Season-Invariant GNSS-Denied Visual Localization for UAVs

Jouko Kinnari ![ORCID], Francesco Verdoja ![ORCID], *Member, IEEE*, and Ville Kyrki ![ORCID], *Senior Member, IEEE*

Abstract—**Localization without Global Navigation Satellite Systems (GNSS) is a critical functionality in autonomous operations of unmanned aerial vehicles (UAVs). Vision-based localization on a known map can be an effective solution, but it is burdened by two main problems: places have different appearance depending on weather and season, and the perspective discrepancy between the UAV camera image and the map make matching hard. In this letter, we propose a localization solution relying on matching of UAV camera images to georeferenced orthophotos with a trained convolutional neural network model that is invariant to significant seasonal appearance difference (winter-summer) between the camera image and map. We compare the convergence speed and localization accuracy of our solution to six reference methods. The results show major improvements with respect to reference methods, especially under high seasonal variation. We finally demonstrate the ability of the method to successfully localize a real UAV, showing that the proposed method is robust to perspective changes.**

Index Terms—**Aircraft navigation, autonomous aerial vehicles, image matching, pose estimation, robot vision systems.**

I. INTRODUCTION

KNOWING the Earth-fixed coordinates of an UAV is one of the basic functionalities required for long-distance autonomous UAV flight. Traditionally, GNSS have been used. However, GNSS are vulnerable to intentional jamming and spoofing attacks by an adversary, and naturally susceptible to blockages and reflections in radio signal paths.

In an ideal localization system, the UAV could infer its location using onboard sensors, without having to depend on availability of any infrastructure. One viable sensor set is a combination of Inertial Measurement Unit (IMU) and camera. Inertial and visual-inertial odometry (VIO) solutions [1] provide tracking for the egomotion of the vehicle in the short term. As these solutions integrate noisy signals, a significant localization error accumulates over a longer period without further correction. Simultaneous Localization and Mapping (SLAM) [2] systems help in reducing this error in case the UAV traverses
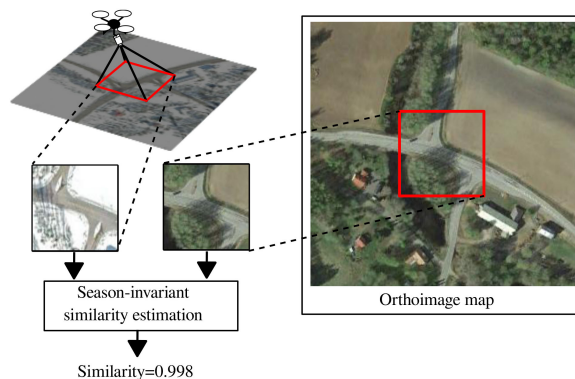
Fig. 1.   We develop a similarity scoring method for UAV localization that is invariant to seasonal appearance change.

the same area a number of times during a mission. However, the correction of accumulated drift provided by SLAM is only partial, and neither SLAM nor VIO systems provide georeferenced coordinates without additional information.

In order to provide georeferenced coordinates, a way of matching sensor observations against a georeferenced map is needed. Providing a match between the observations of the UAV and a map is, however, not a trivial task. Not only do imaging conditions vary due to differences in the imaging hardware, illumination, and weather, but the appearance of the environment changes significantly over seasons.

We propose an image matching approach to season-invariant localization, where the information contained in an image acquired by a UAV is used for verifying or disputing correspondence to an orthoimage map. Using satellite image data, we train a model to learn a similarity measure between orthoimages in a way that is invariant to seasonal change (Fig. 1), and utilize that model for UAV localization in a Monte-Carlo localization (MCL) [3] framework. We demonstrate that, starting from imprecise initialization, the presented method provides significantly shorter time to convergence and smaller localization error than six baseline methods. Moreover, we illustrate the method's operation with real-world data from three UAV flights.

The main contributions of this paper are (i) a solution to visual UAV geolocalization over significant seasonal variation using a Siamese convolutional neural network (CNN), (ii) a method using Gaussian kernel density estimation to evaluate the confidence of the CNN output to be used with MCL, and (iii) a demonstration of the robustness of the solution using real-world and simulation data of flights under significant seasonal change.

II. RELATED WORK

A key functionality in image-based UAV localization is providing a way to find the correspondence between the UAV image and a map. Classical manually engineered feature detectors and descriptors such as SIFT [4] have been proposed by *e.g.*, Cesetti *et al.* [5], but large changes in perspective and seasonal appearance pose challenges in finding correspondences. Feature descriptors specifically hand-crafted for UAV localization have also been proposed by Mantelli *et al.* [6], who modified the BRIEF descriptor [7] to utilize color information. However, we expect color information not to be reliable over significant seasonal appearance change.

Learned features may provide more robust observations for localization. An example of a recent deep learning-based feature detector and descriptor applied to UAV localization is by Hou *et al.* [8], who demonstrate reduction of odometry error of a UAV in short trajectories (750 m) in presence of seasonal appearance change. The approach is based on minimizing reprojection errors of a combination of D2-Net [9] features for map matching and ORB [10] features for visual odometry. The proposed bundle adjustment approach requires accurate knowledge of initial pose and authors show that the solution is not robust to long intervals between keyframes containing map matching features. The need for accurate initialization and good image-to-map feature point matches at short intervals are significant downsides in UAV localization when operating over terrains with long periods of natural ambiguity (*e.g.*, lakes, fields).

Semantic features have been used for finding correspondences between UAV images and a map. In several works, the observed UAV image is first translated into an intermediate terrain class classification (using a single class such as buildings [11] or roads [12], [13] or multiple classes [14]–[16]). Next, features in the semantic representation (such as road [13] and intersection geometry [12], [13], generic shape descriptors [16] or ORB [10] features detected on segmented images [15]) are used as landmarks for localization. Template matching-based methods include computing sum of squared differences (SSD) of semantic classes between map and UAV image [14] and computing the ratio of building to non-building pixels as a matchable descriptor [11].

Instead of matching engineered features, image-to-map matching can be performed in a latent space. Samano *et al.* [17] recently proposed to match UAV images to map by using low-dimensional (16D) embeddings. The projection is learned by finding compatible embeddings for a UAV image and a corresponding semantic map. The embeddings are matched using Euclidean distance. Successful localization is demonstrated for simulated UAV images using MCL. The high reported matching performance and the availability of source code make [17] a good comparison approach for our method. Couturier *et al.* [18] proposed a similar architecture for global descriptor vector extraction.

Instead of detecting visual feature points or relying in any way on a semantic representation, it is possible to use the full image area for finding the correspondence of the UAV image and an orthophoto map. Recent template matching approaches

include the use of Pearson correlation [19] in assessing top-down UAV image similarity to an orthoimage map [20]. In [21], [22], authors match UAV images to precomputed images rendered from preplanned flight paths. To target generality, our focus is on deriving a solution in which the planned path of the autonomous agent is allowed to change during a mission, without requiring significant computation before starting to follow the plan.

Another way to approach the UAV localization problem is to consider it as a sequence of homography transformations between the UAV image and a base map. Yol *et al.* [23] vary homography parameters and maximize mutual information (MI). Goforth *et al.* [24] take a similar approach but add a learning model that transforms the original camera image to a learned feature space to gain a level of seasonal invariance. In both works, starting pose is assumed known. Both solutions track a single hypothesis, which is likely to lead to loss of tracking capability in case of a long period of ambiguity in terrain or due to intermittent matching errors.

To enable tracking of multiple pose hypotheses across ambiguous regions, our work and multiple others (*e.g.*, [6], [15], [17], [20]) use MCL [3] to fuse odometry and map observations.

Outside UAV localization literature, seasonal variations have also been addressed in the context of visual place recognition (VPR) [25], serving as inspiration for our work.

In contrast to the works mentioned above, we present a way to measure the correctness of a pose hypothesis without relying on human-chosen features or semantics. We expect this design choice to allow the matching method to learn a meaningful representation without being constrained to an explicit definition of semantic terrain classes, and without being dependent on existence of those classes in the terrain over which the flight occurs.

III. METHODOLOGY

A. UAV Localization

In the full localization problem, there are 6 degrees of freedom to estimate. To reduce the dimensionality of the problem, we assume that the roll and pitch angles of the UAV can be inferred from the direction of gravity, measurable with an IMU. Altitude is inferred as part of orthoprojection method as presented in earlier work [26]. The state is then defined as

$$X = (x, y, \phi, s) \ , \tag{1}$$

where x, y are longitude and latitude of the UAV position in the map coordinate system, ϕ is the yaw of the UAV and s is a scale parameter, which allows the solution to work in case of scale drift.

To estimate X, we choose MCL, a particle filter tailored for localization over a map $\mathcal{M}$ known in advance. MCL represents the belief on the estimated pose at time t as a set of particles $X_t^r, r = 1 \ldots N$. For initialization, we assume a uniform distribution on an interval for values of x, y and s and uniform distribution over all values of heading for ϕ. When an odometry measurement is available, a new pose for each particle is sampled in accordance with the distribution of the odometry

measurement. A separate observation I_t of the environment acquired in the new pose is then used to update the weight of each particle. The weight w_r of particle r is $w_r = P(X_t^r | \mathcal{I}_t, \mathcal{M})$, *i.e.*, the likelihood that the pose X_t^r represents the true pose, given observation $\mathcal{I}_t$ and map $\mathcal{M}$. The likelihood is estimated based on a similarity measure as described in the following section. This weight is used when resampling a new representative particle set. We follow the particle filter algorithm and low variance sampler described in [3] and resample after each update.

B. Similarity Scoring Function Structure

Given a map $\mathcal{M}$, an observation by the UAV, $\mathcal{I}_t$, and a pose hypothesis X_t^r, we first perform an orthoprojection of the UAV image with the method presented in earlier work [26]. The orthoprojection method in [26] performs VIO and estimates the position of tracked VIO features (landmarks) with respect to drone coordinate frame in meters, assuming sufficient excitation on inertial measurements to resolve scale [27]. By assuming the ground beneath the UAV is planar, the parameters of a plane that best fits the landmark coordinates are resolved and the UAV image is projected to a top-down view by planar homography. This allows creating a projection of the UAV image at a desired resolution (1 m/px in our case) independent of UAV altitude. To tolerate slight drift in estimated scale, our state estimator includes the scale parameter.

From the orthoprojected UAV image, we find the corner points of a square 96×96 m area (1 m/pixel) close to nadir view that is fully visible in the UAV image. Given these corner point locations, we compute what are the corresponding points on the map, if pose hypothesis X_t^r was correct. We then crop a square image patch from both the map $\mathcal{M}$ and the UAV image $\mathcal{I}_t$. We denote these image patches $I_{\mathcal{M}, X_t^r}$ and I_t, respectively. Examples are visualized in Fig. 9.

These image patches are used to compute a similarity score c_t^r, for each pose hypothesis indexed by r at time t, using a similarity function f:

$$c_t^r = f(I_{\mathcal{M}, X_t^r}, I_t) \tag{2}$$

To learn f, we propose an image comparison CNN architecture inspired by [28]. The model structure is shown in Fig. 2. The model contains two Siamese network branches and a separate decision network[1]. The model takes as input the pair of images $(I_t, I_{\mathcal{M}, X})$ and it produces a similarity measure $c \in [0, 1]$.

C. Data

To train the model, we use satellite images from datasets released in [24] and collect additional data from Google Earth historical images to include seasonal variation.

There are a total of 18 different areas with 3 to 15 satellite images acquired per area. For each area, we define a grid of possible sampling locations. For each location, per each epoch, we select one random satellite image from that area, crop a 96×96 m area with random yaw and small random translation around grid

[1]Trained model and instructions for downloading training data available at https://irobotics.aalto.fi/sivl
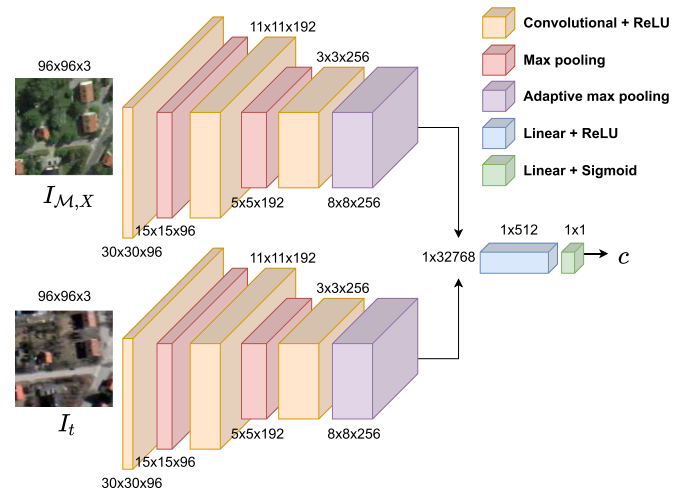

Fig. 2. Model structure for the image similarity network $c = f(I_{\mathcal{M}, X}, I_t)$. Branches share parameters.

point, and label that sample *anchor*. We also generate another sample using the same crop parameters but from another image of the same area, which we label the *positive* sample. A third, *negative* sample is generated by selecting a random location and orientation within the same area. The training dataset consists of 21870 unique locations for sampling, the testing dataset contains 1392 locations, and the weighing function estimation dataset has 5568 locations.

We perform various augmentations during training using [29]. We apply random flips and transposes on the data, to generate more data. We also apply Gaussian noise, various means of blur, sharpening, emboss, brightness, contrast and color changes to gain additional robustness to illumination changes and imaging noise. Additionally, to provide robustness against orthoprojection errors, we apply small geometric transformations.

In training, we use binary cross-entropy loss, setting target to 1 for the pair of anchor and positive samples, and to 0 for the pair of anchor and negative samples. The model is trained using Adam optimizer with learning rate 10^{-5} and a weight decay of 10^{-8} for 1000 epochs.

D. Computing Importance Factor From Similarity Measure

We want to calculate the importance factor w_t^r for each particle r at time t to incorporate the camera observation in the particle set. We compute the importance factor as $w_r^t = P(X_t^r | \mathcal{I}_t, \mathcal{M}) = p(S = s | c_t^r)$, where $S = \{s, u, o\}$ is a variable that determines if the measurement was a match ($S = s$), not a match ($S = u$), or an outlier ($S = o$), and c_t^r is determined by (2). We assume $p(o) = \beta$ where $\beta = 0.05$. As we resample after each update, we assume uninformed priors $p(s) = p(u) = (1 - \beta)/2$. We calculate the importance factor as

$$p(s | c_t^r) = \frac{p(c|s)p(s)}{\sum_{i \in \{s, u, o\}} p(c|i)p(i)} \tag{3}$$

To compute (3), we estimate the probability density function (pdf) $p(c|s)$ from samples corresponding with true pose, using Gaussian kernel density estimation with Scott's bandwidth
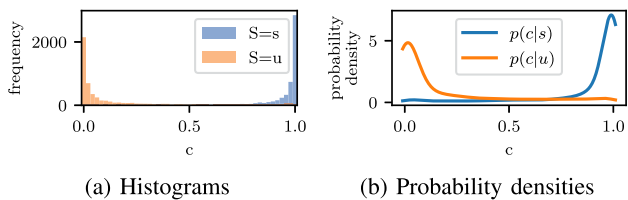
Fig. 3. Histograms of similarity measures and estimated probability densities.

Fig. 4. Orthoimages used in simulated experiments as map (a), "summer" (b), and "winter" (c) measurements. Each orthoimage is 4800 × 2987 m at 1 m/pixel resolution.

rule [30]. Similarly, we estimate a pdf corresponding with incorrect pose $p(c|u)$ for a number of randomly drawn poses. To collect samples corresponding with true pose ($S = s$), we extract pairs of subimages from satellite images corresponding with same area in a similar way as in Section III-C and compute values of c for each pair from (2). To collect samples corresponding with false pose ($S = u$), we extract pairs of subimages from non-corresponding locations. In estimating $p(c|s)$ and $p(c|u)$, we use satellite images from areas that were not used in training f. A histogram of similarity scores for the two classes and corresponding probability density functions are visualized in Fig. 3. The outlier class pdf $p(c|o)$ is assumed to be uniform over the value range of c. The outlier class is included in order to avoid overly confident classifications in regions of c where very small amount of data is available.

IV. EXPERIMENTS

We test the accuracy of the proposed localization system, using a learned similarity score, in two experiments.

In the first experiment (Section IV-B) we want to evaluate the seasonal invariance of the proposed solution. To this end, we test the ability of the model to localize through simulated flights over urban and non-urban locations in our dataset under both *significant* appearance changes (*i.e.*, winter imagery against a map acquired in summer) as well as *minor* appearance changes (*i.e.*, summer imagery against a map acquired in the summer but taken at a different time). We compare the localization performance using our similarity measure method to six other similarity measures.

The second experiment (Section IV-C) attempts to identify how the proposed localization solution works on real UAV data, which include a perspective change as well.

A. Experimental Setting

1) Prior on Initial Pose: In each experiment, the MCL filter is initialized with 1000 particles in a 100 × 100 m area around the true starting position, with scale 0.95 . . . 1.05 and with no a priori information on yaw. This represents a situation where an end user is able to state an inaccurate starting location for the flight of a UAV, without having to input information on initial orientation.

2) Odometry Noise: In all experiments, the UAV takes a sample image approximately every 100 meters. To simulate the impact of odometry noise, we add normally distributed noise with 2 m standard deviation in x and y translation and 1° standard deviation in orientation with respect to the pose increment computed from ground-truth. These parameters are in line
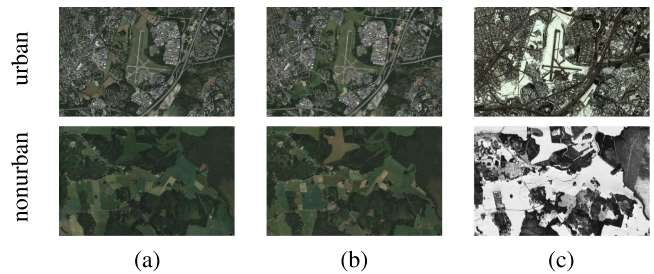
with typical performance reported for monocular visual-inertial odometry solutions in UAVs [31]. Scale noise is assumed to be zero-mean Gaussian with a standard deviation of 0.001.

B. Localization Under Seasonal Appearance Variation

We tested the performance of the proposed localization method under minor and significant seasonal appearance change in both urban and non-urban areas. We selected two testing datasets, representing a non-urban and an urban area, respectively, selected three orthoimages from each dataset (shown in Fig. 4), and formulated an experiment where a simulated UAV flies above each area. One orthoimage acquired during summer was used as map. Another orthoimage acquired during summer was used for generating measurements for the *minor* seasonal appearance case and one orthoimage acquired during winter was used for generating measurements for the *significant* seasonal appearance case.

100 simulated flights were executed for each combination of minor/significant seasonal appearance and urban/non-urban area. In each run, the starting position and yaw of the simulated UAV was randomly selected within the map. Motion of the UAV was simulated with random changes in heading for a duration of 100 updates, making sure the trajectory stays within the map, and localization performance of the MCL algorithm was quantified by computing the weighted mean Euclidean distance to ground-truth position in (x, y)-coordinates.

We compared the localization performance achieved using the proposed method against six other map matching methods. To compare with another learning-based method, we chose Samano *et al.* [17] that has source code and learned weights made available by the authors. To apply that approach to our problem, instead of using a semantic map, unavailable in our scenario, we leveraged their use of an intra-domain loss on UAV images in training their embedding generator network and we generated 16D embedding vectors from both $I_{\mathcal{M}, X_t^r}$ and I_t using their trained image feature extractor and projection modules. We computed particle weights based on distance in embedding space using the linear scaling method proposed by the authors. Performance of deep learning matching methods has rarely been evaluated in relation to more traditional metrics, which are still finding practical use in real localization applications. For this reason, we also compared against the matching method recently proposed by Jurevičius *et al.* [20], using logistic conversion with $v = 0.2$. As other classical image similarity measures, we
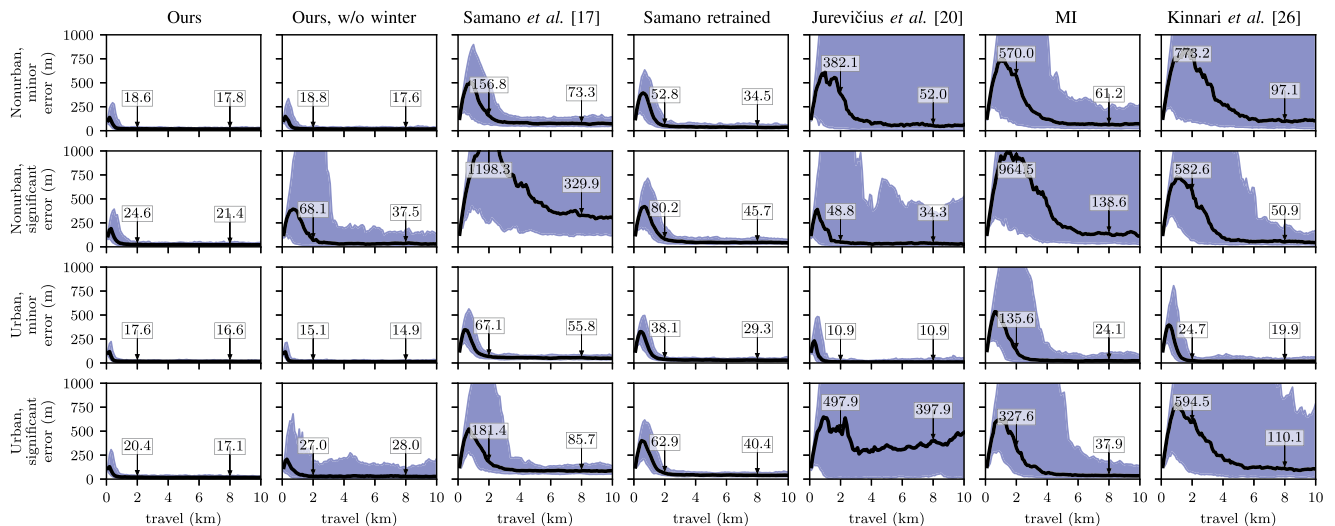
Fig. 5. Errors in simulated localization experiments when using different similarity measures, over different types of terrain. *Minor* appearance change refers to summer-to-summer matching, while *significant* refers to winter-to-summer matching. Medians of mean errors after 20 and 80 updates annotated.

used MI, which has gained attention in other UAV localization approaches [21], [23] and is shown by Mantelli *et al.* [6] to provide marginally superior success rate to abBRIEF. Another classical similarity measure that we used as comparison is Moravec, which we evaluated in a previous work [26].

In addition to Samano *et al.*, Jurevičius *et al.*, MI and our previous approach, we also trained our model without any winter imagery (called "Ours, w/o winter" in Fig. 5) to ablate what we regard as the most important data augmentation source for winter-summer localization. Finally, we trained an embedding vector generator (called "Samano retrained" in Fig. 5) with the model structure of [17] with the same data that we use for our proposed method. The "Samano retrained" model is trained using triplet loss, Adam optimizer, and learning rate 10^{-5} until the testing loss stopped improving (at approximately 200 epochs). For the "Samano retrained" model, we weighed the particles using the linear scaling method in [17].

In our MI and Moravec implementations, we weighed the particles by the Gaussian kernel density estimation method described in Section III-D. Each algorithm was fed the same odometry measurements and camera images, with the exception that the camera image fed to Samano *et al.*'s network was scaled to 128×128 pixels and it was taken from a 95×95 m area to correspond with the design choices in [17]. Each algorithm also used the same map.

Fig. 5 shows the median and 5...95% interval for weighted mean errors computed over all 100 runs in all permutations of urban/non-urban area type and minor/significant appearance change, using our method and the three comparison methods for similarity score computation. The initial error increase in all methods is due to no information on initial yaw. Once particles representing false hypotheses for yaw die out, the error decreases if the correct mode in the search space is found, *i.e.*, if the filter converges to the correct state.

Compared to the reference methods, our method provides both faster convergence time and smaller error bounds after

convergence than all the comparison algorithms. In terms of median of mean errors after convergence, Jurevičius appears to provide lower median for localization error in the case of very high texture (urban environment) and minor appearance change. We suspect this difference may be due to data augmentation by small geometric transformations that we used in training; *i.e.*, the network is trained to give high scores for slightly offset pairs of images. Degradation of convergence and mean error performance on our model trained without winter data show the value of training on data that contains the expected variability. The same can be seen in the comparison between Samano *et al.* network and the retrained Samano model.

C. Localization on Real UAV Data

Besides the simulated experiments using orthoimages, we ran an experiment with three datasets collected with a UAV[2] to identify performance gaps with our model trained on orthoimage data only. The trajectories are shown on a map in Fig. 6 and they cover forest areas, fields, residential areas, and a lake. Ground-truth pose of the UAV in these experiments has been obtained through RTK-GNSS which ensures precision in the centimeter range. Additional characteristics of these flights are listed in Table I. The map used in these experiments was acquired during summer, and the UAV flights took place during autumn months. In UAV dataset 1, deciduous trees are showing autumn colors and in UAV datasets 2 and 3, deciduous trees have dropped leaves.

In this work, the images acquired by the UAV were ortho-projected using ground-truth position information and a digital elevation model (DEM) of the environment where the flight takes place, but in a final use case, orthoprojection can be done *e.g.*, using the method presented in earlier work [26] or by the use of calibrated downward-facing camera and an altimeter. The choice

---

[2]Data provided by Saab Dynamics Ab.
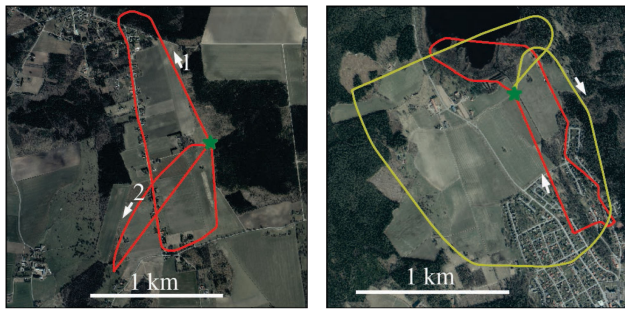
(a) Map of area A (Klockrike, Sweden) and trajectory of UAV dataset 1.
(b) Map of area B (Kisa, Sweden) and trajectory of UAV datasets 2 (red), 3 (yellow)

Fig. 6. ground-truth real UAV trajectories. Starting location marked with green star.

TABLE I
CHARACTERISTICS OF FLIGHTS IN EXPERIMENTS WITH REAL UAV DATA. TRAJECTORY LENGTHS COMPUTED ALONG (x, y) PLANE, AND CAMERA ANGLES BETWEEN NADIR AND CAMERA PRINCIPAL AXIS.

Set	Area	Traj. length (m)	Alt. (m)	Mean camera angle [range] (°)	Acquisition time
1	A	6888	92	50.9 [48.5, 61.6]	Oct 2019
2	B	4080	91	60.7 [55.2, 70.0]	Nov 2019
3	B	6361	92	52.6 [48.9, 118.9]	Nov 2019

to use DEM was made to exclude the impact of possible errors in elevation estimation. The DEM and the orthoimage map of the areas for UAV experiments were purchased from a local map information supplier[3].

The weighted mean error in (x, y)-coordinates for all three UAV flights is visualized in Fig. 7. On all UAV datasets, solution appears to converge after approximately 2 km of travel. The mean errors after 2 km of travel for UAV datasets 1, 2 and 3 were 26.5 m, 29.1 m and 30.6 m, respectively. The localization error and rate of convergence appears to mainly follow the conclusions drawn from the orthoimage experiment. Interestingly, our method without winter data appears to converge faster on dataset 1, possibly due to better fit between the environmental conditions of that particular flight data and the training data. With our method, time to convergence is longer than with the orthoimage experiment. Mean error before convergence (at approximately 1...1.5 km of travel) appears to exceed the 5%...95% interval estimated with the orthoimage experiment with all UAV datasets.

To better understand the performance of our similarity measure with the UAV datasets and understand the potential reason for this performance gap, we computed the similarity measure using ground-truth pose and plotted it for one of the UAV datasets in Fig. 8. For comparison, we also used the UAV image with ten random false poses per image to generate negative examples, and plotted their average similarity measure over the full trajectory. With an ideal method, the true poses would always yield a similarity measure very close to one, and the mean of random poses would be close to zero. From Fig. 8 we see that similarity measure appears to be least reliable over forest areas. This failure
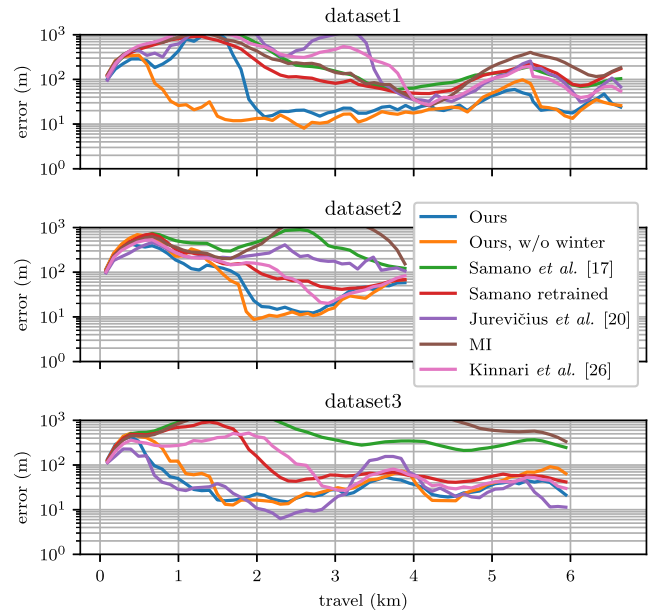
Fig. 7. Errors with different trajectories on real UAV data. Logarithmic vertical scale.

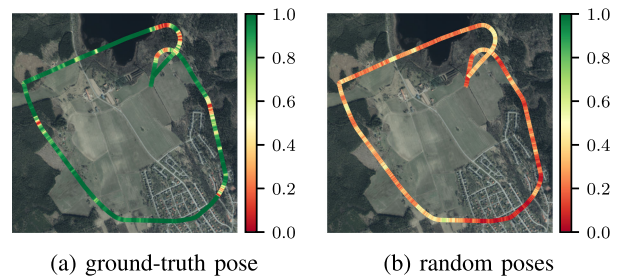(a) ground-truth pose
(b) random poses

Fig. 8. Value of similarity measure computed (a) with true poses of UAV dataset 3 and (b) as average of ten random poses per UAV frame, plotted over a map.

mode appears in all of the UAV datasets and the geometric distortion of tall trees as seen from the UAV compared to the orthoview is a likely explanation to the initial difference in localization error performance between the UAV and orthoimage experiments.

Manual visual inspection of similarity measures produced by individual camera frames also leads to the remark that when the UAV is flying over a forest area where the density of trees is such that trees appear geometrically very distorted in the orthoprojection, similarity measure is often very low. Three exemplary UAV images, corresponding map crops, and similarity measure around true pose are visualized in Fig. 9. Example 3 shows a case where geometric distortion of trees due to orthoprojection appears to affect similarity score significantly. Conversely, in an environment with numerous spatially distinct visual details such as buildings and roads (Fig. 9, Example 1), the similarity measure shows a peak near the true state. The width of the peak is approximately 40 m in translation error and a few degrees in rotation error. Large width of the peak in proportion to the locality of the visual details is possibly affected by data augmentation, alignment errors of training
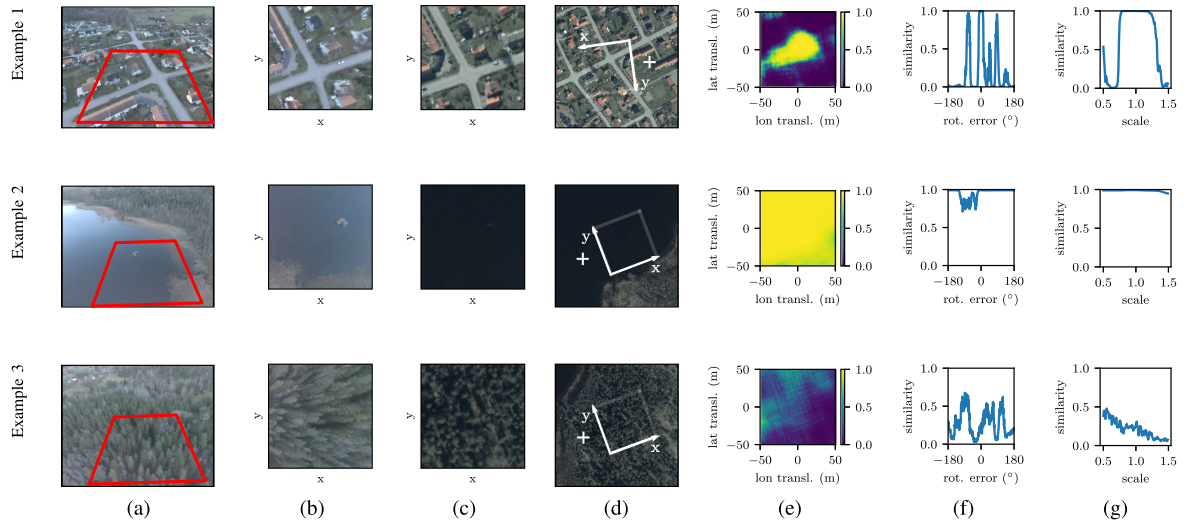
Fig. 9. Similarity measure produced by the learning model under rotation, translation and scaling near true pose in example UAV image frames and poses. (a) $\mathcal{I}_t$ with outline of 96 m × 96 m square used for orthoprojecting I_t highlighted; (b) I_t; (c) $I_{\mathcal{M},X_t^T}$ (here visualized using true pose as X_t^T); (d) true X (white plus sign) and cropping square corresponding with X_t^T overlaid on map; (e) similarity measure value when translating around true pose (translation along map coordinate axes); (f) and (g) similarity measure under rotations and scaling near true pose, respectively.

data, or both. The network also does not appear to have high specificity; *e.g.*, in Fig. 9, Example 1, adjacent intersections appear to produce high scores. The importance of texture is apparent also in Fig. 9, Example 2: there, for an image with extremely few visual features (taken when flying over a lake), the model produces a valid matching score in vicinity of the true state, but is unable to show a peaked output due to natural ambiguity of the environment. In such cases, our solution falls back to giving high likelihood for all particles corresponding to ambiguous terrain area, in effect relying on odometry, until unambiguous terrain areas are observed again.

Image patch extraction time is 0.33 s and inference time of f is 0.13 s at each update on an Intel i7-9750H and NVidia Quadro RTX 3000 using $N = 1000$ particles. We compute f in batches of 100 image patches. Time consumption of both steps scales linearly with N (*e.g.*, for $N = 10000$, times are 3.3 s and 1.3 s, respectively). As we perform an update every 100 m of travel, time between updates is significantly longer than the computing time for typical flight speeds of small UAVs. This suggests the update can be run in real time, also on more resource constrained platforms. We exclude the analysis of computational requirements of VIO from this paper and refer the interested reader to [31].

V. DISCUSSION

The experiments with orthoimages demonstrate that by using a trained model for UAV image-to-map matching, significant reductions in convergence time and localization error can be achieved, compared to reference methods, in cases of both mild and significant seasonal appearance change in urban and non-urban environments.

The comparison to Pearson correlation-based approach [20] and MI-based approach hints towards the interpretation that in this task, trained models appear to outperform classical engineered methods in both convergence time and localization

error, the only exceptions being the performance of [17] on non-urban, significant appearance change, which is considerably out of training data domain in their solution, and the median error after convergence of [20] in the urban, minor appearance change case. In that one case, while [20] is able to achieve lower median error at convergence, we still achieve lower range of error across runs (*i.e.*, narrower 5th–95th percentiles) and faster convergence.

When looking at the comparison with the other learning method [17], our method performs better across all experiments. This can be partially explained by the fact that, in [17], the authors have not specifically trained their embedding generator to be robust to significant seasonal variance and they appear to focus on urban environments. This validates the need for season-invariant methods for visual localization of UAVs. However, also in the case of in-domain data for their method (minor seasonal variation in urban environments), our method still provides faster convergence and smaller error. This, together with the slightly better localization performance of our model compared to the retrained Samano model, hints at the possibility that our method may be a more suitable for this task.

The experiments with UAV data show that the model trained on orthoimages is able to localize also orthoprojected images from a UAV camera. The experiments also demonstrate that there is room for improvement on localization accuracy due to geometric appearance change introduced by an off-nadir viewpoint from a perspective camera.

Further investigation on model structure and dataset composition may yield improved results. The model was not trained specifically for a narrow peak of the output on correct pose; considering the training method and model structure may yield further improvements in localization, while incorporating a portion of UAV data in the training dataset might make the method more robust to perspective distortion.

VI. CONCLUSION

We proposed a method for localizing a UAV with respect to an orthophoto map, in case of significant seasonal appearance change, trained using only satellite images taken at different times of year. We demonstrated the improvement in convergence time and localization error compared to six reference methods in simulated experiments involving significant seasonal appearance change. We showed the ability of the method to be used for localization of a real UAV and identified the most likely error sources for further development.

Our results demonstrate that we can build models that are robust to appearance changes due to seasonal variations. However, seasons are only one source of variation for the dynamically changing operation environments of autonomous agents. The ability of agents to cope with all those variations will be crucial for their deployment in practice.

REFERENCES

[1] D. Scaramuzza and Z. Zhang, "Aerial robots, visual-inertial odometry of," in *Encyclopedia of Robotics*. Berlin/Heidelberg, Germany: Springer, 2020, pp. 1–9. [Online]. Available: https://doi.org/10.1007/978-3-642-41610-1_71-1

[2] C. Cadena et al., "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[3] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. Hoboken, NJ, USA: MIT Press, 2005.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[5] A. Cesetti, E. Frontoni, A. Mancini, A. Ascani, P. Zingaretti, and S. Longhi, "A visual global positioning system for unmanned aerial vehicles used in photogrammetric applications," *J. Intell. Robotic Syst.*, vol. 61, pp. 157–168, 2011.

[6] M. Mantelli et al., "A novel measurement model based on abBRIEF for global localization of a UAV over satellite images," *Robot. Auton. Syst.*, vol. 112, pp. 304–319, 2019.

[7] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Proc. Eur. Conf. Comput. Vis.*, Berlin, Heidelberg: Springer, 2010, pp. 778–792.

[8] H. Hou et al., "UAV pose estimation in GNSS-denied environment assisted by satellite imagery deep learning features," *IEEE Access*, vol. 9, pp. 6358–6367, 2021.

[9] M. Dusmanu et al., "D2-Net: A trainable CNN for joint description and detection of local features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8084–8093, doi: 10.1109/CVPR.2019.00828.

[10] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.

[11] J. Choi and H. Myung, "Brm localization: UAV localization in GNSS-denied environments based on matching of numerical map and UAV images," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 4537–4544.

[12] S. J. Dumble and P. Gibbens, "Airborne vision-aided navigation using road intersection features," *J. Intell. Robotic Syst.*, vol. 78, pp. 185–204, 2015.

[13] A. Volkova and P. W. Gibbens, "More robust features for adaptive visual navigation of UAV in mixed environments," *J. Intell. robotic Syst.*, vol. 90, no. 1, pp. 171–187, 2018.

[14] M. Schleiss, "Translating aerial images into street-map-like representations for visual self-localization of UAV," *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 4213, pp. 575–580, 2019.

[15] A. Masselli, R. Hanten, and A. Zell, "Localization of unmanned aerial vehicles using terrain classification from aerial images," in *Intelligent Autonomous Systems*, E. Menegatti, N. Michael, K. Berns, and H. Yamaguchi, Eds., Cham, Switzerland: Springer International Publishing, 2016, pp. 831–842.

[16] A. Nassar, K. Amer, R. ElHakim, and M. ElHelw, "A deep CNN-based framework for enhanced aerial imagery registration with applications to uav geolocalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 1594–159410.

[17] N. Samano, M. Zhou, and A. Calway, "Global aerial localisation using image and map embeddings," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 5788–5794.

[18] A. Couturier and M. A. Akhloufi, "Convolutional neural networks and particle filter for UAV localization," in *Proc. Unmanned Syst. Technol. XXIII*, 2021, pp. 108–120. [Online]. Available: https://doi.org/10.1117/12.2585986

[19] K. Pearson, "VII. Mathematical contributions to the theory of evolution III. Regression, heredity, and panmixia," *Philos. Trans. Roy. Soc. London. Ser. A*, vol. 187, pp. 253–318, 1896.

[20] R. Jurevičius, V. Marcinkevičius, and J. Šeibokas, "Robust GNSS-denied localization for UAV using particle filter and visual odometry," *Mach. Vis. Appl.*, vol. 30, pp. 1181–1190, 2019.

[21] B. Patel, T. D. Barfoot, and A. P. Schoellig, "Visual localization with google earth images for robust global pose estimation of uavs," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 6491–6497.

[22] M. Bianchi and T. D. Barfoot, "UAV localization using autoencoded satellite images," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 1761–1768, Apr. 2021.

[23] A. Yol, B. Delabarre, A. Dame, J.-E. Dartois, and E. Marchand, "Vision-based absolute localization for unmanned aerial vehicles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 3429–3434.

[24] H. Goforth and S. Lucey, "Gps-denied UAV localization using pre-existing satellite imagery," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 2974–2980.

[25] C. Masone and B. Caputo, "A survey on deep visual place recognition," *IEEE Access*, vol. 9, pp. 19516–19547, 2021.

[26] J. Kinnari, F. Verdoja, and V. Kyrki, "Gnss-denied geolocalization of UAV by visual matching of onboard camera images with orthophotos," in *Proc. 20th Int. Conf. Adv. Robot. (ICAR)*, 2021, pp. 555–562.

[27] A. Martinelli, "Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 44–60, Feb. 2012.

[28] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015.

[29] A. Buslaev, A. Parinov, V. I. I. E. Khvedchenya, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, 2018 Art. no. 125.

[30] D. W. Scott, *Multivariate Density Estimation*. Hoboken, NJ, USA: Wiley, 1992.

[31] J. Delmerico and D. Scaramuzza, "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 2502–2509.