Towards SLAM-based Outdoor Localization using Poor GPS and 2.5D Building Models

Ruyu Liu[*]
Zhejiang University of Technology
Hamburg University

Jianhua Zhang[†]
Zhejiang University of Technology

Shengyong Chen [‡]
Tianjin University of Technology

Clemens Arth[§]
Graz University of Technology

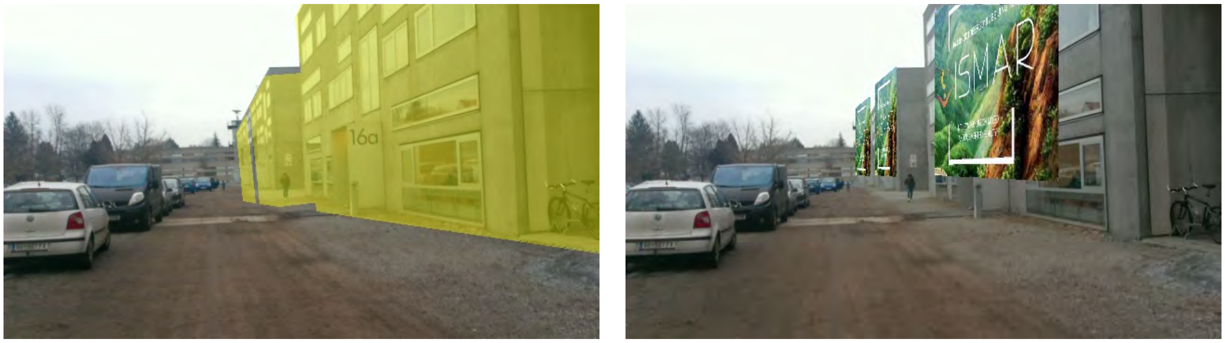


Figure 1: Results from our SLAM-based localization system after optimization. Left: an overlay of the 2.5D building model from OpenStreetMap. Right: ISMAR logos superimposed onto the façades using the surfaces from the 2.5D maps.

Abstract

In this paper, we address the topic of outdoor localization and tracking using monocular camera setups with poor GPS priors. We leverage 2.5D building maps, which are freely available from open-source databases such as OpenStreetMap.

The main contributions of our work are a fast initialization method and a non-linear optimization scheme. The initialization upgrades a visual SLAM reconstruction with an absolute scale. The non-linear optimization uses the 2.5D building model footprint, which further improves the tracking accuracy and the scale estimation. A pose optimization step relates the vision-based camera pose estimation from SLAM to the position information received through GPS, in order to fix the common problem of drift.

We evaluate our approach on a set of challenging scenarios. The experimental results show that our approach achieves improved accuracy and robustness with an advantage in run-time over previous setups.

Index Terms: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically-based Modeling; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality; H.5.2 [Information Interfaces and Presentation]: User Interfaces—Direct Manipulation methods

1 Introduction

Augmented Reality (AR) is commonly expected to change the way we perceive our environment and interact with digital information in the near future. Enriching our surroundings with virtual information has a large potential for entertainment purposes like gaming[1], but also for serious applications in infrastructure maintenance [34] or rescue operations [15], for example.

As the primary requirements for AR include real-time behaviour, the fusion of virtual and real and accurate registration [5], there are a lot of problems to solve. Mobile devices have reached certain maturity to enable real-time processing of sensor data. Similarly, mobile Graphics Processing Units (GPUs) are capable of creating impressive visualization of content. However, the problem of accurate registration is yet a field of active research targeted by a large number of research groups and big commercial companies like Facebook, Microsoft and Google.

A very promising starting point for accurate registration is Simultaneous Localization And Mapping (SLAM)[2]. SLAM has been extensively used in AR systems driven mainly by the work of [20] for creating a model of the local environment and tracking the camera relative to this map. Purely monocular SLAM systems adopt two-frame or multi-frame approaches, scaling in terms of complexity with the available computational resources. Common to these approaches is the requirement to translate the camera over a certain distance during initialization. This requirement stems from the need to establish stable triangle equations, which can be solved within a robust scheme [12]. Some other systems use RGBD-sensors to include a sense of depth for building the local model, which alleviate the necessity for translation at initialization. However, due to the restrictions of the depth sensing technology involved, this comes at the cost of being applicable only within small to medium sized environments indoors.

Given the requirement of translation in the former and the restriction to small environments in the latter group of approaches, applying SLAM in outdoor scenarios robustly is a complicated task. While covering a certain translational motion with a handheld camera is perfectly feasible indoors, having to walk several meters outdoors is cumbersome. Besides other issues owing to camera acquisition outdoors like motion blur, SLAM is inherently built to establish a local coordinate frame without any relations to the global surrounding. Therefore the model created misses the appro-

[*] e-mail: liuliu609470295@gmail.com
[†] e-mail: zjh@zjut.edu.cn
[‡] e-mail: sy@ieee.org
[§] e-mail: arth@icg.tugraz.at

[1] Pokemon GO. `https://www.pokemongo.com`

[2] While SLAM approaches differ by the nature of sensor data used, for the rest of the paper, we use SLAM as a synonym for visual SLAM, i.e. SLAM using camera images.

priate scale factor relating it to globally registered data, ie. any geo-referenced data.

In this work, we propose an approach for outdoor SLAM-based localization leveraging commonly available 2.5D building footprints and poor Global Positioning System (GPS) sensor data. The main contributions of this work is (i) a hybrid graph-based optimization approach to fuse SLAM and data from the 2.5D model with low computational effort, in order to improve the accuracy of both the reconstruction and the camera trajectory, as well as to reduce drift. As a side contribution (ii) we employ a new fast initialization routine for SLAM from a single image and the 2.5D map, applying synthetic depth images in new ways.[3] The experimental results (see Fig. 1 for an example) proof the plausibility of our approach.

2 Related Work

An exhaustive overview of SLAM and localization literature is beyond the scope of this paper. It is worth noting that the terms registration and localization in a global context are often used interchangably in the literature. However, approaches differ widely from determining only a 2D position with several meters of inaccuracy, up to resolving a full 6DOF pose with centimeter-accurate position and sub-degree accurate rotation estimation.

As our approach definitely falls into the latter category, in the following we focus only on the most prominent approaches from SLAM and on those most related to our concept. Similarly, we give an overview of recent vision-based outdoor localization methods and hybrid approaches proposed in the AR community.

Database and Reconstruction-based Approaches

A large number of approaches from this category perform varying degrees of global registration using a query image and retrieving a related image from an offline image database. The most prominent approach in this respect was published by [35], and more recently by [6, 13] and others. Some proposed methods leverage point clouds [17, 26] or sub-parts of point clouds [2, 4] together with source images from different domains. Common to these approaches is the lack of general scalability and the absence of appropriate update mechanisms to cope with changes of the environment over time.

SLAM Systems using Vision and Depth Sensors

A large number of Visual Odometry (VO) and approaches for monocular SLAM, multi-camera systems and systems featuring additional depth sensors were proposed in the past.

The earliest popular vision-only approach within AR was PTAM proposed by [20, 21]. More recently, ORBSLAM2 [27, 28] has become the basis for a lot of follow-up approaches. There are basically two ways to initialize a monocular SLAM system. The first is to use eight-point algorithm [14] to calculate the homography matrix for planar scenes and using the five-point algorithm [31] to calculate the fundamental matrix for non-planar scenes. However, the recovered relative camera pose is ambiguous. In addition, initializing a system is difficult for an untrained user.

The second category leverages visual information such as point, line, plane and or even rich features within Convolutional Neural Networks (CNNs). [16] use traditional vanishing point methods to estimate the structure of a scene. More recently, depth estimation based on deep learning [7, 23, 39] were proposed. These methods learn the feature of depth images through a large number of datasets and finally get the best model to estimate depth values for new input samples.

Although methods based on vision only form the most popular group by far, RGBD cameras have also been proposed to perform close range depth estimation within SLAM. [30] proposed KinectFusion to densely reconstruct areas. [19] proposed InfiniTAM, which is also applicable on suitable mobile devices. Both methods are usable in indoor AR scenarios only, due to the restrictions placed by the depth sensing hardware involved.

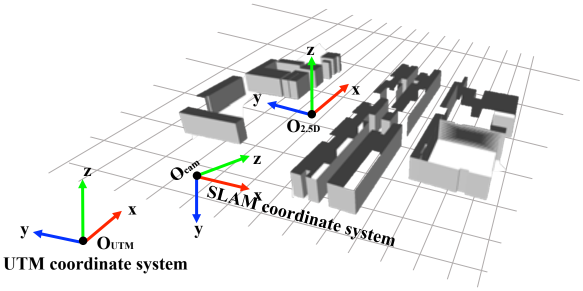


Figure 2: Description of the three coordinate system in our approach. The UTM coordinate system is the global coordinate system and the domain of the sensor data. The 2.5D map coordinate system is the local frame within visible range with a fixed origin within the global coordinate system. The SLAM system is the local coordinate system of the map, respectively the camera.

Resolving Scale Ambiguities

Resolving scale ambiguity has been a dominant topic not only in the multi-view geometry and reconstruction domain mainly, but also in SLAM. [9] propose to add spatially x calibration objects of known size into the scene for determining absolute scale of the camera motion and the scene structure in monocular SLAM systems. [8] for example use a 3D measurement model linked to the camera frame to initialize a monocular SLAM system. [22] estimate the absolute scale of a handheld monocular SLAM system by tracking a user's face.

[38] use multiple cues such as geo-tags, vanishing points and geo-referenced 3D models for global structure-from-motion (SfM) registration. [25] and others fuse visual sensors with an Inertial Measurement Unit (IMU) to obtain metric scale for such reconstructions.

Hybrid Systems for Outdoor Localization and Tracking

Several hybrid solutions to outdoor localization and tracking stem from the AR community. [29, 32, 37] are sophisticated and popular methods about fusing IMU and vision in recent years. [18] and [33] present tracking systems fusing image and sensor information for more accurate outdoor localization with high-quality sensing hardware. [25] and [10] propose visual-inertial SLAM for more precise camera pose estimation, using noisy sensor measurements only.

[36] use video streams and IMU data to perform camera registration. [24] improve the localization accuracy by fusion the SLAM/differential GPS with 3D building models. [1] train a CNN for geo-localization, given a semantic segmentation of the input image and 2.5D models.

Differentiation from Previous Work

Based on ORBSLAM2 [27, 28], closest to our work is the approach of [3] in terms of the use of 2.5D models. However, our system is not based on semantic segmentation and, instead, tightly integrates building model errors into an optimization scheme to closely align the reconstruction to the given 2.5D map. As opposed to a similar idea from [26] and [24], we integrate this error in a novel graph-based optimization scheme in 2D directly in the SLAM system. We are thereby able to bring ORBSLAM2 into the outdoor domain, which has rarely been thoroughly exploited yet.

3 Coordinate System and Initialization

The outdoor localization problem as defined in our case involves three different coordinate systems: (i) the global coordinate system, for which different geodetic formats are available; (ii) the 2.5D map coordinate system, which is a local coordinate system with metric scale; (iii) the SLAM coordinate system, *i.e.,* the local coordinate

[3] https://github.com/lauchlry/Buiding-GPS-SLAM

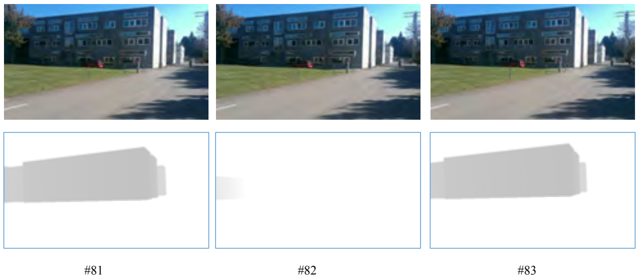


Figure 3: The drift problem of GPS sensors. Top: Three images from dataset 07-08. Bottom: Corresponding depth images calculated from the raw sensor pose. While the first and the last frame have reasonable pose estimates, the pose of the middle frame suffers from a completely wrong estimate due to temporal GPS drift.

system of the reconstructed map without global scale, in which the camera is moving. As the most common longitude-latitude based WGS84 system is not metric, we use Universal Transverse Mercator (UTM) in our setup for the global coordinate system. An illustration of these coordinate systems is given in Fig. 3.

The global coordinate system in UTM is a right-handed coordinate system whose $y-axis$ is pointing to north, the $x-axis$ pointing to the east and the $z-axis$ pointing to the sky respectively. The 2.5D map coordinate system is essentially a cut-out of the actual environment with its coordinate center having a fixed position within the global coordinate system. It is important to note that this coordinate system is introduced mainly to bring all the geo-referenced information into a coordinate system fitting the visual range. This is required to avoid numerical instabilities in the mathematical systems involved[4]. The sensor data samples available from common mobile devices consist of a timestamp, a 3-DOF position information GPS in WGS84 format, and a 3-DOF rotation information expressed in quaternions from the compass and the Inertial Measurement Unit (IMU). Thus, we can describe the sensor-based camera pose in the 2.5D map as a 4×4 matrix $T_{sensor}(i)$ consisting of a rotational $R_{sensor}(i)$ and translational $t_{sensor}(i)$ component where i denotes the i^{th} sensor sample:

$$T_{sensor}(i) = \begin{bmatrix} R_{sensor}(i) & t_{sensor}(i) \\ 0 & 1 \end{bmatrix} \quad . \tag{1}$$

Given the pose from sensors, we are able to generate a pseudo-depth image using the 2.5D map at hand, which essentially gives us an approximate distance value for each pixel in our camera frame. A naive way to create such a depth image is to calculate the distance D from each point $P_{2.5D}$ on the building surface to the camera center $O_{cam} = (X_{cam}, Y_{cam}, Z_{cam})$:

$$D = \sqrt{(X_{2.5D} - X_{O_{cam}})^2 + (Y_{2.5D} - Y_{O_{cam}})^2 + (Z_{2.5D} - Z_{O_{cam}})^2} \tag{2}$$

The depth information required is essentially encoded in the depth channel of a GPU rendering of the building models from the actual pose estimate. Therefore we render the 2.5D map from the current position and automatically retrieve the distance value for each pixel in our image as

$$D' = \frac{2.0 + D_{min} + D_{max}}{D_{max} + D_{min} - (2.0 * D - 1.0)} * (D_{max} - D_{min}) \quad . \tag{3}$$

D_{max} and D_{min} are chosen as the view frustum cut-off distances. Examples of such initial depth images are shown in Fig. 3.

Give the depth channel information, we then retrieve an image mask. Then, we extract ORB key points on the image area covered by the mask and estimate the corresponding distance information for each feature point,*i.e.,* potential SLAM map point using Eqn. 3. Finally, the feature point coordinate $p(x,y)$ in the images and the distance information are used to create a 3D map coordinate $P(X,Y,Z)$, which make up the initial SLAM map with the correct metric scale for subsequent SLAM tracking.

4 Optimization

The individual components used in our system provide complementary cues for accurate localization and tracking. On the one hand, GPS information and the 2.5D map provide a global metrics scale for large-scale outdoor environments. 2.5D maps are easy to obtain from public sources nowadays. From a practical point of view, sensors are small, cheap and low-power, but suffer from inaccuracies. On the other hand, SLAM contributes accurate local registration and tracking, but is hardly applicable in outdoor environments directly. Therefore it is required to fuse all information in a common optimization scheme described in the following.

Due to the liveliness of our system and the varying suitability of information during different states of the system, we establish a segmentation optimization mechanism. According to the active state of the system at a particular time instance, we enable certain groups of parameters within the optimization and leave out others.

At the initialization phase, an initial map based on our distance estimation approach is created. Since the initial map is based on singular feature observations, we are only able to optimize the camera pose, *i.e.,* track the camera relative to a growing, but static map. After individual map points are tracked successfully, *i.e.,* multiple observations have been collected from different camera positions, the optimization of map point positions can be safely enabled. This essentially upgrades the SLAM system to its usual common functional state, where we use the regular reprojection error for optimization. Lastly, as the SLAM system is set up, we add the building model-based error and include the actual sensor information into the optimization.

4.1 Building Model-based Optimization

The major aim of building model-based optimization is to align the camera trajectory and the associated SLAM map closer to the 2.5D map, in terms of the full 7-DOF (*i.e.,* translation, rotation and scale). However, it is equally important to improve the accuracy of localization and tracking within SLAM over time, as the initialization using the depth map rendering inherently added a significant amount of error, which ultimately stems from sensor inaccuracies. This applies to both rotational and translational discrepancies.

In order to solve this problem, our solution is to minimize the distance between the reconstructed 3D map points and the 3D building model façade based on graph optimization, followed by three steps.

4.1.1 Identifying the Corresponding Façade

The yaw angle and the position of current key frame are given by θ and (x,y,z). The horizontal field of view of the camera can be calculated according to the intrinsic parameters, denoted as $\Delta\theta_H$. Let f denote the focal length, and W being the width of input image.

$$\Delta\theta_H = 2 * \frac{\arctan(\frac{W}{2})}{f} \tag{4}$$

Then within$[\theta - \frac{\Delta\theta_H}{2}, \theta + \frac{\Delta\theta_H}{2}]$, we calculate a line of sight, *i.e.,* ray-trace, at intervals of 4°, followed by calculating the intersection points of the line of sight with all façades. At the end, we retain only valid intersection points with façades to buildings. By doing this, the corresponding façades are identified. Let

[4] Although systems like UTM are inherently local systems with respect to the globe, they span several hundreds of kilometers in each direction.

$\mathbf{B} = \{B_{11}, B_{12}, ..., B_{ij}, ..., B_{ml}\}$ represent all façades j of all buildings i in 2.5D map. m denotes the number of buildings, l denotes the number of façades in each building. Given the intersection points, we establish a boolean vector $\mathbf{B}_{isvisible}$, if the building façade is visible, we set the value *isvisible* true, otherwise it is false.

$$\mathbf{B}_{isvisible} = \{B_{11_{isvisible}}, B_{12_{isvisible}}, \ldots, B_{ij_{isvisible}}, \ldots, B_{ml_{isvisible}}\} \quad . \tag{5}$$

4.1.2 Point-Façade Association

To restore the correct scale of the reconstructed point cloud map, only the 3D points belonging to at least a single façade are relevant. We utilize the depth mask from the 2.5D map again given the current sensor pose $T_{sensor}(i)$ to filter out those points which do not belong to buidling façades .

Given the remaining 3D map points and the corresponding building façades, we calculate for the 2D normal distance d between each map point and the building façade it belongs to. Let $\mathbf{P} = \{P_1, P_2, ..., P_i, ..., P_n\}_{f_k}$, $f_k \in F$ be the set of 3D map points visible in key frame k, f_k, where f_k belongs to the collection of all key frames F. Further let p_i represents the orthographic mapping coordinates of P_i onto the ground plane, and we finally determine the closest façade and the smallest distance d following

$$\forall p_i, B_{ij_{visible}} = \underset{B_{ij_{visible}} \in \mathbf{B}_{isvisible}}{\arg\min} \quad \|d(p_i, B_{ij_{visible}})\|^2. \tag{6}$$

4.1.3 Iterative graph optimization

To reduce residual errors, our graph optimization uses each key frame and its associated map points as input. It includes geometric error for map points and reprojection error for the key frame pose.

Geometric Error We use unary edge graph optimization based on the g2o framework and perform non-linear minimization for every 3D map points of each key frame before point-façade association.

$$P_i = \underset{P_i}{\arg\min} \sum_{i=1}^{n} \rho \|Ax_i + By_i + Cz_i + D\|^2 \tag{7}$$

where $P_i = (x_i, y_i, z_i)$ is a 3D map point reconstructed by SLAM, and $Ax + By + Cz + D = 0$ represents a 3D building façade, $D = -(Ax_0 + By_0 + Cz_0)$, $P_{building} = (x_0, y_0, z_0)$ is a 3D point on the corresponding building façade $B_{ij_{visible}}$ according to the results of our point-façade association. Geometric error represents the first kind of edge to connect 3D map points in the graph optimization process.

Reprojection Error After optimizing the map points, we update the pose of key frame which can observes the map point. The map points modified by geometric error have a new position, therefore we update the pose of the corresponding key frames using the reprojection error. In the reprojection error function, K is denoted as the camera intrinsic parameters matrix. η_i are the coordinates of the associated feature point corresponding to map point i, P_i, ρ is the robust Huber cost function. Finally, T_{f_k} is the 6-DOF pose of key frame f_k:

$$T_{f_k} = \underset{T_{f_k}}{\arg\min} \sum_{i=1}^{n} \rho \|\eta_i - KT_{f_k}P_i\|^2 \tag{8}$$

The reprojection error in Eqn. 8 represents the second kind of edge to connect 3D map points and camera poses.

GPS Sensor Optimization Due to the continuous use of depth image information from the actual sensor pose estimate $T_{sensor}(i)$ in the alignment procedure, drift in the GPS information can be identified comparing the number of feature matches between subsequent frames. For pure purposes of visual validation, this is illustrated in Fig. 3.

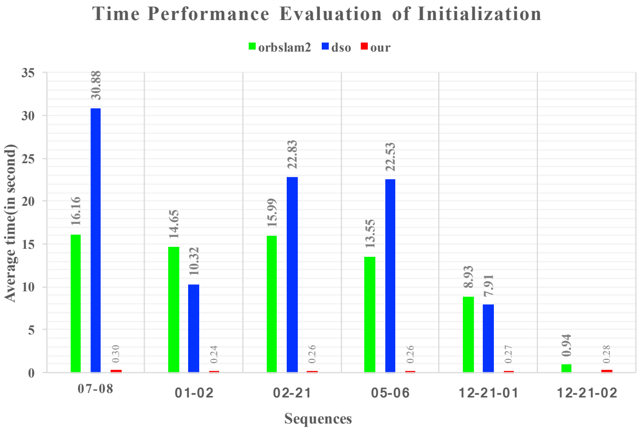


Figure 4: The time cost of initialization. Our method significantly improves the time performance of initialization in all challenging datasets.

In practice, a comparison of the area covered by the building masks, *i.e.,* a rapid change selfsame, triggers a change of the behaviour of our method. In this case we use the estimated pose from SLAM instead of the original sensor-based pose estimate in depth image creation.

Due to the inaccuracies of the sensor values, a weakly constraining error function to incorporate the sensor data at all times into the SLAM system is desirable:

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

$\Delta T_{slam}(i)$ denotes the relative pose between the frame $i-1$ and the frame i from SLAM. Similarly, $\Delta T_{sensor}(i)$ is the relative pose between frame $i-1$ and the frame i from sensor. Considering $T_{slam}(i-1)$ and $T_{slam}(i)$ as two vertices of a binary graph optimization problem, the difference between them serves as an optimal edge. In case the difference between the optimized result and the poses from SLAM exceeds a certain threshold, the optimization result is overall discarded.

5 Experiments

For evaluating our approach, we used a Microsoft Surface tablet equipped with additional sensors to collect one datasets including 5 sequences on the university campus. Each sequence includes RGB images, corresponding depth images created from the sensor pose estimates, inaccurate GPS sensor information and a 2.5D map of the regional environment. The resolution of the images was chosen to 640 × 360 pixels. All sequences include rapid motion and a comparatively small baseline w.r.t. the scene distance. The sequences were named 07-08, 01-02, 02-21, 05-06, and 12-21. Because the 12-21 sequences is very long, we randomly sliced it into two parts to create the sixth sequences. All the experiments were performed offline using a 3.2GHz Inter Core i5 iMac 2015. Some results for individual frames are shown in Figs.5 and 6.

5.1 Initialization Analysis

First, we compare the accuracy and computational performance of our initialization method to the standard ORBSLAM2 implementation on our scenarios.

Runtime Performance Comparison We tested each sequence 10 times using our initialization method, original ORBSLAM2 initialization and DSO (Direct sparse odometry) [11] respectively, recording the time (in s) until successful initialization. Fig. 4 depicts the results of this experiment. Due to the excessive feature matching, ORBSLAM2 requires a considerable number of frames and much more time. Due to the unstable lighting changes in the scene and other reasons, the initialization of DSO takes the most

Table 1: Comparison of initialization performance indices between ORBSLAM2 and our method. The following table records the number of matched points between map points and the following frame after initialization, as well as matches after optimization. The results prove our method can improve the overall system robustness.

Dataset	07-08		01-02		02-21		05-06		12-21-01		12-21-02	
Method	ORBSLAM2	Our	ORBSLAM2	Our	ORBSLAM2	Our	ORBSLAM2	Our	ORBSLAM2	Our	ORBSLAM2	Our
Matches	190	1049	320	337	199	607	227	500	235	667	112	628
Matches after optimization	170	1027	305	331	192	642	219	483	226	650	102	500

time compared to ours and ORBSLAM2, and even fails on the 12-21-02 sequence. Our method is able to complete the initialization procedure using only a single frame, which essentially means instant SLAM operation.

Robustness Comparison In order to verify the robustness of the proposed method, we record the average numbers of matched points between the map and the rest of the frames after initialization. The number of matches reflects the matching performance of initial map points with feature points of the subsequent frames.

From the results given in Tab. 1, our method retains a large number of map points from the initialization on all sequences comparing with the ORBSLAM2. After the optimization stage, a large number of these points are retained, in turn having a high number of observations with feature points in subsequent frames. The ORBSLAM2 method is challenged by the small baseline distance travelled. Therefore the initialization method adopting frame to frame matching leads to a considerably lower number of map points.

The GPS limitation and initialization In extreme cases, pose errors from regular GPS/IMU sensor devices are known to be up to 45deg in rotation and 40m in translation. In practice we experienced errors more in the range of several meters and up to 20 degrees around any axis. Our initialization procedure is able to cope with these errors, nevertheless success also depends on the given environment. On one hand, success or failure of our approach is related to the users operation. When users hold the camera device relatively still or avoiding rapid motion, our initialization is more likely to succeed. On the other hand, once the initialization is successful, the GPS/IMU errors are not crucial because these errors will be rectified in the subsequent optimization. Through our hybrid multi-modality graph-based optimization, we mainly rely on the constraints from the building model and the projection error to ensure the accuracy and scale of the SLAM system is correct. In case of GPS/IMU failure, the system is able to quickly re-initialize once the data from GPS/IMU is acceptable.

5.2 Reconstruction Map Comparison

In the left column of Fig. 7 we compare the reconstructed map using the original ORBSLAM2 approach with our initialization and optimization approach respectively.

The map reconstructed by the original ORBSLAM2 has no real scale information (green), hence, the scale is entirely wrong. In contrast, the maps reconstructed by our method (blue and red) are up to scale and are aligned to the real building models. Although the map reconstructed only by initialization method (blue) has absolute scale, misalignment errors are observable in certain areas. After using the proposed optimization method (red), the reconstructed map matches the real building map very well. The building façades have several large holes and architectural bulges in it, still feature points in the actual images are detected. These feature points cannot be closely aligned with the model of the façade.

5.3 Trajectory Comparison

As a next experiment, we compare the trajectory of the key frames during tracking given the data acquired from consumer-grade sensors (GPS and IMU), the original ORBSLAM2, our initialization and our initialization method including the optimization, respectively. The trajectories are visualized by differently colored discrete points in the right column of Fig. 7.

The camera trajectory estimated by ORBSLAM2 (green) has no scale information, in analogy to the respective reconstructed map. There is an obvious difference between the trajectory from consumer sensors (rose red) to the other methods, as some key frames have no corresponding GPS trajectory points associated at all. From the sensor trajectory, a serious drift problem due to the randomness and uncertainty in consumer sensors is observable.

5.4 Fixed-point Position Comparison

While there are no real ground-truth position measurements available for the entire trajectories, a set of geodetically accurately position measurements are available. Therefore when collecting the sequences, we deliberately chose the initial and final starting points of the sequences at these fixed points. Therefore, in the absence of ultra-precision GPS values, we are still able to give a hint on the accuracy of our initialization and tracking approach, comparing the difference between the camera translation estimated by our method at fixed points and the ground truth position at these points.

Unfortunately, third procedure did not work for all sequences. For the 02-21 sequence, the recorded GPS is overall very noisy, such that matching the fixed points was not possible. However, comparing the final camera position with fixed points on ground in 07-08 and 12-21 sequences, the Euclidean distance error was about 2 meters.

6 Discussion and Conclusion

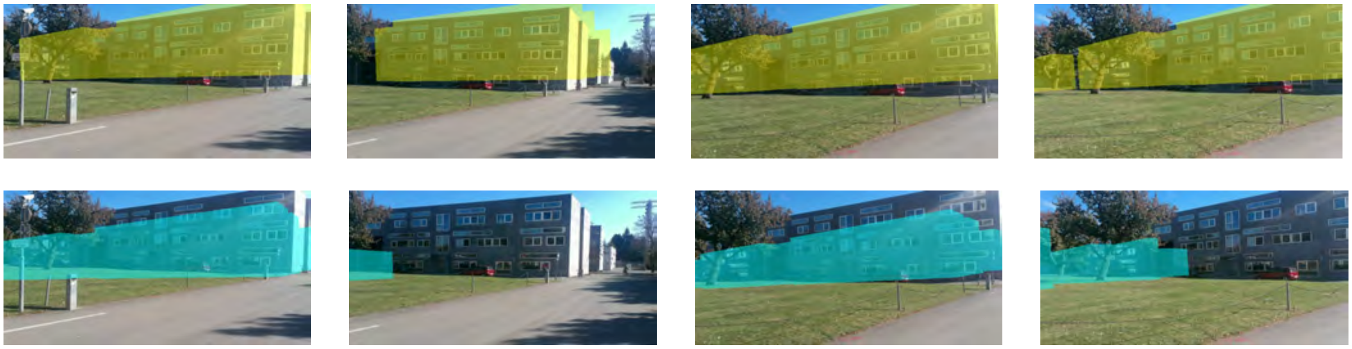
In this paper, we propose an instant initialization algorithm for feature-based monocular SLAM and a hybrid optimization scheme combing multi-modality data in an outdoor AR platform. Our single frame initialization provides an accurate metric camera registration, while our optimization method aligns a large-scale SLAM reconstruction and the associated camera trajectory based on easily-available 2.5D map and poor GPS sensor data. Comparing our method to results published in the state-of-the-art, our method performs plausibly in terms of accuracy, robustness and computational effort, at the benefit of being relatively easy to implement.

While our system has the potential to enable accurate outdoor AR visualization, there exists some limitations in our method. First, due to the complexity of the outdoor localization problem in general, there is a lack of available tools, framework and benchmarks. Our method is not available to work in all outdoor environments, however it is friendly to city scenes. Second, we assume the city environment is planar-world, the urban horizon generally does not fluctuate much, except for individual mountainous cities (*e.g.* Chongqing, China). Third, the feature detection approach used in SLAM is easily fouled by dynamic objects, such as cars or pedestrians, but also by static objects, such as trees, which are obviously not contained in the building depth masks. Any further extension of the map data used, such as information about other super-surface infrastructure or the use of adaptive models to identify dynamic content in imagery is expected to have a positive effect on the performance of the proposed system.

Acknowledgments

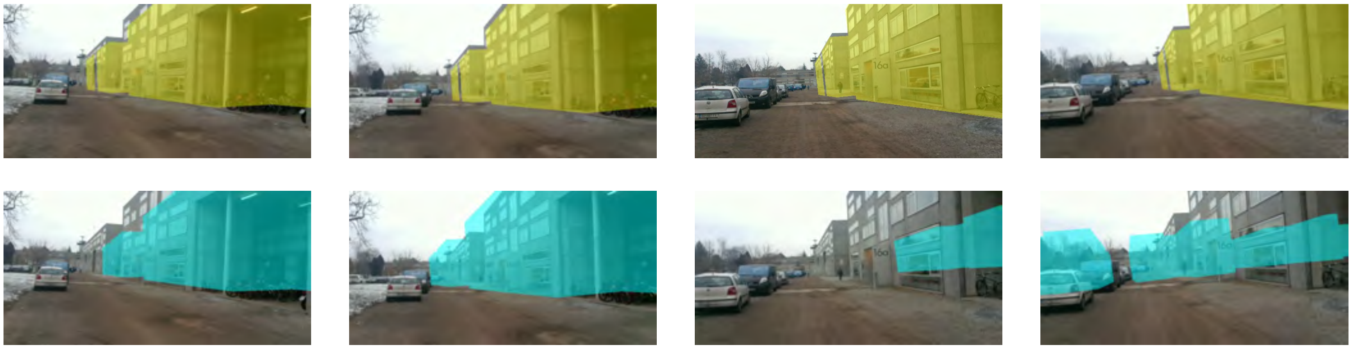
This work was partially supported by National Key R&D Program of China (2018YFB1305200). This publication was partially funded by the Austrian Research Agency (FFG) (No. 859208) and the National Natural Science Foundation of China (61876167 and U1509207).

The authors wish to thank Marco Stranner for providing the required hardware and for his support during data acquisition.



Frame #38 Frame #51 Frame #170 Frame #180

Figure 5: Visual comparison of model reprojection error for Sequence 07-08 using our SLAM-based method (top) and pure GPS and compass-based localization (bottom).



Frame #39 Frame #70 Frame #166 Frame #169

Figure 6: Visual comparison of model reprojection error for Sequence 12-21 using our SLAM-based method (top) and pure GPS and compass-based localization (bottom).

References

[1] A. Armagan, M. Hirzer, P. M. Roth, and V. Lepetit. Learning to align semantic segmentation and 2.5 d maps for geolocalization. In *CVPR*, pp. 3425–3432, 2017.

[2] C. Arth, A. Mulloni, and D. Schmalstieg. Exploiting Sensors on Mobile Phones to Improve Wide-Area Localiz. *ICPR*, pp. 2152–2156, 2012.

[3] C. Arth, C. Pirchheim, J. Ventura, D. Schmalstieg, and V. Lepetit. Instant outdoor localization and slam initialization from 2.5 d maps. *TVCG*, 21(11):1309–1318, 2015.

[4] C. Arth, D. Wagner, M. Klopschitz, A. Irschara, and D. Schmalstieg. Wide area localization on mobile phones. *ISMAR*, pp. 73–82, 2009. doi: 10.1109/ISMAR.2009.5336494

[5] R. T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997.

[6] D. M. Chen, G. Baatz, K. Kser, S. S. Tsai, R. Vedantham, T. Pylvninen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk. City-scale landmark identification on mobile devices. In *CVPR*, pp. 737–744, June 2011. doi: 10.1109/CVPR.2011.5995610

[7] Z. Chen, V. Badrinarayanan, G. Drozdov, and A. Rabinovich. Estim. depth from rgb and sparse sensing. In *ECCV*, pp. 167–182, 2018.

[8] K. Choi, J. Park, Y.-H. Kim, and H.-K. Lee. Monocular slam with undelayed initialization for an indoor robot. *Robotics and Autonomous Systems*, 60(6):841–851, 2012.

[9] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *PAMI*, (6):1052–1067, 2007.

[10] J. Domínguez-Conti, J. Yin, Y. Alami, and J. Civera. Visual-inertial slam initialization: A general linear formulation and a gravity-observing non-linear optimization. In *ISMAR*, pp. 37–45. IEEE, 2018.

[11] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *PAMI*, 40(3):611–625, 2017.

[12] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. doi: 10.1145/358669.358692

[13] A. Fond, M.-O. Berger, and G. Simon. Facade proposals for urban augmented reality. In *ISMAR*, pp. 32–41. IEEE, 2017.

[14] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[15] T. Henkey. *Urban Emergency Management: Planning and Response for the 21st Century*. Elsevier Science, 2017.

[16] J. Huang, R. Liu, J. Zhang, and S. Chen. Fast initialization method for monocular slam based on indoor model. In *IEEE Int. Conference on Robotics and Biomimetics (ROBIO)*, pp. 2360–2365. IEEE, 2017.

[17] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. *CVPR*, pp. 2599–2606, June 2009. doi: 10.1109/CVPR.2009.5206587

[18] B. Jiang, U. Neumann, and S. You. A robust hybrid tracking system for outdoor augmented reality. In *VR*, pp. 3–275. IEEE, 2004.

[19] O. Kähler, V. A. Prisacariu, and D. W. Murray. Real-time large-scale dense 3d rec. with loop closure. In *ECCV*, pp. 500–516, 2016.

[20] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. *ISMAR*, 2007.

[21] G. Klein and D. Murray. Parallel tracking and mapping on a camera phone. *ISMAR*, pp. 83–86, 2009.

[22] S. B. Knorr and D. Kurz. Leveraging the user's face for absolute scale est. in handheld monocular SLAM. In *ISMAR*, pp. 11–17. IEEE, 2016.

[23] Y. Kuznietsov, J. Stuckler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. In *CVPR*, pp. 6647–6655, 2017.

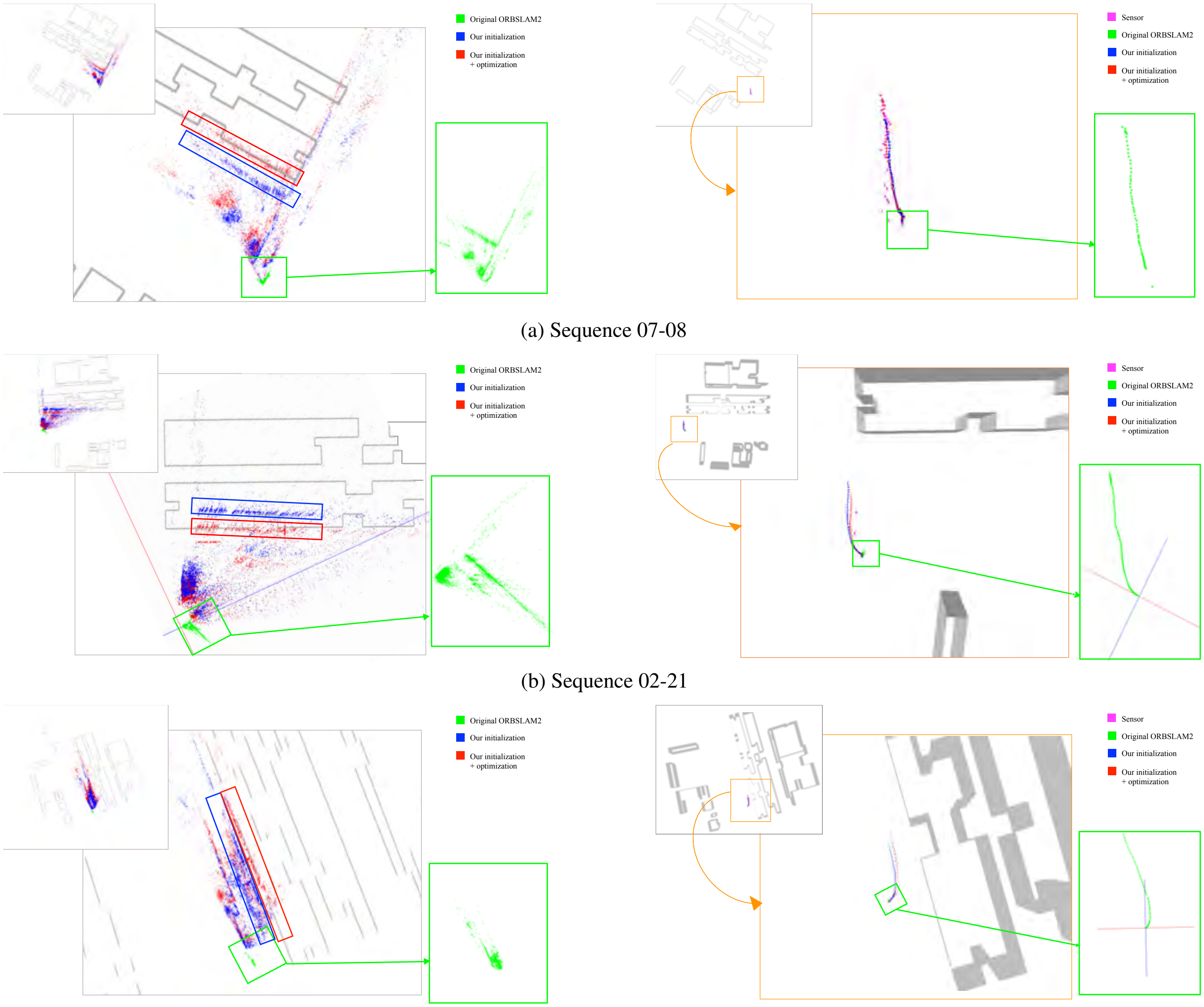
[24] D. Larnaout, V. Gay-Belllile, S. Bourgeois, and M. Dhome. Vision-based differential gps: Improving vslam/gps fusion in urban environment with 3d building models. In *Int. Conf on 3D Vision*, vol. 1, pp. 432–439. IEEE, 2014.

[25] P. Li, T. Qin, B. Hu, F. Zhu, and S. Shen. Monocular visual-inertial State Estimation for Mobile AR. In *ISMAR*, pp. 11–21. IEEE, 2017.

[26] P. Lothe, S. Bourgeois, F. Dekeyser, E. Royer, and M. Dhome. Towards geographical referencing of monocular slam reconstruction using 3d city models: Application to real-time accurate vision-based localization. In *CVPR*, pp. 2882–2889. IEEE, 2009.

[27] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[28] R. Mur-Artal and J. D. Tardós. Orb-slam2: An open-source slam



(a) Sequence 07-08

(b) Sequence 02-21

(c) Sequence 12-21

Figure 7: **Left column:** Comparison of the individual reconstructed map and the remaining gap between the reconstructed maps and the real 2.5D building models for the three methods. The red box marks the reconstructed building façade which is fixed to the real building using our initialization and optimization methods. The green one is the result from original ORBSLAM2 method which has unknown scale and orientation. **Right column**: Comparison of the camera trajectory for different sources. Our method restores the continuous actual trajectory(red), while the original ORBSLAM2 method has no scale (green), and the sensor-based pose estimate exposes serious drift problems (rose red).

system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[29] R. Mur-Artal and J. D. Tardós. Visual-inertial monocular slam with map reuse. *IEEE Robotics & Automation Letters*, 2(2):796–803, 2017.

[30] R. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. *ISMAR*, pp. 127–136, Oct. 2011. doi: 10.1109/ISMAR.2011.6092378

[31] D. Nistér. An efficient solution to the five-point relative pose problem. *PAMI*, 26(6):0756–777, 2004.

[32] T. Qin, P. Li, and S. Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[33] G. Reitmayr and T. W. Drummond. Initialisation for visual tracking in urban environments. In *ISMAR*, pp. 161–172. IEEE, 2007.

[34] G. Schall. *Mobile Augmented Reality for Human Scale Interaction with Geospatial Models: The Benefit for Industrial Applications*. Mobile Computing. Springer Fachmedien Wiesbaden, 2012.

[35] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *CVPR*, pp. 1–7, June 2007. doi: 10.1109/CVPR.2007.383150

[36] J. Ventura, C. Arth, G. Reitmayr, and D. Schmalstieg. Global localization from monocular SLAM on a mobile phone. *TVCG*, 2014.

[37] L. Von Stumberg, V. Usenko, and D. Cremers. Direct sparse visual-inertial odometry using dynamic marginalization. In *ICRA*, pp. 2510–2517. IEEE, 2018.

[38] C.-P. Wang, K. Wilson, and N. Snavely. Accurate georegistration of point clouds using geographic data. In *3DV*, pp. 33–40, 06 2013. doi: 10.1109/3DV.2013.13

[39] Y. Zhang and T. Funkhouser. Deep depth completion of a single rgb-d image. In *CVPR*, pp. 175–185, 2018.