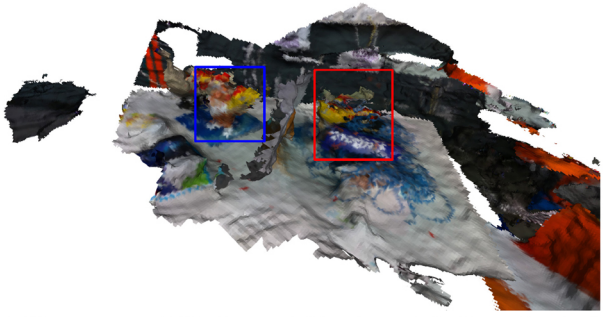
The quality of the meshes produced by BundleFusion can be reviewed to determine the map reconstruction abilities of the dense method in mirror environments. In some cases, reflections of objects are rendered as real objects, conflicting with actual real objects located behind the mirror: see Fig. 12.

For other cases, the virtual environment is projected through an opaque wall in the real world, creating inaccurate geometry where the mirror is treated like a window or open doorway: see Fig. 13. As more images in a sequence are processed, vSLAM methods perform map updates. In the presence of mirrors, map updates can remove and correct inaccuracies, or they can embed and affix mistakes in the map that are later difficult to recover from.

Overall, when mirrors are visible, the maps generated may include visibly identifiable reconstruction failures. Ideally, a planar mirror would be represented as a flat surface with reflective properties, whilst information from the mirror region would be projected back to aid reconstruction of the real parts of the environment. However, existing vSLAM methods do not identify reflections, and the virtual features are merged with the real features, leading to a geometrically inaccurate representation of the environment.

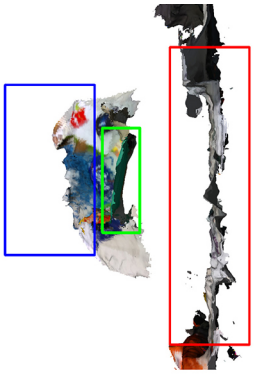


(a) Mirror covered by green card. Blue box: real object (gnome). Green box: card covering mirror. Red box: unaffected real objects (packaging) behind mirror.

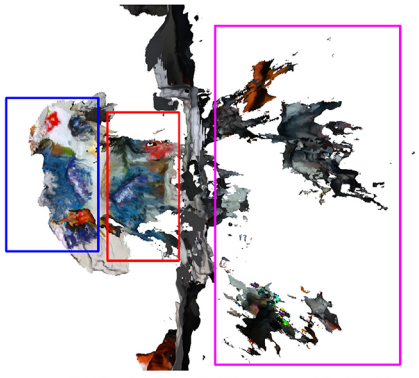


(b) Mirror uncovered. Blue box: real object (gnome). Red box: virtual reflection merged into real objects (packaging) behind mirror.

Fig. 12 BundleFusion meshes for `InfBehind` sequences.



(a) Mirror covered by green card. Blue box: real objects on table. Green box: card covering mirror. Red box: wall in real environment.



(b) Mirror uncovered. Blue box: real objects on table. Red box: reflection of items on table. Pink box: reflections of real environment other than the table, mapped behind the wall.

Fig. 13 BundleFusion meshes for `InLoop3` sequences.

6 Discussion

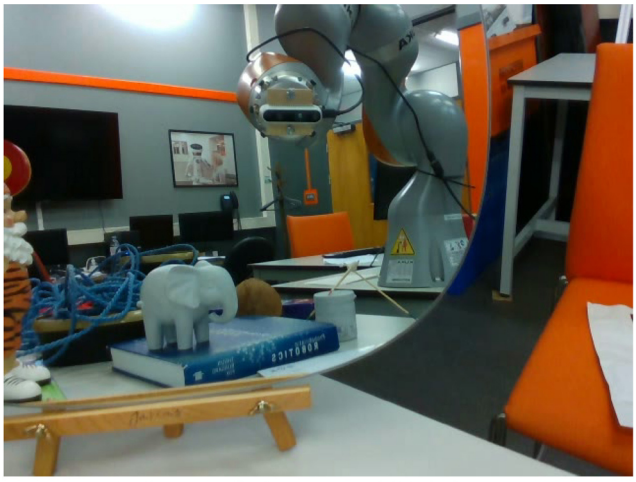
6.1 Localisation

From the analysis of the experimental results, it appears that mirror reflections provide benefits, as well as potential problems, for the performance of vSLAM systems. It was anticipated that mirrors would challenge the different component processes in a visual SLAM system and cause performance to deteriorate. One of the main ways that vSLAM systems were expected to fail was due to the effect reflection might have on feature extraction and tracking. Situations such as real objects and their reflections being visible simultaneously (an example can be see in Fig. 14(a)) could have caused uniqueness assumptions used in feature detection and description to be invalid; if a real object and its reflection were visible at different time in a sequence, place recognition could have mistakenly matched real objects with their mirror counterparts. It was also observed during experiments that the mirrors extended the camera's field of view in such a way that the camera and connected apparatus were present as dynamic objects in an otherwise stationary environment (again see Fig. 14(b)). However, these mirror specific issues did not appear to cause significant or obvious problems for tracking or pose estimation. Any effects on performance from these concerns would require further analysis to uncover.

Early analysis of localisation errors appeared to suggest a correlation between ATE for RGBD ORB-SLAM3 and BundleFusion, and mirror duration. However, the per-frame analysis of the localisation appeared to indicate that the most significant factors influencing the ATE of these methods were trajectory specific factors such as camera motions and the availability of trackable features or textures. It is still possible that mirror presence may have a direct effect on the localisation error of vSLAM systems, e.g., by adding errors to the localisation of features and exacerbating error accumulation. But ultimately, separating the direct influence of mirrors from other factors requires further work, as discussed further in Section 6.3.

Contrary to expectation, it was observed that under some circumstances, mirrors may have improved the localisation performance of vSLAM systems. With the mirror extending the camera's field of view, it was

(a) Many features on the real object (gnome) are also visible in the mirror



(b) Camera and robot arm are visible in the mirror

Fig. 14 Frames from the `OutLoop1` large mirror sequence.

possible for features to be detectable in the mirror region that might otherwise have been a featureless, untrackable surface. Similarly, the mirror is an object in the environment that can occlude or be occluded, thereby creating intersections at discontinuities of boundaries that a camera can perceive as reliable features to track. Rather than simply mitigating or filtering out mirror reflections, it may benefit vSLAM localisation for the system to be aware of the presence of mirrors, and to utilise that knowledge accordingly.

6.2 Reconstruction

In contrast, the qualitative analysis of scene reconstruction by the representative methods demonstrated an inability to compensate for reflections when mapping the 3D environment. In particular, the dense meshes generated by BundleFusion merged real and virtual objects and created inaccurate geometries with reflected spaces being observed as through an opening.

Since visual SLAM is a process that leverages localisation and mapping simultaneously to improve accuracy, it is conceivable that inaccurate reconstruction may also hinder localisation performance, if only through accumulating additional error. As with localisation, it would be useful for visual SLAM systems to identify mirror surfaces, reconstruct the environment accurately, and even take advantage of the extra perspective and extended field of view. The first step in this process would require detection of mirrors using visual SLAM sensors.

6.3 Future investigations

The collection of image sequences to specifically investigate mirror environments is an important initial step to building visual SLAM algorithms that are robust in the presence of, and can potentially make use of, mirror reflections. It is possible that in a complex, multi-component system for vSLAM, some parts may filter or compensate for errors introduced by other parts. Evaluating vSLAM systems was also complicated by the multi-faceted needs of the combined components, including acceptable movements, speeds, lighting conditions, and hardware requirements. Under even more controlled conditions, it might be possible to identify whether component processes (such as feature extraction, tracking, place recognition, and map reconstruction) experience specific problems in mirror environments. This might also help to find the best places in vSLAM systems to make reflection-aware improvements with minimal impact on efficiency.

Additional data collection for the MirrEnv dataset could help to overcome trajectory specific confounding factors, as well as provide sequences for testing specific components and capturing data suitable for direct and stereo methods. Eventually this could also include sequences suitable for sensor-fusion based methods. Such future investigations could help to determine the types of sensors and their arrangements that best provide a desired level of robustness in mirror environments, thereby improving on the situation previously raised in Section 1.

As Section 2.3 noted, the future development of machine learning models that can accurately and reliably detect mirrors could allow these tools

to be incorporated into visual SLAM methods. In doing so, visual SLAM would be capable of detecting each mirror region and utilising the virtual perspective to extend the camera's field of view. Some of the difficulties highlighted, such as mirrors making dynamic objects visible, might even help the mirror detection process [82]. Better environment reconstructions might also aid systems that use vSLAM for additional tasks such as collision avoidance, path planning, and virtual reality applications.

7 Conclusions

This paper has evaluated the performance of three visual SLAM methods (representative of monocular, RGBD, indirect and semi-direct techniques) in a number of mirror- and control-environments. This was accomplished by collecting the MirrEnv dataset of RGBD images and ground truth camera poses, and calculating the trajectory error of the camera poses estimated by the vSLAM algorithms. The results were then used to analyse the influence of mirrors on the localisation errors and quality of the mapping output of vSLAM methods.

The results indicate that RGBD ORB-SLAM3 and BundleFusion might be moderately influenced by the mirror duration, although the influence of mirror reflections was difficult to separate from other confounding factors. In general, RGBD methods had slightly higher localisation errors on mirror sequences compared to their control sequences. From the meshes generated by BundleFusion, it could be seen that the mapping output would integrate reflected objects, even when they conflicted with other real objects behind the mirror. Whilst the effect of mirrors on the localisation error might be marginal, the mapping processes were clearly greatly effected. The leveraged approach of SLAM means that inaccurate reconstructions or dense tracking of the RGBD methods could have contributed to accumulated localisation error.

An extensive discussion was provided on the expected difficulties that visual SLAM methods might encounter in mirror environments, why these expectations did not appear to have significant influence on the localisation error, and how mirrors might even be helping visual SLAM methods in some circumstances. Directions for future investigations were also discussed. Developments in mirror detection could yield opportunities to consider how the presence of mirrors could be utilised to improve map reconstruction, and whether this would consequently improve localisation too. These ideas will be explored in our future work on reflection-aware vSLAM algorithms.

Appendix

This appendix contains further experimental results.

Figure 15 shows how the amount of mirror changes for each of the sequences, assessed using mirror region masks created for every 10th frame in each sequence.

Table 6 shows absolute trajectory error (ATE) for each image sequence, for the 5 vSLAM methods. When methods failed to estimate poses (see Section 5.2), localisation results were not considered reliable, and so were removed.

Table 7 shows the translational components of the relative pose error (RPE_t) for each image sequence, for the 5 vSLAM methods. When methods failed to estimate poses, again the results were removed.

Figure 16 shows those scatter graphs for the representative methods which did not have statistically significant correlation (see Section 5.3).

Author contributions

Peter Herbert: design of work; acquisition, analysis and interpretation of data; creation of new software used; manuscript drafting and revision. Jing Wu: conception and design of work; analysis and interpretation of data; manuscript drafting and revision. Ze Ji: conception and design of work; acquisition, analysis and interpretation of data; manuscript revision. Yu-Kun Lai: conception and design of work; analysis and interpretation of data; manuscript revision.

Availability of data and materials

The MirrEnv dataset is available at https://doi.org/10.17035/d.2023.0292477898.

Acknowledgements

This research was funded by the UK EPSRC through a Doctoral Training Partnership No. EP/T517951/1 (2435656).

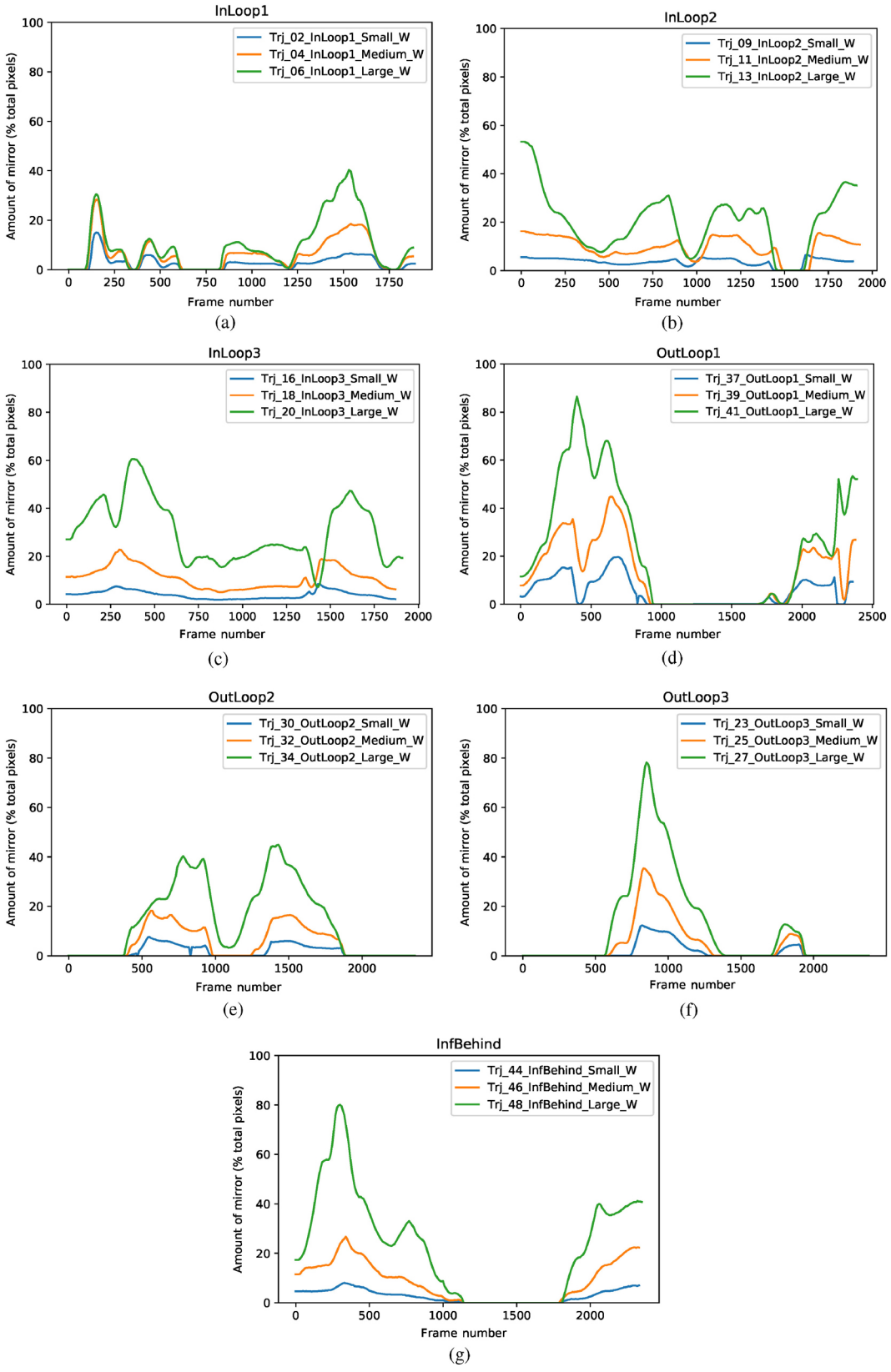


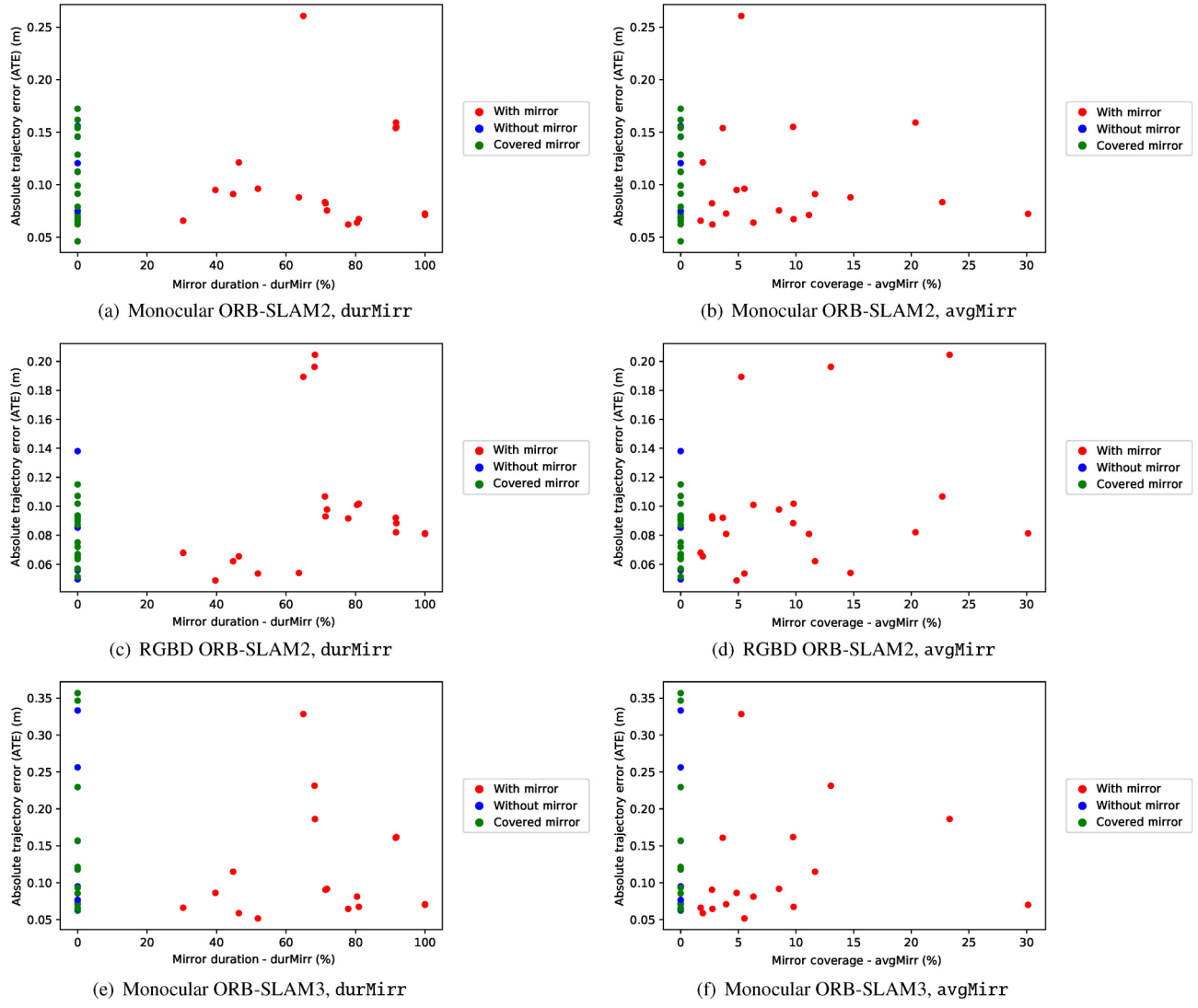
Fig. 15 Proportion of frame covered by mirror, for all sequences, grouped by trajectory. The label format is explained in Section 3.3.

Table 6 ATE (m) for each sequence. Results with significant pose estimation problems were omitted as unreliable. W = with mirror; C = covered mirror

Trajectory	Mirror	Mono. ORB. 2	RGBD ORB. 2	Mono. ORB. 3	RGBD ORB. 3	BundleFusion
InLoop1	None	0.068	0.064	0.071	0.090	0.145
	Small W	0.062	0.092	0.065	0.088	0.184
	Small C	0.070	0.092	0.073	0.089	0.214
	Medium W	0.064	0.101	0.081	0.094	0.218
	Medium C	0.062	0.093	0.066	0.062	0.217
	Large W	0.067	0.102	0.067	0.096	0.190
	Large C	0.068	0.064	0.066	0.093	0.221
InLoop2	None	0.156	0.091	0.157	0.090	—
	Small W	0.154	0.092	0.161	0.087	—
	Small C	0.155	0.092	—	0.093	—
	Medium W	0.155	0.088	0.162	0.087	0.145
	Medium C	0.154	0.094	—	0.087	—
	Large W	0.159	0.082	—	0.082	0.076
	Large C	0.162	0.090	0.157	0.086	0.158
InLoop3	None	—	0.056	0.256	0.056	0.137
	Small W	0.073	0.081	0.071	0.080	0.141
	Small C	0.046	0.056	—	0.056	0.140
	Medium W	0.071	0.081	—	0.081	0.148
	Medium C	0.072	0.075	—	0.074	0.143
	Large W	0.072	0.081	0.070	0.081	0.142
	Large C	0.065	0.067	0.066	0.066	0.151
OutLoop3	None	0.146	0.050	0.077	0.054	—
	Small W	0.066	0.068	0.066	0.069	—
	Small C	0.129	0.072	0.062	0.073	—
	Medium W	0.095	0.049	0.086	0.055	—
	Medium C	0.099	0.067	0.094	0.068	—
	Large W	0.091	0.062	0.115	0.061	—
	Large C	—	0.067	0.229	0.062	—
OutLoop2	None	0.121	0.056	0.064	0.052	0.063
	Small W	0.121	0.066	0.059	0.061	0.059
	Small C	0.112	0.052	0.065	0.047	0.065
	Medium W	0.096	0.054	0.052	0.060	0.063
	Medium C	0.113	0.057	0.118	0.066	0.062
	Large W	0.088	0.054	—	0.074	0.066
	Large C	—	0.065	0.121	0.057	0.067
OutLoop1	None	—	0.138	0.333	0.144	—
	Small W	0.261	0.189	0.328	0.127	—
	Small C	0.146	0.115	0.346	0.130	—
	Medium W	—	0.196	0.231	0.118	—
	Medium C	0.172	0.102	0.357	0.123	—
	Large W	—	0.205	0.186	0.190	—
	Large C	—	—	—	0.166	—
InfBehind	None	0.075	0.085	0.095	0.062	0.143
	Small W	0.082	0.093	0.091	0.065	0.144
	Small C	0.079	0.087	0.086	0.063	0.143
	Medium W	0.076	0.098	0.092	0.078	0.139
	Medium C	0.091	0.107	0.093	0.065	0.142
	Large W	0.083	0.107	—	0.071	0.138
	Large C	—	0.091	0.118	0.061	0.136

Table 7 RPE_t (m) for each sequence. Results with significant pose estimation problems were omitted as unreliable. W = with mirror; C = covered mirror

Trajectory	Mirror	Mono. ORB. 2	RGBD ORB. 2	Mono. ORB. 3	RGBD ORB. 3	BundleFusion
InLoop1	None	0.973	0.416	0.774	0.448	0.468
	Small W	1.149	0.442	0.940	0.439	0.466
	Small C	0.964	0.445	0.805	0.444	0.558
	Medium W	1.043	0.453	0.358	0.447	0.573
	Medium C	0.738	0.439	0.829	0.421	0.566
	Large W	0.943	0.446	0.905	0.449	0.526
	Large C	1.084	0.415	0.903	0.442	0.574
InLoop2	None	0.309	0.381	0.554	0.380	—
	Small W	0.539	0.370	0.463	0.372	—
	Small C	0.519	0.388	—	0.386	—
	Medium W	0.421	0.385	0.304	0.377	0.478
	Medium C	0.394	0.376	—	0.384	—
	Large W	0.442	0.372	—	0.381	0.371
	Large C	0.347	0.385	0.392	0.391	0.513
InLoop3	None	—	0.426	0.391	0.421	0.463
	Small W	0.333	0.388	0.328	0.393	0.450
	Small C	0.448	0.424	—	0.418	0.468
	Medium W	0.330	0.399	—	0.385	0.443
	Medium C	0.331	0.379	—	0.377	0.444
	Large W	0.346	0.404	0.339	0.402	0.446
	Large C	0.350	0.362	0.343	0.361	0.438
OutLoop3	None	0.322	0.441	0.340	0.445	—
	Small W	0.326	0.452	0.329	0.451	—
	Small C	0.337	0.456	0.330	0.460	—
	Medium W	0.335	0.428	0.317	0.441	—
	Medium C	0.335	0.449	0.328	0.448	—
	Large W	0.348	0.452	0.340	0.387	—
	Large C	—	0.438	0.986	0.429	—
OutLoop2	None	0.706	0.528	0.536	0.533	0.602
	Small W	0.947	0.522	0.456	0.517	0.593
	Small C	0.662	0.521	0.876	0.523	0.600
	Medium W	0.793	0.545	0.720	0.522	0.605
	Medium C	0.814	0.528	0.660	0.519	0.599
	Large W	0.878	0.547	—	0.518	0.613
	Large C	—	0.506	0.698	0.526	0.595
OutLoop1	None	—	0.468	1.996	0.377	—
	Small W	5.439	0.430	33.013	0.328	—
	Small C	0.365	0.323	1.075	0.331	—
	Medium W	—	0.443	3.027	0.320	—
	Medium C	0.516	0.300	2.585	0.368	—
	Large W	—	0.502	0.346	0.488	—
	Large C	—	—	—	0.486	—
InfBehind	None	0.221	0.268	0.222	0.267	0.508
	Small W	0.225	0.276	0.219	0.271	0.502
	Small C	0.221	0.268	0.227	0.253	0.505
	Medium W	0.220	0.298	0.237	0.284	0.526
	Medium C	0.288	0.286	0.263	0.250	0.517
	Large W	0.296	0.318	—	0.300	0.531
	Large C	—	0.310	0.482	0.281	0.525



(a) Monocular ORB-SLAM2, durMirr (b) Monocular ORB-SLAM2, avgMirr

(c) RGBD ORB-SLAM2, durMirr (d) RGBD ORB-SLAM2, avgMirr

(e) Monocular ORB-SLAM3, durMirr (f) Monocular ORB-SLAM3, avgMirr

Fig. 16 Scatter plots showing ATE for the three representative methods against mirror quantities. ATE for sequences without a mirror (removed or covered) were included to provide a baseline distribution for comparison to the sequences with a mirror.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

Electronic Supplementary Material

Video clips demonstrating examples of the `InLoop`, `OutLoop`, and `InfBehind` trajectories are available in the online version of this article at https://doi.org/10.1007/s41095-022-0329-x.

References

[1] Taketomi, T.; Uchiyama, H.; Ikeda, S. Visual SLAM algorithms: A survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications* Vol. 9, No. 1, 16, 2017.

[2] Mourikis, A. I.; Roumeliotis, S. I. A multi-state constraint Kalman filter for vision-aided inertial navigation. In: Proceedings of the IEEE International Conference on Robotics and Automation, 3565–3572, 2007.

[3] Mur-Artal, R.; Tardós, J. D. Visual-inertial monocular SLAM with map reuse. *IEEE Robotics and Automation Letters* Vol. 2, No. 2, 796–803, 2017.

[4] Qin, T.; Li, P. L.; Shen, S. J. VINS-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics* Vol. 34, No. 4, 1004–1020, 2018.

[5] Graeter, J.; Wilczynski, A.; Lauer, M. LIMO: Lidar-monocular visual odometry. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 7872–7879, 2018.

[6] Huang, S. S.; Ma, Z. Y.; Mu, T. J.; Fu, H. B.; Hu, S. M. Lidar-monocular visual odometry using point and line features. In: Proceedings of the IEEE International Conference on Robotics and Automation, 1091–1097, 2020.

[7] Abaspur Kazerouni, I.; Fitzgerald, L.; Dooly, G.; Toal, D. A survey of state-of-the-art on visual SLAM. *Expert Systems with Applications* Vol. 205, 117734, 2022.

[8] Huang, B. C.; Zhao, J.; Liu, J. B. A survey of simultaneous localization and mapping with an envision in 6G wireless networks. *arXiv preprint* arXiv:1909.05214, 2019.

[9] Servières, M.; Renaudin, V.; Dupuis, A.; Antigny, N. Visual and visual-inertial SLAM: State of the art, classification, and experimental benchmarking. *Journal of Sensors* Vol. 2021, 1–26, 2021.

[10] Siegwart, R.; Nourbakhsh, I. R.; Scaramuzza, D. *Introduction to Autonomous Mobile Robots*, 2nd edn. Cambridge: MIT Press, 2011.

[11] Pretto, A.; Menegatti, E.; Bennewitz, M.; Burgard, W.; Pagello, E. A visual odometry framework robust to motion blur. In: Proceedings of the IEEE International Conference on Robotics and Automation, 2250–2257, 2009.

[12] Lee, H. S.; Kwon, J.; Lee, K. M. Simultaneous localization, mapping and deblurring. In: Proceedings of the International Conference on Computer Vision, 1203–1210, 2011.

[13] Liu, P. D.; Zuo, X. X.; Larsson, V.; Pollefeys, M. MBA-VO: Motion blur aware visual odometry. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 5530–5539, 2021.

[14] Park, S.; Schöps, T.; Pollefeys, M. Illumination change robustness in direct visual SLAM. In: Proceedings of the IEEE International Conference on Robotics and Automation, 4523–4530, 2017.

[15] Huang, J. W.; Liu, S. G. Robust simultaneous localization and mapping in low-light environment. *Computer Animation and Virtual Worlds* Vol. 30, Nos. 3–4, e1895, 2019.

[16] Huang, J. H.; Yang, S.; Zhao, Z. S.; Lai, Y. K.; Hu, S. M. ClusterSLAM: A SLAM backend for simultaneous rigid body clustering and motion estimation. *Computational Visual Media* Vol. 7, No. 1, 87–101, 2021.

[17] Ma, P.; Bai, Y.; Zhu, J. N.; Wang, C. J.; Peng, C. DSOD: DSO in dynamic environments. *IEEE Access* Vol. 7, 178300–178309, 2019.

[18] Rabiee, S.; Biswas, J. IV-SLAM: Introspective vision for simultaneous localization and mapping. In: Proceedings of the 4th Conference on Robot Learning, 1100–1109, 2020.

[19] Zhou, H. Z.; Zou, D. P.; Pei, L.; Ying, R. D.; Liu, P. L.; Yu, W. X. StructSLAM: Visual SLAM with building structure lines. *IEEE Transactions on Vehicular Technology* Vol. 64, No. 4, 1364–1375, 2015.

[20] Yousif, K.; Bab-Hadiashar, A.; Hoseinnezhad, R. 3D SLAM in texture-less environments using rank order statistics. *Robotica* Vol. 35, No. 4, 809–831, 2017.

[21] Whelan, T.; Salas-Moreno, R. F.; Glocker, B.; Davison, A. J.; Leutenegger, S. ElasticFusion: Real-time dense SLAM and light source estimation. *The International Journal of Robotics Research* Vol. 35, No. 14, 1697–1716, 2016.

[22] Yang, N.; von Stumberg, L.; Wang, R.; Cremers, D. D3VO: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 1278–1289, 2020.

[23] Tan, J. Q.; Lin, W. J.; Chang, A. X.; Savva, M. Mirror3D: Depth refinement for mirror surfaces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 15985–15994, 2021.

[24] Park, D.; Park, Y. H. Identifying reflected images from object detector in indoor environment utilizing depth information. *IEEE Robotics and Automation Letters* Vol. 6, No. 2, 635–642, 2020.

[25] Koch, R.; May, S.; Koch, P.; Kühn, M.; Nüchter, A. Detection of specular reflections in range measurements for faultless robotic SLAM. In: *Robot 2015: Second Iberian Robotics Conference. Advances in Intelligent Systems and Computing, Vol. 417*. Reis, L.; Moreira, A.; Lima, P.; Montano, L.; Muñoz-Martinez, V. Eds. Springer Cham, 133–145, 2016.

[26] Yang, S. W.; Wang, C. C. Dealing with laser scanner failure: Mirrors and windows. In: Proceedings of the IEEE International Conference on Robotics and Automation, 3009–3015, 2008.

[27] Mur-Artal, R.; Montiel, J. M. M.; Tardós, J. D. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics* Vol. 31, No. 5, 1147–1163, 2015.

[28] Mur-Artal, R.; Tardós, J. D. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics* Vol. 33, No. 5, 1255–1262, 2017.

[29] Dai, A.; Nießner, M.; Zollhöfer, M.; Izadi, S.; Theobalt, C. BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics* Vol. 36, No. 4, Article No. 76a, 2017.

[30] Forster, C.; Pizzoli, M.; Scaramuzza, D. SVO: Fast semi-direct monocular visual odometry. In: Proceedings of the IEEE International Conference on Robotics and Automation, 15–22, 2014.

[31] Davison, A. J.; Reid, I. D.; Molton, N. D.; Stasse, O.

MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 29, No. 6, 1052–1067, 2007.

[32] Klein, G.; Murray, D. Parallel tracking and mapping for small AR workspaces. In: Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, 225–234, 2007.

[33] Tang, J. X.; Folkesson, J.; Jensfelt, P. Geometric correspondence network for camera motion estimation. *IEEE Robotics and Automation Letters* Vol. 3, No. 2, 1010–1017, 2018.

[34] Tang, J. X.; Ericson, L.; Folkesson, J.; Jensfelt, P. GCNv2: Efficient correspondence prediction for real-time SLAM. *IEEE Robotics and Automation Letters* Vol. 4, No. 4, 3505–3512, 2019.

[35] Engel, J.; Koltun, V.; Cremers, D. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 40, No. 3, 611–625, 2017.

[36] Engel, J.; Usenko, V.; Cremers, D. A photometrically calibrated benchmark for monocular visual odometry. *arXiv preprint* arXiv:1607.02555, 2016.

[37] Schöps, T.; Sattler, T.; Pollefeys, M. BAD SLAM: Bundle adjusted direct RGB-D SLAM. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 134–144, 2019.

[38] Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-scale direct monocular SLAM. In: *Computer Vision – ECCV 2014. Lecture Notes in Computer Science, Vol. 8690*. Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer Cham, 834–849, 2014.

[39] Gao, X.; Wang, R.; Demmel, N.; Cremers, D. LDSO: Direct sparse odometry with loop closure. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2198–2204, 2018.

[40] Forster, C.; Zhang, Z. C.; Gassner, M.; Werlberger, M.; Scaramuzza, D. SVO: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics* Vol. 33, No. 2, 249–265, 2017.

[41] Kerl, C.; Sturm, J.; Cremers, D. Dense visual SLAM for RGB-D cameras. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2100–2106, 2013.

[42] Engel, J.; Sturm, J.; Cremers, D. Semi-dense visual odometry for a monocular camera. In: Proceedings of the IEEE International Conference on Computer Vision, 1449–1456, 2013.

[43] Whelan, T.; Kaess, M.; Johannsson, H.; Fallon, M.; Leonard, J. J.; McDonald, J. Real-time large-scale dense RGB-D SLAM with volumetric fusion. *International Journal of Robotics Research* Vol. 34, Nos. 4–5, 598–626, 2015.

[44] Tateno, K.; Tombari, F.; Laina, I.; Navab, N. CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 6565–6574, 2017.

[45] Bloesch, M.; Czarnowski, J.; Clark, R.; Leutenegger, S.; Davison, A. J. CodeSLAM - Learning a compact, optimisable representation for dense visual SLAM. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2560–2568, 2018.

[46] Czarnowski, J.; Laidlow, T.; Clark, R.; Davison, A. J. DeepFactors: Real-time probabilistic dense monocular SLAM. *IEEE Robotics and Automation Letters* Vol. 5, No. 2, 721–728, 2020.

[47] Fuentes-Pacheco, J.; Ruiz-Ascencio, J.; Rendón-Mancha, J. M. Visual simultaneous localization and mapping: A survey. *Artificial Intelligence Review* Vol. 43, No. 1, 55–81, 2015.

[48] Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J. J. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics* Vol. 32, No. 6, 1309–1332, 2016.

[49] Duan, C.; Junginger, S.; Huang, J. H.; Jin, K. R.; Thurow, K. Deep learning for visual SLAM in transportation robotics: A review. *Transportation Safety and Environment* Vol. 1, No. 3, 177–184, 2019.

[50] Chen, C. H.; Wang, B.; Lu, C. X.; Trigoni, N.; Markham, A. A survey on deep learning for localization and mapping: Towards the age of spatial machine intelligence. *arXiv preprint* arXiv:2006.12567, 2020.

[51] Wang, K.; Ma, S.; Chen, J. L.; Ren, F.; Lu, J. B. Approaches, challenges, and applications for deep visual odometry: Toward complicated and emerging areas. *IEEE Transactions on Cognitive and Developmental Systems* Vol. 14, No. 1, 35–49, 2022.

[52] Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 573–580, 2012.

[53] Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3354–3361, 2012.

[54] Burri, M.; Nikolic, J.; Gohl, P.; Schneider, T.; Rehder, J.; Omari, S.; Achtelik, M. W.; Siegwart, R. The EuRoC micro aerial vehicle datasets. *International Journal of Robotics Research* Vol. 35, No. 10, 1157–1163, 2016.

[55] Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; Nießner, M. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In:

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2432–2443, 2017.

[56] Silberman, N.; Hoiem, D.; Kohli, P.; Fergus, R. Indoor segmentation and support inference from RGBD images. In: *Computer Vision – ECCV 2012. Lecture Notes in Computer Science, Vol. 7576*. Fitzgibbon, A.; Lazebnik, S.; Perona, P.; Sato, Y.; Schmid, C. Eds. Springer Berlin Heidelberg, 746–760, 2012.

[57] Ming, Y.; Ye, W.; Calway, A. iDF-SLAM: End-to-end RGB-D SLAM with neural implicit mapping and deep feature tracking. *arXiv preprint* arXiv:2209.07919, 2022.

[58] Zhu, Z. H.; Peng, S. Y.; Larsson, V.; Xu, W. W.; Bao, H. J.; Cui, Z. P.; Oswald, M. R.; Pollefeys, M. NICE-SLAM: Neural implicit scalable encoding for SLAM. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12776–12786, 2022.

[59] Handa, A.; Whelan, T.; McDonald, J.; Davison, A. J. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In: Proceedings of the IEEE International Conference on Robotics and Automation, 1524–1531, 2014.

[60] Straub, J.; Whelan, T.; Ma, L.; Chen, Y.; Wijmans, E.; Green, S.; Engel, J. J.; Mur-Artal, R.; Ren, C.; Verma, S.; et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint* arXiv:1906.05797, 2019.

[61] Wang, W. S.; Zhu, D. L.; Wang, X. W.; Hu, Y. Y.; Qiu, Y. H.; Wang, C.; Hu, Y. F.; Kapoor, A.; Scherer, S. TartanAir: A dataset to push the limits of visual SLAM. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 4909–4916, 2020.

[62] Shah, S. M. Z. A.; Marshall, S.; Murray, P. Removal of specular reflections from image sequences using feature correspondences. *Machine Vision and Applications* Vol. 28, Nos. 3–4, 409–420, 2017.

[63] Sirinukulwattana, T.; Choe, G.; Kweon, I. S. Reflection removal using disparity and gradient-sparsity via smoothing algorithm. In: Proceedings of the IEEE International Conference on Image Processing, 1940–1944, 2015.

[64] DelPozo, A.; Savarese, S. Detecting specular surfaces on natural images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1–8, 2007.

[65] Yang, X.; Mei, H. Y.; Xu, K.; Wei, X. P.; Yin, B. C.; Lau, R. Where is my mirror? In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 8808–8817, 2019.

[66] Lin, J. Y.; Wang, G. D.; Lau, R. W. H. Progressive mirror detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3694–3702, 2020.

[67] Mei, H. Y.; Dong, B.; Dong, W.; Peers, P.; Yang, X.; Zhang, Q.; Wei, X. P. Depth-aware mirror segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3043–3052, 2021.

[68] Whelan, T.; Goesele, M.; Lovegrove, S. J.; Straub, J.; Green, S.; Szeliski, R.; Butterfield, S.; Verma, S.; Newcombe, R. Reconstructing scenes with mirror and glass surfaces. *ACM Transactions on Graphics* Vol. 37, No. 4, Article No. 102, 2018.

[69] Hart, J. W.; Scassellati, B. Mirror perspective-taking with a humanoid robot. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence, 1990–1996, 2012.

[70] Zeng, Y.; Zhao, Y. X.; Bai, J. Towards robot self-consciousness (I): Brain-inspired robot mirror neuron system model and its application in mirror self-recognition. In: *Advances in Brain Inspired Cognitive Systems. Lecture Notes in Computer Science, Vol. 10023*. Liu, C. L.; Hussain, A.; Luo, B.; Tan, K.; Zeng, Y.; Zhang, Z. Eds. Springer Cham, 11–21, 2016.

[71] Safeea, M.; Neto, P. KUKA sunrise toolbox: Interfacing collaborative robots with MATLAB. *IEEE Robotics & Automation Magazine* Vol. 26, No. 1, 91–96, 2019.

[72] Safeea, M.; Neto, P. KUKA sunrise toolbox: Interfacing collaborative robots with MATLAB. *IEEE Robotics & Automation Magazine* Vol. 26, No. 1, 91–96, 2019.

[73] Shah, M.; Eastman, R. D.; Hong, T. An overview of robot-sensor calibration methods for evaluation of perception systems. In: Proceedings of the Workshop on Performance Metrics for Intelligent Systems, 15–20, 2012.

[74] Tsai, R. Y.; Lenz, R. K. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation* Vol. 5, No. 3, 345–358, 1989.

[75] Park, F. C.; Martin, B. J. Robot sensor calibration: Solving AX=XB on the Euclidean group. *IEEE Transactions on Robotics and Automation* Vol. 10, No. 5, 717–721, 1994.

[76] Andreff, N.; Horaud, R.; Espiau, B. On-line hand-eye calibration. In: Proceedings of the 2nd International Conference on 3-D Digital Imaging and Modeling, 430–436, 1999.

[77] Daniilidis, K. Hand-eye calibration using dual quaternions. *The International Journal of Robotics Research* Vol. 18, No. 3, 286–298, 1999.

[78] Sharafutdinov, D.; Griguletskii, M.; Kopanev, P.; Kurenkov, M.; Ferrer, G.; Burkov, A.; Gonnochenko,

A.; Tsetserukou, D. Comparison of modern open-source visual SLAM approaches. *arXiv preprint* arXiv:2108.01654, 2021.

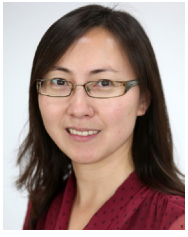
[79] Campos, C.; Elvira, R.; Rodríguez, J. J. G.; M Montiel, J. M.; D Tardós, J. ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM. *IEEE Transactions on Robotics* Vol. 37, No. 6, 1874–1890, 2021.

[80] Zhao, F. FangGet/bundlefusion_ubuntu_pangolin: Aporting for bundlefusion working on ubuntu, with Pangolin as Visualizer. 2020. Available at https://github.com/FangGet/BundleFusion_Ubuntu_Pangolin.

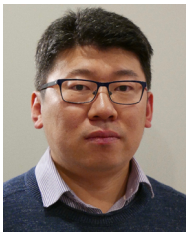
[81] Zhang, Z. C.; Scaramuzza, D. A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 7244–7251, 2018.

[82] Havasi, L.; Szlavik, Z.; Sziranyi, T. The use of vanishing point for the classification of reflections from foreground mask in videos. *IEEE Transactions on Image Processing* Vol. 18, No. 6, 1366–1372, 2009.

Peter Herbert has his B.Sc. degree in mathematics from the University of Manchester, UK, and his M.Sc. degree in data science and analytics from Cardiff University, UK. His current research interests include computer vision, machine learning, and robot navigation.



Jing Wu is a lecturer in the School of Computer Science and Informatics at Cardiff University. Her research interests are in computer vision and visual analytics. She received her B.Sc. and M.Sc. degrees from Nanjing University, China, and her Ph.D. degree from the University of York, UK. She serves on the editorial board of *Displays*, and as a Programme Committee member of CGVC, BMVC, etc.



Ze Ji received his B.Eng. degree from Jilin University, China, M.Sc. degree from the University of Birmingham, UK, and Ph.D. degree from Cardiff University. He is currently a senior lecturer in the School of Engineering at Cardiff University. Prior to his current position, he worked in industry (Dyson, Lenovo, etc.) on autonomous robotics. His research interests include autonomous navigation, robot manipulation, robot learning, simultaneous localization and mapping, and tactile sensing.

Yu-Kun Lai is a professor in the School of Computer Science and Informatics, Cardiff University. He received his bachelor and Ph.D. degrees in computer science from Tsinghua University, China, in 2003 and 2008 respectively. His research interests include computer graphics, computer vision, geometric modelling, and image processing.