Robotics: Science and Systems 2023
Daegu, Republic of Korea, July 10-July 14, 2023

Self-Supervised Lidar Place Recognition in Overhead Imagery Using Unpaired Data

Tim Y. Tang, Daniele De Martini, and Paul Newman
Mobile Robotics Group, Oxford Robotics Institute, University of Oxford
Email: {ttang, daniele, pnewman}@robots.ox.ac.uk

Abstract—**As much as place recognition is crucial for navigation, mapping and collecting training ground truth, namely sensor data pairs across different locations, are costly and time-consuming. This paper tackles these by learning lidar place recognition on public overhead imagery and in a self-supervised fashion, with no need for paired lidar and overhead imagery data. We learn the cross-modal data comparison between lidar and overhead imagery with a multi-step framework. First, images are transformed into synthetic lidar data and a latent projection is learned. Next, we discover pseudo pairs of lidar and satellite data from unpaired and asynchronous sequences, and use them for training a final embedding space projection in a cross-modality place recognition framework. We train and test our approach on real data from various environments and show performances approaching a supervised method using paired data.**

I. Introduction

Lidar is widely considered an ideal sensor for outdoor operation, as it provides a long sensing range, 360° field-of-view (FOV), invariance to lighting, and robustness against weather conditions. For this reason, place recognition, also known as topological localisation, has been extensively researched for lidar [5, 13, 26, 18, 7, 38, 59]. Existing methods require lidar data to have been previously recorded in the places of operation, either as an aggregated point-cloud map or individual scans. When previously recorded sensory data are unreliable or unavailable, off-the-shelf overhead imagery, such as Google satellite images, can be used as an alternative map data source for place recognition. Even under normal operating conditions, overhead imagery can serve as an additional information source for redundancy.

When projected to the x-y plane and expressed as a 2-D scan image, lidars capture geometric features also visible from bird's-eye view aerial photos, providing useful signals for cross-comparison. Nevertheless, localising a lidar in a satellite image map remains challenging, as aerial imagery and range sensor scans are severely different. Recent works were proposed on pose estimation [54, 55] and place recognition [56] of radar and lidar using aerial images. Typically, learning place recognition requires *paired* data to form positive pairs in metric learning as in [56]. This, in turn, relies on accurate time-synced ground truth to query for a satellite image centred at the true centre position of each lidar scan, forming geometric one-to-one correspondences. In practice, collecting and post-processing time-synced GPS and inertial data for ground truth can require substantial cost and time. When live and map data are from the same sensor type, self-supervised learning with artificial paired data can be done by data augmentation [41, 16]. However, self-supervised metric learning is largely unsolved between cross-sensory data without known pairs.

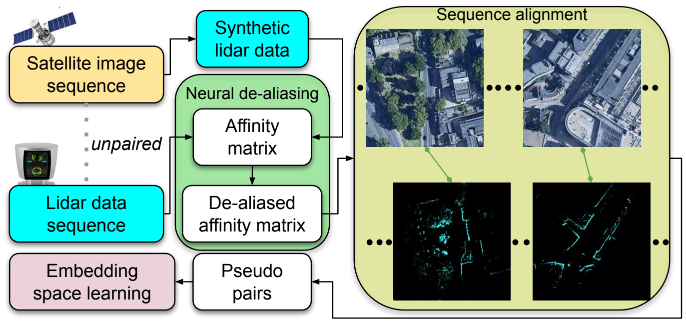


Fig. 1: An overview of the proposed framework: starting from unpaired lidar and satellite data, we first create synthetic lidar data from satellite images. An affinity matrix is built from real and synthetic lidar data, which is de-aliased using a novel learned method. We then discover pseudo pairs of satellite and lidar images with sequence alignment, used to learn a final embedding space for place recognition. Lidar and satellite images have arbitrary heading offsets, but here they are aligned in heading for visualisation purposes.

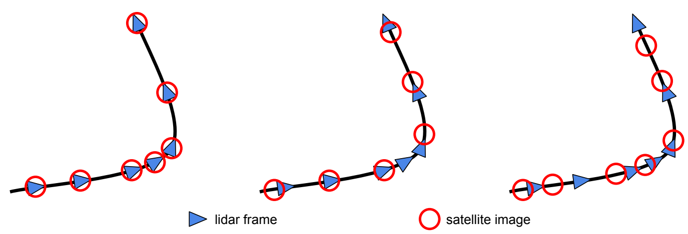


Fig. 2: Different satellite imagery collection strategies: here lidar data are collected at a certain frequency as a vehicle travels along a route. **Left:** in the supervised case, a satellite image is queried at the centre position of each lidar scan, forming corresponding one-to-one pairs. **Middle:** satellite images are sampled uniformly along the route, resulting in unpaired data. **Right:** satellite images are sampled based on asynchronous, previously recorded position measurements, resulting in unpaired data.

We present a self-supervised method for learning lidar place recognition in overhead images from *unpaired* data, with an overview in Figure 1. Suppose a lidar-equipped vehicle travels along a known but never traversed route, we can, for example, sample satellite images uniformly along the route, forming a set of lidar and satellite data without geometric one-to-one correspondences (Figure 2 Middle). Our method learns from unpaired data, relaxing the need for on-board ground truth to collect time-synced, paired lidar and aerial data. Alternatively, if the route has been previously driven by a vehicle with an on-board GPS/INS, then asynchronous position measurements collected on GPS/INS timestamps from the prior drive can also be used to query for satellite images along the route (Figure 2

Right), rather than sampling uniformly. This also mitigates the need for an always-present on-board GPS to collect training data. Figure 2 illustrate the difference between paired and unpaired data under the context of our problem set-up.

To learn from unpaired, cross-sensory data, we first address the sensory difference by utilising existing work on unpaired image-to-image translation. Our method then mines pseudo positive pairs with sequence alignment, where we propose a simple yet effective self-supervised learning strategy to de-alias the noisy affinity matrix resulting from the modality gap. We demonstrate on public datasets that the performance of our method is approaching a supervised approach trained with paired data when tested on unseen places.

II. Related Work

A. Deep Learning for Range Sensor Place Recognition

Neural networks for range sensor place recognition can operate directly on point data or 2-D scan images.

1) Point-based: Early work by Uy and Lee [57] utilises NetVLAD [2] after a PointNet [45] backbone to learn a global descriptor for retrieval. A common strategy utilised by recent methods seeks to learn per-point local descriptors first and then aggregate them into a global descriptor for place recognition using pooling [35, 28], normalisation [60], learned layers [35, 63, 12], or bag of words (BoW) [9].

2) Image-based: Sun et al. [53] expressed lidar data as bird's-eye view images and directly learns a global pose, which is used to seed a Monte Carlo Localiser for pose refinement. Saftescu et al. [46] expressed radar data as images in polar coordinates and learned rotation-invariant embeddings for place recognition via circular padding. Barnes and Posner [3] predicted keypoints and pixel-level local descriptors from radar images and aggregated them to a per-image global descriptor via pooling for place recognition. OverlapNet [7] uses 2-D range, normal, intensity, and semantic images extracted from 3D lidar data to predict the overlap between scans as a proxy for detecting loop closure. OverlapNet was extended to a Transformer [58]-based architecture for rotation-invariant learning in [40], and to handle sequential data in [39].

B. Localisation Using Aerial Imagery

Aerial imagery maps can be used to localise a forward-facing camera image in the geo-localisation problem. In this case, the localisation happens from different view perspectives (forward vs top-down). Still, the sensory nature of the query image and the map database are the same, i.e. RGB data. When a range sensor localises in an aerial imagery map, the view perspectives are consistent as range sensor scans are often expressed as bird's-eye view images. However, the data belong to different modalities, requiring alternative strategies for bridging the sensory gap.

1) Cross-view, intra-modality: Early work by Lin et al. [31] learns cross-view geo-localisation with hand-crafted features and an SVM classifier. CVMNet [21] applies two streams of convolutional layers followed by a NetVLAD layer to learn cross-view matching with a weighted soft-margin ranking loss. Li et al. tackled cross-view retrieval by supplying orientation maps [32], predicting cross-view orientation [50], learning spatial attention [49], and using optimal feature transport [51].

2) Consistent view, cross-modality: The method in [10] accumulates lidar intensity sub-maps and directly localises against overhead imagery using Normalised Mutual Information. Some methods [11, 29] extract hand-crafted features from overhead imagery before comparing them against range sensor data. Zhu et al. [65] learned a matching probability between lidar grid-maps and satellite imagery to enhance a lidar SLAM pipeline. The work in [14] combines ground camera and lidar data to solve registration against overhead imagery in a correlation-maximisation approach. Tang et al. addressed the modality gap between range sensor data and satellite imagery by generating synthetic range sensor images [54, 55] or representing satellite imagery as points [56]. Prior learning-based methods, different from our work, have mostly relied on paired range sensor and satellite data for supervision.

Most recently, Kim et al. [27] trained a semantic segmentation network using hand-annotated satellite images and extracted building outlines while also performing orthorectification. This practically converted satellite imagery to a representation similar to building outlines from OpenStreetMap (OSM). Localisation is then solved by computing mutual information between lidar scans against building outlines.

C. Localisation Using Other Publicly Available Resources

Other off-the-shelf resources, in particular OSM, have been applied for robot localisation. The methods in [6, 15] match Visual Odometry (VO) against road segments extracted from OSM for global localisation. The methods in [43, 61] bridge the modality gap between lidar and OSM by learning domain-invariant semantic descriptors. Cho et al. [8] designed a hand-crafted rotation-invariant descriptor based on the distance to buildings applicable to both lidar point-clouds and OSM data.

D. Cross-Sensory Retrieval With Unpaired Data

Several recent works have targeted cross-sensory retrieval with unpaired data. Yin et al. [62] applied a GAN-based style transfer that generates synthetic lidar images from radar images to achieve radar localisation in lidar maps. Jeong et al. [24] combined style transfer with joint feature space learning to match camera images with infra-red images. The method in [8] uses a hand-crafted descriptor for lidar and OSM data, and therefore does not require paired data for training, but is not directly applicable to our problem set-up. The method in [27] does not require paired lidar and satellite data, but relies on hand-annotated satellite imagery semantics to train semantic segmentation, which is another form of ground truth that is potentially time consuming to acquire.

Though promising, the method by Yin et al. [62] seeks to address the modality gap between two types of range sensors, while the method in [24] targets RGB images and infra-red images, both having a much smaller modality gap than between lidar and aerial images. In our experiments, neither [62] nor [24] were sufficient in our problem of lidar place recognition in satellite imagery with unpaired data.

E. Unpaired Image-to-Image Translation

A core module of our pipeline involves generating synthetic lidar images from satellite images in an unpaired set-up. Unpaired image-to-image translation can be achieved using cycle-consistency [64] between the input and reconstructed images, the assumption of a shared latent space [33, 22], or contrastive learning [44]. Several prior works were proposed to learn multi-modal image generation [22, 30, 1], but empirically they resulted in limited success in generating lidar images from satellite images in our experiments. Our choice fell on CycleGAN [64] as it is a well-established method.

III. Problem Overview

Suppose a vehicle travels along a known route and takes lidar scans at a certain frequency, resulting in a sequence of N lidar images $\mathcal{X} = \{X_1, X_2, \ldots, X_N\}$. Satellite images are queried along the route, forming a sequence of M satellite images $\mathcal{Y} = \{Y_1, Y_2, \ldots, Y_M\}$. In the simplest case, paired data are available through ground truth: for each lidar scan X_i, we have a satellite image Y_j sharing the same centre position, resulting in one-to-one correspondences with $N = M$.

Here, we consider a more general scenario where paired data are unavailable and the lidar and satellite sampling positions differ, for example, if the satellite images are sampled uniformly along the route. In this context, we want to find for each X_i the closest $Y_j \in \mathcal{Y}$; vitally, there are no known correspondences from training data, and the mapping from $\mathcal{X}$ to $\mathcal{Y}$ is neither surjective nor injective.

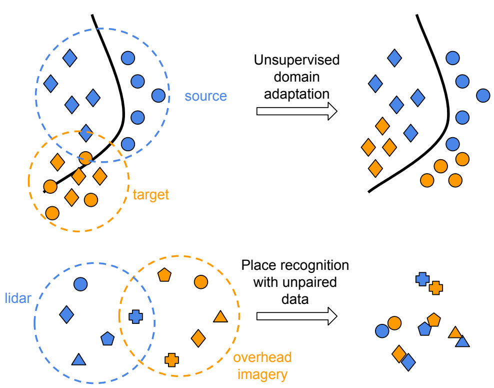


Fig. 3: **Top:** From unpaired data, UDA typically seeks to learn consistent decision boundaries between the source and target modalities. **Bottom:** In our problem, the goal is to search for neighbours across modalities from continuously distributed data.

Here, we distinguish cross-sensory or cross-modal place recognition with unpaired data from the perhaps better studied problem of unsupervised domain adaptation (UDA) [36, 47, 25]. As shown in Figure 3, UDA typically aims to learn consistent decision boundaries among source and target data. Our problem is different in that both lidar and satellite images are distributed *continuously* along a route, and neither can be appropriately classified into a discrete number of distinct categories. Our goal is instead to find the nearest neighbour from modality $\mathcal{Y}$ for each sample in $\mathcal{X}$, which can be achieved by sampling positive pairs from $\mathcal{X}$ and $\mathcal{Y}$ for metric learning. While this is trivial if $\mathcal{X}$ and $\mathcal{Y}$ are paired, our method seeks to extract positive pairs from unpaired data.

IV. Methodology

The core idea of our approach is to exploit the fact that, albeit without one-to-one correspondences, lidar and satellite data follow the same underlying sequence, as they are collected along the same known route. We begin by learning an embedding function for projecting a lidar image to a vector space descriptor using time consistency on lidar only (Section IV-B). In parallel, we learn to generate synthetic lidar images from satellite imagery using CycleGAN [64] (Section IV-C). We can then use this projection to construct an affinity matrix by comparing synthetic and real lidar images.

Though the synthetic lidar images are realistic, there is no guarantee they capture the same regions of the scene as a real lidar situated at the centre of the satellite image. The affinity matrix will then be corrupted by heavy signal aliasing. We propose a simple yet effective learned strategy to de-alias the affinity matrix and improve its signal-to-noise ratio before sequence alignment (Section IV-D). Pseudo pairs are then selected from sequence alignment and used as positive pairs to learn a final embedding space projection for place recognition (Section IV-E). Figure 1 shows the method overview.

A. Data Representation

In our problem set-up, we project 3D lidar data to the x-y plane to form bird's-eye view lidar images, where points with z value less than a threshold are discarded to remove ground points. The intensity in each pixel is the average intensity of all points projected to that pixel.

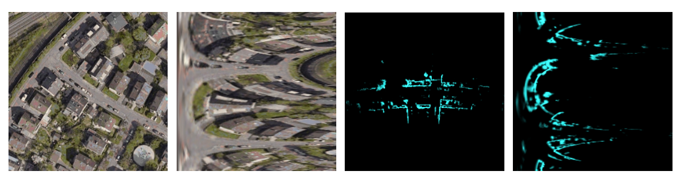


Fig. 4: A satellite image, a lidar image, and their polar counterparts.

Since the heading offset between lidar scans and satellite data is unknown, the place recognition pipeline must be rotation-invariant. Motivated by this, we convert lidar and satellite images to a polar coordinate representation, where the axes are range r and azimuth θ. A rotation in Cartesian space becomes a circular shift in the polar domain along the vertical axis, and, as CNNs are equivariant to vertical shifts, they can be easily trained to be rotation-invariant through data augmentation when applied to polar images. Examples of polar representations of satellite and lidar images are shown in Figure 4. While recent work proposes the Radon transform, which is $SE(2)$ equivariant, for lidar localisation [37], we observed that Radon transform fails to preserve regions in satellite images with a strong gradient, such as building edges.

B. Descriptor Embedding for Lidar Images

We wish to learn a function F, parametrised by a neural network, that projects a lidar image to a descriptor space

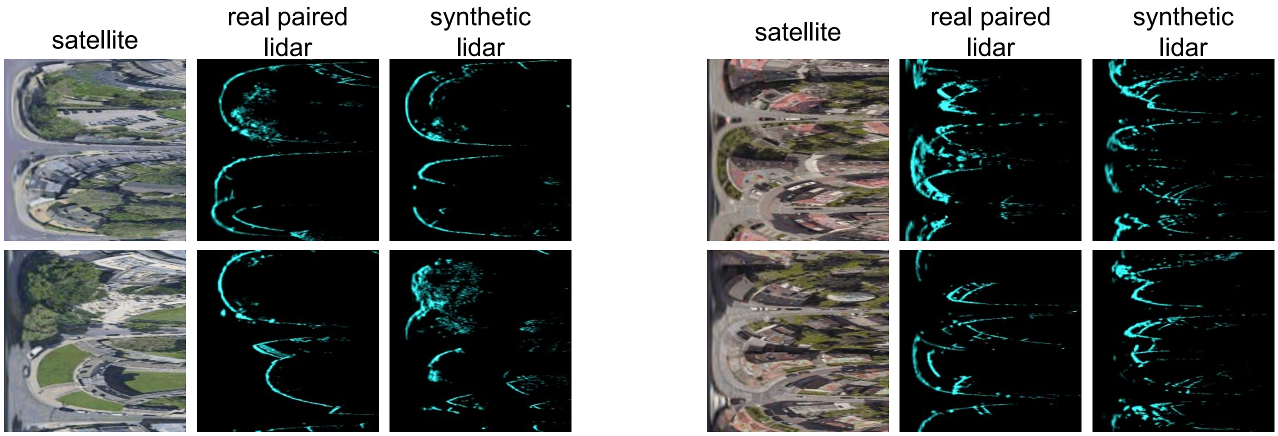


Fig. 5: Comparison between the input satellite image, a real lidar image scanned at the centre of the satellite image (not used during training, for visualisation only), and the synthetic lidar image generated from the satellite image, from the training set trajectories of RobotCar (left) and KITTI (right).

$\mathbb{R}^d$, where closeness in Euclidean space reflects closeness in Cartesian space geometrically. We train the function F in a self-supervised way using Siamese networks with a triplet margin loss. Specifically, given a polar lidar image X_i, we take its temporal neighbour taken τ frames earlier or later, $X_{i\pm\tau}$, to form a positive pair, and take a random sample X^- to form a negative pair, and minimise the following loss:

$$\mathcal{L}_{\text{emb}} = \Big[\|F(X_i) - F(X_{i\pm\tau})\| - \|F(X_i) - F(X^-)\| + m\Big]_+. \tag{1}$$

$[a]_+$ here denotes $\max(a, 0)$, and m is the triplet margin. In our experiments, we set m to 1 and τ to 5. We apply random rotation augmentations to $X_i, X_{i\pm\tau}$, and X^- so that F learns to be rotation invariant.

C. Unpaired Satellite-to-Lidar Translation

To bridge the modality gap between satellite imagery and lidar, we use a variant of CycleGAN applied to the polar images. Specifically, we concatenate each polar lidar or satellite image with an additional single-channel image R, where the values of all pixels on column k of R is $\frac{k}{W}$, with W being the image width. Since height and width now denote azimuth and range values in polar representation, supplying R makes the generator and discriminator aware of the normalised range value of each pixel, and has shown to help stabilise training.

In CycleGAN, we seek to optimise generator functions $G_{X\to Y}, G_{Y\to X}$ and discriminator functions D_X, D_Y, parametrised by neural networks. $G_{X\to Y}$ maps an image from modality $\mathcal{X}$ to $\mathcal{Y}$, while $G_{Y\to X}$ is the counterpart. D_X and D_Y discriminate whether an image in modality $\mathcal{X}$ or $\mathcal{Y}$ is real or fake, respectively.

Given arbitrary X_i and Y_j, an adversarial loss can be applied to $G_{X\to Y}$ and D_Y :

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(G_{X\to Y}, D_Y, X_i, Y_j, R) &= \mathbb{E}_{Y_j\sim\mathcal{Y}} \log D_Y(Y_j, R)\\ &+ \mathbb{E}_{X_i\sim\mathcal{X}} \log\Big(1 - D_Y\big(G_{X\to Y}(X_i, R), R\big)\Big),\end{aligned} \tag{2}$$

and a similar loss is introduced for $G_{Y\to X}$ and D_X.

The cycle-consistency loss can be formulated as:

$$\begin{aligned}\mathcal{L}_{\text{cyc}} &= \mathbb{E}_{X_i\sim\mathcal{X}} \left\|X_i - G_{Y\to X}\big(G_{X\to Y}(X_i, R), R\big)\right\|_1\\ &+ \mathbb{E}_{Y_j\sim\mathcal{Y}} \left\|Y_j - G_{X\to Y}\big(G_{Y\to X}(Y_j, R), R\big)\right\|_1.\end{aligned} \tag{3}$$

Finally, the full loss is:

$$\begin{aligned}\mathcal{L}_{\text{CycleGAN}} &= \mathcal{L}_{\text{GAN}}(G_{X\to Y}, D_Y, X_i, Y_j, R)\\ &+ \mathcal{L}_{\text{GAN}}(G_{Y\to X}, D_X, Y_j, X_i, R) + \lambda\mathcal{L}_{\text{cyc}},\end{aligned} \tag{4}$$

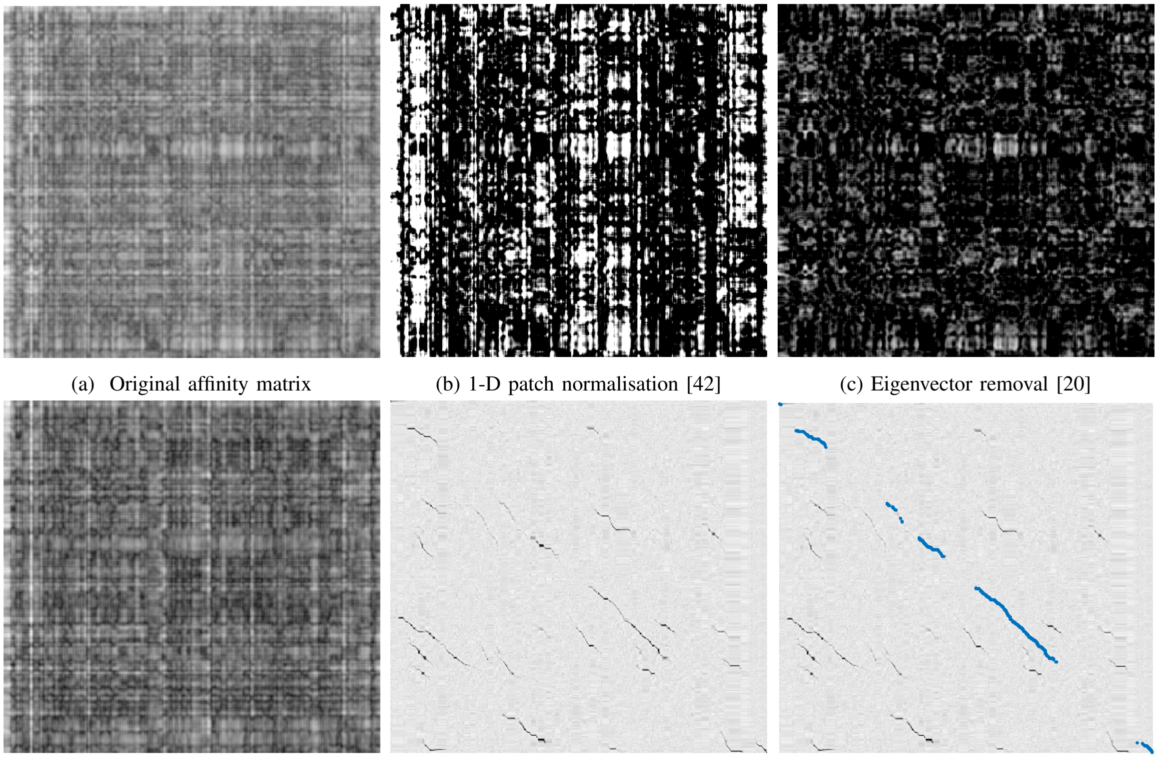
where λ was set to 100 in our experiments. The networks are optimised as:

$$G^*_{X\to Y}, G^*_{Y\to X} = \underset{G_{X\to Y}, G_{Y\to X}}{\operatorname{argmin}} \ \underset{D_X, D_Y}{\operatorname{argmax}} \mathcal{L}_{\text{CycleGAN}}. \tag{5}$$

As the majority of pixels in a lidar image are dark and only a small fraction has range returns, there is no guarantee synthetic lidar images will capture the same regions of the scene as an actual lidar situated at the centre of the satellite image would. Figure 5 shows examples of synthetic lidar images compared to actual lidar images taken at the centre of the satellite image (not used during training in our method). In many cases the synthetic lidar images are visibly significantly different from what an actual lidar would capture, which will result in finding false matches in the descriptor space. As such, relying on unpaired image-to-image translation only is insufficient, unlike the case of radar-to-lidar transfer [62] where locations with strong radar range return will likely also result in lidar return, and pixels with weak or no radar return will likely be not captured in a lidar scan.

D. Sequence Alignment

Though individual nearest-neighbour matching with synthetic lidar images is inadequate, pair-finding can be achieved with sequence alignment. Specifically, given a sequence of lidar images $\mathcal{X} = \{X_1, X_2, \ldots, X_N\}$ and a sequence of satellite images $\mathcal{Y} = \{Y_1, Y_2, \ldots, Y_M\}$ queried along the same route, we first generate synthetic lidar images $\tilde{X}_j = G_{Y\to X}(Y_j)$ using the learned generator, forming a set of synthetic lidar images $\tilde{\mathcal{X}} = \{\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_M\}$. Next, we can



(a) Original affinity matrix (b) 1-D patch normalisation [42] (c) Eigenvector removal [20]

(d) Dimensionality reduction [48, 34] (e) Ours (neural de-aliasing) (f) Pseudo pairs found

Fig. 6: (a): Affinity matrix computed from sequence `2011_10_03_0034` of the KITTI dataset, suffering from high signal aliasing with very poor local contrast. We show 4 methods for de-aliasing the affinity matrix, namely 1-D patch normalisation (b), eigenvector removal (c), dimensionality reduction (d), and our *neural de-aliasing* (e). (f): The highlighted pixels indicate pseudo pairs found from sequence alignment using a modified Smith-Waterman algorithm.

construct an affinity matrix $\mathbf{A} \in \mathbb{R}^{N \times M}$ where each element is the Euclidean distance in descriptor space between X_i and $\tilde{X}_j$ after normalisation:

$$f_i = \frac{F(X_i) - \mu_X}{\sigma_X}, \quad \tilde{f}_j = \frac{F(\tilde{X}_j) - \mu_{\tilde{X}}}{\sigma_{\tilde{X}}} \\ \mathbf{A}_{ij} = \left\| f_i - \tilde{f}_j \right\|, \tag{6}$$

where F is the learned embedding function from Section IV-B. $\mu_X \in \mathbb{R}^d$ and $\sigma_X \in \mathbb{R}^d$ are the mean and standard deviation of $\{F(X_1), \ldots, F(X_N)\}$, and similarly for $\mu_{\tilde{X}}$ and $\sigma_{\tilde{X}}$.

Since synthetic lidar images are not necessarily compatible with real lidar images as described in Section IV-C, the affinity matrix can be heavily corrupted by signal aliasing, resulting in a poor signal-to-noise ratio. Figure 6a shows the affinity matrix for a sequence from the KITTI dataset [17]. Various hand-crafted methods were proposed in prior work to enhance the contrast in the affinity matrix. SeqSLAM [42] proposes 1-D patch normalisation on the affinity matrix. Ho and Newman [20] performed eigendecomposition on the affinity matrix and reconstructed a rank-reduced one by removing eigenvectors corresponding to the largest eigenvalues. The methods in [48, 34] apply dimensionality reduction by keeping the descriptor's most descriptive $k \leq d$ dimensions. These methods were designed primarily for visual place recognition – where the mapping and localising sensors are of the same type and thus the affinity matrix does not suffer from extreme signal aliasing as in our cross-sensory problem.

We propose *neural de-aliasing,* a learned approach for de-aliasing the affinity matrix trained on simulated data. First, we form K random vectors $\{\xi_1, \xi_2, \ldots, \xi_K\}$, where each $\xi_k \in \mathbb{R}^d$ is sampled from a zero-mean Gaussian distribution, and normalised to a unit sphere. Next, we form a random dynamic sequence of length P, $\mathbf{\Phi} = \{\phi_1, \ldots, \phi_P\}$, by travelling from ξ_1 to ξ_K, taking step sizes of $0, 1$, or 2 with various probabilities each. This procedure is repeated, forming a different dynamic sequence of length Q, $\mathbf{\Psi} = \{\psi_1, \ldots, \psi_Q\}$. We can construct an affinity matrix $\mathbf{A}_s \in \mathbb{R}^{P \times Q}$ between $\mathbf{\Phi}$ and $\mathbf{\Psi}$. Here, P and Q can either be less than, equal to, or larger than K. The probability for each step size is a design choice and does not affect performance greatly.

We then add artificial aliasing to corrupt the simulated affinity matrix. We take inspiration from [20], yet, instead of removing eigenvectors, we introduce aliasing by *adding* an eigenvector from the affinity matrix of a KITTI sequence whose real and synthetic lidar embeddings result in heavy aliasing. The detailed procedure is found in Algorithm 1 and a visual example of a simulated affinity matrix $\mathbf{A}_s$ and its alias-corrupted version $\mathbf{A}'_s$ is shown in Figure 7.

We can treat $\mathbf{A}_s$ and $\mathbf{A}'_s$ as single-channel images and apply Pix2Pix [23] to learn to recover a clean affinity matrix from its alias-corrupted counterpart, after appropriate resizing:

$$\mathbf{A}_s = G_{DA}(\mathbf{A}'_s), \tag{7}$$

where $G_{DA} : \mathbb{R}^{H \times W} \rightarrow \mathbb{R}^{H \times W}$ is a matrix de-aliasing function parametrised by a neural network. After G_{DA} is

Algorithm 1: Adding Signal Aliasing

Input:
$\mathbf{A}_s \in \mathbb{R}^{P\times Q}$ # affinity matrix from simulated data
$\{f_1, \ldots, f_N\}$, $\{\tilde{f}_1, \ldots, \tilde{f}_M\}$ # embeddings from KITTI
Output:
$\mathbf{A}'_s$ # affinity matrix with added signal aliasing
Procedure:
$\mathbf{f} \in \mathbb{R}^{d\times(N+M)} \leftarrow \begin{bmatrix} f_1 & \ldots & f_N & \tilde{f}_1 & \ldots & \tilde{f}_M \end{bmatrix}$
$\mathbf{W} \in \mathbb{R}^{(N+M)\times(N+M)} \leftarrow \mathrm{initialise}$
for $i = 1, 2, \cdots, N+M$ **do**
  for $j = 1, 2, \cdots, N+M$ **do**
    # $\mathbf{f}_i$ and $\mathbf{f}_j$ are the i^{th} and j^{th} columns of $\mathbf{f}$
    $\mathbf{W}_{ij} = \|\mathbf{f}_i - \mathbf{f}_j\|$
$\mathbf{\Lambda}, \mathbf{V} \leftarrow \mathrm{eigendecomposition}(\mathbf{W})$
$\lambda^*, \mathbf{v}^* \leftarrow$ largest eigenvalue and the corresponding eigenvector
$\mathbf{v}^* \leftarrow \mathrm{randompermute}(\mathbf{v}^*)$
$\mathbf{N} \in \mathbb{R}^{(N+M)\times(N+M)} \leftarrow \mathbf{v}^*\lambda^*\mathbf{v}^{*\mathrm{T}}$
random crop to a $P \times Q$ patch
$\mathbf{N} \in \mathbb{R}^{P\times Q} \leftarrow \mathrm{randomcrop}(\mathbf{N}, P, Q)$
$\mathbf{A}'_s \leftarrow \mathbf{A}_s + \mathbf{N}$
$\mathbf{A}'_s \leftarrow \mathbf{A}'_s / \max(\mathbf{A}'_s)$

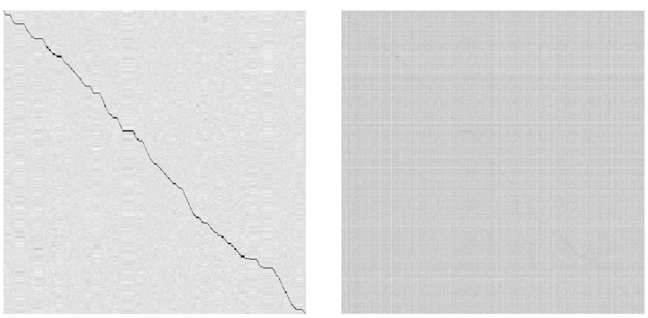


Fig. 7: Example of a simulated affinity matrix $\mathbf{A}_s$ (**left**) and its alias-corrupted version $\mathbf{A}'_s$ (**right**). We train a neural network that takes $\mathbf{A}'_s$ as input and recovers the original, high-contrast affinity matrix $\mathbf{A}_s$.

optimised, we apply G_{DA} to the aliased affinity matrix $\mathbf{A}$ for de-aliasing. Finally, we use the modified Smith-Waterman algorithm in [20] on the de-aliased affinity matrix for sequence alignment, where pseudo pairs are found from the unpaired data, forming a set of pseudo pairs $\mathcal{S}$, with each $(X_p, Y_q) \in \mathcal{S}$ being a pair found by sequence alignment. Figure 6 compares the effect of existing hand-crafted approaches and neural de-aliasing on enhancing the contrast of the affinity matrix. We train neural de-aliasing only once and apply the same learned model to all datasets without further fine-tuning.

E. Learning Place Recognition from Pseudo Pairs

After pseudo pairs are selected from sequence alignment, we use them as positive pairs in metric learning for place recognition. Formally, we wish to learn embedding functions F_X and F_Y that project images of modality $\mathcal{X}$ and $\mathcal{Y}$ respectively to $\mathbb{R}^D$, where F_X and F_Y are parametrised by neural networks. Here F_X and F_Y are separate and different from F as in Section IV-B, which was used for constructing the affinity matrix. To optimise F_X and F_Y, we aim to minimise a bi-directional triplet margin loss:

$$\mathcal{L}_{\mathrm{PR}} = \Big[\|F_X(X_p) - F_Y(Y_q)\| - \big\|F_X(X_p) - F_Y(Y^-)\big\| + m \Big]_+ + \Big[\|F_Y(Y_q) - F_X(X_p)\| - \big\|F_Y(Y_q) - F_X(X^-)\big\| + m \Big]_+, \tag{8}$$

where $(X_p, Y_q) \in \mathcal{S}$, and the negative samples X^- and Y^- are random samples from $\mathcal{X}$ and $\mathcal{Y}$. Here, we set the triplet margin m to 0.1. We again apply random rotation augmentations to the images, so F_X and F_Y learn to be rotation-invariant.

F. Network Architecture and Implementation Details

For the generator networks $G_{X\to Y}, G_{Y\to X}$, and G_{DA}, we use the ResNet [19] generator from the authors' official implementation [1]. For learning embeddings, we use the convolution layers of a VGG16-style [52] backbone followed by a NetVLAD layer [2] for F, and the convolution layers of a ResNet18 backbone, followed by a NetVLAD layer for F_X and F_Y. On lidar and synthetic lidar images, we add a single convolution layer at the top to convert single-channel images to 3 channels, so VGG16 and ResNet18 can consume them. We set the embedding dimensions as $d = 256$ and $D = 2048$.

All of our training is conducted in PyTorch with a batch size of 32. We use RMSProp with a fixed learning rate of 1×10^{-4} in CycleGAN training and ADAM with a fixed learning rate of 2×10^{-4} in all other modules. As no ground truth data are used in training, we cannot split the training data to form a validation set. Instead, we arbitrarily terminate training when the training loss has stabilised for 10 epochs.

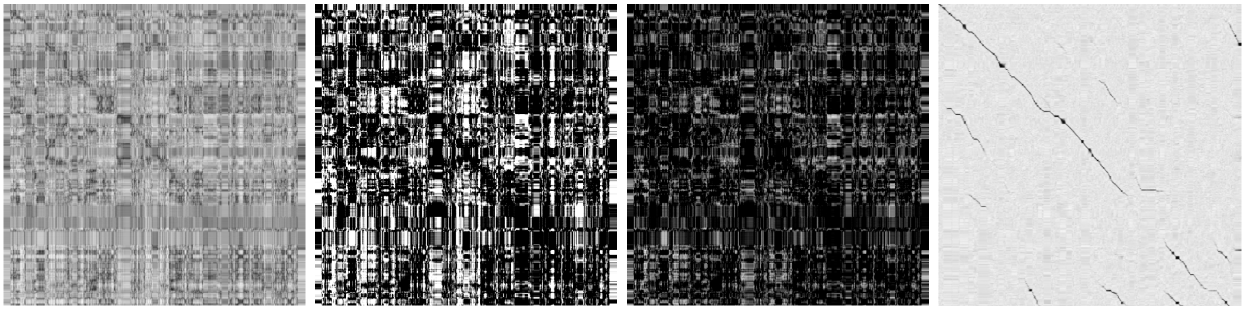
V. Experimental Setup

Our method is validated on the Oxford Radar RobotCar Dataset [4], of which we use the left of the two Velodyne HDL-32E lidars mounted in a tilted configuration, and the KITTI Dataset (raw data) [17], which uses a Velodyne HDL-64E lidar mounted on-top.

Our experiments' trajectories at inference time are unseen during training. As RobotCar features repeated sequences of the same route, we split the trajectory into training and test sets with no overlap, as shown in Figure 9a. We use lidar data collected along the training trajectories from sequences no. 2 and 5 and query satellite data according to the GPS/INS timestamps of sequence no. 7, forming an unpaired training set illustrated in Figure 2 Right.

On KITTI, the training set consists of two long residential sequences, `2011_10_03_0027` and `2011_10_03_0034`. To simulate the scenario where the route is known from a past survey (but never traversed by the lidar-equipped vehicle), we take the paired GPS data but uniformly divide each sequence into 5000 segments based on distance and find the centre of each segment. We then sample satellite images at these centres (as in Figure 2 Middle) with an added random error of $-2\,\mathrm{m}$ to $2\,\mathrm{m}$ to simulate the past survey following a slightly different

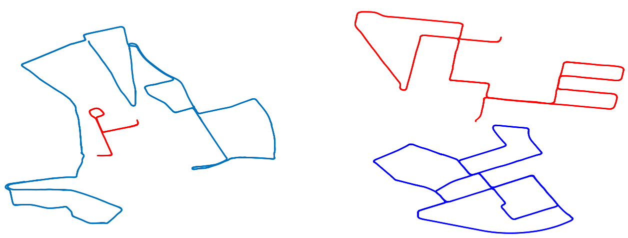
[1]https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix



(a) Original affinity matrix (b) 1-D patch normalisation [42] (c) Eigenvector removal [20] (d) Ours (neural de-aliasing)

Fig. 8: Results of applying *neural de-aliasing* compared to hand-crafted techniques on the affinity matrix of sequence no.2 from RobotCar.

route than when collecting lidar data. The test set consists of another long residential sequence, `2011_09_30_0028`, where we discard the first 1000 frames to avoid overlap with training data, shown in Figure 9b.



(a) RobotCar (b) KITTI

Fig. 9: The RobotCar data were split into training (blue) and test (red). For KITTI, the test set is `2011_09_30_0028` with the first 1000 frames removed (red), as they overlap with `2011_10_03_0027` (blue).

	Ours	Paired baseline	Unpaired baseline	PointLoc [56]
Paired data	✗	✓	✗	✓
Metric $SE(2)$ ground truth	✗	✗	✗	✓

TABLE I: Training data requirements for each evaluated method.

A. Retrieval Accuracy

All test set satellite images $\{Y_j\}$ are mapped to the descriptor space, forming $\{F_Y(Y_j)\}$ where $F_Y(Y_j) \in \mathbb{R}^D \; \forall j$. For each lidar image X_i at test time, we map it to descriptor space as $F_X(X_i) \in \mathbb{R}^D$, and calculate the Euclidean distances against all satellite data descriptors. We find the top-1 satellite image match and the top 1% matches for each lidar image based on descriptor Euclidean distance. This solves the vehicle's place recognition as each satellite image is associated with an x-y position. We compute the percentage of retrievals within a certain threshold to the lidar's ground truth position. In this evaluation, we consider single-frame localisations only and do not use sequential information.

B. Precision and Recall

We sample various descriptor distance thresholds from the minimum distance between each $F_X(X_i)$ to any $F_Y(Y_j)$ to the maximum, and consider matches less than the threshold positives and the rest negatives. Matches (positives or negatives) are considered true matches if the error to the ground truth position is less than $50\,\mathrm{m}$ and false if it is more than $75\,\mathrm{m}$. The precision and recall for all descriptor distance thresholds is computed to generate the Precision-Recall curves.

C. Baselines

Learning lidar-only place recognition in satellite imagery maps is relatively unstudied, and we compare against the following baselines, whose training data requirements are summarised in Table I.

Paired baseline: We trained the same network as ours with the same metric learning approach as in Section IV-E, except with paired training data. Although the paired lidar and satellite images do not need to be perfectly aligned at pixel level, for example the heading ground truth is not required.

Unpaired baseline: This baseline method uses the descriptor embedding function F trained on real lidar images only as in Section IV-B, and directly operates on synthetic lidar images generated from satellite images using the approach in Section IV-C. Specifically, for the unpaired baseline, all test set satellite images are mapped to descriptor space as $\{F\big(G_{Y\to X}(Y_j)\big)\}$, and each lidar image at test time is mapped to descriptor space as $F(X_i)$. The unpaired baseline is similar in spirit as [62] but applied to lidar-satellite place recognition and makes no attempt at sequence alignment.

PointLoc: To the best of our efforts, we implement [56] where satellite images are converted to 2-D points for comparison against lidar data, which we dub PointLoc. PointLoc requires the data to be paired *and* fully aligned at pixel-level through accurate $SE(2)$ ground truth, including heading.

VI. Experimental Results

A. RobotCar Dataset

The affinity matrix for sequence no.2 of RobotCar is shown in Figure 8a. It has much less signal aliasing than the KITTI dataset (Figure 6a). Though hand-crafted techniques such as 1-D patch normalisation and eigenvector removal can increase local contrast to a sufficient extent, we show qualitatively in Figure 8 that our learned method reduces aliasing even further.

The test set trajectory of the RobotCar dataset features approximately $1\,\mathrm{km}$ of urban environment, with 800 lidar frames sampled at $4\,\mathrm{Hz}$. The retrieval performances of our method compared to the baselines are shown in Table II. Though slightly outperformed by the supervised, paired method, almost half of our top-1 retrievals can localise correctly within $60\,\mathrm{m}$ of the true position. Figure 10 shows the Precision-Recall curve, where our method has higher precision than using paired data for recall between 2% and 10%. PointLoc greatly outperforms the other methods, but has the strictest requirements for training data ground truth.

Distance (m)	10	20	30	40	50	60
Top-1						
Paired baseline	30.88	43.50	47.13	48.75	49.75	50.88
Unpaired baseline	8.25	17.00	19.63	20.75	21.75	22.50
PointLoc [56]	**48.00**	**57.00**	**60.63**	**63.75**	**66.50**	**68.88**
Ours	17.00	32.75	37.13	41.25	43.63	45.50
Top 1%						
Paired baseline	29.91	42.70	47.52	49.21	50.77	52.98
Unpaired baseline	6.84	12.55	14.58	16.21	17.31	18.44
PointLoc [56]	**43.14**	**54.15**	**59.29**	**63.22**	**66.46**	**69.33**
Ours	20.05	29.93	34.83	38.74	41.08	43.20

TABLE II: Percentage of top-1 and top 1% retrievals within each distance threshold to the true position, evaluated on the RobotCar Dataset.

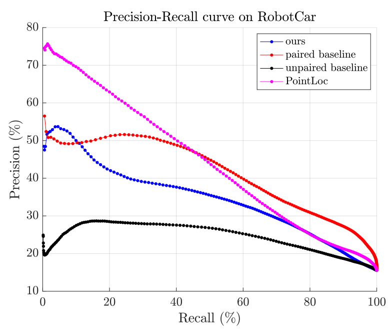


Fig. 10: Precision-Recall curve on the test set of RobotCar.

B. KITTI Dataset

The test set trajectory of KITTI features approximately $4\,\mathrm{km}$ of traversal with more than 4000 lidar frames at $10\,\mathrm{Hz}$. KITTI features much more challenging data than RobotCar for lidar place recognition using overhead imagery. Firstly, KITTI uses a Velodyne HDL-64E lidar compared to Velodyne HDL-32E. Though the higher vertical resolution is ideal for many lidar-based applications, it increases complexity in the resulting lidar images, making it less likely for the synthetic lidar images to be compatible with real lidar scans. This has made it extremely challenging to perform sequence alignment from unpaired data, as indicated by the significantly higher levels of aliasing (Figure 6a) than RobotCar (Figure 8a). Moreover, the test set is in a residential area with many similar places and fewer distinct landmarks, resulting in a high false positive rate. This is further exaggerated when satellite images are expressed as points where several distinctive image features are lost, as demonstrated by the reduced performance of PointLoc. Finally, our test set of KITTI features a longer trajectory, having a much larger satellite map database to search from, inherently making retrieval difficult.

The retrieval performance and Precision-Recall curve are shown in Table III and Figure 11. Even the paired baseline struggles in this environment; nevertheless, our method outperforms the unpaired baseline and, notably, PointLoc.

Distance (m)	10	20	30	40	50	60
Top-1						
Paired baseline	**3.42**	**5.89**	**7.90**	**10.61**	**12.09**	**13.31**
Unpaired baseline	1.41	2.18	3.02	3.38	4.00	5.41
PointLoc [56]	3.08	3.69	4.91	6.15	7.42	8.41
Ours	2.61	4.93	7.02	8.93	9.87	10.92
Top 1%						
Paired baseline	**3.58**	**6.34**	**8.39**	**10.34**	**11.83**	**13.12**
Unpaired baseline	1.17	1.96	2.72	3.46	4.22	4.89
PointLoc [56]	2.68	3.93	5.08	6.18	7.35	8.48
Ours	1.89	3.83	5.43	6.92	8.04	9.01

TABLE III: Percentage of top-1 and top 1% retrievals within each distance threshold to the true position, evaluated on the KITTI Dataset.

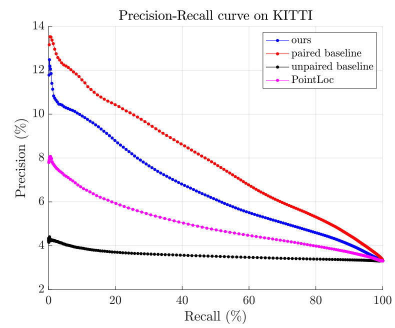


Fig. 11: Precision-Recall curve on the test set of KITTI.

C. Monte Carlo Localisation

Though single frame retrieval using only overhead imagery has limited accuracy, given a stream of lidar data, here we present a Monte Carlo Localisation (MCL) pipeline that can accurately track the pose over long distances. Assuming the route to be taken at test time is known, we can formulate the localisation as a 1-D problem along the known route, where the distance along the route parametrises the state.

Specifically, at time t, the state consists of J particles $\mathbf{P}^t = \{p_1^t, \dots, p_J^t \mid p_j^t \in \mathbb{R}\}$ denoting the distance from the starting point of the trajectory, and the 1-D parametrisation indexes to a 2-D x-y position in the world. We use 2000 particles in our experiments, uniformly initialised along the trajectory. The detailed MCL update step is shown in Algorithm 2.

1) Motion Model: A crucial step in MCL is the motion update, typically provided by vehicle control input or odometry. To demonstrate a pipeline that uses only lidar place recognition in overhead imagery with no need for accurate odometry, we update the motion by sampling the velocity from a uniform mean Gaussian distribution with mean μ_v and standard deviation σ_v. We choose μ_v based on prior knowledge about the vehicle's speed. For the RobotCar Dataset, we set μ_v to $20\,\mathrm{km/h}$ as each sequence features approximately a $10\,\mathrm{km}$ trajectory collected in around 30 minutes. For KITTI, there is no repeated traversal, so we set μ_v to $30\,\mathrm{km/h}$, which is the speed limit in residential areas in Germany. Finally, as the motion update is entirely approximated based on a constant-speed prior, which may not represent the vehicle's true motion at time t, we use a large value of σ_v corresponding to $10\,\mathrm{m/s}$ for both datasets to account for the noise correctly.

2) Measurement Model: For each particle p_j^t, its associated x-y position is used to query the nearest satellite image. The corresponding satellite image is mapped to descriptor space using F_Y, and compared to the live lidar image at time t, X^t.

3) Results: We compute the median of $\mathbf{P}^t$ to find the vehicle's estimated distance along the trajectory, and thus its x-y position, at time t. The error to the ground truth position is plotted for RobotCar in Figure 12 and KITTI in Figure 13

Algorithm 2: 1-D Monte Carlo Localisation

function MCL $(\mathbf{P}^{t-1}, \mu_v, \sigma_v, X^t)$
$\bar{\mathbf{P}}^t = \mathbf{P}^t = \mathbf{S}^t = \emptyset$
for $j = 1, 2, \dots, J$ **do**
  sample v from $V \sim \mathcal{N}(\mu_v, \sigma_v)$
  $p_j^t \leftarrow p_j^{t-1} + v \cdot \Delta t$
  $Y_j^t \leftarrow$ nearest satellite image to p_j^t
  $s_j^t \leftarrow \frac{F_X(X^t) \cdot F_Y(Y_j^t)}{\|F_X(X^t)\| \|F_Y(Y_j^t)\|}$ # cosine similarity
  $\bar{\mathbf{P}}^t \leftarrow \bar{\mathbf{P}}^t \oplus p_j^t$ # add to set
  $\mathbf{S}^t \leftarrow \mathbf{S}^t \oplus s_j^t$
$\mathbf{W}^t \leftarrow \mathrm{softmax}(\mathbf{S}^t)$
for $j = 1, 2, \dots, J$ **do**
  draw p_j^t from $\bar{\mathbf{P}}^t$ with probability w_j^t
  $\mathbf{P}^t \leftarrow \mathbf{P}^t \oplus p_j^t$
return $\mathbf{P}^t$

for our method and the paired baseline. On RobotCar, the particles quickly converged after about 50 frames and localised with small errors afterwards. On KITTI, though MCL lost track near the end of the trajectory, it successfully localised to mostly under $50\,\mathrm{m}$ error for over $2\,\mathrm{km}$, relying solely on comparing lidar data against overhead imagery. Notably, the localisation accuracy of our method is on par with the paired baseline under an MCL set up.

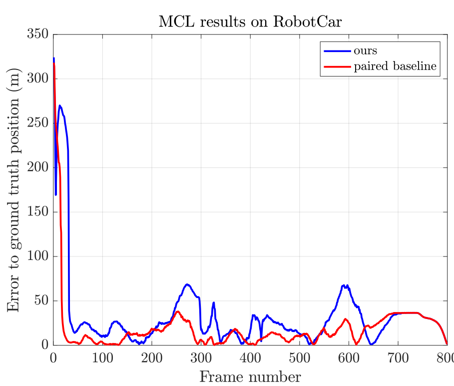


Fig. 12: MCL results on RobotCar.

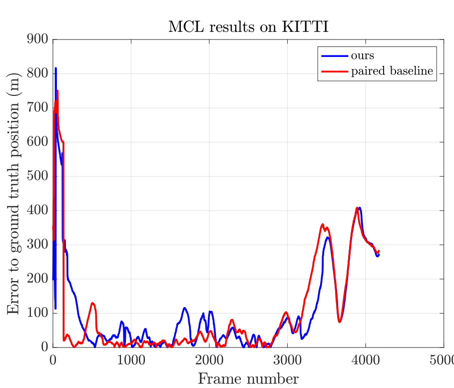


Fig. 13: MCL results on KITTI.

D. Unpaired Radar Place Recognition in Overhead Imagery

Our method was designed for lidar data but can be applied to radar data through radar-to-lidar image translation. Given radar images $\mathcal{Z} = \{Z_1, Z_2, \dots\}$, we apply CycleGAN [64] to learn a function $G_{Z \to X} : \mathbb{R}^{H \times W} \to \mathbb{R}^{H \times W}$ that maps a polar radar image to its synthetic lidar counterpart. Figure 14 shows examples of radar images and the corresponding synthetic lidar images after style transfer on the RobotCar dataset, which also features an on-board Navtech radar.

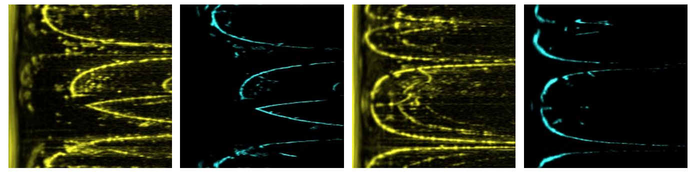


Fig. 14: Radar images (yellow) and corresponding synthetic lidar images.

Distance (m)	10	20	30	40	50	60
Top-1						
Paired baseline	**11.25**	16.25	22.88	30.88	36.88	**46.13**
Ours	9.75	**22.38**	**29.75**	**36.25**	**41.50**	45.50
Top 1%						
Paired baseline	**10.70**	17.83	23.95	31.47	37.80	45.54
Ours	9.26	**22.12**	**30.29**	**36.02**	**41.32**	**45.72**

TABLE IV: Percentage of top-1 and top 1% retrievals within each distance threshold to the true position, evaluated on RobotCar for radar data.

Then, without further retraining, we utilise networks learned with lidar data from our unpaired approach and directly apply them to synthetic lidar images from radar input. Specifically, taking F_X, F_Y trained for lidar-satellite place recognition with found pseudo pairs, at test time, each radar image Z_i is mapped to descriptor space as $F_X\big(G_{Z \to X}(Z_i)\big)$, and compared against the map database of satellite image descriptors $\{F_Y(Y_j)\}$. We compare our approach against a supervised method trained on paired radar and satellite images, using the metric learning approach in Section IV-E. The retrieval performance is shown in Table IV. Notably, the unpaired approach has better accuracy than the supervised approach trained with paired radar and satellite data.

VII. Conclusion and Future Work

In this paper, we show how lidar place recognition in a map of satellite images can be solved without any paired data. Our approach relaxes the need for an on-board, time-synchronised GPS/INS when collecting on-road lidar data for high-precision pose ground truth, as long as the route taken is known from a previous survey or traversal. Though the performance of place recognition in overhead imagery is far from lidar-to-lidar localisation, this capability nevertheless allows a mobile robot to travel to an unvisited place and still localise to a certain extent, and can also add an extra layer of redundancy in standard navigation applications.

Here we focus on solving the pairing problem from unpaired data and rely on metric learning for solving place recognition. Using data with global pose ground truth, future work can target how to further bridge the domain gap between lidar data and overhead imagery to enhance the localisation quality.

References

[1] Amjad Almahairi, Sai Rajeshwar, Alessandro Sordoni, Philip Bachman, and Aaron Courville. Augmented Cyclegan: Learning Many-to-Many Mappings from Unpaired Data. In *International Conference on Machine Learning*, pages 195–204. PMLR, 2018.

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[3] Dan Barnes and Ingmar Posner. Under the Radar: Learning to Predict Robust Keypoints for Odometry Estimation and Metric Localisation in Radar. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Paris, 2020. URL https://arxiv.org/abs/2001.10789.

[4] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The Oxford Radar Robotcar dataset: A Radar Extension to the Oxford Robotcar Dataset. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6433–6438. IEEE, 2020.

[5] Michael Bosse and Robert Zlot. Place Recognition Using Keypoint Voting in Large 3D Lidar Datasets. In *2013 IEEE International Conference on Robotics and Automation*, pages 2677–2684. IEEE, 2013.

[6] Marcus A Brubaker, Andreas Geiger, and Raquel Urtasun. Lost! Leveraging the Crowd for Probabilistic Visual Self-localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3057–3064, 2013.

[7] Xieyuanli Chen, Thomas Läbe, Andres Milioto, Timo Röhling, Olga Vysotska, Alexandre Haag, Jens Behley, and Cyrill Stachniss. OverlapNet: Loop Closing for LiDAR-based SLAM. In *Proceedings of Robotics: Science and Systems*, Corvalis, Oregon, USA, July 2020. doi: 10.15607/RSS.2020.XVI.009.

[8] Younghun Cho, Giseop Kim, Sangmin Lee, and Jee-Hwan Ryu. OpenStreetMap-based LiDAR Global Localization in Urban Environment without a Prior LiDAR Map. *IEEE Robotics and Automation Letters*, 7(2):4999–5006, 2022.

[9] Yunge Cui, Xieyuanli Chen, Yinlong Zhang, Jiahua Dong, Qingxiao Wu, and Feng Zhu. BoW3D: Bag of Words for Real-Time Loop Closing in 3D LiDAR SLAM. *IEEE Robotics and Automation Letters*, 2022.

[10] Lucas de Paula Veronese, Edilson de Aguiar, Rafael Correia Nascimento, Jose Guivant, Fernando A Auat Cheein, Alberto Ferreira De Souza, and Thiago Oliveira-Santos. Re-emission and Satellite Aerial Maps Applied to Vehicle Localization on Urban Environments. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4285–4290. IEEE, 2015.

[11] Can Ulas Dogruer, A Bugra Koku, and Melik Dolen. Outdoor Mapping and Localization Using Satellite Images. *Robotica*, 28(7):1001–1012, 2010.

[12] Juan Du, Rui Wang, and Daniel Cremers. DH3D: Deep Hierarchical 3D Descriptors for Robust Large-Scale 6DoF Relocalization. In *European Conference on Computer Vision*, pages 744–762. Springer, 2020.

[13] Renaud Dubé, Daniel Dugas, Elena Stumm, Juan Nieto, Roland Siegwart, and Cesar Cadena. Segmatch: Segment Based Place Recognition in 3d Point Clouds. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5266–5272. IEEE, 2017.

[14] Florian Fervers, Sebastian Bullinger, Christoph Bodensteiner, Michael Arens, and Rainer Stiefelhagen. Continuous Self-Localization on Aerial Images Using Visual and Lidar Sensors. *arXiv preprint arXiv:2203.03334*, 2022.

[15] Georgios Floros, Benito Van Der Zander, and Bastian Leibe. OpenStreetSLAM: Global Vehicle Localization Using OpenStreetMaps. In *2013 IEEE International Conference on Robotics and Automation*, pages 1054–1059. IEEE, 2013.

[16] Matthew Gadd, Daniele De Martini, and Paul Newman. Contrastive Learning for Unsupervised Radar Place Recognition. In *2021 20th International Conference on Advanced Robotics (ICAR)*, pages 344–349, 2021. doi: 10.1109/ICAR53236.2021.9659335.

[17] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision Meets Robotics: The KITTI Dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[18] Jiadong Guo, Paulo VK Borges, Chanoh Park, and Abel Gawel. Local Descriptor for Robust Place Recognition Using Lidar Intensity. *IEEE Robotics and Automation Letters*, 4(2):1470–1477, 2019.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[20] Kin Leong Ho and Paul Newman. Detecting Loop Closure with Scene Sequences. *International journal of computer vision*, 74(3):261–286, 2007.

[21] Sixing Hu, Mengdan Feng, Rang MH Nguyen, and Gim Hee Lee. CVM-Net: Cross-View Matching Network for Image-Based Ground-to-Aerial Geo-Localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7258–7267, 2018.

[22] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal Unsupervised Image-to-Image Translation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 172–189, 2018.

[23] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-Image Translation with Conditional Adversarial Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017.

[24] Somi Jeong, Seungryong Kim, Kihong Park, and

Kwanghoon Sohn. Learning to Find Unpaired Cross-Spectral Correspondences. *IEEE Transactions on Image Processing*, 28(11):5394–5406, 2019.

[25] Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. Minimum Class Confusion for Versatile Domain Adaptation. In *European Conference on Computer Vision*, pages 464–480. Springer, 2020.

[26] Giseop Kim and Ayoung Kim. Scan Context: Egocentric Spatial Descriptor for Place Recognition within 3d Point Cloud Map. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4802–4809. IEEE, 2018.

[27] Jonghwi Kim, Yonghoon Cho, and Jinwhan Kim. Urban Localization Based on Aerial Imagery by Correcting Projection Distortion. *Autonomous Robots*, pages 1–14, 2022.

[28] Jacek Komorowski. Minkloc3d: Point Cloud Based Large-Scale Place Recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1790–1799, 2021.

[29] Rainer Kümmerle, Bastian Steder, Christian Dornhege, Alexander Kleiner, Giorgio Grisetti, and Wolfram Burgard. Large Scale Graph-based SLAM using Aerial Images as Prior Information. *Autonomous Robots*, 30 (1):25–39, 2011.

[30] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse Image-to-Image Translation via Disentangled Representations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 35–51, 2018.

[31] Tsung-Yi Lin, Serge Belongie, and James Hays. Cross-View Image Geolocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898, 2013.

[32] Liu Liu and Hongdong Li. Lending Orientation to Neural Networks for Cross-View Geo-Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5624–5633, 2019.

[33] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised Image-to-Image Translation Networks. In *Advances in Neural Information Processing Systems*, pages 700–708, 2017.

[34] Yang Liu and Hong Zhang. Visual Loop Closure Detection with a Compact Image Descriptor. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1051–1056. IEEE, 2012.

[35] Zhe Liu, Shunbo Zhou, Chuanzhe Suo, Peng Yin, Wen Chen, Hesheng Wang, Haoang Li, and Yun-Hui Liu. LPD-Net: 3D Point Cloud Learning for Large-Scale Place Recognition and Environment Analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2831–2840, 2019.

[36] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep Transfer Learning with Joint Adaptation Networks. In *International conference on machine learning*, pages 2208–2217. PMLR, 2017.

[37] Sha Lu, Xuecheng Xu, Li Tang, Rong Xiong, and Yue Wang. DeepRING: Learning Roto-Translation Invariant Representation for LiDAR based Place Recognition. *arXiv preprint arXiv:2210.11029*, 2022.

[38] Lun Luo, Si-Yuan Cao, Bin Han, Hui-Liang Shen, and Junwei Li. BVMatch: LiDAR-Based Place Recognition Using Bird's-Eye View Images. *IEEE Robotics and Automation Letters*, 6(3):6076–6083, 2021.

[39] Junyi Ma, Xieyuanli Chen, Jingyi Xu, and Guangming Xiong. SeqOT: A Spatial-Temporal Transformer Network for Place Recognition Using Sequential LiDAR Data. *arXiv preprint arXiv:2209.07951*, 2022.

[40] Junyi Ma, Jun Zhang, Jintao Xu, Rui Ai, Weihao Gu, and Xieyuanli Chen. OverlapTransformer: An Efficient and Yaw-Angle-Invariant Transformer Network for LiDAR-Based Place Recognition. *IEEE Robotics and Automation Letters*, 2022.

[41] Nate Merrill and Guoquan Huang. Lightweight Unsupervised Deep Loop Closure. 2018.

[42] Michael J Milford and Gordon F Wyeth. SeqSLAM: Visual Route-Based Navigation for Sunny Summer Days and Stormy Winter Nights. In *2012 IEEE international conference on robotics and automation*, pages 1643–1649. IEEE, 2012.

[43] Pilailuck Panphattarasap and Andrew Calway. Automated Map Reading: Image Based Localisation in 2-d Maps Using Binary Semantic Descriptors. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6341–6348. IEEE, 2018.

[44] Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive Learning for Unpaired Image-to-Image Translation. In *European conference on computer vision*, pages 319–345. Springer, 2020.

[45] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *arXiv preprint arXiv:1706.02413*, 2017.

[46] Stefan Saftescu, Matthew Gadd, Daniele De Martini, Dan Barnes, and Paul Newman. Kidnapped Radar: Topological Radar Localisation using Rotationally-Invariant Metric Learning. *arXiv preprint arXiv: 2001.09348*, 2020.

[47] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum Classifier Discrepancy for Unsupervised Domain Adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018.

[48] Stefan Schubert, Peer Neubert, and Peter Protzel. Unsupervised Learning Methods for Visual Place Recognition in Discretely and Continuously Changing Environments. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4372–4378. IEEE, 2020.

[49] Yujiao Shi, Liu Liu, Xin Yu, and Hongdong Li. Spatial-Aware Feature Aggregation for Image Based Cross-View Geo-Localization. *Advances in Neural Information Processing Systems*, 32, 2019.

[50] Yujiao Shi, Xin Yu, Dylan Campbell, and Hongdong Li. Where Am I Looking At? Joint Location and Orientation Estimation by Cross-View Matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4064–4072, 2020.

[51] Yujiao Shi, Xin Yu, Liu Liu, Tong Zhang, and Hongdong Li. Optimal Feature Transport for Cross-View Image Geo-:ocalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11990–11997, 2020.

[52] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proceedings of the 3rd International Conference on Learning Representations(ICLR)*, 2015.

[53] Li Sun, Daniel Adolfsson, Martin Magnusson, Henrik Andreasson, Ingmar Posner, and Tom Duckett. Localising Faster: Efficient and Precise Lidar-Based Robot Localisation in Large-Scale Environments. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4386–4392. IEEE, 2020.

[54] T. Y. Tang, D. De Martini, D. Barnes, and P. Newman. RSL-Net: Localising in Satellite Images From a Radar on the Ground. *IEEE Robotics and Automation Letters*, 5(2):1087–1094, April 2020. ISSN 2377-3774. doi: 10.1109/LRA.2020.2965907.

[55] Tim Y Tang, Daniele De Martini, Shangzhe Wu, and Paul Newman. Self-Supervised Localisation between Range Sensors and Overhead Imagery. In *Robotics: Science and Systems (RSS) XVI*, 2020.

[56] Tim Y. Tang, Daniele De Martini, and Paul Newman. Get to the Point: Learning Lidar Place Recognition and Metric Localisation Using Overhead Imagery. In *Proceedings of Robotics: Science and Systems*, Virtual, July 2021. doi: 10.15607/RSS.2021.XVII.003.

[57] Mikaela Angelina Uy and Gim Hee Lee. PointNetVLAD: Deep Point Cloud Based Retrieval for Large-Scale Place Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4470–4479, 2018.

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, 2017.

[59] Kavisha Vidanapathirana, Peyman Moghadam, Ben Harwood, Muming Zhao, Sridha Sridharan, and Clinton Fookes. Locus: Lidar-Based Place Recognition Using Spatiotemporal Higher-Order Pooling. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5075–5081. IEEE, 2021.

[60] Kavisha Vidanapathirana, Milad Ramezani, Peyman Moghadam, Sridha Sridharan, and Clinton Fookes. LoGG3D-Net: Locally Guided Global Descriptor Learning for 3D place Recognition. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2215–2221. IEEE, 2022.

[61] Fan Yan, Olga Vysotska, and Cyrill Stachniss. Global Localization on OpenStreetMap Using 4-bit Semantic Descriptors. In *2019 European Conference on Mobile Robots (ECMR)*, pages 1–7. IEEE, 2019.

[62] Huan Yin, Yue Wang, Jun Wu, and Rong Xiong. Radar Style Transfer for Metric Robot Localisation on Lidar Maps. *CAAI Transactions on Intelligence Technology*, 2022.

[63] Wenxiao Zhang and Chunxia Xiao. PCAN: 3D Attention Map Learning Using Contextual Information for Point Cloud Based Retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12436–12445, 2019.

[64] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.

[65] Minzhao Zhu, Yi Yang, Wenjie Song, Meiling Wang, and Mengyin Fu. AGCV-LOAM: Air-Ground Cross-View Based LiDAR Odometry and Mapping. In *2020 Chinese Control And Decision Conference (CCDC)*, pages 5261–5266. IEEE, 2020.