Journal of Real-Time Image Processing (2021) 18:419–429
https://doi.org/10.1007/s11554-020-00989-6

SPECIAL ISSUE PAPER

Accurate real-time visual SLAM combining building models and GPS for mobile robot

Ruyu Liu[1] · Jianhua Zhang[1] · Shengyong Chen[2] · Thomas Yang[3] · Clemens Arth[4]

Received: 24 January 2020 / Accepted: 18 May 2020 / Published online: 7 June 2020

Abstract
This paper presents a novel 7 DOF (i.e., orientation, translation, and scale) visual simultaneous localization and mapping (vSLAM) system for mobile robots in outdoor environments. In the front end of this vSLAM system, a fast initialization method is designed for different vSLAM backbones, which upgrades the accuracy of trajectory and reconstruction of vSLAM with an absolute scale computed from depth maps generated by building blocks. In the back end of this vSLAM, we propose a nonlinear optimization mechanism throughout which multimodal data are combined for more robust optimization. The modality of building blocks in optimization can improve the tracking accuracy and the scale estimation. By integrating the pose estimated from visual information and the position received through GPS, the optimization further alleviates the drift. The experimental results prove that the proposed method is extremely suitable for outer AR application for outdoor environments, because our method has superior initialization performance, runs in real time, and achieves real scale, higher accuracy, and robustness.

Keywords Robot localization · Building models · Multimodal fusion · Graph optimization

1 Introduction

Nowadays, industrial robots are entering the era of great change with the development of artificial intelligence and robot technologies. Among these technologies, computer vision as a key instrument endows robots with the abilities to automatically sense the industrial environment and complete complex industrial tasks without human intervention [2, 5]. For successfully executing these tasks, robot localization and mapping is one of the essential abilities, which is the guarantee to other complex tasks.

Due to its importance, SLAM has been continuously developing and gradually emerged in the industry. Especially, visual SLAM has received considerable critical attention. Generally speaking, there are two main categories methods in SLAM: the feature-based method (indirect method) and the direct method. Although they have significant differences in terms of principle, structure, and implementation, the main challenges faced by both of them and even most monocular visual SLAM are the scale drift and accumulate error. If there are few feature points in the scene, feature-based methods usually produce a large drift error in both position and orientation or even fail to work. The same problem cannot be eliminated and well solved in the direct method due to dramatic light changing or textureless and structureless environments. Moreover, inaccurate and strenuous vSLAM initialization is also a crucial issue worthy of attention when vSLAM is applied to industrial robots.

In outdoor industrial environments, the common approach for robot outdoor localization relies on sensors such as GPS and compass to determine its spatial position and orientation. However, the built-in sensors in mobile devices only

✉ Jianhua Zhang
zjh@zjut.edu.cn

Ruyu Liu
liuliu609470295@gmail.com

Shengyong Chen
sy@ieee.org

Clemens Arth
arth@icg.tugraz.at

[1] Zhejiang University of Technology, University of Hamburg, Hangzhou, China

[2] Tianjin University of Technology, Tianjin, China

[3] Electrical and Computer Engineering, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114, USA

[4] Graz University of Technology, Graz, Austria

provide a coarse pose. Additionally, GPS signal reception is also influenced by buildings.

Man-made buildings are everywhere in outdoor industrial environments. The building map consisting of building blocks exhibits strong 3D structural regularity and scale information which quite different from the image information. Recently, some methods [1, 23, 25] leverage the building map to help mobile devices or robotic systems obtain better localization and mapping. Textureless building maps not like the texture maps are not affected by changes in lighting conditions and appearances in the outdoor environment. However, the major challenge is to match and integrate the maps carrying only basic 3D building structure information with the rich 2D vision information perceptive by the robot.

In this paper, we propose an accurate localization and mapping method for mobile robots, which fuses the rich visual information, the geometric building blocks and inaccurate GPS positions. Furthermore, this vSLAM includes an easy-using and fast initialization and an optimization based on multimodal data. The contributions of this paper can be summarized as follows:

1. In the front end of vSLAM, we propose a simple but robust initialization method for our vSLAM system. The method leverages building blocks and a coarse GPS position to create depth maps corresponding to images. By doing this, the robot can directly obtain a global map and its initial position in the environment without tediously covering a sufficient outdoor baseline.
2. We design our initialization methods for a feature-based vSLAM and a direct one. Experiments demonstrate that the proposed method is very compatible with different systems and reduces the computation and uncertainty of systems.
3. In the back end of vSLAM, we propose a hybrid graph-based optimization using multiple modalities, which improves the accuracy and robustness of both the mobile robot trajectory and reconstruction and alleviates scale drift.

2 Related work

2.1 Initial positioning of robot system

The aim of initializing the system is to estimate robot position in the environment and assign the depth value to the landmark points by visual perceived. The current initialization method can be divided into three categories.

The first category is based on multi-view geometry [4, 6, 11, 15, 16] and relies on a certain motion baseline. PTAM [11] is a popular vision-only AR (augmented reality) SLAM system that was put forward earlier. Many later feature-based vSLAM methods are inspired by it and split the system into a camera tracking front end and an optimization-based back end. However, PTAM is designed for the AR application, and the initialization process requires user interaction with the system, which is not suitable for robot application. More recently, ORBSLAM [15, 16] is a typical feature-based vSLAM framework which has been the basis for a lot of follow-up approaches. It also uses the multi-view parallax method. There are basically two ways to initialize a monocular vSLAM system. The first is to use the eight-point algorithm [7] to calculate the homograph matrix for planar scenes and using the five-point algorithm [19] to calculate the fundamental matrix for non-planar scenes. The direct methods, such as DSO (direct sparse odometry) [4], optimize the pose hypothesis to get the initial pose. In these methods, the process of moving the baseline requires user's intervention. Sometimes, it is difficult to fast initialize the vSLAM system even by a professional user. Besides, the initial camera pose obtained by these methods has an ambiguous scale.

The second category relies on visual information of environment such as point, line, plane, space structure, and or even rich features within convolutional neural networks [1, 9, 12–14]. Existing research recognizes the crucial role played by scene depth information in vSLAM initialization. The research involves estimating scene depth based on traditional visual components like vanishing points [9, 13], or the recent emergence of depth prediction using deep learning [14]. The former requires complicated and time-consuming calculations, and the latter requires a large number of datasets to learn features of the depth map.

The third category works directly with sensors. Nowadays, mobile robots achieve localization with the GPS, digital compass, and inertial orientation sensors in an outdoor environment. However, the consumer-level sensors can only provide coarse positioning. In more terrible cases, positioning errors can be up to tens of meters, which is intolerable for robots to perform industrial tasks. Although differential GPS can provide centimeter-level positioning, the high price limits its widespread in industrial robots. LiDAR is also very expensive, and its sparse sampling limits its applications. Depth sensors like RGB-D cameras [10, 18] perform close-range depth estimation within vSLAM, but they are unfortunately limited in the indoor environment because of the restrictions of the depth-sensing hardware.

2.2 Multimodal fusion system for tracking and mapping

Only a single modality cannot fully meet the requirement for the instantaneous positioning of a robot, not to mention for the continuous navigation and moving of a robot. Thus, it is necessary for a robot to fuse multimodal data to complete tasks. Multimodal data can provide complementary cues. Fusing them can provide a more reliable, unified and precise description of the environment and the estimation of robot state. [17, 20, 21] are sophisticated and popular methods about fusing IMU and vision in recent years. Hol et al. [8] present an accurate and robust camera pose estimation for augmented reality. Zhu et al. [26] combine color, depth, and IMU modalities to achieve robust environment perception. Liu et al. [12] propose an approach for outdoor localization using visual SLAM, poor GPS, and 2.5D map. Zhang et al. [24] combine visual odometry and light detection and ranging (LiDAR) odometry to complete robot motion estimation and scene mapping. Caselitz et al. [3] explore the feature matching between the image and the 3D LiDAR map to implement camera localization.

2.3 Differentiation from previous work

Closest to our work is the approach of [1] and [12] in terms of the use of building models. [1] pays more attention to the instant localization using a series of algorithms, while our method more focuses on the whole continuous process of robot navigation and moving. Different from [12], our method is targeted at the industrial robot. We theoretically analyze the reasons why the existing vSLAM initialization methods are not friendly to robot applications. Furthermore, we apply the proposed model to more types of vSLAM methods and demonstrate extensively comparative experiments.

3 Limitations in vSLAM initialization

vSLAM can generally be divided into two categories: the direct method based on optical flow and the method based on feature points, which also known as the indirect method. Among many vSLAM systems, ORBSLAM2 and DSO are widely considered as the most typical representatives and benchmarks of two categories, respectively. Therefore, we subsequently take ORBSLAM2 and DSO as examples to briefly cover the working principle of the initialization in these two categories and analyze the uncertainty of initialization.

There are similarities in the initialization of ORBSLAM2 and DSO. They both need to move a distance and rotate a certain angle, which is complex and unrealistic for robot systems. During the motion, the first frame is set as the reference frame I_r, and the current frame is marked as I_c. p_r and p_c are corresponding points between I_r and I_c. K is the camera intrinsic parameter matrix. ORBSLAM2 and DSO calculate the camera pose (rotation matrix R and translation vector t) and build the initial map based on different pipelines.

The ORBSLAM2 extracts and matches the ORB features between the current frame and the reference frame. If the number of extraction and matching points exceeds a certain threshold, the system calculates the homography matrix H and fundamental matrix F, then selects the appropriate model and decomposes the matrix into relative rotation and translation, and further recovers the 3D initial map. At last, a global bundle adjustment (BA) is performed to refine the initial reconstruction map. If there are not enough points, the system resets the reference frame and then repeats the above process.

When the robot translation is too small, the fundamental matrix F will be close to zero and cannot be decomposed to solve the rotation component. Conversely, the system decomposes the homography matrix H to obtain R. However, the initial 3D map points still cannot be accurately recovered by triangulation due to the small translation.

In addition, the scale ambiguity exists in the initialization. The epipolar constraint equation does not change whether the matrix F is multiplied by any proportional coefficient. Hence, the matrix F has the scale invariance. Furthermore, the translation vector from F inherits the scale invariance. In ORBSLAM2, the system takes the translation value t as a unit and fixes the scale.

Overall, there must be a certain translation between the two or multiple frames during the initialization of ORBSLAM2. The unity of translation further causes the scale ambiguity and system uncertainty.

Different from ORBSLAM2, DSO directly uses the photometric consistency between I_r and I_c without any handcrafted features and descriptors. In the initialization of DSO, the system constructs the image pyramid and calculates the camera intrinsic parameter matrix on each layer of the pyramid image, then selects points with an obvious gradient with respect to their surroundings, and makes them well distribute. The inverse depth of each point is set to 1 at first. For the following frames, the system continues using the multi-scale image pyramid to calculate the photometric error over a small neighborhood of pixels in the current frame I_c.

$$E_{p_c} = \sum_{p_c \in \mathcal{N}_{p_c}} \omega_p \left\| (I_c[p_c] - b_c) - \frac{t_c e^{a_c}}{t_r e^{a_r}} (I_r[p_r] - b_r) \right\| \tag{1}$$

where $\mathcal{N}_{p_c}$ is the set of neighbor pixels, $\|\cdot\|$ the Huber norm, t_c and t_r are the expose times of the frame I_c and I_r. Affine bright transfer function $e^{-a_r}(I_r - b_r)$ is used to make photometric camera calibration to operate on sequences without known exposure times. The transformation relationship between p_r and p_c is

$$p_c = K(RK^{-1}p_r + t) \tag{2}$$

The initialization of DSO aims to recover the rotation matrix R, the translation vector t and the brightness transfer function by minimizing the total photometric error over all points. Therefore, the system subsequently optimizes the full photometric.

$$E_{\text{total}} = \sum_{p_c \in p} E_{p_c} \tag{3}$$

$$\begin{aligned} J &= \frac{\partial f(x)}{\partial \rho_r} = \frac{\partial f(x)}{\partial P_c}\frac{\partial P_c}{\partial \rho_r} \\ &= \sqrt{w_h}\rho_r^{-1}\rho_c\left(\nabla I_x f_x\left(t_x - \frac{X_c}{Z_c}t_z\right) + \nabla I_y f_y\left(t_y - \frac{Y_c}{Z_c}t_z\right)\right) \end{aligned} \tag{4}$$

where $f(x)$ is the photometric error function and $P_c = (X_c, Y_c, Z_c)$ is the 3D point in current frame coordinate system. ρ_r and ρ_c represent the inverse depth of the point in the reference frame and the current frame, respectively. $t = (t_x, t_y, t_z)$ is the camera translation. $\sqrt{w_h}$ is Huber norm. ∇I_x and ∇I_y are the gradient of the pixel in the x and y directions. f_x and f_y are the camera focal length.

When the robot translation t is too small, the Jacobian J with the inverse depth of the point in reference frame degenerates to zero, which will preclude an accurate estimation of inverse depth. The error will be accumulated in the following robot localization and mapping. Additionally, DSO sets the inverse depth of all points as 1 to fix the scale, which also causes the scale ambiguity and the system uncertainty.

The goal of the vSLAM initialization is to estimate the starting pose of the robot and triangulate the initial map points of its surroundings. For industrial robots, the initialization should be independent of human operation and leverage the features of the industrial environment as much as possible. However, through the above analysis, both ORB-SLAM2 and DSO need a large enough triangulation baseline from the two frames or multi-frames to initialize the system. This is not easy for a non-expert to operate, let alone robots. Additionally, the ambiguous scale will also cause the robot to fail to perceive the real industrial environment and determine its global position. In a word, the complex and difficult initialization operation will become a big obstacle when applying monocular vSLAM to industrial mobile robots.

4 Proposed initialization method

As discussed earlier, the requirement of sufficient translational motions or a depth-sensing functionality is necessary for initializing vSLAM. However, the RGB-D dense sensor is not reliable in the outdoor environment. More important is that, for a good industrial robotic system, well-designed and robot-friendly vSLAM initialization method is needed, which means to avoid walking for several meters and any scale drift. Hence, we leverage the pose estimation for sensors to generate a pseudo-sense of depth for initialization, which directly obtains the dense depth value from the existing structured map and uses the cheap and simple device. Simultaneously, the proposed method avoids using the complicated method based on deep learning or traditional vanishing point, which is time-consuming and difficult to deploy. Consequently, our method can guarantee to guide the robot to enter the initialization state with minimal time and effort.

The proposed method avoids a series of trivial steps like points selection, matching, triangulation, and optimization. Conversely, it directly uses a single pseudo-depth obtained from the building models at hand as the prior knowledge of the scene geometry, which provides an approximate depth value for each point in the building region of each frame. A simple and straightforward approach of creating the depth map is to compute the distance value D from the robot's position $O_r = (X_r, Y_r, Z_r)$ to the point $P_b = (X_b, Y_b, Z_b)$ on the building model.

$$D = \sqrt{(X_b - X_{O_r})^2 + (Y_b - Y_{O_r})^2 + (Z_b - Z_{O_r})^2} \tag{5}$$

Alternatively, ray-casting and intersecting points with the building surfaces would be another option. Both are time-consuming tasks; however, generating a dense depth image at the same resolution as the current camera frame can be done in a very efficient way.

The obtained depth value needs to be further encoded and rendered into the depth map which has the same resolution as the corresponding image frame. This processing will provide the robot with a depth map according to its actual position and surroundings.

$$D' = \frac{2.0 + D_{\min} + D_{\max}}{D_{\max} + D_{\min} - (2.0 * D - 1.0)} * (D_{\max} - D_{\min}) \tag{6}$$

where $D_{\min}$ and $D_{\max}$ are valid minimum depth value and maximum depth value.

The entire initialization process can be described as the following steps (Fig. 1). Firstly, the system renders the polygonal building models using its pose from sensors. Given the depth information of buildings, the system retrieves a depth mask. Naturally, the depth image only contains reasonable depth values for regions covered by the model, such that other regions can easily be discarded. The second step varies depending on the different vSLAM systems. A mask is first generated by the building models, and then we filter the ORB features within this building mask and send them to the ORBSLAM2 system, while DSO extracts the pixels with a sufficiently high image gradient on the image region covered by the mask. Third, the depth information of the

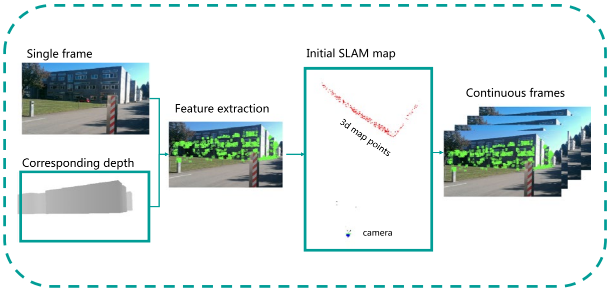


Fig. 1 The initialization process. First, the system leverages the pose from the sensor and building blocks to generate a depth image. Then, the feature points on the building region are extracted. Thereby, the initial vSLAM map which consists of many 3D points is recovered. Subsequently, the vSLAM system collects continuous frames to track and map based on the 3D–2D correspondences

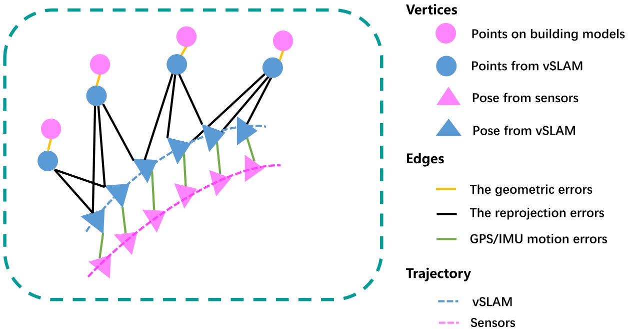


Fig. 2 The framework of our hybrid graph-based optimization. We design the graph optimization method in back end of SLAM, which fuses GPS/IMU, visual SLAM, and building map. The vertices in the graph are camera pose and 3D vSLAM points. The error functions as edges to constrain the vertices

selected point can be calculated using distance estimated by Eq. 6. Thus, 2D plane points together with its corresponding distance information possessing a real scale make up to the potential SLAM map points for initialization and subsequent tracking.

5 Hybrid graph-based optimization

After obtaining the starting pose and initial map, the robot keeps moving and mapping. Different single modalities have their advantages and limitations. For example, sensors built-in robots are cheap, lightweight, and power-saving. They are favorable to be applied in robot platforms but suffer from inevitable imprecision. The building maps together with GPS information contribute a global metrics scale in a wide-area industrial scene. The building models are freely obtained from open map software or reconstructed dense point clouds map. The vSLAM provides robots with an accurate registration in limited local regions; therefore, when it is applicable in outdoor large-scale industrial environments, a large drift may occur. Thus, our system fuses multiple complementary modalities in a uniform optimization to provide the robot with accurate tracking and mapping.

At different stages of the system, our method focuses on the optimization of different modalities. At the initialization stage, the system only optimizes the robot pose while keeping the map static. If the system optimizes the pose and map together, it will disrupt the real scale map created by our initialization approach, since the current vSLAM system possessing a real scale map is not stable and the initial map is also not compatible with the whole vSLAM system. At the following stage, the map points are integrated with multi-view observation from different robot poses. The optimization of robot poses and map points can be safely enabled, where the regular reprojection error and the sensor motion error are used to upgrade the robot states and the reconstructed environment. Eventually, as the system is stabilizing, the geometric error based on the building model will be added to optimization.

We design a hybrid graph optimization method in the back end of vSLAM, which fuses visual information, GPS/IMU, and building models to improve the accuracy of tracking and mapping and keep the scale correct. As shown in Fig. 2, the vertices in graph optimization are camera pose and 3D map points. The multimodal error functions as edges in graph constrain the vertices, and they are the reprojection error based on visual information, the sensor motion error based on GPS/IMU information, and the geometric error based on the building models.

5.1 Reprojection error

Reprojection error function is a common configuration throughout the optimization of vSLAM. In our hybrid optimization method, it is used as the first type of edge to describe the projection relationship between the 3D points and the pixels, and to constrain the robot pose and map points. In particular, after the system optimizes the map points using geometric error based on building models, the reprojection relationship is further used to upgrade the robot pose from which the robot can observe these map points.

$$T_{f_k} = \arg\min_{T_{f_k}} \sum_{k}^{n} \rho \|p_i - KT_{f_k} P_i\|^2 \tag{7}$$

$$T_{f_k}, P_i = \arg\min_{T_{f_k}, P_i} \sum_{k}^{n} \rho \|p_i - KT_{f_k} P_i\|^2 \tag{8}$$

where T_{f_k} represents the pose of frame f_k, p_i is the selected point on the image plane, and P_i is its corresponding 3D space map point. ρ is the robust Huber cost function.

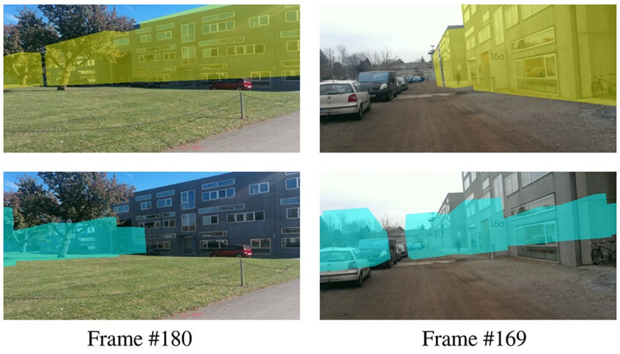

Frame #180 Frame #169

Fig. 3 Visual comparison of building model reprojection error for sequence 07–08 and 12–21 using our SLAM-based method (top) and pure GPS and compass-based localization (bottom)

5.2 Sensor motion error

The accurate pose from the sensors can restrain the estimated 7 DOF pose from our system. However, there commonly exist drift problems in GPS and IMU as illustrated in the bottom of Fig. 3. Due to the usage of multimodal data, the system can easily leverage the changes of the depth map to determine the sensor drift and whether to apply the sensor motion error for optimization. Specifically, the system identifies the drift in sensors by comparing the number of key points matches in the building mask among successive frames. Additionally, we design a weakly constrained error function to integrate the sensor modal data into our system because of inaccurate sensor values.

The sensor motion error is used as the second type of edge and describes the difference between the robot pose from vSLAM and the GPS/IMU. Similar to the method of determining the drift, the system compares the change of the depth map before and after the optimization to determine the quality of the optimization result.

$$T_v(i) = \underset{T_v(i)}{\arg\min} \sum_{i}^{n} \|\Delta T_v(i) - \Delta T_s(i)\|^2 \tag{9}$$

where ΔT_v is the relative robot transformation between two adjacent frames estimated from vSLAM, while ΔT_s is the robot transformation obtained from two adjacent GPS/IMU values.

5.3 Geometric error

Over time, it is equally significant for the system to guarantee the accuracy of robot motion. On the one hand, due to the inaccuracies of the sensors and the resulting rough initial map, the error in the system will be accumulated. On the other hand, a single visual SLAM can hardly guarantee the absolute scale of the trajectory and the reconstructed map.

So as to address this problem, the modality of the building model façade is used to constrain the reconstructed 3D points and to eliminate the gap between estimated reconstructed maps and the real buildings. Hence, we design the third type of edge in our hybrid optimization, which is a geometric error based on the building model façade. The method is divided into three steps.

Determining the visible building Façade to robot The first step is to find all building façade observed by the robot. So the robot must first calculate its own position and field of view. The position $t = (t_x, t_y, t_z)$ can be directly obtained from the robot state estimated from the system. The required horizontal field of view $\Delta\theta_H$ can be calculated by the following formula:

$$\Delta\theta_H = 2 * \frac{\arctan\left(\frac{W}{2}\right)}{f} \tag{10}$$

where f is the camera focal length and W denotes the width of image frame.

Then, the robot calculates the equation of the line-of-sight, i.e., ray-trace within $[\theta - \frac{\Delta\theta_H}{2}, \theta + \frac{\Delta\theta_H}{2}]$ at intervals of 4°. Next, the robot calculates the intersection point between the line of sight and the building model façade. The building façade on which the intersection point closest to the robot's location falls is marked as the visible building façade to robot.

$$\mathbf{F}_v = \{F_{11_v}, F_{12_v}, \ldots, F_{ij_v}, \ldots, F_{ml_v}\} \tag{11}$$

where $\mathbf{F}_v$ represents the set of all visible building façades in the building map. The subscript v indicates the façade is visible to the robot. F_{ij_v} represents the jth façade in the ith building. m denotes the total number of visible buildings, and l denotes the number of visible façades in each visible building.

Point-Façade association In order to recover the environment map with the correct scale, the point cloud from the visual SLAM is aligned with the building models. In the process of optimization, only the 3D points belonging to the building façade participate in the calculation. Using the depth mask, the map points that met the conditions can be easily selected through the correlation between the image features and the map points. These points are then associated with their corresponding building façade using the shortest distance equation.

$$\forall p_i, F_{ij_v} = \underset{F_{ij_v} \in \mathbf{F}_v}{\arg\min} \|d(p_i, F_{ij_v})\|^2 \tag{12}$$

where p_i is the orthographic mapping coordinate of the 3D point P_i onto the ground plane.

5.3.1 Building model-based optimization

The geometric error describes the difference of the distance between the reconstructed map and building models. The system minimizes this distance and aligns the reconstructed map with the real building map by restricting the 3D map points, so as to improve the accuracy of robot trajectory in terms of the full 7 DOF(orientation, translation, and scale).

$$P_i = \arg\min_{P_i} \sum_{i=1}^{n} \rho\|Ax_i + By_i + Cz_i + D\|^2 \tag{13}$$

where $P_i = (X_i, Y_i, Z_i)$ is a 3D point in reconstructed map by vSLAM. $Ax + By + Cz + D = 0$ denotes the 3D building façade, where $D = -(AX_b + BY_b + CZ_b)$. $P_b = (X_b, X_b, X_b)$ is a 3D point on the building façade.

6 Evaluation

6.1 Dataset

We validate our robot localization and mapping system in the real world. The place is located in a city environment full of industrial architectural style. A dataset including 5 sequences is collected by a combination of sensors, including a camera, a GPS/IMU. In each sequence, every frame includes one RGB image, a corresponding depth map which is created through the sensor pose estimated according to Sect. 4, and an inaccurate GPS position. And for each sequence, all RGB images are resized into the 640x360 resolution. A building map of the regional environment is also included. The untextured building map consists of the 2D building footprints and its approximate building height. Therefore, it is also called the 2.5D map [1]. The 2.5D maps can be obtained easily and directly from the map tools such as OpenStreetMap,[1] or it can be extracted from dense point cloud maps [22]. It is worth mentioning that the 2.5D map in our method is commonly composed of the rectangular blocks with regular geometry. In most cases, such abstract building blocks conform to the Manhattan-world assumption [25]. Therefore, the application environment of the proposed method can not only be in the industrial environment that conforms to the Manhattan-world assumption but can be extended to more general urban environment. In each sequence, we deliberately rapidly move the sensors with a comparatively small baseline (i.e., small translation among frames with respect to the scene distance). These sequences are named as 01–02, 05–06, 07–08, and 12–21. We randomly separate the 12–21 sequence into two non-overlapping subsequences since it is too long. All the following experiments are carried out in a 3.2-GHZ iMac 2015.

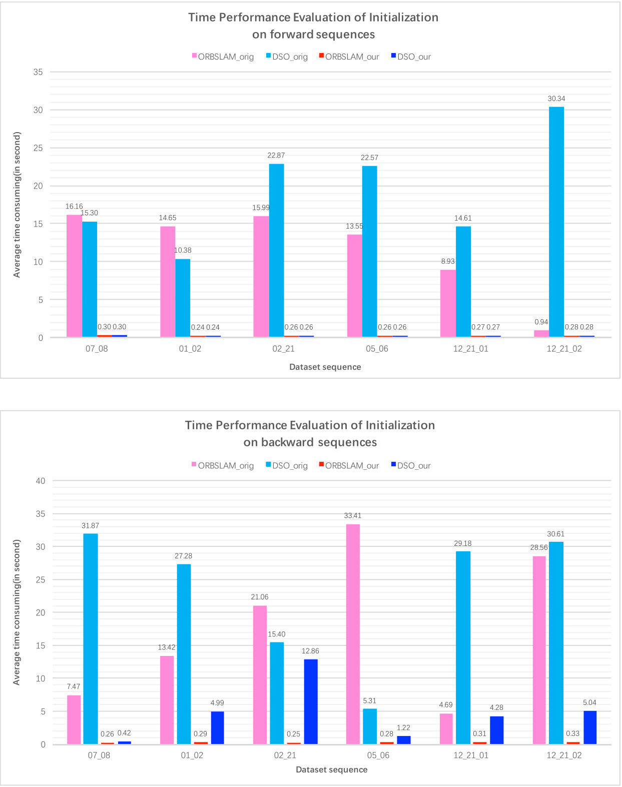


Fig. 4 The time cost of initialization. Our method can combine well with ORBSLAM2 and DSO and significantly improve the time performance of initialization in forward and backward dataset sequences

6.2 Initialization evaluation

In this section, we combine the proposed initialization method with different vSLAM models and further compare it with the initialization methods in ORBSLAM2 and DSO on our dataset.

Time performance evaluation We run each sequence 10 times forward and backward using our initialization with the feature-based method (ORBSLAM2_our), the direct VO (DSO_our), original ORBSLAM2 (ORBSLAM2_orig), and original DSO (DSO_orig), respectively. Then, we record the average time (in seconds) until system successful initialization. In Fig. 4, we illustrate the results of this experiment by histograms. Whether running in forward sequences or backward sequences, our initialization method takes much less time than the traditional initialization methods. In more challenging reverse sequences, the vSLAM system can still be initialized instantly by the proposed method in different models, as shown in the bottom of Fig. 4. During the initialization, the ORBSLAM2 requires sufficient number and quality of matching points in the first two keyframes, and it

[1] https://www.openstreetmap.org.

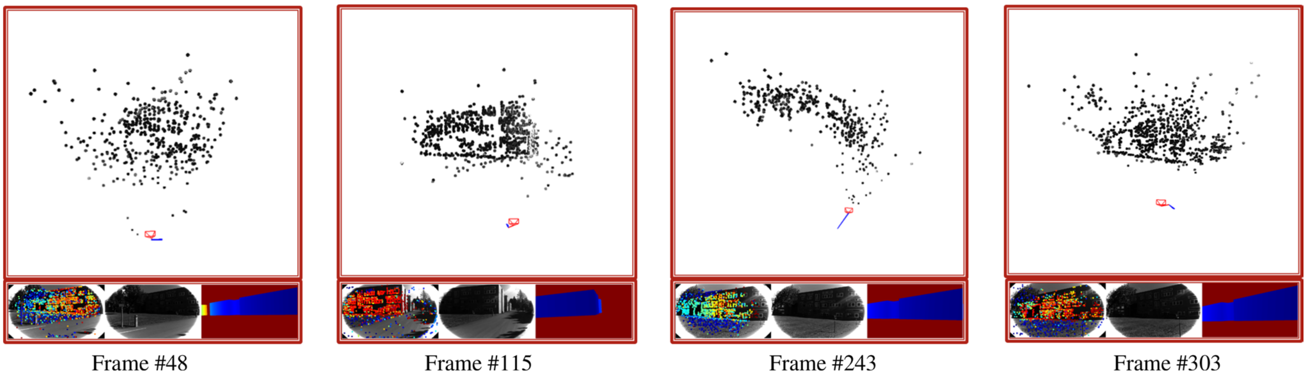


Fig. 5 Our initialization based on DSO framework can initialize or re-initialize the system at any frame in Sequence 07–08

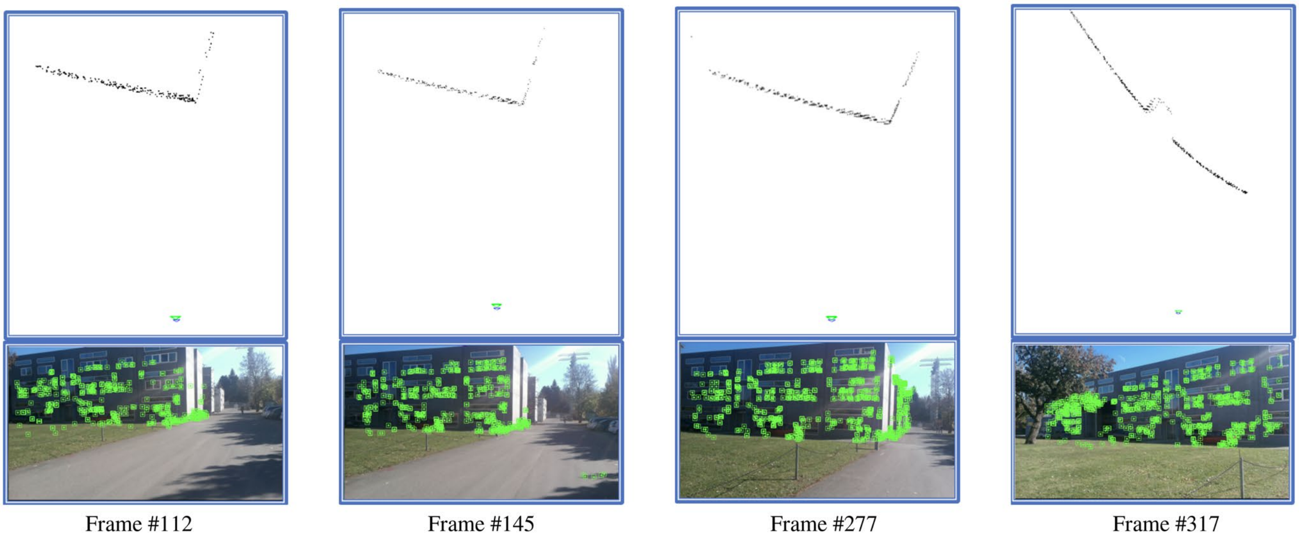
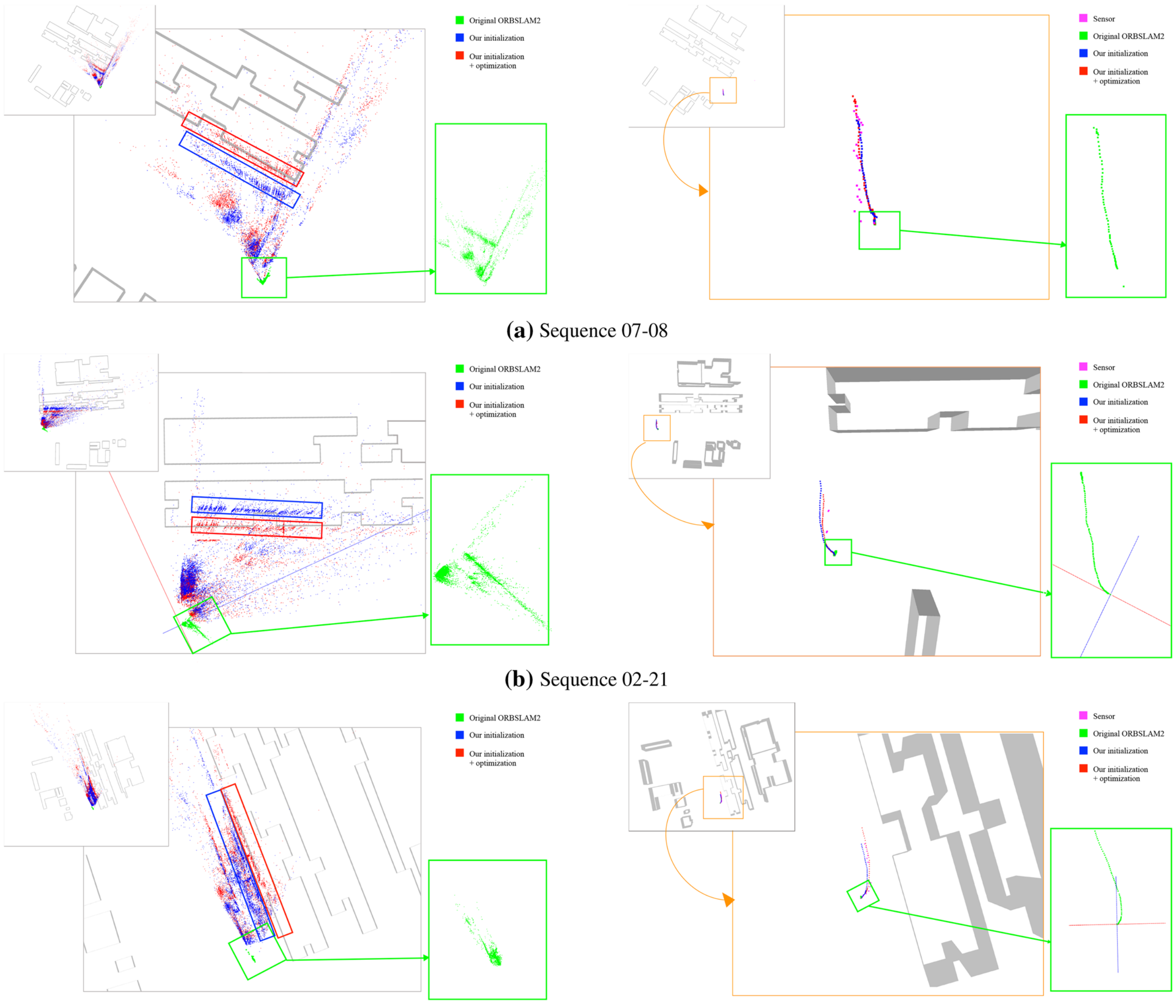


Fig. 6 Our initialization based on ORBSLAM2 framework can initialize or re-initialize the system at any frame in Sequence 07–08

has to last for a considerable number of frames and much more time. Because of the unstable lighting conditions in the environment and complex optimization process, the original DSO takes the most initial time among all methods and even fails during the initialization in some scenes. For example, there are unpredictable initialization failures in Sequence 12_21_02. Our method improves the performance of the original ORBSLAM2 and DSO. Overall, our method can be treated as an instant SLAM initialization, since it needs only a single frame to perfectly complete the initialization and to build the initial world map with a correct scale.

Random initialization In order to complete the tasks, robots may need move freely and flexibly in the industrial environment. Consequently, the robot may initialize or re-initialize at any time and anywhere in the environment. Therefore, the experiments in this part simulate the robot to randomly initialize the system at any frame in each image sequence. As shown in Figs. 5 and 6, ORBSLAM_our and DSO_our methods both have satisfactory results. Our method can quickly estimate the starting robot pose and reconstruct the initial map which fits the geometry of real industrial buildings. Additionally, we purposely show the reconstruction map after accumulating several frames from the initialization frame, which evidence the robustness of our method.

GPS limitation In some cases, there may be very large errors from regular GPS/IMU sensors. For example, the maximum rotation errors may be up to 45 degrees, and the maximum translation errors may be 40 meters. In our experiments, the most common errors are in the range of several meters in translation, and not more than 20 degrees in rotation. As shown in the experimental results, these errors will not impede our initialization method, and the vSLAM map can be successfully initialized. After successful initialization, the subsequent optimization will gradually rectify these errors.



(a) Sequence 07-08

(b) Sequence 02-21

(c) Sequence 12-21

Fig. 7 The reconstructed maps and trajectories by different methods. Left column: The red box denotes the reconstructed building façade which can align well to the real building by our initialization and optimization methods. The green one is the result from original ORBSLAM2 method without accurate scale and orientation. Right column: Our method estimates the real trajectory (red), while the original ORBSLAM2 method has no scale (green), and the sensor-based pose estimate exposes serious drift problems (magenta)

Because in our hybrid multimodality graph-based optimization, the constraints produced by the building blocks and the projection errors can guarantee the accuracy of the vSLAM, and the scale of the map can also keep correct. After initialization, if the system is failure due to some reasons (e.g., fast motion, dramatical light change), the system can be fast re-initialized once the data from GPS/IMU are acceptable as shown in Figs. 6 and 7.

6.3 Map comparison

In this section, we compare the map built by original ORBSLAM2 with the one reconstructed by ours. The results are shown in the left column of Fig. 7.

Because we only use monocular camera to capture images, there is no real scale information in the map built by the original monocular version of ORBSLAM2 (green). Therefore, it can be seen that the scale is entirely wrong. On the contrary, the maps built by the proposed method (blue and red) have the correct scales and can be aligned with real buildings. When we only use the initialization method to reconstruct the map (blue), although there are some misalignment errors in certain areas, it is obvious that the absolute scale is correct. After optimized by our multimodality method, the built map (red) aligns the real building models well. However, if large holds or architectural bulges are in the build façades, the feature points detected in these regions may not be closely aligned with the building models, as also shown in this figure.

6.4 Trajectory comparison

In this experiments, we compare the trajectories obtained by sensors (consumer-grade GPS and IMU), the original ORB-SLAM2, our initialization w/wo the optimization, respectively. We visualize these trajectories in the right column of Fig. 7 by differently colored points.

Like the map constructed by ORBSLAM2 in the last section, again the trajectory (green) computed by it has no absolute scale. The red trajectory obtained from sensors has a obvious difference from others, since there are not corresponding GPS positions associated with some keyframes. By comparison with trajectories obtained by the proposed methods, some keyframes positions in red one have large errors, which just mean the consumer-grade GPS and IMU is not reliable for robot localization due to the randomness and uncertainty in sensors.

6.5 Fixed-point position comparison

While we have no real ground-truth positions for all frames in the sequences, we measure a set of geodetically accurate positions. Therefore, the initial and final points of all sequences are deliberately chosen at these fixed points. Moreover, the robot also passed by most of these fixed points when collecting these sequences.

So we can compute the difference between the positions estimated by our method and the ground truth at these fixed points, through which quantitative results of the accuracy can be obtained. According to our experiments, the average Euclidean distance errors are about 24 centimeters, even when the trajectory is longer than 100 meters. This implies that our method only has about 0.24% errors on average. Therefore, the 3D building models and our hybrid optimization can indeed improve the accuracy of vSLAM.

7 Conclusion

In this paper, we have proposed an instant vSLAM initialization method for robot using building models to generate corresponding depth and a novel optimization framework for robot trajectory tracking and mapping by fusing multimodal information. Our method is validated in real-world scenes from the industrial environment.

The experiment results show that our method can quickly initialize or re-initialize the robot system with high accuracy and robustness and provide the robot with a global metric registration in the environment. Moreover, our optimization method further refines the robot trajectory and reconstructed map using multimodal information. By comparing our SLAM with the state-of-art SLAM methods, our method performs plausibly in terms of accuracy, robustness and computational effort, at the benefit of being relatively easy to implement.

While our method can be applied to the positioning and mapping of robots in the industrial environment, there exist some limitations in our method. For example, there is an objective number of structured building blocks in the common industrial environment; however, other objects in the environment, such as pedestrians, trees, and industrial equipment which are obvious not contained in the building depth masks will influence the effectiveness of the method.

Acknowledgements This work was partially supported by National Key R&D Program of China (2018YFB1305200). This publication was partially funded by the National Natural Science Foundation of China (61876167 and 61802348) and the Natural Science Foundation of Zhejiang Province (LY20F030017).

References

1. Arth, C., Pirchheim, C., Ventura, J., Schmalstieg, D., Lepetit, V.: Instant outdoor localization and slam initialization from 2.5 D maps. TVCG **21**(11), 1309–1318 (2015)
2. Bleser, G., Becker, M., Stricker, D.: Real-time vision-based tracking and reconstruction. J. Real-Time Image Process. **2**(2–3), 161–175 (2007)
3. Caselitz, T., Steder, B., Ruhnke, M., Burgard, W.: Monocular camera localization in 3d lidar maps. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1926–1931. IEEE (2016)
4. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. PAMI **40**(3), 611–625 (2017)
5. Ferreira, F., Veruggio, G., Caccia, M., Bruzzone, G.: A survey on real-time motion estimation techniques for underwater robots. J. Real-Time Image Process. **11**(4), 693–711 (2016)
6. Forster, C., Pizzoli, M., Scaramuzza, D.: SVO: fast semi-direct monocular visual odometry. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 15–22. IEEE (2014)
7. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2003)

8. Hol, J.D., Schön, T.B., Luinge, H., Slycke, P.J., Gustafsson, F.: Robust real-time tracking by fusing measurements from inertial and vision sensors. J. Real-Time Image Process. **2**(2–3), 149–160 (2007)
9. Huang, J., Liu, R., Zhang, J., Chen, S.: Fast initialization method for monocular slam based on indoor model. In: IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 2360–2365. IEEE (2017)
10. Kähler, O., Prisacariu, V.A., Murray, D.W.: Real-time large-scale dense 3D rec. with loop closure. In: ECCV, pp. 500–516 (2016)
11. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: ISMAR (2007)
12. Liu, R., Zhang, J., Chen, S., Arth, C.: Towards SLAM-based outdoor localization using poor GPS and 2.5 D building models. In: 2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Beijing, China, pp. 14–18 (2019)
13. Liu, R., Zhang, J., Yin, K., Wu, J., Lin, R., Chen, S.: Instant SLAM initialization for outdoor omnidirectional augmented reality. In: Proceedings of the 31st International Conference on Computer Animation and Social Agents, pp. 66–70. ACM (2018)
14. Loo, S.Y., Amiri, A.J., Mashohor, S., Tang, S.H., Zhang, H.: CNN-SVO: improving the mapping in semi-direct visual odometry using single-image depth prediction. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 5218–5223. IEEE (2019)
15. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular slam system. IEEE Trans. Robot. **31**(5), 1147–1163 (2015)
16. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras. IEEE Trans. Robot. **33**(5), 1255–1262 (2017)
17. Mur-Artal, R., Tardós, J.D.: Visual-inertial monocular slam with map reuse. IEEE Robot. Autom. Lett. **2**(2), 796–803 (2017)
18. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A.W.: Kinectfusion: real-time dense surface mapping and tracking. ISMAR **11**, 127–136 (2011)
19. Nistér, D.: An efficient solution to the five-point relative pose problem. PAMI **26**(6), 0756–777 (2004)
20. Qin, T., Li, P., Shen, S.: Vins-mono: a robust and versatile monocular visual-inertial state estimator. IEEE Trans. Robot. **34**(4), 1004–1020 (2018)
21. Von Stumberg, L., Usenko, V., Cremers, D.: Direct sparse visual-inertial odometry using dynamic marginalization. In: ICRA, pp. 2510–2517. IEEE (2018)
22. Wei, S., Ji, S., Lu, M.: Toward automatic building footprint delineation from aerial images using cnn and regularization. IEEE Trans. Geosci. Remote Sens. (2019)
23. Zhang, J., Liu, R., Yin, K., Wang, Z., Gui, M., Chen, S.: Intelligent collaborative localization among air-ground robots for industrial environment perception. IEEE Trans. Industr. Electron. **66**(12), 9673–9681 (2018)
24. Zhang, J., Singh, S.: Visual-lidar odometry and mapping: low-drift, robust, and fast. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 2174–2181. IEEE (2015)
25. Zhou, H., Zou, D., Pei, L., Ying, R., Liu, P., Yu, W.: StructSLAM: visual SLAM with building structure lines. IEEE Trans. Veh. Technol. **64**(4), 1364–1375 (2015)
26. Zhu, Z., Xu, F., Yan, C., Hao, X., Ji, X., Zhang, Y., Dai, Q.: Real-time indoor scene reconstruction with RGBD and inertial input. In: 2019 IEEE International Conference on Multimedia and Expo (ICME), pp. 7–12. IEEE (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ruyu Liu (Student Member, IEEE) received the B.S. degree in medical information engineering from Zhejiang Chinese Medical University, Zhejiang, China, in 2016. She is currently a PhD student with Institute of Computer Vision, College of Computer Science and Technology, Zhejiang University of Technology. Her research interests include robot localization, simultaneous localization and mapping (SLAM) and medical image processing.

Jianhua Zhang (Senior Member, IEEE) received the MSc degree at Zhejiang University of Technology in 2009 and the Ph.D. degree at the University of Hamburg in 2012. He was a research assistant at City University of Hongkong in 2008. Now he works with College of Computer Science, Zhejiang University of Technology. His research interests include SLAM, visual learning for autonomous robot, category discovery, object detection, image segmentation, medical image analysis. He is a member of the IEEE.

Shengyong Chen (IEEE M'01-SM'10) received the Ph.D. degree in robot vision from City University of Hong Kong in 2003. He is currently a Professor of Zhejiang University of Technology and Tianjin University of Technology, China. He received a fellowship from the Alexander von Humboldt Foundation of Germany and worked at University of Hamburg in 2006–2007. His research interests include computer vision, robotics, and image analysis. Dr. Chen is a Fellow of IET and senior members of IEEE and CCF. He has published over 100 scientific papers in international journals and he is an inventor of over 100 patents. He received the National Outstanding Youth Foundation Award of China in 2013.

Thomas Yang (Senior Member, IEEE), is the Professor of Electrical and Computer Engineering at Embry-Riddle Aeronautical University. His research interests include Adaptive/statistical signal processing, Wireless communications, Multi-agent systems, Pattern recognition.

Clemens Arth is senior scientist at Graz University of Technology and the deputy director of the CDL for Semantic Computer Vision. His main research interests are Computer Vision algorithms, Augmented Reality technology and Machine Learning. The current focus of his work is on accurate global localization of mobile devices for AR applications.