IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 11, 2019, accepted July 5, 2019, date of publication July 15, 2019, date of current version August 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2928578

Real-Time Direct Monocular SLAM With Learning-Based Confidence Estimation

WEIQI ZHANG[1], ZIFEI YAN[1], GANG XIAO[2], AIDI FENG[1], AND WANGMENG ZUO[1], (Senior Member, IEEE)
[1]Harbin Institute of Technology, Harbin 150001, China
[2]No.962 Hospital of PLA, Harbin 150000, China

Corresponding author: Zifei Yan (yanzifei@hit.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61671182 and Grant 61871381.

ABSTRACT Direct monocular simultaneous localization and mapping (SLAM) methods, for which the image intensity is used for tracking and mapping instead of sparse feature points, have gained in popularity in recent years. However, feature-based methods usually have more accurate camera localization results than most direct methods, though direct methods can work better in a textureless environment. To tackle the localization issue, we develop a novel real-time large-scale direct SLAM model, namely, GCP-SLAM, by integrating the learning-based confidence estimation into the depth fusion and motion tracking optimization. In GCP-SLAM, a random regression forest is trained off-line with pre-defined confidence measures for learning confidence and detecting the ground control points (GCPs). Then, the confidence value along with the selected GCPs is utilized for depth refinement and camera localization. Our proposed method is shown experimentally more reliable in tracking and relocalization than the previous state-of-the-art direct method when compared with feature-based and RGBD SLAMs.

INDEX TERMS Depth estimation, ground control points, motion tracking, random forest, SLAM.

I. INTRODUCTION

Monocular Simultaneous Localization and Mapping (SLAM) [1]–[3] has received consistent attention in robotics, augmented reality and autonomous cars in recent years. Considering of whether feature extractors are needed for the raw sensor measurement, the SLAMs can be devided into feature-based and direct ones. Compared with feature-based methods, direct methods estimate the localization of the sensor and the 3D map by directly using the image intensity information, which can avoid the local feature detection and perform more robust with camera-defocus, motion blur, and especially the textureless and structureless scenario.

Early SLAM methods focus on feature-based methods, where extracting and matching features are computationally expensive. Recent years has witnessed a surge in direct monocular SLAM methods for sparse and dense [4], [5] (semi-dense) tracking and reconstruction. Large-Scale Direct Monocular SLAM (LSD-SLAM) [5] is one representative method, which includes three major components: filtering-based semi-dense depth map estimation, direct image alignment based tracking, and incorporation into global map. Despite its versatility, robustness and flexibility, LSD-SLAM is obviously limited in camera localization accuracy. By far, the localization accuracy of LSD-SLAM is still lower than several state-of-the-art feature-based methods, such as Parallel Tracking and Mapping (PTAM) [6] and ORB-SLAM [7] where bundle adjustment (BA) [8] is adopted for motion optimization. BA is recently exploited in real-time visual odometry (VO) and feature based SLAM, refining a visual reconstruction through minimizing the reprojection error with respect to the measured image points. In contrast, it is computationally impracticable to adopt global and even local BA in LSD-SLAM, which performs localization by optimizing the photometric error based on the corresponding pixel intensities. Moreover, LSD-SLAM actually relies on those pixels with nonvanishing gradient even all pixel intensities are used in the estimation. Besides, scale drift often occurs and impacts the 3D reconstruction, which is a common problem in direct monocular methods.

In this work, we present a Ground Control Points based SLAM (GCP-SLAM) method to obtain accurate

The associate editor coordinating the review of this manuscript and approving it for publication was Jinjia Zhou.

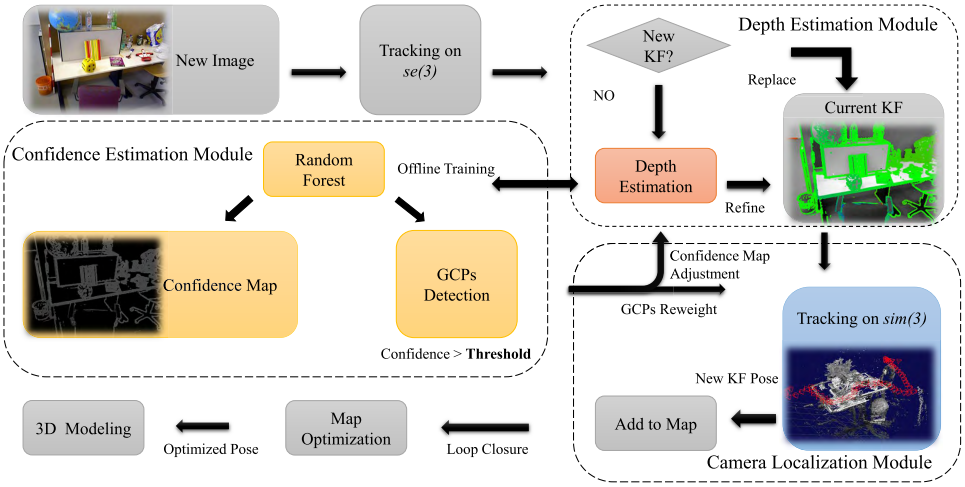


FIGURE 1. Overview of GCP-SLAM, including steps performed by depth estimation module, confidence estimation module and camera localization module.

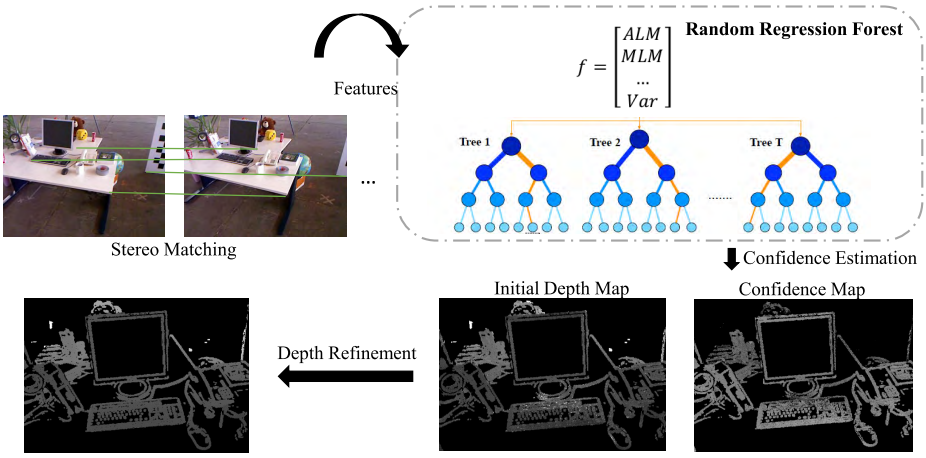


FIGURE 2. Outline of the confidence estimation module.

camera motion and tackle the problem of LSD-SLAM. Our GCP-SLAM is composed of three major modules, i.e., depth estimation module, confidence estimation module, and camera localization module, as illustrated in Fig. 1. In Fig. 2, the steps performed by confidence estimation module are presented, where random regression forests (RF) are utilized as a confidence prediction model. The uncertainty of depth originates from the noise not only on stereo matching, but initial camera pose, propagation and image intensity as illustrated [9]. To precisely model the confidence of depth estimation, besides stereo matching-based features, we also adopt image-based and depth-aware features. In the depth estimation module, we integrate the confidence result into depth refinement to avoid wrong fusion. For resisting the recently initialized and inaccurate depth estimates in motion tracking, LSD-SLAM adds the variance as an additional weighting term in the weighted least squared function, which is one possible factor to impact the camera localization accuracy. In our GCP-SLAM, we define the inaccurate depth estimation based on the relative confidence predicted by the confidence estimation model, and pixels with high confidence value are chosen as the ground control points (GCPs) according to the prediction. In camera localization module, we increase the weight of GCPs and alleviate the adverse effect caused by unreliable estimation for improving the localization accuracy. The experimental results on TUM RGB-D datasets show that our GCP-SLAM performs reliably in tracking and relocalization compared with LSD-SLAM in comparable time. The contribution of this work is three-fold:

- A random regression forest is applied to confidence estimation for small baseline stereo matching in direct monocular SLAM, where the predefined matching cost-based, image-based and depth-aware features are exploited for confidence modeling. We incorporate the confidence value with depth filter and improve the depth estimation based on the Extended Kalman Filter (EKF).

- By choosing a set of pixels with high confidence as GCPs, an improved camera localization method based on minimizing the photometric and geometric error was provided by increasing the weight term of GCPs in direct image alignment on $\mathfrak{sim}(3)$.
- We incorporate the depth refinement and GCPs based camera localization into the LSD-SLAM framework, and achieve accurate and reliable performance in both motion tracking and 3D reconstruction.

The paper is organized as follows. In Section II, we present a brief overview of direct and indirect monocular SLAM methods, and introduce learning based confidence estimation methods in stereo matching and camera localization. Section III provides our confidence estimation model and further introduces the depth refinement and camera localization based on the confidence value. In Section IV, we evaluate the proposed GCP-SLAM on TUM RGB-D datasets and compare with LSD-SLAM, state-of-the-art feature-based and RGBD SLAMs. Finally, Section V gives conclusion and discussion of our method based on the experimental results.

II. RELATED WORKS

This section begins with a brief survey on indirect (feature based) and direct methods for monocular SLAMs. Then, we further review the related work on confidence measure in stereo matching and camera relocalization.

A. FEATURE BASED MONOCULAR SLAMS

Earlier monocular SLAMs mostly focus on the sparse interest points, one representative method of which is proposed by Chiuso *et al.* [10] following the casual scheme of Structure form Motion (SfM). Simple gradient descent feature tracking was used in their method, which makes it unable to match features during high acceleration and only suitable for small scenes with small camera motions. MonoSLAM presented by Davison *et al.* [11] creates a sparse map of landmarks under a probabilistic framework using a single Extended Kalmanm Filter, performing as a real-time system. It is able to cope with faster motion and larger scenes than method proposed by Chiuso. PTAM [6] is the first real-time monocular SLAM method which separated localization and mapping into two threads. It performs pose update by minimizing the geometric error and use BA for further optimization, which brings restriction to small scenes. The state-of-the-art feature-based monocular SLAM method proposed by Mur-Artal *et al.* [7], i.e., ORB-SLAM, can tackle the large scene problem. It is based on the ORiented Brief (ORB) features represented by Bags of Words (BoW) model which benefits place recognition for loop detection. Covisibility graph are suggested in ORB-SLAM to improve the robustness of camera localization, and local BA is adopted to discard outlier observations. ORB-SLAM2 [12] extends the method for stereo and RGB-D cameras and achieves state-of-the-art accuracy on 29 popular public sequences.

B. DIRECT MONOCULAR SLAMS

Recently, monocular SLAMs have witnessed a shift towards direct SLAM methods [5], [13]. Without detection of sparse interest points, direct methods use intensities for camera localization, which are robust for textureless scenes, and show impressive dense (semi-dense) scene reconstructions. For dense monocular SLAMs, Newcombe *et al.* [13] developed a Dense Tracking and Mapping (DTAM) system for real-time tracking and reconstruction of small-scale scenarios. Besides, several dense direct methods have also been proposed [14]–[16]. Engel *et al.* [5] suggested an algorithm for building real-time semi-dense maps of large scale scenes called LSD-SLAM, where OpenFABMAP [17] and pose-graph optimization are utilized for loop closure and optimizing the camera poses, respectively. Multiple-level mapping (MLM) [4] extends LSD-SLAM using a dense approach to increase the density and improves the tracking performance. Similarly, CNN-SLAM [18] deployed predicted depth map from a deep neural network and fused with depth measurements in the direct monocular SLAM, i.e. LSD-SLAM, for accurate and dense reconstruction. While feature-based methods can perform BA to optimize camera localization, pure direct methods generally track camera pose by minimizing the photometric error based on pixel intensities, and cannot achieve state-of-the-art localization performance.

Different from the above direct SLAM methods, Semi-direct Visual Odometry (SVO) [19] estimates camera pose by minimizing the photometric error over pixel intensity patches. The 3D points initialized by feature extraction are estimated using probabilistic model and refined using BA together with the pose. However, the feature correspondence is an implicit result of direct motion estimation. Direct sparse odometry (DSO) [20] jointly optimizes the camera motion, affine brightness parameters, and inverse depth which initialized as LSD-SLAM in a direct probabilistic model, which minimizes photometric error over a window of recent frames. It samples pixels across all image regions that have intensity gradient, which is more sparse than LSD-SLAM, and performs the photometric equivalent of windowed sparse bundle adjustment, showing great outperformance compared with state-of-the-art direct and indirect methods in terms of tracking accuracy and robustness.

C. LEARNING BASED CONFIDENCE ESTIMATION

Confidence estimation based on learning methods have been adopted in stereo matching [21]–[24] and camera relocalization [25]. An earlier method proposed by Spyropoulos *et al.* [21] used a RF trained to predict the correctness of stereo correspondence, and the chosen GCPs are exploited for further optimization using Markov Random Field (MRF) to minimize energy function. Park and Yoon [22] developed two regression forests, where one is used for selecting more important features and the other is adopted to predict the confidence of stereo matching.

(a) (b) (c)

FIGURE 3. Image of Fastmotion dataset (a) without blur or noise effects, (b) with motion blur and (c) with motion blur and low lighting levels.

The prediction results are utilized to adjust the matching cost after rescaling, which benefits the matching procedure. Different features are exploited, e.g., disparity-based features [23] that can be computed in O(1) computation and superpixel-level features [24]. More recently, there has been some stereo confidence estimation methods based on features extracted by deep networks [26]. For camera relocalization, learning predictions are often used to guide the camera pose optimization procedure. Instead of only pointing to candidate locations, the uncertainty associated with each estimation [25] is modeled as mixtures of anisotropic 3D Gaussians using regression forests. Building on the success of confidence estimation based on the random regression forest [21], [22], [25], in this paper, we introduce the learning-based confidence estimation method into monocular SLAM framework for improving the depth estimation and camera localization.

III. MONOCULAR SLAM WITH CONFIDENCE ESTIMATION

A. METHOD OVERVIEW

GCP-SLAM continuously tracks camera poses, predicts semi-dense inverse depth map for each keyframe and maintains a pose graph following the framework of LSD-SLAM. After the process of depth estimation, we predict the confidence and obtain a confidence map for current keyframe. In camera localization, the selected GCPs are exploited for minimizing the photometric error. The major modules of our framework are summarized as follows:

i) **Depth estimation module**: When a new keyframe I_i is created from the most recently tracked image identified by the relative distance and angle to the current keyframe, a semi-dense inverse depth map D_i is initialized by stereo matching with a reference non-keyframe or propagated from the depth map of existing keyframe. The depth map is then enhanced by a set of non-keyframes in the reference frame list according to the estimated confidence map U_i before identifying the next keyframe.

ii) **Confidence estimation module**: We train a random forest regression model off-line with features corresponding to stereo matching, images and depth for predicting the confidence of depth estimation. While obtaining the depth through stereo matching, we create and update the confidence map U_i of ith keyframe using the confidence estimation model and detect GCPs according to the confidence value.

iii) **Camera localization module**: The rigid-body pose $\boldsymbol{\xi} \in \mathfrak{se}(3)$ of a newly captured camera frame relative to the current keyframe is tracked continuously. The GCPs are incorporated with optimization in the objective function, which performs alignment on $\mathfrak{sim}(3)$ with two differently scaled keyframes. Then the pose graph and the reconstruction results of keyframes are maintained.

Apart from the major modules, we continuously perform pose-graph optimization between all keyframes and generate an accumulated semi-dense pointclouds based on the depth map and camera pose of keyframes after each optimization. Large-scale loop closure is detected after adding new keyframe and a reciprocal tracking is also applied to avoid inserting false loop closure.

B. CONFIDENCE ESTIMATION

In LSD-SLAM, the small baseline stereo matching is utilized to initialize and update depth on pixels with large gradient of keyframes (here depth means the inverse depth [27]). Depth estimation uncertainty is one important factor that impacts on tracking procedure, and causes drift in mapping. Considering that the learning based methods are successfully applied in stereo matching to adjust the matching cost and refine the disparity map, we propose a confidence estimation model to discriminate whether the creation or the update of depth is reasonable and reliable.

A regression type of random forest is constructed for precise confidence prediction for depth estimation as illustrated in [21]–[24]. For training our confidence estimation model, we utilize some of the features mentioned in [21], [22] and other features calculated from the image and depth map as the confidence measure. The features should be computationally efficient due to the real-time requirement, thus we exclude the left-right difference (LRD) because it needs stereo matching procedure that costs time, and some other features, e.g., the distance from discontinuity (DD), which are unable to calculate due to the discontinuity of pixels with depth. Note that [22] trained the model by selecting the most important eight features and shown that the eight measures

can accurately predict the unreliable estimation. Here we use features $f = (f_1, f_2, \cdots, f_{16})$, where $f_1, f_2, \cdots, f_8$ are calculated from stereo matching cost, defined as $c(p, d)$. p denotes the pixel in the left image of stereo matching, and p is omitted, e.g., $c(d)$ if the position information is unnecessary. We use d as the depth value corresponding to the matching cost and c_k as the kth minimum matching cost.

Minimum Cost (MC) [21]. Minimum matching cost (c_1) in line stereo matching. We use the square root of cost value while stereo matching is in square form,

$$f_{MC}(p) = -c_1 \tag{1}$$

Maximum Margin (MMN) [21]. The difference between the two smallest cost values (c_1 and c_2) of a pixel,

$$f_{MMN}(p) = c_2 - c_1 \tag{2}$$

Peak Ratio(PKR) [22]. The peak ratio of the matching cost is defined as,

$$f_{PKR}(p) = \frac{c_2}{c_1} \tag{3}$$

Winner Margin (WMN) [22]. Normalization of the difference between the two smallest cost values,

$$f_{WMN}(p) = \frac{c_2 - c_1}{\sum_d c(d)} \tag{4}$$

Attainable Maximum Likelihood (AML) [21]. This feature can be obtained by first subtracting the minimum cost from all cost values, and then converting the cost curve into a probability density function.

$$f_{AML}(p) = \frac{1}{\sum exp(-\frac{(c_2 - c_1)}{\sigma^2_{AML}})} \tag{5}$$

The Maximum Likelihood Measure (MLM) [22]. This feature is also obtained by converting the cost curve into a probability density function for disparity. Instead of subtracting the minimum cost, we assume that the cost follows a normal distribution and the disparity prior is uniform.

$$f_{MLM}(p) = \frac{exp(-\frac{c_1}{\sigma^2_{MLM}})}{\sum_d exp(-\frac{c(d)}{\sigma^2_{MLM}})} \tag{6}$$

The Negative Entropy Measure (NEM) [22]. The negative entropy is defined as,

$$p(d) = \frac{exp(-c_1)}{\sum_d exp(-c_d)} \tag{7}$$

$$f_{NEM}(p) = \sum_d p(d) \log p(d) \tag{8}$$

The Perturbation Measure (PER) [22]. The perturbation measure is defined as an exponential function,

$$f_{PER}(p) = \sum exp(-\frac{c_2 - c_1}{\sigma^2_{PER}}) \tag{9}$$

The geometric error and photometric error are used as f_9 and f_{10}. The geometric error σ_{geo} is the error of depth estimation caused by noise on estimated pose and projection, while the photometric error σ_{pho} encodes how the image intensity errors have effect on the estimated depth. f_{11} is the length of searched epipolar line segment, which is decided by the depth hypothesis. The observation variance of depth is used as f_{12}, which can be calculated by [9],

$$f_{VAR}(p) = \alpha^2(\sigma^2_{geo} + \sigma^2_{pho}) \tag{10}$$

where α is the proportionality constant. Features $f_{13} - f_{16}$ are calculated from the information of depth and the image. The difference between new observation and current estimation of depth value is used as f_{13}. The gradient of pixel in color image is used for f_{14}. f_{15} is the distance of a pixel from its nearest image border. f_{16} is the depth estimation of the pixel itself. With defined features $\boldsymbol{f} = (f_1, f_2, \cdots, f_{16})$, we train a random regression forest for confidence estimation.

After training, the prediction results of this confidence estimation model can serve as confidence for pixels with depth in keyframe, which can be employed to refine depth fusion and enhance the camera localization. The importance of these features are evaluated in section IV.

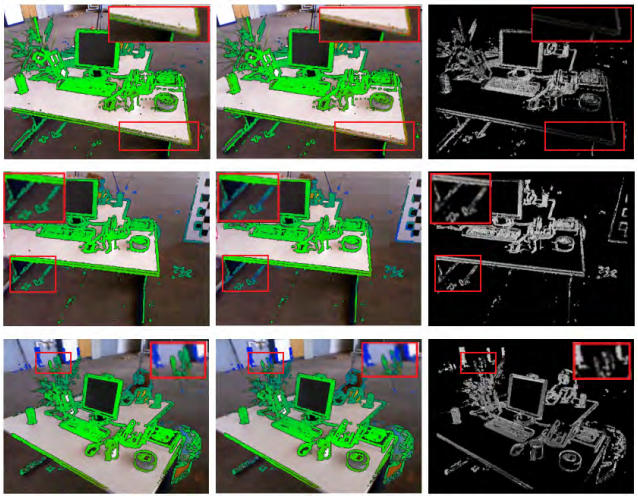
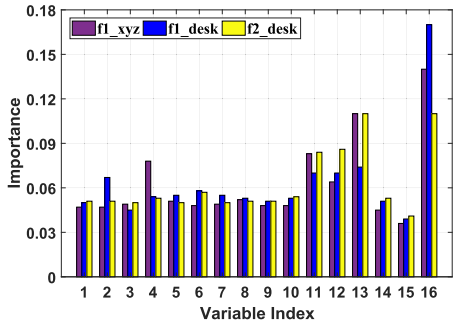


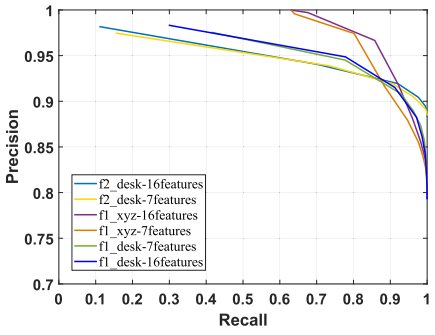
FIGURE 4. Confidence estimation of several frames in *fr2_desk* sequence of TUM RGB-D datasets. Left: Color-coded ground-truth semi-dense depth map. Middle: Color-coded semi-dense depth map estimated by LSD-SLAM. Right: Predicted confidence map projected to 0-255.

C. CONFIDENCE BASED DEPTH REFINEMENT

The goal of this stage is to continuously refine the depth map of the active keyframe based on the confidence results, which measures how coherent each predicted depth value across different estimation. If the confidence value is high, the estimated depth will be considered in fusion. Otherwise, it will be discarded. With this procedure, the precision of depth is effectively improved. After a depth observation of a pixel p in the current keyframe has been obtained, we initialize the depth value $D_0(p)$ directly with the observation if no prior hypothesis for the pixel exists. The corresponding depth variance $V_0(p)$ is initialized with σ_p^2, while the depth



(a)



(b)

FIGURE 5. (a) The importance of each confidence measure used in training the confidence estimation model on TUM RGB-D datasets. (b) The PR curves for confidence estimation model with different number of features on *fr2_desk*, *fr1_desk* and *fr1_xyz* sequences.

confidence $U_0(p)$ is given by confidence prediction result. The creation will be improved during the update so we try to obtain more available depth as we can, but strict rule is applied for the procedure to enhance the depth which will be discussed in experiment part.

New observation d_p with relative confidence value u_p, which is beyond a certain threshold, are incorporated into the prior $D_{k-1}(p)$ and $U_{k-1}(p)$ by multiplying two distribution, corresponding to the update step in the Extended Kalman Filter,

$$D_k(p) = \frac{\sigma_p^2 D_{k-1}(p) + V_{k-1}(p)\, d_p}{\sigma_p^2 + V_{k-1}(p)} \tag{11}$$

$$U_k(p) = \frac{\sigma_p^2 U_{k-1}(p) + V_{k-1}(p)\, u_p}{\sigma_p^2 + V_{k-1}(p)} \tag{12}$$

where σ_p^2 and $V_{k-1}(p)$ denote the observation variance and the prior variance of inverse depth, respectively. The variance [9] of $D_k(p)$ is given by $V_k(p) = \frac{\sigma_p^2 V_{k-1}(p)}{V_{k-1}(p)+\sigma_p^2}$. Except for the depth fusion of active keyframe, we propagate the refined estimated depth from frame to frame with predicted confidence value. The corresponding 3D point is calculated based on the depth $D(p)$ estimated for pixel p on the current keyframe and then projected into the pixel p' in new frame. If there exists valid estimation, we integrate the depth and

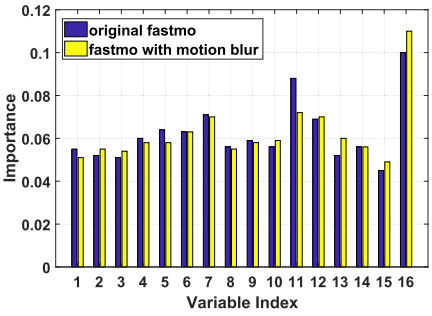


FIGURE 6. The importance of each confidence measure used in training the confidence estimation model on original fastmotion sequence and sequence with motion blur.

confidence value as follows,

$$D_k(p') = \frac{\sigma_{p'}^2 D_{k-1}(p') + V_{k-1}(p')\, d_{p'}}{\sigma_{p'}^2 + V_{k-1}(p')} \tag{13}$$

$$U_k(p') = \frac{\sigma_{p'}^2 U_{k-1}(p') + V_{k-1}(p')\, u_{p'}}{\sigma_{p'}^2 + V_{k-1}(p')} \tag{14}$$

The propagated depth is approximated by $d_{p'} = \left(D(p)^{-1} - t_z\right)^{-1}$, where t_z is the camera translation along the optical axis. The propagated confidence $u_{p'}$ and the variance $\sigma_{p'}$ is given by,

$$u_{p'} = \left(\frac{d_{p'}}{D(p)}\right)^2 U(p) \tag{15}$$

$$\sigma_{p'}^2 = \left(\frac{d_{p'}}{D(p)}\right)^4 V(p) + \sigma_n^2 \tag{16}$$

σ_n^2 is the white noise variance used to increase the propagated uncertainty.

D. $\mathfrak{sim}(3)$ TRACKING BASED ON GCPs

In direct SLAMs such as LSD-SLAM, camera motion is computed using two image alignment by solving an iteratively re-weighted least-squares problem, in which a weighted matrix is suggested to down-weight the large residuals in each iteration for robustness. We propose to up-weight the pixels of high confidence, i.e., GCPs detected by our confidence estimation model, for benefiting the tracking process.

A minimal representation for the camera pose $\boldsymbol{\xi}$ is given by elements of associated Lie-algebra $\mathfrak{se}(3)$ and $\mathfrak{sim}(3)$. To overcome the scale-drift, we use $\mathfrak{sim}(3)$ to estimate not only the motion but also the scale difference between depth. LSD-SLAM proposed a novel method to do image alignment on $\mathfrak{sim}(3)$ for two differently scaled keyframes, where the depth residual r_d is incorporated that penalizes deviations in inverse depth with the photometric residual r_p. The overall objective function minimizing the variance-normalized photometric and geometric error is defined as follows:

$$E(\boldsymbol{\xi}_{ji}) = \sum_{p \in \Omega_{D_i}} \left\| \frac{r_p^2(p, \boldsymbol{\xi}_{ji})}{\sigma^2_{r_p(p,\boldsymbol{\xi}_{ji})}} + \frac{r_d^2(p, \boldsymbol{\xi}_{ji})}{\sigma^2_{r_d(p,\boldsymbol{\xi}_{ji})}} \right\|_\delta \tag{17}$$

where $\boldsymbol{\xi}_{ji} \in \mathfrak{sim}(3)$ denotes the transformation moving a point from frame i to frame j, and $\|\cdot\|_\delta$ denotes the Huber norm. We denote pixel coordinates by $p = (p_x, p_y, 1)^T$, and project it into the 3D position with inverse depth d through the mapping $\pi^{-1}(p, d) = \left(\left(d^{-1}K^{-1}p\right)^T, 1\right)^T$. The pixel p in the ith frame is transformed to the pixel p' in the jth frame,

$$p' = \pi\left(\exp_{\mathfrak{sim}(3)}\left(\boldsymbol{\xi}_{ji}\right)\pi^{-1}\left(p, D_i\left(p\right)\right)\right) \tag{18}$$

the corresponding depth d' of the pixel p' is,

$$d' = \left[\exp_{\mathfrak{sim}(3)}\left(\boldsymbol{\xi}_{ji}\right)\pi^{-1}\left(p, D_i\left(p\right)\right)\right]_3^{-1} \tag{19}$$

where $\pi(\cdot)$ projects a 3D position into the image plane. The photometric residual and the geometric residual are defined as, $r_p\left(p, \boldsymbol{\xi}_{ji}\right) = I_i\left(p\right) - I_j\left(p'\right)$ and $r_d\left(p, \boldsymbol{\xi}_{ji}\right) = d' - D_j\left(p'\right)$, respectively. The variance $\sigma_{r_p}^2$ and $\sigma_{r_d}^2$ are computed as,

$$\sigma_{r_p\left(p,\boldsymbol{\xi}_{ji}\right)}^2 = 2\sigma_I^2 + \left(\frac{\partial r_p\left(p, \boldsymbol{\xi}_{ji}\right)}{\partial D_i\left(p\right)}\right)^2 V_i\left(p\right) \tag{20}$$

$$\sigma_{r_d\left(p,\boldsymbol{\xi}_{ji}\right)}^2 = \left(\frac{\partial r_d\left(p, \boldsymbol{\xi}_{ji}\right)}{\partial D_j\left(p'\right)}\right)^2 V_j\left(p'\right) + \left(\frac{\partial r_d\left(p, \boldsymbol{\xi}_{ji}\right)}{\partial D_i\left(p\right)}\right)^2 V_i\left(p\right) \tag{21}$$

$V_i(p)$ denotes the depth variance of pixel p in the ith keyframe. σ_I^2 is the white noise variance. Considering what mentioned above, we up-weight the residuals of pixels which are chosen as GCPs for $\mathfrak{sim}(3)$ relative pose estimation. The problem in Eqn. (17) is minimized using the re-weighted Gauss-Newton algorithm in a left-compositional formulation: In each iteration, a left-multiplied increment is computed by solving the minimum of a second-order approximation of E,

$$\delta\boldsymbol{\xi}^n = -\left(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{J}\right)^{-1}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{r} \tag{22}$$

where $\boldsymbol{r}$ is the stacked residual vector and $\boldsymbol{J}$ is the Jacobian of the stacked residual vector,

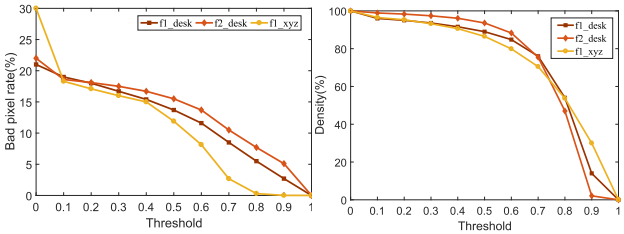
$$\boldsymbol{J} = \left.\frac{\partial r\left(\epsilon \circ \boldsymbol{\xi}^n\right)}{\partial \epsilon}\right|_{\epsilon=0} \tag{23}$$

and $\boldsymbol{W}$ is the diagonal weighted matrix with,

$$\boldsymbol{W}_{ii} = f(\sigma_{r_{p_i}}, \sigma_{r_{d_i}}) \tag{24}$$

According to the prediction results of our confidence estimation model, we set a threshold δ to identify GCPs including the smallest fraction of pixels with wrong depth estimation. There is a trade-off between density and accuracy, thus we set δ based on experiments which will be discussed in Section V. In our GCP-SLAM, we down-weight residuals of pixels with high uncertainty and up-weight the residuals of other pixels with high confidence by $\boldsymbol{W}_{ii} = \boldsymbol{W}_{ii} * (1 - \delta + u_i)$.

To sum up, by integrating GCPs into $\mathfrak{sim}(3)$ tracking, pixels with inaccurate estimation are resisted and better camera localization accuracy can be attained. Note that the predefined confidence measures are computationally efficient and confidence estimation is performed only on pixels with large gradient in keyframes. With this consideration, we can build our improved direct SLAM system in real time on CPU.



(a) (b)

FIGURE 7. (a) Sparsification curves for selected sequences. (b) GCPs density versus threshold on selected sequences.

IV. EXPERIMENTS

In this section, we first evaluate the performance of the random regression forest as confidence estimation model, which shows great efficiency for confidence prediction with various types of data in stereo matching. We evaluate our GCP-SLAM in terms of depth refinement and motion tracking on the TUM RGB-D datasets [28], which contains several long and challenging trajectories with camera rotation, motion blur and rolling shutter artifacts. All the experiments are executed in a PC with eight Intel Cores Xeon E3-1230 V2 CPU (3.3 GHz) and 32 GB RAM. Our GCP-SLAM can be run in real time on CPU for these datasets.

A. EVALUATION ON CONFIDENCE ESTIMATION

We use about 10 pairs of frames (keyframes with reference frames) from each of the sequence *fr*2*_desk*, *fr*1*_desk* and *fr*1*_xyz* of TUM RGB-D datasets, with predefined features to train the confidence estimation model. *fr*1*_xyz* is a simple sequence which only contains translatory motions along the principal axes of Kinect, and keeps the orientation mostly fixed. *fr*1*_desk* contains several sweeps over four desks in a typical office environment with motion blur. *fr*2*_desk* records an office scene with camera defocus, including lots of objects such as desks, a computer monitor, keyboard, etc. We also evaluate our confidence estimation model on simulated sequences from [29], namely fastmotion (fastmo), which generate indoor scenes with fast motion and are augmented with effects simulating the motion blur and noise. The original image with no blur or noise effects, image with motion blur and lower lighting level are shown in Fig. 3. Note that the associated planar depth map are also generated.

We adopt the 16 features introduced in Section III including 8 matching cost-based features, 8 image based and depth-aware features, and assign the label y to each sample according to ground truth depth map. Specifically, the code from the website of the TUM RGB-D dataset is exploited to align color images and the depth images. The difference

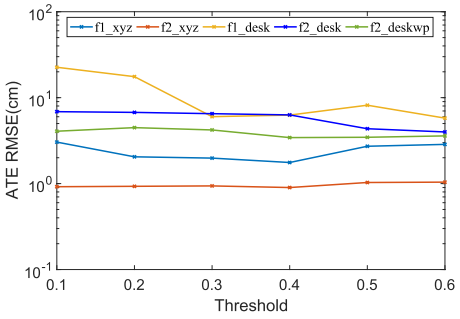


FIGURE 8. The ATE RMSE (cm) using different depth threshold on several sequences.

between estimated depth d_e and ground truth d_{gt} is mapped to the value $y \in [0, 1]$,

$$y = \begin{cases} 1 - |d_{gt} - d_e|/\tau_{err}, & if\ |d_{gt} - d_e| < \tau_{err} \\ 0, & otherwise \end{cases} \quad (25)$$

Fig. 5 (a) shows the importance of each confidence measure for confidence estimation training on the three TUM RGB-D sequences.

One can see that the length of epipolar line, the variance of depth, difference of observation and current estimation of depth and the depth value are more essential for 3 datasets. Moreover, the matching cost based feature MMN and depth value are important for *f*1_*desk*, WMN and difference of new depth observation and current estimation are important for *f*1_*xyz*, while MLM and the depth variance are important for *f*2_*desk*. Besides, other matching cost based features have almost similar contributions to confidence estimation. The importance of the proposed confidence measures show less difference on original fastmotion sequence and the same sequence with motion blur in Fig. 6, among which NEM, length of epipolar line, variance of depth and depth value are also essential. Some statistic features, e.g., AML, are more important for fastmotion datasets, as well as length of epipolar. In contrast, MMN, difference of the observed depth and the current depth estimation, and the depth value are more important for fastmotion with motion blur. Note that the sequence with both motion blur and low lighting level fails on tracking so we cannot train on that sequence. From Fig. 5 (a) and Fig. 6, we know that the importance of confidence measures for training confidence estimation model is affected by some outliers in depth observation related with blur. It seems reasonable that longer length of epipolar line, more stable depth estimation, smaller depth variance bring more accurate depth estimation. The stereo matching cost-based features, e.g., MLM, NEM and WMN play important roles in detecting unreliable depth from raw matching cost but the importance varies according to the images. We also report the precision-recall curves for *fr*2_*desk*, *fr*1_*desk* and *fr*1_*xyz* sequences training and testing with all the proposed features and 7 selected features $\boldsymbol{f}_{selected} = (f_2, f_4, f_6, f_{11}, f_{12}, f_{13}, f_{16})$. However, the RF model show poorer performance by using 7 features on *f*1_*xyz* and *f*2_*desk* in Fig. 5 (b). Without loss of generality, we use all the proposed 16 features to train our confidence prediction model.

TABLE 1. Prediction results of GCPs and non-GCPs using our confidence estimation model. We set $\delta = 0.7$ to balance the density and accuracy of GCPs.

	GCPs		non-GCPs	
Datasets	$y > 0.7$	$y \leq 0.7$	$y \leq 0.7$	$y > 0.7$
fr1_xyz	232,167	6,430	65,118	35,300
fr1_desk	449,550	43,320	98,948	55,663
fr2_desk	552,717	52,390	128,558	76,971
fr2_deskwp	1,067,832	99,947	690,707	146,194
fr3_sitxyz	800,874	64,960	259,015	109,645
fr3_sithalf	568,735	68,589	131,323	40,591
Average Precision	92.15%			

To analyze the performance of various confidence threshold, we report the sparsification curve and density curve in Fig. 7(a) and Fig. 7(b), respectively. The sparsification curve draws the change of bad pixel rates while removing least confident pixels from the estimated depth map according to the confidence prediction. The density curve draws the density of GCPs under different confidence threshold for the keyframe. We choose keyframes excluding the training ones from the three sequences as test sets. We can see that when choosing $\delta = 0.7$ it obtains the highest possible density (about 70%) of GCPs while excludes most (about 90%) wrong depth estimation.

By selecting the best confidence estimation model after training, we test on pairs of images on the sequence *fr*1_*desk*, *fr*2_*desk*, *fr*1_*xyz*, *fr*3_*sitting_xyz*, *fr*3_*sitting_halfsphere* and *fr*2_*desk_with_person*, among which the last three sequences contain dynamic objects. According to the results in Fig. 7(a) and Fig. 7(b), the pixel can be selected as a GCP when the confidence prediction $y > 0.7$. Table 1 reports the prediction results of our regression model on the test set. The first and third column correspond to accurate prediction, while the second and fourth to inaccurate prediction of pixels with reliable estimation (GCPs) and unreliable estimation, respectively. We show raw pixel numbers for these sequences, and the average prediction result of GCPs in the last row. We can see from Table 1 that we achieve 92.15% precision using our confidence estimation model. Fig. 4 shows the ground-truth of depth, the estimated depth map in LSD-SLAM and the corresponding confidence map evaluated by the proposed confidence estimation model of keyframes in sequence *fr*2_*desk*. We can see that the confidence value is visibly darker for unreliable pixels, which contribute little to depth fusion and motion tracking.

Table 2 shows the effect of the number of trees N used in RF on the prediction results and run time for confidence estimation. It demonstrates that the number of trees N is linearly correlated with the run time for prediction, and has little impact on the prediction results when N is higher than 30. Therefore, we set $N = 30$ to train our model for the tradeoff between run time and prediction accuracy.

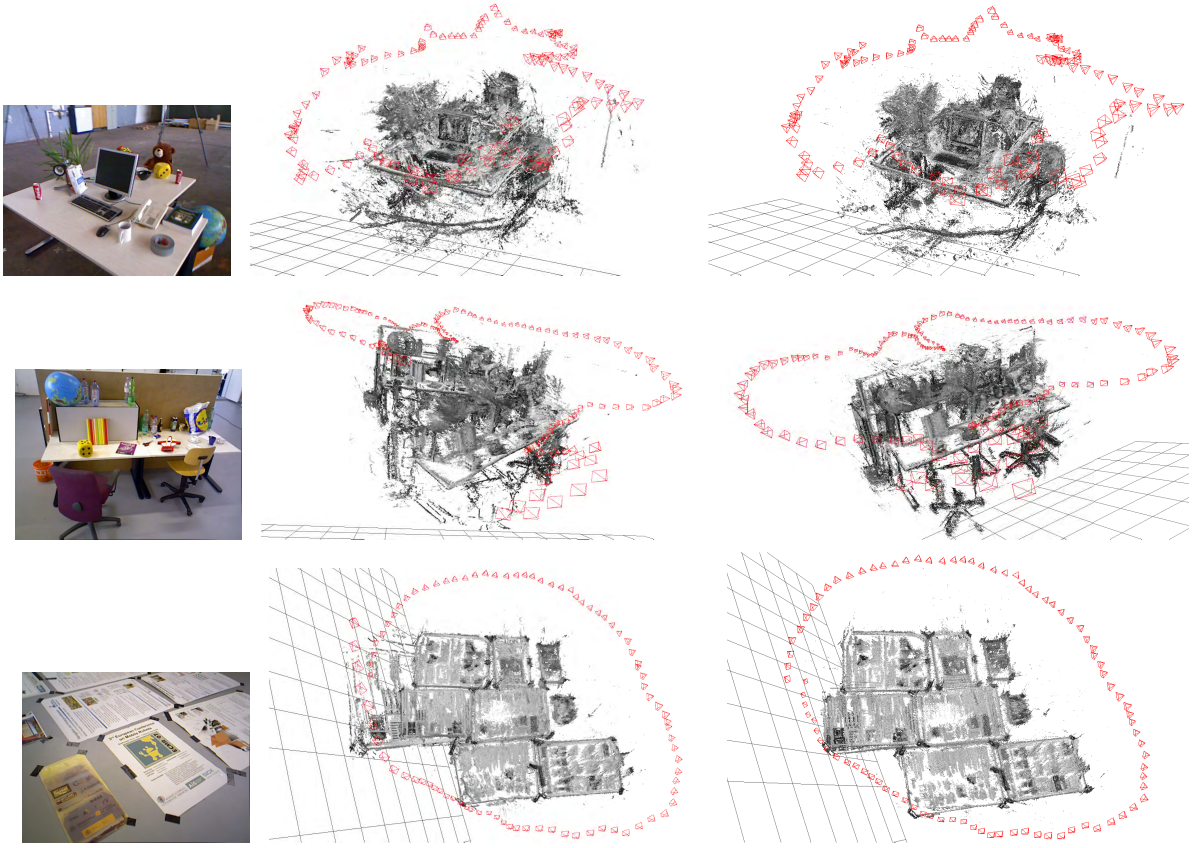


FIGURE 9. Accumulated pointclouds of keyframes using LSD-SLAM and GCP-SLAM. Camera frustums are displayed for keyframes with size corresponding to the scale. (a) Images of sequence *fr2_desk*, *fr3_long_office_household* and *fr3_nostructure_texture_near*. (b) Reconstruction results of LSD-SLAM. (c) Reconstruction results of GCP-SLAM.

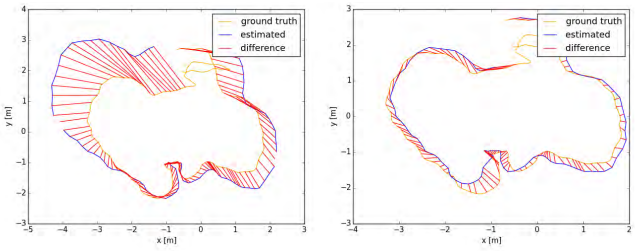


FIGURE 10. Trajectories of *fr3_long_office_household* sequence from the TUM RGB-D dataset generated by LSD-SLAM. Left: the worst keyframe trajectories over 5 executions. Right: the best keyframe trajectories over 5 executions.

TABLE 2. The prediction time (per frame) and results for different number of trees on *f1_xyz* sequence.

# of trees	20	25	30	35	40
Times (ms)	132.0	180.5	241.4	291.8	376.9
Precision (%)	97.30	97.31	97.33	97.34	97.35
Recall (%)	86.59	86.55	86.49	86.45	86.46

B. EVALUATION ON SEMI-DENSE GCP-SLAM

We evaluate our semi-dense GCP-SLAM on the TUM RGB-D datasets with confidence based depth refinement, GCPs-based $\mathfrak{sim}(3)$ tracking, and both of them respectively. The sequences we used including *fr1_xyz*, *fr1_desk*, *fr2_xyz*, *fr1_floor*, *fr2_desk*, *fr2_desk_with_person*, *fr3_long_office_household*, *fr3_sitting_xyz* and *fr3_nostructure_texture_far*. For localization evaluation, we show camera pose accuracy results based on the absolute trajectory error (ATE) RMSE (cm) between the estimated camera translation and the ground-truth. The keyframe trajectories are aligned using a similarity transformation [31] for all methods despite the scale.

We first evaluate our method with confidence based depth refinement under different threshold. The ATE RMSE (cm) with different updating threshold on five sequences of the TUM RGB-D datasets are shown in Fig. 8. According to the experimental results, we discover that setting the threshold to 0.4 for updating the depth stabilizes the performance of motion tracking, though setting larger threshold brings less error on *f2_desk* sequence. Thus we set the depth refinement threshold to 0.4 for the comparison experiment. The unreliable depth shown in Fig. 4 are discarded in this step.

We show comparative results of our method, direct methods including LSD-SLAM and direct dense visual

TABLE 3. Keyframe localization error on the TUM RGB-D benchmark, measured as absolute trajectory RMSE(cm). Results of ORB-SLAM, PTAM and PL-SLAM are extracted from [7] and [30], respectively. Trajectories of RGBD-SLAM are taken from the benchmark website which only available for *fr*1 and *fr*2 sequences. '-' denotes no available data. 'X' denotes tracking failure on that sequence. '*' denotes ambiguity detected on that sequence.

Datasets (# of keyframes)	GCP-SLAM			LSD-SLAM	DVO-SLAM	RGBD-SLAM	PTAM	ORB-SLAM	PL-SLAM
	depth refinement	motion tracking	both						
fr1_xyz (38)	1.7	2.3	**1.6**	6.0	1.16	1.34	1.15	0.9	1.21
fr1_desk (70)	6.3	13.7	**6.1**	39.2	2.10	2.58	X	1.69	-
fr1_floor (100)	39.4	31.9	**27.1**	34.2	5.50	9.00	X	2.99	7.59
fr2_desk (120)	**4.0**	6.1	4.6	6.9	1.70	9.50	X	0.88	-
fr2_xyz (38)	**0.9**	1.0	**0.9**	1.23	1.18	2.61	0.20	0.30	0.43
fr2_deskwp (110)	**3.5**	4.5	4.2	31.73	-	6.85	X	0.63	1.99
fr3_sitxyz (22)	**1.3**	2.2	1.4	6.94	-	-	0.83	0.79	0.07
fr3_longoff (147)	**4.1**	12.3	9.9	36.9	3.50	-	X	3.45	1.97
fr3_nstr_tex_far (29)	3.2	5.8	**3.1**	17.9	-	-	4.92	*	*

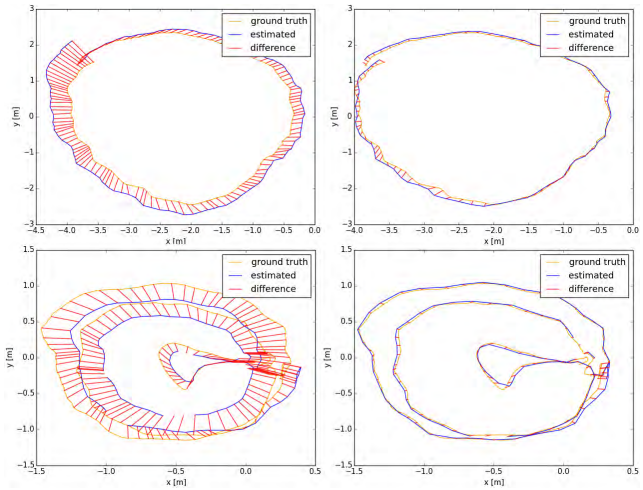


FIGURE 11. Trajectories of *fr3_nostructure_texture_near* and *fr2_dishes* sequences from the TUM RGB-D dataset. Left: Keyframe trajectories generated by LSD-SLAM. Right: Keyframe trajectories generated by our method.

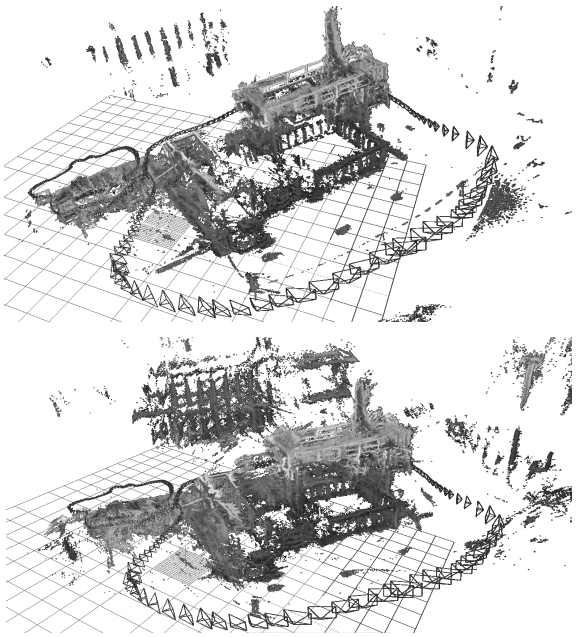


FIGURE 12. Accumulated pointclouds of GCPs of all keyframes from LSD-machine sequence thresholded by different maximum variance.

SLAM (DVO-SLAM) [32], feature-based SLAMs including RGBD-SLAM [33], PTAM, ORB-SLAM and PL-SLAM in Table 3. For fair comparison, the parameter settings in GCP-SLAM for keyframe selection, loop detection and others are the same as LSD-SLAM. Note that the localization error results showed in Fig. 8 and Table 3 are the median over 5 executions in each sequence.

In Table 3, our method significantly improves the accuracy of camera localization compared with LSD-SLAM in terms of confidence based depth refinement, GCPs-based motion tracking and both of them. Our method with depth refinement stabilizes the depth estimation and shows better performance than only using GCPs-based motion tracking. Using both of the depth refinement and GCP-based motion tracking improves the performance on *fr*1*_floor*, which contains much unreliable large-gradient pixels with estimated depth. But the results are similar or even worse on some sequences, e.g., *fr*3*_long_office_household*, compared with only using depth refinement. It demonstrates that if the number of observed pixels in one keyframe is not very sufficient, pixels with less confidence should be kept to benefit the optimization. On some sequences such as *fr*1*_xyz*, *fr*2*_xyz* and *fr*2*_desk_with_person*, our GCP-SLAM is even comparable with or better than the RGBD-based SLAMs. Tracking performs well on the sequence that PTAM, ORB-SLAM and PL-SLAM fail and achieves comparable results on sequence *fr*3*_long_office_household* and *fr*3*_nostructure_texture_far*. The improvement on sequences *fr*2*_desk_with_person* and *fr*3*_sitting_xyz* verify the robustness of our method on sequences including dynamic situation.

Moreover, the original LSD-SLAM fails on the same sequence occasionally, where tracking lost occurred at totally different frame [34], and sometimes may also has problem on detecting loop. The keyframe trajectory error can differ greatly for different runs. In Fig. 10, the best and worst keyframe trajectories for *fr*3*_long_office_household* over 5 executions using

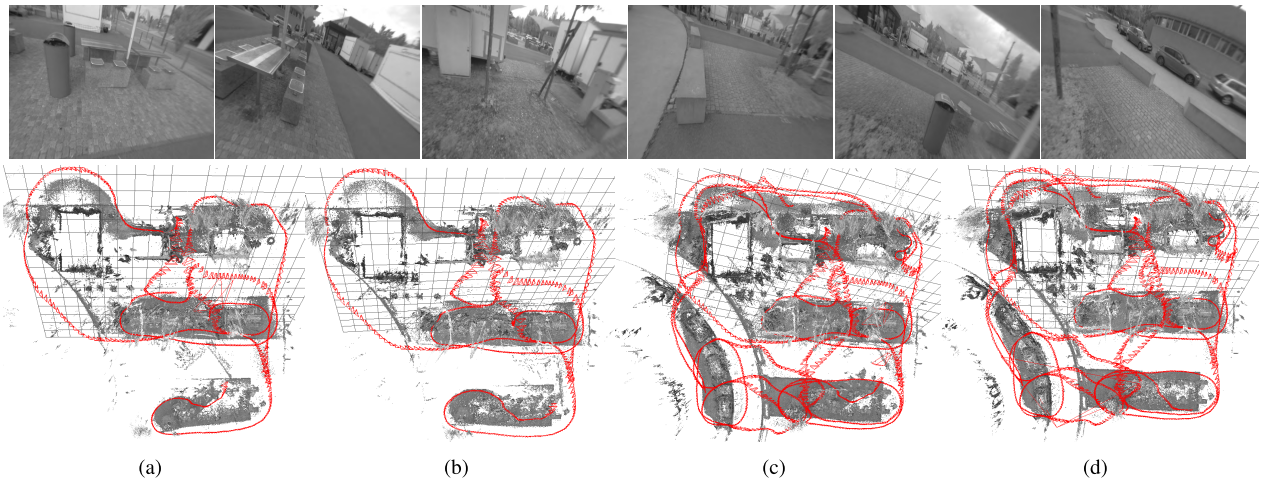


FIGURE 13. Accumulated pointclouds of keyframes at different times generated by (a)(c) LSD-SLAM and (b)(d) GCP-SLAM for LSD-foodcourt sequence with scale variation.

LSD-SLAM are shown. The results are more stable and consistent using our GCP-SLAM. Fig. 9 reports the accumulated pointclouds with trajectories of three sequences generated by LSD-SLAM and GCP-SLAM. The scale drift is obvious on these sequences using LSD-SLAM, and loop closure fails on sequence *fr3_long_office_household* and *fr3_nostructure_texture_near*. The keyframe trajectories of LSD-SLAM and our GCP-SLAM on long sequence with loop including *fr2_dishes* and *fr3_nostructure_texture_near* in one run are illustrated in Fig. 11, from which we can see that our method tracks frames precisely and can alleviate the scale drift caused by depth triangulation, benefiting the loop detection and pose-graph optimization.

Fig. 12 illustrates the accumulated pointclouds of a large scale outdoor scene with different threshold of maximum variance. We can see that the reconstructions constrained by the estimated confidence is more accurate while setting larger threshold of maximum variance, which can preserve 3D points with reliable prediction and exclude more uncertainty as the reconstruction becoming denser. Fig. 13 shows a top view of pointclouds with trajectory of keyframes at different times on foodcourt sequence, and selection of the original frames. LSD-SLAM easily drifts, fails tracking and detecting loop on this sequence, which validates the effectiveness of our GCP-SLAM on large-scale scenes.

V. CONCLUSION

In this paper, we have shown that when adopting bundle adjustment is difficult, the integration of direct SLAM with confidence learning via random regression forest is a promising solution to optimize the localization. The proposed GCPs-based monocular SLAM method, called GCP-SLAM, continually tracks the motion of camera on both $\mathfrak{se}(3)$ and $\mathfrak{sim}(3)$, while maintains a pose-graph of keyframes with probabilistic depth and confidence maps in real-time on a CPU. In contrast to LSD-SLAM, a trained random regression forest model is used to predict confidence value for depth estimation and detect GCPs, which help in avoiding the wrong fusion of depth and improving the accuracy of camera localization. We experimentally show that our learning based confidence estimation model is effective and the proposed semi-dense direct SLAM system performs reliably in tracking and relocalization especially for long sequences with loop. Besides, our framework is capable of reconstructing a more accurate 3D model of the indoor and outdoor environment while excluding severe drift points.

REFERENCES

[1] P. Norgren and R. Skjetne, "A multibeam-based SLAM algorithm for iceberg mapping using AUVs," *IEEE Access*, vol. 6, pp. 26318–26337, 2018.

[2] J. Luo and S. Qin, "A fast algorithm of SLAM based on combinatorial interval filters," *IEEE Access*, vol. 6, pp. 28174–28192, 2018.

[3] L. Zhang, L. Wei, P. Shen, W. Wei, G. Zhu, and J. Song, "Semantic SLAM based on object detection and improved octomap," *IEEE Access*, vol. 6, pp. 75545–75559, 2018.

[4] W. N. Greene, K. Ok, P. Lommel, and N. Roy, "Multi-level mapping: Real-time dense monocular SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2016, pp. 833–840.

[5] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 834–849.

[6] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality (ISMAR)*, Nov. 2007, pp. 225–234.

[7] R. Mur-Artal, J. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[8] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—A modern synthesis," in *Proc. Int. Workshop Vis. Algorithms*. Berlin, Germany: Springer, 1999, pp. 298–372.

[9] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual Odometry for a monocular camera," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1449–1456.

[10] A. Chiuso, P. Favaro, H. Jin, and S. Soatto, "Structure from motion causally integrated over time," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 523–535, Apr. 2002.

[11] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

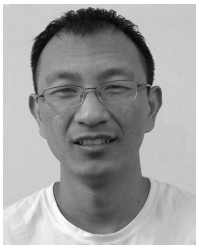
[12] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.
[13] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2320–2327.
[14] R. A. Newcombe and A. J. Davison, "Live dense reconstruction with a single moving camera," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 1498–1505.
[15] M. Pizzoli, C. Forster, and D. Scaramuzza, "REMODE: Probabilistic, monocular dense reconstruction in real time," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May/Jun. 2014, pp. 2609–2616.
[16] J. Stühmer, S. Gumhold, and D. Cremers, "Real-time dense geometry from a handheld camera," in *Proc. Joint Pattern Recognit. Symp.*, 2010, pp. 11–20.
[17] A. Glover, W. Maddern, M. Warren, S. Reid, M. Milford, and G. Wyeth, "OpenFABMAP: An open source toolbox for appearance-based loop closure detection," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2012, pp. 4730–4735.
[18] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6243–6252.
[19] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom.*, May/Jun. 2014, pp. 15–22.
[20] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.
[21] A. Spyropoulos, N. Komodakis, and P. Mordohai, "Learning to detect ground control points for improving the accuracy of stereo matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1621–1628.
[22] M.-G. Park and K.-J. Yoon, "Leveraging stereo matching with learning-based confidence measures," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 101–109.
[23] M. Poggi and S. Mattoccia, "Learning a general-purpose confidence measure based on O(1) features and a smarter aggregation strategy for semi global matching," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 509–518.
[24] S. Kim, D. Min, S. Kim, and K. Sohn, "Feature augmentation for learning confidence measure in stereo matching," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 6019–6033, Dec. 2017.
[25] J. Valentin, M. Niebner, J. Shotton, A. Fitzgibbon, S. Izadi, and P. Torr, "Exploiting uncertainty in regression forests for accurate camera relocalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4400–4408.
[26] Z. Zhong, S. Su, D. Cao, S. Li, and Z. Lv, "Detecting ground control points via convolutional neural network for stereo matching," *Multimedia Tools Appl.*, vol. 76, no. 18, pp. 18473–18488, Sep. 2017.
[27] J. Civera, A. J. Davison, and J. M. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 932–945, Oct. 2008.
[28] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 573–580.
[29] A. Handa, R. A. Newcombe, A. Angeli, and A. J. Davison, *Real-Time Camera Tracking: When is High Frame-Rate Best?* Berlin, Germany: Springer, 2012.
[30] 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 Int. Conf. Robot. Autom.*, May/Jun. 2017, pp. 4503–4508.
[31] M. Nieβner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3D reconstruction at scale using voxel hashing," *ACM Trans. Graph.*, vol. 32, no. 6, Nov. 2013, Art. no. 169.
[32] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 2100–2106.
[33] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the RGB-D SLAM system," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2012, pp. 1691–1696.
[34] A. Huletski, D. Kartashov, and K. Krinkin, "Evaluation of the modern visual SLAM methods," in *Proc. Artif. Intell. Natural Lang. Inf. Extraction, Social Media Web Search FRUCT Conf. (AINL-ISMW FRUCT)*, Nov. 2015, pp. 19–25.

WEIQI ZHANG received the B.S. degree from the Harbin Institute of Technology, Harbin, China, in 2014, where she is currently pursuing the Ph.D. degree in computer science. She has published two papers in academic journals. Her research interests include machine learning, computer vision, and SLAM.

ZIFEI YAN received the Ph.D. degree in computer application technology from the Harbin Institute of Technology, Harbin, China, in 2010. From 2007 to 2009, she was a Research Assistant with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. From 2014 to 2015, she was a Visiting Scholar with the University of Pittsburgh. She is currently a Lecturer with the School of Architecture, Harbin Institute of Technology. She has published more than 20 papers in academic journals and conferences. Her current research interests include machine learning and computer vision.

GANG XIAO received the M.M. degree in clinical medicine from Harbin Medical Sciences University, Harbin, China, in 2002. He is currently an Associate Chief Physician and also the President of the No.962 Hospital of PLA, Harbin. He has published more than 20 papers in academic journals. His research interests include medical biometrics and computerized diagnosis.

AIDI FENG received the B.E. and the M.S. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 2016 and 2018, respectively. Her research interests include machine learning, 3D reconstruction, and SLAM.



WANGMENG ZUO (M'09–SM'14) received the Ph.D. degree in computer application technology from the Harbin Institute of Technology, Harbin, China, in 2007, where he is currently a Professor with the School of Computer Science and Technology. His current research interests include image enhancement and restoration, image and face editing, object detection, visual tracking, and image classification. He has published over 80 papers in top-tier academic journals and conferences. He has served as a Tutorial Organizer at ECCV 2016 and an Associate Editor for the *IET Biometrics* and *Journal of Electronic Imaging*.