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 quadroctopter 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[1]. 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, `majdik@ifi.uzh.ch`, `yvesal@ethz.ch`, `davide.scaramuzza@ieee.org`.

This research was supported by the Scientific Exchange Programme SCIEX-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.

[1] `http://google.com/streetview`

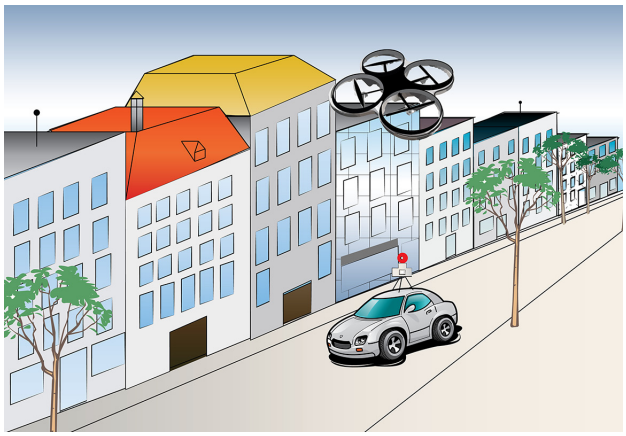

Fig. 1: Illustration of the problem addressed by this work. The global position of the MAV is computed by matching the aerial image taken by the flying vehicle with the closest ground-level geotagged Google Street View image.

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

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

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 MAV 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.

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

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 scene 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 geo-tagged 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,[2] 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:

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

where γ 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 off-line 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

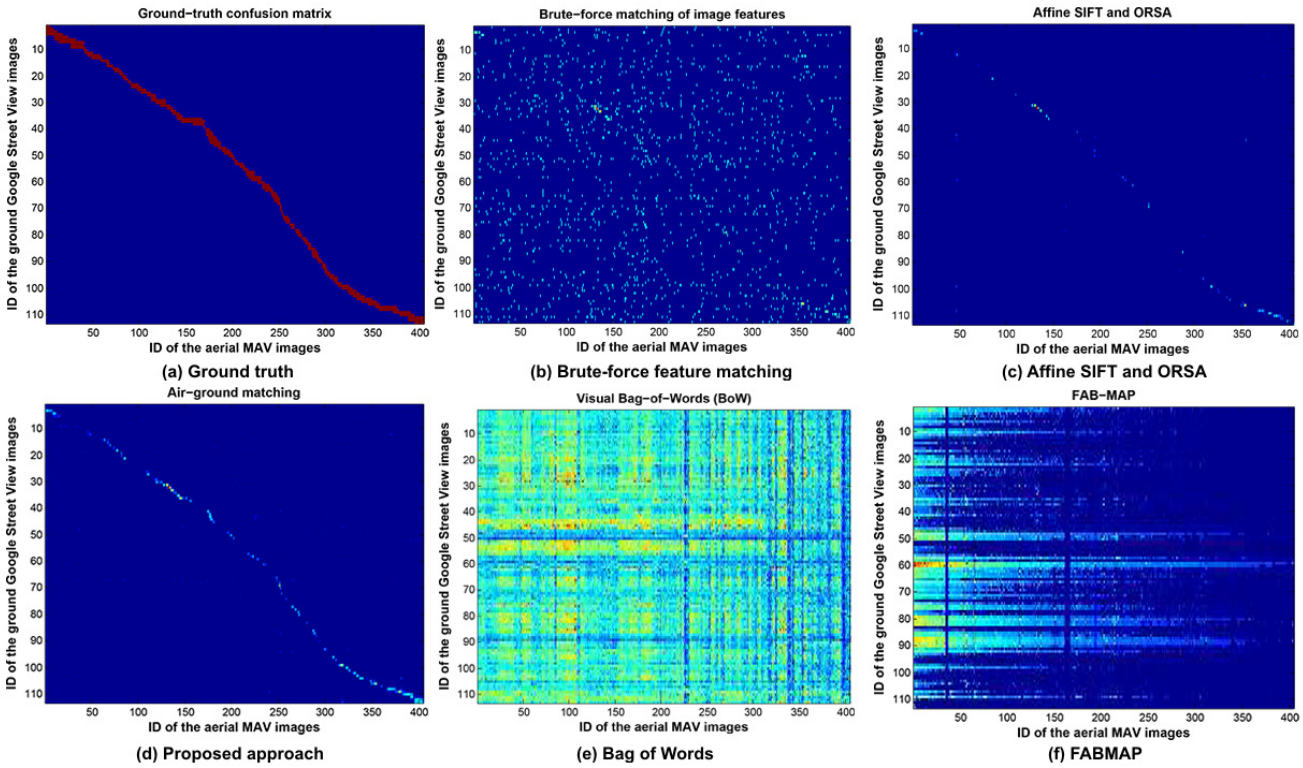

Fig. 4: These plots show the confusion matrices obtained by applying several algorithms described in the literature (b-c, e-f) and the one proposed in the current paper (d). (a) Ground-truth: the data was manually labeled to establish the exact visual overlap between the aerial MAV images and the ground 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

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 [6]. The results of this approach applied to the air-ground dataset are shown in Fig. 4e. We tested different configuration parameters, but the results did not improve (c.f. 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 $\mathcal{I} = \{I_1, I_2, \ldots, I_n\}$ 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_T.

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 ($\theta > 45°$), 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 (ϕ) and the latitude angles (θ) are defined in Fig. 5 on the right. The tilt can thus be defined as

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

TABLE I: Tilting values for which artificial views were made.

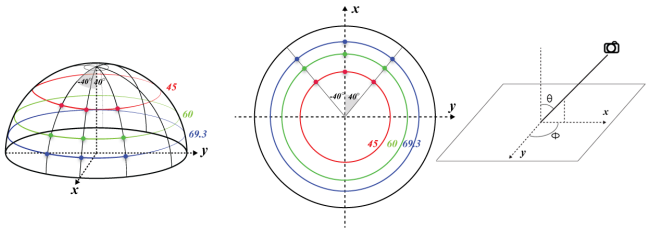

Fig. 5: Illustration of the sampling parameters for virtual view generation. Left: observation hemisphere - perspective view. Right: observation hemisphere - zenith view. The samples are marked with dots.

$tilt = \frac{1}{cos(\theta)}$. The Affine Scale-Invariant Feature Transform (abbrev. ASIFT [19]) detector and descriptor is obtained by sampling various values for the tilt and longitude angle ϕ to compute virtual views of the scene. Further on, SIFT features are detected on the original image and as well on the artificially-generated images. In contrast, in our implementation, we limit the number of tilts considered 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 [19], we make use of just three virtual simulations, i.e., at $0°$, and $\pm 40°$. We illustrate the proposed parameter-sampling method in Fig. 5 and display the different tilt values in Table I. By adopting this efficient sampling method, we managed to reduce the computational complexity by six times—from 60 to 9 artificial views.

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 $\mathcal{I}^p = \{I_1^p, I_2^p, \ldots, I_c^p\}$, 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

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

D_T = database of all the image features of $\mathcal{I}$; ;
for $i \leftarrow 1$ ***to*** n **do**
  V_i = generate virtual views (I_i); // details in IV-A ;
  F_i = extract image features (V_i); ;
  add F_i to D_T;
train D_T using FLANN [23]; ;
$c \leftarrow$ number of cores; ;
// up to this line the algorithm is computed off-line ;
V_a = generate virtual views (I_a); ;
F_a = extract image features (V_a); ;
search approximate nearest neighbor feature matches for F_a in D_T: M_D = 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 IV-B ;
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 = ANN(F_a, F_j^p); ;
  select inlier points: N_j = kVLD(M_j, I_a, I_j^p); ;
$I_b \leftarrow max(N_1, N_2, \ldots, N_c)$;;
return I_b;

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 data bases (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 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 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 α are clustered in bins of $5°$. 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 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

Nr. parallel cores	4	8	16	48	96
Recall at precision 1 (%)	41.9	44.7	45.9	46.4	46.4

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

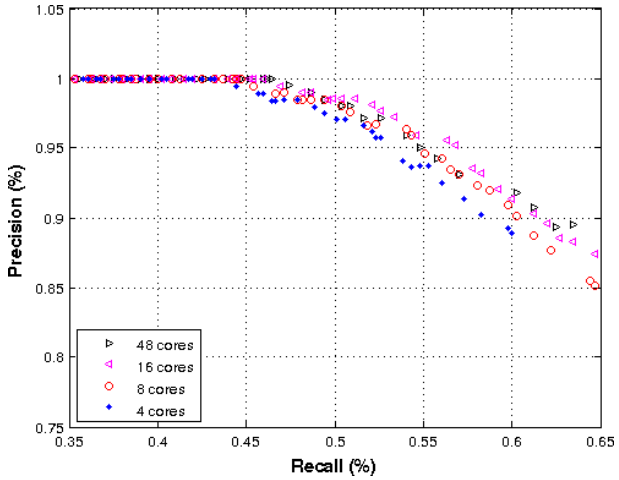

Fig. 6: Performance analysis in terms of precision and recall in case of: 4, 8, 16, and 48 threads were used in parallel. Please note that by selecting just 3% of the total number of possible matches, more then 40% of the true positive matches were detected by the proposed algorithm.

just 4 cores in parallel, a significant number of true-positive matches between the MAV and Google-Street-View images is found without having any erroneous pairing, namely at precision 1. By using 8 cores in parallel, the performance increases by almost 3%. Please note that it is also possible to use two times 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 (c.f. Table II).

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

C. Pairing and acceptance of good matches

Having selected c Google-Street-View images $\mathcal{I}^p = \{I_1^p, I_2^p, \ldots, I_c^p\}$ as described in the previous chapter, 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. Analogous 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 Google-Street-View image feature database D_T, and matched against F_a.

In order to pair the airborne MAV images with the Google-Street-View data and select the best match between 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 a very challenging one for several reasons, and thus, the 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) [26]. Between two key-points of the image, a virtual line is defined and characterized with a SIFT-like descriptor, after the points pass a geometrical consistency check as in [27]. Consistent image matches are searched in the other image by computing and comparing the virtual lines. Further on, the algorithm connects and match a graph consisting of k connected virtual lines. The image feature 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 virtual-view generation and putative-match selection.

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.[3] 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*[4] and *precision rate*.[5] We also

[3] `http://rpg.ifi.uzh.ch`

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

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

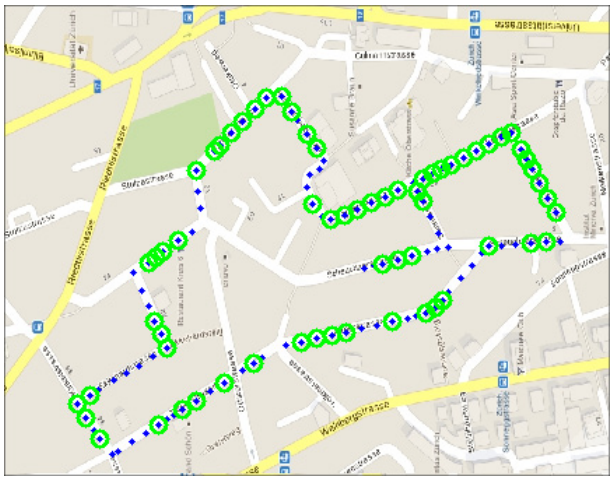

Fig. 7: Bird's-eye view of the test area. The blue dots mark the locations of the ground Google Street View images. The green circles represent those places where the aerial images taken by the urban MAV were successfully matched with the terrestrial image data.

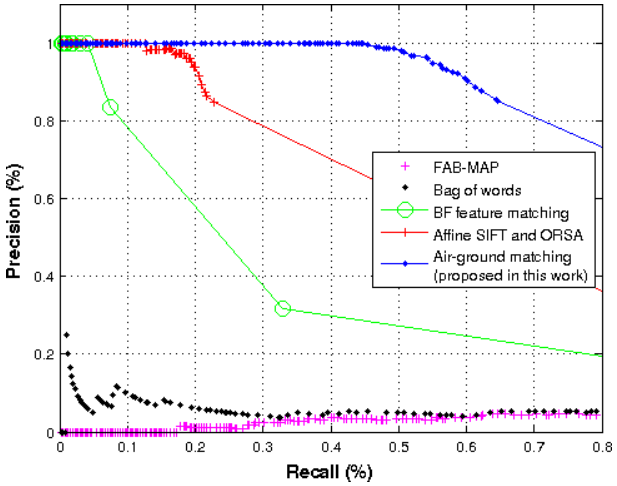

Fig. 8: Comparison of the results. Please note that at precision 1 the proposed Air-ground matching algorithm greatly outperforms in terms of recall the other methods. To visualize all the correctly matched airborne MAV images with the Google Street View images please consult the video attachment of the paper.

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://webdiis.unizar.es/~dorian/`

[7]We used the implementation of [5] publicly available at: `http://www.robots.ox.ac.uk/~mobile/`

Fig. 9: 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 where the MAV is facing the same building from two different sides (streets), having in the filed of view only windows with the same patterns.

Fig. 10: 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 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.

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. In the future, we plan to extend this work by incorporating position tracking and using the global localization algorithm described in the current work to correct the accumulated drifting errors. The time constraints of the proposed algorithm are relaxed, since not all the frames taken by the MAV have to be processed for the global localization of the MAV. Moreover, our architecture is ideal for a cloud-based implementation, where the aerial image of the MAV is sent through the 4G network to server-based search engines.

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

[1] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-SLAM-based navigation for autonomous micro helicopters in GPS-denied environments," *Journal of Field Robotics*, vol. 28, no. 6, pp. 854–874, 2011.

[2] S. Weiss, M. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments," *in ICRA*, 2012.

[3] W. Churchill and P. M. Newman, "Practice makes perfect? managing and leveraging visual experiences for lifelong navigation," in *ICRA*, 2012, pp. 4525–4532.

[4] J. Ibañez Guzmán, C. Laugier, J.-D. Yoder, and S. Thrun, "Autonomous Driving: Context and State-of-the-Art," in *Handbook of Intelligent Vehicles*, 2012, vol. 2, pp. 1271–1310.

[5] M. Cummins and P. M. Newman, "Appearance-only slam at large scale with fab-map 2.0," *I. J. Robotic Res.*, vol. 30, no. 9, 2011.

[6] A. Majdik, D. Gálvez-López, G. Lazea, and J. A. Castellanos, "Adaptive appearance based loop-closing in heterogeneous environments," in *in IROS*, 2011, pp. 1256–1263.

[7] D. Galvez-Lopez and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[8] W. P. Maddern, M. Milford, and G. Wyeth, "Cat-slam: probabilistic localisation and mapping using a continuous appearance-based trajectory," *I. J. Robotic Res.*, vol. 31, no. 4, pp. 429–451, 2012.

[9] G. Vaca-Castano, A. R. Zamir, and M. Shah, "City scale geo-spatial trajectory estimation of a moving camera," in *CVPR*, 2012.

[10] G. Baatz, K. Köser, D. M. Chen, R. Grzeszczuk, and M. Pollefeys, "Leveraging 3d city models for rotation invariant place-of-interest recognition," *I. J. of Computer Vision*, vol. 96, no. 3, 2012.

[11] G. Fritz, C. Seifert, M. Kumar, and L. Paletta, "Building detection from mobile imagery using informative sift descriptors," in *SCIA*, 2005.

[12] T. Yeh, K. Tollmar, and T. Darrell, "Searching the web with mobile images for location recognition," in *in CVPR*, 2004, pp. 76–81.

[13] G. Schindler, M. Brown, and R. Szeliski, "City-scale location recognition," in *CVPR*, 2007.

[14] A. Zamir and M. Shah, "Accurate image localization based on google maps street view," in *in ECCV*, 2010.

[15] M. Bansal, H. S. Sawhney, H. Cheng, and K. Daniilidis, "Geo-localization of street views with aerial image databases," in *ACM Multimedia*, 2011.

[16] M. Bansal, K. Daniilidis, and H. S. Sawhney, "Ultra-wide baseline facade matching for geo-localization," in *ECCV Workshops (1)*, 2012.

[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *I. J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[18] A. Harltey and A. Zisserman, *Multiple view geometry in computer vision (2. ed.)*. Cambridge University Press, 2006.

[19] J.-M. Morel and G. Yu, "Asift: A new framework for fully affine invariant image comparison," *SIAM J. Imaging Sciences*, vol. 2, no. 2, pp. 438–469, 2009.

[20] L. Moisan, P. Moulon, and P. Monasse, "Automatic Homographic Registration of a Pair of Images, with A Contrario Elimination of Outliers," *Image Processing On Line*, 2012.

[21] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *in ICCV*, 2003, pp. 1470–1477.

[22] S. Leutenegger, M. Chli, and R. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *in ICCV*, 2011, pp. 2548–2555.

[23] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *I. Conf. on Computer Vision Theory and Application VISSAPP*, 2009, pp. 331–340.

[24] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE PAMI*, vol. 33, no. 1, pp. 117–128, 2011.

[25] D. Scaramuzza, "1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints," *I. J. of Computer Vision*, vol. 95, no. 1, pp. 74–85, 2011.

[26] Z. Liu and R. Marlet, "Virtual line descriptor and semi-local graph matching method for reliable feature correspondence," in *in British Machine Vision Conference*, 2012, pp. 16.1–16.11.

[27] A. Albarelli, E. Rodolà, and A. Torsello, "Imposing semi-local geometric constraints for accurate correspondences selection in structure from motion: A game-theoretic perspective," *I. J. of Computer Vision*, vol. 97, no. 1, pp. 36–53, 2012.