This article has been accepted for publication in the proceeding of this conference, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI 10.1109/ISMAR-Adjunct.2019.00-62, 2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct).

A two-point map-based interface for architectural walkthrough

Kan Chen* Eugene Lee

Fraunhofer Singapore

ABSTRACT

Architectural walkthrough of 3D buildings usually requires the user to manipulate the position and orientation of a camera for exploration, measurement, and annotation. In this paper, we propose a novel two-point interaction method by interacting with the corresponding 2D floor plans to navigate in 3D buildings. Our method is based on the observation that, naturally, people tend to relate standing tall to seeing far.

Index Terms: Human-centered computing—Interaction design—Interaction design process and methods—User interface design

1 INTRODUCTION

Architectural walkthrough has many applications, such as virtual 3D room presentation (or exploration) for real estate agents (clients), as well as 3D space examination for architects to examine and discuss their designs. However, 3D building walkthrough with only a 3D view using the conventional mouse keyboard-based camera manipulation [3], which is also popular in first person shooter (FPS) games, still give many users unpleasant experiences. This is because the users may often get disoriented or lost due to their current 3D view is only limited in the local viewing space or there are too many similar 3D rooms and levels. On the other hand, many users like architects and real estate clients, compared with a 3D representation, are also more familiar and comfortable with 2D floor plans, which can provide more general overviews of buildings and clearer representations of different levels. Enabling the cross-referencing of 2D floor plans and the corresponding 3D buildings can be beneficial in these applications.

Besides, 2D touch interfaces are currently very popular and largely available in hand-phones and tablets. Existing 2D floor plan-based multi-touch interaction for architectural walkthough still has limitations due to the gap between the limited degrees of freedom (DOFs) in 2D touch and higher DOFs requirement for 3D camera control. For example, the multi-touch architecture demo from Fraunhofer IGD [1] cannot easily handle looking up and down. The iKiosk from UniqueAT [4] requires one additional multi-touch display for looking up and down.

Our map-based interaction aims to navigate in the 3D scene by manipulating the two essential camera settings (camera position and look at position) on the 2D map. By doing so, we can achieve the 2D-3D co-referencing and have better overall views, also fewer operators (two-point) can be more appreciated. Our proposed intuitive interface tackles the difficult problem of closing the gap mentioned above between 2D and 3D based on this observation: as an old Chinese saying, “stand tall and see far”, it is natural that people tend to go high (raising head) to look far and lower head to see up close. This means, naturally, the users can expect the camera is tilting up if its look at position is moving far away, and vice versa. In another word, our approach intuitively utilizes the distance information from the camera's position and look at to achieve tilting (pitch), so as to augment the 2D horizontal directional (yaw) information from them.

*e-mail: {chen.kan, eugene.lee}@fraunhofer.sg

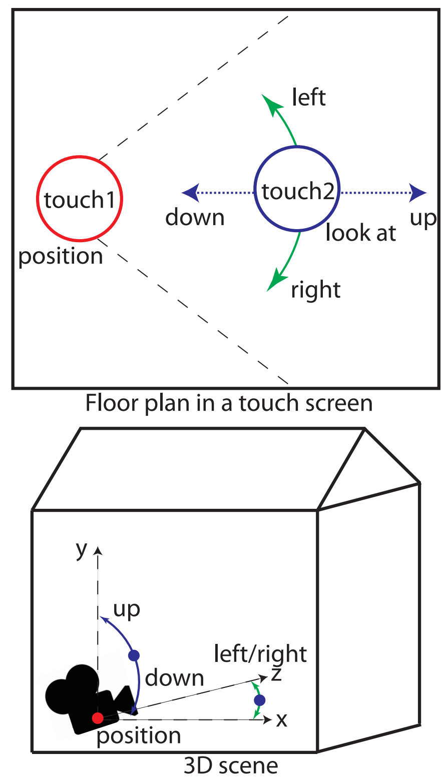

Figure 1: Two-touch interaction on the floor plan for manipulating the camera in the corresponding 3D scene. Touch1 controls camera's position. Touch2 controls the cameras' look at position, left/right: orbiting around touch1, up/down: moving away or closer to touch1.

2 OUR METHOD

In our method, the inputs are 2D floor plans and corresponding 3D buildings, they are aligned and matched to each other. To this end, the locations and orientations of the points in 2D floor plans and 3D buildings are the same. A pre-processing may be performed to ensure they are matched to each other, this can be done manually. The setup (Figure 2) can be one multi-touch display for showing the 2D floor plans of the buildings and handling the interactions with a secondary display for building visualization in 3D, or a side-by-side layout using one multi-touch display. The proposed new interaction schemas as shown in Figure 1 are as follows.

(1) The two-point interactions are performed on the 2D floor plans of different height ($levelHeight$). The first touch point ($touch1$) represents the position of the camera ($camPos$), by tapping on the user desired location or dragging the icon to there on the 2D floor plans, the camera of the 3D scene can move to the corresponding location.

$$\begin{aligned} camPos_{x,z} &= M \cdot touch1_{x,y}, \\ camPos_y &= levelHeight - 0.5 * viewHeight, \end{aligned} \tag{1}$$

This article has been accepted for publication in the proceeding of this conference, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI 10.1109/ISMAR-Adjunct.2019.00-62, 2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct).

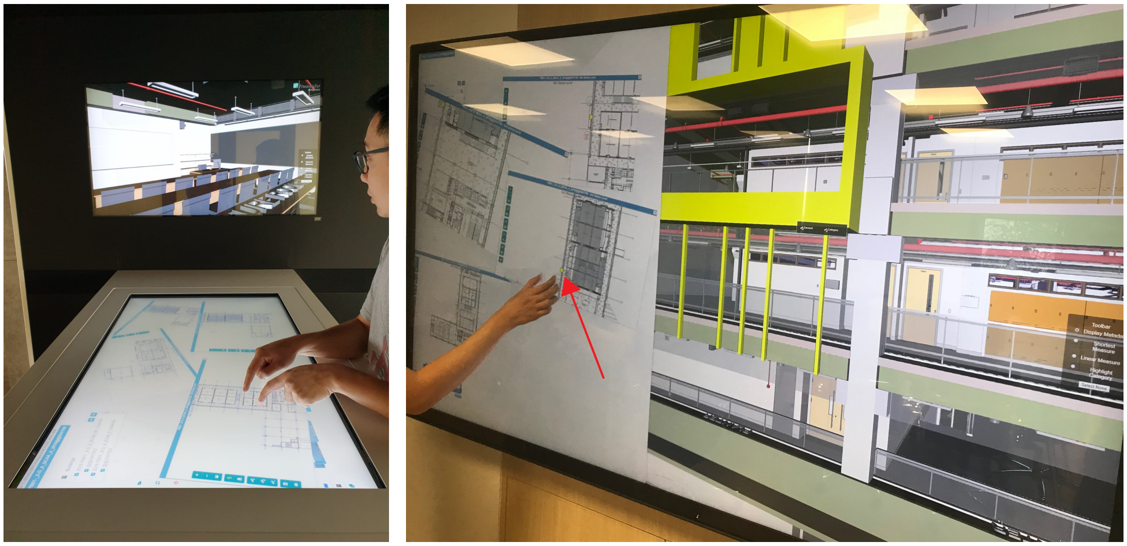

Figure 2: Two possible setups. Left: one multi-touch display and one secondary display. Right: one multi-touch display, the red arrow points to the highlighted element in 2D corresponds to the yellow element in 3D.

where M is a matrix maps the 2D map coordinate to 3D scene coordinate system, and $levelHeight$ is the current floor's height (one floor plan associates one height), and $viewHeight$ represents the camera's look at position's height range, we use 2.4 in our case. Note that the tapping at the desired location will invoke the positioning of both the $touch1$ ($camPos$) and $touch2$ ($camLookat$), whose positions are aligned, in this case, simultaneous dragging two icons is not needed.

(2) The second touch point ($touch2$) represents the look at position ($camLookat$). By orbiting it around the first touch ($camPos$), we can change the camera's horizontal looking direction (left/right, yaw):

$$camLookat_{x,z} = M \cdot touch2_{x,y}. \tag{2}$$

(3) The second touch controls the vertical looking direction (up/down, pitch) by moving it away or closer to the first touch. We propose to use a sigmoid-like function to map the 2D distance between touch2 and touch1 to the height of the camera's look at position in 3D:

$$camLookat_y = camPos_y + \lambda(dist(touch1, touch2)/r)^f, \tag{3}$$

where λ is a scaling factor to map the numerical result based on the touching range r to the range of $viewHeight$, f is a tilting factor, greater f means, the tilting up of the camera happens later but faster, $dist(touch1, touch2)$ is the map distance between two touch points. In our experiments, we use $2.4 \cdot 10^{-6}$ as λ, 400 as r and 6 as f.

(4) Using the camera position and look at position in 3D, we can easily construct the camera matrix for updating the camera in 3D (e.g. using gluLookAt function). Note that, if the user moves the camera position ($touch1$) away from its look at position ($touch2$), the $camLookat_y$ is resetting to $camPos_y$, no matter it was looking up or down, using the same Equation (3), but a faster (larger) f.

Note that the above schemas (also shown in the video) are the main features of the proposed interaction method. There are also other notable related design and implementation pointers as follows, which are helpful to facilitate the walkthrough.
(1) A secondary interaction mode can be toggled by tapping the $touch1$: in this mode, the current camera's look at ($camLooka$) is saved and retained, only the $touch1$ can take effect.
(2) The locations of the two touch points are registered with the 2D floor map, after the map operations like zooming, panning and rotating, their locations are still maintained correctly.
(3) In our implementation, to work with different multi-touch displays, some tolerances are needed in measuring touch interaction to reduce camera oscillation. A slider to adjust $camPos_y$ and other functions like measurement, clipping, reset are also implemented in our prototype. Moreover, the same elements in the 2D plan as well as in 3D scene can be highlighted by tapping on it, as shown in Figure 2, Right. This can be beneficial for 2D-3D co-referencing.

3 Evaluation and Future Work

Comparing with the existing touch-based navigation systems [1,2,4], our method is based on an intuitive paradigm by relating standing tall and seeing far. Also, it is more effective in handling looking up and down without additional touch screen [4] or joystick handle ([2]). As an expert review, we invited groups of domain experts (20+), such as architects, BIM managers, and construction managers, to review our prototype. They all have more than 10 years of industry experiences. They tested our interface and found it is easy-to-use and intuitive. They like our interface and plan to use it in their discussions/case studies/site visits. They also gave constructive comments for us to improve, such as the functionality to focus only on the partial building. Our prototype is currently deployed in one company for evaluation and we plan to conduct more user studies.

Acknowledgments

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative.

References

[1] Fraunhofer IGD, Bene Office Furniture. *Bene Multitouch Exponat*. Demo: https://www.youtube.com/watch?v=sTUCRew8DOE, 2010.
[2] Nintendo. *Nintendo 3DS*. https://www.nintendo.com/3ds/, 2011.
[3] M. J. C. Samonte, Z. N. F. Recio, B. G. Jumaquio, O. P. N. Roxas, and J. S. Tamares. An emerging interactive 3d architectural design walkthrough system for industrial hotel builders. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 760–765. IEEE, 2018.
[4] UniqueAT. *iKiosk*. https://uniqueat.com/interactive-kiosk/, 2019.