Functional Workspace Optimization via Learning Personal Preferences from Virtual Experiences

Wei Liang[1], Jingjing Liu[1], Yining Lang[1], Bing Ning[2], and Lap-Fai Yu[3]

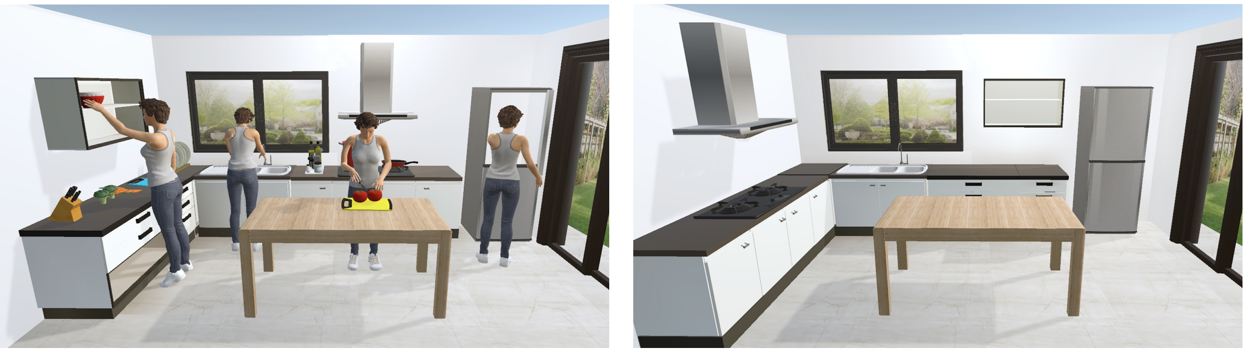


Fig. 1: Our approach learns the personal preferences of using a functional workspace by analyzing a user performing given tasks via VR devices (left), e.g., making a salad in the kitchen. The learned preferences are applied to optimizing the workspace, which results in an updated layout that fits the user's preferences better (right).

Abstract—The functionality of a workspace is one of the most important considerations in both virtual world design and interior design. To offer appropriate functionality to the user, designers usually take some general rules into account, e.g., general workflow and average stature of users, which are summarized from the population statistics. Yet, such general rules cannot reflect the personal preferences of a single individual, which vary from person to person. In this paper, we intend to optimize a functional workspace according to the personal preferences of the specific individual who will use it. We come up with an approach to learn the individual's personal preferences from his activities while using a virtual version of the workspace via virtual reality devices. Then, we construct a cost function, which incorporates personal preferences, spatial constraints, pose assessments, and visual field. At last, the cost function is optimized to achieve an optimal layout. To evaluate the approach, we experimented with different settings. The results of the user study show that the workspaces updated in this way better fit the users.

Index Terms—Affordance, Human-centered Design, Virtual Environments, Workspace Design, Remodeling.

1 Introduction

Scene design is in high demand in the virtual world and in real life. As a unique kind of scene, a workspace has many factors to be considered during the design process, including those related to functionality and ergonomics. In designing a convenient and comfortable workspace to work in, some rules derived from population statistics, are usually incorporated by designers. However, some personal preferences are difficult to satisfy, e.g., different work-flows from different users, different body proportions, or the preferences of placing objects. For example, one person might want to place a tea table on the left side of a sofa in a living room instead of on the right side. Although both arrangements of the tea table (on the right or the left) are reasonable, people's experiences in these two scenarios are different. Some may feel that the placement is convenient, but others do not. This is due to the diversity of personal preferences.

To exploit a user's preferences and design a workspace to serve a user well, designers usually observe or communicate with the user about his daily activities. Experienced designers can summarize the user's preferences from his daily activities and make use of this information to adjust and adapt the design to the specific user. The processes of observing and communicating are usually time-consuming. The effectiveness of the process also depends greatly on the experience of the designer.

Most recently, some automatic scene synthesis approaches [12,17,41,52] have been proposed to generate reasonable scenes, which consider aesthetics, functionality, and general human activities. We attempt to address the problem of remodeling the design of a workspace in a different way, by adapting it to an individual user, incorporating the users preferences in using the workspace. We design a pipeline to learn the user's personal preferences. When a designer completes the design of a workspace for a user, it is convenient to generate a corresponding virtual scene to animate the design by some automatic approaches [32,49]. Then the user can try out the design in a virtual scene via VR devices. During the trial, our approach captures the activities of the user and models his personal preferences, which are later used to optimize the initial design later.

To learn the personal preferences, we design some tasks that are typical for the desired workspace, e.g., making a salad, frying steak, or making soup in a kitchen. The user wears the VR devices (HTC VIVE and VIVE trackers) and enters an initial virtual workspace, in which the user performs the tasks. We capture the user's activities through VR devices. Then, a cost function, which incorporates personal preferences, spatial constrains, pose assessments, and visual field, is formulated to measure how good a layout configuration is for the user.

1. *Wei Liang, Jingjing Liu, and Yining Lang are with the Beijing Institute of Technology. E-mail: liangwei@bit.edu.cn, 3120181009@bit.edu.cn, and lucky@langyining.com.*

2. *Bing Ning is with the Beijing Institute of Fashion Technology. E-mail: ningbing@bift.edu.cn.*

3. *Lap-Fai Yu is with the George Mason University. E-mail: craigyu@gmu.edu.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

An optimization algorithm is applied to search for the optimal solution, resulting in the desired layout.

The main point of our work is to capture human activities and learn personal preferences using consumer-grade VR devices. By applying our approach, an interior design may be improved by adapting it to an individual user according to the learned preferences. Furthermore, our approach may also help some virtual world games, e.g., Job Simulator [24], whose gameplay takes place in indoor scenes, by providing customized scene layouts for each gamer.

In this paper, we take a kitchen as an example workspace to demonstrate our approach because of the diverse functions it serves in different regions. Moreover, we also demonstrate our approach to other workspace scenes in the experiment.

The main contributions of this paper are as follows:

- Propose a pipeline to remodel a workspace by learning the personal preferences from human activities through VR devices.
- Define a cost function to model the personal preferences, spatial constraints, pose assessments, and visual field. The cost function is optimized to develop the remodeled workspace.
- Conduct experiments and perceptual studies on different settings to validate the effectiveness of the approach.

2 Related Work

In this section, we review the previous research in the areas of traditional interior design, automatic layout synthesis, and posture assessment, which are highly related to our problem.

2.1 Traditional Interior Design

In the field of traditional interior design and decoration, there are some basic rules, design standards, and design rationale followed by designers, such as work triangles [18] and relationships between the user and the internal space [19]. Despite the common design criteria, some design approaches tend to explore the characteristics of the individual user, such as user-centered design [23], using questionnaires, face-to-face interviews, and focus groups [1], and so on. Generally, these methods are time-consuming in practice, and the designer's work schedule may be too overloaded.

Nowadays, some commercial software has been developed to help professional designers or general users to design and visualize a design scheme, thus improving the user experience and speeding up the design process. Floorplans [46] is a building floor drawing software. Using this tool, the user can calculate the size of a house accurately in a few seconds. An accurate size calculation for a house helps the user to estimate whether a furniture is suitable for a room before buying it. Sweet Home 3D [13] is an interior design software that helps the user to design and decorate furnitures using a two-dimensional home plane. The user can also preview the entire layout from a 3D perspective. Some software, e.g., Planner 5D [26], can be used to design rooms down to the smallest architectural details, including windows, stairs, and partitions. The software is mainly used to visualize and design the space more realistically and has convenient interfaces that enable users to interact naturally with the designed scenes.

Compared with the traditional interior design approaches, our approach mainly explores the automatic generation of a scene, considering personal preferences.

2.2 Automatic Layout Synthesis

Synthesizing indoor scenes to generate a reasonable layout has practical applications. Some works applied object-related constrains to suggest the arrangements of the given furniture [2, 9, 17, 44, 50]. Kjlaas et al. [22] proposed to automate the process of generating large furnished models of building interiors. The approach selected a template and adapted it to a given room, recursively resolving spatial conflicts in the room template subspaces until a valid furniture configuration was obtained. Genetic algorithms are also used to solve constrained object layout problems [42,51]. Yu et al. [52] employed the relations learned from 3D scene datasets to optimize the layout of a room through an MCMC optimizer. Merrell et al. [35] established layout based on the predefined guidelines from the actual experience of interior designers. The scene understanding techniques [29,30] can help to provide the priorities for the scene and improve the synthesized results.

Considering human factors for scene synthesis has become increasingly popular since the purpose of the synthesized scenes is to support human activities. Fisher et al. [10] synthesized 3D scenes that allow the same activities as real environments represented as noisy and incomplete 3D scans. Qi et al. [41] learned the layout distributions from an indoor scene dataset and sampled it to generate new layouts using Monte Carlo Markov Chain. Human contexts were encoded as contextual relations.

Different from the previous works, which employed predefined rules or learned rules from general human activities, we focus on learning and applying an individual user's preferences to improving an initial layout. The VR devices enable our approach to capture the user's activities and learn their personal preferences [3,8,43]. Driven by the personal preferences, the optimizer searches for an updated layout to fit the individual user better.

2.3 Posture Assessment

When a user works in a workspace, the postures he assumes while working will influence the amount of fatigue experienced. Some postures are determined by the workspace. For example, when a user sits on a chair and writes, his posture depends greatly on the height of the table and the chair. Studying the postures used in a workspace may help to design a reasonable workspace [7].

Many studies have been conducted to model the relation between the comfort of a user and his postures. Some postural analysis techniques have two, usually contradictory, qualities of generality and sensitivity [11]. In a postural analysis study high generality may be compensated for with low sensitivity. The Ovako Working Posture Analysis System [20] had a wide range of use but the results can lack detail. In contrast, NIOSH [45] required detailed information about specific parameters of the postures to achieve a high sensitivity with respect to the defined indices. Priel et al. [40] proposed "Posturegram" to evaluate postures using score cards. It took a few minutes to record and score one gesture. Thus it cannot be applied to the dynamic activities. The increasingly widespread use of VR devices enables a system to conveniently capture dynamic motion. Compared to the traditional motion capture systems [31,38,48], VR devices have the advantage of rapid and accurate data capturing and are also more fun to use. [25,27,36]. We apply the latest VIVE tracker and the Inverse Kinematics algorithm [5] to estimate human postures, based on the features of high precision and low system latency [39].

Some posture assessment methods are proposed in the field of ergonomics. Kemmlert et al. [21] proposed PLIBEL to determine and to identify musculoskeletal tension factors, which were judged by the related user self-evaluation of related questions. The QEC [28] used a combination of observer observation and self-assessment by the observer without considering the lower posture when the observer observed it. RULA [34] was an operation-related upper limb posture assessment method based on an evaluation form that also did not take the lower posture into consideration either. In our approach, we used the REBA [14] to evaluate posture. This is a rapid assessment method for whole body posture that has a combination of upper limb, lower limb, and whole body posture assessment tables to evaluate posture.

3 Overview

We represent a workspace as a layout configuration of it, which is denoted as L. Assume that the workspace consists of N components. For the ith component, we consider two attributes: position (x_i, y_i, z_i) and orientation o_i. To simplify the problem, when the component is located on the floor, e.g., a base cabinet, the position represents the center of the top surface. When the component is hung on a wall, e.g., a shelf, the position represents the center of the bottom surface. The rotation on the floor plane is represented by o_i. Accordingly, $L = \{l_i | l_i = (x_i, y_i, z_i, o_i)\}$, where $i \in \{1, 2, \cdots, N\}$.

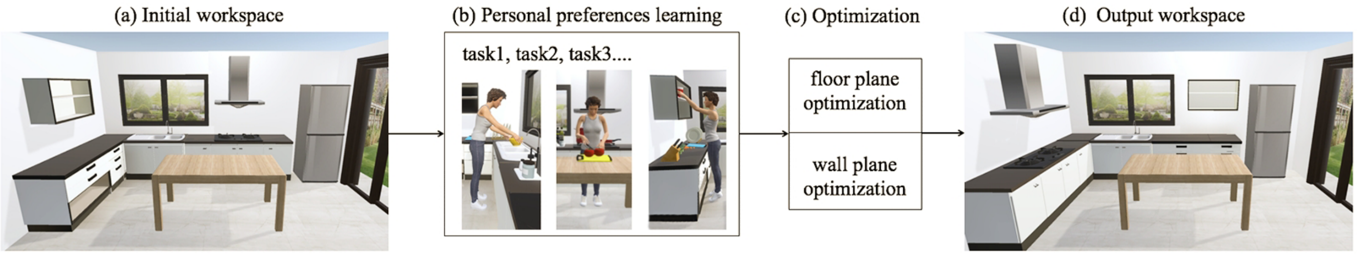


Fig. 2: Overview. Given an initial workspace (a), a user interacts in it and performs some tasks through the HTC VIVE devices, e.g., cooking (b). Through the floor plane optimization and the wall plane optimization (c), we obtain an updated workspace (d).

Given an initial workspace, our approach aims to find an optimal configuration L^* based on the learned personal preferences. The personal preferences refer to how a user uses the workspace, including the preferences on workflow and object storage as well as the user's attributes (i.e., stature and body proportions). It is worth noting that the initial workspace can be configured by referring to the similar scenes in the IKEA Kitchen and can also be initialized by some automatic scene synthesis approaches e.g., [12,52].

The framework is shown in Fig. 2. Our approach includes two stages: personal preferences learning and workspace optimization.

Personal preferences learning. A user enters the initial virtual workspace via the HTC VIVE apparatus. He is given some tasks to perform in the virtual world. To accomplish the tasks, the user navigates in and interacts with the virtual workspace. His activities are captured, based on which the personal preferences are learned. The personal preferences are then used to optimize the initial workspace. We will discuss the details in Sec. 4.

Workspace optimization. Considering the independence between the floor plane and the wall plane, we decompose the optimization into two parts: floor plane optimization and wall plane optimization. The floor plane optimization focuses on optimizing the location of each component along the floor plane, i.e., (x_i, y_i, o_i), which is discussed in Sec. 5. The wall plane optimization focuses on the height of each component, i.e.,z_i, which is discussed in Sec. 6.

4 Learning Personal Preferences

The level of functionality of a workspace usually depends on a users personal preferences of using it. The preferences can be reflected by the user's activities. In this section, we discuss how to learn such personal preferences for an individual user through the observation of his activities.

Our approach defines a set of tasks in the virtual workspace that are typical for the kind of workspace. A user enters the initialized virtual workspace by wearing the HTC VIVE and HTC trackers. He is given the task instructions and is required to perform the tasks one at a time. During this process, all parameters from the HTC VIVE trackers are captured and are used to model the personal preferences of the user.

We take the kitchen as an example workspace to demonstrate our approach. More results of other workspaces are shown in Sec. 7.4.

4.1 Task

In the kitchen scenario, we define three tasks: make a salad, fry a steak, and make soup. For the user, there are three steps to accomplish each task:

(1) Receiving instructions. The user will see an information board in front of him. On the board, the goal of the task and the overall action steps are shown. More detailed information will be given step by step during the user's interaction.

(2) Setting up. In this step, the user is required to place all objects that will be used in the task somewhere in the workspace that he likes. The objects are shown by a menu. The user can use a controller to select and place the objects. We do not set any constraint in this step since we also want to learn the storage preferences of the user. Later, the user may retrieve the stored objects through the menu.

(3) Acting. The user executes the task instructions step by step. It is worth noting that the steps given to the user are abstract and without many details. For example, when the task is frying steak, the instructions are: take out the steak, soak with soy sauce, add pepper and olive oil, and so on. The instructions will not tell the user to take out a bowl, put the bowl on the table, and then go to get the steak out. The reason is that our approach tends to give more flexibility to the user in order to discover his preferences.

After accomplishing one task, the workspace will be cleaned up and reset automatically. The user then turns to the next task and repeats the above three steps, until all tasks are accomplished.

4.2 Human Activity Capture

During the virtual experiences, we want to know the activities of the user at each frame, i.e., location, pose, and which component the user is interacting with.

We provide an HTC VIVE headset, two controllers, and three trackers to the user to navigate and accomplish tasks within the virtual environment. Considering that carrying extra sensors will influence the user's experience, we only only use three extra trackers as suggested [16]. With these devices, the user can control his viewpoint and interact with virtual objects. The headset is worn on the head. The two controllers are held in the hands. The three trackers are attached on the waist and the feet.

Each sensor provides six parameters of itself (three for translation and three for orientation). Using these parameters, we apply the Inverse Kinematics algorithm, which infers the positions of the other joints of the user by the Jacobian inverse technique in real time [5], to obtain the whole pose. A pose includes 16 points, and each point has six degrees of freedom.

Location. The location of the user is regarded as equivalent to the location of the waist. Therefore, we take the coordinate of the waist as the location of the user. During all tasks, the location sequence of the user is written as $H = \{h_1, h_2, \cdots\}$, where h_t is the set of 3D coordinate of the waist at time t.

Interaction. During all tasks, we record the index of the component that the user interacts with at each time. The index sequence is written as $I = \{i_1, i_2, \cdots\}$, where i_t is the index of the component.

Load. We keep a sequence of object mass changes to record when the user transfers objects from one component to another. The sequence is written as $M = \{m_1, m_2, \cdots\}$. We define the mass of each object according to its weight in real life.

Pose. To analyze the user's poses, we record the translation and the orientation for the 16 points.

It is worth noting that we do not use a camera, e.g., Kinect to capture the human activities. The reason is that the accuracy of the camera capturing depends greatly on the viewpoint. When the user has their back or side to the camera, the pose cannot be captured correctly due occlusion, which often happens when navigating in virtual scenes.

4.3 Personal Preferences

We model the personal preferences that influence a user's experience in a workspace. According to such a model, our approach optimizes an initial scene to fit a specific person better. We consider three types of personal preferences, including the preferences on the workflow, on the object storage and the user's attributes (i.e., stature and body proportions).

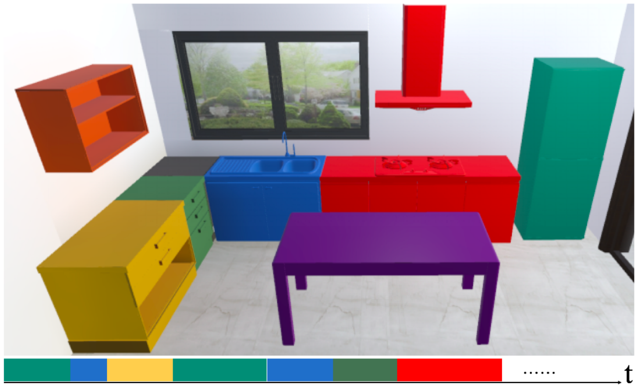


Fig. 3: An example of personal preferences. Each colored bar depicts the user interacting with the component with the same color in the scene. The ordered sequence of the components represents the personal preferences for the workflow.

Preferences on workflow. The preferences on the workflow are reflected by the captured activities even though the activities are noisy and redundant. For example, when the user manipulates one component, his location varies around this component.

A concise summary is that we cluster the user's activities along the time axis according to what component the user interacts with. Once the interaction subject is changed, the activity category changes accordingly. As shown in Fig. 3, each bar with different colors depicts the component with which the user interacts over time. We use the index of the indicated component to represent the clustered category.

After clustering, the location sequence of the user H is represented by an ordered index sequence of the components that the user visits chronologically $P = \{p_1, p_2, \cdots, p_Q\}$, Q is the last cluster of p_k, and $p_k \in \{1, 2, \cdots, N\}$ is the index of the component in the virtual scene. It reflects the user's preferences of using each component.

Preferences on objects storage. The user preferences on the object storage are captured in the setting up process, described in Sec. 4.1. After the setting up, when the user retrieves the corresponding objects to achieve the task, the preferences are reflected in the workflow. For example, in Fig. 3, if the user chose to place the plates into the orange cabinet, he may frequently transfer between that cabinet and the sink. During the optimization, the optimizer will tend to place the shelf and the sink as close together as possible. The shorter distance save the user energy when transferring between them.

User's attributes. To capture the user's attributes, when a user enters the virtual workspace, there is an initialization process. The user is required to stand with a natural posture. Then, we recorded all the parameters of the sensors. Using these parameters, we apply the IK algorithm to infer the other parameters. The natural pose is denoted as θ_0 and will be used as one of the user's attributes to optimize the workspace.

5 Floor Plane Optimization

In this section, we discuss the floor plane optimization, i.e. optimizing the location of each component along the floor plane (x_i, y_i) and the orientation (o_i).

Each component in a workspace plays a role in affording functionality. Different users may expend their efforts differently in the same scene because of their different personal preferences of using the functional component. For example, when accomplishing the task of frying steak, a user is given the instructions as getting out a steak from the refrigerator, soaking it, frying and so on. Different users may have different workflows in detail. When accomplishing the first instruction, one user may follow the order of taking out a plate, putting it on the countertop, getting the steak, cleaning it, and putting the steak on the plate. Another user may follow a different order such as getting the steak, cleaning it, taking out the plate, and putting the steak on the plate. In addition, a person may move among the components, such as the refrigerator, countertop, and oven while carrying different virtual objects, e.g., pan, knife, steak and so on.

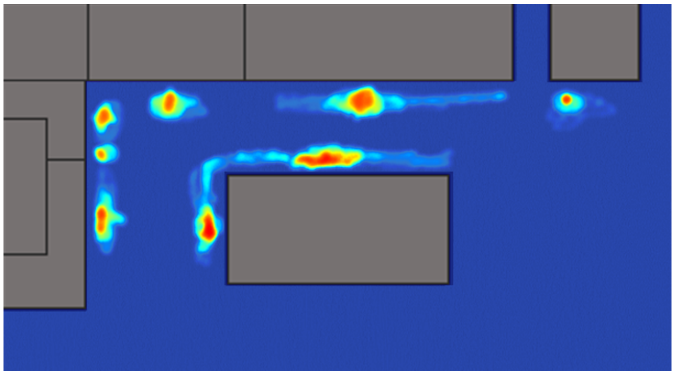


Fig. 4: The position distribution when the user interacts with the corresponding component. We use a heatmap to visualize the visited times at each point when the user interacts with the corresponding component. The redder the color is, the more times the user visited that point.

We attempt to optimize the layout by analyzing these transitions so that the user expends the least effort to achieve the tasks in the updated layout. We define a cost function to consider the personal preferences and spatial constraints as follows:

$$C_{\text{floor}}(P, L) = (1 - \lambda_{\text{s}}) C_{\text{p}}(P, L) + \lambda_{\text{s}} C_{\text{s}}(L). \tag{1}$$

$C_{\text{p}}(P, L)$ is the personal preference cost to penalize those solutions in which the user will expend more energy to complete the given tasks according to his personal preferences. $C_{\text{s}}(L)$ is the spatial cost which constrains the workspace from the perspective of the rationality of the layout, including the accessibility of each component and the free space of the scene. The weight λ_{s} is set as 0.5 experimentally.

5.1 Personal Preference Cost

We consider the travelled distance and the loads carried by the user when he moves around to define a weighted distance as the preference cost:

$$C_{\text{p}}(P, L) = \frac{1}{K} \sum_{k} \frac{D(p_k, p_{k+1})}{D_{\max}} \frac{m_k}{M_{\max}}, \tag{2}$$

where $D(p_k, p_{k+1})$ is the distance between two adjacent visited components along the time axis, whose indexes are p_k and p_{k+1} respectively. The adjacent relation is defined by the preferences on the workflow, learned form the user activities. m_k is the mass of the object the user carries from p_k to p_{k+1}, which is captured during the personal preferences learning process. $D_{\max}$ is the max distance between two components in the scene. $M_{\max}$ is the max mass of all given objects. K is the number of the elements in preferences on the workflow, i.e., P.

Although the user's locations in the initial workspace are obtained, we cannot use the original locations to calculate the cost $D(\cdot)$ for a new scene (i.e., L) directly. The reason is that the distance will change with the layout L changing. Therefore, we simulate a new user's location sequence H' by imagining how the user will work in the updated scene with the layout L.

Given the learned personal preferences P, we generate a location sequence with K points $H' = \{h'_1, h'_2, \cdots\}$. h'_k represents a possible location where the user interacts with the k-th component. From the user's original locations H, we estimate a distribution of the frequency of the visited times at each point when the user interacts with the p_k-th component. Fig. 4 shows an example. The redder region represents that the points are visited more frequently. In the implementation, we take out the k-th and $k+1$-th element in P. p_k and p_{k+1} are the index of the components. Then we sample from the distribution to generate the simulated h'_k and h'_{k+1}. $D(p_k, p_{k+1})$ is calculated as the Euclidean distance of the two locations h'_k and h'_{k+1}.

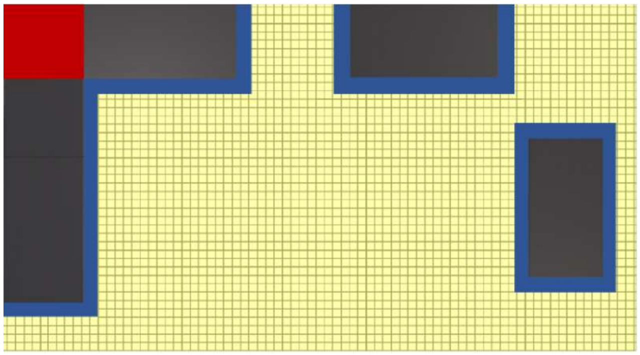


Fig. 5: Free space cost. The red region is unreachable. The blue region is the extended area around a component to ensure a person can walk through. The yellow region is the free space and is used to calculate the free space cost by counting grids.

5.2 Spatial Cost

We define the spatial cost as follows:

$$C_{\mathrm{s}}(L) = (1-\lambda_{\mathrm{f}})C_{\mathrm{a}}(L) + \lambda_{\mathrm{f}}C_{\mathrm{f}}(L), \tag{3}$$

where $\mathrm{C_a}(L)$ is the accessibility cost and $\mathrm{C_f}(L)$ is the free space cost. λ_{f} is the weight, and set as 0.4 experimentally.

Accessibility. In a workspace, each functional component should be accessible to maintain its functionality [4, 37]. We define a collision area in front of each component. To favor the accessibility, the cost increases when a component moves into the accessible space of any other one. Thus, the cost penalty will ensure that there is enough space for a person to interact with the functional component, e.g., opening a cupboard door. The cost is defined as:

$$C_{\mathrm{a}}(L) = \frac{1}{2N}\sum_{i}\sum_{m}\max(0, 1-\frac{2e_{i,m}}{d_i+d_m}), \tag{4}$$

where d_i and d_m are the diagonal length of the component i and m respectively. $e_{i,m}$ represents the distance between the center of the component i and m, calculated by using (x_i, y_i, z_i) and (x_m, y_m, z_m). In the implementation, we set a collision detection area for each component. If one component enters another component's collision area, the cost function is activated.

Free space. Another aspect that is considered in workspace design is the size of the free space. Except for those spaces that are occupied by functional components and the necessary spaces to access each component, a person usually prefers to have more free space. That will make a place look like more spacious. Therefore, we also use free space where a human can reach to increase the accessibility of a scene and calculate it from an overhead view.

We illustrate the calculation process in Fig. 5. Technically, we first detect regions in the scene which are unreachable. Starting from the door, our approach detects the connected region. The connected region is a path with a certain width ($50cm$ in our experiments) which allows a person to walk through. Those unreachable regions are marked in red. Second, we extend the area around each component to ensure that a person can walk through. These regions are labeled as blue. Finally, we put a grid on the remaining regions, (the yellow regions) which are available, and count the number of grids. The cost is formulated as:

$$C_{\mathrm{f}}(L) = 1 - \frac{1}{G}\sum \mathbb{1}_{grid}, \tag{5}$$

where $\mathbb{1}_{grid}$ is an indicator function, representing the available grids. G is the normalization parameter, which is the number of all grids, regardless of whether it is free space or not. It is worth noting that we also consider the wall components. The region which is covered by the projection of a wall component on the ground is also regarded as occupied. This setting will make the optimizer prefer to choose those configurations, in which the floor components have more overlaps with the projects of the components on the wall. That will make more free space available.

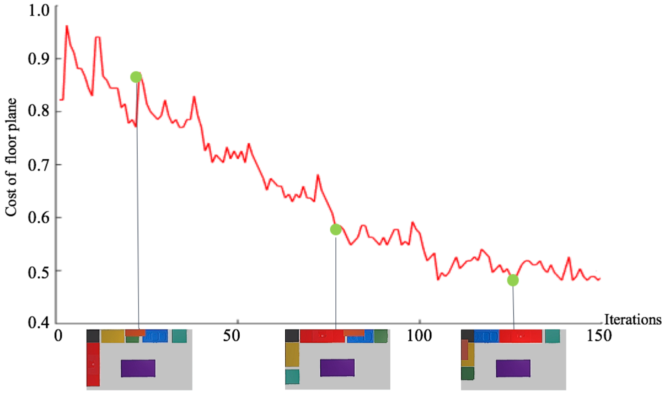


Fig. 6: An example of the optimization. The curve represents the cost change during the optimization. We also visualize three intermediate configurations on the bottom.

5.3 Optimization

To search for a good layout configuration efficiently, our approach applies a simulated annealing algorithm. Given an initial configuration of a workspace L_0, our approach follows two steps iteratively to explore the solution space:

(1) Propose a move $L^i \rightarrow L^*$.

(2) Accept or reject the proposed move based on Metropolis-Hastings acceptance rule:

$$A(L^i, L^*) = \min\left\{1, \frac{p(L^*)}{p(L^i)}\right\}, \tag{6}$$

$p(\cdot)$ is computed using by the defined cost:

$$p(\cdot) = \frac{1}{Z}\exp\frac{-C_{\mathrm{floor}(\cdot)}}{T}, \tag{7}$$

where $C_{\mathrm{floor}(\cdot)}$ is defined in Sections 5.1 and 5.2. T is the temperature of the annealing process. At the beginning of optimization, T is set to a large value, which allows the sampler to more aggressively explore the solution space. Then, T gradually decreases throughout the optimization. Near the end, T attains a small value near zero, thereby allowing the sampler to refine the solution. Z is a constant for normalization. By default, we empirically set T to 1.0 and decrease it by 0.05 every 10 iterations until it reaches zero. We terminate the optimization if the absolute change in the total cost value is less than 5% over the past 20 iterations. In our experiments, a full optimization takes approximately $100 \sim 150$ iterations.

We use three strategies probabilistically to introduce solution changes during the optimization:

Translation. Translation is the basic operation of the optimization that modifies the position of components in a workspace. A component i is selected and its location is updated by a move. Mathematically, we define a linear operation to accomplish the translation move denoted as $(x_i + \Delta x, y_i + \Delta y)$. We sample the variation $(\Delta x, \Delta y)$ using a Gaussian distribution.

In practice, we handle translations differently depending on whether the component is on the ground or attached on the wall. For the floor components, we only modify (x_i, y_i). For the wall components, we modify (y_i, z_i) or (x_i, z_i) according to which wall it is attached to.

Rotation. Rotation works on the orientation of a component. A component i is selected, and its orientation is updated with a rotation change $(o_i + \Delta o)$. The rotation change Δo is generated randomly from a uniform distribution of $\{0°, 90°, 180°, 270°\}$. To ensure a more stable optimization, we alternately modify rotation and translation alternately.

Swapping. To enable more rapid exploration of the arrangement space and avoid becoming stuck in local minima, swapping components spatially in the workspace is proposed. Two components i and j are randomly selected, and their positions are swapped. The swapping operation accelerates the exploration of the solution space. It is worth noting that we ensure some components are grouped together, e.g., the range hood and the cooktop. When selected to be translated or swapped, the grouped components move together.

6 Wall Plane Optimization

In this section, we optimize the height z_i of the component. We consider two factors: (1) pose assessment, which measures how comfortable the user feels when he works with the component, given its height; and (2) visual field, which measures how much of a component the user can see, given the components height. The cost function is defined as:

$$C_{\mathrm{wall}}(L,\Theta) = (1-\lambda_{\mathrm{v}})C_{\mathrm{r}}(L,\Theta)+\lambda_{\mathrm{v}}C_{\mathrm{v}}(L,\Theta). \tag{8}$$

$\mathrm{C_r}(L,\Theta)$ is the pose assessment cost. We generate a pose sequence $\Theta=\{\theta_1,\theta_2,\cdots\}$ by imagining what poses the user will take in the scene, where θ_i is the pose which the user takes when interacting with the ith component. Then the generated sequence is used to measure the cost $\mathrm{C_r}(L,\Theta)$. We will discuss this further in Sec. 6.1. The weight λ_{v} is set as 0.4.

$\mathrm{C_v}(L,\Theta)$ is the visual field cost. It influences the height of the wall component. This term penalizes those solutions in which the users sight is occluded. It ensures that the user can comfortably see the workspace as well as the contents in the wall components. We will discuss this further in Sec. 6.2.

6.1 Pose Cost

We refer to the REBA score [14] to define the post cost. REBA is a posture analysis tool, that evaluates each pose by considering the risk of musculoskeletal injury associated with the posture. The REBA score is obtained from a lookup table according to the angles of 12 points of a pose, including that of the neck, trunk, two legs, two upper arms, two lower arms, two wrists, and two feet. A high score indicates that the posture is more likely to make the user feel tired and has a higher risk of causing injury. The cost is written as:

$$\mathrm{C_r}(L,\Theta) = \frac{1}{N}\sum_i \frac{\mathrm{R}(\theta_i)}{12}, \tag{9}$$

where N is the number of all components. $\mathrm{R}(\theta_i)$ is the REBA score of the user's pose θ_i when using the component i. The highest score in REBA is 12, which is used to normalize the score of one pose to (0,1). Since we cannot use the posture captured directly from the sensors, we generate a pose sequence by assigning some key point positions in the pose, including the position of the hands, feet, waist, and head.

For the hand positions, we use the position of the ith component (x_i,y_i,z_i), which is the top surface center of a floor component or the bottom surface center of a wall component, as the hand locations. It mimics the normal positions of the user's hands when working on a countertop or wall component.

For the feet positions, we assume that the user's feet are always on the floor all the time since any posture with the tiptoe will increase the cost according to the REBA score. According to all positions where the user stand at when he interacts with ith component, we calculate a average distance $\bar{d}$ from this component to the user. Then according to the orientation of the component, the feet location is assigned as $(x_i+\bar{d},y_i)$ when $o_i=0°,180°$ or $(x_i,y_i+\bar{d})$ when $o_i=90°,270°$. z_i is obtained from the two trackers on the feet when the user takes a natural pose in the initial process. The waist and head position are obtained with a similar strategy.

By those key points(two hands, two feet, waist, and head), we apply the Inverse Kinematics algorithm to generate the whole pose θ_i for the component i.

6.2 Visual Field Cost

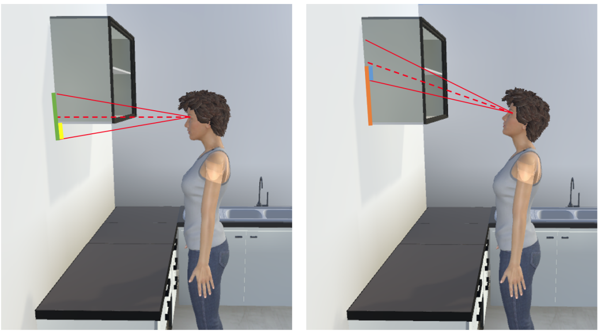


Fig. 7: An example of the visual field cost. The green and orange lines indicate the range covered by the visual field. The yellow and blue lines indicate the actual range covered by the current visual field.

Considering that the user wants to see the work area without occlusion, and he also wants to see most contents in the wall components comfortably, we define the visual field cost as:

$$\mathrm{C_v}(L,\Theta) = 1-\frac{1}{2N}\sum_i\left(\frac{\Delta s_i}{s_i}+\frac{\Delta u_i}{u_i}\right). \tag{10}$$

N is the number of components. s_i is the length that is covered by the user's visual field when the user looks horizontally and there is no occlusion by other components. To calculate s_i, we analyze a standing pose as in Sec. 6.1. Then we adjust the head angle as horizontal. We suppose that the valid visual field of a person is 15°. We project a circular cone on the wall the user is facing and calculate the projected lines' lengths along the wall as s_i, as shown as a green color line in Fig. 7. u_i is the length which is desired to be covered by the user's visual field when the user looks up, as shown with the orange color line in Fig. 7. We set u_i equals to the half height of the wall component as default. This means that our approach encourages those solutions in which the user can see more contents in the lower part of the wall component. The default value can be changed according to the user's requirements.

To estimate Δs_i and Δu_i, we suppose that the valid visual field of a person is 15°. A similar method as in Sec. 6.1 is used to estimate the user's standing pose. Δs_i is the actual length which is covered by the user's visual field when he looks horizontally at the current configuration of (L,Θ), as shown as a yellow line in Fig. 7. We obtain Δs_i by projecting a circular cone and calculating the height of the projected line without occlusions. If the visual field is blocked by another component, the cost will increase. Δu_i is the actual length of the wall component which can be seen by the user. We adjust the head angle from -20° to 20° by 1° interval and calculate all possible values of Δu_i. According to the REBA rule, this is a regular range for the head motion. During the adjustment, we calculate the visual field with the current head angle. The blue line in Fig. 7 is one example of Δu_i. Finally, we take the largest value as Δu_i.

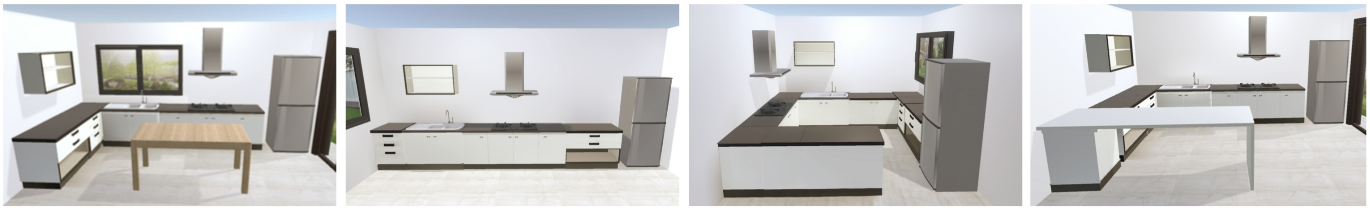
6.3 Optimization

The height of the floor component is set in the range of $65cm$ to $115cm$, and the height of the wall component is set in the range of $140cm$ to $190cm$. To meet the uniformity of the workspace, we use a uniform height for all the components on the floor and wall. The height range is further discretized by $0.1cm$. Since the solution space is not too large $(50*50=2500)$, we use an exhaustive search to determine the optimal height. In the implementation, we enumerate all possible heights. For each pair of heights, we calculate the cost by the Eq. 8. Then we take the solution with the lowest cost as a result.

7 Experiments

We implemented our approach using C# and Unity 5.6 and ran the optimization approach on a PC equipped with 32GB of RAM, a Nvidia Titan X graphics card with 12GB of memory, and a 2.60GHz Intel i7- 5820K processor. The participants worked in a $3m \times 4m$ virtual workspace via the HTC VIVE device (the maximum recommended room size of the device).

Participants. 34 participants were recruited who were unaware of the purpose of the user study. The participants included 19 males and 15 females whose ages ranged from 18 to 50. All the subjects



Scene 1 Scene 2 Scene 3 Scene 4

Fig. 8: Four initial scenes in the user study.

reported normal or corrected-to-normal vision with no color-blindness. 18 subjects reported that they did not have any experience in using VR devices. 11 subjects reported that they cooked every week, and 12 subjects reported that they cooked every month, the other 11 subjects cooked seldomly.

Procedure. First, the participant familiarized themselves with the layout of the initial scene, the tasks, and the operations of the VR devices which took 5 minutes on average.

Second, the participant entered the virtual workspace and received the instructions about the task. Each task took approximately 10 minutes. When one task was accomplished, the scene was reset automatically. Our approach captured the user's activities when he achieved the given tasks. Then, the optimization algorithm was applied to synthesize an updated layout.

After the optimization, the participant was required to interact in the optimized scene and other scenes generated by other approaches that we compared with our approach. Then, we asked the participant to score all scenes. The score range was from 0 to 10. Considering that using VR devices for a long time may make the user feel tired, we carried out the evaluation part in the next day.

We performed the user study on four scenes, shown in Fig. 8, which were created based on traditional kitchen layouts (L-shaped, U-shaped, and one-wall-shaped). The numbers of components in each scene were 6, 5, 7, and 6, respectively.

7.1 Comparison with other approaches

We compare the evaluation results of our approach with the results of the other two approaches. The approaches compared consist of the following:

Our Approach: The workspaces were optimized by our approach after learning the users' personal preferences.

Designer Approach: The workspaces were manually created by a designer who had approximately 3 years experience with interior design. He communicated with each participant and then modified the initial layout according to the communication.

User Approach: The workspaces were manually created by the participants before they experienced in the virtual scene. The participants observed the scene on a computer and were allowed to modify the layout or the component's height as they liked.

We define three metrics to investigate the comparison: (1) convenience, to evaluate how comfortable the user feels during his transition among the components; (2) height, to evaluate how comfortable the user feels about the height of the floor components and the wall components; and(3) overall, to evaluate the overall feeling about the workspace. The score range of each metric is from 0 to 10. We instructed all participants about the metrics and encouraged them to ask questions. During the evaluation, the order of the scenes generated by the different approaches is random. We also recorded the processing time for each approach during the experiment.

24 participants took part in the comparison experiments. They were divided into four groups, and each group was randomly assigned to an initial workspace. Fig. 9 shows the rating results for the three approaches. The blue, orange and green boxes depict the ratings for our approach, the Designer approach and the User approach respectively.

In terms of the convenience rating, our approach obtained the highest score ($M = 6.50$, $SD = 1.47$), followed by the approach of Designer ($M = 6.29$, $SD = 1.24$) and User ($M = 4.79$, $SD = 1.08$). The ratings of our approach and the Designer approaches on convenience are close. However, the Designer approach rating has less diversity.

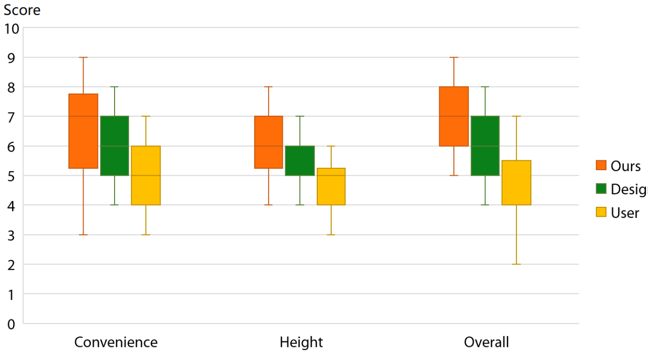


Fig. 9: The box plots of the user rating for the scenes generated by three approaches. The bottom and top edges of the boxes indicate the 25th and 75th percentiles, respectively. The horizontal lines indicate the median rating. The whiskers extend to the most extreme data points.

In terms of the height rating, our approach ($M = 6.29$, $SD = 1.17$) outperforms the approach of Designer ($M = 5.67$, $SD = 1.21$) and the User approach ($M = 4.75$, $SD = 1.09$). We collected the responses about the height settings from the Designer and the User approaches. The designer said he adjusted the component height according to the user's height. 13 of the participants reported they used similar heights with their kitchens. 7 of them reported lowering the heights according to their kitchens. 4 of them reported estimating according to their experiences. We believe that the body proportions captured by the sensors may help our approach to optimize the heights of the components to fit the individual user better.

In terms of the overall rating, our approach ($M = 7.21$, $SD = 1.12$) also performs better than the Designer approach ($M = 6.58$, $SD = 1.04$) and User approach ($M = 4.63$, $SD = 1.18$). It shows that the learned personal preferences may help to generate a personalized layout.

We also performed an ANOVA test on the convenience, height and the overall rating at the $\alpha = 0.05$ significance level. The results were statistically significant between the Our approach and the User approach (convenience: ($F_{[1,23]} = 20.15$, $p < .05$), height: ($F_{[1,23]} = 21.34$, $p < .05$), overall: ($F_{[1,23]} = 57.94$, $p < .05$)); the Designer approach and the User approach (convenience: ($F_{[1,23]} = 19.13$, $p < .05$), height: ($F_{[1,23]} = 7.27$, $p < .05$), overall: ($F_{[1,23]} = 35.60$, $p < .05$)).

The results did not show any statistical significant between our approach and the Designer approach (convenience: ($F_{[1,23]} = 0.27$, $p = 0.61 > .05$), height: ($F_{[1,23]} = 3.16$, $p = 0.08 > .05$), overall: ($F_{[1,23]} = 3.86$, $p = 0.06 > .05$)). However, if we consider the processing time, we can see the difference. Our approach was accomplished within an average time of 3.75 minutes (the shortest is approximately 2 minutes, and the longest is approximately 5 minutes), while the Designer approach spent 38.42 minutes on average (the shortest is approximately 28 minutes, and the longest is approximately 65 minutes). The results show that our approach outperforms the User approach with statistical significance and implements faster than the Designer approach.

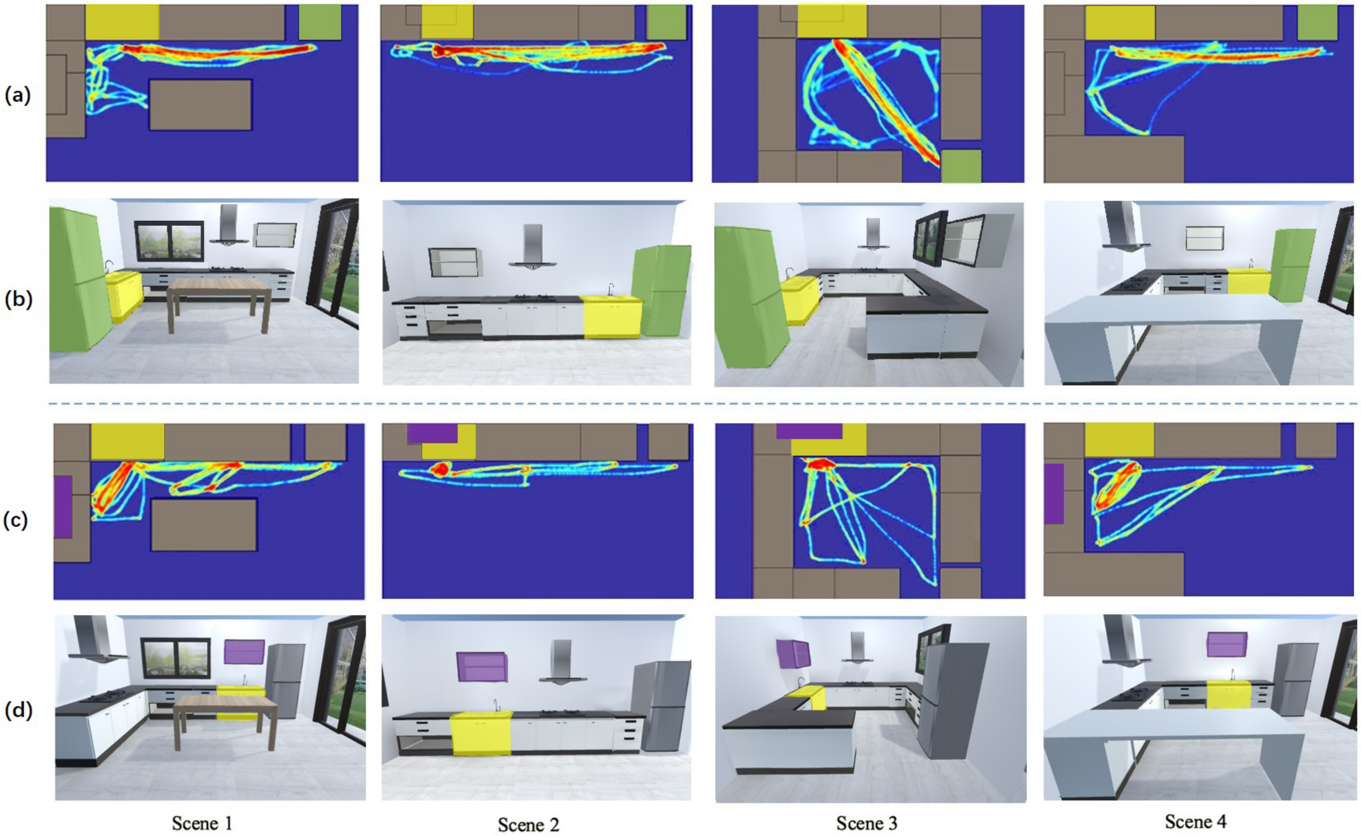


Fig. 10: Two users interact in four scenes. (a) and (c) are the human activities of User1 and User2 captured by the sensors. We use heatmaps to visualize the weighted frequency visited by the user when he/she uses the workspace to achieve tasks. The value of each point is the summation of the carried weight by the user. The redder the color is, the higher the value is. (b) and (d) are the visualizations of the optimized results.

7.2 Personal Preferences across Scenes

The core of our approach is to learn the personal preferences of the user by assuming that the personal preferences will remain across different scenes. To validate this point, we invited ten participants to experience all four scenes shown in Fig 8, different from the experiment in Sec. 7.2 in which each participant experienced one workspace. Our approach generated the optimization results for each of them based on different scenarios.

In Fig. 10, we show examples of two users (User1 and User2). From the first to the fourth column are the four scenes (the same as Fig. 8), respectively. (a) and (c) are activities captured from two users plotted with a heatmap, where the color of each pixel depicts the weighted trajectories (the summation of the objects' mass the user carried at each point) of the user. (b) and (d) are the optimized results according to their personal preferences.

It is interesting to see that there are some similarities in one participant's activities in some details across the four scenes in Fig. 10. For example, User1 preferred to store the food such as vegetables and fruits in the refrigerator (highlighted by green color). When she took out food from the refrigerator, she always went to the sink (highlighted by yellow color) to wash them directly. Based on his preferences of walking between the refrigerator and the sink frequently, our approach optimized the position of the refrigerator next to or close to the sink, User2 transferred between the wall component (highlighted by purple color) and the sink (highlighted by yellow color) more frequently. During his experiences, he preferred to put the plates into the shelf. Additionally, he washed each plate when he was going to use it. This preference is reflected in (c), where the trajectories between the sink and the shelf were redder. Thus, compared with the initial scenes, the sink and the shelf were closer in the optimized results.

The results of the experiment support our hypotheses that the personal preferences of an individual are constant across scenes. Thus, our approach can generate new human activities from the learned preferences to optimize the layout, as discussed in Sec. 5.1. We think that the setting up process is also helpful. Since the storage positions influence the human activities, i.e., different storage positions leading to different human trajectories, our approach requires the user to decide where to put all objects in the setting up process. Although we do not model the preferences for storage in our approach explicitly, it is encoded in the human activities and learned as the preferences later.

7.3 Personal Preferences in Wall Plane

To investigate whether the body proportions influences the optimized results, we analyze some users' data whose heights are the same. In Fig. 11, we show the wall plane optimization of three users, whose heights are all 165cm. The horizontal and vertical axes represent the height of the floor component and the wall component, respectively. Each point's color depicts how comfortable the user feels about the height according to the cost definition. The redder the color is, the smaller is the corresponding cost value is. The white dashed rectangles in Fig. 11 bounded the suggested height ranges mentioned in [33, 47]. From the visualized optimization results, we can see that the optimal solution intervals all fell into the suggested height range. Since the optimal height result may not be unique, we use a nearest rounded optimal height as the output. The optimized height of the floor component User1 is 90cm; the optimized heights of the floor component for both User2 and User3 is 85cm. Based on the pose measurement, we find that the position of the elbow joint of User1 is significantly higher than that of User2 and User3. The difference between User2 and User3 is mainly reflected in the result of the height of the wall component. The optimized wall component height for User2 is 150cm; the optimized wall component height for user3 is 155cm. Both User2 and User3 have a similar body proportion, but User2 stands further from the floor component than User3. This preference may lead to a change of view cost in the wall plane optimization and result in different wall component height results.

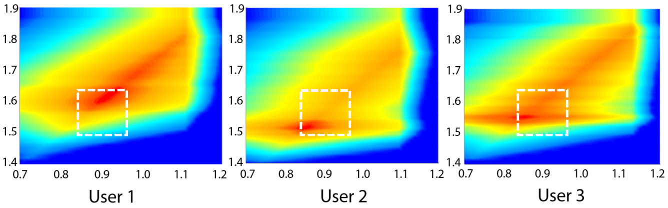


Fig. 11: Wall plane optimization process for three users with a height of 165*cm*. We use heatmaps to visualize the wall plane cost. The horizontal axis represents the height of the floor component; the vertical axis represents the height of the wall component. The redder the color is, the more suitable for the user.

7.4 Optimization for Other Workspaces

To examine the generality of our approach, we experiment on two other workspace scenes: a tailor shop and a garage workbench. We design different tasks in these two scenes. For the tailor shop, the tasks we set are drawing, taking out clothes, sewing, and so on. For the garage workbench, the tasks are changing work clothes, changing a tire for a bicycle, and so on.

Fig. 12 shows the optimized results. For the tailor shop scene, a user preferred to put tools into the shelf. When she started the task, she got the tools from the shelf first and then turned to the desk. Comparing the initial scene with the optimized scene, we can see that the optimized scene brought the wall component and the desk closer. In addition, after sewing the clothes, she usually put them on the dress form. In the optimized scene, the sewing machine was optimized to the position adjacent to the dress form. For the garage workbench, the storage components (the shelf and the high cabinet) were optimized to be apart, which was more consistent with a user's habits of operation. More specifically, he frequently travelled between two shelves and he had less interaction with the high cabinet. In addition, the sink was used only once in his experiences; thus it was moved from the middle to the edge in the updated layout.

The results of the experiment show that our approach can be applied to more than one workspace category. By setting the proper tasks for the kind of workspace, our approach can learn the personal preferences from the user's operations. Then, our approach can optimize a more reasonable layout of the scene with more proper heights of the components.

8 Summary

In this paper, we present an approach for remodeling workspaces based on the personal preferences. We design a pipeline to learn the personal preferences from human activities. Based on the learned personal preferences, two cost functions are defined to evaluate how reasonable the layout of a scene is (floor plane cost function) and how comfortable the user may feel about the height of each component (wall plane cost function). In the floor plane cost function, we consider the personal preference and spatial information. In the wall plane cost function, we consider the pose and visual field of the user. The cost functions are optimized to output an updated workspace, which fits an individual user better.

A virtual reality approach has great benefits for the design of the scene. VR devices offer realistic and immersive experiences, which allows users to experience a designed workspace before actually building them. Compared with other approaches, our approach can learn the user's personal preferences. The learned personal preferences can help to improve the design and make it more suitable for individuals, especially when the user's preference is inconsistent with the statistical data from crowds.

Limitations. The interactions with the objects in the scene are not consistent with the real world. For example, when a user places the objects somewhere in the scene, he needs to select the object from a menu; point to the position; and press the trigger to release the selected object. Due to the limitations of the hardware devices, we simplify parts of the interaction. If VR gloves or other interactive devices can be used, the user's experience in the interaction process will be more realistic.

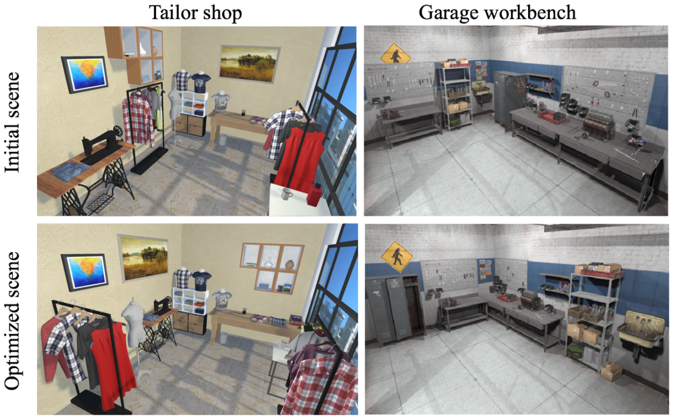


Fig. 12: The optimized results of a tailor shop and a garage workbench. The scenes are optimized based on the user's operational preferences and work trajectory after the user performs the given tasks.

Different from the real world, our approach does not provide realistic haptic feedback to the user, which may affect the user's experience. For example, in a kitchen setting, cutting is not so realistic without tactile feedback. When a user carries an object, he does not feel the weight of the object as in the real world.

Moreover, due to the space constraints of the HTC VIVE device, it is hard to optimize a scene that exceeds the maximum space size. Our approach could be applied to a larger scene if devices such as the Virtuix Omni could be used. In addition, the number, the size, and the style of the components are immutable during the optimization. This may reduce the diversity of the optimized design results.

Future Work. One possible extension is to integrate 3D reconstruction techniques [6, 15] into our approach. With a reconstructed 3D scene or a scanned 3D scene, we can apply our pipeline to optimize an existing workspace conveniently. In this way, if a user wants to redesign an existing workspace, such as a kitchen, our approach can use the current workspace as the initial scene without extra effort. Of course, we can also integrate graphics techniques to extend the initialization process. For example, we can use the works of sketch2scene approach [49] to generate a 3D scene from a sketch. Moreover, if we can estimate the users trajectories and actions accurately in the real environment (e.g., from a video or surveillance camera) in the future, it may help the approach further, given the natural interactions and natural activities. It also allows our approach to integrate with that of a designer. The designer provides an initial design, then our approach helps to optimize the initial design in terms of the personal preferences.

Another possible extension is to take into account the structure of the target scene. For example, the sink should be next to the water supply, and the gas stove needs to be near the gas pipe. With these requirements added, the optimized results will be more practical.

Currently, we provide one initial scene for the user to experience. The tasks are defined based on the characteristics of the workspace and all users achieve the same tasks in one kind of scene. The settings may affect the user preferences determined. It will be more targeted if we provide more initial scenes and define more types of tasks for the user to choose. In addition to personal preferences, we consider accessibility and free space constraints for the spatial cost. In the future, we may explore developing other cost functions for issues such as aesthetics, symmetry, and illumination.

In our current approach, we only consider the personal preferences of one person. It would be interesting to extend our approach to multiple people scenarios, where the preferences from different roles could be considered. For example, if we want to optimize an office coffee bar, we need to consider the different preferences from different people.

Acknowledgments

This research is supported by the Natural Science Foundation of China (NSFC) under award number 61876020 and the National Science Foun-

dation under award number 1565978.

References

[1] C. Abras, D. Maloney-Krichmar, and J. Preece. User-centered design. *Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications*, 37(4):445–456, 2004.

[2] Y. Akazawa, Y. Okada, and K. Niijima. Automatic 3d scene generation based on contact constraints. In *Computer Graphics and Artificial Intelligence*, pages 593–598, 2005.

[3] C. Armbrüster, M. Wolter, T. Kuhlen, W. Spijkers, and B. Fimm. Depth perception in virtual reality: distance estimations in peri- and extrapersonal space. 11(1):9, 2008.

[4] D. K. Ballast, C. FAIA, N. C. No, et al. *Interior Design Reference Manual: Everything You Need to Know to Pass the NCIDQ Exam*. www. ppi2pass. com, 2013.

[5] S. R. Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Robotics and Automation*, 17(1-19):16, 2004.

[6] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *arXiv:1702.04405*, 2017.

[7] J. Dul and B. Weerdmeester. *Ergonomics for beginners: a quick reference guide*. CRC press, 2003.

[8] K. Falko, B. Benjamin, B. Gerd, R. Ulrich, S. Frank, L. Markus, and K. Reinhard. Geometric calibration of head-mounted displays and its effects on distance estimation. *IEEE TVCG*, 18(4):589–596, 2012.

[9] M. Fisher, D. Ritchie, M. Savva, T. Funkhouser, and P. Hanrahan. Example-based synthesis of 3d object arrangements. *ACM TOG*, 31(6):1–11, 2012.

[10] M. Fisher, M. Savva, Y. Li, P. Hanrahan, and M. Nießner. Activity-centric scene synthesis for functional 3d scene modeling. *ACM TOG*, 34(6):179, 2015.

[11] C. Fransson-Hall, S. Byström, and A. Kilbom. Self-reported physical exposure and musculoskeletal symptoms of the forearm-hand among automobile assembly-line workers. *Journal of occupational and environmental medicine*, 37(9):1136–1144, 1995.

[12] 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 TOG*, 36(6):201, 2017.

[13] R. Gäbler. *Sweet Home 3D kompakt, m. CD-ROM*. Brain-Media.de, 2011.

[14] S. Hignett and L. McAtamney. Rapid entire body assessment. In *Handbook of Human Factors and Ergonomics Methods*, pages 97–108. CRC Press, 2004.

[15] B.-S. Hua, Q.-H. Pham, D. T. Nguyen, M.-K. Tran, L.-F. Yu, and S.-K. Yeung. Scenenn: a scene meshes dataset with annotations. *International Conference on 3D Vision*, 2016.

[16] JamesBear. A full body tracking demo using htc vive and trackers. `https://github.com/JamesBear/vive_ik_demo`.

[17] C. Jiang, S. Qi, Y. Zhu, S. Huang, J. Lin, L. F. Yu, D. Terzopoulos, and S. C. Zhu. Configurable 3d scene synthesis and 2d image rendering with per-pixel ground truth using stochastic grammars. *IJCV*, 126(9):920–941, 2018.

[18] R. A. Jones and W. H. Kapple. Kitchen planning principles: Equipment, appliances. Technical report, Small Homes Council-University of Illinois Urbana-Champaign, 1975.

[19] P. Julius. *Human dimension and interior space : A source book of design reference standards*. Whitney Library of Disign, 1979.

[20] O. Karhu, P. Kansi, and I. Kuorinka. Correcting working postures in industry: a practical method for analysis. *Applied Ergonomics*, 8(4):199–201, 1977.

[21] K. Kemmlert. A method assigned for the identification of ergonomic hazardsplibel. *Applied Ergonomics*, 26(3):199–211, 1995.

[22] K. A. H. Kjølaas. Automatic furniture population of large architectural models. *Doctoral dissertation*, Massachusetts Institute of Technology, 2000.

[23] J. Kramer, S. Noronha, and J. Vergo. A user-centered design approach to personalization. *Communication of the ACM*, 43(3):44–48, 2000.

[24] O. Labs. Job simulator. `https://store.steampowered.com/app/448280/Job_Simulator/`.

[25] Y. Lang, W. Liang, F. Xu, Y. Zhao, and L.-F. Yu. Synthesizing personalized training programs for improving driving habits via virtual reality. In *IEEE VR*, 2018.

[26] M. Y. Lee. Designing a children's recreation room. *Teaching Children Mathematics*, 22(2):110–114, 2015.

[27] C. Li, W. Liang, C. Quigley, Y. Zhao, and L. F. Yu. Earthquake safety training through virtual drills. *IEEE TVCG*, 23(4):1275, 2017.

[28] G. Li and P. Buckle. A practical method for the assessment of work-related musculoskeletal risks-quick exposure check (qec). In *Proceedings of the human factors and ergonomics society annual meeting*, volume 42, pages 1351–1355. SAGE Publications Sage CA: Los Angeles, CA, 1998.

[29] W. Liang, Y. Zhao, Y. Zhu, and S.-C. Zhu. What is where: Inferring containment relations from videos. In *25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3418–3424, 2016.

[30] W. Liang, Y. Zhu, and S.-C. Zhu. Tracking occluded objects and recovering incomplete trajectories by reasoning about containment relations and human actions. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[31] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos in the wild. In *CVPR 2009.*, pages 1996–2003.

[32] Z. Lun, M. Gadelha, E. Kalogerakis, S. Maji, and R. Wang. 3d shape reconstruction from sketches via multi-view convolutional networks. In *International Conference on 3D Vision*, pages 67–77. IEEE, 2017.

[33] M. Maguire, S. Peace, C. Nicolle, R. Marshall, R. Sims, J. Percival, and C. Lawton. Kitchen living in later life: Exploring ergonomic problems, coping strategies and design solutions. 2014.

[34] L. Mcatamney and C. E. Nigel. Rula: a survey method for the investigation of work-related upper limb disorders. *Applied Ergonomics*, 24(2):91–9, 1993.

[35] P. Merrell, E. Schkufza, Z. Li, M. Agrawala, and V. Koltun. Interactive furniture layout using interior design guidelines. In *ACM TOG*, volume 30, page 87, 2011.

[36] Mingze and S. P. Smith. Simulating cooperative fire evacuation training in a virtual environment using gaming technology. In *IEEE VR*, pages 139–140, 2014.

[37] M. Mitton and C. Nystuen. *Residential interior design: A guide to planning spaces*. John Wiley and Sons, 2016.

[38] T. Neumann, K. Varanasi, N. Hasler, M. Wacker, M. Magnor, and C. Theobalt. Capture and statistical modeling of arm-muscle deformations. In *Computer Graphics Forum*, volume 32, pages 285–294, 2013.

[39] D. C. Niehorster, L. Li, and M. Lappe. The accuracy and precision of position and orientation tracking in the htc vive virtual reality system for scientific research. *i-Perception,8,3(2017-5-01)*, 8(3), 2017.

[40] V. Z. Priel. A numerical definition of posture. *Human Factors*, 16(6):576–584, 1974.

[41] S. Qi, Y. Zhu, S. Huang, C. Jiang, and S. C. Zhu. Human-centric indoor scene synthesis using stochastic grammar. In *CVPR*, 2018.

[42] T. Tutenel, R. Bidarra, R. M. Smelik, and K. J. De Kraker. Rule-based layout solving and its application to procedural interior generation. In *CASA Workshop on 3D Advanced Media In Gaming And Simulation*, 2009.

[43] van der Veer AH, A. AJT, L. MR, W. HY, and M. BJ. Where am i in virtual reality? In *PLoS ONE*, page 13(10), 2018.

[44] K. Wang, M. Savva, A. X. Chang, and D. Ritchie. Deep convolutional priors for indoor scene synthesis. *ACM TOG*, 37(4):70:1–70:14, 2018.

[45] T. R. Waters, V. Putz-Anderson, A. Garg, and L. J. Fine. Revised niosh equation for the design and evaluation of manual lifting tasks. *Ergonomics*, 36(7):749–776, 1993.

[46] M. Werner. Indoorsim: Simulation of wireless-lan-based indoor positioning systems incorporating cad-floorplans. *Tagungsband des 8. GI/KuVS-Fachgespräch Ortsbezogene Anwendungen und Dienste*, 2011.

[47] M. Wilson et al. Patterns for kitchen cabinets. 1947.

[48] M. Windolf, N. Götzen, and M. Morlock. Systematic accuracy and precision analysis of video motion capturing systems–exemplified on the vicon-460 system. *Journal of Biomechanics*, 41(12):2776–2780, 2008.

[49] K. Xu, K. Chen, H. Fu, W. L. Sun, and S. M. Hu. Sketch2scene:sketch-based co-retrieval and co-placement of 3d models. *ACM TOG*, 32(4):1–15, 2013.

[50] K. Xu, J. Stewart, and E. Fiume. Constraint-based automatic placement for scene composition. *Proceedings - Graphics Interface*, pages 25–34, 2002.

[51] Y.-T. Yeh, L. Yang, M. Watson, N. D. Goodman, and P. Hanrahan. Synthesizing open worlds with constraints using locally annealed reversible jump mcmc. *ACM TOG*, 31(4):1–11, 2012.

[52] 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. In *ACM TOG*, 2011.