2020 20th International Conference on Control, Automation and Systems (ICCAS 2020)
Oct. 13∼16, 2020; BEXCO, Busan, Korea

Viewpoint Estimation for Visual Target Navigation by Leveraging Keypoint Detection

Yunho Choi, Nuri Kim, Jeongho Park, and Songhwai Oh*

Department of Electrical and Computer Engineering, Seoul National University,
Seoul, 08826, Korea ({yunho.choi, nuri.kim, jeongho.park}@rllab.snu.ac.kr, songhwai@snu.ac.kr) * Corresponding author

Abstract: In this paper, we tackle the problem of visual target navigation which is a crucial task for embodied AI. Given a target observation and an initial observation, we extract keypoints using an off-the-shelf keypoint detection model. Then we find sparse keypoint correspondences and estimate the relative camera pose with the PnP algorithm to reach the viewpoint for the target observation. Compared to conventional approaches, our method is faster, and more robust to scene changes and occlusions. We collected a 3D scan dataset in a university building and the proposed method is verified using the dataset.

Keywords: visual navigation, viewpoint estimation, visual servoing, keypoint detection

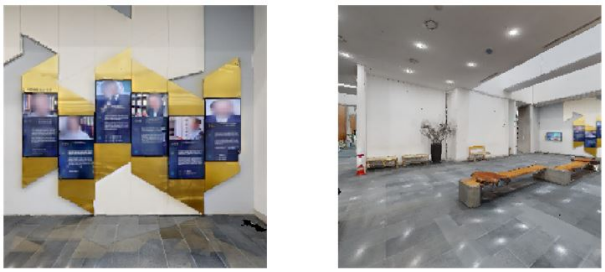
1. INTRODUCTION

Traditional Approaches for robot navigation in novel environments involves mapping, localization, and motion planning based on metric information. However, metric-based representations are often sensitive to actuation noises and not scalable for large environments, which make the traditional approaches brittle for the real world deployment.

In contrast, humans do not use metric representations for navigation. Humans rather use various cues from visual observations such as semantic context and spatial geometry, which make their navigation policy flexible for various visual navigation tasks including the point goal navigation and visual target navigation [1] and robust to actuation noises. Recently, with the development of photo-realistic simulators [1, 2] which can simulate the robot navigation in existing real world environments, there has been a growing body of studies on embodied AI for visual navigation [1–6] showing that the learning-based methods outperform traditional approaches such as SLAM [1] or visual servoing methods [5].

In this paper, we focus on the problem of visual target navigation, where the goal is to reach the target view given a target image. Conventional image-based visual servoing (IBVS) methods are susceptible to the set of detected features and the depth value as shown in [5]. While recent learning-based approaches mitigates these problems by exploiting a representation extracted by a neural network [3,6], they are vulnerable to subtle scene changes and occlusions since they use a global feature computed from a whole image.

Rather, we propose to leverage recent improvements on local keypoint detection techniques [8]. We extract keypoints and its descriptors of both current observation and target observation, find the sparse keypoint correspondences, and calculate the relative camera pose to reach the target view from the keypoint correspondences. Since we exploit sparse keypoint correspondences, our method is suitable for the real-time navigation and robust to scene changes.



(a) Target Observation (b) Initial Observation

Fig. 1: Visual Target Navigation: A target and initial observation are given. Starting from the initial viewpoint, our goal is to reach the viewpoint of the target observation.

Similar techniques are used in the literature of visual localization [7], though these visual localization schemes use the dense feature matching which is time-consuming and assume a 3D model of the environment is given. Li, et al. [5] is the most similar approach to ours in that correspondences between a current observation and target observation are utilized. However, the authors of [5] make an unrealistic assumption that a dense correspondence map between the current and target observation is given, and learn an action model from the correspondence map. On the other hand, our method can find sparse correspondences from an off-the-shelf learned keypoint detection model and directly estimate the viewpoint for the target observation, which is more similar to the way humans navigate.

2. PROPOSED METHOD

2.1 Problem Definition

We consider an embodied agent that navigates in a novel environments. Without any pre-built metric map, the agent navigates solely depending on visual observations from a front camera. We assume that the agents

starts at initial state $s_0 = (x_0, y_0, \theta_0)$ with an initial observation I_0. The observation at time t, I_t, can be either an RGB or an RGB-D image. A target observation, I_g, taken at the target viewpoint s_g is given as an RGB image. I_0 and I_g are assumed to have some overlapping region. Visual target navigation is a task where the agent's goal is to reach the target viewpoint s_g. In this paper, we directly estimate s_g rather than learning a control policy to reach the target.

2.2 Keypoint Detection and Matching

Recent methods on keypoint detection [8,9] jointly detect and describe keypoints in a image. A descriptor of a keypoint can be seen as a local feature, and it can be utilized for various tasks such as place recognition and homography estimation. We first extract keypoints and its 128-D descriptors in I_0 and I_g with an off-the-shelf R2D2 network [8]. Then we find the mutual nearest neighbor matchings between the two sets of keypoints, with the basic ratio test for outlier rejection. More sophisticated techniques such as [10] can also be applied for more robust and faster keypoint matching.

2.3 Computation of Relative Camera Pose

We assume that the camera we use is calibrated beforehand and the camera matrix is known. In addition, we assume observations I_t to be RGB-D images while I_g is an RGB image. This is a reasonable setting if you imagine an agent finding a target viewpoint from a snapshot or street view image (offline) while making use its RGB-D sensor (online). Then, we can get the 3D position of the keypoints in I_t with respect to the current state s_t. With the 2D-3D correspondences acquired above, the relative camera pose can be estimated by PnP algorithm, which we implement by PnP RANSAC function of OpenCV [11] for robustness to outlier matchings.

After moving according to the relative camera pose computed from I_0 and I_g, we can further refine our viewpoint estimation with multi-step iterations of above procedure. If we have RGB observations instead of RGB-D observations, we can estimate a essential matrix from 2D-2D correspondences and get a rotation matrix and a translation vector up to scale. Setting a unit step size for translation and executing the changed procedure iteratively can also be a method for viewpoint estimation, which will not be examined in this paper.

3. EXPERIMENTS

3.1 Dataset

For evaluation of our method, we collected a 3D scan of the university building with the Matterport Pro2 Camera used in [12] which results in 3D mesh data. Then we processed the mesh data for navigation with the same procedure used in [1], and made it loadable in the Habitat simulator [1]. Then we collected 60 pairs of initial I_0, s_0 and target I_g, s_g as seen in Figure 1. We divided the whole dataset into three difficulties as shown in Table 1. The distance and heading towards the target viewpoint and existence of occlusions affect the difficulty. Following experiments will be evaluated on the Habitat simulator, where the navigation with various sensors are supported.

3.2 Evaluation of the Proposed Method

We define some performance metrics to evaluate the proposed method. *Average Initial Distance* is the distance between s_0 and s_g, i.e., $\sqrt{(x_0 - x_g)^2 + (y_0 - y_g)^2}$. *Average Initial Viewpoint Error* is defined as the heading difference between s_0 and s_g, i.e. $|\theta_0 - \theta_g|$. Likewise, *Average Final Distance* and *Average Final Viewpoint Error* are defined similarly. We define a success if the agent's *Average Final Distance* ≤ 0.5 *m* and *Average Final Viewpoint Error* $\leq 10\,^\circ$ both satisfy. If the agents fails the task, *Average Final Distance* and *Average Final Viewpoint Error* are masked as *Average Initial Distance* and *Average Initial Viewpoint Error*.

In Table 1, the difficulties of datasets and the performance of the proposed method for each dataset are evaluated according to the defined performance metrics. The proposed method is hard to be directly compared with [3, 5, 6] since it directly estimates the viewpoint. Here, we evaluate the one step of our algorithm and the performance has room to be further refined. As seen in Table 1, it achieves a success rate of 85% in the moderate difficulty. Considering that the scenes seen in Figure 2 is classified as moderate ones though they lack distinctive features and have bad features resulting from unstable point clouds from leafage, the performance of our method is notable. Some of the successful cases are shown in Figure 3. As seen in Figure 3-(c),(d), our method can robustly estimate the target viewpoints even when there are large occlusions for the target scenes.

Table 1: Performance metrics according to the difficulty of the dataset.

Dataset Difficulty	Moderate	Hard	Extreme
Avg. Init. Dist. / *Avg. Init. VP Error*	2.23 *m* / 42.75°	5.28 *m* / 55.69°	8.39 *m* / 49.50°
Avg. Fin. Dist. / *Avg. Fin. VP Error*	0.71 *m* / 16.64°	3.99 *m* / 43.35°	7.96 *m* / 45.23°
Success Rate	85%	40%	30%

4. CONCLUSION

Our proposed method leverages sparse keypoint correspondences to robustly estimate the target viewpoint, which are experimentally verified with the dataset we collected. Our method also has much room to extend, such as finetuning the feature points [13] and the use of learning-based robust estimator [14]. For future work, we are planning to work on multi-step refinement of the proposed method and learning control policy based on keypoints representations, so that the agent can robustly find the target viewpoint even when started from the ini-

(a)

(b)

Fig. 2: Target observations (left) and initial observations (right) of failure cases. The keypoints and the matching between the two sets of keypoints are depicted on the figure. The number of keypoints and matchings are written in the text overlaid on the left figures. (a) Insufficient features. (b) Leafage.

tial view that does not have any overlap with the target view.

5. ACKNOWLEDGMENT

This work was supported by the Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2019-0-01309, Development of AI Technology for Guidance of a Mobile Robot to its Goal with Uncertain Maps in Indoor/Outdoor Environments)

REFERENCES

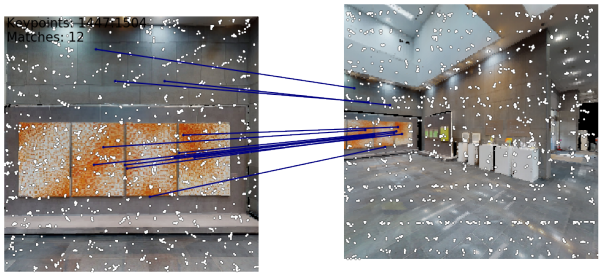
[1] M. Savva, A. Kadian, O.r Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, D. Batra, "Habitat: A Platform for Embodied AI Research." in Proc. of *the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[2] Fei Xia, Amir R. Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, Silvio Savarese, "Gibson Env: Real-World Perception for Embodied Agents." in Proc. of *the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

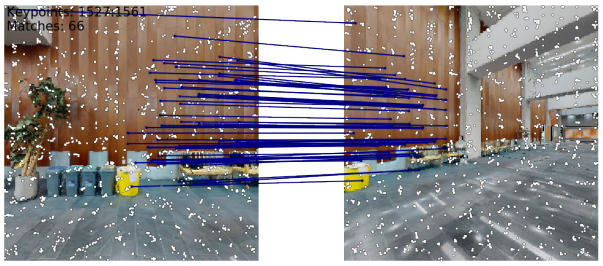
[3] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning" in Proc. of *the IEEE international conference on robotics and automation (ICRA)*, 2017.

[4] A. Kumar, S. Gupta, D. Fouhey, S. Levine, J. Malik, "Visual Memory for Robust Path Following." in Proc. of *the Advances in Neural Information Processing Systems (NIPS)*, 2018.

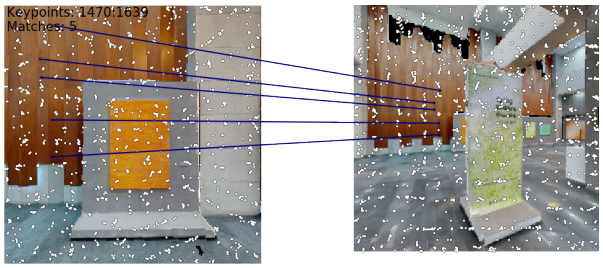
[5] Yimeng Li and Jana Kosecka. "Learning View and Target Invariant Visual Servoing for Navigation." in Proc. of *the IEEE International Conference on*



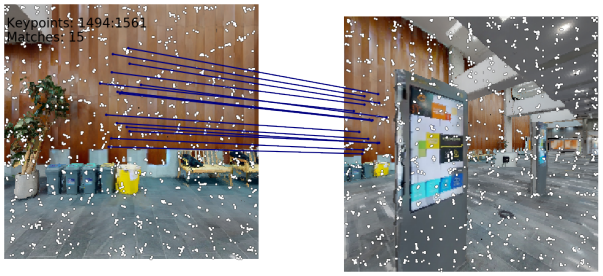
(a)



(b)



(c)



(d)

Fig. 3: Target observations (left) and initial observations (right) of successful cases. The keypoints and the matching between the two sets of keypoints are depicted on the figure. The number of keypoints and matchings are written in the text overlaid on the left figures.

Robotics and Automation (ICRA), 2020.

[6] D. Singh Chaplot, R. Salakhutdinov, A. Gupta, S. Gupta, "Neural Topological SLAM for Visual Navigation." in Proc. of *the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[7] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, A. Torii, "InLoc: Indoor Visual Localization with Dense Matching and View Synthesis." in Proc. of *the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[8] J. Revaud, P. Weinzaepfel, C. De Souza, N. Pion, G. Csurka, Y. Cabon, M. Humenberger, "R2D2: Repeatable and Reliable Detector and Descriptor." in Proc. of *the Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[9] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, T. Sattler, "D2-Net: A Trainable CNN for Joint Description and Detection of Local Features." in Proc. of *the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[10] P.-E. Sarlin, D. DeTone, T. Malisiewicz, A. Rabinovich, "SuperGlue: Learning Feature Matching With Graph Neural Networks." in Proc. of *the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[11] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography." *Communications of the ACM*, 24(6):381–395, 1981

[12] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, Y. Zhang, "Matterport3D: Learning from RGB-D Data in Indoor Environments." in Proc. of *the International Conference on 3D Vision (3DV)*, 2017.

[13] A. Bhowmik, S. Gumhold, C. Rother, E. Brachmann, "Reinforced Feature Points: Optimizing Feature Detection and Description for a High-Level Task." in Proc. of *the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[14] E. Brachmann, C. Rother, "Neural-Guided RANSAC: Learning Where to Sample Model Hypotheses," in Proc. of *the IEEE International Conference on Computer Vision (ICCV)*, 2019.