

Deep UAV Localization with Reference View Rendering

Timo Hinzmann* Roland Siegwart*

Abstract

This paper presents a framework for the localization of Unmanned Aerial Vehicles (UAVs) in unstructured environments with the help of deep learning. A real-time rendering engine is introduced that generates optical and depth images given a six Degrees-of-Freedom (DoF) camera pose, camera model, geo-referenced orthoimage, and elevation map. The rendering engine is embedded into a learning-based six-DoF Inverse Compositional Lucas-Kanade (ICLK) algorithm that is able to robustly align the rendered and real-world image taken by the UAV. To learn the alignment under environmental changes, the architecture is trained using maps spanning multiple years at high resolution. The evaluation shows that the deep 6DoF-ICLK algorithm outperforms its non-trainable counterparts by a large margin. To further support the research in this field, the real-time rendering engine and accompanying datasets are released along with this publication.

1 Introduction

Global Positioning System (GPS) measurements are commonly fused in aerial navigation systems to guarantee long-term stable navigation. However, numerous reasons exist for not fully relying on GPS measurements, for instance, flights in naturally GPS-denied areas (e.g., mountainside), intentionally jammed GPS signals from external sources (e.g., by military), unintentionally jammed GPS signals by on-board sources (e.g., USB 3.0 [28]), or complete GPS receiver failures. On the other hand, the increasing availability of highly-accurate, high-resolution geo-referenced terrain mosaics, and elevation data allows its tight coupling into a visual-inertial navigation system for Unmanned Aerial Vehicles (UAVs). The map data is standardized, stored efficiently in the form of images, and can either be acquired from satellite data or generated from geo-referenced images using commercial (e.g., Pix4Dmapper [1]) or open-source software [22]. Another advantage of localizing within the map, instead of relying purely on GPS, is that it ensures that the UAV pose and the map are consistent, which makes aerial navigation close to the static terrain safer. However, this raises

*All authors are with the Autonomous Systems Lab, ETH Zurich.

the challenge of robust image registration under potential environmental and seasonal changes that we want to address in this paper.

2 Related work

The related work section is subdivided into the subproblems (a) image retrieval and (b) local image alignment. (a) and (b) are further divided into general localization and UAV specific localization methods. The remainder of the paper concentrates on the local image alignment, also referred to as map tracking algorithm. The map tracking algorithm requires a coarse pose initialization, which can be obtained from one of the image retrieval algorithms from Sec. 2.1.

2.1 Image Retrieval and Global Localization

Definition: Given a set of (geo-referenced) database images, retrieve the N best-matching database images for a query image.

Traditional image retrieval methods can be categorized in Bag-of-Words (BOW) and image voting approaches: FAB-MAP [9] is one of the most famous representants of the BOW approach. Proposed by Galvez-Lopez et al., dBoW2 [14] is built up from FAST [39] and BRIEF [6] features with focus on quick real-time query. It is used by the Visual-Inertial (VI) state estimator VINS-mono [38] for re-localization.

Many vertex or image voting approaches with different voting strategies exist: Probabilistic Vertex Voting (PVV) is a scoring scheme proposed by Gehrig et al. [17], that uses BRISK [26] descriptor projection, approximate k-Nearest Neighbors (kNN) search, and scoring based on a binomial distribution. The strategy significantly outperformed the baseline Active Search (AS) [41] in an aerial dataset. The 6-Degree of Freedom (DoF) pose is computed with the Random Sample Consensus (RANSAC) and Perspective-n-Point (PnP) algorithm based on the identified matches. Schönberger et al. proposed Vote-and-verify (VAV) [44], a voting scheme that, in contrast to pure bag-of-words approaches, features spatial verification but aims at being more efficient than standard RANSAC hypothesize-and-verify approaches while maintaining their accuracy. The image retrieval approach computes a vocabulary and visual words from training images. Database images are then used for indexing. For every query, the 6-DoF pose is computed with RANSAC-PnP. This image retrieval approach is part of the incremental structure-from-motion pipeline colmap [42]. Jegou et al. proposed the Vector of Locally Aggregated Descriptors (VLAD) [23] descriptor that uses local image features and produces a compact representation for global indexing. This approach constructs a visual dictionary from the descriptors, computes VLAD descriptors from the visual dictionary, and then creates an index for lookup. As a first Deep Neural Network (DNN)-based global image retrieval method, Arandjelovic et al. proposed NetVLAD [3], a trainable Convolutional Neural Network (CNN) architecture with VLAD layer as main component. Proposed by Sarlin et al., HF-Net [40] is a hierarchical network

based on image retrieval with global descriptors and subsequent 6-DoF pose estimation with local features. The approach compresses the Superpoint [10] and NetVLAD layers using a knowledge distillation scheme. Shetty et al. [46] use a CNN-based scene localization network for image retrieval and a camera localization network for pose regression. However, this approach is not returning the full 6-DoF pose and results in relatively high position and orientation errors.

Several frameworks exist that explicitly address the challenge of environmental changes between query and database image, with the majority of works designed for automotive or, in general, ground-based applications: Bürki et al. [5] propose an appearance-based map management system that summarizes multiple missions and improves the query performance while limiting the accumulation of data. This method is well suited if a location is revisited multiple times with the same sensor setup. Other approaches make use of semantics to improve invariance to appearance changes: Lindsten et al. [29] estimate the geo-referenced 2D position of a UAV by applying environmental classification to both the terrain mosaic and UAV image stream and using subsequent rotation-invariant template matching for alignment. Yu et al. propose VLASE [53], the aggregation of semantic edges into a feature for subsequent VLAD description for ground vehicle localization. For ground-aerial localization, X-View [16] creates a multi-view graph of semantic observations but ignores the size and shape of the segments. Garg et al. introduce LoST [15], a local semantic tensor combining optical and semantic cues. Schönberger et al. [43] make use of a 3D semantic descriptor, assuming access to a depth image for every query image.

The majority of methods discussed above treat the queries independently and not as a video stream, with the following exceptions: Medina et al. [33] evaluate different strategies to aggregate image retrieval votes from a video stream to infer the camera’s location but do not consider motion between the frames. X-View [16] performs graph matching between the query and database graph using Maximum Likelihood Estimation (MLE), neglecting the existence of potentially multiple hypotheses. In contrast, Doherty et al. [11] detect objects of the same semantic class (e.g., cars) and models the multiple data association possibilities with graph-based multi-modal inference [13].

2.2 Local Image Alignment and Map Tracking

Definition: Given a query and a target image, compute the transformation that best aligns both images.

Hand-crafted features and descriptors have been used to match images with small appearance variations: Koch et al. [25] use SIFT [30] features with outlier handling to match nearby aerial images. A combination of correlation and SIFT-like feature matching is employed in [34] to match images recorded of the same terrain but on different days. Surber et al. [50] use weak GPS priors for initialization and direct BRISK matching for visual localization of a UAV in a previously flown mission. Mutual Information (MI) methods have shown to be more robust to appearance changes [52, 36], but come with higher computational

costs due to the Hessian-based iterative optimization.

Higher-level features are employed, for instance, by Patterson et al. in [37]. Ordnance Survey (SO) layers with roads and building outlines are used as a reference map. During the flight, the UAV image stream is converted to the same format, and correlation-based matching is applied to locally align it to the reference map and to estimate the 2D position of the UAV. Shan et al. [45] compute the correlation between UAV image and reference map for global localization. After initialization, a particle filter, Histogram of Oriented Gradient (HOG) features, monocular camera, Inertial Measurement Unit (IMU), and barometer are used for local alignment. Han et al. [21] extract lines and apply Incremental Closest Point (ICP) for reference alignment. Similarly, Son et al. [48] extract building information from satellite images and match the lines to the ones extracted from the live UAV image stream.

Other methods rely on matching of Digital Elevation Maps (DEMs): Sim et al. [47] estimate the geo-registered UAV position by relative motion estimation from image sequences and absolute position estimation from matching the online computed elevation map to the existing elevation map. Grelsson et al. [20] compute the global pose by aligning a geo-referenced 3D model with a local height patch computed from motion stereo. These approaches assume access to a reliable depth map at run-time and the terrain variation to be discriminative (e.g., urban).

The following approaches have been presented that use rendering for reference view generation: Conte and Doherty use correlation-based template matching fused using a Kalman and Point Mass Filter [8]. Altimeter readings are required for scaling. Chiu et al. [7] claim map views are rendered from a 3D model but keep the vision module as a black box and focus on the graph-based optimization.

Substantial improvements have been achieved with the introduction of Deep Learning (DL) and CNNs: Aznar et al. [4] use a simple CNN for visual alignment of UAV and satellite image but requires a compass and altimeter for operation. More recent feature and descriptor approaches, such as Superpoint [10], D2-Net [12], and tracking-based methods like deep Inverse Compositional Lucas-Kanade (ICLK) variants [18, 19, 31] are able to align two images even under extreme viewpoint and appearance changes. However, the approach of Goforth et al. [18, 19] does not simultaneously compute the alignment and six-DoF georeferenced pose, and [31] assumes the availability of depth images for both the target and the reference image, as it is, for instance, the case when using depth cameras.

In this context, this paper introduces a rendering engine that takes georeferenced orthoimages and elevation maps as input and renders optical and depth images given the six-DoF camera pose and desired camera model. The rendering engine is combined with a learning-based six-DoF-ICLK algorithm that is able to align images under severe appearance variations and only requires one depth image per image pair.

The remainder of the paper is structured as follows: Sec. 3 introduces the rendering engine that is used for offline training data generation and online refer-

ence view generation. Sec. 4 embeds the rendering engine in the learning-based ICLK framework for online six-DoF localization. The paper concludes with our experiments, a conclusion, and suggestions for future research directions.

3 Geo-referenced Renderer

This section introduces `aerial_renderer`, a rendering program that uses geo-referenced orthoimages (e.g., optical mosaics of different years) and elevation maps to render a metric depth map and texture images given a query camera pose \mathbf{T}_C^W , camera intrinsics, and distortion parameters. The renderer serves two purposes in this paper:

1. **Offline training data generation and augmentation:** Parameter inputs are the six-DoF query pose in geo-referenced¹ coordinates, camera intrinsics, and camera distortion parameters. The data outputs are the rendered texture images (e.g., from all available optical orthoimages) with the corresponding depth map.
2. **Online reference image generation for map tracking (Sec. 4):** Given a pose estimate \mathbf{T}_C^W , a reference image is generated, which is deemed to be close in scale and rotation to the real world image captured by the UAV. This has the advantage that the image alignment algorithm *merely* needs to be invariant to seasonal or environmental changes. Additionally, for the reference image, a dense depth map is rendered essential for six-DoF alignment. In the online mode, the most recent texture and elevation layers are used for rendering.

4 Map Tracking

Fig. 1 visualizes our proposed six-DoF `aerial_map_tracker`, combining the geo-referenced rendering engine `aerial_renderer` with a learning-based 6DoF-ICLK algorithm. The map tracker is agnostic to the UAV platform, i.e., it may be used by rotary-wing (RW) or fixed-wing (FW) UAVs. In the first step, the current camera pose prior of the UAV, denoted by $\mathbf{T}_{C_{\text{ref}}}^W$, is passed to the rendering engine. During initialization, this pose may come from a GPS-equipped autopilot, or from one of the coarse global localization algorithms described in Sec. 2.1. Once initialized, this prior is obtained from the ICLK’s last iteration. For the query pose $\mathbf{T}_{C_{\text{ref}}}^W$, the rendering engine returns the optical image \mathbf{I}_{ref} and depth map \mathbf{d}_{ref} , given the pre-calibrated camera intrinsics and distortion parameters as well as geo-referenced orthoimage and elevation map. Both, the real optical image taken by the UAV \mathbf{I}_1 and the rendered optical image \mathbf{I}_{ref} are color normalized and fed as single views to the convolutional feature encoder proposed in [31]. The encoder returns four pyramidal layers, ranging from level 0 (752×480 pixels) to level 3 (94×60 pixels). The ICLK

¹e.g., Universal Transverse Mercator (UTM)

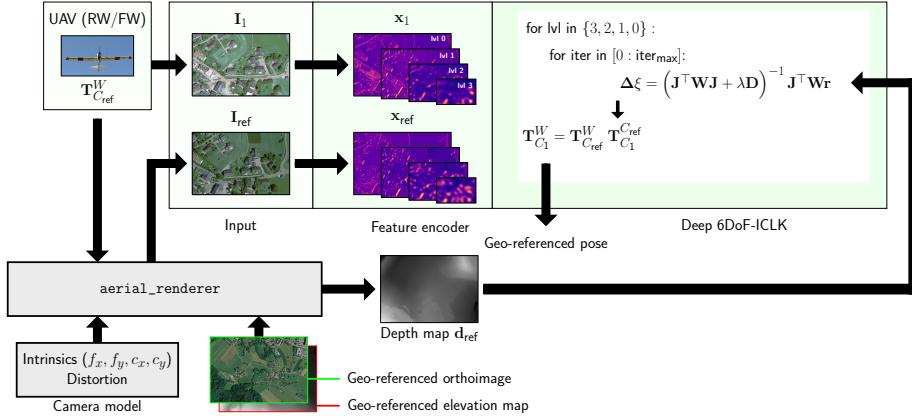


Figure 1: Map tracking framework with Deep 6DoF-ICLK.

algorithm continues to iterate over all levels, starting with the lowest resolution (level 3) up to the original resolution image. For every iteration of the ICLK the six-DoF relative pose is incrementally updated using Levenberg-Marquardt [32, 31]:

$$\Delta\xi = (\mathbf{J}^\top \mathbf{W} \mathbf{J} + \lambda \mathbf{D})^{-1} \mathbf{J}^\top \mathbf{W} \mathbf{r} \quad (1)$$

where the Jacobian \mathbf{J} and residual \mathbf{r} are functions of the depth map \mathbf{d}_{ref} . The diagonal matrix $\lambda \mathbf{D} = \lambda \text{diag } \mathbf{J}^\top \mathbf{W} \mathbf{J}$ is used for regularization of the Hessian matrix with $\lambda = 1e^{-6}$. The weight matrix \mathbf{W} is learned by the convolutional M-Estimator introduced in [31]. Once the ICLK reaches a stopping criteria (e.g., max. number of iterations) the relative transformation matrix $\mathbf{T}_{C_1}^{C_{\text{ref}}} \in \text{SE}(3)$ is computed from $\Delta\xi \in \text{se}(3)$ and used to output the desired map-aligned pose $\mathbf{T}_{C_1}^W$:

$$\mathbf{T}_{C_1}^W = \mathbf{T}_{C_{\text{ref}}}^W \mathbf{T}_{C_1}^{C_{\text{ref}}}. \quad (2)$$

5 Experiments

Implementation The renderer `aerial_renderer` is implemented in OpenGL (C++) [51]. Every pixel of the orthoimage is converted into two triangles, i.e., six vertices, inside the vertex shader program. The `aerial_map_tracker` is implemented in pytorch [35] and adopted from [31]. The `aerial_renderer` is embedded in a ROS [49] wrapper to interface with `aerial_map_tracker`.

Datasets Fig. 2 visualizes the four locations in Switzerland from where training, validation, and testing data is extracted. For each of these four locations, two nearby but non-overlapping areas are selected for training/validation and testing, respectively. The training and validation set are split 80 % to 20 %.

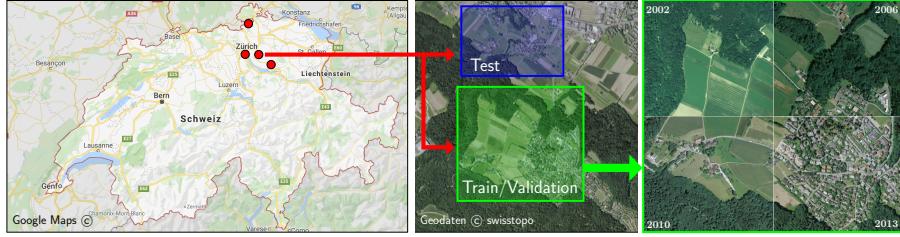


Figure 2: From left to right: Four locations in Switzerland are selected to extract training, validation and testing data. Training, validation and testing data is extracted for every location from nearby but non-overlapping areas. Each of these areas consists again of orthoimages and elevation maps of all available years.

Each of these areas consists again of orthoimages and elevation maps of all available years. All year-to-year combinations are considered for training (e.g., 2002 to 2006, 2010, 2013).

Results Tab. 1 and 2 present the quantitative results in form of the 3D End-Point-Error (EPE) [27, 31], angular, and translational error. The EPE is used as training loss function for all experiments. Tab. 1 evaluates the alignment for the case that \mathbf{I}_1 and \mathbf{I}_{ref} are rendered from the same year, i.e., same orthoimage. In contrast, Tab. 2 shows the error metrics for two different years (2006, 2010).

	EPE					Ang. Error					Transl. Error				
	mean	stdev	median	min	max	mean	stdev	median	min	max	mean	stdev	median	min	max
init	13.88	6.40	12.78	5.53	35.07	0.06	0.03	0.07	0.02	0.12	13.11	5.43	13.56	3.97	25.58
no NN (20)	7.83	8.28	7.18	0.08	35.08	0.05	0.04	0.05	0.00	0.11	8.70	6.96	8.62	0.22	26.99
no NN (50)	3.73	8.19	0.05	0.01	34.67	0.03	0.04	0.00	0.00	0.12	4.46	7.21	0.18	0.01	29.21
NN (20)	3.19	8.55	0.58	0.10	37.82	0.02	0.04	0.01	0.00	0.18	4.37	8.64	1.63	0.24	38.74

Table 1: The error metrics End-Point-Error (EPE), angular and translational error evaluated on the test set for **the same year** (2006, 2006).

First of all, one can see that the deep 6DoF-ICLK algorithm performs on average the best for both cases and in all error metrics, outperforming the variants without trainable parameters independent of the number of iterations. Secondly, for the same year, the non-trainable 6DoF-ICLK is able to reduce the EPE by a factor of 2.1 if the number of iterations is increased by a factor of 2.5, i.e., implying convergence. However, for a different year, the EPE increases by a factor of 1.22 when increasing the number of iterations from 20 to 50, which is a larger error than at initialization, i.e., implying divergence. The deep 6DoF-ICLK algorithm on the other hand, is able to reduce the EPE well below the initial error in both experiments.

The qualitative results are visualized in Fig. 3, presenting four samples taken from the test set (cf. Fig. 2). The first and second row show the framework input

	EPE					Ang. Error					Transl. Error				
	mean	stdev	median	min	max	mean	stdev	median	min	max	mean	stdev	median	min	max
init	13.88	6.40	12.78	5.53	35.07	0.06	0.03	0.07	0.02	0.12	13.11	5.43	13.56	3.97	25.58
no NN (20)	14.14	7.87	13.90	1.41	39.48	0.08	0.04	0.08	0.01	0.16	13.85	6.96	12.25	3.20	24.69
no NN (50)	17.29	10.57	18.18	0.65	44.14	0.11	0.07	0.09	0.00	0.22	19.22	10.80	18.10	1.20	47.48
NN (20)	7.65	8.61	5.75	1.20	40.87	0.07	0.05	0.06	0.01	0.21	10.85	7.18	8.96	2.88	28.70

Table 2: The error metrics End-Point-Error (EPE), angular and translational error evaluated on the test set for **different years** (2006, 2010).

image_0 (i.e., \mathbf{I}_{ref} , from year 2006) and image_1 (i.e., \mathbf{I}_1 , from year 2010) respectively. In the first and, in particular, the last column, one can see significant changes in the infrastructure. The third row visualizes the overlay of image_0 and image_1, given the initialized transformation, which is set to identity in all experiments. The fourth and fifth row show the overlay for the non-trainable 6DoF-ICLK after 20 and 50 iterations, respectively, followed by the deep 6DoF-ICLK in the sixth row. The last row shows the overlay given the ground-truth relative transformation. All methods are tagged with EPE and the number of iterations. Overall, the conclusions that can be drawn are consistent with the quantitative analysis described above: In all samples, the EPE for the non-trainable 6DoF-ICLK rises with an increasing number of iterations, and in 3 out of the 4 samples it is higher than the initial error. Instead, the deep 6DoF-ICLK is able to decrease the EPE in all cases drastically.

Runtime The `aerial_renderer` runs at over 2500 fps on a GeForce RTX 2080 Ti (12GB) for map sizes shown in Fig. 2. The runtime of the 6DoF-ICLK algorithm is given in Tab. 3 in ms, tested on the same GeForce RTX 2080 Ti: It takes the classical 6DoF-ICLK algorithm (*no NN*) approximately 200 ms to align one image-pair when stopping after 20 iterations, which is comparable to the average runtime of the deep 6DoF-ICLK algorithm with the same number of allowed iterations. Increasing the number of allowed iterations of the classical 6DoF-ICLK algorithm by a factor of 2.5 increases the runtime by a factor of 2.26.

	Runtime [ms]				
	mean	stdev	median	min	max
no NN (20)	199.30	6.07	199.72	185.97	214.77
no NN (50)	450.32	12.26	446.63	421.90	476.78
NN (20)	206.85	3.67	207.51	197.99	213.31

Table 3: Runtime of the 6DoF-ICLK algorithm in ms.

6 Conclusion

This work presents a framework for UAV localization in unstructured environments with the help of DL. Firstly, this paper introduces `aerial_renderer`, a real-time rendering engine that uses geo-referenced orthoimages and elevation maps to render depth and optical images given a six-

DoF camera pose \mathbf{T}_C^W and camera model. Secondly, `aerial_map_tracker` is presented, which combines the geo-referenced rendering engine `aerial_renderer` with a learning-based 6DoF-ICLK algorithm. The evaluation shows that the deep 6DoF-ICLK outperforms its non-trainable counterparts by a large margin. An interesting alley for future work would be to *hallucinate* the reference image via view synthesis, taking into account important factors like daytime, season, sun position to reduce the environmental or seasonal change and further ease the image alignment process [24, 2]. Furthermore, we motivate the use of `aerial_renderer` to accelerate the employment of DL for UAV localization in unstructured environments. For instance, one could create infrared orthomosaics and create infrared-optical or infrared-semantical training data for image alignment.

Acknowledgments

The authors thank *Geodaten © swisstopo* for access to the satellite imagery.

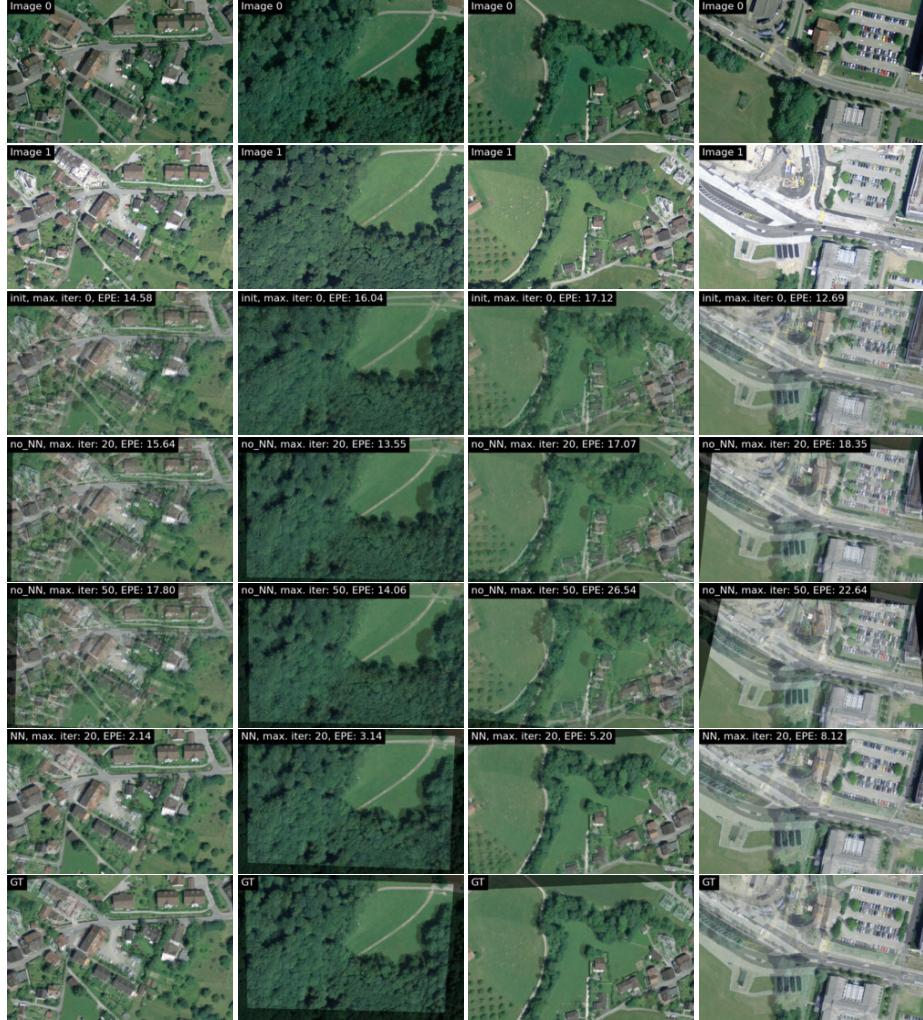


Figure 3: Qualitative evaluation of four samples from the test set shown in Fig. 2. The input images image_0 and image_1 are from the years 2006 and 2010, respectively.

References

- [1] Pix4d mapper. <https://www.pix4d.com/>. Accessed: 2020-08-10.
- [2] I. Anokhin, P. Solovev, D. Korzhenkov, A. Kharlamov, T. Khakhulin, A. Silvestrov, S. Nikolenko, V. Lempitsky, and G. Sterkin. High-resolution daytime translation without domain labels, 2020.
- [3] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.
- [4] F. Aznar, M. Pujol, and R. Rizo. Visual navigation for uav with map references using convnets. In *Conference of the Spanish Association for Artificial Intelligence*, pages 13–22. Springer, 2016.
- [5] M. Bürki, M. Dymczyk, I. Gilitschenski, C. Cadena, R. Siegwart, and J. Nieto. Map management for efficient long-term visual localization in outdoor environments. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 682–688. IEEE, 2018.
- [6] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *Computer Vision (ECCV), 2010 IEEE European Conference on*, pages 778–792.
- [7] H. Chiu, A. Das, P. Miller, S. Samarasekera, and R. Kumar. Precise vision-aided aerial navigation. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 688–695, Sept 2014.
- [8] G. Conte and P. Doherty. A visual navigation system for uas based on geo-referenced imagery. XXXVIII-1/C22, 09 2012.
- [9] M. Cummins and P. Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 2008.
- [10] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236, 2018.
- [11] K. Doherty, D. Fourie, and J. Leonard. Multimodal semantic slam with probabilistic data association. In *2019 international conference on robotics and automation (ICRA)*, pages 2419–2425. IEEE, 2019.
- [12] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. D2-net: A trainable cnn for joint detection and description of local features. *arXiv preprint arXiv:1905.03561*, 2019.

- [13] D. Fourie, J. Leonard, and M. Kaess. A nonparametric belief solution to the bayes tree. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2189–2196. IEEE, 2016.
- [14] D. Gálvez-López and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.
- [15] S. Garg, N. Suenderhauf, and M. Milford. Lost? appearance-invariant place recognition for opposite viewpoints using visual semantics. *Proceedings of Robotics: Science and Systems XIV*, 2018.
- [16] A. Gawel, C. Del Don, R. Siegwart, J. Nieto, and C. Cadena. X-view: Graph-based semantic multi-view localization. *IEEE Robotics and Automation Letters*, 3(3):1687–1694, 2018.
- [17] M. Gehrig, E. Stumm, T. Hinzmann, and R. Siegwart. Visual place recognition with probabilistic voting. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3192–3199. IEEE, 2017.
- [18] H. Goforth and S. Lucey. Aligning across large gaps in time, 2018.
- [19] H. Goforth and S. Lucey. Gps-denied uav localization using pre-existing satellite imagery. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2974–2980, 2019.
- [20] B. Grelsson, M. Felsberg, and F. Isaksson. Efficient 7d aerial pose estimation. In *Robot Vision (WORV), 2013 IEEE Workshop on*, pages 88–95. IEEE, 2013.
- [21] K. Han, C. Aeschliman, J. Park, A. C. Kak, H. Kwon, and D. J. Pack. Uav vision: Feature based accurate ground target localization through propagated initializations and interframe homographies. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 944–950. IEEE, 2012.
- [22] T. Hinzmann, J. L. Schönberger, M. Pollefeys, and R. Siegwart. Mapping on the fly: Real-time 3d dense reconstruction, digital surface map and incremental orthomosaic generation for unmanned aerial vehicles. In *Field and Service Robotics, Results of the 11th International Conference, FSR 2017, Zurich, Switzerland, 12-15 September 2017*, pages 383–396, 2017.
- [23] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3304–3311. IEEE, 2010.
- [24] L. Karacan, Z. Akata, A. Erdem, and E. Erdem. Manipulating attributes of natural scenes via hallucination. *ACM Transactions on Graphics*, 39(1), Nov. 2019.

- [25] T. Koch, X. Zhuo, P. Reinartz, and F. Fraundorfer. A new paradigm for matching uav-and aerial images. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 3(3), 2016.
- [26] S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary Robust Invariant Scalable Keypoints. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.
- [27] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox. Deepim: Deep iterative matching for 6d pose estimation. *International Journal of Computer Vision*, 128(3):657678, Nov 2019.
- [28] H. Lin, C. Lu, H. Tsai, and T. Kung. The analysis of emi noise coupling mechanism for gps reception performance degradation from ssd/usb module. In *2014 International Symposium on Electromagnetic Compatibility, Tokyo*, pages 350–353, May 2014.
- [29] F. Lindsten, J. Callmer, H. Ohlsson, D. Törnqvist, T. B. Schön, and F. Gustafsson. Geo-referencing for uav navigation using environmental classification. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1420–1425. IEEE, 2010.
- [30] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.
- [31] Z. Lv, F. Dellaert, J. Rehg, and A. Geiger. Taking a deeper look at the inverse compositional algorithm. In *CVPR*, 2019.
- [32] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963.
- [33] S. Medina, Z. Dai, and Y. Gao. Where is this? video geolocation based on neural network features. *arXiv preprint arXiv:1810.09068*, 2018.
- [34] C. F. Olson, A. I. Ansar, and C. W. Padgett. Robust registration of aerial image sequences. In *International Symposium on Visual Computing*, pages 325–334. Springer, 2009.
- [35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [36] B. Patel, T. Barfoot, and A. Schoellig. Visual localization with google earth images for robust global pose estimation of uavs. *IEEE Robotics and Automation Letters*, 2020.

- [37] T. Patterson, S. McClean, P. Morrow, and G. Parr. Utilizing geographic information system data for unmanned aerial vehicle position estimation. In *Computer and Robot Vision (CRV), 2011 Canadian Conference on*, pages 8–15. IEEE, 2011.
- [38] T. Qin, P. Li, and S. Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.
- [39] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.
- [40] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019.
- [41] T. Sattler, B. Leibe, and L. Kobbelt. Improving Image-Based Localization by Active Correspondence Search. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 2012.
- [42] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [43] J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler. Semantic visual localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6896–6906, 2018.
- [44] J. L. Schönberger, T. Price, T. Sattler, J.-M. Frahm, and M. Pollefeys. A vote-and-verify strategy for fast spatial verification in image retrieval. In *Asian Conference on Computer Vision*, pages 321–337. Springer, 2016.
- [45] M. Shan, F. Wang, F. Lin, Z. Gao, Y. Z. Tang, and B. M. Chen. Google map aided visual navigation for uavs in gps-denied environment. *CoRR*, abs/1703.10125, 2017.
- [46] A. Shetty and G. X. Gao. Uav pose estimation using cross-view geolocalization with satellite imagery. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1827–1833. IEEE, 2019.
- [47] D.-G. Sim, R.-H. Park, R.-C. Kim, S. U. Lee, and I.-C. Kim. Integrated position estimation using aerial image sequences. *IEEE transactions on pattern analysis and machine intelligence*, 24(1):1–18, 2002.
- [48] K.-H. Son, Y. Hwang, and I.-S. Kweon. Uav global pose estimation by matching forward-looking aerial images with satellite images. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3880–3887. IEEE, 2009.

- [49] Stanford Artificial Intelligence Laboratory et al. Robotic operating system.
- [50] J. Surber, L. Teixeira, and M. Chli. Robust visual-inertial localization with weak gps priors for repetitive uav flights. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 6300–6306. IEEE, 2017.
- [51] M. Woo, J. Neider, T. Davis, and D. Shreiner. *OpenGL programming guide: the official guide to learning OpenGL, version 1.2*. Addison-Wesley Longman Publishing Co., Inc., 1999.
- [52] A. Yol, B. Delabarre, A. Dame, J.-E. Dartois, and E. Marchand. Vision-based absolute localization for unmanned aerial vehicles. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3429–3434. IEEE, 2014.
- [53] X. Yu, S. Chaturvedi, C. Feng, Y. Taguchi, T.-Y. Lee, C. Fernandes, and S. Ramalingam. Vlase: Vehicle localization by aggregating semantic edges. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3196–3203. IEEE, 2018.