



BS3D: Building-Scale 3D Reconstruction from RGB-D Images

Janne Mustaniemi^{1(✉)}, Juho Kannala², Esa Rahtu³, Li Liu¹,
and Janne Heikkilä¹

¹ Center for Machine Vision and Signal Analysis, University of Oulu, Oulu, Finland
`janne.mustaniemi@oulu.fi`

² Department of Computer Science, Aalto University, Espoo, Finland

³ Tampere University, Tampere, Finland

Abstract. Various datasets have been proposed for simultaneous localization and mapping (SLAM) and related problems. Existing datasets often include small environments, have incomplete ground truth, or lack important sensor data, such as depth and infrared images. We propose an easy-to-use framework for acquiring building-scale 3D reconstruction using a consumer depth camera. Unlike complex and expensive acquisition setups, our system enables crowd-sourcing, which can greatly benefit data-hungry algorithms. Compared to similar systems, we utilize raw depth maps for odometry computation and loop closure refinement which results in better reconstructions. We acquire a building-scale 3D dataset (BS3D) and demonstrate its value by training an improved monocular depth estimation model. As a unique experiment, we benchmark visual-inertial odometry methods using both color and active infrared images.

Keywords: Depth camera · SLAM · Large-scale

1 Introduction

Simultaneous localization and mapping (SLAM) is an essential component in robot navigation, virtual reality (VR), and augmented reality (AR) systems. Various datasets and benchmarks have been proposed for SLAM [11, 35, 39] and related problems, including visual-inertial odometry [6, 30], camera re-localization [15, 29, 32], and depth estimation [21, 33]. Currently, there exists only a few building-scale SLAM datasets [28] that include ground truth camera poses and dense 3D geometry. Such datasets enable, for example, evaluation of algorithms needed in large-scale AR applications.

The lack of building-scale SLAM datasets is explained by the difficulty of acquiring ground truth data. Some have utilized a high-end LiDAR for obtaining 3D geometry of the environment [2, 4, 26, 28]. Ground truth camera poses may be acquired using a motion capture (MoCap) system when the environment is small enough [35, 40]. The high cost of equipment, complex sensor setup, and

slow capturing process make these approaches less attractive and inconvenient for crowd-sourced data collection.

An alternative is to perform 3D reconstruction using a monocular, stereo, or depth camera. Consumer RGB-D cameras, in particular, are interesting because of their relatively good accuracy, fast acquisition speed, low-cost, and effectiveness in textureless environments. RGB-D cameras have been used to collect datasets for depth estimation [21, 33], scene understanding [8], and camera re-localization [32, 38], among other tasks. The problem is that existing RGB-D reconstruction systems (e.g. [5, 9, 22]) are limited to room-scale and apartment-scale environments.

Synthetic SLAM datasets have also been proposed [20, 27, 39] that include perfect ground truth. The challenge is that data such as time-of-flight (ToF) depth maps and infrared images are difficult to synthesize realistically. Consequently, training and evaluation done using synthetic data may not reflect algorithm’s real-world performance. To address the domain gap problem, algorithms are often fine-tuned using real data.

We propose a framework to create building-scale 3D reconstructions using a consumer depth camera (Azure Kinect). Unlike existing approaches, we register color images and depth maps using color-to-depth (C2D) strategy. This allows us to directly utilize the raw depth maps captured by the wide field-of-view (FoV) infrared camera. Coupled with an open-source SLAM library [19], we acquire a building-scale 3D vision dataset (BS3D) that is considerably larger than similar datasets as shown in Fig. 1. The BS3D dataset includes 392k synchronized color images, depth maps and infrared images, inertial measurements, camera poses, enhanced depth maps, surface reconstructions, and laser scans. Our framework will be released for the public to enable fast, easy and affordable indoor 3D reconstruction¹.

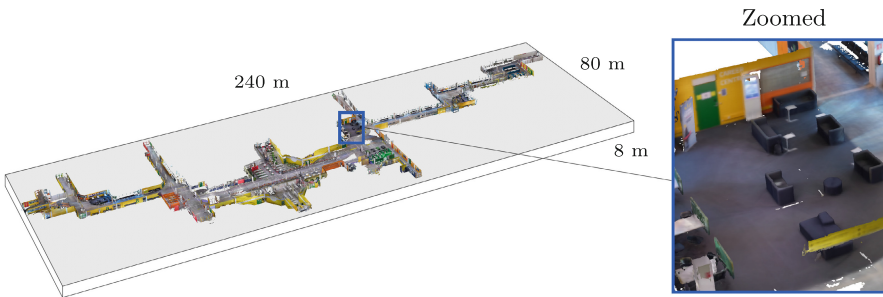


Fig. 1. Building-scale 3D reconstruction (4300 m^2) obtained using an RGB-D camera and the proposed framework. The magnified area (90 m^2) is larger than any reconstruction in the ScanNet dataset [8].

¹ <https://github.com/jannemus/BS3D>.

2 Related Work

This section introduces commonly used RGB-D SLAM datasets and corresponding data acquisition processes. A summary of the datasets is provided in Table 1. As there exist countless SLAM datasets, the scope is restricted to real-world indoor scenarios. We leave out datasets focusing on aerial scenarios (e.g. EuRoC MAV [2]) and autonomous driving (e.g. KITTI [11]). We also omit RGB-D datasets captured with a stationary scanner (e.g. Matterport3D [4]) as they cannot be used for SLAM evaluation. Synthetic datasets, such as SceneNet RGB-D [20], TartanAir [39], and ICL [27] are also omitted.

ADVIO [6] dataset is a realistic visual-inertial odometry benchmark that includes building-scale environments. Ground truth trajectory is computed using an inertial navigation system (INS) together with manual location fixes. The main limitation of the dataset is that it does not come with ground truth 3D geometry. LaMAR [28] is a large-scale SLAM benchmark that utilizes high-end mapping platforms (NavVis M6 trolley and VLX backpack) for ground truth generation. Although the capturing setup includes a variety of devices (e.g. HoloLens2 and iPad Pro), it does not include a dedicated RGB-D camera.

OpenLORIS-Scene [31] focuses on the lifelong SLAM scenario where environments are dynamic and changing, similar to LaMAR [28]. The data is collected over an extended period of time using wheeled robots equipped with various sensors, including RGB-D, stereo, IMU, wheel odometry, and LiDAR. Ground truth poses are acquired using an external motion capture (MoCap) system, or with a 2D laser SLAM method. The dataset is not ideal for handheld SLAM evaluation because of the limited motion patterns of a ground robot.

TUM RGB-D SLAM [35] is one of the most popular SLAM datasets. The RGB-D images are acquired using a consumer depth camera Microsoft Kinect v1. Ground truth trajectory is incomplete because the MoCap system can only cover a small part of the environment. CoRBS [40] consists of four room-scale environments. It also utilizes MoCap for acquiring ground truth trajectories for Microsoft Kinect v2. Unlike [35], CoRBS provides ground truth 3D geometry acquired using a laser scanner. The data also includes infrared images, but not inertial measurements, unlike our dataset.

7-Scenes [32] and 12-Scenes [38] are commonly used for evaluating camera localization. 7-Scenes includes seven scenes captured using Kinect v1. KinectFusion [22] is used to obtain ground truth poses and dense 3D models from the RGB-D images. 12-Scenes consists of multiple rooms captured using the Structure.io depth sensor and iPad color camera. The reconstructions are larger compared to 7-Scenes, about 37 m³ on average. They are acquired using the VoxelHashing framework [23], an alternative to KinectFusion with better scalability.

ScanNet [8] is an RGB-D dataset containing 2.5M views acquired in 707 distinct spaces. It includes estimated calibration parameters, camera poses, 3D surface reconstructions, textured meshes, and object-level semantic segmentations. The hardware consists of a Structure.io depth sensor attached to a tablet computer. Pose estimation is done using BundleFusion [9], after which volumetric integration is performed through VoxelHashing [23].

Sun3D [43] is a large RGB-D database with camera poses, point clouds, object labels, and refined depth maps. The reconstruction process is based on structure from motion (SfM) where manual object annotations are utilized to reduce drift and loop-closure failures. Refined depth maps are obtained via volumetric fusion similar to KinectFusion [22]. We emphasize that ScanNet [8] and Sun3D [43] reconstructions are considerably smaller and have lower quality than those provided in our dataset. Unlike [28, 31, 35], our system also does not require a complex and expensive capturing setup, or manual annotation [6, 43].

Table 1. List of indoor RGB-D SLAM datasets. The BS3D acquisition setup does not require high-end LiDARs [28, 31, 40], MoCap systems [31, 36, 40], or manual annotation [6, 43]. BS3D is building-scale, unlike [8, 32, 36, 38, 40, 43]. Note that ADVIO [6] and LaMAR [28] do not have a dedicated depth camera.

Dataset	Scale	Depth	IMU	IR	Ground truth
7-Scenes [32]	room	Kinect v1	–	–	RGBD-recons
TUM RGBD [36]	room	Kinect v1	✓	–	MoCap
ScanNet [8]	room	Structure.io	✓	–	RGBD-recons
CoRBS [40]	room	Kinect v2	–	✓	MoCap+LiDAR
12-Scenes [38]	apartment	Structure.io	–	–	RGBD-recons
Sun3D [43]	apartment	Xtion Pro Live	–	–	RGBD+manual
OpenLORIS [31]	building	RS-D435i	✓	–	MoCap+LiDAR
ADVIO [6]	building	Tango	✓	–	INS+manual
LaMAR [28]	building	HoloLens2	✓	✓	LiDAR+VIO+SfM
BS3D (ours)	building	Azure Kinect	✓	✓	RGBD-recons

3 Reconstruction Framework

In this section, we introduce the RGB-D reconstruction framework shown in Fig. 2. The framework produces accurate 3D reconstructions of building-scale environments using low-cost hardware. The system is fully automatic and robust against poor lighting conditions and fast motions. Color images are only used for loop closure detection as they are susceptible to motion blur and rolling shutter distortion. Raw depth maps enable accurate odometry and the refinement of loop closure transformations.

3.1 Hardware

Data is captured using the Azure Kinect depth camera, which is well-suited for crowd-sourcing due to its popularity and affordability. We capture synchronized depth, color, and infrared images 30 Hz using the official recorder application running on a laptop computer. We use the wide FoV mode of the infrared camera with 2×2 binning to extend the Z-range. The resolution of raw depth maps and

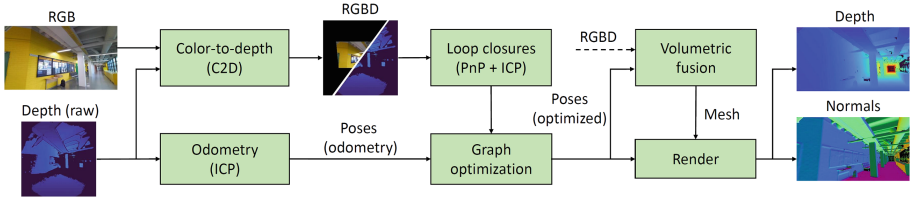


Fig. 2. Overview of the RGB-D reconstruction system.

IR images is 512×512 pixels. Auto-exposure is enabled when capturing color images at the resolution of 720×1280 pixels. We also record accelerometer and gyroscope readings at 1.6 kHz.

3.2 Color-to-Depth Alignment

Most RGB-D reconstruction systems expect that color images and depth maps have been spatially and temporally aligned. Modern depth cameras typically produce temporally synchronized images so the main concern is the spatial alignment. Conventionally, raw depth maps are transformed to the coordinate system of the color camera, which we refer to as the depth-to-color (D2C) alignment.

In the case of Azure Kinect, the color camera’s FoV is much narrower ($90 \times 59^\circ$) compared to the infrared camera ($120 \times 120^\circ$). Thus, the D2C alignment would not take advantage of the infrared camera’s wide FoV because depth maps would be heavily cropped. Moreover, the D2C alignment might introduce artefacts to the raw depth maps.

We propose an alternative called color-to-depth (C2D) alignment where color images are transformed instead. In the experiments, we show that this drastically improves the quality of the reconstructions. The main challenge of C2D is that it requires a fully dense depth map. Fortunately, a reasonably good alignment can be achieved even with a low quality depth map. This is because the baseline between the cameras is narrow and missing depths often appear in areas that are far away from the camera.

For the C2D alignment, we first perform depth inpainting using linear interpolation. Then, the color image is transformed to the raw depth frame. To keep as much of the color information as possible, the output resolution will be higher (1024×1024 pixels) compared to the raw depth maps. After that, holes in the color image due to occlusions are inpainted using the OpenCV library’s implementation of [37]. We note that minor artefacts in the aligned color images will have little impact on the SIFT-based loop closure detection.

3.3 RGB-D Mapping

We process the RGB-D sequences using an open-source SLAM library called RTAB-Map [19]. Odometry is computed from the raw depth maps using the

point-to-plane variant of the iterative closest point (ICP) algorithm [25]. We use the scan-to-map odometry strategy [19] where incoming frames are registered against a point cloud map created from past keyframes. The wide FoV ensures that ICP-odometry rarely fails, but in case it does, a new map is initialized.

Loop closure detection is needed for drift correction and merging of individual maps. For this purpose, SIFT features are extracted from the valid area of the aligned color images. Loop closures are detected using the bag-of-words approach [18], and transformations are estimated using the Perspective-n-Point RANSAC algorithm and refined using ICP [25]. Graph optimization is done using the GTSAM library [10] and Gauss-Newton algorithm.

RTAB-Map supports multi-session mapping which is a necessary feature when reconstructing building-scale environments. It is not practical to collect possibly hours of data at once. Furthermore, having the ability to later update and expand the map is a useful feature. In practice, individual sequences are first processed separately, followed by multi-session mapping. The sessions are merged by finding loop closures and by performing graph optimization. The input is a sequence of keyframes along with odometry poses and SIFT features computed during single-session mapping. The sessions are processed in such order that there is at least some overlap between the current session and the global map build so far.

3.4 Surface Reconstruction

It is often useful to have a 3D surface representation of the environment. There exists many classical [14, 22] and learning-based [1, 41] surface reconstruction approaches. Methods that utilize deep neural networks, such as NeuralFusion [41], have produced impressive results on the task of depth map fusion. Neural radiance fields (NeRFs) have also been adapted to RGB-D imagery [1] showing good performance. We did not use learning-based approaches in this work because they are typically limited to small scenes, at least for the time being. Moreover, scene-specific learning [1] takes several hours even with powerful hardware.

Surface reconstruction is done in segments due to the large scale of the environment and the vast number of frames. To that end, we first create a point cloud from downsampled raw depth maps. Every point includes a view index along with 3D coordinates. The point cloud is partitioned into manageable segments using the K-means algorithm. A mesh is created for each segment using the scalable TSDF fusion implementation [46] that is based on [7, 22]. It uses a hierarchical hashing structure to support large scenes.

4 BS3D Dataset

The BS3D dataset was collected at the university campus using Azure Kinect (Sect. 3.1). Figure 3 shows example frames from the dataset. The collection was done in multiple sessions due to large scale of the environment. The recordings were processed using the framework described in Sect. 3.

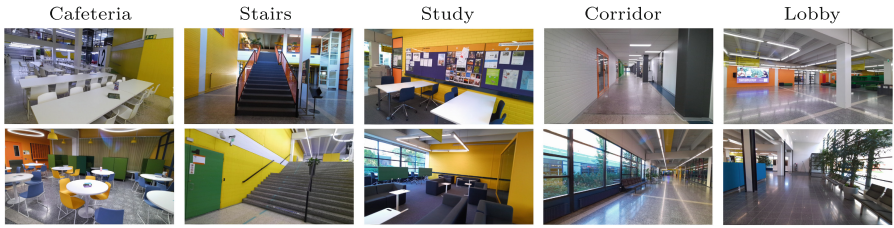


Fig. 3. Example frames from the dataset. Environments are diverse and challenging, including cafeterias, stairs, study areas, corridors, and lobbies.

4.1 Dataset Features

The reconstruction shown in Fig. 1 consists of 47 overlapping recording sessions. Additional 14 sessions, including 3D laser scans, were recorded for evaluation purposes. Most sessions begin and end at the same location to encourage loop closure detection. The total duration of the sessions is 3 h and 38 min and the combined trajectory length is 6.4 km. The reconstructed floor area is approximately 4300 m².

The dataset consists of 392k frames, including color images, raw depth maps, and infrared images. Color images and depth maps are provided in both coordinate frames (color and infrared camera). The images have been undistorted for convenience, but the original recordings are also included. We provide camera poses in a global reference frame for every image. Data also includes inertial measurements, enhanced depth maps and surface normals that have been rendered from the mesh as visualized in Fig. 4.

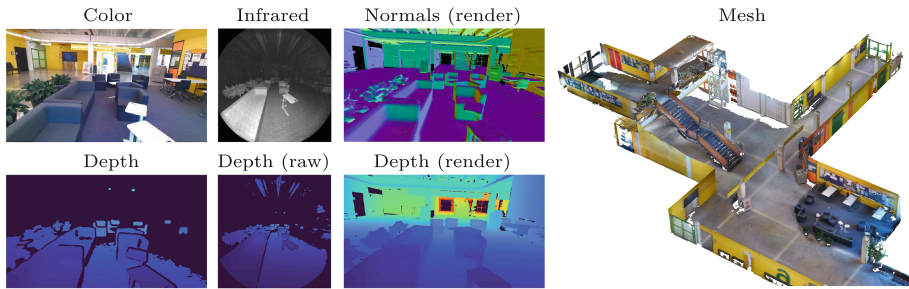


Fig. 4. The BS3D dataset includes color and infrared images, depth maps, IMU data, camera parameters, and surface reconstructions. Enhanced depth maps and surface normals are rendered from the mesh.

4.2 Laser Scan

We utilize the FARO 3D X 130 laser scanner for acquiring ground truth 3D geometry. The scanned area includes a lobby and corridors of different sizes (800 m²). The 28 individual scans were registered using the SCENE software that comes with the laser scanner. Noticeable artefacts, e.g. those caused by mirrors, were manually removed. The laser scan is used to evaluate the reconstruction framework in Sect. 5. However, this data also enables, for example, training and evaluation of RGB-D surface reconstruction algorithms.

5 Experiments

We compare our framework with the state-of-the-art RGB-D reconstruction methods [3, 5, 9]. The value of the BS3D dataset is demonstrated by training a recent monocular depth estimation model [44]. We also benchmark visual-inertial odometry approaches [3, 12, 34] using either color or infrared images to further highlight the unique aspects of the BS3D dataset.

5.1 Reconstruction Framework

In this experiment, we compare the framework against Redwood [5], BundleFusion [9], and ORB-SLAM3 [3]. RGBD images are provided for [3, 5, 9] in the coordinate frame of the color camera. Given the estimated camera poses, we create a point cloud and compare it to the laser scan (Sect. 4.2). The evaluation metrics include overlap of the point clouds and RMSE of inlier correspondences. Before comparison, we create uniformly sampled point clouds using voxel down-sampling (1 cm³ voxel) that computes the centroid of the points in each voxel. The overlap is defined as the ratio of inlier correspondences and the number of ground truth points. A 3D point is considered to be an inlier if the distance to the closest ground truth point is below threshold γ .

Table 2 shows the results for environments of different sizes. All methods are able to reconstruct the small environment (35 m²) consisting of 2.8k frames. The differences between the methods become more evident when reconstructing the medium-size environment (160 m²) consisting of 7.3k frames. BundleFusion [9] only produces a partial reconstruction because of odometry failures. The proposed approach gives the most accurate reconstructions as visualized in Fig. 5. Note that it is not possible to achieve 100% overlap because the depth camera does not observe all parts of the ground truth.

The largest environment (350 m²) consists of 24k frames acquired in four sessions. Redwood [5] does not scale to input sequences of this long. ORB-SLAM3 [3] frequently loses the odometry in open spaces which leads to incomplete and less accurate reconstructions. Our method suffers the same problem when C2D is disabled. Unreliable odometry is likely due to the color camera's limited FoV, rolling shutter distortion, and motion blur. The C2D alignment improves the accuracy and robustness of ICP-based odometry and loop closures.

Without C2D, the frequent odometry failures result in disconnected maps and noticeable drift. We note that the reconstruction in Fig. 1 was computed from $\sim 300k$ frames which is far more than [3, 5, 9] can handle.

Table 2. Comparison of RGB-D reconstruction methods in small, medium and large-scale environments (from top to bottom). Overlap of the point clouds and inlier RMSE computed for distance thresholds γ (mm). Some methods only work in small and/or medium scale environments.

Method	$\gamma = 10$ (mm)		$\gamma = 20$ (mm)		$\gamma = 50$ (mm)	
	Overlap \uparrow	RMSE \downarrow	Overlap \uparrow	RMSE \downarrow	Overlap \uparrow	RMSE \downarrow
Redwood [5]	66.5	5.6	77.9	7.6	87.1	12.6
BundleFusion [9]	72.1	5.5	80.8	6.9	88.3	11.7
ORB-SLAM3 [3]	78.2	5.3	85.2	6.5	91.3	10.6
Prop. (w/o C2D)	66.8	5.7	77.8	7.5	87.0	12.7
Proposed	78.4	5.2	85.7	6.5	91.6	10.6
Redwood [5]	30.4	6.2	44.5	9.8	63.9	19.9
BundleFusion [9]	8.1	6.2	11.1	9.2	14.8	18.8
ORB-SLAM3 [3]	44.3	6.0	57.7	8.7	71.0	16.2
Prop. (w/o C2D)	36.5	6.1	49.2	9.0	64.3	18.3
Proposed	54.1	5.7	64.8	7.7	73.2	13.4
ORB-SLAM3 [3]	9.5	6.3	14.4	9.9	20.8	20.7
Prop. (w/o C2D)	23.1	6.7	40.6	10.9	64.7	22.4
Proposed	34.7	6.4	52.7	10.0	75.0	19.8

5.2 Depth Estimation

We investigate whether the BS3D dataset can be used to train better models for monocular depth estimation. For this experiment, we use the state-of-the-art LeReS model [44] based on ResNet50. The original model has been trained using 354k samples taken from various datasets [13, 16, 24, 42, 45]. We finetune the model using 16.5k samples from BS3D. We set the learning rate to $2e-5$ and train only 4 epochs to avoid overfitting. Other training details, including loss functions are the same as in [44].

For testing, we use NYUD-v2 [21] and iBims-1 [17] that are not seen during training. We also evaluate using BS3D by sampling 535 images from an unseen part of the building. Table 3 shows that finetuning improves the performance on iBims-1 and BS3D. The finetuned model performs marginally worse on NYUD-v2 which is not surprising considering that NYUD-v2 mainly contains room-scale scenes that are not present in BS3D. The qualitative comparison in Fig. 6 also shows a clear improvement over the pretrained model on iBims-1 that contains both small and large scenes. The model trained only using BS3D cannot compete with other models, except on BS3D on which the performance is surprisingly

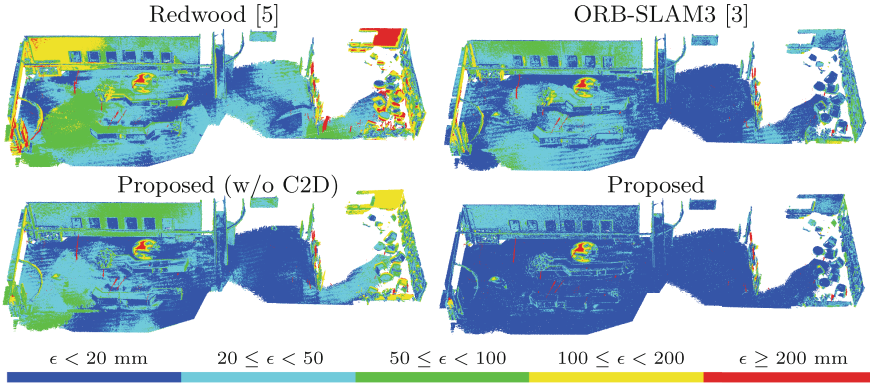


Fig. 5. Reconstructions obtained using Redwood [5], ORB-SLAM3 [3], and the proposed method. Colors depict errors (distance to the closest ground truth point).

good. The poor performance on other datasets is not surprising because of the small training set.

Table 3. Monocular depth estimation using LeReS [44] trained from scratch using BS3D, pretrained model, and finetuned model. NUYD-v2 [21], iBims-1 [17], and BS3D are used for testing.

Training data	NYUD-v2 [21]		iBims-1 [17]		BS3D	
	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑
BS3D	0.181	0.764	0.188	0.763	0.144	0.828
Pretrained	0.096	0.913	0.115	0.890	0.157	0.785
Pre. + BS3D	0.100	0.907	0.098	0.901	0.115	0.881

5.3 Visual-Inertial Odometry

The BS3D dataset includes active infrared images along with color and IMU data. This opens interesting possibilities, for example, the comparison of color and infrared as inputs for visual-inertial odometry. Infrared-inertial odometry is an attractive approach in the sense that it does not require external light, meaning it would work in completely dark environments.

We evaluate OpenVINS [12], ORB-SLAM3 [3], and DM-VIO [34] using color-inertial and infrared-inertial inputs. Note that ORB-SLAM3 has an unfair advantage because it has a loop closure detector that cannot be disabled. In the case of infrared images, we apply a power law transformation ($\bar{I} = 0.04 \cdot I^{0.6}$) to increase brightness. As supported by [34], we provide a mask of valid pixels to ignore black areas near the edges of the infrared images. We adjust the parameters related to feature detection when using infrared images with [3, 12].

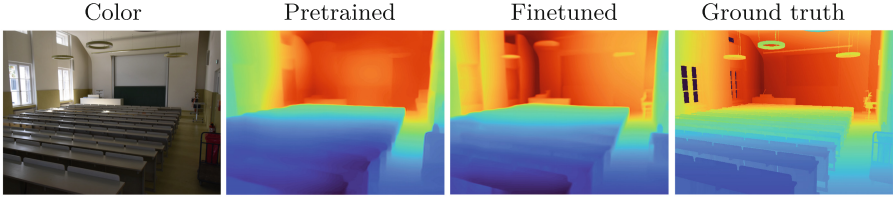


Fig. 6. Comparison of pre-trained and finetuned (BS3D) monocular depth estimation model LeReS [44] on an independent iBims-1 [17] dataset unseen during training.

We use the standard error metrics, namely absolute trajectory error (ATE) and relative pose error (RPE) which measures the drift per second. The methods are evaluated 5 times on each of the 10 sequences (Table 4).

From the results in Table 5, we can see that ORB-SLAM3 has the lowest ATE when evaluating color-inertial odometry, mainly because of loop closure detection. In most cases, ORB-SLAM3 and OpenVINS fail to initialize when using infrared images. We conclude that off-the-shelf feature detectors (FAST and ORB) are quite poor at detecting good features from infrared images. Interestingly, DM-VIO performs better when using infrared images instead of color which is likely due to the infrared camera’s global shutter and wider FoV. This result reveals the great potential of using active infrared images for visual-inertial odometry and the need for new research.

Table 4. Evaluation sequences used in the visual-inertial odometry experiment. Last column shows if the camera returns to the starting point (chance for a loop closure).

Sequence	Duration (s)	Length (m)	Dimensions (m)	Loop
cafeteria	200	90.0	$12.4 \times 15.7 \times 0.8$	✓
central	242	155.0	$25.5 \times 42.1 \times 5.3$	✓
dining	192	109.2	$33.8 \times 25.0 \times 5.5$	✓
corridor	174	77.6	$31.1 \times 4.7 \times 2.4$	✓
foobar	75	37.1	$5.4 \times 14.4 \times 0.6$	✓
hub	124	52.3	$11.4 \times 5.9 \times 0.7$	–
juice	103	42.7	$6.3 \times 8.6 \times 0.5$	–
lounge	222	94.2	$14.4 \times 10.3 \times 1.1$	✓
study	87	40.0	$5.6 \times 9.8 \times 0.6$	–
waiting	139	60.1	$9.8 \times 6.7 \times 0.9$	✓

Table 5. Comparison of visual-inertial odometry methods using color-inertial and infrared-inertial inputs. Average absolute trajectory error (ATE) and relative pose error (RPE). Last column shows the percentage of successful runs.

Method	Color-inertial odometry				Infrared-inertial odometry			
	ATE ↓ (m)	RPE ↓ (deg/s)	RPE ↓ (m/s)	Succ. ↑ (%)	ATE ↓ (m)	RPE ↓ (deg/s)	RPE ↓ (m/s)	Succ. ↑ (%)
OpenVINS [12]	0.347	0.37	0.031	76.0	0.597	0.42	0.057	36.0
ORB-SLAM3 [3]	0.298	0.29	0.026	100.0	0.193	0.29	0.025	24.0
DM-VIO [34]	0.491	0.29	0.033	100.0	0.433	0.29	0.025	100.0

6 Conclusion

We presented a framework for acquiring high-quality 3D reconstructions using a consumer depth camera. The ability to produce building-scale reconstructions is a significant improvement over existing methods that are limited to smaller environments such as rooms or apartments. The proposed C2D alignment enables the use of raw depth maps, resulting in more accurate 3D reconstructions. Our approach is fast, easy to use, and requires no expensive hardware, making it ideal for crowd-sourced data collection. We acquire building-scale 3D dataset (BS3D) and demonstrate its value for monocular depth estimation. BS3D is unique also because it includes active infrared images, which are often missing in other datasets. We employ infrared images for visual-inertial odometry, discovering a promising new research direction.

References

1. Azinović, D., Martin-Brualla, R., Goldman, D.B., Nießner, M., Thies, J.: Neural RGB-D surface reconstruction. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6290–6301 (2022)
2. Burri, M., et al.: The EuRoC micro aerial vehicle datasets. *Int. J. Robot. Res.* **35**(10), 1157–1163 (2016)
3. Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M., Tardós, J.D.: ORB-SLAM3: an accurate open-source library for visual, visual-inertial, and multimap SLAM. *IEEE Trans. Rob.* **37**(6), 1874–1890 (2021)
4. Chang, A., et al.: Matterport3D: learning from RGB-D data in indoor environments. arXiv preprint [arXiv:1709.06158](https://arxiv.org/abs/1709.06158) (2017)
5. Choi, S., Zhou, Q.Y., Koltun, V.: Robust reconstruction of indoor scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5556–5565 (2015)
6. Cortés, S., Solin, A., Rahtu, E., Kannala, J.: ADVIO: an authentic dataset for visual-inertial odometry. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11214, pp. 425–440. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01249-6_26
7. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Conference on Computer Graphics and Interactive Techniques, pp. 303–312 (1996)

8. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: ScanNet: richly-annotated 3D reconstructions of indoor scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5828–5839 (2017)
9. Dai, A., Nießner, M., Zollhöfer, M., Izadi, S., Theobalt, C.: BundleFusion: real-time globally consistent 3D reconstruction using on-the-fly surface reintegration. *ACM Trans. Graph. (ToG)* **36**(4), 1 (2017)
10. Dellaert, F.: Factor graphs and GTSAM: a hands-on introduction. Technical report, Georgia Institute of Technology (2012)
11. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: the KITTI dataset. *Int. J. Robot. Res.* **32**(11), 1231–1237 (2013)
12. Geneva, P., Eckenhoff, K., Lee, W., Yang, Y., Huang, G.: OpenVINS: a research platform for visual-inertial estimation. In: International Conference on Robotics and Automation (ICRA), pp. 4666–4672. IEEE (2020)
13. Hua, Y., et al.: Holopix50k: a large-scale in-the-wild stereo image dataset. arXiv preprint [arXiv:2003.11172](https://arxiv.org/abs/2003.11172) (2020)
14. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the Fourth Eurographics Symposium on Geometry Processing, vol. 7 (2006)
15. Kendall, A., Grimes, M., Cipolla, R.: PoseNet: a convolutional network for real-time 6-DoF camera relocalization. In: IEEE International Conference on Computer Vision (ICCV), pp. 2938–2946 (2015)
16. Kim, Y., Jung, H., Min, D., Sohn, K.: Deep monocular depth estimation via integration of global and local predictions. *IEEE Trans. Image Process.* **27**(8), 4131–4144 (2018)
17. Koch, T., Liebel, L., Fraundorfer, F., Körner, M.: Evaluation of CNN-based single-image depth estimation methods. In: Leal-Taixé, L., Roth, S. (eds.) ECCV 2018. LNCS, vol. 11131, pp. 331–348. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11015-4_25
18. Labbe, M., Michaud, F.: Appearance-based loop closure detection for online large-scale and long-term operation. *IEEE Trans. Rob.* **29**(3), 734–745 (2013)
19. Labbé, M., Michaud, F.: RTAB-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *J. Field Robot.* **36**(2), 416–446 (2019)
20. McCormac, J., Handa, A., Leutenegger, S., Davison, A.J.: SceneNet RGB-D: can 5 m synthetic images beat generic ImageNet pre-training on indoor segmentation? In: IEEE International Conference on Computer Vision (ICCV), pp. 2678–2687 (2017)
21. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7576, pp. 746–760. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33715-4_54
22. Newcombe, R.A., et al.: KinectFusion: real-time dense surface mapping and tracking. In: IEEE International Symposium on Mixed and Augmented Reality, pp. 127–136. IEEE (2011)
23. Nießner, M., Zollhöfer, M., Izadi, S., Stamminger, M.: Real-time 3D reconstruction at scale using voxel hashing. *ACM Trans. Graph. (ToG)* **32**(6), 1–11 (2013)
24. Niklaus, S., Mai, L., Yang, J., Liu, F.: 3D Ken Burns effect from a single image. *ACM Trans. Graph. (ToG)* **38**(6), 1–15 (2019)
25. Pomerleau, F., Colas, F., Siegwart, R., Magnenat, S.: Comparing ICP variants on real-world data sets. *Auton. Robot.* **34**(3), 133–148 (2013)

26. Ramakrishnan, S.K., et al.: Habitat-Matterport 3D Dataset (HM3D): 1000 large-scale 3D environments for embodied AI. arXiv preprint [arXiv:2109.08238](https://arxiv.org/abs/2109.08238) (2021)
27. Saeedi, S., et al.: Characterizing visual localization and mapping datasets. In: International Conference on Robotics and Automation (ICRA), pp. 6699–6705. IEEE (2019)
28. Sarlin, P.E., et al.: LaMAR: benchmarking localization and mapping for augmented reality. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) European Conference on Computer Vision (ECCV), ECCV 2022, vol. 13667, pp. 686–704. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20071-7_40
29. Sattler, T., et al.: Benchmarking 6DoF outdoor visual localization in changing conditions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8601–8610 (2018)
30. Schubert, D., Goll, T., Demmel, N., Usenko, V., Stückler, J., Cremers, D.: The TUM VI benchmark for evaluating visual-inertial odometry. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1680–1687. IEEE (2018)
31. Shi, X., et al.: Are we ready for service robots? The OpenLORIS-scene datasets for lifelong SLAM. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 3139–3145. IEEE (2020)
32. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in RGB-D images. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2930–2937 (2013)
33. Song, S., Lichtenberg, S.P., Xiao, J.: Sun RGB-D: a RGB-D scene understanding benchmark suite. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 567–576 (2015)
34. von Stumberg, L., Cremers, D.: DM-VIO: delayed marginalization visual-inertial odometry. *IEEE Robot. Autom. Lett.* **7**(2), 1408–1415 (2022)
35. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: International Conference on Intelligent Robot Systems (IROS), October 2012
36. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 573–580. IEEE (2012)
37. Telea, A.: An image inpainting technique based on the fast marching method. *J. Graph. Tools* **9**(1), 23–34 (2004)
38. Valentin, J., et al.: Learning to navigate the energy landscape. In: Fourth International Conference on 3D Vision (3DV), pp. 323–332. IEEE (2016)
39. Wang, W., et al.: TartanAir: a dataset to push the limits of visual SLAM. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4909–4916. IEEE (2020)
40. Wasenmüller, O., Meyer, M., Stricker, D.: CoRBS: comprehensive RGB-D benchmark for SLAM using Kinect v2. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1–7. IEEE (2016)
41. Weder, S., Schonberger, J.L., Pollefeys, M., Oswald, M.R.: NeuralFusion: online depth fusion in latent space. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3162–3172 (2021)
42. Xian, K., Zhang, J., Wang, O., Mai, L., Lin, Z., Cao, Z.: Structure-guided ranking loss for single image depth prediction. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 611–620 (2020)

43. Xiao, J., Owens, A., Torralba, A.: Sun3D: a database of big spaces reconstructed using SfM and object labels. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1625–1632 (2013)
44. Yin, W., et al.: Learning to recover 3D scene shape from a single image. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
45. Zamir, A.R., Sax, A., Shen, W., Guibas, L.J., Malik, J., Savarese, S.: Taskonomy: disentangling task transfer learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3712–3722 (2018)
46. Zhou, Q.Y., Park, J., Koltun, V.: Open3D: a modern library for 3D data processing. arXiv preprint [arXiv:1801.09847](https://arxiv.org/abs/1801.09847) (2018)