Air-ground Matching: Appearance-based GPS-denied Urban Localization of Micro Aerial Vehicles

András L. Majdik
Department of Informatics, University of Zurich, Zurich, Switzerland
e-mail: andras@majdik.de
Damiano Verda
Italian National Council of Research, CNR-IEIIT, Genova, Italy
e-mail: damiano.verda@ieiit.cnr.it
Yves Albers-Schoenberg and Davide Scaramuzza
Department of Informatics, University of Zurich, Zurich, Switzerland
e-mail: yves.albers@gmail.com, davide.scaramuzza@ieee.org

Received 14 October 2014; accepted 20 January 2015

In this paper, we address the problem of globally localizing and tracking the pose of a camera-equipped micro aerial vehicle (MAV) flying in urban streets at low altitudes without GPS. An image-based global positioning system is introduced to localize the MAV with respect to the surrounding buildings. We propose a novel air-ground image-matching algorithm to search the airborne image of the MAV within a ground-level, geotagged image database. Based on the detected matching image features, we infer the global position of the MAV by back-projecting the corresponding image points onto a cadastral three-dimensional city model. Furthermore, we describe an algorithm to track the position of the flying vehicle over several frames and to correct the accumulated drift of the visual odometry whenever a good match is detected between the airborne and the ground-level images. The proposed approach is tested on a 2 km trajectory with a small quadrocopter flying in the streets of Zurich. Our vision-based global localization can robustly handle extreme changes in viewpoint, illumination, perceptual aliasing, and over-season variations, thus outperforming conventional visual place-recognition approaches. The dataset is made publicly available to the research community. To the best of our knowledge, this is the first work that studies and demonstrates global localization and position tracking of a drone in urban streets with a single onboard camera. © 2015 Wiley Periodicals, Inc.

1. INTRODUCTION

In this paper, we address the problem of localizing and tracking the pose of a camera-equipped rotary-wing micro aerial vehicle (MAV) flying in urban streets at low altitudes (i.e., 10–20 m from the ground) without a global positioning system (GPS). A novel appearance-based GPS to localize and track the pose of the MAV with respect to the surrounding buildings is presented.

Our motivation is to create vision-based localization methods for MAVs flying in urban environments, where the satellite GPS signal is often shadowed by the presence of the buildings, or is completely unavailable. Accurate localization is indispensable to safely operate small-sized aerial service-robots to perform everyday tasks, such as goods delivery, inspection and monitoring, and first-response and telepresence in the case of accidents.

First, we address the topological localization problem of the flying vehicle. The global position of the MAV is recovered by recognizing visually similar discrete places in the topological map. Namely, the air-level image captured by the MAV is searched in a database of ground-based geotagged pictures. Because of the large difference in viewpoint between the air-level and ground-level images, we call this problem *air-ground matching*.

Secondly, we address the metric localization and position tracking problem of the vehicle. The metric position of the vehicle is computed with respect to the surrounding buildings. We propose the use of textured three-dimensional (3D) city models to solve the *appearance-based global positioning* problem. A graphical illustration of the problem addressed in this work is shown in Figure 1.

In recent years, numerous papers have addressed the development of autonomous unmanned ground vehicles (UGVs), thus leading to striking new technologies, such as

The authors are with the Robotics and Perception Group, University of Zurich, Switzerland—http://rpg.ifi.uzh.ch. András Majdik is also affiliated with the Institute for Computer Science and Control, Hungarian Academy of Sciences, Hungary.
Direct correspondence to Davide Scaramuzza, e-mail: davide.scaramuzza@ieee.org

View this article online at wileyonlinelibrary.com • DOI: 10.1002/rob.21585

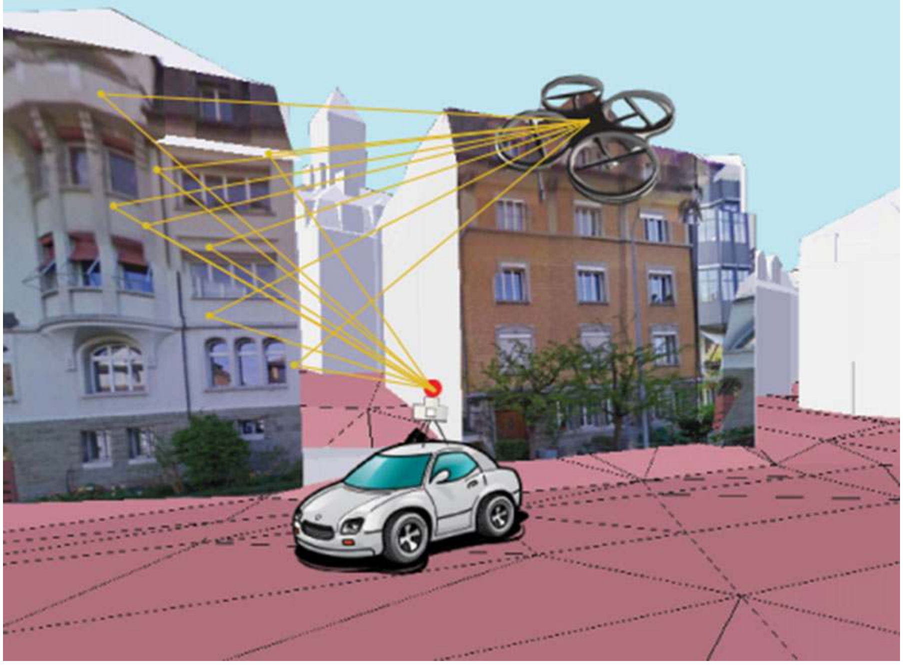


Figure 1. Illustration of the problem addressed in this work. The absolute position of the aerial vehicle is computed by matching airborne MAV images with ground-level Street View images that have previously been backprojected onto the cadastral 3D city model.

self-driving cars. These can map and react in highly uncertain street environments using partially (Churchill & Newman, 2012)—or completely neglecting—GPS (Ibañez Guzmán, Laugier, Yoder, & Thrun, 2012). In the coming years, a similar bust in the development of small-sized unmanned aerial vehicles (UAVs) is expected. Flying robots will be able perform a large variety of tasks in everyday life.

Visual-search techniques used in state-of-the-art place-recognition systems fail at matching air-ground images (Cummins & Newman, 2011; Galvez-Lopez & Tardos, 2012; Morel & Yu, 2009), since, in this case, extreme changes in viewpoint and scale can be found between the aerial images and the ground-level images. Furthermore, appearance-based localization is a challenging problem because of the large changes of illumination, lens distortion, over-season variation of the vegetation, and scene changes between the query and the database images.

To illustrate the challenges of the air-ground image matching scenario, in Figure 2 we show a few samples of the airborne images and their associate Google Street View (hereafter referred to as Street View) images from the dataset used in this work. As observed, due to the different field of view of the cameras on the ground and aerial vehicles and their different distance to the buildings' facades, the aerial image is often a small subsection of the ground-level image, which consists mainly of highly repetitive and self-similar structures (e.g., windows) (cf. Figure 3). All these peculiarities make the air-ground matching problem extremely difficult to solve for state-of-the-art feature-based image-search techniques.

We depart from conventional image-search algorithms by generating artificial views of the scene in order to overcome the large viewpoint differences between the Street View and MAV images, and thus successfully solve their matching. An efficient artificial-view generation algorithm is introduced by exploiting the air-ground geometry of our system, thus leading to a significant improvement of the correctly paired airborne images to the ground level ones.

Furthermore, to deal with the large number of outliers (about 80%) that the large viewpoint difference introduces during the feature-matching process, in the final verification step of the algorithm, we leverage an alternative solution to the classical random sample consensus (RANSAC) approach, which can deal with such a high outlier ratio in a reasonable amount of time.

In this paper, we advance our previous topological localization (Majdik, Albers-Schoenberg, & Scaramuzza, 2013) by computing and tracking the pose of the MAV using cadastral 3D city models, which we first introduced in Majdik, Verda, Albers-Schoenberg, & Scaramuzza (2014). Furthermore, we present an appearance-based global positioning system that is able to successfully substitute

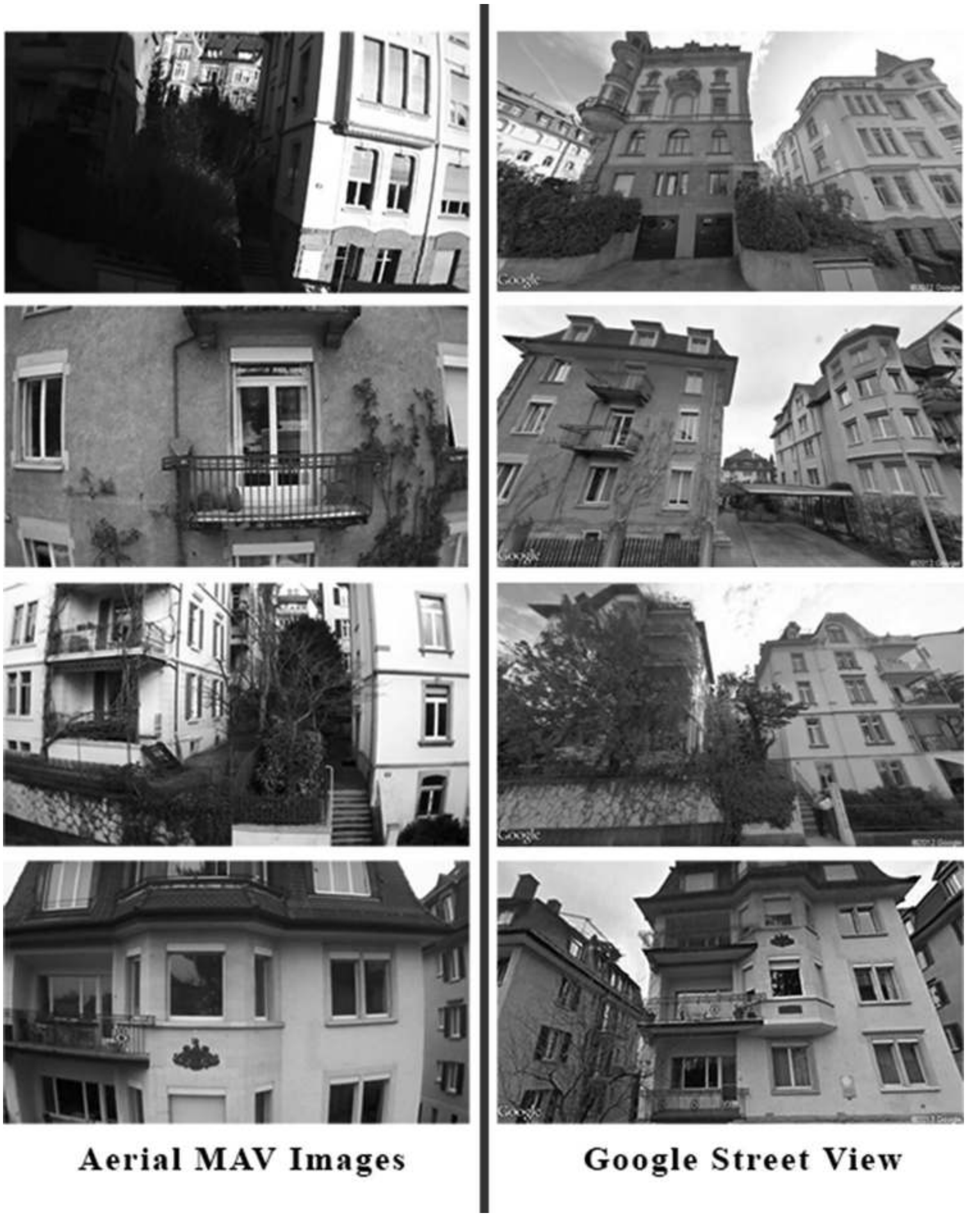


Figure 2. Comparison between airborne MAV (left) and ground-level Street View images (right). Note the significant changes—in terms of viewpoint, illumination, over-season variation, lens distortions, and the scene between the query (left) and the database images (right)—that obstruct their visual recognition.

the satellite GPS for MAVs flying in urban streets. By means of uncertainty quantification, we are able to estimate the accuracy of the visual localization system. We show extended experiments of the appearance-based global localization system on a 2 km trajectory with a drone flying in the streets of Zurich. Finally, we show a real application of the system, where the state of the MAV is updated whenever a new appearance-based global position measurement becomes available. To the best of our knowledge, this is the first work that studies and demonstrates global localization of a drone in urban streets with vision only.

The contributions of this paper are as follows:

- We solve the problem of *air-ground matching* between MAV-based and ground-based images in urban environments. Specifically, we propose to generate artificial

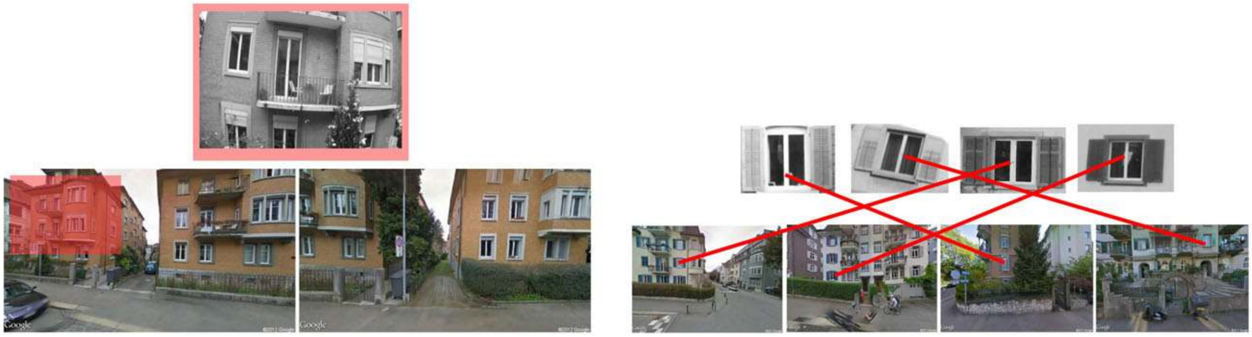


Figure 3. Please note that often the aerial MAV image (displayed in monocolor) is just a small subsection of the Street View image (color images) and that the airborne images contain highly repetitive and self-similar structures.

views of the scene in order to overcome the large viewpoint differences between ground and aerial images, and thus successfully resolve their matching.

- We present a new appearance-based global positioning system to detect the position of MAVs with respect to the surrounding buildings. The proposed algorithm matches airborne MAV images with geotagged Street View images[1] and exploits cadastral 3D city models to compute the absolute position of the flying vehicle.
- We describe an algorithm to track the vehicle position and correct the accumulated drift induced by the onboard state estimator.
- We provide the first ground-truth labeled dataset that contains both aerial images—recorded by a drone together with other measured parameters—and geotagged ground-level images of urban streets. We hope that this dataset can motivate further research in this field and serve as benchmark.

The remainder of the paper is organized as follows. Section II presents the related work. Section III describes the air-ground matching algorithm. Section IV presents the appearance-based global positioning system. Section V describes the position tracking algorithm. Finally, Section VI presents the experimental results.

2. RELATED WORK

Several research works have addressed appearance-based localization throughout image search and matching in urban environments. Many of them were developed for ground-robot simultaneous localization and mapping (SLAM) systems to address the loop-closing problem (Cummins & Newman, 2011; Galvez-Lopez & Tardos, 2012; Maddern, Milford, & Wyeth, 2012; Majdik, Gálvez-López, Lazea, & Castellanos, 2011), while other works focused on position tracking using the Bayesian fashion—such as in Vaca-Castano, Zamir, & Shah (2012), where the authors presented a method that also uses Street View data to track the geospatial position of a camera-equipped car in a citylike environment. Other algorithms used image-search–based localization for handheld mobile devices to detect a point of interest (POI), such as landmark buildings or museums (Baatz, Köser, Chen, Grzeszczuk, & Pollefeys, 2012; Fritz, Seifert, Kumar, & Paletta, 2005; Yeh, Tollmar, & Darrell, 2004). Finally, in recent years, several works have focused on image localization with Street View data (Schindler, Brown, & Szeliski, 2007; Zamir & Shah, 2010). However, all the works mentioned above aim to localize street-level images in a database of pictures also captured at street level. These assumptions are safe in ground-based settings, where there are no large changes between the images in terms of viewpoint. However, as will be discussed later in Section 3.5 and Figure 8, traditional algorithms tend to fail in air-ground settings, where the goal is to match airborne imagery with ground imagery.

Most works addressing the air-ground-matching problem have relied on assumptions different from ours, notably the altitude at which the aerial images are taken. For instance, the problem of geolocalizing ground-level images in urban environments with respect to *satellite* or *high-altitude* (several hundred meters) aerial imagery was studied in Bansal, Sawhney, Cheng, & Daniilidis (2011) and Bansal, Daniilidis, & Sawhney (2012). In contrast, in this paper we aim specifically at low-altitude imagery, which means images captured by safe MAVs flying 10–20 m from the soil.

A downward-looking camera is used in Conte & Doherty (2009) in order to cope with long-term GPS outages. The visual odometry is fused with the inertial sensors measurements, and the onboard video data are registered in a georeferenced aerial image. In contrast, in this paper we use a MAV equipped with a side-looking camera, always facing the buildings along the street. Furthermore, we describe a

[1] By geotag, we mean the latitude and longitude data in the geographic coordinate system, enclosed in the metadata of the Street View images.

method that is able to solve the first localization problem by using image retrial techniques.

World models, maps of the environment, and street-network layouts have been used to localize vehicles performing planar motion in urban environments (Montemerlo et al., 2008). Recently, several research works have addressed the localization of ground vehicles using publicly available maps (Brubaker, Geiger, & Urtasun, 2013; Floros, Zander, & Leibe, 2013), road networks (Hentschel & Wagner, 2010), or satellite images (Kuemmerle et al., 2011). However, the algorithms described in those works are not suitable for the localization of flying vehicles, because of the large viewpoint differences. With the advance of mapping technologies, more and more detailed, textured 3D city models are becoming publicly available (Anguelov et al., 2010), which can be exploited for vision-based localization of MAVs.

As envisaged by several companies, MAVs will be soon used to transport goods,[2] medications and blood samples,[3] or even pizzas from building to building in large urban settings. Therefore, improving localization at small altitude where a GPS signal is shadowed or completely unreliable is of the utmost importance.

3. AIR-GROUND MATCHING OF IMAGES

In this section, we describe the proposed algorithm to match airborne MAV images with ground-level ones. A pseudocode description is given in Algorithm 1. Please note that the algorithm from line 1 to 7 can and should be computed offline, previous to an actual flight mission. In this phase, previously saved geotagged images $\mathcal{I} = \{I_1, I_2, \dots, I_n\}$ are converted into image-feature–based representations F_i (after applying the artificial-view generation method described in the next section) and are saved in a database D_T. Next, for every aerial image I_a we perform artificial-view generation and feature extraction steps (lines 9 and 10). The extracted features F_a are searched in the database D_T (line 11). We select a finite number of ground-level images, using the putative match selection method (line 12) detailed in Section 3.2. Finally, we run in parallel a more elaborate image similarity test (lines 13–16) to obtained the best matching Street View image I_a to the aerial one I_a. In the next sections, we give further details about the proposed algorithm.

3.1. Artificial-view Generation

Point feature detectors and descriptors—such as SIFT (Lowe, 2004), SURF (Bay, Ess, Tuytelaars, & Van Gool, 2008), etc.—usually ensure invariance to rotation and scale. However, they tend to fail in the case of substantial viewpoint changes ($\theta > 45^\circ$).

Table I. Tilting values for which artificial views were made.

Tilt	$\sqrt{2}$	2	$2\sqrt{2}$
θ	45°	60°	69.3°

Our approach was inspired by a technique initially presented in Morel & Yu (2009), where, for a complete affine invariance (six degrees of freedom), it was proposed to simulate all image views obtainable by varying the two camera-axis orientation parameters, namely the latitude and the longitude angles. The longitude angle (ϕ) and the latitude angles (θ) are defined in Figure 4 on the right. The tilt can thus be defined as tilt $= \frac{1}{\cos(\theta)}$. The affine scale-invariant feature transform [ASIFT (Morel & Yu, 2009)] detector and descriptor is obtained by sampling various values for the tilt and longitude angle ϕ to compute artificial views of the scene. Further on, SIFT features are detected on the original image as well as on the artificially generated images.

In contrast, in our implementation, we limit the number of considered tilts by exploiting the air-ground geometry of our system. To address our air-ground-matching problem, we sample the tilt values along the vertical direction of the image instead of the horizontal one. Furthermore, instead of the arithmetical sampling of the longitude angle at every tilt level proposed in Morel & Yu (2009), we make use of just three artificial simulations, i.e., at 0° and ±40°. We illustrate the proposed parameter-sampling method in Figure 4 and display the different tilt values in Table I. By adopting this efficient sampling method, we managed to reduce the computational complexity by a factor of 6 (from 60 to 9 artificial views).

We have chosen this particular discretization in order to exploit the air-ground geometry of the air-ground-matching problem. Thus, we obtained a significant improvement of the correctly paired airborne images to the ground-level ones. Furthermore, we limited the number of artificial views in comparison to the original ASIFT technique in order to reduce the computational complexity of the algorithm. Based on our experiments, using a higher number of artificial views, the performances are not improved.

In conclusion, the algorithm described in this section has two main advantages in comparison with the original ASIFT implementation (Morel & Yu, 2009). First, we significantly reduce the number of artificial views needed by exploiting the air-ground geometry of our system, thus leading to a significant improvement in the computational complexity. Second, by introducing fewer error sources into the matching algorithm, our solution contributes also to obtaining an increased performance in the global localization process.

[2] Amazon Prime Air.

[3] Matternet.

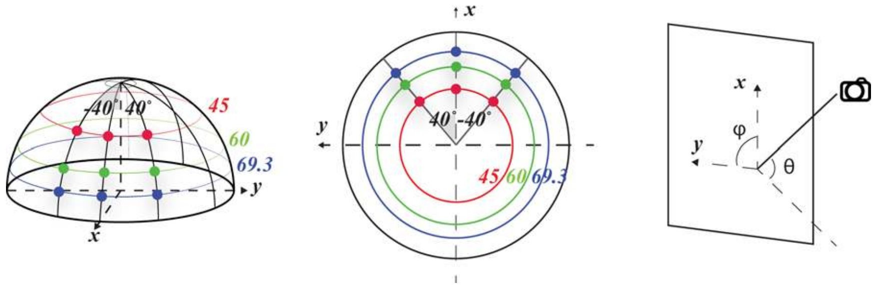


Figure 4. Illustration of the sampling parameters for artificial-view generation. Left: observation hemisphere—perspective view. Right: observation hemisphere—zenith view. The samples are marked with dots.

```
Algorithm 1: Vision-based global localization of MAVs
  Input: A finite set $\mathcal{I} = \{I_1, I_2, \ldots, I_n\}$ of ground geotagged images
  Input: An aerial image $I_a$ taken by a drone in a streetlike environment
  Output: The location of the drone in the discrete map and the best match $I_b$, respectively
  $D_T$ = database of all the image features of $\mathcal{I}$;
  for $i \leftarrow 1$ to $n$ do
      $V_i$ = generate artificial-views ($I_i$); // details in Section 3.1 ;
      $F_i$ = extract image features ($V_i$);
      add $F_i$ to $D_T$;
  train $D_T$ using FLANN (Muja & Lowe, 2009);
  $c \leftarrow$ number of cores;
  // up to this line the algorithm is computed offline ;
  $V_a$ = generate artificial-views ($I_a$);
  $F_a$ = extract image features ($V_a$);
  search approximate nearest-neighbor feature matches for $F_a$ in $D_T$: $M_D = \mathrm{ANN}(F_a, D_T)$ ;
  select $c$ putative image matches $\mathcal{I}^p \subseteq \mathcal{I}$: $\mathcal{I}^p = \{I_1^p, I_2^p, \ldots, I_c^p\}$ // details Section 3.2 ;
  run in parallel for $j \leftarrow 1$ to $c$ do
      search approximate nearest-neighbor feature matches for $F_a$ in $F_j^p$: $M_j = \mathrm{ANN}(F_a, F_j^p)$;
      select inlier points: $N_j = \mathrm{kVLD}(M_j, I_a, I_j^p)$;
  $I_b \leftarrow \max(N_1, N_2, \ldots, N_c)$;
  return $I_b$;
```

3.2. Putative Match Selection

One might argue that artificial-view generation leads to a significant computational complexity. We overcome this issue by selecting only a finite number of the most similar Street View images. Namely, we present a novel algorithm to select these putative matches based on a computationally inexpensive and extremely fast two-dimensional histogram-voting scheme.

The selected, ground-level candidate images are then subjected to a more detailed analysis that is carried out in parallel on the available cores of the processing unit. The experiments show that in selecting only four candidate Street View images, very good results were obtained with the proposed algorithm.

In this step, the algorithm selects a fixed number of putative image matches $\mathcal{I}^p = \{I_1^p, I_2^p, \ldots, I_c^p\}$, based on the available hardware. The idea is to select a subset of the Street View images from the total number of all possible matches and to exclusively process these selected images in parallel, in order to establish a correct correspondence with the aerial image. This approach enables a very fast computation of the algorithm. In case there are no multiple cores available, the algorithm could be serialized, but the computational time would increase accordingly. The subset of the ground images is selected by searching for the approximate nearest neighbor for all the image features extracted from the aerial image and its artificial views F_a. The search is performed using the FLANN (Muja & Lowe, 2009) library, which implements multiple randomized KD-tree or K-means tree forests and autotuning of the parameters. According to the literature, this method performs the search extremely fast and with good precision, although for searching in very large databases (hundreds of millions of images), there are more efficient algorithms (Jégou, Douze, & Schmid, 2011). Since we perform the search in a certain area, we opted for FLANN.

Further on, we apply an idea similar to that of Scaramuzza (2011), where in order to eliminate the outlier features, just a rotation is estimated between two images. In our approach, we compute the difference in orientation α between the image features of the aerial view F_a and the approximate nearest neighbor found in D_T. Next, by using a histogram-voting scheme, we look for that specific Street View image that contains the most image features with the same angular change. To further improve the speed of the algorithm, the possible values of α are clustered in bins of five. Accordingly, a two-dimensional histogram H can be built, in which each bin contains the number of features that count for α in a certain Street View image. Finally, we select the number c of Street View images that have the maximal values in H.

To evaluate the performance of our algorithm, we run several tests using the same 2-km-long dataset and test parameters, only modifying the number of selected candidate Street View images, i.e., the number of parallel cores. Figure 5 shows the obtained results in terms of *recall rate*[4] and *precision rate*[5] for 4, 8, 16, and 48 selected candidate Street View images (parallel cores). The plot shows that, even by using just four cores in parallel, a significant number of true-positive matches between the MAV and the Street View images are found without having any erroneous pairing, namely at precision 1. Using eight putative Street View images processed in parallel on different cores, the recall at precision 1 increases by almost 3%. Please note that it is also possible to use 2 × 4 cores to obtain the same performance. By further increasing the number of cores (e.g., in the case of a *cloud-robotics* scenario), minor improvements in performance are obtained in terms of precision and recall (cf. Table II). In case a pool of 96 candidate Street View images are selected, the number of correct matches at precision 1 is not increased anymore. Therefore, this shows the limitations of the air-ground matching algorithm.

Table II. Recall rate at precision 1 (RR-P1) in the case of the number of putative Street View images analyzed in parallel on different cores (NPC denotes number of parallel cores).

NPC	4	8	16	48	96
RR-P1 (%)	41.9	44.7	45.9	46.4	46.4

More importantly, it can be concluded that the presented approach to select putative matches from the Street View data has a very good performance, and, by just selecting 3% of the total number of possible matches, it can detect more than 40% of the true positive matches at precision 1.

3.3. Pairing and Acceptance of Good Matches

Having selected c Street View images $\mathcal{I}^p = \{I_1^p, I_2^p, \ldots, I_c^p\}$ as described in the preceding section, in the final part of the algorithm we make a more detailed analysis in parallel to compute the final best match for the MAV image. Similarly to line 11 in Algorithm 1, we search for the approximate nearest neighbor of every feature of the aerial image F_a in each selected ground-level image I_j^p. The feature points F_j^p contained in I_j^p are retrieved from the Street View image feature database D_T, and matched against F_a.

To pair the airborne MAV images with the Street View data and select the best match among the putative images, we make a verification step (line 15 in Algorithm 1). The goal of this step is to select the inliers, correctly match feature points, and reject the outliers. As emphasized earlier, the air-ground matching of images is very challenging for

[4] Recall rate = number of detected matches over the total number of possible correspondences.

[5] Precision rate = number of true positives detected over the total number of matches detected (both true and false).

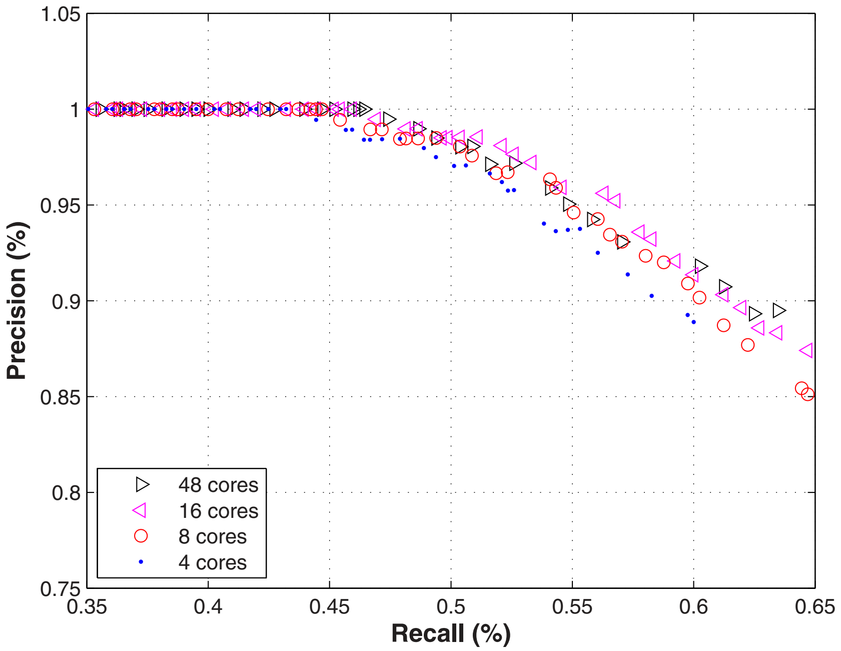


Figure 5. Performance analysis in terms of precision and recall in the case of 4, 8, 16, and 48 threads was used in parallel. Please note that by selecting just 3% of the total number of possible matches, more than 40% of the true positive matches were detected by the proposed algorithm.

several reasons, and thus traditional RANSAC-based approaches tend to fail, or need a very high number of iterations, as shown in the previous section. Consequently, in this paper we make use of an alternative solution to eliminate outlier points and to determine feature point correspondences, which extends the pure photometric matching with a graph-based one.

In this work, we use the virtual line descriptor (kVLD) (Liu & Marlet, 2012). Between two key-points of the image, a virtual line is defined and assigned a SIFT-like descriptor, after the points pass a geometrical consistency check as in Albarelli, Rodolà, & Torsello (2012). Consistent image matches are searched in the other image by computing and comparing the virtual lines. Further on, the algorithm connects and matches a graph consisting of k connected virtual lines. The image points that support a kVLD graph structure are considered inliers, while the other ones are marked as outliers. In the next section, we show the efficiency and precision of this method as well as the artificial-view generation and putative-match selection.

The precision of the air-ground matching algorithm and the uncertainty of the position determination depend on the number of correctly matched image features. Figure 6 summarizes the mean number of inliers matched between airborne and ground images as a function of the distance to the closest Street View image. The results show a Gaussian distribution with standard deviation $\sigma = 5$ m. This means that, if the MAV is within 5 m from a Street View image along the path, our algorithm can detect around 60 correct correspondences.

3.4. Computational Complexity

The main goal of this work is to present a proof-of-concept of the system, rather than a real-time, efficient implementation. The aim of this paper is to present the first appearance-based global localization system for rotary-wing MAVs, similarly to the very popular visual-localization algorithms for ground-level vehicles (Brubaker et al., 2013; Cummins & Newman, 2011). For the sake of completeness, we present in Figure 7 the effective processing time of the air-ground image-matching algorithm, using a commercially available laptop with an eight-core—2.40 GHz clock—architecture.

The air-ground matching algorithm is computed in five major steps: (1) artificial-view generation and feature extraction (Section 3.1); (2) approximate nearest-neighbor search within the full Street View database (line 11 in Algorithm 1); (3) putative correspondences selection (Section 3.3); (4) approximate nearest-neighbor search among the features extracted from the aerial MAV image with respect to the selected ground-level image (line 14 in Algorithm 1); (5) acceptance of good matches (Section 3.3).

In Figure 7 we used the 2-km-long dataset and more than 400 airborne MAV images. All the images were searched within the entire Street View images that could be found along the 2 km trajectory. Notice that the longest

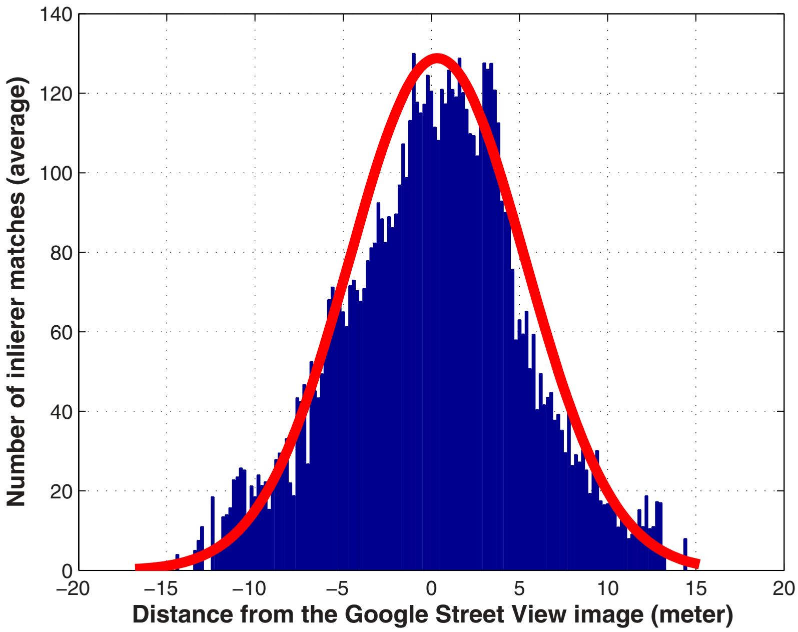


Figure 6. Number of inlier feature points matched between the MAV and ground images as a function of the distance to the closest Street View image.

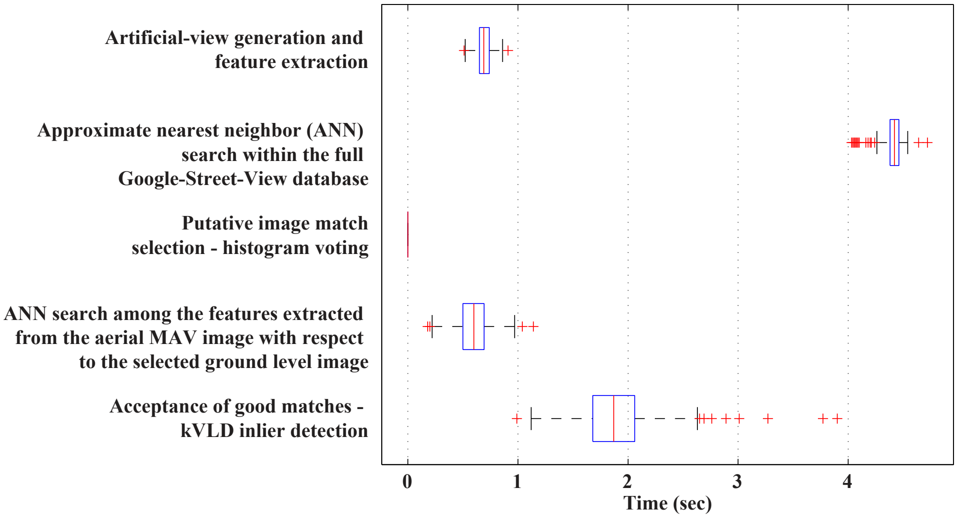


Figure 7. Analysis of the processing time of the air-ground image-matching algorithm. To compute this figure, we used more than 400 airborne MAV images, and all the images were searched within the entire Street View image database, which could be found along the 2 km trajectory.

computation time is the approximate nearest-neighbor search in the entire Street View database for the feature descriptors found in the MAV image. However, this step can be completely neglected once an approximate position of the MAV is known, because in this case, the air-ground matching algorithm can be applied using a distance-based approach instead of a brute-force search.

In a distance-based scenario, the closest Street View images are selected, which are inside of a certain radius from the MAV, e.g., 100 m bound in urban streets. By adopting a distance-based approach, the appearance-based localization problem can be significantly simplified. We have evaluated the air-ground matching algorithm using a brute-force search, because our aim was to solve a more general problem, namely the first localization problem.

In the position tracking experiment (Section 5), we used the distance-based approach, since, in that case, the MAV image is compared only with the neighboring Street View images (usually up to four or eight, computed in parallel on different cores, depending on the road configuration).

Finally, notice that the histogram voting (Figure 7) takes only 0.01 s.

Using the current implementation, on average, an appearance-based global localization—steps (1), (4), and (5)—is computed in 3.2 s. Therefore, if the MAV flies roughly with a speed of 2 m/s, its position would be updated every 6.5 m. The computational time could be significantly reduced by outsourcing the image-processing computations to a server in a cloud-robotics scenario.

3.5. Comparison with State-of-the-art Techniques

Here, we briefly describe four state-of-the-art algorithms, against which we compare and evaluate our approach. These algorithms can be classified into *brute-force* or *bag-of-words* strategies. All the results shown in this section were obtained using the 2-km-long dataset; cf., Appendix A.

3.5.1. Brute-force Search Algorithms

Brute-force approaches work by comparing each aerial image with every Street View image in the database. These algorithms have better precision but at the expense of a very-high computational complexity. The first algorithm that we used for comparison is referred to as *brute-force feature matching*. This algorithm is similar to a standard object-detection method. It compares all the airborne images from the MAV to all the ground-level Street View images. A comparison between two images is done through the following pipeline: (i) SIFT (Lowe, 2004) image features are extracted in both images; (ii) their descriptors are matched; (iii) outliers are rejected through verification of their geometric consistency via fundamental-matrix estimation [e.g., the RANSAC eight-point algorithm (Hartley & Zisserman, 2004)]. RANSAC-like algorithms work robustly as long as the percentage of outliers in the data is below 50%. The number of iterations N needed to select at least one random sample set free of outliers with a given confidence level p—usually set to be 0.99—can be computed as (Fischler & Bolles, 1981)

$$N = \log(1 - p)/\log[1 - (1 - \gamma)^s], \qquad (1)$$

where γ specifies the expected outlier ratio. Using the eight-point implementation ($s = 8$) and given an outlier ratio larger than 70%, it becomes evident that the number of iterations needed to robustly reject outliers becomes unmanageable, on the order of 100 000 iterations, and grows exponentially.

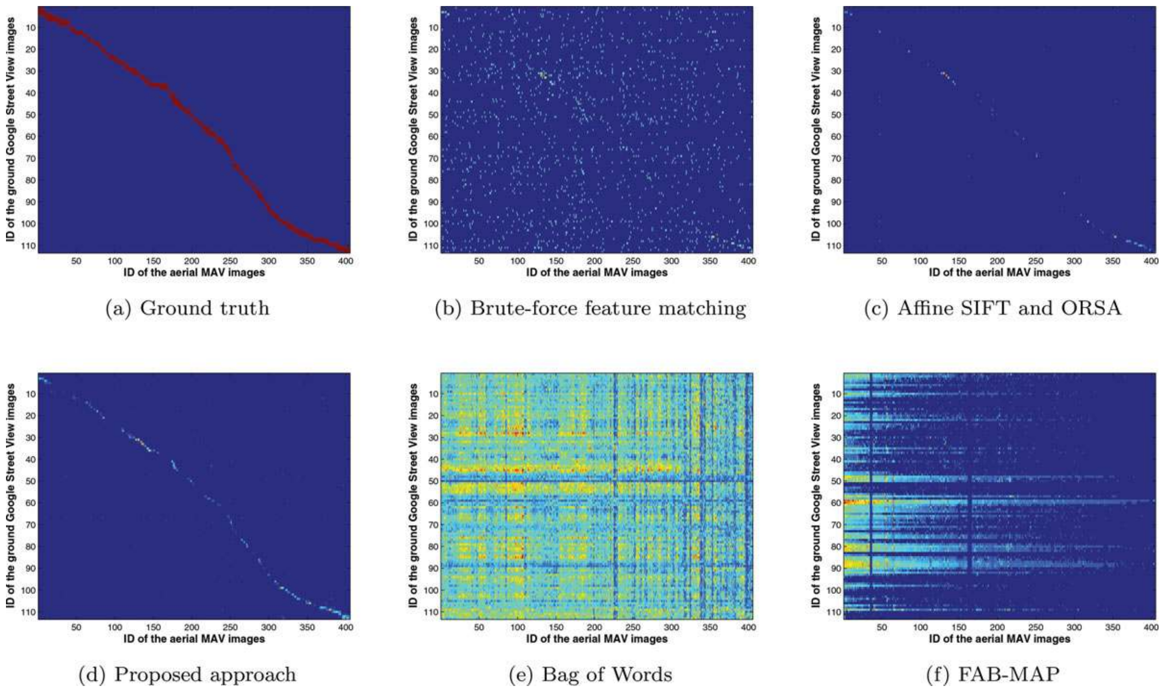
From our studies, the outlier ratio after applying the described feature-matching steps on the given air-ground dataset (before RANSAC) is between 80% and 90%, or, stated differently, only 10–20 % of the found matches (between images of the same scene) correspond to correct match pairs. Following the above analysis, in the case of our dataset, which is illustrated in Figure 2, we conclude that RANSAC-like methods fail to robustly reject wrong correspondences. The confusion matrix depicted in Figure 8(b) reports the results of brute-force feature matching. This further underlines the inability of RANSAC to uniquely identify two corresponding images in our air-ground search scenario. We obtained very similar results using four-point RANSAC—which leverages the planarity constraint between feature sets belonging to building facades.

The second algorithm applied to our air-ground-matching scenario is the one presented in Morel & Yu (2009), here referred to as *Affine SIFT and ORSA*. In Morel & Yu (2009), an image-warping algorithm is described to compute artificially generated views of a planar scene able to cope with large viewpoint changes. ORSA (Moisan, Moulon, & Monasse, 2012) is a variant of RANSAC, which introduces an adaptive criterion to avoid the hard thresholds for inlier/outlier discrimination. The results were improved by adopting this strategy [shown in Figure 8(c)], although the recall rate at precision 1 was below 15% (cf. Figure 17).

3.5.2. Bag-of-words Search Algorithms

The second category of algorithms used for comparison is the bag-of-words (BoW) -based method (Sivic and Zisserman, 2003), devised to improve the speed of image-search algorithms. This technique represents an image as a numerical vector quantizing its salient local features. Their technique entails an offline stage that performs hierarchical clustering of the image descriptor space, obtaining a set of clusters arranged in a tree structure. The leaves of the tree form the so-called visual vocabulary, and each leaf is referred to as a visual word. The similarity between two images, described by the BoW vectors, is estimated by counting the common visual words in the images. Different weighting strategies can be adopted between the words of the visual vocabulary (Majdik et al., 2011). The results of this approach applied to the air-ground dataset are shown in Figure 8(e). We tested different configuration parameters, but the results did not improve (cf. Figure 17).

Additional experiments were carried out by exploiting the joint advantages of the Affine SIFT feature extraction algorithm and that of the bag-of-words technique, referred to as *ASIFT bag-of-words*. In this experiment, SIFT features were extracted also on the generated artificial views for both the aerial and ground-level images. Later on, all the extracted feature vectors were transformed into the BoW representation. Lastly, the BoW vectors extracted from the



(a) Ground truth

(b) Brute-force feature matching

(c) Affine SIFT and ORSA

(d) Proposed approach

(e) Bag of Words

(f) FAB-MAP

Figure 8. These plots show the confusion matrices obtained by applying several algorithms described in the literature [(b),(c) and (e),(f)] and the one proposed in the current paper (d). (a) Ground-truth: the data were manually labeled to establish the exact visual overlap between the aerial MAV images and the ground Street View image; (b) brute-force feature matching; (c) affine-SIFT and ORSA; (d) our proposed air-ground-matching algorithm; (e) bag of words (BoW); (f) FAB-MAP. Notice that our algorithm outperforms all other approaches in the challenging task of matching ground and aerial images. For precision and recall curves, compare to Figure 17.

airborne MAV images were match with the one computed from the Street View images. The results of this approach are shown on Figure 9. Note that the average precision—the area below the precision-recall curve—was significantly improved in comparison with the standard BoW approach (cf. Figure 17).

Finally, the fourth algorithm used for our comparison is *FAB-MAP* (Cummins & Newman, 2011). To cope with perceptual aliasing, in Cummins & Newman (2011) an algorithm is presented in which the coappearance probability of certain visual words is modeled in a probabilistic framework. This algorithm was successfully used in traditional street-level ground-vehicle localization scenarios, but it failed in our air-ground-matching scenario, as displayed in Figure 8(f).

As observed, both BoW and FAB-MAP approaches fail to correctly pair air-ground images. The reason is that the visual patterns of the air and ground images are classified with different visual words, thus leading to a false visual-word association. Consequently, the air-level images are erroneously matched to the Street View database.

To conclude, all these algorithms perform rather unsatisfactorily in the air-ground matching scenario, due to the issues emphasized at the beginning of this paper. This motivated the development of a novel algorithm presented throughout this section. The confusion matrix of the proposed algorithm applied to our air-ground matching scenario is shown in Figure 8(d). This can be compared with the confusion matrix of the ground-truth data [Figure 8(a)]. As observed, the proposed algorithm outperforms all previous approaches. In the Section 6, we give further details about the performance of the described algorithm.

4. APPEARANCE-BASED GLOBAL POSITIONING SYSTEM

In this section, we extend the topological localization algorithm described in the previous section in order to compute the global position of the flying vehicle in a metric map. To achieve this goal, we backproject each pixel onto the 3D cadastral model of the city. Please note that the approach detailed in this section is independent of the 3D model used, thus the same algorithm can be applied to any other textured 3D city model.

4.1. Textured 3D Cadastral Models

The 3D cadastral model of Zurich used in this work was acquired from the city administration and claims to have

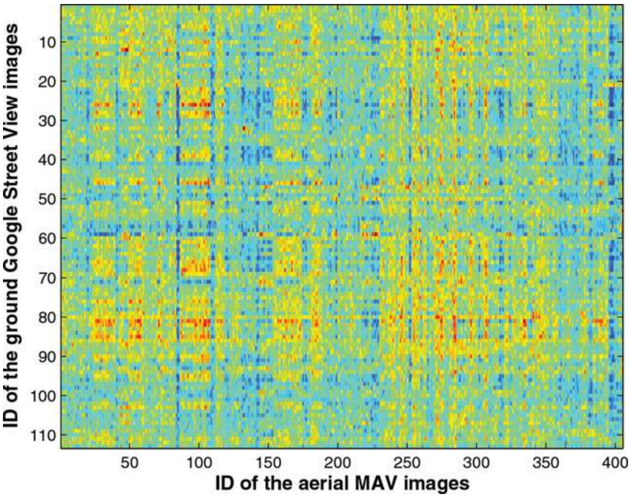


Figure 9. This figure shows the confusion matrix obtained by applying the affine-SIFT feature extraction algorithm and the bag-of-words technique to match the airborne MAV images with the Street View images.

an average lateral position error of $\sigma_l = \pm 10$ cm and an average error in height of $\sigma_h = \pm 50$ cm. The city model is referenced in the Swiss Coordinate System *CH1903* (DDPS, 2008). Note in Figure 11(a) that this model does not contain any textures. By placing virtual cameras in the cadastral model, 2D images and 3D depth maps can be obtained from any arbitrary position within the model, using the Blender[6] software environment.

The geolocation information of the Street View dataset is not exact. The geotags of the Street View images provide only approximate information about where the images were recorded by the vehicle. Indeed, according to Taneja, Ballan, & Pollefeys (2012), where 1,400 Street View images were used to perform the analysis, the average error of the camera positions is 3.7 m and the average error of the camera orientation is 1.9 degrees. In the same work, an algorithm was proposed to improve the precision of the Street View image poses. There, cadastral 3D city-models were used to generate virtual 2D images, in combination with image-segmentation techniques, to detect the outline of the buildings. Finally, the pose was computed by an iterative optimization, namely by minimizing the offset between the segmented outline in the Street View and the virtual images. The resulting corrected Street View image positions have a standard deviation of 0.1184 m, and the orientation of the cameras have standard deviation of 0.476 degrees.

In our work, we apply the algorithm from Taneja et al. (2012) on the dataset used in this work to correct the Street View image poses. Then, from the known location of the Street View image, we backproject each pixel onto the 3D cadastral model [Figure 11(b)]. One sample of the resulting *textured 3D model* is shown in Figure 11(c). By applying this procedure, we are able to compute the 3D location of the image features detected on the 2D images. This step is crucial to compute the scale of the monocular visual odometry (Section 5.1) and to localize the MAV images with respect to the street level ones, thus reducing the uncertainty of the position tracking algorithm. In the next section, we give more details about the integration of textured 3D models into our pipeline.

4.2. Global MAV Camera Pose Estimation

The steps of the algorithm are visualized in Figure 10. For the georeferenced Street View images, depth maps are computed by backprojecting the image from the known camera position onto the 3D model [Figure 10(a)]. The air-ground matching algorithm described in the preceding section detects the most similar Street View in the database for a given MAV image [Figure 10(b)]. Also, the 2D-2D image feature correspondences are computed by the air-ground matching algorithm, shown with green lines in Figure 10(c). The magenta lines are the virtual lines used to distinguish the inlier points from the outlier ones (Section 3.3). Since the depth of every image pixel of the Street View image is known from the 3D city model, 3D-2D point correspondences are computed [Figure 10(d)]. The absolute MAV camera pose and orientation [Figure 10(e)] are estimated given a set of known 3D-2D correspondence points.

Several approaches have been proposed in the literature to estimate the external camera parameters based on 3D-2D correspondences. In Fischler & Bolles (1981), the *perspective-n-point (PnP) problem* was introduced, and different solutions were described to retrieve the absolute camera pose given *n correspondences*. Kneip, Scaramuzza, & Siegwart (2011) addressed the PnP problem for the minimal case in which *n* equals 3, and they introduced a novel parametrization to compute the absolute camera position and orientation. In this work, the efficient PnP (EPnP) algorithm (Moreno-Noguer, Lepetit, & Fua, 2007) is used to estimate the MAV camera position and orientation with respect to the global reference frame. The advantage of the EPnP algorithm with respect to other state-of-the-art noniterative PnP techniques is the low computational complexity and the robustness in terms of noise in the 2D point locations.

Given that the output of our air-ground matching algorithm may still contain outliers and that the model-generated 3D coordinates may depart from the real 3D coordinates, we apply the EPnP algorithm together with a RANSAC scheme (Fischler & Bolles, 1981) to discard the outliers. However, the number of inlier points is reduced by

[6] Blender 3D modeling software environment: http://www.blender.org/.

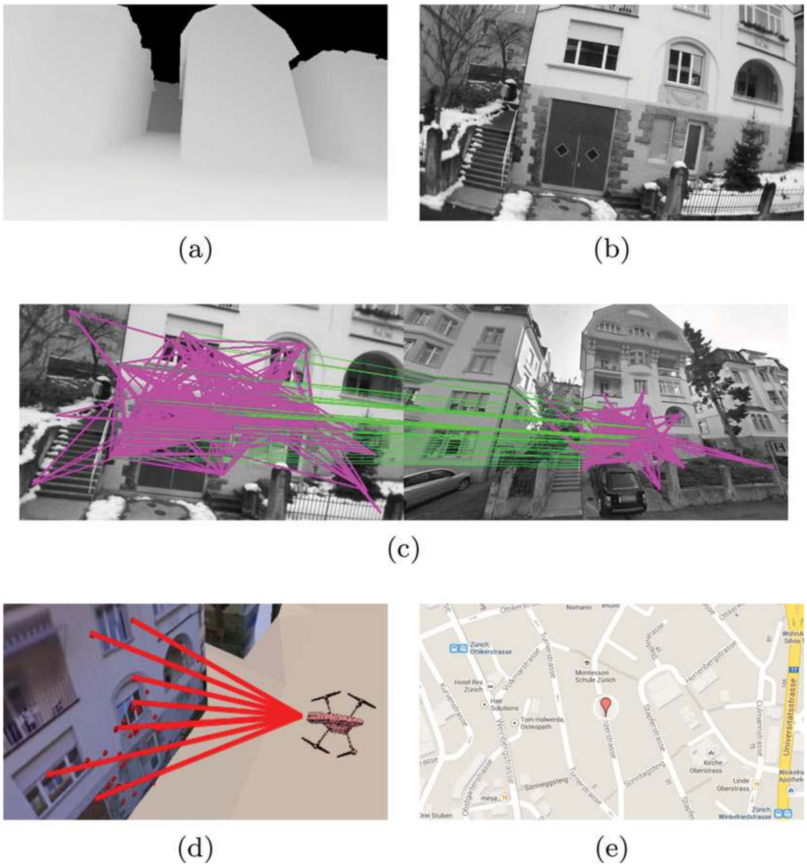


Figure 10. (a) Street View image depth map obtained from the 3D cadastral city model; (b) airborne MAV image; (c) matched feature point pairs (green lines) between the Street View and the MAV image; the magenta lines are the virtual lines used to distinguish the inlier points from the outlier ones (Section 3.3); (d) 3D-2D point correspondences between the texture 3D city model and the MAV image; (e) global position of the MAV, computed based on the 3D-2D point correspondences.

using the EPnP-RANSAC scheme in comparison with the number of inlier points provided by the air-ground matching algorithm, as shown in Figure 12 for a testbed of more than 1,600 samples from the 2 km dataset. This happens because the output of the air-ground matching algorithm may still contain a small amount of outlier matching points, and more importantly, the 3D coordinates of the projected Street View image points have inaccuracies because in the 3D cadastral city model, the nonplanar parts of the facades, e.g., windows and balconies, are not modeled. In the future, by using more detailed city models, this kind of error source could be eliminated.

We refine the resulting camera pose estimate using the Levenberg-Marquardt (Hartley & Zisserman, 2004) optimization, which minimizes the reprojection error given by the sum of the squared distances between the observed image points and the reprojected 3D points. Finally, using only the inlier points, we compute the MAV camera position.

Figures 11(a)–11(c) show examples of how the Street View images are backprojected onto the 3D city model. Moreover, Figure 11(d) shows the estimated camera positions and orientations in the 3D city model for a series of consecutive MAV images. As we do not have an accurate ground-truth (we only have the GPS poses of the MAV), we visually evaluate the accurateness of the position estimate by rendering-out the estimated MAV camera view and comparing it to the actual MAV image for a given position, as shown in Figures 11(e) and 11(f). Figures 11(g)–11(i) again show another example of the estimated camera position (g), the synthesized camera view (h), and the actual MAV image (i).

By comparing the actual MAV images to the rendered-out views [Figures 11(e) and 11(f) and Figures 11(h)–11(i)], it can be noted that the orientation of the flying vehicle is correctly computed by the presented approach. It is very important to correct the orientation of the vehicle in order to correct the accumulated drift by the incremental visual odometry system used for the position tracking of the vehicle. It can be noticed that the position of the vehicle along the street is also correct. However, in the direction

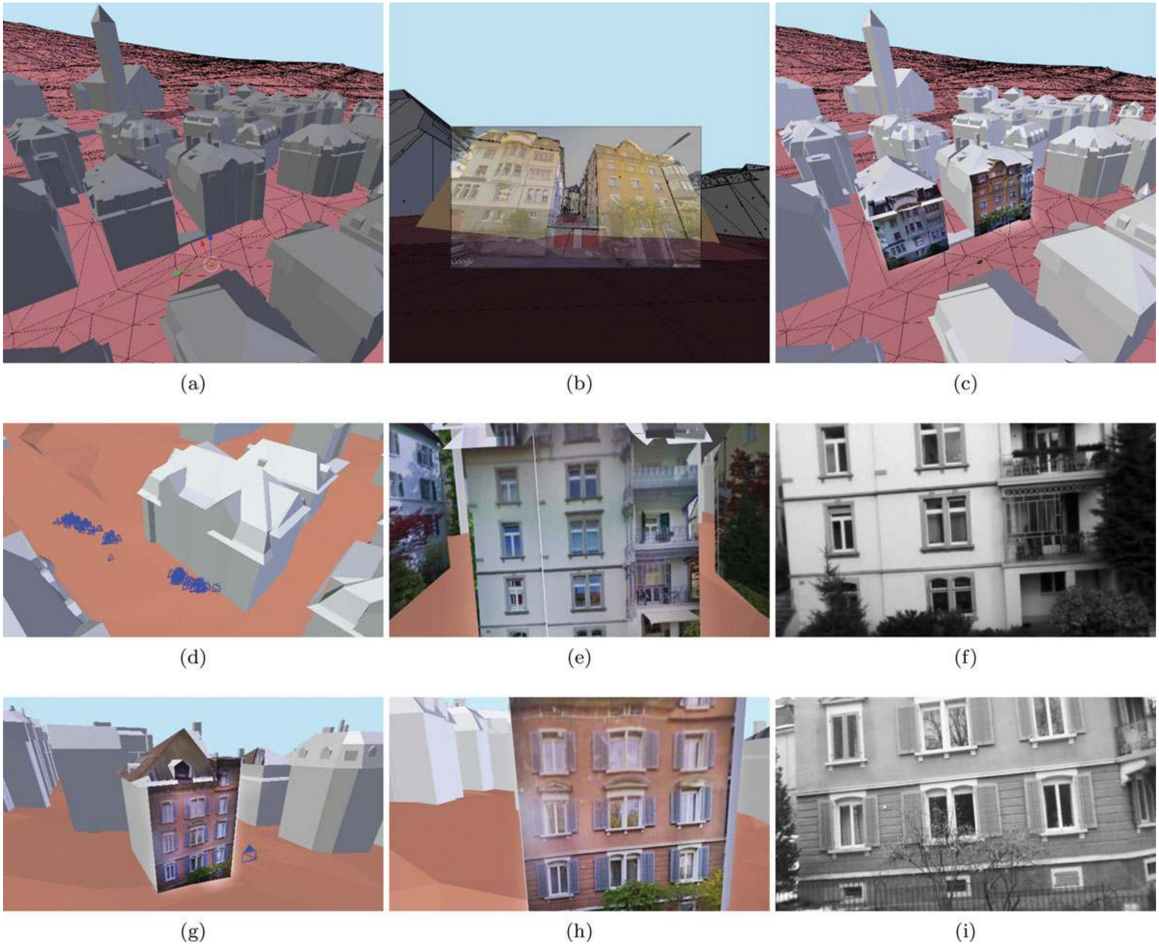


Figure 11. (a) Perspective view of the cadastral 3D city model; (b) the ground-level Street View image overlaid on the model; (c) the backprojected texture onto the cadastral 3D city model; (d) estimated MAV camera positions matched with one Street View image; (e) the synthesized view from one estimated camera position corresponding to an actual MAV image (f); (g)–(i) show another example from our dataset, where (g) is an aerial view of the estimated camera position (h), which is marked with the blue camera in front of the textured 3D model, and (h) is the synthesized view from the estimated camera position corresponding to an actual MAV image (i).

perpendicular to the street, the position still has a small error. This is due to the inaccurateness of the used 3D city model. In the cadastral model, the windows and other small elements that are not exactly in the main plain of the facade are not modeled. Similar results were derived for the remaining MAV-Street View image pairs of the recorded dataset.

The minimal number of correspondences required for the EPnP algorithm is $s = 4$. However, if a nonminimal set of points is randomly selected, then $s > 4$ (in our experiments we used a nonminimal set of points with $s = 8$ matches), and more robust results are obtained (cf. Figure 21).

The results are further improved by estimating the uncertainty of an appearance-based global positioning system using a Monte Carlo approach (Section 5.3). Figures 21(e) and 21(h) show the results of the vision-based estimates filtered using the computed covariance. Note that all the erroneous localizations are removed.

The appearance-based global-localization updates will be used in the next section to correct the accumulated drift in the trajectory of the MAV.

5. POSITION TRACKING

The goal of this section is to integrate the appearance-based global localization algorithm detailed in the previous section into the position-tracking algorithm that estimates the state of the MAV over time. Our aim is to show an application of the vision-based localization system by updating

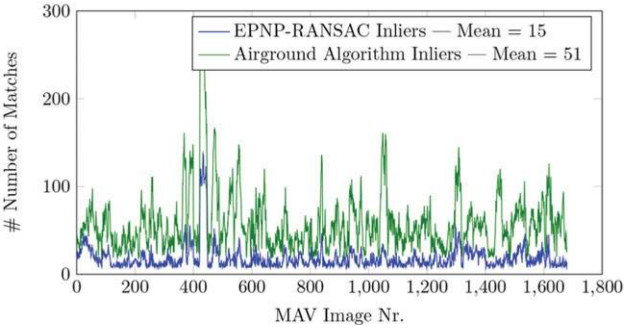


Figure 12. This figure shows the number of detected air-ground matches (green: 2D-2D matching points) for thr MAV—Street View image pairs and the resulting number of matches (blue: 3D-2D matching points) after applying the EPnP-RANSAC algorithm. The number of 3D-2D point correspondences is reduced in comparison to the 2D-2D matching points. This is because of the errors in the backprojection of the Street View images on the cadastral model and the inaccuracies of the 3D model.

the state of the MAV whenever an appearance-based global position measurement becomes available.

The vehicle state at time k is composed of the position vector and the orientation of the airborne image with respect to the global reference system. To simplify the proposed algorithm, we neglect the roll and pitch, since we assume that the MAV flies in near-hovering conditions. Consequently, we consider the reduced state vector $q_k \in \mathbb{R}^4$,

$$q_k := (p_k, \theta_k), \tag{2}$$

where $p_k \in \mathbb{R}^3$ denotes the position and $\theta_k \in \mathbb{R}$ denotes the yaw angle.

We adopt a Bayesian approach (Thrun, Burgard, Fox, 2005) to track and update the position of the MAV. We compute the posterior probability density function (PDF) of the state in two steps. To compute the prediction update of the Bayesian filter, we use visual odometry. To compute the measurement update, we integrate the global position, as soon as this is made available by the algorithm described in the previous section.

The system model f describes the evolution of the state over time. The measurement model h relates the current measurement $z_k \in \mathbb{R}^4$ to the state. Both are expressed in a probabilistic form:

$$q_{k|k-1} = f(q_{k-1|k-1}, u_{k-1}), \tag{3}$$

$$z_k = h(q_{k|k-1}), \tag{4}$$

where $u_{k-1} \in \mathbb{R}^4$ denotes the output of the visual odometry algorithm at time $k-1$, $q_{k|k-1}$ denotes the prediction estimate of q at time k, and $q_{k-1|k-1}$ denotes the updated estimate of q at time $k-1$.

5.1. Visual Odometry

Visual odometry (VO) is the problem of incrementally estimating the egomotion of a vehicle using its onboard camera(s) (Scaramuzza & Fraundorfer, 2011). We use the VO algorithm from Wu, Agarwal, Curless, & Seitz (2011) to incrementally estimate the state of the MAV.

5.2. Uncertainty Estimation and Propagation of the VO

At time k, VO takes two consecutive images I_k, I_{k-1} as input and returns an incremental motion estimate with respect to the camera reference system. We define this estimate as $\delta^*_{k,k-1} \in \mathbb{R}^4$,

$$\delta^*_{k,k-1} := (\Delta s^*_k, \Delta\theta_k), \tag{5}$$

where $\Delta s^*_k \in \mathbb{R}^3$ denotes the translational component of the motion, and $\theta_k \in \mathbb{R}$ denotes the yaw increment. Δs^*_k is valid up to a scale factor, thus the metric translation $\Delta s_k \in \mathbb{R}^3$ of the MAV at time k with respect to the camera reference frame is equal to

$$\Delta s_k = \lambda \Delta s^*_k. \tag{6}$$

We define $\delta_{k,k-1} \in \mathbb{R}^4$ as

$$\delta_{k,k-1} := (\Delta s_k, \Delta\theta_k), \tag{7}$$

where $\lambda \in \mathbb{R}$ represents the scale factor. We describe the procedure to estimate λ in Section 5.5.

We estimate the covariance matrix $\Sigma_{\delta_{k,k-1}} \in \mathbb{R}^{4\times4}$ using the Monte Carlo technique (Thrun, Fox, Burgard, & Dellaert, 2001). The VO at every step of the algorithm provides an incremental estimate $\delta_{k,k-1}$, together with a set of corresponding image points between image I_k and I_{k-1}. We randomly sample five couples from the corresponding point set multiple times (1,000 in our experiments). Each time, we use the selected samples as an input to the five-point algorithm (Nistér, 2004) to obtain the estimate $\{\delta_i\}$. All these estimates form $\mathcal{D} = \{\delta_i\}$. Finally, we calculate the uncertainty $\Sigma_{\delta_{k,k-1}}$ of the VO by computing the sample covariance from the data.

The error of the VO is propagated throughout consecutive camera positions as follows. At time k, the state $q_{k|k-1}$ depends on $q_{k-1|k-1}$ and $\delta_{k,k-1}$,

$$q_{k|k-1} = f(q_{k-1|k-1}, \delta_{k,k-1}). \tag{8}$$

We compute its associated covariance $\Sigma_{q_{k|k-1}} \in \mathbb{R}^{4x4}$ by the error-propagation law:

$$\Sigma_{q_{k|k-1}} = \nabla f_{q_{k-1|k-1}} \Sigma_{q_{k-1|k-1}} \nabla f^T_{q_{k-1|k-1}} + \nabla f_{\delta_{k,k-1}} \Sigma_{\delta_{k,k-1}} \nabla f^T_{\delta_{k,k-1}}, \tag{9}$$

assuming that $q_{k-1|k-1}$ and $\delta_{k,k-1}$ are uncorrelated. We compute the Jacobian matrices numerically. The rows of the Jacobian matrices $\nabla ({}^i f_{q_{k-1|k-1}})$, $\nabla ({}^i f_{\delta_{k,k-1}}) \in \mathbb{R}^{1x4}$ $(i = 1,2,3,4)$

are computed as

$$\nabla\,({}^i f_{q_{k-1|k-1}}) = \left[\ \frac{\partial({}^i f)}{\partial({}^1 q_{k-1|k-1})}\ \frac{\partial({}^i f)}{\partial({}^2 q_{k-1|k-1})}\ \frac{\partial({}^i f)}{\partial({}^3 q_{k-1|k-1})}\ \frac{\partial({}^i f)}{\partial({}^4 q_{k-1|k-1})}\ \right],$$

$$\nabla({}^i f_{\delta_{k,k-1}}) = \left[\ \frac{\partial({}^i f)}{\partial({}^1 \delta_{k,k-1})}\ \frac{\partial({}^i f)}{\partial({}^2 \delta_{k,k-1})}\ \frac{\partial({}^i f)}{\partial({}^3 \delta_{k,k-1})}\ \frac{\partial({}^i f)}{\partial({}^4 \delta_{k,k-1})}\ \right], \qquad (10)$$

where ${}^i q_{k-1|k-1}$ and ${}^i \delta_{k,k-1}$ denote the ith component of $q_{k-1|k-1}$ and $\delta_{k,k-1}$, respectively. The function ${}^i f$ relates the updated state estimate $q_{k-1|k-1}$ and the VO output $\delta_{k,k-1}$ to the ith component of the predicted state ${}^i q_{k|k-1}$.

In conclusion, the state covariance matrix $\Sigma_{q_{k|k-1}}$ defines an uncertainty space (with a confidence level of 3σ). If the measurement z_k that we compute by means of the appearance-based global positioning system is not included in this uncertainty space, we do not update the state and we rely on the VO estimate.

5.3. Uncertainty Estimation of the Appearance-based Global Localization

Our goal is to update the state of the MAV denoted by $q_{k|k-1}$ whenever an appearance-based global position measurement $z_k \in \mathbb{R}^4$ is available. We define z_k as

$$z_k := (p_k^S, \theta_k^S), \qquad (11)$$

where $p_k^S \in \mathbb{R}^3$ denotes the position and $\theta_k^S \in \mathbb{R}$ denotes the yaw in the global reference system at time k.

The appearance-based global positioning system provides the index $j \in \mathbb{N}$ of the Street View image corresponding to the current MAV image, together with two sets of $n \in \mathbb{N}$ 2D corresponding image points between the two images. Furthermore, it provides the 3D coordinates of the corresponding image points in the global reference system. We define the set of 3D coordinates as $X^S: = \{x_i^S\}$ ($\{x_i^S\} \in \mathbb{R}^3\ \forall\ i = 1, \ldots, n$) and the set of 2D coordinates as $\mathcal{M}^D = \{m_i^D\}$ ($\{m_i^D\}, \in \mathbb{R}^2\ \forall\ i = 1, \ldots, n$).

If a MAV image matches a Street View image, it cannot be farther than 15 m from that Street View camera according to our experiments (cf. Figure 6). We illustrate the uncertainty bound of the MAV in a bird's-eye view in Figure 13 with a green ellipse, where blue dots represent Street View camera positions. To reduce the the uncertainty associated with z_k, we use the two sets of corresponding image points.

We compute z_k such that the reprojection error of X^S with respect to $\mathcal{M}^D$ is minimized, that is,

$$z_k = \underset{z}{argmin}\left(\sum_{i=1}^{n}\left\| m_i^D - \pi(x_i^S, z)\right\|\right), \qquad (12)$$

where π denotes the jth Street View camera projection model.

The reprojected point coordinates $\pi(x_i^S, z)$ are often inaccurate because of the uncertainty of the Street View camera poses and that of the 3D model data. The $\mathcal{M}^D$, $\mathcal{X}^S$ sets may contain outliers. We choose then EPnP-RANSAC to compute z_k, selecting the solution with the highest consensus (maximum number of inliers, minimum reprojection error).

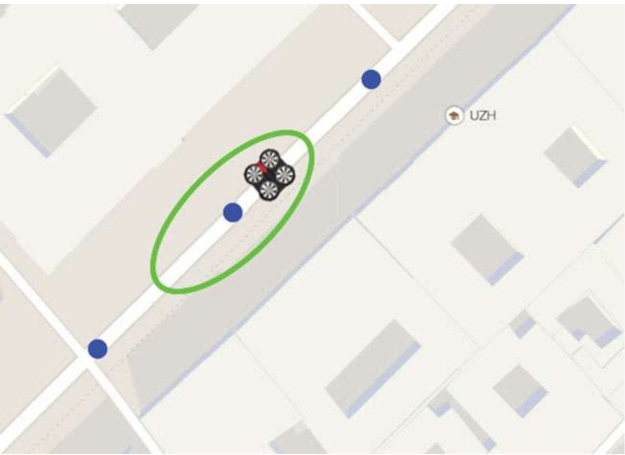


Figure 13. Blue dots represent Street View cameras. If the MAV current image matches with the central Street View one, the MAV must lie in an area of 15 m around the corresponding Street View camera. We display this area with a green ellipse.

Similarly to Section 5.2, we estimate the covariance matrix $\Sigma_{z_k} \in \mathbb{R}^{4x4}$ using the Monte Carlo technique as follows. We randomly sample m corresponding pairs between $\mathcal{M}^D$ and $\mathcal{X}^S$ multiple times (1,000 in the experiments). Each time, we use the selected samples as an input to the EPnP algorithm to obtain the measurement $\{z_i\}$. As we can see in Figure 6, a match with images gathered by Street View cameras farther than 15 m is not plausible. We use this criterion to accept or discard $\{z_i\}$ measurements. All the plausible estimates form the set $\mathcal{Z} = \{z_i\}$. We estimate Σ_{z_k} by computing the sample covariance from the data.

Figure 14 shows the estimated uncertainties of the global localization algorithm for a section of the entire 2 km dataset (Section 6.1). Further details are given in Figure 15, where the Monte-Carlo-based standard deviations are shown along the x, y, and z coordinates and the yaw angle of the vehicle. Based on the computed covariances, a simple filtering rule is used to discard those vision-based position estimates that have a very high uncertainty. Conversely, the appearance-based global positions with high confidence are used to update the position tracking system of the MAV. By applying such an approach, the results can be greatly improved [cf. Figures 21(e) and 21(h)], although the total number of global position updates will be reduced.

5.4. Fusion

We aim to reduce the uncertainty associated with the state by fusing the prediction estimate with the measurement whenever an appearance-based global position measurement is available. The outputs of this fusion step are the

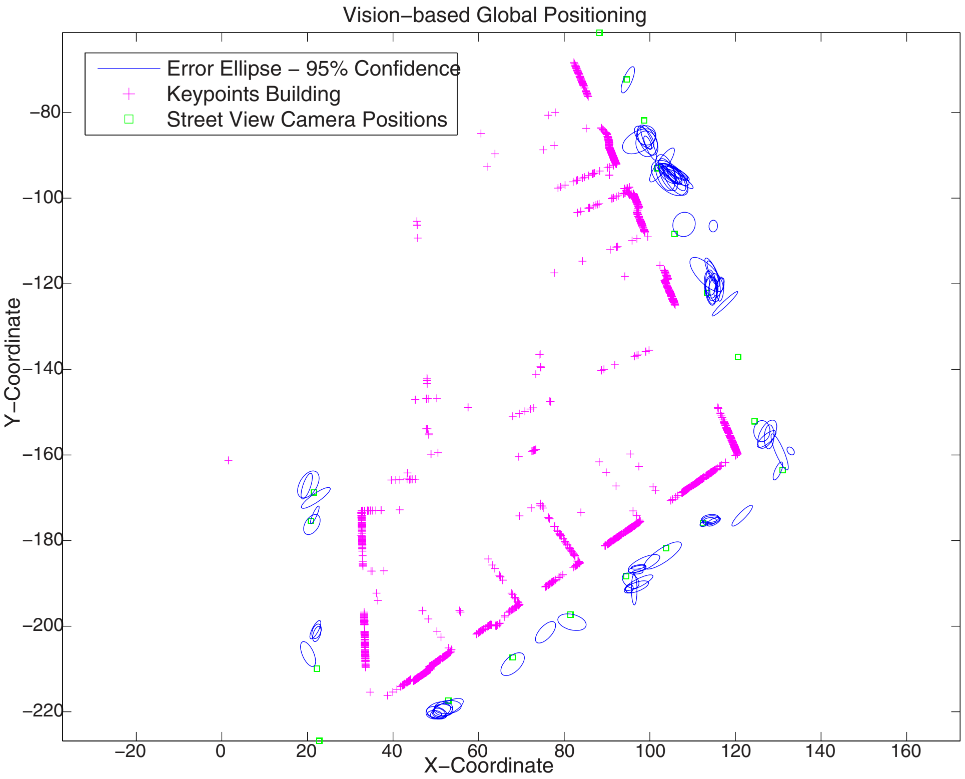


Figure 14. The figure shows the top view of an enlarged subpart of the full trajectory. The blue ellipses show the 95% confidence intervals of the appearance-based global positioning system computed using the outlined Monte Carlo approach. The green boxes correspond to the Street View camera positions. The magenta crosses show the positions of the matched 3D feature points on the building facades. Note that most of the confidence intervals border a reasonably small area, meaning that the accuracy of the vision-based positioning approach can accurately localize the MAV in the urban environment.

updated estimate $q_{k|k}$ and its covariance $\Sigma_{q_{k|k}} \in \mathbb{R}^{4x4}$. We compute them according to Kalman filter equations (Kalman et al., 1960):

$$q_{k|k} = q_{k|k-1} + \Sigma_{q_{k|k-1}}(\Sigma_{q_{k|k-1}} + \Sigma_{z_k})^{-1}(z_k - q_{k|k-1}), \quad (13)$$

$$\Sigma_{q_{k|k}} = \Sigma_{q_{k|k-1}} - \Sigma_{q_{k|k-1}}(\Sigma_{q_{k|k-1}} + \Sigma_{z_k})^{-1}\Sigma_{q_{k|k-1}}. \quad (14)$$

5.5. Initialization

To initialize our system, we use the global localization algorithm, i.e., we use Eq. (12) to compute the initial state $q_{0|0}$ and the Monte Carlo procedure described in Section 5.3 to estimate its covariance $\Sigma_{q_{0|0}}$. In the initialization step, we also estimate the absolute scale factor λ for visual odometry. After the initial position, we need another position of the MAV that is globally localized by our appearance-based approach. Finally, we compute λ by comparing the metric distance traveled, which is computed by the two global localization estimates, with the unscaled motion estimate returned by the VO.

6. EXPERIMENTS AND RESULTS

This section presents the results in two parts. First, the air-ground matching algorithm is evaluated. Second, the results of the appearance-based global positioning system are presented, together with the position-tracking algorithm.

6.1. Air-ground Matching Algorithm Evaluation

We collected a dataset in downtown Zurich, Switzerland; cf., Appendix A. A commercially available Parrot AR.Drone 2 flying vehicle (equipped with a camera—standard mounting) was manually piloted along a 2 km trajectory, collecting images throughout the environment at different flying altitudes up to 20 m by keeping the MAV camera always facing the buildings. Sample images are shown in Figure 2, left column. For more insights, the reader can watch the video file accompanying this article.[7] The full dataset consists of more than 40,500 images. For all the experiments presented in this work, we subsampled the data selecting one image

[7] http://rpg.ifi.uzh.ch.

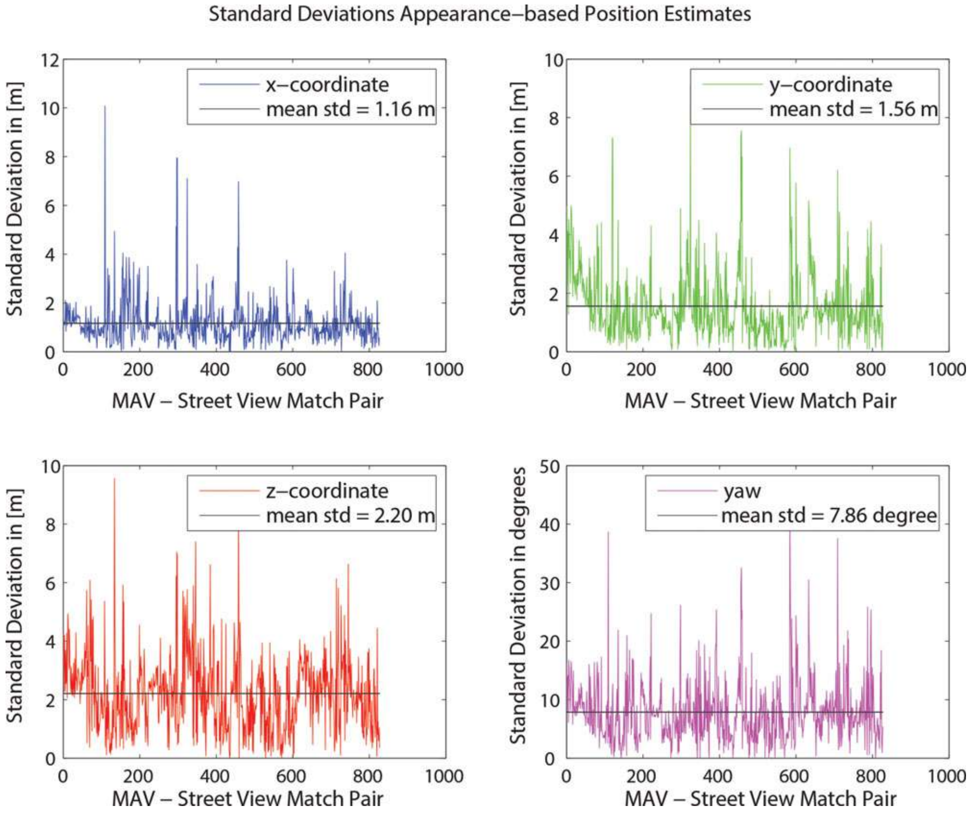


Figure 15. The figure shows the standard deviations computed for matching MAV - Street View image pairs along the *x*, *y*, and *z* coordinates (m) and the yaw angle (degrees) of the vehicle. The mean standard deviation for the *x* coordinate is 1.16 m, for the *y* coordinate it is 1.56 m, and for the *z* coordinate it is slightly bigger, namely 2.20 m. The mean standard deviation for the yaw is 7.86 degrees. Note that in case the uncertainty of the appearance-based global localization algorithm is very large, it is discarded and another image is used to localize the vehicle.

every 100, resulting in a total number of 405 MAV test images. In all the experiments, we used an image resolution of 640 × 360 pixels. All the available Street View data covering the test area were downloaded and saved locally, resulting in 113 discrete possible locations. Since all the MAV test images should have a corresponding terrestrial Street View image, the total number of possible correspondences is 405 in all evaluations. We manually labeled the data to establish the ground-truth, namely the exact visual overlap between the aerial MAV images and the Street View data. The Street View pictures were recorded in the summer of 2009, while the MAV dataset was collected in the winter of 2012; thus, the former is outdated in comparison to the latter. Furthermore, the aerial images are also affected by motion blur due to the fast maneuvers of the MAV. Figure 16 shows the positions of the Street View images (blue dots) overlaid to an aerial image of the area. Also, correctly matched MAV image locations—for which a correct most similar Street View image was found—are shown (green circle).

The different visual-appearance–based algorithms were evaluated in terms of *recall rate*[8] and *precision rate*.[9] We also show the results using a different visualization, namely confusion maps. Figure 8 depicts the results obtained by applying the five conventional methods discussed in Section 3.5 and the algorithm proposed in this work [Figure 8(d)]. The confusion matrix shows the visual similarity computed between all the Street View (vertical axes) images and all the MAV test images (horizontal axes). To display the confusion maps, we used intensity maps, colored as heat maps. Dark blue represents no visual similarity, while dark red denotes a complete similarity. An ideal image-pairing algorithm would detect a confusion matrix coincident to the ground-truth matrix [Figure 8(a)]. A stronger

[8] Recall rate = number of detected matches over the total number of possible correspondences.

[9] Precision rate = number of true positives detected over the total number of matches detected (both true and false).

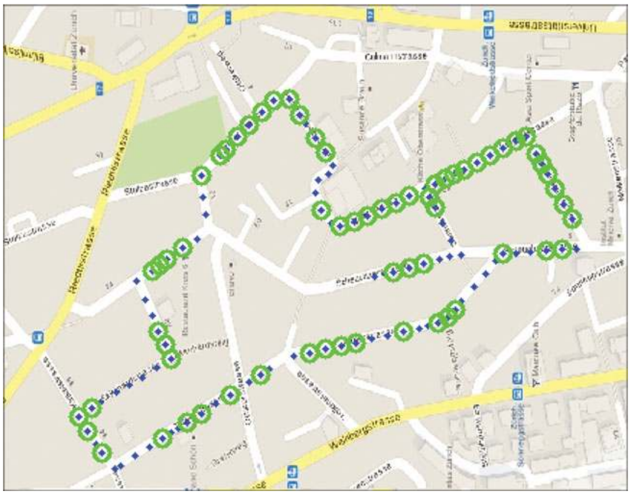


Figure 16. Bird's-eye view of the test area. The blue dots mark the locations of the ground Street View images. The green circles represent those places where the aerial images taken by the urban MAV were successfully matched with the terrestrial image data.

deviation from the ground-truth map shows less accurate results.

6.1.1. Parameters Used in the Experiments

For the bag-of-words[10] approach in Figures 8(e) and 17, a hierarchical vocabulary tree was trained with a *branching factor* of $k = 10$ and *depth levels* of $L = 5$, resulting in $k^L = 100,000$ leaves (visual words) (using both MAV images and Street View images recorded in a neighborhood similar to our test area). The term *frequency-inverse document frequency tf-idf* was used as a weighting type and the L1-Norm was used as a scoring type. In the case of the FAB-MAP[11] algorithm, several parameters were tested to get meaningful results. However, all the checked parameter configurations failed on our dataset. For the experiments presented in the paper, the *FAB-MAP Vocabulary 100k Words* was used. Moreover, a motion model was assumed (bias forward 0.9), and the geometric consistency check was turned on. The other parameters were set according to the recommendations of the authors. For our proposed air-ground matching algorithm, we used the SIFT feature detector and descriptor, but our approach can be adapted easily to use other features as well.

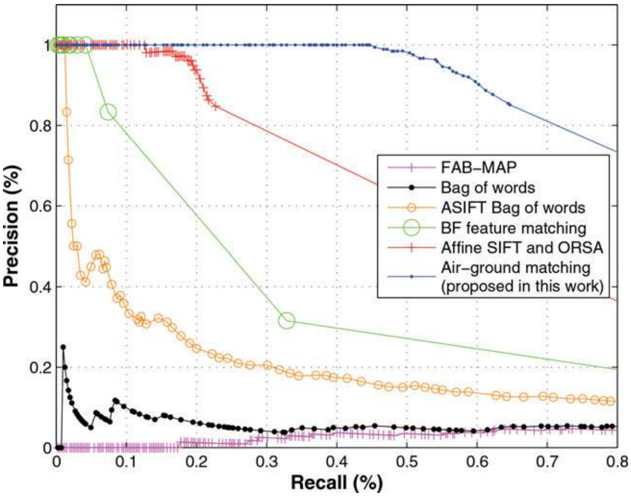


Figure 17. Comparison of the results. Please note that at precision 1, the proposed air-ground matching algorithm greatly outperforms the other methods in terms of recall. To visualize all the correctly matched airborne MAV images with the Street View images, please consult the video attachment of the paper.

6.1.2. Results and Discussion of the Experiments

Figure 17 shows the results in terms of precision and recall. In contrast to object recognition algorithms, in which the average precision is used to evaluate the results, in robotic applications the most important evaluation criterion is usually the recall rate at precision 1. This criterion represents the total number of true-positive detections without having any false-positive match.

Considering the recall rate at precision 1, our proposed *air-ground matching* algorithm (shown with blue in Figure 17) outperforms the second best approach, namely the *ASIFT and ORSA* (red) by a factor of 4. This is because, in our approach, the artificial-views are simulated in a more efficient way. Moreover, to reject the outliers, we use a graph-matching method that extends the pure photometric matching with a graph-based one. These results are even more valuable since the *ASIFT and ORSA* algorithm was applied in a *brute-force fashion*, which is computationally very expensive. In contrast, in the case of our proposed algorithm, we applied the extremely fast putative-match selection method. That is, the results were obtained by selecting just 7% from the total number of Street View images. We show all the correctly matched MAV images with Street View images in the video file accompanying this article, which gives further insight about our air-ground matching algorithm. As observed, other traditional methods, such as the *Visual Bag-of-Words* approach (shown with black in Figure 17), *ASIFT Bag-of-Words* (orange), and *FAB-MAP* (magenta) fail in matching our MAV images with ground-level Street View data. Apparently, these algorithms fail because the visual patterns present in both images are classified in

[10] We used the implementation of Galvez-Lopez & Tardos (2012), publicly available at http://webdiis.unizar.es/dorian/

[11] We used the implementation of Cummins & Newman (2011), publicly available at http://www.robots.ox.ac.uk/ mobile/

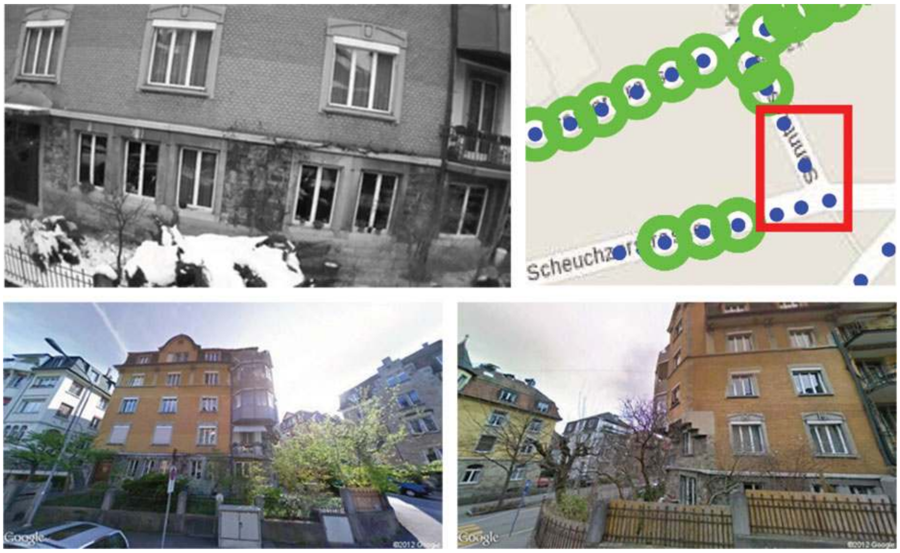


Figure 18. Analysis of the first false-positive detection. Top-left: urban MAV image; top-right: zoom on the global map, where the image was taken; bottom-left: detected match; bottom-right: true positive pairing according to manual labeling. Please note that our algorithm fails for the first time in a situation in which the MAV is facing the same building from two different sides (streets), having in the field of view only windows with the same patterns.

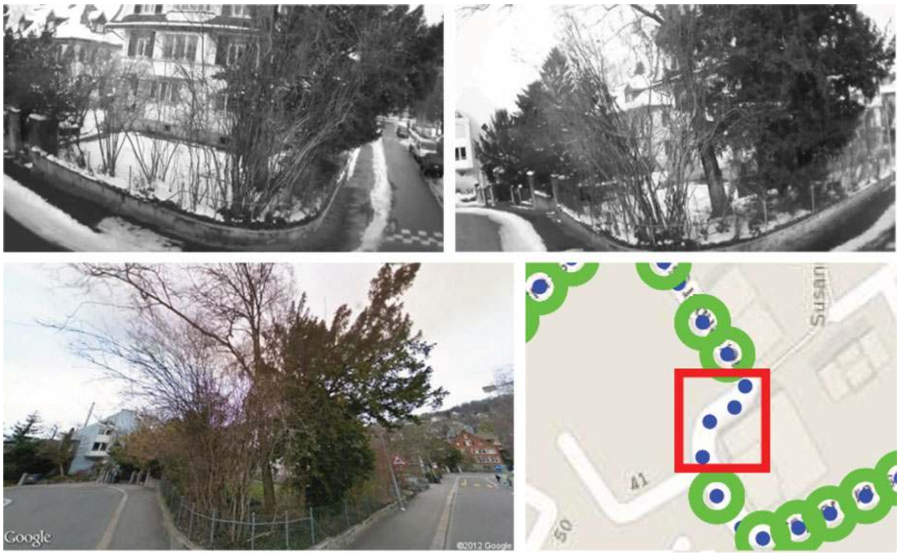


Figure 19. Analysis in case of no detections. Top-left: urban MAV image; top-right: next view of the urban MAV; bottom-left: true positive pairing according to manual labeling; bottom-right: zoom on the global map, where the image was taken. Please note that these robot positions (top raw) are difficult to recognize even for humans. Moreover, the over-season change of the vegetation makes it extremely difficult to cope with their pairing for image feature-based techniques.

different visual words, thus leading to false visual-word associations.

Figure 18 shows the first false-positive detection of our air-ground matching algorithm. After a more careful analysis, we found that this is a special case in which the MAV was facing the same building from two different sides (i.e., from different streets), having only windows with the same patterns in the field of view. Repetitive structures represent a barrier for visual-appearance–based localization algorithms, which can be solved by taking motion into account using a Bayesian fashion, as explained in Section 5. The limitations of the proposed method are shown in Figure 19. Please note that these robot positions (top row) are difficult to recognize even for humans.

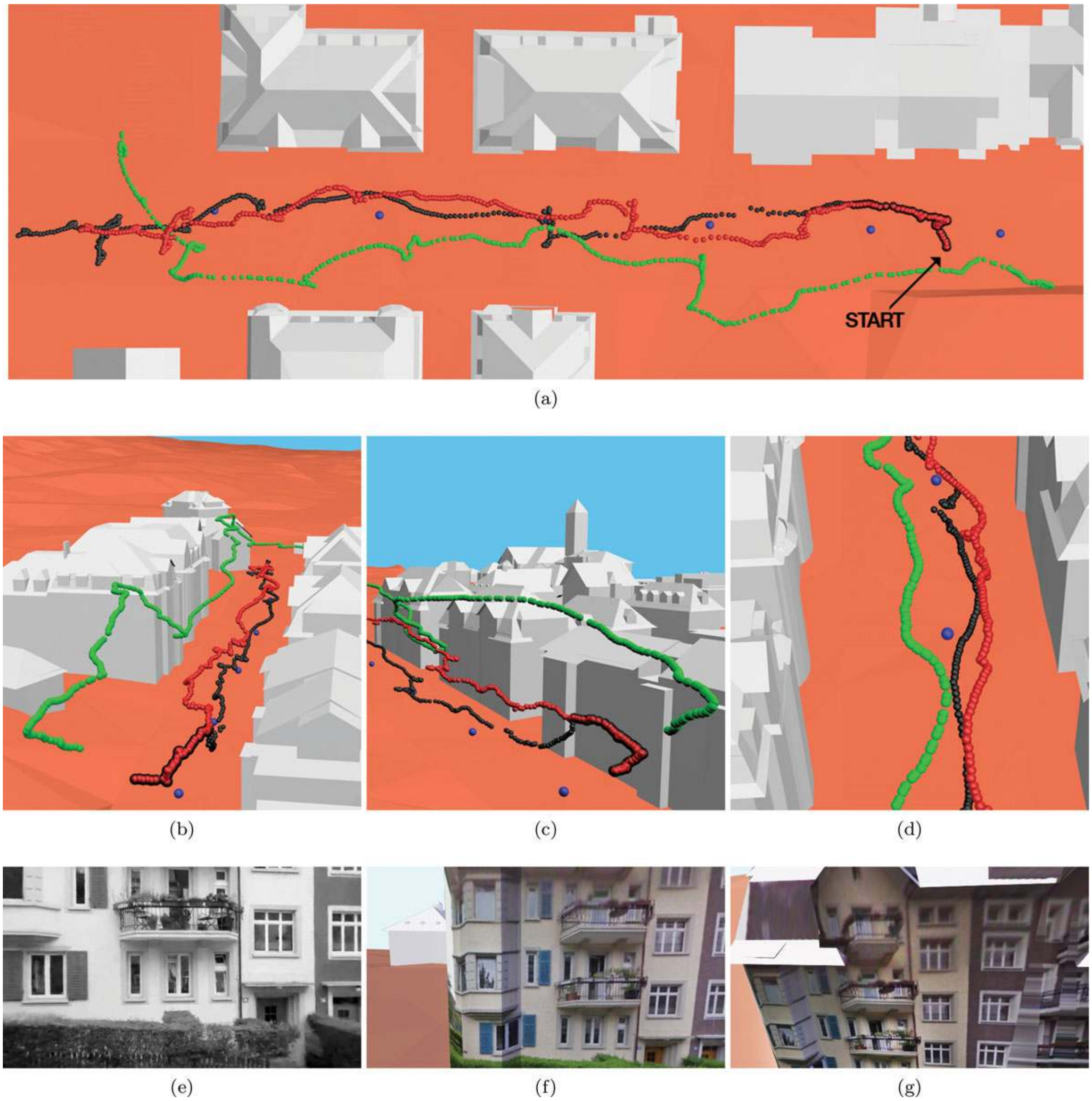


Figure 20. Comparison between path estimates shown within the cadastral 3D city model: Top row: top view of the estimated trajectory of the MAV; we display the Street View image locations in blue, the visual odometry estimate in black, GPS in green, and our estimate in red. Middle row: (b) and (c) altitude evaluation: in the experiment, the MAV flew close to the middle of the street and it never flew over a height of 6 m (above the ground). From this point of view, our path estimate (red) is more accurate than the GPS one (green); (d) enlarged view of the path estimates. Bottom row: we show a visual comparison of (e) the actual view; (f) the rendered view of the textured 3D model corresponding to (e) that the MAV perceives according to our estimate; and (g) the rendered view of the textured 3D model corresponding to (e) that the MAV perceives according to the GPS measurement. To conclude, the algorithm presented in this paper outperforms the other techniques to estimate the trajectory of the MAV flying at low altitudes in an urban environment.

6.2. Appearance-based Global Positioning System Experiments

We collected a second dataset in downtown Zurich using the same platform. The MAV was piloted along a 150 m trajectory, collecting images throughout the environment at different altitudes up to 6 m. The images are synchronized with the GPS data based on their timestamps. Every image is considered a state of the MAV. For the visual odometry, we

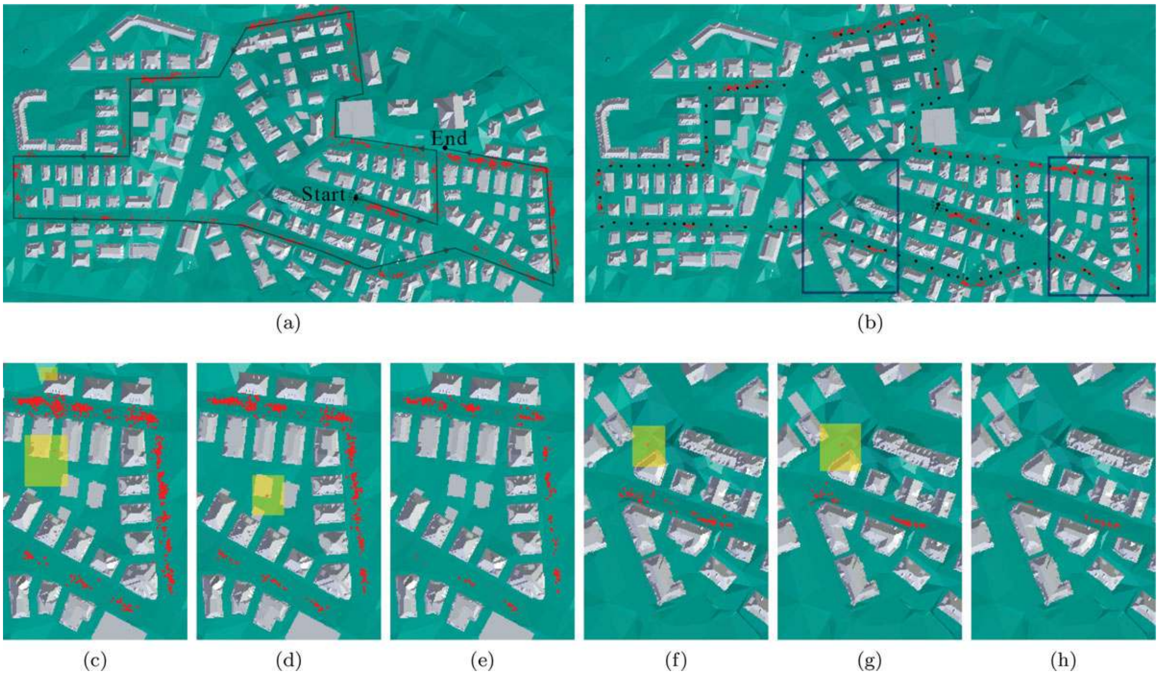


Figure 21. Estimated poses of the MAV along the 2 km trajectory: (a) EPnP-RANSAC with a minimal set of $s = 4$ points; (b) EPnP-RANSAC with a nonminimal set of $s = 8$ matches. The black dots show the location of the geotagged Street View data. Enlarged area 1 for comparison between minimal points set (c), nonminimal points set (d), and filtered using the uncertainty estimation (e); enlarged area 2 for comparison between minimal points set (f), nonminimal points set (g), and filtered using the uncertainty estimation (h). Note that by applying the EPnP-Ransac algorithm with minimal and nonminimal points sets, a few erroneous localizations are computed (highlighted with yellow rectangles in the figures); however, in case the results are filtered based on the uncertainty estimation proposed in this paper, the erroneous positions are completely eliminated [(e) and (h)].

used an average frame rate of three images per meter, and we assumed close-to-hover flight conditions for recording the images. Although we do not have an accurate ground-truth path of the MAV to compare with (since the GPS signal is shadowed between the buildings), we can still evaluate visually the performance of our system (cf. Figure 11). Furthermore, we display our result within the cadastral 3D city model, which can provide a good basis to evaluate the result (cf. Figure 20).

6.2.1. Results and Discussion of the Experiments

In Figures 20(a)–20(d), we display the results using the cadastral 3D model in order to evaluate the trajectories with respect to the surrounding buildings. The Street View image locations are shown in blue, the VO estimate is shown in black, the GPS in green, and our estimate in red. To reduce the drift in the VO estimate, we constrained the orientation to be aligned with the street's dominant direction. However, note in Figure 20(a) that the VO estimate (black) accumulates a significant error alongside the direction of the street. In addition, note that the estimate shown with red, computed with the proposed approach, is the most plausible since the vehicle was navigated close to the center of the street. Thus, our estimate is the most similar to the actual one. The altitude estimate error of the GPS is even more notable in Figures 20(b) and 20(c).

The rendered view of the textured 3D model—which the MAV perceives at the end of the trajectory—is visually more similar to the real one [Figure 20(e)] when it is estimated by the presented algorithm [Figure 20(f)], in comparison with the rendered view computed based on the GPS measurement [Figure 20(g)].

Finally, we show our result in Figure 21, where a bird's-eye view of the 2-km-long test environment is presented. A comparison is shown between the results obtained with the minimal set of correspondences for the EPnP algorithm [Figure 21(a)] and the nonminimal case [Figure 21(b)]. The red points show the computed MAV camera positions with the EPnP-RANSAC algorithm. Enlarged areas are shown in Figures 21(c) and 21(f) for the minimal case and Figures 21(d) and 21(g) for the nonminimal case. By closely comparing these figures, it can be concluded that the position estimates computed from a nonminimal set are more accurate than those from the minimal set. This is illustrated by the fact that the nonminimal position estimates tend to be more organized along smoother trajectories, which is in agreement with the real MAV flight path. Stated differently, the position estimates derived from the minimal set tend to "jump around" more than those from the nonminimal set, i.e., they are more widely spread and less spatially consistent. The reason for more accurate results in the nonminimal case is that the position estimates derived by the EPnP are less affected by outliers and degenerate point configurations. However, in both approaches—the minimal and the nonminimal—a few extreme outliers occur that are clearly not along the flying path, as highlighted by the yellow boxes in Figure 21. One possible cause for these outliers is wrong point correspondence between the Street View images and the MAV images. Another potential explanation is inaccurate 3D point coordinates supplied to EPnP resulting from inaccuracies when the overlay of the Street View images with the cadastral city model is not perfect.

6.2.2. Lessons Learned

Matching airborne images to ground-level ones is a challenging problem because extreme changes in viewpoint and scale can be found between the aerial MAV images and the ground-level images, in addition to large changes of illumination, lens distortion, over-season variation of the vegetation, and scene changes between the query and the database images. Only a complex visual-search algorithm can deal with such a scenario. We demonstrated that a multirotor MAV flying in urban streets, where a satellite GPS signal is often shadowed by the presence of buildings, or is completely unavailable, can be localized by using a textured 3D city model of the environment. Although the 3D city model contains inaccuracies—e.g., the nonplanar parts of the facades are not modeled (windows, balconies, etc.)—the MAV can be accurately localized by means of uncertainty quantification. This paper presented a proof-of-concept appearance-based global positioning system that could be readily implemented in real-time with cloud computing.

7. CONCLUSIONS

To conclude, this work addressed the air-ground matching problem between low-altitude MAV-based imagery and ground-level Street View images. Our algorithm outperforms conventional place-recognition methods in challenging settings, where the aerial vehicle flies over urban streets up to 20 m, often close to buildings. The presented algorithm keeps the computational complexity of the system at an affordable level. A solution was described to globally localize MAVs in urban environments with respect to the surrounding buildings using a single onboard camera and geotagged street-level images together with a cadastral 3D city model. By means of visual inspection and uncertainty quantification, it was shown that the accuracy of the described vision-based approach outperforms that of satellite-based GPS. Therefore, vision-based localization can be either a viable alternative to GPS localization in urban areas, where the GPS signal is shadowed or completely unavailable, or a powerful complement to it in order to enhance localization in areas where the GPS signal strength is weak, i.e., where direct line of sight to the satellites may be obstructed. The presented appearance-based global positioning system is a step toward safe operation (i.e., takeoff, land, and navigate) of small-sized, autonomous aerial vehicles in urban environments equipped with vision sensors.

Acknowledgments

The authors are grateful to Aparna Taneja and Luca Ballan from the ETH Computer Vision and Geometry lab for providing corrected Street View image poses for the data used in this work. This research was supported by the Scientific Exchange Programme SCIEX-NMS-CH (no. 12.097), the Swiss National Science Foundation through project number 200021-143607 (Swarm of Flying Cameras), and the National Centre of Competence in Research (NCCR) Robotics.

APPENDIX A: ZURICH AIR-GROUND MATCHING DATASET

The air-ground matching dataset used in this work is publicly available at rpg.ifi.uzh.ch/data/air-ground-data.tar.gz

The dataset was collected in the downtown area of Zurich, Switzerland. A commercially available Parrot AR.Drone 2 flying vehicle (equipped with a camera, standard mounting) was manually piloted along a 2 km trajectory, collecting images throughout the environment at different flying altitudes up to 20 m by keeping the MAV camera always facing the buildings. The ground-truth was established by manually labeling the data in order to mark the exact visual overlap between the aerial MAV images and the Street View data.

The dataset consists of the following files:

- ./images/MAV Images/—This folder contains the images recorded by an MAV in the city of Zurich, Switzerland.
- ./images/Street View Images/—This folder contains the Street View images corresponding to the area recorded by the MAV.
- ./ground_truth.mat—This Matlab matrix file contains the human-made ground-truth for the MAV images, i.e., the overlap between each drone image and the Street View images.
- ./lat_long.txt—This file contains the GPS data (geotagging) for every database (Street View) image, and the format is according to the Google Street View API: 1)

latitude; 2) longitude; 3) yaw_degree; 4) tilt_yaw_degree; 5) tilt_pitch degree; 6) auxiliary variable.[12]

REFERENCES

Albarelli, A., Rodolà, E., & Torsello, A. (2012). Imposing semi-local geometric constraints for accurate correspondences selection in structure from motion: A game-theoretic perspective. International Journal of Computer Vision, 97(1), 36–53.

Anguelov, D., Dulong, C., Filip, D., Frueh, C., Lafon, S., Lyon, R., Ogale, A., Vincent, L., & Weaver, J. (2010). Google street view: Capturing the world at street level. Computer, 43(6), 32–38.

Baatz, G., Köser, K., Chen, D. M., Grzeszczuk, R., & Pollefeys, M. (2012). Leveraging 3d city models for rotation invariant place-of-interest recognition. International Journal of Computer Vision, 96(3).

Bansal, M., Daniilidis, K., & Sawhney, H. S. (2012). Ultra-wide baseline facade matching for geo-localization. In European Conference on Computer Vision Workshops and Demonstrations. Springer.

Bansal, M., Sawhney, H. S., Cheng, H., & Daniilidis, K. (2011). Geo-localization of street views with aerial image databases. In Proceedings of the 19th ACM International Conference on Multimedia (pp. 1125–1128). ACM.

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (surf). Computer Vision and Image Understanding, 110(3), 346–359.

Brubaker, M. A., Geiger, A., & Urtasun, R. (2013). Lost! leveraging the crowd for probabilistic visual self-localization. In IEEE Conference on Computer Vision and Pattern Recognition.

Churchill, W., & Newman, P. M. (2012). Practice makes perfect? Managing and leveraging visual experiences for lifelong navigation. In IEEE International Conference on Robotics and Automation (pp. 4525–4532).

Conte, G., & Doherty, P. (2009). Vision-based unmanned aerial vehicle navigation using geo-referenced information. EURASIP Journal on Advances in Signal Processing, 2009, 10.

Cummins, M., & Newman, P. (2011). Appearance-only SLAM at large scale with FAB-MAP 2.0. International Journal of Robotics Research, 30(9), 1100–1123.

DDPS (2008). Formulas and constants for the calculation of the swiss conformal cylindrical projection and for the transformation between coordinate systems. Technical report, Federal Department of Defence, Civil Protection and Sport DDPS.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6), 381–395.

Floros, G., Zander, B., & Leibe, B. (2013). Openstreetslam: Global vehicle localization using openstreetmaps. In IEEE International Conference on Robotics and Automation (pp. 1054–1059).

Fritz, G., Seifert, C., Kumar, M., & Paletta, L. (2005). Building detection from mobile imagery using informative sift descriptors. In Scandinavian Conference on Image Analysis. Springer.

Galvez-Lopez, D., & Tardos, J. D. (2012). Bags of binary words for fast place recognition in image sequences. IEEE Transactions on Robotics, 28(5), 1188–1197.

Hartley, R. I., & Zisserman, A. (2004). Multiple view geometry in computer vision, 2nd ed. Cambridge University Press.

Hentschel, M., & Wagner, B. (2010). Autonomous robot navigation based on openstreetmap geodata. In IEEE International Conference on Intelligent Transportation Systems (pp. 1645–1650).

Ibañez Guzmán, J., Laugier, C., Yoder, J.-D., & Thrun, S. (2012). Autonomous driving: Context and state-of-the-art. In Handbook of Intelligent Vehicles (vol. 2, pp. 1271–1310).

Jégou, H., Douze, M., & Schmid, C. (2011). Product quantization for nearest neighbor search. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(1), 117–128.

Kalman, R. E. et al. (1960). A new approach to linear filtering and prediction problems. Journal of Basic Engineering, 82(1), 35–45.

Kneip, L., Scaramuzza, D., & Siegwart, R. (2011). A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO.

Kuemmerle, R., Steder, B., Dornhege, C., Kleiner, A., Grisetti, G., & Burgard, W. (2011). Large scale graph-based SLAM using aerial images as prior information. Journal of Autonomous Robots, 30(1): 25–39.

Liu, Z., & Marlet, R. (2012). Virtual line descriptor and semi-local graph matching method for reliable feature correspondence. In British Machine Vision Conference (pp. 16.1–16.11).

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2), 91–110.

Maddern, W. P., Milford, M., & Wyeth, G. (2012). Cat-slam: Probabilistic localisation and mapping using a continuous appearance-based trajectory. International Journal of Robotics Research, 31(4), 429–451.

Majdik, A., Albers-Schoenberg, Y., & Scaramuzza, D. (2013). Mav urban localization from google street view data. In IEEE International Conference on Intelligent Robots and Systems (pp. 3979–3986).

Majdik, A., Gálvez-López, D., Lazea, G., & Castellanos, J. A. (2011). Adaptive appearance based loop-closing in heterogeneous environments. In IEEE International Conference on Intelligent Robots and Systems (pp. 1256–1263).

Majdik, A., Verda, D., Albers-Schoenberg, Y., & Scaramuzza, D. (2014). Micro air vehicle localization and position tracking from textured 3d cadastral models. In IEEE International Conference on Robotics and Automation.

[12] Please note that this paper is accompanied by videos available at http://youtu.be/CDdUKESUeLc. The dataset used in this work is available at rpg.ifi.uzh.ch/data/air-ground-data.tar.gz

Moisan, L., Moulon, P., & Monasse, P. (2012). Automatic homographic registration of a pair of images, with a contrario elimination of outliers. Image Processing On Line, 2, 56–73.

Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D., Ettinger, S., Haehnel, D., Hilden, T., Hoffmann, G., Huhnke, B., Johnston, D., Klumpp, S., Langer, D., Levandowski, A., Levinson, J., Marcil, J., Orenstein, D., Paefgen, J., Penny, I., Petrovskaya, A., Pflueger, M., Stanek, G., Stavens, D., Vogt, A., & Thrun, S. (2008). Junior: The stanford entry in the urban challenge. Journal of Field Robotics, 25(9), 569–597.

Morel, J.-M., & Yu, G. (2009). Asift: A new framework for fully affine invariant image comparison. SIAM Journal on Imaging Sciences, 2(2), 438–469.

Moreno-Noguer, F., Lepetit, V., & Fua, P. (2007). Accurate noniterative o(n) solution to the pnp problem. In IEEE International Conference on Computer Vision.

Muja, M., & Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In International Conference on Computer Vision Theory and Application VISSAPP (pp. 331–340).

Nistér, D. (2004). An efficient solution to the five-point relative pose problem. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(6), 756–770.

Scaramuzza, D. (2011). 1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. International Journal of Computer Vision, 95(1), 74–85.

Scaramuzza, D., & Fraundorfer, F. (2011). Visual odometry [tutorial]. IEEE Robotics & Automation Magazine, 18(4), 80–92.

Schindler, G., Brown, M., & Szeliski, R. (2007). City-scale location recognition. In IEEE Conference on Computer Vision and Pattern Recognition.

Sivic, J., & Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In IEEE International Conference on Computer Vision (pp. 1470–1477).

Taneja, A., Ballan, L., & Pollefeys, M. (2012). Registration of spherical panoramic images with cadastral 3d models. In International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT) (pp. 479–486). IEEE.

Thrun, S., Burgard, W., Fox, D., et al. (2005). Probabilistic robotics (vol. 1). Cambridge, MA: MIT Press.

Thrun, S., Fox, D., Burgard, W., & Dellaert, F. (2001). Robust monte carlo localization for mobile robots. Artificial intelligence, 128(1), 99–141.

Vaca-Castano, G., Zamir, A. R., & Shah, M. (2012). City scale geo-spatial trajectory estimation of a moving camera. In IEEE Conference on Computer Vision and Pattern Recognition.

Wu, C., Agarwal, S., Curless, B., & Seitz, S. M. (2011). Multicore bundle adjustment. In IEEE Conference on Computer Vision and Pattern Recognition (pp. 3057–3064). IEEE.

Yeh, T., Tollmar, K., & Darrell, T. (2004). Searching the web with mobile images for location recognition. In IEEE Conference on Computer Vision and Pattern Recognition (pp. 76–81).

Zamir, A., & Shah, M. (2010). Accurate image localization based on google maps street view. In European Conference on Computer Vision. Springer.