Self-Supervised Landmark Discovery for Terrain-Relative Navigation

Connor Lee[1], Esmir Mesic[1], and Soon-Jo Chung[1]

***Abstract*— We present a landmark discovery algorithm to automatically detect and identify optimal landmarks for aerial localization in visual terrain-relative navigation (VTRN) pipelines for Global Navigation Satellite Systems (GNSS) denied navigation. Our method employs self-supervised contrastive learning to identify and encode visual landmarks despite illumination, viewpoint, and seasonal changes. Using publicly available aerial imagery, we demonstrate that our approach can detect and re-identify sparse landmarks across seasons and enable localization within 10 meters. Lastly, our method minimizes the storage requirement compared to current VTRN methods, expanding the navigable area size.**

I. Introduction

In absence of Global Navigation Satellite Systems (GNSS), uninhabited aerial vehicles (UAV) can pinpoint their exact geolocation by matching images from their navigational camera (NAVCAM) to known, georeferenced images, in a process known as visual terrain-relative navigation (VTRN). In the last few decades, image registration-based VTRN approaches have dominated GNSS-denied robotic navigation systems, driving applications like planetary entry, landing, and descent (EDL) and cruise missile guidance [1], [2]. These approaches typically rely on registration backends, powered by area-based template matching and/or feature-based homography estimation, to provide precise geolocation [2]. However, they face two problems: First, they fail when faced with seasonal or illumination variation, relying on strategic mission planning [1], [2] or deep learning to compensate [3]. Second, they require georeferenced imagery or extracted local feature descriptors to be stored on memory-constrained UAVs which limits navigation area size.

In contrast, landmark-based VTRN approaches are robust and lightweight, providing accurate but sparser geolocation updates by re-identifying a small set of known landmarks [4], [5], [6]. These landmarks can be encoded with invariances to common VTRN perturbations like seasonal variation and cached as low-dimensional vectors. For landmark-based approaches, the main challenge is choosing a set of landmarks that is large enough to provide a steady rate of geolocation updates, but with each landmark being easily re-identifiable.

Today, convolutional neural networks (CNN) have made it possible to easily detect and identify such landmarks despite appearance and illumination variations, but expert guidance is generally still used to select good landmarks for CNN training. Examples of this include crater detection for lunar EDL [4], and detection of various human-made structures (roads, houses, and buildings) for UAV navigation over urban/suburban areas [7], [8]. Although human expertise helps in these settings, there are three downsides: First, human cognitive and visual biases could result in potentially-useful landmarks being overlooked [9], [10]. Second, humans are not good at pattern recognition in unstructured and noisy terrain, whereas learning-based methods are good, when provided enough data. Third, having humans-in-the-loop means manual and tedious mission planning.

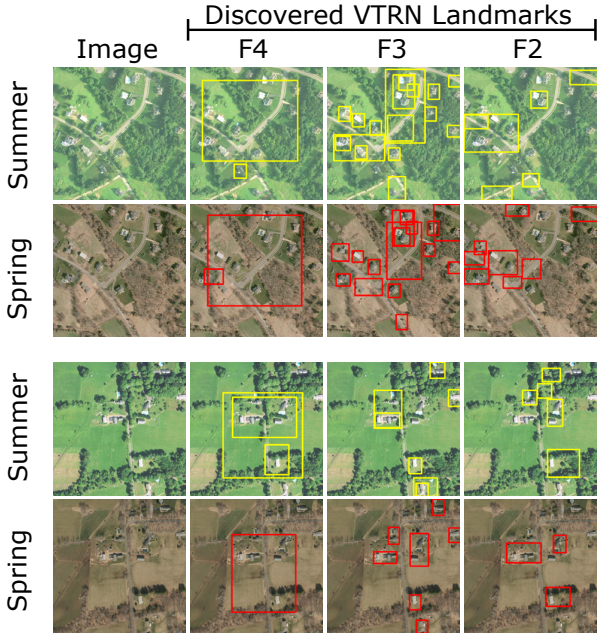

Fig. 1: Examples images from the dataset and proposed landmarks via our discovery module. F2/3/4 are resolution streams of the network activations that the landmarks were extracted from (F4 is lowest). Landmarks from each stream are displayed prior to non-maximum suppression and may overlap with those in other streams. Note that not all discovered landmarks are required to have a matching pair across seasons for localization to work.

In this work, we propose a self-supervised landmark-based VTRN pipeline for UAV localization across seasons. Our primary contribution is a landmark discovery algorithm that learns to automatically identify navigationally-useful and seasonally-robust landmarks (Fig. 1) without requiring human expertise. We investigate the localization potential and robustness of individual components and demonstrate their efficiency over current VTRN techniques.

This abstract is outlined as follows: We briefly go over prior works in Sec. II and follow with our approach (Sec. III), results (Sec. IV), and conclusion (Sec. V).

[1]C. Lee, E. Mesic, and S-J. Chung are with the Graduate Aerospace Laboratories, California Institute of Technology (Caltech), Pasadena, CA, USA. {clee, esmir, sjchung}@caltech.edu

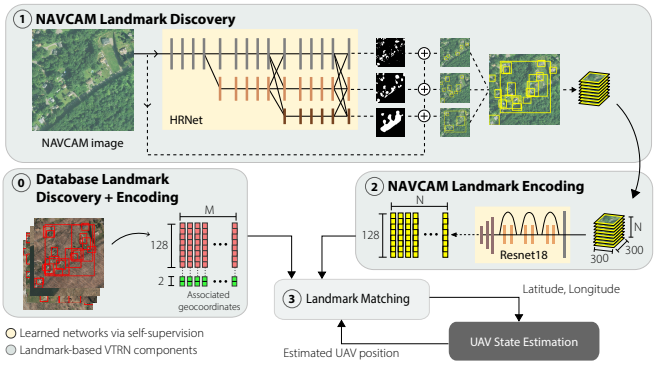

Fig. 2: Flowchart of our proposed landmark-based localization method in a UAV state estimation pipeline. The landmark discovery module (**Step 1**) extracts cropped landmark proposals based on the activations of a CNN. Landmark crops are encoded using a separate network (**Step 2**) and matched (**Step 3**) against a precomputed database of georeferenced landmark encodings (**Step 0**).

II. Relevant Work

Compared to local feature-based approaches, current landmark-based VTRN approaches tend to focus on landmarks with more semantic meaning, such as lunar craters, roads, and buildings [4], [7], [8]. They typically consist of three components: landmark detection, encoding, and matching. For example, [4] localizes lunar craters using a CNN and matches the geometric characteristics of found craters against a georeferenced crater database to get location. Recent works [7], [8] also leverage human-selected landmarks such as road networks and buildings for aerial navigation, but do not operate outside of urban environments. These current approaches reduce the storage overhead of local features but rely on humans to select landmark classes for CNN training. In our work, we seek to automate landmark discovery by directly learning from image data using a self-supervised learning scheme.

Recent VTRN approaches eschew landmarks and directly match globally encoded NAVCAM and database images. [11] trains a CNN autoencoder to densely encode database images along a 1.1 km flight path for fast location querying using a global NAVCAM descriptor during flight. Similarly, [12] discretizes database images of a small UAV flight area along a grid prior to encoding and perform pose refinement via learned local feature matching after reducing location uncertainty via global descriptor matching. [13] also uses global descriptor matching, but requires onboard storage and preprocessing of georeferenced database images before encoding to achieve illumination and viewpoint invariance. In our work, we use global descriptors with discovered landmarks to perform localization with a low storage overhead to enable navigation in larger areas.

III. Approach

An overview of our VTRN pipeline (Fig. 2) is as follows: Prior to flight, landmarks are discovered in georeferenced images using a CNN, encoded with another CNN and tagged with geocoordinates, and cached in an onboard landmark database. During flight, landmarks from NAVCAM images are detected, encoded, and queried against database encodings to find a match. UAV position is updated, and position uncertainty is used to restrict the database search space.

We give an overview of the self-supervised learning scheme we use for CNN training before detailing the landmark discovery, encoding, and matching components that comprise our proposed method.

A. Self-supervised Contrastive Learning

We use a self-supervised contrastive learning (SSCL) scheme similar to [14]. Our training procedure is as follows: for an image $\boldsymbol{x}$, we generate two views $\tilde{\boldsymbol{x}}_{\boldsymbol{i}}, \tilde{\boldsymbol{x}}_{\boldsymbol{j}}$ via random visual perturbations commonly encountered in flight. The views are encoded into 128-d vectors, $\boldsymbol{h}_{\boldsymbol{i}}, \boldsymbol{h}_{\boldsymbol{j}}$, via CNN encoder f. We maximize the cosine similarity between positive vector pairs (generated from the same $\boldsymbol{x}$) and minimize between negative pairs (sampled from within the batch). For a positive pair, the loss is formally defined

$$\mathcal{L}(\boldsymbol{h}_{\boldsymbol{i}}, \boldsymbol{h}_{\boldsymbol{j}}) = -\log \frac{\exp(S_c(\boldsymbol{h}_{\boldsymbol{i}}, \boldsymbol{h}_{\boldsymbol{j}})/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(S_c(\boldsymbol{h}_{\boldsymbol{i}}, \boldsymbol{h}_{\boldsymbol{k}})/\tau)} \quad (1)$$

where S_c denotes cosine similarity and τ is a temperature parameter set to 0.1.

B. Landmark Discovery, Encoder, and Matching

Landmark discovery: Our algorithm uses the activations of an HRNet CNN feature extractor [15] to find landmarks. The HRNet is followed by a global average pool and a fully-connected layer. To focus the network on invariant landmarks, we train this network (Sec. III-A) to predict if two image encodings describe the same location (positive) or not (negative). We create image pairs using random seasonal variations (leaf-on and leaf-off), rotation, perspective, color jitter, and motion blur augmentations.

To localize landmarks, we extract the final activations from the three lowest-resolution streams of the HRNet (Fig. 2), denoted F2, F3, F4 from highest to lowest resolution. Each activation is channel-wise averaged before upsampling to input size and binarized via thresholding. Thresholds are chosen based on percentile values computed over training set activations. Landmarks are localized via contour detection and fitted with a tight bounding box (Fig. 1). Overlapping landmarks with an intersection-over-union (IoU) ≥ 0.4 are non-maximum suppressed (NMS), with preference for landmarks extracted at higher thresholds.

Landmark encoder: We use a Resnet-18 [16] encoder to encode discovered landmarks for lightweight storage and matching. This network is trained on discovered landmark crops (300×300) using the same augmentations as before.

Landmark matching: Prior to flight, landmarks are detected over a target area, encoded, and cached with their associated geocoordinates. During flight, we match NAVCAM landmark encodings against database encodings within R meters of the current position estimate. A landmark pair is a match if its similarity is over a threshold and has

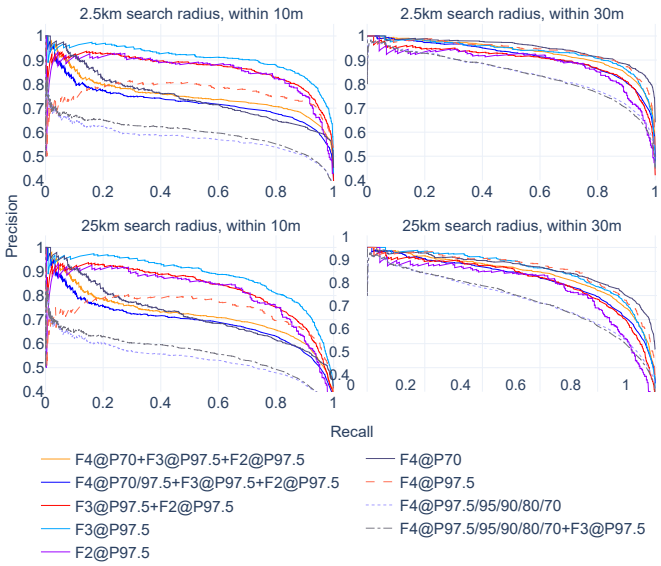

Fig. 3: Joint landmark discovery and encoding matching results. Matching was done with 2.5 km and 25 km search radii. Ground truth matches were counted if a landmark pair were within 10 m or 30 m of one another.

the maximum similarity over other possible pairs. R is hardcoded in this work, but we note that it could possibly be adapted based on current position uncertainties.

C. Implementation and Training Details

We implement networks in PyTorch using the `timm` library [17]. For landmark discovery, convolutions used reflection padding to avoid border effects in activation maps. We train for 1000 epochs, using a batch size of 128 and the Adam optimizer with a learning rate of $1e^{-4}$.

IV. Results

A. Datasets

We train and evaluate our method using the aerial image dataset from [3]. It consists of 3639 coregistered image pairs taken over the state of Connecticut (CT) in the United States during Spring and Summer 2016. Human-made structures, wooded forests, agricultural fields, and bodies of water are present, with "leaf-on vs. leaf-off" seasonal variation. We partition each 1270×1270 image into 600×600 crops with a 10 percent overlap and create train, val, and test splits at a 70:15:15 ratio. Each image has a resolution of 0.6m/pixel, resulting in 2112 km^2 of total landmass covered.

Seasonal effects like snow cover is not captured over this landmass and we leave explicit training and analysis of winter seasonal-invariance for future work. Also, we chose this setting with intuitive landmarks like buildings to easily validate our discovery algorithm, as the landmarks it proposes should overlap with those obvious to humans. In future work, we look to extend to less intuitive settings.

B. Localization Performance

Evaluation method: We evaluate matching performance of our VTRN pipeline via precision-recall analysis and use database search radii R of 2.5 and 25 km, simulating local (relatively lost) and global localization (completely lost) scenarios, respectively. Landmark pairs with maximum cosine similarities are considered as proposed matches in the evaluation. Distances between such landmark pairs are computed using the UTM coordinates at their bounding box centers and we consider ground truth matches to be within 10 and 30 m. Finally, similarity thresholds are applied to generate the precision-recall curves.

We test different configurations of resolution streams and percentile thresholds (Fig. 3) and find that small, highly salient landmarks (F2@P97.5, F3@P97.5) provide reliable position estimates within 10 m of ground truth, especially when paired with tighter search radii. Within 30 m of ground truth, using larger landmarks (F4@P70, F4@P97.5) yields best match rates. Aggregating landmarks from various resolution streams and thresholds does not achieve best performance but has the benefit of more landmarks for more location updates.

C. Ablation Studies

Landmark discovery: We quantify the number and size of the coinciding landmarks discovered using various resolution streams and percentile thresholds (Fig. 4a). In general, masking HRNet activations using high percentile thresholds (P97.5) increases the rate of landmark coincidence and favors small, sparse landmarks. Landmarks are generally easier to match when fewer but more salient landmarks are used, apart from very large landmarks (F4@P70).

Landmark encoding: Our landmark encoder outperforms other common image descriptors when faced with geometric and seasonal perturbations (Fig. 4b). We conduct precision-recall analysis by attempting to distinguish the encodings of an equal number of known matching and non-matching landmark pairs. We compare against ImageNet encodings (512-d) and VLAD [18] descriptors (2048-d). All encodings performed well with random geometrically-transformed landmarks from the same season, but only our encoding method was robust when seasons were varied in each pair, illustrating the benefit of explicitly training for such perturbation.

D. Computation Benchmarks

We benchmark landmark discovery and encoding using 600×600 NAVCAM images. Using a Nvidia Titan RTX and an Intel Core i9-7900X, images can be processed at 17 Hz. Benchmarks on an Nvidia Jetson AGX Orin, simulating small UAV use cases, sees slower rates of 8 Hz, due to slower processing during the CPU portions of the landmark localization step. We note that significant speed improvements can be made with smaller network architectures.

As our landmarks are sparse and low-dimensional, our method usually requires less onboard storage compared to techniques that densely encode a flight area [11], [12] or

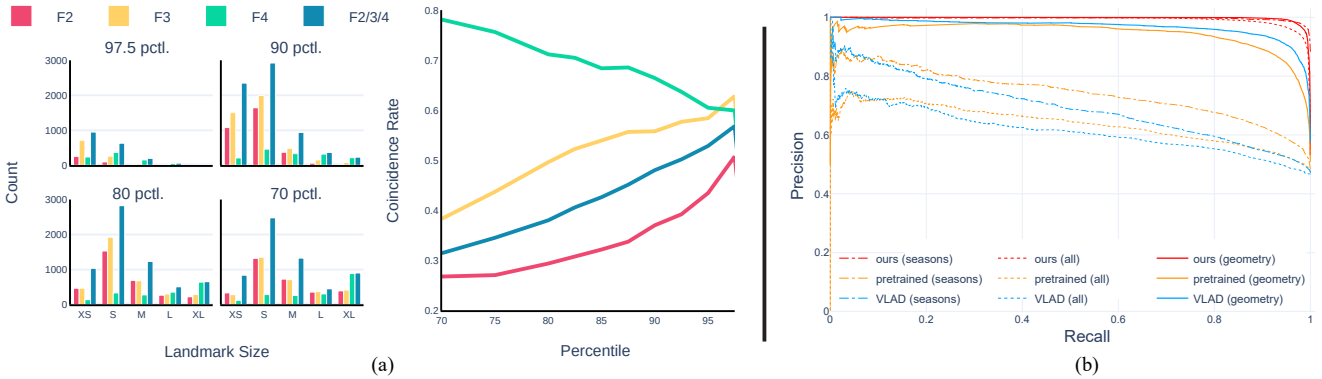

Fig. 4: Ablation on landmark discovery and encoding using the Connecticut test set. (a) Metrics of coinciding landmarks discovered at different resolution streams and various percentile threshold values. (b) Effect of common VTRN perturbations (*seasons only*, *geometry only*, *all*) on the encoding similarities of known matching and non-matching landmark pairs.

require onboard reference orthoimagery [1], [3], [8], [13]. For example, to localize over Salisbury, CT, which covers 155 km^2 of farmland, forests, and suburbs, 1.5 Gb is needed to store high-resolution, orthorectified reference images (assuming 0.6 m/pixel NAIP imagery) for methods that use database images during flight. Furthermore, an encoder-based method that lacks landmark detection like [11] would require roughly 19 Gb (extrapolated from their benchmarks). In contrast, our method requires between 90 to 200 Mb for the same landmass depending on configuration.

V. Conclusion

We presented the first landmark discovery algorithm for aerial VTRN. We showed that SSCL can find optimal landmarks for aerial navigation without human guidance and can consistently re-identify them across seasons. In conjunction with a seasonally-invariant CNN encoder, our discovery algorithm proposes landmarks that enable robust localization capabilities over large landmasses while demanding much less storage memory required by other methods. For future work, we aim to leverage our approach to better utilize sparse local features for more precise localization and pose estimation, integrate into a state estimation pipeline for UAV flight, and test in rugged mountainous and desert terrain where landmark selection is not as intuitive for humans.

Acknowledgment

This project was funded by the Boeing Company with R. K. Li as Boeing Project Manager. The authors thank R. K. Li and A. Tolstov of The Boeing Company for stimulating technical discussions.

References

[1] J. R. Carr and J. S. Sobek, "Digital scene matching area correlator (dsmac)," in *Image Processing For Missile Guidance*, vol. 238. International Society for Optics and Photonics, 1980, pp. 36–41.

[2] A. E. Johnson, S. B. Aaron, H. Ansari, C. Bergh, H. Bourdu, J. Butler, J. Chang, R. Cheng, Y. Cheng, K. Clark *et al.*, "Mars 2020 lander vision system flight performance," in *AIAA SciTech 2022 Forum*, 2022, p. 1214.

[3] A. T. Fragoso, C. T. Lee, A. S. McCoy, and S.-J. Chung, "A seasonally invariant deep transform for visual terrain-relative navigation," *Science Robotics*, vol. 6, no. 55, p. eabf3320, 2021. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.abf3320

[4] L. Downes, T. J. Steiner, and J. P. How, "Deep learning crater detection for lunar terrain relative navigation," in *AIAA SciTech 2020 Forum*, 2020, p. 1838.

[5] L. Matthies, S. Daftry, S. Tepsuporn, Y. Cheng, D. Atha, R. M. Swan, S. Ravichandar, and M. Ono, "Lunar rover localization using craters as landmarks," in *2022 IEEE Aerospace Conference (AERO)*. IEEE, 2022, pp. 1–17.

[6] J. Vander Hook, R. Schwartz, K. Ebadi, K. Coble, and C. Padgett, "Topographical landmarks for ground-level terrain relative navigation on mars," in *2022 IEEE Aerospace Conference (AERO)*. IEEE, 2022, pp. 1–6.

[7] T. Wang, Y. Zhao, J. Wang, A. K. Somani, and C. Sun, "Attention-based road registration for gps-denied uas navigation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 4, pp. 1788–1800, 2021.

[8] 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 Conf. Comput. Vis. Pattern Recog. workshops*, 2018, pp. 1513–1523.

[9] M. Kwon, E. Biyik, A. Talati, K. Bhasin, D. P. Losey, and D. Sadigh, "When humans aren't optimal: Robots that collaborate with risk-aware humans," in *2020 15th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2020, pp. 43–52.

[10] A. M. Hussain Ismail, J. A. Solomon, M. Hansard, and I. Mareschal, "A perceptual bias for man-made objects in humans," *Proceedings of the Royal Society B*, vol. 286, no. 1914, p. 20191492, 2019.

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

[12] S. Chen, X. Wu, M. W. Mueller, and K. Sreenath, "Real-time geo-localization using satellite imagery and topography for unmanned aerial vehicles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, 2021, pp. 2275–2281.

[13] P. Yin, I. Cisneros, S. Zhao, J. Zhang, H. Choset, and S. Scherer, "isimloc: Visual global localization for previously unseen environments with simulated images," *IEEE Trans. Robot.*, pp. 1–17, 2023.

[14] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Int. Conf. Mach. Learning*. PMLR, 2020, pp. 1597–1607.

[15] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, 2020.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.

[17] R. Wightman, "Pytorch image models," https://github.com/rwightman/pytorch-image-models, 2019.

[18] R. Arandjelovic and A. Zisserman, "All about vlad," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 1578–1585.