Supplementary Material for Learning View Selection for 3D Scenes

Yifan Sun
Institution1
Institution1 address
firstauthor@i1.org

Qixing Huang
Institution2
Institution2 address
firstauthor@i1.org

Dun-Yu Hsiao, Li Guan, Gang Hua
Institution2
First line of institution2 address
secondauthor@i2.org

In this supplementary material, we provide a more detailed and explicit explanation of our dataset used for experimental evaluation. We also provide more statistics of our experimental results and additional visualizations.

1. Dataset Overview

The original input of our method is composed by three different parts, namely 3D models of the real scenes, the areas of interest for each individual scene, and cameras with poses (extrinsic parameters) and predefined intrinsic parameters. In this section, we present the dataset for our experiments and define the hyper-parameters for pre-processing before the camera view prediction module in detail.

1.1. Models

Our dataset consists of CAD models of more than 300 real stores. These CAD models were first artificially built by 3D designers based on the actual layouts of these scenes. These CAD models were then compared against actual 3D reconstructions of these scenes to justify their accuracy. The objects in each model include large objects such as walls, shelves, counters, to small objects such as ovens and lamps. We involve these high-resolution models as a substitute to the unrealistic shapes of reconstructions from multi-view stereo (MVS) methods to narrow the gap between the experimental evaluation and the practical performance and improve the accuracy of our evaluation metrics. For simplicity, in the follow renderings of the coverage task we only keep the geometry of each object, and no texture mapping is included.

Figure 1 provides visualizations of 60 store models rendered from the top view. The sizes of rendered models have been normalized for the convenience of visualization. The unoccupied space including corridors and empty rooms is rendered dark and the occupied space including as walls and shelves is rendered light. The core module in the layout of a store is the aisle, the region of which is indicated by the largest connected dark component (see Figure 2). Shelves are arranged inside these aisles.

An 1:1 model before scaling typically has an aisle area of 20 meters^2 to 60 meters^2 and a height of 3 meters to 3.5 meters. Shelves are 1.5 meters to 2 meters in height and corridors are at least 1m in width. For discretization of the scene model, we choose the $85 \times 85 \times 7$ grid to reduce the space complexity of training and testing of our view prediction network while guaranteeing that the topological structures of the scenes are unchanged.

1.2. Areas of Interest

Each CAD model is associated with a set of areas for detection (or to be covered). These areas can accumulate along the aisles, on shelf sides, or at the entrances, which are in practice the critical areas to be surveilled by cameras in a store. The areas of interest in raw data are marked as coarse bounding boxes in the design, e.g., one bounding box for each shelf or one bounding box for each corridor. To create a general representation for both the occupied areas and the unoccupied, and also to speed up the process of generating the ground truth of coverage from camera position candidates, we discretize each area by a 3D grid into a set of entities for detection (i.e., we sample each vertex of the 3D grid). At each vertex, we assign an entity with zero volume and a normal direction. The normal direction $\boldsymbol{n}$ at position (x_e, y_e, z_e), generated inside bounding box centered at (x_b, y_b, z_b) is defined as $\boldsymbol{n} = (x_e - x_b, y_e - y_b, z_e - z_b)$. Under this representation, we can apply the metric

$$s := \sum_{e \in \mathcal{E}} \max_{1 \le i \le n} \left(w_{ie} \cdot (\cos^{\alpha}(\theta_{ie}) + \delta_{\max}) - \delta_{\max} \right) - \lambda n$$

defined in the main paper to quantify the coverage from each view of the camera.

In our experiments with 1:1 models, we fix the resolution of coverage entity grid to be 20 centimeters, i.e., 20 centimeters between neighboring entities. A non-opaque actor is added to the scene for each entity for detection when rendering with Unreal Engine.

Figure 1. A subset of 60 scenes from our dataset. A scene is typically composed of aisles with shelves, a dinning section with tables and chairs, and some other small rooms. The floor is rendered darker than other objects as they are farther from the view point. Actors for entity detection are removed from the scenes in this figure for a clearer view. The size of each scene in the image is not proportional to its actual size.

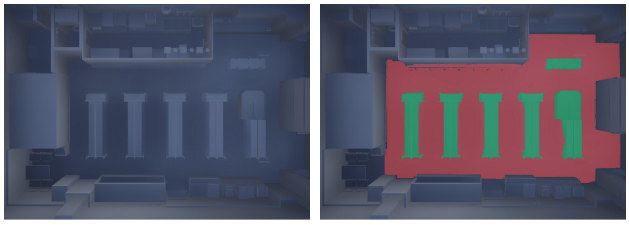

Figure 2. We focus on the coverage of areas of interest inside aisles (the colored part) of the scenes in the dataset. The corridors are colored red and the shelve are colored green.

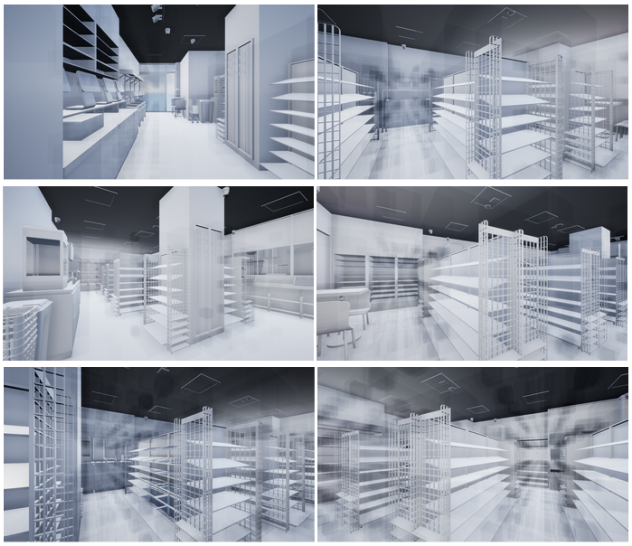

Figure 3. This figure shows the details of a scene model and the result of the rendering process. We consider the set of views from a series of cameras under a single scene. The actors are rendered as black cubes with 10% opacity in this figure.

1.3. Cameras and Visualization

The cameras used for all our experiments are fixed to have a half angle-of-view of 45 degrees. We do not add restrictions to the minimum and maximum visible range since the size of a store model is bounded ($\sim$10 meters) in each dimension. The height of cameras are set to adjust between 100 centimeters and 120 centimeters, which simulates the behavior of a robot with a camera carrier on its top. Despite our experiment setting, both the modules in our method and rendering techniques can be generalized for different types of cameras.

Given a series of extrinsic parameters of cameras, we render specific views captured from these cameras. An example of rendered views under a camera series with a high coverage score is shown in Figure 3. The entities are rendered as black cubes instead of a zero-volume vertex only for visualization purpose and in all our experiments we follow the original definition stated above to compute the coverage rates and scores.

For realistic rendering in qualitative evaluation of coverage, we replace cameras with green angled light sources that share the same intrinsic(angle) and extrinsic parameters. Faces in the model that are lighted by these light sources are covered by the cameras. This visualization technique does not show the coverage of entities precisely, but we still take the area colored by the light sources as an approximation of coverage rate and scores (Figure 4).

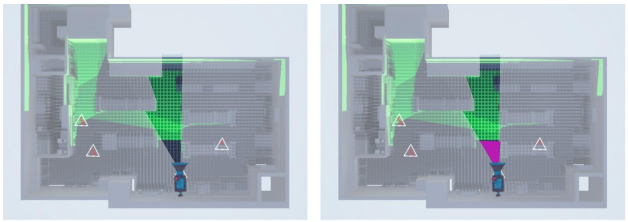

Figure 4. In the visualization of qualitative rendering in the main paper, we use light sources that share the same parameters with cameras to replace them to find the covered faces of the model. When the visible area of one camera is cropped (the dark part of the left figure above), we see that a part of the vision cone that is close to the camera (the pink area in the right figure) fails to get colored. Due to our experimental results the loss from these parts is subtle and we still use this visualization as an approximation of the coverage qualitatively.

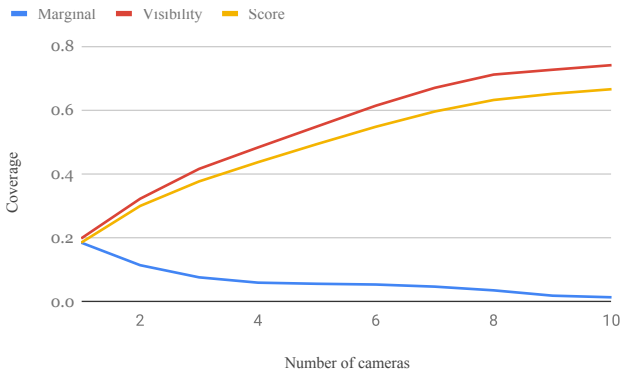

Figure 5. The average coverage ratio with respect to the number of cameras. The ratio from our angular metric(yellow) follows the binary visibility(red) in both global and marginal(blue) coverage.

In Figure 5 we show the increment in coverage ratio when placing more cameras. The marginal coverage drops to less than 5% both in the metric of binary visibility and the angular metric with hyper-parameters. Here we manually set the hyper-parameter $\delta_{\max}$ in the context of the camera intrinsic configuration stated above for better visualization.

2. Ablation study for optimization near optimum

To show the robustness of our method under a proper initialization close to the solution of the Next-Best-View(NBV) method, we present more experimental results

in this section. Note that the NBV method does not compute an "optimum," and we use this word only to emphasize that the NBV method under high-resolution grid is the state-of-the-art method and achieves high coverage score in practice.

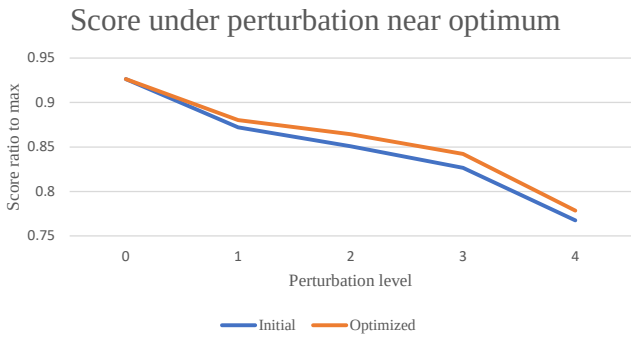

Figure 6. This figure shows the experimental results of robustness testing. The horizontal axis represents the perturbation level. The amplitude of perturbation is proportional to the level, with level 1 corresponding to a uniformly random one within $\pm$20cm, $\pm 10^{\circ}$. The vertical axis represents the relative single-camera score on average. In general, our optimization reduces the loss in coverage score brought by perturbations on camera transformations. Note that even the optimum may not achieve the possible maximum score since some of the entities for coverage may be generated inside objects, and the number of cameras is limited. We do not consider perturbation larger than level 4 as the randomness of initialization becomes too large.

Instead of implementing the NBV method on camera positioning grids with different resolutions, we consider the more general case that camera transformations from the NBV method on fine grids are randomly perturbed to some extent. We apply this strategy of initialization based on two reasons. The first one is that in the camera position discretization of NBV method, there is no rule for selecting the origin and the block size of the grid under different scenes. In fact, a camera sampled from a fine-grid may be inside an object, and it is no longer valid (i.e., we cannot place a camera inside a solid object). A random perturbation can reduce the effect of these incorrect camera placements. The second is that the amplitude of perturbation is continuous and therefore makes the set of camera position candidates for the NBV method continuous as well. We can densely sample the amplitude along with the perturbation for robustness testing of our method. For this test, we fix the number of cameras in a scene.

The result of the robustness test is plotted as in Figure 6, with each perturbation level taking the average of a group of 6 cases. Compared to the ground truth, the approximated score function learned from the network smooths high-frequency noises and helps to optimize the camera transformations to a local maximum of the approximated function.

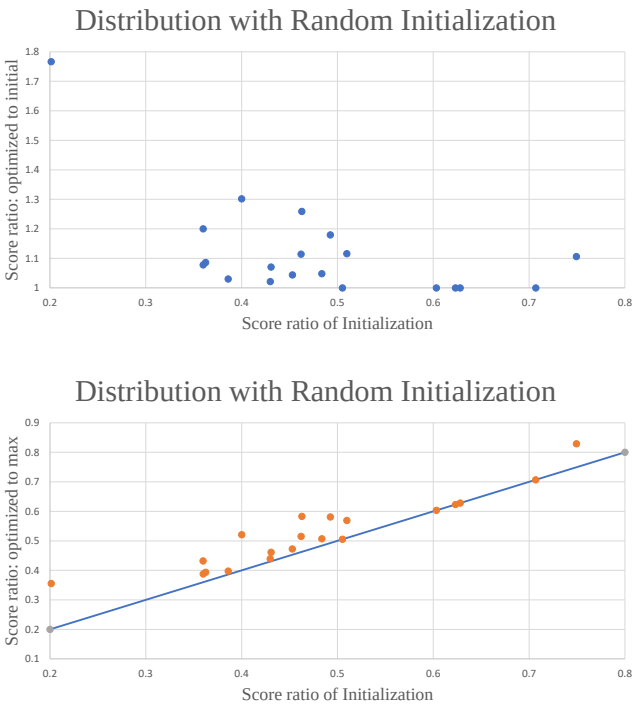

Figure 7. This figure shows the experimental results of the random initialization test. The horizontal axis of both sub-figures represents the ratio between the average score of each camera under random initialization and the possible maximum, bounded by the total number of entities for detection in the scene. The vertical axis shows the ratio between the scores of optimized and initial positioning(possible maximum) above(below). Our optimization method is expected to have better performance for worse initializations.

When the perturbation is large, it is relatively more likely to find a local optimum on the approximated score function, which also scores high in the real case. As the initialization approaches the global optimum, it becomes harder for optimization on an approximate function to precisely find a local optimum on the manifold of the real function within the neighborhood.

3. Additional experimental results for random initialization

In this section, we extend our experiments to random initializations. Although our method outperforms the baseline visually, as is shown in the main paper, we now provide statistics on coverage scores under more test cases. All the scenes are randomly chosen from our test set, and cameras are positioned with random translation and rotation parameters in each scene.

Figure 7 shows the improvement in the score of each camera when our method is applied to pose sets from ran-

dom initialization. When camera poses are randomly generated, the initial score approximately follows a bell-shaped distribution centered around 40% to 50% of the maximum possible score. According to the random initialization test, we can see that even with the smooth approximate function, there are still many local optimums distributed all over the loss surface. As the score of an initialization gets farther away from the global optimal, the performance gain of our optimization increases both in absolute and relative scores.