SceneViewer: Automating Residential Photography in Virtual Environments

Shao-Kui Zhang, Hou Tam, Yi-Xiao Li, Tai-Jiang Mu, and Song-Hai Zhang, *Member, IEEE*

Abstract—Selecting views is one of the most common but overlooked procedures in topics related to 3D scenes. Typically, existing applications and researchers manually select views through a trial-and-error process or "preset" a direction, such as the top-down views. For example, literature for scene synthesis requires views for visualizing scenes. Research on panorama and VR also require initial placements for cameras, etc. This article presents SceneViewer, an integrated system for automatic view selections. Our system is achieved by applying rules of interior photography, which guides potential views and seeks better views. Through experiments and applications, we show the potentiality and novelty of the proposed method.

Index Terms—Interior photography, view selection, 3D interior scene

1 Introduction

Research on 3D scenes has advanced over the last decades, where the topics spread over a wide range, including scene understanding [1], [2], [3], scene synthesis [4], [5], [6], style compatibility [7], scene detailing [8], interaction [9], [10], industrial applications[1], etc. Among all the topics around 3D scenes, selecting a view to visualize or render a scene should be the most fundamental step since we can not perceive scenes intuitively, given only text-based configurations. However, view selection is either ignored by setting an empirical fixed direction or conducted manually in the existing literature.

First, for 3D scene synthesis [4], it is essential to render generated scenes for evaluation. However, since no rules of 3D scene viewing are considered, current state-of-the-art techniques depend on fixed predefined views, e.g., top-down views [11], [12], [13], [14]. Consequently, the evaluations may result in perceptual biases due to the selections of views, for predefined or fixed views are not always suitable for every scene. Recent online businesses on estates would like to show the best views of their properties. Their businesses mainly depend on exhibiting rendered photos taken from the floorplans to potential buyers with an excellent first impression, so selecting the views accurately and appealingly saves time for marketing and gives the customers a better understanding of the rooms. Handa et al. [15] generate datasets for computer vision by using 3D scenes through different views, demonstrating that good choices of views yield informative datasets. The rendering focuses on developing as realistic an image as possible [16], but we argue that selecting positions and directions also matters considering the camera.

Nevertheless, automatic view selection in 3D scenes is not well explored previously. Existing literature on view selections is still designated for 3D meshes [17]. It lacks semantic channels such as layouts of objects and does not consider human-centric constraints such as connections of rooms and visually pleasing qualities. Furthermore, the hypothesis space of possible views for a mesh is often distributed on a regular geometry such as a sphere or regular polyhedron [17], [18]. However, potential views for 3D scenes lie in a more arbitrary and irregular space, i.e., a view could be positioned anywhere among any set of objects within a scene.

In this paper, we propose SceneViewer for automatic view selections in 3D scenes. Our method automatically generates appealing perspective views representing the underlying 3D scene, given a typical floorplan with multiple rooms, as shown in Fig. 1. This is achieved by adopting the interior photography rules, where "probe views" are generated first instead of being exhausted in the entire 3D space. To generate the set of "probe views", we geometrically formulate the rules of "one-point perspective (OPP)" and "two-point perspective (TPP)" [19], [20] w.r.t. room shapes. OPP refers to a single vanishing point formed by a room shape, while TPP refers to two vanishing points. The OPP is dedicated to capturing the architectures and symmetrical perfections (section 3.1). The TPP is dedicated to showing

[1] https://www.kujiale.com/

- *Shao-Kui Zhang, Hou Tam, and Tai-Jiang Mu are with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. E-mail: {zhangsk18, th21}@mails.tsinghua.edu.cn, taijiang@tsinghua.edu.cn.*
- *Yi-Xiao Li is with the Academy of Arts & Design, Tsinghua University, Beijing 100084, China. E-mail: liyixiao20@mails.tsinghua.edu.cn.*
- *Song-Hai Zhang is with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China, and also with the Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing 100084, China. E-mail: shz@tsinghua.edu.cn.*

Manuscript received 5 October 2021; revised 6 October 2022; accepted 12 October 2022. Date of publication 17 October 2022; date of current version 10 November 2023.
This work was supported in part by the Natural Science Foundation of China under Grant 62132012, and in part by the Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology.
(Corresponding author: Song-Hai Zhang.)
Recommended for acceptance by V. Popescu.
This article has supplementary downloadable material available at https://doi.org/10.1109/TVCG.2022.3214836, provided by the authors.
Digital Object Identifier no. 10.1109/TVCG.2022.3214836

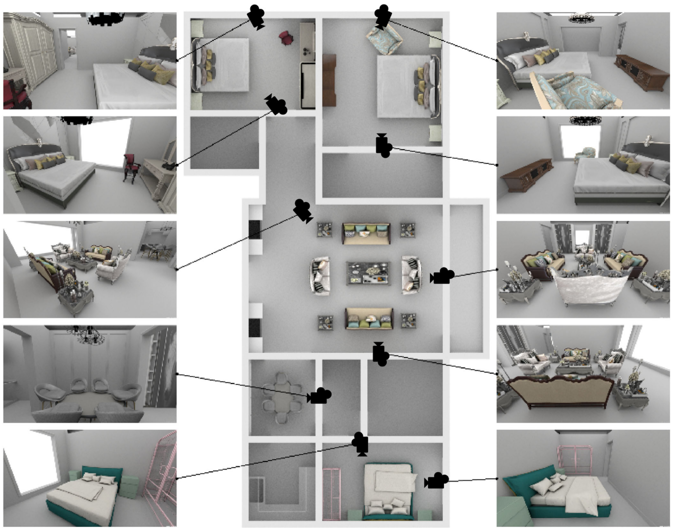


Fig. 1. We present a framework for viewing 3D scenes by automatically generating photographs. Our framework positions and directs a set of appealing views into scenes following the rules of residential photography. Subsequently, our method also supports generating a series of views for floorplans and more applications (seeing section 4), thus giving users a more perceptible and faster understanding of 3D scenes.

the human perception of depth (section 3.1). In the simplest case, OPP could be satisfied by placing the camera facing the middle of a wall. TPP could be satisfied by placing the camera in front of a corner and orienting it to the corner. However, this merely works for rooms with well-aligned layouts and cuboid shapes. For example, as shown in Fig. 2, to view the main content of the room, placing the camera along with all positions on the red dotted lines satisfies the constraint of OPP. However, most views are not favourable because some views may be blocked by the walls or include very few objects. The fundamental problem is that room shapes are irregular. Therefore, we computationally generalize residential photography, where five schemes are proposed to adapt complex geometries while keeping the OPP and TPP. Our method can generate probe views for arbitrary room shapes by doing so.

One could easily derive a series of views satisfying OPP and TPP from a room, but not all the generated probe views are guaranteed to be informative and aesthetic. To this end, several constraints, i.e., the content and aesthetic constraints, are proposed to refine the final human-centric views. The former measures the "informative extent" of views, e.g., a view should contain as many objects and connections of rooms as possible. The latter measures the visual effects of the views, e.g., the trisection rules [21]. As shown in Fig. 1, our method automatically generates individual photographs and a series of views, where further mappings from perspective views to floorplans are available, thus composing the basic features of SceneViewer. Section 3 will illustrate how we formulate interior photography and develop SceneViewer. In Section 4, we conduct experiments to verify our method and present potential applications that could be exclusively achieved using SceneViewer[2].

In summary, our work makes the following contributions:

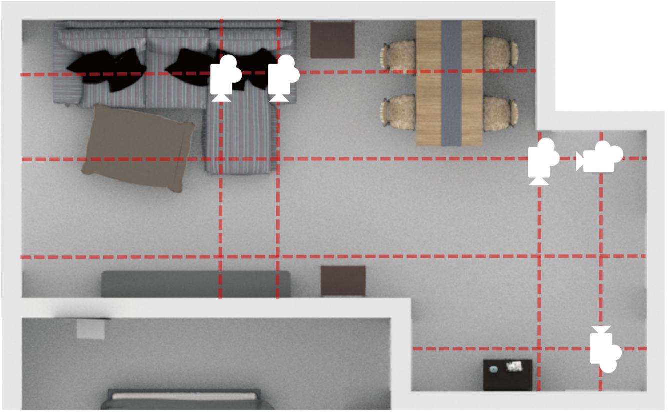


Fig. 2. A problem of one-point perspective (OPP) in residential photography. Although each line guides the camera to satisfy OPP, most camera positions are still unfavourable due to irregular room shapes in real life, e.g., the views at the white cameras.

- We present SceneViewer for automatically selecting views for 3D scenes by computationally formulating the rules of residential photography.
- We formulate a set of constraints that quantitatively measure how views are informative and visual-pleasingly satisfy residential photography.
- We present several applications, including (series of) views generation, trajectory, view-based 3D scene synthesis, etc., to demonstrate the potentiality of our SceneViewer.

2 Related Works

Research on 3D Scenes. Scene synthesis techniques require showcasing the synthesized results to debug and conduct experiments, e.g., user study [6], [14], [22]. Scene understanding techniques require a high-quality training set [2], [23], where existing literature pursues realistic rendering [24], [25] and view selections are indeed ignored. [26] learns and predicts human poses given furniture groups using captured views from sensors. [27] inserts objects given views of 3D scenes. [8], [9], [28] interactively synthesize 3D scenes, but training users to control the camera would be time-consuming. [15] renders a computer vision dataset using different views derived from 3D scenes and [29] trains neural networks to evaluate 3D scenes based on various views from datasets. [7], [30] make the style and colour of rooms consistent, so different views of verifying the results are needed. Our method is a tool for automatically photographing 3D scenes, bridging computer graphics and computer vision [31].

Viewpoint Selections. Existing literature focuses on selecting views for objects, e.g., CAD models, meshes, flows, etc. [32], [33] select views for volume rendering and [34] is intended for streamlines. [35] suggests camera views for point cloud segmentation. [36] locates optimal views based on feature extraction. [37] investigates camera control in cinematography. [38], [39] select views for CAD models, but the optimization loss is the number of primitives seen by the camera. The aesthetic of views for 3D scenes is more related to semantics. [18], [40] generate a series of views for a model, but the views are derived from regular shapes, e.g., dodecahedron, and icosahedron, which is unsuitable

[2] Code is publicly available at https://github.com/Shao-Kui/3DScenePlatform#sceneviewer.

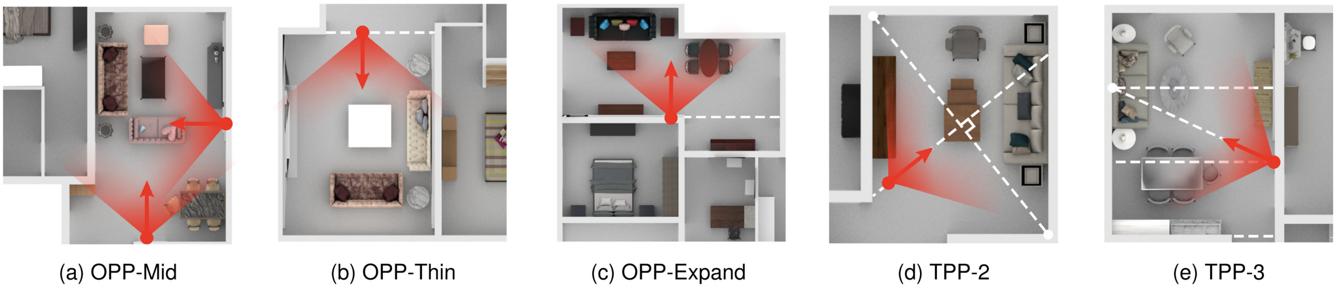

(a) OPP-Mid (b) OPP-Thin (c) OPP-Expand (d) TPP-2 (e) TPP-3

Fig. 3. The derived basic views. (a): OPP-Mid, where each view is positioned in the wall middle behind the view and directed toward the wall normal. (b) and (c): OPP-Thin & OPP-Expand, where each view is positioned toward a target wall and directed against the normal of the target wall. The former prefer narrower views, thus following the adjacent walls of the probe wall. The latter prefers wider views, thus expanding virtual walls based on the probe wall. (d): TPP-2, where views are placed in front of two adjacent walls as much as possible. (e): TPP-3, based on OPP-Thin and OPP-Expand, views are placed at the trisection point and are directed toward another trisection point of the probe wall.

for arbitrary room shapes. [41] measures the quality of different viewpoints for 3D objects. [42] selects upright orientations for 3D models. Please refer to an insightful survey [17] for more comparisons on selecting viewpoints for 3D models. Additionally, [43] estimates the visibility of a scene given a view based on many primitives and objects. Viewpoints are also used for generating pathways. [44], [45] generate a series of viewpoints (nodes) connected to guide a tour in a given 3D scene. [46] generates camera trajectories outside scenes or objects for automatic explorations. [47] develops an interactive system to offer users as much environmental knowledge as possible. In contrast, we focus on how every single view satisfies the rules of residential photography, where the connected pathways are our additional features (see Section 4.3). To the best of our knowledge, we are the first to formulate a view selection method for 3D scenes that adhere to the rules of photography.

Residential photography has formed its matured and detailed standards in industries. These standards are mainly based on the desire to take photos that are as visual-pleasing as possible, in line with how humans see scenes and remember them [19]. In response to this, the rules of "One Point Perspective" and "Two Point Perspective" [48] are summarized, where the visual plane is carefully adjusted, and the height should be consistent with human eyes [20]. Residential photography emphasizes essential elements in the visual center [49] to guide viewers' attention. For a more exquisite composition, photographers could also refer to the rule of thirds [50], where photographers arrange objects according to the intersections of vertical and horizontal lines trisecting a photo, e.g., the dominant object is often exhibited at the bottom third horizontally. Since users are more concerned about objects than walls, we typically try involving as many objects as possible via assigning fewer biases on room shapes [21] by tilting the camera. Additionally, [51] also investigates the problem of balancing objects in photos.

3 Formulating Residential Photography

Generating informative and visual-pleasing photographs for scenes is difficult due to the sizeable hypothetical space in complicated 3D environments. In other words, the layouts of 3D scenes could be arbitrary when placing potential probe views, unlike views derived from regular shapes, which are adopted for viewing meshes [18], [34], [36]. To "narrow" the space of possible probe views, we adapt rules from residential photography to arbitrary room shapes. Even so, many probe views may still satisfy the rules, so a set of constraints are incorporated to filter them further.

In general, the process of finding a set of n views $\hat{\mathcal{S}} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n\}$ for a room follows Equation (1), where $\mathcal{S}$ is the set consisting of all probe views derived from a room shape $\mathcal{R}$. The constraints function $C_{view}(\mathbf{c}|\mathcal{R}, \mathcal{O})$ is used to measure the goodness of a view $\mathbf{c}$, given the room shape $\mathcal{R}$ and its corresponding content $\mathcal{O}$. In this paper, each view is parameterized as $\mathbf{c} = (\vec{\zeta}, \vec{\beta})$, where $\vec{\zeta}$ positions the probe view and $\vec{\beta}$ is the normalized direction. The room shape $\mathcal{R}$ is formulated as a polygon, and the content $\mathcal{O}$ typically contains a set of furniture objects

$$\hat{\mathcal{S}} = \underset{\mathcal{S}' \subset \mathcal{S}, |\mathcal{S}'| = n}{\arg\max} \sum_{\mathbf{c} \in \mathcal{S}'} C_{view}(\mathbf{c}|\mathcal{R}, \mathcal{O}), \tag{1}$$

$$C_{view}(\mathbf{c}) = \lambda_c C_c(\mathbf{c}|\mathcal{R}, \mathcal{O}) + (1 - \lambda_c) C_a(\mathbf{c}|\mathcal{R}, \mathcal{O}). \tag{2}$$

Based on residential photography [19], [20], we adapt the one-point and two-point perspectives for deriving candidate views $\mathbf{c} \in \mathcal{S}$. However, there are further concerns. First, the shapes of modern rooms and floorplans are irregular. For example, a room shape could be a concave polygon with many edges (walls) instead of a simple rectangle. As a result, we extend the OPP and TPP to make it more robust on various polygons and discover more potential probe views as shown in Fig. 3. Second, residential photography should bring meaningful content to humans and be visually pleasing. However, there may exist many unappealing probe views subjecting to the rules for a given room shape. Therefore, we utilize constraints in $C_{view}(\cdot)$ for further filtering out less informative and visually unpleasing probe views in $\mathcal{S}$.

A room with s sides is abstracted as an s-gon composed by a set of edges $\mathcal{R} = \{\mathbf{r}_i | i = 1, 2, \ldots, s\}$. Contents $\mathcal{O}$ of a room is a set of objects (e.g., furniture), windows, doors, etc. $C_{view}(\cdot)$ considers the content constraints $C_c(\cdot)$ and the aesthetic constraints $C_a(\cdot)$. The former calculates the number of contents perceived by a probe view, e.g., the number of furniture, connections of rooms, etc. The latter analyzes how visually comfortable a probe view is, including the law of the third, angles of the camera, etc. These two constraints are balanced with λ_c. In the following subsections, Sections 3.1 and 3.2 detail how we adapt residential photography with room geometry. Sections 3.3 and 3.4 formulate the content constraints and aesthetic constraints, respectively.

In addition to the one-point and two-point perspectives, the generated views should be subject to the following two independent rules:

1) A view should give fewer biases on ceilings [21].
2) A human-centric view should be as tall as a camera held by a person [20].

Note that the above rules are optional but affect people's visual perception. Rule 1 suggests that photographs should accommodate more content on walls and grounds, i.e., the camera should be pushed lower or rotated towards the ground, which refers to the "pitch" of the camera. We adopt rule 1 by applying Rodrigues' formula to $\vec{u} \times \vec{\zeta}$ where $\vec{u}$ is the up vectors of views. Rule 2 suggests that photographs should not violate the view height of a person, e.g., humans seldom watch objects clinging to the ground or watching from top-down views. We utilize rule 2 by empirically assigning the camera's height slightly lower than human eyes.

Algorithm 1. The Pipeline of Finding One-Point Perspective Views

```
Input: Polygon of the room's inner side $\mathcal{R}$;
Output: A set of one-point perspective views $\mathcal{S}_{opp} \subset \mathcal{S}$;
$\mathcal{S}_{opp} := \emptyset$
for $r_i \in \mathcal{R}$ do
  Find the middle point $m$ of $r_i$
  Calculate the ray $n_i$ originated from $m$ and following the normal of edge $r_i$
  Find the predecessor wall $r_{pre}$ of edge $r_i$
  Find the successor wall $r_{nxt}$ of edge $r_i$
  for $r_j \in \mathcal{R} \setminus \{r_i\}$ do
    $p$ := LineIntersection($n_i$, $r_j$)
    if $p$ exists then
      $t1$ := LineIntersection($r_{pre}$, $r_j$)
      $t2$ := LineIntersection($r_{nxt}$, $r_j$)
      $\vec{\zeta} := (t_1 + t_2)/2$
      $\vec{\beta} := m - \vec{\zeta}$
      Push $(\vec{\zeta}, \vec{\beta})$ to $\mathcal{S}_{opp}$
    end if
  end for
end for
```

3.1 One-Point Perspective

The one-point perspective refers to a single vanishing point given a view in a room shape, requiring the central symmetry w.r.t walls. The OPP aims to show the interior as the architect intended, in all its symmetrical perfection, while creating a visually stunning shot of the views [20]. However, OPP may not always be satisfied. For example, the wall behind the L-shape sofa in Fig. 2 is longer than its two opposite walls, and choosing either to place the camera gives unfair biases to the dinning table with four chairs. The core problem is that room shapes are irregular in most cases. Humans have the prior knowledge to find a rectangular area satisfying OPP in a room, but not machines. Therefore, we propose three strategies that try to generate views satisfying OPP, as illustrated in Figs. 3a, 3b, and 3c:

1) assuming walls behind the camera satisfy the rule of central symmetry (OPP-Mid).
2) assuming opposite walls with complements satisfying the rule (OPP-Thin).
3) assuming the opposite walls with expansions satisfying the rule (OPP-Expand).

First of all, we assume the main contents of a room may be arranged in front of a wall, thus satisfying the central symmetry rule, so a camera is positioned in each wall middle behind it, towards the direction of wall normals, shown in Figs. 3a as "OPP-Mid". This assumption is a straightforward solution since we assume that the content is spread across a room.

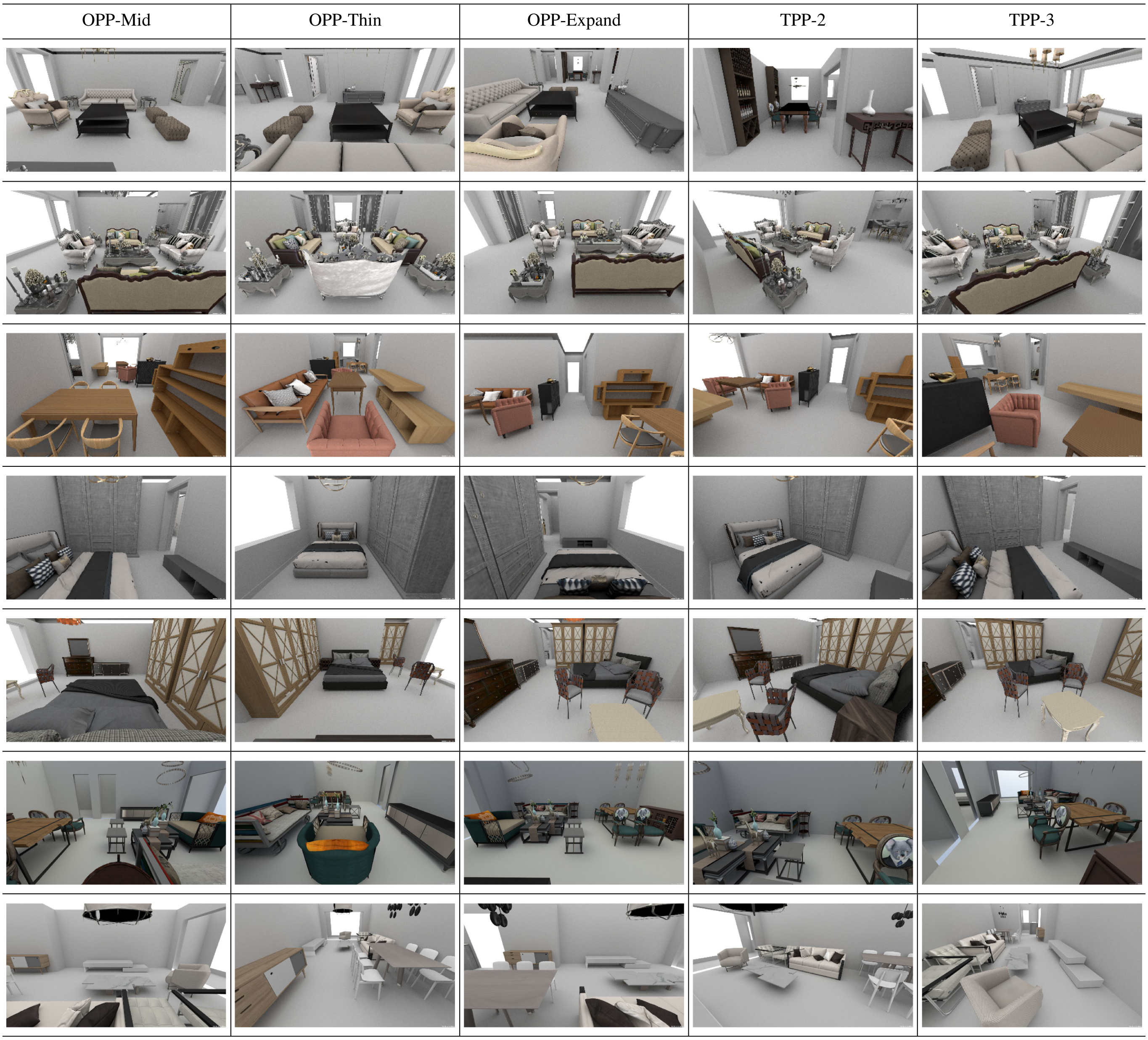
However, the primary content of a room usually gathers in a sub-area. Simply placing the camera in front of each wall could be far away from the content. The walls or other objects will also likely occlude the content. Therefore, we further present "OPP-Thin" and "OPP-Expand" to seek positions along the sub-areas of a room. If the rule is not satisfied by the walls behind the camera, it should be satisfied by opposite walls. Nevertheless, suppose the room shapes are irregular. In that case, we could fit the length of the walls behind the camera by complementing/clipping the opposite walls, as shown in Fig. 3b or fit the room shape by expanding the opposite walls across the room, as shown in Fig. 3c. Thus, the former prefers a narrower view, and the latter prefers a broader view.

Algorithm 2. Expanding Virtual Walls

```
Input:
An edge $r_i$ and its corresponding endpoints $p_1$ and $p_2$;
Polygon of the room's inner side $\mathcal{R}$;
Output: Two virtual and expanded "edges" $r_{pre}$ and $r_{nxt}$;
$N$ := -1
$P$ := -1
$r_{pre}$ := null
$r_{nxt}$ := null
$m := (p_1 + p_2)/2$
for $r \in \mathcal{R} \setminus \{r_i\}$ do
  $p$ := LineIntersection($r_i$, $r$)
  if $p$ exists then :
    $d := ||m - p||$
    if $(p_2 - p_1) \cdot (p - m) > 0$ then
      if $d > N$ then
        $N := d$
        $r_{nxt} := p$
      end if
    else
      if $d > P$ then
        $P := d$
        $r_{pre} := p$
      end if
    end if
  end if
end for
```

The idea of OPP-Thin is to make the area in front of a camera as rectangular as possible. Views of OPP-Thin focus on the quasi-rectangular region, excluding other parts of the room. As described in Algorithm 1, OPP-Thin initially selects

TABLE 1
Views Generated by Applying OPP and TPP to Irregular Room Shapes



Photographs in the same row belong to the same room and photographs in the same column belong to the same view-selecting strategy.

a probe wall r_i. A ray n_i is then cast from the center of r_i following its normal direction. The ray can intersect the extended line of the other wall r_j. If there is an intersection, i.e., "LineIntersection($\cdot$)" returns a point p, we subsequently calculate the intersections of r_j with r_{pre} and r_{nxt}, which are the adjacent walls of r_i. This process yields a virtual wall connecting $t1$ and $t2$. Finally, a probe view $\mathbf{c}$ is calculated as $((t_1+t_2)/2, m-(t_1+t_2)/2)$. This process is repeated to try each pair of walls.

In contrast, OPP-Expand tries to expand the region as much as possible, which is achieved by replacing the calculation of r_{pre} and r_{nxt} in Algorithm 1 to Algorithm 2, where a probe wall r_i tries to intersect all extended lines of others and find two endpoints. The endpoints should be as far away from r_i as possible. As a result, t_1 and t_2 would be derived by two expanded virtual walls, so a probe view has a broader perception of the room.

Due to various shapes, these three strategies may overlap, thus generating the same results. Our strategies are proposed to deal robustly with irregular shapes. OPP-Mid will be a straightforward solution if a room shape is a regular rectangle. Table 1 shows the qualitative OPP views of several rooms generated by our method, where each room is guaranteed to yield an OPP-Mid, an OPP-Thin and an OPP-Expand. We could see the views neatly concerning the room shapes.

3.2 Two-Point Perspective

When a photographer is taking a picture of a room, she/he usually includes two or three walls to give viewers a sense of depth so that the viewers can perceive the space. For example, if a photo has only one relatively complete wall, where other walls are not seen, or only tiny parts are shown, it would be hard for the viewers to determine their locations w.r.t the room shape. If two or more walls are presented, locations are easier to be determined, referring us to the two-point perspective (TPP) [20]. To this end, two more strategies are proposed based on TPP. First, to capture two walls in one view, the best camera position is usually located in one corner, looking towards the opposite corner [20]. However, as shown in Fig. 3d, irregular room shapes often prevent the cameras from capturing the opposite corners since they are likely to be occluded by other walls or two adjacent walls with different lengths. Thus, we formulate this rule with the modification, referred to as TPP-2 in Fig. 3d, that the camera should be as far away from the two walls as possible to capture as much content w.r.t the two walls as possible.

Algorithm 3. Detecting Whether an Object is Semantically Visible

```
Input:
  An object $o$ to be tested with vertices $v_1, v_2, \dots, v_n \in o$;
  A probe view $\mathbf{c} = (\vec{\zeta}, \vec{\beta})$ to be tested;
  Polygon of the room's inner side $\mathcal{R}$;
  Horizontal and vertical FoV $2\phi$ and $2\theta$ (aspect ratio);
  Contents of the room $\mathcal{O}$;
Output: is $o$ seen by $\mathbf{c}$;
  $o_{pre} := o$
  $\vec{\mathbf{v}} := \vec{\beta} \times \vec{u} / |\vec{\beta} \times \vec{u}|$
  $\vec{\mathbf{h}} := \vec{\beta} \times \vec{\mathbf{v}} / |\vec{\beta} \times \vec{\mathbf{v}}|$
  for $v \in o$ do
    $\vec{t} := v - \vec{\zeta}$
    $\vec{\mathbf{v}}' := -(\vec{\mathbf{v}} \cdot \vec{t})\vec{\mathbf{v}} + \vec{t}$
    $\vec{\mathbf{h}}' := -(\vec{\mathbf{h}} \cdot \vec{t})\vec{\mathbf{h}} + \vec{t}$
    if $(\vec{\beta} \cdot \vec{\mathbf{v}}')/|\vec{\beta}||\vec{\mathbf{v}}'| < \cos\theta$ then
      $o := o \setminus \{v\}$
    else if $(\vec{\beta} \cdot \vec{\mathbf{h}}')/|\vec{\beta}||\vec{\mathbf{h}}'| < \cos\phi$ then
      $o := o \setminus \{v\}$
    end if
  end for
  for $v \in o$ do
    if isCross($v - \mathbf{c}$, $\mathcal{R}$) then
      $o := o \setminus \{v\}$
    end if
  end for
  $o := o \setminus$ isOccluded$(o, \mathcal{O})$
  if $|o|/|o_{pre}| >= M$ then
    return True
  else
    return False
  end if
```

Each TPP-2 probe view is generated by traversing each pair of adjacent walls. A probe view is calculated by Equations (3) and (4). The direction $\vec{\beta}$ takes the cross product of camera up vector $\vec{u}$ with the vector connecting t_1 and t_2, which are the two opposite endpoints of the two walls, respectively. The position $\vec{\zeta}$ starts from the middle point of t_1 and t_2 and draws back along the $-\vec{\beta}$. If the horizontal field of view (FoV) 2ϕ is sufficient to accommodate two walls, the length is directly calculated using ϕ. Otherwise, the camera draws back until it touches a wall

$$\vec{\beta} = (t_1 - t_2) \times \vec{u}, \tag{3}$$

$$\vec{\zeta} = \frac{t_1 + t_2}{2} - \vec{\beta} \cdot \frac{||t_1 - t_2||}{2 \tan \phi}. \tag{4}$$

The best position to place a camera is often a third along the length of the back wall towards the opposite wall since it avoids photographing the extra side wall at an overly oblique angle [20]. This intuition implies that the camera should give a primary bias to the front wall while giving fewer biases to the side walls. The "back wall" and the "opposite wall" lead us to Algorithm 1, which aims at finding two relative walls. To apply "a third of the way", referred to as TPP-3 in Fig. 3e, we make the following modifications: a view is taken from one of the trisection points along r_j and is directed towards another trisection point on r_i. In practice, views of TPP-3 are derived concurrently with both OPP-Thin and OPP-Expand.

Note that TPP-3 theoretically still belongs to TPP since it aims at more than one vanishing point. Table 1 also shows the qualitative TPP views of several rooms.

3.3 Content Constraints

A probe view should perceive as many objects as possible, which leads us to occlusion detection. However, whether or not an object is in sight depends on various conditions. For example, the occlusion of an object could be attributed to two or more objects. Therefore, we present Algorithm 3 to decide if an object o is visible concerning a probe view $\mathbf{c}$. First, for each vertex v of an object, we calculate if it is on the camera's horizontal and vertical visual planes, i.e., a v could only be seen if it can be projected within the horizontal and vertical visual plane of the frustum. Then, we check if the room shape occludes the vertices. The function "isCross" detects if two geometries cross each other. The function "isOccluded" conducts a raycasting-based approach [52] based on [53] to calculate how many vertices in object o are occluded by other objects in $\mathcal{O}$. If the $(\vec{\zeta}, \vec{\beta})$ of probe views does not vary, rendering-based approaches such as [54] could also be leveraged for the acceleration. [39] could also be assembled to prevent perspective distortions if our method is applied in the real world instead of digital 3D scenes. Finally, Algorithm 3 decides the visibility of o based on a tolerance M. The visibility is measured by the ratio of visible vertices to the number of all vertices. If M takes a smaller value, more objects could be perceived in the results, while the visible part of each object may be small, e.g., a stool leg, and vice versa. Note that M is never set more significant than 0.5 since an object could occlude the vertices of itself.

To better perceive the room's layout, a view should be set to see the connections between rooms, e.g., doors and corridors, so we formulate the constraints for windows and doors. The overall process of counting the number of connections follows Algorithm 3, where the function "isCross" takes the $\mathcal{R} \setminus \omega$ as input and ω is the wall that a door or a window belongs. Due to the speciality of connections, this constraint is measured by $\sum_{\hat{o}} \alpha(\hat{o}, \mathbf{c}) / \sum_{o'} |\alpha(o', \mathbf{c})|$, where $\alpha(\cdot, \cdot)$ calculates the projected area of a connection, $\hat{o}$ and o' are from the sets of visible connections and all connections. Consequently, the constraints contribute by a weighted sum: $C_c(\cdot) = \lambda_{obj} C_{obj}(\cdot) + \lambda_{win} C_{win}(\cdot)$, where C_{obj} and C_{win} refer to the number of objects and area of connections, respectively. λ_{obj} and λ_{win} are the weights for them.

3.4 Aesthetic Constraints

In addition to content constraints, a view should also be visually aesthetic. First, from an aesthetic view, the orientations of objects w.r.t the camera should be carefully considered, e.g., we appreciate a bed from the front view. Otherwise, if a probe view has the same direction w.r.t a double bed, it would either be blocked by the bed or lack details of the bed. Thus, we formulate the layout constraint as $\sum_{o \in \mathcal{O}_{sim}} \vec{\beta} \cdot \vec{\theta_o}$, where $\vec{\theta_o}$ denotes the normalized direction vector of the object o. The $\mathcal{O}_{sim} \subset \mathcal{O}$ is the functional symmetries extracted with the guidance of PlanIT [13], a tool used to generate the relation graphs for scene synthesis. According to [13] and [28], only dominant objects with properties including “left-right”, “front-back”, “corner-left”, and “corner-right” would be added to $\mathcal{O}_{sim}$.

The rule of thirds is one of the best-known composition rules used in painting and photography, motivated by the golden ratio [55]. With the rule of thirds, the focus of interest must be placed at lines' intersections that divide the frame into thirds from top to bottom and from left to right [21]. Commonly, the dominant furniture stands at the intersection of the third to the bottom [20]. Subsequently, viewers' eyes are led along the trisection lines through the image, thus creating a more balanced composition and an aesthetically pleasing photo. One advantage of this approach is that it avoids directly placing dominant objects at the center of the photo [56]. When placing a dominant object, such as a table at the center of the photo, no matter where the other objects are arranged around the dominant object, the balance of the composition will not be improved. As a result, this constraint is formulated by a sign function, which returns 1 if a ray cast from intersections of the thirds touches a dominant object. Otherwise, it returns 0. We assemble the idea of MageAdd [28], which interactively inserts objects in real-time based on partitioning objects into dominant and subordinate objects to determine the dominance of objects.

We also constrain the direction of the camera concerning walls. If a probe view is geared too close to walls, it would contain much less meaningful content from walls causing obvious imbalance [51]. An extreme example should be that the direction of a camera is perpendicular to the normal of the wall behind it, where half the view is filled with the wall. To alleviate this, we propose an offset constraint $C_{off}(\cdot)$ measured by Equation (5). Specifically, for $\mathbf{c}$ derived from TPP-2, since $\mathbf{c}$ is towards two walls, no penalty will be imposed. If $\mathbf{c}$ is derived from OPP-Mid, the wall r_i is used according to Algorithm 1. Otherwise, the virtual wall r_j is used. The returned value equals the included angle between wall normals and $\vec{\beta}$. Similar to content constraints, aesthetic constraints are weighted and summed as $C_a(\cdot) = \lambda_{lay} C_{lay}(\cdot) + \lambda_{trd} C_{trd}(\cdot) + \lambda_{off} C_{off}(\cdot)$, where $C_{lay}(\cdot)$, $C_{trd}(\cdot)$ and $C_{off}(\cdot)$ refer to layout constraints, the rule of thirds and offset respectively. λ_{lay}, λ_{trd} and λ_{off} are weights for them

$$C_{off} = -\begin{cases} 0, & \text{if } \mathbf{c} \text{ is derived from TPP-2} \\ \arccos(\vec{\beta} \cdot n_i), & \text{if } \mathbf{c} \text{ is derived from OPP-Mid}. \\ \arccos(\vec{\beta} \cdot n_j), & \text{otherwise} \end{cases} \tag{5}$$

Table 2 shows views generated by discarding one constraint in each column. Since constraints are designed to work together, sometimes removing a single constraint may not yield poor results. Therefore, we give negative coefficients to the discarded constraints to amplify the adverse effect. For the content constraints, the results contain as much content as possible. For the layout constraint, it favours views that capture a set of objects facing the camera. For the rule of the third, the method guarantees that a dominant object appears at the trisection points of images. As for the offset constraint, since it can be well-satisfied by being perpendicular to walls behind it, a set of views could all comply with it. However, there are unappealing cases. For example, the rule of the third may pull an object closer so that the object is at the trisection point. However, this single object would occupy most parts of the image, resulting in a visual imbalance. Similarly, the connection constraint may cause the view to be too inclined to capture more windows and doors. As a result, each constraint is necessary, and $C_{view}(\mathbf{c})$ incorporates all of them.

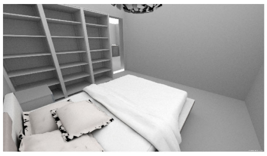
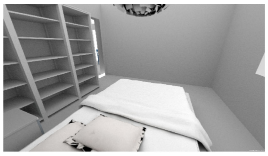
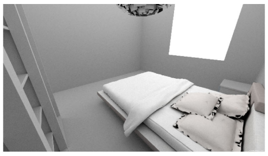
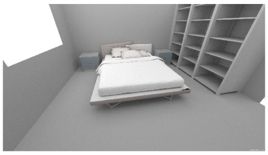
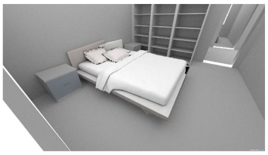
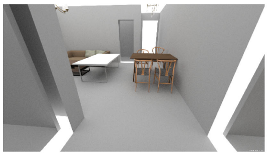
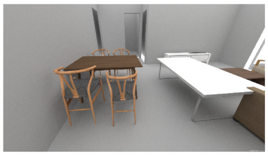
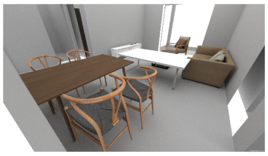
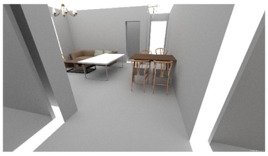
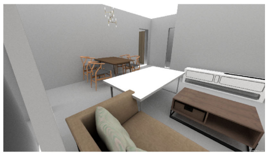
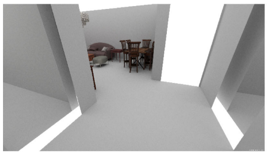
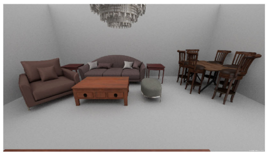
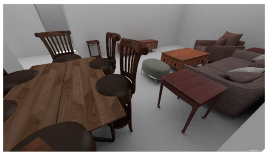
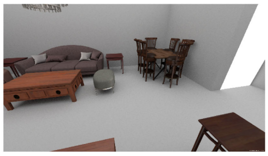
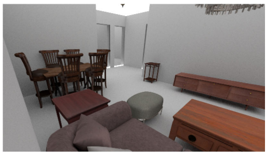
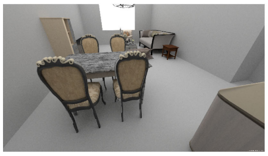
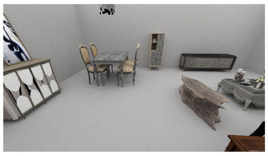
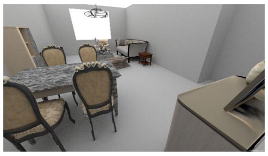
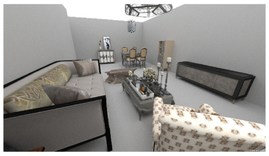
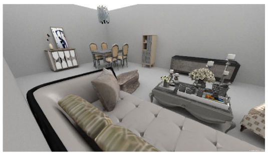
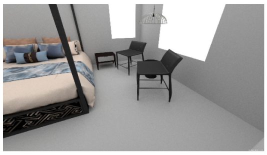
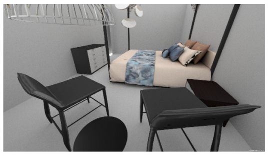
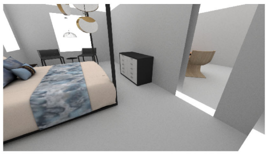
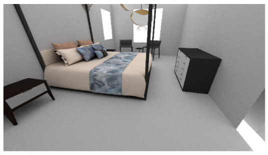
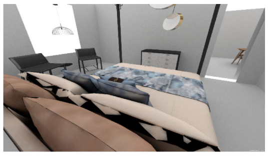
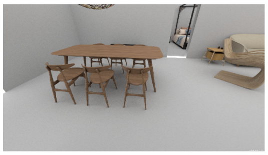
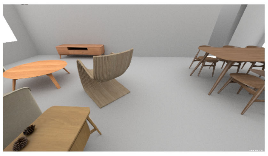
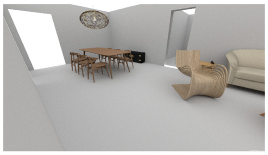
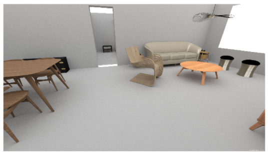
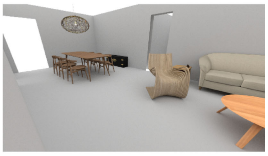
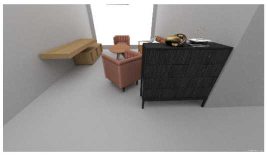
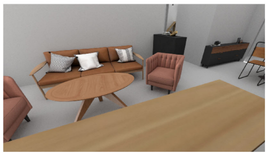
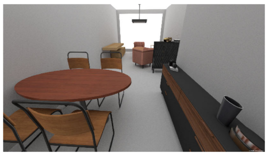
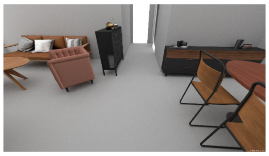
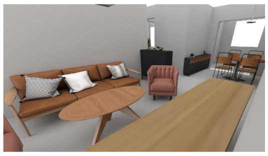
3.5 View Mapping

After having a set of independent views derived from each room, a floorplan could be already explored. However, this still isolates the views since connections among the views are inferred manually, e.g., finding a view of the exact next room based on the current view. Thus, we propose a way to automatically “map” individual views together with the floorplan.

Algorithm 4. Mapping Perspective Views Together With the Orthographic View

```
Input:
  Orthographic view $V_{orth}$;
  Perspective views and the positions of their corresponding
    viewpoint on the orthographic view $\hat{\mathcal{S}} = \{(\mathbf{c}_1, p_{\mathbf{c}_1}), (\mathbf{c}_2, p_{\mathbf{c}_1}),$
    $\ldots, (\mathbf{c}_n, p_{\mathbf{c}_1})\}$;
  Maximum perspective views per room $k$;
Output: Mapping result;
  $V_{selected} := \{\}$
  for each room do
    $V_{selected} := V_{selected}$ + selectViews(room, $k$)
  end for
  sort $V_{selected}$ by viewpoint's x-coordinate
  ($V_{left}$, $V_{right}$, $V_{top}$, $V_{bottom}$) := arrangeViews($V_{selected}$)
  sort $V_{left}$, $V_{right}$ by viewpoint's z-coordinate respectively
  sort $V_{top}$, $V_{bottom}$ by viewpoint's x-coordinate respectively
  while IntersectionExist() do
    for $\mathbf{c}_i$, $\mathbf{c}_j$ in ($V_{left}$, $V_{right}$, $V_{top}$, $V_{bottom}$) do
      if mappingLineIntersect(($\mathbf{c}_i$, $p_{\mathbf{c}_i}$), ($\mathbf{c}_j$, $p_{\mathbf{c}_j}$)) then
        swapPosition($\mathbf{c}_i$, $\mathbf{c}_j$)
      end if
    end for
  end while
```

TABLE 2
Qualitative Results When we Discard a Constraint

No #Objects	No Connections	No Layout	No "Third" Rule	No Offset
				
				
				
				
				
				
				

Each cell contains a relatively unappealing case due to the lack of a constraint.

Algorithm 4 formulates the process of view mapping. This algorithm selects no more than k perspective views for each room and places them around the orthographic view with as few intersections between the mapping lines as possible.

In order to provide users views from different positions, the function "selectViews" selects the top-k scattered viewpoints that cover the most area of each room. Since k is small, we calculate the convex hull of all available viewpoints and select k points from the hull to satisfy the most-scattered condition. If there are less than k points in the hull, a random pick from the interior points is adopted since the convex hull covers all the points.

The function "arrangeView" adaptively assigns the perspective views to the four sides around the orthographic view and gives an initial mapping layout. The fundamental strategy is to evenly split the perspective views to the left and right columns according to the x-coordinate of their viewpoints. The top and bottom spaces are utilized to place the middlemost perspective views when there are too many views to be arranged to make the result more compact. Suppose the top or bottom row capacity is sufficient for the selected middlemost views. Which row to choose depends on the overall distance to the corresponding side. Depending on the number of excess views, it may also take both the top and bottom space. Since all views are arranged to one of the four sides, an initial mapping layout is obtained with a fixed margin between each view. The views along each side are sorted to determine their initial positions and to avoid intersections of mapping lines within the same side. As shown in Algorithm 4, the views arranged to the left/right sides are sorted by the z-coordinate of their viewpoints. The x-coordinate sorts those from the top and bottom sides.

Though the sorting and arrangement above can significantly reduce intersections of lines, further intersection check is needed: we add a distance check between endpoints and line segments to the regular segment intersection test in Algorithm 4[3]. The distance check is triggered if the distance between an endpoint and a mapping line is less than the endpoint radius. The placement of the intersected views is swapped until no lines intersect.

4 Experiments and Applications

4.1 Setup and Results

We utilize the recently released 3D-Front dataset [57], [58], which contains nearly 10,000 floorplans with more than 70,000 rooms and nearly 10,000 3D models, to demonstrate our method fully. The aspect ratio of views is set as 1920:1080. The vertical FoV is set as 75°. For the height of cameras, as suggested by rule 2, we set a typical value of 1.3 meters.

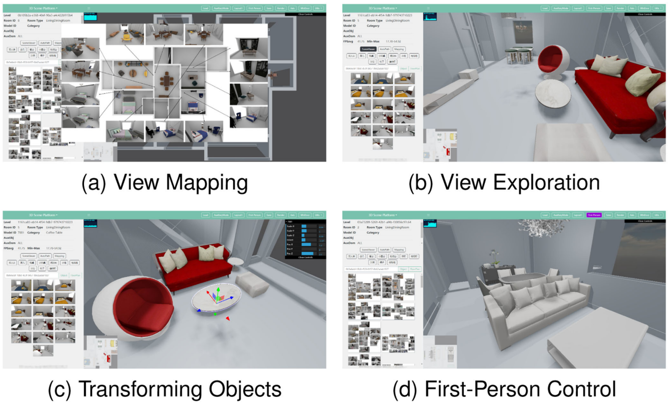
The weight of Equation (2) is set as 0.5. The weights of content constraints are set as 1.0 for objects and 0.6 for connections. The weights of aesthetic constraints are set as 3.0 for layout, 1.0 for the rule of thirds, and 10.0 for camera directions. The interactive platform is rendered based on Three.js[4], a popular rendering engine on top of WebGL, and other results are rendered using Mitsuba [16], a photo-realistic rendering system. Our method is implemented with Numpy (Scipy) and Shapely mainly for operating geometries, e.g., processing room shapes. The system is deployed on a desktop computer with a Titan RTX, 32GB memory, and an AMD Ryzen 2700x CPU.

Table 1 tabulates the generated views of several rooms based on the above constraints settings. Each row belongs to a specific room, and each column belongs to one of the views illustrated in Fig. 3. Normally, views are selected based on Equation (1), regardless of which basic views they are. In Table 1, to ensure that each room contains all the basic views to show how they differ, views in different columns are independent w.r.t Equation (1). In other words, Each strategy in Fig. 3 is guaranteed to be selected once and only once in each row. Ordinary results are shown in Fig. 5 and the supplementary materials, available online. Our method could capture different levels of details in rooms, leveraging the variety of proposed view-selecting strategies, e.g., capturing an entire room or focusing on a coherent group.

4.2 User Study

We conducted several user studies to verify how our method can generate views satisfying for residential photography.

Third-Party Evaluation. We invited 10 non-professional participants to adjust the cameras manually. We obtained a set of "handmade" photographs, leveraging the platform above, which enables participants to interact with 3D scenes through both the orbital control and the first-person control, as shown in Fig. 4. Before the experiment, one technical staff and a detailed manual told each participant how to control the platform. The technical staff also ensured participants had a basic knowledge of photography by showing them several well-designed photographs derived from the dataset. The rooms were selected with coherent groups greater than 2 to enable sufficient variations of views. The technical staff stood by the participant in case of technical questions during the experiment. In this way, we collected a set of over 400 views considered the ground truth (GT), which will be compared in the following experiments. Next, we conducted a user study to verify our ability to generate individual photographs, as shown in Fig. 6[5]. We invited 58 subjects to compare the views automatically generated by our method and the GT views collected before. These subjects are also amateurs, such as university students, workers, engineers, designers, researchers, social media, etc. Each subject was presented with a series of questions. For each question, a subject compared two photos and rated them respectively, where two photos were taken in the same room with the same layout. The rating scores range from 0 (very poor) to 5 (exceptionally appealing), considering the aesthetic, richness, goodness, and information obtained from the photo. Each questionnaire was uniquely and randomly generated, i.e., rooms were arbitrarily assigned to subjects, and two photos were selected randomly. Results are shown in Table 3, where we report the ratings' average (AVG) and standard deviation (STD). We could conclude that our method can generate photographs competitive with manual operation.


(a) View Mapping

(b) View Exploration

(c) Transforming Objects

(d) First-Person Control

Fig. 4. A platform for visualizing, exploring and manipulating 3D indoor scenes. More details are included in the supplementary video, available online.

Professional Evaluation. Next, we evaluate our method from the perspective of professional photographers. Another 10 subjects were recruited, and all reported professional skills in residential photography. They were invited to conduct a similar process described in the third-party evaluations to annotate ground truth views using the platform in Fig. 4. Subsequently, annotated views were organized. We recreated the questionnaire, where each question contains a professionally designed view and a generated view by our method in the same room. The newly recruited professionals were further invited to conduct the new questionnaire. Instead of purely finishing a questionnaire, this process is also considered an interview. Technical staff was nearby to

[3] The thickness of a line segment and the radius of the endpoint is also taken into account when testing whether two mapping lines intersect.

[4] https://threejs.org/

[5] See the supplementary document for more details about the UI, available online.

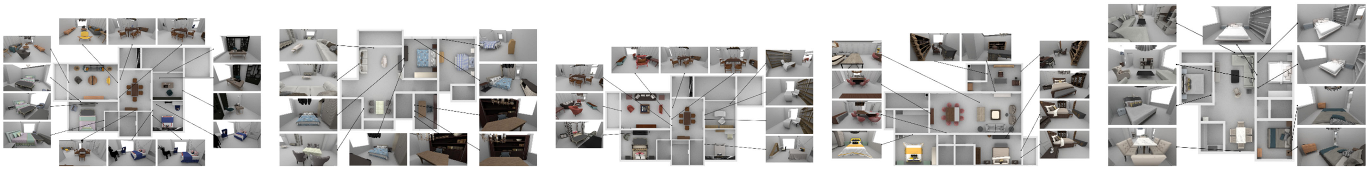
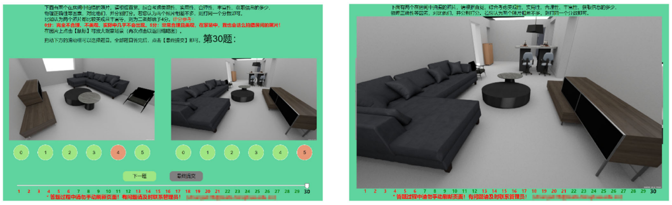


Fig. 5. The results of view mapping, where generated photos are mapped around floorplans. More results are included in the supplementary materials, available online.

record opinions, reasons and suggestions about why they favoured one view. Note that each subject not only answered questions involving her/his photos but also answered questions involving photos annotated by the others, leading to a more comprehensive evaluation. Table 4 shows the averages and standard deviations of ratings. Since professionals have insights, their results are slightly better, but our method are still comparable. The qualitative summary of the interviews will be discussed in Section 5.

Mapping Evaluation. We conduct one more user study to verify the ability to generate a series of photographs for floorplans. Another 20 subjects were invited, and each subject was presented with two series of photos: the traditional solution and the view mapping of this paper. The traditional solution is adopted by most platforms, e.g., [59], where they typically show a top-down orthographic view of the entire floorplan and several accompanying photos captured by ordinary users. Each subject was only given one mapped view of each floorplan. For both methods, two sets of floorplans were shown to users, and each set contained 20 floorplans. The two sets do not intersect, and floorplans have approximately the same scales. Subjects were asked to watch all floorplans and sorted them by preference carefully. Results are shown in Table 5, where we report the average time consumption for sorting and the visual satisfaction of the two methods (brackets contain the standard deviations). Consequently, our method alleviates the time required to understand floorplans and improves users' visual satisfaction. According to the Kruskal-Wallis H-Test (brackets contain p-values), there are significant statistical differences between our method and the traditional solution. Thus, we can conclude that our method of generating a series of photographs for floorplans is more effective than the traditional method. We also interviewed several subjects, and their responses indicate that our method has a sense of wholeness and compactness, which reduces the time consumed to understand the floorplans.



下面有两个在房间中拍摄的照片，请根据直觉，综合考虑美观性、实用性、合理性、丰富性、获取信息的多少、物理正确性等因素，对比他们，并分别打分。若您认为两个照片相差不多，则打同一个分数即可。
比如认为两个照片都比较美观且丰富等，则为二者都给了4分。评分参考：
0分：完全不合理、不美观、实际中几乎不会出现。5分：非常合理且美观，在家装中，我也会这么拍摄房间的照片！
在图片上点击【鼠标】可放大观察场景（再次点击以返回缩略图）。
拖动下方的滑动框可以选择题目。全部题目答完后，点击【最终提交】即可。第30题：

* 答题过程中请勿手动刷新页面！有问题请及时联系管理员！ [redacted]

(a)

* 答题过程中请勿手动刷新页面！有问题请及时联系管理员！ [redacted]

(b)

Fig. 6. The user study platform of section 4.2. (a): Each user is presented with a series of questions containing a manual photo and a photo taken by our method. Temporally skipping a question or jumping to another question is enabled. (b): Each photo can be zoomed in or out.

4.3 View Visualization

In this section, we implement a platform aiming at manipulating 3D scenes with the help of SceneViewer to show the potentiality of our method. First, as shown in Fig. 4a, we embed the view mapping into the platform to give the user a more accessible and quicker understanding of floorplans. As verified in Section 4.2, view mapping notably reduces the time consumption of selecting favoured floorplans for each user. Note that view mapping focuses on how results are effectively presented to users instead of how results are recommended to users, which belongs to the topic of recommendation systems [60].

Second, when arranging objects in floorplans, we often need to wander the 3D scenes, causing trial-and-error and time-consuming operations on changing views. As shown in Fig. 4b, by clicking on the "SceneViewer" button, individual views are generated at the left search bar in the form of rendered images. The camera is smoothly translated into the 3D scene by clicking on a rendered image. Users can easily access different rooms and switch to different views within each room, so the manipulations, such as transforming objects, adding objects, and deleting objects, can be performed more efficiently.

Additionally, the generated views are not "still". Users could play an automatic animation connecting views on a floorplan if they favour a quicker tour instead of interactions. A trajectory is a user-oriented tour, so it should go through each view at least once. Meanwhile, the trajectory should pass each view at most once to reduce redundancies. Therefore, this leads us to the Hamilton pathfinding algorithm, which is an np-complete problem [61], [62]. Efforts [63], [64] were made to heuristically solve this problem by efficiently finding the approximated and sufficiently close results. In this paper, we utilize the algorithm of Angluin and Valiant [64] to find such a trajectory.

The basic idea of [64] is that we start from a random vertex p, and each time we choose an edge (p, v') from the graph $G = (V, E)$. If v' is not an existing vertex in the Hamilton path, we add v' and set it to the new p. Otherwise, a short-circuit case is detected, and we cut off the edge (u, v') where u is the neighbour of v'. Then, the edge (p, v') is added to the path and p is set to u. However, additional concerns are inevitable due to the complexity of 3D scenes. Specifically, an initial graph $G = (V, E)$ is formulated and passed to [64], where V refers to the generated views, but E is not explicitly given in 3D scenes. For three reasons, We can not assume that E connects all the vertices resulting in a complete graph. First, a transition between two views should not go through walls; second, a transition between two views in different rooms should not happen; and third, edge connections should consider the distances between vertices. Additionally, a trajectory should be physically as

TABLE 3
User Study: Third-Party Evaluation

-	Bedroom		Living&Dinning Room		Total	
Metric	GT	Ours	GT	Ours	GT	Ours
AVG	3.228	3.359	3.209	3.198	3.221	3.296
STD	1.09	1.096	1.157	1.203	1.117	1.142

TABLE 4
User Study: Professional Evaluation

-	Bedroom		Living&Dinning Room		Total	
Metric	GT	Ours	GT	Ours	GT	Ours
AVG	3.644	3.492	3.797	3.52	3.707	3.503
STD	0.993	0.94	0.937	0.923	0.973	0.933

TABLE 5
User Study: Comparing Mapped Views

Metric	Ground Truth		Ours
Time Consumption	1395.0 (416.896)		880.0 (271.819)
Kruskal-Wallis H-Test		13.534 (0.0)	
Visual Satisfaction	2.6 (1.281)		4.05 (0.865)
Kruskal-Wallis H-Test		13.093 (0.0)	

short as possible and be thematically successive, i.e., several adjacent transitions need to focus on the same room instead of "wandering" among rooms. Consequently, to determine E and address concerns in 3D scenes, we proposed the following rules to complement [64]:

1) Two views do not share an edge in E if they are in two different rooms.
2) We no longer uniformly choose a random edge (p, v') from G. Instead, we choose an incident edge with the nearest distance w.r.t v'.
3) Extra-views are placed at the positions of doors.
4) Each extra view can be passed more than once.
5) Each photography-satisfied view belongs to and only belongs to the room where it is generated.
6) Each extra-view belongs to two rooms which are the rooms on both sides of the door.

Rules 1 and 3 address the problem of showing connections between rooms. Rule 2 guides thematically successive trajectories. Rule 1, 5, and 6 are for constructing the edge set E. Finally, rule 4 guarantees the existence of trajectories because it is very likely to have a room with only one rotation to pass in and out. The animated results are shown in the supplementary materials, available online.

4.4 View-Based Scene Synthesis

In this section, we implement a scene synthesis application to show the additional feature of our method, thus further strengthening our contribution to view generations. We observe that top-down views are not standard in our daily lives. 3D scene modelling tools such as Kujiale [65] and Planner5d [66] utilize the orbital control that translates and rotates the perspective camera based on the top-down views, which is convenient for users to insert/translate/rotate/rescale objects in the "god's perspective". Nevertheless, the orbital control does not consider the daily views, i.e., residential photography. In this application, we facilitate the manipulation of an object in 3D scenes [65], [66] by viewing the effect of manipulation with our efficient view generation.

Foremost, intuitively, views of a room being appealing are necessary for the layout of the room to be appealing. This intuition motivates us to incorporate automatic photography into interactive scene synthesis [8], [9], [28], where we iteratively suggest views to users and require users to fill in the room based on the views. Filling in rooms could be accomplished by either traditional interactions [65], [66] or MageAdd [28], a framework for synthesizing 3D scenes through "popping" up objects with transformations based on cursor movements in real-time.

Starting from an empty room, we first suggest the best view, typically faced toward windows or doors, according to content constraints (Section 3.3). Subsequently, our method iteratively generates appropriate room views when the contents are changed. Users should fill the room with objects using the suggested view for each iteration. This iterative process terminates once the worst view of the room satisfies Equation (1) (Section 3) and the room has no space left. We formulate two rules during iterations to fully explore the variation of views. First, we suggest alternatively best and worst views according to Equation (1). For example, if the best view is suggested in the current iteration, the worst view will be suggested in the next iteration. Second, swapping views are only triggered by dominant objects, e.g., a double bed, coffee table, etc. Selecting and transforming a dominant object is independent of other objects, whereas inferring subordinate objects depends on the dominant objects they belong to [6], [14], [28], which

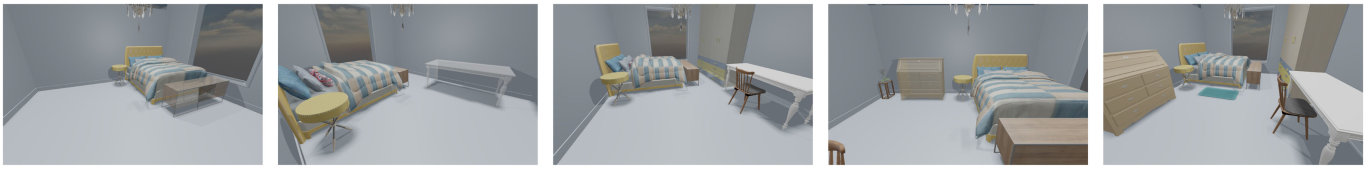


Fig. 7. A scenario of view-based scene synthesis, where the transparent object is the object to be added. We first suggest a view w.r.t the double bed, and the room is further filled with several subordinate objects relating to the bed. After the user inserts a desk, the camera is transformed toward another wall hinting the user to add a cabinet. This process continues until the room is appropriately filled. More details are included in the supplementary video, available online.

allows users to exploit a view as much as possible. A scenario is shown in Fig. 7.

5 Discussions and Limitations

This section presents further discussions on our method and its limitations, giving insights for future work.

Preparing Method Input. This paper focuses on automatic photography in digital 3D scenes, where the input, i.e., the scene configurations, can either be captured by standard 3D scene modelling tools [65], [66] or be provided by 3D scene datasets [57], [58], [67]. For pure 3D scene meshes and 3D scans, existing literature is available for semantic segmentation [68], [69], model retrieval [70] and registering [71] to convert them into the configurations required by our method. These approaches may not always work, for real-world scans could be complex, so we recommend using the 3D scene modelling tools above. However, developing tools for acquiring scene configurations is beyond the scope of our paper.

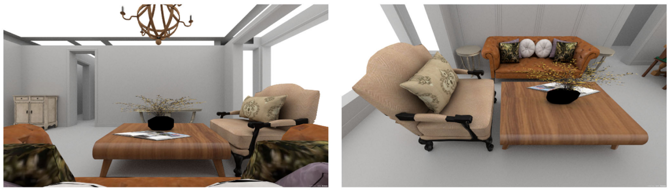
Object-Oriented Photography. One of our motivations is to enable users to perceive as much content as possible. However, sometimes a user may want to take a specific object as the theme while other accompanied contents are considered less significant. For example, this happens when a furniture seller tries to advertise her/his products. The photos would only focus on the commodity objects, and other objects contribute to the reality of the 3D scene, which is exclusively elaborated for the objects they would like to sell. Fig. 8 shows a few preliminary results of group-oriented photographs as alternative future directions of automatic photography.

Photo Clipping. To enable views to be aesthetic and contain more content, ceilings are considered less critical, and a camera should give more attention to walls and grounds. Therefore, three methods are adopted. First, photographers could lower their cameras so that grounds are "lifted" in the photos, pushing the camera closer to the ground. Second, one could rotate the pitch of the camera towards the ground. However, this results in more vanishing points, thus visually affecting the aesthetic of views. Third, photographers could cut off the upper image patch containing parts of the ceilings while preserving the lower after obtaining the views, but the extent of how many pixels to cut, how to preserve the aspect ratio and how this potentially influences the calculation of constraints should be carefully investigated in the future.

Balance of View Content. We may also force views to be balanced w.r.t objects, but this would potentially break the rules of photography or distract cameras, as shown in Fig. 9. Future research should be conducted to improve the balance of objects.

Rounded Walls. Because our method is executed based on lines of polygons, i.e., room shapes, some exceptional cases, such as rounded walls, should be discretized to polygons beforehand. The mapping of our method focuses on floorplans. However, given a house with multiple floors, our method could only generate separate maps for each floor. So an improvement could be a more comprehensive mapping on multiple floors.

Alternative Types of Views. Our method does not cover all types of "shots" humans would have taken. A person may try other alternative views in addition to OPP and TPP. We have interviewed the professionals in Section 4.2 and summarised their suggestions, which are also future directions of automatic residential photography. First, though views generated by our method follow the OPP and TPP rules, it would be more appealing to slightly tune the camera direction to achieve a better perspective, as shown in Fig. 10a.



(a) (b)

Fig. 8. The photography for an object group includes a coffee table and two sofas, where other objects serve as foils. (a) Mimicking first-person views. (b) Using the bounding square of the group as the virtual room shape and calculating its one-point perspective view.

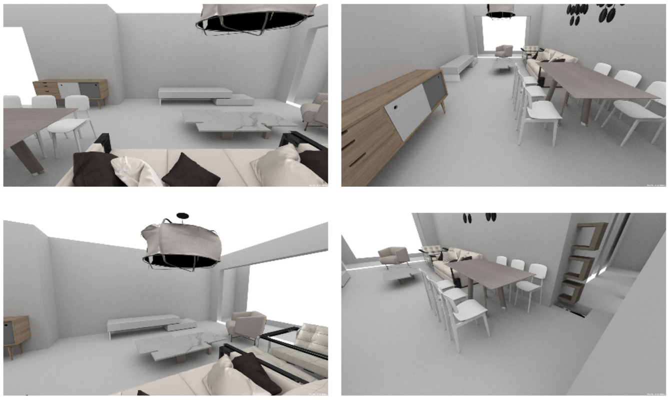


Fig. 9. The balancing problem to be explored in the future. Top: The original results. Bottom: The balanced results based on objects. More advanced techniques should be made so that the balancing neither violates the existing photography rules nor drives cameras in weird directions.

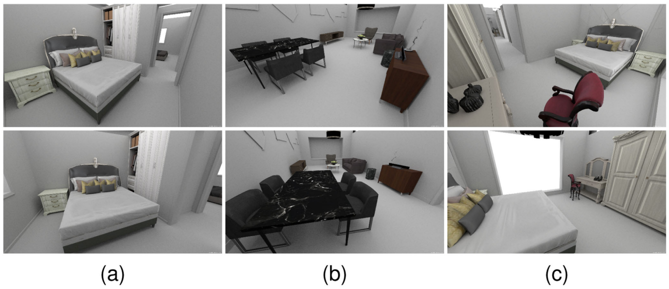


Fig. 10. Typical views based on the shots taken by professionals (top row) and the corresponding views generated (bottom row) in the same rooms. (a): A better perspective could be achieved by slightly tuning the camera direction. (b) Sometimes, our method may create a sense of oppression. (c) Sometimes an object (e.g., the double bed) should be fully exposed.

Second, the compositions of photos can be optimized to avoid a sense of oppression, as shown in Fig. 10b. Third, sometimes, it is better to force an object to be fully exposed, e.g., the double bed in Fig. 10c.

Views Generated w.r.t the Constraints. Currently, constraints of our method are used for evaluating the probe views. In contrast, those views are not generated according to a specific constraint, i.e., the views are generated according to OPP and TPP rules regardless of constraints. For example, the method does not guarantee that the objects of interest satisfy the rule of thirds. Instead, the views might be chosen because the probe views have the suitable objects already placed, satisfying the rule of the thirds and other constraints since constraints work together.

FoV and Camera Height. We also notice two hyper-parameters: the FoV and the camera's height. The FoV is set as 75° because it is a choice to capture more content [72]. However, humans feel differently given different lenses (FoVs). It is generally believed that the typical focal length is the focal length that best represents the human eye's perception of the surrounding environment. The field of view covered by a standard lens is between 40° and 50°, similar to what a human would see. A relatively large FoV increases the perceived depth and the distances between objects. This increment causes near objects to appear closer, thus visually being bigger than they are, while distant objects appear smaller and farther away [73]. Selecting FoVs is a trade-off between mimicking human eyes and the sense of depth. Note that optical distortions of a real-world camera could be corrected given the right lens profiles, and a virtual camera's projective distortions could be corrected using an existing method such as [39]. Similarly, the camera height is set lower than human eyes because it gives fewer biases to the ceilings. In daily life, we may take photos with our knees bent, which is also a trade-off between mimicking human eyes and catching more content. Thus, investigating how humans feel given different FoVs and heights of a camera is also a promising future direction.

Dataset. As a method designated for 3D scenes, our method also suffers from implausible cases from the dataset. As shown in Fig. 11, if a room is occupied with a simple but colossal object, the constraints are hard to be met for given probe views. Our method may also fail to adjust views given a room with an implausible layout, e.g., the orientation of a double bed is inconsistent with its nightstands. Hypothesizing views for tiny rooms is also not as capable as dealing with regular rooms. Consequently, they are also worthy of investigation.

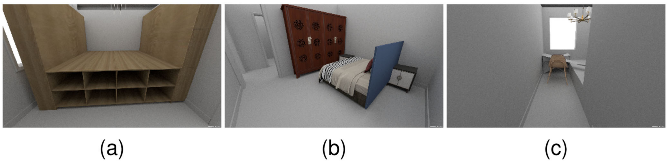


Fig. 11. Three representative room layouts that are abnormal in the dataset. (a): A room is filled up with a huge object. (b): An implausible layout. (c): A room with a tiny size.

6 Conclusion and Future Works

In this paper, we proposed and demonstrated a framework, SceneViewer, to generate views for 3D scenes by computationally formulating residential photography. Our approach is flexible in that it can generate individual views by selecting the best viewpoints and combining them with the floorplan, providing users with both local details and overall understanding. Extensive experiments and applications show that our method is effective and has various academic and industrial uses. We hope this work could advance future research, especially the literature relying upon showcasing 3D scenes.

Our method effectively visualizes indoor scenes in virtual environments, thus setting up a foundation for future works such as trajectory generation for movie cameras. We provide a way for selecting camera positions and animating them efficiently. By adding constraints such as character importance or the location of a critical storyline, a similar idea can be derived to support generating character-aware, story-aware, and even audience-interactable movie camera trajectory.

Acknowledgments

The authors would like to thank all reviewers for their thoughtful comments.

References

[1] L. Nan, K. Xie, and A. Sharf, "A search-classify approach for cluttered indoor scene understanding," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 1–10, 2012.

[2] S. Satkin, J. Lin, and M. Hebert, "Data-driven scene understanding from 3D models," 2012.

[3] K. Xu et al., "Organizing heterogeneous scene collections through contextual focal points," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 1–12, 2014.

[4] S.-H. Zhang, S.-K. Zhang, Y. Liang, and P. Hall, "A survey of 3D indoor scene synthesis," *J. Comput. Sci. Technol.*, vol. 34, no. 3, 2019, Art. no. 594.

[5] Q. Fu, X. Chen, X. Wang, S. Wen, B. Zhou, and H. Fu, "Adaptive synthesis of indoor scenes via activity-associated object relation graphs," *ACM Trans. Graph.*, vol. 36, no. 6, pp. 1–13, 2017.

[6] S.-H. Zhang, S.-K. Zhang, W.-Y. Xie, C.-Y. Luo, Y.-L. Yang, and H. Fu, "Fast 3D indoor scene synthesis by learning spatial relation priors of objects," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 9, pp. 3082–3092, Sep. 2022.

[7] T. Liu, A. Hertzmann, W. Li, and T. Funkhouser, "Style compatibility for 3D furniture models," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 1–9, 2015.

[8] L.-F. Yu, S.-K. Yeung, and D. Terzopoulos, "The clutterpalette: An interactive tool for detailing indoor scenes," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 2, pp. 1138–1148, Feb. 2015.

[9] S. Zhang, Z. Han, and H. Zhang, "User guided 3D scene enrichment," in *Proc. 15th ACM SIGGRAPH Conf. Virtual-Reality Continuum Appl. Ind.*, 2016, pp. 353–362.

[10] M. Yan, X. Chen, and J. Zhou, "An interactive system for efficient 3D furniture arrangement," in *Proc. Comput. Graph. Int. Conf.*, 2017, pp. 1–6.

[11] S. Qi, Y. Zhu, S. Huang, C. Jiang, and S.-C. Zhu, "Human-centric indoor scene synthesis using stochastic grammar," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5899–5908.

[12] K. Wang, M. Savva, A. X. Chang, and D. Ritchie, "Deep convolutional priors for indoor scene synthesis," *ACM Trans. Graph.*, vol. 37, no. 4, 2018, Art. no. 70.

[13] K. Wang, Y.-A. Lin, B. Weissmann, M. Savva, A. X. Chang, and D. Ritchie, "PlanIT: Planning and instantiating indoor scenes with relation graph and spatial prior networks," *ACM Trans. Graph.*, vol. 38, no. 4, 2019, Art. no. 132.

[14] S.-K. Zhang, W.-Y. Xie, and S.-H. Zhang, "Geometry-based layout generation with hyper-relations among objects," *Graph. Models*-vol. 116, 2021, Art. no. 101104.

[15] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla, "Understanding real world indoor scenes with synthetic data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4077–4085.

[16] W. Jakob, "Mitsuba renderer," 2010, [Online]. Available: http://www.mitsuba-renderer.org

[17] X. Bonaventura, M. Feixas, M. Sbert, L. Chuang, and C. Wallraven, "A survey of viewpoint selection methods for polygonal models," *Entropy*, vol. 20, no. 5, 2018, Art. no. 370.

[18] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung, "On visual similarity based 3D model retrieval," in *Comput. Graph. Forum*, vol. 22, no. 3, pp. 223–232, 2003.

[19] R. Hicks and F. Schultz, *Interiors: A Guide to Professional Lighting Techniques Interior Shots (Pro-lighting)*, Brighton, U.K.: RotoVision, 1996.

[20] M. G. Harris, *Professional Interior Photography*, New York, NY, USA: Taylor & Francis, 2003.

[21] D. Prakel, *Basics Photography 01: Composition*, vol. 1, Worthing, U.K., AVA Publishing, 2006.

[22] L.-F. Yu, S. K. Yeung, C.-K. Tang, D. Terzopoulos, T. F. Chan, and S. Osher, "Make it home: Automatic optimization of furniture arrangement," *ACM Trans. Graph.*, vol. 30, no. 4, 2011, Art. no. 86.

[23] Y. M. Kim, N. J. Mitra, D.-M. Yan, and L. Guibas, "Acquiring 3D indoor environments with variability and repetition," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 1–11, 2012.

[24] W. Li et al., "Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset," in *Proc. Brit. Mach. Vis. Conf.*, 2018, p. 77.

[25] J. Zheng, J. Zhang, J. Li, R. Tang, S. Gao, and Z. Zhou, "Structured3D: A large photo-realistic dataset for structured 3D modeling," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 519–535.

[26] M. Savva, A. X. Chang, P. Hanrahan, M. Fisher, and M. Nießner, "PiGraphs: Learning interaction snapshots from observations," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–12, 2016.

[27] Y. Liang, L. Fan, P. Ren, X. Xie, and X.-S. Hua, "Decorin: An automatic method for plane-based decorating," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 8, pp. 3438–3450, Aug. 2020.

[28] S.-K. Zhang, Y.-X. Li, Y. He, Y.-L. Yang, and S.-H. Zhang, "Mageadd: Real-time interaction simulation for scene synthesis," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 965–973.

[29] A. Luo, Z. Zhang, J. Wu, and J. B. Tenenbaum, "End-to-end optimization of scene layout," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3754–3763.

[30] Y. He et al., "Style-compatible object recommendation for multiroom indoor scene synthesis," 2020, *arXiv:2003.04187*.

[31] M.-M. Cheng, Q.-B. Hou, S.-H. Zhang, and P. L. Rosin, "Intelligent visual media processing: When graphics meets vision," *J. Comput. Sci. Technol.*, vol. 32, no. 1, pp. 110–121, 2017.

[32] U. D. Bordoloi and H.-W. Shen, "View selection for volume rendering," in *Proc. IEEE Vis.*, 2005, pp. 487–494.

[33] M. Ruiz, I. Boada, M. Feixas, and M. Sbert, "Viewpoint information channel for illustrative volume rendering," *Comput. Graph.*, vol. 34, no. 4, pp. 351–360, 2010.

[34] J. Tao, J. Ma, C. Wang, and C.-K. Shene, "A unified approach to streamline selection and viewpoint selection for 3D flow visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 3, pp. 393–406, Mar. 2013.

[35] S. Yang, J. Xu, K. Chen, and H. Fu, "View suggestion for interactive segmentation of indoor scenes," *Comput. Vis. Media*, vol. 3, no. 2, pp. 131–146, 2017.

[36] S. Takahashi, I. Fujishiro, Y. Takeshima, and T. Nishita, "A feature-driven approach to locating optimal viewpoints for volume visualization," in *Proc. IEEE Vis.*, 2005, pp. 495–502.

[37] M. Christie, P. Olivier, and J.-M. Normand, "Camera control in computer graphics," in *Comput. Graph. Forum*, vol. 27, no. 8, pp. 2197–2218, 2008.

[38] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich, "Automatic view selection using viewpoint entropy and its application to image-based modelling," in *Comput. Graph. Forum*, vol. 22, no. 4, pp. 689–700, 2003.

[39] P.-P. Vázquez and M. Sbert, "On the fly best view detection using graphics hardware," in *Proc. 4th Int Conf. Vis., Imag., Image Process.*, 2004, pp. 625–630.

[40] M. Eitz, R. Richter, T. Boubekeur, K. Hildebrand, and M. Alexa, "Sketch-based shape retrieval," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 1–10, 2012.

[41] L. Neumann, M. Sbert, B. Gooch, and W. Purgathofer et al., "Viewpoint quality: Measures and applications," in *Proc. 1st Eurographics Workshop Comput. Aesthetics Graph., Vis. Imag.*, 2005, pp. 185–192.

[42] Z. Liu, J. Zhang, and L. Liu, "Upright orientation of 3D shapes with convolutional networks," *Graphical Models*, vol. 85, pp. 22–29, 2016.

[43] S. Freitag, B. Weyers, and T. W. Kuhlen, "Efficient approximate computation of scene visibility based on navigation meshes and applications for navigation and scene analysis," in *Proc. IEEE Symp. 3D User Interfaces*, 2017, pp. 134–143.

[44] P. Barral, G. Dorme, and D. Plemenos, "Intelligent scene exploration with a camera," in *Proc. Int. Conf. 3IA*, 2000, pp. 3–4.

[45] C. Andújar, P. Vázquez, and M. Fairén, "Way-finder: Guided tours through complex walkthrough models," in *Comput. Graph. Forum*, vol. 23, no. 3, pp. 499–508, 2004.

[46] B. Jaubert, K. Tamine, and D. Plemenos, "Techniques for off-line scene exploration using a virtual camera," in *Proc. Int. Conf. 3IA*, 2006, Art. no. 1.

[47] S. Freitag, B. Weyers, and T. W. Kuhlen, "Interactive exploration assistance for immersive virtual environments based on object visibility and viewpoint quality," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces*, 2018, pp. 355–362.

[48] J. D'amelio, *Perspective Drawing Handbook*. Chelmsford, MA, USA: Courier Corporation, 2004.

[49] M. Claassens, "Interior photography: The ins and outs," Ph.D. dissertation, Faculty of Human Sciences, Central Univ. Technology, Bloemfontein, Free State, 1997.

[50] B. P. Krages, "The art of composition," New York, NY, USA: Allworth Communications, 2005.

[51] J. Webb, *Basics Creative Photography 01: Design Principles*. London, U.K.: Bloomsbury Publishing, 2017.

[52] I. Pantazopoulos and S. Tzafestas, "Occlusion culling algorithms: A comprehensive survey," *J. Intell. Robotic Syst.*, vol. 35, no. 2, pp. 123–156, 2002.

[53] D. Cohen-Or, G. Fibich, D. Halperin, and E. Zadicario, "Conservative visibility and strong occlusion for viewspace partitioning of densely occluded scenes," in *Comput. Graph. Forum*, vol. 17, no. 3, pp. 243–253, 1998.

[54] H. Weghorst, G. Hooper, and D. P. Greenberg, "Improved computational methods for ray tracing," *ACM Trans. Graph.*, vol. 3, no. 1, pp. 52–69, 1984.

[55] B. Gooch, E. Reinhard, C. Moulding, and P. Shirley, "Artistic composition for image creation," in *Proc. Eurographics Workshop Rendering Techn.*, 2001, pp. 83–88.

[56] B. Krages, *Photography: The Art of Composition*. New York, NY, USA: Simon and Schuster, 2012.

[57] H. Fu et al., "3D-front: 3D furnished rooms with layouts and semantics," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10933–10942.

[58] H. Fu et al., "3D-future: 3D furniture shape with texture," 2020, *arXiv:2009.09633*.

[59] ziroom.com, "Ziroom," Jun. 2022. [Online]. Available: https://www.ziroom.com/

[60] P. Nagarnaik and A. Thomas, "Survey on recommendation system methods," in *Proc. 2nd Int. Conf. Electron. Commun. Syst.*, 2015, pp. 1603–1608.

[61] H. R. Lewis, “Computers and intractability. A guide to the theory of NP-completeness,” *J. Symbolic Log.*, vol. 48, no. 2, pp. 498–500, 1983.

[62] H. S. Wilf and H. S. Wilf, *Algorithms and Complexity*, vol. 986, Englewood Cliffs, NJ, USA: Prentice-Hall, 1986.

[63] F. Rubin, “A search procedure for hamilton paths and circuits,” *J. ACM*, vol. 21, no. 4, pp. 576–580, 1974.

[64] D. Angluin and L. G. Valiant, “Fast probabilistic algorithms for hamiltonian circuits and matchings,” *J. Comput. Syst. Sci.*, vol. 18, no. 2, pp. 155–193, 1979.

[65] kujiale.com, “Kujiale,” Dec. 2020. [Online]. Available: https://www.kujiale.com/

[66] planner5d.com, “Planner5d,” Dec. 2020. [Online]. Available: https://planner5d.com/

[67] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, “Semantic scene completion from a single depth image,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1746–1754.

[68] A. Dai, M. Nießner, M. Zollöfer, S. Izadi, and C. Theobalt, “Bundlefusion: Real-time globally consistent 3D reconstruction using on-the-fly surface re-integration,” *ACM Trans. Graph.*, vol. 36, no. 3, pp. 1–18, 2017.

[69] T. Vu, K. Kim, T. M. Luu, X. T. Nguyen, and C. D. Yoo, “SoftGroup for 3D instance segmentation on 3D point clouds,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2698–2707.

[70] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep learning on point sets for 3D classification and segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.

[71] Z. J. Yew and G. H. Lee, “RPM-Net: Robust point matching using learned features,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11824–11833.

[72] M. Claassens, “Interior photography: The ins and outs,” Ph.D. dissertation, Faculty of Human Sciences, Central Univ. Technology, Bloemfontein, Free State, 1997.

[73] C. Marquardt, *Wide-Angle Photography*, San Rafael, CA, USA: Rocky Nook, 2018.

Shao-Kui Zhang received the BS degree in software engineering from Northeastern University, Shenyang, in 2018. He is currently working toward the PhD degree with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include computer graphics, 3D scene synthesis, and intelligent 3D scene interaction.

Hou Tam received the bachelor’s degree in computer science and technology from Tsinghua University, in 2021. He is currently working toward the master’s degree with the Department of Computer Science and Technology, Tsinghua University.

Yi-Xiao Li received the bachelor’s degree in arts & design from Tsinghua University, Beijing, in 2020. She is currently working toward the master’s degree with the Academy of Arts & Design, Tsinghua University, Beijing. Her research interests include human-computer interaction and virtual reality.

Tai-Jiang Mu received the bachelor’s and PhD degrees in computer science and technology from Tsinghua University, in 2011 and 2016, respectively. He is currently an assistant researcher with the Graphics and Geometric Computing Group in the Department of Computer Science and Technology, Tsinghua University. His research interests include computer graphics and visual media learning.

Song-Hai Zhang (Member, IEEE) received the PhD degree in computer science and technology from Tsinghua University, Beijing, in 2007. He is currently an Associate Professor with the Department of Computer Science and Technology, Tsinghua University. His research interests include computer graphics and virtual reality.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.