PoserNet: Refining Relative Camera Poses Exploiting Object Detections * - Supplementary Material

Matteo Taiana, Matteo Toso, Stuart James, and Alessio Del Bue

Pattern Analysis and Computer Vision (PAVIS), Istituto Italiano di Tecnologia (IIT), Genoa, Italy
{name.surname}@iit.it

1 Introduction

In the main paper, we introduced a novel approach to relative pose refinement (PoserNet) and evaluated its effectiveness in correcting initial relative pose estimates. We then investigated the effect of PoserNet on Motion Averaging performed via either an optimisation-based method (EIG-SE3 [1]), or a deep-learning-based method (MultiReg [7]). In both cases, we used as training and testing dataset a collection of graphs generated using images from 7-Scenes[3].

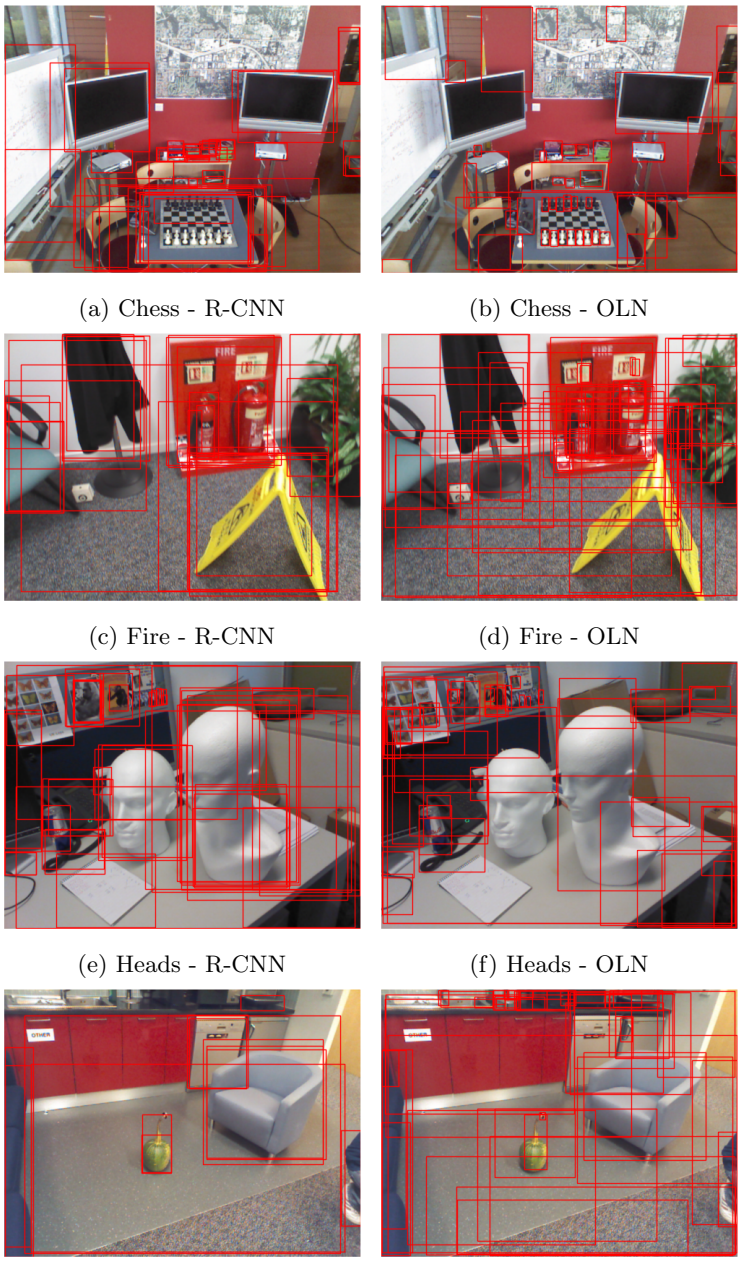
In this document, we provide additional details about *i)* how the 7-Scenes graphs were obtained by generating and matching ROI detections; *ii)* the implementation details for our PoserNet and for the third-party MultiReg [7] and EIG-SE3 [1] methods; *iii)* a more detailed discussion of the error distribution after motion averaging (Table 4 of the main paper); *iv)* the effect of increasing PoserNet's complexity by increasing its "depth"; *v)* PoserNet generalisation capabilities.

2 Generating Graphs from 7-Scenes

In this section, we provide additional details on how we generated object detections and on how we established matches between them. We provide an example of the resulting graphs in Figure 2. Moreover, we share general considerations on the difficulties we encountered in the process.

ROIs generation – We computed ROIs for our experiment applying the pre-trained Object Localisation Network (OLN) [4] on the images of the 7-Scenes dataset. Examples of ROIs computed via OLN, and a comparison with those obtained using a part of Faster R-CNN [5] are shown in Figure 1. We opted for using OLN in our work because we found it more reliable at localising small objects. We selected the 50 best-scoring detections for each image, a number empirically large enough to capture most objects present in the scene. We discarded detections larger than 25% of the image size, as they were likely associated with

* This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 870743.



(a) Chess - R-CNN (b) Chess - OLN

(c) Fire - R-CNN (d) Fire - OLN

(e) Heads - R-CNN (f) Heads - OLN

(g) Pumpkin - R-CNN (h) Pumpkin - OLN

Fig. 1: Different approaches to ROI generation: ROIs computed with the pre-trained ResNet50 component of the Faster R-CNN [5] network, and by OLN [4]. We found OLN more reliable at identifying small and re-occurring objects.

a large scene area, rather than with an object, and could possibly carry information detrimental to the relative pose estimation task.

Camera selection – We then sampled image sets, as candidates for the nodes of the graphs. In most of our experiments we used graphs composed of eight nodes; we compared the performances of PoserNet on large graphs (125 nodes) and small graphs (eight nodes), and in the paper we show how the latter lead to better performance. Moreover, in the paper we show how training PoserNet on small graphs still allows it to generalise well on large graphs, making the choice of using small graphs not restrictive.

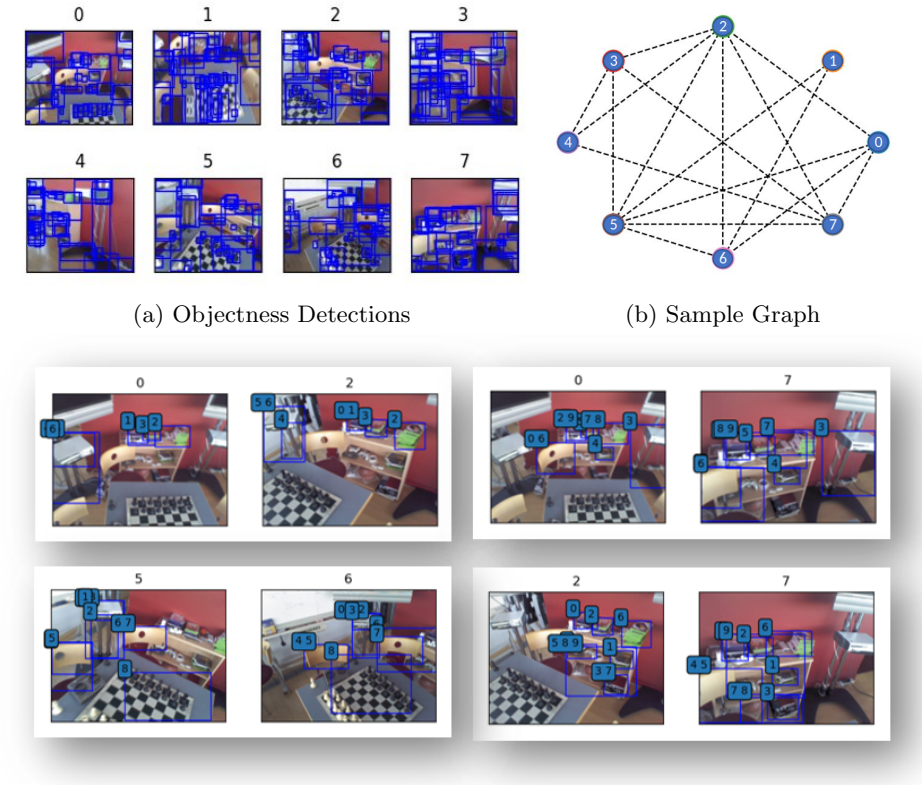
ROIs matching – To establish which candidate nodes are connected, we matched ROIs across images. We passed each image pair through the pre-trained SuperGlue [6] model, generating a set of matched keypoints for the pair of images. Of those keypoints, we kept only the ones contained in the bounding boxes (BB) associated with the ROIs. We then defined as matched all BB pairs that shared at least 15 matched keypoints, with the keypoints spanning at least 30% of the area of the bounding box. These two criteria were defined to ensure that matching bounding boxes enclose the same scene element, and that the shared scene elements constitute a large chunk of the bounding box. We created one edge between two images in a graph if the images were connected via at least five matched BBs.

Relative pose estimation – For each edge, we provide two estimates of the relative transformation mapping points between the connected nodes. The estimates were generated using the OpenCV [2] implementation of the 5-point algorithm, using as inputs either all matched keypoints, or the centres of the matched BBs. Keypoint-based initialisation and BB-based initialisation result in estimates with different levels of noise: the keypoints - aside from some noise and possible mismatches - are much closer to the ideal input of the 5-point algorithm than the BB centres. Even if the two BBs were tightly fitting the same object in the two views, the BB centres would not correspond to projections of the same 3D points; and the ROIs are rarely well centred on the objects.

Graph's topology – In the previous steps we generated all elements necessary to build our graph (nodes and edges), but we now have to check how connected the graph is. The presence of isolated nodes or having multiple disconnected subgraphs is not an issue for PoserNet per se, as the former are ignored and the latter are optimised in parallel. For this reason, we accept the large graphs as they are. In the small-graph case, however, given the limited number of nodes available, we reject all graphs with isolated nodes or isolated subgraphs.

3 Implementation Details

In this section, we discuss implementation details for the third-party code referenced in the paper (MultiReg and EIG-SE3) and for PoserNet. We provide general information on the hardware and computational time involved in the experiments, and on the changes implemented in the third-party code we used in our experiments.



(a) Objectness Detections (b) Sample Graph

(c) Matched Detections

Fig. 2: Sample input graph. We select eight images and obtain objectness detections via OLN (a) and use them as nodes in a graph (b). The number on top of each image represents the image ID, which also corresponds to the node ID as shown in (b). Two nodes are connected if they share at least five matched detections. In (c) we show matched detections for pairs of images. Detections matched across each image pair and indicated with the same number (blue box).

PoserNet. The MLPs used in PoserNet for updating the embedding of the edges (Ψ_e) and of the nodes (Ψ_n) consist in three fully connected layers, each followed by leaky ReLUs. The number of units in each layer is 32. We employed the Adam optimiser, with a learning rate of 10^{-3} for the experiments training on small graphs, and of 10^{-2} for the experiments training on large graphs. The difference was motivated by the different speed of learning we observed in the two cases. We adopted a learning rate scheduler, which reduced the value of the learning rate after three epochs without improvements on the validation loss. Regarding α, the parameter used to tune the strength of the different components of the loss, we found that the best results are obtained by setting $\alpha = 0.1$.

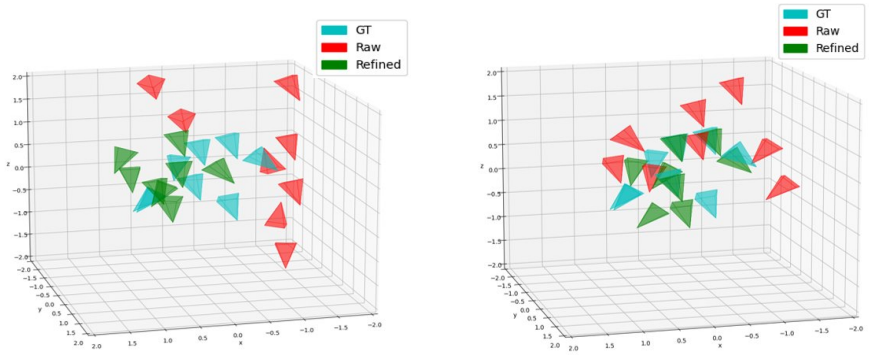
MultiReg. To use MultiReg [7] in our experiments, we implemented as few changes as possible to the original code. First, we added data loaders to process our 7-Scenes graphs; this narrowly follows the process used by the original MultiReg method to process the ScanNet dataset. Unlike in the original method, we do not augment data by randomly sub-sampling each graph; the ScanNet training data used in the original paper was composed by 60 graphs, each containing from 50 to over 100 nodes. Our training set, instead, contains $14k$ graphs of 8 nodes each, making the augmentation step unnecessary, if not detrimental. As a last change, we relaxed the conditions for classifying an edge as outlier: the original work labels as outliers and it ignores all graph edges with an associated rotation error of over 15° or a translation error of more than 0.15 m. We removed this constraint, as it would lead us to ignore a large number of edges. Training MultiReg on the eight-nodes sequences took on average 12 hours on a GeForce RTX 2080 Ti with 11 GB of RAM, while evaluation on the test set required less than one minute.

EIG-SE3: We ran the EIG-SE3 algorithm using the Matlab code originally released by the authors of [1], with no modification and just looping over all testing graphs. This process does not require training, and processing the testing set ($7k$ 8-nodes graphs) requires approximately 50 minutes. This is significantly longer than MultiReg's evaluation time, though the process could be made more efficient, by parallelising it and evaluating in batches.

We provide examples of the absolute camera poses generated by EIG in Figure 3, comparing the poses obtained starting from relative poses with and without PoserNet refinement. In the example we can see how PoserNet can lead to significantly more accurate poses, both for relative poses initialised using matched keypoints or the matched detection BB centers. We can see how, in the case of BB-based initialisation, using the raw input results in camera poses so noisy that EIG cannot properly align them to the ground truth poses. We also point out that the poses obtained starting from the PoserNet-refined inputs have the correct orientation, and the discrepancies with respect to the ground truth poses are mostly limited to translations. This is not surprising, as the results in Tables 2,4 of the main paper show how PoserNet improvement of the relative poses is mostly noticeable in the relative rotations rather than in the translation direction.

4 Highlighting advantages of PoserNet

To better highlight the significant benefit of refining the relative pose with PoserNet, in Fig. 4 we provide plots of the rotation and translation error distributions over all eight-node testing graphs. To increase interpretability, we sorted the sequences in ascending order of error on the PoserNet-refined predictions. Moreover, given the large number of sequences and likely presence of outliers, we smooth the curves using a rolling average over a window of 50 graphs.



(a) BB-based initialisation (b) Keypoint-based initialisation

Fig. 3: Effects of PoserNet on the motion averaging output of EIG. We compare ground-truth camera poses (azure) with the output of EIG initialised with the raw relative camera poses (red) and the ones refined with PoserNet (green). For relative poses obtained with both the bounding-box (a) and key-point (b) initialisation method, EIG's output is significantly improved by using PoserNet to refine the input relative poses.

5 Ablation on the depth of PoserNet

To select the optimal number of rounds of message passing (characterised as GNN "depth") for the GNN in PoserNet, we trained four versions of PoserNet with depths ranging from two to five. For this experiment we trained each model for the same amount of time (24 hours), initialising the edges of the small graph dataset with the keypoint-based relative poses. As it can be seen in Table 1, the highest accuracy is achieved with a GNN of depth 2, which is the value we used for all the remaining experiments described in the main document.

	2-level	3-level	4-level	5-level
Relative orientation error	**7.5**	7.6	9.4	11.2
Translation direction error	**14.7**	14.9	16.1	17.2

Table 1: Median performance of PoserNet as a function of GNN depth, errors expressed in degrees. Best performance (in bold) was achieved with a two-level GNN.

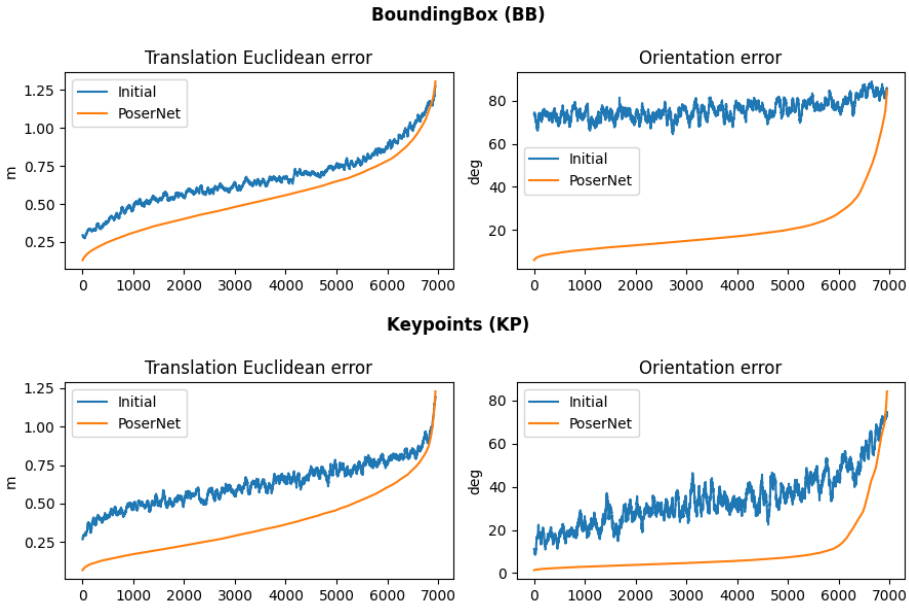


Fig. 4: EIG-SE3 Motion Averaging performances on raw inputs and on inputs refined by PoserNet. There is a substantial error reduction both for relative poses computed from the detection centers (Top) and keypoints (Bottom).

6 Ablation on the keypoint descriptors

In all our experiments, the initial relative poses are obtained applying the OpenCV 5-point algorithm to either the 2D centres of matched bounding boxes, or keypoints SuperGlue. These two choices represents opposite use cases, with the BB centres providing a very rough initialisation (even if the bounding boxes are correctly matched, their centres are not guaranteed to correspond to the same 3D point), and SuperPoint+SuperGlue being the state-of-the-art for keypoint detections and matching. We assumed that any other common choice of keypoint matching would result in input relative pose accuracy subsumed in the range of values covered by our two initialisation strategies, and that showing how PoserNet can significantly improved on both could be generalised to the other cases. To support this point, we show in Table 2 how estimating the relative pose starting from points matched via SIFT or ORB is more accurate than using BB centres, and less accurate than using SuperPoint + SuperGlue. Moreover, we must point out that while using ORB provided better results than SIFT, the ORB matches caused the 5-point algorithm to fail in 30% of the image pairs.

Metric	BB	SIFT	ORB*	GLUE
R	96.48	53.53	74.84	**36.27**
t	89.30	90.73	90.20	**87.23**

Table 2: Median rotation (R) and translation (t) error using SIFT, ORB, SuperPoint+SuperGlue (GLUE), and matched BB center (BB) to compute the relative poses. $*$ Failed on 30% of pairs.

7 Generalisation of PoserNet to novel scenes

To assess PoserNet generalisation capabilities, we evaluate on each of the seven scenes a model trained exclusively on the other six. For these tests, we consider only eight-node graphs, and we use the key-point based relative pose initialisation approach. The results of this leave-one-out scheme, reported in Table 3, suggest PoserNet can generalise well to novel scene: averaging the performances over all seven tests results in average rotation and translation orientation errors compatible with the those obtained training on the full dataset.

	Office	Pumpkin	RedKitchen	Stairs	Chess	Fire	Heads	Median	All
Rotation	5.05	9.93	7.33	9.39	7.27	8.36	8.96	8.04	7.31
Translation	10.59	13.05	16.82	18.07	21.03	11.15	47.54	19.75	14.54

Table 3: Median performance of PoserNet on each scene when trained only on the other six. The overall performance over the seven tests (Median) is comparable with the performances obtained training PoserNet on the full dataset (All).

References

1. Federica Arrigoni, Beatrice Rossi, and Andrea Fusiello. Spectral synchronization of multiple views in se (3). *SIAM Journal on Imaging Sciences*, 9(4):1963–1990, 2016.
2. G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
3. Ben Glocker, Shahram Izadi, Jamie Shotton, and Antonio Criminisi. Real-time rgb-d camera relocalization. In *International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, October 2013.
4. Dahun Kim, Tsung-Yi Lin, Anelia Angelova, In So Kweon, and Weicheng Kuo. Learning open-world object proposals without learning to classify. *IEEE Robotics and Automation Letters (RA-L)*, 2022.
5. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

6. Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020.
7. Zi Jian Yew and Gim Hee Lee. Learning iterative robust transformation synchronization. In *International Conference on 3D Vision (3DV)*, 2021.