iSimLoc: Visual Global Localization for Previously Unseen Environments With Simulated Images

Peng Yin, Member, IEEE, Ivan Cisneros, Shiqi Zhao, Ji Zhang, Howie Choset, Fellow, IEEE, and Sebastian Scherer, Senior Member, IEEE

Abstract—The camera is an attractive device for use in beyond visual line of sight drone operation since cameras are low in size, weight, power, and cost. However, state-of-the-art visual localization algorithms have trouble matching visual data that have significantly different appearances due to changes in illumination or viewpoint. This article presents iSimLoc, a learning-based global relocalization approach that is robust to appearance and viewpoint differences. The features learned by iSimLoc’s place recognition network can be utilized to match query images to reference images of a different stylistic domain and viewpoint. In addition, our hierarchical global relocalization module searches in a coarse-to-fine manner, allowing iSimLoc to perform fast and accurate pose estimation. We evaluate our method on a dataset with appearance variations and a dataset that focuses on demonstrating large-scale matching over a long flight over complex terrain. iSimLoc achieves 88.7% and 83.8% successful retrieval rates on our two datasets, with 1.5 s inference time, compared to 45.8% and 39.7% using the next best method. These results demonstrate robust localization in a range of environments and conditions.

Index Terms—Aerial visual terrain navigation, GPS denied localization, hierarchical global relocalization, sim-to-real.

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) have become popular in different nonmilitary and commercial applications such as cargo transport [1], surveillance [2], precision agriculture [3], and search–rescue tasks [4], [5]. Current UAVs primarily rely on GPS as their only source for global position information, making their localization systems fragile to GPS outages, an issue which is often addressed with safety pilots [6]. However, in the future, beyond visual line of sight flight will require a backup source of global position to achieve sufficient reliability.

To this end, the visual terrain-relative navigation (VTRN) [7] method can be viewed as an alternate method for GPS redundancy. In general cameras are more attractive for UAVs, since cameras are passive, able to run over long distances, have a low SWaP-C (size, weight, power, and cost), and have a large field of view (FOV). As shown in Fig. 1, challenges of VTRN for localization include the following.

1) Appearance Changes: Appearances of a particular area may change drastically under different illumination and weather conditions, making data association challenging.

2) Viewpoint Differences: When revisiting the same place, the UAV cannot guarantee that it will revisit the exact same position, orientation, and altitude, which requires the method to be robust to variations.

3) High-Similarity: In large-scale UAV navigation, there is a high likelihood of areas with repeated and homogeneous geometries, such as forests and flat ground, leading to false data associations.

We propose a novel learning-based VTRN approach, iSimLoc, which provides robust global re-localization in large-scale and urban environments. iSimLoc trains a place recognition network by matching simulated images to real images, thus enabling it to perform localization in previously unseen environments.
model for nonvisited environments by leveraging “overhead imagery.” We define overhead imagery as any orthorectified North-aligned imagery of some region of the Earth’s surface; in our case, we use either Earth-observing satellite imagery or imagery collected by high-altitude aircraft. As it is based on our previous work [8], [9], iSimLoc is also invariant to appearance changes caused by illumination and viewpoint differences. When relocalizing within a large area, iSimLoc relies on a coarse-to-fine localization method to balance efficiency and accuracy. The contributions of iSimLoc are as follows:

1) **Real-to-Sim Conditional-Domain Transfer: A Conditional Domain Transfer Module (CDTM) to transform the raw images into a constant simulated domain.** The CDTM extracts both geometric and conditional features from raw images for conditional invariant place recognition. Our experimental results on urban and terrain areas demonstrate that the CDTM improves recognition ability for nonvisited areas.

2) **Viewpoint-Invariant Place Recognition: Most VTRN systems depend on fixed viewpoints between test/reference queries.** iSimLoc, on the other hand, calculates viewpoint-invariant descriptors and estimates relative orientation through usage of the Pose Estimation Module (PEM). Specifically, the PEM utilizes spherical harmonic features, which are orientation-equivariant for the same location.

3) **Hierarchical Localization System:** For global relocalization, iSimLoc matches hierarchically starting with a coarse estimate at a high altitude and refining with repeated cropping of images to make more accurate estimates. This method balances efficiency and accuracy, which is essential for relocalization in large maps.

In our experiment results, we present an extensive evaluation of our system on two unique datasets: 1) The **CMU Campus** dataset consists of 160 trajectories taken under different lighting conditions targeting an urban environment, which was collected using a quadcopter flying on Carnegie Mellon University’s campus; 2) and the Aerial-view Large-scale Terrain-Oriented (ALTO) dataset [10] consists of one 150 km trajectory covering both urban and natural terrain, which was collected using a helicopter. On both datasets, iSimLoc outperforms all other relative place recognition methods and achieves a 88.7% successful recognition rate for urban areas and 83.8% for natural terrain. Leveraging overhead images helps our system attain higher generalization ability for unseen environments. We include a discussion section and a conclusion section to analyze the advantages and shortcomings of the current iSimLoc approach, as well as to examine potential future work.

II. RELATED WORKS

Visual geo-localization is defined as finding geographic coordinates (and possibly camera orientation) for a given query image. Based on survey [11], visual geo-localization is divided into city-scale and natural localization. The main challenges are environmental condition changes, viewpoint differences, and large-scale relocalization.

To deal with changes in environmental conditions, Mishkin et al. [12] modified BoW approach [13] with multiple descriptors and adaptive thresholds to better cope with large-scale changes in environments. Bhavit et al. [14] introduced a visual terrain navigation method where reference images are rendered from Google Earth satellite images. However, because such overhead images from Google Earth are captured several years prior to test time, visual differences between reference/testing streams reduce localization accuracy. To deal with this issue, Mollie et al. [15] utilized an Autoencoder network to transfer raw images into overhead images, ignoring local dynamic differences or environmental condition changes. However, this method cannot handle illumination, weather, and seasonal changes, reducing generalization ability for unseen conditions. In their recent work [7], Anthony et al. provided a seasonally invariant deep transform neural network to convert seasonal images into a stable and invariant domain for visual terrain navigation. This method targets high-altitude flying modes as depicted in Fig. 1, where there exist rich unchanging geometric features that persist even in different seasons. However, in lower altitudes, this method’s transfer-ability will be negatively affected by occlusions of 3-D objects and highly variable lighting conditions from different times of the day. Another solution to deal with disturbances from environmental conditions is to match horizontal lines extracted from a query image against those rendered from digital elevation models [16]. Baatz et al. [17] demonstrated terrain localization by leveraging this method. Similarly, Bertil et al. [18] introduced an accurate camera localization for unmanned surface vessels by aligning horizontal lines with coastal structures. A significant drawback of horizontal line-based approaches is dependence on rich 3-D geometric structures, such as mountainous or coastal areas. Thus, performance will be reduced in homogeneous and plain environments. Similar to [7], our iSimLoc method transfers raw images into a constant domain.

Viewpoint difference poses another significant challenge for accurate localization. As depicted in Fig. 1, orientation and altitude differences can significantly change the appearance of a location from the original perspective. Fragoso et al. [7] and Baatz et al. [17] conducted image alignment over very high altitude flights (7, 000 ~ 15, 000 m). In such cases, top-down images are able to capture rich distinguishable geometric features for accurate matching. However, not all applications can leverage this benefit since the Federal Aviation Administration sets the flying altitude limit for UAV drones to around 120 m. Helicopters usually fly at ≥ 300 m when at lower altitudes, encountering changing viewpoints and environmental conditions. Additionally, current visual terrain navigation methods usually assume that relative orientations between raw inputs of drones/helicopter and reference images are small. Thus, these methods focus on position estimation and do not include robust relative orientation estimation. In real applications, a noisy GPS signal may result in a sizeable initial orientation estimation error, resulting in image alignment failures and, consequently, a loss of accuracy. Yet, most image-based alignment methods are unable to estimate corresponding orientations [7], [14], [15]. Jeong-Kyun et al. [19] introduced a vanishing point-based camera orientation estimation method. This method is suitable
for urban indoor and outdoor scenes, as detection of vanishing points is based on line segments. Shichao et al. [25] proposed CubeSLAM for camera pose estimation and localization based on extracted cubic objects. Both line features and cubic objects can be relatively easily captured in urban environments; however, they are sparsely present in natural terrain. Baatz et al. [16] used semantic information (tree, river) as constraints for camera pose estimation. However, a significant limitation comes from unreliability of detecting semantic objects. When using visual terrain navigation at lower altitudes, image distortion caused by viewpoint changes will further reduce feature stability. To extract viewpoint-invariant descriptors in both low and high altitudes—similar to our previous 3-D place recognition work [9], [26]—iSimLoc utilizes spherical harmonics [27] to extract orientation-equivariant features from spherical perspectives. iSimLoc also estimates the relative orientation between test and reference based on constant amplitude of spherical harmonics. This ability further improves online localization robustness even for long-term visual terrain navigation.

Most existing VTRN methods focus on local relocalization against a reference image and share the assumption that the robot has relatively good estimates of its position and orientation. However, in real-world applications, environmental conditions and viewpoints (both orientation and altitude) may change dramatically and simultaneously. In addition, similar and repeated geometric features in natural terrain environments, such as in forests and flat plain ground, will further reduce localization success rates. Most VTRN methods can, thus, hardly deal with global relocalization in large-scale environments without a great deal of assistance from GPS.

In Table I, we compare different properties of current VTRN methods. Domain transfer-based methods are mainly designed to deal with changing environmental conditions. Notably different to [7], they ignore viewpoint differences, making them most suitable for local relocalization, whereas iSimLoc also includes viewpoint-invariant feature extraction to handle viewpoint differences. On the other hand, most geometric-based methods mainly target changing viewpoints under constant environmental assumptions. Recently, Cai et al. [23] provided a ground-to-aerial localization method based on the extracted visual features between omnidirectional image and the google satellite image, and Liu et al. [24] also used a similar pipeline with spherical perspectives. However, few methods consider both conditional and viewpoint differences in visual terrain localization at the same time. In contrast to our previous work i3dLoc [9] on Robotics: Science and Systems 2021, where we use a place descriptor for local relocalization, iSimLoc aims to provide robust visual global relocalization for large-scale environments by leveraging overhead imagery.

### III. System Overview

iSimLoc uses publicly available orthorectified North-aligned aerial imagery and omnidirectional camera imagery from a UAV as input to provide condition- and viewpoint-invariant global relocalization in large-scale terrain/urban environments. The iSimLoc framework consists of the following three steps.

1. **Data Collection** to provide paired simulation/raw data to train the CDTM.
2. **Place Feature Extraction** for viewpoint-invariant place recognition.
3. **Global Relocalization** for hierarchical robust localization.

#### A. Data Collection

We use two data collection platforms to generate paired simulated and real-world data for training and evaluation. The first

---

**TABLE I**

PROPERTIES OF DIFFERENT VTRN METHODS

| Method            | Class   | Localization Accuracy | Varying Environments | Varying Conditions | Varying Viewpoints | Varying Altitudes | Global Localization |
|-------------------|---------|-----------------------|----------------------|-------------------|-------------------|-------------------|---------------------|
| Baatz et al. [17] | Nonlearning | N/A         | ✗                   | ✗                 | ✗                 | ✗                 | ✗                   |
| Jeong-Kyun et al. [19] | Nonlearning | N/A         | ✗                   | ✓                 | ✓                 | ✓                 | ✓                   |
| Baatz et al. [16] | Nonlearning | N/A         | ✗                   | ✗                 | ✗                 | ✗                 | ✓                   |
| Fluckler et al. [20] | Nonlearning | N/A         | ✓                   | ✓                 | ✗                 | ✗                 | ✓                   |
| Michael et al. [21] | Nonlearning | N/A         | ✓                   | ✓                 | ✓                 | ✓                 | ✓                   |
| Anthony et al. [7] | Learning  | ≤ 50 m          | ✗                   | ✓                 | ✗                 | ✗                 | ✗                   |
| Patel et al. [14]  | Learning  | ≤ 10 m          | ✓                   | ✓                 | ✗                 | ✗                 | ✗                   |
| Bianchi et al. [15] | Learning  | ≤ 10 m          | ✓                   | ✓                 | ✗                 | ✗                 | ✗                   |
| Arandjelovic et al. [22] | Learning  | N/A          | ✗                   | ✓                 | ✓                 | ✓                 | ✗                   |
| Cai Sudong  et al. [23]  | Learning  | N/A          | ✗                   | ✗                 | ✓                 | ✓                 | ✓                   |
| Liu Liu et al. [24]  | Learning  | ≤ 5 m          | ✗                   | ✓                 | ✓                 | ✓                 | ✓                   |
| iSimLoc (ours)     | Learning  | ≤ 20 m          | ✓                   | ✓                 | ✓                 | ✓                 | ✓                   |
dataset is recorded under changing conditions and at different altitudes around the Carnegie Mellon University campus. The recording platform is a quadcopter with a mounted omnidirectional camera. We name it “CMU Campus” dataset. The second dataset (“ALTO”) is a flight from Cambridge, Ohio, to Pittsburgh, Pennsylvania using a helicopter with a downward-facing pinhole camera.

For the CMU Campus dataset, we use the waypoint following mode to make repeated passes along fixed trajectories to collect data on the same path but under different conditions (illumination, weather, time, etc.). ALTO dataset includes 150 km trajectory. We generate a paired sim-to-real dataset for both platforms by exporting the trajectories’ GPS information and collecting publicly available nadir overhead imagery from Google Earth.

Due to the low frequency of the GPS data, we interpolate data among two GPS points to generate time-synced sim-to-real paired images. And please note here, the sync images are not the hard requirements in our system, otherwise, it can be the limitation of our method. In Section V, we expand further on the details of our data generation.

B. Feature Extraction

Due to condition and viewpoint differences, extracting invariant place descriptors is the most critical factor in visual localization. In iSimLoc, we first use a conditional domain transfer learning module (CDTM) to convert raw images into a constant geometric domain. As depicted in Fig. 3, the CDTM is forced

Fig. 2. iSimLoc system framework. For high and low altitudes, iSimLoc extracts a condition-(illumination) and viewpoint-invariant place descriptor. Only the descriptor needs to be stored and matched. Larger FOV perspectives help iSimLoc to provide an initial guess, while narrower FOV perspectives provide rich local geometric features for accurate localization. iSimLoc matches in a hierarchical manner, which enables us to balance search efficiency and accuracy.

Fig. 3. The network structure of iSimLoc. iSimLoc consists of a CDTM to transform raw visual inputs (X) into images of a different domain (e.g., simulation environment) (Ŷ), where environmental conditions (Z_CM) of the target domain are applied to improve generalization ability for unseen environments, and a PEM to simultaneously estimate viewpoint-invariant place features through the PEM loss and estimate the relative orientations through correlation estimation. Similarly colored networks indicate that these networks share weights (e.g., all orange Encoders share the same weights).
to extract conditional and geometric features with an orthogonal relationship distribution. With extracted geometric features, iSimLoc learns viewpoint invariant place descriptors with the Pose Extraction Module (PEM). The PEM can estimate place descriptors’ similarities while ignoring their orientation differences, and predict the relative yaw difference given matched descriptors for a particular area. Since spherical viewpoints will reduce geometric differences under varying altitudes, the PEM is also robust to local altitude differences. Due to these invariance properties, the iSimLoc place feature model is able to recognize places under different viewpoints, which increases sampling efficiency for global relocalization.

C. Global Relocalization

Without GPS assistance, current VTRN methods may quickly encounter tracking failures in repeated and homogeneous terrain areas, especially for long-term navigation tasks. Our global relocalization procedure uses extracted iSimLoc place features and a particle filter [28] in a coarse-to-fine hierarchical refinement method to overcome the limitations of other methods. Higher altitude images provide a coarse position estimation over a large search area, reducing the number of initial particles needed. At lower altitudes, visual inputs capture more geometric structures, which produces more accurate final matches. We visualize this process in Fig 2. Since features are orientation invariant, particles only need to be sampled within Euclidean space $\mathbb{R}^3$, instead of $SO(3)$ space, significantly reducing the total number of particles used.

IV. OUR APPROACH

As we stated in Section I, the UAV localization system is expected to deal with appearance differences between the raw image inputs and overhead images from google satellite, invariant to the difference of the local viewpoints, and provide robust global localization under large-scale without initials. Fig. 3 shows the system structure of iSimLoc, which mainly contains the following three modules.

1) A CDTM to convert raw images into a constant overhead image domain.

2) A PEM to recognize the place and orientation based on estimated features’ similarities.

3) A hierarchical global relocalization module to track position at both low and high altitudes.

In this section, we will investigate each individual module.

A. Conditional Domain Transfer Module

To generate constant geometric features from visual inputs under different environmental conditions, we construct a CDTM, which includes a feature encoder, a conditional decoder, and a discriminator. Our CDTM is inspired by style transfer networks such as CycleGAN [29], which teach a network to embed in its weights the stylistic features of the target domain and thus are able to convert an image from a source domain (e.g., horses) to the target domain (e.g., zebras). Our application, though, requires translating from several source style domains (e.g., different lighting conditions at different times of day) into a single target style domain (e.g., the satellite image style, or the simulation environment style), thus using a fixed network-like CycleGAN would require training several networks, one for each source–target mapping. Our CDTM, on the other hand, is mapping-agnostic; rather than learning the specific source-target style mapping of the training data, the CDTM learns to separate style from the content of the imagery (which we refer to as “conditional” features and “geometric” features, respectively).

In this way, we only need to train a single network, and we can reuse this network to translate many different source–target style mappings; we need only to include the target style encoding ($Z_{Cv}$) to condition the style translation to a specific style (the domain of image $Y$).

Raw visual images include both geometric features $Z_G$, which depend on 3-D geometric structures, and conditional features $Z_C$, which encode an image’s stylistic appearance that is caused by the combination of environmental conditions (illumination, etc). As depicted in Fig. 3, the CDTM encoder outputs a tensor of size $N \times M \times C$, which we split evenly (channel-wise) into the two tensors $Z_{G}$ and $Z_{C}$. At the start of the training process, there is no real characteristic difference between the two named tensors, but due to the loss functions that we describe below the encoder is trained to encode separate characteristics of the image into these two portions. When fully trained, we can separate the $Z_{GX}$ and $Z_{CX}$ encoding of an input image $X$, and substitute a $Z_{CV}$ encoding from a target image $Y$, so that when the encoding $Z_{CV}$ and $Z_{GX}$ are concatenated and decoded back into the generated image $Y$, it will have the style characteristics of the target image domain $Z_{CV}$. At inference time, we only need one encoder and one decoder, as well as a stored target domain tensor $Z_{CV}$, in order to condition the style of the input images $X$ to the target domain.

Before introducing the feature extraction process, we analyze the relationship of information entropy to viewpoints. Given an image $X$, $H(Z_G|Z_C|X)$ and $I(Z_G;Z_C|X)$ represent joint entropy and mutual entropy. $H(Z_G|Z_C,X)$ and $H(Z_C|Z_G,X)$ are conditional entropies based on features $Z_C$ and $Z_G$, respectively. The target of place recognition is to extract condition-invariant geometric features, related to conditional information $H(Z_G|Z_C,X)$; and restrict extracted feature’s uncertainty given the same image $X$, which is relative to joint entropy $H(Z_G,Z_C|X)$. To learn condition-invariant place descriptors, we focus on the following.

1) Increasing conditional entropy $H(Z_G|Z_C,X)$ to enrich geometric features $Z_G$ extracted from raw image $X$, which is independent of conditional feature $Z_C$, and

2) Reducing joint entropy $H(Z_G,Z_C|X)$, which measures joint features’ ($\{Z_G;Z_C\}$) differences when revisiting the same place.

Directly improving conditional entropy $H(Z_G|Z_C,X)$ in place recognition tasks is intractable since each unique area may drastically vary in appearance due to different combinations of environmental conditions (illumination, weather, seasons), and it is hard to access all potential $Z_G \rightarrow Z_C$ pairs. Alternatively,
using the information theory perspective, \( H(ZG|ZC, X) \) can be converted into
\[
H(ZG|ZC, X) = H(ZG|X) - I(ZG; ZC|X) \tag{1}
\]
where \( H(ZG|X) \) measures the diversity of geometric features \( ZG \) based on observation \( X \). This is important because, in place recognition tasks, higher variance in place features provides better distinguishability. We design a generative adversarial network (GAN) \cite{30} to enhance the diversity of \( H(ZG|X) \); for this, we used the standard GAN loss metric \( \mathcal{L}_{GAN} \) defined as
\[
\mathcal{L}_{GAN} = \min_{\theta, \phi} \max_{\beta} \mathbb{E}(\log(D_\beta(Y))) + \mathbb{E}(\log(1 - D_\beta(\tilde{Y}))) \tag{2}
\]
where \( Y \) and \( \tilde{Y} \) are real and generated overhead images, respectively, \( D_\beta \) is the discriminator to distinguish \( Y \) and \( \tilde{Y} \), \( P_\theta \) is the encoder, and \( Q_\phi \) is the decoder, and \( \beta, \theta, \phi \) are the learnable parameters. As shown by Goodfellow et al. \cite{30}, with iterative updating of the generative network (decoder \( Q_\phi \)) and discriminator modules, an adversarial network is able to push distribution of \( \tilde{Y} \) toward target distribution \( Y \). On the other hand, since mutual entropy \( I(ZG; ZC|X) \) measures overlaps between geometric features \( ZG \) and condition features \( ZC \), reducing \( I(ZG; ZC|X) \) indicates the minimum projection from \( ZC \) onto \( ZG \). We apply an orthogonal loss metric \( \mathcal{L}_{Orth} \) to enhance the orthogonal relationship between features \( ZG \) and \( ZC \)
\[
\mathcal{L}_{Orth} = \left( \frac{ZG \cdot ZC}{\|ZG\|_2 \cdot \|ZC\|_2} \right)^2. \tag{3}
\]
Through the combination of (2) and (3), we can indirectly increase the entropy \( H(ZG|X) \) and reduce the mutual entropy \( I(ZG; ZC|X) \), then increase the conditional entropy \( H(ZG|ZC, X) \) as mentioned in (1). For more details, please refer to our previous work \cite{9}.

To reduce joint entropy \( H(ZC, ZG|X) \), similar to CycleGAN \cite{29}, we construct an L1 loss metric between raw image \( X \) and reconstructed image \( \hat{X} \) as demonstrated in Fig. 3. Using raw image \( X \), iSimLoc generates overhead image \( \hat{Y} \) with geometric feature \( ZG_{\hat{Y}} \) and condition feature \( ZC_{\hat{Y}} \); then using \( \hat{Y} \), iSimLoc reconstructs \( \hat{X} \) with \( ZG_{\hat{X}} \) and \( ZC_{\hat{X}} \). In our previous work \cite{9}, we prove that decreasing \( H(ZG, ZC|X) \) corresponds to reducing image reconstruction uncertainty given sample data \( X \). We formulate the reconstruction loss as
\[
\mathcal{L}_{Recon} = H(ZG_{\hat{X}}, ZC_{\hat{X}}|P_\theta (X), ZG_{\hat{Y}}, ZC_{\hat{Y}}|P_\theta (\hat{Y})) - \log(Q_\phi(ZG_{\hat{X}}, ZC_{\hat{X}}|X)) \tag{4}
\]
The original \( H(ZG, ZC|X) \) is transformed into its upper bound \( \mathcal{L}_{Recon}(\hat{X}, \hat{X}) \). Based on (2), (3), and (4), we construct the loss metric for the CDTM as
\[
\mathcal{L}_{CDTM} = \mathcal{L}_{GAN} + \mathcal{L}_{Orth} + \mathcal{L}_{Recon}. \tag{5}
\]

**B. Pose Estimation Module**

In iSimLoc, the PEM is designed to predict viewpoint-invariant place descriptors and extract relative orientations. The PEM module is combined with a pretrained ResNet18 model from torchvision\(^1\) for deep feature extraction and a spherical feature-based viewpoint-invariant descriptor extractor and orientation estimator.

1) **Viewpoint-Invariant Descriptor**: Because of the greater FOV provided by spherical (omnidirectional) imagery compared with pinhole camera imagery, it seems natural to use spherical imagery for viewpoint-invariant place recognition. However, traditional convolutional neural networks are not well suited to use with spherical images, since angular resolution is not uniform across these types of images. Instead of a traditional convolution, we apply the spherical convolution, which utilizes spherical harmonics present in spherical projection images. Spherical convolutions avoid space-varying distortions in Euclidean space by convolving spherical signals in the harmonic domain. The mathematical model of the spherical convolution into harmonic domain shows that it is orientation-equivariant. Spherical convolution of \( SO(3) \) signals \( f \) and \( h \) are functions: \( SO(3) \to \mathbb{R}^K \) in rotation group \( SO(3) \) is defined as
\[
[f \ast_{SO(3)} h](R) = \int_{SO(3)} f(R^{-1}Q)h(Q)dQ \tag{6}
\]
where \( R, Q \in SO(3) \). As the proof in \cite{31} shows, spherical convolutions are orientation-equivariant
\[
[f \ast_{SO(3)} L_Q h](R) = [L_Q[f \ast_{SO(3)} h]](R) \tag{7}
\]
where \( L_Q(Q \in SO(3)) \) is a rotation operator for spherical signals. Convolution of two spherical signals \( f \) and \( h \) in the harmonics domain is computed in three steps: We first expand \( f \) and \( h \) to their spherical harmonic basis \( H_f \) and \( H_h \), then compute the point-wise product of harmonic coefficients, and finally invert the spherical harmonic expansion; we depict this in Fig 4. In our previous works \cite{8}, \cite{9}, we have utilized this approach to extract orientation-invariant place descriptors for either 3-D point clouds or spherical images. For more details on spherical harmonic properties, we suggest the reader refers to the original work in \cite{27}. To leverage viewpoint-invariant feature extraction, we utilize VLAD networks \cite{22} to cluster local orientation-equivariant features into global place descriptors.

\(^1\)Online. Available: https://pytorch.org/vision/stable/models.html
With the assistance of the CDTM module, iSimLoc is able to learn condition- and viewpoint- invariant place descriptors.

2) Orientation Estimation: As shown in the PEM module of Fig. 3, given extracted spherical harmonic features from two relative spherical images, we apply an estimation module to directly obtain relative orientations. As shown in [32], the spherical correlation $\bar{C}$ between two spherical signals $f_1$ and $f_2$ is their inner product $\bar{C} = \langle f_1, f_2 \rangle$. If $f_2$ is the rotated version of $f_1$, relative orientation $r \in SO(3)$ can be estimated by maximizing $\bar{C}$

$$\text{argmax}_{r \in SO(3)} \langle f_1, r^{-1}f_2 \rangle.$$  

(8)

Based on the orthogonal property of the spherical harmonics and the magnitudes property of harmonic signals [33], the above equation can be evaluated using the spherical Fourier coefficients. Since this part is beyond the scope of this article, please refer to [32] for a more detailed derivation.

3) Pose Loss Metric: To enable end-to-end training for viewpoint-invariant place feature extraction, we apply individual triplet-loss to both the raw image domain and the simulation (i.e., overhead) image domain separately, and also a cross triplet-loss between the two domains. To illustrate the loss functions, we first describe a few necessary definitions. In both real-world and simulated image domains, for each query image $M_k$, we provide the following tuples $T_k = \{S_k^i, \{S^\text{rot}_{i}^j\}_k, \{S^\text{pos}_i\}_k, \{S^\text{neg}_i\}_k\}$. $S_k$ is encoded spherical place descriptors from $M_k$, $\{S^\text{rot}_{i}^j\}_k$ is a set of descriptors from the same position of $M_k$ but for different orientations. As was done in relative place recognition work [22], we also provide positive $\{S^\text{pos}_i\}_k$ and negative $\{S^\text{neg}_i\}_k$ features based on distance to $M_k$. We construct paired tuples within raw image domain $S^V$ and simulation image domain $S^S$. Ideally, for each scene, the place descriptor should be invariant to orientation and sensitive to translation differences, thus we design the following loss function:

$$L_{\text{Individual}}(T_k) = \max_{i \in \{S^\text{pos}_i\}_k, j \in \{S^\text{neg}_i\}_k} (|\lambda_1 + d(S_k^i, S^\text{pos}_i) - d(S_k^i, S^\text{neg}_i)|)_+ + \max_{i,j,k} (|\lambda_2 + d(S^\text{rot}_{i}^j, S^\text{pos}_i) - d(S^\text{rot}_{i}^j, S^\text{neg}_i)|)_+$$  

(9)

where $d(\cdot)$ denotes Euclidean distance, and $(\cdot)_+$ denotes the hinge loss, $\lambda_1$ and $\lambda_2$ are constant thresholds to control the margin between feature differences of different Euclidean distances. This loss function applies to both $S^V$ and $S^S$ domains. We also define a domain learning metric to reduce cross-domain feature differences

$$L_{\text{CrossDomain}}(T_k) = \max_{i \in \{S^\text{pos}_i\}_k, j \in \{S^\text{neg}_i\}_k} (|\lambda_3 + d(S^V_k, S^S^\text{pos}_i) - d(S^V_k, S^S^\text{neg}_i)|)_+ .$$  

(10)

where $\lambda_3$ is constant threshold to control margin between raw and simulated image features. In both of the above equations, variables $S$ without a superscript denote images from the same domain. By combining both individual metrics and cross domain metric, we obtain the following loss function for the PEM:

$$L_{\text{PEM}} = L_{\text{Individual}}^V + L_{\text{Individual}}^S + L_{\text{CrossDomain}}.$$  

(11)

In our application, $\lambda_1$, $\lambda_2$ are set to 0.5 and $\lambda_3$ is set to 1.0. During the training procedure, we first train the domain transfer module with paired real-world and simulated images; then we use the pretrained transfer model to produce conditional- and viewpoint-invariant place descriptors for use in training the rest of the network.

C. Hierarchical Localization

Visual ambiguity is unavoidable during high-altitude flying in large-scale terrain relative navigation. Therefore, we devise a hierarchical localization module that helps iSimLoc achieve robust localization with a particle-filter-based coarse-to-fine searching strategy. Each particle represents a spherical image generated from the google satellite\(^2\) with fixed position and altitude. In contrast with place features used in other works [14], [15], our place descriptor is symmetric to viewpoint differences, thus each particle uses our extracted place descriptor as the representation of one potential area.

As shown in Fig. 5, given one test image at full resolution, we generate spherical projections (256 × 256) at different altitudes. Higher altitude images capture a larger context for coarse global localization, which will reduce the number of initial particles. Given a potential search area of size $[M_1 \times M_2]m^2$, the potential FOV of each particle is based on altitude $H$. We define the active search radius $r$ as $r = H \tan 45^\circ$ (we assume a camera FOV angle of 90°). At the lowest resolution level, particles are sampled uniformly on a reference overhead image, and we define a ratio $R_{\text{olp}} \in [0, 1]$ to control the overlaps between two sampled areas. The initial number of particles $P_{\text{init}}$ is

\(^2\)[Online]. Available: https://github.com/zhengjie9510/google-map-downloader

\[\text{Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.}\]
Hierarchical Localization.

Algorithm 1: Hierarchical Localization.

Input: Extracted observation $Y$, Potential searching area $[M_1, M_2]$, initial altitude $H$ and overlap ratio $R_{olp}$.

Output: Estimated Position $P_{est}$

1. Generate initial particles $P_{init}$ based on (12);
2. For each particle, evaluate the weighting $\omega_i$ via (13);
3. Calculate the particles’ convergence $N_{eff}$;
4. If $N_{eff} < 0.5 \times N_{max}$ then
   5. Reduce the altitudes to $H \times 50\%$;
   6. Re-Sampling new particles number with $D_{decay}$;
5. else
6. Re-Sampling new particles and go to Step 3.
7. If $H \geq H_{target}$ then
8. Go back to Step 2;
11. Output the estimated position $P_{est}$;

decided by

$$P_{init} = \frac{M_1 \cdot M_2}{(H \cdot (1 - R_{olp}))^2} \quad (12)$$

where a higher ratio $R_{olp}$ results in more initial particles.

The weight of each particle $\omega_i$ is estimated by computing the cosine similarity,

$$\omega_i = \cos(S_i, S_c) = \frac{S_i \cdot S_c}{\|S_i\| \cdot \|S_c\|} \quad (13)$$

where $S_c$ is the encoded feature from the aerial vehicle image, and $S_i$ is the feature of $i$th particle. New particles are regenerated from the highest weighted particles, with random translations added. The distribution of particles converges to a local area through iterative updates. We use weights to estimate particles’ convergence $N_{eff} = 1/(\sum(\omega_i)^2)$, and determine whether to change resolution level. When particles converge into potential areas ($N_{eff} < 0.5 \times N_{max}$), we decrease altitude and only keep $D_{decay} = 70 \sim 90\%$ of particles, then new particles are resampled around the remaining original weighted particles. By iteratively updating particles, our method accurately matches while reducing the computational burden. Our hierarchical searching procedure is also given in Algorithm 1.

Given a $[M_1 \times M_2]$ reference overhead image, searching for the best match through brute-force directly at the lowest altitude $H_1$ has a complexity of $O(P_{init}) \cdot O(C)$. Bound $O(C)$ includes reference image generation, feature extraction, and feature similarity calculation, and is a constant for different altitudes. Assume the highest altitude is $\alpha$ times $H_1$, then we calculate the computation complexity ratio $\mathcal{C}$ between Brute-force and Hierarchical searching by

$$\mathcal{C}_{\text{Brute-Force}} = \frac{O(M_1, M_2) \cdot O(C)}{(H \cdot (1 - R_{olp}))^2} \cdot \frac{D_{init}^4 \cdot O(C)}{\sum_{i=0}^{N_{max}} \left(\frac{M_1 \cdot M_2}{\alpha H \cdot (1 - R_{olp})^2} \cdot D_{decay}^4 \right) \cdot O(C)}$$

$$= \frac{\alpha^2 \cdot (1 - D_{decay})}{1 - D_{max}_{\text{decay}}} \quad (14)$$

where $l_{max}$ is the maximum number of layers of hierarchical searching. When we let $\alpha = 3$, $l_{max} = 5$ and $D_{decay} = 0.8$, we expect hierarchical approach to be around 2.7 times faster than the Brute-force method. Besides increased matching efficiency, our hierarchical search method also increases robustness to orientation and altitude variations.

V. EXPERIMENTS

We evaluate the performance of various VTRN methods with respect to changing illumination, low/high altitudes, and large flights. Two datasets are utilized for comprehensive evaluation: 1) the CMU Campus dataset and 2) the ALTO dataset.

The CMU Campus dataset is designed to test the invariance of VTRN methods to different lighting and altitudes, as well as test 3-D localization during the take-off/landing phase of a flight. This dataset was recorded with a DJI Mavic 2 Pro quadcopter equipped with a GoPro Max omnidirectional camera as shown in Fig. 6.

The ALTO dataset contains a 150 km trajectory from Ohio to Pennsylvania. The focus of this dataset is to capture complex natural terrain over longer trajectories at higher altitudes. This dataset was collected using a helicopter with a downward-facing pinhole camera and a GPS; Fig. 8 shows this platform.

Visual Terrain Relative Navigation Datasets

1) CMU Campus dataset includes 10 trajectories within the Carnegie Mellon University campus. For each trajectory, we recorded four passes at different times of
For each of the 10 trajectories around the CMU campus, we fly at 4 different times in order to capture slightly different appearances.

Fig. 8. ALTO dataset Helicopter platform. A helicopter equipped with a GPS and a downward-facing pinhole camera is used to collect the 150 km long trajectory for localization.

Table II shows the differences between our two VTRN datasets. The average distance of the CMU Campus trajectories is 1600 m, and each trajectory includes four passes under the different times of the day. The ALTO dataset includes 150 subtrajectories with an average distance of 1,000 m. For both datasets, there is no overlap between training and evaluation datasets. Fig. 7 shows some example images of different areas of CMU in the CMU Campus dataset. Fig. 9 shows example scenarios encountered in the ALTO flight. For the details of the ALTO dataset, please refer to the GitHub repo.

Evaluation Methods and Metrics: We evaluate the localization performance of our method on both CMU Campus and ALTO datasets via traditional place retrieval metrics. The evaluation dataset consists of cross-domain reference and testing queries on the same trajectories. Each testing frame can find its correspondences on reference queries. Successful place feature matching is based on testing queries’ retrieval poses. If the deviation distance of retrieval and target query is within a given threshold (20 m for CMU Campus dataset and 40 m for ALTO dataset), place recognition is counted as successful; otherwise, it is unsuccessful. We use average recall of top 10 and top-N retrievals, receiver operating characteristic (ROC) curve, feature difference, relative orientation distributions, and global relocalization success rates to analyze place recognition accuracy on

Table II: Properties of CMU Campus and ALTO Datasets

| Light Cond. | Distance       | Area       |
|-------------|----------------|------------|
| Varying     | 1,600 m x 4    | Urban      |
| Constant    | 1,000 m x 150  | Urban, Terrain |

4[Online]. Available: https://github.com/MetaSLAM/ALTO
evaluation trajectories of both datasets. We compare iSimLoc with learning-based feature learning methods (CycleGAN [29], AlexNet [36], NetVLAD [22], CALC [37]), and nonlearning based geometric methods HoG [38], Bag-of-words (BoW) [13] and CoHoG [39]).

All learning-based methods are trained with the same hardware: an Ubuntu 18.04 system with 64 GB of RAM and one Nvidia 1080Ti GPU. We chose to set the default visual input dimensions to $256 \times 256$. At inference time, this resolution is more suitable for the limited computation and memory available on a real-world small UAV payload (e.g. one NVIDIA Xavier NX). To train iSimLoc, we use only 10% of paired sim/real data from each dataset for domain-transfer training, and we use 60% of the overhead images for place recognition training; we evaluate on cross sim/real domains with the remaining 40% of the data. For other methods, we use 60% for training, and leave 40% as unseen environments for evaluation.

In condition-invariant analysis, we fix viewpoints and calculate place recognition average recalls of different methods under changing conditions. In viewpoint-invariant analysis, we fix environmental conditions and calculate average recall under different viewpoints (translations, orientations, and altitude ratios).

In hierarchical localization analysis, we analyzed successful global relocalization rates on both datasets using different global localization methods.

### A. Domain-Invariant Place Recognition Analysis

This section evaluates visual place recognition performance for different environmental conditions. We provide raw images and paired overhead images for the same perspectives for a fair comparison. Furthermore, we compare our CDTM module with existing nonlearning method HoG to verify place recognition ability. We first show domain adaptation on both CMU Campus and ALTO datasets by transferring raw images into overhead images. Based on CDTM module’s orthogonal feature constraints, we extract paired geometry features ($Z_G$) and condition features ($Z_C$) from the same image. As stated in Section. IV-A, additional conditional features from overhead images assists in domain transfer of real images. We compared recognition ability with and without orthogonal extraction to verify the above property.

Fig. 10 shows domain transfer on unseen environments. After training the network model on our two datasets, we pick unseen trajectories to examine image reconstruction performance. The
Fig. 11. Real-to-Sim feature differences for different datasets. Each subfigure represents the difference matrix between real images (x-axis) and corresponding overhead images (y-axis). Similarities are calculated by cosine distance. The first three rows show results on the same trajectory of CMU Campus datasets under illuminations of [10 A.M., 1 P.M., 7 P.M.]. The last row shows matching results on ALTO dataset.

Fig. 12. ROC analysis for cross-domain place recognition. Each subfigure shows the ROC curves of different methods for unseen environments on CMU Campus under conditions [10 A.M., 1 P.M., 7 P.M.] and ALTO datasets.

First rows of both sections in the figure show the input images; the last rows show the target images, and the third rows show images generated with our CDTM module. For comparison, we also show images generated with CycleGAN [29] network modules in the second rows. Different from CycleGAN, under varying illumination of CMU Campus dataset, our CDTM module also encodes condition styles into the image reconstruction. CycleGAN requires that target images must follow the same style, which restricts image reconstructions on unseen environments or conditions. Fig. 11 shows feature difference matrices between test query images and reference overhead images. We notice that both nonlearning and learning methods have varying performance if conditions change. In general, AlexNet and methods with domain transfer modules show higher robustness. With different domains as constraints, NetVLAD shows less generalization ability than AlexNet. However, the large size of the AlexNet model makes it ill-suited for real-time inference. With a domain transfer module (CDTM and CycleGAN), HoG shows higher robustness to domain differences. When comparing two different domain transfer modules, our CDTM module is able to generalize better than CycleGAN for unseen environments. Among learning-based methods, CALC and original NetVLAD are more sensitive to lighting changes. They cannot capture rich

| Table III: AVERAGE RECALL OF TOP 10 RETRIEVALS FOR DIFFERENT ENVIRONMENTS |
|----------------|----------------|----------------|----------------|----------------|
| Method         | 10 A.M.        | 1 P.M.         | 7 P.M.         | Terrain        |
| NetVLAD [22]   | 46.8%          | 52.1%          | 36.4%          | 68.1%          |
| CALC [39]      | 64.5%          | 70.8%          | 56.2%          | 80.2%          |
| AlexNet [36]   | 98.5%          | 95.8%          | 76.0%          | 90.7%          |
| HoG [58]       | 80.2%          | 87.5%          | 73.0%          | 86.5%          |
| CycleGAN [29]+HoG | 93.7%          | 94.9%          | 88.5%          | 89.3%          |
| iSimLoc(CDTM)+HoG | 96.6%          | 98.3%          | 93.8%          | 92.8%          |

Red indicates best result, and blue indicates second best.
Fig. 13. Localization results for different viewpoints on different datasets. For each dataset, we pick one trajectory from the same domain and generate test/reference queries with different pitch angles $[5^\circ, 15^\circ, 30^\circ]$ and yaw angles $[15^\circ, 45^\circ, 90^\circ, 135^\circ]$, and then analyze the average recall for top-$N$ retrievals.

geometric structures for proper place retrievals, especially on the ALTO dataset.

Table III shows the analysis of average recall for top 10 retrievals for different methods; domain transfer based on Cycle-GAN and our CDTM further improve the place retrieval accuracy compared to methods that do not utilize domain transfer. In this table, we highlight the best results in red and the second best in blue. We also notice that AlexNet performs well at the 10 A.M. case from the CMU Campus dataset, while CDTM performs well in all conditions. In Fig. 12, we present the ROC curves of different methods. CDTM has robust true-positive rates compared with other methods. However, while the nonlearning HoG method also performs well on the 1 p.m. CMU Campus dataset, it is not robust for all time conditions. Overall, the CDTM-based domain transfer module provides a high average recall rate with top 10 retrievals with confident estimation.

B. Viewpoint-Invariant Place Recognition Analysis

On both CMU Campus and ALTO datasets, our PEM extracts viewpoint-invariant descriptors for place retrievals from extracted condition-invariant features from our (CDTM) module. For this experiment, both reference and test images are from the same domain to focus on robustness to viewpoint differences. Our place descriptor constrains global localization to the Euclidean domain. Moreover, given matching testing and reference images, the PEM estimates relative orientations in parallel. In this section, we investigate robustness to viewpoint differences
and the accuracy of orientation estimation separately. Especially, for the ALTO dataset, due to the top–down perspective being different from the required inputs of the PEM module, we manually transform the top–down perspective images to omnidirectional images through the equilib library.\(^5\)

First, as shown in Fig. 13, for each dataset, we analyze viewpoint-invariance by applying several orientations on yaw ([15\(^\circ\), 45\(^\circ\), 90\(^\circ\), 145\(^\circ\)]) and pitch ([5\(^\circ\), 15\(^\circ\), 30\(^\circ\)]) angles to testing images. Most place recognition methods show high retrieval rates but only for small viewpoint changes. For a fixed pitch angle, place recalls of different methods drop significantly as the yaw angle is changed. As the yaw angle changes, iSimLoc shows higher and more consistent performance. With only viewpoint differences, iSimLoc’s top 5 place retrievals on both datasets is above 90%.

We further analyze viewpoint-invariance from a different perspective. For both datasets, we analyze the similarity of iSimLoc features given reference images from a local area with different rotations and scales. For each subfigure of Fig. 14, an image pixel corresponds to relative translation difference, and the center of the images are ground truth matching areas. We take reference images for different FOVs, e.g., scale 1.0 means reference and test images have the same perspective. Scales 0.8 and 1.2 indicate reference images are taken at 80% and 120% of the altitude of testing images. For local translation, orientation, and scale differences, iSimLoc shows a higher similarity score on both CMU Campus and ALTO datasets, which will improve global relocalization robustness for scale differences.

In addition, given a matched image pair, relative orientation estimation uses the same spherical features that are used for the viewpoint-invariant descriptor as depicted in Section IV-B2.

As shown in Fig. 15, on both CMU Campus and ALTO datasets, relative orientation between testing and reference queries can be evaluated according to maximum spherical correlation \(C\) as we stated in Section IV-B2. For both datasets, we present the relative orientation error distribution, which has a domain within \([-30 \sim 30]^\circ\). Compared to performance on CMU Campus dataset, the estimator shows higher orientation error on the ALTO dataset, and this is mainly caused by textureless terrain environments, which increase the difficulty of accurate orientation estimation.

In general, the PEM module provides accurate viewpoint-invariant place recognition, which improves global localization efficiency by reducing the sampling space from 6-DoF space to Euclidean space. The orientation estimation module concurrently estimates relative orientation between testing images and matched query images. However, as we can see in the feature matching results, there are outliers due to common and similar textures, which is something that becomes more abundant in long-term navigation. In the following section, we investigate hierarchical localization performance in terms of robustness, accuracy, and efficiency.

C. Hierarchical Localization Analysis

Hierarchical matching improves global localization efficiency over a large search area without losing matching accuracy, as depicted in Section IV-C. The robustness of hierarchical localization can be further boosted by ignoring condition and viewpoint changes, which is done with the aid of the domain transfer module (CDTM) and viewpoint-invariant module (PEM). We analyze different global localization methods on the CMU Campus dataset as shown in Fig. 16, i.e., fixed 120 m and 40 m altitudes and high-to-low altitudes. Global localization uses the same iSimLoc place feature and particle filter under the exact

\(^5\)[Online]. Available: https://github.com/haruishi43/equilib
initial particles requires more processing time. With hierarchical matching, we can match well even with few initial particles. Specifically, the hierarchical matching approach enables a more robust initial estimate at a higher initial altitude, which further helps achieve successful localization. Regarding efficiency, we notice that a hierarchical approach speeds up localization by a factor of $4 - 10$. Compared with the performance on CMU Campus dataset, localization over the ALTO dataset shows lower success rates, likely caused by less-distinguishable geometric features in the terrain.

VI. IMPLEMENTATION DETAILS

This section will describe the details of our implementation procedure for iSimLoc. In the training procedure, we first train the domain transfer module for 20 epochs with an initial learning rate of 0.001. Then we change the learning rate of the domain transfer to 0.00005, set the initial learning rate of the PEM to 0.001, and jointly train both modules with another 20 epoch.

Before conducting the iSimLoc localization for nonvisited areas, we generate spherical projections through the Google API, which can provide a top-down view from different altitudes, then use the equilib library to generate spherical view with a resolution of $256 \times 256$. According to (12), for the $1000 \times 1000$ m areas with $R_{olp} = 0.8$ and altitude at $H = 100$ m, the initial particle size is around 2500. As we investigate in Table IV, the configurations of $R_{olp}$ and $H$ significantly affect the final successful rates. But in the real-world UAV terrain relative navigation, robots also need to balance localization accuracy and onboard computation capacity.

VII. DISCUSSION AND LIMITATION

As depicted in Section III, our iSimLoc method consists of a condition- and viewpoint-invariant place feature learning module and a hierarchical localization module for VTRN task. The CDTM of iSimLoc improves localization accuracy with the assistance of conditional features of target domains as demonstrated by results shown in Fig. 10. As depicted in Section V-A, our CDTM reconstruction is based on both geometric and conditional features, which further improves localization accuracy for unseen environments compared to CycleGAN and AlexNet by 3.75% and 5.09% on average. However, the current domain transfer module can only be used with low-resolution images ($128 \times 128$ or $256 \times 256$), which reduces it’s ability to capture rich geometric features. The PEM that is shown in Section V-B demonstrates robustness to viewpoint differences and orientation estimation compared to other methods by $5 - 10\%$ as depicted in Fig. 13. However, the current spherical convolution network in the PEM has very shallow layers (four spherical convolutions), which may reduce its feature extraction ability. Finally, as analyzed in Section IV-C, our hierarchical searching method balances matching efficiency and accuracy. As shown in Table IV, by using the hierarchical method, iSimLoc can reach up to 80% successful retrieval rates for large-scale place recognition compared to 40% for the following best method AlexNet.
Fig. 16. Comparison of different global localization methods on CMU dataset. The X-Y positions are relative to the target position. In each sub-figure, we plot particle distributions and density analysis. The first two rows show global localization with iSimLoc place features at constant altitudes, 120 m and 40 m, respectively: we can notice that the particles in the constant 40 m case better converge on the ground truth [0,0] position than in the constant 120 m case, but it requires many more particles to achieve this accuracy. The last row shows localization using our hierarchical approach (from a 120 m altitude perspective), which quickly converges on a tighter distribution around the ground truth [0,0] point than does the constant 120 m case, and it does so with fewer particles than in the constant 40 m case.

### Table IV

Global Relocalization Robustness and Efficiency

| Localization Method       | ACC$_{0.9}$ | ACC$_{0.8}$ | ACC$_{0.7}$ | ACC$_{0.9}$ | ACC$_{0.8}$ | ACC$_{0.7}$ | Time$_{0.9}$ | Time$_{0.8}$ | Time$_{0.7}$ |
|---------------------------|-------------|-------------|-------------|-------------|-------------|-------------|--------------|--------------|--------------|
|                           | CMU Campus Dataset | ALTO Dataset | Run Time    | Hierarchical Localization |
| CALC [37] (100% Altitude) | 10.1%       | 6.8%        | 4.3%        | 10.8%       | 7.1%        | 4.1%        | 2.4s         | 1.9s         | 1.6s         |
| HOG [38] (100% Altitude)  | 12.1%       | 9.3%        | 7.2%        | 11.4%       | 8.5%        | 5.6%        | 2.1s         | 1.7s         | 1.5s         |
| NetVLAD [23] (100% Altitude) | 24.7%      | 18.3%       | 12.4%       | 22.5%       | 16.8%       | 13.6%       | 3.1s         | 2.5s         | 2.1s         |
| AlexNet [36] (100% Altitude) | 45.8%      | 31.6%       | 23.5%       | 39.7%       | 28.4%       | 19.2%       | 3.3s         | 2.6s         | 2.2s         |
| iSimLoc (100% Altitude)   | 88.7%       | 84.5%       | 74.8%       | 83.8%       | 76.1%       | 72.4%       | 1.5s         | 1.2s         | 0.9s         |
| iSimLoc (50% Altitude)    | 89.1%       | 84.2%       | 78.4%       | 85.4%       | 79.2%       | 70.3%       | 6.2s         | 5.0          | 4.3s         |
| iSimLoc (33% Altitude)    | 91.8%       | 85.3%       | 76.1%       | 86.9%       | 81.7%       | 72.5%       | 12.9s        | 10.2s        | 9.0s         |

Constant Altitude Localization

|                           | ACC$_{0.9}$ | ACC$_{0.8}$ | ACC$_{0.7}$ | Time$_{0.9}$ | Time$_{0.8}$ | Time$_{0.7}$ |
|---------------------------|-------------|-------------|-------------|--------------|--------------|--------------|
| iSimLoc (100% Altitude)   | 59.2%       | 51.9%       | 48.8%       | 52.5%        | 48.8%        | 43.1%        | 1.5s         | 1.2s         | 1.1s         |
| iSimLoc (50% Altitude)    | 61.6%       | 52.5%       | 48.2%       | 54.7%        | 49.2%        | 45.3%        | 6.2s         | 5.1s         | 4.4s         |
| iSimLoc (33% Altitude)    | 63.5%       | 55.3%       | 49.7%       | 53.6%        | 51.4%        | 45.9%        | 13.1s        | 10.8s        | 9.1s         |

Red indicates the best result, and blue indicates the second best. The floats (0.7, 0.8, 0.9) indicate the decay rates $D_{decay}$ of the particles as mentioned in Section IV-C. The baseline is set at $D_{decay} = 0$ and initial localization altitude at the maximum height (120 m for CMU dataset, and 200 m for ALTO dataset).

Since iSimLoc is designed to give the top matches for global relocalization, a complete system must be combined with online image alignment methods and visual odometry for continuous pose estimation. The run time analysis in Table IV, shows the hierarchical version of iSimLoc has 88.7% (Campus) and 83.8% (ALTO) correct matches with 1.5 s computation time, and 89.1% and 85.4% correct matches with a 6.2 s computation time.

However, the limitation of iSimLoc is also notable. One major limitation lies in the final localization accuracy. For our approach, the place recognition can only achieve rough global localization (i.e., successful retrieval is set under 20 m), which can not satisfy highly accurate online localization (i.e., 1−10 m). One potential solution is to combine the global place descriptor-based approach with one of the current state-of-the-art feature-based image matching approaches [40]. Another major drawback is the inference efficiency. As shown in Table IV, the hierarchical localization time is around 10−15 s, which can not satisfy the real-time requirements. However, in the real-world application, the robot is only expected to run the online relocalization within a potential area (i.e., 500 m × 500 m) for one time. One potential option to achieve online efficient localization performance is to combine the...
iSimLoc-based global localization with real-time online tracking within a limited searching window (i.e., 50 m × 50 m). On the other hand, since the computation time depends on the number of particles used and the geographic size of the search area; thus such particle evaluation procedure can be further sped up via parallel computing on GPU/FPGA.

VIII. CONCLUSION

This article presented iSimLoc, a hierarchical global relocalization method for VTRN with the assistance of satellite imagery. iSimLoc can learn a place recognition model with only simulated images and only requires a small portion (10%) of paired sim/real data to train the domain-transfer module compared to other methods that require 60% of the data for domain transfer. Because of the viewpoint-invariant property, it is capable to recognize the same place even with orientation and altitude differences, and the hierarchical matching method helps iSimLoc balance efficiency and robustness for global relocalization.

In future research, we intend to improve the efficiency of hierarchical localization by parallelizing the method. Additionally, we plan to improve the current domain-transfer module’s ability to capture rich geometric details to integrate iSimLoc with existing image alignment methods for accurate pose estimation.

ACKNOWLEDGMENT

The authors would like to thank Near Earth Autonomy, Pittsburgh, PA, for assistance in creating the “ALTO” dataset. We especially thank Lingyun Xu for her suggestions on both methods and experiments.

REFERENCES

[1] E. Larson et al., “Autonomous underwater survey apparatus and system,” U.S. Patent Appl. 16/173,567, May 2 2019.
[2] E. Guisado-Pintado, D. W. Jackson, and D. Rogers, “3D mapping efficacy of a drone and terrestrial laser scanner over a temperate beach-dune zone,” Geomorphology, vol. 328, pp. 157–172, 2019.
[3] F. Pallottino, M. Bicoca, P. Nardi, S. Figorilli, P. Menassiti, and C. Costa, “Science mapping approach to analyze the research evolution on precision agriculture: World, eu and italian situation,” Precis. Agriculture, vol. 19, no. 6, pp. 1011–1026, 2018.
[4] M. Silvagni, A. Tonoli, E. Zenerino, and M. Chiaberse, “Multipurpose UAV for search and rescue operations in mountain avalanche events,” Geomatics, Natural Hazards Risk, vol. 8, no. 1, pp. 18–33, 2017.
[5] M. B. Bejiga, A. Zeggada, A. Noufildj, and F. Melgani, “A convolutional neural network approach for assisting avalanche search and rescue operations with UAV imagery,” Remote Sens., vol. 9, no. 2, 2017, Art. no. 100.
[6] M. Mittal, R. Mohan, W. Burgard, and A. Valada, “Vision-based autonomous UAV navigation and landing for urban search and rescue,” in Proc. Robot. Res.: 19th Int. Symp. ISR, 2022, pp. 575–592.
[7] A. T. Fragoso, C. T. Lee, A. S. McCoy, and S.-J. Chung, “A seasonally invariant deep transform for visual terrain-relative navigation,” Sci. Robot., vol. 6, no. 55, 2021, Art. no. eabt3320.
[8] P. Yin, F. Wang, A. Egorov, J. Hou, Z. Jia, and J. Han, “Fast sequence-matching enhanced viewpoint-invariant 3-D place recognition,” IEEE Trans. Ind. Electron., vol. 69, no. 2, pp. 2127–2135, Feb. 2022.
[9] P. Yin, L. Xu, J. Zhang, H. Choset, and S. Scherer, “3dLoc: Image-to-range cross-domain localization robust to inconsistent environmental conditions,” in Proc. Robot. Sci. System. Robot., Sci. 2021, pp. 1–9.
[10] I. Cisneros, P. Yin, J. Zhang, H. Choset, and S. Scherer, “Alto: A large-scale dataset for UAV visual place recognition and localization,” 2022, arXiv:2207.12317.
[11] J. Brejcha and M. Čadik, “State-of-the-art in visual geo-localization,” Pattern Anal. Appl., vol. 20, no. 3, pp. 613–637, 2017.
[12] M. Mishkin, M. Perdoch, and J. Matas, “Place recognition with WxBS retrieval,” in Proc. CVPR Workshop Vis. Place Recognit. Changing Environ., 2015, pp. 1–8.
[13] H. Jézouf, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, “Aggregating local image descriptors into compact codes,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 9, pp. 1704–1716, Sep. 2012.
[14] B. Patel, T. D. Barfoot, and A. P. Schoellig, “Visual localization with Google Earth images for robust global pose estimation of UAVs,” in Proc. IEEE Int. Conf. Robot. Automat., 2020, pp. 6491–6497.
[15] M. Bianchi and T. Barfoot, “UAV localization using autoencoded satellite images,” IEEE Robot. Automat. Lett., vol. 6, no. 2, pp. 1761–1768, Apr. 2021.
[16] G. Baatz, O. Sauer, K. Köser, and M. Pollefeys, “Large scale visual geo-localization of images in mountainous terrain,” in Proc. Eur. Conf. Comput. Vis., Springer, 2012, pp. 517–530.
[17] G. Baatz, O. Sauer, K. Köser, and M. Pollefeys, “Leveraging topographic maps for image to terrain alignment,” in Proc. 2nd Int. Conf. 3D Imaging, Model., Process., Visual. Transmiss., 2012, pp. 487–492.
[18] B. Grellson, A. Robinson, M. Felsberg, and F. S. Khan, “GPS-level accurate camera localization with horizonnet,” J. Field Robot., vol. 37, no. 6, pp. 951–971, 2020.
[19] J.-K. Lee and K.-J. Yoon, “Real-time joint estimation of camera orientation and vanishing points,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 1866–1874.
[20] K. Pluckter and S. Scherer, “Precision UAV landing in unstructured environments,” in Proc. Int. Symp. Exp. Robot., 2020, pp. 177–187.
[21] M. J. Milford and G. F. Wyeth, “SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights,” in Proc. IEEE Int. Conf. Robot. Automat., 2012, pp. 1643–1649.
[22] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “Netvlad: CNN architecture for weakly supervised place recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 5297–5307.
[23] S. Cai, Y. Guo, S. Khan, J. Hu, and G. Wen, “Ground-to-aerial image geo-localization with a hard exemplar reweighting triplet loss,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 8391–8400.
[24] L. Liu and H. Li, “Lending orientation to neural networks for cross-view geo-localization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 5624–5633.
[25] S. Yang and S. Scherer, “CubeSLAM: Monocular 3D object slam,” IEEE Trans. Robot., vol. 35, no. 4, pp. 925–938, Aug. 2019.
[26] P. Yin, F. Wang, A. Egorov, J. Hou, J. Zhang, and H. Choset, “Seqsphereval: Sequence matching enhanced orientation-invariant place recognition,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2020, pp. 5024–5029.
[27] C. Esteves, C. Allen-Blanchette, A. Makadia, and K. Daniilidis, “Learning so(3) equivariant representations with spherical CNNs,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 52–68.
[28] P. Yin et al., “MRS-VPR: A multi-resolution sampling based global visual place recognition method,” in Proc. Int. Conf. Robot. Automat., 2019, pp. 7137–7142.
[29] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 2223–2232.
[30] I. Goodfellow et al., “Generative adversarial nets,” in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 2672–2680.
[31] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling, “Spherical CNNs,” in Proc. Int. Conf. Learn. Representations, 2018, pp. 1–15.
[32] L. Bernreiter, L. Ott, J. Nieto, R. Siegwart, and C. Cadena, “PHASER: A robust and correspondence-free global pointcloud registration,” IEEE Robot. Automat. Lett., vol. 6, no. 2, pp. 855–862, Apr. 2021.
[33] P. J. Kostelec and D. N. Rockmore, “FFTs on the rotation group,” J. Fourier Anal. Appl., vol. 14, no. 2, pp. 145–179, 2008.
[34] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “Airsim: High-fidelity visual and physical simulation for autonomous vehicles,” in Field and Service Robotics, Cham, Switzerland: Springer, 2017. [Online]. Available: https://arxiv.org/abs/1705.05065
[35] N. Gorelick, M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, and R. Moore, “Google earth engine: Planetary-scale geospatial analysis for everyone,” Remote Sens. Environ., vol. 202, pp. 18–27, 2017.
[36] N. Sänderhaus, S. Shirazi, F. Dayoub, B. Ucpeft, and M. Milford, “On the performance of convnet features for place recognition,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2015, pp. 4297–4304.
[37] N. Merrill and G. Huang, “Lightweight unsupervised deep loop closure,” in Proc. Robot., Sci. Syst., Pittsburgh, PA, USA, Jun. 2018, pp. 1–10.

[38] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2005, vol. 1, pp. 886–893.

[39] M. Zaffar, S. Ehsan, M. Milford, and K. McDonald-Maier, “Cohog: A light-weight, compute-efficient, and training-free visual place recognition technique for changing environments,” IEEE Robot. Automat. Lett., vol. 5, no. 2, pp. 1835–1842, Apr. 2020.

[40] A. Wang et al., “Superglue: A stickier benchmark for general-purpose language understanding systems,” Adv. Neural Inf. Process. Syst., vol. 32, 2019.

Peng Yin (Member, IEEE) received the bachelor’s degree in automation from Harbin Institute of Technology, Harbin, China, in 2013, and the Ph.D. degree in mechatronic engineering from the University of Chinese Academy of Sciences, Beijing, China, in 2018.

He is a Research Post-doctoral with the Department of the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. His research interests include LiDAR SLAM, place recognition, 3D perception, and reinforcement learning.

Dr. Yin was a Reviewer for several IEEE Conferences ICRA, IROS, ACC, RSS.

Ivan Cisneros received the B.S. degree in electrical engineering with a minor in computer science from Harvard University, Cambridge, MA, USA, in 2016. He is currently working toward the master’s degree in robotics with the Robotics Institute at Carnegie Mellon University (CMU), Pittsburgh, PA, USA.

He worked full-time with NASA-JPL on several flight projects for 3 years before starting his graduate studies with CMU. His research interests include SLAM, visual localization, 3D Perception, and deep learning.

Shiqi Zhao received the bachelor’s degree in mechanical design, manufacturing and automation from Dalian University of Technology, Dalian, China, in 2018, and the master’s degree in mechanical engineering from the University of California San Diego, CA, USA, in 2020.

He works as an Intern with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. His research interests include place recognition, 3D perception, and deep learning.

Ji Zhang received the Ph.D. degree in robotics from Carnegie Mellon University, Pittsburgh, PA, USA, in 2017.

He is a Systems Scientist with the Robotics Institute, Carnegie Mellon University, where he leads in the development of a series of autonomous navigation algorithms. His work was ranked #1 on the odometry leaderboard of KITTI Vision Benchmark between 2014 and 2021. He founded Kaarta, Inc, a CMU spin-off commercializing 3D mapping and modeling technologies, and stayed with the company for 4 years as chief scientist. His research interests are in robotic navigation, spanning localization, mapping, planning, and exploration.

Howie Choset (Fellow, IEEE) received the B.S. Eng. degree in computer science and the B.S. Econ. degree in entrepreneurial management from the University of Pennsylvania (Wharton), Philadelphia, PA, USA, both in 1990, the M.S. and Ph.D. degrees in mechanical engineering from California Institute of Technology (Caltech), Pasadena, CA, USA, in 1991 and 1996, respectively.

He is currently a Professor of robotics with the Carnegie Mellon University, Pittsburgh, PA, USA. His research group reduces complicated high-dimensional problems found in robotics to low-dimensional simpler ones for design, analysis, and planning.

Sebastian Scherer (Senior Member, IEEE) received the B.S. degree in computer science from Carnegie Mellon University (CMU), Pittsburgh, PA, USA, in 2004, the M.S. and Ph.D. degrees in robotics from CMU, in 2007 and 2010, respectively.

He is an Associate Research Professor with the Robotics Institute, Carnegie Mellon University. His research focuses on enabling autonomy for unmanned rotorcraft to operate at low altitude in cluttered environments. Dr. Scherer is a Siebel scholar and a recipient of multiple paper awards and nominations, including AIAA@Infotech 2010 and FSR 2013. His research has been covered by the national and internal press including IEEE Spectrum, The New Scientist, Wired, Der Spiegel, and the WSJ.