MAV Urban Localization from Google Street View Data
András L. Majdik, Yves Albers-Schoenberg, Davide Scaramuzza

Abstract—We tackle the problem of globally localizing a camera-equipped micro aerial vehicle flying within urban environments for which a Google Street View image database exists. To avoid the caveats of current image-search algorithms in case of severe viewpoint changes between the query and the database images, we propose to generate virtual views of the scene, which exploit the air-ground geometry of the system.

To limit the computational complexity of the algorithm, we rely on a histogram-voting scheme to select the best putative image correspondences. The proposed approach is tested on a 2km image dataset captured with a small quadrotor flying in the streets of Zurich. The success of our approach shows that our new air-ground matching algorithm can robustly handle extreme changes in viewpoint, illumination, perceptual aliasing, and over-season variations, thus, outperforming conventional visual place-recognition approaches.

MULTIMEDIA MATERIAL

Please note that this paper is accompanied by a video demonstration available on our webpage along with the dataset used in this work: rpg.ifi.uzh.ch

I. INTRODUCTION

In this paper, we deal with the problem of globally localizing a Micro Aerial Vehicle (MAV) in urban environments using exclusively images captured by means of a single onboard camera and at low altitudes (i.e., 10-20 meters from the ground). The global position of the MAV is recovered by recognizing visually-similar discrete places in the map. Namely, the air-level image captured by the MAV is searched in a database of ground-based geotagged pictures, notably Google Street View image data. Because of the large difference in viewpoint between the air-level and ground-level images, we call this problem air-ground matching. A graphical illustration of our scenario is shown in Fig. 1.

The motivation behind this work is to develop autonomous flying vehicles that could one day operate in urban environments where GPS signal is shadowed or completely unavailable. In these situations, such technology is crucial to correct the drift induced by ego-motion-estimation devices (e.g., inertial measurement units, or inertial-visual odometry [1], [2]).

The authors are with the Artificial Intelligence Lab—Robotics and Perception Group—http://rpg.ifi.uzh.ch, University of Zurich, Switzerland, majdk@ifi.uzh.ch, yvesal@ethz.ch, davide.scaramuzza@ieee.org.

This research was supported by the Scientific Exchange Programme SCIX-NMS-CH project 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 Robotics.

†http://google.com/streetview

In recent years, numerous research papers have addressed the development of autonomous Unmanned Ground Vehicles (UGV), leading thus to striking new technologies like self-driving cars. These can map and react in highly-uncertain street environments partially using [3]—or completely neglecting—GPS systems [4]. In the next years, a similar bust in the development of small-sized Unmanned Aerial Vehicles (UAV) is expected. Flying robots could perform a large variety of tasks in everyday life, e.g., medication or other goods delivery, inspection and modeling of industrial and historical buildings, search and rescue missions, monitoring, etc.

Visual-search techniques used in state-of-the-art place-recognition systems may perform poorly with air-ground image matching, since in this case—besides the challenges present in ground visual search algorithms used in UGV applications, such as illumination, lens distortions, over-season variation of the vegetation, and scene changes between the query and the database images—extreme changes in viewpoint and scale can be found between the aerial MAV images and the ground-level images.

To illustrate the challenges of the air-ground image matching scenario, in Fig. 2 we show a few samples of the airborne images and their associate Google-Street-View images from the dataset used in this work. As can be observed, due to the different fields of view of the ground cameras 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 mainly consists of highly-repetitive and self-similar structures (e.g., windows) (c.f. Fig. 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 Google-Street-View and MA V images, and, thus, successfully solve their matching. An efficient virtual-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 ground level ones. One might argue that this leads to a significant computational complexity. We overcome this issue by selecting only a finite number of the most similar Google-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 using only 4 cores (candidate images) very good results are obtained with the proposed algorithm. 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 time.

To summarize, this paper advances the state-of-the-art with the following contributions:

- It solves the problem of air-ground matching between MAV-based and ground-based images in urban environments. Specifically, we propose to generate artificial views of the image in order to overcome the large viewpoint differences between ground and aerial images, and, thus, successfully solve their matching.
- We present a new algorithm to rapidly detect putative corresponding image matches using a computationally-inexpensive and extremely-fast histogram-voting scheme. Furthermore, the algorithm automatically scales to the limitations of the available and computational power, e.g., number of existing cores of the processor units.
- The proposed approach is a novel-image search technique that can robustly pair images with severe differences in viewpoint, scale, illumination, perceptual aliasing, repetitive structures, and changes in the scene between the query and the database images.
- We provide the first ground-truth labeled dataset that contains both aerial images—recorded by a drone and other measured parameters simultaneously—and geotagged ground-level images of urban streets. We hope that this dataset can serve as a benchmark and motivation for further research in other robotics labs in this field.

The remainder of the paper is organized as follows: Section II gives a review of the related work; Section III shows the limitations of the state-of-the-art; Section IV presents the proposed air-ground matching algorithm in detail; Section V presents the results in comparison with other approaches from the literature; finally we conclude in Section VI.
II. 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 [5]–[8], while other works focused on position tracking using the Bayesian fashion—such as in [9], where the authors presented a method that also uses Google-Street-View data to track the geospatial position of a camera-equipped car in a city-like environment. Other algorithms used image-search-based localization for hand-held mobile devices to detect Point Of Interest (POI), such as landmark buildings or museums [10]–[12]. Finally, in the recent years, several works have focused on image localization with Google-Street-View data [13], [14]. However, all the works mentioned above aim to localize street-level images in a database of pictures also captured at the 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 III and Fig. 4, traditional algorithms tend to fail in air-ground settings, where the goal is to match airborne imagery with ground one.

Most works addressing the air-ground-matching problem have relied on different assumptions than ours, notably the altitude at which the aerial images are taken. For instance, in [15], [16] the problem of geo-localizing ground level images in urban environments with respect to satellite or high-altitude (several hundred meters) aerial imagery was studied. In contrast, in this paper we aim specifically at low-altitude imagery, which means, images captured by safe MAVs flying at 10-20m from the ground.

As envisaged by the firm Matternet, MAVs will soon be used to transport goods, such as medications, blood samples, or even pizzas from building to building in large urban settings. Therefore, improving localization at small altitude where GPS signal is shadowed or completely unreliable is of utmost importance. To the best of our knowledge, we are the first to present an in-depth analysis of air-ground matching between ground-level images (recorded by a car) and low-altitude aerial images (recorded by a MAV flying close to the buildings’ facades at 10-20 meters from the ground).

III. 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.

A. Brute-force search algorithms

Brute-force approaches work by comparing each aerial image to every Google-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 Google-Street-View images. A comparison between two images is done through the following pipeline: (i) SIFT [17] 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., RANSAC 8-point algorithm [18]). 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:

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

where $\gamma$ specifies the expected outlier ratio. Using the 8-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, in 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%–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 Fig. 2, we conclude that RANSAC-like methods fail to robustly reject wrong correspondences. The confusion matrix depicted in Fig. 4b reports the results of the 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 4-point RANSAC—which leverages the planarity constraint between features sets belonging to building facades.

The second algorithm applied to our air-ground-matching scenario is the one presented in [19], here referred to as Affine SIFT and ORSA. In [19], an image-warping algorithm is described to compute artificially-generated views of a planar scene able to cope with large viewpoint changes. ORSA [20] 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 Fig. 4c), although the recall rate at precision 1 was below 15% (c.f. Fig. 8).

B. Bag-of-words search algorithms

The second category of algorithms used for comparison are the bag-of-words (BoW) based methods [21], 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

\[^2\]http://matternet.us
Fig. 4: These plots show the confusion matrices obtained by applying several algorithms described in the literature (b-c, e-f) and the one proposed in the current paper (d). (a) Ground-truth: the data was manually labeled to establish the exact visual overlap between the aerial MAV images and the ground Google-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 and in the challenging task of matching ground and aerial images. For precision and recall curves, compare to Fig. 8.

Finally, the fourth algorithm used for our comparison is FABMAP [5]. To cope with perceptual aliasing, in [5] an algorithm is presented where the co-appearance 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 failed in our air-ground-matching scenario, as displayed in Fig. 4f.

As observed, both BoW and FABMAP 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, leading, thus, to a false visual-word association. Consequently, the air-level images are erroneously matched to the Google-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 in the next section. The confusion matrix of the proposed algorithm applied to our air-ground matching scenario is shown in Fig. 4d. This can be compared with the confusion matrix of the ground truth data (Fig. 4a). As observed, the proposed algorithm outperforms all previous approaches.

IV. AIR-GROUND MATCHING OF IMAGES

In this section, we describe the proposed algorithm in details. A pseudo-code description is given in Algorithm 1. Please note that the algorithm from line 1 to 7 can and should be computed off-line, previous to an actual flight mission. In this phase, previously saved Google-Street-View images \( I_1, I_2, \ldots, I_n \) are converted into image-feature–based representations \( F_i \), after applying the virtual-view generation method described in the next section, and are saved in a database \( D_F \).

A. Virtual-view generation

Point feature detectors and descriptors—such as SIFT [17], BRISK [22], etc.—usually ensure invariance to rotation and scale. However, they tend to fail in case of substantial viewpoint changes (\( \theta > 45^\circ \)), as we have shown in the previous section (c.f. Fig. 4b-c).

Our approach was inspired by a technique initially presented in [19], where, for a complete affine invariance (6 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 (\( \phi \)) and the latitude angles (\( \theta \)) are defined in Fig. 5 on the right. The tilt can thus be defined as...
<table>
<thead>
<tr>
<th>Tilt</th>
<th>( \sqrt{2} )</th>
<th>2</th>
<th>( 2\sqrt{2} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \theta )</td>
<td>45°</td>
<td>60°</td>
<td>69.3°</td>
</tr>
</tbody>
</table>

**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 street-like environment

**Output:** The location of the drone in the discrete map, respectively the best match \( I_b \)

1. \( D_T \) = database of all the image features of \( \mathcal{I} \); 
2. for \( i \leftarrow 1 \) to \( n \) do
3. \( V_i \) = generate virtual views (\( I_i \)); // details in IV-A;
4. \( F_i \) = extract image features (\( V_i \));
5. \( \text{add} F_i \) to \( D_T \);
6. train \( D_T \) using FLANN [23];
7. \( c \leftarrow \) number of cores;
8. // up to this line the algorithm is computed off-line;
9. \( V_a \) = generate virtual views (\( I_a \));
10. \( F_a \) = extract image features (\( V_a \));
11. search approximate nearest neighbor feature matches for \( F_a \) in \( D_T \): \( M_D = \text{ANN}(F_a, D_T) \);
12. select \( c \) putative image matches \( I^p \subseteq \mathcal{I} \):
13. \( \mathcal{I}^p = \{ I^p_1, I^p_2, \ldots, I^p_c \} \) // details in section IV-B;
14. run in parallel for \( j \leftarrow 1 \) to \( c \) do
15. search approximate nearest neighbor feature matches for \( F_a \) in \( I^p_j \): \( M_j = \text{ANN}(F_a, I^p_j) \);
16. select inlier points: \( N_j = \text{kVLD}(M_j, I_a, I^p_j) \);
17. \( I_b \leftarrow \max(N_1, N_2, \ldots, N_c) \);
18. return \( I_b \);

In conclusion, the algorithm described in this section has two main advantages in comparison with the original ASIFT implementation [19]. Firstly, 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. Secondly, by introducing less error sources into the matching algorithm, our solution contributes also to obtaining an increased performance in the global localization process.

**B. Putative match selection**

In this step, the algorithm selects a fixed number of putative image matches \( \mathcal{I}^p = \{ I^p_1, I^p_2, \ldots, I^p_c \} \), based on the available hardware. The idea is to select a subset of the Google-Street-View images from the total number of all the 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 virtual views \( F_a \). The search is performed by using the FLANN [23] library that implements multiple randomized KD-tree or K-means tree forests and auto-tuning of the parameters. According to the literature, this method performs the search extremely fast and with a good precision, although, for searching in very large databases (100 millions of images), there are more efficient algorithms (c.f. [24]). Since we perform the search in a certain area, we opted for FLANN.

Further on, we apply a similar idea to [25], 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 \( \alpha \) 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 Google-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 \( \alpha \) are clustered in bins of \( 5^\circ \). Accordingly, a two-dimensional histogram \( H \) can be built, in which each bin contains the number of features that count for \( \alpha \) in a certain Google-Street-View image. Finally, we select those \( c \) number of Google-Street-View images that have the maximal values in \( H \).

To evaluate the performance of our algorithm, we run several tests using the same dataset and test parameters, and only modifying the number of cores used. Fig. 6 shows the obtained results in terms of precision and recall for 4, 8, 16, and 48 cores. The plot shows that, even by using


TABLE II: Recall rate at precision 1 in case of the number of putative Google Street View images analyzed in parallel on different cores.

<table>
<thead>
<tr>
<th>Nr. parallel cores</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>48</th>
<th>96</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall at precision 1 (%)</td>
<td>41.9</td>
<td>44.7</td>
<td>45.9</td>
<td>46.4</td>
<td>46.4</td>
</tr>
</tbody>
</table>

Fig. 6: Performance analysis in terms of precision and recall in case of: 4, 8, 16, and 48 threads were 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.

V. EXPERIMENTS AND RESULTS

A. The experimental dataset

We collected a dataset in downtown Zurich, Switzerland. A commercially available Parrot AR.Drone 2 flying vehicle was manually piloted along a 2km trajectory, collecting images throughout the environment at different flying altitudes by keeping the MAV camera always facing the buildings. Sample images are shown in Fig. 2, left column. For more insights, kindly check the video file accompanying this article. The full dataset consists of more than 40,500 images. For all the experiments presented in this work, we sub-sampled the data selecting one image from every 100, resulting in a total number of 405 MAV test images. All the available Google-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 Google-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 Google-Street-View data. The Street View pictures were recorded in summer 2009 while the MAV dataset was collected in winter 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. Fig. 7 shows the positions of the Google-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 Google-Street-View image was found—are shown (green-circle).

B. Evaluation criteria and parameters used for the experiments

The different visual-appearance–based algorithms were evaluated in terms of recall rate and precision rate. We also

3http://rpg.ifi.uzh.ch

4Recall rate = Number of detected matches over the total number of possible correspondences

5Precision rate = Number of true positive detected over the total number of matches detected (both true and false)
Fig. 7: Bird’s-eye view of the test area. The blue dots mark the locations of the ground Google 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.

show the results using a different visualization, namely confusion maps. Fig. 4 depicts the results obtained by applying the four conventional methods discussed in Section III and the algorithm proposed in this work (Fig. 2d). The confusion matrix shows the visual similarity computed between all the Google-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. A dark blue represents no visual similarity, while a dark red color is a complete similarity. An ideal image pairing algorithm would detect a confusion matrix coincident to the ground-truth matrix (Fig. 4a). A stronger deviation from the ground-truth map shows less accurate results.

For the Bag-of-Words\(^6\) approach in Fig. 4e and Fig. 8, a hierarchical vocabulary tree was trained with 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 Google-Street-View images recorded in a neighborhood similar to our test area). Term frequency-inverse document frequency tf-idf was used as weighting type and the L1-Norm as scoring type. In the case of FABMAP\(^7\) algorithm, several parameters were tested to get meaningful results. However, all checked parameter configurations failed on our dataset. For the experiments presented in the paper, the FABMAP 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.

C. Results interpretation

Fig. 8 shows the results in term of precision and recall. Opposite to object recognition algorithms, where the average precision is used to evaluate the results, in robotic applications the most important evaluation criteria is usually the recall rate at precision 1. This criteria 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 on Fig. 8) outperforms the second best approach, namely the ASIFT and ORSA (red) by a factor of 4. This is because, in our approach, the virtual 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. Namely, the results were obtained by selecting just 7% from the total number of Google-Street-View images. We show all the correctly-matched MAV images with Google-Street-View images in the video file accompanying this article, which gives a 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 Fig. 8) and FABMAP (magenta) fail in matching our MAV images with ground level Google-Street-View data. Apparently, these algorithms fail because the visual patterns present in both images are classified in different visual words, thus, leading to false visual-word associations.

Fig. 9 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, where 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 dynamics into account in

\(^6\)We used the implementation of [7] publicly available at: http://webdis.unizar.es/~dorian/

\(^7\)We used the implementation of [5] publicly available at: http://www.robots.ox.ac.uk/~mobile/
a Bayesian fashion. The limitations of the proposed method are shown in Fig. 10. Please note that these robot positions (top row) are difficult to be recognized even for humans. Moreover the over-season change of the vegetation makes it extremely difficult to cope with the pairing of them for image feature based techniques.

VI. CONCLUSIONS

To conclude, this paper solves the air-ground matching problem of low-altitude MAV-based imagery with ground level Google-Street-View images. Our algorithm outperforms conventional methods from the literature in challenging settings, where the aerial vehicle flies over urban streets up to 20 meters, often flying close to buildings. Furthermore the presented algorithm keeps the computational complexity of the system at an affordable level.

REFERENCES