https://doi.org/10.1109/ICAR46387.2019.8981653

Map Point Optimization in Keyframe-Based SLAM using Covisibility Graph and Information Fusion

Edison Kleiber T. Concha[1], Diego Pittol[1], Ricardo Westhauser[1],
Mariana Kolberg[1], Renan Maffei[1] and Edson Prestes[1]

***Abstract*— Keyframe-based monocular SLAM (Simultaneous Localization and Mapping) is one of the main visual SLAM approaches, used to estimate the camera motion together with the map reconstruction over selected frames. These techniques represent the environment by map points located in the three-dimensional space, that can be recognized and located in the frame. However, these techniques usually cannot decide when a map point is an outlier or obsolete information and can be discarded. Another problem is to decide when combining map points corresponding to the same three-dimensional point. In this paper, we present a robust method to maintain a refined map. This approach uses the covisibility graph and an algorithm based on information fusion to build a probabilistic map, that explicitly models outlier measurements. In addition, we incorporate a pruning mechanism to reduce redundant information and remove outliers. In this way, our approach manages to reduce the map size maintaining essential information of the environment. Finally, in order to evaluate the performance of our method, we incorporate it into an ORB-SLAM system and measure the accuracy achieved on publicly available benchmark datasets which contain indoor images sequences recorded with a hand-held monocular camera.**

I. INTRODUCTION

In recent years the interest in using cameras as sensors in SLAM has increased, and some authors have been concentrating on building 3D models using visual information [1], [2], [3]. The reasons for this interest are not only because of their low power consumption, small size, and cost, but also for their ability to provide rich information about the surrounding environment, such as color, texture, motion, and structure.

Environment models or maps serve as essential resources for an autonomous robot by providing it with the necessary relevant information about the scenario. Their use enables robots to perform their tasks more reliably, flexibly, and efficiently. For instance, a map can inform path planning or provide an intuitive visualization for a human operator. Besides, they allow limiting the error produced in estimating the state of the robot.

Recently, many keyframe-based SLAM methods have been presented which were proven effective to accurately estimate trajectories while geometrically reconstructing the unknown environment. These methods represent the scene as a set of sparse 3D landmarks corresponding to discriminative features in the environment (e.g., points, lines, polygons) [4], [5], [6], and retain a selected subset of previous observations called keyframes that explicitly represent past knowledge gained [7], [3]. A common assumption underlying these representations is that the landmarks are distinguishable and provide a descriptor which establishes a data association between each measurement and the corresponding landmark. In this sense, the robot can operate for an extended time and revisit a place several times, while new information is continuously added in the map. However, this becomes problematic since the size of the map grows only with the mapping duration and not with the size of the area explored. Furthermore the new information can be repetitive, or, even worse, outliers. The inclusion of a single outlier degrades the quality of the estimate, which in turn degrades the capability of discerning outliers later on. Therefore, it is necessary to have approaches that can deal with repetitive information and outliers to maintain a refined representation of the map.

In this paper, we present a new method to maintain a refined map through the pruning of map points that can have a direct influence on the performance of the visual SLAM process. The approach aims at the remotion of outliers generated from a poor depth estimate, or map points visualized only in some frames, that over time becomes obsolete information. Our method also deals with repetitive information in order to maintain a good quality map and counteract the effect of the frequent addition of features. This paper is organized as follows: Section II deals with related works. Section III presents our method in details. Section IV shows the results of the experimental validation of the proposed approach. Finally, Section V provides our conclusions.

II. RELATED WORK

Over the last decade, numerous efforts have been made towards minimizing the computational requirements of SLAM by reducing the number of variables (observations and poses) in the state space, while keeping the sparse structure of the problem. Due to the popularity of graph-based optimization solutions for SLAM, researchers investigated how to reduce the number of nodes in the SLAM graph. Some approaches focused on determining which node to remove from the graph and how to treat the resulting graph. Konolige et al. [8] clustered nodes in the graph according to their spatial distance. They remove the least recently used nodes among each cluster, in order to keep a limited number of nodes and still capture the dynamic nature of the environment. A similar idea has been introduced by Eade et al. [9], who propose

[1] Institute of Informatics, Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil ektconcha, dpittol, rswesthauser, mariana.kolberg, rqmaffei, prestes@inf.ufrgs.br

This work was supported by CNPq and CAPES [Finance Code 001]

to remove nodes without data or associated with similar observations of existing nodes in the vicinity. Moreover, Ila et al. [10] use an information-theoretic approach which only keeps non-redundant poses and highly informative measurements to the graph. Finally, Johannsson et al. [11] reuse already existing poses in previously mapped areas, keeping the number of poses bounded by the size of the explored environment and use the new measurements to improve the map.

Keyframe-based approaches are among the first attempt to reduce the pose graph for visual SLAM. Such approaches create a sparse pose graph, where each node is a selected keyframe representing the prominent visual appearances and variations in order to keep the density of poses constant. In 2007, PTAM was introduced by Klein and Murray [7], which provides simple but effective methods for keyframe selection, patches matching, point triangulation, camera localization, and relocalization after tracking failure. Unfortunately, several factors limit its application, and one of the biggest problems is the constant insertion of keyframes, even if the camera is looking at the scene from different viewpoints, that causes excessive growth of the map size. More recently, the ORB-SLAM system has been proposed [3], expanding the versatility of PTAM to environments that are intractable for that system. ORB-SLAM integrates a very efficient place recognition system to perform relocalization and loop closing. In addition, it incorporates a pruning mechanism to maintain a compact reconstruction, which detects redundant keyframes and delete them, using a minimum visual change criterion instead of using a distance criterion to other keyframes as PTAM. This mechanism allows a flexible map expansion during the exploration.

In this sense, our approach uses a covisibility graph [12] similar to the one employed by ORB-SLAM and an information fusion algorithm as Forster et al. [13] or Pizzoli et al. [14], to reduce the number of outliers and combine repetitive information in the map. Then, we represent the depth information of each map point as a mixture of probability distributions and take advantage of the keyframe neighborhoods created in the covisibility graph to update these probabilities and thus maintain the highest possible accuracy of depth estimation. Additionally, to reduce outliers in the map and the number of redundant keyframes, a pruning policy based on the depth accuracy of the map points is performed.

III. MAP POINT OPTIMIZATION IN KEYFRAME-BASED SLAM

In the following, we introduce our approach that uses a covisibility graph [12] and an algorithm of information fusion [14] to maintain a refined map in keyframe-based monocular SLAM process.

A. Map Representation

Our map is represented as a set of map points $\{{}^{w}p_j\}_{j=1}^{M}$, keyframes $\{{}^{w}K_i\}_{i=1}^{N}$, and an undirected weighted graph G known as covisibility graph, where each node is a keyframe, and an edge between two keyframes exists if they share at least θ common map points.

Each map point ${}^{w}p_j$ corresponds to an ORB feature, which represents a textured planar patch in the world whose position has been triangulated from different views. Each map point stores:

${}^{w}X_j \in R^{3\times 1}$	3D position in the world coordinate system
$n_j \in R^{3\times 1}$	viewing direction
d_{min}, d_{max}	minimum and maximum distances at which it can be observed, according to the scale invariance limits of the ORB features
D_j	representative ORB descriptor, whose hamming distance is minimum to all other associated descriptors in the keyframes where the map point is observed

On the other hand, each keyframe ${}^{w}K_i$ in the world stores:

F_i	set of all ORB features extracted in the frame
${}^{w}T_i \in SE(3)$	camera pose, which is a rigid body transformation that transforms map points from the world to the camera coordinate system
$K \in R^{3\times 3}$	camera intrinsics parameters, including focal length and principal point

B. Probabilistic Depth Sensor

During the map creation, the robot collects information and builds a representation of the environment where it is located. In keyframe-based SLAM, cameras are commonly used to get such information by performing depth measurements through the captured images. These measurements are always subject to errors called noise, and there may also be seemingly random measurements that are caused by photometric inconsistency. In this way, when the task is to create an accurate map of the environment from such noisy measurements, a probabilistic approach is necessary. Thus, we model each depth measurement d obtained by the sensor as a distribution that mixes a good measurement model with a bad one, as in the work of Forster et al. [13]. The good measurement is normally distributed around the true depth $\hat{d}$ whereas the bad one is uniformly distributed in all possible depth locations in the interval $[d_{min}, d_{max}]$, which is known to contain the true depth. Our mixture model distribution is defined as

$$p\left(d \mid \hat{d}, \rho\right) = \rho \mathcal{N}\left(d \mid \hat{d}, \tau^2\right) + (1-\rho)\, U(d \mid d_{min}, d_{max}), \tag{1}$$

where d is the depth measurement, the parameter ρ indicates the purity of the measurement (inlier probability) and τ^2 is the variance of a good measurement, which can be computed geometrically by assuming a fixed variance of one pixel in the image plane defined by the relative position of the cameras that produced the measurement [14].

C. Depth Bayesian Inference

The uncertainties in sensors arise not only from the imprecision and noise in the measurements, but also from the ambiguities and inconsistencies present in the environment, and by the inability to distinguish them. Information fusion algorithms can exploit redundant data to alleviate such effects. Briefly, we can define information fusion as the process of integrating multiple information sources to obtain improved and useful information as accurately possible [15]. The creation of new map points is necessary to represent the new information every time that a new keyframe is selected. In this sense, we first update the covisibility graph by adding a new node for the new keyframe and creating the covisibility edges, as shown in Fig. 1. Due to these updates in the covisibility graph, a vicinity is generated for this new keyframe which we use to compute new depth measurements. Later, we collect and combine these measurements using information fusion to infer a single depth estimation with the higher possible accuracy. In order to achieve this goal, we use Bayesian inference [16].

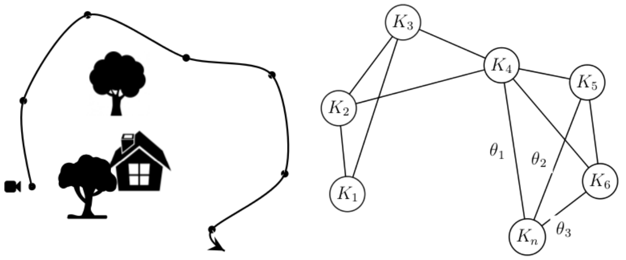


Fig. 1: Covisibility graph with nodes representing keyframes and edges representing their covisibility [12].

Given a new keyframe ${}^{w}K_n$ and a covisibility graph G, for each feature found in the new keyframe there is a set of noisy depth measurement denoted by d_i for $i = 1,...,k$ obtained from the triangulation of the new keyframe with k neighbours of the covisibility graph. In order to simplify the problem and allow the use of Bayesian estimation we assume that all measurements $d_1,...,d_k$ are independent. Therefore the depth posterior is approximated by the product of the marginal probabilities,

$$p(\hat{d},\rho \mid d_1,...,d_k) \propto p(\hat{d},\rho)\prod_k p(d_k \mid \hat{d},\rho), \tag{2}$$

with $p(\hat{d},\rho)$ being a prior on the true depth and the inlier probability.

Vogiatzis et al. [17] prove that the posterior can be approximated by the product of $Gaussian \times Beta$ distributions which minimizes the Kullback-Leibler divergence from the true posterior,

$$p(\hat{d},\rho \mid d_1,...,d_k) \propto \mathcal{N}(\hat{d} \mid \mu_k,\sigma_k^2)Beta(\rho \mid a_k,b_k), \tag{3}$$

which can be parametrized with four parameters, where the first two parameters, a_k and b_k control the Beta distribution. The other two, μ_k and σ_k^2 represent the mean and variance of the Gaussian distribution. This leads to

$$q(\hat{d},\rho \mid a_k,b_k,\mu_k,\sigma_k^2) \propto \mathcal{N}(\hat{d} \mid \mu_k,\sigma_k^2)Beta(\rho \mid a_k,b_k). \tag{4}$$

Once defined the posterior, the Bayesian estimation allows us to integrate new measurements in order to update the posterior that describes a new knowledge state, as given by

$$\begin{aligned} q(\hat{d},\rho \mid a_k,b_k,\mu_k,\sigma_k^2) = {} & q(\hat{d},\rho \mid a_{k-1},b_{k-1},\mu_{k-1},\sigma_{k-1}^2) \\ & p(d_k \mid \hat{d},\rho) \end{aligned} \tag{5}$$

Then, using Eq. (1) and (4) in Eq. (5), and the definition of Gaussian and Beta distributions, as well as the properties of the gamma function, we obtain

$$\begin{aligned} q(\hat{d},\rho \mid a_k,b_k,\mu_k,\sigma_k^2) = {} & \frac{a_{k-1}}{a_{k-1}+b_{k-1}}\mathcal{N}(d_k \mid \mu_{k-1},\tau_k^2+\sigma_{k-1}^2) \\ & q(\hat{d},\rho \mid a_{k-1}+1,b_{k-1},m,s^2) \\ & + \frac{b_{k-1}}{a_{k-1}+b_{k-1}}U(d_k \mid d_{min},d_{max}) \\ & q(\hat{d},\rho \mid a_{k-1},b_{k-1}+1,\mu_{k-1},\sigma_{k-1}^2), \end{aligned} \tag{6}$$

and we can observe that a_k and b_k can be thought of as probabilistic counters of how many inlier and outlier measurements have occurred during the lifetime of the map point. Furthermore m and s^2 represent the new mean and variance obtained from the product of two Gaussian functions, and are given by

$$m = \frac{\sigma_{k-1}^2 d_k + \tau_k^2 \mu_{k-1}}{\tau_k^2+\sigma_{k-1}^2}, \tag{7}$$

$$s^2 = \frac{\tau_k^2\sigma_{k-1}^2}{\tau_k^2+\sigma_{k-1}^2}, \tag{8}$$

Finally, using Eq. (6) and matching the first and second moments of the Gaussian and Beta distributions, we can obtain the new posterior parameters a_k, b_k, μ_k, σ_k^2, from the old parameters a_{k-1}, b_{k-1}, μ_{k-1}, σ_{k-1}^2, and the new measurement d_k. For more details on this derivation we refer to the work in [17].

D. Map Point Optimization

To maintain a refined map, we need a method to detect and reject outliers and obsolete informations on all observations made on the new keyframe. Firstly, we define as obsolete informations those map points that over time do not contribute to the SLAM process. In this way, we have considered obsolete those map points which have not been observed in at least three keyframes, this ensure that more than one measurement has been used in estimating the depth posterior distribution. Secondly, in order to identify outliers we use a method that uses only the information contained in the depth posterior distribution. Our approach takes into account the amount of inlier measurements that have occurred and the amount of information gained. In this sense, given the depth posterior for all observations in the new keyframe we assign a state for each of them using the conditional defined as:

$$S(q_k) = \begin{cases} \text{Converged,} & \text{if } \frac{a_k}{a_k+b_k} > \eta_{in} \text{ and } \sigma^2 < \sigma_{cnvg}^2 \\ \text{Diverged,} & \text{if } \frac{a_k-1}{a_k+b_k-2} < \eta_{out} \\ \text{Update,} & \text{otherwise,} \end{cases} \tag{9}$$

where q_k represent the depth posterior distribution defined in (4), σ_{cnvg} is the gained information threshold, whereas η_{in} and η_{out} are the inlier and outlier threshold, respectively. In (9) we can observe that there are three possible outcomes.

First, if the mode[1] of the beta distribution is less than η_{out} then we conclude that the depth estimation has failed to converge due to unreliable measurements. Therefore this map point is removed from the map. Second, if the mean[2] of the beta distribution is bigger than η_{in} and the variance of the normal distribution is less than σ^2_{cnvg} then we assume that the depth has converged to a good estimate and this map point remains stable; otherwise it waits for new measurements to be integrated.

Note that the mean of the beta distribution is used to evaluate convergence while the mode of the same distribution is used to evaluate divergence. However, the mean could also be used to evaluate the divergence, but this value rapidly tends to extremes as the parameters of the posterior are updated through equation (5). Due to this behavior, we chose to use the mode, allowing to maintain the map points for more time waiting for new measurements to be integrated. Finally, to deal with repetitive information, the map point is searched in the neighboring keyframes, if a match is found it is integrated using the method explained in the previous section and once again the criteria of Eq. (9) is applied.

IV. EXPERIMENTS

The proposed method was evaluated using the TUM RGB-D benchmark [18]. The benchmark contains 39 sequences that were captured by a Microsoft Kinect sensor. Each sequence contains both the color and depth images in full sensor resolution (640x480) at video frame rate (30 Hz), and the ground truth for camera motion that was provided by a motion capture system with eight high-speed tracking cameras (100 Hz). We have selected 10 sequences, which are also used in other works [3], [2]. Our experiments were performed on a desktop computer with Ubuntu 14.04, equipped with Intel Core 2 Quad processor and 4GB of RAM. The keyframe-based monocular SLAM system and our method are implemented in C++. The values for the parameters of our method are established as in the work of Vogiatzis et al. [17] with the inlier threshold $\eta_{in} = 0.7$, the outlier threshold $\eta_{out} = 0.05$ and the variance threshold σ^2_{cnvg} was set at $1/1000th$ of the bounding volume size $d_{max} - d_{min}$.

Next, in order to allow quantitative comparison between obtained trajectories and ground truths, we computed two error metrics proposed in [18]: the relative pose error and the absolute keyframe trajectory error. Finally, to compare our results, we executed the original ORB-SLAM system over the same sequences.

A. Removing outliers

To evaluate the resulting map, we used a quantitative metric such as the map size. Table I provides a summary of the performance of ORB-SLAM with and without our proposed method on the 10 sequences. This table shows the number of keyframes and map points, the relative and the absolute error of the trajectory. Comparing the sizes of the maps obtained by ORB SLAM with and without our proposed method, we can see that our approach always manages to reduce a significant percentage of points and keyframes. On the other hand, it is important to check if the process of decreasing the map size affects the accuracy of the estimated trajectory, which is done in the analysis of pose errors.

B. Relative Pose Error (RPE)

The relative pose error measures the difference between the estimated and the true motion and is used to evaluate the local accuracy or the drift of a visual odometry system over a fixed time interval Δ. To compute the RPE, the relative transformation between consecutive poses of the estimated trajectory P and the ground truth Q are compared at time step i using

$$E_i = (Q_i^{-1} Q_{i+\Delta})^{-1} (P_i^{-1} P_{i+\Delta}), \tag{10}$$

thus from a sequence of n camera poses are obtained $m = n - \Delta$ individual relative pose errors. Later, the root mean square error (RMSE) is computed over all translational components of these errors along the sequence.

Fig. 2 shows the relative pose error obtained in our experiments for *fr3_long_office*, *fr2_desk* and *fr2_desk_person* sequences. In Fig. 2d, 2e and 2f the camera trajectories estimated by ORB-SLAM with our method are compared to the ground truth trajectories. We can observe that on each of these plots, the camera trajectory estimated is close to the ground truth. The results are very similar to the ones in Fig. 2a, 2b and 2c which are estimated by original ORB-SLAM. Furthermore, Table I shows the quantitative results of the RMSE in meters (m) for the relative pose error computed in each sequence. The results show a similar RMSE, implying that the result of the trajectory estimation is usually not degraded by the removal of obsolete information and outliers. In fact, generally there was a decrease in the relative pose error. The worst difference of RPE in terms of percentage happened in dataset number 7, but, even in this case, the difference in terms of absolute distance was of only $2cm$ ($0.021m$).

C. Absolute Keyframe Trajectory Error (ATE)

We also checked the absolute keyframe trajectory error, which focuses on global consistency and is used to evaluate the performance of visual SLAM systems. The absolute distances between the estimated keyframes trajectory P and the ground truth Q at time step i are compared using the absolute trajectory error,

$$F_i = Q_i^{-1} S P_i, \tag{11}$$

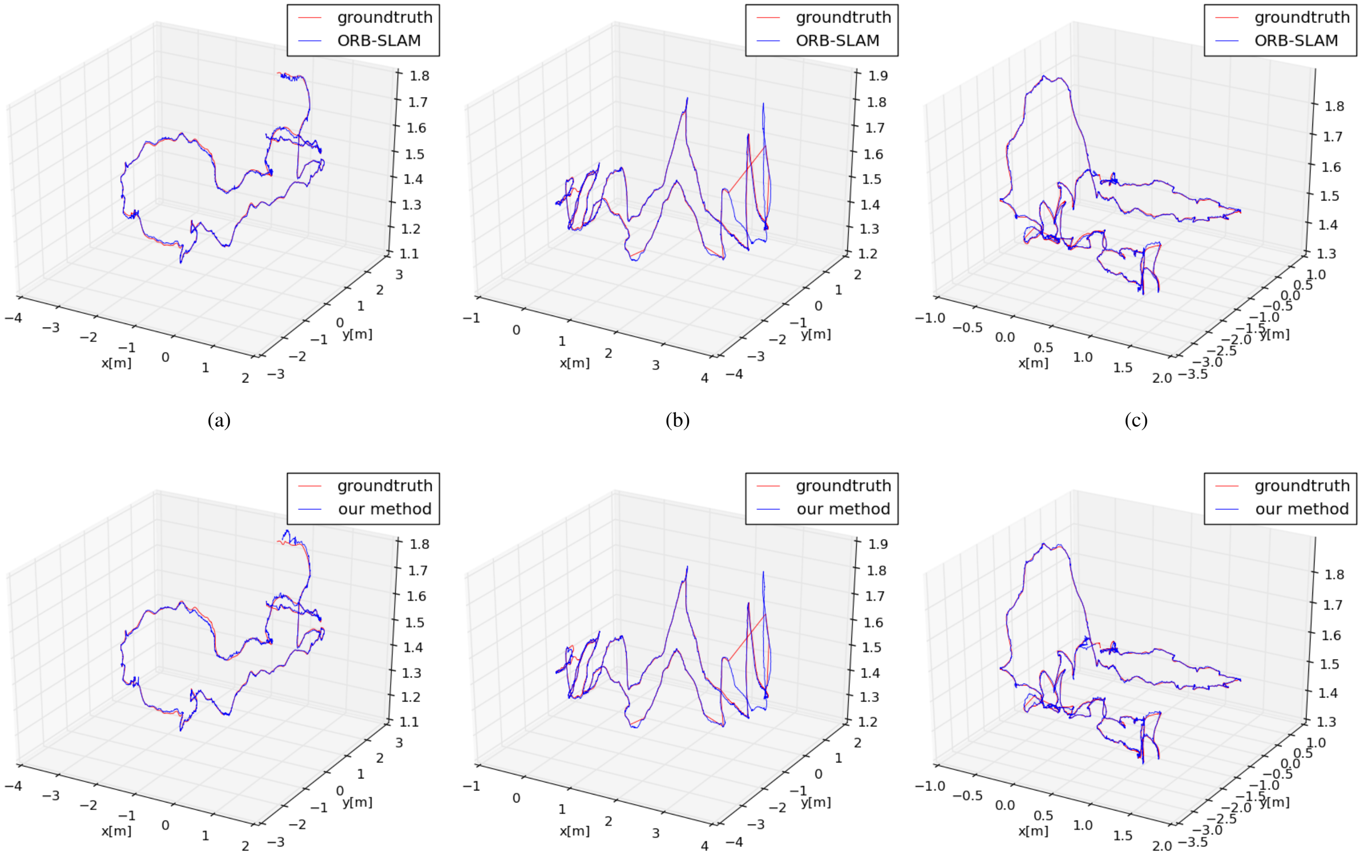
where S is the rigid-body transformation corresponding to the least-square solution to the alignment problem, which maps the estimated keyframes trajectory P onto the ground truth Q. Then similar to the RPE, the RMSE is computed over all translational components of the relative pose error in all time indices.

[1] The mode of a beta distribution $Beta(a,b)$ is given by $\frac{a-1}{a+b-2}$.

[2] The mean of a beta distribution $Beta(a,b)$ is given by $\frac{a}{a+b}$.

TABLE I: Quantitative evaluations for 10 sequences from the TUM benchmark [18]. From left to right, the columns show: the dataset name; the path length; the number of keyframes and map points in both methods (the original ORB-SLAM and the proposed method); the root mean square error (RMSE) of absolute keyframe trajectory (ATE); and the relative pose error (RPE) in terms of translation. We also show the percentages of reduction for the four metrics.

	Dataset	Length [m]	Keyframes			Points			ATE (RMSE) [m]			RPE (RMSE) [m]		
			ORB	Our	%	ORB	Our	%	ORB	Our	%	ORB	Our	%
1	*fr1_desk*	9.263	63	60	**-4.8**	3236	2785	**-13.9**	0.014	0.014	**-2.1**	0.034	0.024	**-28.1**
2	*fr2_xyz*	7.029	37	29	**-21.6**	1582	1355	**-14.3**	0.002	0.002	**-4.0**	0.017	0.018	**+4.6**
3	*fr3_str_tex_far*	5.884	25	23	**-8.0**	1911	1698	**-11.1**	0.009	0.008	**-13.6**	0.097	0.070	**-27.8**
4	*fr3_walk_halfsph*	7.686	45	36	**-20.0**	1433	1035	**-27.8**	0.016	0.016	**-3.0**	0.269	0.178	**-33.7**
5	*fr3_str_tex_near*	5.050	49	43	**-12.2**	3188	2969	**-6.9**	0.011	0.011	**-3.5**	0.032	0.040	**+26.5**
6	*fr3_sit_halfsph*	6.503	76	69	**-9.2**	2570	2096	**-18.4**	0.234	0.010	**-95.7**	0.235	0.113	**-51.9**
7	*fr3_nstr_tex_near*	13.456	67	62	**-7.5**	4542	4032	**-11.2**	0.014	0.013	**-4.3**	0.061	0.085	**+40.7**
8	*fr3_long_office*	21.455	198	170	**-14.1**	10175	8250	**-18.9**	0.010	0.014	**+39.2**	0.157	0.158	**+0.7**
9	*fr2_desk*	18.880	177	149	**-15.8**	7288	5914	**-18.9**	0.008	0.017	**+104.8**	0.093	0.089	**-4.5**
10	*fr2_desk_person*	17.044	119	106	**-10.9**	4565	3863	**-15.4**	0.010	0.008	**-16.7**	0.100	0.114	**+14.6**



(a)

(b)

(c)

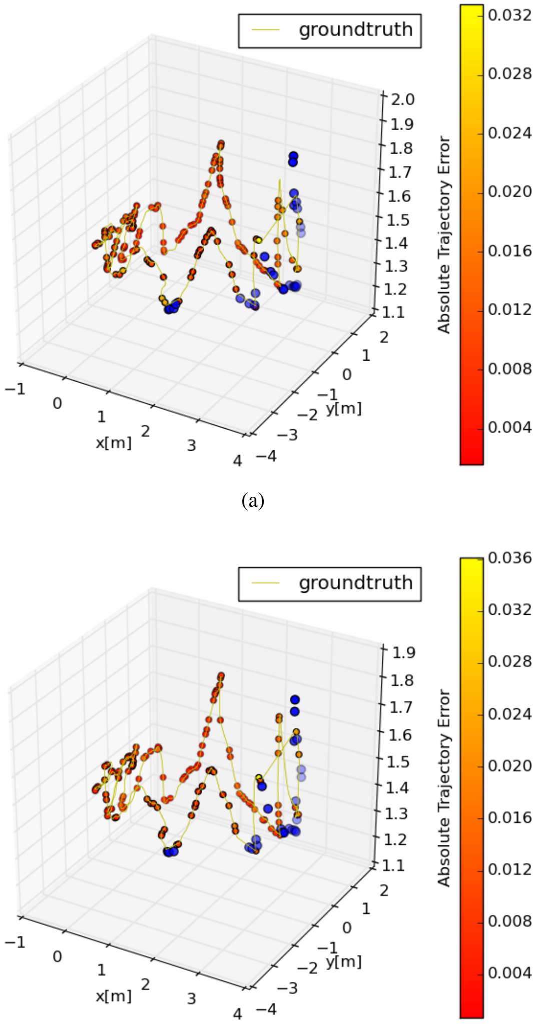
(d)

(e)

(f)

Fig. 2: Relative Pose Error evaluation for *fr3_long_office*, *fr2_desk* and *fr2_desk_person* sequences, showing the results obtained by ORB-SLAM without our method (2a) (2b) (2c) and with our method (2d) (2e) (2d).

Fig. 3 presents the comparison of ATE in a single dataset. Each point represents a keyframe, and the blue points are those keyframes in the estimated trajectory which are not considered in the absolute trajectory error computation because their matches were not found onto the ground truth at the time of alignment. Fig. 3a, representing the results of the original ORB-SLAM, shows regions with a high density of keyframes where there is possibly repeated information, outliers or obsolete information. In Fig. 3b, representing our approach, this density decreases. This happens because ORB-SLAM has a policy that removes keyframes whose 90% of the map points have been seen in at least other three



(a)

(b)

Fig. 3: Absolute Keyframe Trajectory Error evaluation for *fr2_desk* sequence, showing the results obtained by ORB-SLAM without our method (3a) and with our method (3b).

keyframes. Therefore, if the size of the map points decreases, then the number of keyframes also decreases. Finally, Table I shows the quantitative results of the RMSE in meters (m) for the absolute keyframe trajectory computed in each sequence, where we can observe that the RMSE produced by ORB-SLAM with our method is very close to the produced by the original ORB-SLAM. The worst result in terms of ATE was measured in dataset number 9, in which the error value doubled. However the original error was so small that the increase in terms of absolute value was of less than $1cm$ ($0.009m$).

V. CONCLUSIONS

In this paper, we have demonstrated how the combined use of the covisibility graph with an information fusion algorithm allows us to maintain a refined map during the keyframe-based SLAM process. We represent the depth information of each map point as a mixture of distributions and we take advantage of the keyframe neighborhoods of covisibility graph to improve depth accuracy. We also perform a pruning strategy based on the depth accuracy; in this way, our method removes possible outliers and deals with repetitive information. Finally, we have incorporated our method within a visual SLAM system, and its effectiveness was demonstrated through extensive experiments on publicly available data, showing that the presented method is beneficial because it reduces the size of the map representation without compromise the quality of the trajectory estimate when the robot is continuously operating in the same environment.

REFERENCES

[1] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, “DTAM: Dense tracking and mapping in real-time,” in *International Conference on Computer Vision (ICCV)*, Nov. 2011, pp. 2320–2327.

[2] J. Engel, T. Schops, and D. Cremers, “LSD-SLAM: Large-scale direct monocular SLAM,” in *European Conference on Computer Vision (ECCV)*, Sept. 2014, pp. 834–849.

[3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM: A versatile and accurate monocular SLAM system,” *IEEE Transactions on Robotics*, vol. 31, pp. 1147–1163, Oct. 2015.

[4] Y. Lu and D. Song, “Visual navigation using heterogeneous landmarks and unsupervised geometric constraints,” *IEEE Transactions on Robotics*, vol. 31, pp. 736–749, June 2015.

[5] H. Zhou, D. Zou, L. Pei, R. Ying, P. Liu, and W. Yu, “StructSLAM: Visual slam with building structure lines,” *IEEE Transactions on Vehicular Technology*, vol. 64, pp. 1364–1375, Apr. 2015.

[6] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, “PL-SLAM: Real-time monocular visual slam with points and lines,” in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 4503–4508.

[7] G. Klein and D. Murray, “Parallel tracking and mapping for small ar workspaces,” in *Proc. IEEE International Symposium on Mixed and Augmented Reality*, Nov. 2007, pp. 225–234.

[8] K. Konolige and J. Bowman, “Towards lifelong visual maps,” in *Proc. IEEE International Conference on Intelligent Robots and Systems*, Oct. 2009, pp. 1156–1163.

[9] E. Eade, P. Fong, and M. E. Munich, “Monocular graph slam with complexity reduction,” in *Proc. IEEE International Conference on Intelligent Robots and Systems*, Oct. 2010, pp. 3017–3024.

[10] V. Ila, J. M. Porta, and J. Andrade-Cetto, “Information-based compact pose SLAM,” *IEEE Transactions on Robotics*, vol. 26, pp. 78–93, Feb. 2010.

[11] H. Johannsson, M. Kaess, M. Fallon, and J. J. Leonard, “Temporally scalable visual slam using a reduced pose graph,” in *Proc. of (ICRA)*. IEEE, May 2013, pp. 3638–3643.

[12] E. Stumm, C. Mei, and S. Lacroix, “Probabilistic place recognition with covisibility maps,” in *Proc. IEEE International Conference on Intelligent Robot and Systems (IROS)*, Nov. 2013, pp. 4158–4163.

[13] C. Forster, M. Pizzoli, and D. Scaramuzza, “SVO: Fast semi-direct monocular visual odometry,” in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 15–22.

[14] M. Pizzoli, C. Forster, and D. Scaramuzza, “REMODE: Probabilistic, monocular dense reconstruction in real time,” in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 2609–2616.

[15] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “Octomap: an efficient probabilistic 3d mapping framework based on octrees,” *IEEE Autonomous Robots*, vol. 34, pp. 189–206, Apr. 2013.

[16] C. M. Bishop, *Pattern Recognition and Machine Learning*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[17] G. Vogiatzis and C. Hernández, “Video-based, real-time multi-view stereo,” *Image and Vision Computing*, vol. 29, pp. 434–441, June 2011.

[18] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of rgb-d slam systems,” in *Proc. IEEE International Conference on Intelligent Robot and Systems (IROS)*, Oct. 2012, pp. 573–580.