USVLoc : Infrared Topographic Based Cross-view Localization in Maritime Environment*

Alexandre Foucher[1 [0009−0006−5723−9516]], Cédric Seguin[1 [0000−0001−7057−4710]], Dominque Heller[1 [0009−0006−4717−8449]], and Johann Laurent[1 [0000−0001−7133−759X]]

Université Bretagne-Sud, Lab-STICC CNRS UMR 6285, Lorient, France
{firstname}.{lastname}@univ-ubs.fr

Abstract. In the marine environment, geolocation is crucial for autonomous navigation of surface systems. Nowadays, most autonomous navigation methods still mainly rely on GNSS for localization. However, GNSS is vulnerable to jamming and spoofing and needs to be compensated to improve overall resilience. Visual Geo-localization is a passive alternative and represents a significant challenge in maritime environments. Besides, over the past decade, we observed an increasing complexity in the architectures of neural networks. Although this trend is proportional to the evolution of hardware capabilities, it is necessary to enable small embedded systems to use these approaches in real time. This paper explores the limit and the trade-off for lightweight visual geo-localization using limited field-of-view IR and RGB cameras for a horizon based cross-view correlation for 3 degrees of freedom localization. We propose to correlate ground images with the simulated horizon from the elevation model using a lightweight cross-modal network and compact visual descriptors. We also present a geotagged dataset for visual navigation of IR and RGB data recorded from an inland maritime system. Our results highlight the feasibility of GNSS-free navigation for real-time maritime systems, with positioning to within 20 m and 10° of deviation in 3 degrees of freedom. Opening new possibilities for robust, real-time, autonomous operations on small embedded systems.

Keywords: Computer Vision · Visual-Geolocalization · Embedded System · Maritime.

1 Introduction

Unmanned systems have become increasingly prominent in recent years in industrial, military, and academic fields. Their low economic cost and unique advantages, such as safety and security of operations in remote environments, make them valuable solutions. Among those systems, Unmanned Surface Vehicles (USVs) demonstrate their adaptability in various scenarios such as surveillance and reconnaissance.

* This study is being conducted with the funding support of Agence de l'Innovation de Défense - Ministère des Armées - France.

Thanks to the growing capabilities of Artificial Intelligence (AI) and embedded systems, it is now possible to execute automation chains in real time on board. Despite these advances, systems still depend on remote supervisors to monitor operations and intervene in the event of a malfunction. Following this observation, the major step forward in this field is to reduce this human dependence by designing more autonomous systems.

One critical aspect of achieving this autonomy is localization, which enables subsequent navigation planning. Particularly USVs mostly rely on accurate position measurements provided by Global Navigation Satellite System (GNSS). However, the GNSS accuracy can be compromised due to vegetation and structure obstruction but also by malicious attacks such as spoofing and jamming. The potential loss of accurate geolocation can compromise long-term autonomous operations. In this context, it is essential to develop alternative positioning methods for USVs that do not rely on any external sources for safety.

In maritime environments, vessels commonly use rotating radar sensors to identify obstacles and coastlines, allowing them to accurately determine their position [10, 16]. However, some issues arise: radar effectiveness is highly dependent on their height above water, and for small USVs, it is not possible to achieve sufficient height. Added to that the price of the sensor, which is not insignificant. Moreover, by emitting long-range waves, radar makes systems detectable and more vulnerable in a potentially hostile environment.

In this article, we address the challenge of accurately positioning USVs operating in coastal areas or inland waters without using external sources or emitting sensors. Visual localization is an ideal alternative in this context, as it allows position to be estimated solely from visual information gather by camera sensor. In addition, as distances can be significant in maritime context, cameras are ideal sensors, enabling passive and distant perception. Although visual localization is a significant academic topic, only a limited amount of articles consider this approach for USVs, as the field of autonomous maritime systems is relatively emerging.

The maritime environment is challenging due to dynamic factors, including waves, tides and weather conditions. HorizonNet[9], a 360-degree topographic correlation method in archipelagos, has highlighted the increasing difficulty of localization when the field of view (FoV) is reduced. However, small USVs are mainly equipped with cameras with limited fields of view, which reduces the amount of visual information available.

In addition to this challenging environment, we've noticed a lack of publicly available data, which makes it almost impossible to compare and replicate existing methods. Some datasets exist for maritime environment but only object detection/classification and segmentation task are represented. To date, we have not identified any dataset for maritime localization providing continuous logs with position information including latitude, longitude, and heading such as Kitti[8] dataset can provide for autonomous driving field.

Our contribution. In this paper, following previous observation, we propose a lightweight architecture for cross-view localization based on neural network and designed to suit small embedded systems constraints. As USVs navigate on water surface, our proposal consists of 3 Degrees of Freedom (DoF) method, which aims to recover latitude, longitude, and heading of the system.

In order to train and validate our proposed model, we also present a publicly available dataset of geotagged data from multiple USV logs, including infrared and color images, GNSS position and heading as well as data from Inertial Measurement Unit (IMU). We also demonstrate the capabilities of our proposal using infrared data for a robust solution in challenging weather conditions and during nighttime over classical color camera.

2 Related Work

2.1 Visual Geo-localization

Visual geo-localization can be described as determining the geolocation and possibly the camera orientation for a given image without any additional information. This approach differs from visual SLAM, as it is a single-shot image retrieval task without a mapping step. Indeed, SLAM works well in structured environments and at short distances, making it unsuitable for maritime applications. Literature offers multiple approaches that can be categorized into two main scales of geo-localization: local and global. As global methods aim to localize anywhere on the planet, they are not limited to specific areas, whereas local methods can be specifically adapted to natural or urban environments [4]. However, most recent methods based on deep learning rely on their datasets for their geolocation knowledge, limiting their geolocation predictions to only those that were learned during training [2, 27]. This limitation restricts the use of these methods in areas where dataset coverage is insufficient or outdated. To address this issue, it is possible to correlate ground images with other types of data that are less dependent on the availability of ground truth data itself, this is named cross-view geo-localization.

2.2 Cross-View geo-localization

Cross-view geo-localization is a specific approach within the visual geo-localization field, which involves matching ground images with another different source of information. More commonly, it refers to the correlation between satellite images and ground images. Even if the earth is totally scanned by satellite imagery, the most accurate methods are still limited to urban areas [13, 22, 5, 14].

However, cross-domain geo-localization isn't limited to satellite views only. Correlation with the horizon is also a valid approach in natural environments.

2.3 Horizon Correlation

Horizon matching methods estimate the observer's location by correlating extracted horizon features with a preliminary known digital elevation model. The

observed horizon contains a variety of topographical information, such as elevation and depth, which can be used to determine a precise location. This approach was first developed for planetary rovers on Mars [24, 6]. Still, this method requires a clear and distant view of the terrain's elevation, favoring terrestrial systems. Also, high computational power may be required depending on the correlation method used, limiting real-time application [19, 9, 15, 25].

In a maritime context on a USV, a research team [9] obtained a mean localization error of 2.72m for a limited search area with a 360° field of view. As a result, they highlight that the localization error grew exponentially as the field of view was reduced, stopping at 18.96m mean error for a 120° field of view. Based on this observation, we will compare our method aimed at compensating for this limitation.

2.4 Siamese Architecture

Although there are some variations between the architectures proposed in the literature, most follow an approach based on Siamese networks. Siamese networks consist of a convolutional neural network (CNN) structure followed by an aggregation layer that generates a consistent scene descriptor that can be compared using a loss function.

Commonly used CNNs are VGG [23] and ResNets [11]. Although they were developed between 2014 and 2017, they remain predominant in the literature. Subsequently, an aggregation layer is responsible for combining the features extracted from the CNN into a single visual descriptor, also called an "embedding" or "feature vector" depending on the research field. Historically, NetVLAD [1] was the first major breakthrough and is still in use to this day. More recently, some lighter alternatives such as GeM [18] or SAFA [21] can be mentioned.

However, as identified by Berton et al [2], over the past decade, we have seen architectures become increasingly complex due to the use of larger models and an ever-growing number of intermediate layers such as transformers [26]. Although this trend is proportional to the evolution of hardware capabilities, it is now necessary to revisit this philosophy to enable small embedded systems to use these approaches in real time.

3 Method

To obtain accurate geolocation from an image despite the limited field of view, we propose a deep learning based pipeline, as shown in Figure 1. This pipeline follows the common cross-view architecture, relying on the correlation of two inputs from different perspectives. The main objective of this approach is to enable two separate networks, working on two separate inputs, to extract the same information from both as visual descriptor. Their main advantage is that after the learning process, the similarity of two inputs is proportional to the distance between their respective descriptors. Here our approach is based on ground

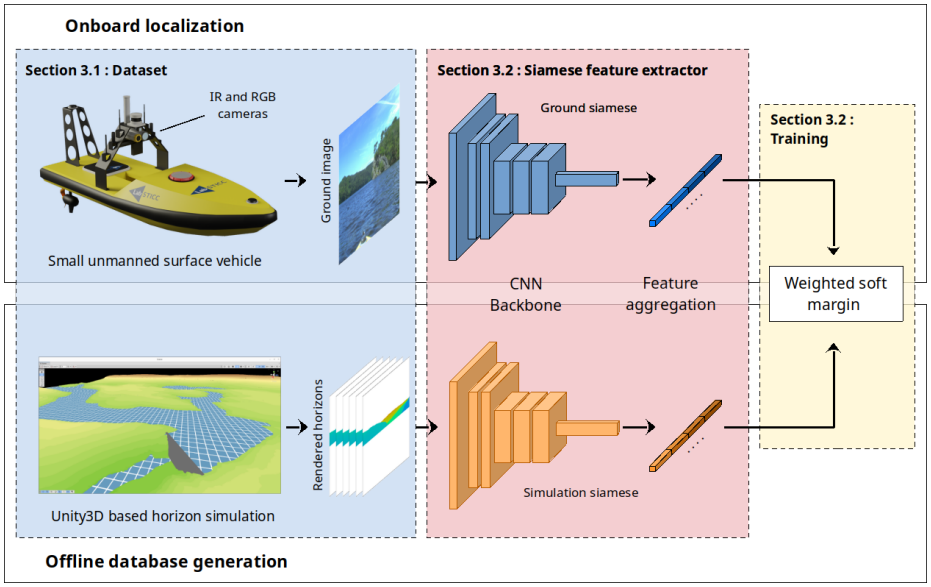

Fig. 1. Diagram of the cross-view approach, which consists of two networks sharing the same architecture but using different weights for simulated horizons and ground image inference. The two networks share a soft-margin triplet loss in order to converge together during training.

image from the USV perception and from rendered image based on previously known local Digital Elevation Map (DEM).

Cross-correlation is based on an image search method in which the highest correlation score is inversely proportional to the distance between descriptors. These descriptors are compared using a K-Nearest Neighbor (KNN) search to retrieve the highest correlation hypotheses. Since descriptors are n-dimensional mathematical vectors, to compare them, we can calculate the distance in their respective spaces, such as the euclidean distance, which will be used throughout the rest of this article.

As our approach is based on elevation maps, we assume that we know approximately the location where the system operates to limit the searching area. This initial hypothesis is not a strong one, as the approximation covers an area of several dozen square kilometers and is not limited to a narrow zone. The principle is to generate a sufficiently exhaustive dataset of digital horizons rendered from topographic data in the first instance that cover all the targeted area. Subsequently, after training a model to recognize similarities and differences in the horizon, a vectorial database is created to store all the descriptors from the simulation dataset. It is important to notice that all computationally expensive processes are performed beforehand, including terrain simulation, horizon rendering, and descriptor database generation. Finally, only the database of descriptors and the trained model are stored onboard the USV, limiting memory

usage and required computing power. For localization, the drone will then need to perform only an inference and a search for the most similar horizons. As mentioned, we distinguish between the dataset, which represents image sets for deep-learning training, and the database, which represents descriptor storage.

3.1 Dataset

In deep learning, the amount and the data quality are crucial to train and evaluate a model. Typically, using public datasets is a good idea to enhance data diversity and facilitate comparisons in benchmarking performances. However, in the maritime domain, most available datasets are limited to image segmentation [3] or object detection and classification tasks [17]. In our case, our method requires not only GNSS geolocation but also heading and topographic elevation information. To address these specific requirements, we created a custom dataset from the logged data of our laboratory's USV. The dataset is made of two parts: a real sampled data part from USV's log and a second made from simulated horizon using DEM of the same area.

Topographic Simulation. As logs were recorded at the Guerledan Lake, we managed to retrieve topographic data from the national French DEM dataset RGE ALTI. This elevation map is made of $1km^2$ grids of all France, each composed of a 1000×1000 matrix representing the elevation with an accuracy of 1 m. Subsequently, the DEM was imported as a terrain object in a simulator made with the Unity3D engine. We picked the Unity3D engine due to its ease of use and the flexibility offered by its Shader Graph tool, which enables high control over visual rendering rules as shown in Figure 2. The simulator is designed to simulate horizon views captured by a 50° FoV camera at a given geographical location to match our USV specifications.

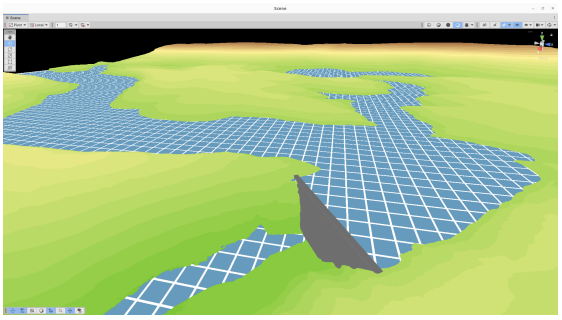

Fig. 2. Unity3D scene rendered using a topographic shader to highlight and enhance the visibility of elevations. The grid on the water indicates all the positions where the camera is placed to render and generate the database.

Notably, in contrast to the real world, the simulation terrain is completely clear of any vegetation or buildings. Because of that, we chose to generate the color of the rendered horizon based on depth ranging from blue for nearby elements to red for distant elements. This depth render add meanfull perspective informations for the retrieval task, allowing to distinguish between two horizons that overlap but are on distant planes. Also, water and sky are intentionally removed to force the model to learn from the horizon.

To create a large dataset, we automated the generation process. We first initialize a matrix of points on the water surface of the terrain. The distance between them, noted r, depends on the resolution of the grid and the camera angle rotation step count, noted n, resulting in n captures with a $360°/n$ step. We set $r = 15m$ and $n = 30$ for the rest of the paper, because after multiple tests, we found that this was a good compromise between maximum theoretical accuracy and the number of images generated. For our $25km^2$ test area we end up with 91290 images for a total weight of $1.6GB$. This amount is less than the maximum expected for a $25km^2$ area, because images are only captured from water.

Ground image. To collect real data for the dataset, we use a small USV equipped with SBG Elipse-D using dual GNSS antennas to compute accurate and stable heading even in static conditions. The system is also equipped with a 50° FoV Pylon color camera and 60° FoV FLIR IR camera. This difference in FoV will be discussed further in the section devoted to results. In addition, we use Robot Operating System (ROS) on the onboard Nvidia Jetson Xavier to manage and synchronize sensors, but also to record every sensor's data as rosbag, enabling replays.

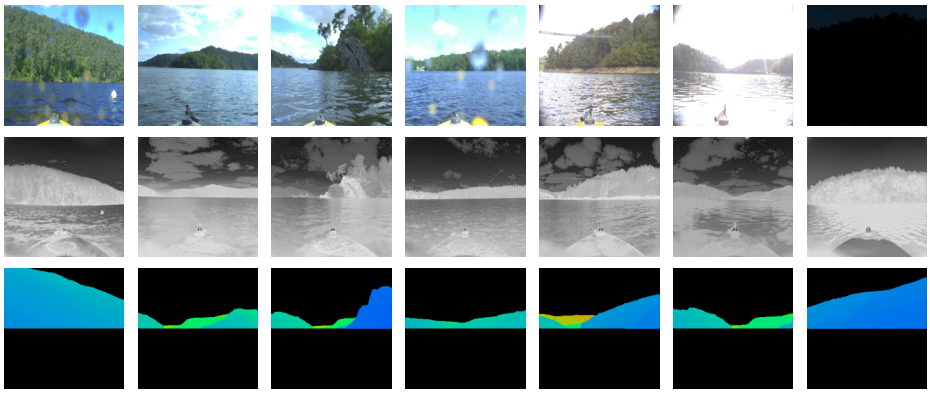

Fig. 3. Samples from the training set of the dataset sorted by column, with the corresponding RGB, IR, and rendered images for each sample.

The dataset was generated using a ROS node that synchronizes geolocation, heading, and image data. The node was fed with several logs created during different sessions conducted over multiple years at the Guerledan lake in France. Since some recordings were made at night, some RGB images will appear dark, while IR images will remain bright, as shown in Figure 3. All data were subsequently split into three sets: training, validation, and testing. The training and validation sets are respectively made of 60151 and 6683 distinct and randomly shuffled geotagged samples. Whereas test sets consist of five complete logs that are never seen in either training or validation presented below in Table 1.

Table 1. Table describing the 5 test sets present in the dataset.

Test	Images	Difficulty	Desc
0	900	Easy	Simple trajectory, cloudy weather preventing sun glare
1	900	Medium	Complex trajectory, some visual occlusion and water droplet on lens
2	900		
3	900	Hard	Area not well covered in the training set, few visual anchors and mostly natural vegetation, sun glare
4	753	Hard	Night time

3.2 Siamese Feature Extractor

Our method relies on horizon recognition rather than satellite images, which shares similarities with cross-view image retrieval architectures. The objective is to retrieve the most similar rendered horizon from a ground-level image. To achieve this, we require feature extraction model made of a CNN backbone followed by an aggregation layer.

Table 2. Model specification Table based on the official implementation of model inside Pytorch. Parameters refer to the number of trainable variables in a model, while GFLOPS corresponds to the computational cost of a model.

Model	Parameters	GFLOPS
ResNet18 [11]	11.7M	1.81
ResNet34 [11]	21.8M	3.66
ResNet50 [11]	25.6M	4.09
ResNet101 [11]	44.5M	7.80
VGG11 [23]	132.9M	7.61
VGG13 [23]	133.1M	11.31
VGG16 [23]	138.4M	15.47

As our goal is to enable real time cross-view for small embedded systems, size of the model and, above all, the number of operations required are important factors. According to the model specifications presented in Table 2, ResNet18 is the smallest model available in the VGG and ResNet family, requiring only 1.81 GLOPS. According to this observation, we decided to base our approach on the ResNet family of models and explore the trade-off between lightweight models and performance.

Subsequently, the output dimensionality of each descriptor (D) can vary significantly depending on the aggregation layer used. For instance, NetVLAD [1] returns a descriptor of 32768-D, whereas SFRS [7] uses 4096-D and GeM [18] uses 512-D. High dimensionality, such as NetVLAD, can have a substantial impact on performance and speed for retrieving candidates. For instance, a database based on NetVLAD would be 64 times heavier than a database using a 512-D descriptor. In fact, we obtain a simulated dataset of 91290 horizons, representing a theoretical database weighing 12 GB using NetVLAD. This may not permit the database to be loaded completely into RAM for fast access or may limit the size of the target area.

Taking these considerations into account, we opted for a less complex approach, utilizing only a ResNet18 CNN backbone followed by a GeM [18] aggregation layer and a Fully Connected (FC) layer with a desired dimensionality as output. In the results section, we will perform several tests on small-dimensional descriptors to measure their impact on the method. The initial assumption is that the smaller the descriptor, the lighter the database and therefore the faster the search for nearest neighbors, but at the detriment of the quality of the neighbors found.

3.3 Training

Relevance score. To identify the similarity between our geotagged ground image and simulated horizon, we propose a relevance score formula (cf. Eq. 1). This formula returns a score in the range of $]-\infty; 1]$, where higher scores indicate higher similarity. The score depends on the difference in position and orientation between two positions $\{lat, lon, hdt\}$.

$$rel = 1 - \frac{\Delta_{\text{hdt}}}{a} - \frac{\Delta_{\text{pos}}}{b} \tag{1}$$

Considering the limitation of the simulated dataset grid resolution shown in the previous section and that a relevance score higher than 0 between a ground image and a simulated horizon is a match, we set $a = 12°$ and $b = 20m$. As the relevance score is computed from heading distance and position distance, $rel \geqslant 0$ means at least $\Delta_{hdt} \leqslant 12°$ with a perfect position match or $\Delta_{pos} \leqslant 20m$ with a perfect heading match.

Online Triplet Mining. The model alone is not enough to guarantee quality, training is essential, as is architecture, including data mining, loss function,

and training parameters. The training is based on semi-supervised online triplet mining, which has been shown to be efficient in the training process [20]. To generate a batch for a training iteration, we gather ground images from the logged dataset and retrieve the k nearest simulated horizons based on geographical distance. Among these k horizons, we randomly select one horizon, which will be the positive elements for the triplets. We therefore assume that with a randomized dataset, all other samples within the current batch will act as negative horizon. All possible triplet combinations are generated online among all available horizons in the batch, giving us 240 triplets for a batchsize of 16.

Ultimately, we chose to use weighted soft margin loss with $\alpha = 10$ (cf. Eq. 2) in accordance with its original implementation [12]. Soft margin results in more efficient training as it is a continuous loss, rather than traditional triplet margin loss, which turns easy triplets into zero loss.

$$\mathcal{L}_{soft} = ln(1 + e^{\alpha(d_{pos} - d_{neg})}) \tag{2}$$

Image augmentation. In order to artificially improve the size and diversity of our dataset, several augmentation techniques were added during the training process. However, since the cross-view method involves two different image sources, some augmentations are not relevant to both. For example, to improve the robustness of the application in bad weather, we randomly add rain and fog to some ground images with random intensity. But, for our simulated horizon images, this specific weather augmentation makes no sense, as these generated images will never be affected by weather conditions. Furthermore, in our case, one of the main augmentation techniques is horizontal image flipping. Since we train our Siamese network to learn the shape and depth characteristics of the horizon, artificially flipping our image doubles the size of the dataset. But in this specific case, we must apply horizontal flipping to all batch images, including ground truth and simulated ones, because the relevance score is based on geographic location and orientation. So if random images were flipped, their score would not always correspond to the actual similarity.

4 Result

4.1 Model performance

As well knowed, recall is the metric used to evaluate the quality of network. It quantify the amount of inference where a positive sample is successfully retrieved. Also noted R@k, this notation stands for recall among the first k nearest elements. Here we evaluate model based on the R@1 as it is the most difficult and also R@5.

In our specific case, a positive element is an element with a relevance score higher than zero. As our relevance formula is used with $a = 12.0°$, $b = 20.0m$ and based on our grid generation setting, this means that among our 91290

simulated horizons only 5 samples are considered positive. In other word, R@1 represent the percentage of inference where the network successfully retireves 1 of 5 positives elements among 91290.

Table 3. Model performance evaluation based on recall R@1 and R@5 with a relevance threshold greater than zero. The evaluation is performed on several combinations of aggregation layers on top of a ResNet backbone. All models are trained for 10 epochs on the same dataset, but with different image types: RGB, IR, and both, then tested on RGB and IR. Each result is averaged over 3 runs with seeds 0, 1, and 2.

Backbone	Aggr.	Desc. Dim.	Train set	RGB		IR	
				R@1	R@5	R@1	R@5
ResNet18	NetVLAD	32768	RGB	6.27± 0.9	17.48± 0.9	0.28± 0.2	1.10± 0.2
			IR	0.00± 0.0	0.00± 0.0	7.91± 1.8	22.92± 2.9
			Both	16.20± 1.1	40.08± 2.4	19.90± 1.2	48.82± 1.0
	GeM	512	RGB	15.42± 2.8	37.89± 4.1	0.06± 0.1	0.31± 0.2
			IR	0.00± 0.0	0.00± 0.0	21.26± 4.1	51.43± 2.9
			Both	**21.11**± 0.4	45.99± 1.3	21.91± 3.1	50.03± 3.9
	GeM + FC	256	RGB	17.09± 1.4	37.01± 3.2	0.48± 0.2	1.75± 0.5
			IR	0.00± 0.0	0.00± 0.0	23.16± 2.9	49.38± 3.0
			Both	18.35± 1.8	41.42± 1.9	18.49± 0.3	42.70± 0.6
		128	RGB	13.76± 1.5	34.50± 1.2	0.18± 0.2	0.63± 0.4
			IR	0.01± 0.0	0.06± 0.1	21.65± 2.0	49.19± 1.0
			Both	20.27± 3.9	42.20± 2.3	19.10± 1.6	44.73± 0.5
		64	RGB	17.46± 3.0	38.65± 2.1	0.47± 0.3	1.03± 0.6
			IR	0.00± 0.0	0.04± 0.1	22.27± 4.0	49.32± 4.9
			Both	19.29± 2.2	44.00± 1.5	20.44± 3.5	46.48± 4.1
ResNet50	GeM	2048	RGB	18.60± 3.2	41.06± 3.3	0.21± 0.1	1.10± 0.8
			IR	0.00± 0.0	0.01± 0.0	**24.08**± 0.5	**52.92**± 0.8
			Both	19.18± 3.1	**46.57**± 0.7	21.98± 2.2	52.25± 1.5

In order to evaluate the impact of the aggregation layer and descriptor size on the method's performance, we compute the mean recall and standard deviation for each model. Each model is trained on three distinct seeds 0, 1, and 2, but also on three different types of data: IR, RGB, and both (IR and RGB combined) for 10 epochs each as loss slowly reach a floor and to prevent to long experimentation. Finally, each model is tested using the five test sets presented in the dataset section for IR and RGB images. In this way, the experiment also highlighted the potential versatility of a model capable of processing both IR and RGB images.

As illustrated in Table 3, although NetVLAD was predominant in the literature, it is the least efficient in our case. This is probably due to the contrast between the large size of the descriptors and the reduced size and complexity of our dataset. On the other hand, the native GeM model along ResNet50 achieves the best performance with for most of the case with RGB-R@5 of 46.57% and also IR-R@5 of 52.92%. It is also interesting to note that the model trained only on IR images achieves better results than the one trained on RGB images. This could be explained because there is a nighttime recording in the test set, which necessarily favors IR vision. But also potentially due to our IR camera, which

has a field of view 10° wider than that of the color camera, captures more information on the horizon, but also because IR images are easier to interpret, as the distinction between water, ground, and sky is clearer.

Furthermore, the model trained on both types of images performs better on RGB images than the model trained only on RGB images. Similarly, our GeM variant with FC layer does not show any significant difference in performance for the dimension of distinct descriptors. This observation allows us to further our research on reducing our descriptor size in order to also reduce the size of the generated database and the nearest neighbor search time.

4.2 Onboard

As our goal is to enable small embedded system to run in real time visual geolocalization method, we performed a benchmark on Jetson Orin embedded GPU. Note that this benchmark is for raw reference only. In practice, it is possible to further improve performance through model optimizations.

Table 4. Onboard model performance computed on Jetson Orin platform. Inference time and KNN time are averaged over 1000 runs. The neural network uses the pytorch library with a float32 model. KNN is performed using the annoy library to retrieve the 5 nearest neighbors.

Backbone	Aggr.	Desc. Dim.	Database size	Inference (ms)	KNN@5 (ms)
ResNet18	NetVLAD	32768	12 GB	20.63 ± 0.74	719.57 ± 81.09
	GeM	512	187 MB	**7.07** ± 0.84	0.55 ± 0.10
	GeM + FC	256	94 MB	7.58 ± 0.85	0.19 ± 0.03
		128	47 MB	7.33 ± 0.80	0.11 ± 0.02
		64	**24 MB**	7.29 ± 0.87	**0.08** ± 0.01
ResNet50	GeM	2048	748 MB	15.78 ± 1.01	4.01 ± 0.61

As Table 4 highlight, NetVLAD is not suitable for embedded use, as the generated database weighs approximately 12 GB, which slows down neighbor search to approximately 719 ms. We also note that the aggregation layer only has a significant impact on inference time, as in the case of NetVLAD, which is three times slower than the default GeM layer. However, the backbone layer also represents a computational load, as illustrated by the difference between ResNet18 and ResNet50, which is 2.2 times slower for a same aggregation layer.

5 Conclusion

In this article, we propose a revisit approach for lightweight geolocation on small embedded system without GNSS, equipped only with passive sensors, including a camera with a limited field of view. We proposed and demonstrated the effectiveness of topographic based cross-view for a restricted field of view in a natural

environment. Using a digital elevation model to render simulated horizons as a reference for a method. This work highlights the feasibility and potential of autonomous navigation in areas where ground images are limited and GNSS is no longer reliable.

This approach was especially developed for small embedded systems with limited computation power and battery capacity, using the light vision model ResNet18 and a highly compressed embedding database with a minimum of 64 dimensions descriptors. In addition to this, we propose a dataset for visual geo-localization, providing IR and RGB images along GNSS latitude, longitude and heading information as exists in the field of autonomous vehicles. Based on this dataset, our proposition achieves recall@5 of 45.99% at a distance less than 20m and heading of less than 12°. Using compact descriptor for cross-view task in an area of $25km^2$ using only a 50° field of view camera.

Our goal is to enable a system to navigate in an unknown environment knowing only the elevation of the terrain. We therefore need to develop an enhance available dataset to guarantee that approach can be generalized to unknown area.

References

1. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: Netvlad: Cnn architecture for weakly supervised place recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5297–5307 (2016)
2. Berton, G., Masone, C., Caputo, B.: Rethinking visual geo-localization for large-scale applications. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4878–4888 (2022)
3. Bovcon, B., Muhovič, J., Perš, J., Kristan, M.: The mastr1325 dataset for training deep usv obstacle detection models. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE (2019)
4. Brejcha, J., M Čadík, M.: State-of-the-art in visual geo-localization. Pattern Analysis and Applications **20**(3), 613–637 (2017)
5. Downes, L.M., Steiner, T.J., Russell, R.L., How, J.P.: Wide-area geolocalization with a limited field of view camera in challenging urban environments. arXiv preprint arXiv:2308.07432 (2023)
6. Ebadi, K., Coble, K., Kogan, D., Atha, D., Schwartz, R., Padgett, C., Vander Hook, J.: Toward autonomous localization of planetary robotic explorers by relying on semantic mapping. In: 2022 IEEE Aerospace Conference (AERO). pp. 1–10. IEEE (2022)
7. Ge, Y., Wang, H., Zhu, F., Zhao, R., Li, H.: Self-supervising fine-grained region similarities for large-scale image localization. In: European conference on computer vision. pp. 369–386. Springer (2020)
8. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
9. Grelsson, B., Robinson, A., Felsberg, M., Khan, F.S.: Gps-level accurate camera localization with horizonnet. Journal of field robotics **37**(6), 951–971 (2020)
10. Han, J., Cho, Y., Kim, J.: Coastal slam with marine radar for usv operation in gps-restricted situations. IEEE Journal of Oceanic Engineering **44**(2), 300–309 (2019)

11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
12. Hu, S., Feng, M., Nguyen, R.M., Lee, G.H.: Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7258–7267 (2018)
13. Hu, S., Lee, G.H.: Image-based geo-localization using satellite imagery. International Journal of Computer Vision **128**(5), 1205–1219 (2020)
14. Huang, G., Zhou, Y., Zhao, L., Gan, W.: Cv-cities: Advancing cross-view geo-localization in global cities. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (2024)
15. Liu, Z., Guo, F., Liu, H., Xiao, X., Tang, J.: Cmlocate: A cross-modal automatic visual geo-localization framework for a natural environment without gnss information. IET Image Processing **17**(12), 3524–3540 (2023)
16. Ma, H., Smart, E., Ahmed, A., Brown, D.: Radar image-based positioning for usv under gps denial environment. IEEE transactions on intelligent transportation systems **19**(1), 72–80 (2017)
17. Nanda, A., Cho, S.W., Lee, H., Park, J.H.: Kolomverse: Korea open large-scale image dataset for object detection in the maritime universe. IEEE Transactions on Intelligent Transportation Systems (2024)
18. Radenović, F., Tolias, G., Chum, O.: Fine-tuning cnn image retrieval with no human annotation. IEEE transactions on pattern analysis and machine intelligence **41**(7), 1655–1668 (2018)
19. Saurer, O., Baatz, G., Köser, K., Ladickỳ, L., Pollefeys, M.: Image based geo-localization in the alps. International Journal of Computer Vision **116**(3), 213–225 (2016)
20. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 815–823 (2015)
21. Shi, Y., Liu, L., Yu, X., Li, H.: Spatial-aware feature aggregation for image based cross-view geo-localization. Advances in Neural Information Processing Systems **32** (2019)
22. Shugaev, M., Semenov, I., Ashley, K., Klaczynski, M., Cuntoor, N., Lee, M.W., Jacobs, N.: Arcgeo: Localizing limited field-of-view images using cross-view matching. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 209–218 (2024)
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
24. Stein, F., Medioni, G.: Map-based localization using the panoramic horizon. IEEE transactions on robotics and automation **11**(6), 892–896 (1995)
25. Tomešek, J., Čadík, M., Brejcha, J.: Crosslocate: cross-modal large-scale visual geo-localization in natural environments using rendered modalities. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3174–3183 (2022)
26. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
27. Vo, N., Jacobs, N., Hays, J.: Revisiting im2gps in the deep learning era. In: Proceedings of the IEEE international conference on computer vision. pp. 2621–2630 (2017)