Contents lists available at SciVerse ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Section on Touching the 3rd Dimension

A survey of 3D object selection techniques for virtual environments

Ferran Argelaguet [a],*, Carlos Andujar [b]

[a] VR4i Research Group, INRIA Rennes, France
[b] MOVING Research Group, Universitat Politècnica de Catalunya, Spain

A R T I C L E I N F O

Article history:
Received 31 August 2012
Received in revised form
7 December 2012
Accepted 11 December 2012
Available online 20 December 2012

Keywords:
3D interaction
3D selection
Virtual pointing
Virtual reality

A B S T R A C T

Computer graphics applications controlled through natural gestures are gaining increasing popularity these days due to recent developments in low-cost tracking systems and gesture recognition technologies. Although interaction techniques through natural gestures have already demonstrated their benefits in manipulation, navigation and avatar-control tasks, effective selection with pointing gestures remains an open problem. In this paper we survey the state-of-the-art in 3D object selection techniques. We review important findings in human control models, analyze major factors influencing selection performance, and classify existing techniques according to a number of criteria. Unlike other components of the application's user interface, pointing techniques need a close coupling with the rendering pipeline, introducing new elements to be drawn, and potentially modifying the object layout and the way the scene is rendered. Conversely, selection performance is affected by rendering issues such as visual feedback, depth perception, and occlusion management. We thus review existing literature paying special attention to those aspects in the boundary between computer graphics and human–computer interaction.

1. Introduction

In the last decades we have witnessed enormous improvements in spatial input devices and motion tracking systems. These advances have motivated the development of a plethora of interaction techniques relying on six DoFs (Degrees of Freedom) input devices and user gestures. Interaction through natural gestures is gaining further popularity since the recent mass commercialization of low-cost solutions for full-body tracking, which is enabling the deployment of natural interfaces outside virtual reality labs. We will use the term 3D interaction to refer to interaction tasks requiring users to make some gestures in free (unconstrained) 3D space. These gestures typically involve one or both hands, and might also involve the user's head and other parts of the body.

The design of appropriate 3D interaction techniques for virtual environments (VEs) is a challenging problem [19,51]. On the positive side, interacting in free space with natural gestures opens a new world of possibilities for exploiting the richness and expressiveness of the interaction, allowing users to control simultaneously more DoFs and exploiting well-known real-world actions. On the negative side, 3D interaction is more physically demanding and might hinder user tasks by increasing the required dexterity. Compare for example the act of selecting an object using a mouse pointer to that of grasping a 3D object in free space. Mouse movement involves small, fast muscles whereas grasping often requires a complex arm movement involving larger and slower muscles [23,48]. Furthermore, current immersive VEs, even the most sophisticated ones, neither fail to provide the same level of cues for understanding the environment, nor reproduce faithfully the physical constraints of the real world [74]. For this reason, although humans are used to perform 3D interaction gestures in the real world, users of IVEs often encounter difficulties in understanding 3D spatial relationships and controlling multiple DoFs simultaneously.

Object selection is one of the fundamental tasks in 3D user interfaces [19] and the initial task for most common user interactions in a VE. Manipulation tasks often depend on (and are preceded by) selection tasks. As a consequence, poorly designed selection techniques often have a significant negative impact on the overall user performance. In this survey, we review major 3D interaction techniques intended for 3D object selection tasks. We do not consider indirect selection techniques, e.g. selecting from a menu or performing semantic queries. A 3D object selection technique requires the user to gesture in 3D space, e.g. grabbing an object or pointing to something (see Fig. 1). Two main 3D selection metaphors can be identified: virtual hand [78] and virtual pointing [63,54]. In the early days, virtual hand techniques were more popular as they map identically virtual tasks with real

* Corresponding author. Tel.: +33 299847487.
E-mail addresses: fernando.argelaguet_sanz@inria.fr (F. Argelaguet), andujar@lsi.upc.edu (C. Andujar).

0097-8493/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cag.2012.12.003

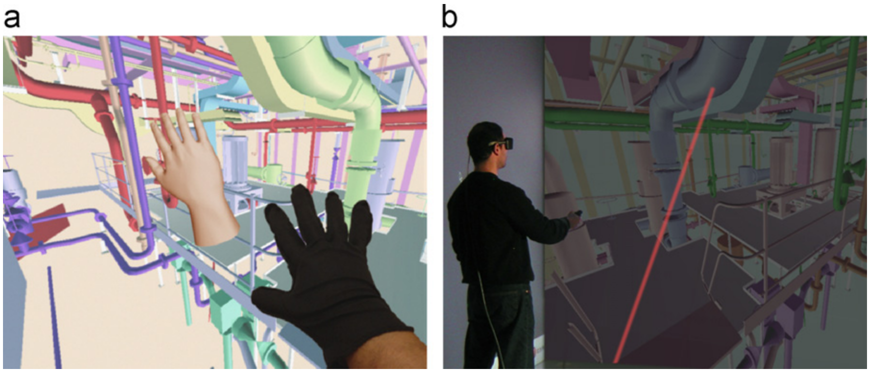

Fig. 1. Object selection using different metaphors and devices: (a) virtual hand and (b) virtual pointing through a hand-held spatial input device.

tasks, resulting in a more natural interaction. Lately, it has been shown that overcoming the physical constraints of the real world provides substantial benefits, e.g. letting the user select objects out of reach by enlarging the user's virtual arm [75], or using virtual pointing techniques such as raycasting [63]. In fact, raycasting selection is one of the most popular techniques for 3D object selection tasks [16]. A number of user studies in the literature have found that virtual pointing techniques often result in better selection effectiveness than competing 3D selection metaphors [19]. Unlike classical virtual hand techniques, virtual pointing techniques allow the user to select objects beyond their area of reach and require relatively less physical movement.

Selection through virtual pointing, though, is not free from difficulties. The selection of small or distant objects through virtual pointing remains to be a difficult task. Some techniques address the selection of small objects by increasing the size of the selection tool [36,73], at the expense of requiring disambiguation mechanisms to guess the object the user aims to select [30]. Noise from tracking devices and the fact that the interaction takes place in free space with no physical support for the hands [55] further hinders the accurate selection of small targets [43]. The user also has to keep the tool orientation steady until the selection confirmation is triggered, for example, by a button press. Such a confirmation action is likely to produce a change in the tool orientation, nicknamed Heisenberg effect [20], potentially causing a wrong selection. Occlusion is another major handicap for accomplishing spatial tasks [33]. Most interaction techniques for 3D selection and manipulation require the involved objects to be visible. A common solution for selecting occluded objects is to navigate to an appropriate location so that the targets become unoccluded. However, this navigate-to-select approach is impractical for selection-intensive applications. Therefore occlusion management techniques are often essential for helping users discover and access potential targets.

A number of approaches have been proposed to improve user performance in terms of task completion times and error counts [15]. A common strategy is to apply human control models such as the optimized initial impulse model [62] and Fitts' Law [34,35]. While the optimized initial impulse model refers to the accuracy a user can achieve given the movement required to perform an action, Fitts' Law estimates the time required to acquire a target. However, as users are bounded by human motor skills, there is a natural trade-off between speed and accuracy. In a typical scenario, high-accuracy rates will produce high task completion times and vice-versa.

In the context of the real usage of 3D interfaces, the subjective impressions of the users about an interaction technique can play a larger role than merely speed. The inability to select objects precisely may prove to be overly annoying and thus frustrate users. A performance increase might not be desirable if it is achieved at the expense of increasing the cognitive load of the task, or using techniques requiring extensive training.

The rest of this paper is organized as follows. Section 2 reviews existing human pointing models. In Section 3 we review major techniques for 3D object selection and extend previously proposed classifications [18,76,29] with a number of additional criteria to further elucidate the potential benefits and drawbacks of existing selection techniques. A comprehensive summary of the reviewed techniques is given in Table 1. Section 4 analyzes major factors influencing selection performance and proposes some usability guidelines. Finally, Section 5 provides some concluding remarks and future research directions.

2. Human pointing models

In order to point to (*acquire*) an object (*the target*), the user is required to perform a set of gestures (*movements*) to position the selection tool (e.g. his finger) over it. For each movement, the final position of the selection tool (*endpoint*) determines whether the acquisition is accomplished (the endpoint is inside the target) or not (the endpoint is outside the target). Once the target is acquired, the user has to trigger some selection mechanism to confirm the acquisition (e.g. pressing a button).

Pointing tasks involving physical interaction are constrained by the human psychomotor behavior. Several human pointing models have been proposed in order to model these aiming movements, to allow a better understanding of the processes involved and provide reliable prediction models of performance. From all the existing human motor models, Fitts' law provides by far the most successful and complete explanation. Fitts' law is one of the few quantitative measures in human–computer interaction and has motivated the development of guidelines for improving 2D and 3D pointing tasks. These guidelines are discussed in Section 4.

2.1. Fitts' law

Fitts' law [34], which emerged from experimental psychology, is a well known human psychomotor behavior model which has been widely adopted in numerous areas, including human factors, ergonomics and human–computer interaction. Fitts' law estimates the time required to perform an aimed movement considering only the physical properties underlying the acquisition task (the size of the target and the amplitude of the movement required to acquire it).

The most common formulation of Fitts' Law was proposed by MacKenzie [58] which asserts that the time T to acquire a target of effective width W which lies at a distance A is governed by the

Table 1
Summary of the classification of selection techniques. (x,y,z,θ,φ) refers to the dominant hand position, and yaw and pitch angles. $(x_n,y_n,z_n,\theta_n,\varphi_n)$ refers to the user's non-dominant hand and $(x_e,y_e,z_e,\theta_e,\varphi_e)$ to the user's eye. We assume a user-centered coordinate system.

Technique	Selection tool	Selection control DoFs			Disambiguation mechanism	CD ratio	Motor and visual space relationship
		Origin	Orientation	Dominant			
Virtual-hand [64]	Hand avatar	(x,y,z)	None	(x,y,z)	N/A	Isomorphic	Offset/clutching
Go–go [75]	Hand avatar	(x,y,z)	None	(x,y,z)	N/A	Anisomorphic	Offset/clutching CD ratio
Bubble-cursor [93]	Adjustable sphere	(x,y,z)	None	(x,y,z)	Heuristic	Isomorphic	Offset/clutching
Silk cursor [107]	Axis aligned box	(x,y,z)	None	(x,y,z)	N/A	Isomorphic	Offset/clutching
Raycasting [63]	Ray	(x,y,z)	(θ,φ)	(θ,φ)	N/A	Isomorphic	Coupled
Virtual pads [1]	Ray	(x,y,z)	(θ,φ)	(θ,φ)	N/A	Anisomorphic	Coupled
Direct image plane [53]	Ray	(x,y,z)	None[a]	(x,y)	N/A	Isomorphic	Offset/clutching
Raycasting from the eye[8]	Ray	(x_e,y_e,z_e)	(θ,φ)	(θ,φ)	N/A	Isomorphic	Coupled
View finder [7]	Ray	(x_e,y_e,z_e)	(θ,φ)	(θ,φ)	N/A	Isomorphic	Coupled
Eye-gazed selection [87,27]	Ray	(x_e,y_e,z_e)	(θ_e,φ_e)	(θ_e,φ_e)	N/A	Isomorphic	Coupled
Occlusion selection [73]	Ray	(x_e,y_e,z_e)	(x,y,z)	(x,y)	N/A	Isomorphic	Offset
One-eyed cursor [98]	Ray	(x_e,y_e,z_e)	(x,y,z)	(x,y)	N/A	Isomorphic	Offset/clutching
Two-handed pointing [64]	Ray	(x,y,z)	(x_n,y_n,z_n)	(x,y,z,x_n,y_n,z_n)	N/A	Isomorphic	Coupled
IntenSelect [30]	Ray	(x,y,z)	(θ,φ)	(θ,φ)	Behavioral	Isomorphic	Coupled
Smart ray [41]	Ray	(x,y,z)	(θ,φ)	(θ,φ)	Manual	Isomorphic	Coupled
Sticky ray [85]	Cone	(x,y,z)	(θ,φ)	(θ,φ)	Heuristic	Isomorphic	Coupled
Flashlight [54]	Cone	(x,y,z)	(θ,φ)	(θ,φ)	Heuristic	Isomorphic	Coupled
Sense shapes [68]	Cone	(x,y,z)	(θ,φ)	(θ,φ)	Behavioral	Isomorphic	Coupled
Shadow cone selection [84]	Cone	(x,y,z)	(θ,φ)	(θ,φ)	Manual	Isomorphic	Coupled
Probabilistic pointing [80]	Cone	(x,y,z)	(θ,φ)	(θ,φ)	Heuristic	Isomorphic	Coupled
Enhanced cone selection [83]	Cone	(x,y,z)	(θ,φ)	(θ,φ)	Behavioral	Isomorphic	Coupled
Aperture [36]	Adjustable cone	(x_e,y_e,z_e)	(x,y,z)	(x,y,z)[b]	Heuristic	Isomorphic	Offset
iSith [71]	Two rays	$(x,y,z)(x_n,y_n,z_n)$	$(\theta,\varphi)(\theta_n,\varphi_n)$	$(x,y,z,\theta,\varphi)(x_n,y_n,z_n,\theta_n,\varphi_n)$	Manual	Isomorphic	Coupled
Flexible pointing [69]	Curved ray	(x,y,z)	$(\theta,\varphi)(x_n,y_n,z_n,\theta_n,\varphi_n)$	$(x,y,z,\theta,\varphi)(x_n,y_n,z_n,\theta_n,\varphi_n)$	N/A	Isomorphic	Coupled
Depth ray [41,93]	Ray and 3D cursor	(x,y,z)	(θ,φ)	(z,θ,φ)	Manual	Isomorphic	Coupled
Lock ray [41]	Ray and 3D cursor	(x,y,z)	(θ,φ)	(z,θ,φ)	Manual	Isomorphic	Coupled
Flow ray [41]	Ray	(x,y,z)	(θ,φ)	(θ,φ)	Manual	Isomorphic	Coupled
Friction surfaces [1]	Ray	(x,y,z)	(θ,φ)	(θ,φ)	N/A	Anisomorphic	CD ratio
PRISM [38]	Ray	(x,y,z)	(θ,φ)	(θ,φ)	N/A	Anisomorphic	CD ratio
Adaptative pointing [48]	Ray	(x,y,z)	(θ,φ)	(θ,φ)	N/A	Anisomorphic	CD ratio
SQUAD [49]	Ray and adjustable sphere	(x,y,z)	(θ,φ)	(θ,φ)	Manual	Isomorphic	Coupled

[a] The orientation of the selection ray is determined by a vector orthogonal to the screen plane.
[b] The third DoF is used to adjust the apex angle of the selection cone.

following relationship:

$$T = a + b\,\log_2\left(\frac{A+W}{W}\right)$$

where a and b are regression coefficients and the logarithmic term is called index of difficulty (ID). The intercept a is sensitive to additive factors such as reaction times (e.g. time to locate the target or time to trigger the selection confirmation) and the inverse of the slope $1/b$ is the index of performance (or the throughput) of the task.

The application of Fitts' law ranges from estimating the time required to perform an assembly operation through the evaluation of different input devices [59], up to estimating times for pressing a button with a mouse or selecting an object in 3D space [40]. Several works have extended the Fitts' Law formulation to higher dimensional tasks [66] and to account for noise [47] and latency [97].

Wingrave and Bowman [101] showed that Fitts' law still holds when pointing in virtual environments. Instead of considering the size of the target, they observed that W was related to the visual size of the target and A to the amplitude of the movement computed considering the angle covered by hand rotations. Poupyrev et al. [77] went further, by defining the size of an object W according to the vertical φ and horizontal ϕ angles an object occupies in the user's field of view. Further studies analyzing whether 3D object selection techniques are modeled by Fitts' law can be found in [6,50,89].

2.2. Optimized initial impulse model

Fitts' law only accounts for movement time according to the target's characteristics and the empirical parameters a and b. However, it does not provide any insight on how subjects perform acquisition tasks. Different human performance models have been proposed to explain the logarithmic speed-accuracy trade-off defined by Fitts' law.

The human movement model which better accounts for Fitts' Law is the *optimized initial impulse model* proposed by Meyer et al.

[62]. According to this model, acquisition tasks are subdivided in two phases. In the first phase, called ballistic phase, a fast and inaccurate movement is made towards the target. If the target is not acquired, during the corrective phase, iterative slow correction movements are executed in close loop feedback until the target is acquired.

Ballistic movements are intended to cover the whole distance towards the target, but due to limitations of the human motor system, the endpoint of the movement is randomly distributed over the desired endpoint [81]. This variability depends on the muscle groups involved in the movement [23], with bigger muscle groups introducing higher variability than smaller ones. On the other hand, *corrective movements* are slow movements where precision is the main requirement. They are needed when the target is undershot or overshot.

In their experiments Meyer et al. [62] defined the speed-accuracy ratio for ballistic movements. They stated that the standard deviation of the movement's endpoint is proportional to the speed average (D/T):

$$S = k\frac{D}{T} \tag{1}$$

where S is the standard deviation of the endpoint, D is the distance covered and T is the movement time. This relationship determines the trade-off between the speed of the movement and the precision needed. Faster movements result in higher endpoint variability, thus requiring more corrective movements to acquire the target. On the other hand, slow movements result in smaller endpoint variability and thus require fewer corrective movements.

Experimental observations show that, given a task, users minimize movement times by balancing the speed of the ballistic movement with the required corrective sub-movements [92]. The speed profiles clearly showed two distinct movement phases (see Fig. 2), with fast movements followed by sequences of slower movements. MacKenzie et al. [57] already concluded that velocity profiles also depend on W and A and not only on the ID. During the acceleration phase, A determines the maximum movement speed, regardless of the target size. In contrast, W determines the deceleration phase and the corrective movements required to acquire the target.

3. Classification of selection techniques

A number of taxonomies have been proposed to classify existing 3D selection techniques. In Bowman et al. [18] classification, interaction techniques are decomposed into subtasks and classified according to them (see Fig. 3). Following [18], a selection technique has to provide means to indicate an object (*object indication*), a mechanism to confirm its selection (*confirmation of selection*) and visual, haptic or audio feedback to guide the user during the selection task (*feedback*). One limitation of this classification is that the choice of a suitable feedback is often highly coupled with the object indication subtask, which introduces some redundancy into the classification. For example, raycasting-based techniques typically draw virtual rays to assist the user during the task. In addition, the classification proposed by Bowman et al. does not consider the varying purpose of the feedback during the indication and confirmation subtasks. While feedback guides user's actions during indication tasks, it has to show if the selection was successful in confirmation tasks.

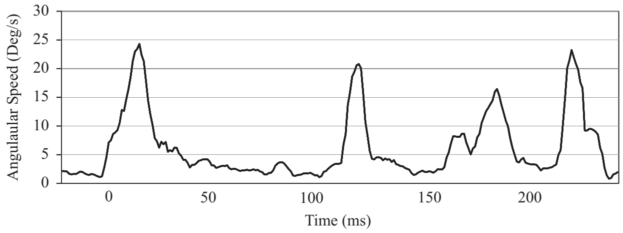

Fig. 2. Example of a velocity profile for a 3D acquisition task using raycasting selection. Ballistic and corrective phases of the movement are clearly visible.

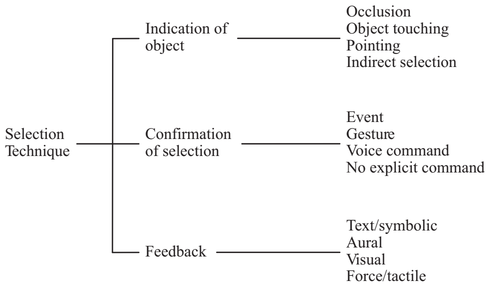

Fig. 3. Classification of selection techniques by task decomposition proposed by Bowman et al. [18].

Poupyrev et al. [76] proposed an alternative classification based on interaction metaphors (see Fig. 4). The classification has several levels. The first level distinguishes exocentric and egocentric techniques, depending on whether the user interacts from outside (third-person view) or inside (first-person view) the environment. In the second level, egocentric meta-phors are further subdivided into virtual hand and virtual pointer metaphors, and exocentric metaphors are subdivided into world-in-miniature and automatic scaling. In contrast to Bowman's classification, Poupyrev's classification disregards technique differences like feedback and confirmation mechanisms. The classification is not exclusive as exocentric and egocentric metaphors can be combined.

The above taxonomies provide a broad view of selection techniques but consider a relatively small number of design variables. In this survey, instead, we propose the classification of the selection techniques according to their intrinsic characteristics, considering the underlying selection tool and how the user controls it. Our classification provides a more complete characterization of existing 3D selection techniques to enable interaction designers to choose the best selection technique for a given task [4]. We also analyze each selection technique according to major human control models. The criteria we used for our classification (Table 1) are described below.

3.1. Selection tool

Object selection techniques require an underlying selection tool in order to perform an intersection test or a proximity test against the virtual environment for determining the selected object. The underlying selection tool is typically a 1D/3D shape, the most common ones being rays, cones, cubes and spheres. Simple shapes accelerate the intersection test with the 3D environment decreasing the overhead of the selection process, although in practice this overhead can be neglected unless continuous highlighting of the indicated object is desired, thus forcing the intersection/proximity test to be carried out every frame.

The shape of the selection tool is a key issue as it will determine its control (e.g. degrees of freedom), spatial range and accuracy. Referring to Poupyrev's classification, virtual hand and virtual pointing techniques are completely determined by the selection tool. Virtual hand techniques use 3D cursors (the underlying selection tool is, e.g. a sphere, cube or hand avatar), while virtual pointing techniques employ virtual rays (the underlying selection tool is typically a ray or a cone).

For the majority of techniques, the selection tool does not change its shape during the interaction process, although a few techniques provide mechanisms to alter the shape of the selection tool. For example, the Bubble Cursor [93] employs a sphere-like selection tool that automatically expands to reach the object closest to its center. Another example is Aperture Selection [36] which uses a cone as a selection tool and allows users to manually adjust its apex angle.

Although most techniques employ either 3D cursors or rays, alternative solutions do exist. The Depth Ray and the Lock Ray [41] adopt a hybrid approach combining a ray with a 3D cursor constrained along the ray. When the selection trigger is activated, the object intersected by the ray and closest to the 3D cursor is selected. Another example is the iSith technique [71] which employs two selection rays. A 3D point is computed from the two rays which is then used to perform a proximity test against the scene. Finally, the Flexible Pointing [69] allows the user to bend the selection ray to select partially occluded objects. A complete classification of existing selection techniques considering their selection tool is presented in the second column of Table 1.

3.2. Tool control

Once the shape of the selection tool is fixed, the selection technique also determines how the user is able to control it. The simplest way to control a 3D cursor is through a tracked hand (as in the Virtual Hand). In contrast, virtual rays can be controlled in a variety of ways, e.g. using the hand position and wrist orientation as in classic raycasting, or casting the ray from the user's viewpoint and going through the user's hand, as in Occlusion Selection [73] (see Fig. 5). Other techniques determine the ray's orientation by the vector defined by both hands [64], bending the ray according to the position of the non-dominant hand [69], or through eye tracking [87].

Virtual pointing techniques whose tool origin is located at the hand position, suffer from the eye–hand visibility mismatch. As analyzed by Argelaguet et al. [6], the mismatch between the eye position E and the hand position H introduces two potential conflicts. First, the solid angle subtended by potential targets with respect E and H might differ. Second, due to inter-object occlusion, some objects can appear occluded from the hand but not from the eye and vice versa (see Fig. 6). These conflicts will bias how the user perceives the difficulty of the task; some objects might seem easy to select but in practice they are not. In order to avoid this conflict, Argelaguet et al. [6] proposed a new virtual ray control approach for raycasting selection named raycasting from the eye. In their approach, the virtual ray origin matches the eye position but its orientation is controlled through wrist rotations (see Fig. 5c). Since the ray is cast from the eye, the set of visible objects and the set of selectable objects match. This approach showed a significant improvement on selection performance over rayscasting in cluttered environments.

3.2.1. Selection tool DoFs

The control mechanism determines the degrees of freedom required to control the selection tool. A 3D cursor controlled by the hand position involves three DoFs (one for each dimension), while a virtual ray controlled by hand position and orientation involves five DoFs, three to determine the ray's origin and two for the ray's orientation. The number of DoFs the user has to control is a measure of the complexity of the selection technique. The higher the DoFs, the more complex the tool control is but the higher its expressiveness. However, some DoFs are more relevant than others. For example, in the absence of visibility mismatch, any scene object can be pointed to by adjusting only the ray orientation, keeping the ray origin fixed. Therefore, although a virtual ray has up to five DoFs, in most situations the two orientation DoFs suffice to indicate any object.

As previously stated, the number of effective DoF depends on the technique. The Smart Ray [41], which was conceived to select semitransparent objects in cluttered environments, requires the

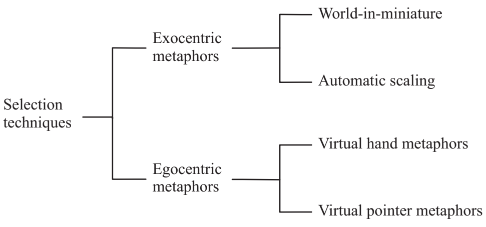

Fig. 4. Classification of selection techniques by interaction metaphor proposed by Poupyrev et al. [76].

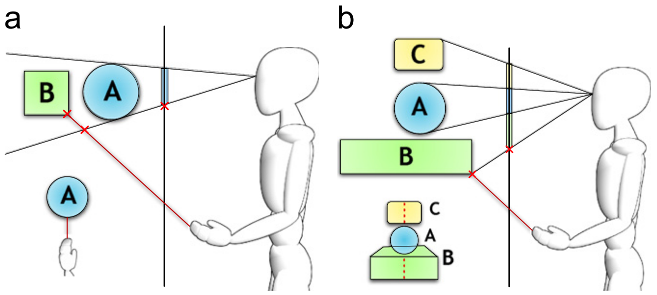

Fig. 6. Eye–hand visibility mismatch conflicts [6]. (a) The user can select an object which is hidden by another object. The last visible point on the ray is projected over the screen projection of the occluding object, leading to misinterpretation. (b) The visible object A cannot be selected because it cannot be reached by a ray emanating from the user's hand. The dotted line shows the path followed by the ray-scene intersection as it might be seen on the screen as the user rotates his hands upwards; the path skips object A.

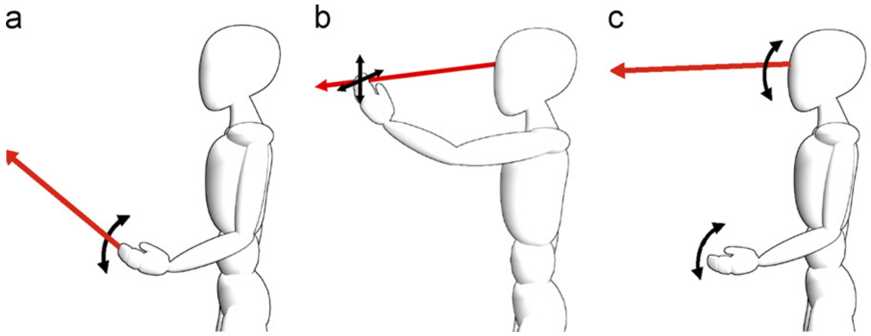

Fig. 5. Examples of different virtual pointing techniques: (a) raycasting, (b) occlusion selection and (c) raycasting from the eye.

user to point to the desired target from several directions, thus using all five DoFs. The Depth Ray and the Lock Ray [41] require the user to provide an additional DoF, as the selection tool is a 3D cursor constrained along a ray. Two-handed techniques also increase the number of DoFs the user has to control. In Two-Handed pointing [64] the virtual ray is constrained by the position of both hands resulting in six positional DoFs. A variation of two-handed pointing is the Flexible Pointing [69], which employs a Bézier curve as a selection tool. Two control points are determined by the position of the user hands (six DoFs) and the remaining control point is computed considering the orientation of both hands (four DoFs). Another two-handed technique is the iSith [71], where the user controls two virtual rays, requiring up to 10 DoFs. In contrast to classic raycasting, the origin of the rays plays an important role, as the selected object is determined by the intersection of both rays. In Table 1, columns 3–5 show the maximum DoFs and the dominant DoFs in major selection techniques.

3.2.2. *Control–display ratio*

The control–display ratio (CD ratio) determines how translations and rotations of the input device (x) are transferred to the selection tool (X). More precisely, the CD ratio is defined as $\Delta x/\Delta X$. Systems using an isomorphic mapping between the pointing device and the display have a unit CD ratio, which means than the movement of the pointing device is the same as the movement of the selection tool in the virtual environment. For example, moving the mouse 1 cm causes the cursor to move 1 cm too. When the CD ratio differs from one, the movement is scaled (CD ratio <1) or downscaled (CD ratio >1). The effect of a constant CD ratio on performance has been extensively explored in the literature but results are still inconclusive [10]. According to Fitts' law, a constant CD ratio affects the amplitude of the movement and the target's size at the same level, keeping the index of difficulty unchanged.

The first 3D selection technique proposing a dynamic CD ratio was the go-go technique [75] by Poupyrev et al. Go–go uses a virtual hand metaphor that adjusts the CD ratio according to the distance between the user's hand and its torso, increasing the limited area of reach of classic virtual hand techniques. When the distance is smaller than a given threshold, the CD ratio is set to one. Above the threshold, user movements are mapped non-linearly to the virtual hand. A factor k, where $0<k<1$, is used to adjust the non-linear component. Go-go allows the user to stretch its virtual arm to select distant objects, but the precision decreases as users move their hand further because movements are magnified. Although the CD ratio depends on the distance between the user's hand and torso, it may result in unnatural and unpredictable movements. Some studies show that people tend to judge their hand movement mainly on the basis of the on-screen movement of the cursor and adapt their hand movement accordingly [48].

Following König et al. [48], CD ratio-based techniques can be classified into three groups: manual switching, target oriented and velocity oriented techniques.

Manual switching techniques provide mechanisms allowing the user to manually control the CD ratio. The most common approach is based on reducing the CD ratio when additional accuracy is required. For example, Vogel et al. [95] proposed the use of gestures to allow the user to switch between isomorphic raycasting and anisomorphinc raycasting with a CD ratio greater than one. Although they obtained higher selection times with their approach (mainly due to mode switches), they get lower error rates than standard raycasting.

The second group, *target oriented* techniques, is based on reducing the CD ratio when the selection tool enters or approaches an object, following a sticky metaphor. Although this approach is useful for the selection of isolated targets, its performance tends to degrade in cluttered environments. In the context of interaction with 2D GUIs embedded into a virtual environment, Argelaguet et al. [1] proposed an anisomorphic raycasting approach to automatically modulate the CD ratio according to the width and height of the GUI window, obtaining lower error rates and increased user comfort.

Finally, *velocity oriented* techniques dynamically adjust the CD ratio according to the input device speed. Considering the optimized initial impulse model [62], accuracy can be decreased during ballistic phases (through a CD ratio lower than one) and increased during corrective movements (through a CD ratio higher than one). As a result, during ballistic movements the amplitude of the movement A decreases, while during corrective movements the size of the target W increases. This approach has been widely adopted for 2D mouse interaction and is often referred to as mouse acceleration. The PRISM technique proposed by Frees and Kessler [37] applies a velocity-oriented CD ratio for manipulation and selection tasks in 3D space. Fig. 7 shows how the CD ratio varies according to the speed of the input device. Movements below a minimum speed (MinS) are considered noise and thus ignored. Corrective movements (speeds between MinS and SC) are scaled down, increasing precision. For ballistic movements (speed higher than SC), they applied a 1:1 CD ratio. However, changes in the CD ratio introduce a spatial offset between the physical device and the virtual selection tool. After a sequence of corrective movements the position of the input device no longer matches the position of the virtual device. In order to solve this issue, when the speed exceeds a maximum speed (MaxS) the CD ratio is increased until the offset is recovered. König et al. [48] proposed the Adaptive Pointing which is based on a similar mapping (see Fig. 8), but it also takes into account the accumulated offset between the physical device and the virtual selection tool in order to avoid high discrepancies. Similar to PRISM, Adaptive Pointing in combination with raycasting resulted in reduced error rates and slightly better selection times.

3.3. *Motor and visual space relationship*

Two main interaction spaces are involved during the selection of a 3D object in a VE: the motor space and the visual space. The

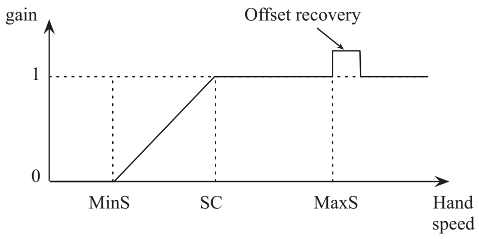

Fig. 7. Control–display ratio function for PRISM. Adapted from [38].

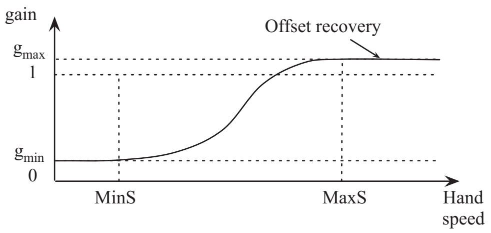

Fig. 8. Control–display ratio function for Adaptive Pointing [48]. The influence of the offset is not considered in this plot. Adapted from [48].

motor space (or working space) is the physical space available for the user to operate, which is constrained by the degrees of freedom available and the virtual reality setup. For example, a user controlling six DoFs hand-held device inside a CAVE is constrained by the six DoFs of the input device, the CAVE's walls and its own body limitations. In contrast, the *visual space* is the visual representation of the environment perceived by the user, which is independent from the selection technique employed and it is constrained by the field of view of the display.

Motor and visual spaces can be coupled, as in the Virtual Hand technique, or decoupled, as in a typical desktop setup where the movement of the mouse on a horizontal plane (mouse pad) is transformed into the movement of a 2D cursor on a vertical plane (screen). The motor space is thus mapped onto the visual space by a rotation and a translation.

When both spaces are coupled, absolute pointing can be accomplished relying solely on the proprioceptive feedback of the hand. However, when motor and visual spaces are decoupled, proprioceptive feedback no longer suffices and visual feedback is critical. The user has to continuously sample the selection tool location with gaze to execute accurate corrective movements [82].

In addition, the selection tool and its control determines which objects within the visual space may afford direct manipulation. In other words, these two components transform the motor space into another space which defines the scope of the user's actions. This transformed space will be referred to as the *control space*. The intersection between the control space and the visual space determines which objects afford direct manipulation.

For the classic Virtual Hand technique, the control space matches the motor space as depicted in Fig. 9a. Objects outside the motor space are not selectable. In contrast, for virtual pointing techniques, the control space matches the visual space, thus allowing the user to select objects outside the motor space. Changes in the CD ratio modify the relationship between the motor and the control space. A CD ratio lower than one scales the control space increasing the area affording direct manipulation at the expense of decreasing the accuracy and vice-versa. For example, the non-linear mapping of the CD ratio provided by the go-go technique [75] increases the control space (see Fig. 9b). Techniques using a dynamic CD ratio, such as PRISM [38] and Adaptive Pointing [48], reduce the control space when precision is required and vice-versa, but require an offset recovery mechanism to avoid an excessive decoupling between the motor and the control space.

The motor and the control space can also be decoupled by introducing a constant offset or allowing the user to determine the transformation between both spaces (clutching). When tracking the user's hand position, the virtual representation of the hand can be coupled with the position of the real hand or a constant offset can be added (translation and/or rotation). For example, in occlusion selection [73], the pointing direction is defined by roughly aligning the hand with the eye position, thus requiring the user to keep its arm extended. Introducing a vertical offset allows the user to keep his hand in a lower position, reducing fatigue levels (see Fig. 10). This offset can be beneficial also when using projection-based systems to keep the real hand from occluding the projected content. Another example is the Virtual Pad metaphor [2] which allows the user to decouple the working and the visual space when interacting with 2D graphical user interfaces embedded in 3D space. This decouple did not introduce any performance loss although a constant offset and rotation was introduced.

Clutching mechanisms allow the user to relocate the control space. It accounts for hand repositioning [44] at the expense of introducing an offset between the selection tool and the physical device. Relocating the control space allows the user to select objects otherwise unreachable (see Fig. 11) at the expense of increased user attention. A trigger is needed to enable and disable the clutching. This trigger can be explicit like pressing a button, or implicit like a 2D mouse where the clutching is achieved by lifting the mouse.

However, decoupling the motor and visual spaces may result in performance loss. Humans seem to achieve optimum manipulation performance when haptic and graphic displays of objects are superimposed [96], particularly during rotation tasks. However, moderate disparity in orientation between haptic and graphic displays appears to have no significant effect on object translation. This higher tolerance to object translation with

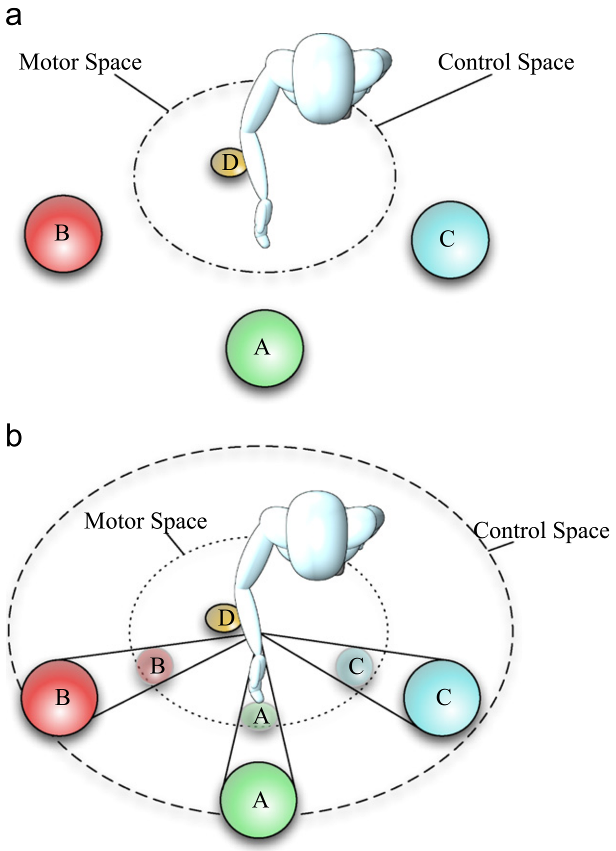

Fig. 9. Mapping between motor and control space for the Virtual Hand and the Go–Go techniques. In classic Virtual Hand selection (a), only objects inside the working space are selectable. (b) The Go–Go technique alters the mapping between the motor space and the control space, allowing the user to select objects outside the motor space.

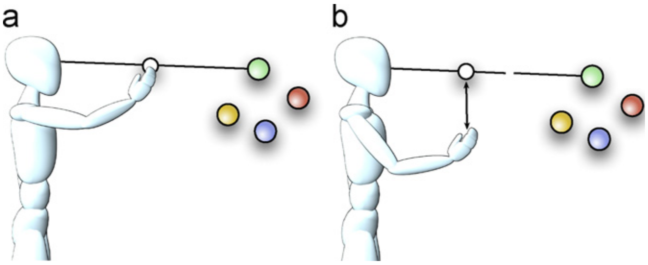

Fig. 10. Occlusion selection might require the user to keep its hand roughly aligned with the eye position (a). By introducing an offset between the motor and the visual space, the user can keep a more comfortable position (b).

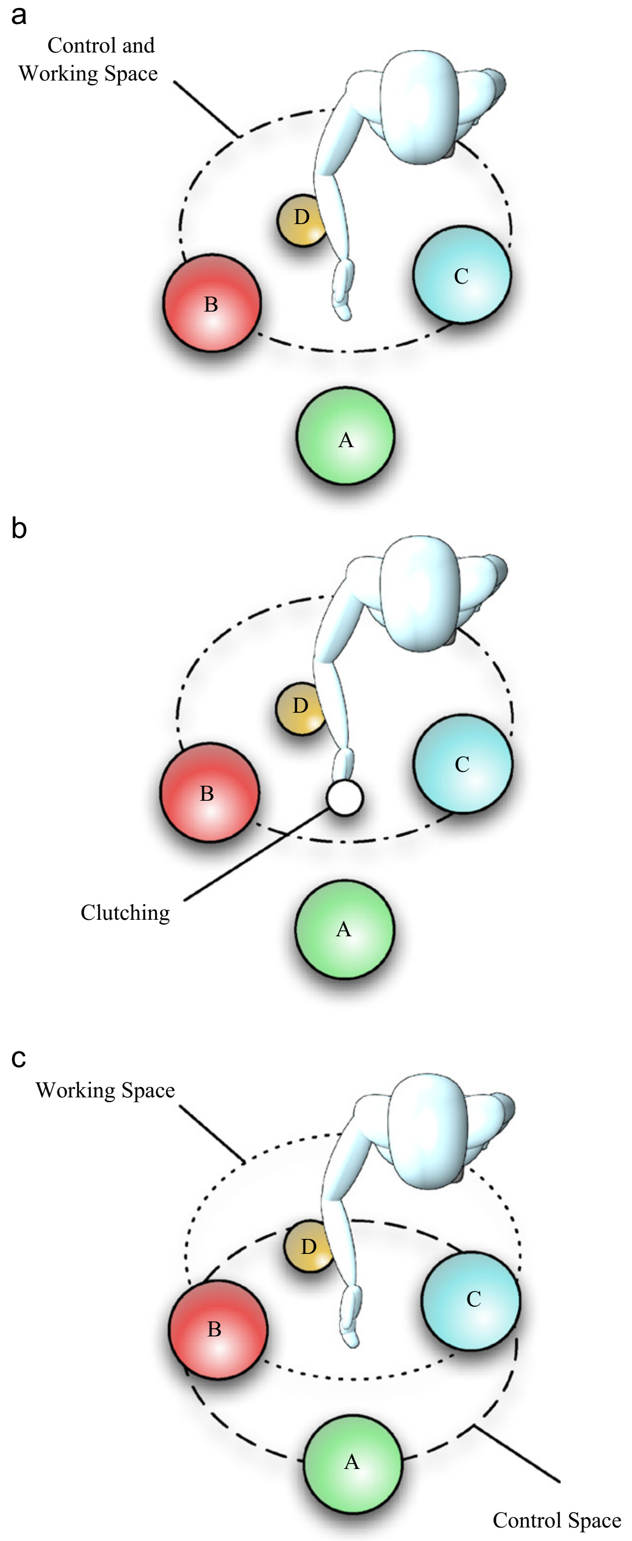

Fig. 11. A clutching mechanism allows the user to relocate the control space. The object A is not selectable as it is outside the control space (a). The user places the 3D cursor (b) and fixes its position with the clutching mechanism. The user returns its hand to the center of the motor space and disengages the clutching mechanism. Now the control space is centered at the previous 3D cursor position (c) and object A is now selectable.

respect to rotations explains why most virtual hand techniques can provide clutching mechanisms while virtual pointing techniques cannot.

3.4. Disambiguation mechanism

As previously discussed, a common strategy to improve selection performance relies on employing volumetric selection tools [54,84] such as cones and spheres. However, volumetric tools are prone to indicate more than one object at once, specially in dense environments [93]. In these situations, the selection technique has to provide some mechanism to disambiguate the selection. We classify disambiguation mechanisms into three groups: manual, heuristic and behavioral.

In *manual approaches*, the user has to decide, among the indicated targets, which target is the desired one. Manual approaches provide the maximum flexibility at the expense of increasing the cognitive load of the user due to additional selection steps. The simplest solution consists in using a button to cycle among all indicated objects [44], but this approach does not scale well if too many objects are indicated. Instead of cycling among the selected objects, we can display them and allow the user to perform additional selections steps. For example, in the Flow Ray proposed by Grossman et al. [41], objects intersected by the virtual ray are displayed in a list or a pie menu, thus letting the user select the desired one in a second selection step. Similarly, Kopper et al. [49] proposed a progressive refinement approach called SQUAD. By refining the selection through a QUAD-menu user interface, in which the indicated objects are split into four groups, the user performs simple selections until the selection only contains a single object. A different approach relies on increasing the amount of DoF, using the extra DoFs for disambiguating the selection. For example, the Depth Ray [41] provides a 3D cursor constrained along the virtual ray. The 3D cursor is controlled by pulling the hand forwards and backwards (additional DoF). This additional DoF allows the user to disambiguate the selection, as the object closest to the 3D cursor which has been already intersected by the selection ray will be the selected one.

The second group of disambiguation techniques employ *heuristics* to guess which object the user is willing to select. Objects are ranked according to a heuristic and the higher ranked object is selected. The easiest approach considers the distance between the objects and a central axis of the selection volume, as in flashlight selection [54], where the object closest to the axis of the selection cone is selected. Schmidt et al. [80] extended this naive approach by proposing probabilistic pointing-based selection algorithms. Heuristic methods have also been proposed in the context of disambiguating 3D locations in volume rendering applications [100].

Finally, *behavioral approaches* take into account user's actions prior to the selection confirmation. Instead of applying a heuristic when the user triggers the selection, they continuously rank objects during the selection process, gathering statistical information. IntenSelect [30], enhanced cone selection [83] and Sense Shapes [68] follow this approach. The data considered for ranking the objects includes the time the object is inside the selection volume, its distance to the center of the volume, the number of times the object enters the volume, the objects' visible pixels within the selection tool and the average or minimum pixel distance to the center of the volume's center. These approaches are particularly useful for selecting moving targets. If we track the moving target with the selection tool its selection weight will increase with respect to static objects.

In summary, manual approaches provide total control to users at the expense of increased cognitive load or additional selection steps. As stated by Kopper [49], there is a tradeoff between the usage of precise selection techniques and selection techniques requiring manual disambiguation. Precise selection techniques will perform better when selecting easy targets (e.g. big targets) or in high density environments, while manual disambiguation techniques will perform better in low density environments or when selecting potentially difficult targets (e.g. small targets). On the other hand, heuristic and behavioral techniques do not introduce any further selection steps, but as they are not completely accurate, they might result in unwanted selections and thus require the user to repeat the selection. In Table 1, column 6 shows the disambiguation mechanisms provided by each of the considered selection techniques.

3.5. Selection trigger

The final step of the selection task is the selection confirmation. Bowman et al. [18] consider four different confirmations alternatives: event, gesture, voice command and no explicit command. The most common option is to press a button conveniently placed in the pointing device, often called *press to select*. Steed in [83] considered two additional options: *hold and select* and *dwell on object*. For hold and select instead of triggering the selection when pressing the button, it is triggered when the button is released, which may be less sensitive to the Heisenberg effect. In contrast, for the dwell on object approach, the selection is triggered when the user points to an object during a fixed amount of time. Dwell time thresholds introduce a fixed constant latency, being sensitive to the Midas Touch effect: for high precision selections fixations may occur at objects that do not interest the user, resulting in unwanted selections. This is also the case when using eye-gazed Selection [87,27]. Müller [65] concludes that the dwell time must be in the range of 350–600 ms, although it might vary according to the requirements of the application.

Gestures can also be used as selection triggers. The simplest approach is to perform a pinch gesture [20]. However, if we are using the hand as a pointing device, the gestures used for the selection confirmation have to minimize hand instability to minimize the Heisenberg effect. Vogel and Balakrishnan [95] proposed two additional gestures focusing on the minimization of such side effects, namely *AirTap* and *Thumb Trigger*, in combination of visual and auditory feedback.

Finally, interfaces which combine direct manipulation and voice input employ voice commands as triggering mechanisms (‘*Select that*’), as in Bolt’s Put-that-there [13]. More complex voice commands can be used if the elements of the VE have semantic information [72] (‘*Select that bolt*’).

3.6. Feedback

Selection techniques involving spatial interaction require users to perform gestures to control the selection tool. The gesture can be a simple grasp operation or a pointing operation. If no feedback is provided, the user has to rely only in proprioception and depth perception to ensure that the gesture results in the selection of the intended virtual object. Although interaction with nearby objects can be achieved only by proprioception [64], several studies revealed that users without any selection feedback are unable to efficiently interact with the VE [102]. As pointed out by Wingrave et al., “*users do not have a model of interaction with the environment but a model of how to respond to the feedback the environment provides*” [102]. Therefore providing feedback is critical [39].

A selection technique has to provide, at least, visual feedback to drive user’s actions. The simplest option consists in displaying a virtual representation of the user’s actions in the environment, for example, drawing the user’s hand avatar or displaying the pointing direction. The visual feedback allows users to observe how their gestures map with the virtual tool. In situations where the CD ratio differs from one or when the shape of the selection tool changes over time, a good visual representation of the selection tool is a key feature to ensure usability. But in general, proper visual feedback highly depends on the interaction technique.

Moreover, after the selection confirmation, the selected target can be highlighted [63]. Changes on its visual properties allow the user to ensure that the object selected is the right one. For example, changing its color or displaying the object in wire frame. In contrast, continuous highlighting of the object indicated by the selection tool has to be used carefully. It might cause excessive popping and be a distracting factor, particularly while interacting with cluttered scenes. Furthermore, object highlighting requires to check every frame which is the object indicated by the selection tool, thus potentially increasing the application’s overhead. In general, increasing the amount of visual feedback does not always improve user performance [78] and might even reduce selection performance [101,41]. On the other hand, providing redundant information might allow to bypass aptitude and expertise, allowing unexperienced users to perform as experienced ones. Selectable areas should be indicated to avoid confusion [39], as selectability may change dynamically over time [83]. If selectable areas can be outside the viewing frustum, users can be provided with markers to guide them towards the desired object [106].

In addition to visual feedback, introducing different feedback modalities, like haptic and acoustic feedback, can also be beneficial [43]. Although it is not assured that including additional feedback results in performance improvements [21], users often prefer the addition of extra feedback [94]. Active haptic feedback can assist users during the selection process [70,94].

However, it requires a fine tuning of the forces applied and in dense environments it might be counterproductive as the user might be guided to the wrong object. An easier approach is to provide passive haptic feedback (physical constraints), which can further increase interaction precision [43]. The most adopted solutions rely on using prop-based physical constraints [45] or physical surfaces [17,56]. Both provide spatial references, which are intuitive to learn and speed up 2D pointing tasks in free space. The user’s body can be also used as a frame of reference, as the user is able to determine its own body position by proprioception. One clear example is the Go–Go technique [75]; the user is aware of the distance between its body and its hand. Using the non-dominant hand [44] can also be considered, as it provides a frame of reference for the dominant hand, and the user can employ it to perform two tasks in parallel.

Lastly, auditory feedback [94] can reinforce user’s actions. For example, it can inform the user when a target has been highlighted or successfully selected. However, similar to haptic feedback, when interacting on dense environments it might produce distracting effects and playing the same sound multiple times might become annoying.

4. Factors influencing performance

A number of usability guidelines exist for 2D user interfaces, however, in general, they are not directly applicable to 3D user interfaces. 3D user interfaces are significantly more difficult to design, implement and use than their 2D counterparts. 3DUIs are based on real-world characteristics such as naive physics, body awareness, environmental awareness, social awareness and social skills [46].

There are a few works explicitly focusing on usability guidelines for 3D user interfaces, being the work of Gabbard [39], Hal [44] and Bowman [18,19] notable exceptions. Usability guidelines are useful during the first stages of the design as they avoid known usability issues and speed up the whole design process. Usability issues might arise due to intrinsic factors, mainly determined by the nature of the selection task, and due to extrinsic factors introduced by input and output devices. From a usability point of view, a selection technique has to (1) provide rapid selection, (2) be accurate and error-proof, (3) be easy to understand and control and (4) produce low levels of fatigue.

Additional requirements depend on the application, e.g. support sparse or dense environments, provide mechanisms to select semi-transparent or occluded objects, and do not interfere with the user's immersion in the virtual environment. In the rest of this section we review major factors influencing usability and performance, and provide some guidelines for adopting or extending pointing selection techniques.

4.1. *Target geometry*

Object's size and location have a direct effect on selection performance. Following Fitts' law and the Optimized Initial Impulse Model, several guidelines can be proposed to increase user's performance in selection tasks. As proposed by Balakrishnan [10], options rely on decreasing the distance to the target, increasing the size of the target, or modifying both at the same time. Selection time increases if the amplitude of the movement (*A*) increases and/or the object size (*W*) decreases, and vice-versa, which has been corroborated by different studies [93,70,12,78].

Regarding target distance, a first approach for reducing it focuses on modifying the layout of selectable items. Items can be laid out in a way that the distance between them is minimized [28]. However, this approach is limited to graphical user interfaces in which potential targets are known a priori. In contrast, a more general approach is to reduce *A* only in control space, preserving the original layout of the elements in the visual space. For example, it can be accomplished by ignoring the empty space between targets (see Fig. 12a). However, objects' boundaries in the control space become closer. Without the proper feedback, as the new boundaries of objects are not visible to users, target boundaries might become unclear.

Techniques attempting to increase *W* have focused on increasing the area of influence of the selection tool, increasing the activation area of targets in control space or dynamically increasing the size of the targets. The area of influence can be increased using volumetric tools as in Flashlight [54] or Aperture Selection [36]. This approach allows for fast selections in sparse environments (see Fig. 12b), but in cluttered environments their performance tends to degrade as disambiguation mechanisms are required. Increasing the activation area of a target can also be done only in control space considering the objects close to the selection tool. For example, the Bubble Cursor [93] subdivides the control space into Voronoy cells according to the layout of the objects in the environment. Instead of selecting the object by placing the selection tool over it, the object is selected if the selection tool is inside the Voronoy cell enclosing the object (see Fig. 12c). In other words, the object selected is the object closest to the 3D cursor. Another example is the Sticky Ray [85], which selects the object closest to the selection ray. However, improvements on selection performance depend on the density of the VE, again improvements are more apparent in sparse environments. Furthermore, as the visual representation is kept unmodified, additional feedback is required to show changes in the control space.

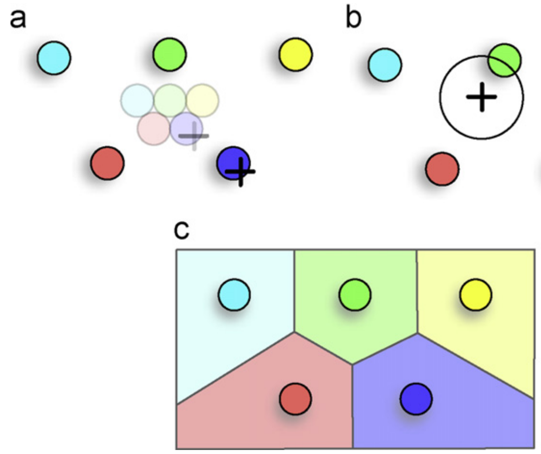

Fig. 12. Three approaches to improve the acquisition of small targets without changing their visual representation: (a) reduce the distance between targets only in control space, (b) increase the size of the selection tool, and (c) increase the area of influence of each target.

Moreover, *W* can also be increased both in control and visual space, known as Expanding Targets. This approach relies on dynamically increasing the size of targets near the selection tool (see Fig. 13). Expanding Targets has his origins in the Graphical Fisheyes Views [79] in which the displayed visual elements are rearranged in order to increase the visibility of the object of interest. Although it was not originally designed as a selection aid, it has been applied for 2D graphical user interfaces [11]. Regarding Fisheye Menus, we have to note that typically the size of targets is only increased in visual space, the more famous example is the Apple's dock menu. However, if the size is also increased in motor space, considering that the time to acquire isolated targets depends mostly on the final target size and not on the initial one [10], by increasing the size of targets, users are provided with a larger target area to interact with. Several studies support the use of Expanding Targets for 2D acquisition tasks [26], but only one work explored its viability in 3D object selection [5]. The evaluation presented in [5] showed that 3D Expanding Target techniques are also viable but only for simple environments.

Although theoretically increasing the size of targets or its activation area will result in a decrease on the index of difficulty of the task, a general drawback of increasing *W*, as reported by Wingrave et al. [102], is that it will induce users to decrease their accuracy as they no longer need to be accurate. The decrease of the index of difficulty could be compensated by a decrease on the index of performance (1/*b*) resulting in similar selection times. Approaches increasing *W* should be considered in scenarios with high error rates, as increasing *W* effectively decreases error rates.

Finally, techniques that increase *W* and reduce *A* at the same time focus on CD ratio adjustments (already discussed in Section 3.3). According the optimized initial impulse model, the ballistic movement will cover most of the distance *A* towards the target, while the size of the target *W* will influence corrective movements. In summary, *A* can be reduced by decreasing the CD ratio during ballistic movements and *W* can be increased by increasing the CD ratio during corrective movements. However, techniques exploiting this concept only adjust the CD ratio during corrective

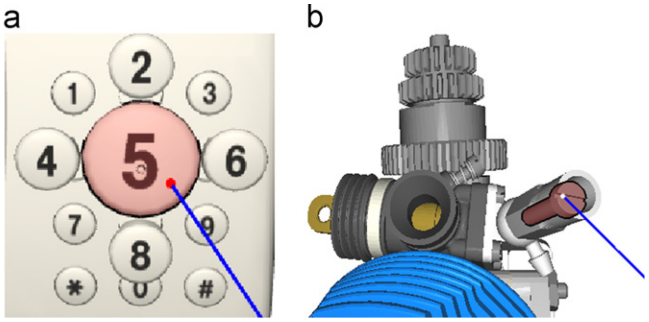

Fig. 13. Two 3D expanding targets techniques [5]: (a) scale potential targets and (b) show the target completely unoccluded.

movements to avoid excessive decoupling between the pointing tool and its visual representation [38,48].

4.2. Object distance and area of reach

As stated in the previous section, the control space determines which objects can be selected. Virtual hand techniques allow the user to select only the objects inside the working space unless decoupling mechanisms are employed. Although clutching or CD ratio based techniques can be used to extend the control space, clutching mechanisms introduce additional cognitive overhead and CD ratio based techniques, like the Go–Go [75], cannot provide enough precision when selecting distant objects.

In contrast, the control space for virtual pointing techniques matches the visual space, thus all objects in the visual space are selectable. However, as the selection tool is mainly governed by hand's rotations, its precision is limited to the user's hand angular accuracy and stability. The further away an object is the higher the accuracy is required. Although the theoretical control space matches the visual space, the precision slightly decreases as the distance increases. Nevertheless its precision is higher than that provided by virtual hand techniques.

Depth perception becomes an additional limiting factor when selecting distant objects. The level of required depth perception varies from one selection technique to another. For example, virtual hand metaphors require higher depth perception as the hand's depth is used to control the virtual tool. In contrast it is less important for virtual pointing techniques and even less for image plane techniques.

At this point, it can argued that selection techniques oriented towards the selection of small or distant objects are superfluous, as navigating towards the target to obtain an easier selection appears to be a logical alternative. Two main issues arise in navigate-to-select approaches. First, navigating to obtain an easier selection can be also "potentially" difficult. Navigation in cluttered environments requires proper navigation techniques and the selection of small objects will require to scale the VE. In addition, the navigation in dynamic environments is even more challenging as the target might move outside the user's field of view. In this situations, progressive refinement techniques [49,24] or techniques which take into account moving objects [30] are better suited. Furthermore, the user has to be provided with mechanisms to easily switch between selection and navigation tasks.

Second, when navigating in homogeneous VEs, such as a molecular model, the user can lose crucial context information. For example, while navigating towards the target, the user might lose track of it. If this happens, the user has to go back to the starting position, locate again the target and restart the navigation task. Although for some situations the navigate-to-select approach might be desirable, for selection-intensive scenarios the ability to select small and distant objects is necessary.

4.3. Object density

Until now, we considered selection tasks when the objects are isolated, but in a standard scenario objects might be surrounded by other objects. As the object density increases, occlusions between objects are likely to increase. Occlusion is present in reality and provides important depth cues for spatial perception. However, occlusion is not always a desired feature. It can have a detrimental impact on tasks requiring to locate an object (discovery), obtain information encoded in the object (access) or obtain spatial information of the object and its context (spatial relationship) [33]. Occlusion is a common issue for cluttered environments, high object density leads to occluded objects from the user viewpoint, reducing user's selection performance [93]. To avoid occlusion, in controlled situations, the environment can be rearranged [83]. However, in most situations it is not possible as the environment is fixed and context information should be preserved. Occlusion might increase the time required to discover an object in the virtual environment and in the worst case scenario the object will be fully occluded requiring the user to navigate in order to locate it. Furthermore, although the user sees the target in the environment, occlusion still results in reduced object visual sizes and restricted access to targets [77] which will affect user performance [83,93]. In these situations the user has two main choices, navigate to find an unoccluded view of the target or perform the selection from the occluded viewpoint. Improvements can be focused on the discovery phase or on the access phase, although it remains unclear whether improvements in the discovery phase will also improve access phase.

Standard interaction techniques based on virtual constraints like damping, snapping or trolling, which are useful in sparse scenarios [44,39], will be difficult to control in cluttered environments. Users tend to complain mainly about flickering effects [102]. CD ratio based techniques better adapt to dense environments, although the overall index of performance depends on the level of occlusion.

A different solution is to employ occlusion management techniques. Elmqvist and Tsigas [33] analyzed a broad range of techniques for occlusion management and identify five main design patterns: Multiple Viewports (using two or more separate views of the scene), Virtual X-Ray (turn occluded objects visible), Tour Planners (a precomputed camera animation reveals the otherwise occluded geometry), Volumetric Probes (user controls an object which alters the environment in its neighborhood) and Projection Distorters (nonlinear projections integrate two or more views into a single view).

Despite having so many options to deal with occlusion, when considering direct interaction in VEs the alternatives are limited and there is no single solution that completely solves occlusion issues. Projection distorters [32] do not integrate well in immersive environments as we can hardly modify the user perspective. Tour planners involves navigation and the user has to stick to the predefined navigation paths, lacking flexibility. On the other hand, virtual x-ray [22] and volumetric probes [31] allow users to manually remove occluders in order to get an unoccluded view of the intended target (see Fig. 14). However, these alternatives increase the cognitive load of the user, potentially increasing selection time. Moreover removing occluders may remove useful context information. The most common solution is to employ semi-transparency [93]. However, spatial relationships between semi-transparent objects may become unclear to users and access tasks can be compromised. An alternative solution is the World-in-Miniature metaphor [86], which provides the user with an additional viewport displaying the virtual environment from a third-person perspective. In order to afford for direct manipulation, the content displayed in the WIM can be manually adjusted to contain only a portion of the VE [103] or automatically adjust the content removing potential occluders [3] (see Fig. 15).

4.4. Input and output devices

The staggering number of devices available for use in VEs makes the development of 3DUIs significantly harder than their 2D counterparts [105,19]. Input and output devices affect the usability of existing interaction techniques [14]. Typically, interaction techniques are designed and evaluated taking into account only one hardware configuration, due to time, availability and budget limitations. Furthermore, it does not exist the best hardware solution nor a standard VR hardware platform [104]. For example, wireless devices should be preferred over wired ones but wireless devices are more expensive and the battery life

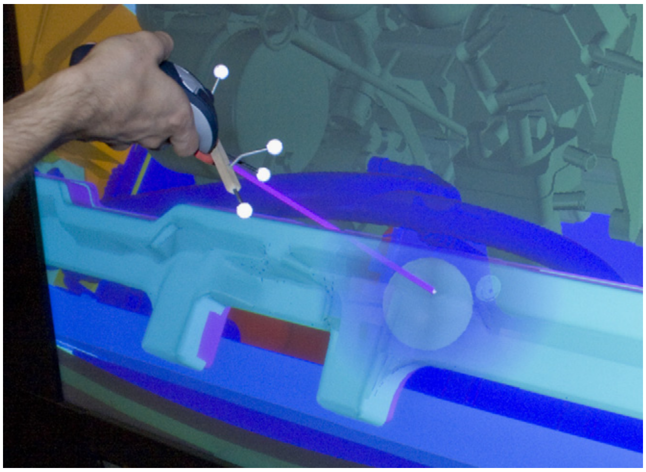

Fig. 14. In cluttered environments, objects might be occluded from the user's viewpoint. Occlusion management techniques such as virtual X-Ray can be employed to improve their selection. Image from [9].

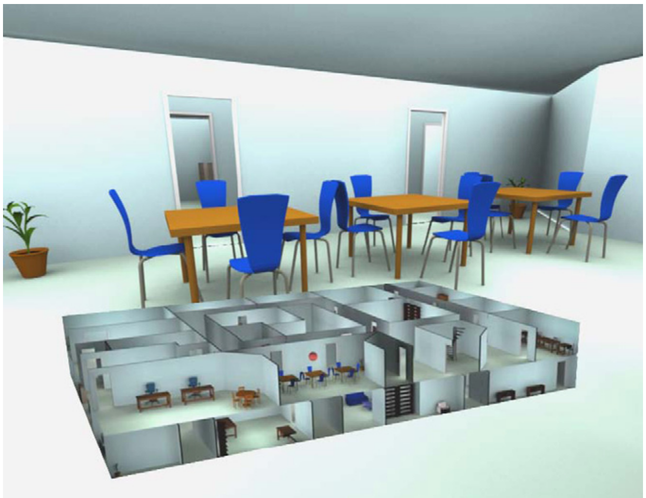

Fig. 15. WIM enhanced with cut-aways [3]. The miniature replica provides a cut-away view of a part of the model according to the viewing direction and the user's hand position inside the WIM (shown as a red sphere). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

might be an issue. At the end, this might result in the development of techniques that are only usable for a specific setup, being the comparison with other existing techniques unfair.

4.4.1. Input device DoFs

When designing a new interaction technique, it is important to consider the matching between the DoFs required for the interaction technique and the DoFs provided by the input device [44]. It is recommended to minimize the number of DoFs required, as the more DoFs used the harder is the control of the selection tool [19]. Virtual Hand techniques only require three DoFs for the hand position, and raycasting techniques are mainly governed by the yaw and pitch of the hand (only two DoFs). Employing six DoFs for tasks requiring less could be confusing if the input device is not well constrained [43], as changes in the unused DoFs are not visible to the user. Wingrave et al. [106] observed that users performing with raycasting tend to move their arms forward and backward to select objects placed at different depths. This is totally unnecessary as raycasting is almost insensitive to hand position, specially for selecting distant objects. This behavior is common for novice users which unknowingly hinder their ability and thus they have to be taught not to perform in that way.

On the other hand, if the input device is not able to provide the amount of DoFs required the interaction technique must provide additional mechanisms in order to control the DoFs independently [61].

4.4.2. Ergonomics

The physical device employed has to match the functionality of the interaction technique [44]. It makes no sense to employ a sphere-shaped device for virtual pointing, as the way of grasping the device should provide the pointing direction by proprioception. Furthermore, most of the existing input devices are equipped with a number of buttons. The mapping between the buttons and the functionalities of the interaction technique is crucial for its usability. For example, a button press in a hand held device (like a wand) introduces instability when the button is pressed.

Performance is also tightly coupled with the muscle groups involved [77,25], smaller muscle groups achieve higher motor precision than bigger ones [108,48]. For selection tasks requiring high accuracy input devices relying on smaller muscle groups should be employed .

4.4.3. Displays

Emerging and specialized devices, such as holographic [41] and tabletop displays [42], require specific interaction techniques as they present unique conditions in terms of working and visual areas. Available display devices range from semi-immersive displays, LCD screens and projection based systems, to fully immersive displays like head mounted displays and CAVEs systems. Each display has its own field of view and provides the user with different levels of immersion [90]. Head mounted displays (HMD), typically have greater field of regard (amount of physical space surrounding the user in which visual images are displayed). On the other hand, HMDs have reduced resolution in comparison with projection based systems.

The field of view determines the amount of information visible at a time, the more information displayed the easier is to locate an object without head movements. For example, when using raycasting selection in a fully immersive device, the selection ray is displayed entirely, allowing the user to easily determine the origin and the orientation of the ray. However, in a semi-immersive display, only a fraction of the ray will be inside the viewing frustum. Furthermore, the orientation of the display also plays an important role. Pointing gestures will differ from vertical displays (e.g. powerwall) and horizontal displays (e.g. tabletop).

Displays can be classified into non-obstructive and obstructive displays [19]. In non-obstructive displays, the user is able to see his own body. However, two conflicting situations may arise [42,91]. First, objects exhibiting positive parallax (objects behind the projection screen) might not afford direct interaction. For example, for virtual hand techniques objects cannot be "touched" as the working space is restricted by the projection screen. Second, objects exhibiting negative parallax (objects between the projection screen and the user) might induce depth sorting conflicts. The user's hands and arms might occlude objects that are virtually closer, presenting stereo fusion issues and vergence and accommodation conflicts. Similar issues arises when using 2D cursors to select 3D content [7], most 2D cursor based approaches present stereo fusion issues, the depth mismatch between the cursor and the target object prevents the user to fuse both objects (see Fig. 16). In addition, user's actions can obstruct the visualization of the virtual environment. For example, when selecting an object with the virtual hand technique, the user's hand will occlude the projection of the object during corrective movements,

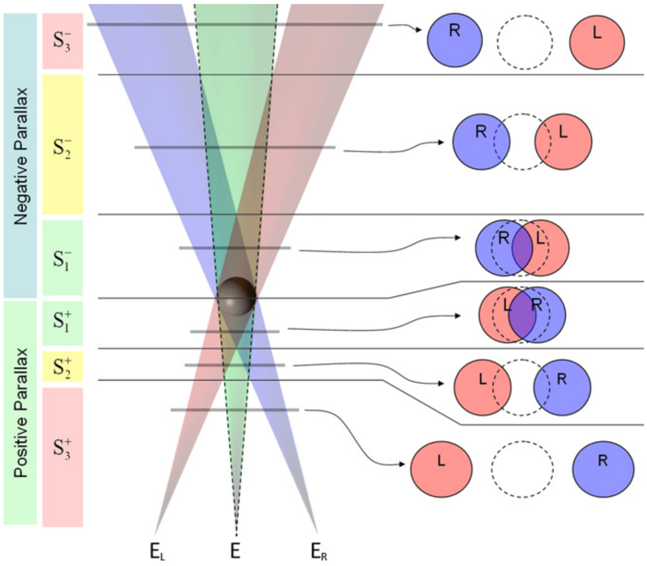

Fig. 16. Six different configurations for the left *L* and right *R* screen projections of an object and the selection area (dashed circle). The position of the projection screen where these situations occur is shown on the left. The virtual object is the sphere at the intersection of the three cones. Note that for most of the situations the overlap between the projections of the cursor and the object do not provide adequate feedback. Image from [7].

increasing the chance of erroneous selections specially for small objects.

On the other hand, obstructive displays (e.g. HMD) do not present the limitations of non-obstructive displays. However, if needed, the application has to provide a virtual representation of the user's body. If the user's body is not correctly tracked, proprioceptive information will conflict with the virtual avatar thus hindering interaction. Nevertheless, non-obstructive displays are more common than obstructive displays as they provide higher visual fidelity, are less sensitive to head rotations and do not require the user to wear heavy equipment.

4.5. User fatigue

One of the most known issues in virtual reality applications is fatigue. The reduction of the fatigue is especially important if a hand-held device is used, as fatigue levels can raise rapidly. Interacting with our own body can raise fatigue levels and extended use may induce simulation sickness [52] and muscular strain. Selection techniques are more prone to arm and wrist strain, while for example, navigation techniques are more prone to induce simulation sickness.

Arm and wrist effort can be extrapolated taking into account the degrees of freedom required to control the selection tool. Virtual hand techniques will require more arm effort than virtual pointing techniques, while virtual pointing techniques will require more wrist effort. Other device-ray mapping such as occlusion selection, which require to keep the arm roughly aligned with the eye, will require increased arm effort.

In the absence of input filtering (discussed below), hand fixation is required to reduce hand trembling and stabilize the selection tool, but it requires additional user effort. Hand fixation is tightly coupled with the precision required; the greater the impact of hand trembling the higher the hand fixation should be. Moreover, the position and orientation of the working space with respect to the user plays an important role in the user's comfort. For example, the user can accomplish manipulation tasks with the arm lowered in a relaxed position by defining a convenient working space [2]. Ideally, VE applications should allow users to define their working spaces according to the their own preferences, the physical condition of the user and the desired speed/accuracy balance.

4.6. Application performance, latency and noise

In order to ensure smooth interaction the VR application has to keep a high and constant frame rate [99], avoid large end-to-end latency [97,60] and filter data from noisy input signals [48,43,39]. If the application does not ensure these requirements, it might reduce interaction performance [88], hinder high precision tasks, and also break immersion and presence. Noisy tracking devices in combination with users' hand trembling [48] decrease the precision of the selection technique. Selection techniques have different tolerance levels to noise. Virtual hand metaphors are more tolerant to noise as they only rely on positional tracking data, but virtual pointing techniques are less tolerant to noise, as they mainly rely on rotational data. Some CD ratio based techniques behave as a noise filter for low-amplitude noise, and volumetric selection tools do not require to be accurate.

When high precision is required, a device with low latency and low noise should be used. If not possible, band-pass filters or Kalman filters can be applied to reduce noise of the input signal [48]. However, too much filtering increases the end-to-end latency. Pawar and Steed in [70] state that 60 ms is the maximum latency that can be introduced without degrading interaction. In situations with high latency, Wingrave et al. in [106] observed that users performed steady movements and relied on proprioception rather than on the visual feedback. Obviously these behaviors trade off speed and accuracy. Moreover, changes in latency with respect to time, referred to as temporal jitter, also hinder interaction and thus should be avoided. People can detect small fluctuations in latency likely as low as 16 ms [70].

4.7. User's preferences

Users interacting with VEs account for different preferences. Knowing these preferences allows interface designers to determine which are their preferred interaction techniques [106]. For example, computer game experience has been found to be an important factor both in terms of task performance and users' preferences [106]. Users have different levels of expertise and perform actions in different ways, thus requiring selection techniques suited for their skills.

As stated before, manipulating input devices in free space can easily raise fatigue [44]. We can provide users with recalibration mechanisms to allow them to define their working space, obtaining a more comfortable positions. In addition, according to the user's method of interaction the designer may personalize the behavior of the interaction technique to behave like the user wants. As Wingrave et al. show in [102], subtle versions of the same interaction technique can be provided, and we can let the user choose they preferred configuration. In their experiment, they employed raycasting and occlusion selection with a snapping mechanism (the selection tool bends to the closest object within a range). The user could introduce a rotational (raycasting) or a translational (occlusion selection) offset to the virtual hand with respect to the real hand, thus allowing for a more comfortable interaction, and change the threshold of the snapping mechanism. They results showed that there was not a trend when tunning the selection techniques, each user had his own preferences.

On the other hand, instead of letting the user customize his interaction, we can adapt the available techniques to better suit the user. Octavia et al. in [67] explored how to choose automatically the most suitable interaction technique for a certain situation.

They gathered physiological data to measure user frustration, user experience and the mental workload. They observed that frustration measures were strongly correlated to the task completion time. Users accepted the technique adaptation; they did not bother when the system automatically chose the best suitable technique if the performance was slightly improved. However, in their study they knew a priori the intended targets and only considered two selection techniques.

5. Conclusions and future outlook

The act of pointing to graphical elements is one of the fundamental tasks in human–computer interaction. Although 3D interaction techniques for target selection have been used for many years, they still exhibit major limitations regarding effective, accurate selection of targets in real-world applications. Some of these limitations are concerned with visual feedback issues (occlusion, visibility mismatch, depth perception in stereoscopic displays) and the inherent features of the human motor system (instability when interacting in free space, speed-accuracy trade-off, neuromotor noise). More efficient 3D interaction techniques can be designed by devising new strategies for controlling the selection tool and for providing appropriate visual feedback, drawing the inspiration from Fitts' law, occlusion management literature and depth perception studies.

The user performance during a pointing selection task depends on a number of domain-specific factors (such as the shape, layout and density of the targets) as well as hardware-related factors (DoFs, noise, latency and accuracy of the input hardware; field-of-view, resolution, level of immersion and depth quality of the display hardware). Considering all these factors simultaneously as independent variables in controlled experiments is clearly not practical. This fact limits the validity of the findings reported in the 3D user interface literature to a specific domain and a particular hardware setup. The lack of de-facto standard datasets for testing purposes (more common in other scientific communities) along with the plethora of VR hardware setups makes it difficult to make fair comparisons. Furthermore, many of the selection and facilitation techniques we have explored in this paper have been proposed and evaluated in isolation, whereas in the real world selection tasks are mixed with other primary tasks such as manipulation and navigation. These are issues that must still be addressed.

Interaction in VR systems is more physically demanding than traditional interfaces. Usually, users have to stand to benefit from head-coupled perspective rendering, and most interactions take place in the 3D space without physical support for the arms. Recent advances in low-cost but low-accuracy motion capture sensors is pushing 3D user interfaces towards an even more extensive use of the human body. Although progress is being made, and users do not have to carry heavy equipment, the appropriateness of existing 3D user interfaces (and 3D pointing techniques in particular) is still lagging behind when it comes to their use during prolonged periods of time.

Optical depth sensors such as the Kinect are particularly attractive for VR interfaces as the user does not need to carry any device nor wear any marker. Confirming a selection with such controller-free interfaces is more difficult, although future advances in tracking accuracy will enable the recognition of subtle gestures. Another issue is how controller-free interfaces can provide a smooth integration of the selection technique with typical follow-on tasks such as manipulation [16].

An important question is to which extent current interaction techniques for 3D object selection will stand the test of time. A major hindrance for effective user interaction in VEs is the fact that current technology fails to provide the same level of cues for understanding the environment and does not reproduce faithfully the physical constraints of the real world. Improvements in motion capture technology will allow for a more accurate tracking of the user's actions, and better displays will enhance the user's perception of the virtual environment. We believe though that the major conclusions of this survey will still be valid despite forthcoming advances in VR technology. We can provide the user with extremely realistic volumetric displays with no convergence and accommodation mismatch, and perfectly accurate tracking systems, but pointing gestures will still be limited by the human motor system, which is unlikely to improve in the near future. Although new and better interaction techniques will arise, or in a mid-term future, brain-computer interfaces might partially replace traditional gesture-based interfaces, the techniques and performance models we have reviewed in this paper are likely to play an important role in current and upcoming VR applications.

References

[1] Andujar C, Argelaguet F. Anisomorphic ray-casting manipulation for interacting with 2D GUIs. Comput Graph 2007;31(1):15–25.

[2] Andujar C, Argelaguet F. Virtual pads: decoupling motor space and visual space for flexible manipulation of 2D windows within VEs. In: IEEE symposium on 3D user interfaces, 3DUI'07; 2007.

[3] Andujar C, Argelaguet F, Trueba R. Hand-based disocclusion for the world-in-miniature metaphor. Presence: Teleop Virt Environ 2010;19(6):499–512.

[4] Argelaguet F. Pointing facilitation techniques for 3D object selection on virtual environments. PhD thesis, Universitat Politècnica de Catalunya; 2011.

[5] Argelaguet F., Andujar C. Improving 3D selection in immersive environments through expanding targets. In: SG'08: proceedings of the 9th international symposium on smart graphics; 2008. p. 45–57.

[6] Argelaguet F, Andujar C. Efficient 3D pointing selection in cluttered virtual environments. IEEE Comput Graph Appl 2009;29(6):34–43.

[7] Argelaguet F, Andujar C. Visual feedback techniques for virtual pointing on stereoscopic displays. In: Proceedings of the 16th ACM symposium on virtual reality software and technology, VRST '09. ACM; 2009. p. 163–70.

[8] Argelaguet F, Andujar C, Trueba R. Overcoming eye-hand visibility mismatch in 3D pointing selection. InL VRST '08: proceedings of the 2008 ACM symposium on virtual reality software and technology; 2008. p. 43–6.

[9] Argelaguet F, Kunert A, Kulik A, Froehlich B. Improving co-located collaboration with show-through techniques. In: IEEE Symposium on 3D user interfaces; 2010. p. 55–92.

[10] Balakrishnan R. "Beating" Fitts' law: virtual enhancements for pointing facilitation. Int J Hum–Comput Stud 2004;61(6):857–74.

[11] Bederson BB. Fisheye menus. In: UIST '00: proceedings of the 13th annual ACM symposium on user interface software and technology; 2000. p. 217–25.

[12] Boeck JD, Weyer TD, Raymaekers C, Coninx K. Using the non-dominant hand for selection in 3D. In: IEEE symposium on 3D user interfaces (3DUI '06); 2006. p. 53–8.

[13] Bolt RA. "Put-that-there": voice and gesture at the graphics interface. In: SIGGRAPH '80: proceedings of the seventh annual conference on computer graphics and interactive techniques. ACM, 1980. p. 262–70.

[14] Bowman DA, Badillo B, Manek D. Evaluating the need for display-specific and device-specific 3D interaction techniques. In: ICVR'07: proceedings of the 2nd international conference on virtual reality, Springer-Verlag, 2007. p. 195–204.

[15] Bowman DA, Gabbard JL, Hix D. A survey of usability evaluation in virtual environments: classification and comparison of methods. Presence: Teleop Virt Environ 2002;11(4):404–24.

[16] Bowman DA, Hodges LF. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In: I3D '97: proceedings of the 1997 symposium on interactive 3D graphics; 1997. p. 35–ff.

[17] Bowman DA, Hodges LF, Bolter J. The virtual venue: user-computer interaction in information-rich virtual environments. Presence: Teleop Virt. Environ. 1998;7(5):478–93.

[18] Bowman DA, Johnson DB, Hodges LF. Testbed evaluation of virtual environment interaction techniques. In: VRST '99: proceedings of the ACM symposium on virtual reality software and technology. ACM, 1999. p. 26–33.

[19] Bowman DA, Kruijff E, LaViola JJ, Poupyrev I. 3D user interfaces: theory and practice. Addison Wesley; 2004.

[20] Bowman DA, Wingrave CA, Campbell J. Using pinch gloves for both natural and abstract interaction techniques in virtual environments. In: HCI international; 2001. p. 629–33.

[21] Brewster S. Multimodal feedback for the acquisition of small targets. Ergonomics 2005;48(22):1129–50.

[22] Burns M, Finkelstein A. Adaptive cutaways for comprehensible rendering of polygonal scenes. ACM Trans Graph 2008;27(5):1–7.

[23] Card SK, Mackinlay JD, Robertson GG. A morphological analysis of the design space of input devices. ACM Trans Inf Syst 1991;9(2):99–122.
[24] Cashion J, Wingrave CA, LaViola JJ. Dense and dynamic 3D selection for game-based virtual environments. IEEE Trans Visualization Comput Graph 2012;18(4):634–42.
[25] Casiez G, Vogel D, Balakrishnan R, Cockburn A. The impact of control–display gain on user performance in pointing tasks. In: Human–computer interaction, vol. 23; 2009. p. 215–50.
[26] Cockburn A, Brock P. Human on-line response to visual and motor target expansion. In: GI '06: proceedings of graphics interface; 2006. p. 81–7.
[27] Cournia N, Smith JD, Duchowski AT. Gaze vs hand-based pointing in virtual environments. In: CHI '03: CHI '03 extended abstracts on Human factors in computing systems; 2003. p. 772–3.
[28] Dachselt R, Hübner A. A survey and taxonomy of 3D menu techniques. In: EGVE 06: proceedings of the 12th eurographics symposium on virtual environments; 2006. p. 89–99.
[29] Dang N-T. A survey and classification of 3D pointing techniques. In: 2007 IEEE international conference on research, innovation and vision for the future. IEEE, March 2007. p. 71–80.
[30] de Haan G, Koutek M, Post FH. IntenSelect: using dynamic object rating for assisting 3D object selection; 2005. p. 201–9.
[31] Elmqvist N. BalloonProbe: reducing occlusion in 3D using interactive space distortion. In: VRST '05: proceedings of the ACM symposium on virtual reality software and technology; 2005. p. 134–7.
[32] Elmqvist N, Tsigas P. View-projection animation for 3D occlusion management. Comput Graph 2007;31(6):864–76.
[33] Elmqvist N, Tsigas P. A taxonomy of 3D occlusion management for visualization. IEEE Trans. Visualization Comput Graph 2008;14(5):1095–109.
[34] Fitts P. The information capacity of the human motor system is controlled by the amplitude of movement. J. Exp. Psychol. 1954;6(47):381–91.
[35] Fitts P, Peterson J. Information capacity of discrete motor response. J. Exp. Psychol. 1964;2(67):103–12.
[36] Forsberg A, Herndon K, Zeleznik R. Aperture based selection for immersive virtual environments. In: UIST '96: proceedings of the 9th annual ACM symposium on user interface software and technology; 1996. p. 95–6.
[37] Frees S, Kessler D. Precise and rapid interaction thought scaled manipulation in immersive virtual environments. IEEE Virtual Reality 2005:99–106.
[38] Frees S, Kessler GD, Kay E. PRISM interaction for enhancing control in immersive virtual environments. ACM Trans Comput–Hum Interact 2007;14(1).
[39] Gabbard JL. A taxonomy of usability characteristics for virtual environments. Master's thesis, Department of Computer Science, Virginia Tech; 1997.
[40] Grossman T, Balakrishnan R. Pointing at trivariate targets in 3D environments. In: CHI '04: proceedings of the SIGCHI conference on human factors in computing systems. ACM, 2004. p. 447–54.
[41] Grossman T, Balakrishnan R. The design and evaluation of selection techniques for 3D volumetric displays. In: UIST '06: proceedings of the 19th annual ACM symposium on user interface software and technology. ACM, 2006. p. 3–12.
[42] Grossman T, Wigdor D. Going deeper: a taxonomy of 3D on the tabletop. In: IEEE TABLETOP '07, 2007. p. 137–44.
[43] Herndon KP, van Dam A, Gleicher M. The challenges of 3D interaction: a CHI'94 workshop. SIGCHI Bull 1994;26(4):36–43.
[44] Hinckley K, Pausch R, Goble JC, Kassell NF. A survey of design issues in spatial input. In: UIST '94: Proceedings of the 7th ACM symposium on user interface software and technology; 1994. p. 213–22.
[45] Hinckley K, Pausch R, Goble JC, Kassell NF. Passive real-world interface props for neurosurgical visualization. In: CHI '94: proceedings of the SIGCHI conference on human factors in computing systems. ACM; 1994. p. 452–8.
[46] Jacob RJK, Girouard A, Hirshfield LM, Horn MS, Shaer O, Solovey ET, Zigelbaum J. Reality-based interaction: a framework for post-WIMP interfaces. In: Proceeding of the twenty-sixth annual SIGCHI conference on human factors in computing systems, CHI '08. ACM, 2008. p. 201–10.
[47] Jagacinski RJ, Monk DL. Fitts' law in two dimensions with hand and head movements. J Mot Behav 1985;17(1):77–95.
[48] König W, Gerken J, Dierdorf S, Reiterer H. Adaptive pointing - design and evaluation of a precision enhancing technique for absolute pointing devices. In: Human–computer interaction, INTERACT 2009, vol. 5726, 2009. p. 658–71.
[49] Kopper R, Bacim F, Bowman DA. Rapid and accurate 3D selection by progressive refinement. In: IEEE symposium on 3D user interfaces (3DUI'11); 2011. p. 67–74.
[50] Kopper R, Bowman DA, Silva MG, McMahan RP. A human motor behavior model for distal pointing tasks. Int J Hum–Comput Stud 2010;68(October (10)):603–15.
[51] Kulik A. Building on realism and magic for designing 3D interaction techniques. IEEE Comput Graph Appl 2009;29(6):22–33.
[52] LaViola Jr. JJ. A discussion of cybersickness in virtual environments. SIGCHI Bull 2000;32(1):47–56.
[53] Lee S, Seo J, Kim GJ, Park C-M, Evaluation of pointing techniques for ray casting selection in virtual environments. In: Third international conference on virtual reality and its application in industry, vol. 4756(1); 2003. p. 38–44.
[54] Liang J, Green M. JDcad: a highly interactive 3D modeling system. Comput Graph 1994;18(4):499–506.
[55] Lindeman RW, Sibert JL, Hahn JK. Hand-held windows: towards effective 2D interaction in immersive virtual environments. In: IEEE virtual reality; 1999. p. 205–12.
[56] Linderman R, Sibert J, Hahn J. Towards usable VR: an empirical study of user interfaces for immersive virtual environments, In: Proceedings of the SIHCHI conference on human factors in computing systems; 1999. p. 64–71.
[57] MacKenzie CL, Marteniuk RG, Dugas C, Liske D, Eickmeier B. Three-dimensional movement trajectories in Fitts' task: implications for control. Q J Exp Psychol A: Hum Exp Psychol 1987;39(4):629–47.
[58] MacKenzie IS. Fitts' law as a research and design tool in human–computer interaction. Hum Comput Interact 1992;7(1):91–139.
[59] MacKenzie IS, Kauppinen T, Silfverberg M. Accuracy measures for evaluating computer pointing devices. In: CHI '01: proceedings of the SIGCHI conference on human factors in computing systems, CHI '01. ACM; 2001. p. 9–16.
[60] MacKenzie IS, Ware C. Lag as a determinant of human performance in interactive systems. In: CHI '93: proceedings of the INTERACT '93 and CHI '93 conference on human factors in computing systems. ACM; 1993. p. 488–93.
[61] Martinet A, Casiez G, Grisoni G. The design and evaluation of 3D positioning techniques for multi-touch displays. In: IEEE symposium on 3D user interfaces, 3DUI'10; 2010. p. 87–94.
[62] Meyer DE, Abrams RA, Kornblum S, Wright CE, Smith JEK. Optimality in human motor performance: ideal control of rapid aimed movements. Psychol Rev 1988;95(3):340–70.
[63] Mine MR. Virtual environments interaction techniques. Technical Report TR94-018, Department of Computer Science, University of North Carolina at Chapel Hill; 1995.
[64] Mine MR, Brooks FP Jr., Sequin C. Moving objects in space: exploiting proprioception in virtual-environment interaction. In: SIGGRAPH '97: proceedings of the 24th annual conference on computer graphics and interactive techniques; 1997. p. 19–26.
[65] Müller-Tomfelde C, Baranauskas C, Palanque P, Abascal J, Barbosa S. Dwell-based pointing in applications of human computer interaction. In: Baranauskas C, Palanque P, Abascal J, Barbosa SDJ, editors, Human–computer interaction - INTERACT 2007, vol. 4662, Berlin, Heidelberg; 2007. p. 560–73.
[66] Murata A. Extending effective target width in Fitt's law to two-dimensional pointing task. Int J Hum–Comput Interact 1999;11:137–52.
[67] Octavia J, Raymaekers C, Coninx K. Adaptation in virtual environments: conceptual framework and user models. Multimedia Tools Appl 2010:1–22.
[68] Olwal A, Benko H, Feiner S. SenseShapes: using statistical geometry for object selection in a multimodal augmented reality system. In: ISMAR '03: proceedings of the 2nd IEEE/ACM international symposium on mixed and augmented reality; 2003. p. 300.
[69] Olwal A, Feiner S. The flexible pointer: an interaction technique for selection in augmented and virtual reality. In: Conference supplement of UIST '03 (ACM symposium on user interface software and technology); 2003. p. 81–2.
[70] Pavlovych A, Stuerzlinger W. The tradeoff between spatial jitter and latency in pointing tasks. In: EICS '09: proceedings of the 1st ACM SIGCHI symposium on engineering interactive computing systems, ACM; 2009. p. 187–196.
[71] Peter H, Roland W, Bues BM. ISith – intersection-based spatial interaction for two hands. In: IEEE symposium on 3D user interfaces, 3DUI'06; 2006. p. 59–61.
[72] Pfeiffer T. Towards a linguistically motivated model for selection in virtual reality. In: IEEE virtual reality 2012 (VR'12). IEEE; March 2012. p. 89–90.
[73] Pierce JS, Forsberg A, Conway MJ, Hong S, Zeleznik R, Mine MR. Image plane interaction techniques in 3D immersive environments. In: I3D '97: proceedings of the 1997 symposium on interactive 3D graphics; 1997. p. 39–ff.
[74] Plumert JM, Kearney JK, Cremer JF. Distance perception in real and virtual environments. In: Proceedings of the 1st symposium on applied perception in graphics and visualization – APGV '04, New York. New York, USA: ACM Press; August 2004. p. 27.
[75] Poupyrev I, Billinghutst M, Weghorst S, Ichikawa T. The Go-Go interaction technique: non-linear mapping for direct manipulation in VR. In: UIST '96: proceedings of the 9th annual ACM symposium on user interface software and technology; 1996. p. 79–80.
[76] Poupyrev I, Ichikawa T. Manipulating objects in virtual worlds: categorization and empirical evaluation of interaction. J Visual Lang Comput 1999;10(1): 19–35.
[77] Poupyrev I, Weghorst S, Billinghurst M, Ichikawa T. A framework and testbed for studying manipulation techniques for immersive VR. In: VRST '97: proceedings of the ACM symposium on virtual reality software and technology, ACM; 1997. p. 21–8.
[78] Poupyrev I, Weghorst S, Billinghutst M, Ichikawa T. Egocentric object manipulation in virtual environments: empirical evaluation of interaction techniques. Comput Graph Forum 1998;3(17):41–52.
[79] Sarkar M, Brown MH. Graphical fisheye views of graphs. In: CHI '92: proceedings of the SIGCHI conference on Human factors in computing systems; 1992. p. 83–91.
[80] Schmidt G, Baillot Y, Brown DG, Tomlin EB, Swan JEII. Toward disambiguating multiple selections for frustum-based pointing. In: IEEE symposium on 3D user interfaces (3DUI'06); 2006. p. 87–94.
[81] Schmidt RA, Zelaznik H, Hawkins B, Frank JS, Quinn Jr. JT. Motor-output variability: a theory for the accuracy of rapid motor acts. Psychol Rev 1979;86(5):415–51.

[82] Smith BA, Ho J, Ark W, Zhai S. Hand eye coordination patterns in target selection. In: Proceedings of the 2000 symposium on eye tracking research & applications, ETRA '00, ACM; 2000. p. 117–22.

[83] Steed A. Selection/towards a general model for selection in virtual environments. In: IEEE symposium on 3D user interfaces (3DUI'06); 2006. p. 103–10.

[84] Steed A, Parker C. 3D selection strategies for head tracked and non-head tracked operation of spatially immersive displays. In: Eighth international immersive projection technology workshop; 2004. p. 163–70.

[85] Steinicke F, Ropinski T, Hinrichs K. Object selection in virtual environments with an improved virtual pointer metaphor. In: Computer vision and graphics, international conference, ICCVG; 2004. p. 320–6.

[86] Stoakley R, Conway MJ, Pausch R. Virtual reality on a WIM: interactive worlds in miniature. In: CHI '95: proceedings of the SIGCHI conference on human factors in computing systems; 1995. p. 265–72.

[87] Tanriverdi V, Jacob RJK. Interacting with eye movements in virtual environments. In: CHI '00: proceedings of the SIGCHI conference on human factors in computing systems, ACM; 2000. p. 265–72.

[88] Teather RJ, Pavlovych A, Stuerzlinger W, MacKenzie IS. Effects of tracking technology, latency, and spatial jitter on object movement. In: IEEE symposium on 3D user interfaces; 2009. p. 43–50.

[89] Teather RJ, Stuerzlinger W. Pointing at 3D targets in a stereo head-tracked virtual environment. In: IEEE symposium on 3D user interfaces (3DUI'11); 2011. p. 87–94.

[90] Tyndiuk F, Thomas G, Lespinet-Najib V, Schlick C. Cognitive Comparison of 3D interaction in front of large vs. small displays. In: VRST '05: proceedings of the ACM symposium on virtual reality software and technology; 2005. p. 117–23.

[91] Valkov D, Steinicke F, Bruder G, Hinrichs K. 2D touching of 3D stereoscopic objects. In: Proceedings of the 2011 annual conference on human factors in computing systems, CHI '11. ACM; 2011. p. 1353–62.

[92] van Galen GP, de Jong WP. Fitts' law as the outcome of a dynamic noise filtering model of motor control. Hum Movement Sci 1995;14(4–5):539–71.

[93] 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, 3DUI '07; 2007. p. 115–22.

[94] Vanacken L, Raymaekers C, Coninx K. Evaluating the influence of multimodal feedback on egocentric selection metaphors in virtual environments. Haptic and audio interaction design; 2006. p. 12–23.

[95] Vogel, D., Balakrishnan R. Distant freehand pointing and clicking on very large, high resolution displays. In: UIST '05: proceedings of the 18th annual ACM symposium on user interface software and technology. ACM Press; 2005. p. 33–42.

[96] Wang Y, MacKenzie C. Effects of orientation disparity between haptic and graphic displays of objects in virtual environments. In: INTERACT '99; 1999. p. 391–8.

[97] Ware C, Balakrishnan R. Reaching for objects in VR displays: lag and frame rate. ACM Trans Hum Comput Interact 1994;1(4):331–56.

[98] Ware C, Lowther K. Selection using a one-eyed cursor in a fish tank VR environment. ACM Trans Hum Comput Interact 1997;4(4):309–22.

[99] Watson B, Spaulding V, Walker N, Ribarsky W. Evaluation of the effects of frame time variation on VR task performance. In: Virtual reality annual international symposium. IEEE; 1997. p. 38–44.

[100] Wiebel A, Vos FM, Foerster D, Hege H-C. WYSIWYP: what you see is what you pick. IEEE Trans Visualization Comput Graph 2012;18(December (12)):2236–44.

[101] Wingrave, CA, Bowman DA. Baseline factors for raycasting selection. In: Proceedings of virtual reality international; 2005.

[102] Wingrave CA, Bowman DA, Ramakrishnan N. Towards preferences in virtual environment interfaces. In: EGVE '02: proceedings of the workshop on virtual environments; 2002. p. 63–72.

[103] Wingrave CA, Haciahmetoglu Y, Bowman DA. Overcoming world in miniature limitations by scaling and scrolling. In: IEEE symposium on 3D user interfaces (3DUI'06); 2006. p. 11–6.

[104] Wingrave CA, LaViola JJ. Reflecting on the design and implementation issues of virtual environments. Presence: Teleop Virt Environ 2010;19(2):179–95.

[105] Wingrave CA, Laviola Jr. JJ, Bowman DA. A natural, tiered and executable UIDL for 3D user interfaces based on concept-oriented design. ACM Trans Comput–Hum Interact 2009;16(4):1–36.

[106] Wingrave CA, Tintner R, Walker BN, Bowman DA, Hodges LF. Exploring individual differences in raybased selection: strategies and traits. In: IEEE virtual reality; 2005. p. 163–70.

[107] Zhai S, Buxton W, Milgram P. The "Silk Cursor": investigating transparency for 3D target acquisition. In: CHI '94: proceedings of the SIGCHI conference on human factors in computing systems; 1994. p. 459–64.

[108] Zhai S, Milgram P, Buxton W. The influence of muscle groups on performance of multiple degree-of-freedom input. In: Proceedings of the SIGCHI conference on human factors in computing systems: common ground, CHI '96, ACM; 1996. p. 308–15.