Desktop Orbital Camera Motions Using Rotational Head Movements

Thibaut Jacob[1], Gilles Bailly[1], Eric Lecolinet[1], Géry Casiez[2], Marc Teyssier[1]
[1]LTCI, CNRS, Telecom ParisTech, France, [2]Université de Lille, France
{thibaut.jacob,gilles.bailly,eric.lecolinet,marc.teyssier}@telecom-paristech.fr,
gery.casiez@univ-lille1.fr

ABSTRACT

In this paper, we investigate how head movements can serve to change the viewpoint in 3D applications, especially when the viewpoint needs to be changed quickly and temporarily to disambiguate the view. We study how to use yaw and roll head movements to perform orbital camera control, i.e., to rotate the camera around a specific point in the scene. We report on four user studies. Study 1 evaluates the useful resolution of head movements. Study 2 informs about visual and physical comfort. Study 3 compares two interaction techniques, designed by taking into account the results of the two previous studies. Results show that head roll is more efficient than head yaw for orbital camera control when interacting with a screen. Finally, Study 4 compares head roll with a standard technique relying on the mouse and the keyboard. Moreover, users were allowed to use both techniques at their convenience in a second stage. Results show that users prefer and are faster (14.5%) with the head control technique.

Keywords

Head motion; 3D interaction; camera control; transfer function

1. INTRODUCTION

Manipulating 3D objects require designers to frequently change the viewpoint to avoid occlusion, see details, perceive depth or get a global view of the scene [6]. Interaction techniques using the mouse or the keyboard have been proposed to manipulate the viewpoint [6, 9], but this requires users to use the same modality (hand gestures) to control both what they see and manipulate. As a result, users must continuously switch between editing tasks and camera control, which interrupts their workflow, may impair their attention and increase execution time [6, 12, 28].

In the physical world, humans use head and eyes movements to control what they see and limbs movements to manipulate objects. Using head movements as an additional input channel to control camera motion may thus provide better comfort and increase the interaction bandwidth. This may be especially true when the viewpoint needs to be changed quickly and temporarily to disambiguate the view. A quick glance often suffices for this purpose, head interaction seem an efficient and natural way of performing to-and-fro temporary movements of the camera. Moreover, head-camera coupling may provide a more ecological visual perception of 3D scenes [18].

Head movement has been used in several studies to improve the feeling of immersion in Virtual Reality environments, either using head-mounted displays [32] or CAVEs [11]. But this approach has seldom been investigated for desktop workstations [20] although such an approach can be implemented at little cost as many computers have an integrated webcam.

In this paper, we investigate how to best define head-camera couplings to favor both comfort and efficiency [6]. We focus on *orbital control* because this type of camera motion is frequently used in 3D software (Blender, SketchUp), especially in 3D room-planning applications (IKEA Home Planner) or 3D sound interfaces. We focus on screen desktop environments because they are still the most used for 3D editing.

In this context, we report the findings of four user studies. Study 1 investigates the widest angles at which users can rotate the head on yaw and roll axes while maintaining a high level of physical and visual comfort. Results show that,

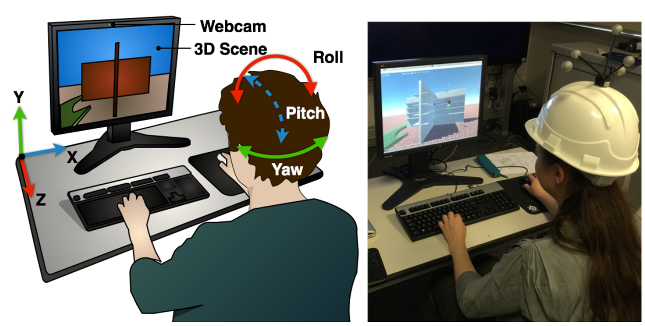


Figure 1: *Left.* Envisioned system: Yaw or roll rotations are captured with a webcam for orbital camera control. *Right.* Experimental setup: The head orientation is measured using a tree target attached to a cap.

SUI '16, October 15-16, 2016, Tokyo, Japan

DOI: http://dx.doi.org/10.1145/2983310.2985758

when taking into account both criteria, larger head angles can be performed for roll (35°) than for yaw (26°).

Study 2 investigates the useful resolution [1] i.e. the smallest movements that can be willingly operated by users. Results show that (a) the useful resolution is 1° for a 95% success rate for both head yaw and roll and (b) that accuracy decreases with larger starting angles and with smaller amplitudes.

Building on these studies, we designed a transfer function for controlling orbital camera motion with the head. We derived two multimodal interaction techniques combining head movement (either roll or yaw) for controlling the camera and mouse input for selecting objects in a 3D scene. Study 3 compares the performance (success rate, time of completion) of these two techniques on a 3D task. Results show that participants performed better and preferred using roll than yaw.

Finally, Study 4 compares the technique using head roll with a standard technique relying on the mouse/keyboard. Results show that users are faster (14.5%) with our technique and they find it more comfortable.

2. RELATED WORK

2.1 Camera controls in 3D software

3D applications such as Unity Editor or Blender provide multiple camera controls (Table 1) that rely on the mouse and/or the keyboard. Camera *tilting* or *orbit control* (the camera rotates around a specific point in the scene, keeping this point in the center of the viewport) are generally performed by dragging the mouse or by pressing dedicated keys. Camera *translations* usually rely on arrow keys. The mouse wheel is typically used for *zooming*. Other camera controls (e.g., *fly mode* or roll motions) generally require using buttons on a toolbar.

As shown in Table 1, floor planning applications provide orbit control on two axes (yaw and pitch) so that the user can rotate around the scene both horizontally and vertically. Horizontal orbit control is especially useful for changing the viewpoint quickly. More generally, applications for 3D room planning, 3D sound editing or real-time strategy games (e.g. StarCraft) constrain camera motion and only provide a subset of camera controls. These applications tend to favor orbital camera control with a fixed height to obtain an isometric point of view, which is especially appropriate in this context.

Software	Orbit	Pan	Tilt
Blender, Maya	all axes	all axes	all axes
SketchUp	yaw, pitch	y, z	yaw, pitch
Google Earth	no	all axes	all axes
MeshLab	yaw, pitch, roll	x, y, z	no
IKEA Home Planner	yaw, pitch	x, z	no
floorplanner.com	yaw, pitch	x, z	no
RoomSketcher	yaw, pitch	z	yaw, pitch

Table 1: Camera controls commonly used in 3D applications

Numerous academic interaction techniques have been proposed for controlling the camera[38, 17, 39, 22, 3]. In this wide literature, we focus on head-camera coupling.

2.2 Head-camera coupling

Head-camera coupling has been investigated in different contexts such as desktop workstations [4], mobile devices [19, 10], tabletops [35] or CAVE systems [11], as well as different applications such as VR [22, 13], video-conferencing [20, 39], video games [23], teleoperation [29], accessibility [21] or surgery [33]. Therefore, different mappings were proposed:

Fishtank. Fishtank VR [38] is probably the most famous head-camera coupling. It enhances the perception of distance in 3D scenes with motion parallax. This technique couples the position of the camera to the position of the user's head. While it provides a sense of immersion in 3D scenes, this technique only allows relatively small displacements and cannot be used to inspect the different sides of an object.

Tilting. Tilting consists of rotating the camera around its own center. In VR [23], it is generally coupled to head yaw and pitch (e.g., in Occulus Rift games) but other mappings have been proposed in the literature [17, 39]. Some techniques combine multiple camera controls, such as [27] which combines Fishtank and tilting to extend the user's field of view.

Orbiting. In head-camera coupling systems, orbit control is performed around the vertical axis and coupled to the yaw rotation of the head [22]. In contrast, we aim at understanding which of yaw or roll rotations of the head is most appropriate for orbiting the camera.

2.3 Transfer function

Various transfer functions can be applied to transform head movements into camera motions. One approach consists in scaling head movements with a constant value (Control-Display gain or CD gain) [36, 24]. Teather et al. [36] considered gain values of 2, 3 and 5 to increase the amplitude of camera motion in a Fishtank system. Results showed no significant effect on time nor accuracy but participants preferred a 1:3 gain. Mulder et al. [27] amplified tilting rotations by a factor of 2 to extend the users' field of view. Poupyrev et al. [30] studied the performance of non-isomorphic rotations (1.8:1) against a 1:1 mapping. Results showed a 13% speed improvement for the latter for large amplitudes. Other amplification factors (1:1, 2:1, 3:1, 4:1) have been tested in [24], with a 15% speed improvement for a 3:1 gain without significant loss in accuracy compared to 1:1.

Non-linear gains have also been considered, as in [31] where, for 3D rotations, the gain remains constant to 1 under a certain threshold then becomes non-linear. PRISM [16] is a rotation technique where the rotation gain depends on the speed of the user. Offset recovery is provided in order to null the offset that is progressively accumulated. A drawback of this approach is its non nulling-compliance [30, 6].

3. DESIGN RATIONALE

We now describe the design rationale that motivated the studies and the techniques presented in this article.

3.1 Camera control and 3D editing

Editing a 3D virtual scene with a desktop application is a complex task involving many operations such as adding and removing objects, editing vertices position or modifying object properties, etc. Camera control is crucial to avoid occlusion, observe objects under different perspectives, explore the 3D scene or enhance depth perception through motion.

However, as mentioned in the introduction, camera control might interrupt the users' workflow, impair attention and increase completion time [6, 12, 28, 34].

As an example, let's consider a user who wants to move an object outside of the field of view of the camera. The user will have to 1) rotate and move the camera as far as possible while maintaining the object in the field of view; 2) drag the object in the direction of the desired location until reaching the border of the 3D scene; 3) repeat these operations until the object is close enough to the target location. Our objective is to reduce the cost of these interleave operations and to let users focus on their primary task, the edition of the 3D scene.

Using head movement to control the camera enables leaving the hands free for performing manipulation tasks, as in the physical world. This may be especially useful when the viewpoint needs to be temporarily changed, as when moving objects in the previous example, or if a quick to-and-fro movement of the camera is needed to disambiguate the view.

3D applications (Table 1) provide various types of camera motions, which availability depends on the application. However, interviews with CG artists and an analysis of 3D editing tools (Maya, Blender, MeshLab) and floor planning applications showed that orbit control is especially useful when editing a scene because it maintains the focus on the objects of interest, which allows users to easily rotate around them. Consequently, we chose to focus our study on this type of camera control.

While we focus on orbital control in this article, other head movements could also be used to control other camera movements such as zooming, panning etc. as will be discussed in the last section. However, controlling the camera with the head is probably mostly useful for camera controls that occur very frequently. Controlling many camera movements with the head might be ambiguous and hard to perform. Standard techniques relying on the mouse or the keyboard are probably more appropriate for camera controls that are seldom used.

3.2 Head movement

The head has six degrees of freedom (Figure 1) three degrees of rotation (yaw, roll, pitch) and three degrees of translation (T_x, T_y, T_z). However, they cannot be fully combined due to biomechanical constraints. We conducted a brainstorming session to find out which kind of input would be most suited for controlling orbital motion. We did not consider translations because they are difficult to perform while seating in a working chair. While the T_x translation seemed easier to perform, we observed that in fact participants tended to roll the head. We also withdrew pitch rotations because a pilot study revealed that participants did not spontaneously use such rotations for orbital control. We thus retained two types of head movements: *yaw* and *roll* rotations, which have a respective average amplitude of 70° and 40° in both directions [37].

In order to make the technique simple and mostly similar to what people do in the physical world, we decided to use position control and to limit the need of clutching. Hence, an appropriate rotation of the head produces an homothetic movement of the camera, according to the CD gain coefficient (which calculation is detailed below). Besides simplicity, another reason we chose position control rather than rate control is that the latter is more suitable for isometric or elastic devices (e.g. a joystick). Such devices feature a self-centering mechanism to return to their neutral state when released [8], which is not the case with the head. Finally, clutching tends to require additional movements, thus additional time [8, 14].

However, while head movement seem an efficient and natural way for performing temporary movements of the camera, it may not be appropriate for long editing sequences. Rotating the head is convenient if the user needs to hold her position for a limited amount of time, but a static posture may become unconfortable for longer periods. Head rotations may also be insufficient for covering long distances. For instance, a rotation of more than 360° would force users to go over (if even possible) their physiological limit. To alleviate this limitation, we made our technique *fully compatible* with existing mouse or keyboard camera controls. Hence, users can still use the mouse or the keyboard to change the default viewpoint. Once it is defined, head motion can be used for frequent and short camera movements.

3.3 CD gain

The CD gain is a unit-free coefficient which maps head movement to camera motion. With a CD gain of 1, the camera and the head are rotating by an identical amount. It turns proportionality farther and faster for a larger gain and covers less angular distance but offers higher precision for smaller values. A task can be characterized by the maximum camera angular distance A_{max}^{cam} and (depending on the smallest target size) the minimum angle W_{min}^{cam} needed to properly accomplish this task. The minimal CD gain is represented by the following formula:

$$CDgain_{min} = \frac{A_{max}^{cam}}{A_{max}^{head}} \quad (1)$$

where A_{max}^{head} is the widest angle users can perform during head rotations. This value might depend on the rotation axis (yaw vs. roll). Similarly, the maximal CD gain is:

$$CDgain_{max} = \frac{W_{min}^{cam}}{\hat{R}} \quad (2)$$

where $\hat{R}$ is the *useful resolution* of the device, i.e., the smallest movement users can willingly operate.

The task cannot be achieved if $CDgain_{min} > CDgain_{max}$ as, either the farthest target could not be reached or the smallest target could not be selected. Clutching is then necessary to move the camera over longer distances without impairing precision (see Discussion). The next sections present two user studies we conducted to estimate A_{max}^{head} and the useful resolution $\hat{R}$.

3.4 Resolution measurement

Device Resolution. While our technique is intended to work with non expensive equipment such as an embedded webcam (see the Discussion and the video), we used an ART motion capture system (www.ar-tracking.com/) for the sake of precision in the following experiments. The setup consisted of a cap with a mounted passive tree target (Figure 1).

We first conducted a pilot study to estimate R_{device}, the precision of our motion capture system. Due to environmental conditions (distance between the camera and the markers, lighting conditions, etc.) raw measurements are noisy even when the markers are perfectly stable. In order to get a better estimation of the device resolution, we used a method inspired by Bérard et al. that consists in positioning the input device at a fixed position and recording output during a given amount of time [5]. The estimated resolution then equals four times the standard deviation. We collected raw data for one hour with a passive tree target (ART TT3) fixed on a table. Results showed a R_{device} precision higher than 0.05° for all rotation axes, a value almost twenty times more precise than human head movements (as will be seen in Study 2).

Effective resolution of the head. We distinguish the useful resolution $\hat{R}$, which is the smallest movement users can willingly operate, and the *effective resolution* R_{eff}^{head} (or head noise) which represents the amount of movement users perform while remaining still because of uncontrolled tremor. This value is useful to ensure the stability of the interaction and avoid uncomfortable jitters. To estimate R_{eff}^{head}, we conducted a pilot study where participants were asked to remain still during 1 minute. A visual feedback (white cursor) represented the current orientations (yaw and roll) to guide participants. For each second interval we computed the range of movement, and obtained an average range of movement R_{eff}^{head} of 0.2°. This value is 4 times larger than the noise of the motion capture system we previously mentioned ($R_{device} = 0.05°$).

4. STUDY 1: COMFORT

The goal of this study was to estimate the widest angle users can perform during yaw and roll head rotations while maintaining a high level of *physical* and *visual* comfort. Biomechanic studies state that humans can perform larger yaw (70°) than roll rotations (40°) [37]. However, these studies focused on physiological amplitudes and did not consider the level of visual nor physical comfort when interacting with a screen.

4.1 Experimental design

Participants and Apparatus. 12 participants (3 females) aged 23 to 33 ($\bar{x} = 28; \sigma = 2.66$) were recruited from our institution via mailing lists and received a handful of candies for their participation. The setup (Figure 1) was a MacBook Pro laptop connected to an external 17-Inch screen, an external keyboard and a mouse. Participants controlled the keyboard using their non-dominant hand and the mouse using the other hand. The screen was at a distance of 50 cm from the participant. The seat could not be rotated.

Stimulus and task. Participants performed a visual search task. The stimulus was a 3-letter word [7] displayed on the top of the screen. Participants had to find this word in a 7×7 grid full of distractors as fast and accurately as possible.

Conditions. We both tested *Yaw* and *Roll* ROTATIONS. As said before, we withdrew pitch head movements because they were not spontaneously used by participants for this task in a pilot study. We also controlled the DIRECTION of the rotations (*Left* or *Right*).

Procedure. We first asked the participants to put their head in resting position and calibrated our tracking system to that the participants' resting positions corresponded to 0°. The system then indicated a rotation axis (Yaw or Roll) and a direction (Left or Right). Participants turned their head until reaching the largest amplitude ensuring (1) visual comfort (the screen remains in the field of view of the participant) and (2) physical comfort (no muscular tension of the neck nor the eyes). Once they adopted the chosen posture, they pressed the space bar to start the visual search task. The trial finished as soon as the participant clicked on the target word.

Design. We used a within-participant design. The order of appearance of the conditions was counterbalanced between participants. Each condition was repeated seven times. The name and the location of the targets were randomly picked from a set of pre-defined values. For each trial, we measured the rotation angle when participants found the target word. In summary, the experimental design was: 12 Participants × 2 ROTATIONS (Yaw and Roll) × 2 DIRECTIONS (Left and Right) × 7 Repetitions = 336 selections.

4.2 Results and discussion

The results are summarized in Table 2. ANOVA revealed a significant effect for ROTATION on Angle ($F_{1,33} = 26.1, p < .0001$). A post Tukey test showed that users performed wider angles for *Roll* (35.3°) than *Yaw* (25.8°) rotations. No DIRECTION or ROTATION × DIRECTION effect was found on angles. ANOVA confirmed that the conditions had no effect on the visual search task (speed and accuracy).

Because bio-mechanic studies showed that humans can perform larger yaw than roll rotations [25], we expected "comfortable" rotations to be larger for yaw. Our results show an opposite effect because participants had to perform (1) comfortable head movements while (2) looking at the screen. Participants reported that yaw rotations quickly became visually uncomfortable because the screen shifted out of their field of view. This effect will be emphasized with larger screen as the user's field of view will cover less screen space. In contrast, participants' field of view remained unchanged during roll rotations.

Rotation	**Direction** (°)	**Angle** (°)	CI
Yaw	Left	25.0	3.7
	Right	26.5	3.2
Roll	Left	36.8	6.9
	Right	33.9	6.9

Table 2: Average and 95% confidence interval (CI) for the maximum comfortable angles for Rotation and Direction.

5. STUDY 2: USEFUL RESOLUTION

The objective of this experiment was to study the *useful resolution* of rotational head movements, i.e. the smallest movements that can be willingly operated by users.

5.1 Experimental design

Participants and Apparatus. We used the same participants and apparatus than in the previous study.

Method. We followed a methodology similar to Aceituno et al. [1] to find the useful resolution of the head. This method aims at defining the *smallest displacement users can reliably produce*. It differs from methods aiming at defining the smallest target size that users can acquire (such as [5]) because it focuses on controlling ballistic and corrective sub-movements users perform. This method has been successfully used to investigate human limits in small unidirectional mouse movements. In contrast, we apply this

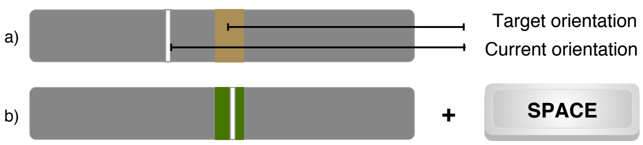


Figure 2: Participants first move the handle with the head to reach the initial orientation. When the initial orientation is reached, the section becomes green. Participants then press the spacebar to start the trial.

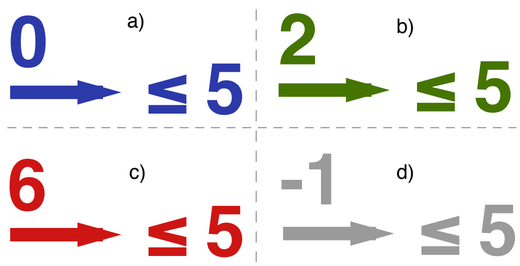


Figure 3: The different states of a trial are color coded: during trial (a), trial validated (b), trial failed (c), trial canceled (d).

method to find the human limits in small rotational (yaw and roll) head movements.

The main procedure of this method is the following: (1) choose a wide maximum amplitude (A) to test the resolution (R_A); (2) ask participants to perform a rotational head movement smaller than A. Repeat this action k times and compute the success rate (S_A); (3) choose a lower amplitude (A) and return to step 2; (4) the useful resolution $\hat{R}$ is the greatest R_A for which the success rate (S_A) $\geq 95\%$.

Factors. We controlled four factors: ROTATION was either *Yaw* or *Roll*. DIRECTION was either *Left* or *Right*. We also considered the INITIAL ORIENTATION of the head because we hypothesized it could influence the useful resolution of head rotations due to biomechanical constraints. We thus defined three different values of INITIAL ORIENTATION for each type of ROTATION: 0°, 10°, 20° for yaw and 0°, 20°, 30° for roll. These values are compatible with the findings of Study 1: (1) they are under the comfortable angles for each type of rotation; (2) initial rotations are larger for roll than for yaw. Finally, we controlled RESOLUTION, i.e. the maximal amplitude participants should not exceed. We chose the following values from pilot studies: 1°, 0.8°, 0.6°, 0.4°.

Task. Users performed a rotation of the head from an initial orientation along a specific direction with an amplitude inferior to a threshold value. A trial was divided into three steps as illustrated in Figure 2 and Figure 3 : (1) participants first rotated their head to an initial orientation by bringing the white vertical bar (handle) inside the corresponding interval (Figure 2 a). (2) Then, they pressed the space bar (Figure 2 b). (3) The system indicated the direction and the maximum amplitude they could rotate the head (Figure 3 a). Participants performed the corresponding movement. They were instructed to do their best to stay within this interval. The trial started with the first rotation of the head captured through the motion capture device and ended either when the movement exceeded the maximum amplitude (failed trial, Figure 3 c) or the head *stopped moving* within the maximum amplitude (Figure 3 b) during 1 second [1]. Any movements reported in the opposite direction to the target canceled the current trial [1] (Figure 3 d). The participant would then repeat the trial until passing or failing it. A change in direction was detected when the amplitude exceeded one count unit.

Count unit. The count unit (*Count*) was estimated from the effective resolution of the head (head noise). This is a main difference with the protocol used in [1]. The reason is that hand noise is very small in comparison to head noise due to the stable position of the mouse on the table. In contrast, head tremor (0.2°) is larger than the resolution of the device (0.05°).

We thus decided to refine the definition of the count unit as:

$$Count = max(R_{device}, R_{eff}^{head}) = max(0.05, 0.2) = 0.2° \quad (3)$$

so that the amplitude of uncontrolled movements when users wanted to remain still would not be inferior to a count unit.

Design. We used a within subject experimental design. ROTATION, DIRECTION and INITIAL ORIENTATION were counterbalanced between participants. RESOLUTION was presented in descending order. In summary, the experimental design was: 12 participants × 2 ROTATIONS × 2 INITIAL ORIENTATIONS × 5 RESOLUTIONS × 10 repetitions = 2400 trials.

5.2 Results

The results are summarized in Figure 4. ANOVA confirmed a significant effect of RESOLUTION on accuracy ($F_{4,190} = 62.5, p < .0001$). However, no other significant effect was found. Results show that participants required at least 5 counts to successfully complete the task 95% of the time. From these results, we estimate the value of the useful resolution $\hat{R}$ for both yaw and roll rotations: $\hat{R} = 5 \times 0.2 = 1°$.

6. STUDY 3: TECHNIQUE COMPARISON

The goal of this study was to compare the impact of the rotation axes on both speed and accuracy on a task involving head-camera coupling. To achieve this, we derived two

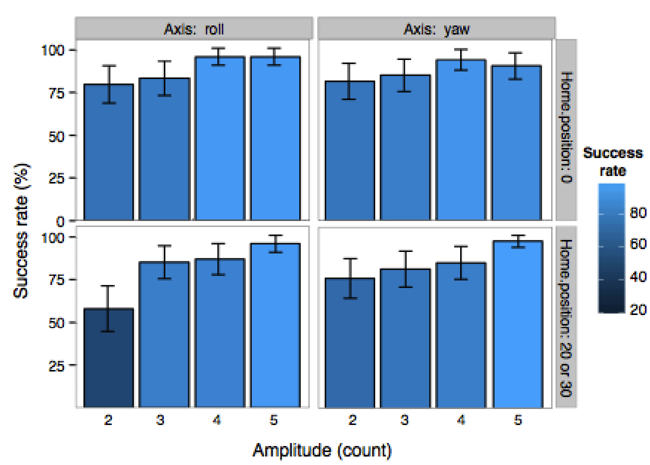


Figure 4: Success rates across all participants in percentage on both axes for both orientations and amplitudes. Amplitudes shown in counts.

interaction techniques.

Interaction Techniques. With both techniques, users perform orbital camera control with head movements while selecting and manipulating objects with the mouse. These techniques only differ in the movement controlling the camera: roll rotation in the first case, and yaw rotations in the second case.

We initially decided to compare our techniques to PRISM [16] because this technique provides an advanced transfer function designed for rotational motion. PRISM dynamically adjusts the control/display gain to provide increased control when moving slowly. However, it can accumulate an offset value representing the angular displacement between the head and the object being manipulated. PRISM provides offset recovery when the user exceeds a certain angular speed. But, pilot studies revealed that this technique does not work well for high gain values because the offset is quite important and it is difficult to predict when the offset recovery will occur. With a gain value of 6 or 7, the camera can turn 12 to 18° per iteration with the consequence that participants get lost in the 3D scene. We thus decided to discard this technique from our study.

6.1 Experimental design

Participants and Apparatus. Twelve participants (4 females) aged 23 to 36 ($\bar{x} = 27.4; \sigma = 3.8$) were recruited via mailing lists and word-of-mouth. They received a handful of candies for participation. 5 of the participants performed the study 1 or the study 2. We used the same apparatus as in the previous experiment.

Task. The task, which is inspired from floor planner tasks, involves the placement of a 3D object (a ball) in a target (a bowl on a shelf) in a 3D scene, as shown on Figure 5. This task combines both a head and mouse pointing task: (a) Participants first perform a pointing task with the mouse to select the ball in front of them; (b) They rotate the view (through roll or yaw head rotations depending of the technique) in the direction of the target, which is indicated by a green arrow on the floor of the 3D scene; (c) Once the head of the participants is aligned with the target, the target is highlighted in green; (d) participants then performs a pointing task with the mouse to align the mouse cursor (that carries the ball) with the bowl; (e) They can then throw the ball inside the bowl by pressing the mouse button and (f) return to the initial position to start the next trial. Feedback indicates whether the shoot is correct or missed. A missed shot can either be due to a misalignment between the camera and the target (head movement error) and/or between the cursor and the target (mouse pointing error).

Technique parameters. In this study, we wanted the participants to be able to perform a full orbital movement in each direction ($A^{cam}_{max} = \pm 180$) in order to cover 360°. Based on the findings of studies 1 and 2, we computed the CD gain for yaw and roll rotations as follows:

$$CDgain(yaw) = \frac{A^{cam}_{max}}{A^{head}_{max}(yaw)} = \frac{180}{26} = 6.9 \tag{4}$$

$$CDgain(roll) = \frac{A^{cam}_{max}}{A^{head}_{max}(roll)} = \frac{180}{35} = 5.1 \tag{5}$$

We could thus compute the theoretical smallest target width for yaw and roll rotations:

$$W^{cam}_{min}(yaw) = CDgain(yaw) \times \hat{R}(yaw) = 6.9 \times 1 = 6.9. \tag{6}$$

$$W^{cam}_{min}(roll) = CDgain(roll) \times \hat{R}(roll) = 5.1 \times 1 = 5.1. \tag{7}$$

Conditions. We controlled three factors in this study: (1) Target Width defines the angular range for which the participant is aligned with the target. We chose four values in the vicinity of the CD gain of our techniques: 4°, 5.1°, 6° and 6.9°; (2) Target Distance: As we wanted to cover $2 \times 180°$, we chose both close and far targets: 28°, 68°, 113° and 158°; (3) Direction: The target was located either on the left or right side of the participant.

Procedure. The experimenter first explained the task. Participants then practiced the two techniques before starting the experiment. During this phase, they were free to change mouse sensitivity to their liking. Participants were generally satisfied with the default settings. They could also reverse the direction of the mapping toward positive or negative angles (i.e. rotating the head to the right would either rotate the camera to the right or to the left). Indeed, we noticed that some participants did not have the same mental model of the scene (11/12 used the default mapping on the roll axis, 8/12 on the yaw). Participants then performed the experiment during 1 hour. We invited them to take a break after each series of 5 trials. At the end of the experiment, participants were asked to answer a NASA TLX questionnaire to assess cognitive load, fatigue, strategies of use and ease of use for the various techniques.

Design. We used a repeated measures-within subject experimental design. Technique was counter-balanced between participants. Width, Distance and Direction were countbalanced between techniques. For each condition, participants performed 5 trials. We measured completion time and success rate. In summary, the experimental design was : 12 participants × 2 Rotations × 2 Directions × 4 target Widths × 4 target angular Distances × 5 repetitions = 3840 trials.

6.2 Results

Accuracy. Repeated measures multi-way ANOVA reveals a significant effect for Rotation on Accuracy ($F_{1,11} = 13.2, p < .001$). A post-hoc Tukey test shows that Roll (92%) is more accurate than Yaw (88%). ANOVA also reveals a significant effect for Width on Accuracy ($F_{3,33} = 10.3, p < .0001$). A post-hoc Tukey test shows that the smallest target (4°: 85%) is significantly less accurate than the other targets (5.1°: 92%; 6°: 91%, 6.9°: 92%). ANOVA also reveals a significant effect for Direction on Accuracy ($F_{1,11} = 7.2, p < .01$). A post-hoc Tukey test shows that Left (91%) is more accurate than Right (88%). No Distance or interaction effects were

Figure 5: The user must 1) pick up a ball on the floor using the mouse cursor; 2) rotate the view (using yaw or roll head movements depending on the condition) in the direction of the target (which is indicated by a green arrow on the floor); 3) Once the cursor is aligned with the target (highlighted in green), shoot the ball in the target by pressing the mouse button.

found on accuracy.

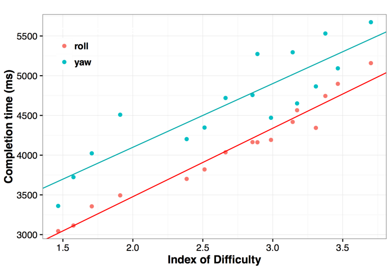
Selection time. ANOVA reveals a significant effect for ROTATION on Selection time ($F_{1,11} = 61.8, p < .0001$). A post-hoc Tukey test shows that Roll (4.0s) is faster than Yaw (4.6s). ANOVA also reveals significant effects for DISTANCE ($F_{3,33} = 59.7, p < .0001$) and WIDTH ($F_{3,33} = 24.1, p < .0001$) on Selection time. Post-hoc Tukey tests confirm that selection time increases with distance and decreases with target width. No DIRECTION or interaction effects was found.

Return time. ANOVA reveals a significant effect for DISTANCE on Return time ($F_{3,33} = 78.3, p < .0001$). A post-hoc Tukey test confirmed the return time increases with distance. ANOVA also reveals a significant effect for DIRECTION ($F_{3,33} = 7.8, p < .01$). A post-hoc Tukey test shows that Left (1.5s) is faster than Right (1.6s). No ROTATION, WIDTH or interaction effect was found on return time.

Qualitative measures. Kruskal-Wallis tests reveal no effect for ROTATION on physical, temporal or mental demand. 8/12 participants preferred Roll and 10 participants found Roll faster than Yaw. 11 Participants reported that Yaw was too sensitive.

6.3 Modeling

We define the index of difficulty of the task, ID, as $\log(1+\frac{D}{W})$ where D is the angular distance and W the angular width. A linear regression gives the following coefficients: T = 1.7 + 0.6 ID for roll ($R^2 = 0.96$) and T = 2.5 + 0.5 ID for yaw ($R^2 = 0.78$). While the model for roll is consistent with Fitts' law [15], it does not seem to be the case for yaw (Figure 6). One possible explanation is that some IDs are too high in the latter case. Another reason might be that, in the case of the yaw axis, the screen shifts out of user view more drastically.



Tech / size	4	5.1	6	6.9
roll left	88	92	95	95
roll right	85	92	93	95
roll total	86	92	94	95
yaw left	88	93	89	90
yaw right	81	90	87	88
yaw total	84	91	88	89

Figure 6: Left: Linear regression model of roll and yaw rotations. Right: Average success rate for each technique, direction and target size.

7. STUDY 4: EXTERNAL VALIDITY

Finally, we performed a last experiment consisting of a 3D manipulating task to see how well our technique fares against a more mainstream approach.

7.1 Experimental design

Participants and Apparatus. Ten participants (2 females) aged 24 to 37 ($\bar{x} = 28.2; \sigma = 4.9$) were recruited. We used the same apparatus as the one used in the previous study.

Task. We chose a standard docking task that requires the user to place a sphere in a specific 3D position (Figure 7). This task is similar to the task of the previous study except that participants must move the ball into the target (Figure 7). The task thus combines *camera control* and *object displacement*.

Users performed orbital *camera control* either with roll head movements or with the mouse or the keyboard. Roll head movements allowed up to 180° of orbital motion in each direction, with the same gain (5.1) as in the previous experiment. Hysteresis was applied to roll movements to prevent jitter and increase stability. We reproduced the mouse and keyboard behavior of the Blender application to orbit the camera: Users had to press (and hold) the wheel button of the mouse to control the camera and the same gain of 2.5 was used. Alternately, users could also use dedicated keys (4 and 6 on the numerical pad). A single key press performed a rotation of 15°. Maintaining a key pressed allowed a continuous camera control with a speed of $360°/s$.

Users could *displace the object* in the camera plane by dragging the mouse (left button). They could also use a 3D helper (Figure 7) with 3 axes and 3 planes, which was attached to the object. This representation is commonly used in 3D applications to constrain the displacements of objects along one (axis) or two (plane) dimensions.

Conditions. We controlled three factors: (1) TECHNIQUE, which was either Head or Mouse/Keyboard, (2) Target DISTANCE, which had three values (68°, 113° and 158°) and DIRECTION, which was either on the left or right side.

Procedure and design. Participants performed several practice rounds for each technique before the experiment and could change angular mapping direction during this stage. Participants performed two sessions. In the first session, participants performed four consecutive blocks with each TECHNIQUE. The order of technique was counterbalanced between participants. Each block contained 6 trials corresponding to 3 DISTANCE × 2 DIRECTION. The order of trials was randomized between blocks and participants.

A second session was performed just after the first one. Participants were now free to choose to use the Head or Mouse/Keyboard *or both* during each trial. We conducted this second session to investigate users preference and whether participants would combine the two techniques. This "free" condition was not performed simultaneously with the previous ones to avoid probable order effect, i.e. participants would have been using these two techniques in a variable number of times depending on their preferences and on presentation order.

Breaks were scheduled between each block, allowing participants to rest or give feedback. At the end of the experiment, they were asked to answer a NASA TLX questionnaire for the three techniques. The experiment lasted 45 minutes

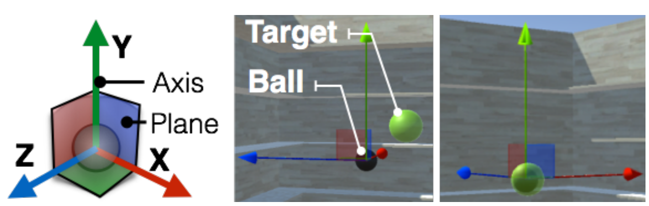


Figure 7: *Left.* The ball can be displaced in the scene by pulling either on its axes or on its planes. *Right.* The participant had to displace the ball into the target using its axes or planes to validate the trial.

where users tested the different conditions. Overall, the experimental design was: 10 participants × 2+1 TECHNIQUES × 4 blocks × 2 DIRECTIONS × 3 target angular DISTANCES = 720 trials.

7.2 Results

We considered three different time measurements in this experiment: (1) Alignment time: the time taken to align the ball with the target (including the time needed to rotate the viewpoint); (2) Return time: the time needed to return to the home position after aligning the ball with the target; (3) Total time: the overall amount of time to complete the trial. Overall, Head Roll is faster than Mouse/Keyboard for Total (14.5%), Alignment (14.7%) and Return (13.4%) time. Repeated measures multi-way ANOVA reveals no effect (or interaction effect) for BLOCK or DIRECTION.

Total time. ANOVA reveals a significant effect for DISTANCE ($F_{2,18} = 20.2, p < 0.001$) and TECHNIQUE ($F_{1,9} = 9, p < 0.01$) on Total time. A post-hoc Tukey test shows that the largest distance (158°: 11.2s) takes more time than the two other distances (68°: 7.8s; 113°: 8.2s). A post-hoc Tukey test also reveals that Head Roll (8.3s) is faster than Mouse/Keyboard (9.8s).

Alignment time. ANOVA reveals a significant effect for DISTANCE on Alignment time ($F_{2,18} = 15.3, p < 0.001$) and Technique ($F_{1,9} = 7.1, p < 0.01$). A post-hoc Tukey test shows the largest distance (158°: 9.6s) takes more time than the two other distances (68°: 6.7s; 113°: 7s). A post-hoc Tukey test also reveals that Head Roll (7.1s) is faster than Mouse/Keyboard (8.4s).

Return time. ANOVA reveals a significant effect for DISTANCE ($F_{2,18} = 36.2, p < 0.001$) and TECHNIQUE ($F_{1,9} = 12, p < 0.001$) on Return time. A post-hoc Tukey test shows the largest distance (158°: 1.6s) takes more time than the two other distances (68°: 1.1s; 113°: 1.2s). A post-hoc Tukey test also reveals that Head Roll (1.2s) is faster than Mouse/Keyboard (1.4s).

Post-Experiment. During the "Free" condition, 8 participants used a technique involving Head Roll: 6 of them used Head Roll+Keyboard and 2 used Head Roll + Mouse. These participants reported that the combination of Head Roll with either mouse or keyboard was faster and more comfortable. Amongst those who did not use Head Roll, 1 used only the mouse and 1 used only the keyboard.

NASA TLX. Kruskall-Wallis tests revealed no effect for TECHNIQUE on cognitive load, fatigue and ease of use.

8. DISCUSSION

8.1 Human Factors

We investigated how to perform orbital camera control with rotational head movements. Study 1&2 helped to tune the transfer function. Study 3 compared techniques using roll or yaw head movements, and study 4 compared head roll interaction and mouse/keyboard interaction.

Head roll is an efficient input modality for head-camera coupling. Yaw might appear more appropriate for orbital camera control because (1) rotations are performed around the same axis for the head and for the camera, and (2) maximum yaw amplitude is larger than for roll. However, this study shows that participants are faster and more accurate with roll head movements and prefer them when interacting with a screen. Indeed, participants reported that yaw rotations are visually uncomfortable because the screen rapidly shifts out of their field of view. The comfortable range of movement for yaw (26°) is smaller than the range on the roll axis (35°). Most participants also reported that the technique using yaw was too sensitive, due to the high CD gain on this axis. This informs us that the range of "comfortable" yaw movement is too small for 360° camera control.

Direction is more intuitive with roll. Head control may introduce confusion about the direction of camera motion. Eleven out of 12 participants used the default mapping for roll but only 8 for yaw. One participant was not sure whether he should turn the head to the left or to the right.

Users are more accurate on the left side. During study 3, participants were slightly more accurate on the left side, possibly because of the *ocular dominance* effect (also called eye preference) [26] which is the tendency to prefer visual input from one eye to the other. A Miles test [26] revealed that 7 out of the 12 participants in Study 3 had a left dominant eye. A post analysis suggests that users are more accurate for the direction corresponding to their dominant eye (91.6%, $\sigma = 1.2$) than the other direction (88.6%, $\sigma = 1.8$). More participants would however be needed to validate this hypothesis.

Head roll motion and/or mouse/keyboard. Study 4 showed that Head Roll is faster than Mouse/Keyboard for Total completion time, Return time and Alignment time. When proposed to combine these techniques in the final "Free" condition, 8/10 participants chose to combine the head with either the mouse (2/10) or the keyboard (6/10). The Free condition was preferred by 80% of the participants and the total time was lower (7s) than for the Head Roll (8.3s) and Mouse/Keyboard conditions(9.8s). Note however that the Free condition cannot be statistically compared to the two other ones because it was performed in a second session for the reasons above explained. Some participants reported that "keyboard is useful for long distance, once Roll movements start being less comfortable" or to be "very fast by performing Head Roll and keyboard/Mouse movements at the same time".

8.2 Interaction technique

Head roll appears a promising modality to control orbital camera motion. However, an effective deployment requires considering additional factors which are task and platform dependent.

Controlling other camera movements. Even though we focus on orbital camera rotation in this article, other types of camera motions such as zooming, panning, other rotations, are needed in actual applications. Our goal was to make interaction faster and more comfortable when performing very common and frequent viewport manipulations in 3D applications. On the one hand, camera controls that are less often used can be performed using classical techniques that rely on the mouse, the keyboard or additional widgets located on the toolbar. On the other hand, head movement can also be used to control other camera motions without hampering the use of the proposed solution. For instance, yaw and pitch could be used for controlling other rotations and translations along the perpendicular axis of the screen for zooming in and out. We envision to evaluate to which extent these different movements can be performed independently in future work.

Activation. Some application might need a mechanism to specify when the technique is activated to avoid accidental camera viewpoint changes. One solution consists of using an explicit delimiter such as a key, but this would require using the hands. Another solution is to trigger the technique when the user performs a head movement larger than a certain amplitude. This value must be small enough to allow novice users to notice it, and large enough to avoid accidental activation. To determine this value, we recorded head movements of 4 participants over one hour in their usual desktop environment. Results revealed an average movement range on each side of $\pm7.1°$ for yaw, $\pm7.5°$ for pitch and only $\pm5°$ for roll. Another solution enables the technique by default when a dedicated application is running (e.g. Floor planning). Interestingly, it is worth noticing that, no participant complained about potential undesired activation of the Head Roll technique in the Free condition. This can be explained by the fact that roll head movements seem pretty robust against undesirable activation. This may be a supplemental advantage of roll head movements over yaw head movements, which are more likely to be performed inadvertently.

9. APPLICATIONS

We developed three applications using the proposed interaction technique to illustrate its principle. While we used the ART motion capture system for the sake of precision in the previous experiments, we implemented the following application with a regular webcam and a commercial face tracking technology (http://visagetechnologies.com).

Hand-free interaction. Rough drawing allow users to give form to an idea and to evaluate it quickly. While easy in 2D applications, hand-free drawing or free manipulation of objects is more complex in 3D applications because it involves drawing or moving an object, which requires to constantly have the hand on the interaction device but also to manipulate the view. We implemented our technique in a room planner which allows users to focus on the drawing task without being disturbed by a physical hand movement to perform viewport manipulation (Figure 8) with the mouse or keyboard.

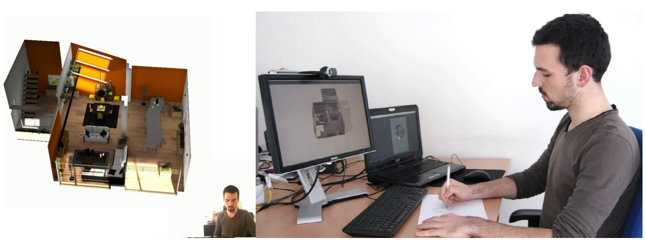


Figure 8: Hand-free interaction. User rotating the 3D scene with the head while drawing on a piece of paper.

3D editor. Moving an object in the 3D space generally requires multiple steps for manipulating the object and the camera. We implemented a script editor as an add-on for Unity, a real-time game engine widely used in the field, to check if our method was suitable on a daily basis by CG artists. The participants of our focus study, long-time users of this software, reported that they had the feeling to displace objects faster, thus be more productive for 3D manipulation tasks. It required them only a few minutes to adjust their workflow.

Immersion. Immersion is becoming preeminent in video games. Common solutions either rely on expensive hardware (*Oculus Rift, HTC Vive*) or are often limited to a Fish Tank effect [2]. To expand our technique within immersive environments, we created a First Person Shooter game. The main camera is overlapping the player's head, thus adding our orbital camera manipulation to basic FPS interactions (based on arrows keys to move the character and the mouse for a Mouse Look effect). Using a slight camera rotation, the player can perform subtle moves to reveal enemies behind obstacles while staying hidden. Our method requires less physical engagement from the player than traditional absolute mapping to achieve similar immersive effect.

These prototypes also show that this technique can be combined with other camera manipulations (scroll to zoom, pan) to provide a wider panel of interaction modalities. Moreover, as computing power is crucial for games and for 3D artists, it is worth noticing that it does not require much computing power, which makes it compatible with cpu-intensive and memory demanding tasks.

10. CONCLUSION AND FUTURE WORK

In this paper we reported how head movements can serve to efficiently change the viewpoint in 3D applications while letting the hands free to manipulate 3D objects with the mouse and the keyboard, a feature that seems especially useful when the viewpoint needs to be temporarily changed, as when moving objects in the 3D scene, or if a quick to-and-fro movement of the camera is needed to disambiguate the view. This research was focused on orbital camera control, the most common way to perform viewport rotations in 3D space. With four users studies, our findings are: 1) the useful resolution of the head is $\pm1°$ for head yaw and roll, 2) head roll provides a wider comfortable amplitude than head yaw, 3) head roll is more efficient and precise than head yaw for 3D orbital control, 4) Head Roll is faster than common mouse or keyboard techniques for orbital camera control and let the hands free for additional input, 5) interaction combining Head Roll and Keyboard is preferred by users. Our findings gives designers of 3D software a solid basis to integrate head-based multi-modal interaction in their software. Future work is required to demonstrate (1) how this technique can be applied in various 3D applications; (2) how it can be applied to other camera controls and (3) what is the impact of the input device (motion capture system vs. webcam) on usability.

11. ACKNOWLEDGMENTS

This research work was supported by the Agence Nationale de la Recherche (ANR 13 CORD 008) and by the EQUIPEX Digiscope ANR-10-EQPX-0026.

12. REFERENCES

[1] J. Aceituno, G. Casiez, and N. Roussel. How low can you go?: Human limits in small unidirectional mouse movements. In *ACM CHI'13*, pages 1383–1386, 2013.

[2] K. W. Arthur, K. S. Booth, and C. Ware. Evaluating 3d task performance for fish tank virtual worlds. *ACM TOIS*, 11(3):239–265, 1993.

[3] R. Balakrishnan and G. Kurtenbach. Exploring bimanual camera control and object manipulation in 3d graphics interfaces. CHI '99, pages 56–62, 1999.

[4] F. Bérard. The Perceptual Window: Head Motion as a new Input Stream. *INTERACT*, pages 1–8, 2014.

[5] F. Bérard, G. Wang, and J. R. Cooperstock. On the limits of the human motor control precision: The search for a device's human resolution. In *INTERACT'11*, pages 107–122. Springer-Verlag, 2011.

[6] D. A. Bowman, E. Kruijff, J. J. LaViola, and I. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., 2004.

[7] BSi. Visual acuity test types - part 1: Test charts for clinical determination of distance visual acuity - specification. BS 4274-1:2003, British Standards Institution, 2003.

[8] G. Casiez and D. Vogel. The effect of spring stiffness and control gain with an elastic rate control pointing device. ACM CHI '08, pages 1709–1718, 2008.

[9] M. Christie, P. Olivier, and J.-M. Normand. Camera control in computer graphics. *Computer Graphics Forum*, 27:2197–2218, 2008.

[10] A. Crossan, M. McGill, S. Brewster, and R. Murray-Smith. Head tilting for interaction in mobile contexts. In *ACM MobileHCI'09*, 2009.

[11] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti. Surround-screen projection-based virtual reality: The design and implementation of the cave. In *SIGGRAPH '93*, pages 135–142. ACM, 1993.

[12] L. D. Cutler, B. Fröhlich, and P. Hanrahan. Two-handed direct manipulation on the responsive workbench. In *ACM I3D '97*, pages 107–114, 1997.

[13] K. M. Fairchild, B. H. Lee, J. Loo, H. Ng, and L. Serra. The heaven and earth virtual reality: Designing applications for novice users. *IEEE VR*, pages 47–53, 1993.

[14] M. Fiorentino, A. E. Uva, M. Dellisanti Fabiano, and G. Monno. Improving bi-manual 3d input in cad modelling by part rotation optimisation. *Comput. Aided Des.*, 42(5):462–470, May 2010.

[15] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6):381, 1954.

[16] S. Frees, G. D. Kessler, and E. Kay. PRISM interaction for enhancing control in immersive virtual environments. *TOCHI*, 14(1), 2007.

[17] A. Fuhrmann, D. Schmalstieg, and M. Gervautz. Strolling Through Cyberspace With Your Hands In Your Pockets: Head Directed Navigation In Virtual Environments, 1998.

[18] E. J. Gibson, J. J. Gibson, O. W. Smith, and H. Flock. Motion parallax as a determinant of perceived depth. *Journal of Experimental Psychology*, pages 40–51, 1959.

[19] T. R. Hansen, E. Eriksson, and A. Lykke-Olesen. Use your head: Exploring face tracking for mobile interaction. In *ACM CHI EA'06*, pages 845–850, 2006.

[20] C. Harrison and A. K. Dey. Lean and zoom: Proximity-aware user interface and content magnification. In *ACM CHI'08*, pages 507–510, 2008.

[21] R. Jagacinski and D. Monk. Fitts' law in two dimensions with hand and head movements. *Journal of Motor Behavior*, 7:77–95, 1985.

[22] D. R. Koller, M. R. Mine, and S. E. Hudson. Head-tracked orbital viewing: an interaction technique for immersive virtual environments. In *ACM UIST '96*, pages 81–82, 1996.

[23] A. Kulshreshth, J. J. LaViola, and Jr. Evaluating performance benefits of head tracking in modern video games. In *SUI '13*, pages 53–60. ACM, 2013.

[24] J. Laviola and M. Katzourin. An Exploration of Non-Isomorphic 3D Rotation in Surround Screen Virtual Environments. In *IEEE 3DUI*, 2007.

[25] E. LoPresti, D. M. Brienza, J. Angelo, L. Gilbertson, and J. Sakai. Neck range of motion and use of computer head controls. In *ACM Assets '00*, pages 121–128, 2000.

[26] W. Miles. Ocular dominance in human adults. *Journal of General Psychology*, 3:412–429, 1930.

[27] J. D. Mulder and R. V. Liere. Enhancing fish tank vr. In *VR '00*, pages 91–. IEEE Computer Society, 2000.

[28] M. Ortega and T. Vincent. Direct drawing on 3d shapes with automated camera control. In *ACM CHI'14*, pages 2047–2050, 2014.

[29] C. Pittman, J. J. LaViola, and Jr. Exploring head tracked head mounted displays for first person robot teleoperation. In *ACM IUI'14*, pages 323–328, 2014.

[30] I. Poupyrev, S. Weghorst, and S. Fels. Non-isomorphic 3D rotational techniques. In *ACM CHI'00*, pages 540–547, 2000.

[31] I. Poupyrev, S. Weghorst, T. Otsuka, and T. Ichikawa. Amplifying spatial rotations in 3D interfaces. 1999.

[32] W. Qi, R. M. Taylor, II, C. G. Healey, and J.-B. Martens. A comparison of immersive hmd, fish tank vr and fish tank with haptics displays for volume visualization. In *APGV '06*, pages 51–58. ACM, 2006.

[33] R. Reilink, G. de Bruin, M. Franken, M. A. Mariani, S. Misra, and S. Stramigioli. Endoscopic camera control by head movements for thoracic surgery. In *BioRob*, pages 510–515. IEEE, 2010.

[34] A. L. Simeone, E. Velloso, J. Alexander, and H. Gellersen. Feet movement in desktop 3d interaction. In *IEEE 3DUI*, pages 71–74, 2014.

[35] M. Spindler, W. Büschel, and R. Dachselt. Use your head: Tangible windows for 3d information spaces in a tabletop environment. In *ACM ITS '12*, pages 245–254, 2012.

[36] R. J. Teather and W. Stuerzlinger. Exaggerated head motions for game viewpoint control. In *Future Play '08*, pages 240–243. ACM, 2008.

[37] G. Thibodeau and K. Patton. *Anatomy and Physiology Third Edition*. Mosby, 1996.

[38] C. Ware, K. Arthur, and K. S. Booth. Fish tank virtual reality. In *INTERACT and CHI '93*, pages 37–42, 1993.

[39] K. Yamaguchi, T. Komuro, and M. Ishikawa. Ptz control with head tracking for video chat. In *CHI EA '09*, pages 3919–3924, 2009.