

Head Tracking in First-Person Games: Interaction using a Web-Camera

Torben Sko and Henry J. Gardner

School of Computer Science, College of Engineering and Computer Science
The Australian National University, Canberra, ACT, 0200, Australia
Torben.Sko@gmail.com, Henry.Gardner@anu.edu.au

Abstract. Recent advances in face-tracking technology have made it possible to recognize head movements using a commodity web-camera. This development has created exciting possibilities for enhancing player enjoyment during computer game play. In order to ascertain the real-world potential for head gestural input to First Person Shooter games, we have developed seven diverse interaction techniques and integrated these with a modern games engine. Evaluation of the techniques was carried out with four focus groups made up of expert games developers and experienced end-users. One of the techniques was further refined and subjected to a follow-up comparison test with promising results. A set of guidelines for the future development of head interaction techniques for computer games has been derived from the studies. All of the techniques have been built upon freely available software and open-sourced to encourage further research in this area.

Keywords: input and interaction technology, face tracking, head tracking, computer game, game engine, first person shooter

1 Introduction

Although real-time head tracking has been an interaction technology for many decades [1], its use as a commodity input mechanism for desktop computing is only just being realized. Thanks in large part to advances in computer vision (for example [2]), head tracking can now be achieved using an off-the-shelf web-camera and face-tracking software (for example [3]).

In light of these advances, HCI researchers have begun to study face-tracked head gestures as input to computer games. In their 2006 paper, Wang et al. studied two basic interaction techniques in two game contexts [4]. They showed that test participants experienced a greater “sense of presence” and satisfaction with their head-tracked techniques. Likewise, in 2008, Yim et al. [5] proposed a head tracking game built upon the popular work of Johnny Lee [6]. Continuing this line of research, we look to more systematically develop a set of interaction techniques, which map out important parts of the design space of head-gestural input to First-Person Shooter

(FPS) games and then evaluate the real-world potential for the techniques using focus groups.

In the following section, we present a simple taxonomy for head-tracked interaction in games and use this to position and describe the collection of seven head-tracked techniques that we have developed. We describe further aspects of the implementation of these techniques, and their integration into a game engine, in Section 3. The conduct and overall observations arising from the focus-group evaluation of these techniques is described in Section 4, which includes a technique-by-technique summary of the results. One of the techniques, singled out for its market potential, was subjected to a follow-up comparison test as described in Section 5. In Section 6, we present some preliminary design guidelines for head-tracked interaction techniques in games. Finally, overall conclusions and future work are presented in Section 7.

2 Interaction Techniques

The complete design space for head-tracked interaction with games is huge. One needs to consider genres [7] (shooters, driving, role playing...), platforms (PC, console...), virtual dimensions (2D, 3D), perspectives (first-person, third-person, top-down...) and players (single, multiple local, multiple remote). As such, it is necessary to consider just a subset of this space. In our work, we chose to specialize to *single-user, first-person-shooter* (FPS) games due to their wide popularity and the natural correlation between the user's head and the first-person-view. We chose to work on the *Windows PC platform* due to its popularity, large software support and the ideal manner (for face tracking) in which users sit front-on to the screen. We also chose to specialize to techniques, which demanded only *light integration with the underlying games engine*, implying that the underlying game logic was only augmented, rather than substantially altered, by use of the new techniques. This last restriction meant that the new techniques should be applicable beyond a particular game scenario and other researchers could more easily emulate their implementation.

In their 2006 paper, Wang et al. [4] describe what appears to have been the first serious research into the design and development of face and head-tracked interaction techniques for computer games. They used a three-level design focus with categories of "presence", "role-playing" and "control" all related to the cognitive processes or activities of the user. They developed two basic interaction techniques: avatar appearance and control in a third person game and dodging-and-peeking (which they referred to as "bullet time") in a FPS game. Their evaluation showed the techniques to enhance end-user experience in various ways. In particular both were shown to significantly increase player presence.

When we came to consider techniques for our own work, we wished to position them in the context of a taxonomy. While the categorization of [4] was interesting, its focus on the effect on the player meant that techniques often overlapped categories. For instance, the dodging-and-peeking technique of [4] could arguably fall under all of their three categories, resulting in a poor classification. Our approach was to take a designer's perspective to derive a taxonomy that regards the effects the techniques

have on the game rather than on the player. We present a simple two-level taxonomy, which categorizes techniques into either “ambient” or “control” techniques. Ambient (or perceptual) techniques have no effect on the state of the virtual world and are designed simply to enhance the visual and/or audio feedback based on the user’s head position. Control techniques are focused on the controlling the state of the game. We can further refine this category into social and non-social (or gameplay) control techniques. Social control techniques are those concerned with communicating with avatars and other players. They are generally more relevant to other game types such as role-playing games and, by default, require a deep integration with the game engine. Given that we are only considering lightly integrated techniques, they have been excluded from this work. Gameplay control techniques pertain to all other forms of in-game control, such as looking and aiming in FPS games. With our new taxonomy the dodging-and-peeking technique of [4] can now be unambiguously described as a gameplay control technique. For the sake of brevity, we refer to “gameplay control” techniques simply as “control” techniques in the rest of this paper.

When considering the category of “control” techniques, we observed that head-gestures are most naturally associated to the concept of looking. Natural movements of the head, and natural *looking* metaphors, motivate controls such as *zooming* (leaning into the screen) and *peering* (leaning and tilting). For these techniques it is natural to map camera position to head position in a continuous manner, which is what we did with our first two interaction techniques: a *continuous zoom* technique and a *peering* technique.

Head movements can also be used as triggers for controls. We studied trigger controls by implementing three techniques: a *spinning* technique which causes a rapid camera rotation (by 180°) in response to flicking the head, a *threshold zooming* technique which activates when the head comes within a preset proximity of the screen, and an *iron sighting* technique. This last technique was inspired by the engaging metaphor of iron sighting (aiming down the barrel of a weapon).

The “ambient” techniques consisted of two techniques: *head-coupled perspective* (HPC) [8,9] which provides a 3D effect by making the desktop display replicate the appearance of a window, and a new ambient technique motivated by the unsteady effect created by a *hand-held camera* (the “Handy-Cam” technique). All of the seven interaction techniques are described in more detail in the following subsections and, with the exception of spinning, are illustrated in Fig. 1. A video of these techniques is also provided online [10].

2.1 Continuous and Threshold Zooming

The two zooming techniques let the user zoom into the scene by moving their head towards the screen. Whilst previous work explored the use of a head-controlled, *staggered* zoom to enhance browsing on a laptop [11], our work chose to explore the idea of *continuous* and *threshold* zooming. For the continuous zooming, the amount of zoom was directly and continuously proportional to the distance between the user’s head and the screen as defined by the following function:

$$\text{cam.FOV} = \text{zoom}_{\max} \cdot f(\text{learnt.depth} - \text{head.depth}, \alpha, \beta)^{\nu} \quad (1)$$

In this equation, `cam.FOV` corresponds the field of view of the virtual camera while `f()` represents a function that normalizes its first argument over the second two, which in this case corresponds to the range $[\alpha, \beta]$. The constants parameterized the shape of the mapping. In our default implementation, $\alpha = 0.05\text{m}$, $\beta = 0.3\text{m}$, $\text{zoom}_{\text{max}} = 40^\circ$, $\nu = 2.0$ (meaning the scaling was parabolic rather than linear) and `learnt.depth` initially starts at `0.7m`, implying that `30cm` forward lean would result in a 40° reduction in the FOV. A full implementation would also induce some form of disadvantage, such accuracy reduction, to ensure the use of the technique was fair.

The threshold zooming induced a full zoom (zoom_{max}) when the user's head crossed a proximity threshold. If described using the same function as before, then $\alpha = \beta$, where both represent the threshold point. Worth noting was that users were unable to shoot whilst using the threshold zoom.

2.2 Spinning

The spinning technique allows the user to perform a rotation of the camera – defaulted to 180° – in response to a rapid flick of their head beyond an angular threshold. A subsequent spin could only be performed once the user had re-centered their head. This technique was designed for when the player was attacked from behind and appears to be new in the literature.

2.3 Peering

The peering technique lets the player look around objects in the virtual world by physically leaning their head to either side, following a similar concept implemented by Wang et al. [4] for dodging bullets. Unlike their technique, which only considers the *horizontal offset* of the user's head, we also considered the *roll*. The following formula describes how our version of peering-to-the-right functions (a similar equation would be employed for peering-to-the-left):

$$p^+ = f(f(\text{head.roll}, \alpha, \beta) + f(\text{head.off}, \gamma, \delta), 0, 1)^\nu \quad (2)$$

$$\text{cam.sideways} += \text{check_collisions}(p^+. \text{peer.sideways}) \quad (3)$$

$$\text{cam.height} += p^+. \text{peer.lower} \quad (4)$$

$$\text{cam.roll} += p^+. \text{peer.roll} \quad (5)$$

In these equations, α and β represent the minimum and maximum amount of required head roll (radians) respectively, while γ and δ represent the minimum and maximum amount of required sideways head movement (meters) respectively. As with before, `f()` is a normalizing function. In our default implementation, $\alpha = 0.15$ (rad), $\beta = 0.3$ (rad), $\gamma = 0.05\text{m}$, $\delta = 0.1\text{m}$, $\nu = 2.0$, `peer.lower` = 10, `peer.sideways` = 30, `peer.roll` = 15° . To avoid allowing the player to peer through the walls, we needed to ensure that we checked for collisions. A complete implementation would also alter the player's hit boxes and aiming position in an effort to maintain game balance.

2.4 Iron Sighting

The iron sighting technique allows the user to aim down the barrel of their weapon by tilting their head to the right, similar to how they would iron sight in real life. As the user rolls their head past a threshold, the virtual weapon is raised up and a slight zoom occurs. To enhance this effect, the sniper rifle was modified to feature a real-time scope, allowing the user to realistically target a far off enemy using the actual weapon model, rather just using a decreased FOV. As with other games that feature iron sighting, a full implementation would also induce an increase in accuracy and a reduction in player speed, to ensure the technique was balanced. This technique appears to be new in the literature.

2.5 Head-Coupled Perspective

The first of the two ambient techniques, Head-Coupled Perspective (HCP), is a classic enabling technology for fish-tank virtual reality [8,9] and has recently been publicized by the work of Johnny Lee [6] and has been subsequently used by Yim et al. [5] to develop a “Dodge It” game. Based on the user’s head position, the parallax of the 3D scene is altered by varying the virtual camera’s field of view, offset, orientation and vanishing point. Given that we have followed a standard implementation, the equations for this technique have been excluded.

2.6 Handy-cam

The handy-cam technique adds natural movement to the view in an effort to replicate the unsteady appearance of movies filmed using a hand-held camera. This was achieved by coupling the rotation (and upon the request of one participant, the offset) of the player’s head to those of the virtual camera using equations of the form:

$$\text{Rotation: } \left[\begin{array}{l} \text{cam.yaw} += \alpha (\text{head.yaw}^\nu) \quad (6) \\ \text{cam.pitch} -= \alpha ((\text{head.pitch} - \text{learnt.pitch})^\nu) \quad (7) \end{array} \right.$$

$$\text{Offsets: } \left[\begin{array}{l} \text{cam.height} += h \cdot \text{head.height} \quad (8) \end{array} \right.$$

$$\text{cam.sideways} += s \cdot \text{head.sideways} \quad (9)$$

$$\text{cam.depth} -= d (\text{head.depth} - \text{learnt.depth}) \quad (10)$$

Where $\alpha = 40^\circ$, $\nu = 1.1$, $h = 10$, $s = 10$, $d = 20$, `learnt.pitch` initially equals 0.15 (rad), and, as with before, `learnt.depth` initially starts at 0.7m. Although existing FPS games sometimes simulate a hand-held camera motion to represent player movement through a scene, our handy-cam technique also works when the player is stationary. Furthermore, it imparts a subtle and responsive movement to the scene. It has not been previously reported in the literature.

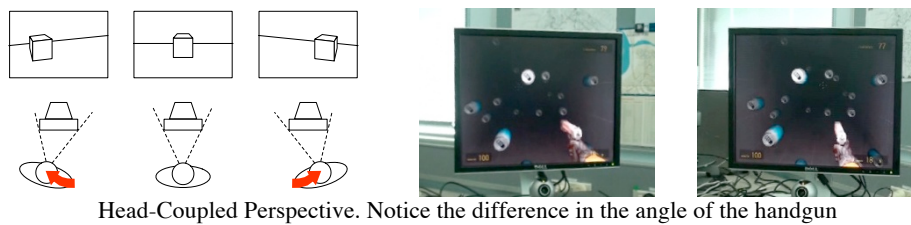
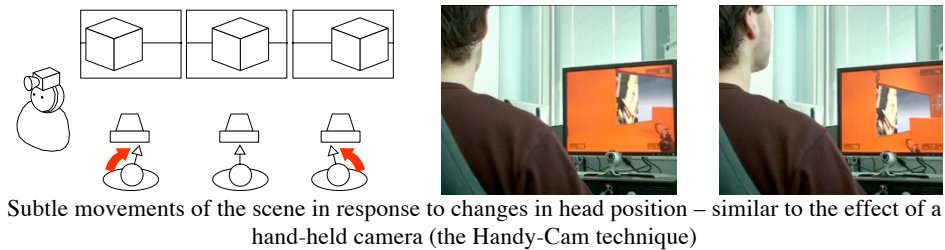
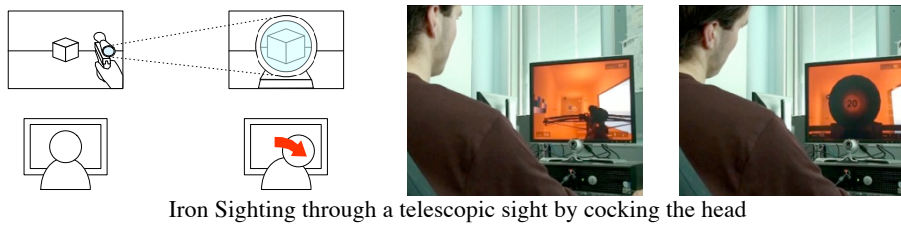
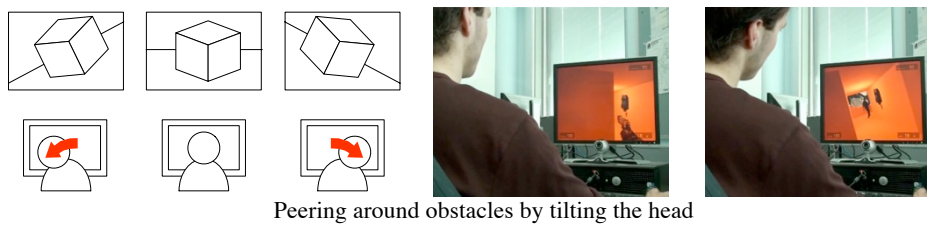
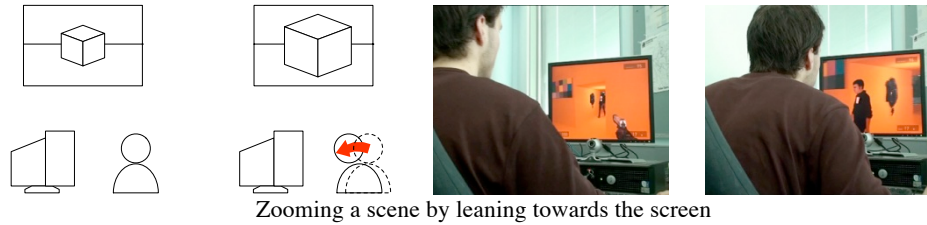


Fig. 1. Illustration of the techniques discussed in the paper, excluding the spinning technique. Head and body movements have been exaggerated for the sake of demonstration.

3 Implementation

The head tracking was performed using a standard web camera and v2.1 of Seeing Machine’s FaceAPI [3], which tracks the head with 6 degrees of freedom without the use of markers. This avoids the need to have the user wear specialized equipment (which, as Yim et al. [5] mention, could deter users). This software was integrated into Valve’s Source [12] game engine via the use of client code and library calls. The overall system is shown in Fig. 2 and is freely available online [13]. The integration process required the game engine to be modified in a number of ways:

Firstly, some of the techniques required functionality not already present in the engine including peering, spinning and iron sighting. Additionally, the rifle’s scope was modified to show a real-time magnification of the view for use with iron sighting. Advice on these modifications was obtained from Valve’s support resources [14, 15].

Secondly, a layer of supporting logic was developed for the techniques. This included smoothing the head data over a specified number of data frames, learning the neutral position of the head, gracefully resetting to the neutral position following a tracking loss, resetting the tracking when the system suspected it to be inaccurate, and correcting for an arcing effect when moving the head directly towards the screen.

Finally, to get the most from the focus group evaluation, it was important that the techniques were of a near-commercial quality. For example, most techniques featured an ease-in (v) to avoid abrupt movements and the crosshair’s position was made to be dynamic to ensure that it remained accurate in spite of moving the camera. Likewise for the iron-sighting technique, the modified rifle was refined until it was at a high level of aesthetic quality.

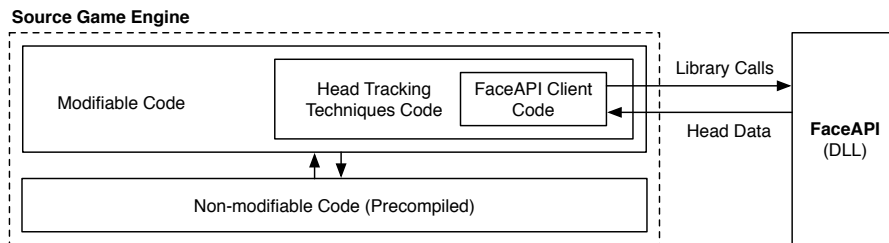


Fig. 2. System diagram for the implementation of the head-tracked interaction techniques in the Source game engine

4 Focus Group Evaluation

The primary usability attribute of a computer game is that of *fun*. However, the mere concept of head-gestural input to an FPS game is a *fun* idea in itself. We therefore expected that the novelty value of our new techniques would be high but we wished to look beyond their immediate reception to get feedback about their potential for commercial FPS games in the near future. We chose a focus group approach with a double-layer design [16] to explicitly compare and contrast the views of two classes

of *experts*: games developers and experienced end-users. This form of analysis was chosen due to the freedom it provided. Not only were we keen to collect feedback, we also wanted to gather the participant's suggestions and ideas.

A set of four focus groups were conducted: two with developers and two with end users with 15 participants in total. The two industry focus-groups included staff from two games companies of international reach. One of these focus groups also included staff from the company which produced the face-tracking software. Occupations included lead designer, lead software engineer, technical developer, marketing communications officer and a game producer. The end-user groups were conducted with university students all of whom had considerable experience with FPS games. Prior to the focus group interview, each participant completed informal tasks (such as exploration or combat) with each technique separately. This was followed by using three of the techniques simultaneously, namely handy cam, peering and zooming. The platform used was a 2.13GHz machine, with 2.00GB of memory and a nVidia GeForce 7600 GS graphics card. The video signal was provided by a USB Logitech QuickCam Pro 4000, placed directly under a 17" monitor as shown in Figure 1. The familiarization phase was conducted with all members of the group being present and the ensuing discussion was recorded and combined with the interview process.

The focus group interview was conducted away from the computer and was driven by a set of guiding questions. These questions started out by asking about the background of the participants including their experience and preferences. Questions about the techniques started at a general level (What is your impression of each of the techniques?) and then moved to cover specific issues such as the commercial viability of head tracking (Do you see head tracking as being something that may appear in games in the future?), the viability the techniques (Could you see yourself continually using any of these techniques or are they just a novelty?), and suggestions as to how they might be improved (Do you think any of the techniques could be improved?). Between one and a half and two hours were needed for each focus group (with 4 participants in most of the groups) and the study was conducted over a week.

Each of the authors conducted an independent analysis of the abridged focus group transcripts [16], which comprised a total of 144 pages of single-spaced text. General observations derived from these analyses were then pooled and are summarized in the next subsection. This is then followed by subsection 4.2, which provides an individual critiquing of all the techniques. In keeping with advice from the focus group literature [16] we report qualitative trends rather than quoting statistics.

4.1 General Observations

Considerable excitement was evinced in all of the focus groups, which confirmed our caution about the novelty effect of head-mounted interaction. One of the industry participants stated, "You've done a good job. I'm really impressed" while an end-user stated, "...when I first came into the room, I found that you actually control by your head, I was really impressed". As such, this initial reaction justified and the strong need for follow-up probing about their real-world potential as a commercial product.

The general excitement did not mean that group participants felt that head-mounted interaction had an immediate future with FPS games. To the contrary, a large amount of discussion time considered other genres of games which appeared to be more suited to the techniques – stealth games (much like “Thief” [17]) were particularly mentioned as well on-rails shooters and games which include the visible control of avatars.

Many participants reflected that they would not use any of the techniques for fast paced or competitive gameplay, as echoed by the statement of one end user who said, “when it’s quick in competition, it’s all buttons”.

The reliability of the head-tracking software was a particular concern when participants were first getting used to the system. The version of the FaceAPI being used had occasional problems with certain facial characteristics such as beards and darker skin tones. In many cases the system tuned itself over time but there were still occasional drop-outs. It was pointed out in discussion that there is little room for tolerance for tracking drop-outs in a commercial product – particularly if the technique is to be used for control. In such cases, users will turn head-tracking off and use the keyboard alternatives. It is claimed that the reliability issue has been addressed in a more recent release of FaceAPI [3]. There was only one direct mention of latency as an issue of the system and this was with respect to the implementation of the Head-Coupled Perspective technique.

4.2 Technique Critiques

Although several participants provided some form of preferential ordering for the techniques, drawing out a clear ordering across all groups of the focus study was difficult. As, such, the following discusses the merits, criticisms and the potential of each technique separately, as reflected across all participants.

Peering. This technique clearly received the greatest amount of positive feedback of all of the control techniques, with several end users naming it as their favorite technique. Suggested improvements for the technique included providing a finer level of control, as well as allowing the player to shoot from the peering position. Of all the techniques, it was felt that peering could benefit the most from focusing the game content around its use.

Iron sighting. This technique caused the greatest split among participants. While one participant with a firearms background showed great fondness for the technique, most others expressed reservations about it. These comments included that it induced neck pain, that it placed the head in an awkward playing position, that a mouse click would be taken in preference to a head gesture, and finally that peering, which shares a similar action, would be chosen in preference. Some attributed these issues to the frequency, urgency, duration and extent of the action required. We feel that introducing some feedback could help alleviate these issues, given that most participants tilted their heads further than was required. Some participants suggested that closing one eye may be a more appropriate gesture.

Zooming. This technique generally received less attention than the others, but on the whole, the feedback for both zooming techniques was positive. Of the two, the continuous zooming was preferred, as it provided a greater level of control, although many participants felt the movement range was too large and that it lacked feedback regarding the end of this range. To improve the technique, some users suggested adding a depth of field and making the zoom length dependent on the gun.

Spinning. This technique, as a concept, received a large amount of positive feedback, although only a few participants noted that they would use a head gesture in preference to a keystroke. Many reflected that they felt the technique would be better suited to games consoles, where there is a greater tradeoff between the speed required for turning verses that for aiming. Many also suggested that the spin be 90° as this would be more natural and less disorientating. One participant voiced concerns about the effect of false positives, as these would significantly disadvantage the player.

Head-Coupled Perspective. This technique generally received an enthusiastic response. However, as the technique requires a low latency, accurate tracking and quite some tuning to accurately create the 3D effect, it places unrealistic expectations on the tracking system. As such, some participants reflected that they felt the technique to be ineffective, and one participant stated they would discard it altogether. Despite failing to create a real sense of depth, many participants still liked the way that it injected life into the scene, and some suggested that it would be better suited to either special moments during a first person game, such as peering through a virtual window, or for use in viewing the map in a top-down, strategy game.

Handy-cam. In spite of its subtle nature, this technique received only positive feedback from all participants, with several stating that it added *life* to the game and made it *more realistic*. One industry representative was particularly receptive to the technique and felt that it had market potential. Unlike the similar HCP technique, the handy cam technique places fewer expectations on the system, catering for a small level of latency and inaccuracy. As reported in the next section, only 2 of the 8 participants of the follow up study noted of any issues to do with the response of the technique. It was felt that this technique could be improved by making it work with relative, rather than absolute, head positions. In contrast, the level of integration was not raised as an issue, boding well in regards to the ease in which this technique could be integrated into other games.

Combining Techniques. Where requested, the handy-cam, peering and continuous zooming techniques were simultaneously activated. Whilst most participants noted that this combination of techniques was overwhelming, they enjoyed the increased freedom that this combined technique introduced to the game. Some felt they could become acclimatized to the combined technique in event they were provided with more time with a better-tuned version of it.

5 Follow-up Test of the Handy Cam Technique

Given the large amount of positive feedback the handy-cam technique received, we felt it warranted further testing to see whether players indicated a preference for having this ambient technique switched on during play. We conducted a brief comparison test using our modified game engine and some early levels from the game Half-Life 2 [18]. For this study, the handy-cam technique was modified to include roll, with the following replacing Eqns. 6 and 7 from Sec. 2.6:

$$\text{cam.roll} -= r (\text{head.roll}^y) \quad (11)$$

$$\text{cam.yaw} += y (\text{head.yaw}^y) \quad (12)$$

$$\text{cam.pitch} -= p ((\text{head.pitch} - \text{learnt.pitch})^y) \quad (13)$$

In this tuned version $r = 65^\circ$, $y = 90^\circ$, $p = 150^\circ$, $w = 100$, $h = 50$, $d = 75$, learnt.pitch initially equals 0.15 (rad), and, learnt.depth initially equals at 0.7m. From the variables it can be seen that this version is much more accentuated than the version used in the focus group study.

Eight new end-user participants were asked to spend a total of 20 minutes playing the game for order-balanced conditions with and without the handy-cam switched on. They were then asked four questions using Likert scales between one and five: Which condition did they prefer? How immersed in the game experience did they feel? How realistic was the experience? How aware were they of the ambient head-tracking technique being in operation? The median rating for the preference question was 4, showing a slight preference for having the handy cam on, and the median rating for awareness was 2, showing a low awareness of the head tracking system. Taken together, these two results argue well for the use of the handy-cam as an unobtrusive add-on for FPS games. We note however, that one outlier participant greatly preferred playing with the handy cam technique off, suggesting that this variant of the technique would still benefit from being an optional component. Additionally, some tracking issues occurred. In particular, the system failed to properly recognize one of the participants, implying the need to further refine the technique to make it more robust. The immersion and realism questions showed no significant difference between the two conditions.

6 Derived Design Guidelines

Despite being somewhat preliminary work, the four focus groups and follow up testing continually raised several important issues to do with the design of head-tracked interaction techniques. We have abstracted these observations, and combined them with our own experience in developing and testing these techniques to derive draft guidelines for others wishing to develop commercially viable (i.e. based on hardware that is affordable, intuitive, unimposing, etc), head-tracking techniques for use with computer games:

Cater for a lack of reliability. Unlike a keyboard and mouse, commercial head tracking will arguably always suffer from some issues of reliability - be it latency, inaccuracy or simply dropping out altogether. As such, one should avoid using head tracking for performing critical control functions, given that the game would either need to be paused or feedback would need to be provided, in the event the system becomes unreliable. Such occurrences might detract from the user experience if they occur too frequently. A preferred approach is to use the head tracking to supplement the experience so that gameplay could continue without significant impact following a tracking loss.

Help the tracking system. Given the issues of reliability, avoid inducing movements that the head tracking system may find harder to accurately interpret. From our experience with the FaceAPI, avoid large rotations, rapid movements, or requiring movements near the edge of the usage region. For the latter point, we suggest softening out an effect so the outer regions have less impact than the center, making the user more inclined to stay within a reliable usage region.

Make it natural. A technique should try to cater towards a natural affordance or action, allowing for them to be performed either unwittingly or intuitively. In the latter case, one must then consider what is considered intuitive for *most* people.

Do not strain the user. A technique should avoid placing a player under any form of physical duress. In particular, avoid making the player continuously hold awkward poses or perform quick, snapping actions. Developers must be responsible for the safety of their users, especially if a game is to be released to market.

Emphasize the continuous nature of the head data. A great advantage of the head data is that it provides continuous information, which allows the player to conduct partial movements. Given the difficulty in replicating this using buttons, head tracking can provide a finer level of control than a keyboard or mouse click.

Induce and reward the player. Thought should be given in focusing the game content around the techniques, particularly for control techniques, to induce the player into using them. If a technique requires something of the player, ensure that their efforts are rewarded. A player will cease to use something if it gives them no benefit. Worth noting, is that guideline must be considered in regards to the first guideline.

Make it general. In developing a technique, ensure that its implementation does not discriminate against particular users. For example, some users sit back, some slouch and others may be just be interpreted oddly by the head tracking. As such, avoid *pre-set* options. Either allow the user to set where they wish to, say, invoke an action, or preferably, refine and develop the technique until it works uniformly across all user types. Part of this will probably require making the technique work on *relative* movements, rather than *absolute* ones.

Be unobtrusive. A successful technique cannot expect too much of the user. Techniques that require a setup process, continual resetting, or any other form of inconvenience will deter users.

Feedback. In an effort to train the player in using a technique, feedback is very important to make players aware of the limits. For example, letting the user know how close they are to the edge of the working camera range, or how close they are to fully completing or activating an action will avoid them moving further than required. Auditory feedback is particularly good as it avoids cluttering the screen, however, elegant visual feedback could also be effective.

Unintentional movements. Unlike other forms of input, head tracking is susceptible to unintentional input. Sources of these may include external distractions, causing the player to turn their head away from the screen, and uncontrollable physiological responses, such as jumping during a scary, in-game moment. Techniques that do not reverse the actions of an unintentional movement upon returning to a neutral head position may cause the player frustration.

7 Conclusion

We have continued work initiated by others to more systematically explore the design space of head-tracked interaction with computer games. We have developed five interaction techniques for control and two for ambient interaction.

While, on the whole, the feedback received from the focus groups was positive, we conclude that the control techniques (peering, zooming, iron sighting and spinning) are most useful for games with specially designed, head-centric content or as an optional control for more mainstream games. This is primarily because of the high performance demands that gamers place on control devices, which, unfortunately, commercial head tracking systems do not seem to meet at this time.

By contrast, we see a greater potential for the ambient techniques in the FPS game genre, given the energy and realism they bring to the game. Given that these techniques provide nonessential functionality (at least in the form we propose) they are better at catering for the unreliable nature of the head tracking system. Unfortunately though, the HCP technique still places unreasonable demands on the latency and configuration of the system. In contrast, the handy-cam technique, a technique first proposed in this paper, does not place these requirements on the system but still achieves a similar effect. Follow up testing showed that the majority of users preferred playing with the handy-cam technique switched on. For this reason, we believe this technique has the potential for the greatest immediate, commercial impact of all of the techniques studied here.

In the future, we hope to conduct more substantial and quantitative analysis of the handy-cam technique. We also wish to study the natural movement of players during gameplay as this will aid in utilizing the head-tracking data in future work.

Finally, to encourage further research in this field of work, the techniques discussed in this paper have been open-sourced and can be downloaded from [13].

Acknowledgements. The authors gratefully acknowledge the donation of v2.1 of the FaceAPI from the Seeing Machines company for the purposes of this study. No commercial funding was received for this research.

8 References

1. Sutherland, I.E.: A Head-Mounted Three-Dimensional Display. In: Proceedings of the December 9-11, 1968, fall joint computer conference, part I, pp. 757--764. ACM, New York (1968)
2. Viola, P., Jones, M.T.: Robust Real-Time Face Detection. *J. Computer Vision* 57 (2), pp 137--154 (2004)
3. Face-API, <http://www.faceapi.com>
4. Wang, S., Ziong, X., Zu, Y., Wang, C., Zhang, W., Dai, X., Zhang, D.: Face-Tracking as an Augmented Input in Video Games: Enhancing Presence, Role-Playing and Control. In: Proceedings of the SIGCHI conference on Human Factors in computing systems, pp. 1097--1106. ACM, New York (2006)
5. Yim, J. Qiu, E. Graham, T.C.N.: Experience in the Design and Development of a Game Based on Head-Tracking Input. In: Proceedings of the 2008 Conference on Future Play: Research, Play, Share, pp. 236--239. ACM, New York (2008)
6. Lee, J.C.: Hacking the Nintendo Wii Remote. *J. IEEE Pervasive Computing* 7 (3), pp. 39--45. IEEE Educational Activities Department, Piscataway (2008)
7. Crawford, C.: *The Art of Computer Game Design*. McGraw-Hill/Osborne Media, Berkeley (1984)
8. Arthur, K.W., Booth, K.S., Ware, C.: Evaluating 3D Task Performance for Fish Tank Virtual Worlds. *J. ACM Transactions on Information Systems (TOIS)* 11 (3), pp. 239--265. ACM, New York (1993)
9. Reikimoto, J.: A Vision-Based Head Tracker for Fish Tank Virtual Reality – VR without Head Gear. In: Proceedings of the Virtual Reality Annual International Symposium (VRAIS'95), pp. 94--100. IEEE Computer Society, Washington(1995)
10. Video Demonstration of the Techniques, <http://www.youtube.com/watch?v=qWkpdtFZoBE>
11. Harrison, C., Dey, A.K.: Lean and Zoom: Proximity-Aware User Interface and Content Magnification. In: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, pp. 507--510. ACM, New York (2008)
12. Valve Corporation, <http://www.valvesoftware.com>
13. Software Download Page, http://www.torbensko.com/software/head_tracking
14. Source Engine SDK Developer Wiki, http://developer.valvesoftware.com/wiki/SDK_Docs
15. Valve Support Forums, <http://store.steampowered.com/forums>
16. Krueger, R.A., Casey, M.A.: *Focus Groups: A Practical Guide for Applied Research*. Sage Publications Inc. (2000)
17. Thief Product Information, <http://www.eidosinteractive.co.uk/games/info.html?gmid=148>
18. Half-Life 2 Information Page. <http://orange.half-life2.com/hl2.html>