2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

Embedding Temporally Consistent Depth Recovery for Real-time Dense Mapping in Visual-inertial Odometry

Hui Cheng[1], Zhuoqi Zheng[1], Jinhao He[1], Chongyu Chen[1], Keze Wang[1], and Liang Lin[1]

Abstract— **Dense mapping is always the desire of simultaneous localization and mapping (SLAM), especially for the applications that require fast and dense scene information. Visual-inertial odometry (VIO) is a light-weight and effective solution to fast self-localization. However, VIO-based SLAM systems have difficulty in providing dense mapping results due to the spatial sparsity and temporal instability of the VIO depth estimations. Although there have been great efforts on real-time mapping and depth recovery from sparse measurements, the existing solutions for VIO-based SLAM still fail to preserve sufficient geometry details in their results. In this paper, we propose to embed depth recovery into VIO-based SLAM for real-time dense mapping. In the proposed method, we present a subspace-based stabilization scheme to maintain the temporal consistency and design a hierarchical pipeline for edge-preserving depth interpolation to reduce the computational burden. Numerous experiments demonstrate that our method can achieve an accuracy improvement of up to 49.1 cm compared to state-of-the-art learning-based methods for depth recovery and reconstruct sufficient geometric details in dense mapping when only 0.07% depth samples are available. Since a simple CPU implementation of our method already runs at 10-20 fps, we believe our method is very favorable for practical SLAM systems with critical computational requirements.**

I. Introduction

For unmanned intelligent systems such as autonomous vehicles, simultaneous localization and mapping (SLAM) is an important tool for perceiving the physical world. To obtain the physical information of the environment, SLAM systems have to find ways to obtain its real distances to the surroundings, i.e., the real depth values. In principle, the real depth can be either directly obtained by range sensors [1] or estimated by calibrated cameras. Among these methods, the monocular visual-inertial system [2], [3] plays an important role in SLAM systems because of its least amount of required data and immediate application to mobile devices [4]. Although visual-inertial odometry (VIO) has well addressed the problem of self-localization in real-time, its mapping result is still insufficient for practical use due to the spatial sparsity and temporal instability of the estimated depth values. The main reason is that the depth values are estimated by matching visual feature points (*i.e.* landmark points) among sequential frames, which is with inevitable mismatches due to its strong dependency on the scene appearance. Therefore, dense 3D mapping of the scene using a monocular visual-inertial system remains a challenging problem, especially when there are critical computational and storage requirements.

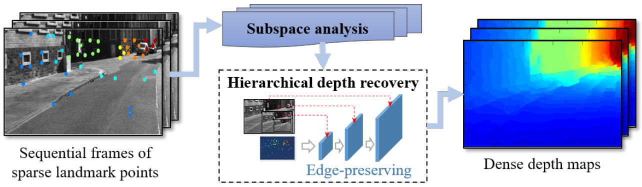

Subspace analysis
Hierarchical depth recovery
Edge-preserving
Sequential frames of sparse landmark points
Dense depth maps

Fig. 1. An intuitive illustration of the proposed method.

Since a sequence of sufficiently dense depth maps can be fused for reconstructing a large-scale 3D scene [5], the key to VIO-based dense 3D mapping is the recovery of dense depth maps from sparse depth values. In the literature, there have been extensive works on recovering dense depth maps, which are designed for tackling the input depth values with different sparsity levels. Considering that the depth maps obtained by range sensors are with noise and holes [6], plenty of filtering and inpainting methods [7]–[9] are proposed to reduce the noise and fill in the holes. These methods can produce high-quality depth maps by embedding edge-preserving filters into their formulations. Although these methods can only handle a small portion of depth missing in real-time, their idea of preserving the depth boundaries is still critical to high-quality depth recovery. For stereo matching or VIO, the number of available depth values becomes much smaller [2], [3] compared to the resolution of the depth map, which raises the problem of reconstructing a dense depth map from very sparse depth values and the corresponding high-resolution intensity image. Existing solutions to this problem [10]–[14] are with limitations that hinder their application to VIO-based SLAM systems. For example, the representation-based depth reconstruction requires large amount of computations [10], [12] and even carefully designed sampling strategies [10]. Recent learning-based methods [13], [14] have demonstrated their capability in recovering dense depth maps after sufficient training. However, these methods fail to preserve depth boundaries and require large amount of memory for storing their parameters. In addition, the important temporal consistency of sequential depth maps is not taken into account in these methods.

In this paper, we embed depth recovery into the VIO-based SLAM system with considerations of the temporal consistency of sequential frames, the preservation of boundaries, and high computational efficiency. As shown in Fig. 1,

*This work is partially supported by the National Natural Science Foundation of China (NSFC) under Grant 61602533, NSFC-Shenzhen Robotics Projects (U1613211), Science and Technology Program of Guangzhou, China (201510010126), and The Fundamental Research Funds for the Central Universities.

[1] The authors are with the School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, Guangdong, 510006 China. Correspondence should be addressed to Chongyu Chen: chency47@mail.sysu.edu.cn.

978-1-5386-8094-0/18/$31.00 ©2018 IEEE

in the proposed method, a temporal stabilization scheme based on subspace analysis is designed to guarantee the temporal consistency of landmark points, the edge-preserving filters are employed to achieve high recovery accuracy, and a hierarchical processing pipeline is used to reduce the computational burden. The contributions of our work are twofold. First, we propose an effective solution to the challenging problem of depth recovery from extremely sparse depth samples, which achieves superior recovery quality compared to the existing solutions in the context of VIO. Second, we realize real-time dense mapping by embedding our method into VIO-based SLAM systems, which demonstrates a lightweight and effective solution to real-time SLAM.

II. Proposed Method

The embedding of the proposed method into the VIO-based SLAM is illustrated in Fig. 2, where the red rectangles indicate the key components of our method. In this section, we describe these key components in detail.

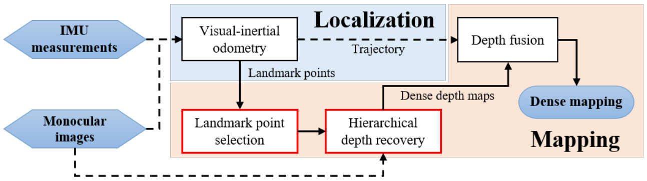


Fig. 2. An illustration of embedding our method into the VIO-based SLAM.

A. *Stabilizing Landmark Points for Temporally Consistency*

The primary step of VIO is the extraction of visual features from the intensity images. Although there have been several types of visual features (such as the robust ORB [15] feature and efficient KLT [16] feature) which can be extracted in real-time and matched with high accuracy, the extracted feature points are still temporally unstable. The major reason is the rapid change of the scene appearance, which may be caused by lighting changes, occlusion, and fast motion. Even when the feature point is physically stable (*e.g.*, a fixed point on the ground), its integer coordinates in the camera coordinate system still reduces its location accuracy, especially when the distance between the feature point and the camera is large.

After the matching of feature points between sequential frames, a set of landmark points that indicate the temporal correspondences can be obtained. Considering that stable landmark points indicate the same physical place, we propose to formulate the feature extraction as a sensing process of the physical coordinates of landmark points. In this way, both spatial inaccuracy and temporal instability are regarded as additive noises, and the real coordinates can be estimated by reducing such noises. It should be noted that such noises are difficult to be explicitly modelled due to the diversity of feature types and scene appearances, which makes the point stabilization very challenging.

In this work, we propose to utilize subspace analysis for stabilizing the landmark points. To be more specific, there are three steps. First, sparse landmark points from feature-based VIO is represented in the world coordinates. Then, by stacking the coordinates of all landmark points in the same frame as a column $\mathbf{a}_i$, *i.e.*, $\mathbf{a}_i = [x_1^{(i)}, y_1^{(i)}, z_1^{(i)}, \ldots, x_p^{(i)}, y_p^{(i)}, z_p^{(i)}]^\top$, where $i \in [1, \ldots, n]$ represents the column number, we construct a matrix $A = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n,] \in \mathbb{R}^{3p \times n}$ for p landmark points that appear in sequential n frames. At last, we perform the low-rank and sparse decomposition (LRSD) [17] on A, *i.e.*,

$$A = L + S, \tag{1}$$

where L and S are the low-rank and sparse components of A. Based on the fact that the world coordinates of a stable landmark point should be identical during the sensing process, we impose a rank-1 constraint on L in the decomposition process. By averaging every row of the matrix L and reorganizing the column vector into 3D coordinates, we stabilize the 3D coordinates of each landmark point.

The proposed stabilization process is expected to be effective. The reasons are twofold. First, the subspace spanned by the column vector of coordinates is expected to be low-rank because of the physical identity of the real landmark points. Therefore, the low-rank structure of L can well capture the real coordinates. Second, the sparse structure of S can well capture the temporal disturbances because the rapid scene changes are expected to be temporally sparse. Generally, there are usually very few sudden changes in videos. The final step of averaging the rows of L further reduces the coordinate inconsistency. The effectiveness of the proposed subspace-based stabilization scheme will be demonstrated in Section III-A.

B. *Hierarchical Edge-preserving Depth Recovery*

With the temporally stable but spatially sparse landmark points and their depth values, we utilize edge-preserving filters to recover the dense depth map, which includes the estimation of missing depth values and the smoothing of existing depth values. Standard edge-preserving filters [18]–[20] can be written in a similar form, *i.e.*,

$$y_i = \sum_{j \in \Omega_i} w_{ij}(I) x_j, \tag{2}$$

where x_j is the j-th pixel of the input depth map $\mathbf{x}$, y_i is the i-th pixel of the output depth map $\mathbf{y}$, Ω_i represents the pixels inside the filtering window of pixel i, and I is the guidance image. In general, the computation of the weights is of $O(HWh^2)$ complexity for recovering a $H \times W$ depth map using a filtering window of $h \times h$. Iterative filtering is usually required for higher recovery quality, which makes the edge-preserving depth recovery very time-consuming.

Inspired by the recent advances of deep convolutional network [21] and iterative depth recovery [9], we propose a hierarchical structure for edge-preserving depth interpolation and refinement. Three types of edge-preserving filters are utilized, *i.e.*, the joint bilateral filter (JBF) [18], joint trilateral filter (JTF) [19], and the guided filter (GF) [20]. In particular, the JBF weight, *i.e.*.

$$w_{ij}^{\text{JBF}} = \exp\left(-\frac{2\|i-j\|^2}{h^2}\right) \exp\left(-\frac{(I(i)-I(j))^2}{2\sigma_c^2}\right), \tag{3}$$

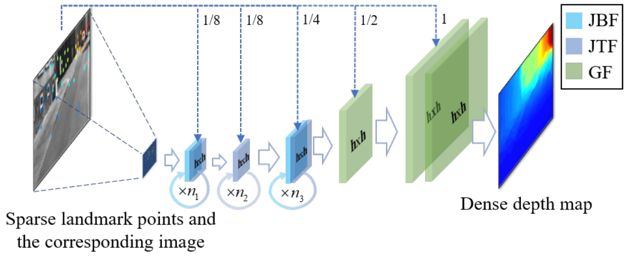


Fig. 3. The proposed processing pipeline for depth recovery.

is used for depth interpolation, and the JTF weight, *i.e.*

$$w_{ij}^{\text{JTF}} = w_{ij}^{\text{JBF}} \exp(-\frac{(x_i - x_j)^2}{2\sigma_z^2}), \tag{4}$$

is used for depth refinement at the smallest two scales. For largest two scales, the GF is used for fast depth refinement due to its high efficiency. The proposed processing pipeline is shown in Fig. 3, where the processing blocks for JBF, JTF, and GF are marked by different colors and the captions on the block indicate the size of the filtering window. All the processing blocks use the downsampled intensity images of the same resolution as the depth map as the guidance, except that in the final refinement by GF, only the depth map is used as the guidance. The pipeline can be roughly divided into 3 stages, which are illustrated from left to right in Fig. 3.

The first stage includes the downsampling of the input depth map into $H/8 \times W/8$ and the processing blocks at this scale. It can be seen that there are two iterative processing blocks at this scale. In the first iterative block, both JBF and JTF are used. The JBF is used for depth interpolation to estimate the missing depth values according to its surrounding available pixels, after which the JTF is used to refine the estimated depth values. Such "JBF+JTF" process is performed for n_1 iterations until most pixels are with depth values. Then, the JTF is applied for n_2 times for further depth refinement.

In the second stage, we individually deal with edge pixels and other pixels of the depth map because the depth edge is very sharp while the depth surface is very smooth. A difference-based scheme is proposed to quickly find out the pixels around depth boundaries. That is, the low-resolution depth map is enlarged to the resolution of $H/4 \times W/4$ using the linear and nearest interpolations, respectively, resulting in two enlarged depth maps D_1 and D_2. Then, the pixel-wise depth differences between D_1 and D_2 are compared with a threshold. So that the pixels with sufficiently large depth difference can be found, which are marked as edge pixels. Using the enlarged depth map D_1 as input, we use JBF to re-estimate the depth values for the edge pixels and JTF for the refinement of other pixels. This "JBF+JTF" process is performed for n_3 iterations to produce an edge-preserving depth map at this scale.

The third stage includes the processing at two scales, *i.e.*, $H/2 \times W/2$ and $H \times W$. For the depth refinement, we use GF instead of JTF for a better trade-off between accuracy and efficiency. Validations for the choice of edge-preserving filters at different scales will be discussed in Section III-A.

In general, larger parameters of n_1, n_2, and n_3 lead to higher recovery accuracy, while higher efficiency can be obtained when they are smaller. One can set these parameters subject to the processing capability of the used hardware. In our implementation on a common PC with a CPU of 3.4 GHz, we set $\sigma_c = 8$, $\sigma_z = 23$, $h = 7$, $n_1 = 15$, $n_2 = 10$, and $n_3 = 5$ for a processing framerate of 10 fps. Such parameter setting not only guarantees the convergence of iterative filtering [9], but also inherits the effective characteristic of the recursive neural network (RNN) [22]. We perform more iterations in the smallest scale because this scale is extremely critical to the overall recovery quality. Note that the parameters for the small scales do not essentially affect the efficiency. Therefore, even when the iteration numbers are not small, the CPU implementation of our method can still run at real-time.

III. Experimental Results

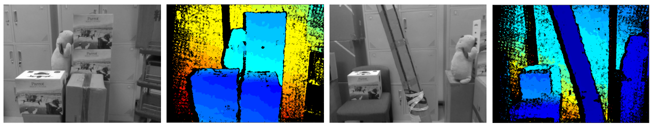
In this section, we thoroughly assess the proposed method by experiments. First, we individually evaluate the proposed landmark point stabilization scheme and the hierarchical depth recovery architecture. The relationship between the recovery accuracy and the parameter setup is demonstrated to support the rationality of our implementation. Second, we conduct comparisons between the proposed method and recent solutions to depth recovery and dense mapping.

Three popular quantitative metrics (as described in [13] and [14]) are used in this section, *i.e.*, the root-mean-square error (RMSE), mean absolute relative error (REL), and the inlier ratio (*i.e.*, the percentage of predicted pixels where the relative error is within a threshold). Both the RMSE and REL are calculated using real depth values (in meter), which are the lower the better. The inlier ratio is better when it is higher.

A. Ablation Study

Our method requires a set of feature points and their depth values as the input. In our experiments, these points are obtained by the localization part of the state-of-the-art VIO-based SLAM system, *i.e.*, the monocular visual-inertial system (VINS) [3]. In particular, potential feature points are detected according to the "good feature to track" principle [23], and the KLT sparse optical flow algorithm [16] is employed for the tracking of feature points. In general, 100-300 feature points are tracked for a good localization in VINS. Thus, usually less than 300 feature points with the estimated depth values are used as the input of our method, which are extremely sparse compared to the image resolution and make the dense mapping very challenging.

For the ablation study of the proposed method, we capture 5 RGB-D sequences of resolution 640 × 480 with synchronized inertial measurements using the ZR300 visual-inertial sensor. These sequences are captured at the framerate of 30 fps in common indoor environments where the lighting is stable. Each sequence is of length abound 60 seconds. Example frames of the captured sequences are shown in



(a) Image #1 (b) Depth #1 (c) Image #2 (d) Depth #2

Fig. 4. Examples of our captured images and depth maps.

Fig. 4. The depth maps captured by ZR300 are used as the ground-truth depth maps.

1) Stabilization of landmark points: Herein, the proposed method for stabilizing the landmark points takes 10 frames of feature points as the input. For comparison, we also use the simple average to stabilize the landmark points. Note that only the feature points appear in no less than 2 key frames are selected for stabilization.

Because the accuracy of the coordinates of landmark points is in proportion to the accuracy of depth recovery, we directly compare the depth accuracy to assess the stabilization methods. The quantitative scores of REL and RMSE on the captured 5 sequences are shown in Table I. These results clearly demonstrate the effectiveness of the proposed subspace-based stabilization method because the proposed method leads to the highest recovery accuracy for most sequences. For the sequence where the simple average works slightly better (*e.g.*, sequence No. 4 where there are less feature perturbations), the proposed method still achieves a comparable recovery accuracy (the score differences are no more than 0.005).

TABLE I

ASSESSMENT ON THE LANDMARK POINT STABILIZATION METHODS

Method	No stabilization		Simple average		Subspace-based	
Sequence No.	REL	RMSE	REL	RMSE	REL	RMSE
1	0.187	0.312	0.194	0.325	**0.150**	**0.270**
2	0.271	0.456	0.269	0.452	**0.230**	**0.402**
3	0.138	0.238	**0.136**	**0.234**	**0.136**	**0.234**
4	0.119	0.217	**0.116**	**0.214**	0.120	0.221
5	0.108	0.214	0.107	0.215	**0.092**	**0.192**
Mean	0.165	0.287	0.164	0.288	**0.146**	**0.264**

2) The choice of edge-preserving filters: As described in Section II-B, the proposed hierarchical processing pipeline for depth recovery can be divided into 3 stages. We would like to point out that the choices of edge-preserving filters in the second and the third stages are relatively flexible. In general, JTF and GF are chosen for higher recovery quality and higher efficiency, respectively.

The supports of our statement are shown in Table II, where each row presents a profile for choosing the filters. The scores are obtained by conducting experiments on sequence No. 5. The bold fonts indicate the best results. It can be observed that both REL and RMSE remain similar across the four profiles. Therefore, we can choose the combination of filters subject to the computational capability of the utilized hardware. In the following, we use the profile presented in the third row for the comparisons with other methods.

TABLE II

EVALUATION OF DIFFERENT PROFILES OF FILTERS

	Scale		REL	RMSE	Time	Framerate
1/4	1/2	1				
JTF	JTF	JTF	0.093	0.195	256 ms	4 fps
JTF	JTF	GF	0.093	0.194	158 ms	6 fps
JTF	GF	GF	0.092	0.192	103 ms	10 fps
GF	GF	GF	0.093	0.194	66 ms	15 fps

B. Quantitative Evaluations of Depth Completion

The problem of depth completion is very important to autonomous vehicles because sparse depth measurements are usually provided by practical sensing systems. Researchers propose the KITTI depth completion dataset [24] to evaluate the methods for completing LiDAR depth maps that have thousands of depth measurements in one frame. However, our method is designed for the depth map with only hundreds of depth values. To conduct experiments on this dataset, we generate very sparse depth maps by preserving the depth values on the selected points. Two schemes of point selection are tested with our proposed method. The first scheme is the uniformly random selection of all pixels in the depth map, which is denoted as “Ours-random”. Each score for the random selection The second scheme selects the points with “good features” [23] according to the content of the input intensity image, which is denoted as “Ours-feature”.

In our experiments, 1000 depth maps with ground truths are used to calculate the quantitative scores, which are provided in the “manually selected validation and test data sets” of the KITTI depth completion dataset. Three methods for recovering a dense depth map from very sparse depth measurements are compared, *i.e.*, the method based on residual neural network [13] (denoted as “Res-NN”), the auto-encoder-based method [11] (denoted as “Auto-encoder”), and the method based on deep regression network [14] (denoted as “DRN”). The quantitative scores are presented in Table III, where there are three inlier ratios (*i.e.*, δ_1, δ_2, and δ_3) corresponding to three different thresholds (see [14] for more details). The bold fonts indicate the best results, and the results of different methods are listed according to their ranking. It is demonstrated in these results that the proposed method using the VIO feature points consistently outperforms other methods when there are only 200 or 500 available feature points. Significant performance improvement upon both the Res-NN and auto-encoder-based method are demonstrated, even though these two methods use more available depth samples. The performance gains of our method compared to the second-best method DRN are 20.6 cm in RMSE and up to 1.8% in the inlier ratios when only 200 feature points are available. The accuracy improvement increases to 49.1 cm in RMSE when 500 feature points are available. The proposed method performs slightly worse when random feature points are used, which is expected because random points may not preserve sufficient structural information.

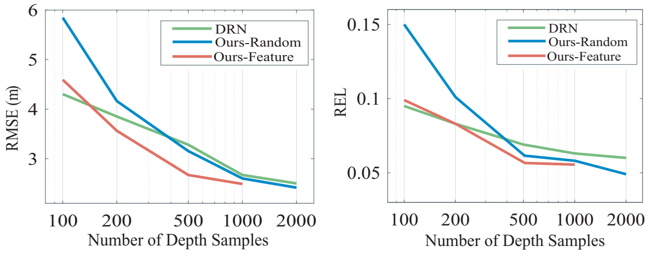
For further comparisons between our method and DRN, we change the number of available depth samples from 100 to 2000 and obtain the RMSE and REL curves shown in Fig. 5. One can see that the proposed method with VIO landmark points as the input outperforms DRN when the sample number is greater than 200. Considering that DRN

TABLE III

COMPARISONS ON THE DEPTH COMPLETION DATASET

Method	Samples	RMSE	REL	$\delta_1(\%)$	$\delta_2(\%)$	$\delta_3(\%)$
Res-NN [13]	225	4.500	0.113	87.4	96.0	98.4
Ours-random	200	4.125	0.088	91.2	97.6	98.3
DRN [14]	200	3.851	0.083	91.9	97.0	98.6
Ours-feature	200	**3.645**	**0.081**	**93.7**	**97.8**	**98.9**
Auto-encoder [11]	650	7.140	0.179	70.9	88.8	95.6
DRN [14]	500	3.378	0.073	93.5	97.6	98.9
Ours-random	500	3.142	0.064	94.8	98.3	99.1
Ours-feature	500	**2.887**	**0.062**	**95.6**	**98.7**	**99.2**

requires training and large amount of storage, our method is believed to be more favorable in practice due to its training-free characteristic and real-time processing capability.

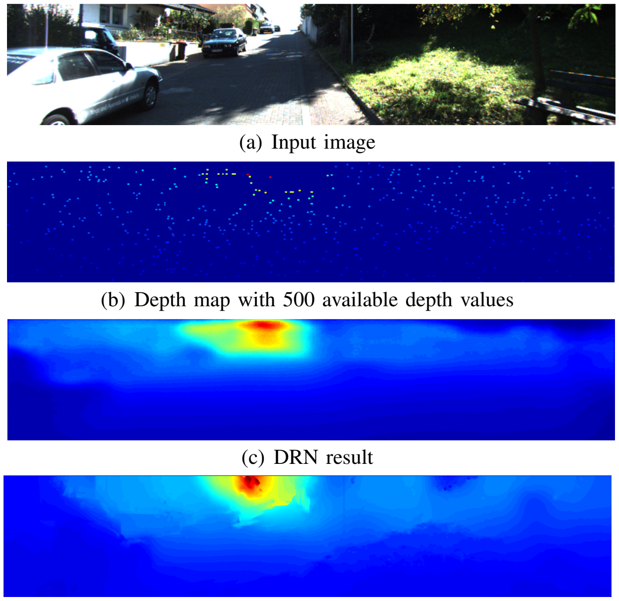


(a) The RMSE curve

(b) The REL curve

Fig. 5. The RMSE and REL curves obtained by our method and DRN.

Beside the quantitative scores, we also demonstrate the visual quality of the depth maps recovered by our method in Fig. 6. Note that there are only 500 (0.17% of the image resolution of 1216 × 240) available depth samples. It can be seen that there are more recognizable geometric details (such as the cars and the telegraph pole) in the depth map obtained by our method.



(a) Input image

(b) Depth map with 500 available depth values

(c) DRN result

(d) Ours result

Fig. 6. A visual comparison between the results of DRN and our method.

C. Embedding Depth Recovery into the VIO-based SLAM

After the thorough assessment of the proposed method in the key task of depth recovery, we turn to the validation of our method in a practical VIO-based SLAM system. Our experiments are conducted by embedding our method into a recently developed visual-inertial SLAM system (VI-MEAN) [25]. The aforementioned dataset captured by ZR300 is used as the test data because synchronized inertial measurements are available.

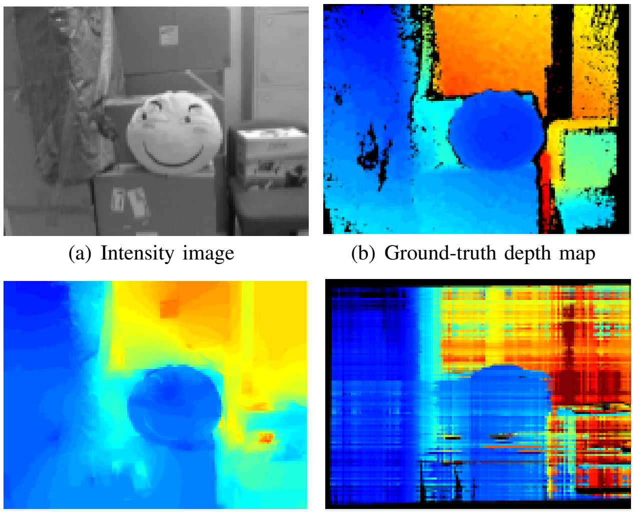
1) Quantitative comparisons: First, we compare the quality of depth maps provided by the proposed method and the motion stereo used in VI-MEAN. Quantitative results are presented in Table IV. Less than 200 feature points in each frame, equivalently no more than 0.07% of the pixel number 640 × 480, are used as the input of our method. One can see that both RMSE and REL scores of our method are only one third of that of motion stereo, which demonstrates the overwhelming superiority of our method. Because the processing framerates of our method and VI-MEAN are similar, our method is believed to be more suitable for dense mapping in VIO-based SLAM systems.

TABLE IV

EVALUATIONS ON THE CAPTURED DATASETS

Method	VI-MEAN		Ours	
	REL	RMSE (m)	REL	RMSE (m)
1	0.349	0.660	**0.150**	**0.270**
2	0.340	0.633	**0.230**	**0.402**
3	0.352	0.677	**0.136**	**0.234**
4	0.340	0.753	**0.120**	**0.221**
5	0.296	0.611	**0.092**	**0.192**
Mean	0.335	0.668	**0.146**	**0.264**

2) Visual results: Visual comparisons of the recovered depth maps are demonstrated in Fig. 7. As can be expected, the depth maps provided by our method are closer to the ground-truth depth maps, while the depth maps obtained by motion stereo are with severe artifacts. Such artifacts result in a blurry dense mapping with insufficient geometric details. As shown in Fig. 8 (a), the boundaries of some large objects (*e.g.*, the boxes) are blurred in the 3D scene reconstructed



(a) Intensity image

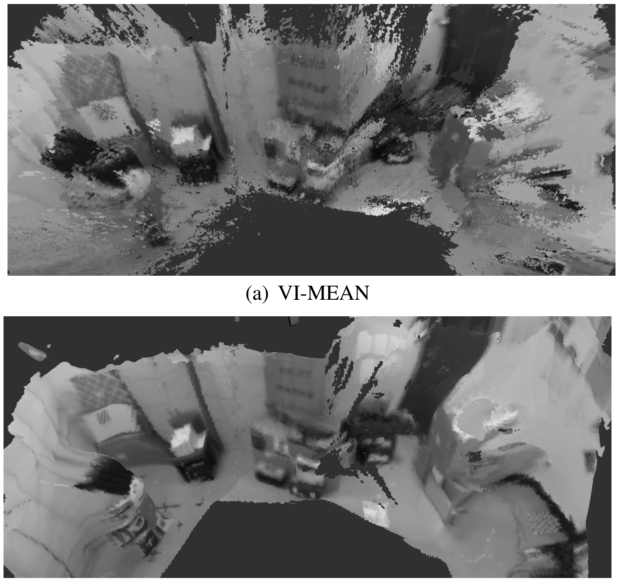
(b) Ground-truth depth map

(c) Our method

(d) Motion stereo

Fig. 7. A visual comparison of the results on the captured dataset.

by motion stereo. In contrast, using the dense depth maps provided by the proposed method, we reconstruct a 3D scene with more geometric details shown in Fig. 8 (b). It can be seen that the reconstructed 3D scene has both sharper boundaries around the object and more realistic textures on the object surfaces. Many geometric details that cannot be seen in the results of VI-MEAN can be found in the results of our method. The visual comparisons from more camera angles are shown in the video attachment of this paper.



(a) VI-MEAN

(b) Embedding our method to VI-MEAN

Fig. 8. The comparisons between dense mapping results.

IV. Conclusion

In this paper, we present a light-weight and effective solution to the problem of recovering a dense depth map from extremely sparse depth samples, which can work even when there are less than 0.07% available depth samples. The proposed solution contains a hierarchical processing pipeline with recursive edge-preserving filters, whose structure is quite similar to that of the deep neural networks. The high recovery accuracy and efficiency bring inspirations to the design of future methods for depth recovery and image restoration. The proposed landmark point stabilization method successfully imposes temporal consistency to the recovered depth maps, which makes our method more suitable for dense mapping in SLAM systems. Although a simple CPU implementation of our method can already run at real-time, it is still worth pointing out that our method can be further accelerated by parallel implementation.

References

[1] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems Conference*, 2014.

[2] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[3] Y. Lin, F. Gao, T. Qin, W. Gao, T. Liu, W. Wu, Z. Yang, and S. Shen, "Autonomous aerial navigation using monocular visual-inertial fusion," *Journal of Field Robotics*, vol. 35, no. 1, pp. 23–51, 2018.

[4] T. Schneider, M. T. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart, "Maplab: An open framework for research in visual-inertial mapping and localization," *IEEE Robotics and Automation Letters*, 2018.

[5] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. Mcdonald, "Real-time large-scale dense RGB-D SLAM with volumetric fusion," *International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 598–626, 2015.

[6] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012.

[7] C. Richardt, C. Stoll, N. A. Dodgson, H.-P. Seidel, and C. Theobalt, "Coherent spatiotemporal filtering, upsampling and rendering of RGBZ videos," *Comp. Graph. Forum*, vol. 31, no. 2pt1, pp. 247–256, May 2012.

[8] J. Yang, X. Ye, K. Li, C. Hou, and Y. Wang, "Color-guided depth recovery from RGB-D data using an adaptive autoregressive model." *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3443–3458, 2014.

[9] C. Chen, J. Cai, J. Zheng, T. J. Cham, and G. Shi, "Kinect depth recovery using a color-guided, region-adaptive, and depth-selective framework," *ACM Transactions on Intelligent Systems and Technology*, vol. 6, no. 2, pp. 1–19, 2015.

[10] L. K. Liu, S. H. Chan, and T. Q. Nguyen, "Depth reconstruction from sparse samples: Representation, algorithm, and sampling," *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1983–1996, June 2015.

[11] C. Cadena, A. R. Dick, and I. D. Reid, "Multi-modal auto-encoders as joint estimators for robotics scene understanding." in *Robotics: Science and Systems*, 2016.

[12] F. Ma, L. Carlone, U. Ayaz, and S. Karaman, "Sparse sensing for resource-constrained depth reconstruction," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 96–103.

[13] Y. Liao, L. Huang, Y. Wang, S. Kodagoda, Y. Yu, and Y. Liu, "Parse geometry from a line: Monocular depth estimation with partial laser observation," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5059–5066.

[14] F. Ma and S. Karaman, "Sparse-to-dense: Depth prediction from sparse depth samples and a single image," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[15] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, 2016.

[16] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial Intelligence*, 1981, pp. 674–679.

[17] T. Zhou and D. Tao, "GoDec: Randomized lowrank and sparse matrix decomposition in noisy case." in *International Conference on Machine Learning, ICML 2011*, 2011, pp. 33–40.

[18] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, "Digital photography with flash and no-flash image pairs," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 664–672, Aug. 2004.

[19] S. W. Jung, "Enhancement of image and depth map using adaptive joint trilateral filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 2, pp. 269–280, 2013.

[20] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, June 2013.

[21] Y. Shi, K. Wang, C. Chen, L. Xu, and L. Lin, "Structure-preserving image super-resolution via contextualized multi-task learning," *IEEE Trans. Multimedia*, vol. PP, no. 99, pp. 1–1, 2017.

[22] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, "Conditional random fields as recurrent neural networks," in *IEEE International Conference on Computer Vision*, 2015, pp. 1529–1537.

[23] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition*, Jun 1994, pp. 593–600.

[24] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant CNNs," in *International Conference on 3D Vision (3DV)*, 2017, pp. 11–20.

[25] Z. Yang, F. Gao, and S. Shen, "Real-time monocular dense mapping on aerial robots using visual-inertial fusion," in *2017 IEEE International Conference onRobotics and Automation (ICRA)*. IEEE, 2017, pp. 4552–4559.