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

GOPE: Geometry-Aware Optimal Viewpoint Path Estimation Using a Monocular Camera

Nuri Kim, Yunho Choi, Minjae Kang and Songhwai Oh*

Department of Electrical and Computer Engineering and ASRI, Seoul National University, * Corresponding author
Email: {nuri.kim, yunho.choi, minjae.kang}@rllab.snu.ac.kr and songhwai@snu.ac.kr

Abstract: The goal of the optimal viewpoint path estimation is to generate a path to the optimal viewpoint location where the robot can best see the Point of Interest (POI). There are several learning-based methods to find an optimal viewpoint, but these methods are limited to a specific object POI and it is necessary to newly learn in a situation where a new POI is added, and not robust to the environment changes. In this paper, we propose an algorithm that generates a path to the optimal viewpoint by using the geometrical features of the environment in the situation where the target POI is in the field of view. This method makes it easy to add new POIs and is robust to environmental changes because it uses semantic and geometric information. We assume that the robot can make a simple estimation of the geometric characteristics of the surrounding environment by using pretrained networks or by using sensor values. We collected the Kwanjeong street dataset for testing our algorithm. In this dataset, the distance accuracy of our method to reach the optimal viewpoint of the POI achieved 81.8% and 70.9% for template matching accuracy.

Keywords: Visual Navigation, Deep Learning, Optimal Viewpoint.

1. INTRODUCTION

Robot localization in a lost situation is an important problem in the long-horizon visual navigation problem that needs to travel a very long path [2, 5, 11]. Since there are localization errors in a robot, if a robot gets lost while navigating in the absence of GPS, the robot navigates to nearby places to find a known point of interest (POI) to update its position. In this stray situation, it is better to find a POI which should exist nearby. Therefore, to reduce the localization uncertainty of a robot, the robot finds an optimal viewpoint of a POI where the POI is best viewed. Even though there are several learning-based methods [3, 7–10, 13] to find an optimal viewpoint, these methods are limited to a specific object POI and it is necessary to newly learn in a situation where a new POI is added, and not robust to the environment changes.

In this paper, we propose a method to generate a path toward an optimal point where the POI to be searched is best viewed using geometric information about the surroundings. Semantic information is extracted using pretrained depth estimation, surface normal estimation, and text detection networks, which helps to determine the geometry around the robot and create a suitable path to get to the optimal viewpoint. As shown in Figure 1, path generation consists of three steps. First, the POI is centered in the image to obtain accurate semantic information. Here, the rotation angle of the robot is calculated using depth information. Second, the most suitable text proposal is selected through matching text proposals and a template of the POI. Comparing the size of the POI and the whole image, the distance for the optimal viewpoint from a POI is calculated. Lastly, using the fact that the optimal viewpoint is in front of the POI, a path to the optimal viewpoint is created. In summary, we use all steps to create command as follows: “Turn 30° to the right, then move 7m in the direction of 30°, and turn 80° to the left.”

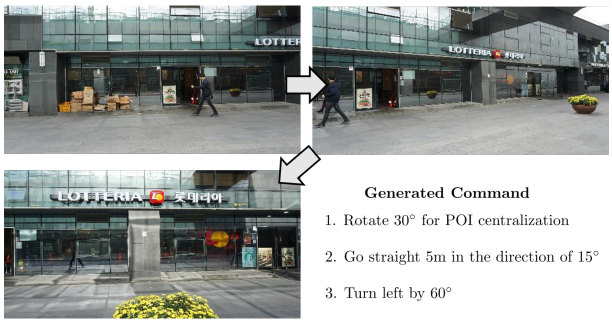


Fig. 1: Example output of our optimal viewpoint path estimation algorithm. When the upper image came in, our algorithm firstly finds the target (Sign of Lotteria) and makes a robot to rotate. Secondly, it finds an optimal viewpoint, where the target is most visible, and move to the optimal viewpoint.

Then the robot moves according to this command (see Figure 1).

To test our algorithm, we collected the Kwanjeong street dataset, which consists of 44×7×12 images of 30 degrees at each grid node. As a result, our algorithm shows the distance accuracy of 81.8% and template matching accuracy of 70.9% on the Kwanjeong dataset.

2. RELATED WORK

The problem of optimal viewpoint path estimation is related to various tasks, e.g., visual navigation and active viewpoint estimation. Below, we briefly introduce representative methods.

There are many papers in visual navigation tasks [2,5]. [2] generates topological graph for each location and connect the relationship between nodes to transfer information about the location to other locations. [5] uses Transformer [12] for effectively following a path.

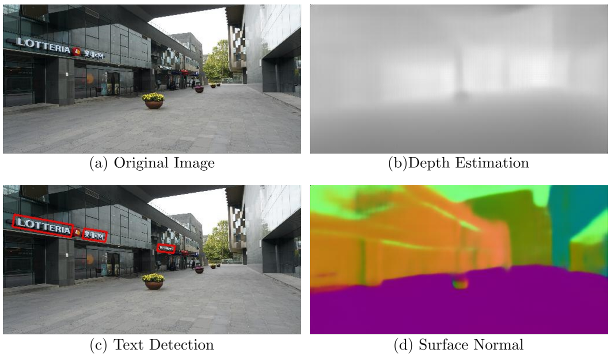


Fig. 2: Output of the sub-algorithms in optimal viewpoint path estimation. **(a)** Original image, **(b)** Depth estimation, **(c)** Text estimation, **(d)** Surface normal estimation.

In active viewpoint estimation, the key problem to tackle is occlusion of the target objects [3, 13]. For reducing occlusion, [3] finds an optimal viewpoint of an object by constructing a 3d object. Although [3] works well, it did not generate a path for the object. The most similar work with ours is [13], which learns to move to reduce the occlusion of the target object. These researches all used learning techniques and tested only on the simulator, while our work use pretrained visual geometric knowledge for environment-robust optimal viewpoint path estimation and tested on the real-world.

3. PROPOSED APPROACH

3.1 Sub-Algorithms

The proposed optimal viewpoint path estimation algorithm uses four sub-algorithms (depth estimation, surface normal estimation, text detection, and template matching). Since sub-algorithms are important for the performance of our algorithm, the sub-algorithms are explained in detail. The result of the sub-algorithm can be seen in Figure 2.

3.1.1 Depth estimation

For depth estimation, we used pretrained Taskonomy [14]. The output of the algorithm gives a value from 0 to 1. If the closer it gets to 0, the farther it gets 1. In order to transform this value to the meter, we used the distance correction value M, which is multiplied to the depth estimation value (M = 19.2).

3.1.2 Surface normal estimation

We used pretrained Taskonomy [14] for surface normal estimation. The output of the estimation is from -1 to 1 with a 3-dimensional vector. Because it was trained in an indoor environment, there are some errors in the outdoor environment, such as the environment we tested.

3.1.3 Text detection

The CRAFT [1] was used for text detection. This algorithm is the best performing text detection algorithm that is currently open-source. When we tested the CRAFT algorithm [1], the algorithm is the best working when the

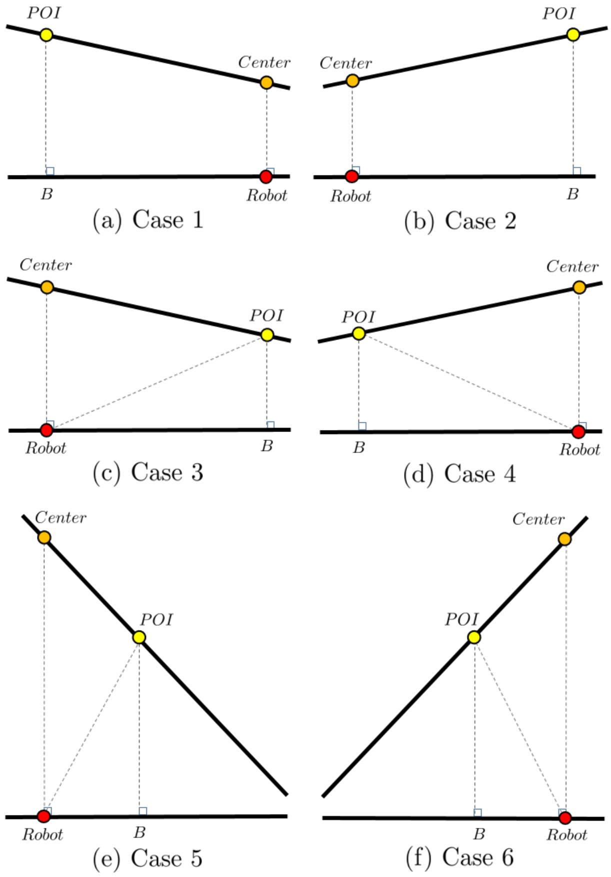


Fig. 3: Six arrangements of a POI and a robot. (a) Case 1: $\overline{PB} > \overline{CR}$ and POI on the right side, (b) Case 2: $\overline{PB} > \overline{CR}$ and POI on the left side, (c) Case 3: $\overline{PB} < \overline{CR}$, POI on the right side, and $\angle CPR < 90°$, (d) Case 4: $\overline{PB} < \overline{CR}$, POI on the left side, and $\angle CPR < 90°$, (e) Case 5: $\overline{PB} < \overline{CR}$, POI on the right side, and $\angle CPR > 90°$, (f) Case 6: $\overline{PB} < \overline{CR}$, POI on the left side, and $\angle CPR > 90°$.

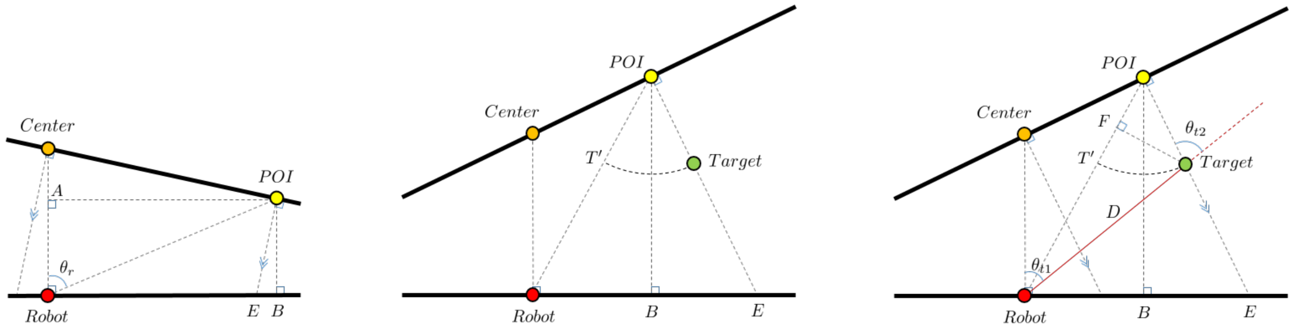
size of the text is within a certain distance. Therefore, in order to better detect a small text, which is located far away, we use multi-scale images as input. Then the output text boxes are collected to detect multi-scale text detection.

3.1.4 Template matching

Template matching is used to find the target POI. Here, the template of the POI is an object image taken from various angles. After resizing the template and the text box to be tested to a size of 224x224, we calculate a cosine similarity of two images using the last feature point of the ResNet18 [6] pretrained with ImageNet [4]. After measuring the cosine similarity between the template of the POI and the detected text detection box, we extract the template with the highest value. If the similarity score is 0.5 or higher, the text box is considered as the POI.

3.2 Optimal Viewpoint Path Estimation

In order to estimate the optimal viewpoint, we divided the problem into six cases using the relative position of a



(a) POI Centralization (b) Target Estimation (c) Path Estimation

Fig. 4: The three steps of the optimal viewpoint path estimation. **(a)** A robot makes the POI into the center of the image by rotating as an amount of θ_r. **(b)** Using the height of the POI and the surface normal vector of the POI plane, the location of the target is estimated. **(c)** Using the geometric information from the sensor, calculate the path to reach the optimal viewpoint (θ_{t1}, D, and θ_{t2} are the path parameters).

POI and a center from a robot (see Figure 3). Since the cases mostly affect to the rotation direction of a robot, we only explain a single case of Figure 3(c) where the plane of a POI is located on the left side of a robot, the POI is on the right side of the screen, and the distance to a center is shorter than the distance to a POI from a robot. We will notice if there is a change in the equation for each case. As shown in Figure 4, the process of the optimal viewpoint path estimation is composed of three steps: (1) POI centralization, (2) Target estimation, and (3) Path estimation.

3.2.1 POI centralization

Firstly, we centralize the POI to the camera center for better geometric estimation from the visual input. The robot extracts the multiple text boxes and compares the boxes with the POI templates to select the most similar text box with the target POI. Then, surface normals corresponding to the text box are extracted and then averaged to obtain an accurate surface normal vector of the POI. We do the same for the image center.

To centralize the POI, we calculate the rotation angle between the camera center and the POI (θ_r in Figure 4(a)). By measuring the depth of the POI (P) and the center (C) from the camera plane, we can calculate the $\overline{PB}$ and $\overline{CR}$. Then, the angle between the surface normal of the plane containing the POI and the camera plane surface normal is the $\angle PCR$. Using $\overline{CR}$, $\overline{PB}$, $\angle PCR$, we can calculate θ_r:

$$\theta_r = \tan^{-1}\frac{(\overline{CR} - \overline{PB})\tan(\angle PCR)}{\overline{PB}}. \tag{1}$$

The robot rotates by the θ_r for making the robots' center align with the POI.

3.2.2 Target estimation

Target estimation refers to the process of estimating where the target which is the optimal viewpoint of a POI. The optimal viewpoint is located on the surface normal line of the POI ($\overline{PE}$ in Figure 4(b)). Since the POI is not easily visible when it is too close or too far from POI, the target with a proper distance was set as the target. The proper distance is defined by using the ratio of the height of the POI in the image and the total image height. We used the height because the width of the POI can be changed according to the angle change of a robot, but the height is directly proportional to the distance between a robot and a POI.

Note that even though the center of the robot becomes closer to the POI, the center and the POI cannot be aligned perfectly because of the localization error of a robot. As can be seen in Figure 4(b), the distance between the center and the POI is not zero. In this situation, the position of the POI is estimated using the template matching and the surface normal ($\overrightarrow{PE}$) and the depth ($\overline{PB}$) of the POI, and the depth of the center ($\overline{CR}$) is estimated. As in the POI centralization, the angle ($\angle EPB$) between the surface normal vector of the plane to which the POI belongs and the surface normal vector of the robot camera can be obtained.

Then, the height of the POI is calculated, and the ratio (λ_r) between the height of the POI and the height of the current image is calculated. $\overline{PT'}$, which is the most suitable distance for viewing a POI, is calculated as follows (Figure 4 (b)):

$$\overline{PT'} = \overline{PR}\cdot\sqrt{\lambda_r/\lambda_{or}}, \tag{2}$$

where $\overline{PR} = \sqrt{\overline{PB}^2 + ((\overline{PB} - \overline{CR})/\tan(\angle EPB))^2}$. λ_{or} is an optimal ratio of a POI's height and an image height. Since $\overline{PT'} = \overline{PT}$, calculating the point which has the distance $\overline{PT'}$ on the surface normal vector ($\overrightarrow{PE}$) of a POI, we can estimate the target position.

3.2.3 Path estimation

After finding the target location, the robot generates the path to approach a target viewpoint. The robot firstly rotate left as θ_{t1} and move forward D meters, and then rotate right as θ_{t2} to reach the optimal viewpoint for a POI (see Figure 4 (c)). For this, the robot calculates the

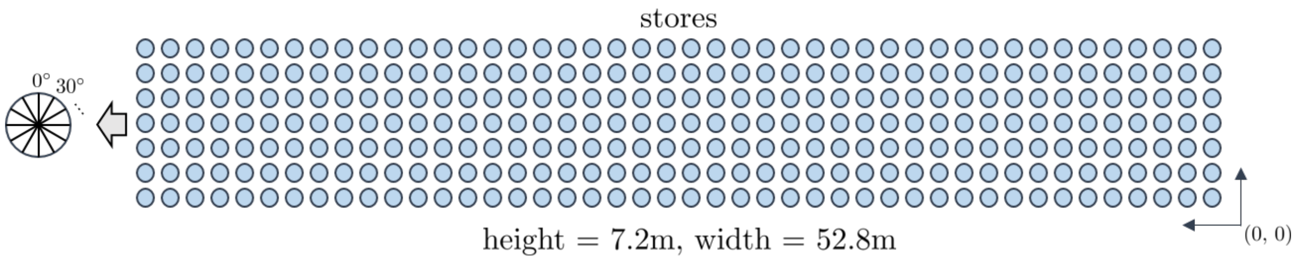


Fig. 5: Data nodes of the Kwanjeong street dataset. Each node has 12 images at 30°intervals. The right bottom point is the (0, 0) and there are seven stores on the top.

Table 1: Distance accuracy and Template matching accuracy for each target POI in the Kwanjeong Dataset.

	Frangcors Fancy	CU	Hugh Gimbap	Lotteria	Paris Baguette	Cafe Pascucci	Lotte Tour	mean ACC
Distance Accuracy	85.8	93.6	89.6	78.4	68.2	82.0	75.3	81.8
Template matching accuracy	97.2	98.8	36.0	63.5	79.3	91.2	31.6	70.9

$\angle PRB$:

$$\angle PRB = \tan^{-1}\frac{\overline{PB}}{\overline{RB}}, \quad \text{where } \overline{RB} = \frac{\overline{PB} - \overline{CR}}{\tan(\angle TPA)}. \tag{3}$$

Since $\triangle PRB$ is a right triangle, $\angle RPB = \pi/2 - \angle PRB$. Using this, we can calculate the $\angle RPT$. There is a difference while calculating $\angle RPT$ for each cases. For case 1 and 2,

$$\angle RPT = \pi/2 - \angle PRB + \angle TPA, \tag{4}$$

and for other cases,

$$\angle RPT = \pi/2 - \angle PRB - \angle TPA. \tag{5}$$

Since $\overline{TF} = \overline{PF} \sin(\angle RPT) = (\overline{PR} - \overline{PT} \cos(\angle RPT)) \tan(\angle PRT)$,

$$\angle PRT = \tan^{-1}\frac{\overline{PT}\sin(\angle RPT)}{\overline{PR} - \overline{PT}\cos(\angle RPT)}. \tag{6}$$

We calculate the θ_{t1}, $D = \overline{RT}$, and θ_{t2} using eq (3), eq (4), eq (5), and eq (6):

$$\begin{aligned} \theta_{t1} &= \angle CRT = \pi/2 - \angle PRB + \angle PRT, \\ D &= \sin(\angle RPT)\frac{\overline{PT}}{\sin(\angle PRT)}, \\ \theta_{t2} &= \angle RPT + \angle PRT. \end{aligned} \tag{7}$$

In summary, (1) the robot is rotated by θ_r to centralize the POI. (2) The target viewpoint point is calculated using the height of the POI on the image and the surface normal of the POI. (3) Then, we tackled the problem of generating the optimal viewpoint path by calculating θ_{t1}, D, and θ_{t2}. For the case 1, 4, and 5, the θ_{t1} is positive (turn left) and θ_{t2} is negative (turn right).For other cases, θ_{t1} is negative (turn right) and θ_{t2} is positive (turn left).

4. EXPERIMENTS

4.1 Kwanjeong Street Dataset

To test the optimal viewpoint path estimation algorithm, we collect a total of 3696 images from the street next to the Seoul National University Library (Kwanjeong street dataset). There are 44 horizontal and 7 vertical nodes at 1.2m intervals (see Figure 5). At each node, 12 images are collected at 30° intervals at each node because it is important to move at an accurate angle to estimate the optimal viewpoint. The original size of the image is 6000 $\times$ 3376 (width $\times$ height) and there are a total of seven POIs (Frangcors Fancy, CU, Hugh Gimbap, Lotteria, Paris Baguette, Cafe Pascucci, Lotte Tour) in the Kwanjeong street dataset. Optimal viewpoints of the POIs are labeled as positive when the POI has a height of 10% of the total image height and the POI is in the right front. Then, we estimate the height of the POIs in the optimal viewpoint to use it as λ_{or}. The test set consists of 1679 pairs of a starting point and a target POI. Even if only a small portion of the target POI is visible, it is included in the test set. Since the POI candidates cannot be obtained when text is not detected from the starting point, these are not included in the test set.

4.2 Performance Analysis

We made two measures to calculate the accuracy of our proposed optimal viewpoint algorithm. Firstly, a distance accuracy is defined that the robot is well reached an optimal viewpoint of a POI when the final position is within $5m$ of the target POI and the viewpoint angle is within 60 degrees from the optimal viewpoint angle. We averaged the matching accuracy over each POI (see Table 1). Since our method is to reduce the uncertainty of the POI measurement, the second one measures how accurate the template matching result of the final point is. If the template matching accuracy is 1 when the template matching result is 0.5 or higher (Table 1).

As a result, the distance accuracy was 81.8%, and the template matching accuracy was 71.3%. The Paris Baguette shows the lowest distance accuracy (68.2%), because of the poor template matching. Although there was no ground truth box of POI for each image, it was not

Table 2: Ablation study on three key components in our algorithm for each target POI in the Kwanjeong Dataset.

Centralization	Rotation	Translation	Frangcors Fancy	CU	Hugh Gimbap	Lotteria	Paris Baguette	Cafe Pascucci	Lotte Tour	mean dist. ACC
✗	✔	✔	81.0	91.2	88.4	79.5	68.5	79.0	74.7	80.3
✔	✗	✔	76.8	90.7	82.9	60.3	53.9	69.5	51.9	69.3
✔	✔	✗	46.0	71.5	65.9	35.2	37.2	42.6	44.3	49.0
✔	✔	✔	85.8	93.6	89.6	78.4	68.2	82.0	75.3	81.8

quantitatively evaluated, but after visualizing the failure cases, we discover that it failed to match the template and found a wrong bounding box. It means that our algorithm can be improved by a better template matching algorithm.

4.3 Ablation Study

We measured the impact of three key elements of the algorithm on distance accuracy. First, we measured how much the POI centralization affects performance. Additionally, we show how much the distance accuracy deteriorates when the robot is not translated or rotated (see Table 2).

As a result, POI centralization caused an improvement in distance accuracy by 1.5%, and rotation increased accuracy by 12.5%, and translation increased accuracy by 32.8%. Therefore, translation is the most important component for optimal viewpoint path estimation. The POI centralization did not affect much on the distance accuracy since we design a path based on unaligned cases for the robustness of our algorithm.

4.4 Qualitative Results

For qualitative results, we choose six examples with different POIs. The left column is images from the starting point, and the right column is the images taken from the estimated optimal viewpoint (Figure 6). The POI is written on the bottom right of each starting point image, and the guidance command is written on the right of the paired images. In almost all examples, we can see the robot has reached the POI quite close. If the POI which is located too far cannot be reached, as in the third example. The problem is occurred because of the inaccurate depth prediction algorithm, which gives a depth value between 0 and 1. This problem seems to be tackled by using a depth sensor to the robot and using it as an actual value.

5. CONCLUSION

In this paper, we proposed an algorithm for generating a path for finding an optimal viewpoint of a target POI. This algorithm geometrically calculates the parameters for generating a path. The geometric information is extracted from pretrained deep learning networks, which consists of depth, surface normal, text detection, template matching algorithms. By dividing six cases of geometric structures, we efficiently tackle the problem.

Additionally, we collected the Kwanjeong dataset which has thousands of data nodes and images for algorithm evaluation. Even though the dataset has a hard occlusion of target POIs at the starting point, our algorithm shows 81.8% of distance accuracy and 70.9% of template matching accuracy. It means that our optimal viewpoint finding algorithm can effectively reduce the uncertainty in POI detection while visual navigation.

Acknowledgements: This work was supported by 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] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[2] D. S. Chaplot, R. Salakhutdinov, A. Gupta, and S. Gupta, "Neural topological slam for visual navigation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[3] R. Cheng, Z. Wang, and K. Fragkiadaki, "Geometry-aware recurrent neural networks for active visual recognition," in *Neural Information Processing Systems (NIPS)*, 2018, pp. 5081–5091.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[5] K. Fang, A. Toshev, L. Fei-Fei, and S. Savarese, "Scene memory transformer for embodied agents in long-horizon tasks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] D. Jayaraman and K. Grauman, "Learning to look around: Intelligently exploring unseen environments for unknown tasks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[8] T. Patten, W. Martens, and R. Fitch, "Monte carlo planning for active object classification," *Autonomous Robots*, vol. 42, no. 2, pp. 391–421, 2018.

[9] S. Radmard, D. Meger, J. J. Little, and E. A. Croft, "Resolving occlusion in active visual target search of high-dimensional robotic systems," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 616–629, 2018.

[10] A. Saran, B. Lakic, S. Majumdar, J. Hess, and S. Niekum, "Viewpoint selection for visual failure

Fig. 6: Results of the our optimal viewpoint path estimation. The left column shows the images from the starting point and the right column shows the images from the estimated optimal viewpoint. We wrote the generated guidance sentences next to the paired images.

detection," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[11] G. J. Stein, C. Bradley, and N. Roy, "Learning over subgoals for efficient navigation of structured, unknown environments," in *Conference on Robot Learning (CoRL)*, 2018.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *Neural Information Processing Systems (NIPS)*, 2017.

[13] J. Yang, Z. Ren, M. Xu, X. Chen, D. Crandall, D. Parikh, and D. Batra, "Embodied visual recognition," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[14] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.