2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

CamVox: A Low-cost and Accurate Lidar-assisted Visual SLAM System

Yuewen Zhu, Chunran Zheng, Chongjian Yuan, Xu Huang and Xiaoping Hong

Abstract— **Combining lidar in camera-based simultaneous localization and mapping (SLAM) is an effective method in improving overall accuracy, especially at outdoor large scale scenes. Recent development of low-cost lidars (e.g. Livox lidar) enable us to explore such SLAM systems with lower budget and higher performance. In this paper we propose CamVox by adapting Livox lidars into visual SLAM (ORB-SLAM2) by exploring the lidars' unique features. Based on the unique scan pattern of Livox lidars, we propose an automatic lidar-camera calibration method that will work in uncontrolled scenes. The long depth detection range also benefit a more accurate mapping. Comparison of CamVox with visual SLAM (VINS-mono) and lidar SLAM (LOAM) are evaluated on the same dataset to demonstrate the performance. We open sourced our hardware, code and dataset on GitHub[1].**

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) is a key technique in autonomous robots with growing attention from academia and industry. Initially cameras provided rich angular and color information and enabled the development of visual SLAM such as ORB-SLAM [1]. Following that the inertial measurement units (IMU) had the cost dropped thanks to their massive adoption in smart phone industry, and the utilization in SLAM becomes straightforward to gain additional modality and performance such as in VINS-mono [2] and ORB-SLAM3 [3]. Among the additional sensors, depth sensors (stereo camera, RGB-D camera) provide direct depth measurement and enables accurate performance in SLAM applications such as ORB-SLAM2 [4]. Lidar, as the high-end depth sensor, provides long range outdoor capability, accurate measurements and system robustness, and has been widely adopted in more demanding applications such as autonomous driving [5], but also typically comes with a hefty price tag. As the autonomous industry progresses, recently many new technology developments have enabled commercialization of low-cost lidars, e.g. Ouster and Livox lidars.

Featuring a non-repeating scanning pattern, Livox lidars pose unique advantageous in low-cost lidar-assisted SLAM system. We in this paper present the first Livox lidar assisted visual SLAM system (CamVox) with accurate and real-time performance. Our CamVox SLAM built upon the state-of-the-art ORB-SLAM2 by using Livox lidar as the depth sensor, with the following contributions:

1) Extrinsic calibration of lidar and camera can sometimes be very challenging, especially in the field where no calibration target could be manually setup. The non-repeating scanning Livox lidar introduced a new type of lidar scanning. This unique feature can be utilized to device an automatic extrinsic calibration procedure between the camera and the lidar at almost any scenes without controlled target.
2) Thanks to the strong sunlight-resisting performance of the lidar, the proposed SLAM system was field evaluated at an outdoor large-scale scene under strong sunlight. CamVox demonstrated high accuracy and robustness compared to other state-of-the-art outdoor SLAM frameworks.
3) We open-sourced the hardware, code and dataset of this work to provide an out-of-the-box SLAM solution. To our knowledge this is also the first open-sourced visual SLAM solution assisted by a Livox lidar.

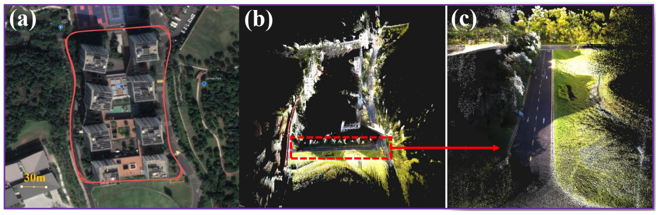

Fig. 1. Example of CamVox performance. (a) the trajectory of the robot from CamVox; (b) the dense RGBD map from CamVox; (c) a magnified and rotated part of (b) demonstrating high point cloud quality.

Fig. 1 shows an example trajectory and map construction from CamVox. The rest of this paper is structured as the following. Related work is reviewed in Section II. The CamVox system hardware and software framework is described in detail in Section III. The results and evaluation are presented in Section IV. Finally, we conclude and remark the outlook in Section V.

II. RELATED WORK

Due to the rich angular resolution and informative color information, cameras could provide surprisingly good localization and mapping performance through a simple projection model and bundle adjustment. One of the most famous SLAM systems with monocular camera is ORB-SLAM [1]. ORB-SLAM tracks the object by extract the ORB features in the image and use loop-closure detection to optimize the map and pose globally, which is usually fast and stable. However, it cannot accurately recover the real scale factor since an absolute depth scale is unknown to a camera.

*Research is supported by SUSTech start-up fund and DJI-SUSTech joint lab.

All authors are with the School of System Design and Intelligent Manufacturing, Southern University of Science and Technology, China {zhuyw2019,zhengcr,ycj2020,huangx2020}@mail.sustech.edu.cn, hongxp@sustech.edu.cn.

[1] https://github.com/ISEE-Technology/CamVox

978-1-7281-9077-8/21/$31.00 ©2021 IEEE

To improve over this issue, Mur et al. proposed ORB-SLAM2 [4], adding the support of stereo camera and RGBD camera for depth estimation. However, there are drawbacks with both of these sensors, especially in estimating the outdoor objects with long depths. The stereo camera requires a long baseline for accurate long-depth estimation, which is usually limited in real world scenes. Additionally, the calibration between the two cameras is susceptible to mechanical changes and will adversely influence the long-depth estimation accuracy. The RGBD camera is usually susceptible to sun light with a finite range of less than 10 meters typically.

Fusing camera and IMU is another common solution, because camera can partially correct IMU integral drift, calibrate IMU bias while IMU can overcome the scale ambiguity of monocular system. Qin et al. proposed Visual-Inertial Monocular SLAM (VINS) [2], which is an excellent camera and IMU fusion system. Similarly, Campos et al. extended ORB-SLAM2 by fusing camera and IMU measurement and proposed ORB-SLAM3 [3]. However, consumer-grade IMU only works well in relatively low precision and suffers from bias, noise and drift while high-end IMU is prohibitively costly.

Lidar on the other hand, provides a direct spatial measurement. Lidar SLAM framework has been developed. One pioneering work is LOAM [6]. Comparing to visual SLAM, it is more robust in a dynamic environment, due to the accurate depth estimation from lidar point cloud. However, due to the traditional rotating scanning and sensing mechanism, lidar cannot provide as much vertical angular resolution or color information as camera does, so that it is prone to failure in environments with less prominent structures like tunnels or hallways. The lacking of loop-closure detection also makes the algorithm difficult to estimate global pose and map optimization. To add loop-closure detection in LOAM, Shan et al. [7] proposed an enhanced LOAM algorithm LeGO-LOAM. Comparing to LOAM, LeGO-LOAM improves feature extraction with segmentation and clustering for efficiency improvement, and adds loop-closure detection for long run drift reduction.

Combining lidar and camera in a SLAM framework become an ideal solution. While obtaining a point cloud with accurate depth information, it could make use of the high angular and color information from camera. Zhang et al. proposed VLOAM [8], which fuses monocular camera and lidar in a loosely coupled manner. Similar to LOAM, the estimation is claimed to be accurate enough and no loop-closure is needed. Shin et al. [9] also tried to combine monocular camera and lidar together using direct method rather than feature points to estimate the pose. In addition, it tightly couples the visual data and point clouds, and output the estimated pose. Shao et al. [10] went further fusing the stereo camera, IMU and lidar together. They demonstrated a good performance in outdoor large-scale scenes taking advantage of stereo Visual-Inertial Odometry (VIO) loop closure. VIO and lidar mapping are loosely coupled without further optimization at back-end. It is also limited by its complexity and cost.

Lidar was typically too costly to be useful in many applications. Fortunately, Livox unveiled a new type of lidars based on prism scanning [11]. Due to the new scanning method, the cost can be significantly lowered to enable massive adoption. Furthermore, this new scanning method allows non-repeating scanning patterns[2], ideal for acquiring a relatively high-definition point cloud when accumulated (Fig. 2). Even for 100 ms accumulation, the density of Livox Horizon is already as high as 64 lines and continue to increase. This feature can be extremely beneficial in calibrating the lidar and camera, where the traditional multiline lidar lacks the precision for space between the lines. The prism design also features a maximal optical aperture for signal reception and allows long range detection. For example, the Livox Horizon could detect up to 260 m under strong sunlight[3]. With this type of lidar, Lin et al. proposed Livox-LOAM [12], which is an enhancement of LOAM adapted to Livox Mid. Based on this, Livox also released a LOAM framework for Livox Horizon, named livox_horizon_loam[4].

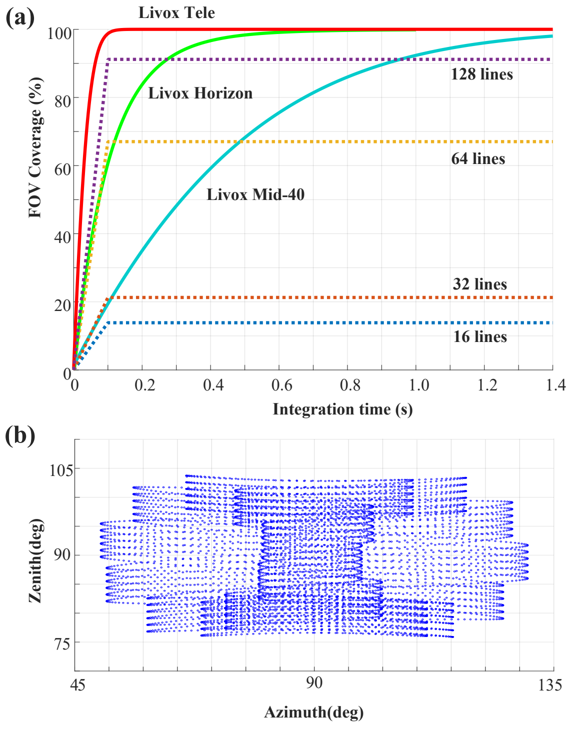

Fig. 2. (a) Point cloud density of three Livox lidar models compared to traditional lidars as a function of integration time. (Figure from Livox) (b) Livox lidar Horizon scanning pattern in 0.1 s. The pattern does not repeat itself.

Because of the long detection range, extrinsic parameter calibration between the camera and the lidar becomes even more important. In [13], the proposed solutions to the lidar-camera calibration can be classified in two categories. The

[2] https://github.com/ISEE-Technology/CamVox/blob/main/pics/horizon.gif

[3] https://www.livoxtech.com/horizon/specs

[4] https://github.com/Livox-SDK/livox_horizon_loam

first one is whether the calibration process needs a calibration target, while the second is whether the calibration can work without human intervention. During these years, many calibration techniques are based on fixed calibration target or manual effort, like [14] and [15]. In [16], Pandey et al. use Cramer-Rao-Lower-Bound (CRLB) to prove the existence of calibration parameters and estimate them by calculating the minimum variance unbiased (MVUB) estimator. Iyer et al. proposed a network called CalibNet, a geometrically supervised deep network to estimate the transformation between lidar and camera in [17]. No specific scene is required for the above two methods. In addition, Levinson et al. proposed an online calibration method in [13], in which they claimed that such a method can calibrate the lidar and camera in real time and it was suitable in any scenes. But so far, the calibration still remains as a challenge task and there is no open-source algorithm for calibrating lidar and camera in uncontrolled scenes. Livox lidar's non-repeating scanning pattern could provided a much easier solution as we will demonstrate.

III. CamVox framework

The proposed CamVox is based on ORB-SLAM2 (RGBD model) with separate RGBD input preprocessing and automatic calibration methods at uncontrolled scenes. The framework utilizes lidar-assisted visual keyframes to generate local mapping, and exhibits high robustness thanks to the back-end lightweight pose-graph optimization at various levels of bundle adjustment (BA) and loop closure from ORB-SLAM2.

In the original ORB-SLAM2, keypoints are classified in two categories, close and far, where close points are those with high certainty in depth and can be used for scale, translation and rotation estimations while the far points are only used for rotation estimation and hence less informative. With the dense, long range and accurate points obtained from Livox lidars fused with camera image, many more close points could be assigned than traditional RGBD cameras (due to detection range) or stereo vision cameras (due to limited baseline). As a result, the advantages from both the camera (high angular resolution for ORB detection and tracking) and lidar (long range and accurate depth measurement) can be exploited in a tightly coupled manner.

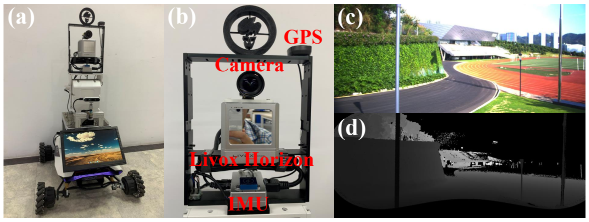

Fig. 3. (a) The complete robot platform. CamVox hardware is mounted on top of this robot. An additional RGBD camera is mounted for comparison. (b) CamVox hardware close-up including a camera, Livox Horizon, and IMU. Additional GPS/RTK is used for ground truth estimation. (c-d) an example of acquired RGB image, and depth image from lidar point cloud (colored in depth).

A. Hardware and software

The CamVox hardware includes a MV-CE060-10UC rolling shutter camera, a Livox Horizon lidar and an IMU (Inertial Sense μINS). Additional GPS-RTK (Inertial Sense μINS) is used for ground truth estimation. Hard synchronization is performed with all of these sensors by a trigger signal of 10 Hz. The camera outputs at each trigger signal (10 Hz). The lidar keeps a clock (synced with GPS-RTK) and continuously outputs the scanned point with an accurate timestamp. In the meantime, the IMU outputs at a frequency of 200 Hz synced with the trigger. Data from the GPS-RTK is also recorded for ground truth comparison. An Intel Realsense D435 RGBD camera is mounted for comparison. The whole system mounts on a moving robot platform (Agile X Scout mini). It is noted that the sale price of the Livox Horizon lidar (800 USD) is significantly lower than other similar performance lidars (10k – 80k USD) and this allows building the complete hardware system within a reasonable budget.

The software pipeline runs on several parallel threads as shown in Fig. 4. In addition to the major threads from ORB-SLAM2, an additional RGBD input preprocessing thread is added to capture data from synchronized camera and lidar (IMU corrected) and process them into a unified RGBD frame. An automatic calibration thread can be triggered to calibrate the camera and lidar, which happens automatically when the robot is detected not moving or by human interaction. The calibrated result is then evaluated and output for potential parameter update.

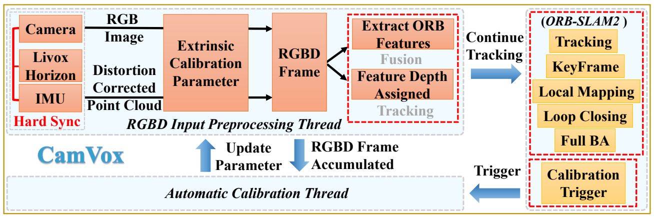

Fig. 4. CamVox SLAM pipeline. In addition to the ORB-SLAM2 major threads, a RGBD input preprocessing thread is used to convert lidar and camera data to the RGBD format, and an automatic calibration thread can be automatically/manually triggered for camera and lidar extrinsic calibration, which is shown in Section C.

B. Preprocessing

The preprocessing thread takes the raw points from the lidar, corrected by the IMU and projected into a depth image according to the extrinsic calibration with camera. The RGB image is then combined with the depth image as the output of the RGBD frame, where the two images are formatted with equal size and pixel-wise corresponded as shown in Fig. 3(c) and Fig. 3(d). Further tracking thread operations such as ORB feature extraction, keypoints generation are then performed based on the output. Since the lidar continuously scan the environment, each data point is obtained at a slightly different time and needs correction from IMU. This is different from a camera, whereas an image is obtained at almost an instant (within 10 ms frame readout time). To partially correct this lidar point distortion, the robot pose is calculated from the IMU pose at each lidar timestamp, and transforms the corresponding lidar point to the initial lidar coordinate when trigger signaled and camera image acquired at a frequency of 10 Hz.

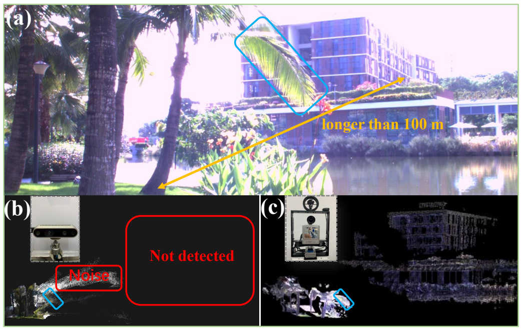

Fig. 5. Student dormitory at SUSTech, captured by (a) RGB camera, (b) Realsense D435 and (c) CamVox RGBD output. The blue rectangles show the position of the coconut tree leaf at the three pictures. It is clear that typical RGBD camera are much worse than the CamVox RGBD output in detection distance and resistance to sunlight noise.

Furthermore, with the help of long distance Livox lidar, we were able to detect reliably many depth-associated camera feature points beyond 100 meters. In comparison, the Realsense RGBD camera was not able to detect points beyond 10 meters and suffer from sunlight noises. This is clearly demonstrated in Fig. 5. As a result, in CamVox we can specify the close keypoints to those points with associated depth less than 130 meters. This is far more than the 40 times (ORB-SLAM2 default) the stereo baseline, which is on the order of 10 cm typically used in commercial stereo cameras.

C. Calibration

Calibration accuracy is vitally important in CamVox due to the long-range capability of the lidar. A small angular mismatch could result in a large absolute deviation at a large depth. Controlled calibration target such as checkerboards are not always available in the field and misalignment could happen after a random mechanical failure or a collision. An automatic calibration method needs to be developed at an uncontrolled scene and update the parameters if a better calibration match is found. Thanks to the non-repeating nature of Livox lidars, as long as we could accumulate a few seconds of scanning points, the depth image could become as high resolution as a camera image (Fig. 6) and the correspondence to the camera image becomes easy to find. Therefore, we are able to do this calibration at almost all field scenes based on the scene information automatically. The triggering of this automatic calibration is set when the robot is detected to be stationary in order to eliminate the motion blur. We accumulate lidar points for a few seconds while remaining still. Camera image is also captured.

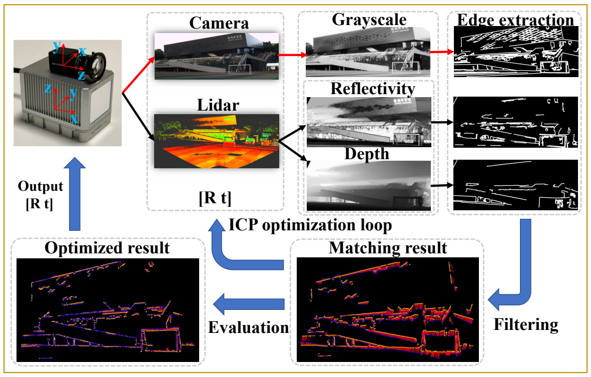

Fig. 6. Calibration thread pipeline with examples. The captured data from camera and lidar (remaining still and accumulate for a few seconds) are processed with initial calibration parameters to form three images (grayscale, reflectivity and depth, the latter two are from lidar). Calibration is performed on the edges detected from these images until a satisfactory set of parameters is obtained.

The overall calibration algorithm is structured as in Fig. 6. The dense point cloud is first projected onto the imaging plane by initial external parameters using both reflectivity and depth values, and contour extractions are then performed to compare with the camera image contour. The cost function is constructed by an improved ICP algorithm, which is optimized by Ceres [18] to get the relatively more accurate external calibration parameters. Cost function from these new parameters is then evaluated against the previous values and a decision is made whether to update the extrinsic calibration parameter at input preprocessing thread.

Suppose the coordinate value of a point in the lidar coordinate system is $X = (x, y, z, 1)^T$, the z coordinate value of a point in the camera coordinate system is Z_c and the pixel position of the point in 2D image is $Y = (u, v, 1)^T$. Given an initial external transform matrix from lidar to camera T_{lidar}^{cam} and camera's intrinsic parameter (f_u, f_v, c_u, c_v), we can project the 3D point cloud to a 2D image by Eq. (1) and Eq. (2).

$$P_{rect} = \begin{pmatrix} f_u & 0 & c_u & 0 \\ 0 & f_v & c_v & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{1}$$

$$Z_c Y = P_{rect} T_{lidar}^{cam} X \tag{2}$$

After projecting lidar point cloud to 2D image, we use histogram equalization on all images to enhance contrast and extract the edge using Canny edge detector [19]. The edges extracted from the depth image and the reflectivity image are combined because they are both from the same lidar but separate information. To extract more reliable edges, the following two kinds of edges are filtered out on those edge images. The first kind is the edge that is less than 200 pixels in length. The second kind of edge are the interior ones that are cluttered together, identified as non-contour lines whose curvatures are greater than the set threshold. Finally, some characteristic edges that are present in both camera image and lidar image are obtained and edge matching are performed according to the nearest edge distance.

An initial matching result is shown at lower right of Fig. 6, where the orange line is the edge of the camera image, the blue line is the edge of the lidar image, and the red line is the distance between the nearest points. Here we adopted the ICP algorithm [20] and use K-D tree to speed up the search for the nearest neighbor point. However, sometimes in a wrong match very few points actually participated in the calculation of distance, the value of the cost function is trapped inside this local minimum. In this case, we improve the cost function in ICP by adding a degree of mismatching. The improved cost function is shown in Eq. (3), where n is number of camera edge points that are within distance threshold with lidar edge points, m is the number of nearest neighbors, N is number of all camera points, b is weighing factor. We found a value of 10 for b is a good candidate to start as the default value. Note here that the cost function is an averaged value for matching points and thus can be used to compare horizontally at different scenes.

$$CF = \sum_{i=1}^{n}\sum_{k=1}^{m} \frac{\text{Distance}(P_i^{cam}, P_{ik}^{lidar})}{n \times m} + b \times \frac{N-n}{N} \tag{3}$$

In optimizing this cost function, we adopted coordinate descent algorithms [21] and iteratively optimized (roll, pitch, yaw) coordinates by Ceres [18]. This seems to result in a better convergence.

IV. Results

In this section we present the evaluation results of CamVox. Specifically, we will first show the results of the automatic calibration. The effect of choosing depth threshold for close keypoints is also evaluated. Finally we evaluate the trajectory of CamVox as compared to some of the main stream SLAM frameworks and give a time analysis.

A. Automatic calibration results

Our automatic calibration result is shown in Fig. 7. Shown in Fig. 7(a) is an overlay of lidar points onto the RGB image when the sensor and the camera are not calibrated (with a misalignment of more than 2 degrees). The cost function has a value of 7.95. The automatic calibration is triggered and calibrates the result as shown in Fig. 7(b), where the cost function value is 6.11, while the best possible calibration result done manually is shown in Fig. 7(c) with value 5.88 (as the ground truth). The automatic calibration delivers a very close result to the best manual calibration. Additionally, thanks to the image-like calibration scheme, the automatic calibration works robustly on most uncontrolled scenes. Several different scenes are evaluated in Fig. 7(d-f) and cost values in the optimization equation gradually converges to a low value shown in 7(g-i). We found a cost value of 6 in our implementation is a relatively good number. Notice that this algorithm does not require any assumptions, such as straight lines and flat surfaces, which are sometimes imposed in other automatic calibration algorithms.

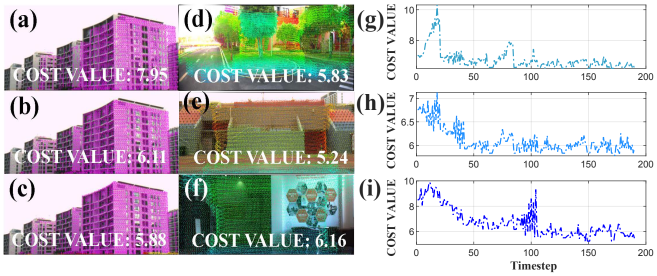

Fig. 7. An example of RGB camera and point cloud overlay after calibration. (a) not calibrated. (b) automatically calibrated. (c) the best manual calibration. The automatic calibration algorithms is verified at various scenes, (d) outdoor road with natural trees and grasses, (e) outdoor artificial structures, (f) indoor underexposed structures. (g-i) represent the cost value evolution in the optimization process corresponding to the scenes on the left.

B. Evaluation of depth threshold for keypoints

Because the lidar could detect 260 meters, there are many keypoints in the fused frame that we could characterize as close. These points help significantly in tracking and mapping. From Fig. 8(a-d), by setting the close keypoints depth threshold from 20m to 130m, we see a significant increase in both mapping scale and number of mapped features. In Fig. 8(e) We evaluated the number of matching points tracked as a function of time in the first 100 frames (10 FPS) after starting CamVox. An increase of feature numbers is observed as more frames is captured (Fig. 8(f) 5 frames after start), and the larger threshold obviously tracked more features initially that is helpful in starting a robust localization and mapping. In CamVox, we recommend and set the default value to 130 meters.

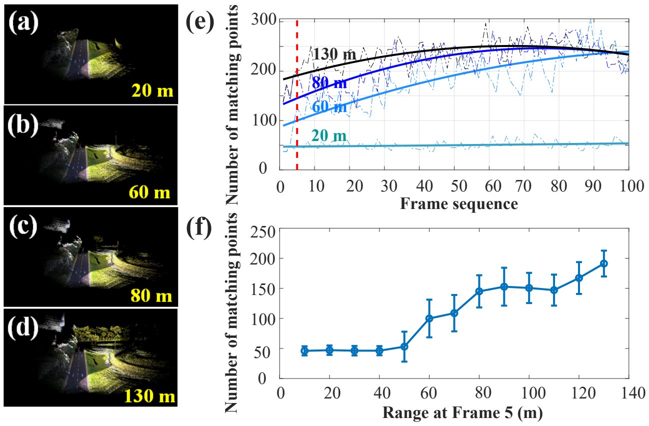

Fig. 8. Evaluation of close keypoints threshold. (a-d) The reconstructed point cloud map by selecting different values of close keypoints threshold. (e) the number of matching points as a function of frame sequence (10 FPS) from start. (f) the number of matching points as a function of close keypoints threshold, evaluated at the beginning (5th frame) of SLAM process.

C. Comparison of trajectory

The comparisons of the trajectories from CamVox, two mainstream SLAM framework and the ground truth are evaluated on our SUSTech dataset shown in Fig. 9 and TABLE I using evo tools [22] . Due to the accurate calibration, rich depth-associated visual features and their accurate tracking, CamVox system is very close to ground truth and significantly outperformed the other frameworks such as livox_horizon_loam [5] and VINS-mono [2].

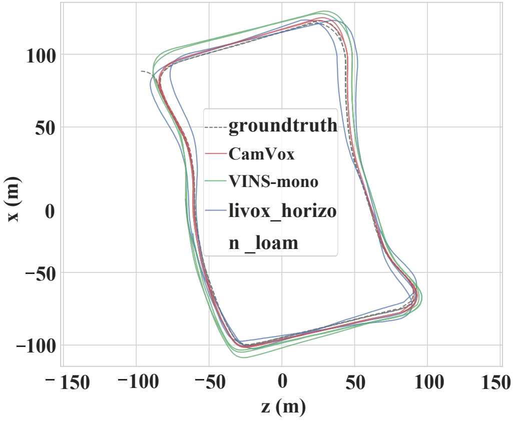

Fig. 9. Trajectories from livox_horizon_loam, VINS-mono and CamVox together with ground truth from SUSTech dataset.

D. Timing results

The timing analysis of the CamVox framework was performed as illustrated in TABLE II. The evaluation was performed on a onboard computer system Manifold 2C which has a 4-core Intel Core i7-8550U processor. With such a system, CamVox is able to perform in real-time. Although the automatic calibration takes about 58s to finish, this thread only runs occasionally while needed and the robot is in a stationary state. Furthermore, the update of parameter could happen at a later time and such calculation time will not be an issue for real time performance.

TABLE I
ABSOLUTE POSE ERROR (APE) (UNIT: M)

APE	CamVox	VINS-mono	livox_horizon_loam
max	**3.3**	27.2	9.9
mean	**1.7**	6.7	6.2
median	**1.6**	6.1	6.5
min	**0.2**	2.8	1.8
rmse	**1.8**	7.5	6.5
sse	**16066.5**	50788.6	101223.7
std	**0.7**	3.5	1.9

V. CONCLUSION AND PERSPECTIVE

To summarize, we have proposed CamVox as a new low-cost lidar-assisted visual SLAM framework, aiming to combine the best from both worlds, i.e., the best angular resolution from camera and the best depth and range from lidar. Thanks to the unique working principle of Livox lidar, an automatic calibration algorithm that could perform in uncontrolled scenes is developed. Evaluations of this new framework was carried out in automatic calibration accuracy, depth threshold for close keypoints classification and trajectory comparison. It could also run with real time performance on an onboard computer. We hope that this new framework could be useful for robotic and sensor research and an out-of-the-box low-cost solution for the community.

[5] https://github.com/Livox-SDK/livox_horizon_loam

TABLE II
TIME ANALYSIS

	Framework	CamVox	ORB-SLAM2
Setting	Dataset	SUSTech	TUM
	Resolution	1520×568	640×480
	Camera FPS	10 Hz	30 Hz
	ORB Features	1500	1000
Thread	Calibration	58.16 s	/
	Tracking	42.27 ms	25.58 ms
	Mapping	252.41 ms	267.33 ms
	Loop Closing	7821.22 ms	598.70 ms
RGBD	IMU Corretcion	0.89 ms	/
Preprocessing	Pcd2Depth	16.35 ms	/

ACKNOWLEDGMENT

The authors thank Fu Zhang at HKU and colleagues at Livox Technology for helpful discussion and support. The authors also thank anonymous reviewers for their valuable comments and suggestions.

REFERENCES

[1] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[2] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[3] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual-inertial and multi-map slam," *arXiv preprint arXiv:2007.11898*, 2020.

[4] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[5] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[6] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, no. 9, 2014.

[7] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4758–4765.

[8] J. Zhang and S. Singh, "Visual-lidar odometry and mapping: Low-drift, robust, and fast," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2174–2181.

[9] Y.-S. Shin, Y. S. Park, and A. Kim, "Direct visual slam using sparse depth for camera-lidar system," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[10] W. Shao, S. Vijayarangan, C. Li, and G. Kantor, "Stereo visual inertial lidar simultaneous localization and mapping," *arXiv preprint arXiv:1902.10741*, 2019.

[11] Z. Liu, F. Zhang, and X. Hong, "Low-cost retina-like robotic lidars based on incommensurable scanning," *arXiv preprint arXiv:2006.11034*, 2020.

[12] J. Lin and F. Zhang, “Loam livox: A fast, robust, high-precision lidar odometry and mapping package for lidars of small fov,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3126–3131.

[13] J. Levinson and S. Thrun, “Automatic online calibration of cameras and lasers.” in *Robotics: Science and Systems*, vol. 2, 2013, p. 7.

[14] R. Unnikrishnan and M. Hebert, “Fast extrinsic calibration of a laser rangefinder to a camera,” *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-05-09*, 2005.

[15] A. Geiger, F. Moosmann, Ö. Car, and B. Schuster, “Automatic camera and range sensor calibration using a single shot,” in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 3936–3943.

[16] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, “Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information.” in *AAAI*. Citeseer, 2012.

[17] G. Iyer, R. K. Ram, J. K. Murthy, and K. M. Krishna, “Calibnet: Geometrically supervised extrinsic calibration using 3d spatial transformer networks,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1110–1117.

[18] S. Agarwal, K. Mierle, and Others, “Ceres solver,” http://ceres-solver.org.

[19] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[20] P. J. Besl and N. D. McKay, “Method for registration of 3-d shapes,” in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–606.

[21] S. J. Wright, “Coordinate descent algorithms,” *Mathematical Programming*, vol. 151, no. 1, pp. 3–34, 2015.

[22] M. Grupp, “evo: Python package for the evaluation of odometry and slam.” https://github.com/MichaelGrupp/evo, 2017.