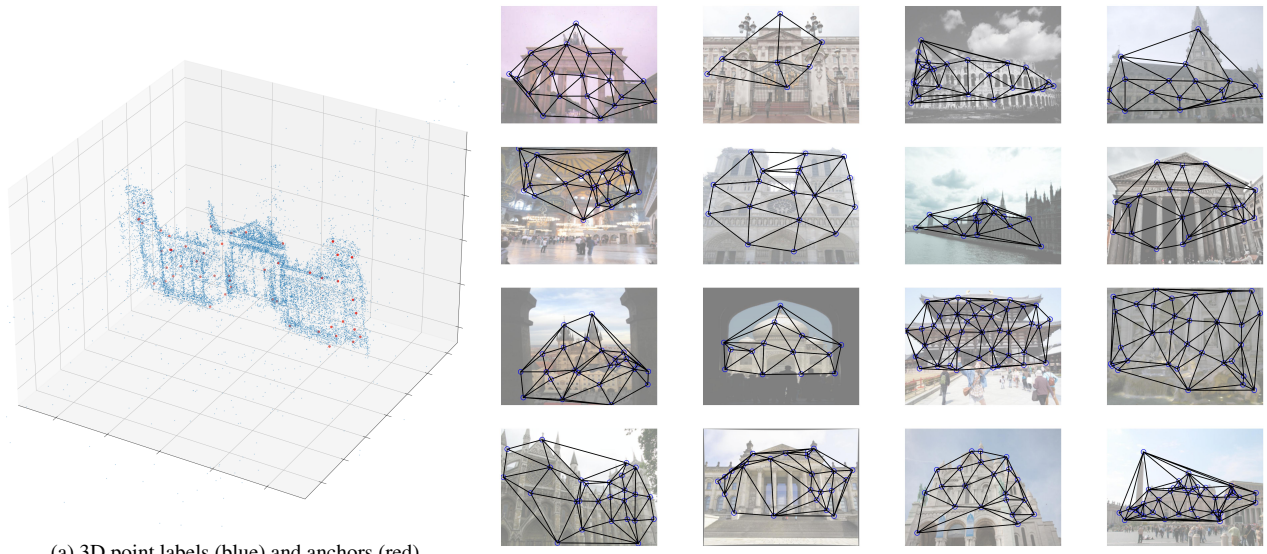

(a) 3D point labels (blue) and anchors (red)

(b) Examples (raw images with built graphs) for visual graph matching

Figure 6. The 3D points in (a) are detected by colmap [31, 32] which are available as labels in IMC-PT [18]. The blue points denote our selected anchors, based on which our IMC-PT SparseGM-50 is built, as shown in (b). The lines connecting anchors are the edges we build through Delaunay triangulation.

Table 7. Comparison of existing visual graph matching benchmarks.

dataset name	# instances	# classes	avg # nodes	avg # edges	# universe	partial rate	best-known f1
CMU house/hotel	212	2	30	\	30	0.00%	100% (RRWM [7])
Willow ObjectClass	404	5	10	\	10	0.00%	97.8% (GANN [46])
CUB2011	11,788	200	12	\	15	20.00%	83.2% (PCA-GM [44])
Pascal VOC Keypoint	8,702	20	9.07	\	6 to 23	28.50%	62.8% (BBGM-Multi [29])
IMC-PT-SparseGM-50 (ours)	25,765	16	21.36	54.71	50	57.28%	72.9% (ours)
IMC-PT-SparseGM-100 (ours)	25,765	16	44.48	123.99	100	55.52%	71.5% (ours)

A. Details of IMC-PT-SparseGM Benchmark

A.1. Visualization

See Fig. 6a for a visualization of the 3D point cloud built from the collection of Reichstag photos, where most points gather near the main part of the building. The red points are the anchors (50 anchors in this example) that are regarded as the keypoints in our visual graph matching benchmark. These anchor points are then projected back to the 2D images, whereby the visibility of anchors is judged by the method described in the main paper. Examples of images and keypoints from our benchmark are visualized in Fig. 6.

A.2. Details about hyperparameters

We elaborate on the following three insights of our proposed approach to transforming the original IMC-PT image matching dataset to our IMC-PT-SparseGM benchmark for visual graph matching. These insights are omitted in the main paper due to page limitations. Our approach only involves four hyperparameters:

1) To extract some keypoints that can well represent the feature of the original building and to reduce the impact of noise, we set one hyperparameter of the frequency threshold of keypoints existence, screening out keypoints frequently appearing in the sample images.

2) To reduce the complexity of graph matching, we randomly extract a hyperparameter of 50 keypoints from the keypoints we selected before. During the extraction, to maintain a good representation of the original building, we set a hyperparameter of the minimal euclidean distance of two keypoints to ensure that the extracted keypoints are relatively evenly distributed in the main part of the building.

3) For every sample image of a certain building, we intend to check whether the extracted keypoints exist in the image. Using the annotation of the whole keypoints existing in the image, for every selected keypoints, we calculate its minimal euclidean distance between the keypoints in the image and judge its existence in the image based on another minimal euclidean distance hyperparameter threshold which indicates whether the keypoint is close to the present image or not, removing some of the keypoints covered by the exterior scene to some extent.

Table 8. Number of visual graphs in each class of IMC-PT-SparseGM benchmark. * refers to test class.

class name	brandenburg_gate	grand_place_brussels	palace_of_westminster	reichstag*
# visual graphs	1,363	1,083	983	75
class name	taj_mahal	westminster_abbey	buckingham_palace	hagia_sophia_interior
# visual graphs	1,312	1,061	1,676	889
class name	pantheon_exterior	sacre_coeur*	temple_nara_japan	colosseum_exterior
# visual graphs	1,401	1,179	904	2,063
class name	notre_dame_front_facade	prague_old_town_square	st_peters_square*	trevi_fountain
# visual graphs	3,765	2,316	2,504	3,191

Table 9. F1 (%) on SPair-71k (unfiltered setting). Our methods are marked as gray.

GM Network	PMH	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	dog	horse	mbike	person	plant	sheep	train	tv	mean
ZACR	ZACR	32.9	33.3	45.7	24.6	62.0	13.5	36.0	56.2	17.4	47.5	32.7	19.0	40.7	42.7	37.3	34.8	52.5	60.0	38.3
PCA-GM	None	36.5	25.6	48.9	24.7	50.7	29.1	19.2	54.6	30.1	39.1	42.9	34.0	31.3	27.1	70.5	31.1	56.6	75.2	40.4
BBGM	None	42.9	43.8	65.3	34.6	62.6	47.6	25.6	68.0	38.6	62.0	57.8	42.8	44.1	36.0	83.2	45.4	86.7	90.3	54.3
Ngmv2	None	45.4	42.3	61.0	31.2	62.2	53.3	34.2	65.3	37.0	59.5	54.7	41.3	44.8	38.9	77.5	44.2	77.8	89.9	53.4
	Thresholding	50.2	42.9	63.4	29.9	62.1	53.9	34.8	65.7	37.3	62.7	56.1	43.8	45.7	41.8	77.1	45.2	79.0	90.4	54.6±0.5
	Dummy node	47.7	41.6	62.1	30.3	59.0	49.7	27.4	68.3	33.9	62.4	57.3	46.7	46.4	42.7	78.7	43.5	80.5	89.5	53.8±0.4
	AFAT-U(ours)	50.3	43.5	63.8	32.4	59.0	60.1	39.7	68.6	36.1	63.6	56.5	46.3	51.4	43.3	77.0	51.2	81.1	89.4	56.3±0.4
	AFAT-I(ours)	50.4	43.6	63.9	32.1	61.2	58.5	38.0	68.4	35.7	62.7	56.4	47.7	51.9	44.3	78.5	50.7	79.2	91.2	**56.4**±0.6
GCAN	Dummy node	49.0	41.3	64.0	30.3	57.3	55.0	37.4	64.8	36.6	63.0	58.0	44.4	46.4	42.6	68.4	42.3	83.2	91.9	54.2±0.3
	AFAT-U(ours)	46.7	43.3	65.8	33.3	61.5	54.9	35.2	68.4	37.7	59.9	56.0	47.6	47.2	43.5	80.3	47.7	83.8	89.0	**55.7**±0.4
	AFAT-I(ours)	46.8	44.3	65.9	32.4	61.5	53.8	33.7	68.4	38.1	60.1	56.3	47.9	48.3	43.8	81.2	48.4	82.9	88.0	**55.7**±0.4

A.3. More Details

Tbl. 7 shows comparison among our released IMC-PT-SparseGM benchmark and other existing vision graph matching benchmarks. Note that in our released IMC-PT-SparseGM benchmark, the edges are previously built through Delaunay triangulation, thus saving users' time of online graph-building. Tbl. 8 exhibits number of visual graphs (with raw images) in each class of IMC-PT-SparseGM benchmark.

In addition, the anchors are not fixed in IMC-PT-SparseGM, and can be edited via tuning hyperparameters, allowing users to build data that fulfills their own demands. We provide code and instructions for users to build their own data in IMC-PT-SparseGM dataset page: https://github.com/Thinklab-SJTU/IMCPT-SparseGM-dataset.

B. Results on SPair-71k Dataset

To further validate the general effectiveness of our proposed methods, we also perform experiments on SPair-71k (http://cvlab.postech.ac.kr/research/SPair-71k/) dataset. The dataset contains 18 categories of total 70,958 image pairs, including 53,340 for training and 12,234 for testing. The image pairs are different in scale, truncation, and occlusion, whereby outliers are prevalent. We still follow the "unfiltered" setting and show our experimental results in Tbl. 9. Consistent with the experimental results on other datasets, our methods outperform other PMH methods on both GM network embodiments.

Algorithm 2 GreedyTopK

Input: confidence matrix $\mathbf{D}_{conf}$; k; permutation matrix $\mathbf{P}$.
Output: final permutation matrix $\tilde{\mathbf{P}}$.

$\mathbf{D}_{conf} = \mathbf{D}_{conf} \odot \mathbf{P}$; ▷ filter the matching confidence
set $\tilde{\mathbf{P}}$ to all-zero matrix; set $m = 0$; ▷ initialization
while $m < k$ **do**
  $r, c = \text{argmax}\,(\mathbf{D}_{conf})$; ▷ the most confident match
  $\tilde{\mathbf{P}}_{r,c} = 1$; ▷ select this match
  $\mathbf{D}_{conf\,r,c} = 0$; ▷ to select next match
  $m = m + 1$; ▷ count selected matches
end while
return $\tilde{\mathbf{P}}$; ▷ final matching result, for testing

C. GreedyTopK Algorithm for Post-process

In our proposed framework, a GreedyTopK algorithm is adopted in inference stage to greedily select top-k matches based on the confidence from the output of Hungarian algorithm, i.e., the permutation matrix $\mathbf{P}$. Algorithm 2 shows the procedure of GreedyTopK algorithm, where $\odot$ denotes element-wise product.