Lifted Semantic Graph Embedding for Omnidirectional Place Recognition

Chao Zhang[1] Ignas Budvytis[2] Stephan Liwicki[1] Roberto Cipolla[1,2]

[1]Toshiba Europe Ltd, Cambridge, United Kingdom
[2]University of Cambridge, United Kingdom

Abstract

Typical place recognition is dependent on the visual appearance and camera position of query images, without explicit use of domain knowledge and geometric relationships between key features in the scene. We exploit semantic grouping of pixels, and camera-pose robust scene graphs to perform structure-based visual localization for place recognition. In particular, we first formulate place recognition as an image retrieval task. Then, we lift the omnidirectional input images into 3D space, and compute a rotation and translation invariant semantic graph embedding to encode query and reference images. Finally, place information is obtained through graph similarity matching. Our graph representation is a simple addition to standard image embeddings with minimal overhead, but contains awareness of objects and their geometric relationships. In our experiments, we show improvement over typical place recognition, especially in environments with repetitions and dynamic appearance changes.

1. Introduction

Visual place recognition determines the camera's location given its current view. It is an important problem in computer vision and robotics [1, 38, 36, 33], and is relevant for a wide range of applications (*e.g.*, autonomous driving [13], augmented reality [27]). Efforts have been made to overcome challenges such as illumination and appearance changes [33] and viewpoint variations [35]. Traditional approaches formulate the task as image retrieval, and estimate the query location using the labels of the most visually similar images from a reference dataset. Typically, visual place recognition is tackled with perspective images, as many large scale datasets are available (*e.g.* Aachen Day-Night, RobotCar Seasons and InLoc [30, 26, 36]). Recent work explores place recognition with panoramic and omnidirectional images [19, 6]. Similarly, in our work we focus on equirectangular projections of omnidirectional images. Omnidirectional images provide a maximum field of view and enable speedup during database acquisition for a thorough map coverage. Furthermore, the calibration parameters of different cameras are normalized to the same representation on the sphere. Here we also note, the popularity of omnidirectional capturing devices such as GoPro Max, Insta360 and Ricoh Theta is rapidly increasing, making omnidirectional images in computer vision increasingly relevant, for example in object detection [11], semantic segmentation [42], depth estimation [47] and camera re-localization [45].

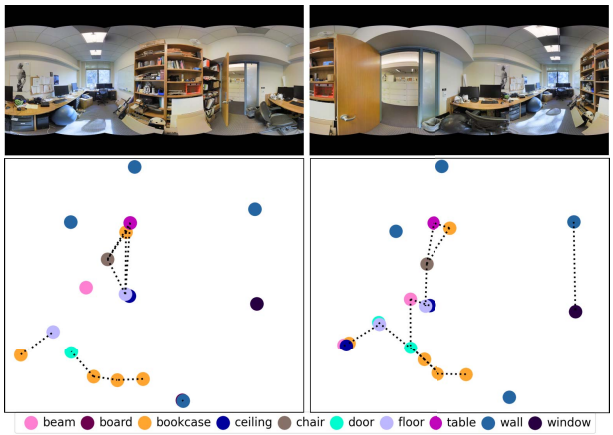


Figure 1. Two visually **dissimilar** views in the **same** room are challenging for omnidirectional visual localization due to distortions and object discontinuity. Our method groups object pixels and introduces a relationship graph by lifting images into 3D and using graph embeddings for comparison. (Colors for illustration purpose only)

Visual feature-based localization struggles with appearance changes caused by annual seasons or variable indoor surfaces such as monitors or bed linen. In [37], semantic information was exploited to score the feature correspondence. Semantic visual localization [31] combines semantics and 3D geometry in a generative descriptor learning for robust localization. In contrast, indoor appearance changes have not received much attention, though a recent dataset

2475-7888/21/$31.00 ©2021 IEEE
DOI 10.1109/3DV53792.2021.00147

was introduced in [40] to evaluate camera re-localization on different types of scene changes. In our work, we exploit instance level segmentation, and use segmentation for known classes in particular. We reason, since the reference dataset contains information about the domain, it is reasonable to utilize common objects in the scenes. Alternatively the objectness score may be employed directly. Furthermore, we alleviate the issue of appearance repetitions by utilizing geometric object relationships. In the following we focus on this concepts in particular.

Scene graph for semantic image retrieval [22] explicitly captures attributes of objects and relationships between objects. Replacing textual queries with scene graphs allows describing the semantics accurately. A scene graph is composed of nodes and edges. For a given image, nodes represent objects, and edges denote the relationship between them. To handle the scene graph representation, graph neural network approaches [23, 15] are used. An approach for image-to-image retrieval using scene graph similarity was proposed recently by [44].

In contrast to image scene graphs, we propose to lift the relations of objects in 2D images into 3D space. There are few works on 3D scene graph representation and generation [2, 39]. Inspired by these, we propose to represent an equirectangular image in 3D space. Our idea is to semantically aggregate expressive local features to form objects, and use geometric relationships to reduce object ambiguity, *e.g*. office desks or chairs all look similar. Our hypothesis is that the semantically aggregated features have enhanced discriminating information as location-based relationships between pixels are considered during aggregation through iterative graph-based message passing. Furthermore, we use simple spatial information of objects in 3D scene graphs to make our graphs translation and rotation invariant.

Our scene graph generated from equirectangular images is visualized in Fig. 1. We first predict instance segmentation and dense depth using omnidirectional input views. With per-pixel instance label and depth, we generate 3D scene graphs where each node is attached to CNN features. Edges are built by thresholding the distance between objects. Finally we apply graph similarity learning to optimize graph embeddings for the place recognition task. To validate our approach we use both synthetic and real datasets for evaluation: (i) Inspired by Clevr[21], OmniClevr is created to verify our hypothesis that relationship graphs are important for retrieval. (ii) Experiments on Stanford-2D3DS [3] demonstrate that our proposed method using predicted graphs outperforms state-of-the-art methods. In summary, our contributions are:

1. We employ object instance segmentation to group pixel-features using semantics;

2. We propose to lift the 2D scene graph into 3D space and use object distances to encode edges, to ensure translation and rotation invariance;

3. We exploit graph neural networks to formulate image similarity learning, and show superior results in synthetic and real datasets.

2. Related Work

Our work is related to omnidirectional image understanding such as object detection and depth estimation, and also 3D scene graph representations and their applications.

Object Detection Modern object detectors in 2D images are usually based on a two-stage approach. For instance, region-based CNN (R-CNN) [16] first makes a set of region proposals, then a convolutional network regresses the bounding boxes and class of an object. Mask R-CNN [18] added a branch for object mask prediction and is the most popular approach for object instance segmentation. Recently, with growing interest towards omnidirectional views, object detection in omnidirectional images emerged. Su *et al*. [34] facilitate omnidirectional object detection via a network distillation which extracts the tangent plane of spherical images, and applies rectangular kernels. Yang *et al*. [43] utilize perspective-projection based detector on a real-world dataset. However, the annotations of the objects are rectangular shape and would have been distorted on the sphere. Coors *et al*. [11] propose sphere convolution to solve the distortion in equirectangular images.

Depth Estimation The majority of recent CNN architectures for dense depth estimation follow the encoder-decoder structure. The encoder takes a RGB input and summarizes it to features at much smaller resolution, while the decoder regresses these features to the desired output by upsampling. For equirectangular images, Zioulis *et al*. [47] propose a set of rectangular filter banks to handle the projection distortions by increasing the receptive field of convolution kernels. Rather than working on panoramic or equirectangular images directly, the cube map representation simplifies the view to 6 cube faces. Cheng *et al*. [8] apply cube padding to reduce the information loss along edges between faces. Wang *et al*. [41] extend cube padding to spherical padding and proposed a two-branch encoder-decoder network for panorama depth estimation. Chen *et al*. [7] introduce a distortion-aware module and exploited strip pooling to preserve more context information.

3D Scene Graphs Armeni *et al*. [2] generate 3D scene graphs which holds object relationships, camera poses and room-building hierarchies. Here, a semi-automatic framework is proposed to alleviate manual labeling. The method employs existing detection methods on multiple perspective images sampled from an equirectangular image. However, this sample and detect procedure inevitably increases the

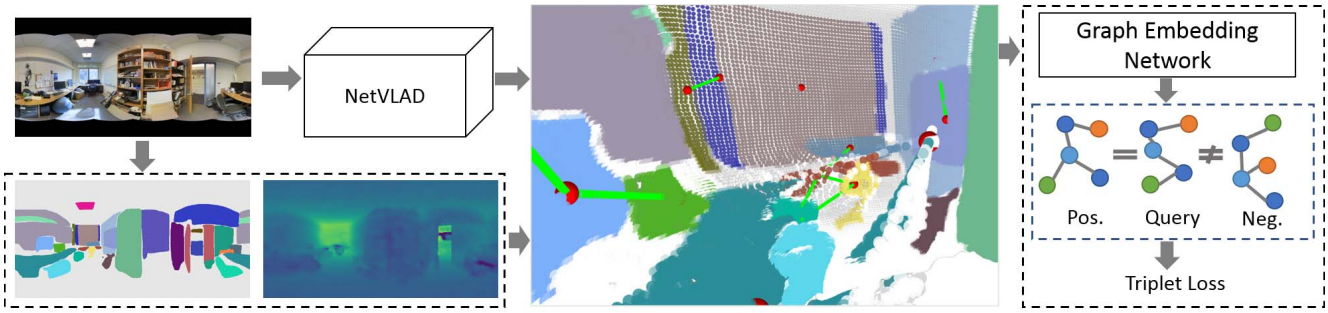


Figure 2. Our method is composed of three main parts: feature extraction, scene graph generation and graph similarity learning. We predict instance segmentation and dense depth given equirectangular input. These are used to lift images into 3D space. Here, red dots denote predicted object instance and green lines show the edges connecting nearby objects. We leverage a graph embedding network to learn the graph vector for the retrieval task.

computation and relies on post-processing for panoramic object detection. Towards the goal of 3D scene understanding, Wald *et al*. [39] introduce a semantically rich indoor scene graph dataset and propose a learning-based method to regress scene graphs. However, the method requires scene point cloud as input which limits its applicability.

Garanderie *et al*. [12] adapt contemporary automotive datasets with style and projection transformation to recover scene depth and 3D pose of vehicles. However, object semantics such as vehicles are not suitable for outdoor visual localization task. For indoor scene, Guerrero-Viu *et al*. [17] use object location and class information present in the scene to recover instance segmentation masks and place them inside the 3D room layout. Although semantic and spatial information are also exploited in our method, our goal is to enable graph similarity learning, rather than reconstruct 3D room layout.

3. Proposed Graphical VLAD (GraphVLAD)

The overall pipeline of our method for omnidirectional place recognition is shown in Fig. 2. Our method includes traditional feature encoding, our scene graph generation and the graph similarity learning. Our scene graphs are build from NetVLAD [1] features, instance proposal and depth estimation. This process transforms an equirectangular images to nodes with semantically aggregated features and 3D positions. This all enables graph-based similarity learning using both semantics and geometry. In the following, we demonstrate how to adapt existing methods to infer semantics and geometry for indoor panorama images. In particular, we will first introduce the feature extractor used by our work. Then, we describe how we generate scene graph representations from equirectangular images. Finally, we present our graph embedding network.

3.1. NetVLAD Image Feature Encoding

We follow the image retrieval approach to address equirectangular place recognition. Typical methods either concatenate activations of certain layers or use max pooling [5, 4]. State-of-the-art methods which finetune network end-to-end for place recognition include NetVLAD [1] for VLAD [20] and [28] for Fisher Vector [29]. In this work, we use NetVLAD as our feature base, but any feature extractor may be used. Later, we explain how to exploit geometric relationships for our features.

A trainable VLAD layer was proposed in NetVLAD [1] to mimic VLAD in a CNN framework. Please refer to [1] for detail. Briefly, given N D-dimensional local pixel descriptors as input, and K automatically found cluster centres, the output of NetVLAD at each pixel is a $K \times D$ matrix relating to the descriptor distance to each cluster center. Finally all pixel matrices are simply aggregated into a single matrix. It is then flattened and normalized, followed by PCA to reduce the dimensionality. The reduced vector is then used as the image embedding for the image retrieval task. In our method, we use NetVLAD's output with crucial changes: Instead of summing up all N pixel matrices in a orderless way, we maintain the spatial structure so we can later apply instance-level and graph-level aggregation. Thus, NetVLAD generates a $H' \times W' \times K \times D$ feature map for images of size $H \times W$, where $H' = H/16$ and $W' = W/16$.

3.2. Generating Lifted Scene Graphs

In this section, we describe our method to generate 3D scene graphs from equirectangular images. First we extract object instances, then we lift the images into 3D and utilize camera-motion invariant properties for our graphs.

Equirectangular Instance Segmentation The goal of our instance segmentation is to identify each object and group its pixel features into nodes of the graph. While general objectness may be applied, we found it is beneficial to build on known object categories in the scene. This is possible, as we have access to a reference dataset that contains relevant objects. To the best of our knowledge, there is no instance segmentation method that is trained on large-scale equirectangular datasets, although omnidirectional object detection dataset exist [9]. On the other hand, standard pla-

nar Mask R-CNN [18] have strong pretrained models on planar large-scale datasets like COCO [25]. Unfortunately, the simple application of Mask R-CNN[1] on equirectangular images produced unstable segmentation results. Thus we refined the model on image segmentation of Stanford-2D3DS, using training set alone, to classify 13 relevant classes. Finally, to overcome the problem of discontinuity, we generated two sets of detections: one on original equirectangular image, and one on the equirectangular projection after $180°$ rotation around the vertical axis. Final detections are filtered by non-maximum suppression (NMS). We found this works well to identify and segment objects reliably. We keep detected objects with at least 0.7 objectness confidence and the intersection over union (IoU) threshold of NMS is set to 0.5. Un-segmented pixels are aggregated as "unknown" node and is added to the scene graph to contain global image features. Finally we note, future improvement on instance level segmentation will directly translate into better performance for our method.

Equirectangular Depth Estimation Extracting camera invariant relationships between objects is possible by lifting images into 3D geometry. Given depth, we can generate a 3D scene from 2D equirectangular images. In [47] two architectures to estimate depth maps from equirectangular images were presented. We use the RectNet architecture, since it was the best performing method. Again, we increase accuracy by finetuning to the training split of Stanford-2D3DS. The input image size we used was 512×256. As before, any depth estimation module could be used, and future improvements will directly translate into improved performance.

Pose-Invariant Scene Graphs Now we have object masks from instance segmentation and the 3D location of each pixel from the depth. We generate the nodes of the graph by aggregating the pixel features for each object through summation. We also extract the median 3D coordinate location of each object, and span an edge between objects that have small euclidean distances (Fig. 2). Note, self-loop is not added to edges. Thus our graph is camera pose independent and contains information of objects and their spatial relationships in 3D world.

3.3. Graph Embedding via Similarity Learning

Each image is expressed through a scene graph as given in the section above. We now apply graph similarity learning to obtain our graph embedding network weights. In particular, a graph embedding module translates each graph into a vector, and a similarity metric is learnt to measure the similarity between graphs. We adapt the graph embedding network from [24] to learn the graph similarity for image retrieval.

[1]https://github.com/facebookresearch/detectron2

A scene graph $G = (V, E)$ is represented as a set of nodes V and edges E. Optionally, each node $i \in V$ is associated with a feature vector x_i and each edge $(i, j) \in E$ associated with a feature vector x_{ij}. In our work, nodes are objects, and the associated feature vector is the aggregation of NetVLAD descriptor at each pixel as described in Section 3.1, *i.e.* the sum of residues of K clusters with D dimensional features as a $K \times D$ matrix. Our edges are based on threshold distance values between object locations, and thus no descriptor is attached. Now the graph embedding module comprises three parts: the encoder, the propagation layers and the final graph aggregator. Before we start the description of our design, we emphasize that a simple sum of all nodes in our graph will result in an image descriptor that is similar to original NetVLAD as in essence all pixel features are summed.

The encoder maps the node and edge features to initial node and edge vectors h_i^0 and e_{ij}^0 respectively. Since we use NetVLAD [1] to extract per-pixel feature maps our encoding is already rich, and we simply assign $h_i^0 = x_i$ while e_{ij}^0 is unused.

Our propagation layer passes messages between nodes along edges. Here, node features h_i^t get new representation h_i^{t+1} using messages $m_{j \to i}$

$$m_{j \to i} = f_m(h_i^t, h_j^t) \tag{1}$$

$$h_i^{t+1} = f_n(h_i^t, \frac{1}{\|E_i\|} \sum_{(j,i) \in E_i} m_{j \to i}) \tag{2}$$

where $E_i \subset E$ contains all edges (j, i) connecting to i. Both f_m and f_n use sum of gating function

$$f = \sigma(\text{MLP}^i_{\text{gate}}(h_i, h_j)) \odot h_i^t + \sigma(\text{MLP}^j_{\text{gate}}(h_i, h_j)) \odot h_j^t \tag{3}$$

but with varying learnable weights. Here, $\text{MLP}^i_{\text{gate}}$ and $\text{MLP}^j_{\text{gate}}$ are gating MLPs also with different learnable weights. We use gating without additional feature MLP to avoid overfitting and emulate the summing of NetVLAD features. If no edges (j, i) exist, $h_i^t = h_i^{t+1}$.

Lastly, our aggregator takes the set of final node representations h_i^T and computes a graph level representation as follows:

$$h_G = \text{MLP}_G(\sum_{i \in V} \sigma(\text{MLP}_{\text{gate}}(h_i^T)) \odot \text{MLP}_{\text{node}}(h_i^T)) \tag{4}$$

which transforms node representations and use weighted sum of nodes with gating functions. Both MLP_{gate} and MLP_{node} are row kernels convoluted with features of each cluster. Following [1], MLP_G is a PCA based conversion of the feature matrices.

Using graph representations h_G for all images, we train our graph similarity module on a set of example triplets. In particular, given triplets where G_q is the query graph, G_p

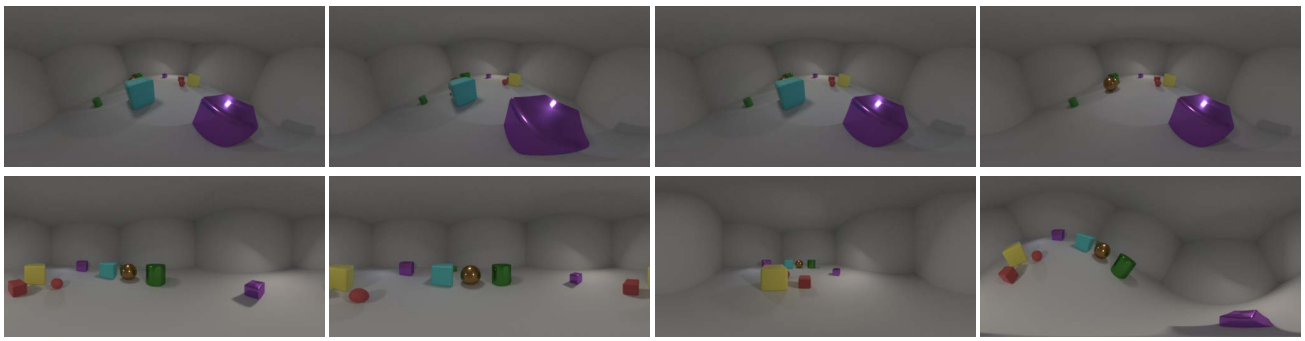


Figure 3. Example query images from OmniClevr are shown. Top row illustrates object changes: None, Move, Addition and Deletion; Bottom row shows camera motions: None, Translation, Zooming and Roll (both left to right).

the positive graph and G_n the negative graph, we optimize the margin-based triplet loss [32]:

$$L = \mathbb{E}_{G_q,G_p,G_n}[\max\{0, d(G_q,G_p) - d(G_q,G_n) + \gamma\}] \quad (5)$$

where $d(G_q,G_p) = \|h_{G_q} - h_{G_p}\|^2$ is the Euclidean distance, and $\gamma = 1.0$ is the margin used in our experiments. We note, weights for NetVLAD, instance segmentation and depth estimation are not trainable.

4. Datasets and Training Setup

In this section, we introduce our datasets and evaluation protocol for the task of omnidirectional image retrieval. Overall, two datasets with synthetic and real data are used.

4.1. Datasets

Inspired by Clevr [21], we create an equirectangular version of Clevr called OmniClevr for the image retrieval study. To demonstrate the strength of our method for equirectangular place recognition, we use Stanford-2D3DS [3] to evaluate our method and conduct ablation experiments.

OmniClevr Dataset In order to verify the idea that explicit knowledge of semantics and geometry is beneficial for omnidirectional image retrieval, we introduce our OmniClevr dataset (Fig. 3). This Blender-based synthetic environment includes a room of size $16m \times 16m$ and height $5m$, containing objects of different shapes and colors, and a randomly placed camera is used to observe the scene. We fix the camera height at $2m$, and its location is limited to the central $10m \times 10m$ area of the room. We create training data of 200 scenes with randomly selected 5-10 objects per scene, each with random color, shape and material as defined by the Clevr dataset [21].[2] For each scene, 10 images with random camera position and orientation are rendered and used for similarity learning. The validation consist of 50 different scenes. Similarly, 10 images per scene are used as reference images. As for query images, we design multiple subsets to systematically investigate the effect of various factors on retrieval. The first split is called "object", which includes 4 object-centric variations: none, move, addition and deletion. Moves may apply to all objects, but are limited to a $1m \times 1m$ area. Addition and deletion is limited to one object only. The second split is called "camera". This is to study the robustness of image retrieval against camera motion. To render query images, we randomly choose one reference image and start with the selected camera position and rotation. Then we render images with 3 optional camera movements: (i) horizontal translation, (ii) zoom in/out and (iii) roll rotation. [3] For each case we render 5 query images. Finally, 35 query images per scene are generated for validating purpose. In total, this gives us 2,000 images for training and 2,250 images for validation. Specifically, 500 and 1,750 images are used as reference images and query images, respectively.

Stanford-2D3DS Stanford-2D3DS [3] is a real indoor dataset consisting of 6 building-scale areas. There are in total 1,413 equirectangular images captured with annotated room labels. The dataset features similar room layout and repetitive texture which makes it a challenging dataset for image retrieval. In our evaluation protocol, we first split all rooms into two geographically disjoint splits: training and validation. To minimise the bias towards any specific room type, *e.g.* offices and hallways, we ensure that the ratio of each room type falling into training and validation is the same. Training split is used to learn a similarity metric so that image embeddings captured in the same room are similar, while images from different rooms are faraway. The training split includes 884 images while validation set contains 529 images. During validation, images from the validation set are further partitioned into reference set and query set. 107 images among 529 are used as query images. Images from training set are also added to reference set during validation as distractors, resulting in a reference set of 1,306 images.

[2] https://github.com/facebookresearch/clevr-dataset-gen

[3] Camera roll rotation is uniformly sampling within the range of [-pi, pi]. Camera translation and zooming is implemented by uniformly sampling within the range of [7m, 7m] along X and Y axis, respectively.

Method	Dist.(m)	Recall@1	Recall@5	Recall@10
NetVLAD	-	73.2	84.6	90.7
GraphVLAD	0	90.6	95.9	97.4
	1	90.9	96.5	97.6
	3	**92.1**	**97.1**	98.0
	7	92.1	97.0	**98.1**
	Inf	91.6	96.5	97.8
NetVLAD_t	-	83.2	94.0	96.3
$\text{GraphVLAD}_\text{t}$	3	**93.2**	**97.2**	**98.2**

Table 1. Quantitative comparison on OmniClevr. Average of recalls of all validation types are reported. $\text{GraphVLAD}_\text{t}$ with $d = 3m$ performs best. All our experiments except $d = 0$ use single propagation layer.

4.2. Training Setup and Details

On both OmniClevr and Stanford-2D3DS, we use NetVLAD with pretrained VGG-16 backbone. The weights of pretrained NetVLAD are taken from [10]. To obtain NetVLAD features, images are first upsampled to 2048×1024 and fed to NetVLAD as input. We use the 128×64 output where each pixel has a descriptor of 64×512. Instance segmentation and depth are resized to 128×64 using nearest neighbor downsampling to match NetVLAD output spatial dimension. Our method is implemented using PyTorch 1.80 with Geometric extension [14]. We use an Adam optimizer with learning rate set to 10^{-4}. In all experiments the network is trained with batch size 16 up to 1k epochs. We report the recalls for top N retrieved database images on the query set.

5. Results

In this section we present the results of our method (denoted GraphVLAD): (i) on an ablation study using a synthetic dataset and (ii) on a real indoor dataset.

5.1. Results on OmniClevr

For our first image retrieval experiment, we evaluate our method on the OmniClevr dataset. Inspired by Clevr [21], this dataset features equirectangular representation of a synthetic environment with compositions of objects. Ground truth instance segmentation and depth information are used in our method assuming they are predicted perfectly. Alternatively, it is trivial to train instance segmentation and depth estimation as shown in Section 3.2.

Baseline comparisons Our GraphVLAD uses generated scene graphs with NetVLAD features at the nodes. The edges are based on varying distance thresholds d, and we note $d = 3m$ performed best. We apply one propagation layer. In Table 1, we report recalls of top N retrieved images on combined object split and camera split. GraphVLAD improves upon NetVLAD with Recall@1 of 92.1 versus 73.2. Here we also note, off-the-shelf NetVLAD was pretrained on real images, and it is not expected to perform well on OmniClevr. Therefore, we also trained NetVLAD on OmniClevr from scratch, denoted NetVLAD_t. Similarly, $\text{GraphVLAD}_\text{t}$ uses these retrained NetVLAD features. Now Recall@1 improves to 83.2 for NetVLAD, while our $\text{GraphVLAD}_\text{t}$ now scores 93.2. We note, overall performance is improved with graphs, and GraphVLAD directly benefits from improved NetVLAD features.

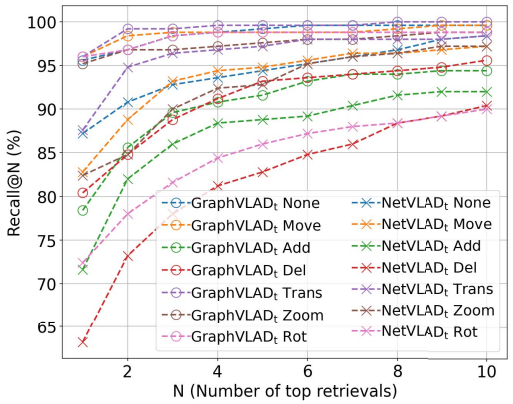
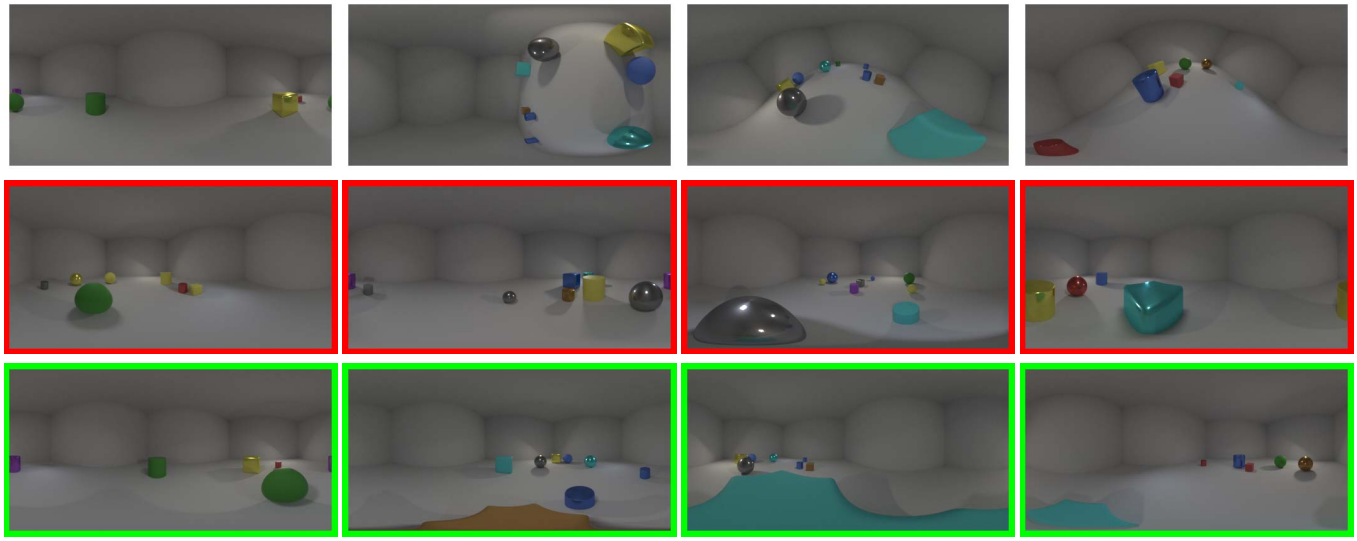


Figure 4. Recalls of $\text{GraphVLAD}_\text{t}$ and NetVLAD_t on OmniClevr for each validation type. Our $\text{GraphVLAD}_\text{t}$ is invariant to camera motion and more robust to object dynamics.

Effect of object dynamics In Fig. 4, we show the ROC curves of NetVLAD_t and $\text{GraphVLAD}_\text{t}$ on the different groups of the OmniClevr validation set. Static scenes (denoted None) are relatively easy for image retrieval while object dynamics such as object motion and changes pose more challenging tasks. Here, NetVLAD_t struggles especially with addition or deletion of objects, resulting in a 71.6 and 63.2 recall@1, respectively. In contrast, $\text{GraphVLAD}_\text{t}$ benefits from the graph embedding which is more robust than appearance alone, resulting in much better recall@1 for addition and deletion of 78.4 and 80.4 respectively. Object motion is less challenging, but our method again outperforms NetVLAD_t.

Effect of camera motion Another critical factor evaluated in Fig. 4 is view variation due to camera motion, especially across reference set and query set. We note, NetVLAD_t performs particularly poor for camera under roll rotation and zoom, with only 72.4 and 82.4 recall@1, respectively. In stark contrast, $\text{GraphVLAD}_\text{t}$ achieves 96.0 and 95.2 recall@1 respectively, benefiting from our rotation and translation invariant graph embedding. Similarly, translation is improved for $\text{GraphVLAD}_\text{t}$, while NetVLAD performs reasonably.

Effect of message passing Message passing in graph similarity learning plays an important role. Specifically, we control context awareness through larger receptive fields via the edge distance thresholds in our graphs. In Table 1 we



(a) (b) (c) (d)

Figure 5. Qualitative comparison of GraphVLAD$_t$ with NetVLAD$_t$ on OmniClevr. Proposed GraphVLAD$_t$ retrieves the correct database images when camera is **translated** (a), **rotated** (b) or **zoomed** (c), and an object is **modified** (d). From top to bottom, query image, NetVLAD retrieval and our retrieval are shown.

apply GraphVLAD (with pretrained weights) using varying distance thresholds d. Without message passing ($d = 0m$) recall@1 is reduced to 90.6, while $d = 1$ improves performance to 90.9. Overall, we achieve best performance with a larger set of edges, at the cost of efficiency. Our best results are achieved with $d = 3$ at 92.1 recall@1. Thus we believe accumulating neighbourhood features is beneficial for our method.

Qualitative results In Fig. 5, we show qualitative retrieval results. Comparing to NetVLAD$_t$, our GraphVLAD$_t$ shows its strength under challenging scene changes. In particular, our method enables correct matching under camera translation and rotation in Fig. 5(a) and Fig. 5(b). Furthermore, appearance distortion due to zooming is handled since 3D position is estimated and used in graph building Fig. 5(c). When a new object is added to the scene, our method is robust against dynamics because of message passing between nearby objects Fig. 5(d).

5.2. Results on Stanford-2D3DS

For our second experiment of place recognition, we show the results on the real indoor Stanford-2D3DS dataset described in Section 4.1. We evaluate our method using both ground truth scene graphs and our predicted ones.

Baseline comparisons We first discuss Table 2, which compares image retrieval performance of our method to the baselines. First let us discuss our baseline alternatives. We compare ResNet-50 trained on Places365 [46], NetVLAD trained on Pitts30k [1], and NetVLAD trained or refined on images of Stanford-2D3DS. For ResNet-50, images are resized to 224×224 and the feature map after average pooling is used as image descriptor. For NetVLAD, input images

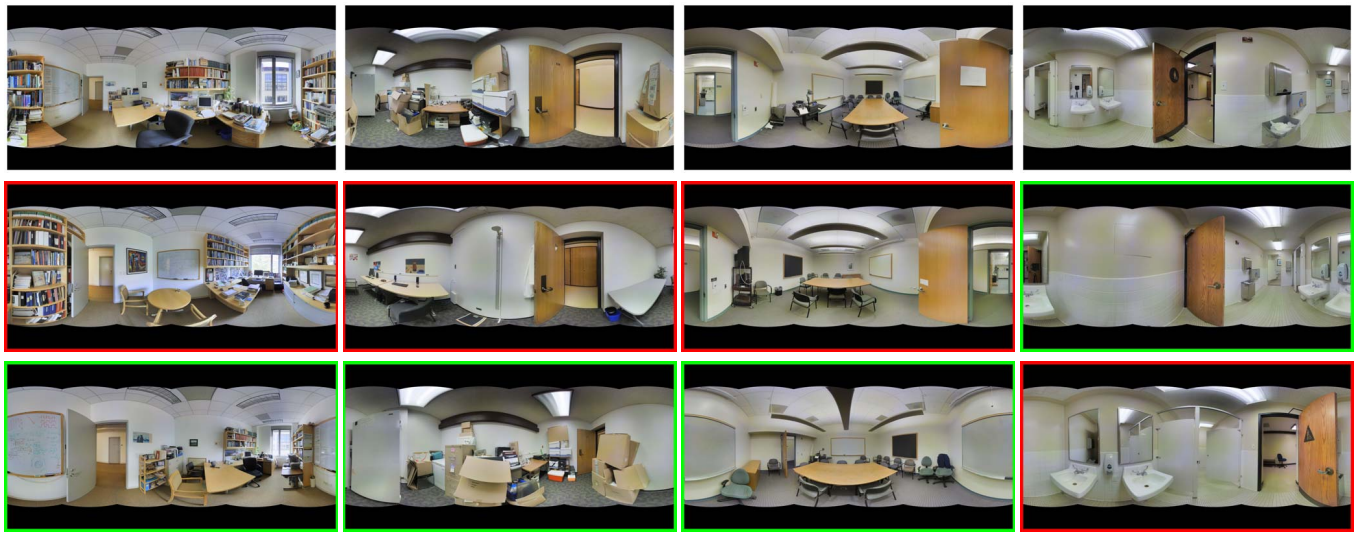
Method	R@1	R@5	R@10	Train setting
ResNet-50	35.5	57.0	65.4	Places365
NetVLAD	53.3	72.9	85.1	2D3DS
	58.8	82.2	85.9	Pitt30k
	53.2	72.0	82.2	Pitt30k+2D3DS
GraphVLAD$_{gt}$	68.3	87.5	91.4	d=0.5, 2 Prop
GraphVLAD$_{seg}$	67.3	87.5	91.3	
GraphVLAD$_{depth}$	63.5	83.2	92.5	
GraphVLAD	62.6	86.9	92.5	

Table 2. Comparison of GraphVLAD and baseline methods on Stanford-2D3DS. ResNet-50 was pretrained on Places365. NetVLAD results using different training data are reported. Our method using either GT graphs or predicted graphs performs better than other baselines.

are of size 2048×1024 and PCA with whitening is applied to result in an image embedding of size 4096. Pretrained NetVLAD benefits from vast amount of training, resulting in best performance for real images at 58.8 recall@1. In contrast ResNet-50 only reaches 35.5 recall@1, showing that triplet loss trained for discriminative features is preferred for image retrieval task. Nevertheless, finetuning NetVLAD did not yield improve results for the real dataset images. Using pretrained NetVLAD features, our proposed GraphVLAD outperforms at 62.6 recall@1 using the same input image without additional data.

Effect of message passing Like in OmniClevr, the receptive field of the graph message passing increases with larger distance threshold for edge generation. Table 3 investigates the impact of distance threshold d [4] under the assumption

[4] Distance threshold is related to the spatial scale of the environment and the distribution of objects, and therefore dataset dependent.



(a) (b) (c) (d)

Figure 6. Qualitative comparison on Stanford-2D3DS. NetVLAD retrieves visually similar but structure inconsistent reference images (a,b,c). While our method is robust to camera rotation and zooming of equirectangular images. Wrong matching is observed when semantics and textures are similar (d). From top to bottom, query image, NetVLAD retrieval and our retrieval are shown.

Method	R@1	R@5	R@10	Dist.(m)	#of Prop
GraphVLAD$_{gt}$	66.3	87.5	90.4	0.0	-
	67.3	87.5	90.4	0.1	1
	67.3	86.5	92.3	0.5	1
	64.4	87.5	91.4	1.0	1
	59.6	88.5	93.3	1.5	1
	68.3	87.5	91.4	0.5	2
	67.3	87.5	91.3	0.5	3

Table 3. Ablation study of our method using GT graphs of Stanford-2D3DS dataset. We study the effect of different distance threshold d and number of propagation layers.

of perfect instance segmentation and depth. Increasing d between 0m and 0.5m improves recall results, while further increase diminishes performance. Peak performance is reached as $d = 0.5m$ with 67.3 recall@1. We note, while recall@5 is reduced toward $d = 0.1m$, overall more global information is captured, increasing recall@10. With larger context receptive graphs where $d = 1.0$ and above, graph embedding disambiguate with object repetition is reduced, as less distinguished relationships are seen. Finally we evaluate multiple propagation iterations, and achieve similar results but at the cost of more computation.

Effect of predicted graphs Based on previous findings using ground truth graphs, we fix $d = 0.5$ and use 2 propagation iterations. Table 3 shows the ablations. We investigate the effect of predicted depth as described in Section 3.2, denoted GraphVLAD$_{seg}$. Since we employ object location augmentation during training, retrieval performance is competitive with GraphVLAD$_{gt}$. We then evaluate predicted semantic with true depth in GraphVLAD$_{depth}$. The performance drops from 68.3 to 63.5 recall@1. Thus we conclude that quality object segmentation improves pixel grouping, and therefore recall results. Finally, using both predicted depth and segmentation achieves 62.6 recall@1, which is still competitive and improves upon NetVLAD with the same input data. Again, we emphasize, improved segmentation performance will directly translate into better recall results. The ground truth is an indication of top performance to be reached.

Qualitative results In Fig. 6, we compare NetVLAD and GraphVLAD qualitatively. Without the knowledge of semantics, NetVLAD descriptors only capture global appearance similarity while ignoring subtle structure difference, as shown in Fig. 6(a). Using scene graphs as representation, our method further lifts a graph to 3D space and enable the embedding to be robust to camera motions such as zooming and roll rotation. An example of large translation is shown in Fig. 6(b). Camera is translated and rotated 90 degree in Fig. 6(c). In places such as toilets where both semantic and texture are similar, it is difficult to distinguish between rooms Fig. 6(d).

6. Conclusions

We presented a place recognition method based on graph similarity learning that operates on omnidirectional query images. Built on pixel-wise NetVLAD features, we predict instance segmentation to guide local features aggregation and to form nodes in a scene graph. Depth estimation is employed to further lift scene graphs to 3D, and we extract a rotation and translation invariant spatial relationship for graph similarity learning. In our evaluation we improve on original NetVLAD by introducing our GraphVLAD with equivalent input image data.

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, pages 5297–5307, 2016. 4321, 4323, 4324, 4327

[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 *ICCV*, pages 5664–5673, 2019. 4322

[3] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 4322, 4325

[4] Artem Babenko and Victor Lempitsky. Aggregating local deep features for image retrieval. In *ICCV*, pages 1269–1277, 2015. 4323

[5] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *ECCV*, pages 584–599. Springer, 2014. 4323

[6] Jean-Baptiste Boin, Dmytro Bobkov, Eckehard Steinbach, and Bernd Girod. Efficient panorama database indexing for indoor localization. In *2019 International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–7. IEEE, 2019. 4321

[7] Hong-Xiang Chen, Kunhong Li, Zhiheng Fu, Mengyi Liu, Zonghao Chen, and Yulan Guo. Distortion-aware monocular depth estimation for omnidirectional images. *IEEE Signal Processing Letters*, 28:334–338, 2021. 4322

[8] Hsien-Tzu Cheng, Chun-Hung Chao, Jin-Dong Dong, Hao-Kai Wen, Tyng-Luh Liu, and Min Sun. Cube padding for weakly-supervised saliency prediction in 360 videos. In *CVPR*, pages 1420–1429, 2018. 4322

[9] Shih-Han Chou, Cheng Sun, Wen-Yen Chang, Wan-Ting Hsu, Min Sun, and Jianlong Fu. 360-indoor: towards learning real-world objects in 360deg indoor equirectangular images. In *WACV*, pages 845–853, 2020. 4323

[10] Titus Cieslewski, Siddharth Choudhary, and Davide Scaramuzza. Data-efficient decentralized visual slam. In *ICRA*, pages 2466–2473. IEEE, 2018. 4326

[11] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In *ECCV*, pages 518–533, 2018. 4321, 4322

[12] Greire Payen de La Garanderie, Amir Atapour Abarghouei, and Toby P Breckon. Eliminating the blind spot: Adapting 3d object detection and monocular depth estimation to 360 panoramic imagery. In *ECCV*, pages 789–807, 2018. 4323

[13] Anh-Dzung Doan, Yasir Latif, Tat-Jun Chin, Yu Liu, Thanh-Toan Do, and Ian Reid. Scalable place recognition under appearance change for autonomous driving. In *ICCV*, pages 9319–9328, 2019. 4321

[14] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. 4326

[15] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *ICML*, pages 1263–1272. PMLR, 2017. 4322

[16] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014. 4322

[17] Julia Guerrero-Viu, Clara Fernandez-Labrador, Cédric Demonceaux, and Jose J Guerrero. What's in my room? object recognition on indoor panoramic images. In *ICRA*, pages 567–573. IEEE, 2020. 4323

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 4322, 4324

[19] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, Teddy Furon, and Ondřej Chum. Panorama to panorama matching for location recognition. In *International Conference on Multimedia Retrieval*, pages 392–396, 2017. 4321

[20] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, pages 3304–3311. IEEE, 2010. 4323

[21] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, pages 2901–2910, 2017. 4322, 4325, 4326

[22] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *CVPR*, pages 3668–3678, 2015. 4322

[23] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 4322

[24] Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. Graph matching networks for learning the similarity of graph structured objects. In *ICML*, pages 3835–3845. PMLR, 2019. 4324

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. 4324

[26] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017. 4321

[27] Sven Middelberg, Torsten Sattler, Ole Untzelmann, and Leif Kobbelt. Scalable 6-dof localization on mobile devices. In *ECCV*, pages 268–283. Springer, 2014. 4321

[28] Eng-Jon Ong, Sameed Husain, and Miroslaw Bober. Siamese network of deep fisher-vector descriptors for image retrieval. *arXiv preprint arXiv:1702.00338*, 2017. 4323

[29] Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-scale image retrieval with compressed fisher vectors. In *CVPR*, pages 3384–3391. IEEE, 2010. 4323

[30] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking

6dof outdoor visual localization in changing conditions. In *CVPR*, pages 8601–8610, 2018. 4321

[31] Johannes L Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic visual localization. In *CVPR*, pages 6896–6906, 2018. 4321

[32] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015. 4325

[33] Erik Stenborg, Carl Toft, and Lars Hammarstrand. Long-term visual localization using semantically segmented images. In *ICRA*, pages 6484–6490. IEEE, 2018. 4321

[34] Yu-Chuan Su and Kristen Grauman. Learning spherical convolution for fast features from 360 imagery. *Advances in Neural Information Processing Systems*, 30:529–539, 2017. 4322

[35] Niko Sünderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. *Robotics: Science and Systems XI*, pages 1–10, 2015. 4321

[36] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In *CVPR*, pages 7199–7209, 2018. 4321

[37] Carl Toft, Erik Stenborg, Lars Hammarstrand, Lucas Brynte, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. Semantic match consistency for long-term visual localization. In *ECCV*, pages 383–399, 2018. 4321

[38] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *CVPR*, pages 1808–1817, 2015. 4321

[39] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions. In *CVPR*, pages 3961–3970, 2020. 4322, 4323

[40] Johanna Wald, Torsten Sattler, Stuart Golodetz, Tommaso Cavallari, and Federico Tombari. Beyond controlled environments: 3d camera re-localization in changing indoor scenes. In *ECCV*, pages 467–487. Springer, 2020. 4322

[41] Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. Bifuse: Monocular 360 depth estimation via bi-projection fusion. In *CVPR*, pages 462–471, 2020. 4322

[42] Kailun Yang, Xinxin Hu, and Rainer Stiefelhagen. Is context-aware cnn ready for the surroundings? panoramic semantic segmentation in the wild. *IEEE Transactions on Image Processing*, 30:1866–1881, 2021. 4321

[43] Wenyan Yang, Yanlin Qian, Joni-Kristian Kämäräinen, Francesco Cricri, and Lixin Fan. Object detection in equirectangular panorama. In *ICPR*, pages 2190–2195. IEEE, 2018. 4322

[44] Sangwoong Yoon, Woo Young Kang, Sungwook Jeon, SeongEun Lee, Changjin Han, Jonghun Park, and Eun-Sol Kim. Image-to-image retrieval by learning similarity between scene graphs. *arXiv preprint arXiv:2012.14700*, 2020. 4322

[45] Chao Zhang, Ignas Budvytis, Stephan Liwicki, and Roberto Cipolla. Rotation equivariant orientation estimation for omnidirectional localization. In *ACCV*, 2020. 4321

[46] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 4327

[47] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, and Petros Daras. Omnidepth: Dense depth estimation for indoors spherical panoramas. In *ECCV*, pages 448–465, 2018. 4321, 4322, 4324