Classification of Interaction Techniques in the 3D Virtual Environment on Mobile Devices

Eliane Balaa[1], Mathieu Raynal[1], Youssef Bou Issa[2], and Emmanuel Dubois[1]

[1] IRIT, University of Toulouse, France
[2] Faculty of Engineering, Antonine University, Baabda, Lebanon
elianebalaa@yahoo.com

Abstract. 3D Virtual Environments (3DVE) are more and more used in different applications such as CAD, games, or teleoperation. Due to the improvement of smartphones hardware performance, 3D applications were also introduced to mobile devices. In addition, smartphones provide new computing capabilities far beyond the traditional voice communication. They are permitted by the variety of built-in sensors and the internet connectivity. In consequence, interesting 3D applications can be designed by enabling the device capabilities to interact in a 3DVE. Due to the fact that smartphones have small and flat screens and that a 3DVE is wide and dense, mobile devices present some constraints: the environment density, the depth of targets and the occlusion. The pointing task faces these three problems to select a target. We propose a new classification of the existing interaction techniques, according to three axis of classification: a) the three discussed problems (density, depth and occlusion); b) the first two subtasks of the pointing task (navigation, selection); and c) the number of targets selected by the pointing technique (1 or N). In this paper we will begin by presenting a state of the art of the different pointing techniques in existing 3DVE, structured around three selection techniques: a) Ray casting, b) Curve and c) Point cursor. Then we will present our classification, and we will illustrate the classification of the main pointing techniques for 3DVE. From this classification, we will discuss the type of interaction that seems the most appropriate to perform this subtask optimally.

Keywords: Interaction techniques, 3D Virtual environment, mobile devices, environment density, depth of targets, occlusion, Augmented Reality.

1 Introduction

Nowadays, with built-in mobile operating system, smartphones provide new computing capability and connectivity far beyond traditional voice communication [12]. Internet connectivity and applications, games, gyroscopic sensor [13], built-in camera, accelerometer, touchpad and touch screen are very common on such devices. In parallel, three-dimensional technologies have been recently introduced in different applications and areas such as desktop computing and CAD [8] (computer aided design systems), 3D games, VE (virtual environments), AR (augmented reality), ubiquitous computing and tele-operation [19]. Extending the use of 3D applications onto

R. Shumaker and S. Lackey (Eds.): VAMR 2014, Part I, LNCS 8525, pp. 3–13, 2014.

smartphone provides a greater value to both. Few applications to the concept are launched: 3D games and Head-tracking are some examples. Interesting 3D applications on mobile devices can be designed through the mapping of the device abilities to the interaction requirements in the 3D environment. Consequently, the smartphone will be promoted as the new generation of multi-use device. Researchers in 3D virtual environment have developed many new techniques and metaphors for 3D interaction to improve 3D application usability on mobile devices. But they are all facing the same kinds of problems or factors that are specific to the use of 3D on mobile devices. Indeed due to the fact that smartphones have small and flat screens and that a 3DVE is wide and dense with a large number of targets of various sizes, mobile devices present specific constraints:

1. The environment density [17]: to display a large number of targets with various sizes on the relatively small screen of mobile devices, the scene must be shrunk, and the virtual space size and its components must be minimized. Various objects are very close to each other. In this case, how to accurately select one of these objects?
2. The depth of targets [7]: the elements of the environment are displayed in 3D and are identified by three coordinates (X, Y, Z). They are arranged in several levels of depth while the display is a flat touch screen (x,y). But how to point at the appropriate depth to reach an object that is in depth and behind other, even transparent, layers? Depth causes two problems: On the one hand, to know where the pointer is in depth in relation to the target; on the other hand, to point target in depth quickly and with accuracy adds additional complexity to the pointing task.
3. The occlusion [18]: One or more elements of the scene can hide the target either partially or totally and thus reduce its visibility and ability to be selected. Furthermore, the user uses his finger to point a target using the touch screen which creates an additional occlusion while targeting a small object. How to reach an object if it is hidden by another or by the interaction process?

Designing an appropriate interaction technique for 3D on mobile devices must therefore consider these constraints in order to identify the most appropriate solution to the selected context.

In light of these three major concerns, and to contribute to the comparison and design of interaction techniques in 3D on mobile devices, we propose a new classification space. The originality of our approach is to particularly emphasize usage constraints (density, depth, and occlusion) in the context of the pointing task. It is no longer a technologically centered approach but a real user centered approach.

2 Classification

As a first step we chose to focus on the pointing task only, among the three well known tasks of Bowman [1]. Indeed, the pointing task, which is the most used, faces the three mentioned problems. Despite the large amount of different techniques proposed in the literature, these three problems are never solved at the same time. Typically, suggested interaction techniques are used to solve one or two of the three problems. To refine this analysis and better compare or design the pointing techniques in 3D, we propose to consider the pointing task as a set of three sequential subtasks:

1. Navigation refers to exploring the content of the 3DVE in different ways to search for the target and visualize it [4]. In this phase, the user may change his viewpoint through the environment or move the scene to visualize the target.
2. Selection refers to moving a pointer to reach the target position once the target is visible. [5]
3. Validation consists of validating or confirming the pointing task, when the pointer reaches accurately the target. When the technique selects multiple targets during the selection task, the validation task can also select the desired target from the N selected targets. [1]

In addition, the target environment can be either dense or sparse. In order to select a target in a dense environment, multiple objects close to the target may be hard to avoid while trying to select the target. As a result, the target will have to be selected from a first subset of objects. In a sparse environment, such kind of problem doesn't occur, and thus only one object is selected. Consequently we will consider two cases: a) the interaction techniques selecting multiple objects and: b) the interaction techniques selecting 1 object.

Finally problems that have been raised above can therefore be involved at different steps of the pointing task. This is why we propose a new classification of the existing interaction techniques, according to three axis of classification: a) the three subtasks of the pointing tasks (navigation, selection, validation); b) the three discussed problems (density, depth and occlusion) on the first two subtasks of the pointing task, and c) the number of targets selected by the pointing technique during the validation subtask (1 or N).

Table 1. Classification structure

1	2	3	4	5	6	7	8
PT	Navigation			Selection			Validation
	Dens.	Depth	Occl.	Dens.	Depth	Occl.	1 or N

Our classification space consists in a double-entry table. Each pointing technique (PT) is represented in a line. Columns depict the three sub-task of the pointing task. Subtasks are depicted by the columns and divided in sub columns with each encountered problems:

- The first column presents the name of the pointing technique and its reference.
- Columns 2 to 4 relate to the navigation subtask
  - In column 2, the target is in a high density environment but it is visible. Usually, navigation will not solve this problem of density because its goal is only to display the target. However, some interaction techniques can indirectly use navigation to bring target closer and therefore make easier the selection task.

  - In column 3, the target is in depth, but visible. Like density, navigation task is not necessarily required, but can be used to facilitate the selection task.
  - In column 4, the target is occluded. To visualize it, the user has to change the viewpoint searching for the target. This case covers interaction techniques that benefit from the Navigation to solve the occlusion problem.
- Columns 5 to 7 relate to the selection subtask
  - In column 5, the user must point a target that is close to several others (density problem). The technique must allow the selection with accuracy of the target among the N surrounding.
  - In column 6, the user must be able to point with the pointing technique the target that would be distant in depth. The technique must enable to quickly and accurately select this target.
  - In column 7, the position target is known but the target is occluded or partially occluded. The technique must allow the selection of the desired target despite obstacles.
- Column 8 treats the validation problem. We distinguish situations where one target is selected during the selection phase and the case where multiple targets are selected. In this last case, it requires a new selection from N preselected targets.

For each cell of the table, we note "-" if the sub-task is not covered by the pointing technique. If the subtask is covered, we note "**P**" if the problem is present but is not resolved by the pointing technique; "**PS**" if the problem is partially solved and "**S**" if the problem is solved by the pointing technique.

Thus this table allows to quickly see the advantages and disadvantages of a pointing technique, and also allows comparing two techniques together. This representation also allows us to study more precisely the resolution of a problem for a subtask by analyzing the various solutions proposed for this particular column.

In section 3 we use this classification space and its sub-categories to compare existing interaction techniques. We highlight the differences of the various techniques to demonstrate that existing classifications do not sufficiently highlight the weaknesses and forces of each technique.

3 Existing Techniques

A well-established classification [4] proposes to classify the selection technique used in 3D Virtual Environment into 3 categories: point cursor, ray casting and curve. We will present the main techniques of these three categories and show the main problems they face.

The simplest pointing technique consists of moving a pointer in the 3DVE. The pointer movement is controlled by a pointing device. For example, 3D Point cursor [5] is an interaction technique where a cursor is moved in the virtual environment to select a visible target. When the user press a button, the 3D position and orientation of an input device are linearly transformed into coordinates in the virtual environment where a cursor is displayed. To reach the target, the user moves his input device. Consequently, the coordinates of the cursor change. If the cursor points at the target, the target is highlighted. Then the selection is confirmed by clicking on the input device button. However, this task is complicated since it requires more accurate movements and multiple adjustments in depth.

In case of dense and crowded environment, the target can be occluded. Two problems may occur: the first is the occlusion of the target and the second is the density problem: due to the fact that the target is surrounded by multiple objects that makes the selection of this target a complicated task.

To point the target precisely, the user has to make fine displacement of the crosshair. To point the target accurately without making a big number of displacements, researchers introduce the volume selection with multiple occlusion levels. In the Silk cursor technique [20], the cursor is replaced by a semi-transparent rectangular volume cursor. The user hand is transformed into a silk cursor. When the target is inside the rectangular volume, the user close his hand and the selection is ended. The occlusion levels are used to solve the depth problem. Silk cursor provides information on where the object is placed relatively to another object with no measure of how much they are spatially separated. In the transparent sphere technique [4], the crosshair is replaced by a transparent sphere and the target is one of the objects inside the sphere. To select it, this technique uses list menu or circulation techniques. The menu technique solves the density problem but lose the information about the depth position.

In summary, pointing techniques, based on the point cursor, do not support the navigation phase. They require an additional accuracy effort from the user to point a target in depth. In the selection phase, Silk cursor and transparent sphere partially solve the depth problems. In transparent sphere, the density problem is partially resolved because it requires the search of the target from a set of items.

With the ray casting techniques, the depth problem is solved. In ray cursor [2,5] (see Fig.1-A), the user controls the origin position and the orientation of a virtual ray by adjusting the input device position. Then, a virtual ray is emitted from the user through the target direction. This ray intersects the targeted object. The first object intersected by the virtual ray is highlighted and the selection is confirmed by clicking the input device button. Therefore, this technique does not solve the occlusion problem because if a target is positioned in front of the desired target, it is the first target which is selected and not the desired target. In Go-Go technique [10] (see Fig.1-B), the virtual ray is replaced by a virtual hand. Go-Go is an arm-extension technique using a nonlinear transformation. The user real hand is transformed in a virtual hand in the virtual environment. The virtual hand is kept on the ray extending from the torso and going through the physical hand. Go-Go is limited by the real hand length and the difficulty of grabbing; these limitations are solved by the HOMER [3] interaction technique which combines a ray casting and an arm-extension techniques. HOMER can select objects at any position with no restrictions and perform easy selection and manipulation.

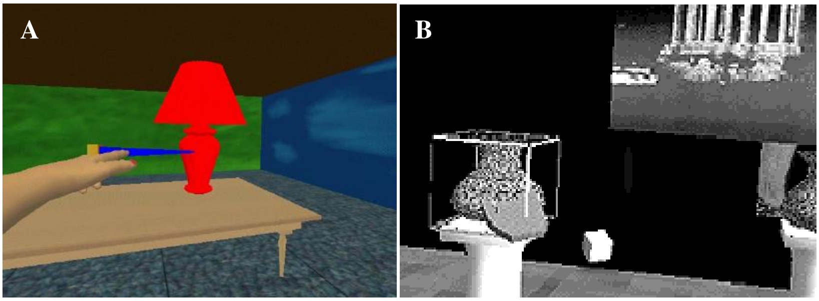

Fig. 1. A) Ray casting technique [3] B) Go-go technique [10]

In the selection phase, Ray cursor, Go-Go and HOMER solve the depth problem thanks to the virtual ray projected to infinity. These techniques are used in a sparse environment to point a visible target. However, they do not solve the occlusion problem because the pointer selects the first target encountered in the case of the ray cursor technique and a visible target in the case of the Go-Go and HOMER techniques. The density problem is not solved either: the Go-Go and HOMER techniques need multiple adjustments to point the target and the ray cursor requires a higher selection precision. Consequently, in case of a dense environment, these techniques are not accurate to select an object.

To solve the occlusion problem, when the ray passes through multiple targets, different techniques are proposed to identify and point the correct target from the group of the aligned objects. The "Ray With Variable Length" technique [4] aims to change the ray length by changing the position of the ray endpoint. The closest objet to the endpoint is highlighted and selected when the user pushes a tracked wand button. Another technique is the Depth ray [5], in addition to the ray, a depth marker travels between the objects following the input device displacements. Each time it comes close to an intersected object, the object becomes red. However, the Depth ray uses a continuous displacement of the input device that can cause modifications of the ray position. This limitation is solved by the Lock ray technique [5] which is a Depth ray version aiming of locking the ray then the depth marker appears to avoid confusion between selection and disambiguation phases. Another technique consists of using a menu to select a target from the N pointed objects of the ray casting. The menu technique consists of displaying the visible and the occluded objects intersected by the ray. Then the target is selected from the menu. Different techniques exist: a 2D flower menu in the Flower Ray technique [5] (which is a 3D extension of the splatter technique [11]) ; a virtual sphere menu in the Daisy menu technique [8] ; a circular menu in the Ring menu technique [8] or a list menu in the Floating menu technique [4, 11]. Using a ray cursor and adding a menu technique solves the occlusion and the depth problems in the selection phase. The limitation of these combined techniques is in the huge number of objects displayed in the menu. These techniques are efficient if the ray crosses exactly the target. Otherwise a density problem is detected.

To overcome the density problem, some techniques use a volume rather than a ray. A selection volume is defined by a ray and the volume around it. This volume can have different forms: a) a cone in the case of the spotlight technique [8] or a cylinder in the case of the transparent cylinder technique [4]. The transparent cylinder technique consists of displaying a cylinder around the ray (see Fig. 2 A.). All objects contained in the cylinder are selected. Then the target is selected using a circulation or Floating menu techniques. The huge number of objects inside the cylinder is solved using a dynamic selection volume (case of the spotlight). The spotlight technique consists of displaying a cone around the ray. The spread angle of the cone is adjusted to minimize the number of objects inside it. Then the target is selected using a special numeric algorithm. Once found, the area around the object is gradually shrunk to minimize the total number of the selected objects. Consequently, the transparent dynamic selection volume solves the density problem, in the selection phase.

Another way to solve the density problem consists of designing hybrid techniques. They combine techniques based on the point cursor metaphor to others based on the ray casting metaphor: for example, the hybrid technique formed by the combination of a ray cursor and a bubble cursor technique [5], or the one formed by the combination of a ray cursor and a transparent sphere technique [4] (see Fig. 2 B.). To do that, these techniques add a dynamic sphere volume at the ray endpoint to augment the selection area and minimize the pointing problem in a dense volume. The hybrid technique formed by the combination of the ray cursor and the bubble cursor provides a shape transformation and warp to select one target. And the hybrid technique formed by the combination of the ray cursor and the transparent sphere selects a group of objects. Based on the number of objects inside the dynamic selection area, researchers propose: a) circulation or menu techniques or b) a special numeric algorithm to select one target. In the selection phase, these techniques solve the depth and the density problems.

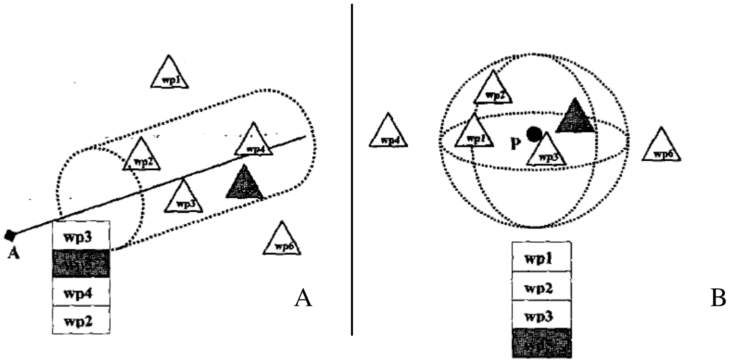

Fig. 2. A) Transparent cylinder and B) transparent sphere technique [4]

To resolve the occlusion problem during the selection phase, Curve technique presents a possible solution. Curve with fixed length [4] is an interaction technique where a fixed curve is moved in the virtual environment to select a visible target. Unlike the curve with fixed length where the user has to move many time in order to select a target, a Curve with variable length [4] technique move the destination endpoint along the three dimensions in order to select its target. In the selection phase, Curve with fixed length and Curve with variable length solve the depth and the occlusion problem because the curve help avoid targets which is in front of the desired target. In case of dense and crowded environment, the target can be occluded and the curve will select multiple objects. The curve techniques are based on the transparency of the environment. Thus, the user can detect the position of the target and make the adjustment to reach it. The Flexible pointer [9] technique is visualized by a flexible curve. The flexible pointer direction is determined by vector formed by the two hands. The amount of curvature is defined by the orientation of each the hand. It is used to point visible, partially or fully occluded objects and to point around objects. Using a curved ray has the advantage of selecting a deep target without passing through the environment objects. The problem of density is solved by the flexibility of the curve in the case of the flexible pointer technique or by using the Virtual pointer metaphor technique [16]. The Virtual pointer metaphor is a ray casting metaphor, using a dynamic area selection and a menu or selection algorithm, to point the target and then to draw a Bezier curve graph pointing the target. Curve also solves the depth problem. The transparency added to the curve metaphor solves the occlusion problem. Solving the density problem is achieved by two methods a) the form and the length of a flexible curve in a transparent environment or b) a ray based technique in a non-transparent environment. In the selection phase, the Flexible pointer length and the Virtual pointer metaphor solve the depth, the occlusion and the density problems.

Most of the presented techniques are only concerned with the selection phase. The navigation subtask is not supported by the proposed techniques and requires different approaches. This complicates the interaction task in the 3DVE. The following techniques cover the selection and the navigation phases. Using the two ray selection technique [4], a first virtual ray is emitted. Multiple targets could be intersected. To select the target from those that have been intersected by the first ray, the user changes his point of view and emits a new ray. The target is located at the intersection of the two virtual rays. The shadow cone [14] used the same principle, using virtual cone instead of rays. The Smart ray [5] proposes to emit several rays in different viewpoints (see Fig. 3). For each ray emitted, the technique adds a weight to the selected objects and the weight are adjusted in function of two conditions: a) if the target is hit several times by the ray and b) if the ray is closer to the center of the selected object. The target with the highest weight can be selected. In conclusion, in the selection and the navigation subtasks, these techniques solve the occlusion and the depth problems.

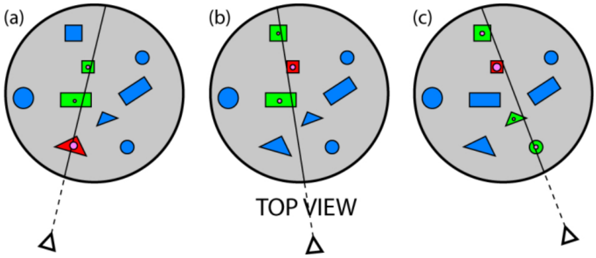

Fig. 3. Smart ray [5]

At last, the world-in-miniature (WIM) technique [15] uses a small representation of the environment to manipulate and interact with the environment objects. The user holds the small version. He uses a simple virtual hand for indirect manipulations of the full-scale object by interacting with their icon in the small version. The user manipulates the miniature objects then the full-scale objects are affected. The WIM is a point cursor technique and used for selection and navigation.

4 Discussion

Error! Reference source not found. summarizes the classification of the techniques we have presented in Section 3. We grouped the techniques in relation to the classification of [4].

Table 2.

PT		Navigation			Selection			Validation
		Dens	Depth	Occl	Dens	Depth	Occl	
Point cursor	3D Point cursor [5]	-			P	P	S	1
	Silk cursor [20]	-			P	PS	S	1
	Transparent sphere [4]	-			S	PS	S	2
	WIM [15]	S			P	PS	S	1
Ray casting	Ray cursor [2,5]	-			P	S	P	1
	Go-Go [10]	-			P	PS	PS	1
	Homer [3]	-			P	S	S	1
	Variable Length [4]	-			P	S	S	N
	Depth ray [5]	-			P	S	PS	1
	Lock ray [5]	-			P	S	S	2
	spotlight [8]	-			S	S	S	2
	Transparent cylinder [5]	-			S	S	S	2
	Two ray selection [4]	S			S	S	S	2
	Shadow cone [14]	S			S	S	S	2
	Smart ray [5]	S			S	S	S	N
Hybrid	Ray cursor + Bubble cursor [5]	-			S	S	P	1
	Ray cursor + Transparent sphere[4]	-			PS	S	S	1
Curve	fixed or variable length [4]	-			P	PS	S	1
	Flexible pointer [9]	-			P	S	S	1
	Virtual pointer metaphor [16]	-			P	S	S	1

This table allows us to draw some information about the different techniques. At first, we can see that few of them can make both selection and navigation. The techniques we present are more specific to the selection.

We also note that point cursor techniques do not solve the problem of depth: it is difficult to locate the pointer position in depth relative to the target. In addition, the pointing task in depth is difficult with a pointer.

The depth problem is solved using ray casting techniques. However, this technique causes the occlusion problem. Indeed, the virtual ray emitted can cross another target before the desired. In this case, the validation should be done in several stages. On the one hand, point to the right target with the virtual ray, and then select the desired target among those crossed by the virtual ray. This generates two selection and/or validation steps. Therefore, it may increase the pointing time.

The occlusion problem can be solved by curves techniques. Curves techniques allow user to avoid targets located in front of the desired one. However, it poses afresh the depth problem because curves must be moved, and it can cause several movements to point precisely target.

5 Conclusion

In light of these considerations, this work proposes a categorization of interaction techniques. Our classification offers a different way of analyzing the existing interaction techniques which may help in the process of evaluating such techniques. The use of such categorization help users to choose the ones best suited for their needs, to improve existing techniques or to design new techniques. This new classification is based on two criteria: problems and factors affecting pointing performance and the interaction phases.

References

1. Bowman, D.A.: Principles for the Design of Performance-oriented Interaction Techniques, ch. 15
2. Bowman, D.A., Johnson, D.B., Hodges, L.F.: Testbed evaluation of virtual environment interaction techniques. In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST 1999), pp. 26–33. ACM, New York (1999)
3. Bowman, D.A., Hodges, L.F.: An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In: Proceedings of the 1997 Symposium on Interactive 3D Graphics (I3D 1997), pp. 35–38. ACM, New York (1997)
4. Dang, N., Le, H., Tavanti, M.: Visualization and interaction on flight trajectory in a 3D stereoscopic environment. In: IEEE 2003, pp. 9.A.5-1–9.A.5-10 (2003)
5. Grossman, T., Balakrishnan, R.: The Design and Evaluation of Selection Techniques for 3D Volumetric Displays. In: Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST 2006), pp. 3–12. ACM, New York (2006)
6. Grossman, T., Balakrishnan, R.: The bubble cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2005), pp. 281–290. ACM, New York (2005)
7. Hwang, M.S., Jeon, J.W.: Design of the 3D Input Method Based on Touch Device for Mobile. In: Fifth International Joint Conference on INC, IMS and IDC IEEE, NCM 2009, pp. 1607–1610 (2009)

8. Liang, J., Liang, O., Green, M.: Geometric Modeling Using Six Degrees of Freedom Input Devices. In: 3rd Int'l Conference on CAD and Computer Graphics (1993)
9. Olwal, A., Feiner, S.: The Flexible Pointer: An Interaction Technique for Augmented and Virtual Reality. In: UIST 2003, ACM Symposium on User Interface Software and Technology, pp. 81–82 (2003)
10. Poupyrev, I., Billinghurst, M., Weghorst, S., Ichikawa, T.: The Go-Go Interaction Technique: Non-linear Mapping for Direct Manipulation in VR. In: Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology (UIST 1996), pp. 79–80. ACM, New York (1996)
11. Ramos, G., Robertson, G., Czerwinski, M., Tan, D., Baudisch, P., Hinckley, K., Agrawala, M.: Tumble! Splat! Helping Users Access and Manipulate Occluded Content in 2D Drawings. In: Proceedings of the Working Conference on Advanced Visual Interfaces (AVI 2006), pp. 428–435. ACM, New York (2006)
12. Robbins, D.C., Cutrell, E., Sarin, R., Horvitz, E.: ZoneZoom: Map Navigation for Smartphones with Recursive View Segmentation. In: Proceedings of the Working Conference on Advanced Visual Interfaces (AVI 2004), pp. 231–234. ACM, New York (2004)
13. Schäfers, T., Rohs, M., Spors, S., Raake, A., Ahrens, J.: Designing Low-Dimensional Interaction for Mobile Navigation in 3D Audio Spaces. In: Proceedings of 34th International Conference of the Audio Engineering Society, AES (2008)
14. Steed, A., Parker, C.: 3D Selection Strategies for Head Tracked and Non-Head Tracked Operation of Spatially Immersive Displays. In: Proceedings of 8th International Immersive Projection Technology Workshop, pp. 13–14 (2004)
15. Stoakley, R., Conway, M., Pausch, R.: Virtual Reality on a WIM: Interactive Worlds in Miniature. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 1995), pp. 265–272. ACM Press/Addison-Wesley Publishing Co, New York, NY (1995)
16. Steinicke, F., Ropinski, T., Hinrichs, K.: Object selection in virtual environments using an improved virtual pointer metaphor. In: Proceedings of International Conference on Computer Vision and Graphics (ICCVG), pp. 320–326 (2006)
17. Vanacken, L., Grossman, T., Coninx, K.: Exploring the effects of environment density and target visibility on object selection in 3D virtual environments. In: IEEE Symposium on 3D User Interfaces, pp. 117–124 (2007)
18. Vanacken, L., Grossman, T., Coninx, K.: Multimodal selection techniques for dense and occluded 3D virtual environments. International Journal of Human-Computer Studies 67(3), 237–255 (2009)
19. Zhai, S., Milgram, P., Rastogi, A.: Anisotropic human performance in six degree-of-freedom tracking: An evaluation of three-dimensional display and control interfaces. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 27(4), 518–528 (1997)
20. Zhai, S., Buxton, W., Milgram, P.: The "Silk Cursor": Investigating transparency for 3D target acquisition. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 1994), pp. 459–464. ACM, New York (1994)