SpaGBOL: Spatial-Graph-Based Orientated Localisation

Tavis Shore[1] Oscar Mendez[2] Simon Hadfield[1]
University of Surrey[1] Locus Robotics[2]
{t.shore, s.hadfield}@surrey.ac.uk, omendez@locusrobotics.com

Abstract

Cross-View Geo-Localisation within urban regions is challenging in part due to the lack of geo-spatial structuring within current datasets and techniques. We propose utilising graph representations to model sequences of local observations and the connectivity of the target location. Modelling as a graph enables generating previously unseen sequences by sampling with new parameter configurations. To leverage this newly available information, we propose a GNN-based architecture, producing spatially strong embeddings and improving discriminability over isolated image embeddings. We outline SpaGBOL, introducing three novel contributions. 1) The first graph-structured dataset for Cross-View Geo-Localisation, containing multiple streetview images per node to improve generalisation. 2) Introducing GNNs to the problem, we develop the first system that exploits the correlation between node proximity and feature similarity. 3) Leveraging the unique properties of the graph representation - we demonstrate a novel retrieval filtering approach based on neighbourhood bearings. SpaGBOL achieves state-of-the-art accuracies on the unseen test graph - with relative Top-1 retrieval improvements on previous techniques of 11%, and 50% when filtering with Bearing Vector Matching on the SpaGBOL dataset. Code and dataset available: github.com/tavisshore/SpaGBOL.

1. Introduction

Localisation is essential in many robotics applications. Techniques like Global Navigation Satellite Systems (GNSS) provide absolute positioning data but often fail in environments like urban canyons, where occlusions and reflections interfere with satellite signals. Image-based localisation offers an alternative approach, enabling a machine to determine its position by capturing images of its surroundings and comparing them to pre-recorded geo-referenced images. Most modern vehicles are equipped with cameras, simplifying the adoption of image-based localisation.

Two main retrieval-based image localisation techniques

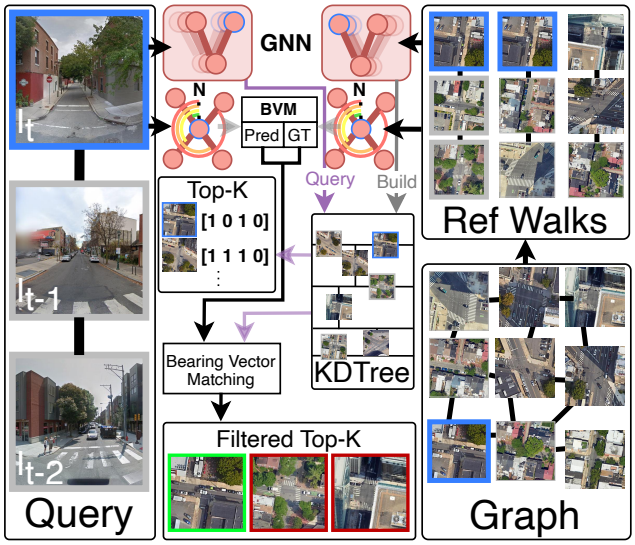

Figure 1. At inference time, a KDTree is constructed from exhaustive reference walks sampled from the city's graph. A randomly selected query walk passes through the network, retrieving corresponding embeddings from the KDTree ordered in descending similarity. These are further filtered to the set of compatible nodes with Bearing Vector Matching (BVM).

are: image-to-image localisation, where query and reference images are taken from the same perspective, and Cross-View Geo-localisation (CVGL), where street view query images are matched with a database of satellite images. Both with the same objective - returning the geographic coordinates of the retrieved image. Existing CVGL techniques primarily focus on sparse streetview-satellite image pairs - randomly sampled from across vast regions, disregarding the geo-spatial structure and relationships between neighbouring regions. Sequential CVGL extends single-image techniques, querying multiple images to strength representations - extracting features with cross-frame information. This provides a more practical solution, and estimates position with higher confidence and precision. These datasets and techniques succeed in learning related features between the viewpoints but still consider data as sequences of separate image pairs with no spatial

structure beyond chronology. Reference data remains unstructured with no geo-spatial metadata, limiting real-world representational accuracy. This can make it challenging to recognise new sequences which partially overlap or combine several existing sequences seen during training. To improve the feasibility of CVGL, research should be focused to regions most likely to experience GNSS communication failure, dense urban city centres. The design of image localisation techniques should progress to expect any possible sequence of images within the considered regions.

We propose structuring image localisation data as graph networks. This adds crucial geo-spatial information, enabling the generation of unseen sequences of desired length. Progressing to this data representation is relatively simple as the target of our system, urban canyons within dense city centres, generally have existing accurate graph representations within many Geographic Information Systems (GIS). We therefore propose utilising GNNs to improve CVGL within this novel representation, storing sets of streetview images and satellite images at junctions (graph nodes), with connecting roads represented as the graph edges between these nodes. A brief overview of the proposed system is displayed in Figure 1. To solidify our proposal into the progression of CVGL towards real-world feasibility, we release the *Spatial-Graph-Based Orientated Localisation (SpaGBOL)* dataset: a dense multi-city graph-based CVGL dataset with multiple streetview images per satellite image - allowing for generalisation across time, weather, and lighting. This dataset is split into training and test sets, comprising of 9 cities and 1 city respectively. We prove the positive impact that graph representation has on CVGL performance due to strengthened feature representation and filtering by neighbourhood road bearings - valid within this city-scale due to neighbouring node's close proximity.

In summary, our research contributions are:

- Introduce a new direction for CVGL research, moving from sparse cross-view image retrieval and sequential image retrieval into spatially-strong dense image retrieval, moving the field closer to real-world feasibility for assisting GNSS techniques in urban environments.
- Propose an introductory GNN model utilising data along graph walks to create strong representations, also exploiting derived characteristics to filter retrievals with Bearing Vector Matching (BVM), greatly improving performance.
- Release a dense multi-city graph-based CVGL dataset, *SpaGBOL*, containing train and test set graphs with corresponding images from a sample of the densest city centres across the globe.

2. Related Works

2.1. Cross-View Geo-Localisation

The predominant technique for CVGL is embedding retrieval. Novel techniques are being proposed at an increasing rate, aiming to improve performance by manipulating extracted features, [1], [2], [3].

Deep learning was first introduced to CVGL by Workman and Jacobs [4], utilising CNNs for correlated feature extraction across viewpoints, proving their suitability. Lin et al. [5] extended this by regarding each query as unique - using euclidean similarities for retrieving clusters. Vo and Hays [6] then utilised aerial rotational information with an auxiliary loss, observing the impact of image misalignment - leading to our incorporating of a compass in order to aid system performance. CVM-Net [7] appended NetVLAD [8] to a siamese CNN architecture, aggregating residuals of local features to cluster centroids - improving accuracy though greatly increasing complexity. Zhu et al. [9] leveraged activation maps to estimate orientation. Sun et al. [10] created a capsule network following a ResNet backbone, improving upon CVM-Net performance by approximately 10%. Liu and Li [11] inserted orientation information to the problem, improving the representational robustness of their latent space. Shi et al. [12] developed a spatial attention mechanism, improving feature alignment between views. Regmi et al. [13] created a conditional GAN to synthesise aerial representations of ground-level panoramas. Shi et al. [14], [15] proposed techniques for increasing the similarity of features across viewpoints before applying them to limited-Field-of-View (FOV) data. This is important due to the ubiquity of monocular cameras compared with panoramic cameras; essential for wide-spread feasibility and adoption. [15] computes feature correlation between ground-level images and polar-transformed aerial images, shifting and cropping at the strongest alignment before performing image retrieval.

Toker et al. [16] proposed synthesising streetview images from aerial image queries before performing image retrieval. L2LTR [17] developed a CNN+Transformer network, combining a ResNet backbone with a vanilla ViT encoder. TransGeo [1] proposed a transformer that uses an attention-guided non-uniform cropping strategy to remove uninformative areas. In GeoDTR [18] and their following work GeoDTR+ [19], Zhang et al. disentangle geometric information from raw features, learning spatial correlations among visual features to increase performance. Zhu et al. [2] introduce *SAIG*, an attention-based backbone for CVGL, representing long-range interactions among patches and cross-view relationships with multi-head self-attention layers. BEV-CV [3] introduces Birds-Eye-View (BEV) transforms, further reducing the representation difference between viewpoints to create more simi-

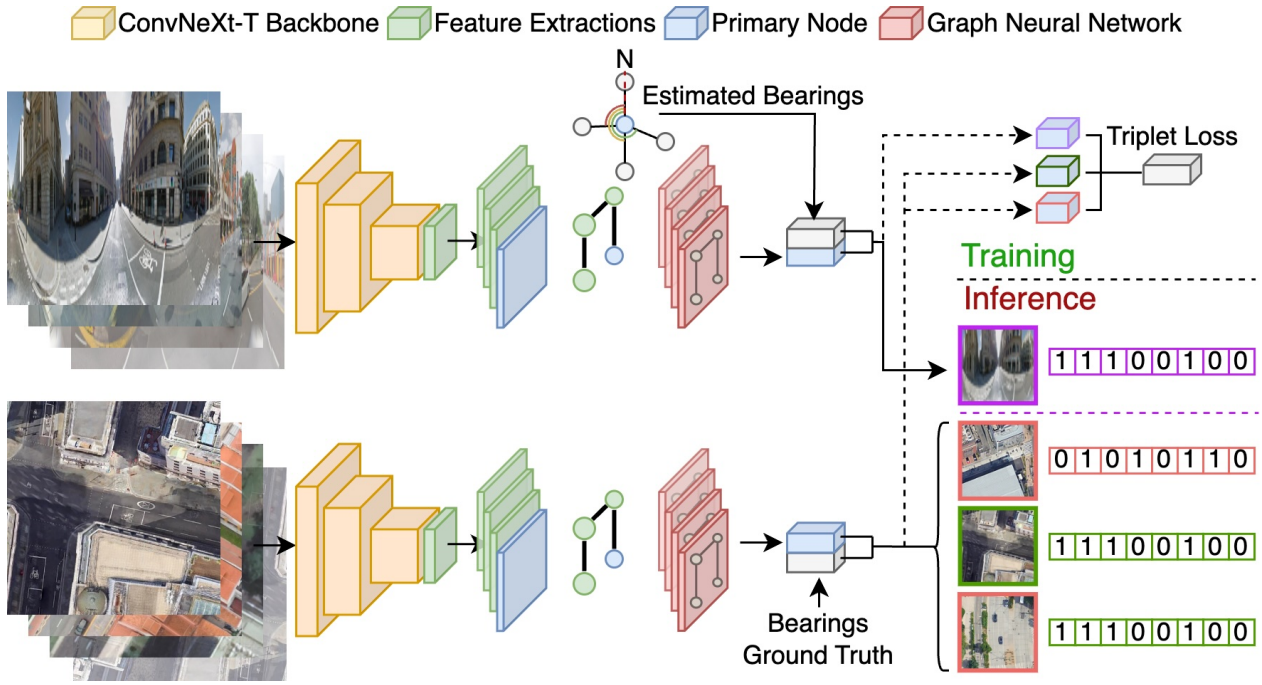

Figure 2. SpaGBOL is a two-branch neural network with no weight-sharing, from left to right the network performs the following actions: (1) Image feature extraction with ConvNext-T, (2) Depth-first walk image features → GNN embedding (red), (3) Produce neighbour bearing vectors, (4) Perform embedding retrieval from the KDTree, (5) Filter retrievals with bearings to return final geo-coordinates.

lar embeddings. Sample4Geo [20] propose two sampling strategies for CVGL, sampling geographically for optimal training initialisation, and mining hard-negatives according to visual similarities between embeddings. Generally, the above works all focus on developing more similar embeddings for either sparsely sampled image pairs or relatively limited image sequences. In contrast, we transition CVGL to methods that more closely represent real GNSS-denied regions, advancing the field towards practical application.

2.2. Graph-Based Localisation

Graph-networks and GNNs have not previously been utilised in the field of CVGL. They have however been applied to related fields, from localising objects within scene graphs to mapping out environments for graph-based SLAM. We outline some key related works that contributed to our proposition of their application for CVGL.

Graph-based SLAM techniques construct a graph mapping of an environment while simultaneously localising an agent within the map. Heinzle et al. [21] introduce pattern recognition within road networks - aiming to perform automatic localisation of city centres. Grisetti et al. [22] display an overview of Graph-based SLAM methods, representing generally GNSS-denied indoor environments as graphs, localising within the graph using probabilistic techniques. Kümmerle et al. [23] introduces the use of aerial priors alongside sensor data to improve map creation for graph-based SLAM. Annaiyan et al. [24] use stereo imaging to construct and localise UAVs within a graph-based map. He et al. [25] combine visual-LIDAR data to construct 3D maps of environments, merging with a pose graph optimisation procedure. Vysotska and Stachniss [26] present a search heuristic aiming to efficiently find matches between an image sequence and a database using a data association graph. Johnson et al. [27] introduce a framework for semantic image retrieval based on scene graphs, outperforming methods that only use low-level image features. Liu et al. [28] leverage object level semantics and spatial environment understanding for localisation, improving performance where extreme appearance changes occur. Giuliari et al. [29] use Spatial Commonsense Graphs to localise objects in partial scenes where nodes represent objects, and edges represent pairwise distances between them. Finally we outlined examples of practical applications of both graph structures and GNNs. [30] represent water utility networks as graphs, using Graph Convolutional Networks (GCNs) to predict nodal pressures, and localise leaks. In a similar manner, [31] introduce graphs and GNNs to localise epileptic seizure onset zones, where nodes represent different regions of the brain. Murai et al. [32] developed a graph-based collaborative localisation system for robots, globally localising via efficient peer-to-peer communication. Most prior graph and GNN works have attempted to learn similarities between related examples from the same domain. In our work we attempt to preform cross-view graph matching between images on the ground, and those from a satellite.

Figure 3. Corpus graph of London City Centre. Each graph is square with sides of length 2km. Nodes (junctions) are shown here in blue, with black edges (roads).

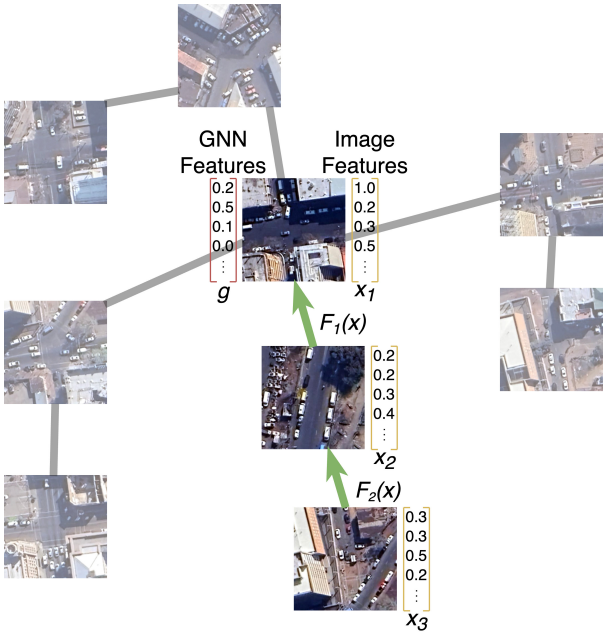

Figure 4. Random depth-first walk sample of length 3. Image features are extracted from each node, passing through a GNN to produce the final node embedding.

3. Methodology

3.1. CVGL Graph Representation

To store geographically dense collections of images with a strong spatial structure we propose a graph representation, improving feasibility and extending the potential techniques suitable for CVGL - an example graph is shown in Figure 3. We represent cities $i \in \{London, Tokyo, ...\}$ as separate graphs $G_i = (N_i, E_i)$ with nodes $N_i = \{n_1, n_2, ..., n_N\}$ and edges $E_i = \{e_{1,2}, e_{1,3}, ..., e_E\}$. Nodes n represent road junctions and edges $e_{a,b}$ represent roads connecting nodes a and b. Figure 5 shows how the graphs are separated into train/validation/test sets. For each node we collect a satellite image and 5 corresponding panoramic streetview images captured over an extended period. Both image types are RGB: $I_t \in \mathbb{R}^{3\times W\times H}, t \in \{street, sat\}$. Each node holds attributes - $n_i = \{I_{sat}, I^{1..5}_{street}, L, \Psi, B\}$, where location $L = \{\phi, \lambda\}$ contains geographical latitude and longitude coordinates, $\Psi \in \mathbb{R} : \{-180° \leq \Psi \leq 180°\}$ is the north-centred camera yaw, and $B = \{\beta_1, ..., \beta_K\}$ are north-aligned bearings to it's K neighbouring nodes - where $\beta \in \mathbb{R} : \{-180° \leq \beta \leq 180°\}$.

The panoramic streetview image (I^*_{street}) FOV is varied to evaluate the feasibility of using monocular cameras. Cameras are assumed to be fixed to the vehicle in a forward-facing configuration. Where FOV, $\Theta \in \{360°, 180°, 90°\}$:

$$I_{street} = \text{fov\_crop}\left(I^*_{street}, \Theta, \Psi\right) \tag{1}$$

The proposed system takes randomly sampled query walks (exhaustive for reference set) W^j_i of length $l \in \{1, ..., 5\}$ as input from each node n_j in graph G_i

$$W^j_i = \text{random\_walk}(G_i(n_j)) \tag{2}$$

A walk representation is shown in Figure 4, randomly selecting one depth-first walk from the target node's available walks. This walk is then extracted from the corpus graph as a subgraph - passing the streetview images, satellite images, and other attributes through the corresponding branches within the SpaGBOL network. The training/validation/testing walks are sampled from disconnected graphs and subgraphs, as shown in Figure 5.

3.2. SpaGBOL Neural Network

During training, corresponding streetview and satellite image walks are passed through *SpaGBOL*, shown in Figure 2. The network's upper and lower branches are identical but do not share any weights. Streetview queries are passed through the upper branch and corresponding satellite targets through the lower branch. Each branch first embeds it's inputs through CNN backbones:

$$feat_{street} = \text{CNN}_{\text{street}}\left(I^{rand(0-4)}_{street}\right) \tag{3}$$

$$feat_{sat} = \text{CNN}_{\text{sat}}\left(I_{sat}\right). \tag{4}$$

A sequence of GNN layers then process the results, as

$$h^{k+1}_{n_j} = \sigma\left(\Omega^k \cdot \text{AGG}\left(\{h^k_{n_u}, \forall u \in W_F\}\right)\right) \tag{5}$$

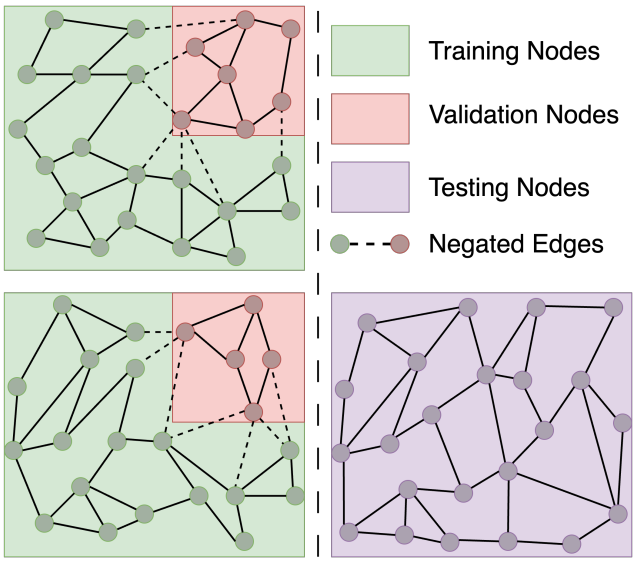

Figure 5. Splitting corpus graphs into train/validation/test sets. Validation graphs are unconnected subgraphs of each training graphs. The test graph is a wholly unseen city graph.

where $h_{n_j}^{k+1}$ is the updated embedding of node, n_j at layer $k+1$, σ is an activation function, Ω^k is a weight matrix for layer k, AGG is a mean-based aggregating function combining features from neighbouring nodes, $h_{n_u}^k$ is the embedding of node n_u at layer k, W_F is the set of walk image features where $F \in \{feat_{street}, feat_{sat}\}$. The output graph embedding from the final layer is then $h_{n_j}^L$. For the streetview branch these final embeddings are notated as η_{street}^j while the satellite branch embeddings are η_{sat}^j.

The network is trained using a triplet loss function, with the objective of producing similar GNN embeddings for corresponding streetview and satellite walks. We select walk triplets by deeming a walk of streetview images as the anchor, it's corresponding walk of satellite images as the positive, and randomly selecting an unrelated walk of satellite images as the negative. More specifically, we utilise the Triplet Loss Function:

$$\mathcal{L} = \sum_{i=1}^{N} \left[\|\eta_{street}^a - \eta_{sat}^p\|_2^2 - \|\eta_{street}^a - \eta_{sat}^n\|_2^2 + \alpha \right] \quad (6)$$

where η_{street}^a, η_{sat}^p, and η_{sat}^n are the anchor, positive, and negative embeddings, respectively, $\|\cdot\|_2$ is the Euclidean norm, and α is the margin.

3.3. Bearing Vector Matching

A significant benefit of utilising graphs for CVGL is the ability to efficiently filter route proposals. We pre-compute both the number of neighbours at each node, and the relative bearings (azimuth) to each neighbour. These relative bearings $\theta \in \{\beta_0, ..., \beta_K\}$ are calculated using the geographic

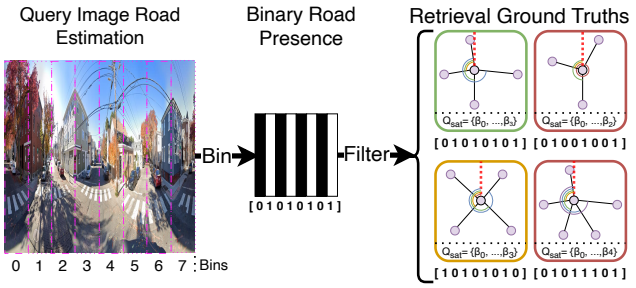

Figure 6. Road bearings may be estimated from panoramic streetview images into a configurable number of bins. These can then be matched against quantised bearings for retrievals.

coordinates of the two nodes (a and b):

$$\beta_b = \text{acos}(\sin(\phi_a)\cdot\sin(\phi_b) + \cos(\phi_a)\cdot\cos(\phi_b)\cdot\cos(\Delta\lambda)) \quad (7)$$

where $\Delta\lambda$ is the difference in longitude and ϕ is the latitude of each node. These bearings are then quantised into V bins, in the bearings vector $Q = (Q_0, Q_1, ..., Q_V)$:

$$Q_v = \begin{cases} 1 & \exists\theta \in \{\beta_0, ..., \beta_K\} \text{ such that } \frac{v}{V} < \frac{\theta}{2\pi} \leq \frac{v+1}{V} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

This creates a binary code describing the arrangement of roads at this junction. Bins of equal width are used where bin width $\omega = \frac{V}{360}$ degrees, shifted by $\frac{\omega}{2}$ degrees as the camera is expected to be forward-facing, leaving the forwards road appearing in the centre of the midpoint bin. All reference bearing vectors $Q_{sat} = \{Q_0, ..., Q_N\}$ are computed prior to evaluation.

At query time, bearings Q_{street} may similarly be estimated from the streetview images. For example a semantic segmentation or BEV system can recognise areas of road in different directions. The query and reference junction vectors are then used to filter the image retrievals, discarding retrievals with incompatible bearing vectors. More formally, a retrieval is compatible if any bitwise shift operation of the query matches the retrieval. This operation results in filtered reference retrievals Q_{sat}^* from the overall reference set Q_{sat} whose bearing vectors equal the queries at some shift.

$$Q_{sat}^* = \{Q \in Q_{sat} | \exists v \text{ such that } Q_{street} = \text{shift}(Q, v)\} \quad (9)$$

Performance can be further increased if the vehicle's yaw is known. In this case, the input to the shift operation is defined by the yaw. Figure 6 illustrates the bearing filtering technique. The right-hand side displays retrievals determined from SpaGBOL along with their pre-computed bearing vectors. These are filtered using the query bearings vector, determined from the query image. In this example, the red-outlined embeddings are discarded as their

SpaGBOL			
Region	Nodes	Edges	Walks
Tokyo	4,815	7,942	95,044
London	3,155	4,124	30,634
Philly	2,272	3,782	47,774
Brussels	2,190	3,403	35,959
Boston	1,567	2,403	26,180
Guildford	1,472	1,773	11,247
Chicago	1,159	1,935	25,824
New York	1,103	1,983	29,668
Singapore	1,043	1,567	15,241
Hong Kong	995	1,440	13,270
Total	19,771	28,912	330,841
Streetview	98,855		
Satellite	19,771		
VIGOR-Graph			
Region	Nodes	Edges	Walks
New York	3,880	6,771	96,176
San Francisco	3,288	5,337	67,942
Seattle	3,039	4,697	51,370
Chicago	2,295	3,771	49,212
Total	12,502	20,576	264,700

Table 1. Graph Attributes - No. unique walk samples of length 4

vectors don't match the queries. The orange-outlined embedding is a partial match, with the correct road positions but misaligned. The green-outlined embedding shows a perfect match. Once retrievals have been filtered, the potential retrievals can be greatly narrowed down, increasing the probability of a correct localisation.

4. Results

4.1. Datasets

The most significant current CVGL datasets (CVUSA [33] and CVACT [11]) are unsuitable for conversion to a graph structure as the data is too sparse. We convert the older benchmark dataset VIGOR [34] into a graph structure, enabling similar assessment. VIGOR contains densely collected image pairs from four cities within the USA: New York, San Francisco, Chicago, and Seattle. To convert VIGOR to a graph representation, we first retrieve the graphs for each of these cities, with the same characteristics as SpaGBOL - nodes represent junctions and edges represent roads. Each node is then assigned the image pairs closest to their geographical coordinates. This results in 10,207 training nodes and 2,295 testing nodes - the system is evaluated with sampled walks in the same manner as with SpaGBOL. SpaGBOL's and VIGOR-Graph's characteristics are displayed in Table 1, with the total number of walks (when walk length $n = 4$) to demonstrate the extensive sampling capabilities when using graph structures.

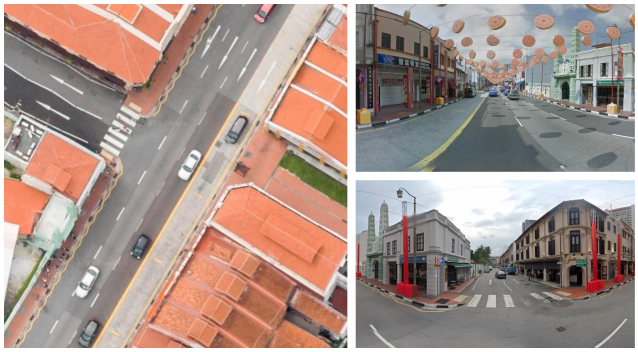

Figure 7. SpaGBOL node data: satellite image & 2/5 corresponding FOV-cropped streetview images - shown at different yaws.

SpaGBOL contains 18,204 Satellite-Streetview training+validation pairs and 1,567 testing pairs from across 10 cities, covering $2km^2$ per city. Satellite images are north-aligned with a resolution of 0.2metres / pixel covering $50m^2$ (note some of these images may have been captured from drones and other aerial image sources). Streetview images are yaw-aligned panoramas with a resolution of 2048×512. When limiting the FOV, images are cropped to the desired FOV with yaw rotated away from the previous node. We use Boston's graph as the test set, with the remaining cities used for training - separating a ninth of each training graph for validation, as shown in Figure 5. More in-depth information about the SpaGBOL dataset is given in the Supplementary Material.

4.2. Implementation Details

Image features are extracted with a ConvNext-T [35], producing 768-dimension embeddings. The sampled walk embeddings are passed through a GNN which outputs refined 64-dimension embeddings. All image embeddings affect network learning, but only the target node embeddings are retained for evaluation. A KDTree of satellite image embeddings is constructed. This is then queried with each streetview image to retrieve the K closest embeddings. Training occurs end-to-end, randomly sampling walks of length n for each node per epoch, also randomly selecting the streetview image from each node's streetview set. SpaGBOL is trained with walk triplets for 100 epochs using an AdamW optimiser with an initial learning rate of 1e-4 and a ReduceLROnPlateau scheduler. Graphs during validation and testing are distinct subsets, with one random query walk per node and exhaustive reference walks.

4.3. Evaluation

We evaluate with Top-K recall accuracy, similar to previous works [1], [3], and [20], though we enhance performance with retrieval filtering. A query is deemed successful if the correct node is within the Top-K retrievals. Top-K uses the absolute value of K for retrievals whereas

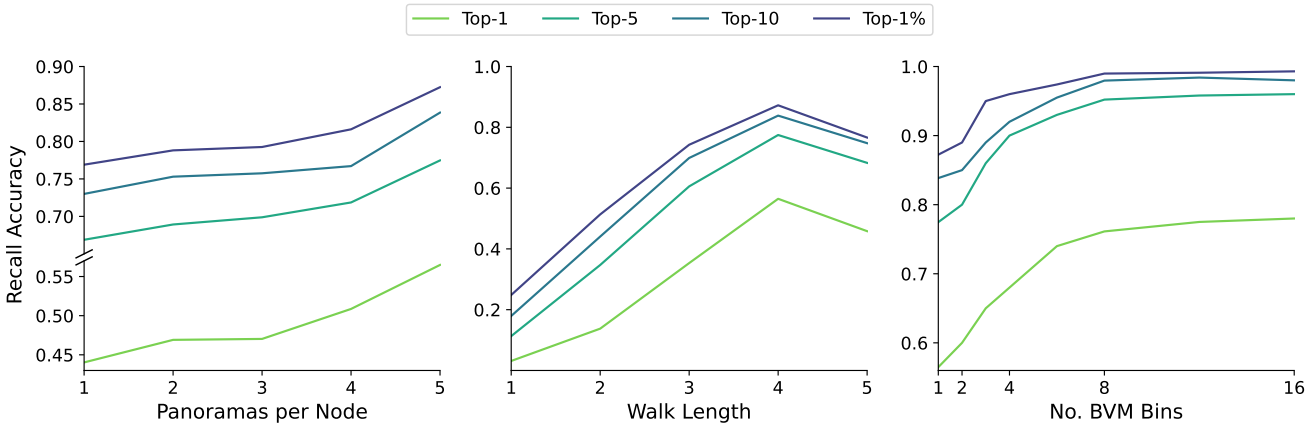

Figure 8. Impact on recall accuracies when SpaGBOL characteristics are varied.

Top-K% uses the K% length of the database. As we are proposing and releasing a novel dataset, we evaluate against previous CVGL works whose code is publicly available. We train each approach according to the optimal configurations outlined in their papers/code. As this is the first work to propose graph-based representations and techniques for CVGL, we performed a variety of experiments on previous works, aiming to increase fairness. One experiment averaged each embedding along sampled walks, another reduced potential reference embeddings at each stage along a walk by performing Top-K retrievals sequentially - aiming to increase retrieval accuracy. We found empirically that prior works achieve the greatest accuracy when treating each node as an independent retrieval. Thus we train competing techniques in this mode, to provide the strictest baseline possible. We also evaluate how each technique performs with limited-FOV images, including those originally designed for panoramic inputs. Table 2 outlines the performance for each work. *SpaGBOL* displays the performance of the network with simple embedding retrieval. *SpaGBOL+B* demonstrates how a system can exploit the ability to filter embeddings based on the angles and presence of neighbouring node's edges. For limited-FOV evaluation, these are only extracted from visible regions of the scene, impacting filtering capabilities. BVM is not utilised where FOV is below $180°$ due to lack of the required visual information. *SpaGBOL+YB* improves BVM's potential, displaying the increase in retrieval success when the yaw of the vehicle is also known, i.e. with access to a simple compass.

Results from both datasets show that our proposal achieves significant improvements over previous works, specifically when performing CVGL in densely sampled city-scale graphs. We demonstrated in Figure 8 that the inclusion of multiple streetview images per node improves generalisation - increasing test performance by approximately 10% for each metric, when increasing from one streetview image per node to five. Also showing that when evaluating with the SpaGBOL dataset, the optimal walk length was four, with performance dropping when exceeding this. Utilising our GNN-based network achieves performance increases of 11.18% on Top-1 retrievals on SpaGBOL. Also showing that the filtered GNN embeddings are more robust to reduced FOV inputs with our Top-1 relatively decreasing by approximately 12% compared to previous state of the art (SOTA) performance's reduction of 26%, when reducing input FOV to $180°$. Utilising graph characteristics which allow for our bearing filtering proposal, demonstrates that this can achieve relative performance increases beyond our standard retrieval system of $\approx$ 35% when FOV is $360°$, and 67% when FOV is $180°$.

4.4. Ablation Study

To verify components contribute as intended within our proposed system, we display an ablation of constituents in Table 3. The base model is only the feature extraction, trained for single-image retrieval as it has no graph walk capability. Adding our GNN greatly improved performance, outputting geo-spatially strong embeddings from the more discriminative network. We then add bearing vector filtering which further boosts performance around 15% by removing incompatible nodes. Finally, adding the camera yaw to the system optimised performance by filtering with aligned bearing vectors. We determine the optimal walk length of our system with the SpaGBOL dataset - varying the walk length of all sampled walks. Visible in Figure 8, the system's performance dramatically increases when walk lengths are larger than two - with the optimal for this dataset being random walks of length four. To improve generalisation of our network and future works, we include multiple streetview panoramas for each node in the graphs. These images were captured across a period of around a decade - leading to varying content, weather, and lightning.

SpaGBOL

FOV	360°				180°				90°			
Model	Top-1	Top-5	Top-10	Top-1%	Top-1	Top-5	Top-10	Top-1%	Top-1	Top-5	Top-10	Top-1%
CVM [7]	2.87	12.96	21.51	28.33	2.68	9.83	15.12	20.23	1.02	5.87	10.15	14.81
CVFT [14]	4.02	13.02	20.29	27.19	2.49	8.74	14.61	19.91	1.21	5.74	10.02	13.53
DSM [15]	5.82	10.21	14.13	18.62	3.33	9.74	14.66	21.48	1.59	5.87	10.11	16.24
L2LTR [36]	11.23	31.27	42.50	49.52	5.94	18.32	28.53	35.23	6.13	18.70	27.95	34.08
GeoDTR+ [19]	17.49	40.27	52.01	59.41	9.06	25.46	35.67	43.33	5.55	17.04	24.31	31.78
SAIG-D [2]	25.65	51.44	62.29	68.22	15.12	35.55	45.63	53.10	7.40	21.76	31.14	37.14
Sample4Geo [20]	50.80	74.22	79.96	82.32	37.52	**64.52**	71.92	76.39	6.51	20.61	30.31	36.12
SpaGBOL	**56.48**	**77.47**	**83.85**	**87.24**	**40.88**	63.79	**72.88**	**78.28**	**18.63**	**43.20**	**54.05**	**61.20**
SpaGBOL+B	64.01	86.54	92.09	94.64	52.01	82.20	89.47	93.62	-	-	-	-
SpaGBOL+YB	76.13	95.21	97.96	98.98	66.82	92.69	96.38	97.30	-	-	-	-

VIGOR-Graph

FOV	360°				180°				90°			
Model	Top-1	Top-5	Top-10	Top-1%	Top-1	Top-5	Top-10	Top-1%	Top-1	Top-5	Top-10	Top-1%
CVM [7]	1.83	7.80	11.90	22.75	1.79	5.49	9.63	16.99	1.39	4.31	8.58	15.08
CVFT [14]	5.01	12.99	18.48	28.93	1.96	6.28	9.89	16.51	1.31	3.57	6.28	11.29
DSM [15]	6.19	16.51	22.14	32.64	1.05	2.31	3.70	7.67	0.44	1.48	2.66	5.36
L2LTR [36]	6.41	17.52	26.45	37.91	3.09	8.37	12.20	20.78	1.87	6.75	10.12	17.08
GeoDTR+ [19]	3.09	11.07	17.08	28.24	2.05	6.71	11.20	20.22	1.48	5.19	9.37	17.43
SAIG-D [2]	7.63	17.47	24.92	36.17	5.27	14.55	21.79	32.81	2.88	7.97	13.16	21.00
Sample4Geo [20]	**32.03**	54.73	64.10	75.90	**13.92**	31.07	36.17	54.23	1.35	4.40	7.93	14.81
SpaGBOL	31.88	**57.99**	**67.47**	**77.56**	13.36	**31.53**	**41.66**	**54.59**	**6.51**	**18.95**	**27.07**	**41.22**
SpaGBOL+B	47.99	74.63	83.45	91.40	19.17	42.53	52.88	66.25	-	-	-	-
SpaGBOL+YB	58.21	81.49	88.69	94.32	21.88	47.25	58.17	69.96	-	-	-	-

Table 2. Benchmark Dataset Test Recall Accuracies.

	Train			
Model	Top-1	Top-5	Top-10	Top-1%
ConvNeXt-T	52.93	70.80	88.01	92.87
C+GNN	79.04	95.80	97.36	99.75
C+G+Bearing	84.03	97.92	99.55	99.91
C+G+B+Yaw	85.89	98.82	99.29	99.97
	Test			
ConvNeXt-T	15.00	44.80	60.00	67.58
C+GNN	56.48	77.47	83.85	87.24
C+G+Bearing	64.01	86.54	92.09	94.64
C+G+B+Yaw	76.13	95.21	97.96	98.98

Table 3. Ablation study demonstrating the performance impact from each component of SpaGBOL.

5. Conclusion & Future Work

In this paper, we successfully progress CVGL towards real-world application, demonstrating the benefits of advancing the field from single-image and image-sequence representations towards explicitly structured graphs. We release a comprehensive novel dataset focused on regions most likely to benefit from CVGL - dense GNSS-denied urban regions. We have presented an approach using graph representations and GNNs to significantly aid CVGL by exploiting the relationship between image features, their geographic proximity, and geo-spatial structures. Furthermore we have demonstrated how performance may be boosted by implementing BVM according to observed road bearings. Evaluating against previous approaches, we increase retrieval performances by more than 11.18% for Top-1 retrievals - boosting up to 49.86% when utilising the BVM capabilities of graph representation.

5.1. Future Work

We have demonstrated the utility of graphs for CVGL, effectively verifying various benefits of such approaches. However, there are some limitations that must be addressed in future works. Although closer to real-world feasibility than prior datasets/techniques, the granularity of our dataset limits precision - only capable of localising to the nearest road junction. Within our test set, the median length of edges is 73 metres. This could be naively addressed by incorporating additional sensors for localising between nodes, such as using an IMU for measuring between successful retrievals. Future works may overcome this obstacle by introducing hierarchical structures such as sub-graph representations for each edge on the corpus graph, allowing for secondary localisation once the nearest node has been determined against the city-scale graph.

6. Acknowledgements

This work was partially funded by the EPSRC under grant agreement EP/S035761/1, FlexBot - InnovateUK project 10067785, and the author was financially supported by G-Research.

References

[1] Sijie Zhu, Mubarak Shah, and Chen Chen. Transgeo: Transformer is all you need for cross-view image geo-localization. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1152–1161, 2022. 2, 6

[2] Yingying Zhu, Hongji Yang, Yuxin Lu, and Qiang Huang. Simple, effective and general: A new backbone for cross-view image geo-localization, 2023. 2, 8

[3] Tavis Shore, Simon Hadfield, and Oscar Mendez. Bev-cv: Birds-eye-view transform for cross-view geo-localisation, 2023. 2, 6

[4] Scott Workman and Nathan Jacobs. On the location dependence of convolutional neural network features. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 70–78, 2015. 2

[5] Tsung-Yi Lin, Yin Cui, Serge Belongie, and James Hays. Learning deep representations for ground-to-aerial geolocalization. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5007–5015, 2015. 2

[6] Nam N. Vo and James Hays. Localizing and orienting street views using overhead imagery. In *European Conference on Computer Vision*, 2016. 2

[7] Sixing Hu, Mengdan Feng, Rang M. H. Nguyen, and Gim Hee Lee. Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7258–7267, 2018. 2, 8

[8] Relja Arandjelović, Petr Gronát, Akihiko Torii, Tomás Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:1437–1451, 2015. 2

[9] Sijie Zhu, Taojiannan Yang, and Chen Chen. Revisiting street-to-aerial view image geo-localization and orientation estimation. *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 756–765, 2020. 2

[10] Bin Sun, Chen Chen, Yingying Zhu, and Jianmin Jiang. Geocapsnet: Ground to aerial view image geo-localization using capsule network. *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 742–747, 2019. 2

[11] Liu Liu and Hongdong Li. Lending orientation to neural networks for cross-view geo-localization. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5617–5626, 2019. 2, 6

[12] Yujiao Shi, Liu Liu, Xin Yu, and Hongdong Li. Spatial-aware feature aggregation for image based cross-view geo-localization. In *Neural Information Processing Systems*, 2019. 2

[13] Krishna Regmi and Mubarak Shah. Bridging the domain gap for ground-to-aerial image matching. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 470–479, 2019. 2

[14] Yujiao Shi, Xin Yu, Liu Liu, Tong Zhang, and Hongdong Li. Optimal feature transport for cross-view image geo-localization. *ArXiv*, 2019. 2, 8

[15] Yujiao Shi, Xin Yu, Dylan Campbell, and Hongdong Li. Where am i looking at? joint location and orientation estimation by cross-view matching. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4063–4071, 2020. 2, 8

[16] Aysim Toker, Qunjie Zhou, Maxim Maximov, and Laura Leal-Taix'e. Coming down to earth: Satellite-to-street view synthesis for geo-localization. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6484–6493, 2021. 2

[17] Hongji Yang, Xiufan Lu, and Ying J. Zhu. Cross-view geo-localization with layer-to-layer transformer. In *Neural Information Processing Systems*, 2021. 2

[18] Xiaohan Zhang, Xingyu Li, Waqas Sultani, Yi Zhou, and Safwan Wshah. Cross-view geo-localization via learning disentangled geometric layout correspondence, 2023. 2

[19] Xiaohan Zhang, Xingyu Li, Waqas Sultani, Chen Chen, and Safwan Wshah. Geodtr+: Toward generic cross-view geolocalization via geometric disentanglement, 2023. 2, 8

[20] Fabian Deuser, Konrad Habel, and Norbert Oswald. Sample4geo: Hard negative sampling for cross-view geo-localisation, 2023. 3, 6, 8

[21] Frauke Heinzle, Karl-Heinrich Anders, and Monika Sester. Graph based approaches for recognition of patterns and implicit information in road networks. In *Proceedings of the 22nd international cartographic conference*, pages 11–16. ICA Washington, DC, 2005. 3

[22] Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010. 3

[23] 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*, 30:25–39, 2011. 3

[24] Arun Annaiyan, Miguel A. Olivares-Mendez, and Holger Voos. Real-time graph-based slam in unknown environments using a small uav. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1118–1123, 2017. 3

[25] Jinhao He, Yuming Zhou, Lixiang Huang, Yang Kong, and Hui Cheng. Ground and aerial collaborative mapping in urban environments. *IEEE Robotics and Automation Letters*, 6(1):95–102, 2021. 3

[26] Olga Vysotska and Cyrill Stachniss. Lazy data association for image sequences matching under substantial appearance changes. *IEEE Robotics and Automation Letters*, 1(1):1–8, 2016. 3

[27] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3668–3678, 2015. 3

[28] Yu Liu, Yvan Petillot, David Lane, and Sen Wang. Global localization with object-level semantics and topology. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4909–4915, 2019. 3

[29] Francesco Giuliari, Geri Skenderi, Marco Cristani, Yiming Wang, and Alessio Del Bue. Spatial commonsense graph for object localisation in partial scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19518–19527, June 2022. 3

[30] Garar Örn Gararsson, Francesca Boem, and Laura Toni. Graph-based learning for leak detection and localisation in water distribution networks*. *IFAC-PapersOnLine*, 55(6):661–666, 2022. 11th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2022. 3

[31] Daniele Grattarola, Lorenzo Livi, Cesare Alippi, Richard Wennberg, and Taufik A. Valiante. Seizure localisation with attention-based graph neural networks. *Expert Systems with Applications*, 203:117330, 2022. 3

[32] Riku Murai, Joseph Ortiz, Sajad Saeedi, Paul H. J. Kelly, and Andrew J. Davison. A robot web for distributed many-device localization. *IEEE Transactions on Robotics*, 40:121–138, 2024. 3

[33] Scott Workman, Richard Souvenir, and Nathan Jacobs. Wide-area image geolocalization with aerial reference imagery. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1–9, 2015. 6

[34] Sijie Zhu, Taojiannan Yang, and Chen Chen. Vigor: Cross-view image geo-localization beyond one-to-one retrieval. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5316–5325, 2021. 6

[35] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s, 2022. 6

[36] Hongji Yang, Xiufan Lu, and Yingying Zhu. Cross-view geo-localization with layer-to-layer transformer. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 29009–29020. Curran Associates, Inc., 2021. 8