

Vision-based Global Localization of Unmanned Aerial Vehicles with Street View Images

Xuejiao Yan, Zongying Shi, Yisheng Zhong

Department of Automation, Tsinghua University, Beijing 100084, P. R. China
E-mail: yanxj15@mails.tsinghua.edu.cn, szy@mail.tsinghua.edu.cn, zys-dau@mail.tsinghua.edu.cn

Abstract: In this paper, a vision-based method is proposed to globally localize an unmanned aerial vehicle (UAV) flying in the urban scenarios using Google Street View as prior information. Since the trajectory of the UAV with absolute scale cannot be recovered with the help of monocular visual odometry only, we leverage the known positional information of Google Street View to localize vehicles with respect to the surrounding buildings. First, a place recognition algorithm is presented to solve air-ground matching problem by searching aerial images taken from onboard camera within the geotagged image database. Then based on the matched images by place recognition and tracked features via visual odometry, a novel absolute scale estimation mechanism is proposed to estimate the pose of UAV in the real world coordinate system and to correct the scale drift of visual odometry. Furthermore, the pose graph with UAV's poses as vertexes and relative transformations as edges is optimized with SE(3) constraints to get trajectory of UAV in the real world coordinate system. Experiments are conducted on a 2km dataset recorded onboard by a camera-equipped UAV flying in the urban streets of Zurich, Switzerland, and the results show that our method is able to metrically localize the UAV without previously visiting or building 3D models in advance.

Key Words: Vision-based localization, unmanned aerial vehicle, Google Street View, absolute scale estimation

1 Introduction

In recent years, there has been a growing interest in research on autonomous visual navigation of robots. Vision-based methods have advantages that the camera is cheaper, safer and more compact compared with other active sensors such as laser range finder, and it is more reliable than GPS when robots in dense downtown cores where GPS signals might be blocked. There are numerous works on vision-based navigation of both unmanned ground vehicles (UGVs) and unmanned aerial vehicles (UAVs). During the past years, rapid progress in navigation of UGVs leads to striking new technologies like self-driving which can map and track vehicles in uncertain street environments with cameras [2]. As for UAVs, one approach is to use a set of external cameras to track the vehicle, which cannot be applicable to navigation in large outdoor environment [3]. Alternatively, some researchers make use of onboard cameras for landmark detection with requirements of prior knowledge of landmarks [4]. In order to estimate the pose of vehicles as well as the position of visual landmarks especially in unknown environments, visual simultaneous localization and mapping (vSLAM) or visual odometry (VO) are proposed and widely applied [5, 7].

One of the most important research subjects of vision-based navigation is outdoor navigation for its various application in people's daily life. Outdoor navigation faces more challenges such as changes of illumination. To solve the problem of outdoor navigation, image-based localization with monocular camera has been studied over many years. The image retrieval technique is often applied to image-based localization to recognize the query location. Street View images are usually used to form image database for querying, since they are public and easy to be obtained. Considering the problem of localizing a camera-equipped UAV flying in the outdoor urban streets at low altitudes (i.e. 10-20

m from the ground), we use Google Street View images for providing prior information. The image retrieval-based localization can be categorized as localization recognition and fine localization methods [8]. We review some recent works of each category as follows.

In the first category, the problem of image matching is studied to find the best match of images captured by moving cameras in the street view images database. In [10], the authors extract SIFT descriptors from 100,000 Google Street View images to establish trees which are then searched by a nearest-neighbor method. Zheng Wang et. al [11] propose a faster scheme for partitioning geo-tagged reference image database by constructing a topic model to divide the reference dataset into scene groups, and the query image is localized by searching the visually similar images in each scene group using fast library for approximate nearest neighbors (FLANN) [22]. A memory scalable image localization system is presented in [12], which uses distributed kd-trees created on overlapping geographic cells. In [14], the method utilize GPS module along with inertial sensors to calculate the Estimated Position Error (EPE) to limit the number of candidate images and then all those of candidate images are fed to a homography verification algorithm to find the best match feature in the database. Majdik et al. [15] use street view images to localize a UAV by matching images obtained from onboard camera to Street View images which called air-ground matching. Their key contribution is matching images with strong view point changes using Affine-SIFT (ASIFT) [6]. Conversely, the problem of matching images captured from a UGV to satellite imagery is studied to localize the vehicle, which called Ground to Satellite Matching [16]. They warp the ground-based panoramas to form the top-down view of the scene that are compared to a grid of satellite locations with Bayesian Localization framework. The methods described above just solve the problem of place recognition and roughly locate the vehicles in the outdoor environment.

In the second category, a few methods are proposed based

This work is supported by National Natural Science Foundation (NNSF) of China under Grants 61374034 and 61210012.

on place recognition. 2DTriPnP [8] is a 2D and robust fine localization method using Google street view. It utilizes the projective geometry in two dimensions and solves nonlinear equations to estimate the location of a query image. In [17], the authors formulate the image-localization problem as a regression on an image graph with geo-tagged images as nodes and edges connecting close-by images. With linear combination of image feature vectors along edges, they find subset of geo-tagged images database and predict the query location by interpolating locations of matched images in the graph. The two methods mentioned above merely provide fairly accurate 2D positions. Pratik et.al [18] model the problem of fine localization as a non-linear least squares estimation in two phases. They estimate 3D position of points in the environment from a short monocular trajectory and then find matched panoramas to compute the 6DOF transformation between the moving camera imagery and the geo-tagged rectilinear panoramic views. The problem of absolute scale is solved by combining visual odometry and IMUs to generate accurate visual-inertial odometry (VIO). 3D prior information of surroundings are also used to localize vehicles. A three-step approach for global metric localization of 3D vehicle trajectories from dashcam videos in the wild is proposed by ShaoPin Change et. al [19]. In order to get global metric localization, they align the 3D trajectory recovered from Structure from Motion (SFM) to a global metric 3D map created from RGBD Panoramic Google Street View. In [20], street Views panoramas with their depth map offline are trained as a Bag of Words (BoW) database and metric urban localization is completed by matching 3D Street View points to a moving monocular camera. Besides, Majdik et al. extend their work on place recognition [15] by establishing 3D models of buildings using Google Street View to get the trajectories of UAV [1].

Accurate localization is extremely important for UAVs to complete tasks, such as aerial photography, goods delivery and telepresence in the case of accidents. In this paper, we deal with the problem of localizing the poses of UAVs equipped with a monocular camera when flying in urban streets at low altitude. Google Street Views are considered as a database of accurate geo-tagged imagery to provide prior information. Our work belongs to fine localization which is complementary to those place recognition algorithms. Unlike works in [1, 19, 20], we are not interested in using the panoramas to build an accurate large 3D model of the environment in advance. Instead of getting 2D positions only, we aim at recovering 6DoF poses of UAVs without building new large scale maps where Street View exists. Our method consists of two steps. Firstly, a place recognition algorithm is presented to solve air-ground matching problem [1]. Secondly, a novel absolute scale estimation mechanism is proposed to recover the poses of UAV in the real world coordinate system based on the matched images, and map points and camera poses via visual odometry. Furthermore, the trajectory is optimized with SE(3) constraints.

The main contributions of this paper are as follows: (1) an effective vision-based global localization method is presented to estimate the poses of UAVs relative to the surrounding buildings and recover the trajectory of vehicles in the real world without building cadastral 3D models or land-

marks; (2) an absolute scale estimation mechanism is proposed to estimate the absolute scale and correct the accumulated drift and scale drift induced by monocular visual odometry.

The rest of the paper is organized as follows. Section 2 overviews the framework of our method. Section 3 introduces the place recognition algorithm about air-ground matching problem. Section 4 describes the metric localization method for UAV in detail, including tracking features in an image sequence, absolute scale estimation and trajectory optimization. Section 5 presents the experimental results. Finally conclusions are given in Section 6.

2 Overall Framework

Our motivation is to create a vision-based localization method for the UAV flying in the urban scenarios with the help of Google Street View images. In fact, we formulate localization as the problem of estimating the position of street view images relative to monocular image sequences obtained from the onboard camera of UAV. Our method only makes use of the monocular camera and visual odometry estimation, and it consists of two steps: place recognition and metric localization.

Firstly, the global positions of the UAV are recovered by recognizing visually-similar discrete places in a Google Street View database. Namely, the image of buildings in the streets captured by the UAV is searched in the database of geo-tagged pictures. Since geo-tagged images are usually obtained by the cameras installed on the ground vehicle, they cannot be used directly for UAV localization. This problem is called air-ground matching due to the large difference in viewpoint between the air-level images from UAV and ground-level images [1]. In addition to common challenges of vision-based localization such as the large changes of illumination, lens distortion and season variation of vegetation, the extreme changes in viewpoint and scale makes air-ground matching problem more difficult to be solved. The matching problem is settled by generating virtual views just like [1], and a coarse-to-fine scheme with histogram-voting is proposed.

Secondly, the 3D positions of map points in the environment are estimated from an aerial image stream using ORB-SLAM without loop closing thread [7], and the 6DoF transformation of matched street view images with respect to the camera trajectory are computed with PnP-RANSAC. Since the GPS positions of Google Street Views are known, the absolute scale of visual output and the onboard camera positions relative to the positions of street view images can be estimated. After pose graph optimization with SE(3) constraints, the trajectory of UAV in the real world coordinate system can be recovered. Fig. 1 illustrates the structure of our method.

Our method is to metrically locate UAV and recover its trajectory in the real world coordinate system with prior information. In this work, the construction of a consistent map and previously visiting the environment are not required, which means this method is not to compute the loop closures for environment reconstruction. Experiments are conducted on the dataset provided by Robotics and Perception Group

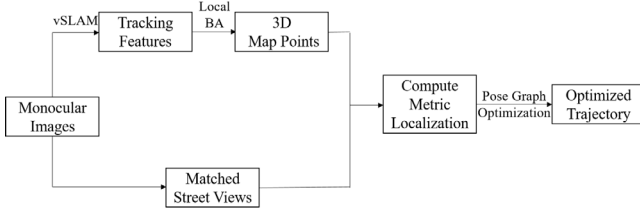


Fig. 1: Structure of our method

from University of Zurich¹. The dataset includes aerial UAV images along a 2km trajectory and 113 Google Street View images covering the area where UAV flies.

3 Place Recognition: Air-ground Matching of Images

In this section, the place recognition algorithm about air-ground matching problem is described in details. In order to match monocular images from UAV with Google Street Views accurately and quickly, a coarse-to-fine scheme is proposed for match selection. We constitute a candidate set of Google Street View images with a histogram-voting mechanism at first and then select the best match from the candidates, which means visually-similar discrete places is recognized in a Google Street View database. A pseudo-code description is given in Table 1. Please note that line 1 of the algorithm is computed off-line. In this step, Google Street View images $I = \{I_1, I_2, \dots, I_n\}$ are converted into image features using ASIFT descriptors and are saved in a database D_T .

Table 1: Place Recognition Algorithm

Input: An aerial image I_a taken by onboard camera of UAV
Output: I_a 's best match I_b in Google Street View database
$I = \{I_1, I_2, \dots, I_n\}$
1. D_T = database of all the image ASIFT features of I
Train D_T using FLANN
2. F_a = extract ASIFT image features of I_a
3. Search approximate nearest neighbor feature matches for F_a in D_T : $M_D = ANN(F_a, D_T)$
4. Select c images as a candidate set $I^P \subseteq I$ using histogram-voting scheme $I^P = \{I_1^P, I_2^P, \dots, I_c^P\}$
5. Run in parallel for j from 1 to c
{ Search approximate nearest neighbor feature matches for F_a in F_j^P : $M_j = ANN(F_a, F_j^P)$;
Select inlier points: $N_j = kVLD(M_j, I_a, I_j^P)$ }
6. $I_b \leftarrow \max(N_1, N_2, \dots, N_c)$
7. return I_b

In the step of coarse match selection, the purpose is selecting a fixed number of candidate images $I^P = \{I_1^P, I_2^P, \dots, I_c^P\}$. First of all, we need represent images with appropriate feature detector and descriptor. There are many common point feature detectors and descriptors, such as SIFT, BRISK etc. Although these detectors and descriptors are invariance to rotation and scale to some extent, they may fail in the case of substantial viewpoint changes. Aiming at solving air-ground matching problem, our approach makes use of ASIFT descriptors to generate virtual views

and extract features. ASIFT is a method to simulate all image views obtained by varying the two camera-axis orientation parameters [6], i.e. the latitude angles (θ) and longitude angles (ϕ) shown in Fig 2. Thus the definition of tilt is $tilt = \frac{1}{\cos \theta}$. ASIFT detector and descriptor are realized by sampling various values for the tilt and longitude angle ϕ to compute virtual views of the image. Next, SIFT features are detected on the original and artificially-generated images. Considering the air-ground geometry, we limit the number of tilt values and just use three virtual simulations, i.e. at 0° and $\pm 40^\circ$ just as presented in [1].

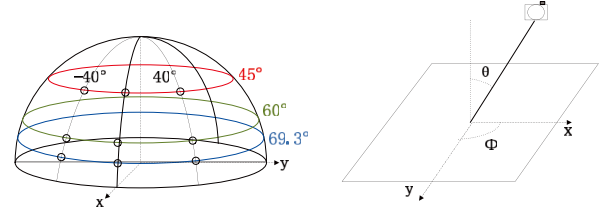


Fig. 2: Sampling parameters for virtual view generation

After extracting ASIFT features of the aerial image I_a , the approximate nearest neighbors for all the image features extracted from I_a and its virtual views are searched using FLANN [22] for the purpose of rapidity. Next, a histogram-voting scheme is utilized to find a fixed number of specific Google Street View images that indicates the most features with the same angular change. A two-dimensional histogram is built, in which each bin contains the number of approximate nearest neighbors of features that count for different virtual views in a certain image in the database. Finally, we can select top c database images to form a candidate subset.

Having selected c candidate images $I^P = \{I_1^P, I_2^P, \dots, I_c^P\}$, we look for the best match for the UAV image in the step of fine match selection. The search process can be completed in parallel. As shown in Table 1, the approximate nearest neighbor of every feature of aerial image F_a is searched in each candidate I_j^P . In order to select inlier points and correctly match features, Virtual Line Descriptor (kVLD)[9] and epipolar constraints are used instead of traditional RANSAC-based approaches. The candidate image that has maximum number of matched inlier points is considered as the best match for aerial image.

The vision-based place recognition algorithms are evaluated in terms of recall rate and precision rate. One of the most important evaluation criteria for place recognition is recall rate at precision 100%, which represents the true positive detections over the total number of possible matches detected [15]. Our place recognition algorithm is tested on the dataset¹, and the recall rate is 48.6% at precision 100%. The confusion matrix² is used as another visualization to show the result, as shown in Fig 3. Here, the confusion matrix illustrates the visual similarity between all the aerial images from UAV camera and all the Google Street View images. We make use of heat maps to display the confusion matrix. In Fig 3, the white color represents no visual similarity, while the blue or red color is complete similarity. Thereinto, the blue is for the ground truth and the red denotes the results of

¹Dataset is download from <http://rpg.ifi.uzh.ch>.

²https://en.wikipedia.org/wiki/Confusion_matrix

our method. An ideal place recognition algorithm would detect a confusion matrix coincident to the ground truth. From Fig 3, the result of our method is generally consistent with ground truth matrix without strong deviation, namely all the recognized places are correct except some places do not be matched with the database images. Our method plays a good performance.

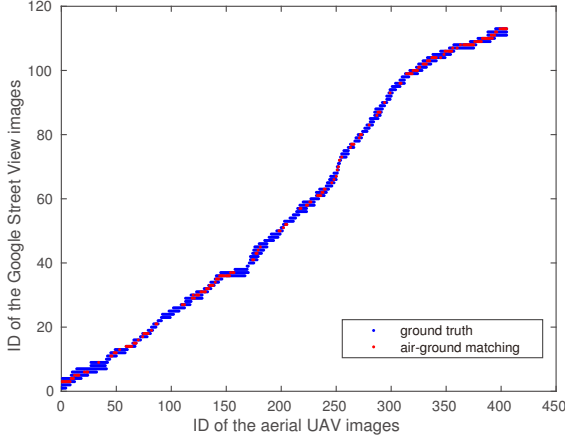


Fig. 3: Confusion matrix of our method and ground truth

4 Metric Localization of UAV

This section illustrates how to estimate 6DoF poses of flying vehicle in the global coordinate system on the basis of the place recognition algorithm presented in the previous section. Avoiding building cadastral 3D models or landmarks in advance like [1, 19, 20], we use the tracking and mapping threads of ORB-SLAM [7] to track features in an image sequence taken by onboard camera to get 3D positions of map points, propose absolute scale estimation mechanism and optimize trajectory with SE(3) constraints on pose graph.

4.1 Feature Tracking in Image Sequence

The tracking and mapping threads of ORBSLAM [7] is utilized as visual odometry to get 3D map points and the poses of UAV. The input is an aerial image stream $S = \{s_1, s_2, \dots, s_m\}$ obtained from a monocular camera. Please note that the 3D positions of map points are in the SLAM world coordinate system, not in the real world coordinate system. SLAM world coordinate system does not have the absolute scale due to the characteristics of monocular problem, while the real word coordinate system is with the absolute scale. Besides, the trajectory of UAV does not have an absolute scale since its a monocular problem.

ORB features is chosen to represent images. At first, the relative pose between two images is computed by triangulating an initial set of map points to create initial map. It's achieved by extracting ORB features in the current image searching for matches in the reference image and computing in parallel two geometrical models, namely a homography and a fundamental matrix. Finally a heuristic is used to select a model to recover the relative pose. After map initialization, the tracking thread is in charge of localizing camera with every image and deciding when to insert a new keyframe. Once tracking was successful for last image s_{i-1} , a constant

velocity motion model is utilized to predict the camera pose of s_i and build a local visible map with the help of the covisibility graph of keyframes. Then the map is reprojected into the image s_i to search more map point correspondences, and camera pose is optimized with all matches. Finally the tracking process decides whether a new keyframe is inserted. In the local mapping process, local Bundle Adjustment (BA) is applied to achieve an optimal reconstruction of map points and camera poses after keyframe insertion, recent map points culling and new map point creation. Fig 4 shows an overview of tracking and mapping threads of ORB-SLAM system.

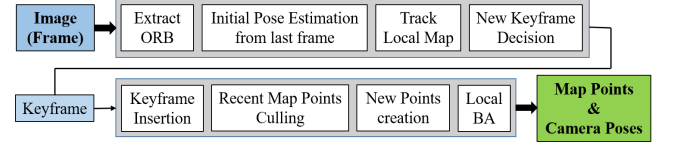


Fig. 4: Tracking and mapping threads of ORB-SLAM

Once the best match g_j of an image s_i in the Google Street View database can be found, we have 2D coordinates of matched points in Google Street View image and 3D position of map points from tracking. Then PnP-RANSAC is applied to estimate the relative pose between matched Google Street View image and image s_i , which means the pose T_{js} of Google Street View image g_j in the SLAM world coordinate system without absolute scale can be recovered.

4.2 Absolute Scale Estimation Mechanism

The pose of onboard camera estimated by visual odometry must be converted in metric unit, so absolute scale should be estimated with the help of known GPS positions of matched GPS Street View images. The iterative method presented in [21] is used to compute the absolute scale.

When two different Google Street View images are matched with aerial images taken by UAV camera, we can get the distances between the two places both in SLAM world coordinate system d_{si} and in real world coordinate system d_{gi} . As UAV flying, we obtain a set of pairs $\{(d_{s1}, d_{g1}), (d_{s2}, d_{g2}), \dots, (d_{sn}, d_{gn})\}$. However, both measurements are inaccurate due to measurement noise. So approximately there is $d_{si} \approx \lambda d_{gi}$, where λ is the absolute scale. Assume the measurement noise is Gaussian with constant variance, namely

$$\begin{aligned} d_{gi} &\sim N(\mu_i, \sigma_g^2) \\ d_{si} &\sim N(\lambda\mu_i, \sigma_s^2) \end{aligned} \quad (1)$$

where $\mu_1, \dots, \mu_i$ denote true but unknown distances and σ_g^2, σ_s^2 are the variances of the measurement errors.

The problem of scale estimation can be considered as a maximum likelihood estimation. One approach to solve this problem is to minimize the following negative log-likelihood

$$\mathcal{L}(\mu_1, \mu_2, \dots, \mu_n, \lambda) \propto \frac{1}{2} \sum_{i=1}^n \left(\frac{(d_{si} - \lambda\mu_i)^2}{\sigma_s^2} + \frac{(d_{gi} - \mu_i)^2}{\sigma_g^2} \right) \quad (2)$$

By solving $\partial \mathcal{L} / \partial \mu_i = 0$, $\partial \mathcal{L} / \partial \lambda = 0$, we can get a unique and global minimum

$$\lambda^* = \frac{h_{ss} - h_{gg} + \sqrt{(h_{ss} - h_{gg})^2 + 4h_{sg}^2}}{2\sigma_s^{-1}\sigma_g h_{sg}} \quad (3)$$

where

$$\begin{cases} h_{ss} = \sigma_g^2 \sum_{i=1}^n d_{si}^2 \\ h_{gg} = \sigma_s^2 \sum_{i=1}^n d_{gi}^2 \\ h_{sg} = \sigma_s \sigma_g \sum_{i=1}^n d_{si} d_{gi} \end{cases}$$

When a new pair of measurements comes, the scale λ^* is updated by formula (3).

If the number of measurement pairs is small, the estimation of scale is not converged. Besides, scale drift exists with monocular problem. So a scale-chosen mechanism is proposed. In each iteration, we record the scale λ_i , and compute average λ_{aver} and standard deviation σ_λ^i of every three scales $\{\lambda_{i-2}, \lambda_{i-1}, \lambda_i\}$. Two threshold values are settled, i.e. r_{in} and r_{out} . If $\frac{\sigma_\lambda^i}{\lambda_{aver}} \leq r_{in}$, λ_i is chosen as the absolute scale λ^* . When $\frac{\sigma_\lambda^j}{\lambda_{aver}} \geq r_{out}$, we consider that the scale has changed, that is to say the scale between iteration i and j can be considered basically invariant. The position of UAV camera between iteration i and j given by VO is scaled to λ_i . Then restart to compute the scale from $j+1$ and another absolute scale is estimated if $\frac{\sigma_\lambda^m}{\lambda_{aver}} \leq r_{in}$ ($m \geq j+1$). We call this process scale-reset.

To demonstrate the scale estimation mechanism, we conduct an experiment on the dataset¹. We set $r_{in} = 0.04$ and $r_{out} = 0.02$. The values of λ^* and $\frac{\sigma_\lambda^i}{\lambda_{aver}}$ in each iteration are depicted in Fig 5, where green dashed line represents the moment when $\frac{\sigma_\lambda^i}{\lambda_{aver}} \leq r_{in}$, which means we can consider the scale as absolute scale, and red dashed line denotes the moment when $\frac{\sigma_\lambda^j}{\lambda_{aver}} \geq r_{out}$, where the scale starts to change.

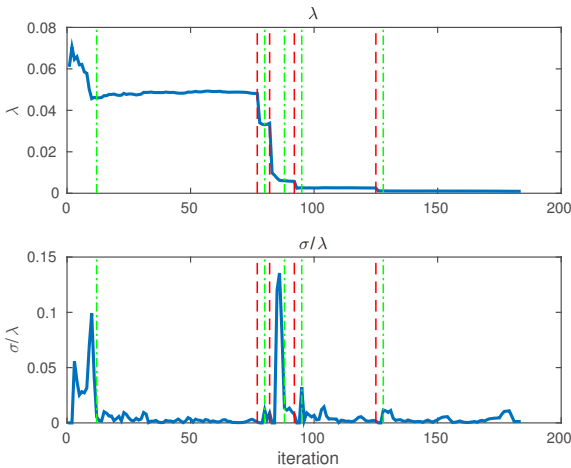


Fig. 5: Absolute scale λ^* and σ_λ/λ in each iteration

With the help of the scale estimation mechanism, the scaled positions of UAV in metric unit is recovered.

4.3 Trajectory Optimization and Global Metric Localization

As the positions of UAV with the absolute scale is recovered, these positions can be converted to the real world coordinate system. To localize the UAV in the real world co-

ordinate system, computing the relative pose T_{gs} between the real word coordinate and SLAM world coordinate system with absolute scale is indispensable. The pose in SLAM world coordinate frame T_{js} can be recovered from vSLAM, and then can be converted in metric unit with absolute scale using scale estimation scheme, which is denoted as T_{js}^λ . As GPS positions of the matched Google Street View images and onboard attitude data are known, the pose in the real word coordinate system T_{gj} can be obtained. T_{gs} is computed by $T_{gs} = T_{gj} \cdot T_{js}^\lambda$. According to the absolute scale estimation mechanism, T_{gs} in different scales is computed when $\frac{\sigma_\lambda^i}{\lambda_{aver}} \leq r_{in}$. Finally, the trajectory of UAV in the real world can be recovered.

Due to the absolute scale estimation mechanism designed to solve scale-drift problem, there is a phenomenon we called scale-abruption. If $\frac{\sigma_\lambda^i}{\lambda_{aver}} \leq r_{in}$ and $\frac{\sigma_\lambda^j}{\lambda_{aver}} \geq r_{out}$, the poses between iteration i and j are scaled to λ_i . Then the scale should be recalculated from iteration $j+1$ and another absolute scale will be gotten if $\frac{\sigma_\lambda^m}{\lambda_{aver}} \leq r_{in}$ ($m \geq j+1$). As a result, scale-abruption is mainly reflect in those poses between iteration j and iteration m with unknown scale as shown in Fig 6. So a pose graph optimization is performed over SE(3) constraints to distribute the scale error along the whole trajectory.

Given a pose graph of binary edges, the error in an edge can be defined as:

$$e_{ij} = \log_{SE(3)} (T_{ij} T_{j\omega} T_{i\omega}^{-1}) \quad (4)$$

where $T_{j\omega}, T_{i\omega}$ are the poses of UAV in trajectory, and T_{ij} is the relative SE(3) transformation between $T_{j\omega}$ and $T_{i\omega}$ just before the pose graph optimization. The $\log_{SE(3)}$ [13] transforms to the tangent space, so that the error is a vector in $\mathbb{R}^6$. The goal is to optimize the SE(3) poses in the trajectory by minimizing the cost function:

$$C = \sum_{i,j} e_{ij}^T \Lambda_{ij} e_{ij} \quad (5)$$

where Λ_{ij} is the information matrix of the edge, which we set to the identity.

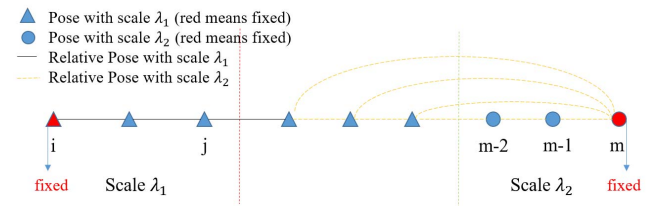


Fig. 6: Vertices and edges in SE(3) optimization

Applying g2o frame to optimize pose graph, the setting of vertices and edges is illustrated in Fig 6. After pose graph optimization, the optimized trajectory of UAV in the real word coordinate system can be recovered.

5 Experimental Results

To demonstrate the feasibility of our visual global localization method, experiments are carried out on the dataset¹ recorded onboard by a camera-equipped UAV flying in the urban streets of Zurich, Switzerland, at low altitudes (10-20 meters above the ground). This 2km dataset, which includes

time synchronized aerial images, GPS and IMU sensor data, ground Google Street View images and ground truth data, can be considered as a benchmark to evaluate appearance-based localization. The real world coordinate system as reference frame is International WGS 84/UTM zone 32N coordinate system. Since the coordinate values in the International WGS 84 / UTM zone 32N are very large, we convert the real world coordinates to local coordinate system to depict trajectories for better illustration, as shown in Fig 7 and Fig 8.

Firstly, ORB-SLAM is tested on the dataset without using Google Street View information, and the ground truth path (in red) and the UAV positions estimated by ORB-SLAM (in blue) are shown in Fig 7. In fact, the poses estimated by monocular vSLAM don't have absolute scale, so these poses are unified to the same scale as the ground truth to make a comparison. As shown, the estimated positions by ORB-SLAM are inaccurate, especially in the latter half of the trajectory as a result of scale drift and lack of loop closure. The trajectory in the real word coordinate system cannot be recovered without absolute scale and scale drift is difficult to be corrected using vSLAM only.

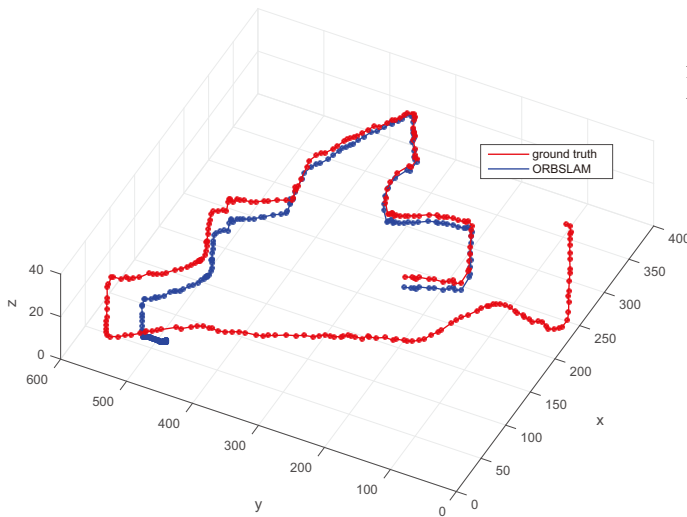


Fig. 7: Comparison between ground truth path and trajectory estimated by ORB-SLAM

Then the global positions of UAV with absolute scale and the trajectory optimized with applying SE(3) constraints to pose graph are recovered with absolute estimation mechanism and optimization method we propose. Fig 8 depicts unoptimized trajectory (in blue) with absolute scale estimation only, optimized trajectory (in green) with SE(3) optimization over pose graph and ground truth path (in red).

From Fig 8, we can see that optimizing pose graph with SE(3) constraints can distribute the scale error along the trajectory when comparing unoptimized trajectory and optimized trajectory. The "scale-abruption" is eliminated after optimization. This figure shows how the pose graph optimization improves localization over the unoptimized. The errors between estimated position and ground truth both in unoptimized and optimized trajectory are plotted below in Fig 9. The dashed lines represent the position errors between unoptimized poses and ground truth in three axes, while the

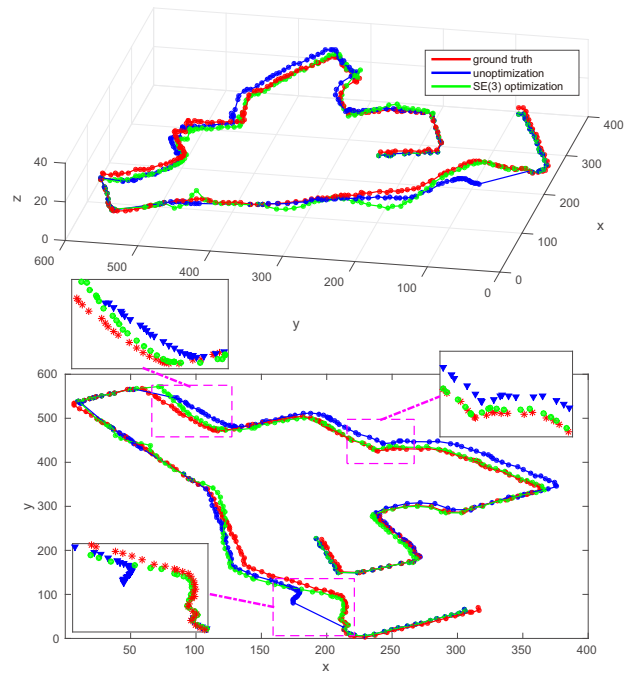


Fig. 8: Comparison results of trajectories

full lines represent the position errors of optimized trajectory.

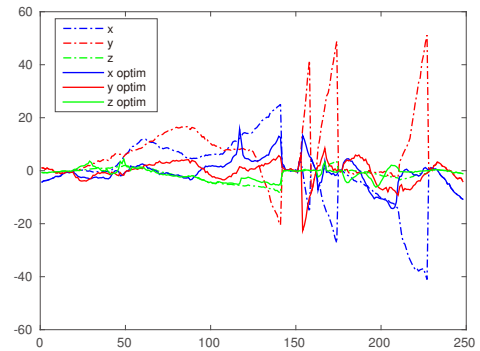


Fig. 9: Position errors in x , y , and z axis direction

As shown in Fig 9, position errors of optimized trajectory are mostly within 10 m, while the errors of unoptimized trajectory reach up to 40 m. Thus it can be seen that position errors can be reduced after pose graph optimization, especially when "scale-abruption" occurs. Besides, the optimized trajectory is smoother than the unoptimized trajectory.

Experimental results demonstrate that our method performs well in recovering the UAV trajectory in the real world coordinate system with the help of Google Street View images.

6 Conclusions

In this paper, a novel method of global metric localization using Google Street View is implemented to recover the pose of UAV with a monocular camera in the real world coordinate system. On the basis of a state-of-the-art visual SLAM algorithm and air-ground matching method, a novel absolute scale estimation mechanism and pose graph optimization are

completed to recovery the 6DoF pose of the UAV in metric unit and correct the accumulated drift. This method consists of two steps, namely place recognition and metric localization. In the first step, a coarse-to-fine scheme for match selection is proposed to match monocular images with Google Street Views accurately and quickly. Secondly, a novel absolute scale estimation mechanism is presented to compute the absolute scale and the poses are optimized with SE(3) constraints to recover trajectory and correct scale drift induced by monocular vSLAM. Experiments are conducted on the dataset recorded in the urban streets of Zurich, Switzerland, and results show that our method is able to metrically localize an UAV without pre-visit locations to build a map where Street View exists. The approach can be regarded as a complement of GPS to localize the UAV's positions in the real world coordinate system. We believe that our method paves the way towards a new convenient and widely used urban localization for the UAV.

References

- [1] Majdik A L, Verda D, AlbersSchoenberg Y, et al. Airground Matching: Appearancebased GPSdenied Urban Localization of Micro Aerial Vehicles[J]. *Journal of Field Robotics*, 2015, 32(7):1015-1039.
- [2] Ibaez-Guzmn J, Laugier C, Yoder J D, et al. *Autonomous Driving: Context and State-of-the-Art*[M]. Springer London, 2012.
- [3] Klose S, Wang J, Achtelik M, et al. Markerless, vision-assisted flight control of a quadcopter[C]// *Ieee/rsj International Conference on Intelligent Robots and Systems*. IEEE, 2010:5712-5717.
- [4] Eberli D, Scaramuzza D, Weiss S, et al. Vision Based Position Control for MAVs Using One Single Circular Landmark[J]. *Journal of Intelligent & Robotic Systems*, 2011, 61(1-4):495-512.
- [5] Scaramuzza D, Fraundorfer F. Visual Odometry [Tutorial][J]. *Robotics & Automation Magazine IEEE*, 2011, 18(4):80-92.
- [6] Morel J M, Yu G. ASIFT: A New Framework for Fully Affine Invariant Image Comparison[J]. *Siam Journal on Imaging Sciences*, 2009, 2(2):438-469.
- [7] Mur-Artal R, Montiel J M M, Tards J D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System[J]. *IEEE Transactions on Robotics*, 2015, 31(5):1147-1163.
- [8] Sadeghi H, Valae S, Shirani S. 2DTriPnP: A Robust Two-Dimensional Method for Fine Visual Localization Using Google Streetview Database[J]. *IEEE Transactions on Vehicular Technology*, 2017, 66(6):4678-4690.
- [9] Liu Z, Marlet R. Virtual Line Descriptor and Semi-Local Graph Matching Method for Reliable Feature Correspondence[C]// *British Machine Vision Conference*. 2012:16.1-16.11.
- [10] Zamir A R, Shah M. Accurate Image Localization Based on Google Maps Street View[C]// *Computer Vision - ECCV 2010, European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings*. DBLP, 2010:255-268.
- [11] Wang Z, Qureshi F Z. Topic Models for Image Localization[J]. 2013.
- [12] Zhai A, Clements M, Zakhora A. Large Area Cell Based Image Localization[C]// *IEEE International Symposium on Multimedia*. IEEE Computer Society, 2014:357-362.
- [13] T. Barfoot, *State Estimation for Robotics: A Matrix Lie Group Approach*. Draft in preparation for publication by Cambridge University Press, 2016.
- [14] Salarian M, Ansari R. Improved Image Retrieval for Efficient Localization in Urban Areas Using Location Uncertainty Data[C]// *IEEE International Symposium on Multimedia*. IEEE, 2017:181-184.
- [15] Majdik A L, Albers-Schoenberg Y, Scaramuzza D. MAV urban localization from Google street view data[C]// *Ieee/rsj International Conference on Intelligent Robots and Systems*. IEEE, 2013:3979-3986.
- [16] Viswanathan A, Pires B R, Huber D. Vision based robot localization by ground to satellite matching in GPS-denied situations[J]. 2005:870-876.
- [17] Torii A, Sivic J, Pajdla T. Visual localization by linear combination of image descriptors[C]// *IEEE International Conference on Computer Vision Workshops*. IEEE, 2011:102-109.
- [18] Agarwal P, Burgard W, Spinello L. Metric localization using Google Street View[J]. 2015:3111-3118.
- [19] Chang S P, Chien J T, Wang F E, et al. Extracting Driving Behavior: Global Metric Localization from Dashcam Videos in the Wild[J]. *European Conference on Computer Vision*. Springer, 2016. 136-148
- [20] Yu L, Joly C, Bresson G, et al. Monocular Urban Localization using Street View[C]. 2016 14th International Conference on Control, Automation, Robotics and Computer Vision, 2016. 1-6.
- [21] Engel J, Sturm J, Cremers D. Scale-aware navigation of a low-cost quadcopter with a monocular camera[J]. *Robotics & Autonomous Systems*, 2014, 62(11):1646-1656.
- [22] Muja M. Fast approximate nearest neighbors with automatic algorithm configuration[C]// *International Conference on Computer Vision Theory and Application Vissapp*. 2009:331-340.