2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Geolocation on Cartographic Maps with Multi-Modal Fusion

Mengjie Zhou[1], Liu Liu[2], Yiran Zhong[3], and Andrew Calway[1]

***Abstract*— We explore the geolocation problem, aiming to localize ground-view images on cartographic maps, without the need of any GPS priors. This task mimics the human wayfinding ability and offers high scalability and robustness by using the compact and semantic representations of maps. Current methods often rely on 2D maps to encode dense contextual information for ground-to-map matching. In this paper, we lift ground-to-map matching to a 2.5D space, where heights of structures (e.g. buildings) provide richer geometric information to guide the matching process. We propose a new approach to learning representative embeddings from multi-modal data. Specifically, we establish a projection relationship between 2D and 2.5D space. The projection is further used to combine multi-modal features from the 2D and 2.5D maps using an effective pixel-to-point fusion method. By encoding crucial geometric cues, our method learns discriminative location embeddings for matching panoramic images and maps. Additionally, we construct the first large-scale multi-modal geolocation dataset to validate our method and facilitate future research. Both single-image based and route based geolocation experiments are conducted to test our method. Extensive experiments demonstrate that the proposed method achieves significantly higher geolocation accuracy and faster convergence than previous 2D map-based approaches.**

I. Introduction

We study the problem of geolocation using cartographic maps. Through linking the semantic information on geo-referenced maps to the content in ground-view images, the geographic coordinates are determined at which the images are captured, without relying on coarse pose priors or online GPS signals. This is akin to human wayfinding ability that uses survey maps or You Are Here schematic maps to locate and navigate in complex 3D world. It differs from previous geolocation methods, which involve matching location images with a large-scale database of geo-referenced images, either ground or aerial/satellite [1], [2].

The cartographic map is typically defined by spatial arrangement of semantic entities, such as roads, buildings, and rivers, etc. Using maps as the reference source offers manifold benefits over geo-referenced images. First, maps provide compact and semantic representations that are largely independent of capture time and dynamic objects, and hence offer the potential for more robust matching. On contrary, image-based methods have to deal with changes in appearance over long timescales and in the presence of dynamic objects. Second, maps are ubiquitous and freely available for most of the planet, contrasting with the difficulty in obtaining sufficient ground images. Motivated by these, an increasing number of works are exploring map-related tasks, processing cartographic maps in various forms such as raw data, 2D tiles or 2.5D models.

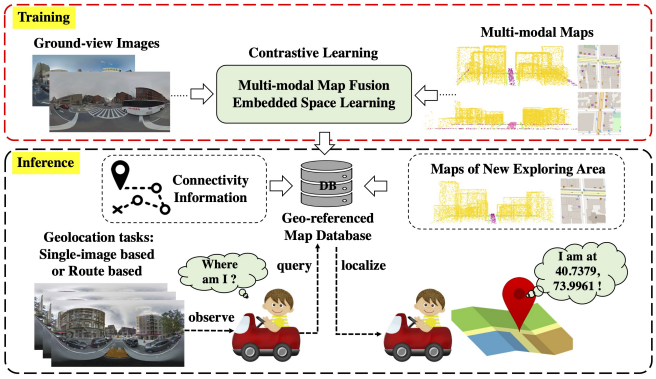

Fig. 1: Illustration of querying multi-modal maps for geolocation. During the training phase, the precollected ground-view images and multi-modal maps are fed to our contrastive learning framework, fusing multi-modal maps and learning image and map embeddings. In the testing stage, we collect multi-modal maps covering the area of interest and use our learned network to extract map embeddings to build a geo-referenced database. Embeddings of query images are matched to the database to estimate geo-locations. The connectivity within multi-modal maps are used for route-based geolocation.

In this paper, we explore geolocation by incorporating multi-modal maps to leverage richer contextual and geometric information for greater discrimination. This process involves querying a ground-view image with respect to a large-scale and geo-referenced multi-modal map database consisting of 2D map tiles and 2.5D map models. An example scenario is illustrated in Figure 1. The 2D map tiles are rendered images that depict small local regions within the localized area. The 2.5D map model refers to an untextured 3D model constructed by augmenting a 2D cartographic map with height information. Compared with detailed 3D models, the 2.5D model is compact, easily obtained, and still contains sufficient structural information for geolocation. It is noteworthy that 2.5D maps are now accessible through the majority of map services, including Google Maps and OpenStreetMap (OSM) [3]. Given that 2D maps encompass dense contextual information while 2.5D maps provide explicit geometric information, our approach integrates multiple map modalities which enables the capture of complementary information, leading to more robust and discriminative representations of the localized area.

Geolocation with multi-modal maps presents inherent challenges due to the substantial differences in appearance

[1] Mengjie Zhou and Andrew Calway are with the University of Bristol, UK. [mengjie.zhou, andrew.calway]@bristol.ac.uk.
[2] Liu Liu is with Huawei KooMap Dept., Beijing, China. liuliu33@huawei.com
[3] Yiran Zhong is with Shanghai AI Lab, Shanghai, China. corresponding author, zhongyiran@gmail.com.

between images and maps, as well as the necessity for effective feature fusion across the 2D and 2.5D map domains. To address these, our approach proposes fusing 2D and 2.5D maps within the same feature space using effective fusion techniques. We examine various methods and discover that pixel-to-point fusion delivers superior performance. Specifically, we first design a pipeline to automatically extract 2.5D map models from OSM and convert them to point clouds using the surface sampling strategy. We then build a Triplet-like architecture with an metric learning loss to learn an embedded space for intra- and inter-modal discrimination.

For training and evaluation, we built a large-scale geolocation dataset, containing 113,767 panoramic images and geo-tagged multi-modal maps for areas of New York and Pittsburgh. We perform both single-image based and route based geolocation to validate our method. Experiments show that our multi-modal map based method outperform previous method [4]. In summary, the main contributions are:

- A large-scale multi-modal dataset for geolocation, comprising pairs of panoramic images, 2D map tiles and 2.5D map models, covering cities;
- A simple yet effective network architecture for fusing and linking multi-modal features in an embedded space;
- State-of-the-art performance for two subsequent tasks, i.e., single-image based and route based geolocation.

II. Related Work

Map-based Geolocation Map data, such as OSM, has been used previously for geolocation. One of the first examples was that by Panphattarasap et al. [5] in which a compact 4-bit descriptor indicating the presence or not of semantic features (junctions and building gaps) is used to link OSM raw data to ground-view images to achieve matching and hence geolocation. Follow on work by Samano et al. [4], [6] generalized the approach in [5] by linking images to 2D map tiles using a learned embedded latent space, achieving significantly better performance and greater efficiency by incorporation into a particle filtering framework. Vojir et al. [7] also extended the approach by exploiting depth and building instance labels in the map data to avoid the need for co-located image-map tile pairs, although the reliance on specific semantic features reduces generality.

Others have looked at combining GPS priors with map based geolocation. For example, Sarlin et al. [8] introduced a finer geolocation approach that achieves sub-meter accuracy by matching neural bird's-eye-view (BEV) representations of images with neural maps derived from 2D semantic maps. Armagan et al. [9] and Hai et al. [10] achieved finer geolocation using 2.5D building maps. They employ either rendering of 2.5D maps for 2D-2D alignment or directly implement 2D-3D alignment. These methods have shown that increased accuracy and resolution can be achieved by using GPS. However, although GPS is low-cost and readily available, it is susceptible to loss of accuracy in challenging environments such as urban canyons and its use means a reliance on an external infrastructure, which may not be appropriate for all applications.

Multi-modal Fusion To fuse 2D and 2.5D maps, we examine multi-modal fusion in the image-point cloud domain. Wang et al. [11] proposed decorating point cloud with corresponding image features at early stage. Li et al. [12] compared early-stage and mid-level fusion, demonstrating improved performance by fusing image features with deep point cloud features, rather than raw points. Similarly, Bai et al. [13] highlighted the performance drop induced by bad illumination and sensor misalignment. They introduced a robust solution to image-point cloud fusion using a soft-association mechanism. Liu et al. [14] offered alternative approaches to the conventional method. They unified multi-modal features in a shared BEV representation space, aiming to preserve both geometric and semantic information.

III. Network Architecture

A. Overall Network Architecture

The overall network architecture for learning location embeddings is given in Fig. 2. It is structured in a Triplet-like shape with three individual branches, namely, map tile branch, point cloud branch, and panorama branch. The two upper branches are used to learn multi-modal map features and the bottom branch is used to learn semantic features from panoramic images. All learned features from different modalities are used for subsequent feature alignment by employing contrastive learning in the embedded space. Note that there is no weight sharing between branches because the domains (and modalities) of data inputs are different. In the following sections, we provide technical details of each branch and our feature fusion strategies.

B. Map Tile Branch

The map tile branch is used to extract features from the map tile input, which is an image of a local region of the 2D map. The map tile encoder is built upon the ResNet18 network [15], including four convolutional blocks to produce a 512-channel feature volume $\mathbf{F}_{\text{tile}}$ with a resolution that is 1/32 of the original input.

C. Point Cloud Branch

The 2.5D map can be processed into the form of voxel, mesh or point cloud. Compared with mesh and voxel, we use the point cloud for the generalization, efficient storage, and broad usage. In the point cloud branch, the feature encoder is built upon popular backbones used for point cloud representation learning. We study both MLP-based [16], [17], [18] and MLP-Transformer [19] based structure as the feature encode backbones and demonstrate consistent performance improvement brought by the 2.5D map. After the point cloud encoder, the original input is encoded into a shape of $C_{3D} \times N$ feature volume $\mathbf{F}_{\text{point}}$, where N is the number of points and C_{3D} is the number of feature channels.

D. Multi-modal Fusion

The output features from the map tile branch and point cloud branch are fused for the subsequent multi-modality feature learning. The low spatial resolution of the feature

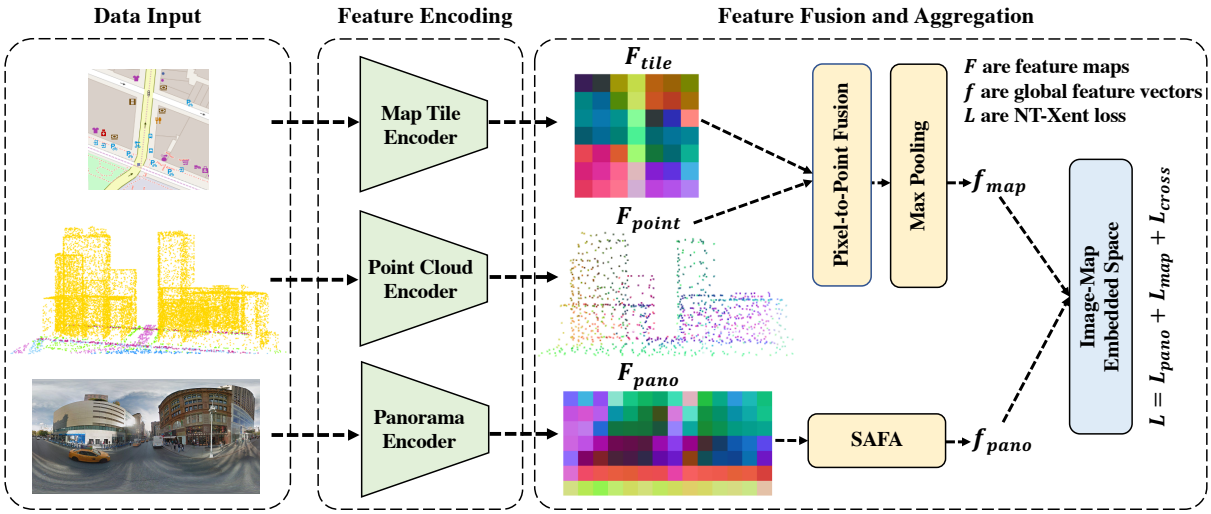

Fig. 2: The overall network architecture consists of a map tile branch, a point cloud branch, and a panorama branch. Each branch consists of an independent feature encoder. The fusion block employs the pixel-to-point projection from 2D space to 2.5D space. The feature aggregators, max pooling, and spatial-aware feature aggregation (SAFA) produce the global feature vectors, which embed semantic and geometric information to achieve feature alignment via contrastive learning. The color of the input point cloud is uniquely encoded by the semantic category as shown in Fig. 5. For better visualization, each feature map is projected into the RGB space.

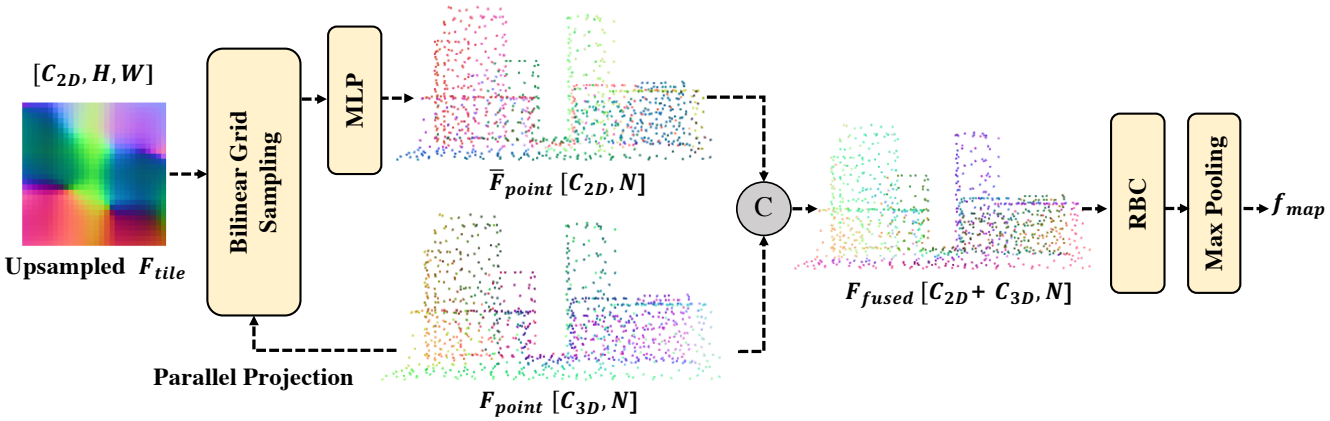

Fig. 3: Pixel-to-Point Fusion. To create a global semantic feature vector, we use bilinear grid sampling and parallel projection to project upsampled tile features $\mathbf{F}_{\text{tile}}$ to the same shape as point cloud features $\mathbf{F}_{\text{point}}$. We then concatenate and fuse these features using Conv1 $\times$ 1-BN-ReLU operations (RBC) and a max pooling aggregator. To visualize each feature map, we project it into the RGB space.

map has a negative impact on pixel-to-point fusion [20]. To recover the spatial resolution, we perform an additional bilinear upsampling after the map tile feature encoder to quadruple the size of its feature map. After the point cloud feature encoder, we incorporate an additional projection module, consisting of a multilayer perceptron (MLP) block, to reduce the feature dimension of the point cloud to that of a map tile.

Given feature volumes $\mathbf{F}_{\text{tile}}$ $(C_{2D} \times H \times W)$ and $\mathbf{F}_{\text{point}}$ $(C_{3D} \times N)$ from individual encoders, we perform pixel-to-point feature fusion. We establish a parallel projection relationship between 2D and 2.5D space:

$$x_i = (\overline{x}_i + 0.5W_g - C_x)\frac{(W-1)}{(W_g-1)} \tag{1}$$

$$y_i = (\overline{y}_i + 0.5H_g - C_y)\frac{(H-1)}{(H_g-1)} \tag{2}$$

where $(\overline{x}_i, \overline{y}_i)$ is the point coordinate, and (x_i, y_i) is the projected pixel coordinate. (W,H) and (W_g,H_g) are the size of the feature map in pixel and geographic level, respectively, while (C_x, C_y) represents the geographic coordinate of central point.

Subsequently, as depicted in Fig. 3, we generate the projected feature volume $\overline{\mathbf{F}}_{\text{point}}$ $(C_{2D} \times N)$ through bilinear grid sampling at $(\overline{x}, \overline{y})$ with the feature volume $\mathbf{F}_{\text{tile}}$. This projected volume is then concatenated with $\mathbf{F}_{\text{point}}$ after passing through a MLP including three Conv1 $\times$ 1-BN-ReLU blocks. Finally, an additional Conv1 $\times$ 1-BN-ReLU block and a max pooling operator are applied to fuse and aggregate multi-modal feature volume $\mathbf{F}_{\text{fused}}$, processing it into a unified global feature vector with the desired embedding size. As highlighted in [16], max pooling, being a symmetric function, is well-suited for processing unordered point cloud.

E. Panorama Branch

The panoramic image offers a broader field of view (FOV), enabling larger observation of the surrounding environment

and enhancing feature discrimination. We use ResNet50 as the panorama encoder as suggested in [4]. After passing through four convolutional blocks, the input panoramic image is transformed into a 512-channel feature volume with a 1/32 resolution of the original size. We leverage the spatial-aware feature aggregation (SAFA) module [21] to localize the salient features and encode the relative spatial layout information.

IV. Model Training

Our model is trained in an end-to-end way via contrastive learning. We combine intra-modal and inter-modal discrimination to formulate the loss function during training, which is inspired by the pioneering work [22]. Given an input panoramic image $\mathbf{I}_i$, we construct augmented versions $\mathbf{I}_i^{t_1}$ and $\mathbf{I}_i^{t_2}$ using rotation, color jittering, normalization, erasing and Gaussian noising augmentations in sequence. The augmented versions $\mathbf{M}_i^{t_1}$ and $\mathbf{M}_i^{t_2}$ of the map tile $\mathbf{M}_i$ are constructed similarly. For the point cloud $\mathbf{P}_i$, $\mathbf{P}_i^{t_1}$ and $\mathbf{P}_i^{t_2}$ are constructed using random shuffle, jittering, and points removal sequentially.

With our encoding and aggregation module, we separately extract global feature vectors $\mathbf{q}_i^{t_1}$ and $\mathbf{q}_i^{t_2}$ from $\mathbf{I}_i^{t_1}$ and $\mathbf{I}_i^{t_2}$. With our fusion block, we get the fused global feature vector $\mathbf{r}_i^{t_1}$ for $(\mathbf{M}_i^{t_1}, \mathbf{P}_i^{t_1})$ and $\mathbf{r}_i^{t_2}$ for $(\mathbf{M}_i^{t_2}, \mathbf{P}_i^{t_2})$. The optimization goal is to maximize the similarity of positive pairs while minimizing the similarity of negative pairs in a mini-batch. For the panorama-modal discrimination, the loss is given by:

$$\mathcal{L}_{\text{pano}} = \frac{1}{2B} \sum_{i=1}^{B} \left[l(\mathbf{q}_i^{t_1}, \mathbf{q}_i^{t_2}) + l(\mathbf{q}_i^{t_2}, \mathbf{q}_i^{t_1}) \right] \tag{3}$$

The loss of the map-modal discrimination is given by:

$$\mathcal{L}_{\text{map}} = \frac{1}{2B} \sum_{i=1}^{B} \left[l(\mathbf{r}_i^{t_1}, \mathbf{r}_i^{t_2}) + l(\mathbf{r}_i^{t_2}, \mathbf{r}_i^{t_1}) \right] \tag{4}$$

The loss of the cross-modal discrimination is given by:

$$\mathcal{L}_{\text{cross}} = \frac{1}{2B} \sum_{i=1}^{B} \left[l(\mathbf{q}_i, \mathbf{r}_i) + l(\mathbf{r}_i, \mathbf{q}_i) \right] \tag{5}$$

$$\mathbf{q}_i = \frac{1}{2}(\mathbf{q}_i^{t_1} + \mathbf{q}_i^{t_2}) \tag{6}$$

$$\mathbf{r}_i = \frac{1}{2}(\mathbf{r}_i^{t_1} + \mathbf{r}_i^{t_2}) \tag{7}$$

We use the NT-Xent loss [23] as the function of $l(\mathbf{z}_i, \mathbf{h}_i)$ for the positive pair of $\mathbf{z}_i$ and $\mathbf{h}_i$:

$$l(\mathbf{z}_i, \mathbf{h}_i) = -\log \frac{\exp(d(\mathbf{z}_i, \mathbf{h}_i)/\tau)}{\sum_{k=1}^{B} \exp(d(\mathbf{z}_i, \mathbf{h}_k)/\tau)} \tag{8}$$

where B is the mini-batch size, τ is the temperature coefficient, and $d(.)$ is the cosine similarity function, which executes the dot product between L_2 normalized feature vector. Finally, the overall loss function is formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{pano}} + \lambda_1 \mathcal{L}_{\text{map}} + \lambda_2 \mathcal{L}_{\text{cross}} \tag{9}$$

where λ_1 and λ_2 are weighting factors to control the influence of each loss component, which we set to be equal as suggested in [22], [4].

V. The Dataset

We construct the first large-scale multi-modal dataset for geolocation by incorporating co-located ground-view images, 2D map tiles and 2.5D map models. The images are collected from the StreetLearn dataset [24], consisting of 113,767 high-resolution panoramic images (1664×832 pixels) named with unique string identifiers in the cities of New York (Manhattan) and Pittsburgh. In the metadata, there is detailed information about the geographical position (lat/long coordinates and altitude in meters), camera orientation (pitch, roll, and yaw angles), and the connected neighbors of each location. To generate the training/testing/validation split, we use the same approach proposed in [4]. There are two testing sets from areas of Union Square and Wall Street, each containing 5000 locations, covering around 75.6 km and 73.1 km trajectories, respectively. The validation set is generated from the area of Hudson River with the same size as the testing set, covering around 69.3 km trajectory. The remaining 98,767 locations are designated for training. There are diverse scenes in different areas, including skyscrapers, highways, parks, and riversides located on regular street grids (Union Square, Hudson River) or narrow streets with irregular intersections (Wall Street).

For each location, the corresponding multi-modal map data is automatically generated from the public map service, OpenStreetMap (OSM) [3], as illustrated in Fig. 4. With OSM metadata, the 2D map tiles are rendered using Mapnik [25], each with 256×256 pixels, centered on the geo-tagged location, and aligned with the heading direction The 2.5D map models are automatically processed through the following pipeline from OSM to the point cloud. First, we utilize Blender [26] to render the OSM metadata into several triangle-mesh structural models, each representing an individual semantic category as depicted in Fig. 5. In OSM, the height is defined either in meters or levels. Subsequently, we use the Barycentric coordinate system [27] to uniformly sample a certain number of points on each triangle. The number of points is determined by the sampling density (0.1 m) and surface area. Finally, we merge points of each semantic category to a completed point cloud covering the whole area and crop the corresponding 2.5D map of a local region given the geo-tagged location. The local region represents an area with a geographical size of 152×152 m^2, centered at the geo-tagged location and aligned with the heading direction. The dataset and code are available at `https://github.com/ZhouMengjie/2-5DMap-Dataset`.

VI. Experiments

A. Setting

We implement our network in Pytorch. All models (baselines or ours) are trained in an end-to-end manner for 60 epochs on 4 Nvidia A100 GPUs. We use ImageNet [28] pre-trained weights to initialize the map tile encoder and

Fig. 4: Multi-modal map dataset. The 2D map (a) and 2.5D map (b) are from the area of Wall Street, which includes narrow streets and highways with irregular intersections. The 2D map (c) and 2.5D map (d) are from the area of Union Square, which includes densely distributed skyscrapers, brownstones, townhouses, and parks located on regular street grids. For the 2.5D map, the unique color is encoded by the semantic category as shown in Fig. 5.

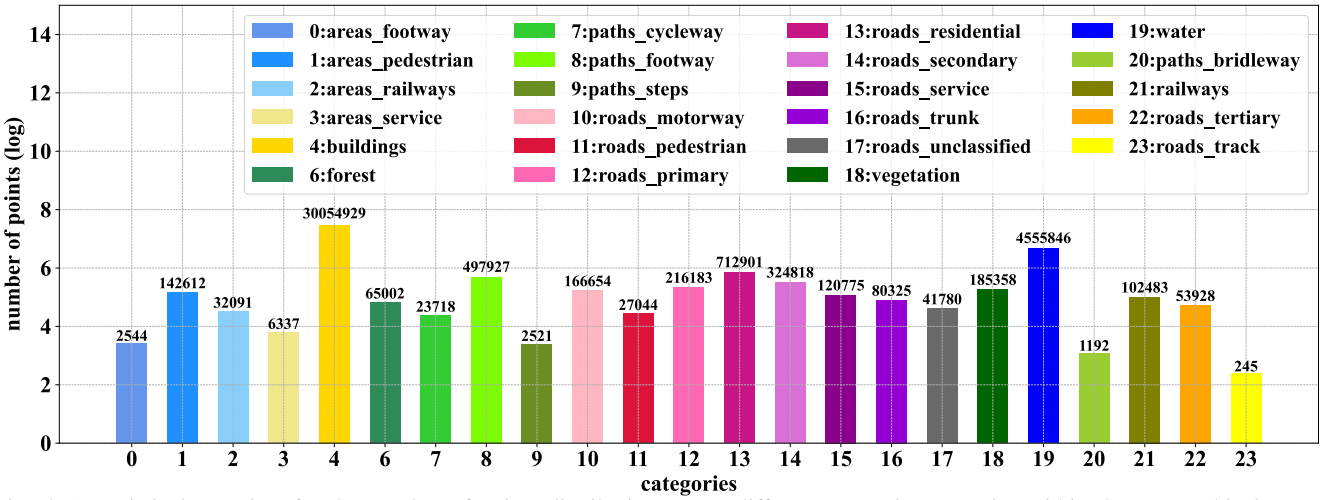

Fig. 5: A statistical overview for the number of points distribution across different semantic categories within the geographical area of Manhattan. Semantic categories are initially labeled 0 to 23 and then encoded as 24-D one-hot vectors for network input if required. Note that Category 5 (Coastline) is absent in Manhattan's semantic categories. For ease of display, we project the data into log space but annotated the actual number of points at the top of each bar.

Places365 [29] for the panorama encoder. The other modules are initialized randomly.

The panorama and map tiles are resized to 448×224 and 224×224 respectively. The dense point cloud is firstly normalized to the range of -1 to 1 and then downsampled to 1024 points with the farthest point sampling strategy as suggested in [17]. The network output initially has an embedding size of 4096. To minimize redundancy and enhance computation and storage efficiency, we use the Principal Component Analysis (PCA) for flexible feature dimension reduction. This results in a final embedding size of either 128 or 16, depending on the geolocation methods used.

During back-propagation, we use the Adaptive Sharpness-aware Minimization (ASAM) strategy [30] combined with the AdamW optimizer to optimize the network. The AdamW optimizer has an initial learning rate of 1×10^{-4} and a weight decay of 0.03. The cosine annealing scheduler is used to gradually decrease the learning rate to a minimum (0). We use a batch size of 32 and the temperature in Eq. (8) is set to 0.07. The model performing best on the validation set is chosen for subsequent geolocation tasks.

B. Geolocation Results

We validate our learned location embeddings in two geolocation strategies, i.e., single-image based geolocation and route based geolocation. For the former, we use the Top-k recall rate to evaluate the geolocation performance on the Hudson River, Wall Street, and Union Square test sets. That is, given a query panoramic image, we retrieve the Top-k geo-referenced maps by measuring the similarity (L_2 distance) between their 128D global semantic features. If the matching reference map is ranked within the Top-k list, a query panoramic image is considered to be localized successfully. The Top-k recall rate shows the percentage of correctly localized queries. For the latter, we use 500 randomly generated routes consisting of 40 adjacent locations in the area of Hudson River, Wall Street and Union Square. The test data is provided by [4], and the distance between each location is around 10 m. We adopt the Top-1 recall rate as our evaluation metric, which is measured by the percentage

of correctly localized routes as a function of route length. Specifically, a route is considered to have been successfully localized at step t if and only if the matching reference maps from step $t-4$ to t are all ranked first.

Single-image Based Geolocation In our study, we evaluate the recall rate for the top $k\%$ of the dataset, where $k\%$ represents a fraction of the dataset size. We use the state-of-the-art single-modal method [4] as the baseline. The comparisons are given in Fig. 6 (a)-(c). It shows that using 2.5D maps can yield significantly better performance compared to single-modal method. Specifically, the Top-1 recall improvements on the Hudson River, Wall Street, and Union Square are 19.08%, 18.24% and 26.9%, respectively. Some qualitative geolocation examples are given in Fig. 7. Following ablation studies are also conducted on this task.

Route Based Geolocation In large cities with many repetitive scenes, a single descriptor is not sufficiently discriminative. We implement a route based geolocation method [5] with modifications, i.e., rather than storing all route candidates in advance, we generate them online based on connectivity between adjacent locations. To further improve the efficiency, we adopt a culling strategy that iteratively eliminates 50% of route candidates at each movement until at least 100 remain.

The comparisons are given in Fig. 6 (d). Our method outperforms with the state-of-the-art [4]. When moving to the location with a route length of 5, our multi-modal method already achieves over 75% geolocation accuracy, which is more than 10% higher than [4]. The results show that fusing multi-modal map features for the route based geolocation leads to higher accuracy and faster convergence.

C. Ablation Study

Fusion Strategy To study the fusion of multi-modal features, we examine three design options: global fusion with add or concatenation operators and pixel-to-point fusion. The global fusion block is given in Fig. 8. Initially, a map tile encoder extracts the feature volume $\mathbf{F}_{\text{tile}}$, which is then fed into a SAFA module to create a C_g-channel global feature vector $\mathbf{f}_{\text{tile}}$. Similarly, the point cloud encoder extracts the feature volume $\mathbf{F}_{\text{point}}$, which is then projected into two $C_g/2$-channel global vectors using max and average pooling. These vectors are concatenated to form a C_g-channel global feature vector $\mathbf{f}_{\text{point}}$. Finally, the multi-modal global feature vectors are either concatenated or added along the channel dimension and projected to the desired embedding size after a fully connected layer.

Table I illustrates the Top-1 recall rate localizing in Hudson River, Wall Street, and Union Square. As shown, the pixel-to-point fusion with upsampled map tile features exhibits the highest success rate across all testing sets. When compared to global fusion using the add operator, there are notable performance gains of 3.62%, 4.4%, and 6.42% observed in different geolocation areas.

Point Sampling Strategy We conduct a comparison between two point cloud sampling strategies – random point sampling (RPS) and farthest point sampling (FPS). We sampled 256, 512, 1024, and 2048 points for each strategy to process the point cloud. Based on the data in Table II, we find that FPS generally provides better geolocation accuracy, and increasing the number of sampled points results in better performance. This is likely because FPS preserves more structure information compared to RPS. After considering the trade-off between efficiency and accuracy, we decided to use FPS with 1024 points as our sampling strategy.

TABLE I: Comparison between multi-modal methods using global fusion with concatenation and add operator, and pixel-to-point fusion in different testing areas. Method denoted with * utilize the four-fold upsampled map tile feature as an input to the fusion block.

Fusion Strategy	Hudson River	Wall Street	Union Square
Concatenate	63.38	56.98	74.10
Add	64.08	56.26	76.54
Pixel-to-Point	66.96	60.00	81.50
Pixel-to-Point*	**67.70**	**60.66**	**82.96**

TABLE II: Robustness of multi-modal method to density variation and the number of points. Various types of point clouds are generated by the random point sampling and farthest point sampling in the area of Union Square. The Top-1 recall rate (%) is calculated to evaluate the geolocation performance.

Sampling Strategy	256	512	1024	2048
Random Point Sampling	49.72	67.50	77.10	80.70
Farthest Point Sampling	**73.58**	**81.12**	**82.96**	**83.66**

Point Cloud Encoder We study multi-modal methods utilizing different point cloud encode backbones. Specifically, we implement pixel-to-point fusion for Pointnet [16] and DGCNN [18] based methods since they do not employ any further point sampling during the forward pass. For Pointnet++[17] and Point Transformer[19], the number of points is reduced to 256 and 16, respectively. Note that this particular process has been known to cause a notable decrease in performance, based on previous experiments. Therefore, we choose to utilize global fusion with the addition operator for these two methods in order to evaluate their performance. Table III presents the comparisons. When combined with global fusion, employing Point Transformer as the point cloud encoder yields the best performance. When adopting pixel-to-point fusion, using DGCNN as the point cloud encoder achieves superior results. In conclusion, our investigations reveal that employing either MLP-based [16], [17], [18] or MLP-Transformer [19] based structures as the feature encode backbones consistently leads to improved performance when integrating the 2.5D map.

Semantic Category In Fig. 5, the 2.5D map comprises 24 distinct semantic categories. Certain mainstream methods [9], [10] solely employ building information to generate the 2.5D map for fine geolocation tasks. In this work, we investigate the performance enhancement achieved by incorporating richer semantic information within the 2.5D map. We find that there is a performance degradation of 9.34%, 10.08% and 2.38% for the Hudson River, Wall

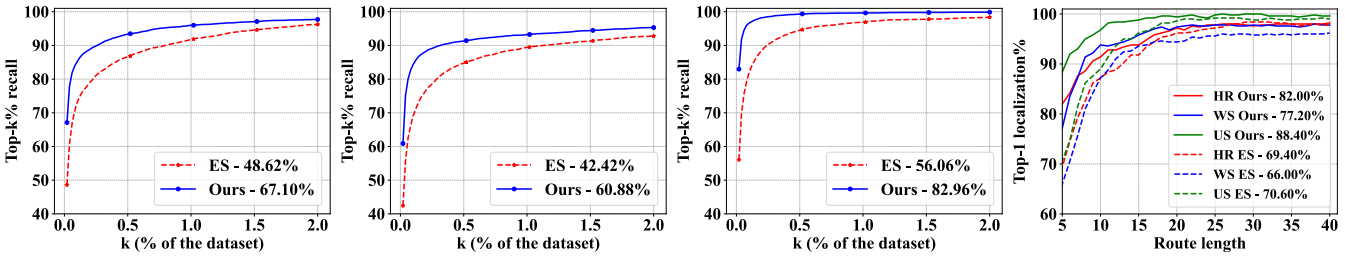

Fig. 6: Comparison between single-modal and multi-modal methods on two geolocation tasks. The Embedded Space Descriptor (ES) [4] serves as the single-modal baseline. Single-image based geolocation: The Top-k% recall rate evaluates the performance in the area of Hudson River (a), Wall Street (b), and Union Square (c). The Top-1 recall rate is shown in the lower-right legend. Route-based geolocation (d): The Top-1 recall rate relative to route length is evaluated for three areas. The result at step 5 is shown in the lower-right legend.

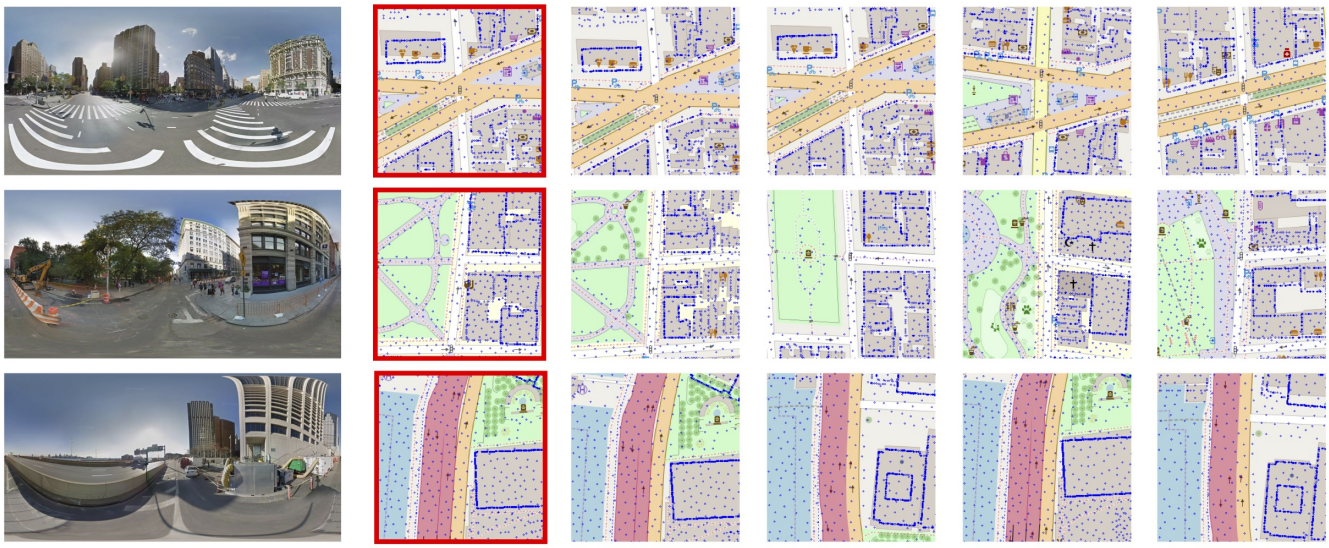

Fig. 7: Top-5 retrieved maps (b)-(f) given a query panoramic image (a). The correct related map of the query is outlined in red.

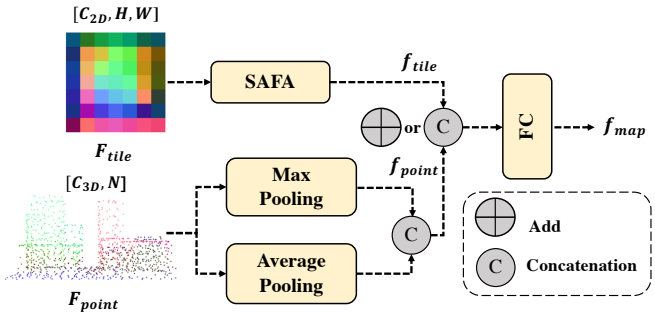

Fig. 8: Procedure of global fusion. The feature volume outputs from separate feature encoders are initially aggregated into single-modal global feature vectors. Subsequently, these individual global feature vectors are combined through either concatenation or addition, resulting in a fused global feature vector after passing through a fully-connected layer.

Street and Union Square areas, respectively. By including points from other semantic categories, such as water bodies and residential roads, the Top-1 accuracy further increases. The results affirm the significance of incorporating diverse semantic information within the 2.5D map to achieve superior geolocation outcomes across varied urban landscapes,

TABLE III: Comparison between multi-modal methods using different point cloud encoders. The Top-1 accuracy is calculated for three testing sets. Models denoted with * utilize the pixel-to-point fusion, whereas the remaining models adopt global fusion with the add operator.

Encoder	Hudson River	Wall Street	Union Square
Pointnet	63.60	57.36	75.22
Pointnet++	64.32	57.02	76.08
Point Transformer	64.02	58.56	77.26
DGCNN	64.08	56.26	76.54
Pointnet*	65.38	57.76	78.72
DGCNN*	**67.70**	**60.66**	**82.96**

particularly in more sparsely built areas like Wall Street as shown in Fig. 4(b).

D. Complexity Analysis

We analyze the computational cost and complexity of various methods on an Nvidia 3090 GPU by evaluating their Top-1 accuracy, inference time, memory utilization, and model size. Results are given in Table IV. Our multi-modal methods outperform single-modal approaches, with larger

TABLE IV: Complexity comparison on Union Square between single-modal (ES, Ours-2d, Ours-2.5d) and multi-modal methods. Memory is the maximum GPU memory occupied by tensors in an inference loop (batch size of 1). Size is the model size. 2to3 represents model with pixel-to-point fusion.

Model	Top-1 (%)	Time (ms)	Memory (MB)	Size (MB)
ES	56.06	2.37	53.20	378.73
Ours-2d	65.04	2.13	32.63	131.17
Ours-2.5d	49.24	3.14	85.97	156.08
Ours-concat	74.10	4.45	94.94	263.58
Ours-add	76.54	4.35	89.61	199.58
Ours-2to3	82.96	5.62	88.81	164.93

success rates and smaller model sizes, but they require more inference time and memory usage.

VII. Conclusion

In this paper, we proposed a novel approach for geolocating ground query images using multi-modal maps. Unlike previous methods, which only used 2D maps as the georeferenced database, we extended the 2D maps to 2.5D maps, where the heights of structures can be used to support ground-to-map matching. A new multi-modal representation learning framework is proposed to learn location embeddings from panoramic images and multi-modal maps. We also constructed the first large-scale multi-modal geolocation dataset to facilitate future research. Extensive experiments demonstrate that our multi-modal embeddings achieve significantly higher geolocation accuracy in both single-image based and route based geolocation. Future work will focus on geolocation with videos or sequences comprising limited FOV images, which are more common and easier to capture from larger and diverse areas.

References

[1] T. Sattler, M. Havlena, K. Schindler, and M. Pollefeys, "Large-scale location recognition and the geometric burstiness problem," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1582–1590.

[2] L. Liu and H. Li, "Lending orientation to neural networks for cross-view geo-localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5624–5633.

[3] Open Street Map. [Online]. Available: https://www.openstreetmap.org

[4] N. Samano, M. Zhou, and A. Calway, "You are here: Geolocation by embedding maps and images," in *European Conference on Computer Vision*. Springer, 2020, pp. 502–518.

[5] P. Panphattarasap and A. 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)*. IEEE, 2018, pp. 6341–6348.

[6] M. Zhou, X. Chen, N. Samano, C. Stachniss, and A. Calway, "Efficient localisation using images and openstreetmaps," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5507–5513.

[7] T. Vojir, I. Budvytis, and R. Cipolla, "Efficient large-scale semantic visual localization in 2d maps," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[8] P.-E. Sarlin, D. DeTone, T.-Y. Yang, A. Avetisyan, J. Straub, T. Malisiewicz, S. R. Bulo, R. Newcombe, P. Kontschieder, and V. Balntas, "Orienternet: Visual localization in 2d public maps with neural matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 632–21 642.

[9] A. Armagan, M. Hirzer, P. M. Roth, and V. Lepetit, "Learning to align semantic segmentation and 2.5d maps for geolocalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[10] H. Li, T. Fan, H. Zhai, Z. Cui, H. Bao, and G. Zhang, "Bdloc: Global localization from 2.5d building map," in *International Symposium on Mixed and Augmented RealityISMAR*. IEEE, 2021, pp. 80–89.

[11] C. Wang, C. Ma, M. Zhu, and X. Yang, "Pointaugmenting: Cross-modal augmentation for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 794–11 803.

[12] Y. Li, A. W. Yu, T. Meng, B. Caine, J. Ngiam, D. Peng, J. Shen, Y. Lu, D. Zhou, Q. V. Le *et al.*, "Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 182–17 191.

[13] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C.-L. Tai, "Transfusion: Robust lidar-camera fusion for 3d object detection with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 1090–1099.

[14] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. L. Rus, and S. Han, "Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *2023 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2023, pp. 2774–2781.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[16] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[17] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[18] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *Acm Transactions On Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.

[19] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 16 259–16 268.

[20] Y.-C. Liu, Y.-K. Huang, H.-Y. Chiang, H.-T. Su, Z.-Y. Liu, C.-T. Chen, C.-Y. Tseng, and W. H. Hsu, "Learning from 2d: Contrastive pixel-to-point knowledge transfer for 3d pretraining," *arXiv preprint arXiv:2104.04687*, 2021.

[21] Y. Shi, L. Liu, X. Yu, and H. Li, "Spatial-aware feature aggregation for image based cross-view geo-localization," *Advances in Neural Information Processing Systems*, vol. 32, pp. 10 090–10 100, 2019.

[22] M. Afham, I. Dissanayake, D. Dissanayake, A. Dharmasiri, K. Thilakarathna, and R. Rodrigo, "Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9902–9912.

[23] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[24] P. Mirowski, A. Banki-Horvath, K. Anderson, D. Teplyashin, K. M. Hermann, M. Malinowski, M. K. Grimes, K. Simonyan, K. Kavukcuoglu, A. Zisserman *et al.*, "The streetlearn environment and dataset," *arXiv:1903.01292*, 2019.

[25] Mapnik. [Online]. Available: https://mapnik.org

[26] Blender. [Online]. Available: https://github.com/vvoovv/blender-osm/wiki/Documentation

[27] M. Meyer, A. Barr, H. Lee, and M. Desbrun, "Generalized barycentric coordinates on irregular polygons," *Journal of graphics tools*, vol. 7, no. 1, pp. 13–22, 2002.

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

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

[30] J. Kwon, J. Kim, H. Park, and I. K. Choi, "Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5905–5914.