Accurate Indoor Localization through Constrained Visual SLAM

Olivier Gomez, Achkan Salehi, Vincent Gay-Bellile, Mathieu Carrier

Accurate Indoor Localization through Constrained Visual SLAM

Olivier Gomez, Achkan Salehi, Vincent Gay-Bellile and Mathieu Carrier
CEA, LIST, Laboratoire Vision et Ingénierie des Contenus
Point Courrier 94, Gif-sur-Yvette, F-91191 France
Email: {olivier.gomez2, achkan.salehi, vincent.gay-bellile, mathieu.carrier}@cea.fr

***Abstract*— In this paper, we focus on navigation in indoor environments using Visual SLAM (VSLAM). We propose an approach to suppress the known drifting issue of VSLAM and express its localization in building coordinate frame. It relies on a database built offline through a coarse to fine strategy that registers and refines a VSLAM reconstruction by taking advantage of the building 3D model. The database can then be extended online when the user goes out and comes back in the known environment. We present experimental results on synthetic and real data.**

I. Introduction

Guiding a user inside a building requires an accurate localization. To provide this localization, existing methods often rely on WIFI/Bluetooth/RFID beacons [2], [3]. However the precision of methods based on beacons depends especially on their number and their disposition on the site. Furthermore these methods need to equip their environment beforehand and are thus invasive.

An alternative is to use vision based solutions such as Visual Simultaneous Localization And Mapping (VSLAM). They provide real-time localization with no prior knowledge about the environment nor additional equipment. However the resulting localization is expressed in an arbitrary coordinate frame and suffers from drift due to error accumulation. Recent methods try to overcome the drift issue of VSLAM by adding an IMU sensor. Visual-Inertial SLAM (VISLAM) [5] provides a more accurate localization but not drift free for long trajectories. Furthermore, the resulting localization is still not expressed in the building coordinate frame. VISLAM methods are thus not well suited for the intended application.

To tackle VSLAM limitations for user guidance application, we propose an approach which uses a previously built database of the environment (expressed in the building coordinate frame) in order to constrain the VSLAM reconstruction. The resulting localization is thus expressed in the building coordinate frame and do not drift as long as the user operates in the known part of the environment, *i.e.* where the database is available. To build that database, we use the $3D$ textureless model of the building, which can easily be obtained from blueprints or with $3D$ scanning, to register and refine a SLAM reconstruction in the model coordinate system with a coarse to fine approach. Whenever the user operates in an unknown environment, our approach behaves as a VSLAM and its localization accumulates error over time. However when the user comes back in a known environment the SLAM drift can be estimated and corrected through a pose-graph optimization. That corrected trajectory is then added to the database leading to its enrichment.

Compare to existing visual solutions such as [1] that also exploit a database for indoor localization, the contribution of the proposed framework are: 1) the coarse to fine approach for building accurately the database and express it in the building coordinate frame. 2) online localization through VSLAM constrained to the database rather than relocalization on each frame. It guarantees a continuity of service even when the relocalization fails. 3) online database extension.

In this paper, we first present the VSLAM algorithm which is at the core of both the database construction and the online tracking. Secondly, we detail the database construction with on going work to improve its accuracy. Then we present our VSLAM algorithm constrained to the database as well as the online database extension process. Finally, we show some results of our framework on synthetic and real data.

II. Visual SLAM

The Visual Simultaneous Localization And Mapping presented is a key-frame based VSLAM [6]. It provides continuous frame-to-frame pose estimation from $2D$/$3D$ matching. When the camera displacement is large enough, a new key-frame is labeled and new $3D$ points are triangulated. Because that reconstruction is often imprecise, it needs to be refined. Consequently, a local Bundle Adjustment (BA) [6] is performed to simultaneously optimize the camera poses $\{P_j\}_{j=0}^{N_c}$ of the last N_c key-frames and the N_p $3D$ points $\{Q_i\}_{i=0}^{N_p}$ they observe by minimizing the re-projection error in the key-frames. The cost function of BA, minimized with the Levenberg-Marquardt algorithm, is given by:

$$B\Big(\{P_j\}_{j=0}^{N_c}, \{Q_i\}_{i=0}^{N_p}\Big) = \sum_{i=0}^{Np} \sum_{j \in A_i} \rho\Big(q_{i,j} - \pi(KP_jQ_i)\Big) \qquad (1)$$

where $q_{i,j}$ is the $2D$ observation of Q_i in camera j, K is the camera calibration matrix, $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ is the perspective projection function, A_i denotes the set of key-frame indices observing Q_i and ρ is the Geman-McClure kernel to deal with outliers. VSLAM tends to drift, however when a loop occurs this drift can be observed and corrected. The loop closure detection is performed in a re-localization thread that uses of a Vocabulary Tree (VT) structure [8] to index each key-frame by their $2D$ observations. Once a key-frame is created, the re-localization thread queries the VT with the key-frame

observations to find the most similar indexed key-frame that does not share any $3D$ point. Once a loop closure is detected, $3D$/$3D$ correspondences between the $3D$ points observed by the current key-frame e and those observed by the most similar indexed key-frame b are established. From these correspondences, a similarity ΔS_{err} encoding the drift error in 7 Degree of Freedom (DoF), rotation, translation and scale, is estimated. The drift correction is achieved with a pose-graph optimization that minimizes the relative pose errors $\{r_{i,j}\}_{i,j=b,b+1}^{e-1,e}$ between successive cameras and the error $r_{e,b}$ deducted from the loop closure [10]. Each pose P_i and relative pose $\Delta P_{i,j}$ are transformed to a similarity S_i and $\Delta S_{i,j}$ by adding a scale set to 1. Only the loop constraint $\Delta S_{e,b}$ has a scale $s_{err} \neq 1$. The pose-graph optimization problem is expressed as the minimization of the following cost function:

$$\chi^2(S_b, ..., S_e) = \sum_{i=b}^{e-1} \sum_{j=i+1} \left(r_{i,j}^T \, r_{i,j} \right) + r_{e,b} \tag{2}$$

where S_b is fixed and the relative position error $r_{i,j}$ is:

$$r_{i,j} = log_{Sim(3)}(\Delta S_{i,j} \cdot S_i \cdot S_j^{-1}) \tag{3}$$

where $\Delta S_{i,j} = \hat{S}_j \cdot \hat{S}_i^{\,-1}$ is the initial relative pose between camera i and j computed before optimization. For the loop constraint $\Delta S_{e,b} = \Delta S_{err}$.

III. Proposed method

Our solution provides real-time indoor localization through the use of a database to reduce VSLAM drift, even when no loop occurs, and express the localization in the building coordinate system. This section details in §III-A the construction of the database, then explains in §III-B the process to constrain the VSLAM in known environment with that database and finally §III-C presents how the database is extended when an unknown environment is explored. As this paper is a work in progress, sections §III-A and §III-C are divided into "achieved" and "ongoing" work subsections to ease the discussion.

A. Database construction

1) Achieved work: The database construction is performed off-line through a coarse to fine approach in 4 steps. The resulting database contains a set of key-frame camera poses and a $3D$ point cloud along with their $2D$ observations in the key-frames.

First step: Create an initial reconstruction of the environment using the VSLAM described in §II. During the VSLAM tracking, some key-frames are associated to an absolute pose expressed in the building coordinate frame. The absolute poses can be obtained for example with visual markers placed in the environment and removed after the database construction or by any other approach that can give an absolute pose in the building model.

Second step: Express the VSLAM reconstruction in the buildings coordinate frame. Therefore a similarity (7 DoF) is estimated from the absolute poses and is applied on the VSLAM reconstruction.

Third step: Perform a coarse non-rigid correction through a pose-graph optimization. The purpose is to shift the key-frames poses toward their absolute poses without breaking the epipolar geometry between cameras. Therefore a pose-graph optimization is performed to distribute high relative pose errors around the shifted cameras along the entire trajectory.

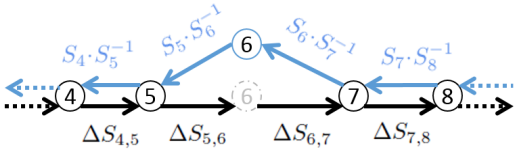

Fig. 1. Illustration of the third step of the database construction. In this example the 6^{th} key-frame has an absolute pose associated and is shifted towards it, resulting in large errors $r_{5,6}$ and $r_{6,7}$ (see equation (3)) that are minimized through pose-graph optimization while fixing the 6^{th} camera pose.

This optimization requires to firstly compute the initial relative position $\Delta S_{i,j}$ of the graph constraints. Then the key-frame poses are shifted towards their absolute poses. Finally, the sum of relative pose errors along the graph $\sum_{i,j} (r_{i,j}^T \, r_{i,j})$ is minimized while fixing the shifted camera poses (Figure 1). As the pose-graph optimization only concerns camera poses, the $3D$ points have to be re-triangulated.

Fourth step: The final step is a non linear refinement of the reconstruction through a BA that is constrained to the building model. The aim of this final step is to align at best the subset $\mathcal{Q}$ of the $3D$ point cloud that corresponds to the building, e.g. walls, with the building model. Currently, $\mathcal{Q}$ is determined using a simple ray-tracing as in [4]. Once $\mathcal{Q}$ is known, we use a simple proximity criterion to associate each $Q_i \in \mathcal{Q}$ to a plane W_i of the building mesh and project Q_i on the plane W_i. Then, for $Q_i \in \mathcal{Q}$, re-projection error of Q_i is replaced by $\rho(q_{i,j} - \pi(K P_j M_{W_i} Q_i^W))$ in equation (1) where Q_i^W is Q_i expressed in W_i coordinate frame and limited to two degree of freedom ($z_{Q_i^W} = 0$). M_{W_i} is the change of basis matrix from the plan W_i coordinate frame to building model coordinate frame.

2) Ongoing work: The ray-tracing method of [4] for determining $\mathcal{Q}$ is not well adapted to cluttered environments since occluding objects will mistakenly be associated to the walls. This inevitably leads to an inaccurate refinement of the reconstruction. In order to alleviate this problem, we seek to use a neural network to segment each key-frame, keeping only observations of the building structure that may be associated with a plane in the model, (*e.g.* wall, door or window frame. Therefore, $Q_i \in \mathcal{Q}$ is determined through a majority vote on the class labels of its observations as given by the corresponding segmented key-frames. We use an Enet [9] based architecture for binary pixel-wise labeling. We have used the Nyu dataset [7], as well as interior scenes that we gathered for training. However, the genericity of the obtained

network is not yet satisfactory, and we plan as a future work to train it on an augmented version of our dataset.

B. Constrain VSLAM to a database

The first stage is to initialize the VSLAM algorithm with the database to fix the coordinate frame and the scale. Then, when a new key-frame is detected, $3D/3D$ correspondences between the online reconstructed $3D$ point by VSLAM algorithm and $3D$ point belonging to the database are established. These correspondences are used to constrain the bundle adjustment and thus reduce the VSLAM drift.

Initialization. First of all, every database key-frame is indexed in the VT structure described in §II. VSLAM initialization is achieved by querying VT with the current frame observations to find the most similar database key-frame. When a similar viewpoint is found, $3D$ points of database, observed by this key-frame, are matched with the current frame observations. These $2D/3D$ associations are used to compute a pose to initialize the VSLAM algorithm. Furthermore, the $3D$ points that are inliers after the pose estimation form its initial $3D$ point cloud.

Find 3D/3D correspondences. During VSLAM localization, the re-localization thread tries to find, for each key-frame, a corresponding one in the database. When it succeeds, image matching determines $2D/2D$ associations between these two images from which are deduced $3D/3D$ associations between the $3D$ points they respectively observe. Key-frames successfully matched with the database are referred as re-localized. During online localization, only non re-localized key-frames are indexed in the VT to avoid redundancy.

Constrained Bundle Adjustment. $3D$ points that have a correspondent in the database are fixed in the bundle adjustment (equation 1), their positions are previously updated with the ones of the database points. These constraints reduce the SLAM drift as demonstrated in section IV.

C. Database extension

At some point, the user may explore an unknown part of the environment which is characterized by re-localization failure over several consecutive key-frames. As our constrained VSLAM behaves as a VSLAM in unknown environment, its localization inevitably drift over time. However, when the user goes back in a known environment, re-localization resumes and $3D/3D$ associations with the database are computed as well as a similarity ΔS_{ext}, encoding VSLAM drift error.

1) Achieved work: Correcting the VSLAM drift directly through a constrained BA may fail due to high re-projection errors, resulting in a bad convergence of the optimization. In order to continuously provide a localization, ΔS_{ext} is applied to the last N_c cameras and $3D$ points they observe. At this point, the constrained VSLAM localization could successfully resume, however, the database can not be extended by this additional trajectory since the epipolar geometry is now broken between cameras $l - N_c$ and $l - N_c + 1$, where l is the index of the current re-localized key-frame. In order to reduce the error, and thus maintain the epipolar constraint, a pose-graph

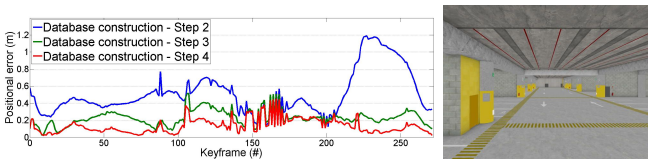

Fig. 2. **Database creation on the synthetic sequence.** Left: error in position at different step of its construction, see §III-A. Right: one image of the synthetic sequence.

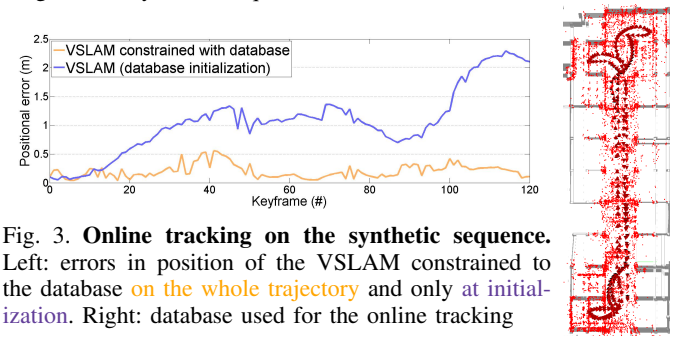

Fig. 3. **Online tracking on the synthetic sequence.** Left: errors in position of the VSLAM constrained to the database on the whole trajectory and only at initialization. Right: database used for the online tracking

optimization is performed. We note f the index of the second to last re-localized key-frame. The sum of relative pose errors $\sum_{i=f}^{l-N_c} \sum_{j=i+1} \left(r_{i,j}^T \, r_{i,j}\right)$ is minimized while fixing cameras f and $l-N_c+1$. For the relative pose error $r_{(l-N_c),\,(l-N_c+1)}$ (see equation 3), $\Delta S_{(l-N_c),(l-N_c+1)}$ is computed before applying ΔS_{ext} to S_{l-N_c+1}. Finally, the key-frames from $f+1$ to $l-1$ and the $3D$ points they observe, after their re-triangulation, are simply added to the database.

2) Ongoing work: Our current solution corrects VSLAM drift ΔS_{ext}. Yet, the result is a coarse correction of the trajectory and may not be as accurate as the database built offline. To improve the accuracy of online extensions, we aim to refine them with the BA constrained to building model, described in §III-A. As it is computationally expensive to perform such a BA online, it will be achieved in a separate low-important thread in order not to perturb VSLAM localization.

IV. Experiments

We present results based on achieved works, *i.e.* §III-A1, §III-B and §III-C1. The database construction and online tracking are assessed on both synthetic and real data. The constrained VSLAM performs a localization at 60Hz (640x480 resolution) on a Microsoft Surface Pro 4 tablet with an Intel Core i7-6650U @2.2GHz.

Synthetic data. The sequence represents a $173\text{x}52\,\text{m}$ parking lot as illustrated in Figure 2. The VSLAM algorithm described in §II is applied on this sequence. Several loops are detected and corrected. The resulting reconstruction includes 272 key-frames and 23777 $3D$ points. Then the second and third steps of the database construction are performed by using 8 well distributed absolute poses. They come from the groundtruth and have been slightly degraded in order to be closer to a real scenario. Figure 2 shows the error in position, with respect to the groundtruth, after each optimization step of the database construction. At the end of the second step, the mean error is $0.5101\,\text{m}$, $0.226\,\text{m}$ after the third step and

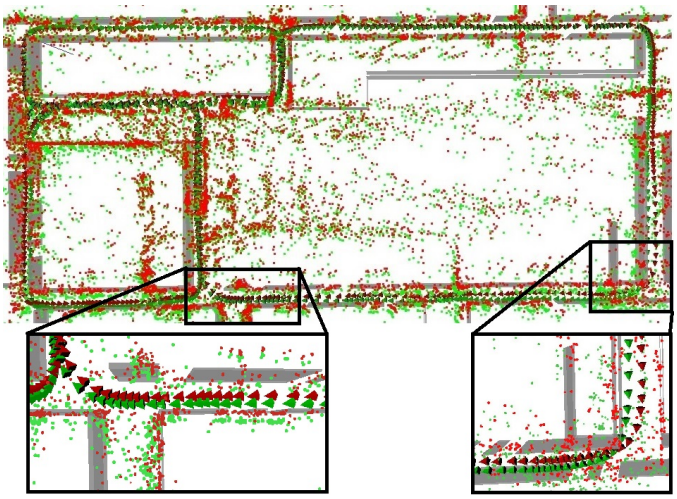

Fig. 4. **Database creation on the office building sequence.** In gray the building model, in green (*resp.* in red) the database obtained at the end of the third (*resp.* the fourth) step described in §III-A. The final $3D$ point cloud (after step four) is well aligned with the building model.

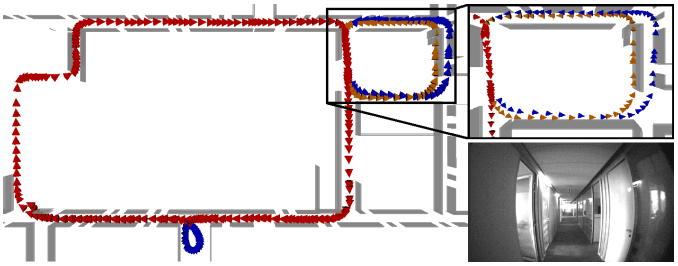

Fig. 5. **Online tracking and database extension on the office building sequence.** Left: top-down view of the trajectory with re-localized key-frames represented in red and key-frames added to the database in blue. Top Right: A closeup view of one database extension that required a pose-graph optimization before being added in the database. The cameras poses before this optimization are represented in orange. Right: one image of this sequence.

the final error is 0.1321 m. For this sequence, both the third and the fourth steps halve the error in position. The former is performed in 0.597 s while the latter is more time consuming and takes 19 s. The resulting database is illustrated in Figure 3.

Using the previously created database, online tracking is performed on a second synthetic sequence where the trajectory is confined to the known environment. Figure 3 shows the error in position of two different VSLAM execution. The first one uses the database at initialization and during the overall trajectory as a constraint in the BA, whereas the second one only uses it for its initialization. While the first execution has a small error (mean error 0.1852 m), the second one has a higher and steadily growing error (mean error 1.0338 m) with a maximum of 2.2925 m. The second execution accumulates error over time which demonstrates that constraining the VSLAM to the database reduce drastically the drift.

Real data. The sequence represents a 76x34 m office building floor with long corridors in an environment where there is no major occlusion of the walls. The online VSLAM algorithm detailed in §II is applied on this sequence, where multiple loops are detected and corrected. Four visual markers have been placed in the outer corners of the floor to obtain the required absolute poses for the second and third steps of the database construction. The reconstruction for the office building includes 337 key-frames and 16937 $3D$ points. Figure 4 shows the database construction on this sequence where the fourth step drastically improves the accuracy, since it results in a well alignment between the $3D$ point cloud and the model.

The online tracking is realized 6 months after the database creation, and visual markers used for its construction have been removed. The re-localization successfully finds $3D$/$3D$ associations with the database. The constrained VSLAM localization does not drift as long as it remains in the known environment. Unknown environments have also been explored, resulting in two databases expansion as illustrated in Figure 5. For the first one, the trajectory is short, thus no drift occurs. However for the second exploration, the drift is important and a graph optimization is performed to correct the trajectory enabling thereafter the constrained BA to converge. The improvement of the estimated trajectory is illustrated in Figure 5. The key-frames poses do not cross the wall anymore after graph optimization.

V. Conclusion

In this paper, we present a VSLAM constrained to a pre-build database. It is obtained through a coarse to fine approach that exploit the building model to improve the accuracy. We demonstrate that our localization solution suppress the VSLAM drift and express its localization in the building coordinate frame while extending the database online when unknown environment is explored. This makes our approach suitable for user guiding applications. To deal with more challenging environment and increase the accuracy of the online database extension, we will improve our framework as mentioned in §III-A2 and §III-C2.

References

[1] E. Deretey, M. T. Ahmed, J. A. Marshall, and M. Greenspan, "Visual indoor positioning with a single camera using pnp," in *IPIN, 2015*.

[2] G.-y. Jin, X.-y. Lu, and M.-S. Park, "An indoor localization mechanism using active rfid tag," in *Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006.*

[3] L. Kanaris, A. Kokkinis, A. Liotta, and S. Stavrou, "Fusing bluetooth beacon data with wi-fi radiomaps for improved indoor localization," *Sensors*, vol. 17, no. 4, 2017.

[4] D. Larnaout, V. Gay-Bellile, S. Bourgeois, and M. Dhome, "Fast and automatic city-scale environment modelling using hard and/or weak constrained bundle adjustments," *Machine Vision and Applications*, vol. 27, no. 6, pp. 943–962, 2016.

[5] S. Leutenegger, P. T. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart, "Keyframe-based visual-inertial slam using nonlinear optimization." in *Robotics: Science and Systems*, 2013.

[6] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Real time localization and 3d reconstruction," in *CVPR, 2006.*

[7] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[8] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *CVPR, 2006.*

[9] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *arXiv preprint arXiv:1606.02147*, 2016.

[10] H. Strasdat, J. Montiel, and A. J. Davison, "Scale drift-aware large scale monocular slam," *Robotics: Science and Systems VI*, 2010.