2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Global Aerial Localisation Using Image and Map Embeddings

Noe Samano, Mengjie Zhou and Andrew Calway

***Abstract*— We present a purely vision based geolocation method for aircraft flying over urban and suburban environments. The method is based on matching aerial images with geolocated map tiles using a shared low dimensional embedded space of descriptors. The Euclidean distance between descriptors is used as a similarity measure between domains. The similarity between the observation and map locations is then integrated with visual odometry to track the aircraft's position and yaw using a particle filter. Furthermore, we propose an efficient method to generate map descriptors in testing time based on interpolation, allowing compact representation of large areas giving the potential for high levels of scalability. We experimented in different cities with areas above 20 km^2 in size and preliminary results based on a database of aerial imagery demonstrate that the method gives good results.**

I. INTRODUCTION

Geolocation refers to finding the global location (longitude and latitude) of a sensor in an environment, which is an essential requirement for autonomous robots. Currently, the standard solution is to use the Global Positional System (GPS). However, even the most sophisticated and costly systems are susceptible to signal loss and true autonomy is compromised by reliance on an external infrastructure . For this reason, many researchers have focused their attention on developing alternative geolocation methods.

In the field of aerial vehicles, extensive work has been proposed on geolocation using vision based methods, including visual odometry [1] and, more extensively, matching aerial imagery to databases of geo-tagged images [2]–[4]. The former suffer from significant drift issues, whilst an inherent disadvantage of the latter is the dependence on the time of capture of the database images. Moreover, the logistics to maintain an extensive image database are not trivial.

There has been considerable work done on matching aerial imagery with invariant features extracted from cartographic maps, typically in the form of road networks [5]–[12]. Although significant advances have been reported, these methods are still limited to areas rich in this particular feature. In common with these works, we share the idea that using invariant descriptors extracted from maps can lead to more robust geolocation. Maps offer a semantic representation of the world which is invariant to environmental and weather conditions, and to sensing devices, as well as being compact and regularly maintained and updated. However, in contrast to the above works, we seek to make full use of all semantic characteristics within the maps, to give greater generalisation.

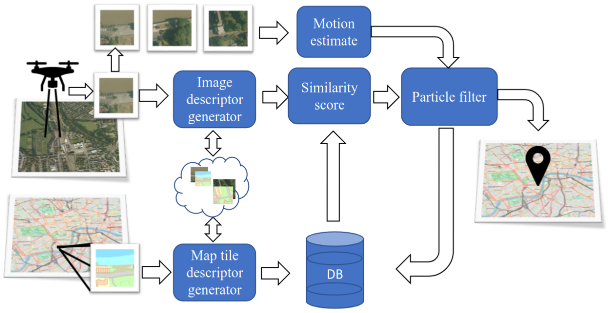

Fig. 1: Our method uses descriptor vectors from a learnt embedded space to compare an aircraft's aerial view with tiles from a cartographic map to get a similarity measure. The similarity score coupled with visual odometry extracted from a sequence of images allow us to track the geolocation and yaw of the aircraft using a particle filter.

In our previous work [13], we demonstrated that descriptors from a learnt embedded space linking street level panoramic images and map tiles could be combined over routes to enable highly accurate and robust geolocation in urban environments. In this work, we extend the idea to show that the method can also be applied to estimating the geolocation and yaw of an aircraft based on aerial imagery. Although the central concept of learning an embedded space between the aerial images and map tiles is similarly employed, there are important and significant challenges in applying the method to an aerial scenario.

For instance, a key assumption in [13] was that the images were located on the road network, whereas for an aerial application no such constraints can be imposed. This significantly increases the size of the tracking space and hence the scope for error, and the computational and storage space required to maintain a database of map descriptors. We address both of these issues, adopting a Monte Carlo localisation (MCL) [14] approach for efficient and robust geolocation over time and a descriptor interpolation strategy for scalable representation of the map.

An overview of the approach is shown in Fig.1. It uses two networks to extract low-dimensional descriptors (16D) from aerial images and map tiles in such a way that semantically similar locations are close to each other. We split the map into a grid of tiles and generate a descriptor for each one, along with those for a subset of tile orientations. Using a particle filter to implement MCL, we then track the aircraft geolocation and yaw by comparing the descriptor computed from aerial images with those interpolated from

The authors are with the Department of Computer Science, University of Bristol, Bristol, U.K. {obed.samanoabonce, mengjie.zhou,andrew.calway@bristol.ac.uk}

978-1-7281-9077-8/21/$31.00 ©2021 IEEE

the tile grid for particle locations and yaw angles. We use visual odometry between frames based on ORB features [15] extracted from the aerial images to provide motion prediction within the filter.

The performance of the approach was investigated using aerial images and cartographic maps obtained from the Digimap [16] and Ordnance Survey (OS) Mastermap [17] databases. These were used both for training in order to learn the embedded space and for testing, by simulating multiple aircraft flights over several urban and suburban areas in the UK. Results demonstrate that the method is effective and has significant potential as an alternative scalable approach to aerial geolocation.

The paper is organised as follows. Section II presents a review of related work and Section III describes the data used in the experiments. The embedding network and training procedure is described in Section IV and details of the MCL particle filter are given in Section V. Experiments and results are presented in Section VI, and the paper concludes with a discussion on the direction of future work.

II. RELATED WORK

In this section, we review the literature focusing on aerial geolocation methods using visual information. We can distinguish two main approaches, those using images and those using maps as reference databases.

Among proposals in the first group, traditional computer vision methods such as feature-based similarity scores [18], [19], bag-of-visual-words BoVW [20] and template matching [21], [22] have been proposed. However, normally these techniques only work when the similarity of the query's image and the reference image is high. In [2], authors extract local features from the robot observation and perform image registration with a reference satellite image. Accuracy is then improved using a semantic shape matching algorithm. In [4], a learning approach is used to estimate a drone's position in an environment previously visited. In [3], authors present a cross-view method to estimate the global pose of a UAV using satellite imagery. The method uses two Siamese neural networks. One localises the scene, and the other estimates the camera pose. Our network does something similar to their scene recognition network but using map tiles images instead of satellite images. Other than that, our methods are different.

Among the approaches using maps are [23]–[25]. In [23], images captured by a drone are segmented and classified into semantic groups. Then images are compared against an environmental map to estimate the 2D position using template matching. Patterson et al. [24] estimate 2D UAV's position using map data from the Ordnance Survey as the reference and a motion estimate from an IMU. Images and maps are first converted to a common representation, and then a correlation-based matching is used to estimate position. Shan et al. [25] use Google Maps to aid the navigation of UAV in GPS-denied environments. The method is initialized using correlation. Then optical flow and homography are used to predict position in subsequent frames. Matching between image and map is done by registration of Histogram of Oriented Gradients and a particle filter. In [26] authors propose a global localization method based on matching segmented buildings on captured images with a reference numerical map using area ratios as features. Combining these features with visual and inertial odometry, global localisation is achieved. However, the method is still limited to areas rich on buildings. Over the recent years, rendered images from Google Earth or elevation maps have also been proposed as a reference for pose estimation and localisation [27], [28].

TABLE I: Geographic areas used for training and testing in this work [17], [29]–[35]. The year refers to the aerial imagery. "A" is the area in km^2.

City	Tile Names	Year (20–)	A	Use
London	TQ28SE, TQ27NE	15/16	50	Train
Reading	SU77SW	13	25	Train
Countryside	SO82	13	100	Train
London	TQ38SW, TQ37NW	15/16	50	Test
Bristol	ST57SE	17,16,14	25	Test
Oxford	SP50SW	16	25	Test

Significant work using road networks as source of reference information has also been published [5]–[12]. In [6], [8], [9] road intersections are used for visual-aided navigation using similarity transforms to match the query features with the reference road network. In [10], intersections and roads in observations are matched with roads extracted from OpenStreetMaps (OSM) to estimate location. Accuracy is then improved by aligning images using an affine transformation. Máttyus and Fraundorfer [11] also match road segments on sequences of images to road networks. Detection of roads in the aerial image is performed by tracking cars and verified using pixel color road detector. Matching against the road network database is performed using an efficient hashing algorithm. The method is scalable to large areas, but it requires traffic on the roads to work. Similarly, Li et al. [12] also use sequences of images and registration against a road network. Improvements in the stitching of sequences and registration methodology are reported to make it faster and more robust.

III. DATASET

In this preliminary work, we used topographic map data from the Mastermap database provided by the Ordnance Survey [17] (dated November 2019). For aerial imagery we used high resolution (25 cm) images from Getmapping Plc [29]–[35] accessed using Digimap [16]. We used map tiles rendered at 256×256 pixels and each tile corresponded to an area of approximately 95×95 m^2 giving an approximate ground sample distance (GSD) of 0.37 $m/pixel$.

A summary of the areas, including tile names according to the British National Grid, extension in km^2 and the use given, is shown in Table I. The "year" column refers to the aerial data only. In London, the area is covered mainly by imagery captured in 2015, except for a small section in the south where newer data was available. For Bristol, we downloaded imagery data from three different years to

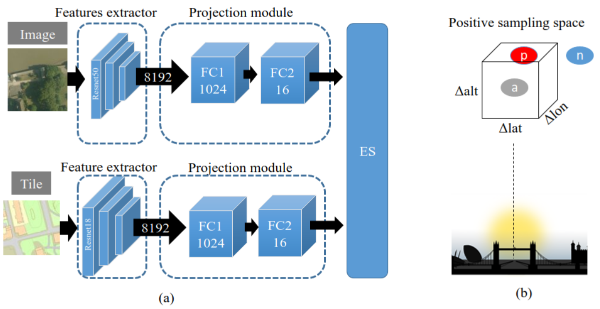

Fig. 2: a) Network architecture b) Positive examples are sampled from a cuboid space centred in the anchor location.

test our algorithm under different conditions. Note that none of the testing areas overlap geographically with those for training.

IV. Embedding Network

A. Network architecture

Full details of the network architecture can be found in our previous work [13]. A simplified diagram is shown in Fig 2a. The network has two branches, one for processing images and the other for map tiles. Each branch has a feature extractor and a projection module. Feature extractor modules are based on Resnet50 and Resnet18 architectures [36] for images and maps, respectively. The projection modules are fully connected layers. The input to the map network is a down sampled 128×128 pixel map tile. As opposed to that in [13], where the image network's input was a list of 4 images, here it is a single 128×128 pixel image. Outputs of both branches are 16-D vectors that are normalized and scaled to form a hypersphere manifold of radius 32.

B. Training Procedure

The training procedure is similar to the reported in [13]. We generate matched pairs as follows. Let $\mathcal{A} := A_1, A_2, ..., A_n$ represent a collection of geographic areas for training. And let X and Y denote a mosaic of map tiles and images, respectively, covering the area. We start the training process by taking n random locations among the available areas in $\mathcal{A}$ to form a set L_B of locations in the training batch. Then, for each location $l_i \in L_B$, we sample K neighbour locations to be used as positives in the triplet loss by applying a random shift $\Delta l \sim U(-30, 30)$ m to the anchor location. Afterwards, for each resulting location inside the batch, we crop 256×256 tiles and images from X and Y respectively. Random scaling $s \sim U(0.707, 1.414)$ is also applied to simulate changes in altitude. This process is similar to sampling from a cuboid space around the central location; see Fig 2b for an illustration. We also apply a small random rotation $r \sim N(0, 5)$ degrees, but limited to small values since we aim to distinguish between descriptors from samples of different orientations. Finally, tiles and images are resized to 128×128 pixels before been feed to the network. Additionally, we include a standard normalization and random erasing [37] in both inputs.

For the loss function, we used the weighted soft-margin ranking loss proposed in [38]. It is defined as $\mathcal{L}_{wgt}(d) = ln(1 + e^{\alpha d})$, where d is the difference between a matched pair descriptor distance (Euclidean) and an unmatched pair descriptor distance, and α is a weighting factor that helps to improve convergence [38]. We also adopted a similar strategy to [39] and included bidirectional cross-domain and intra-domain ranking constraints to force the network to preserve embedding structure in both data representations. Our loss function is therefore given by

$$\mathcal{L} = \sum_{i,j,k,l,m} \mathcal{J}_{wgt}(\mathbf{x}_{ik}, \mathbf{y}_{il}, \mathbf{y}_{jm}) + \mathcal{J}_{wgt}(\mathbf{y}_{ik}, \mathbf{x}_{il}, \mathbf{x}_{jm})$$
$$+ \sum_{i,j,k,l,m,k \neq l} \mathcal{J}_{wgt}(\mathbf{x}_{ik}, \mathbf{x}_{il}, \mathbf{x}_{jm}) + \mathcal{J}_{wgt}(\mathbf{y}_{ik}, \mathbf{y}_{il}, \mathbf{y}_{jm}) \tag{1}$$

where $\mathcal{J}_{wgt}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathcal{L}_{wgt}(d(\mathbf{x}, \mathbf{y}) - d(\mathbf{x}, \mathbf{z}))$, $\mathbf{x}_{ik}$ and $\mathbf{y}_{ik}$ denote the embedded vectors corresponding to the kth augmentation of the map tile and image at location i, respectively, i.e. the outputs from the sub-networks in Fig 2a, and $d(.,.)$ denotes Euclidean distance. Note that i and j refer to different locations, i.e. $1 \leq i, j \leq N$ and $i \neq j$, and k, l and m refer to the K augmentations per location, i.e. $1 \leq k, l, m \leq K$.

For training, we used a batch size of 25 locations and $K = 10$ augmentations for 1×10^6 iterations using the Adam optimizer [40] with learning rate of 4×10^{-5}. Image and map feature extractors were initialized with Places365 [41] and ImageNet [42] weights, respectively. Projection modules in both branches were initialized randomly and we formed triplets using the batch all mining strategy [43], [44].

V. Monte Carlo Localisation

The Monte Carlo localization (MCL) method [14] is a non parametric implementation of the Bayes filter to solve non linear filtering problems. The latter allows recursive estimation of the posterior probability distribution $bel(x_t)$ of a state x at time t according to

$$bel(x_t) = \eta p(z_t|x_t) \int p(x_t|x_{t-1}, u_t) bel(x_{t-1}) dx_{t-1} \tag{2}$$

where $p(x_t|x_{t-1}, u_t)$ is the state transition probability given control input u_t, $p(z_t|x_t)$ is the probability of the measurement z_t given state x_t and η is a normalisation factor. In this work, our objective is to find the geolocation in terms of latitude (lat) and longitude (lon) and yaw (θ) of an aircraft and hence our state is defined as $x_t = (lat_t, lon_t, \theta_t)$.

In MCL, $bel(x_t)$ is represented by a collection of N samples called particles $\boldsymbol{X_t} := x_t^1, x_t^2, ..., x_t^N$, each of which is a sample of the state x_t at discrete time instances. Associated with each particle is a weight ω_t^n, which represents the importance of the sample. Particles are propagated between states using a motion model representing $p(x_t|x_{t-1}, u_t)$ and updated at each time step using an observation model representing $p(z_t|x_t)$. The weighted mean of the particles then gives the current estimate of the state, i.e., $\hat{x_t} = \eta_t \Sigma_{n=1}^N \omega_t^n x_t^n$, where $\eta_t = 1/\sum_{n=1}^N \omega_t^n$.

A. Observation Model

Our approach is built upon the idea of comparing images with map tiles using the embedded space to provide the observation model. The Euclidean distance is used as a similarity measure between embedded space descriptors corresponding to aerial images and map tiles. Let $\boldsymbol{F}(\cdot)$ and $\boldsymbol{G}(\cdot)$ represent the image and map networks respectively. Then, given particle state x_t^n, our measurement z_t^n is $z_t^n = dist(\boldsymbol{F}(Y_t), \boldsymbol{G}(X_t^n))$, where Y_t is the aerial image captured by the aircraft, X_t^n is the map tile at location (lat_t^n, lon_t^n) rotated by yaw angle θ_t^n and $dist()$ is the Euclidean distance. Since the embedded descriptors lie within a hypersphere of radius 32, we then define the probability of obtaining the measurement as $P(z_t^n|x_t^n) = (64 - z_t^n)/64$ and the particle weights are then updated according to $\omega_t^n = \omega_{t-1}^n P(z_t|x_t^n)$, followed by normalisation $\omega_t^n \rightarrow \eta_t \omega_t^n$.

To speed up localisation, we precompute descriptors for all the map tiles needed to cover the testing area at discrete locations and orientations and then interpolate amongst them to estimate a descriptor for a given state. The descriptors are stored in a multidimensional array $\mathcal{D} : W \times H \times \Theta$ where W, H are the width and height of the map in tiles and Θ is the number of discrete orientations of the map (8 in this work). Hence, each location (i, j, k) in $\mathcal{D}$ stores a 16-D map tile descriptor $\mathbf{d}_{ijk} = \boldsymbol{G}(X_{ijk})$, where X_{ijk} are map tiles defined on a regular spatial grid with spacing Δ_W and Δ_H and rotated by angle $k\Delta_\Theta$. At test time, given particle state x_t^n, descriptor $\boldsymbol{G}(X_t^n)$ is estimated amongst its eight nearest spatial neighbours within $\mathcal{D}$ using trilinear interpolation. This process is faster than using the model to compute descriptors at each iteration, and we only need 5.8 MB to store an area of 50 km^2.

B. Motion Model

We estimate the motion control u_t using visual odometry based on point correspondences between aerial images captured at times t and $t-1$. We model the transformation between the images as a homography $\boldsymbol{H}$ and estimate this using ORB features [15] combined with RANSAC [45]. Here we have assumed that images are being captured at sufficient height to allow a planar surface approximation. This suffices for the data used in the experiments. A more robust algorithm may be required when deploying in a real-world application. We then decompose $\boldsymbol{H}$ to extract the aircraft's relative translation $\hat{\boldsymbol{t}}$ and rotation angle $\hat{\phi}$ [46]. This then allows propagation of each particles' state using the estimated translation and rotation:

$$\begin{aligned} \theta_t^n &= \theta_{t-1}^n + \hat{\phi}_t + \mathcal{N}_\theta \\ \boldsymbol{\mathcal{X}_t^n} &= \boldsymbol{\mathcal{X}_{t-1}^n} + \alpha R(\gamma_t^n)\hat{\boldsymbol{t}}_t + \boldsymbol{\mathcal{N}_{\mathcal{X}}}, \end{aligned} \tag{3}$$

where $\boldsymbol{\mathcal{X}_t^n}$ represents the 2-D geolocation of the particle x_t^n in meters, α is the ground sample distance (GSD) and $R(\gamma_t^n)$ is a 2x2 rotation matrix defined by angle $\gamma_t^n = -(\theta_{t-1}^n + \hat{\phi}_t)$. Noise components $\mathcal{N}_\theta$ and $\boldsymbol{\mathcal{N}_{\mathcal{X}}}$ are zero mean Gaussian with standard deviation $\sigma = 5$ degrees and covariance $\boldsymbol{\Sigma} = [10, 0; 0, 10]$ m, respectively.

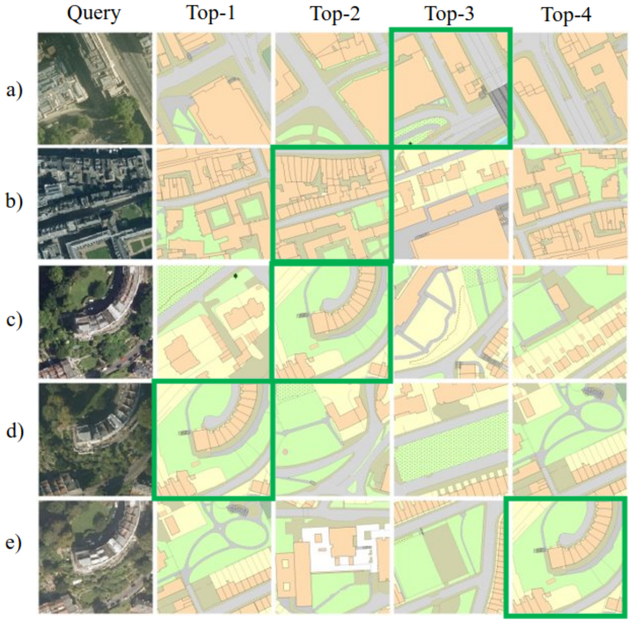

Fig. 3: Retrieving examples given a query image in (a) London, (b) Oxford, (c) Bristol 2017, (d) Bristol 2016 and (e) Bristol 2014. The ground truth map tiles are outlined in green. Note that (c-e) corresponds to the same place but with aerial images captured in different years.

C. Implementation details

To perform our experiments, we simulated an aircraft flying over the testing areas parallel to the ground using Newton's kinematic equations. The time step between updates has been set to 1 s. For any test trajectory, the aircraft's initial state is set randomly. The initial velocity is sampled from $\mathcal{N}(5, 1)$ m/s. Then, at each step an acceleration command $\boldsymbol{a} \sim \mathcal{N}(0, 0.33)$ m/s^2 and a turn command $t \sim \mathcal{N}(5, 0)$ degrees is applied but constraining the velocity between 0 and 10 m/s. When the aircraft approaches a boundary a systematic turn $t = 5$ degrees command is applied. Besides, we simulate changes in the altitude by scaling observed images according to $scale = 2^{0.25sin(2\pi t/50)}$. Hence, the sampling ground distance vary from 0.31 to 0.44 $m/pixel$.

For all experiments, we initialized N = 20000 particles uniformly distributed in the testing area. We perform resampling using the low variance method when the number of effective particles, given by $N_t^{eff} = \frac{1}{\Sigma_{n=1}^{N}(\omega_t^n)^2}$, is less than $2N/3$ and drop 10% in every resampling step until only 5000 remain. All experiments were performed on an i7-6700 CPU @3.40 GHz with 8 cores. The average time to complete a step in the simulation is 75 ms.

VI. EXPERIMENTAL RESULTS

A. Embedding Images and Maps

A qualitative evaluation of our network is presented in Fig. 3, in which we show examples of retrieved map tiles given a query image in different testing areas. The ground truth map tile is outlined in green. It is interesting to note

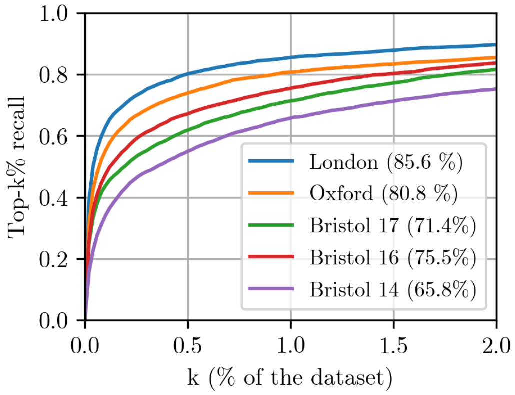

Fig. 4: Top-k% recall for the three testing areas, where k% is the fraction of the dataset size. Bristol 17, Bristol 16 and Bristol 14 correspond to the same geographic area but using aerial imagery captured in 2017, 2016 and 2014 respectively. The value in brackets is the top-1% recall.

the semantic similarity between the query image and the retrieved maps. Note that the last three examples, Fig. 3c-e, correspond to the same geographic location but using images from different years. The images were ranked in the top-4 in the three cases despite changes in illumination and capturing conditions. Besides, using imagery from old years is more challenging due to the potential mismatch with the current maps and the observations. Although ranking using data from 2014 is not as good as that of more recent years, such rankings are still encouraging given the number of possible locations on the map, indicating that the network uses a diversity of features to match locations. It still allows effective geolocation, albeit slightly less reliable, as shown in Fig. 5 and described below.

We also evaluated the retrieving ability of the network quantitatively throughout a top-k rank curve. We used 5000 random generated pairs in each of the testing areas to create the plot. Results are shown in Fig. 4. The percentage in brackets is the top-1% recall. We can observe that the network has learned to match pairs very well, achieving above 65% top-1% for all testing areas. Unsurprisingly, the best performance is achieved in London, which keeps a high similarity with the training data and is a dense urban environment rich in features. In contrast, the performance in Bristol is lower, especially when using data from 2014. We believe it is related, among other factors, to mismatches between images and maps.

B. Geolocation and Yaw Estimation

To evaluate geolocation performance of our method, we randomly generated 100 trajectories per area of different lengths. Long trajectories (between 1 km and 7 km and around 4 km on average) were simulated to show the performance in long term localisation. In the case of Bristol, the same trajectories were used when using data from different years. Example trajectories for each area are shown in Fig.5a-c, where we compare the ground truth (GT) with the estimation provided by visual odometry alone (VO) and when combined with our embedded descriptors (VO+ES). Observe that our method converges to the true trajectory in all cases, correcting the drift accumulated by visual odometry.

In Fig.5d-f, we show the Mean Localisation Error (MLE), i.e. the distance between the estimated position and the ground truth, and the yaw estimation error for a set of test trajectories in London, Oxford, and Bristol, respectively. Five trajectories are reported for London and Oxford. Meanwhile, in Bristol, we present two trajectories but using observations from different years. Large errors are observed initially prior to the filter converging but convergence is achieved for almost all trajectories by 200 time steps. After convergence, the method maintains track with small localisation (typically around 100 m) and orientation errors. Occasionally, estimates may degrade as a result of perceptual aliasing, for example, when the aircraft flies over parks or rivers, as can be verified in Fig.5b-c. However, the method is able to recover once more informative features are captured.

To evaluate convergence, Fig.5g shows the number of trajectories as a function of the time step in which convergence was detected, where we deemed a trajectory as localised if the absolute error drops below 95 m (the size of a tile). Observe that 78.2% of the total number of trajectories (500) converged and 61.4% did this by 200 time steps.

We also evaluated the tracking ability of our method by computing the RMSE for every trajectory after convergence. Results for localisation and yaw are shown in Fig.5h and Fig.5i, respectively. Observe that the location error is typically between 50 m and 100 m, which we regard as reasonable considering the size of testing areas. The yaw RMSE is less than 45 degrees for the 78.8% of the converged trajectories with a typical value between 5 and 10 degrees.

Error variation in London is smaller because of the high similarity between this area and the training data. Besides, as it was pointed above, London's environment is very dense, and thus there are many features to match. However, experiments in Oxford and Bristol demonstrate that the method can work in different environments. Although these areas are only half in extension as London, they present challenging characteristics, for example, large extensions of land covered by parks, sparse features and of course changes in environmental conditions. Even under these circumstances, we observe good convergence rates and tracking performance.

VII. CONCLUSIONS

We have presented a new methodology to globally geolocate an aircraft in an environment using visual-only information and map tiles as a reference. Our methodology relates the observed images with an abstract representation of the world in the form of a map, which is invariant to the capturing conditions and can be maintained and stored easily. Furthermore, we have proposed a method to generate new descriptors using linear interpolation, and therefore it is not computationally expensive. Map descriptors for an area of 50 km^2 can be stored in a file of only 5.8 MB. Our preliminary results based on simulations show that the method is feasible

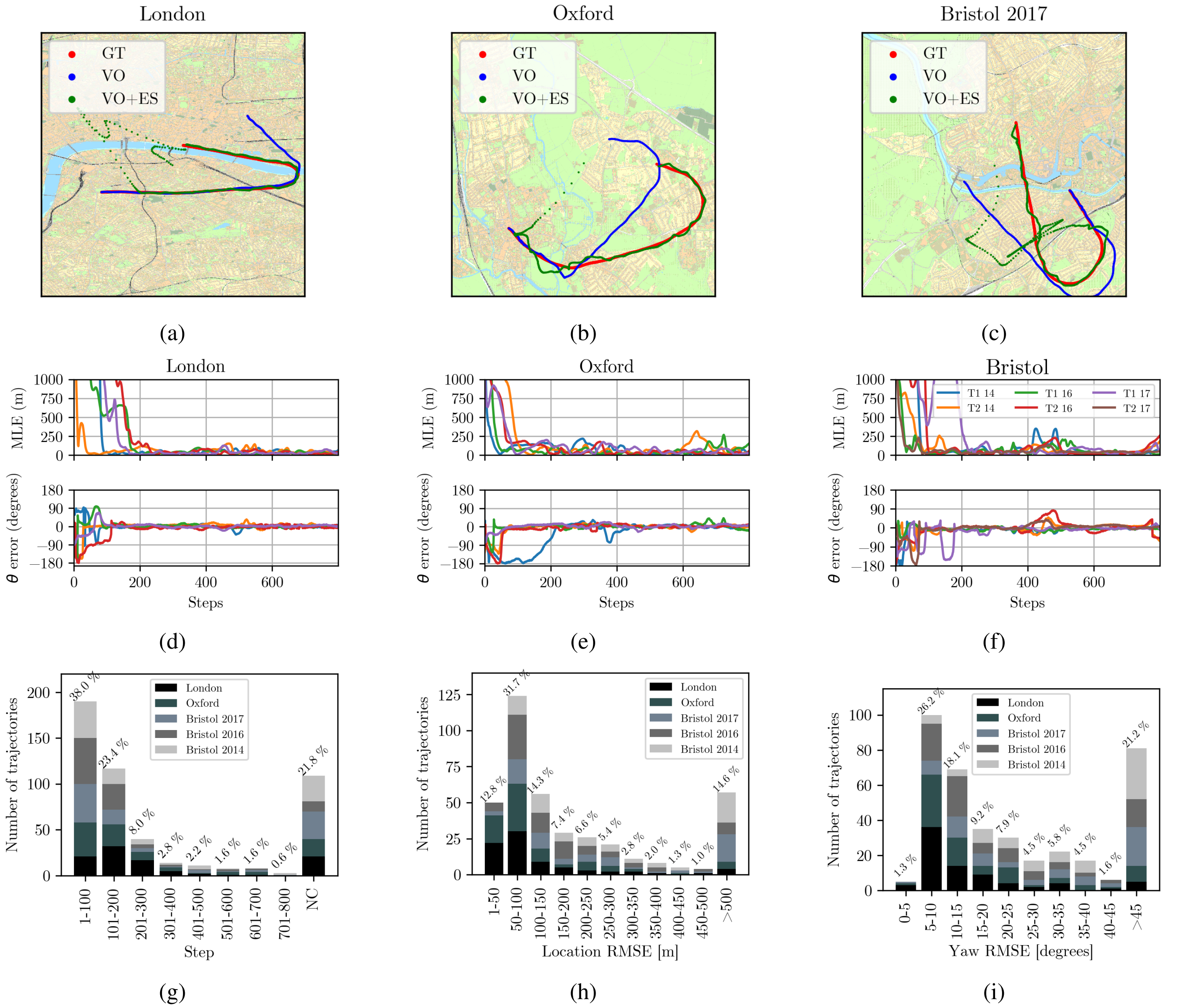

Fig. 5: Area map and example trajectories comparing ground truth (GT), visual odometry alone (VO) and visual odometry combined with our descriptors (VO+ES) in (a) London (b) Oxford and (c) Bristol; Mean localisation error (top) and yaw error (bottom) for five trajectories in (d) London and (e) Oxford; (f) Mean localisation error and yaw error for two test trajectories using aerial images from 3 different years in Bristol; (g) Number of trajectories successfully localised as a function of the step by area; Trajectory RMSE distribution for geolocation (h) and yaw (i) after convergence by area.

and scalable to large areas. In future work, we plan to collect real data and test the method on it. Besides, we plan to experiment with multi-scale levels to improve robustness.

ACKNOWLEDGMENTS

We gratefully acknowledge the support of the Mexican Council of Science and Technology (CONACyT) and the Chinese Scholarship Council (CSC). We also would like to acknowledge the Ordnance Survey, Getmapping Plc and Digimaps for making available the data. The Mastermap data is under copyright of ©Crown copyright and database rights 2020 Ordnance Survey (100025252), and the aerial data is under copyright of ©Getmapping Plc.

REFERENCES

[1] V. Guizilini and F. Ramos, "Visual odometry learning for unmanned aerial vehicles," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 6213–6220.

[2] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1513–1523.

[3] 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)*. IEEE, 2019, pp. 1827–1833.

[4] A. A. Cabrera-Ponce and J. Martinez-Carranza, "Aerial geolocalisation for mavs using posenet," in *2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS)*. IEEE, 2019, pp. 192–198.

[5] S. Z. Li, J. Kittler, and M. Petrou, "Matching and recognition of road networks from aerial images," in *European Conference on Computer Vision*. Springer, 1992, pp. 857–861.

[6] L. Wu and Y. Hu, "Vision-aided navigation for aircrafts based on road junction detection," in *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, vol. 4. IEEE, 2009, pp. 164–169.

[7] K. Kozempel and R. Reulke, "Camera orientation based on matching road networks," in *2009 24th International Conference Image and Vision Computing New Zealand*. IEEE, 2009, pp. 237–242.

[8] J. Jung, J. Yun, C.-K. Ryoo, and K. Choi, "Vision based navigation using road-intersection image," in *2011 11th International Conference on Control, Automation and Systems*. IEEE, 2011, pp. 964–968.

[9] S. J. Dumble and P. W. Gibbens, "Airborne vision-aided navigation using road intersection features," *Journal of Intelligent & Robotic Systems*, vol. 78, no. 2, pp. 185–204, 2015.

[10] D. Costea and M. Leordeanu, "Aerial image geolocalization from recognition and matching of roads and intersections," *arXiv preprint arXiv:1605.08323*, 2016.

[11] G. Máttyus and F. Fraundorfer, "Aerial image sequence geolocalization with road traffic as invariant feature," *Image and Vision Computing*, vol. 52, pp. 218–229, 2016.

[12] Y. Li, H. He, D. Yang, S. Wang, and M. Zhang, "Geolocalization with aerial image sequence for uavs," *Autonomous Robots*, vol. 44, no. 7, pp. 1199–1215, 2020.

[13] N. Samano, M. Zhou, and A. Calway, "You are here: Geolocation by embedding maps and images," in *European Conference on Computer Vision*, 2020.

[14] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 2. IEEE, 1999, pp. 1322–1328.

[15] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[16] Digimap. [Online]. Available: https://digimap.edina.ac.uk/

[17] Ordnance Survey (GB), OS MasterMap® Topography Layer [FileGeoDatabase geospatial data], Scale 1:1250, Tiles: GB, Updated: 28 November 2019, Using: EDINA Digimap Ordnance Survey Service, Downloaded: May-Oct 2020. [Online]. Available: https://digimap.edina.ac.uk

[18] B. Fan, Y. Du, L. Zhu, and Y. Tang, "The registration of uav down-looking aerial images to satellite images with image entropy and edges," in *International Conference on Intelligent Robotics and Applications*. Springer, 2010, pp. 609–617.

[19] 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*, vol. 24, no. 1, pp. 1–18, 2002.

[20] M. Divecha and S. Newsam, "Large-scale geolocalization of overhead imagery," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2016, pp. 1–9.

[21] G. Conte and P. Doherty, "Vision-based unmanned aerial vehicle navigation using geo-referenced information," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, pp. 1–18, 2009.

[22] A. Yol, B. Delabarre, A. Dame, J.-E. Dartois, and E. Marchand, "Vision-based absolute localization for unmanned aerial vehicles," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 3429–3434.

[23] F. Lindsten, J. Callmer, H. Ohlsson, D. Törnqvist, T. B. Schön, and F. Gustafsson, "Geo-referencing for uav navigation using environmental classification," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 1420–1425.

[24] T. Patterson, S. McClean, P. Morrow, and G. Parr, "Utilizing geographic information system data for unmanned aerial vehicle position estimation," in *2011 Canadian Conference on Computer and Robot Vision*. IEEE, 2011, pp. 8–15.

[25] 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," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2015, pp. 114–119.

[26] J. Choi and H. Myung, "Brm localization: Uav localization in gnss-denied environments based on matching of numerical map and uav images," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4537–4544.

[27] B. Patel, T. D. Barfoot, and A. P. Schoellig, "Visual localization with google earth images for robust global pose estimation of uavs," *IEEE Robotics and Automation Letters*, 2020.

[28] T. Hinzmann and R. Siegwart, "Deep uav localization with reference view rendering," *arXiv preprint arXiv:2008.04619*, 2020.

[29] Getmapping Plc, High Resolution (25cm) Vertical Aerial Imagery (2015/2016) [WMS web map service], Scale 1:500, Tiles: tq27ne,tq28se,tq37nw,tq38sw, Updated: 2015/2016, Using: EDINA Aerial Web Map Service, Accessed: 2020-05-21. [Online]. Available: https://digimap.edina.ac.uk

[30] ——, High Resolution (25cm) Vertical Aerial Imagery (2016) [JPG geospatial data], Scale 1:500, Tiles: sp50nw, Updated: 25 October 2016, Getmapping, Using: EDINA Aerial Digimap Service, Downloaded: 2020-09-04 12:17:57.596. [Online]. Available: https://digimap.edina.ac.uk

[31] ——, High Resolution (25cm) Vertical Aerial Imagery (2014) [JPG geospatial data], Scale 1:500, Tiles: so82, Updated: 25 October 2014, Getmapping, Using: EDINA Aerial Digimap Service, Downloaded: Downloaded: 2020-09-01 18:44:22.379. [Online]. Available: https://digimap.edina.ac.uk

[32] ——, High Resolution (25cm) Vertical Aerial Imagery (2013) [JPG geospatial data], Scale 1:500, Tiles: su77sw, Updated: 11 October 2013, Getmapping, Using: EDINA Aerial Digimap Service, Downloaded: 2020-10-17 17:44:10.04. [Online]. Available: https://digimap.edina.ac.uk

[33] ——, High Resolution (25cm) Vertical Aerial Imagery (2014) [JPG geospatial data], Scale 1:500, Tiles: st57se, Updated: 25 October 2014, Using: EDINA Aerial Digimap Service, Downloaded: 2020-08-26 11:53:58.227. [Online]. Available: https://digimap.edina.ac.uk

[34] ——, High Resolution (25cm) Vertical Aerial Imagery (2016) [JPG geospatial data], Scale 1:500, Tiles: st57se, Updated: 25 October 2016, Using: EDINA Aerial Digimap Service, Downloaded: 2020-08-26 11:53:58.227. [Online]. Available: https://digimap.edina.ac.uk

[35] ——, High Resolution (25cm) Vertical Aerial Imagery (2017) [JPG geospatial data], Scale 1:500, Tiles: st57se, Updated: 5 November 2017, Using: EDINA Aerial Digimap Service, Downloaded: 2020-08-26 11:53:58.227. [Online]. Available: https://digimap.edina.ac.uk

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.

[37] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation." in *AAAI*, 2020, pp. 13 001–13 008.

[38] S. Hu, M. Feng, R. M. Nguyen, and G. Hee Lee, "Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2018.

[39] L. Wang, Y. Li, and S. Lazebnik, "Learning deep structure-preserving image-text embeddings," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

[41] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Trans on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1452–1464, 2017.

[42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[43] S. Ding, L. Lin, G. Wang, and H. Chao, "Deep feature learning with relative distance comparison for person re-identification," *Pattern Recognition*, vol. 48, no. 10, pp. 2993–3003, 2015.

[44] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv:1703.07737*, 2017.

[45] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[46] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.