Label Guidance based Object Locating in Virtual Reality

Xiaoheng Wei 1  Xuehuai Shi 1  Lili Wang 1,2,3 *

1 State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China
2 Peng Cheng Laboratory, Shenzhen, China
3 Beijing Advanced Innovation Center for Biomedical Engineering, Beihang University, Beijing, China.

*Lili Wang is the corresponding author: wanglily@buaa.edu.cn

ABSTRACT

Object locating in virtual reality (VR) has been widely used in many VR applications, such as virtual assembly, virtual repair, virtual remote coaching. However, when there are a large number of objects in the virtual environment (VE), the user cannot locate the target object efficiently and comfortably. In this paper, we propose a label guidance based object locating method for locating the target object efficiently in VR. Firstly, we introduce the label guidance based object locating pipeline to improve the efficiency of the object locating. It arranges the labels of all objects on the same screen, lets the user select the target labels first, and then uses the flying labels to guide the user to the target object. Then we summarize five principles for constructing the label layout for object locating and propose a two-level hierarchical sorted and orientated label layout based on the five principles for the user to select the candidate labels efficiently and comfortably. After that, we propose the view and gaze based label guidance method for guiding the user to locate the target object based on the selected candidate labels. It generates specific flying trajectories for candidate labels, updates the flying speed of candidate labels, keeps valid candidate labels, and removes the invalid candidate labels in real time during object locating with the guidance of the candidate labels. Compared with the traditional method, the user study results show that our method significantly improves efficiency and reduces task load for object locating.

Index Terms: Virtual Reality, Object Locating, Label Guidance

1 INTRODUCTION

Recently, VR technology has made significant progress and has been applied to many industries such as manufacturing, entertainment, and education. Object locating is widely used [1], especially in multi-person collaborative operation applications such as virtual assembly, virtual repair, and virtual remote coaching. However, inefficient object locating methods will greatly reduce the users' experience of these applications.

Locating the target object with the label guidance is an idea for efficiently object locating in VR. However, it brings three challenges. In the VE with a large number of labeled objects, the user has to search for the target in all directions. If other objects block the label of the target object, the user also needs to walk around to find it, which makes it more difficult for the user to find the target object. So the first challenge is to design the pipeline of label guidance for locating the target object efficiently and comfortably. In the VE that
contains a large number of labeled objects, the user can not find the
candidate labels quickly. So the second challenge is to construct
the sorted label layout so the user can select the candidate labels
efficiently. When using the candidate labels to guide the user
to locate the target object, if the candidate labels are too close or too
far away from the user’s position, the moving speed of the candidate
labels is too fast or slow, or the invalid candidate labels are not
removed in time, the user can not locate the target object accurately
and comfortably. So the third challenge is to guide the user to locate
the target object accurately and comfortably.

In this paper, we propose a label guidance based object locating
method to improve the efficiency of locating the target object in VR
applications. For the first challenge, we propose a label guidance
based object locating pipeline, which arranges the labels of all ob-
jects on the same screen, lets the user select the label target first, and
uses the flying labels to guide the user to the target object. In order
to let the user select the candidate labels efficiently and comfortably,
we summarize five principles for constructing the label layout for
object locating. For the second challenge, we design and construct
the two-level hierarchical sorted and oriented label layout based on
five summarized principles. The user selects the candidate labels in
the two-level hierarchical sorted and oriented label layout. For the
third challenge, we propose the view and gaze based label guidance
method to generate specific flying trajectories for candidate labels,
update the flying speed of candidate labels, keep valid candidate
labels and remove the invalid ones in real time during the process
of object locating. Finally, the user uses the valid flying candidate
label to locate the target object. We design a user study of two tasks
to evaluate the performance of our method. Compared with the
traditional method, the results show that our method significantly
improves efficiency and reduces task load for object locating. Figure
1 shows the process of a user locating an object by our method.

In summary, the contributions of our method are as follows: 1) we
propose a label guidance based object locating pipeline to improve
the efficiency of locating the target object in VR applications; 2) we
introduce a two-level hierarchical sorted and oriented label layout to provide sort and orientation cues of the target object; 3) we
introduce a view and gaze based label guidance method to optimize
the label guidance path, update the guidance speed of labels, keep
valid labels and remove the invalid labels in real time during the
process of object locating; 4) we design a user study to evaluate the
efficiency of our method.

2 RELATED WORK
In this section, we briefly review the previous work on the label
layout and out-of-view guidance related to our method.

2.1 Label Layout
Label layout is widely studied in two-dimensional and three-
dimensional images. Fink et al. [5] used a circle and cluster layout
to place labels on the focus regions of the two-dimensional map.
Heinsohn et al. [11] placed labels for the dynamic nature of the focus
region, and users can obtain details on their demand through the cluster layout during the overview. Kouvril et al. [13] extracted the
hierarchical structure of the objects and labeled different levels of
objects to deal with large hierarchical environments.

Tatzgern et al. [27] constrained the placement of labels in 3D
object space according to user viewpoint transformation. Cmolik et
al. [3] proposed a hybrid label layout, which determines the place-
ment position and type of labels according to the threshold specified
by the user. Zhou et al. [34] determined a 2D label layout plane by
user view direction and arranged the labels on the circle of the 2D
plane in alphabetical order. Grassett et al. [6] summarized the basic
rules and criteria for placing labels in VR scenes, and they proposed
an image-based approach to identify geometric constraints for plac-
ing labels. McNamara et al. [21] used eye tracking to calculate the
user’s potential object of interest and adjusted the placement strategy
of label information in the complex VE. Jia et al. [12] established
the label placement constraints according to the user’s semantic
perception of the image. The above three label placement methods all had objects scattered in multiple locations of a scene. Generally, label layout requires
that labels cannot occlude each other, whether for an in-view object
or multiple objects of a scene. The position of the label should be
adjusted according to the user’s viewpoint and always remain in the
user’s current view, and the distance between the label and its anchor
object should be as close as possible. Our method provides the label
layout that remains in the user’s current view, with no occlusion
among labels, and good interactivity.

When the scene has too many labels, the display of the user’s
current view quickly becomes cluttered. Zhang et al. [33] reduced the
number of labels displayed by scoring building importance and
scheduling annotations in the video. Tatzgern et al. [28] created a
temporally coherent layout for compact label annotations, thus
avoiding visual interference when the viewpoint changes. Then they
[29] created an information hierarchy to cluster a large number of
labels by weighted calculation of user-defined spatial and non-spatial
attributes. The above methods adaptively adjusted the information
density by reducing the labels displayed on the screen by filtering or
clustering operations. We also use this idea for the object locating
task, cluster the labels according to their initial letter, and then
expand the labels according to the selected initial letter.

2.2 Out-of-View Object Guidance
The guidance technology of out-of-view objects is divided into
types according to visualization methods. One is to visualize
out-of-view objects as abstract visualized symbols and encode
the information of out-of-view objects into the attributes of the vis-
ualized symbols. Peterson [22] reduced the label visual cluster
by using stereoscopic discrimination. Schwerdtfeer [24] compared
the frame, tunnel, and arrow visualization guidance. Renner [23]
evaluated AR-based guiding techniques based on images, funnel,
arrow and proposed a SWAVE guidance technique based on eye
gaze information. EyeSee360 [7] is a 2D visualization technique
with distance-encoding and direction-encoding. The flyingarrow [8]
flew to the location of the out-of-view object according to the user’s
current sight. Bork [2] proposed a mirror ball of all virtual ob-
jects’ reflections to provide positive hints and a 3D radar method for
visualizing user positions and out-of-view objects.

The other is to use non-visual cues, such as vibro-tactile cues
[15, 17], auditory cues [20, 31] or blend several cues [18].

Another guidance technology is labeling. The advantage of label-
ing is to support bidirectional retrieval workflows for object-to-label
and label-to-object lookups [16]. Kruijff [14] evaluated the impact
of virtual label characteristics such as color, size, and leader lines
on the search performance and gave suggestions on label design in
wide FOV augmented reality displays. Lin [16] explored the design
space of labels in AR applications for situated visual search and
compared three representative AR labeling techniques that encode
different objects’ different information. They demonstrated that
angle-encoded labels with directional cues perform best, and our
method also uses the labels to guide the user to the target object.

3 METHOD
In order to locate the target object in the VE with a large number
of objects more efficiently and comfortably, we propose a label
 guidance based object locating method. In this section, we first
describe our label guidance based object locating pipeline in Section
3.1. Then we introduce the two-level hierarchical sorted and orien-
tated label layout to provide orientation cues for the target object
in Section 3.2. At last, we introduce the view and gaze based label
guidance method to optimize the guidance path, update the guidance
speed of labels, keep valid labels and remove the invalid labels in
real time during the process of object locating in Section 3.3.
3.1 Label Guidance based Object Locating Pipeline

When a user stands in the VE with many objects around him, it is not easy for the user to locate the target object with a specific label. The user has to turn around and look in all directions to search for the target object. In order to improve the efficiency of object locating, the idea of our method is to arrange the labels of all objects on the same screen, let the user select the candidate labels first, and use the candidate labels to guide the user to the target object. The pipeline of our label guidance based object locating has the following steps.

Firstly, we construct the two-level hierarchical sorted and oriented label layout. The two-level hierarchical sorted and oriented label layout contains the first-level sorted circle layout and the second-level sorted circle layout. The first-level sorted circle layout is a circle displayed in the user’s view. The initial letters of labels are arranged on the edge of the first-level sorted circle layout in counterclockwise alphabetical order. The second-level sorted circle layout appears in the user’s view when the first-level sorted circle layout disappears. It is a concentric circle, and the sorted labels with orientation cues are arranged on the edge of all circles in the second-level sorted circle layout.

Secondly, the user selects the target label with head movement on the two-level hierarchical sorted and oriented label layout. We refer to the center of the user view as the gaze point since the use of gaze capture device will bring two problems. First, the gaze point position may be located in the current view’s edge area, which will break the in-view principle in subsection 3.2. Second, since the user’s gaze position and gaze direction may jump continuously over a while, if the location and the normal direction of the label layout change according to the user’s gaze position and gaze direction frame by frame precisely, the label layout will jump and flicker on the screen, which will cause motion sickness.

When the user needs to select the label, the first-level sorted circle layout is displayed on the screen. Then the user performs the initial letter selection. In the initial letter selection, the user uses gaze based dwell-time method [32] to select the initial letter, i.e., the user’s gaze stays on the initial letter for 400ms. After that, the first-level sorted circle layout disappears, and the second-level sorted circle layout unfolds, on which the labels with the selected initial letter are shown. The user selects the candidate labels on the second-level sorted circle layout. In the candidate labels selection, we first calculate the orientation of each label from the label position and the center position of the second-level sorted circle layout. Those labels whose orientations are less than 90° from the moving direction of the gaze point are regarded as candidate labels. Then we record the user’s gaze point in each frame, compute the moving direction of the gaze point by the least-squares fitting function, and the gaze moving direction is used to select the candidate labels.

Thirdly, the candidate labels guide the user to locate the target object by flying in a specific trajectory. The candidate labels fly back to their anchor objects according to the optimized paths to avoid labels penetrating the user’s body during the guidance process. Simultaneously, the user locates the target object with the guidance of candidate labels. The user moves his head and uses his gaze to follow and select the target label from the flying candidate labels. The moving direction of the candidate labels and the moving direction of the user’s gaze is calculated in real time. If the moving direction of the candidate deviates from the user’s gaze moving direction by more than 90° during the flight, it will be regarded as the invalid candidate label and be removed. The user uses the valid flying candidate labels to locate the target object.

3.2 Two-level Hierarchical Sorted and Orientated Label Layout

Previous work on labeling objects in VR and AR has introduced some basic principles of label layout, such as no occlusion among labels, closer distance between the label and its anchor object, and excellent interactivity. In the label guidance based object locating task, in order to search the labels and locate the corresponding objects efficiently, we summarize five principles.

Principle 1 In View. There are many objects in the VE around the user. When the user needs to locate the target object, she/he needs to look around constantly with the traditional method. The idea of the label-guided method is to find the label of the target first and then guide the user to locate the target through the movement of the label. In order to find the label efficiently, the initial position of the label is preferably within the user’s view, so that the user does not need to look around. After the label is found, if the anchor object is not in the current view, the label will fly and guide the user to turn his head and transform to the view containing the anchor object.

Principle 2 Sorted Label. To further speed up the label selection, the label arrangement needs to be organized in an orderly manner. Usually, the user is most familiar with the alphabetical order so that the labels can be arranged in that order. Due to a large number of objects, the speed of label selection is still limited if all labels are arranged in alphabetical order. Given that the user is very familiar with the first letter index of the dictionary and the way words are arranged alphabetically under the user’s first letter, it is an excellent choice to use a hierarchical method for label layout arrangement.

Principle 3 Interaction. The layout label allows users to make convenient and robust label selection with only minor movements, combined with the steering to track labels and locate anchor objects. There are no restrictions on the specific interaction method. That is to say, users can use the handle or hand-free mode to select labels.

Principle 4 Orientation. The display position of the label on the screen can encode the orientation information of the object indicated by the label, helping the user obtain the relative position of the label and the anchor object, understand the potential moving direction of the flying label and effortlessly follow the label guidance.

Principle 5 Distance. If the orientation cue of the anchor object contained in the label is a specific value, this constraint is too strict because the label position should reflect orientation information about its anchor object and not be strictly limited to a precise direction value. So a range of orientations is usually used. Range size should be determined by the distance between the label and its anchor object, and the greater the distance, the greater the range.

According to the five principles above, we propose the two-level hierarchical sorted and oriented label layout. Based on Principle 1, all labels can be displayed in the user view by pressing the button on the handle when the user needs the label guidance.

Based on Principle 2 and Principle 3, we design a two-level hierarchical sorted and oriented label layout, which contains the first-level sorted circle layout and the second-level sorted circle layout. The first-level sorted circle layout is a circle displayed in the user’s view. The initial letters of the labels are arranged in counterclockwise alphabetical order from the three o’clock position in the first-level sorted circle layout. The first-level sorted circle layout is placed in the center of the user’s view, and its radius is calculated using the central vision of the human field of view (FOV) [30]. After the initial letter selection, the first-level sorted circle layout disappears. The second-level sorted circle layout with the sorted label for the selected initial letter unfolds. The second-level sorted circle layout is a concentric circle displayed in the user’s view. The sorted labels are placed on each circle of the concentric circle. The second-level sorted circle layout centers on the selected initial letter and uses the central vision of FOV to compute its radius.

Based on Principle 4 and Principle 5, the labels on the second-level sorted circle layout are placed to indicate the orientation of their anchor objects in the VE.

Given the label set \( L \), the anchor object set \( O \) of \( L \), the user gaze point \( G \), and the user view direction \( d \), the maximum number of circles \( N \) in the second-level sorted circle layout \( MCL \), the maximum number of iterations for relaxation \( N_{\text{r}} \), the second-level sorted...
circle layout \( MCL \) is calculated by Algorithm 1.

**Algorithm 1: Second-level Sorted Circle Layout**

**Data:** label set \( L \), object set \( O \), user gaze point \( G \), user view direction \( d \), maximum number of circles \( N \), maximum number of iterations for relaxation \( #N_{g} \)

**Result:** second-level sorted circle layout \( MCL \)

\[
\begin{align*}
1 & \quad \Pi \leftarrow \text{plane}(G, d); \\
2 & \quad \text{SCL} \leftarrow \text{initArray}(); \\
3 & \quad \text{for } l_{i} \in L \text{ do} \\
4 & \quad \quad \quad \uparrow \\
5 & \quad \quad \quad \text{ran}_{i} \leftarrow \text{project}(\uparrow \Pi); \\
6 & \quad \quad \quad l_{i}\text{.dis} \leftarrow \text{dis}(\uparrow \Pi); \\
7 & \quad \quad \quad l_{i}\text{.rad} \leftarrow \text{rad}(\uparrow \Pi); \\
8 & \quad \quad \quad l_{i}\text{.ran} \leftarrow \text{ran}(\uparrow \Pi); \\
9 & \quad \quad \quad \text{SCL} \leftarrow \text{SCL} + l_{i}; \\
10 & \quad \text{end} \\
11 & \quad \text{SCL} \leftarrow \text{initArray}(N); \\
12 & \quad \text{MCL} \leftarrow \text{maxSortedSubseq} (\text{MCL}[k], \text{SCL}); \\
13 & \quad \text{for } i \in [0, \text{len}(\text{SCL})] \text{ do} \\
14 & \quad \quad \quad \text{MCL}[k], \text{SCL} \leftarrow \text{insert} (\text{MCL}[k], \text{SCL}, \text{SCL}[i]); \\
15 & \quad \text{end} \\
16 & \quad \text{MCL}[k] \leftarrow \text{relax} (\text{MCL}[k], k, #N_{c}); \\
17 & \quad k \leftarrow k + 1; \\
18 & \text{end}
\end{align*}
\]

Each label \( l \) in \( L \) has four attributes: \( \text{dis}, \text{rad}_{p}, \text{rad}, \) and \( \text{ran}. \text{dis} \) is the projection length in screen space of the distance between the anchor object’s position of \( l \) and gaze point position in the VE. Since labels are placed on \( MCL \), we only need to record the label’s radian to determine the position of the label on \( MCL \). \( l\text{.rad} \) is the initial radian of \( l \) in \( MCL \). \( l\text{.rad} \) is the current radian of \( l \) in \( MCL \). \( l\text{.ran} \) is the sliding range of \( l \) in \( MCL \), which is an interval \([l\text{.ran}_{\text{min}}, l\text{.ran}_{\text{max}}]\) formed by \( l\text{.ran}_{\text{min}} \) and \( l\text{.ran}_{\text{max}} \).

Firstly we use the gaze point \( G \) as the center point, and the user’s view direction \( d \) as the normal vector to generate the layout plane \( \Pi \) (line 1). The single circle layout \( SCL \) is initialized in line 2. We initialize the attributes of each label \( l \) in the label set \( L \) and store them in \( SCL \) (lines 3-11). For each label \( l_{i} \) in \( L \) (line 3), we get the vector \( \uparrow \Pi \) from \( l_{i} \)’s anchor object position \( o_{l} \) to the gaze point \( G \) (line 4). Then we project \( \uparrow \Pi \) on \( \Pi \) to get the vector \( \uparrow \Pi \) (line 5). After that, we initialize the attributes of \( l_{i} : \text{dis}, \text{rad}_{p}, \text{rad}, \) and \( \text{ran}. \text{dis} \) is initialized as the modulus length of \( \uparrow \Pi \) (line 6). \( l_{i}\text{.rad} \) is calculated by \text{radian} function (line 7). As shown in Figure 2, we get the intersection \( p \) of \( \uparrow \Pi \) on the edge of the unit circle in the screen space, and \( l_{i}\text{.rad} \) is set as the radian of \( p \) on the edge of the unit circle. \( l_{i}\text{.ran} \) is set as the same as \( l_{i}\text{.rad} \) (lines 8). \( l_{i}\text{.ran} \) is the range \([\text{ran}_{\text{min}}, \text{ran}_{\text{max}}]\), where \( \text{ran}_{\text{min}}, \text{ran}_{\text{max}} \) are calculated by Equation 1 (line 9). After the attributes of \( l_{i} \) are initialized, we add \( l_{i} \) to \( SCL \) (line 10).

**Algorithm 2: Insert**

**Data:** circle layout \( c \), \( SCL \), label \( l \)

**Result:** circle layout \( c \), \( SCL \)

\[
\begin{align*}
1 & \quad l_{i}, l_{j} \leftarrow \text{binarySearch}(l, c); \\
2 & \quad \text{InRan} \leftarrow (l_{i}\text{.rad} \in l\text{.ran}) \lor (l_{j}\text{.rad} \in l\text{.ran}); \\
3 & \quad \text{RanRan} \leftarrow (l_{i}\text{.rad} \in l\text{.ran}) \lor (l_{j}\text{.rad} \in l\text{.ran}); \\
4 & \quad \text{if } \text{InRan} \lor \text{RanRan} \text{ then} \\
5 & \quad \quad \quad l_{k} \leftarrow \text{nearest}(l_{i}, l_{j}, l); \\
6 & \quad \quad \quad l_{k}\text{.rad} \leftarrow \text{median}(l_{i}\text{.ran}, l_{j}\text{.ran}); \\
7 & \quad \quad \quad c \leftarrow c + 1; \\
8 & \quad \quad \quad \text{SCL} \leftarrow \text{SCL} + l; \\
9 & \quad \text{end}
\end{align*}
\]

10 return \( c \), \( SCL \);

\[
\text{ran}_{\text{min}} = l_{i}\text{.rad} - f(l_{i})/2 \\
\text{ran}_{\text{max}} = l_{i}\text{.rad} + f(l_{i})/2 \\
f(l) = (1 - e^{-l\text{.dis}}) \times \pi / 4.0
\]

We initialize the second-level sorted circle layout \( MCL \) in line 12. The second-level sorted circle layout is a concentric circle displayed in the user’s view, which contains up to \( N \) circles. We initialize the current circle index \( k \) of the second-level sorted circle layout \( MCL \) as 0 in line 13. Then we add all labels in the single circle layout \( SCL \) to the specified circle in \( MCL \) (lines 14-21). In line 15, we use Dynamic Programming to get the longest sorted label subsequence in \( SCL \). All labels in the subsequence are removed from \( SCL \) and added to \( MCL \). In order to add more labels to \( MCL[k] \) as many as possible, we traverse each remaining label \( SCL[l] \) in \( SCL \) (line 16), try to add \( SCL[l] \) to \( MCL[k] \) and return the updated \( MCL[k] \) and \( SCL \) by the function \text{insert}. The details of function \text{insert} are shown in Algorithm 2. After all remaining labels in \( SCL \) have performed function \text{insert}, we perform \text{relax} function for \( MCL[k] \) to ensure that the labels arranged on \( MCL[k] \) do not occlude each other (line 19). The details of \text{relax} function are shown in Algorithm 3. Finally, we add \( k \) to 1 (line 20). If there are still remaining labels in \( SCL \), we will continue to iterate and try to arrange these labels in \( MCL[k+1] \).

The details of the function \text{insert} in Algorithm 1 are shown in Algorithm 2. The inputs of Algorithm 2 are the circle layout \( c \), single circle layout \( SCL \), and the label \( l \) need to be inserted into \( c \). Algorithm 2 returns the updated \( c \) and the updated \( SCL \).

**Figure 2: Computation of the label’s radian.**
of \( l \) and the next label \( l'' \) of \( l \) in \( c \) (line 6). After that, we get the next label \( l'' \) of \( l \) in \( c \) (line 7). If \( l \) and \( l'' \) overlap (line 8), we update the radial of \( l \) by the function \( \text{subRad} \) (line 9), and update the radial of \( l'' \) by the function \( \text{addRad} \) (line 10). The details of the functions \( \text{subRad} \) and \( \text{addRad} \) are shown in Equation 2.

\[
\begin{align*}
\text{subRad}(l, l', k) &= l' - \frac{\pi}{2} + \delta, \\
\text{addRad}(l, l', k) &= l' + \frac{\pi}{2} + \delta,
\end{align*}
\]

\( l \cdot \text{rad} = l' \cdot \text{rad} \) (2)

Then we perform the same operation on \( oo[\text{len}(oo) - i] \) as \( oo[0] \) (lines 12-18). After \( oo \) traversal, we initialize \( oo \) as empty, and we use the function \( \text{overlappedArr} \) to add all overlapped labels in \( c \) into \( oo \) (line 20). If \( oo \) is not empty and \( iter \) is larger than \#\( \text{N} \) (line 21), we remove the overlapped label \( oo[0] \) with the maximum degree of overlap in \( oo \) from \( c \) (line 22). Then we add \( iter \) to 1 to perform the next relaxation iteration (line 24). After all relaxation iterations are ended, we return the updated \( c \) (line 26).

Algorithm 3: Relax

\[
\begin{align*}
\text{Data:} & \quad \text{circle layout} \ c, \ \text{circle index} \ k, \ \text{maximum number of iterations for relaxation} \ #\text{N}\text{N}_0 \\
\text{Result:} & \quad \text{circle layout} \ c \\
\text{iter} & \leftarrow 0; \ oo \leftarrow \emptyset; \\
oo & \leftarrow \text{overlappedArr}(c, k); \\
\text{while} \ oo \neq \emptyset \text{ or } iter < \#\text{N}\text{N}_0 \text{ do} \\
\text{for} \ i \in [0, \text{len}(oo)] \text{ do} \\
\text{l} & \leftarrow oo[i]; \\
l_1, l_2 & \leftarrow \text{getPreNextLab}(c, l); \\
l_r & \leftarrow \text{getNextLab}(c, l); \\
\text{if} \ \text{overlap}(l, l_1) \text{ then} \\
l & \leftarrow \text{subRad}(l, l_1, k); \\
l_1, l_2 & \leftarrow \text{addRad}(l, l_1, k); \\
\text{end} \\
l & \leftarrow oo[\text{len}(oo) - i]; \\
l_1, l_2 & \leftarrow \text{getPreNextLab}(c, l); \\
l_r & \leftarrow \text{getNextLab}(c, l); \\
\text{if} \ \text{overlap}(l, l_1) \text{ then} \\
l & \leftarrow \text{subRad}(l, l_1, k); \\
l_1, l_2 & \leftarrow \text{addRad}(l, l_1, k); \\
\text{end} \\
oo & \leftarrow \emptyset; \ oo \leftarrow \text{overlappedArr}(c, k); \\
\text{if} \ oo \neq \emptyset \text{ and } iter > \#\text{N}\text{N}_0 \text{ then} \\
\text{c} & \leftarrow c - oo[0]; \\
\text{end} \\
\text{iter} & \leftarrow iter + 1; \\
\text{return} \ c.
\end{align*}
\]

3.3 View and Gaze based Label Guidance

In this section, we propose the view and gaze based label guidance method for guiding the user to locate the target object in VR more efficiently. The view and gaze based label guidance method first generates a specific flying trajectory for each candidate label to ensure that the distance between the candidate label and the user is even. During the process of the label guidance, the flying candidate labels will not be too close or too far away from the user. During the process of the label guidance, this method updates the flying speed of the candidate labels to ensure that the user can keep up with the flying candidate labels efficiently. Moreover, it keeps valid candidate labels and removes the invalid ones in real time to ensure that the user locates the correct candidate label.

We compute the specific flying trajectory \( \Psi \) of each candidate label \( l \) based on the initial position \( p_l \), terminal position \( p_e \), and the user’s viewpoint position \( p_v \). The details are shown in Figure 3. We take two trivial points \( p_{m1} \) and \( p_{m2} \) in the line segment formed between \( p_l \) and \( p_e \). Then, we get a vector \( \vec{v}_{m1} \) passing through \( p_{m1} \) with \( p_l \) as the start point, the modulus length of \( \vec{v}_{m1} \) is \( |p_l - p_{m1}| \), and mark the end point of \( \vec{v}_{m1} \) as \( p_{m1}' \). Similarly, we get \( \vec{v}_{m2} \) passing through \( p_{m2} \) with \( p_l \) as the start point, the modulus length of \( \vec{v}_{m2} \) is \( |p_l - p_{m2}| \), and mark the end point of \( \vec{v}_{m2} \) as \( p_{m2}' \). Finally, we use \( p_v \), \( p_{m1}' \), \( p_{m2}' \), and \( p_e \) to generate a Bezier curve as \( \Psi \).

![Figure 3: Visualization of the flying trajectory generation.](image)

The larger \( \alpha \) means that the direction between \( \vec{v}_{m1} \) and \( \vec{v}_{m2} \) is closer, which means that the guidance effect is better and the flying speed of \( l \) can be faster. We also record the distance \( d_{dis} \) between \( l \) and the gaze point \( p_e \) in screen space. The larger \( d_{dis} \), the more difficult for the user to follow \( l \). And the flying speed of \( l \) should be reduced. The flying speed \( s \) of \( l \) is calculated by Equation 4.

\[
s = s + (1 - s) \times \alpha \times d_{dis}
\]
At last, we keep valid candidate labels and remove the invalid ones during the label guidance. For each flying candidate label $l$, if the angle between the $l$’s flying direction $\vec{v}_l$ and the gaze moving direction $\vec{v}_g$ is greater than 90°, then $l$ has no effect on locating the target object in VR, and $l$ will be regarded as an invalid label and removed from the candidate labels in the process of label guidance.

4 RESULTS AND DISCUSSION: USER STUDIES

We design user studies to evaluate the performance of our label guidance based object locating method. We first design a pilot study to explore the effect of using label guidance to locate objects (Sect 4.1). Then, we conduct a user study to further evaluate our label guidance based method’s efficiency and task load. (Sect 4.2).

4.1 Pilot Study

The intuitive idea is that when there are only a few objects in the VE, the object locating task can be completed efficiently without additional guidance or only using a simple label guidance method. Thus, we design a pilot study to explore the effect of using the label guidance method in the VE with different numbers of labels.

There are two scenes and three experimental conditions in the pilot study. The two scenes are the same except for the number of objects, 16 and 60, respectively. For the target object locating task, a traditional method is that users browse the entire scene to visually search for the target object (PCC1). When there are too many objects and labels, it will be difficult to search for the target object. A straightforward solution is to arrange the labels of objects on the screen in circular and alphabetical order, and then the user finds the target object according to the target label (PCC2). The third condition is our label guidance based object locating method (PEC).

We recruit N=12 participants (4 female, 8 male) aged between 23 and 28. 3 of the participants had experience with VR applications. We use a within-subject design with each participant completing each of the 5 tasks in 3 conditions $\times$ 2 scenes. The task is to locate the target object whose name is displayed at the center of the screen. Once the participant finds the target object, they will indicate it by pointing the handle ray at it and pressing a button.

Table 1: Pilot study object locating time, in seconds. Statistical difference is denoted with an asterisk.

| Scene | Condition | Avg ± std. dev. | Comparison | $p$ |
|-------|-----------|-----------------|------------|-----|
| S1    | PCC1      | 10.5 ± 6.9      | PCC1 - PEC | 0.025* |
|       | PCC2      | 9.8 ± 1.8       | PCC2 - PEC | < 0.001* |
|       | PEC       | 6.3 ± 1.2       |            |     |
| S2    | PCC1      | 22.3 ± 16.3     | PCC1 - PEC | < 0.001* |
|       | PCC2      | 12.1 ± 2.9      | PCC2 - PEC | < 0.001* |
|       | PEC       | 7.0 ± 1.2       |            |     |

Figure 4: EC1: single sorted circle layout in the second level(left). EC2: strict orientated circle layout in the second level (right).

**Hypotheses** The two-level hierarchical sorted and orientated label layout (EC3) is designed to improve the efficiency of object locating in scenes with a large number of labels. The two-level hierarchical layout reduces the number of labels displayed on the screen by clustering labels according to initial letter, while the sorted and orientated label layout reduces the layout circle number. The view and gaze based label guidance optimizes label motion trajectory and speed. Thus, we formulate the following hypotheses.

**H1:** Using the hierarchical circles to find labels (EC1, EC2, and EC3) will be more efficient than the traditional method (CC1 and CC2). Using the two-level hierarchical sorted and orientated label layout (EC3) will be even more efficient than using a single sorted circle layout (EC1) and strict orientated circle layout (EC2).

**H2:** The user task load with the hierarchical circles (EC1, EC2, and EC3) will be lower than traditional method (CC1 and CC2), and user task load with EC3 will be lower than those of EC1 and EC2.

**H3:** EC3 is easy to use.

**Experimental scene.** The experimental scenes are shown in Figure 5. The workbench scene is an indoor scene. 90 tools and instruments are regularly placed on the workbench. The pipe factory is an outdoor scene. 87 intricate pipes are irregularly arranged. They will partially block each other.

**Hardware.** We use the HTC Cosmos VR systems with two hand-held controllers, allowing the users to awake label guidance in

---

Table 1 shows the completion time for the three conditions in two scenes. The data conforms to the normal distribution by Shapiro-Wilk test [25]. PCC1, PCC2 are compared with PEC by using ANOVA. Whether in the scene of few objects (S1) or in the scene of a large number of objects (S2), the efficiency in our method (PEC) is significantly higher than that of surrounding visually search (PCC1) and arranging all labels on the screen (PCC2).

The reduction in the time overhead of our method for locating the object is due to the fast search for a given label from the two-level hierarchical sorted and orientated label layout in the presence of a large number of labels. It is necessary to apply our method.

We also compare the same method in different scenes by using ANOVA. The result is: (PCC1, $p = 0.006^*$), (PCC2, $p = 0.004^*$) and (PEC, $p = 0.069$). As the number of objects in the scene increases, the efficiency of traditional methods, whether surrounding visually search (PCC1) or arranging all labels on the screen (PCC2), will become significantly worse. However, the performance of our method is independent of the number of labels.
the VE. The HMD is connected to the workstation with a 3.6GHz Intel(R) Core(TM) i7-9900KF CPU, 16GB of RAM, and an NVIDIA GeForce GTX 3080 graphics card.

Participants. We recruit N=32 participants (12 female, 20 male), whose ages are between 19 and 33. 14 of our participants had experience with VR applications.

Procedure. Before the experiment starts, we allow participants to fully train in a simple scene until they fully understand the details of the five methods. When the participant finds the target object, he should point the handle ray at the target object and presses the end key to complete an object locating trial and then a prompt to continue the next trial will appear in the center of the screen. The participant repeats the above steps. During the whole process, the participant can only turn around and move his head. We balance the order of each condition in a Latin square (5 groups). Each participant completes 150 study trials: 5 conditions × 2 scenes × 15 trials. The participant can take breaks between each trial. After the experiment, the participant fills out subjective post-experiment questionnaires.

Metrics. We use two objective metrics: time and the rotation angle. Time to locate an object in each trial starts when the participant presses the start key of the handle and ends when the participant points to the target object and presses the end key. We record the cumulative fovTime of the target object in the central FOV to remove outliers. If the fovTime takes up more than 50% of the time, we abandon the data of this trial. We use rotation angle to record the accumulated head rotation angle of the participant. The moment to record rotation angle is the same as recording time.

We record the perceived value with two subjective metrics: user task load, measured with a standard NASA TLX questionnaire [9, 10] and usability of our method, measured with a 5-point Likert scale question. The Likert scale has five questions. Each question needs to score from 1 to 5. 1 means extremely disagree, and 5 means extremely agree. These five questions are: Q1: whether the two-level hierarchical sorted and oriented label layout (EC3) can effectively improve the efficiency? Q2: whether EC3 is very simple and easy to learn? Q3: whether EC3 dose not confuse you when completing tasks? Q4: whether EC3 is convincing and reasonable? Q5: whether you enjoy using EC3 in the future?

Statistical analysis. The time to locate the target object, the task load and the usability scores are compared across the five conditions (CC1, CC2, EC1, EC2, and EC3) by using a one-way repeated measures ANOVA. Firstly, we use the Shapiro-Wilk test [25] to verify the distribution normality assumption. Secondly, we use the Mauchly test [19] to verify the sphericity assumption. When the sphericity assumption is violated, we apply a Greenhouse-Geisser correction. Then, we conduct a population ANOVA to investigate whether the null hypothesis of no statistically significant difference between the five conditions could be rejected. When null hypothesis is rejected (p < 0.05), the differences between the four pairs (EC3 vs CC1, CC2, EC1, EC2) are analyzed by post-hoc tests, using the Bonferroni correction to reduce the level of significance (α < 0.016). We use Cohen’s d [4] to quantify the effect size.

4.2.2 Experimental results

The time and the pairwise comparisons among conditions are displayed in Table 2. The time in the workbench and pipe both violate the sphericity assumption. After applying the Greenhouse-Geisser correction, the overall ANOVA reveals significant differences between the five conditions: (F1,126.27.036 = 27.768, p < 0.001) for the workbench scene and (F1,420.73.860 = 56.009, p < 0.001) for the pipe scene. Post-hoc analysis shows that the time of EC3 is significantly shorter than that of CC1, CC2, EC1, and EC2 in both scenes. The effect size for all pairwise is V.large or higher.

Table 2: The time to locate object, in seconds. Statistical significance (p < 0.016) is denoted with an asterisk.

| Scene | Condition | Avg ± std. dev. (XC-EC) | p Cohen’s d | Effect size |
|-------|-----------|--------------------------|-------------|-------------|
| CC1   | 25.4 ± 12.9 | 74.2% < 0.001* | 2.05 | Huge        |
| CC2   | 11.6 ± 3.89  | 43.7% < 0.001* | 1.75 | V.large     |
| S1    | EC1       | 8.8 ± 1.6 | 25.2% < 0.001* | 1.51 | V.large     |
| S2    | EC2       | 8.4 ± 1.4 | 21.6% < 0.001* | 1.31 | V.large     |
| EC3   | 6.6 ± 1.3  |              |             |             |

Table 3: The Rotation angle during object locating, in degrees. Statistical significance (p < 0.016) is denoted with an asterisk.

| Scene | Condition | Avg ± std. dev. (XC-EC) | p Cohen’s d | Effect size |
|-------|-----------|--------------------------|-------------|-------------|
| CC1   | 642 ± 379 | 87.9% < 0.001* | 2.09 | Huge        |
| CC2   | 114.9 ± 45.5 | 32.5% 0.010* | 1.04 | Medium      |
| S1    | EC1       | 99.4 ± 37.7 | 21.9% 0.011* | 0.64 | Medium      |
| S2    | EC2       | 75.2 ± 29.4 | -3.32% 0.509 | 0.08 | V.small     |
| EC3   | 73.7 ± 29.6 |              |             |             |

The rotation angle and the pairwise comparisons among conditions are displayed in Table 3. The rotation angle in workbench scene and pipe scene both violate the sphericity assumption. After applying the Greenhouse-Geisser correction, the overall ANOVA reveals significant differences between the four conditions (F1,029.51.456 = 100.676, p < 0.001) for the workbench scene and (F1,028.44.240 = 22.583, p < 0.001) for the pipe scene. Post-hoc analysis shows that EC3 rotation angle is significantly smaller than CC1, CC2 and EC1 but not significantly different from EC2. The effect size between EC3 and CC1 is "Large" or higher. The effect size between EC3 and CC2 is "Very large" for pipe scene.

The NASA-TLX scores are shown on Figure 6. The positive score performance is replaced with its complement so that smaller is always more favorable. We test the sphericity assumption on six aspects of NASA-TLX and apply the Greenhouse-Geisser correction when necessary. The mental is verified with p = 0.666, the physical is verified with p = 0.546, the temporal is verified with p = 0.972, the performance is violated with p < 0.001, the
effort is verified with \( p = 0.327 \) and the frustration is verified with \( p = 0.211 \). After applying the Greenhouse-Geisser correction on performance, the overall ANOVA reveals significant differences with \( F(1.097, 5.485) = 9.516, p = 0.023 \). Compared with CC1, EC3 has significant improvement in all six aspects. Compared with CC2, EC3 has significant improvement in five aspects except in temporal. Compared with EC1, EC3 significantly improves mental, performance, effort, and frustration. Compared with EC2, EC3 has significant improvement in mental, performance and effort. For the overall score, EC3 has significant improvement compared with CC1, CC2, EC1, and EC2.

![Figure 6: NASA-TLX scores for individual questions. Significant difference are denoted with the asterisk and line.](image)

We design a Likert scale to investigate the usability of our two-level hierarchical sorted and orientated label layout. The results show that participants believe that EC3 can help them improve efficiency (4.92), and they feel easy (4.25) and comfortable when using EC3. Overall, EC3 does not confuse them (3.89). The layout of EC3 is reasonable (4.25), and they enjoy (4.88) using it in the future.

### 4.2.3 Discussion

The study is conducted to evaluate the performance of our two-level hierarchical sorted and orientated label layout for the object locating task in VR. The experimental results show that our method achieves a significant improvement over other methods in both efficiency and task load, and our method is easy to use.

For H1, the mean value and standard deviation in Table 2 show that using the hierarchical circles (EC1, EC2 and EC3) is much faster and requires fewer head rotation angles than the traditional methods (CC1 and CC2). ANOVA analysis shows that EC3 is significantly faster than other methods and significantly reduces the accumulated head rotation angle compared with CC1, CC2 and EC1. However, there is no significant difference in the rotation angle between EC2 and EC3. EC3 is more efficient than CC1, CC2, EC1, but not more efficient than EC2 because of no significant difference in the rotation angle. The results do not support H1.

Firstly, we discuss time. Compared with the traditional methods, the user only needs to retrieve the initial clustering labels to find the target label instead of changing his perspective to retrieve all labels. Compared with EC1, EC3 encodes the orientation information, and thus EC3 places the labels closer to corresponding objects. Compared with EC2, the insertion and relaxation in EC3 reduce the number of label layout circles. Compared with one label selection based on dwell-time in EC1 and EC2, EC3 selects multiple candidate labels based on the movement of gaze point. To sum up, EC3 is significantly faster than other methods. Secondly, we discuss rotation angle. CC1, CC2 and EC1 do not consider the orientation between the label and the object. The two processes of the user searching the label and locating the object are unrelated. In EC2 and EC3, the orientation of label and the gaze point is basically the same as object and the gaze point, so no additional head rotation angle will be introduced. However, compared with the strict orientated circle layout in the second level (EC2), changing the position of label on the circle in EC3 may cause the guidance path to be not optimal, which introduces an additional rotation angle.

For H2, Figure 6 shows that the mean score of EC1, EC2 and EC3 in six aspects is lower than that of CC1 and CC2. Compared with other methods, EC3 has significantly improvement in mental, performance and frustration. The load of EC3 is significantly lower than that of CC1, CC2, EC1, and EC2. The results support H2.

CC1 requires the user to wrap around and change his perspective to search, memorize and compare from different perspectives, and thus the task load of CC1 is the highest. CC2 arranges all labels on one screen. Limited by the FOV of the device, the user has to change his perspectives to find the target label. In addition, the selection method based on dwell time also brings trouble to users, which makes it necessary for users to move the gaze point carefully to select. The label layout of EC3 is more reasonable and the label selection of EC3 is more natural, thus the task load of EC3 is lower than both EC1 and EC2. The discussion about H1 can explain that the task load of EC3 in performance is significantly lower than EC1 and EC2.

For H3, the effort of NASA-TLX task load shows that the method in EC3 requires less effort. The results support H3. The result of Likert scale questions shows that users believe that EC3 is easy to learn and that they will enjoy using EC3 in the future.

## 5 Conclusion, limitations and future work

We have proposed an efficient object locating method based on label guidance to improve the efficiency of locating the target object in VR applications. A two-level hierarchical sorted and orientated label layout is designed to provide sort and orientation cues of the target object. A view and gaze label guidance method is introduced to improve the efficiency of locating the target object. Our method achieves a significant improvement in efficiency and a significant reduction of task load.

There are some limitations in our method. The first limitation is that the efficiency of our method will reduce in the VE that has many labels with the same initials. Although all labels are arranged in alphabetical order on each circle of the second-level sorted circle layout, the second-level sorted circle layout will contain many circles in this case, so it still takes much time for the user to find the candidate labels. Thus, one possible future work is to propose an adaptive range calculation method, which dynamically expands the range of the labels with the same initials. The second limitation is that the flying trajectories of some candidate labels may be similar when there are many objects with the same initials in the same region of the VE, which will reduce the accuracy of the object locating. The second future work is to introduce a trajectory similarity parameter to ensure the uniqueness of the flying trajectory. When there are multiple similar flying trajectories, these trajectories are deformed by the trajectory similarity parameter to ensure the uniqueness of each trajectory. The third limitation is that our method can not guide the user to locate the target object efficiently when the other object completely occludes the target. So another possible future work is to combine our method with multiperspective visualization to remove occlusions of the target object in the process of label guidance. The fourth limitation is the relationship between label placement and label size. We slide the handle button to resize labels so that the labels can display information clearly without overly occluding the scene. We fix the size of the labels before the user study. We do not consider the dynamic size, which needs to be synchronized and enlarged when moving to the target object.

## Acknowledgments

This work was supported by National Key RD plan 2019YFC1521102, by the National Natural Science Foundation of China through Projects 61932003 and 61772051.
References

[1] A. Bodonyi and R. Kunkli. Efficient object location determination and error analysis based on barycentric coordinates. *Visual Computing for Industry, Biomedicine, and Art*, 3(1):1–7, 2020.

[2] F. Bork, C. Schneidler, U. Eck, and N. Navab. Towards efficient visual guidance in limited field-of-view head-mounted displays. *IEEE transactions on visualization and computer graphics*, 24(11):2983–2992, 2018.

[3] L. Cmolik, V. Pavlovec, H.-Y. Wu, and M. Nollenburg. Mixed labeling: Integrating internal and external labels. *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[4] J. Cohen. Statistical power analysis for the behavioral sciences. *Academic press*, 2013.

[5] M. Fink, J.-H. Haurnert, A. Schulz, J. Spoorhase, and A. Wölfle. Algorithms for labeling focus regions. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2583–2592, 2012.

[6] R. Grasset, T. Langlotz, D. Kalkofen, M. Tatzgern, and D. Schmalstieg. Image-driven view management for augmented reality browsers. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 177–186. IEEE, 2012.

[7] U. Gruenefeld, D. Enhenga, A. E. Ali, W. Heuten, and S. Boll. Eyecatch: Designing a visualization technique for out-of-view objects in head-mounted augmented reality. In *Proceedings of the 5th symposium on spatial user interaction*, pp. 109–118, 2017.

[8] U. Gruenefeld, D. Lange, L. Hammer, S. Boll, and W. Heuten. Flynningarrow: pointing towards out-of-view objects on augmented reality devices. In *Proceedings of the 7th ACM international symposium on pervasive displays*, pp. 1–6, 2018.

[9] S. Hart. Nasa-task load index (nasa-tlx): 20 years later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50:904–908, 2006.

[10] S. Hart and L. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Advances in psychology*, 52:139–183, 1988.

[11] N. Heisohun, A. Gerasch, and M. Kaufmann. Boundary labeling methods for dynamic focus regions. In *2014 IEEE Pacific Visualization Symposium*, pp. 243–247. IEEE, 2014.

[12] J. Jia, S. Elezovikj, H. Fan, S. Yang, J. Liu, W. Guo, C. C. Tan, and H. Ling. Semantic-aware label placement for augmented reality in street view. *The Visual Computer*, 37(7):1805–1819, 2021.

[13] D. Koufli, L. Cmolik, B. Kozlíková, H.-Y. Wu, G. Johnson, D. S. Goodsell, A. Olson, M. E. Gröller, and I. Viola. Labels on levels: labeling of multi-scale multi-instance and crowded 3d biological environments. *IEEE transactions on visualization and computer graphics*, 25(1):977–986, 2018.

[14] E. Kruijff, J. Orlosky, N. Kishishita, C. Trepkowski, and K. Kiyokawa. The influence of label design on search performance and noticeability in wide field of view augmented reality displays. *IEEE transactions on visualization and computer graphics*, 25(9):2821–2837, 2018.

[15] V. Lehtinen, A. Oulasvirta, A. Salovaara, and P. Nurmi. Dynamic tactile guidance for visual search tasks. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pp. 445–452, 2012.

[16] T. Lin, Y. Yang, J. Beyer, and H. Pfister. Labeling out-of-view objects in immersive analytics to support situated visual searching. *IEEE Transactions on Visualization and Computer Graphics*, 2021.

[17] R. W. Lindeman, Y. Yanagida, J. L. Silbert, and R. Lavine. Effective vibrotactile cueing in a visual search task. In *Proc. of Interact 2003*, pp. 89–96, 2003.

[18] A. Marquardt, C. Trepkowski, T. D. Eibich, J. Maiiero, E. Kruijff, and J. Schöning. Comparing non-visual and visual guidance methods for narrow field of view augmented reality displays. *IEEE Transactions on Visualization and Computer Graphics*, 26(12):3389–3401, 2020.

[19] J. W. Mauchly. Significance test for sphericity of a normal n-variate distribution. *The Annals of Mathematical Statistics*, 11(2):204–209, 1940.

[20] J. P. McIntire, P. R. Havig, S. N. Watamaniuk, and R. H. Gilkey. Visual search performance with 3-d auditory cues: Effects of motion, target location, and practice. *Human factors*, 52(1):41–53, 2010.

[21] A. McNamara, K. Boyd, J. George, W. Jones, S. Oh, and A. Sutherland. Information placement in virtual reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 1765–1769. IEEE, 2019.

[22] S. Peterson, M. Axholt, and S. R. Ellis. Managing visual clutter: A generalized technique for label segregation using stereoscopic disparity. In *2008 IEEE Virtual Reality Conference*, pp. 169–176. IEEE, 2008.

[23] P. Renner and T. Pfeiffer. Attention guiding techniques using peripheral vision and eye tracking for feedback in augmented-reality-based assistance systems. In *2017 IEEE symposium on 3D user interfaces (3DUI)*, pp. 186–194. IEEE, 2017.

[24] B. Schwirdfeger and G. Klinker. Supporting order picking with augmented reality. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pp. 91–94. IEEE, 2008.

[25] S. S. Shaphiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52:591–611, 1965.

[26] J. Talmon, E. Ammenwerth, J. Breder, N. De Keizer, P. Nykänen, and M. Rigby. Stare-hi—statement on reporting of evaluation studies in health informatics. *International journal of medical informatics*, 78(1):1–9, 2009.

[27] M. Tatzgern, D. Kalkofen, R. Grasset, and D. Schmalstieg. Hedgehog labeling: View management techniques for external labels in 3d space. In *2014 IEEE Virtual Reality (VR)*, pp. 27–32. IEEE, 2014.

[28] M. Tatzgern, D. Kalkofen, and D. Schmalstieg. Dynamic compact visualizations for augmented reality. In *2013 IEEE Virtual Reality (VR)*, pp. 3–6. IEEE, 2013.

[29] M. Tatzgern, V. Orso, D. Kalkofen, G. Jaccucci, L. Gambirini, and D. Schmalstieg. Adaptive information density for augmented reality displays. In *2016 IEEE Virtual Reality (VR)*, pp. 83–92. IEEE, 2016.

[30] C. Trepkowski, D. Eibich, J. Maiiero, A. Marquardt, E. Kruijff, and S. Feiner. The effect of narrow field of view and information density on visual search performance in augmented reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 575–584. IEEE, 2019.

[31] E. Van der Burg, C. N. Olivers, A. W. Bronkhorst, and J. Theeuwes. Pip and pop: nonspatial auditory signals improve spatial visual search. *Journal of Experimental Psychology: Human Perception and Performance*, 34(5):1053, 2008.

[32] W. Xu, H.-N. Liang, Y. Zhao, T. Zhang, D. Yu, and D. Monteiro. Ring-text: Dwell-free and hands-free text entry for mobile head-mounted displays using head motions. *IEEE transactions on visualization and computer graphics*, 25(5):1991–2001, 2019.

[33] B. Zhang, Q. Li, H. Chao, B. Chen, E. Ofek, and Y.-Q. Xu. Annotating and navigating tourist videos. In *Proceedings of the 15th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 260–269, 2010.

[34] Z. Zhou, L. Wang, and V. Popescu. A partially-sorted concentric layout for efficient label localization in augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 27(11):4087–4096, 2021.