PEnG: Pose-Enhanced Geo-Localisation

Tavis Shore[1], Oscar Mendez[2], and Simon Hadfield[1]

***Abstract*—Cross-view Geo-localisation is typically performed at a coarse granularity, because densely sampled satellite image patches overlap heavily. This heavy overlap would make disambiguating patches very challenging. However, by opting for sparsely sampled patches, prior work has placed an artificial upper bound on the localisation accuracy that is possible. Even a perfect oracle system cannot achieve accuracy greater than the average separation of the tiles. To solve this limitation, we propose combining cross-view geo-localisation and relative pose estimation to increase precision to a level practical for real-world application. We develop PEnG, a 2-stage system which first predicts the most likely edges from a city-scale graph representation upon which a query image lies. It then performs relative pose estimation within these edges to determine a precise position. PEnG presents the first technique to utilise both viewpoints available within cross-view geo-localisation datasets, referring to this as Multi-View Geo-Localisation (MVGL). This enhances accuracy to a sub-metre level, with some examples achieving centimetre level precision. Our proposed ensemble achieves state-of-the-art accuracy - with relative Top-5m retrieval improvements on previous works of 213%. Decreasing the median Euclidean distance error by 96.90% from the previous best of 734m down to 22.77m, when evaluating with 90° horizontal FOV images. Code is available here: github.com/tavisshore/peng.**

***Index Terms*—Localisation, Vision-Based Navigation, Computer Vision for Transportation**

I. Introduction

LOCALISATION is vital in the majority of mobile robotics applications. Common techniques such as Global Navigation Satellite Systems (GNSS) provide absolute positioning data to clients. These are prone to failure in certain environments. One example are dense urban canyons such as New York City where tall buildings cause signal occlusions & reflections, preventing successful satellite communication. Another example are regions of conflict where malicious actors purposefully disrupt positioning by spoofing signals, inserting erroneous information.

Image localisation may provide a solution as agents can fully self-localise using onboard sensors, removing requirements for external communication. These techniques aim to relate an agent's query image with previously seen geo-tagged images, determining an updated position according to feature and positional similarities with these references. A large proportion of mobile robots are already equipped with cameras, increasing the viability of image localisation.

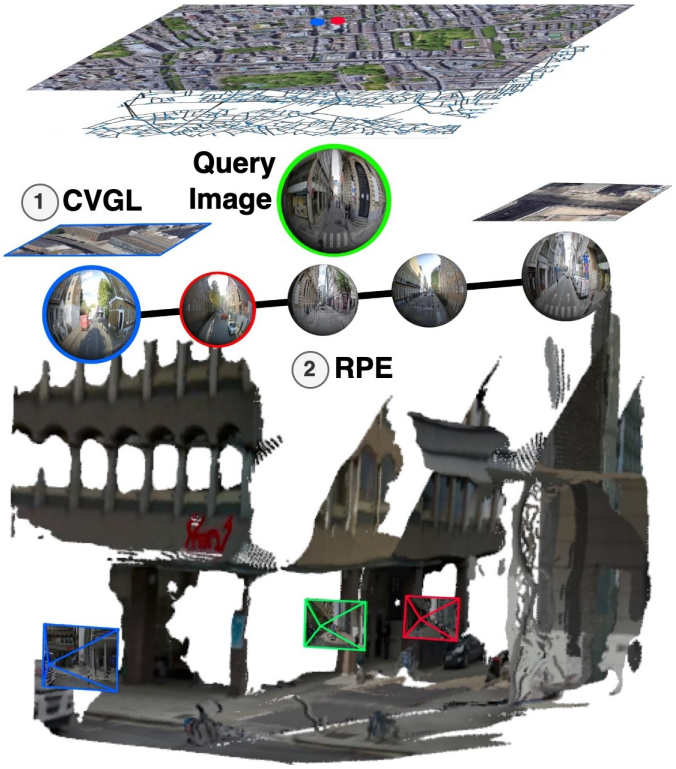

Fig. 1. PEnG Multi-View Stages: 1) City-scale satellite image with underlying graph network, Cross-View Geo-Localisation (CVGL) estimates candidate edges within city's graph. 2) Relative Pose Estimation (RPE) along these edges achieves refined geographic poses. Green denotes a query input, blue and red display two known reference images.

Cross-View Geo-localisation (CVGL) is an increasingly popular branch of image localisation research, offering a viable form of generalisable wide-scale image localisation. The objective is to relate a street-level query image to a database of reference satellite images - returning the geographic coordinates of the highest correlating known satellite image. Pose estimation is a related field that aims to determine the pose of a camera within a scene. These techniques generally operate on a smaller scale than CVGL, localising within a few metres, rather than in whole cities. They generally operate as continuous prediction, rather than retrieval problems, and operate in N-Degrees of Freedom (DoF) as opposed to simple geographic coordinates. Pose estimation has two primary subfields - Absolute Pose Estimation (APE) and Relative Pose Estimation (RPE). APE aims to determine the position and orientation of a camera within a 3D world coordinate frame. RPE aims to compute the same, but with respect to a reference camera.

Manuscript received: November, 23, 2024; Revised February, 13, 2025; Accepted February, 21, 2025.

This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the EPSRC under grant EP/S035761/1 and FlexBot - InnovateUK project 10067785.

[1]Tavis Shore and Simon Hadfield are with the Centre for Vision, Speech and Signal Processing, University of Surrey, United Kingdom {t.shore, s.hadfield}@surrey.ac.uk

[2]Oscar Mendez is with Locus Robotics, Kent, United Kingdom omendez@locusrobotics.com

Digital Object Identifier (DOI): see top of this page.

We propose leveraging the advantages of both techniques in a single two-stage system to achieve high-precision city-scale localisation, shown in top-down order in Figure 1. Taking as input a street-level image, the first stage performs city-wide CVGL, predicting the most recently observed road junction. Operating the CVGL stage at the scale of the road junctions helps to keep the reference set lean and discriminative, ensuring efficient and accurate results of coarse location retrieval.

The second stage takes the CVGL sub-region predictions and performs RPE along neighbouring roads, merging likelihoods from both stages to determine a final 3-DoF pose. This novel combination of learned computer vision techniques achieves a reduction in the median localisation error from 734m to 22.77m, evaluating with 90° crops of the StreetLearn dataset [1].

In summary, our research contributions are as follows:

- Introduce the first technique for performing precise image localisation in a city-scale by utilising information from both image viewpoints in CVGL datasets, creating Multi-View Geo-Localisation (MVGL).
- Introduce emulating a simple compass, filtering reference embeddings according to a configurable yaw threshold, greatly increasing localisation precision.
- Demonstrate strong generalisation to cities not seen in training - localising with a median error of 22.77m within the large dense region of Manhattan, considering an area of $36.1km^2$.

II. Related Works

A. Camera Pose Estimation

RPE can be divided into two categories: feature matching, and pose regression. More traditional camera localisation techniques often utilise structure-based methods, representing a scene with an explicit SfM or SLAM reconstruction [2, 3, 4]. This often requires a large number of images to have already been captured within a scene, limiting generalisation.

Shotton et al. [5] introduce a novel method called Scene Coordinate Regression Forest (SCoRe Forest) for inferring the pose of an RGB-D camera relative to a known 3D scene using a single image with decision forests. Kendall et al. propose PoseNet [6], the first CNN designed for end-to-end 6-DOF camera pose localisation, evaluating the network thoroughly to prove the viability of deep learning for the field. In their following paper [7], they apply a principled loss function based on the scene's geometry to learn camera pose without any hyper-parameters, achieving state of the art (SOTA) results, reducing the performance gap to traditional methods. Sattler et al. [4] propose using a prioritised matching approach, considering features more likely to yield 2D-to-3D matches, terminating searches once sufficient matches have been found. Brachmann et al. [8] propose *DSAC*, a differentiable counterpart to RANSAC, replacing the deterministic hypothesis selection with a probabilistic selection, deriving the expected loss with respect to all learnable parameters. Applying this to image localisation achieved higher accuracies than previous deep learning based methods. Clark et al. [9] propose extending to sequential camera pose estimation, designing an RNN which achieves smoothed poses and greatly reduced localisation error. Sarlin et al [10] propose *HFNet* - performing coarse-to-fine image localisation by predicting local features and global descriptors for 6-DoF localisation simultaneously.

Map-free Relocalisation [11] introduces using a single photo from a scene for metric scaled re-localisation, negating the requirement to construct a scaled map of the scene. Rockwell et al. [12] propose *FAR*, combining correspondence estimation and pose regression techniques to utilise the benefits from both to provide precision and generalisation. Wang et al. [13] and Leroy et al. in the follow-up paper [14] propose *Dust3r* and *Mast3r* respectively. Both are techniques for dense unconstrained stereo 3D reconstruction of arbitrary image collections, with no prior information. Mast3r achieves SOTA performance in various fields including camera calibration and dense 3D reconstruction. Moreau1 et al. [15] propose *CROSSFIRE* - using NeRFs as implicit scene maps and propose a camera re-localisation algorithm for this representation. *CROSSFIRE* achieves SOTA accuracy and is capable of operating in dynamic outdoor environments.

Similar to how *FAR* proposed combining multiple pose estimation paradigms to achieve SOTA performance in that particular sub-field, we propose combining multiple image localisation techniques to achieve high precision localisation in large scale regions with different input modalities.

B. Cross-View Geo-Localisation

Current CVGL techniques primarily focus on embedding retrieval - extracting reduced dimensionality representations of reference satellite images, aiming to return geo-coordinates from those most similar to query images. Techniques are being increasingly proposed to improve performance by manipulating extracted features, [16], [17], [18].

Workman and Jacobs [19] first propose CNNs for learning feature relationships across viewpoints. This was extended by Lin et al. [20], treating each query uniquely, utilising Euclidean similarities for retrieval. Vo and Hays [21] add rotation information through an auxiliary loss, evaluating misalignment impact. CVM-Net [22] add NetVLAD [23] to the CNN, aggregating local feature residuals to cluster centroids. Liu and Li [24] increase access to orientation information, improving the latent space robustness. [16] computes feature correlation between ground-level images and polar-transformed aerial images, shifting and cropping at the strongest alignment before performing image retrieval. L2LTR [25] developed a CNN+Transformer network, combining a ResNet backbone with a vanilla ViT encoder to increase performance over SOTA.

In GeoDTR [17, 26], Zhang et al. separate geometric information from the raw features, learning spatial correlations within visual features to enhance performance. BEV-CV [18] introduces Birds-Eye-View (BEV) transforms to the field, reducing representational differences between viewpoints to create more similar embeddings. Sample4Geo [27] propose two CVGL sampling strategies, geographically sampling for optimal training initialisation, mining hard-negatives according to feature similarities between viewpoints. SpaGBOL [28]

propose progressing the CVGL field from single and sequential representations to graph-based representation, allowing for more geo-spatially strong embeddings.

Fine-grained CVGL estimates the pose of a streetview image within it's known corresponding satellite image. Shi et al. [29] propose utilising a differentiable Levenberg-Marquardt (LM) to iteratively search for the optimal camera pose. In [30], Lentsch et al. horizontally divide panoramas into *slices* and apply cross-view attention to create a ground slice-specific aerial feature map, learning the correlation between each slice across viewpoints. Xia et al. [31] estimate a dense probability distribution over the satellite image, capturing the underlying multi-modal distribution - achieving results robust against small orientation variation. In their subsequent work *CCVPE* [32], they propose a translational equivariant encoder to learn orientation-aware descriptors for localisation and orientation estimation, achieving SOTA results.

Fine-grained CVGL methods provide computationally efficient and robust results but heavily depend on prior knowledge of the correct corresponding satellite image. We aim to overcome the limitation of fine-grained techniques in distinguishing between estimating a pose within the correct corresponding satellite image and an incorrect one.

CVGL approaches to date have followed a retrieval paradigm where the accuracy of results is limited by the granularity of the geo-referenced database. Sparsely sampled data can lead to higher retrieval rates due to greater feature dissimilarities, while densely sampled data may enhance localisation precision but decrease performance, as overlapping satellite image patches increase the likelihood of incorrect retrievals.

III. Methodology

A. City-Scale Geo-Localisation Data Representation

We frame CVGL as a graph comparison problem, similar to the technique demonstrated in *SpaGBOL* [28]. Where *SpaGBOL* established a lower bound on localisation precision by only applying graph nodes at road junctions, we incorporate orders of magnitude more nodes by placing *secondary nodes* along existing edges, enhancing the density of data. These graphs now have two classes of nodes, denoted *primary* nodes N - representing road junctions, and *secondary* nodes Q - captured along roads at varying intervals. This significant increase in data density greatly increases the precision upper bound. Figure 2 shows a section of this graph representation of Manhattan.

We represent each region in the dataset $i \in \{Manhattan, ...\}$ as a separate graph $G_i = (N, Q, E)$ with primary nodes $N_i = \{n_1, n_2, ..., n_N\}$, secondary nodes $Q_i = \{q_1, q_2, ..., q_Q\}$, edges $E_i = \{e_{1,2}, e_{1,3}, ..., e_E\}$. Edges $e_{a,b}$ represent roads connecting primary nodes a and b. Each node in both classes has attributes - $\{I_{sat}, I_{street}, L, \Psi, B\}$, containing a panoramic streetview image and a satellite image - both RGB: $I_j \in \mathbb{R}^{3\times W\times H}, j \in \{street, sat\}$, location $L = \{\phi, \lambda\}$ consists of geographical latitude and longitude coordinates, $\Psi \in \mathbb{R} : \{-180° \leq \Psi \leq 180°\}$ is the north-aligned camera yaw, and $B = \{\beta_1, ..., \beta_K\}$ are north-aligned bearings to K neighbouring nodes - where $\beta \in \mathbb{R} : \{-180° \leq \beta \leq 180°\}$.

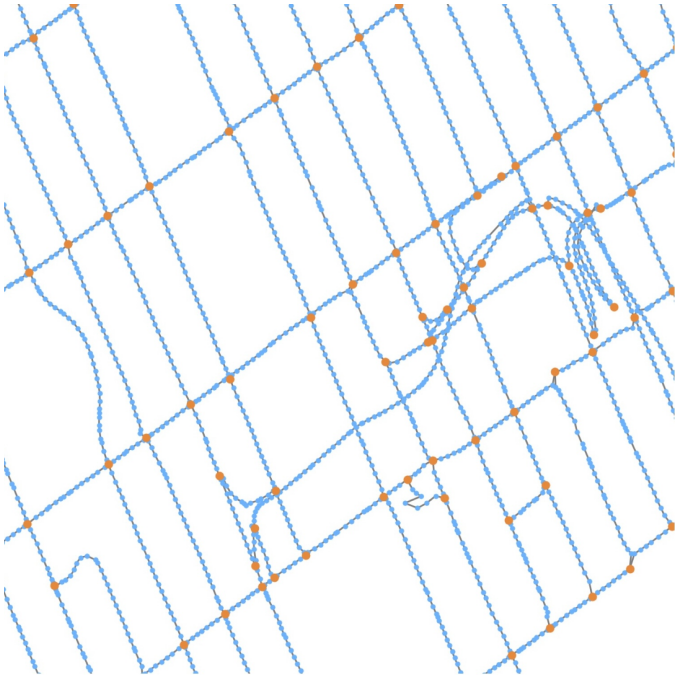

Fig. 2. Section of Manhattan graph with primary (orange) and secondary (blue) nodes displayed. Most edges have a constant yaw, motivating the utilisation of a compass.

We limit the streetview image's (I_{street}) Field-of-View (FOV) to increase the technique's feasibility as a large proportion of existing vehicles possess monocular cameras. Cameras are assumed to be fixed to the vehicle in a forward-facing configuration. We experiment with FOVs, $\Theta \in \{70°, 90°, 120°\}$.

B. PEnG Procedure

Our proposed technique, PEnG, operates in two stages utilising multiple views, described in Figure 3: initially estimating *candidate primary nodes* with graph-based CVGL (shown on the left-hand side) before performing RPE relative to the secondary nodes present along each *candidate edge* until a threshold is met, or all candidate edges have been processed. The main purpose of the first stage is to reduce the number of reference images when performing relative pose estimation. This enables city-scale pose estimation as without it, pose estimation takes orders of magnitude longer.

1) Graph-Based Cross-View Geo-Localisation: We perform CVGL following the standard procedure as used within previous works [18, 22, 16]. We implement a siamese-like network of CNN feature extractors f with no weight sharing. The network is parametrised by θ to produce similar embeddings η_t from corresponding streetview-satellite image pairs. Creating a database of reference embeddings offline, querying this database for retrievals during online operation.

$$\eta_t = f_\theta\left(I_t\right), t \in \{street, sat\} \tag{1}$$

In the first stage, CVGL retrievals are performed on primary nodes N_i to provide efficient and accurate initial filtering.

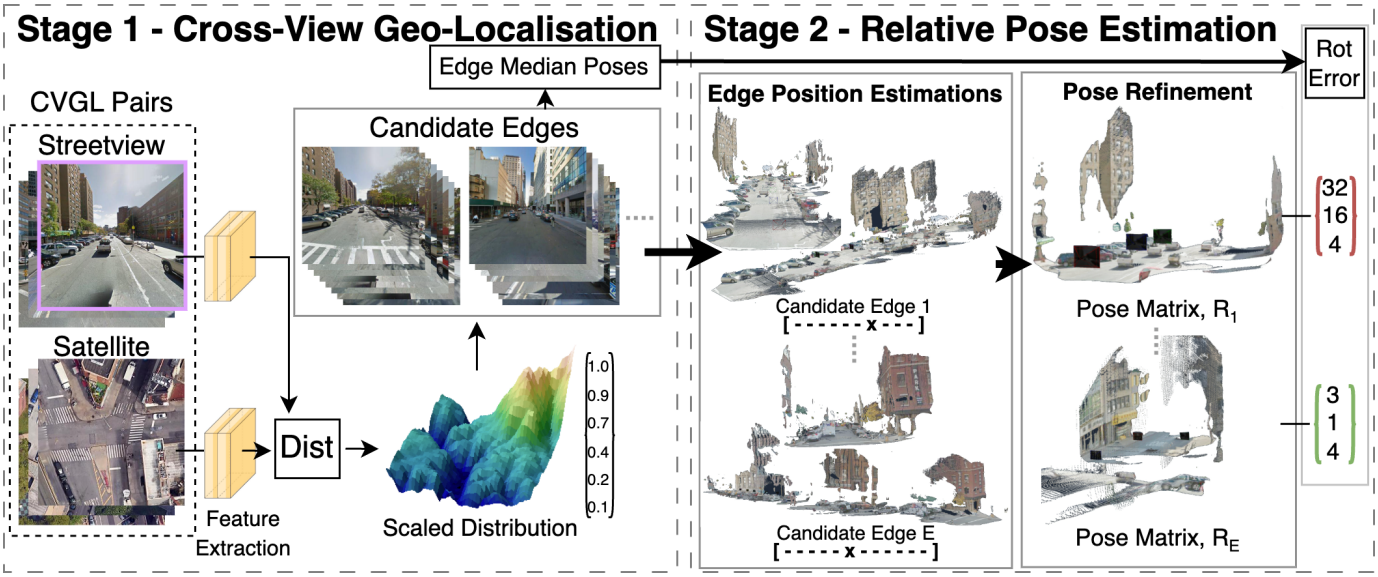

Fig. 3. Overview of PEnG MVGL technique. Features are extracted and sorted by cross-view Euclidean distance. Stage 1 determines a scaled distribution of CVGL retrievals from only primary nodes, determining secondary nodes with a threshold. Stage 2 executes through edges consecutively until an error threshold is met or completion. This stage has two sections: an initial estimate of position along edge to determine adjacent reference images before refining the pose estimate against these. The pose error is the MSE between the estimate and the edge's median pose.

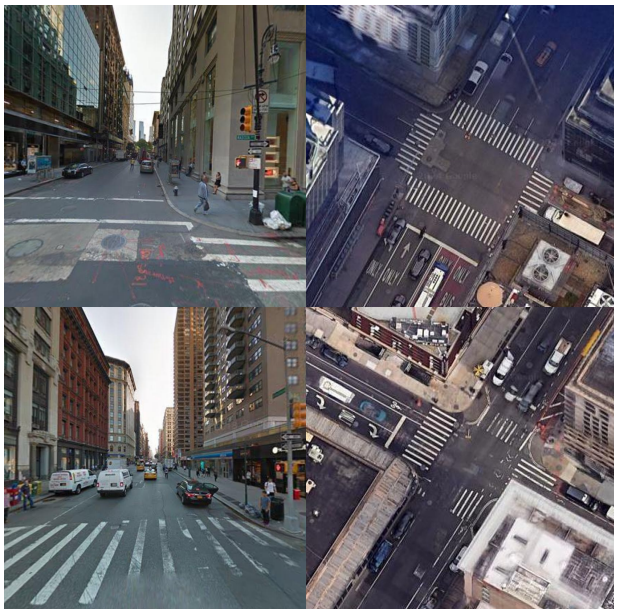

Fig. 4. Example primary node (road junction) cross-view image pairs. Left-hand side shows 90° crops from panoramas and the right-hand side shows aerial images at zoom 20.

Retrieved embeddings are ordered by descending similarity with the query. They are then min-max scaled, $\mathcal{M}$, giving a confidence score c_i for each candidate node—concluding this stage.

$$c_i = \mathcal{M}\left(\frac{\eta_i^{query} \cdot \eta^{ref}}{\|\eta_i^{query}\|\|\eta^{ref}\|}\right) \tag{2}$$

Candidate nodes, C_k, are passed to the second stage depending on the confidence threshold θ_c, and maximum number of candidates k.

$$C_k = \{c_i | c_i > \theta_c \text{ or } i < k\} \tag{3}$$

2) Pose Refinement: For each candidate node, c, we select that candidate's connected edges, $E_c = \{e_{i,j} | i = c \text{ or } j = c\}$. We then filter these edges by matching the compass heading and the edge's yaw within the graph. For every remaining candidate edge, we then perform RPE in two stages: first estimating a coarse position of the query image along an edge before refining this relative to the two neighbouring reference secondary nodes. The calculation of median edge rotational pose is displayed in Figure 5.

Inspired by [14], we determine the relative pose of query images against each candidate edge's secondary node, before combining the poses across the entire edge. For each image pair along an edge I^1 & I^2, we determine the set of cross-image pixel correspondences. We then use a transformer network based on Mast3r [14] to predict 3D pointmaps in real-time as needed, $X^{1,1}, X^{1,2}$, from 2D points x^i between these images, expressed in the coordinate frame of I^1. The pointmaps are then compared $X^{1,1} \longleftrightarrow X^{1,2}$, computing the relative poses with RANSAC & PnP [33] expressed in equations 4 and 5.

The objective of PnP is to minimise the reprojection error between the 3D points and their corresponding 2D image projections:

$$x^i = K(RX^i + t) \tag{4}$$

Where x^i is the projected 2D point, X^i is the 3D world point, K is the estimated camera intrinsic matrix, R & t are the rotation and translation matrices. RANSAC randomly samples 4 points for PnP, optimising to estimate R and t.

We compute reprojection error $e_i = ||x_i - K(RX_i + T)||$, rejecting outliers based on a predefined threshold ϵ. We then

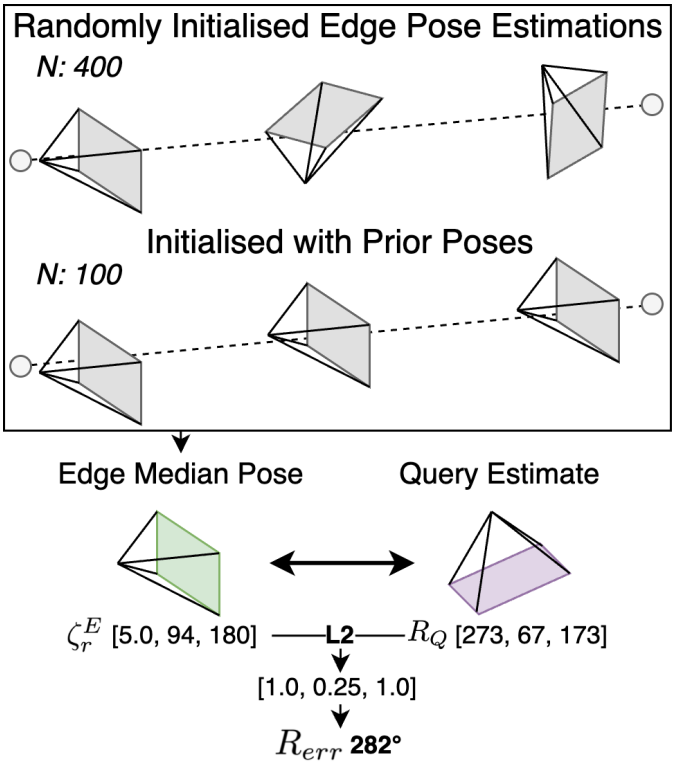

Fig. 5. Pose estimates within each candidate edge are scored by their 3-axis Euclidean distance with the mean rotational pose of the secondary nodes. This is possible due to the known orientations of edges within graph representations. N denotes the maximum number of RANSAC iterations in the pose estimation.

maximise the number of inliers $e_i \leq \epsilon$ to achieve the best pose estimate (R^*, t^*):

$$(R^*, t^*) = \underset{R,t}{\text{argmax}} \sum_i \mathbf{1}(e_i \leq \epsilon) \tag{5}$$

where $(e_i \leq \epsilon)$ is the indicator function - equals 1 if e_i is less than or equal to a predefined threshold ϵ, 0 otherwise.

Precomputation - All reference poses, P^r, are estimated prior to system operation, calculating a median 3-DoF rotational matrix for each edge ζ_r^E. As this is a preprocessing step, a larger number of iterations, N, are used compared to during inference. These pre-determined poses then initialise optimisation processes during operation, reducing the required number of iterations - leading to lower operating times without effecting performance.

Operation - Algorithm 1 is executed for each query image, until thresholds such as Maximum Rotational Error θ_{re} or No. Candidate Nodes θ_n are achieved. Rotational error R_{err} is the 3-DoF summed Euclidean distance between the query rotation R_Q and the median edge rotation ζ_r^E. This is calculated with an $[X, Y, Z]$ axis weighting of $[1, 0.25, 1]$ as roll has a smaller impact on performance. Where a query has multiple pose estimations and an L2 distance threshold has not been met, each pose is given a confidence score - rotational errors are summed and min-max scaled to between 0 & 1.

Algorithm 1 PEnG Algorithm

Require: Graph $G = (N, Q, E)$, Reference Primary Node Database η_N^{sat}, Query and Reference images I_Q^{street} I_R^{sat}, Thresholds $\theta_x \in \{\theta_{pe}, \theta_n, ...\}$, Reference Poses ζ_r^E

Ensure: R_Q

Stage 1 - CVGL
$\eta^{street} = f_\theta(I^{street})$
$S = \mathcal{N}\left(\left(\sum_k \eta_{ik}^{ref} \eta_k^{query}\right)\right)$

Stage 2 - Pose Estimation
$i = 0$
while thres$(R_{err} \leq \theta_x)$ **do**
  $E_{cand} = \text{filter}(N(S_i), \Psi)$
  $I^{pairs} = \textit{exhaustive}(E_{cand} + I^{street})$
  $t_p = \text{RPE}_{position}(I^{pairs})$
  $(R^i, t^i) = \text{RPE}_{pose}(I^{t_{p-1}}, I^{t_{p+1}})$
  $R_{err}^i = \text{sim}_{euc}(R^i, E_{cand})$
  $i = i + 1$

return Absolute Pose Estimations R_Q

Confidence scores from both stages are considered to determine a final pose estimation, calculated by scaling the relative poses to between the edge's ground truth limits.

IV. Results

A. Datasets

The feature extractors for both PEnG and previous works are trained with the CVUSA dataset [34], cropping streetview images to various FOVs, portraying front-facing road-aligned monocular images. This dataset contains $35,532$ streetview-satellite training pairs and $8,884$ validation pairs. CVUSA satellite images have a resolution of 750×750 and streetview panoramas of 1232×224, both north-aligned. We evaluate with the StreetLearn Manhattan dataset [1]. Example image pairs are shown in Figure 4. Manhattan is selected for evaluation as it qualifies as an urban canyon - an environment category that often experiences GNSS failure. The city's data are converted from unconnected images into a graph representation. This contains $53,289$ images, comprising $2,622$ primary nodes and $50,667$ secondary nodes. The graph covers approximately 31.6km^2. Satellite images are north-aligned with a resolution of 0.20metres/pixel covering 50m^2 (some images may have been captured from drones and other aerial image sources). Streetview images are yaw-aligned panoramas with a resolution of 1664×832. The median distance between the primary nodes is 116m, and the median distance between adjacent secondary nodes is 9.83m. As both training and evaluation datasets contain yaw values at image capture, we can produce limited-FOV front-facing crops, emulating a monocular camera - our expected input for real-world CVGL application for autonomous vehicles.

B. Implementation Details

Image features are extracted with a ConvNext-T [35] pre-trained on ImageNet-1K [36], producing 768-dimension em-

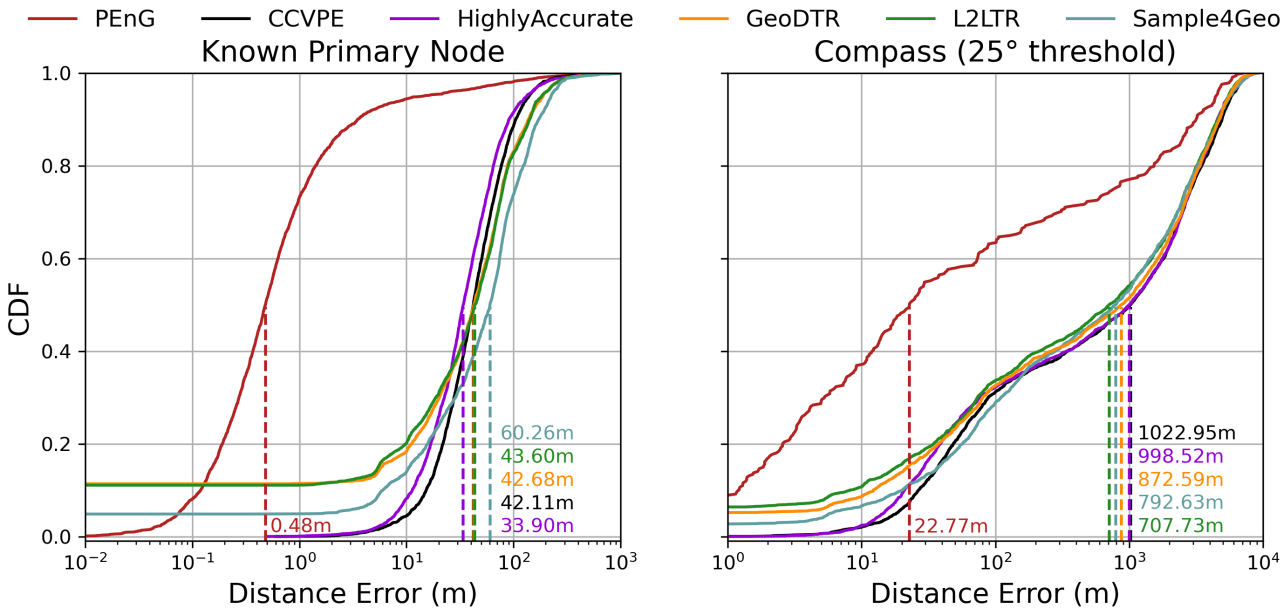

Fig. 6. Cumulative Distribution Functions show the significant decrease in distance error achieved with PEnG. Some works are non-zero at $x = 0$ as there is 0m error when they correctly retrieve the corresponding correct satellite image.

beddings. When evaluating against SpaGBOL [28] we instead use their trained feature extractor - a combination of a ConvNext-T CNN with a GraphSage GNN, generating low-dimensional vector representations. We perform this second evaluation with randomly sampled depth-first walks from the graph. We filter candidate edges by emulating a compass alongside the query, discarding incompatible graph edges. This is possible due to the graph representation - with known orientations between the primary node and it's connected edges.

All existing CVGL baselines are also augmented with this compass filtering technique to ensure a balanced assessment. The pointsmap extraction transformer is trained with the InfoNCE loss and a simple mean distance loss function for 35 epochs. We use a median pose error threshold of 3°, halting execution if a match is found with a weighted Euclidean distance below this. In the rare case that all edge pose estimates have an error larger than this threshold, the estimate with lowest error is selected. The feature extractor is trained with FOVs $\in \{70°, 90°, 120°\}$ for 50 epochs using an AdamW optimiser with an initial learning rate of $1e-4$ and a ReduceLROnPlateau scheduler. The preset poses stored for reference points are calculated offline with a learning rate of 0.1 and 400 iterations, which are refined when online with a learning rate of 0.1 and 100 iterations.

C. Ablation Study

To verify the contribution of each constituent in the proposed system, we display an ablation study in Table I. CVGL shows the performance of the simple ConvNeXt-T feature extractor, evaluated in the same method as previous works - filtering by primary nodes initially to reduce the reference set. *1 Pose* performs pose estimation against an entire edge's reference images, determining a relative 2-DoF pose between primary nodes. *2 Pose* follows *1 Pose* with a refined pose estimation relative to the 2 adjacent reference secondary nodes, determined in the first pose estimation step - this enables a high precision final estimate. *Pose Priors* is the addition of estimating the pose of all secondary nodes prior to querying, increasing the accuracy of reference poses and offloading a portion of computation to an offline stage.

TABLE I
ABLATION STUDY

Config	Med (m)	Top-1m	Top-5m	Top-25m
CVGL	961	6.06	6.94	9.27
1 Pose	32.12	7.02	25.45	45.12
2 Pose	28.94	7.31	26.41	47.91
Pose Priors	22.77	9.12	29.18	51.37

Evaluated with 90° Crops

The ablation shows the vast decrease in median distance error achieved by combining these two localisation techniques, the median error decreases by an order of magnitude. Having a pose refinement stage after the initial position estimation further decreases median error by $\approx$ 3m. Finally, estimating reference poses prior to operation increased accuracy relatively by $\approx 10\%$.

D. Evaluation

We evaluate with distance-based Top-K accuracy, displaying Euclidean distance errors as Cumulative Distribution Function (CDF) plots in Figure 6. Table II shows these in discretised metrics, defining estimates as successful if they are within K-metres of the ground truth. All works follow the 2-stage process of CVGL within the primary node set before estimating position within the secondary node set. To demonstrate the generality of the PEnG approach we present results with both a traditional retrieval first stage, PEnG, and a graph-based first stage, PEnG*.

TABLE II
EVALUATION

Model	Type	Med (m)	Top-1m	Top-5m	Top-25m
	FOV	70°			
L2LTR [37]	CVGL	826	6.48	7.55	10.03
GeoDTR+ [26]	CVGL	903	5.19	6.03	8.47
Sample4Geo [27]	CVGL	897	6.79	7.78	10.41
Beyond [29]	FG	999	0.00	0.53	8.96
CCVPE [38]	FG	843	0.08	0.50	12.36
PEnG	MVGL	**26.82**	**7.25**	**27.43**	**49.01**
SpaGBOL [28]	GB	634	6.37	7.70	10.56
PEnG*	MVGL	*29.31*	*6.86*	*26.16*	*47.75*
	FOV	90°			
L2LTR [37]	CVGL	750	6.64	8.01	10.64
GeoDTR+ [26]	CVGL	854	6.06	7.25	9.80
Sample4Geo [27]	CVGL	734	8.35	9.31	12.43
Beyond [29]	FG	1032	0.00	0.34	9.34
CCVPE [38]	FG	941	0.11	0.65	11.14
PEnG	MVGL	**22.77**	**9.12**	**29.18**	**51.37**
SpaGBOL [28]	GB	529	6.33	7.25	9.69
PEnG*	MVGL	*34.91*	*7.17*	*25.10*	*47.29*
	FOV	120°			
L2LTR [37]	CVGL	732	7.82	9.19	12.05
GeoDTR+ [26]	CVGL	893	6.75	7.63	10.60
Sample4Geo [27]	CVGL	703	**9.50**	10.68	14.42
Beyond [29]	FG	977	0.04	0.46	8.05
CCVPE [38]	FG	979	0.00	0.76	11.94
PEnG	MVGL	**37.72**	4.04	**21.21**	**44.93**
SpaGBOL [28]	GB	501	*6.90*	7.86	10.14
PEnG*	MVGL	*45.46*	3.36	*22.04*	*44.24*

Stage 1 threshold $= 0.9$. Best CVGL result in **bold**, best graph-based result in *italic*. CVGL denotes cross-view geo-localisation works, FG shows fine-grained cvgl works, and MVGL denotes multi-view geo-localisation.

To ensure a fair comparison when evaluating our two-stage technique against previous CVGL works, we adopt the two-stage retrieval approach to assist these prior works. We fist run the first CVGL stage, selecting the final estimate by running the CVGL system again on the collection of secondary nodes and choosing the most similar retrieved secondary node. For previous works, we retain the ground truth corresponding satellite image, as seen in Figure 6, where the distributions do not start from 0. In real-world applications, this would be infeasible since the reference set cannot contain precisely geographically aligned ground truth satellite images. However, this approach provides a stronger baseline for comparison. For fine-grained CVGL methods, we first applied our CVGL network in the initial stage, followed by their pose estimation techniques for the selected secondary nodes. The final estimate is the one with the lowest error to the median pose, consistent with the PEnG technique.

The evaluation shows that our proposal achieves significant improvements over current SOTA. With $90°$ images, we achieve a 96.90%% reduction in median error, and an approximate 213%% increase in Top-5m accuracy. We note that using $90°$ FOV images achieves a relative decrease in the median error of $\approx$ 4m compared to $70°$. This is due to the increase in information available to each stage. However, further increasing the FOV to $120°$ yields a decrease in localisation precision. This may be caused by the input image dimensionality limitation of our model - due to the backbone pre-training, the maximum image resolution for the system is 512×384, placing an upper bound on how much information can pass through the system. Another hindrance is experienced from extracting perspective images from a $360°$ panorama. When increasing the horizontal FOV beyond $90°$, these crops begin to display visibly distortion.

Within the discretised Top-Km metrics, PEnG performs slightly worse than previous works where $K < 5$ due to the inherent zero error bias in existing CVGL works. As K reaches 25m, performance is significantly higher across Fields-of-View (FOVs). As precisely centred ground-truth corresponding satellite images are known for each query streetview image in CVGL, they tend to perform unrealistically well with these Top-K metrics. This peculiarity of previous evaluation protocols is visible in Figure 6 where at $x = 0$, previous works start from a non-zero values.

V. CONCLUSION & FUTURE WORK

A. Conclusion

We successfully propose and demonstrate the utility of combining graph-representations, CVGL, and relative pose estimation techniques. This ensemble is proven to be a viable strategy for progressing CVGL within a large city-scale environment towards practicality, reducing median distance errors from hundreds of metres down to often centimetre level accuracy. PEnG achieves SOTA localisation precision when evaluated within the Manhattan region of $36.1km^2$, reducing the median error from Sample4Geo's previous best of 734m down to 22.77m when operating with $90°$ FOV. In our ablation studies, we thoroughly demonstrate the significance of each portion of the 2-stage architecture, validating that the combination results in the maximum precision possible for PEnG. We release code for converting the StreetLearn dataset into the graph representation outlined above, along with PEnG technique's code and corresponding pretrained weights, enabling future works to build upon the technique and further evaluate this ensemble.

B. Future Work

There are various limitations within the work and several aspects which can be optimised to advance the field towards real-world application. Firstly, storage requirements of PEnG place a limitation on the system, more so than previous CVGL system designs. However, as physical size and cost of storage reduces tremendously over recent years, with 1TB MicroSD's available for less than £100, PEnG may be applied across dense urban centres, potentially selecting which city representation with GPS, before communication becomes unreliable inside the urban canyon, and switching to PEnG. Secondly, the significant viewpoint disparity in CVGL means that the performance of the first stage constrains the potential precision of the second stage. A more probabilistic fusion approach could help mitigate this limitation.

ACKNOWLEDGMENT

This work was partially funded by the EPSRC under grant agreement EP/S035761/1 and FlexBot - InnovateUK project 10067785.

References

[1] P. Mirowski et al. "The StreetLearn Environment and Dataset". In: *ArXiv* abs/1903.01292 (2019).

[2] E. Royer et al. "Monocular Vision for Mobile Robot Localization and Autonomous Navigation". In: *IJCV* 74.3 (2007), pp. 237–260.

[3] A. Irschara et al. "From structure-from-motion point clouds to fast location recognition". In: *CVPR*. 2009, pp. 2599–2606.

[4] T. Sattler, B. Leibe, and L. Kobbelt. "Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization". In: *TPAMI* 39.9 (2017), pp. 1744–1756.

[5] J. Shotton et al. "Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images". In: *CVPR* (2013), pp. 2930–2937.

[6] A. Kendall, M. Koichi Grimes, and R. Cipolla. "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization". In: *ICCV* (2015), pp. 2938–2946.

[7] A. Kendall and R. Cipolla. "Geometric Loss Functions for Camera Pose Regression with Deep Learning". In: *CVPR* (2017), pp. 6555–6564.

[8] E. Brachmann et al. "DSAC — Differentiable RANSAC for Camera Localization". In: *CVPR* (2016), pp. 2492–2500.

[9] R. Clark et al. "VidLoc: A Deep Spatio-Temporal Model for 6-DoF Video-Clip Relocalization". In: *CVPR* (2017), pp. 2652–2660.

[10] P. Sarlin et al. "From Coarse to Fine: Robust Hierarchical Localization at Large Scale". In: *CVPR*. 2019, pp. 12708–12717.

[11] R. Arnold et al. "Map-free Visual Relocalization: Metric Pose Relative to a Single Image". In: *ECCV*. 2022, pp. 690–708.

[12] C. Rockwell et al. "FAR: Flexible, Accurate and Robust 6DoF Relative Camera Pose Estimation". In: *CVPR* (2024), pp. 19854–19864.

[13] S. Wang et al. "DUSt3R: Geometric 3D Vision Made Easy". In: *CVPR*. 2024, pp. 20697–20709.

[14] V. Leroy, Y. Cabon, and J. Revaud. "Grounding Image Matching in 3D with MASt3R". In: *ECCV*. 2024, pp. 71–91.

[15] A. Moreau et al. "Crossfire: Camera relocalization on self-supervised features from an implicit representation". In: *ICCV*. 2023, pp. 252–262.

[16] Y. Shi et al. "Where Am I Looking At? Joint Location and Orientation Estimation by Cross-View Matching". In: *CVPR* (2020), pp. 4063–4071.

[17] X. Zhang et al. "Cross-view Geo-localization via Learning Disentangled Geometric Layout Correspondence". In: *AAAI*. 2022.

[18] T. Shore, S. Hadfield, and O. Mendez. "BEV-CV: Birds-Eye-View Transform for Cross-View Geo-Localisation". In: *IROS* (2023), pp. 11048–11055.

[19] S. Workman and N. Jacobs. "On the location dependence of convolutional neural network features". In: *CVPRW*. 2015, pp. 70–78.

[20] T. Lin et al. "Learning deep representations for ground-to-aerial geolocalization". In: *CVPR*. 2015, pp. 5007–5015.

[21] N. Vo and J. Hays. "Localizing and Orienting Street Views Using Overhead Imagery". In: *ECCV*. 2016, pp. 494–509.

[22] S. Hu, M. Feng, and R. Nguyen. "CVM-Net: Cross-View Matching Network for Image-Based Ground-to-Aerial Geo-Localization". In: *CVPR*. 2018, pp. 7258–7267.

[23] Relja A. et al. "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition". In: *TPAMI* 40 (2015), pp. 1437–1451.

[24] L. Liu and H. Li. "Lending Orientation to Neural Networks for Cross-View Geo-Localization". In: *CVPR*. 2019, pp. 5617–5626.

[25] H. Yang, X. Lu, and Y. Zhu. "Cross-view Geo-localization with Layer-to-Layer Transformer". In: *NeurIPS*. Vol. 34. 2021, pp. 29009–29020.

[26] X. Zhang et al. "GeoDTR+: Toward Generic Cross-View Geolocalization via Geometric Disentanglement". In: *TPAMI* 46 (2023), pp. 10419–10433.

[27] F. Deuser, K. Habel, and N. Oswald. "Sample4Geo: Hard Negative Sampling For Cross-View Geo-Localisation". In: *ICCV* (2023), pp. 16801–16810.

[28] T. Shore, O. Mendez, and S. Hadfield. "SpaGBOL: Spatial-Graph-Based Orientated Localisation". In: *WACV*. 2025.

[29] Y. Shi and H. Li. "Beyond Cross-view Image Retrieval: Highly Accurate Vehicle Localization Using Satellite Image". In: *CVPR*. 2022, pp. 16989–16999.

[30] T. Lentsch et al. "SliceMatch: Geometry-Guided Aggregation for Cross-View Pose Estimation". In: *CVPR* (2022), pp. 17225–17234.

[31] Z. Xia et al. "Visual Cross-View Metric Localization with Dense Uncertainty Estimates". In: *ECCV*. 2022, pp. 90–106.

[32] Z. Xia, O. Booij, and J. Kooij. "Convolutional Cross-View Pose Estimation". In: *TPAMI* 46.5 (2024), pp. 3813–3831.

[33] M. Fischler and R. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". In: *Commun. ACM* 24.6 (1981), pp. 381–395. ISSN: 0001-0782.

[34] S. Workman, R. Souvenir, and N. Jacobs. "Wide-Area Image Geolocalization with Aerial Reference Imagery". In: *ICCV*. 2015, pp. 1–9.

[35] Z. Liu et al. "A ConvNet for the 2020s". In: *CVPR* (2022), pp. 11966–11976.

[36] J. Deng et al. "ImageNet: A large-scale hierarchical image database". In: *CVPR*. 2009, pp. 248–255.

[37] H. Yang, X. Lu, and Y. Zhu. "Cross-view Geo-localization with Layer-to-Layer Transformer". In: *NeurIPS*. Vol. 34. 2021, pp. 29009–29020.

[38] X. Wang et al. "Fine-Grained Cross-View Geo-Localization Using a Correlation-Aware Homography Estimator". In: *NeurIPS* 36 (2024).