Maps from Motion (MfM): Generating 2D Semantic Maps from Sparse Multi-view Images

Matteo Toso[1] Stefano Fiorini[1] Stuart James[1,2] Alessio Del Bue[1]

[1]Istituto Italiano di Tecnologia (IIT) {name.surname}@iit.it
[2]Durham University stuart.a.james@durham.ac.uk

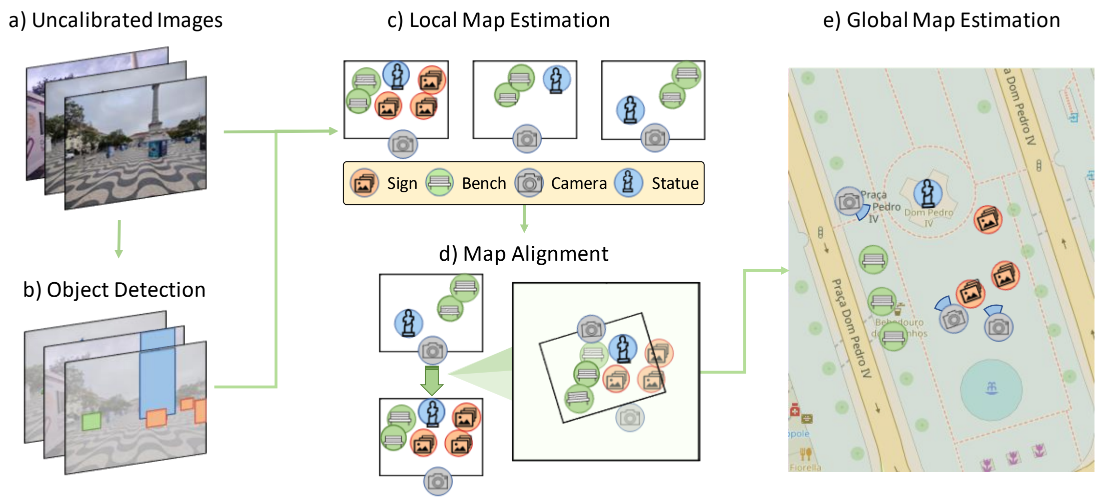

Figure 1. *The Maps from Motion (MfM) Concept.* From a set of uncalibrated images (a) we extract 2D detections of static urban objects (b), and generate local top-down 2D maps representing the spatial arrangement of the objects with respect to each image (c). We then learn how to register all local maps in the same reference frame (d), to generate a common global map with all objects present in the scene (e).

Abstract

World-wide detailed 2D maps require enormous collective efforts. OpenStreetMap is the result of 11 million registered users manually annotating the GPS location of over 1.75 billion entries, including distinctive landmarks and common urban objects. At the same time, manual annotations can include errors and are slow to update, limiting the map's accuracy. Maps from Motion (MfM) is a step forward to automatize such time-consuming map making procedure by computing 2D maps of semantic objects directly from a collection of uncalibrated multi-view images. From each image, we extract a set of object detections, and estimate their spatial arrangement in a top-down local map centered in the reference frame of the camera that captured the image. Aligning these local maps is not a trivial problem, since they provide incomplete, noisy fragments of the scene, and matching detections across them is unreliable because of the presence of repeated pattern and the limited appearance variability of urban objects. We address this with a novel graph-based framework, that encodes the spatial and semantic distribution of the objects detected in each image, and learns how to combine them to predict the objects' poses in a global reference system, while taking into account all possible detection matches and preserving the topology observed in each image. Despite the complexity of the problem, our best model achieves global 2D registration with an average accuracy within 4 *meters (*i.e. *below GPS accuracy) even on sparse sequences with strong viewpoint change, on which COLMAP has an* 80% *failure rate. We provide extensive evaluation on synthetic and real-world data, showing how the method obtains a solution even in scenarios where standard optimization techniques fail. Find more information at matteot90.github.io/MapsFromMotion.*

1. Introduction

2D semantic maps provide top-down abstract representations of an environment annotated with the location of easily identifiable landmarks, and play an important role in everyday life. In most public spaces, like museums and parks, we are used to finding our way via minimalist maps that mark our current location as a red, "You are here 📍" dot.

In recent years, works like OrienterNet [51], SNAP [52] and Flatlandia [59] have highlighted the advantages of using semantic maps in Computer Vision. They are more storage efficient than traditional 3D maps [1, 36, 54], requiring $1-10\%$ of the memory used by a set of reference images or a 3D point cloud [59]. Moreover, semantic maps provide an abstract representation that is robust to temporal changes. While a 2D map might sound limiting, in several scenarios (*e.g.* autonomous cars or robotics), cameras have roll angle 0 and y-axis colinear with the gravity direction [58], and have a constant height from the ground [12]; this makes localization as GPS location and a viewing direction sufficient. We explore the possibility of localizing on abstract global 2D maps composed only of the spatial top-down view layout of the objects observed in the images.

Existing approaches for generating 2D maps with object annotations present several limitations, in terms of time (*e.g.* requiring manual user annotations [9]), computational cost (*e.g.* large SfM reconstructions [59]), and specialized additional data (*e.g.* aerial images [52]). In contrast, we suggest that directly reasoning in 2D to combine partial maps from different viewpoints is a more efficient approach.

Given a sparse set of images, we frame the reconstruction of the 2D map as the registration of partial maps, based on the estimated arrangement of the objects detected in each image. As shown in Figure 1, we take a set of uncalibrated, unsorted images of the scene (1.a) and extract from them detections of common urban objects (1.b). Then, using the nominal camera intrinsics and monocular depth estimation, we generate local maps representative of the observed objects' layout in the camera reference system (1.c). We want to align these local maps in a common reference system (1.d), resulting in a global map (1.e); this, however, requires inferring a transformation (roto-translation and scale) from each map to a common global reference frame. We name this novel problem *Maps from Motion* (MfM).

The proposed task presents several difficulties: using only the estimated location of objects is an efficient representation [59], but it carries less information than traditional BEV images. Representing objects as a single coordinates also limits the semantic classes available, since larger elements like roads and buildings, while typically used in semantic maps, are too large to be localized as a single point without making the problem unstable. Meanwhile, using sparse images and focusing on common urban objects, that typically have plain and standardized appearance, makes establishing object matches from the input images unreliable. Given the success of Graph Neural Network (GNN) to address geometrical reasoning problems [17, 23, 53], we frame MfM as a graph problem, assigning a node to each detection and attempting to regress its location in the global map. In this formulation, we use same-map edges to force the network to preserve the topology of each local map, and same-class edges to account for all possible detections matches, instead of explicitly matching the input detections. This amounts to training a network to find the best alignment between the local maps, while preserving the object's layout observed in each image. To investigate the ability of graph networks to solve the MfM problem, we compare several architectures, with and without an attention mechanism. Experimenting on the Flatlandia dataset [59], we show how, despite the noisy detections and the absence of explicit cross-image detection matches, we can achieve object and camera localization accuracy comparable to COLMAP, while achieving a 60% lower failure rate on sparse sequences with strong viewpoint changes. Even in this challenging scenario, the best-performing implementation of our solution achieves a median localization error of less than 4 meters, better than standard GPS accuracy (4.9 m[1]). Our contributions are the following:

- We introduce a new problem (MfM) that provides an object scene map from a sparse set of uncalibrated images to automatize 2D map making procedures.
- We propose a new graph structure to address the MfM correspondence and registration problem, along with a GNN framework that estimates the positions of cameras and objects in a 2D global reference frame.
- We provide a new dataset and an evaluation protocol for MfM demonstrating the feasibility of the problem, and offer a comparison against relevant baselines.

2. Related Work

We discuss the creation (Section 2.1), representation (Section 2.2), and use (Section 2.3) of 2D semantic maps.

2.1. Creation of Semantic Maps

The creation of semantic maps has often been intertwined with 3D modeling, as their models - *e.g.* 3D point-cloud reconstructions from Structure from Motion (SfM) - provide enough information to allow localizing 3D objects [32, 49, 59]. Alternatively, 3D objects can be parameterized as ellipsoids [67], which, in turn, can also be used jointly for camera pose estimation from elliptical detections [10, 20, 41]. This parameterization is also used in SLAM [16, 40, 42]. Such methods result in accurate models, but they generally are computation and memory intensive, and prone to failure if the images are too sparse or have large viewpoint changes.

[1] www.gps.gov/performance/accuracy

Other methods use object detections to create object-maps. If camera calibration is available, an option is to use 3D projection [38] and refining using graph neural networks [39] to produce an object-level map. However, these approaches impose strong limitations on the diversity of objects (i.e. trees) and the camera parameters they can handle. Object detection are also used to re-identify buildings for camera localization [63], matching detections across images [13], and making feature matching more robust [4].

Alternatively, some works have focused on on generating accurate bird-eye-view maps from input images [34, 50] or on improving maps' accuracy by fusing street-level images with additional sensor (*e.g.* like satellite or aerial images [52]). Such additional data provide helpful localization cues, but aerial data and fleets of sensorized cars are viable only for large enterprises, limiting accessibility.

Finally, platforms like OpenStreetMaps [9] generate large semantic maps via crowd-sourcing. This is time and resource expensive, with annotation density varying significantly across areas, and conflicting or old annotations can result in outdated maps. Unlike these methods, we directly compute minimalist 2D semantic maps from images without an intermediate 3D model, characterizing the local maps as sets of 2D coordinates with a class label.

2.2. Graph Representation of scenes

The use of graphs to represent scenes is most common in scene graphs from image [11, 18, 24] or 3D models [2, 21]. Here, the content of a scene is encoded in graph form and passed to a GNN, trained on tasks like relational labeling [21], object localization [22] and robot navigation [48].

Moreover, many works use graph to encode the spatial representation of 2D Map data for representing and inferring map attributes [3, 25, 26, 65] or for downstream tasks like traffic management [44]. Other works, closer to MfM, align 3D representations to a 2D reference map [7, 31], propagating the 2D information to elements of the 3D model. This problem is simpler than MfM, involving a single alignment problem instead of multiple ones.

GNNs are also used to solve the 3D camera pose estimation problems. Recent works have formalized this problem as motion averaging [43, 57, 64] and Bundle Adjustment (BA) [6] through Graph Neural Networks (GNNs) [14, 15, 28, 33], typically modeling the scene as a graph of connected cameras to update or regress the 6DoF camera pose.

These approaches rely on 3D data to make their problems more treatable. Such data, however, is often not available, making it appealing to find a solution to the challenging problem of sparse scene estimation problem in 2D.

2.3. Use of Semantic Maps

Most semantic map-based localization methods require specialized hardware or large datasets. Qin *et al*. [45] use semantic information (e.g., lanes, signs, obstacles) but rely on surround-view cameras. Wen *et al*. [62] localize with lane lines and poles from HDMaps built using LIDAR odometry. Qin *et al*. [46] employ precise semantic maps created with RTK-GNSS and IMU. Gawel *et al*. [19] compare reference and query graphs, needing odometry, a graph database, and registered images. Cheng *et al*. [8] localize GPS coordinates within 30m and refine poses with 2D-3D PnP. Kong *et al*. [30] introduce a graph-based scene representation, similar to our approach, but without fine-grained localization. Recent works [37, 51, 61, 66] use OpenStreetMaps [9] reference maps, containing vector tiles for urban elements (e.g., building outlines, roads). Localization is achieved by comparing visual query embeddings to reference map information [37, 61, 66], or regressing 3DoF poses using bird's-eye view neural maps generated by GNNs [51].

3. Methodology

MfM addresses the problem of generating a **2D global map** from a set of sparse and wide-baseline images. As illustrated in Figure 2, this is done across three main steps. First, the images are turned into **local maps** (Figure 2.a), representing the spatial arrangement of the objects detected in the images and their classes. These maps are obtained by reprojecting the center of the object detections in 3D and then projecting it on the ground plane, as detailed in Section 3.1. MfM combines the local maps by forming a graph (Figure 2.b), in which the detections are represented as nodes, the local maps as subgraphs, and the cross-map connections as edges (Section 3.2). We then train a GNN to aggregate the information and regress the location of each detected object in the reference system of a global map (Figure 2.c), while preserving the spatial layout of each local map.

3.1. Local Map Estimation

Given a set of K images $\mathcal{I} = \{I_i\}_{i\in[1,\dots,K]}$, we estimate a set of local maps $\mathcal{M} = \{M_i\}_{i\in[1,\dots,K]}$. First, we use the Panoptic object detection algorithm [29] to identify the objects in the scene. This results in a detection o_{ij} with semantic label l_{ij} for each object j observed in the image. We then estimate the distance d_{ij} of j from image i using a monocular depth estimation algorithm [47] and averaging the predicted per-pixel depth over the detection, following [59]. Finally, we represent the coordinates of the detection in the image as a bounding box b_{ij}. The use of monocular depth estimation and averaging the depth of the whole detection into a single point will introduce some noise, and for this reason we also evaluate the proposed approach using perfect depth (Sec. 4.3) and perform ablation tests on the effect of noise on the local maps (Sec. 4.4).

Finally, we generate the local maps by reprojecting the center of each bounding box in 3D, and then projecting it onto the scene's ground plane. We then define the location

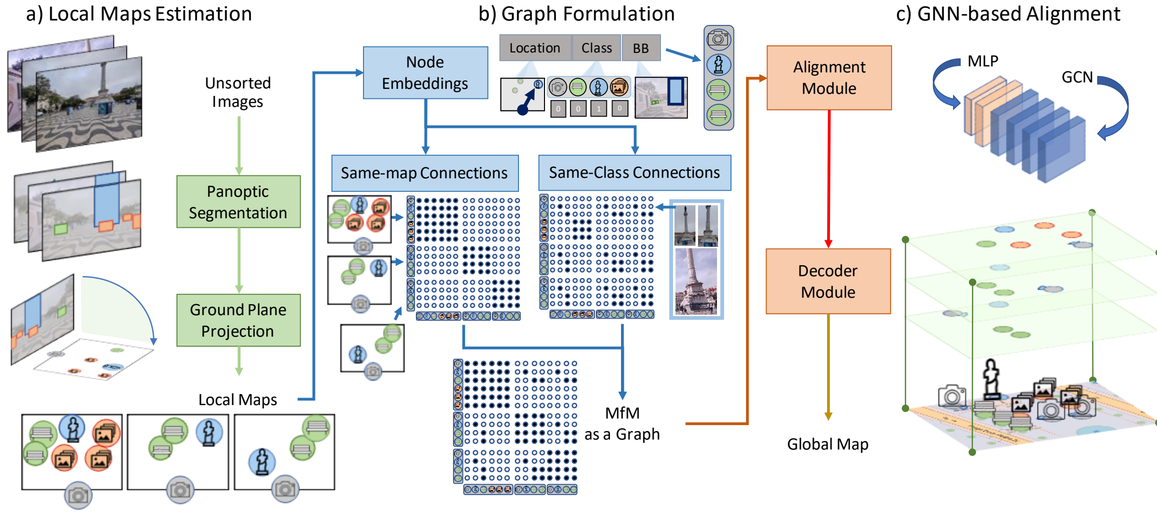

Figure 2. *The MfM Pipeline*. a) We extract 2D maps representing the spatial arrangement of detected objects, from the image's point of view. b) The maps are encoded as a graph, with a node for each detection, and edges connecting detections from the same image (Same-Map) or with the same class label (Same-class). c) A GNN predicts the location of all object and the cameras in one reference frame.

$\mathbf{c}_{ij} \in \mathcal{R}^2$ of j in the local map M_i as:

$$\begin{bmatrix} \mathbf{c}_{ij} \\ 1 \end{bmatrix} = \Pi \left(d_{ij} \; K_i^{-1} \begin{bmatrix} \mathbf{b}_{ij} \\ 1 \end{bmatrix} \right), \tag{1}$$

where Π is the projection from 3D to a horizontal plane orthogonal to the image plane I_i, $\mathbf{b}_{ij}$ is the center of the bounding box b_{ij}, and K_i is the intrinsic camera matrix associated with image i, obtained from the nominal characteristics of the camera. Equation (1) projects the objects out of the image plane assuming a pinhole camera model, and we define the 2D location of the camera that captured the image as the origin of the local map, *i.e.* as coordinates $[0, 0]$.

3.2. Graph Formulation for the MfM Problem

We frame the MfM alignment problem in graph form, and solve it using distribution of class labels and spatial layout, instead of explicitly matching detections across views.

First, we define for each image I_i an undirected subgraph $G_i = \{V_i, E_i\}$, with nodes V_i and edges E_i . We assign a node to each object detection, and one to the camera associated with I_i. The edges E_i connect all nodes V_i, making G_i a fully connected graph. We then aggregate the local maps in a large, undirected graph $\tilde{\mathcal{G}} = \{\tilde{V}, \tilde{E}\}$, whose nodes are defined as the union of all subgraph's nodes $\tilde{V} = \cup_{G_i} V_i$. The edges $\tilde{E}$ are defined as $\tilde{E} = (\cup_{G_i} E_i) \cup E_l$, where E_l is the set of edges connecting all nodes associated to the same class label, *i.e.* possible detection matches.

From the set of edges $\tilde{E}$, we define a binary adjacency matrix $A \in \{0, 1\}^{|\tilde{V}| \times |\tilde{V}|}$; this is a binary matrix with block diagonal elements connecting object detections from the same image and off-block diagonal values connecting detections from different images but with compatible classes, *i.e.* that could correspond to the same object. No such same-class edges are assigned to the camera nodes, as each represents the unique point of view of the corresponding images.

Given the graph $\tilde{\mathcal{G}}$, we encode in each of its nodes i the relevant information from the corresponding image and local map. We define the embedding $\psi(c_{ij}, l_{ij}, b_{ij}) = [c_{ij}, \sigma(l_{ij}), \epsilon(b_{ij})]$, where c_{ij} are the coordinates of the object or camera in its local map; $\sigma(l_{ij})$ is the one-hot encoding of the class; and $\epsilon(b_{ij})$ is the bounding box fitted to the detection of the object in the input image. For the camera nodes there is no detection, and we set $\epsilon(b_{ij}) = 0$.

3.3. GNN-based Alignment Module

We then introduced a three-stage alignment module. *i)* An encoder $\Psi(\cdot)$ composed of a fully connected linear layer with GeLu activation projects the aggregated initial embedding $\psi = [\psi(c_{ij}, l_{ij}, b_{ij})]_{\tilde{V}}$ into a higher-dimensional space, such that $\psi' = \Psi(\psi) \in \mathcal{R}^F$, where F is the dimension of the embedding space. *ii)* The projected embedding is fed to a GNN ($\Xi(\cdot, \cdot)$), producing an updated embedding $\psi''(\cdot) = \Xi(\psi', \tilde{\mathcal{G}})$. The proposed method can be adapted for a wide range of popular GNN, and a comparative study is available in Section 4. *iii)* We project into coordinates on a map as $\hat{c} = \Phi(\psi'') \in \mathcal{R}^2$, where Φ is a decoder based on a fully-connected layer. This outputs the 2D positions of the objects in a common reference frame, *i.e.* the global map. The alignment module is trained in supervised fashion, minimizing the linear combination of three different losses:

Euclidean Camera-Object Pose Loss is the difference between the predicted $\hat{c}_j$, and GT c_j^{GT} pose of object j:

$$\mathcal{L}_e = \frac{1}{|\tilde{\mathcal{V}}|} \sum_{j=1}^{|\tilde{\mathcal{V}}|} \|\hat{c}_j - c_j^{GT}\|_2^2, \quad (2)$$

Cross-Map Consistency Loss is the variance of the $\hat{c}_j$ predicted for detections of object j, plus the distance between mean pose μ_j and ground truth c_j^{GT}, *i.e.* it is minimized when nodes matching the same object have the same predicted global pose, and this coincides with the GT pose:

$$\mathcal{L}_\mu = \frac{1}{T} \sum_{j=1}^{T} \left(\frac{1}{N_j} \sum_{i=1}^{N_j} \|\hat{c}_i - \hat{\mu}_j\|_2^2 + \|c_j^{GT} - \hat{\mu}_j\|_2^2 \right) \quad (3)$$

$$\hat{\mu}_j = \frac{1}{N_j} \sum_{j=1}^{N_j} \hat{c}_j, \quad (4)$$

where T is the number of physical objects in the scene and N_j is the number of nodes corresponding to object j.

Self-Similarity Loss measures the consistency between the input local map and the predicted object locations in the global reference system. For each subgraph G_i, the original coordinates in the local map $c^0_{G_i} = \{c_i\}_{i \in G_i}$ and the predicted coordinates in the global reference system $\hat{c}_{G_i} = \{\hat{c}_i\}_{i \in G_i}$ define the self-consistency loss as:

$$\mathcal{L}_\sigma = \sum_{G_i \in \tilde{G}} \|c^0_{G_i} - (\lambda R \cdot \hat{c}_{G_i} + \tau)\|, \quad (5)$$

$$\lambda, R, \tau = \Theta(c^0_{G_i}, \hat{c}_{G_i}) \quad (6)$$

where Θ is a 2D Procrustean alignment algorithm that finds the rigid transformation - defined by a rotation angle θ, a scaling factor λ and a translation τ - that solves the problem:

$$\Theta(c_i, c_j) = \arg\min_{\theta,\tau,\lambda} \|c_i - (\lambda R(\theta) \cdot c_j + \tau)\|. \quad (7)$$

To solve equation (7), we design a differentiable 2D alignment algorithm; implementation and a complete derivation are available in the Supplementary Material. This loss reflects the fact that, within each local map, the relative spatial arrangement of the objects should be unchanged, regardless the coordinate system (global or local).

Total Loss uses hyperparameters λ_e, λ_μ and λ_σ to balance the three loss terms:

$$\mathcal{L} = \lambda_e \mathcal{L}_e + \lambda_\mu \mathcal{L}_\mu + \lambda_\sigma \mathcal{L}_\sigma, \quad (8)$$

Given the intrinsic ambiguity in choosing the reference system of the reconstruction, we expressing the GT always in the reference system of the first subgraph. At inference, the 2D coordinates regressed by the Alignment Module $\hat{c}$ are assumed to be the location of the observed objects in a global reference system, as illustrated in Figure 2.c.

City	Barcelona	Berlin	Lisbon	Vienna	Paris	Avg.
Scenes	9	2	22	3	20	11.2
Avg. Objects	18.2	70.5	27.1	65.7	17.4	39.8
Avg. Images	16.2	86	31	127	21.9	56.4
Track Length	5.8	4.3	4.7	5.3	4.9	5

Table 1. Statistics on the scenes from extracted from Flatlandia. We report for each city the number of distinct sub-scenes and their average number of objects, images and average track length.

4. Experiments

The MfM problem is new, and there are no readily available datasets and baselines in the literature. Therefore, we construct a dataset from Flatlandia [59], a related dataset for single-view camera localization on object maps in 3DoF. We also generate a synthetic dataset used for ablations on MfM. Both datasets are described in Section 4.1.

All models are first tested on the real-world data, to investigate whether the MfM problem is solvable with the proposed approach (Section 4.3). Then, we use the synthetic dataset to perform ablation tests on the effect of visibility and noise in the local maps on MfM (Section 4.4).

All evaluations report performance in terms of localization accuracy in meters for the predicted camera (μ_c) and objects (μ_o) and their respective standard deviations σ_c, σ_o computed over three runs initialized with different seeds; for each experiment we highlight the **best** and <u>second best</u> result. We also show, where appropriate, the failure rate of the methods. This is defined as the percentage of scenes where the error is significant *i.e.* greater than 7.5 meters, or for which it was not possible to generate a reconstruction. The latter case happened, for example, when COLMAP cannot find enough keypoint matches to initialize the SfM reconstruction process. As a final remark on evaluation, MfM is also subject to gauge freedom [35] as most reconstruction algorithms do, *i.e.* the global map predicted by MfM is not in the same reference frame of the ground truth data. We, therefore, use the alignment algorithm of Equation (7) to register the predicted objects and camera locations onto the ground truth global map. This also allows us to report the performance on a metric scale.

4.1. Datasets

***MfM Dataset - based on* [59].** The Flatlandia [59] dataset addresses the problem of 3DoF visual localization, and provides 2D local and reference maps obtained in 20 locations over 5 European cities. The dataset provides 20 reference maps annotated with a total of 2967 static objects, 6.3k reference images, and 2k query images. For the latter, the dataset provides local maps and their correspondences to the reference maps' objects. The local maps are provided in two forms *i)* ground truth: where the position of objects and the camera is based on the global map (i.e. perfect maps);

and *ii)* noisy ones obtained using MiDaS [47], a generic monocular depth estimation model, to estimate the layout of detected objects (Depth). These represent a noisy limit case. The local maps are obtained using the intrinsic parameters provided by the image metadata, without a calibration step. This is a reasonable approximation, with values within 5% from the intrinsics calibrated using COLMAP. The dataset's ground truth is obtained by reconstructing the scenes using all data and the COLMAP [55] SfM approach, the gold standard for reconstruction from multi-view images and a common baseline for scene reconstructions.

We construct a graph where each image from [59] is a node. Image pairs are connected if they have detections of at least three objects in common. Then, we extract all connected subgraphs from each graph—sets of images with a sufficient number of matched detections, *i.e.* three, for our specific task. This pre-processing results in two dataset:

- **MfM Large.** We use the entire dataset and produce 56 sub-graps, with 32 images on average per subgraph and 7 detections per image; the sequences are then split between training (51%), validation (24%) and testing (25%), ensuring no image and no object appears in more than one split.
- **MfM Small.** We extract from MfM Large subgraphs of five images with matched detections. This results in $70k$ sequences, evenly split into training, validation and testing.

Our dataset split addresses the uneven geographical distribution of data in [59], which results in a wide range of number of scenes per city, each with a different amount of images and objects (Table 1). To provide a more balanced dataset, we distribute the sequences to approach a 50% training, 25% testing and validation split, with a similar variety in the number of images and of objects. This does not prevent generalization, as shown in Section 4.4.

MfM Synthetic Dataset. For the Synthetic dataset, we generate maps by creating semantic maps with N_O objects uniformly distributed in a box defined within the range $[-1, 1]$. Each object is randomly assigned a label from among N_C possible classes. Subsequently, we generate N_M local semantic maps by randomly choosing a point of origin, selecting a subset of objects with each object observed with a probability ϕ, and choosing a random, compatible viewing direction. This setup offers an ideal testing scenario for proving the concept of the proposed method. In this evaluation, we set $N_C = 5$, $N_M = 8$ and $N_O = 7$, and we provide an ablation of these values in the Supplementary Material.

4.2. Graph Neural Network Architectures

We consider different architectural solutions for our **Alignment module**, considering two categories of GNN models: *i)* Attention-based and *ii)* Non-Attention-based. The attention-based models are designed to assign varying weights to the edges between nodes in the graph. This is achieved through a trainable attention mechanism that assigns larger weights to certain edges while diminishing the significance of others. Attention allows the model to focus on specific relationships within the graph. In contrast, the non-attention-based approach uses a more straightforward method for information aggregation. In these models, every edge in the graph is treated based on a pre-defined adjacency matrix, with no dynamic weight training capability. For these experiments, we adopt four different architectures, applying four consecutive layers of the following:

- **GCN** [28] uses a graph convolutional layer to aggregate information from neighboring nodes of G, as $\psi'' = \Xi(\psi', \tilde{\mathcal{G}}) = \Theta \sum_{v \in \mathcal{N}(u) \cup \{u\}} (d_u d_v)^{-\frac{1}{2}} A_{uv} {\psi'}_v$. Here $A_{uv} \in A$, $\mathcal{N}(u)$ is the neighborhood of node u, d_u is its vertex degree, and θ are the GNN weights.
- **GAT** [5] incorporates an attention mechanism between the node embedding, as $\psi'' = \Xi(\psi', \tilde{\mathcal{G}}) = \alpha_{uu} \Theta \psi'_u + \sum_{v \in \mathcal{N}(u)} \alpha_{uv} \Theta \psi'_v$, where α_{uv} is the attention mechanism.
- **SuperGAT** [27] builds upon the previous method, GAT. The main difference is that SuperGAT employs two types of attention mechanisms, enabling self-supervised learning of which edges carry the most information between nodes.
- **TransformerGCN** [56] combines the multi-head attention mechanism of the Transformer[60] and a fusion mechanism, applied to a standard GCN. $\psi'' = \Xi(\psi', \tilde{\mathcal{G}}) = \Theta_1 \psi'_u + \sum_{v \in \mathcal{N}(u)} \alpha_{uv} \Theta_2 \psi'_v$, with α_{uv} computed via multi-head dot product attention.

4.3. MfM Evaluation

The first set of experiments evaluates the MfM solution accuracy on the real-world scenes from **MfM Dataset**. We consider four different scenarios of decreasing difficulty: *a)* the standard graph formulation as proposed in (Section 3.2) with noisy local maps as input, generated using monocular depth estimation; *b)* the standard graph formulation with no known matches between the detections in different views but with the ground truth position of each node in its local maps; *c)* the graph is modified to include the ground truth detection matches, connecting nodes from different subgraphs only if they represent the same object, with noisy local maps as input; *d)* the graph with ground-truth detection matches with the the ground truth local maps.

COLMAP Baseline. We compare the performance of MfM against those of COLMAP, using the default features and matcher. Since COLMAP estimates for each scene a set of 3D camera poses and a 3D point cloud, its output cannot be directly compared with the 2D GT data of the MfM dataset, and has to be first projected into 2D points on a horizontal map. After identifying the ground plane of the reconstruction via PCA, the 2D camera locations are obtained by directly projecting the camera centers on it; for the object locations, instead, we identify and cluster the 3D points corresponding to all detections of each object, and project the center of the cluster on the ground plane.

	Method	Fail (%)	$\mu_c \pm \sigma_c$ (m)	$\mu_o \pm \sigma_o$ (m)
Small Scenes	COLMAP (Baseline)	80	**2.50** $\pm$ 2.64	**1.16** $\pm$ 2.09
	MfM + GCN	34	3.70$\pm$ 2.12	3.82 $\pm$ 1.47
	MfM + GAT	**30**	3.75 $\pm$ 2.17	3.58 $\pm$ 1.66
	MfM + SuperGAT	36	3.58 $\pm$ 2.11	3.65 $\pm$ 1.57
	MfM + Transf.GCN	37	3.73 $\pm$ 2.24	3.48 $\pm$ 1.59
Large Scenes	COLMAP (Baseline)	45	**1.12** $\pm$ 1.52	**1.20** $\pm$ 0.79
	MfM + GCN	15	4.01 $\pm$ 1.74	2.56 $\pm$ 1.15
	MfM + GAT	**5**	3.67 $\pm$ 1.76	2.66 $\pm$ 1.28
	MfM + SuperGAT	10	3.82 $\pm$ 1.68	2.74 $\pm$ 1.23
	MfM + Transf.GCN	10	3.77 $\pm$ 1.78	2.72 $\pm$ 1.62

Table 2. *MfM Dataset Evaluation* (Section 4.3): Average camera (μ_c) and object (μ_o) error, their standard deviation (σ_c and σ_o), and the failure percentage (Fail) of MfM using noisy inputs (Depth Local Maps), compared against standard COLMAP.

	Method	Fail (%)	$\mu_c \pm \sigma_c$ (m)	$\mu_o \pm \sigma_o$ (m)
Small Scenes	COLMAP (Baseline)	80	**2.50** $\pm$ 2.64	**1.16** $\pm$ 2.09
	MfM + GCN	**32**	3.83$\pm$ 2.15	3.76 $\pm$ 1.61
	MfM + GAT	34	3.71 $\pm$ 2.13	3.58 $\pm$ 1.46
	MfM + SuperGAT	35	3.62 $\pm$ 2.21	3.67 $\pm$ 1.66
	MfM + Transf.GCN	37	3.63 $\pm$ 2.15	3.51 $\pm$ 1.53
Large Scenes	COLMAP (Baseline)	45	**1.12** $\pm$ 1.52	**1.20** $\pm$ 0.79
	MfM + GCN	10	3.98 $\pm$ 1.73	2.84 $\pm$ 1.59
	MfM + GAT	15	3.62 $\pm$ 1.64	2.81 $\pm$ 1.87
	MfM + SuperGAT	**5**	4.14 $\pm$ 1.83	2.70 $\pm$ 1.49
	MfM + Transf.GCN	10	3.61 $\pm$ 1.50	2.83 $\pm$ 1.39

Table 3. *MfM Dataset Evaluation* (Section 4.3): Average camera (μ_c) and object (μ_o) error, their standard deviation (σ_c and σ_o), and the failure percentage (Fail) of MfM using perfect inputs (GT Local Maps), compared against standard COLMAP.

	Method	Fail (%)	$\mu_c \pm \sigma_c$ (m)	$\mu_o \pm \sigma_o$ (m)
Small Scenes	COLMAP (Baseline)	80	**2.50** $\pm$ 2.64	**1.16** $\pm$ 2.09
	MfM + GCN	35	3.72 $\pm$ 2.26	4.01 $\pm$1.42
	MfM + GAT	30	3.60 $\pm$ 2.13	3.49 $\pm$1.53
	MfM + SuperGAT	**26**	3.59 $\pm$ 2.07	3.44 $\pm$1.63
	MfM + Transf.GCN	31	3.65 $\pm$ 2.10	3.87 $\pm$ 1.39
Large Scenes	COLMAP (Baseline)	45	**1.12** $\pm$ 1.52	**1.20** $\pm$ 0.79
	MfM + GCN	10	4.48 $\pm$ 1.43	2.91 $\pm$ 1.63
	MfM + GAT	10	3.94 $\pm$ 1.74	2.58 $\pm$ 1.20
	MfM + SuperGAT	**5**	3.70 $\pm$ 1.98	2.67 $\pm$ 1.42
	MfM + Transf.GCN	**5**	3.97 $\pm$ 1.92	2.73 $\pm$ 1.39

Table 4. *MfM Dataset Evaluation with Known Detection Matches* (Section 4.3): Average camera (μ_c) and object (μ_o) error, their standard deviation (σ_c and σ_o), and the failure percentage (Fail) of MfM using noisy inputs (Depth Local Maps).

MfM Dataset Evaluation. Table 2 and Table 3 report the results of scenarios *a* and *b* respectively. The COLMAP baseline achieves more accurate localization, as expected. SfM pipelines uses more precise feature point matches together with outlier-free correspondences. This information is not available in the MfM pipeline, which localizes objects on a map rather than computing an accurate 3D point cloud. However, when compared to MfM, COLMAP exhibits a significantly higher failure rate, reaching 45% on MfM Large and a notable 80% on MfM Small. In contrast, the average failure rates for MfM in both scenarios

	Method	Fail (%)	$\mu_c \pm \sigma_c$ (m)	$\mu_o \pm \sigma_o$ (m)
Small Scenes	COLMAP (Baseline)	80	**2.50** $\pm$ 2.64	**1.16** $\pm$ 2.09
	MfM + GCN	30	3.86 $\pm$ 2.27	3.77 $\pm$ 1.56
	MfM + GAT	27	3.57 $\pm$ 2.05	3.53 $\pm$1.53
	MfM + SuperGAT	**25**	3.64 $\pm$ 2.10	3.49 $\pm$1.54
	MfM + Transf.GCN	31	3.86 $\pm$ 2.16	3.59$\pm$1.42
Large Scenes	COLMAP (Baseline)	45	**1.12** $\pm$ 1.52	**1.20** $\pm$ 0.79
	MfM + GCN	10	3.95 $\pm$ 1.78	2.67 $\pm$ 1.42
	MfM + GAT	**5**	3.91 $\pm$ 1.81	2.77 $\pm$ 1.33
	MfM + SuperGAT	**5**	3.84 $\pm$ 1.73	2.73 $\pm$ 1.45
	MfM + Transf.GCN	**5**	3.66 $\pm$ 1.68	2.64 $\pm$ 1.15

Table 5. *MfM Dataset Evaluation with Known Detection Matches* (Section 4.3): Average camera (μ_c) and object (μ_o) error, their standard deviation (σ_c and σ_o), and the failure percentage (Fail) of MfM using perfect inputs (GT Local Maps).

are around 34% and 10%, respectively. This disparity arises due to the sparse nature of input images (see [59]), leading to insufficient feature matches for initializing the Bundle Adjustment (BA) process that often fails. It is noteworthy that among the MfM models, there is a minimal disparity between the two configurations—when provided with the correct position (scenario *b*) v/s a noisy position (scenario *a*) in the local maps The results highlight the models' resilience to noise in the initial embedding, showcasing comparable performance across scenarios. As a final remark, the proposed approach is significantly more efficient than COLMAP. On both large and small scenes, MfM on average needs 2.8 ms to estimate the global map, with SAGE being the fastest at 1.8 ms and SuperGAT being the slowest at 3.5 ms. For a fair comparison, this only includes the inference time of the models, not the pre-processing time needed *e.g.* for object detection and local map projection. For reference, COLMAP's bundle adjustment stage requires on average 89.8 s for MfM large and 0.81 s for MfM short.

MfM Dataset Evaluation with Known Detection Matches. Table 4 and Table 5 report the results of scenarios *c* and *d* respectively, where the graph is modified to include the ground truth detection matches with the correct position and with a noisy position. Looking at the performance, we can see that while the average results are comparable to Table 2 and Table 3, the additional information has a positive impact on the failure rate (-5%), with an average failure rate of 28% for MfM small and 6.5% for MfM large. Comparing the two scenarios *c* and *d*, MfM models demonstrate resilience to the noise introduced in the initial embedding. The largest performance gap is observed for the GCN architecture, and is attributed to the model's challenges in mitigating the propagation of incorrect information within the graph.

4.4. Ablation

We conduct two experiments on the Synthetic MfM dataset: *i)* varying noise level on the objects' locations in

Δ_{xy}	0.0m		2.5m		5m		7.5m	
Method	F	μ	F	μ	F	μ	F	μ
MfM + GCN	0	1.0	0	0.7	1	1.0	1	1.1
MfM + GAT	0	0.3	0	0.3	0	0.6	0	0.9
MfM + SuperGAT	0	0.3	0	0.3	0	0.7	0	0.9
MfM + Transf.GCN	0	0.2	0	0.2	0	0.3	0	0.4

Table 6. *Effect of Local Map Noise on MfM* (Section 4.4): Failure percentage F and per-object error μ (m) as a function of noise in the local maps' accuracy (Δ_{xy}).

ϕ	1.0		0.75		0.5		0.25	
Method	F	μ	F	μ	F	μ	F	μ
MfM + GCN	0	1.0	75	5.8	97	6.4	99	6.7
MfM + GAT	0	0.3	75	5.7	97	6.4	99	6.2
MfM + SuperGAT	0	0.3	76	5.8	97	6.0	99	6.3
MfM + Transf.GCN	0	0.2	75	5.7	97	6.2	99	4.5

Table 7. *Effect of Visibility on MfM* (Section 4.4): Failure percentage F and per-object error μ (m) as a function of visibility ϕ.

the local maps, and *ii)* changing the visibility, *i.e.* the fraction of the scene's objects observed by each local map.

We apply displacements in a random direction and an amplitude in the range $[0, \Delta_{xy}]$, with $\Delta_{xy} \in \{0, 2.5, 5, 7.5\}$ meters. The results, reported in Table 6, show the robust performance of all methods, featuring low failure rates and precise localization despite the injection of noise in the initial map coordinates. Nevertheless, the discernible impact of noise becomes evident as mean errors increase at higher noise levels. The GCN model exhibits a constant slight rise in failure rates, possibly attributable to its relatively lower complexity compared to alternative solutions.

We then test the robustness of MfM to occlusions and other visibility-reducing effects. Table 7 reports performance for different levels of visibility $\phi \in \{1.0, 0.75, 0.50, 0.25\}$, where ϕ is the probability of each scene's object being observed by a given camera. With perfect detections ($\phi = 1.0$), the mean object localization achieves sub-meter accuracy (maximum error is 0.96 for GCN), with zero failures. However, as visibility decreases, we observe a substantial reduction in localization accuracy. This makes good visibility fundamental to solve MfM, but as long as enough objects are partially observed we can achieve accurate localization, irregardless of camera view. In contrast, SfM requires matching visual features, *i.e.* overlapping fields of view, which is a stricter requirement.

We then test the generalization capabilities of MfM, by performing leave-one-out cross-validation over the imbalanced data of the 5 cities. Table 8 shows the performance of two models - MfM + TransformerGCN and MfM + CGN - tested on sequences from one city and trained on the remaining ones; on average, the weighted mean of these results differs from the values reported in Tab. 3 by only 5.8%, confirming that the model can generalize to unseen cities.

Test Set	MfM+GCN Fail	MfM+GCN $\mu_c \pm \sigma_c$	MfM+GCN $\mu_o \pm \sigma_o$	MfM+TransformerGCN Fail	MfM+TransformerGCN $\mu_c \pm \sigma_c$	MfM+TransformerGCN $\mu_o \pm \sigma_o$
Barcelona	**0**	3.9 ± 1.1	**1.7** ± 0.5	**0**	4.2 ± 1.4	**1.9** ± 0.6
Berlin	**0**	5.1 ± 1.5	3.6 ± 1.4	**0**	5.3 ± 1.5	3.6 ± 1.2
Lisbon	10	**3.5** ± 2.0	$\underline{2.7} \pm 1.2$	10	**3.3** ± 1.8	$\underline{2.7} \pm 1.2$
Vienna	10	**3.5** ± 1.8	3.0 ± 1.7	15	$\underline{3.5} \pm 2.0$	3.4 ± 1.7
Paris	15	**3.5** ± 1.8	3.0 ± 1.7	10	$\underline{3.5} \pm 2.0$	3.4 ± 1.7
Mean	$\underline{9.8}$	$\underline{3.6} \pm 1.7$	$\underline{2.7} \pm 1.4$	$\underline{8.3}$	3.6 ± 1.8	2.9 ± 1.3
Table 3	10	4.0 ± 1.7	2.8 ± 1.6	10	3.6 ± 1.5	2.8 ± 1.4

Table 8. *Generalization of MfM*. Performance testing on a city and training on the others, reporting the average camera ($\mu_c \pm \sigma_c$) and object ($\mu_o \pm \sigma_o$) error and the fraction of failed scenes (Fail).

5. Conclusion

In the paper, we propose a solution to the novel task of MfM, *i.e.* generating 2D object maps by fusing into a global reference frame a set of local 2D semantic maps, representing the partial 2D map as observed from their view point. This initial method provides a first step towards developing tools to automatically generate annotations on 2D maps from uncalibrated images, without having to generate a 3D reconstruction of the scene or establish matches between the images. We shown how the proposed approach, MfM, can provide accurate maps even in the presence of limited information, such as noisy input maps and no availability of cross-view match between the detected objects. Even in these challenging scenarios, MfM provides an average localization within GPS accuracy. Moreover, when applied to spares sequences with large viewpoint changes, MfM has approximately a failure rate $3x$ lower than COLMAP.

The main limitation of the approach, as demonstrated in the ablation experiments, lies in its dependency on objects' covisibility among the multi-view images. Based on empirical estimations, every image must share at least three detections with another view. Images containing only classes not observed in the other images will result in disconnected subgraphs in $\tilde{\mathcal{G}}$, making the alignment impossible.

Future work will expand the approach to predict accurate detection matches across maps, to improve the information aggregation capabilities of graph-based learning models. This will allow extending the approach to tasks like multi-view object localization. Additionally, depth estimation is not the most accurate way for generating top-view maps, and was used to provide a lower-accuracy limit; future work will employ more sophisticated approaches, like generating segmented BEV maps from each image.

Acknowledgments

This work is supported by PNRR MUR under Project Cod. PE0000013 "FAIR" and investment 1.5 (M4C2) "RAISE"; and by the EU's Horizon research and innovation programmes "DCitizens" and "Bauhaus of the Seas Sails" (grant agreement No 101079116 and No 101079995).

References

[1] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building rome in a day. In *ICCV 2009*, pages 72–79, 2009. 2

[2] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5664–5673, 2019. 3

[3] Ayush Bandil, Vaishali Girdhar, Hieu Chau, Mohamed Ali, Abdeltawab Hendawi, Harsh Govind, Peiwei Cao, and Ashley Song. Geodart: A system for discovering maps discrepancies. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 2535–2546. IEEE, 2021. 3

[4] A. Benbihi, C. Pradalier, and O. Chum. Object-guided day-night visual localization in urban scenes. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 3786–3793, Los Alamitos, CA, USA, 2022. IEEE Computer Society. 3

[5] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2021. 6

[6] Lucas Brynte, José Pedro Iglesias, Carl Olsson, and Fredrik Kahl. Learning structure-from-motion with graph attention networks. *ArXiv*, abs/2308.15984, 2023. 3

[7] Yanming Chen, Guoli Li, Xiaoqiang Liu, Yueqian Shen, Jia Li, and Qin Tian. Integrating openstreetmap tags for efficient lidar point cloud classification using graph neural networks. *International Journal of Digital Earth*, 17(1):2297946, 2024. 3

[8] Wentao Cheng, Sheng Yang, Maomin Zhou, Ziyuan Liu, Yiming Chen, and Mingyang Li. Road mapping and localization using sparse semantic visual features. *IEEE Robotics and Automation Letters*, 6(4):8118–8125, 2021. 3

[9] OpenStreetMap Contributors. Planet dump retrieved from https://planet.osm.org, 2017. 2, 3

[10] Marco Crocco, Cosimo Rubino, and Alessio Del Bue. Structure from motion with objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4141–4149, 2016. 2

[11] Helisa Dhamo, Fabian Manhardt, Nassir Navab, and Federico Tombari. Graph-to-3d: End-to-end generation and manipulation of 3d scenes using scene graphs. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 3

[12] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. *IEEE Robotics and Automation Magazine*, 13(2):99–110, 2006. 2

[13] Cathrin Elich, Iro Armeni, Martin R. Oswald, Marc Pollefeys, and Joerg Stueckler. Learning-based relational object matching across views. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7, 2023. 3

[14] Stefano Fiorini, Stefano Coniglio, Michele Ciavotta, and Enza Messina. Sigmanet: One laplacian to rule them all. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023. 3

[15] Stefano Fiorini, Stefano Coniglio, Michele Ciavotta, and Enza Messina. Graph learning in 4d: A quaternion-valued laplacian to enhance spectral gcns. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12006–12015, 2024. 3

[16] Duncan Frost, Victor Prisacariu, and David Murray. Recovering stable scale in monocular slam using object-supplemented bundle adjustment. *IEEE Transactions on Robotics*, 34(3):736–747, 2018. 2

[17] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se(3)-transformers: 3d roto-translation equivariant attention networks. *Advances in neural information processing systems*, 33:1970–1981, 2020. 2

[18] Sarthak Garg, Helisa Dhamo, Azade Farshad, Sabrina Musatian, Nassir Navab, and Federico Tombari. Unconditional scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16362–16371, 2021. 3

[19] Abel Gawel, Carlo Del Don, Roland Siegwart, Juan Nieto, and Cesar Cadena. X-view: Graph-based semantic multi-view localization. *IEEE Robotics and Automation Letters*, 3 (3):1687–1694, 2018. 3

[20] Paul Gay, Cosimo Rubino, Vaibhav Bansal, and Alessio Del Bue. Probabilistic structure from motion with objects (psfmo). In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[21] Paul Gay, James Stuart, and Alessio Del Bue. Visual graphs from motion (vgfm): Scene understanding with object geometry reasoning. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*, pages 330–346. Springer, 2019. 3

[22] 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*, pages 19518–19527, 2022. 3

[23] Francesco Giuliari, Gianluca Scarpellini, Stefano Fiorini, Stuart James, Pietro Morerio, Yiming Wang, and Alessio Del Bue. Positional diffusion: Graph-based diffusion models for set ordering. *Pattern Recognition Letters*, 2024. 2

[24] Jiuxiang Gu, Handong Zhao, Zhe Lin, Sheng Li, Jianfei Cai, and Mingyang Ling. Scene graph generation with external knowledge and image reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1969–1978, 2019. 3

[25] Songtao He, Favyen Bastani, Satvat Jagwani, Edward Park, Sofiane Abbar, Mohammad Alizadeh, Hari Balakrishnan, Sanjay Chawla, Samuel Madden, and Mohammad Amin Sadeghi. Roadtagger: Robust road attribute inference with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10965–10972, 2020. 3

[26] Zongcai Huang, Peiyuan Qiu, Li Yu, and Feng Lu. Msengrp: A geographic relations prediction model based on multi-layer similarity enhanced networks for geographic relations completion. *ISPRS International Journal of Geo-Information*, 11(9):493, 2022. 3

[27] Dongkwan Kim and Alice Oh. How to find your friendly neighborhood: Graph attention design with self-supervision. In *International Conference on Learning Representations*, 2020. 6

[28] Thomas. N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2017. 3, 6

[29] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollar. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[30] Xin Kong, Xuemeng Yang, Guangyao Zhai, Xiangrui Zhao, Xianfang Zeng, Mengmeng Wang, Yong Liu, Wanlong Li, and Feng Wen. Semantic graph based place recognition for point clouds. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8216–8223. IEEE, 2020. 3

[31] Guoli Li, Xiaoqiang Liu, Xinyu Cai, Yao Chen, and Yanming Chen. Unleashing the power of openstreetmap tags: a graph neural network approach for efficient lidar point cloud classification. In *International Conference on Remote Sensing, Mapping, and Geographic Systems (RSMG 2023)*, pages 734–739. SPIE, 2023. 3

[32] Kejie Li, Daniel DeTone, Yu Fan (Steven) Chen, Minh Vo, Ian Reid, Hamid Rezatofighi, Chris Sweeney, Julian Straub, and Richard Newcombe. Odam: Object detection, association, and mapping using posed rgb video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5998–6008, 2021. 2

[33] Jielun Liu, Ghim Ping Ong, and Xiqun Chen. Graphsage-based traffic speed forecasting for segment network with sparse data. *IEEE Transactions on Intelligent Transportation Systems*, 23(3):1755–1766, 2020. 3

[34] Zhiguo Ma, Yutong Zhang, and Meng Han. Predicting maps using in-vehicle cameras for data-driven intelligent transport. *Electronics*, 12(24), 2023. 3

[35] Daniel D Morris, Kenichi Kanatani, and Takeo Kanade. Uncertainty modeling for optimal structure from motion. In *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*, pages 200–217. Springer, 2000. 5

[36] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. OpenMVG: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer, 2016. 2

[37] A. Calway N. Samano, M Zhou. You are here: Geolocation by embedding maps and images. In *In Proc. of the European Conference on Computer Vision (ECCV)*, 2020. 3

[38] Ahmed Samy Nassar, Sébastien Lefèvre, and Jan Dirk Wegner. Simultaneous multi-view instance detection with learned geometric soft-constraints. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6559–6568, 2019. 3

[39] Ahmed Samy Nassar, Stefano D'aronco, Sébastien Lefèvre, and Jan D Wegner. Geograph: graph-based multi-view object detection with geometric cues end-to-end. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 488–504. Springer, 2020. 3

[40] Lachlan Nicholson, Michael Milford, and Niko Sunderhauf. Quadricslam: Dual quadrics as slam landmarks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018. 2

[41] Lachlan Nicholson, Michael Milford, and Niko Sünderhauf. Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam. *IEEE Robotics and Automation Letters*, 4(1):1–8, 2019. 2

[42] Parv Parkhiya, Rishabh Khawad, J. Krishna Murthy, Brojeshwar Bhowmick, and K. Madhava Krishna. Constructing category-specific models for monocular object-slam. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4517–4524, 2018. 2

[43] Pulak Purkait, Tat-Jun Chin, and Ian Reid. Neurora: Neural robust rotation averaging. In *European Conference on Computer Vision*, pages 137–154. Springer, 2020. 3

[44] Kun Qin, Yuanquan Xu, Chaogui Kang, and Mei-Po Kwan. A graph convolutional network model for evaluating potential congestion spots based on local urban built environments. *Transactions in GIS*, 24(5):1382–1401, 2020. 3

[45] Tong Qin, Tongqing Chen, Yilun Chen, and Qing Su. Avp-slam: Semantic visual mapping and localization for autonomous vehicles in the parking lot. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5939–5945, 2020. 3

[46] Tong Qin, Yuxin Zheng, Tongqing Chen, Yilun Chen, and Qing Su. A light-weight semantic map for visual localization towards autonomous driving. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11248–11254, 2021. 3

[47] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020. 3, 6

[48] Zachary Ravichandran, Lisa Peng, Nathan Hughes, J. Daniel Griffith, and Luca Carlone. Hierarchical representations and explicit memory: Learning effective navigation policies on 3d scene graphs using graph neural networks. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 9272–9279, 2022. 3

[49] Cosimo Rubino, Marco Crocco, and Alessio Del Bue. 3d object localisation from multi-view image detections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1281–1294, 2018. 2

[50] Avishkar Saha, Oscar Mendez, Chris Russell, and Richard Bowden. "the pedestrian next to the lamppost" adaptive object graphs for better instantaneous mapping¡br/¿. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19528–19537, 2022. 3

[51] Paul-Edouard Sarlin, Daniel DeTone, Tsun-Yi Yang, Armen Avetisyan, Julian Straub, Tomasz Malisiewicz, Samuel Rota Bulo, Richard Newcombe, Peter Kontschieder, and Vasileios Baltnas. OrienterNet: Visual Localization in 2D Public Maps with Neural Matching. In *CVPR*, 2023. 2, 3

[52] Paul-Edouard Sarlin, Eduard Trulls, Marc Pollefeys, Jan Hosang, and Simon Lynen. SNAP: Self-Supervised Neural Maps for Visual Positioning and Semantic Understanding. In *NeurIPS*, 2023. 2, 3

[53] Gianluca Scarpellini, Stefano Fiorini, Francesco Giuliari, Pietro Morerio, and Alessio Del Bue. Diffassemble: A unified graph-diffusion model for 2d and 3d reassembly, 2024. 2

[54] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[55] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[56] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. In *IJCAI*, 2021. 6

[57] Matteo Taiana, Matteo Toso, Stuart James, and Alessio Del Bue. Posernet: Refining relative camera poses exploiting object detections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 3

[58] Carl Toft, Erik Stenborg, Lars Hammarstrand, Lucas Brynte, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. Semantic match consistency for long-term visual localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2

[59] Matteo Toso, Matteo Taiana, Stuart James, and Alessio Del Bue. You are here! finding position and orientation on a 2d map from a single image: The flatlandia localization problem and dataset. In *arXiv preprint arXiv:2304.06373*, 2023. 2, 3, 5, 6, 7

[60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 6

[61] Tomas Vojir, Ignas Budvytis, and Roberto Cipolla. Efficient large-scale semantic visual localization in 2d maps. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2020. 3

[62] Tuopu Wen, Zhongyang Xiao, Benny Wijaya, Kun Jiang, Mengmeng Yang, and Diange Yang. High precision vehicle localization based on tightly-coupled visual odometry and vector hd map. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 672–679, 2020. 3

[63] Fei Xue, Ignas Budvytis, Daniel Olmeda Reino, and Roberto Cipolla. Efficient large-scale localization by global instance recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17348–17357, 2022. 3

[64] Zi Jian Yew and Gim Hee Lee. Learning iterative robust transformation synchronization. In *International Conference on 3D Vision (3DV)*, 2021. 3

[65] Yifang Yin, Jagannadan Varadarajan, Guanfeng Wang, Xueou Wang, Dhruva Sahrawat, Roger Zimmermann, and See-Kiong Ng. A multi-task learning framework for road attribute updating via joint analysis of map data and gps traces. In *Proceedings of The Web Conference 2020*, pages 2662–2668, 2020. 3

[66] Mengjie Zhou, Xieyuanli Chen, Noe Samano, Cyrill Stachniss, and Andrew Calway. Efficient localisation using images and OpenStreetMaps. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5507–5513, 2021. 3

[67] Matthieu Zins, Gilles Simon, and Marie-Odile Berger. 3D-Aware Ellipse Prediction for Object-Based Camera Pose Estimation. In *3DV 2020 - International Virtual Conference on 3D Vision*, Fukuoka, Japan, 2020. 2