Pacific Graphics 2019
C. Theobalt, J. Lee, and G. Wetzstein
(Guest Editors)

Volume 38 (2019), Number 7

Imitating Popular Photos to Select Views for an Indoor Scene

Rung-De Su, Zhe-Yo Liao, Li-Chi Chen, Ai-Ling Tung, Yu-Shuen Wang

National Chiao Tung University, Taiwan

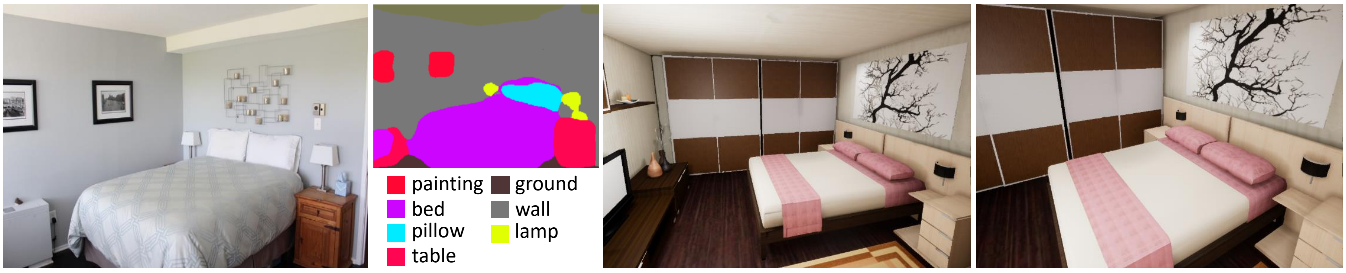

Figure 1: *We select views for a 3D indoor scene by imitating popular photos on the Internet. (Left) A popular photo. (Middle left) We apply the scene parsing method to identify the object of each pixel, and visualize the result in respective colors. (Middle right) The initial view is determined by locating centroids of corresponding objects at the same position. (Right) Our system fine tunes the view to make contours of the corresponding objects similar.*

Abstract

Selecting informative and visually appealing views for 3D indoor scenes is beneficial for the housing, decoration, and entertainment industries. A set of views that exhibit comfort, aesthetics, and functionality of a particular scene can attract customers and facilitate business transactions. However, selecting views for an indoor scene is challenging because the system has to consider not only the need to reveal as much information as possible, but also object arrangements, occlusions, and characteristics. Since there can be many principles utilized to guide the view selection, and various principles to follow under different circumstances, we achieve the goal by imitating popular photos on the Internet. Specifically, we select the view that can optimize the contour similarity of corresponding objects to the photo. Because the selected view can be inadequate if object arrangements in the 3D scene and the photo are different, our system imitates many popular photos and selects a certain number of views. After that, it clusters the selected views and determines the view/cluster centers by the weighted average to finally exhibit the scene. Experimental results demonstrate that the views selected by our method are visually appealing.

CCS Concepts

• ***Computing methodologies*** → *Collision detection;* • ***Hardware*** → *Sensors and actuators; PCB design and layout;*

1. Introduction

The preview to an indoor scene is critical for the housing, decoration, and entertainment industries because images are the most convenient media to distribute and convey information. A set of views that express comfort, aesthetics, and functionality of an indoor scene can attract customers to view details of the scene and facilitate business transactions.

Methods for selecting the best view for a 3D object have been thoroughly investigated in the field of computer graphics. However, to the best of our knowledge, no existing method selects a set of optimal views for an indoor scene that is composed of several 3D objects. Selecting views for a 3D indoor scene and a single 3D model are different in two aspects. First, several objects can exist in a scene. The selected views should help viewers reconstruct the object arrangements in 3D. Second, objects behind the camera are invisible. Finding a set of views that are dissimilar, visually appealing, and able to highlight important characteristics of a scene constitutes a demanding task. In addition, principles for selecting views can be different according to particular purposes and functionalities of a scene. For example, professional photographers would usually consider showing large space and object arrangements when taking photos of a living room. However, when taking photos of a bedroom, the priority is more likely to be making customers feel relaxed and comfortable. Hence, low-level features, such as entropy and saliency, are insufficient to express semantics. Selecting views based on such features will potentially fail to fulfill purposes of exhibiting different scenes.

Since defining rules to select optimal views for a 3D indoor scene is challenging, we achieve the goal by imitating popular photos on the Internet. To imitate a specific photo, we first apply the scene parsing method [CZP*18] to label the object of each pixel, and then compute the object contours. Because 3D virtual objects are often manually created, we assume that the object labels are known in advance. The goal is to select a view for the 3D indoor scene by optimizing the similarity of corresponding object contours between the view and the photo. Specifically, the contour errors of corresponding objects between the selected view and the imitated photo are measured. A set of camera parameters, such as position, view direction, and field of view, which can minimize the error is then determined.

The selected view is not guaranteed similar to the imitated photo because object arrangements in the 3D scene and the photo (real world) can be different. These views are invalid and often contain significant contour errors. An intuitive way to prevent such a problem is to find photos, in which the object arrangements are highly similar to those in the 3D scene, before the imitating process. However, measuring the similarity of object arrangements between a photo and a 3D scene is not easy because either scene reconstruction from a photo or view selection for a 3D scene should proceed. Therefore, in this study, we select views for the 3D scene by imitating many popular photos. The idea is based on voting and the assumption that visually appealing yet dissimilar views are not many. In other words, popular photos, in which object arrangements are similar to those in the 3D scene, would guide our system to select views from a small number of normal distributions. Therefore, we cluster the many selected views and choose only the cluster centers, denoted as representative views, to exhibit the 3D scene. Note that the cluster center is a weighted average of views. The weight of a view is determined based on its contour error. In this way, popular photos, in which the object arrangements are dissimilar to the 3D scene, have little influence on the final result.

We demonstrate the feasibility of our method on a variety of indoor scenes, such as living rooms, bedrooms, bathrooms, and kitchens. The selected views shown in Figures 1, 2, and 4 are visually appealing and can exhibit important characteristics of the scenes. We also conducted a user study with 60 participants to evaluate whether our objective ranking and the subjective ranking are consistent. The results demonstrate the feasibility of our system.

2. Related Works

Best view selection. Finding the best view of a 3D model is beneficial for many applications, such as thumbnail generation [MS09], object recognition [DDA*04], and image-based modeling [VFSH03]. The goal of this view is to reveal as much information about an object as possible so that people can distinguish the object from others. Blanz et al. [BTB99] introduced four attributes to determine a so-called *canonical view*: goodness for recognition, familiarity, functionality, and aesthetic. Psychophysical experiments have demonstrated that a canonical view often corresponds to the classical *three-quarter view* [BTB99, Pal81].

A large body of literature has focused on selecting the best views for 3D models. Many of these studies are based on low-level features, such as entropy [VFSH01, VFSL02] and saliency [LVJ05, SLT13, TFTN05]. The methods aim to identify a view that can reveal as many features as possible. However, since a certain type of low-level feature is not sufficient to capture the semantics of a 3D model, Polonsky et al. [PPB*05] presented a visual descriptor that comprises several features, such as surface area entropy, visibility ratio, curvature entropy, and surface entropy, and maximized the descriptor to obtain the best view. In addition, Kucerova et al. [KVC13] implemented three approaches (i.e., based on geometry, entropy, and visual attention) to determine the best views of a set of 3D models. They then compared the results and conducted a user study to identify which method outperforms the others in which kinds of 3D models.

View selection methods that consider semantics are diverse. Mortara et al. [MS09] partitioned the model into semantic components, and then attempted to obtain the view that can reveal most of the components. Denton et al. [DDA*04] pointed out that the best view is a view that is dissimilar to others. Their presented method removes equivalent views in the view space to achieve the goal. Laga [Lag10] assumed that the best views of a 3D model are views that allow discriminating the model from other models. He formulated the best view selection problem as a classification and feature selection problem. Vázquez [Váz09] computed the stability of a view according to depth maps, without prior knowledge on the geometry or orientation of an object, to select best views. Liu et al. [LZH12] applied Internet images to vote for the best view. The object's shape and saliency are utilized to determine the view that is similar to an Internet image. Different from the previous methods, our system selects views for an indoor scene rather than an individual object. Problems, such as occlusions, aesthetics, and different object arrangements between the 3D scene and the photo, occur. In addition, our system has to select several views rather than a single view to exhibit an indoor scene. A system that is similar to ours was presented by Genova et al. [GSCF17], which was built to generate training sets for the computer vision tasks. To reduce the gap between synthesized and real samples, the system selects views that are similar to real photos by minimizing the spatial distribution of each semantic object category.

3. View Selection Algorithm

We present a fully automatic system to select views of an indoor scene by imitating popular photos on the Internet. To achieve the aim, the first goal is to obtain object arrangements of each photo. We apply the scene parsing technique [CZP*18] to identify the object of each pixel, and then determine object contours by tracing the segmentation boundaries. On the other hand, we assume that each object, such as a chair, sofa or bed, in a 3D scene is known in advance because 3D scenes are in general created manually or by an optimal object arrangement method [AON05, YYT*11]. To obtain the object label of each pixel in the rendered views, we first colorize each 3D object by a unique color according to its label and then render the scene without illumination. Under this circumstance, the label of each pixel in the view can be easily identified by color examination. After that, similarly, we trace segmentation boundaries to extract object contours in each view.

Popular photos can be retrieved from many web pages, such as Google and Instagram. Users can query a keyword on the page and then download the photos for our system to imitate. In our implementation, we retrieve photos from Flickr because the Flickr API [†] provides the number of views of each photo. This number, to a certain degree, reflects how appealing a photo is. The first 200 most-viewed photos on the page, which were clicked to watch from thousands to 0.1 million times, were used in this study.

Our system selects as many views as the photos it imitates. Because object arrangements in the scene and the photo can be different, a part of the selected views would be dissimilar to the imitated photos. These views are invalid and often contain significant contour errors. On the other hand, since visually appealing yet dissimilar views in a 3D scene are not many, popular photos, in which the object arrangements are similar to those in the scene, would guide our system to select views from a small number of normal distributions. Accordingly, we apply the mean-shift algorithm to cluster the selected views. The center view in each cluster is computed by weighted average, in which the weight of a view is determined based on the contour error.

3.1. Imitating the View from a Photo

Given a 3D scene $\mathbf{S}_{3D}$, our goal is to find a view $\mathbf{S}_{2D}$ that is visually similar to a popular photo $\mathbf{R}$. Let $\{s,t\} \in \mathcal{L}$ be the pair of corresponding objects, and let $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, ...\}$ be the object contour, where $\mathbf{p}_i \in \mathcal{R}^2$ is a contour point. We compute the camera position (x,y,z), view direction (θ,ϕ), and field of view λ, that can minimize the deviation of corresponding object contours $\mathbf{P}^s$ and $\mathbf{P}^t$. The view direction is defined in the spherical coordinate, and the up vector of the camera is upward. Specifically, we compute the view by minimizing

$$E(\mathbf{v}) = \frac{1}{\alpha} \sum_{\{s,t\}\in\mathcal{L}} \frac{1}{\beta|\Psi_{s,t}|} \sum_{\{i,j\}\in\Psi_{s,t}} \left|\mathbf{p}_i^s - \mathbf{p}_j^t\right| \tag{1}$$

where $\mathbf{v} = (x,y,z,\theta,\phi,\lambda)$, $0 < \alpha \leq 1$ is the ratio of the summed area of the reference objects, Ψ is the set of corresponding points, and β is the diagonal length of an image. It is worth noting that high similarity between the selected view and the imitated photo implies a large corresponding object area. Accordingly, we divide the mean contour error by a scalar α, which is expected to be a large value.

Minimizing Equation 1 is challenging because objects in a 3D scene and a photo can be different, and the correspondence of objects is unknown. Therefore, we adopt a heuristic method that solves the camera parameters in three steps. First, we find h pairs of corresponding objects between the 3D scene and the photo ($h = 3$ in our implementation). The pairs can be arbitrary, and all possible combinations will be tested. For each pair set, we compute an initial view of the scene by aligning the centroids of corresponding objects. Then, we refine the view by minimizing contour errors (Equation 1). While there will be many views selected according to different object combinations, we keep only one view that has the lowest energy, which implies that the view is the most similar to the photo in object arrangements. Finally, we test the remaining corresponding object pairs one by one and check if adding the pair to the pair set can further reduce the energy. The system stops when no pairs of corresponding objects can be added.

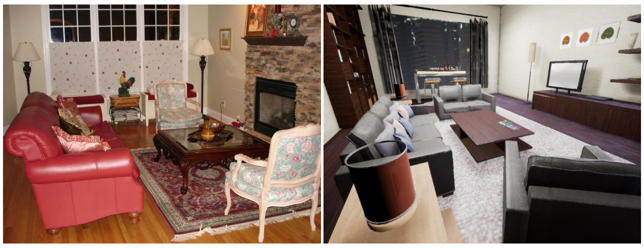

Figure 2: *(Left) A popular photo on the Internet. (Right) Our system imitates the photo to select views. Notice that the object arrangements in the 3D scene and the photo can be partially similar. The chairs and the TV are not considered when the system selects the view. Three pairs of corresponding objects that are arranged similarly are allowed.*

Figure 2 shows a selected view and the imitated photo. We point out that object arrangements in the view and the photo can be partially similar because the minimum number of corresponding objects is three. In other words, although minimizing Equation 1 expects the corresponding areas in the view and the photo to be large, our system does not attempt to find a view that looks the same to the photo, which is impossible.

3.1.1. Initial View

Our system determines the view by starting from a set of corresponding objects between the 3D scene and the photo. However, although the corresponding objects are set as known variables, Equation 1 is still nonlinear because the correspondence of contour points is unknown. To prevent the system from rapidly falling into a local minimum, we first compute an initial view by aligning the objects' centroids. Then, the view is iteratively updated to minimize the deviation of corresponding contours. Since defining a unique view demands six unknown variables, this initial view is determined by three pairs of corresponding objects. Let $\mathbf{c}^s$ and $\mathbf{c}^t$ be the centroids of corresponding objects in the 3D scene $\mathbf{S}_{3D}$ and the photo $\mathbf{R}$, respectively. To obtain the initial view, we compute the camera parameters $\mathbf{v} = \{x,y,z,\theta,\phi,\lambda\}$ that can minimize the term

$$E_c(\mathbf{v}) = \sum_{i=1}^{h} \left|\mathbf{M}\mathbf{c}_i^s - \mathbf{c}_i^t\right| \tag{2}$$

where $h = 3$ is the number of corresponding objects, and $\mathbf{M}$ is the projection matrix determined from the unknown camera parameters. To prevent target objects from being occluded, we further consider the term

$$E_o(\mathbf{v}) = \sum_i x_i, \quad \text{where } x_i = \begin{cases} \varepsilon & \text{if } \mathbf{c}_i^s \text{ is occluded} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

[†] https://www.flickr.com/services/api/

when computing the parameters. We set ϵ to a large value ($\epsilon = 100$ in our experiments) because object occlusion is not allowed. To accelerate the occlusion test, we applied the bounding volume hierarchy technique [Eri04] to examine the occlusion of centroids.

We minimize the objective function $E_c + E_o$ by applying the downhill simplex algorithm to obtain the initial view. Since there are six unknown variables, the initial simplex is defined by seven randomly generated vertices. Each vertex corresponds to a solution. Because target objects should be in front of the camera, we initialize the simplex vertices based on the positions of objects rather than pure randomness. Specifically, the camera positions (x, y, z) are constrained to be away from the centroids within a reasonable distance; the view directions (θ, ϕ) should point at the center of objects, and the field of view λ is within 40-80 degrees. In each step of the optimization, we transform the temporary camera parameters to the projection matrix for evaluating the view quality (Equation 1) and then update the simplex vertices. The process repeats until its shape is shrunk to a point, which is the final solution. Because of the random strategy used in the downhill simplex method, we repeat the minimization process ten times and choose the optimum.

3.1.2. View Refinement

Corresponding objects in the initial view $\mathbf{S}_{2D}$ and the photo R would appear at similar positions. However, the orientations of these objects may be different (see Figure 1). As stated previously, $\mathbf{P}^s$ and $\mathbf{P}^r$ are the corresponding contours in $\mathbf{S}_{2D}$ and R, respectively. Our goal is to fine tune the view by making contours $\mathbf{P}^s$ and $\mathbf{P}^r$ as similar as possible. In other words, we assume that $\mathcal{L}$ is known, which is obtained from the previous step, and minimize Equation 1. Because the unknown view and the correspondence of contour points are correlated, the objective function is minimized by applying the iterative closest points (ICP) method [Zha94], i.e., these two unknown variables are alternatively and iteratively updated. At one step, we compute the set $\Psi_{s,t}$ based on the distance of points. The pair $\{i, j\}$ is added to $\Psi_{s,t}$ if either $\mathbf{p}_i^s$ is closest to $\mathbf{p}_j^r$ or $\mathbf{p}_j^r$ is closest to $\mathbf{p}_i^s$. At the other step, similar to the way of minimizing Equation 2, we apply the downhill simplex method to update the view and minimize the distance of corresponding contour points. The process repeats until the system converges.

We add the fourth corresponding object to the system and check whether Equation 1 can be further reduced because the goal is to consider as many objects in the popular photo as possible. The view is then updated by the above-mentioned algorithm. Once the fourth corresponding object is successfully added, we consider the fifth, sixth, etc., until the objective function stops decreasing.

3.2. Representative Views

Our system imitates a certain number of popular photos to select views for an indoor scene. If object arrangements in the photo and the 3D scene are similar, the photo guides our system to select a valid view. However, if this is not the case, the selected view is invalid and visually dissimilar to the photo. Under this circumstance, the contour error determined by Equation 1 is large. Considering that visually appealing yet dissimilar views of an indoor scene are not many, the valid views will be selected from a small number of normal distributions. Accordingly, we apply the mean-shift algorithm to cluster views. The cluster centers, denoted as representative views, will be selected to exhibit the 3D scene. The cluster center is a weighted average of views, in which the weight of each view is determined by its contour error. This strategy can minimize negative effects caused by the invalid views.

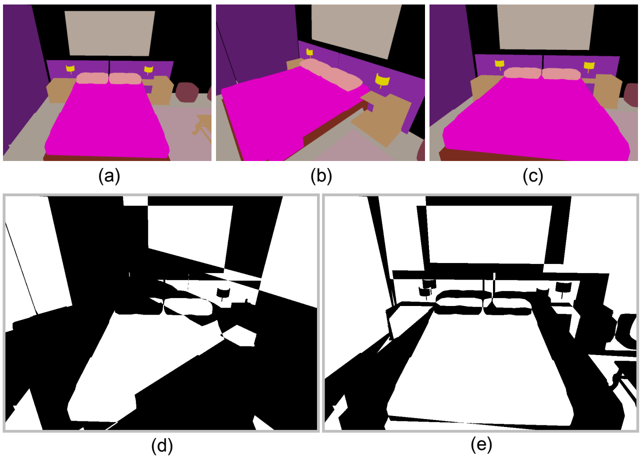

Figure 3: *We obtain visual similarity of two views by comparing object distributions. (a - c) Three views selected for a 3D scene, in which the object of each pixel is represented by color. (d) and (e) The comparisons of views (a) (b) and views (a) (c). Black and white indicate that objects at an identical position are different and the same, respectively. We count the number of black pixels to determine the distance of two views.*

Generating view clusters can be achieved by merging views if their camera parameters are numerically alike. In our implementation, we normalize each parameter to the range of [0, 1] before computing the camera distance D_c. Considering that distinct camera parameters do not guarantee that the views are visually different, we additionally consider object distributions between views and measure their visual distance D_v. To achieve this goal, we colorize predefined 3D objects by a unique color and render the views without considering illumination, as illustrated in Figure 3 (a)-(c). Since colors represent object labels, we measure the visual distance of two views by counting the number of corresponding pixels (i.e., located at the same position) that are in different colors (Figure 3 (d)-(e)). Note that the pixels of floor, wall, or ceiling are not considered during the computation of D_v. The distance is then normalized by the number of valid pixels of the two views. Then, we sum the distances to determine view similarity of i and j by using the term

$$D_c(i,j) + D_v(i,j), \text{ where } D_c(i,j) = \left|\mathbf{v}_i - \mathbf{v}_j\right|. \tag{4}$$

We compute the representative view of each cluster by weighted average. The weight of each selected view is determined based on its contour error. Specifically, let i be the view index and $E(\mathbf{v})$ be the contour error (Equation 1), we determine the weight by

$$w_i = e^{\frac{-(E(\mathbf{v}_i)-\mu)^2}{2\sigma^2}}, \tag{5}$$

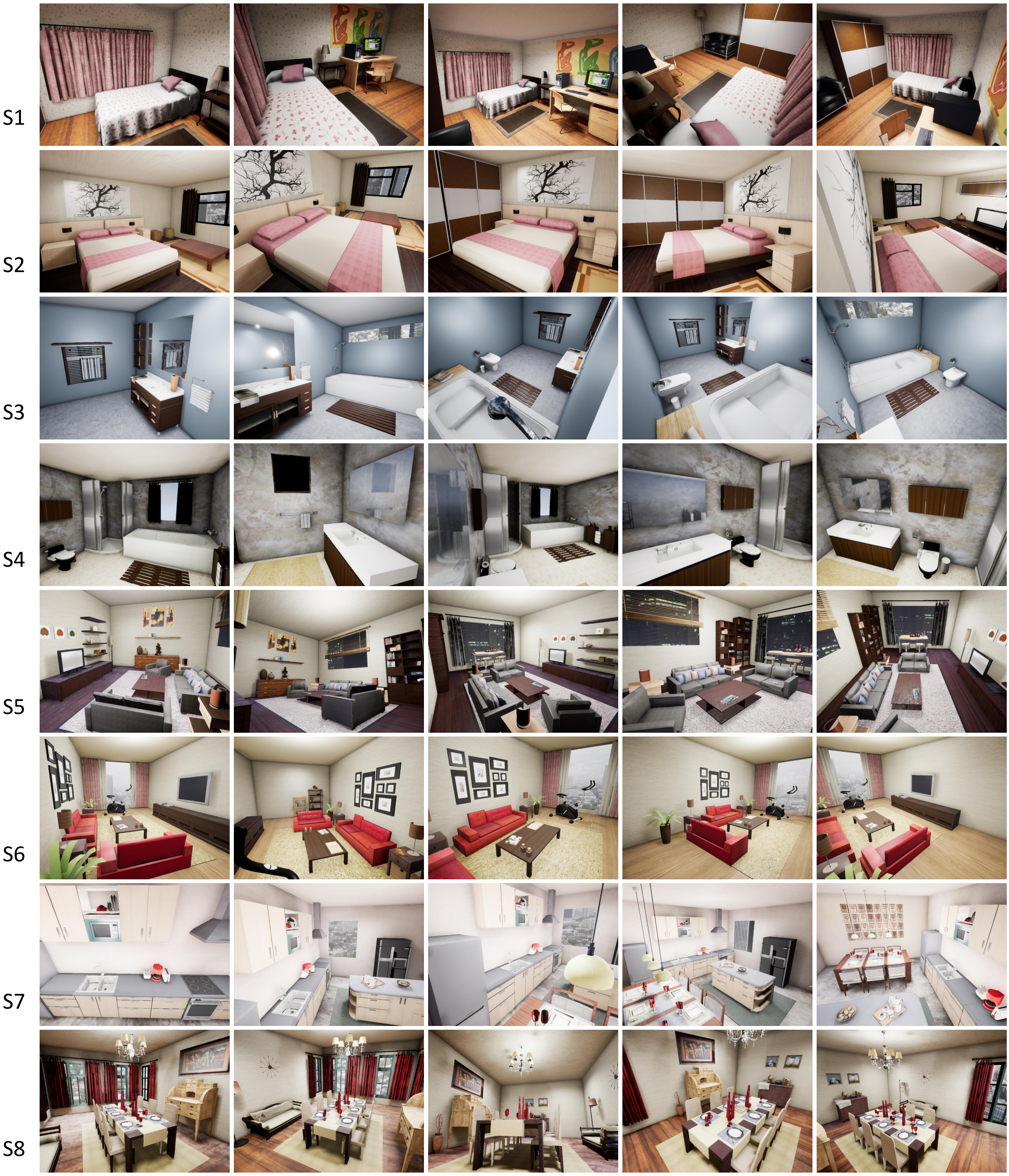

Figure 4: *The five top-ranked views of indoor scenes selected by our system. S1-S8 are the scene indexes.*

where μ and σ are the minimum and the standard deviation of $E(\mathbf{v})$, respectively, among the views.

3.3. View Ranking

We rank the representative view mainly based on the averaged contour errors ($E(\mathbf{v})$ in Equation 1) of the corresponding cluster. A small error implies high similarity to the imitated photos. Let i be the cluster index, and Φ_i be the set of views in cluster i. We score the representative view by

$$G_i = \frac{1}{|\Phi_i|-1} \times \frac{\sum_{j\in\Phi_i} w_j \cdot E(\mathbf{v}_j)}{\sum_{j\in\Phi_i} w_j}. \qquad (6)$$

Note that we divide the mean contour error by $|\Phi_i|-1$ because a very small cluster is potentially an outlier. This strategy also encourages the system to rank the cluster with more views anterior because the view is more popular.

4. Results and Discussion

We have implemented our algorithm and run the program on a desktop PC with Intel Core i7 3.0 GHz CPU. A variety of indoor scenes, such as bedrooms, living rooms, kitchens, and bathrooms, were tested and the results are shown in Figures 1, 2, and 4. As can be seen, the views selected by our system appear reasonable and visually appealing. No object is exceptionally close to the camera; no views are simply composed of a wall, ceiling, or floor; and there are no views in which objects are upside down. Since the goal of our system is to imitate how photographers take pictures rather than maximizing the view information, the selected views are not always informative. These results are reasonable because people will be willing to pay attention to show the functionality of furniture when several views are selected to exhibit a scene.

The computation cost of our system is proportional to the number of photos it considers. In our current unoptimized implementation, the system takes about 20 minutes to hours to select views for a scene. The time is mainly caused by finding the best correspondence of objects in the 3D scene and the photo. Specifically, although the system computes only one view for the scene when it imitates a photo, it has to test many object correspondences. In our current implementation, objects in different categories do not correspond, and objects on the photo that have too small areas (less than 1%) are not considered because of the small value of α in the objective function (Equation 1). To improve the system performance, in the future, we plan to leverage the GPU acceleration and skip the photos in which the object arrangements are very different from that of the 3D scene.

4.1. Evaluation

Our system selects a view similar to the popular photo by applying a heuristic method. To understand whether this method works well in view selection, we evaluate the system as follows.

4.1.1. Stability of downhill simplex optimizer

The initial vertex positions used in the downhill simplex method are randomly generated. In other words, the system does not guarantee that the obtained solutions are identical when it converges, although the objective function is untouched. However, we observed that the results are quite similar in the experiments. To verify the observation, we repeated the optimization process 100 times, and then visualized the results. That is, we computed 100 views for a 3D scene by imitating the same photo, where the corresponding objects between the scene and the photo are predefined. After that, we determined the central view from the results and compute the camera distance between this central view and each individual view. The distance is then normalized to the range among [0, 1]. Figure 5 left shows the histogram of distance distribution. We also show the views that are closest and farthest from the central view in Figure 5 right. Notice that the views are visually similar.

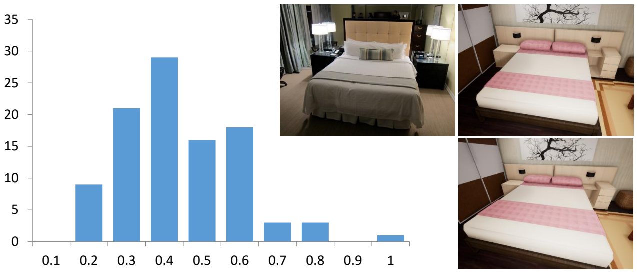

Figure 5: *We tested the stability of the downhill simplex method by repeating the experiment 100 times under the same condition, except the initial shape of simplex. The photo used in the experiment is shown in the middle. (Left) We computed the distance from the central view to each individual view and showed the histogram of distance distribution. The horizontal and the vertical coordinates indicate the distance and the number of views, respectively. (Right) The views that are closest (top) and farthest (bottom) to the central view. As indicated, our system is stable to the initial shape of simplex because the closest and the farthest views are visually similar.*

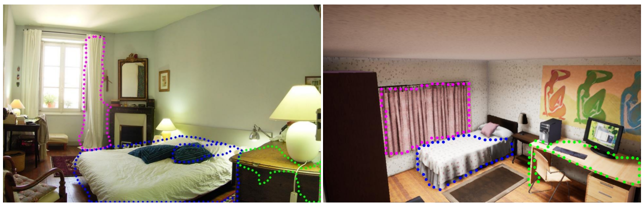

Figure 6: *Left and right are the photo and the selected view. The overall object arrangements in the scenes are similar. However, the contours of the corresponding objects are not (indicated by the dotted curves). Under this circumstance, our system degenerates to retain the object positions because it solves the camera position, view direction, and the view angle to minimize the objective function.*

4.1.2. Different shapes and arrangements of scene objects

Corresponding objects in the 3D scene and the photo can be different in shape. Under this circumstance, the selected view holds a large contour error, although the object arrangements are similar. Figure 6 shows an example. Similarly, the object arrangements in the 3D scene and the photo can be different. While there are fewer

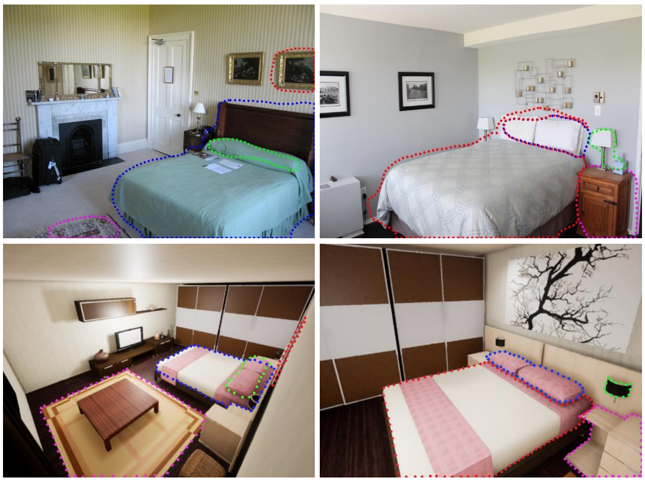

Figure 7: *(Top) Popular photos. (Bottom) The selected views. While the selected view is dissimilar to the corresponding photo, its contour error is large and will be assigned a small weight during the computation of representative views. The contour errors of the left and right examples are 0.99 and 0.13, and the assigned weights are 0.03 and 0.90, respectively.*

than three pairs of corresponding objects between the scene and the photo, we simply neglect the photo because computing the initial view necessitates at least three pairs of centroids. In the case that there are enough corresponding objects but the objects are in different arrangements, the selected view can be dissimilar to the photo. These selected views hold a large contour error as well.

Our system considers many popular photos to ease the problems caused by different object shapes and object arrangements. Specifically, it selects many views for an indoor scene, clusters the views by using the mean-shift algorithm and then determines the representative view of each cluster by the weighted average. Since the views with large contour errors will be assigned a small weight, they have little influence on determining the representative view. Figure 7 shows the selected views, the corresponding photos, the contour errors, and the weights of the views. Therefore, although the selected views are not always satisfactory, in our experiment, the representative views are adequate because they must be similar to a certain number of popular photos.

4.2. Aspect Ratio of a View

The most common aspect ratios used today in photography are 4:3 and 16:9. The vertical and horizontal coordinates can be switched. However, we set the view aspect ratio to 4:3 in our current system because it is the majority of the popular photos. Since our system is example based, it can be easily adapted to select views of other aspect ratios, by imitating a different set of photos.

4.3. Limitations

The selected view can be dissimilar to the photo because of several reasons, such as different shapes and arrangements of corresponding objects, and imperfect scene parsing results (Figure 8). Besides, our system applies a heuristic method to minimize contour errors for view selection. The searching space is large when there are many corresponding objects, and the selected views are not guaranteed optimal. These views may look awkward. To ease the problem, our system imitates a large number of photos and then determines the representative view by the weighted average. However, the problem would appear if the number of imitated photos is small because a smaller view cluster is less reliable. We believe that imitating more photos can improve the quality of the results, with the price of more computational cost. Finally, our system does not *learn* how to select views for indoor scenes, but *imitate* popular photos on the Internet to achieve the goal. It has to consider many photos and takes lots of computation when selecting views for each scene.

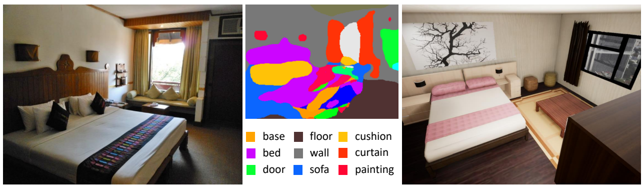

Figure 8: *(Left and middle) The scene parsing method misclassifies a large part of the bed as sofa. (Right) The selected view is dissimilar to the photo.*

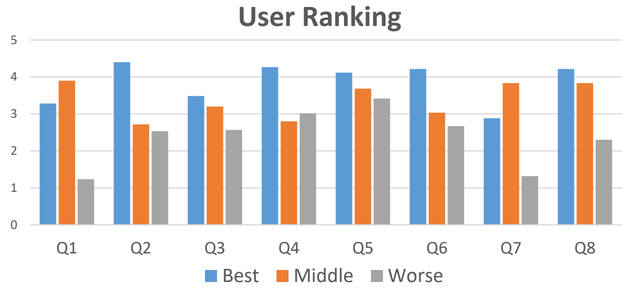

Figure 9: *Subjective scores of the views that are rated the best, middle, and the worse by our system. The order of the scene is the same to the order in Figure 4.*

5. User Study

Our system ranks the view based on the popularity and degree of similarity to the imitated photos. To evaluate whether the computed and subjective rankings are consistent, we conducted a user study and asked the participants to rate the views selected by our system. Specifically, we created a questionnaire with eight questions and posted the questionnaire on the Internet. In each question, three views, which were ranked at the first, middle and the last positions, were listed. The order of questions was randomly generated. The participants were asked to rate the photos subjectively according to their preferences. The best to the worst views were rated by 5 to 1, respectively. To control the experiment, we instructed the participant on the first page of the questionnaire that a view with a high score should be attractive and can motivate people to explore the scene. In the end, we got the answers from 60 participants.

Figure 9 shows the results. In each view set (question), we listed the bar charts from left to right according to the rank given by our system to facilitate interpretation. Ideally, the heights of the bar charts should be monotonously decreasing from left to right because a high quality view deserves a high score. As can be seen, our ranking algorithm was able to reflect the participants' preference. One exception was in S1 (Figure 4). The participants gave the top-ranked view the second highest score. The variation was small since the top-rank view was visually appealing as well. The top-ranked view in S7 has the same problem. The participants disliked the top-ranked view by our system because the kitchen counter was too close to the camera. However, we found that many photos in the database that were similar to this view. Perhaps the reason was diverse preferences of ordinary people in different regions and ages.

6. Conclusions

We have presented a system to select views for indoor scenes by imitating popular photos on the Internet. The system achieves the goal by minimizing the contour deviation of corresponding objects between the selected view and the photo. Considering that object arrangements in the scene and the photo can be different, it imitates a certain number of photos, clusters the selected views, and then determines the representative view of each cluster by weighted average. Although popular photos are not always high-quality, and object arrangements in the photos may be dissimilar to those of the 3D scene, this automatic process allows the system to select views for various scenes as long as there are photos to imitate. Currently, we refer to the number of views as quality. In the future, we will consider other attributes, such as the number of likes, downloads, and reshares. It is also possible to apply a machine learning technique to determine the quality of the photos.

Our current system is designed to select views for 3D virtual indoor scenes, in which objects in the scenes are well-defined. Considering that 3D scanners are becoming increasingly inexpensive, and scene-understanding techniques have been improved by advancements in deep neural networks, we plan to present a system that can take photos for real scenes in future. Specifically, 3D point clouds of a scene can be scanned by a drone, and then the categories of objects in the scene are identified. In this way, we extend the view selection system for virtual scenes to an automatic photo taking system in reality. In addition, taking visually appealing photos should consider not only the view, but also lighting conditions. We plan to study light settings in popular photos and incorporate the knowledge in the future system.

Acknowledgements

We thank anonymous reviewers for their insightful comments and suggestions. We are also grateful to several Flickr users (shawn.cornelius, Marc Lacoste, Christopher Porter, h-squared) who let us use their photos in the paper, and all the participants who joined the user study. This work is partially supported by the Ministry of Science and Technology, Taiwan, under Grant No. 105-2221-E-009 -135 -MY3 and 107-2221-E-009 -131 -MY3.

References

[AON05] Akazawa Y., Okada Y., Niijima K.: Automatic 3d scene generation based on contact constraints. In *Proc. Conf. on Computer Graphics and Artificial Intelligence* (2005), pp. 593–598. 2

[BTB99] Blanz V., Tarr M. J., Bülthoff H. H.: What object attributes determine canonical views? *Perception 28*, 5 (1999), 575–599. 2

[CZP*18] Chen L.-C., Zhu Y., Papandreou G., Schroff F., Adam H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV* (2018). 2

[DDA*04] Denton T., Demirci M. F., Abrahamson J., Shokoufandeh A., Dickinson S.: Selecting canonical views for view-based 3-d object recognition. In *Proceedings of the International Conference on Pattern Recognition* (2004), vol. 2, pp. 273–276. 2

[Eri04] Ericson C.: *Real-Time Collision Detection*. CRC Press, Inc., Boca Raton, FL, USA, 2004. 4

[GSCF17] Genova K., Savva M., Chang A. X., Funkhouser T.: Learning where to look: Data-driven viewpoint set selection for 3d scenes. *arXiv preprint arXiv:1704.02393* (2017). 2

[KVC13] Kucerova J., Varhanikova I., Cernekova Z.: Best view methods suitability for different types of objects. In *Proceedings of the Spring Conference on Computer Graphics* (2013), pp. 55–61. 2

[Lag10] Laga H.: Semantics-driven approach for automatic selection of best views of 3d shapes. In *Proceedings of the 3rd Eurographics conference on 3D Object Retrieval* (2010), pp. 15–22. 2

[LVJ05] Lee C. H., Varshney A., Jacobs D. W.: Mesh saliency. 659–666. 2

[LZH12] Liu H., Zhang L., Huang H.: Web-image driven best views of 3d shapes. *The Visual Computer 28*, 3 (2012), 279–287. 2

[MS09] Mortara M., Spagnuolo M.: Semantics-driven best view of 3d shapes. *Computers & Graphics 33*, 3 (2009), 280–290. 2

[Pal81] Palmer S.: Canonical perspective and the perception of objects. *Attention and performance* (1981), 135–151. 2

[PPB*05] Polonsky O., Patané G., Biasotti S., Gotsman C., Spagnuolo M.: What's in an image? *The Visual Computer 21*, 8 (2005), 840–847. 2

[SLT13] Shtrom E., Leifman G., Tal A.: Saliency detection in large point sets. In *Proceedings of the IEEE International Conference on Computer Vision* (2013), pp. 3591–3598. 2

[TFTN05] Takahashi S., Fujishiro I., Takeshima Y., Nishita T.: A feature-driven approach to locating optimal viewpoints for volume visualization. In *Visualization, 2005. VIS 05. IEEE* (2005), pp. 495–502. 2

[Váz09] Vázquez P.-P.: Automatic view selection through depth-based view stability analysis. *The Visual Computer 25*, 5-7 (2009), 441–449. 2

[VFSH01] Vázquez P.-P., Feixas M., Sbert M., Heidrich W.: Viewpoint selection using viewpoint entropy. In *VMV* (2001), vol. 1, pp. 273–280. 2

[VFSH03] Vázquez P.-P., Feixas M., Sbert M., Heidrich W.: Automatic view selection using viewpoint entropy and its application to image-based modelling. In *Computer Graphics Forum* (2003), vol. 22, pp. 689–700. 2

[VFSL02] Vázquez P.-P., Feixas M., Sbert M., Llobet A.: Viewpoint entropy: a new tool for obtaining good views of molecules. In *ACM International Conference Proceeding Series* (2002), vol. 22, pp. 183–188. 2

[YYT*11] Yu L.-F., Yeung S.-K., Tang C.-K., Terzopoulos D., Chan T. F., Osher S. J.: Make it home: automatic optimization of furniture arrangement. 86. 2

[Zha94] Zhang Z.: Iterative point matching for registration of free-form curves and surfaces. *International journal of computer vision 13*, 2 (1994), 119–152. 4