A CALCULATING ROUNDNESS

Let $G = (V, E)$ be an undirected graph to be constructed for computing $f_r(L_i)$. Each vertex $v \in V$ owns an attribute $p_v = (x, y)$, which represents the position of the vertex in a scene. There is also an attribute d_e for each edge $e = (u, v) \in E \quad (u, v \in V)$, which stands for the length of e. In our context, an edge (u, v) maps a straight path between two points p_v and p_u in the scene through which customers travel. As a result, d_e could be directly calculated as the Euclidean distance between vertices u and v, formulated as:

$$d_e = ||p_u - p_v||_2. \tag{5}$$

Then the key lies in deciding every vertex $v \in V$ with its p_v and every edge $e = (u, v) \in E$ for the scene.

Normally, taking the "width" of an edge into account would be unnecessary, but that is not the case for an actual pathway in a traffic flow. Any path between two points must have a minimum width of W_m to satisfy the continuous exploration. To achieve this, two shelves split by an edge in G must yield a distance no smaller than W_m. When the path is represented by a centered-aligned line segment, any shelf by its sides keeps a distance no shorter than $\frac{W_m}{2}$ from it. Figure 4a of the main paper illustrates the distance $\frac{W_m}{2}$ kept by patterns.

We begin to construct G by constructing a subgraph $G_l = (V_l, E_l)$ for every pattern l in the room. All patterns are designed to be surrounded by several paths, considering the exterior part of G_l. The exterior vertices are connected linearly, thus forming a polygon-shaped ring that bounds a pattern. Generally, the polygon is in a similar shape to the pattern's tight bounding box, but is enlarged due to the $\frac{W_m}{2}$ distance constraints. As for the interior part, the vertices and edges are added following the pattern's properties. For example, a radial pattern is expressed by a center vertex that connects the exterior vertices through multiple edges. A grid pattern may embody a series of interior vertices connected adjacently. For each pattern l, G_l is a union of all vertices and edges in the interior and the exterior. Then G is comprised of the vertices and edges of all patterns, including v_{in} and v_{out}.

Two exterior edges from different patterns may be close to or even overlap with each other when they are put into an actual room. We notice that only one actual route instead of two should be recognized. Supposing $e_1 = (u_1, v_1)$ and $e_2 = (u_2, v_2)$ are two edges with actual line segments (p_{u_1}, p_{v_1}) and (p_{u_2}, p_{v_2}) on lines $\Upsilon_1 : a_1x + b_1y = c_1$ and $\Upsilon_2 : a_2x + b_2y = c_2$, respectively. For simplicity, the two edges can be merged only when: (i) $a_1b_2 - b_1a_2 = 0$, which means they are parallel; (ii) the distance:

$$d_e = \frac{|c_1 - c_2|}{\sqrt{{a_1}^2 + {b_1}^2}} \leq D_m; \tag{6}$$

(iii) when projected on any line parallel to them, they share an overlap part. Among the conditions above, D_m typically equals $\frac{W_n}{2}$, which is a threshold small enough to merge edges. After a single step of merging, the two original edges become three new edges embodied by actual line segments (p_{i_1}, p_{i_2}), (p_{i_2}, p_{i_3}), and (p_{i_3}, p_{i_4}), with the segment (p_{i_2}, p_{i_3}) in the middle originated from the previously overlapped part. They both lay on the same line L:

$$a_1x + b_1y = \frac{d_{e_1}c_1 + d_{e_2}c_2}{d_{e_1} + d_{e_2}}, \tag{7}$$

which represents a continuous route (p_{i_1}, p_{i_4}) between the original patterns.

After a series of merging operations, the original graph could change a lot. The attributes of vertices and edges should be modified accordingly, while different patterns may now share a common exterior consisting of various vertices connected as a new circuit.

When there is more than one patterns in the room, G could be interpreted as two points with one or several isolated rings. We need to selectively connect them to construct G. For the connection between a point p and a ring r, we project p on r and determine the projection point p_r. Then we add a new edge with segment (p, p_r). However, it is valid only when $||p - p_r||_2 <= D_c$ and (p, p_r) does not intersect with other rings. For the connection between two rings r_1 and r_2, we find two points p_1 and p_2 on them with the minimum distance and add an edge with segment (p_1, p_2). The undirected graph G is eventually completed.

$G = (V, E)$ may not be connected in some cases, such as no pattern exists or several patterns are isolated, where the roundness $f_r(L_i)$ is set to zero. Otherwise, $f_r(L_i)$ is calculated. We consider G a weighted tree, with each edge having a weight equal its attribute d_e. Then we have:

$$f_r(L_i) = \frac{\Sigma_{e_l \in \chi} d_{e_l}}{\Sigma_{e \in \chi} d_e}, \tag{8}$$

where χ is the set of all edges in the longest path. Every vertex is visited less than once, so no circuit exists in the longest path. Since G may not be acyclic, the longest-path problem is NP-complete, so we use an approximation to compute it.

Since G may not be acyclic, the longest-path problem is in NP-complete. Therefore, we use an approximation method to compute the longest path as follows:

(i) Compute the maximum spanning tree $T_1(G)$. We then use it to determine the only path $S = \{v_{in}, v_1, ...v_k, v_{out}\}$ that connects v_{in} and v_{out}. For simplicity, v_{in} is named as v_0, and v_{out} as v_{k+1}. $T_1(G)$ is discarded since we do not need it in the following steps.

(ii) Iterate i for the range from 0 to k. For each i, we iterate all neighbors of v_i and v_{i+1} and check if there exists a path $S_i = \{v_i, v_{n_i}, v_{j_1}, v_{j_2}..., v_{j_l}, v_{n_{i+1}}, v_{i+1}\}$ in G, where $(v_i, v_{n_i}), (v_{n_{i+1}}, v_{i+1}) \in E(G)$, and $\{v_{n_i}, v_{j_1}, v_{j_2}..., v_{j_l}, v_{n_{i+1}}\}$ is a path in $G - S$. Based on triangular inequality, if S_i exists, it is certainly longer than $\{v_i, v_{i+1}\}$. We then substitute the old one in S and form a new path S',which becomes the new S. Then we continue the iteration from v_{i+1}.

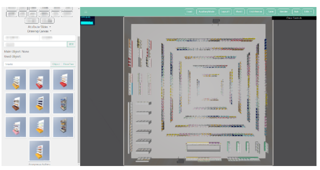
(iii) If the iteration in the former step does not succeed in discovering a replacing path even for once, we continue for step (iv). Otherwise, we perform another iteration for step (ii).

(iv) Compute the maximum spanning tree $T(G)$ that contains S. S stays unchanged in this step.

(v) Iterate e for every $e \in E(G) - \chi$ where $\chi = \{(v_0, v_1), (v_1, v_2), ..., (v_k, v_{k+1})\}$ consists of the edges derived from S. For each e, adding it into $T(G)$ would result in one and only one circuit $C = \{(v_i, v_{i+1}), (v_{i+1}, v_{i+2}), ...(v_{i+j-1}, v_{i+j}), (v_{i+j}, v_{p_1}), (v_{p_1}, v_{p_2}), ...e, ..., (v_{p_q}, v_i)\}$. We then compare the total path length for both $C_1 = \{(v_i, v_{i+1}), (v_{i+1}, v_{i+2}), ...(v_{i+j-1}, v_{i+j})\}$ and $C_2 = \{(v_{i+j}, v_{p_1}), (v_{p_1}, v_{p_2}), ...e, ..., (v_{p_q}, v_i)\}$. If C_2 is longer, then we find a substitution for C_1 in S. e is thus added to $T(G)$, and the shortest edge in C_1 is removed to ensure that $T(G)$ is still a tree. S is changed due to the substitution. On finishing the actions, we break the iteration.

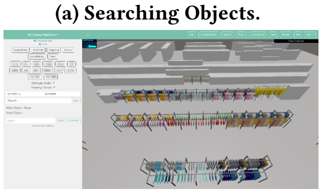
Table 2: The efficiency of our method concerning different sizes of rooms. Times are recorded as seconds.

	10m	15m	20m	25m	30m
10m	106				
15m	126	163			
20m	154	187	261		
25m		280	366	423	
30m			441	531	588
40m				669	745
50m					1042



(a) Searching Objects.

(b) Transforming Objects.

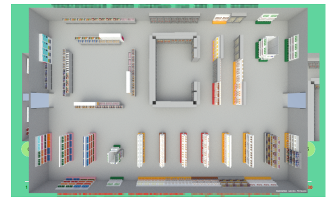


(c) Level of Details.

(d) Box Selections.

Figure 11: The web-based platform for manually designing large-scale commercial layouts, where we can manipulate objects just like using similar industrial applications such as Kujiale [14] or Plannar5d [25]. The "level of details" and "box selections" further enable efficient interactions with a large scale of objects.

(a)

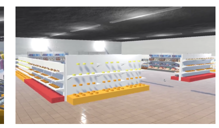


(b)

Figure 12: (a): The platform for the user study on the aesthetic and plausibility, where we could answer questions, temporarily save answers and jump to other questions. (b): Users can also optionally zoom in each presented scene.

(a) Fresh Foods

(b) Clothes



(c) Snacks

Figure 13: The VR-oriented client for conducting the immersive user study to measure the scenes.

(vi) If the iteration in (v) was ended by a successful exchange, we jump to (v) for another iteration. Otherwise, the longest path S is determined, and we output the result.

B DIVERSITY

We generated 1000 scenes and counted the patterns distributed in them. In general, 43.2% of the scenes have incorporated every proposed pattern. 99.0% of the scenes have incorporated three types of patterns. 9.4% of the scenes are dominated by a specific pattern, i.e., a pattern occupies more than 30% of the scene area. For example, a shop is mainly arranged with a cross pattern. The overall assembling rates of the four patterns in all scenes are 29.86% (Linearity), 17.68% (Cross), 19.40% (Circulation), and 33.06% (Radiation), respectively. The assembling rates are counted over areas since a pattern type may have a small number but occupy large areas.

C EFFICIENCY

We ran our method on different sizes of floor plans. The average cost times are shown in Table 2. Each cell refers to a time consumption value in seconds concerning a floor plan size. For example, the cost time for size 20m*20m is 261s. This experiment uses a single core to clarify the time consumption. When leveraging multiple cores, using a ten-core processor, our method achieves ten results given a similar time.