AirMouse: Finger Gesture for 2D and 3D Interaction

Michael Ortega[1] , Laurence Nigay[2]

[1] PRIMA - INRIA Rhône-Alpes; michael.ortega@inria.fr
[2] IIHM - University of Grenoble, CNRS, LIG; laurence.nigay@imag.fr

Abstract. This paper presents AirMouse, a new interaction technique based on finger gestures above the laptop's keyboard. At a reasonably low cost, the technique can replace the traditional methods for pointing in two or three dimensions. Moreover, the device-switching time is reduced and no additional surface than the one for the laptop is needed. In a 2D pointing evaluation, a vision-based implementation of the technique is compared with commonly used devices. The same implementation is also compared with the two most commonly used 3D pointing devices. The two user experiments show the benefits of the polyvalent technique: it is easy to learn, intuitive and efficient by providing good performance. In particular, our conducted experiment shows that performance with AirMouse is promising in comparison with a touchpad and with dedicated 3D pointing devices. It shows that AirMouse offers better performance as compared to FlowMouse, a previous solution using fingers above the keyboard.

Keywords: AirMouse, interaction, 2D/3D pointing, computer vision, Fitts' law.

1 Introduction

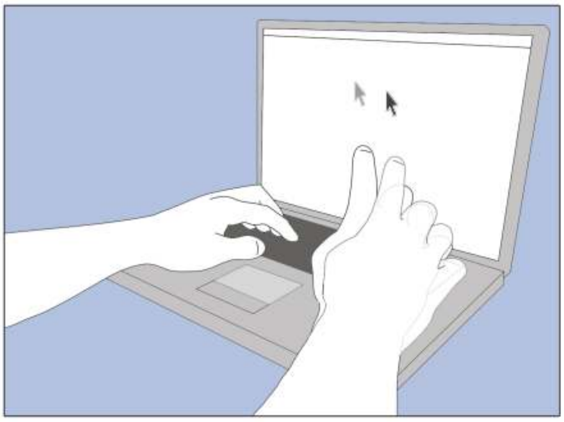

Fig. 1 AirMouse is an interaction technique, which consists of moving fingers in 3D above the keyboard for directly pointing and manipulating objects on screen.

Interaction devices such as a mouse require an additional surface to operate on. Laptops are widely used today, and additional space is an issue in a context where space is at a premium as is the case for example of the table in a train. More generally, the cumbersome problem is one of the «grand challenge» questions described by Bowman et al. [2]; they argue that being cumbersome "has a huge impact on the usability of the systems". Touchpads and key-joysticks are solutions for this problem, however the pointing performances are low and they are not efficient for 3D interaction.

For 3D interaction, the current existing devices are bulky and expensive. Previous works, like [22, 17, 28], proposed solutions using a finger above the keyboard. However, the solutions imply arm tiredness and cognitive load due to transformation between the manipulation space and the display space. Moreover, quoting the authors of Flowmouse [28]: "pointing performance with FlowMouse was significantly worse than with a trackpad".

Facing these issues, we present AirMouse for 2D and 3D interaction that is based on finger gestures performed above the keyboard. AirMouse is a mix of efficient pointing techniques (namely Ray-casting and Virtual-Hand techniques [2]) adapted for 2D and 3D pointing. The key features of AirMouse include:

No additional surface: The additional surface, beside the laptop, is suppressed.

2D/3D interaction: The technique supports mixed 2D and 3D interaction in the same application. This feature is important since 2D and 3D views are more and more common and their combinations valuable as explained in [26]. With AirMouse no additional time is needed for the user to move her/his hand from one input 2D device to another 3D one.

Reduced homing time: The switching time between the keyboard and the pointing device is drastically reduced. This feature is particularly important for limited-mobility users.

Easy to learn and easy to use: Based on a natural and direct way of pointing and manipulating, the concept is easily adopted by new users.

Reasonably low cost: The technique can be implemented at reasonably low cost.

Low tiredness: Compared to similar existing methods [22,28,17], AirMouse does not force the user to move his forearm. The hand palm can be left beside the keyboard, and only forefinger movements are needed. Other fingers can stay on the keyboard.

Low cognitive load: No rotation between the manipulation space and the display space decreases the cognitive load implied by previous methods [14].

Good performance for 2D and 3D pointing: The performance of the AirMouse is better than the FlowMouse technique and is promising in comparison with a touchpad and with dedicated 3D pointing devices.

This paper presents a low-cost vision-based implementation of the AirMouse technique in order to validate the technique itself. A final product could be industrially implemented on all the laptops with an unobtrusive and no bulky system of cameras integrated in the corners of the laptop screens. Indeed our technique on a laptop is promising for 3D games or professional 3D applications (architecture, interior design applications used by sellers at clients home) and may foster even more 3D applications on laptops.

The paper is organized as follows: we first discuss related work before explaining the AirMouse technique and its vision-based implementation. We then present a formal evaluation and its results of the two implemented pointing techniques (2D and 3D) that we compare with more traditional 2D and 3D pointing devices.

2 RELATED WORK

On the one hand, the mouse is the most commonly used device for desktop applications, for experts and occasional users. The device is easy-to-learn, low cost, and, compared to other desktop pointing devices like trackballs, touchpads or key-joysticks, it offers the best performance for time completion of pointing tasks [16, 7]. On the other hand, it is natural to designate an object using the forefinger which is routinely performed in everyday life. Towards naturalness and intuitiveness of the interaction, several studies therefore focused on using a finger for showing, selecting and manipulating objects displayed on screen instead of the mouse [24].

With multi-touch interactive surfaces as in [10] the mouse is replaced by the forefinger. Multi-touch interactive surfaces define a very dynamic research area. When focusing on large surfaces such as a table, such setting cannot be used in everyday work environment. Moreover, although interaction with multi-touch interactive surfaces is natural and intuitive, there are also identified limitations. One of the main issues is the tiring effect of lifting and moving the arm between the different points of the surface. Moreover such movements are not possible for limited-mobility users.

In our work and as opposed to multi-touch interactive surfaces, we focus on fingerbased interaction for a laptop setting including a traditional keyboard. Previous studies have been conducted on using fingers while keeping the keyboard for interaction. First, in [3], the mouse is replaced by a joystick placed in the middle of the keyboard. The keyboard and joystick combination reduces the homing time in comparison with the keyboard and mouse combination. However, the performance and the usability of the joystick are far from the mouse capacities [7]. A different approach is presented in [19] and [22] with the FingerMouse: this freehand pointing technique is based on computer vision for tracking a fingertip. The screen cursor moves according to the user gestures in a horizontal plane just above the keyboard. The selection, equivalent to a mouse click, is performed by pressing the SHIFT key. More recently, another computer vision-based pointing gesture technique, namely the FlowMouse, has been proposed in [29]. FlowMouse uses one camera, and detects the complete hand 2D horizontal movements (using optical flow cues) above the keyboard. Tha hand translates and then force the user to move its arm. The technique has been experimentally evaluated: a Fitts' law study demonstrated that while pointing performance was worse than a touchpad, the interaction was intuitive, easy to learn and easy to use. Finally the Visual Touchpad [17] is a mixed technique that combines the "Finger- Mouse" technique, 2D hand gesture techniques and a virtual keyboard. Two cameras are used to detect if the fingertip is on or above the virtual keyboard. The system could be considered as a low-cost tabletop display or touch-screen, but with a dissociation of the horizontal tracked surface (i.e., a quadrangle surface replacing the keyboard) from the vertical computer screen.

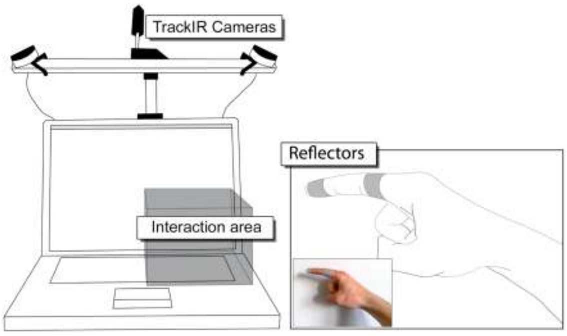

Fig. 2 One set-up for the AirMouse technique. Two trackIR devices are placed on top of the laptop in order to provide a large 3D interaction space.

The aforementioned techniques do not require additional space to operate on, which is an important issue for laptops used in various contexts. They also reduce the homing time which is responsible for 42% of the time required to move the hand from the keyboard to the mouse, point, and go back to the keyboard [4]. However these techniques have two main limitations. First, the forearm has to move above the keyboard which is tiring. Letting the palm beside the keyboard, and only moving the fingers should be less tiring. Secondly, the transformation, here a rotation, between the plane of the finger gesture and the one of the cursor movements increases the cognitive load, which decreases performances and increases tiredness of the user [14].
These techniques aim at replacing the mouse. These techniques as well as the mouse are not adapted for 3D pointing or manipulation, due to their lack of dimensions. Additional modifier keys are then required for 3D interaction. Specific devices exist for manipulating objects in three dimensions, like PHANTOMs [18] and the spacemouse [5]. Most of them are expensive, bulky and also involve switching time when changing from using the 3D pointing device to the keyboard. Moreover, the 3D pointing device being next to the keyboard, the action workspace defined by the position of the device is deported from the screen that defines the virtual workspace. A large translation between the action and virtual workspaces decreases the interaction performances [20].

The AirMouse technique, for which an implementation is presented in the next section, extends the FingerMouse and FlowMouse possibilities by considering 3D finger gestures. AirMouse therefore supports both 2D and 3D pointing.

3 TECHNIQUE OVERVIEW AND IMPLEMENTATION

The AirMouse technique consists of using fingers over the keyboard for interacting in two and/or three dimensions with desktop applications. The first goal is to decrease the tiredness implied by previously presented techniques. One constraint is then to allow users only to move the fingers, letting the palm beside of the keyboard, as shown in figure 1. The second goal is to reduce the cognitive load implied by a transformation between the manipulation space and the working space. By only moving the fingers, it is not possible to perfectly fit the two spaces, and a scale as well as a translation are required. However, the rotation, which is the transformation with a strongest impact on cognitive load, is suppressed.

These two aforementioned goals implies that we initially assumed that it was not useful for the users to visually identify the 3D interaction volume, i.e. the volume in which gestures are possible since fingers are tracked by the system. An informal evaluation showed that our assumption was right and that it is not important for users to know the limits of the tracking area. Indeed, users do not look at their fingers but only at the screen (same as when using a mouse). They visually understand the interaction workspace, seeing the limits on the screen. A scale transformation between the tracking area and the displayed area is therefore possible. This allows us to reduce the gesture amplitude and therefore the tiring effect while preserving enough precision for pointing tasks. The tracking area has been defined considering the medium size of a hand: for right handed users, the defined area horizontally corresponds to the right half part of the keyboard, and vertically corresponds to the lower half part of the screen (see Figure 2). However, a calibration step allows to readjust this area for users with very small or very big hands.

The main technological issue is first to track a part or the whole fingers in 3 dimensions. The vision-based implementation proposed in this paper is based on vision reconstruction algorithms [6], and is able to track many points in 3D. Two trackIR [12] camera devices are placed on top of the laptop screen (see Figure 2). Each trackIR device is composed of one infra-red camera that is circled by infrared LEDS. Thus, a reflector placed in front of the device allows us to reflect the infrared light from the LEDS back to the camera. Using the trackIR SDK, we implemented an algorithm for reconstructing the three dimensional position of the reflector area. Many reflectors can be used.

As shown in Figure 2, the prototype, that has been done for validating the technique, includes two TrackIR cameras far from the top of the screen. For sure this prototype is not usable in all the usage contexts of a laptop. However, we focus here on validating the AirMouse technique. Using cameras with large enough focal distance will enable us to fix the cameras at the two top corners of the screen. Another future solution would be to use a single camera providing depth cues [13].

In this implementation of the AirMouse technique, we only focus on 2D and 3D pointing. In order to provide an intuitive and natural way of pointing, we decided to use isotonic interaction instead of isometric interaction.

2D Pointing The forefinger is commonly used for designating far or proximal objects. For this action, we can consider the finger as defining an infinite ray which intersects with the designated object. This technique [1] is commonly called Ray-Casting [2]. It is a natural and conceptually simple [27] pointing technique. The Ray-Casting technique sounds adapted for AirMouse: while using the keyboard, the user moves her/his forefinger and the cursor will be displayed on the designated position on screen. For this technique, two reflector ring are used (see Figure 2): one on the forefinger tip, and another one on the forefinger third phalanx. Thus, the two recorded 3D points allow us to define a line whose intersection with the screen plan gives the 2D position of the cursor.

3D pointing Only the forefinger tip reflector is used here. The tracking by the two cameras, combined with a classical reconstruction algorithm [6], gives one three-dimensional point. The 3D cursor then moves according to the three-dimensional position of the fingertip. In order to preserve the directness of the interaction, no rotation transformation is applied between the tracking area and the displayed area. There is a direct mapping: left/right and up/down movements of the fingertip are directly mapped to cursor movements in the same direction. Direct mapping is also provided for the depth: while the fingertip is going away from the screen, the cursor moves ”closer” to the user. Nevertheless as explained above, we apply a scale transformation between the tracking area and the displayed area for reducing gesture amplitudes and therefore tiredness while maintaining enough precision for pointing tasks (see Figure 2 Interaction area).

Selection The selection, equivalent to a mouse click, is performed by clicking the touchpad button of the laptop.

The goal of this implemented technique of AirMouse is to experimentally study the 2D and 3D pointing tasks. Nevertheless we point out that AirMouse intends to replace the mouse by providing a large range of possible interaction techniques that need to be further studied. For example for this implementation, the clutching aspects are not examined. We nevertheless show in the last section of this paper an implementation of AirMouse for an existing 3D modeler that supports smoothly integrated 2D and 3D pointing tasks with no need for activation/deactivation.

4 2D EVALUATION

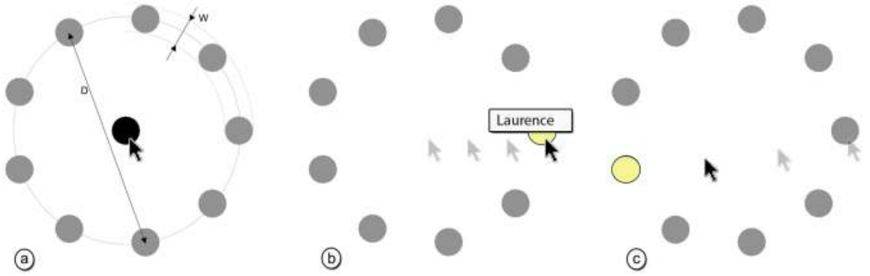

Fig. 3 Combination of multidirectional tapping with click-and-write. **a**: Initial cursor position, with amplitude and width visual representation. **b**: First target is reached (and clicked), the subject wrote her first name. **c**: ‘RETURN’ has been pressed, the cursor is going to the next target.

In this controlled experiment, we evaluated the performance of the above implemented technique of AirMouse as a pointing device, using 2D and 3D Fitts’ law studies. The two tasks have been performed by 15 subjects with no prior experience with 3D interaction devices. They were right handed, all rated themselves as advanced computer users and had normal or corrected normal vision. The two tasks are based on the recommendation given by Soukoreff et al. in [25], using the Fitts’ law [8]. Finally, at the end of each experiment, we asked participants to freely comment on the techniques and then to rank-order each of the experimented pointing devices respectively in terms of performance, satisfaction and tiring effect. As pointed out in [23] subjective satisfaction may be the key determinant of success.

4.1 Pointing Task

The goal of this evaluation is to position the 2D pointing performance of AirMouse in relation with the performance of other traditional device. We therefore compared Air-Mouse with the three well-known and commonly used pointing devices: the traditional mouse, the touchpad and the key-joystick. Traditional mouse is an isotonic device and every advanced computer user can be considered as an expert, i.e. the time performance of this device are partly due to the advanced knowledge of the users. The touchpad is also an isotonic device with a limited interaction space. The efficiency of use of such a device can be optimized by improving the scaling factor: however, the limit is fixed by the corresponding obtained precision quality. In the experiment, the scaling factor has been chosen empirically, computing the mean of three users parameters. The same value has been kept for all subjects. The key-joystick is an isometric device, i.e., it controls the cursor by speed. It can be found on a large variety of laptop, but it is not often used. None of the subjects has regularly used this device before the experiment, or just a few times for testing it.

This two-dimension Fitts'law task is administrated using the multidirectional tapping task paradigm [25], described in the ISO9241-9 standard, in which the subjects have to successively clicked on circular targets placed along a circle (see Figure 3). This paradigm presents the advantage of controlling the effect of direction. The distance between two successively clicked targets corresponds to the amplitude (*D*) of the movement, and the size of each target is the width (*W*). Combining different widths, four different difficulties (*ID*), from 3 to 6, are proposed to the subjects (according to the formula $ID=log2(D/W + 1)$).

For starting the trial, the subject must click on the centered target. The first target is then highlighted. Because of the amplitude difference, the movement, which consists of reaching the first target from the starting point, is not kept in final results. In order to prove that the proposed device can reduce the homing time [3] (the time needed to reach the device from the keyboard, and vice versa), we decided to use the click-and-write technique, as proposed in [19, 7, 3], and combine it with the multidirectional tapping task [25]. Thus, a click on a target opens a small command line, in which the subjects were asked to write her/his first name and to press the 'RETURN' key for closing the command line. Since we do not implement a mode switching between pointing and typing, the cursor disappears at the bottom of the screen when the user starts typing. This evaluation is composed of 12 sessions: 3 sessions per device. Each session is composed of 12 trials: 3 trials per ID. One trial is composed of 9 clicks corresponding to the multidirectional tapping tasks. We then obtained 9x12x3 = 324 pointing events per device. For each device, the first session is considered as a training session, but subjects do not know it. The results of this session could be kept for analyzing learning effects of each device but they are not considered for the comparison of the device time performance. Considering the (ID, Session) couples, all the arrangements are used, in order to avoid learning/tiring or influence effects between the tested devices

4.2 Results

Quantitative Results. For each trial, three times are recorded:

1. **Homing1**: elapsed time between the "RETURN" key press and pointing start

2. **Pointing Time**: elapsed time between pointing start and the click on the target
3. **Homing2**: elapsed time between the click and the first letter key press.

The experience data are analyzed within the framework of General Linear Model Procedure from SAS Software. There is a significant effect of device for all the recorded time types (Homing1: *F = 1976.3; p < 0.0001*; Pointing Time: *F = 283.41; p < 0.0001*; Homing2: *F = 24.04; p < 0.0001*). The classification of the device performances can be deduced from Figure 4. For Homing1, AirMouse is close to zero while mouse and touchpad are quite similar, but faster than key-joystick. Homing2 is similar for each device, from 0.655s for AirMouse to 0.548s for the mouse. This similarity is due to the time needed to find the first key to press on the keyboard. This value is not dependent to the ponting device. In Pointing Time, mouse is faster, followed by the touchpad. Then, AirMouse and key-joystick are close but key-joystick is slower. This result confirms the work presented by Douglas and al. [7].

Considering the total time (Homing1 + Pointing + Homing2), the device parameter has a significant effect (*F = 300; p < 0.0001*) and pairwise comparisons show a significant difference between all the devices (*p < 0.0001* for each combination). The corresponding curve presented in Figure 4 presents the final classification of the devices. Despite of the very good performances of AirMouse for Homing1, Figure 4 allows us to point up the very good performances of the mouse for total time, maybe because of the expertise of the subjects. However, key-joystick is the slower device. AirMouse is slower but comparable to touchpad performances.

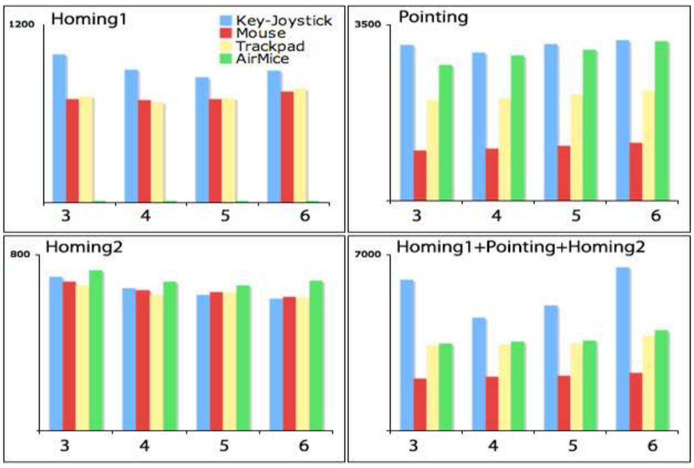

Fig. 4 Mean Time (in milliseconds) needed for each device and each difficulty for Homing1, Pointing, Homing2 and the sum of the 3 time values.

Qualitative Results. Concerning the perceived performances, all the subjects consider the key-joystick as the slowest one. This is confirmed by the quantitative results previously presented. This could be explained by the fact that they never used the device before the experiment. The AirMouse is higher ranked, although it was the first time the participants use it. The mouse is considered as the faster device by 77%

of the subjects. They are all experts, and the quantitative results confirm it. The AirMouse and the touchpad are quite similar, with a slight advantage for the AirMouse. However, subjects have never classified the touchpad as the faster device, in constrast with the AirMouse that has been classified as the faster device by 23% of the subjects.

Concerning the preference classification, the key-joystick is unanimously the less appreciated device (lower rates for all the subjects), mainly because of the fastidious learning time required. For the touchpad, results show the same pattern as previously, i.e. it is never the favorite device, but the second or the third selected device. AirMouse seems to be the most favorite device, i.e. it is classified as the most comfortable, intuitive and easy to learn device by 70% of the subjects. However, 77% of the subjects has classified the mouse in first or second position. Again, this could be explained by the mouse expertise of all the subjects.

Considering the tiring effect, only 10% of the subjects express a small feeling of tiredness in the hand for the AirMouse. But they consider it as a side effect of the experiment, i.e. the technique is new, and the hand is contracted for moving as fast as possible.

5 3D EVALUATION

5.1 Pointing Task

As for the above 2D Fitts' law study, the goal of this evaluation is to position the 3D pointing performances of AirMouse in relation with other traditional device performances. We then compared our technique for 3D translation pointing with two wellknown devices: the PHANTOM [18], an isotonic arm-based pointing device, which can provide haptic feedback (not used in the experiment) ; the SpaceNavigator [5], an isometric joystick.

Because of their isotonic property, and excluding the grasping action of the stylus, the movements with the PHANTOM and with the AirMouse for selecting and manipulating an object in translation are close. However, compared to theAirMouse, the PHANTOM is expensive and bulky. The comparison between the AirMouse and the spacemouse is interesting because of the popularity of the spacemouse. It is a low cost device, commonly used by designers for manipulating objects in 3D modelers. However, mainly because of its isometric property, its pointing time performances are lower than the ones of the PHANTOM [30]. Moreover, despite the tuning possibilities, the manipulation of such a device is not easy to learn and implies training time.

The PHANTOM and the AirMouse have a limited workspace, that we can consider as a cube. After preliminary tests, an empirical scale of each device workspace has been defined in relation to human skills, and then used for all the subjects. The scale of the AirMouse workspace has been defined in order to avoid subjects to move their hands for moving the cursor from the left to the right of the screen, allowing them to do it with only finger movements. Similarly the scale of the PHANTOM has been fixed in order to minimize arm movements.

3D pointing devices are usually used in manipulation tasks. Then, in order to fit with reality, the 3D pointing performances are evaluated with the same principle

presented in [31] and recently used in [11]: subjects have to manipulate a tetrahedron and bring it inside another bigger one. Fitts' law studies can be used according to the Prince technique proposed in [15], the cursor being an area cursor.
Because 3D rotation is not considered and only 3D translation is used, the tetrahedrons are thus replaced by spheres. After selecting a green sphere with the end-effector of the device (represented by a small radius sphere, the same for each device), the subject has to bring it into a transparent spherical area (see Figure 5 for a snapshot of the 3D environment of the evaluation). Before the selection, the target area is not displayed, in order to avoid any anticipation of the recorded movement. The radius of the green manipulated sphere R is fixed. This manipulated object corresponds to an area cursor. While the distance D between the two spheres is fixed during the experiment, the radius of the target area R' is modified in order to define 3 different levels of difficulties (IDs): 3,4,5, according to the Fitts' law formula used in Prince:

$$ID = \log_2(D/(R'-R) + 1)\ . \tag{1}$$

The evaluation has been performed on a traditional laptop without simulation of visual stereoscopy. In order to improve the depth perception, real-time shadows have been added. The horizontal position of each sphere is represented by a black disk projected onto the bottom plan (see Figure 5) and perceived by the subjects in their peripheral vision. This evaluation is composed of 9 sessions: 3 sessions per device. Each session is composed of 21 trials: 7 trials per ID. We then obtained 9x21=189 events per device. For each device, the first session is considered as a training session, but subjects do not know it. The results of this session could be kept for analyzing learning effects of each device but they are not considered for the comparison of the device time performances. Considering the (ID, Session) couples, all the arrangements are used, in order to avoid learning/tiring or influence effects between the tested devices.

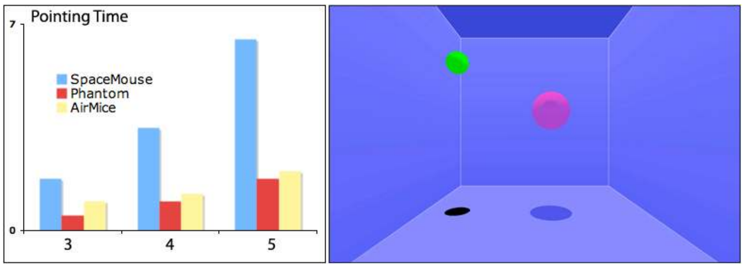

Fig. 5 Left: Mean Pointing Time (in seconds) for each device and for each ID. Subjects are slower with the SpaceMouse and faster with the PHANTOM. AirMouse is placed in between, but closer to the PHANTOM. **Right**: Snapshot of the 3D pointing evaluation environment. The green sphere on the left must be moved as fast as possible into the transparent red area. The transparent area size is modified during the experiment for defining different difficulties (IDs).

5.1 Results

Quantitative Results : Pointing Time. The experience data are analyzed within the framework of General Linear Model Procedure from SAS Software. For each trial, Pointing Time (PT) has been recorded between the date of the click required for selecting the green sphere to be manipulated and the date of sphere disappearance into the spherical target area. PT values (in seconds) are presented in Figure 5. There is a significant effect of the device ($F = 122.11; p < 0.0001$), with mean times decreasing from 4.9s ($SD = 4.4$) with the spacemouse, through 2.6s ($SD = 1.9$) with the AirMouse, to 1.8s ($SD = 1.2$) with the PHANTOM ; a 63% reduction in PT across the three conditions. The spacemouse is the slower device. Observing subjects during the experiment, we could notice that pointing movements are natural and intuitive for the PHANTOM and the AirMouse. However, using the spacemouse, subjects usually try to decompose the pointing movement: first, following the shadow cues, they try to adjust the position in the horizontal plan, then they adjust the height. This decomposition is not observed with other tested devices, so we suppose that it is linked to the spacemouse capabilities. This could explain the spacemouse low performances. Moreover, after the learning effect of the first session, the movement is more direct, but still slower than the movements with the two other tested devices.

Post-hoc pairwise comparisons showed a significant difference between the AirMouse and the spacemouse ($p < 0{:}0001$) and a less significant difference between the Air-Mouse and the PHANTOM ($p = 0.0168$). As expected by the Fitts'Law study, the difficulty (ID) has a significant effect on task performance ($F = 95.84; p < 0.0001$). Figure 5 shows both results, the effect of the device and the effect of the ID on PT. Mean PT increases with the ID for all the devices. It is higher with the spacemouse and lower with the PHANTOM. AirMouse is placed in between, but closer to the PHANTOM.

There is a significant effect of the session ($F = 23.05; p < 0.0001$), with mean times decreasing from 3.8s ($SD = 3.7$) for session 1, through 3.0s ($SD = 3.4$) for session 2, to 2,5 ($SD = 1.8$) for session 3; a 34% reduction of PT across the three conditions. However, session 1 and session 3 are significantly different ($p = 0.0003$), but session 2 and session 3 are not ($p = 0.06$). This effect is explained by the learning effect between session 1 and session 2, that seems to disappear between session 2 and session 3.

Qualitative Results. Concerning the perceived performances, without ambiguity, all subjects estimated that they are slower with the spacemouse. This is confirmed by the quantitative results previously presented. The main quoted reason is the learning stage linked to the sensitivity of the sensors. Concerning the PHANTOM and the AirMouse, 55% of the subjects are not able to know which of the two devices offer the best performance, and 33% took a decision and said that the PHANTOM is faster.

Concerning the preference classification, the spacemouse is unanimously the less favorite device. In contrast, the PHANTOM and the AirMouse are considered as intuitive, without learning stage. The viscosity of the PHANTOM (i.e. the resistance provided by the arm mechanism) is considered as helpful for precision by 44% of the subjects, but 33% consider it as a disturbing side effect. Compared to AirMouse, the PHANTOM is considered as less transparent and more invasive.

Concerning the tiring effects, 10% of the subjects express a feeling of tiredness in the hand for the AirMouse. 10% of the subjects also express a feeling of tiredness using the PHANTOM, because of the movements of the hand and the forearm. However, as for the previous evaluation, they explained it by the stress of the experiment, trying to perform the tasks as fast as possible.

6 DISCUSSION AND FUTURE WORK

Fig. 6 Left: Use of the two hands for 3D rotation: metaphor of planting a pin inside the object to be rotated. **Right**: Mixed 2D and 3D Interaction with AirMouse in Autodesk 3D Studio Max. the 2D pointing technique has been plugged to the mouse cursor while 3D pointing becomes active when the user moves over the perspective 3D view.

As expected, these evaluations allow us to position AirMouse in relation to other existing and commonly used pointing devices. Results show that the performances of AirMouse in 2D pointing are not better than the ones of the mouse, but are comparable to the ones of the touchpad, and are better than the ones of key-joystick. Since the pointing performance with FlowMouse has been reported to be significantly worse than with a touchpad [28], we conclude that AirMouse offers better pointing performance than FlowMouse. In 3D pointing, AirMouse is not so far from the PHANTOM and a lot faster than the spacemouse. Finally, qualitative results show that the performance is not the most important criterion. Subjects prefer to use a device that is intuitive and easy to learn while providing correct performance. Based on these criteria, AirMouse is appreciated by most of the users. They consider the technique as promising, and useful for laptop configuration.

Based on these encouraging results obtained for 2D /3D pointing tasks, it is now possible to further investigate AirMouse for other object manipulation tasks. AirMouse opens a vast world of possibilities in terms of interaction techniques. For example we plan to explore two-handed AirMouse interaction and gesture recognition as in [11, 9, 28] or based on real world metaphor: for example the user can perform a gesture similar to the one of turning a page in a book in order to scroll to the next page of a document. While for pointing tasks, mode switching between pointing and keyboard was not a key issue since the technique supports a direct designation of the objects on screen and therefore the cursor can move and disappear while typing, natural and efficient mode switching [28] is a primarily issue for the other tasks that we study and envision.

Since AirMouse seems very promising for 3D pointing, we first study the use of the AirMouse for full 3D manipulation and we started to investigate 3D rotation. A prototype has been designed and will be described in a next paper.

Finally an interesting feature of AirMouse is to support both 2D and 3D interaction in the same application. In order to informally evaluate the combined usage of 2D and 3D interaction, we tested the vision-based implementation of AirMouse in the context of a 3D modeler called Autodesk 3D Studio Max (3DSMax). Figure 6 shows a screenshot of the application. The two pointing techniques (2D and 3D) have been mixed: the 2D pointing technique, used in the experiment and based on two reflectors, has been plugged to the mouse cursor. The switch between 2D and 3D is based on the application mode defined by the cursor position. When the cursor of the mouse is over the 3D view, 3D pointing becomes active and 3D movements of the fingerTip are used. To sum up, the user can interact with 3DSMax, by moving the desktop cursor with its forefinger and clicking on icons. The user can also perform 3D manipulation as soon as the cursor is within the 3D scene: the arrow cursor is then replaced by a small sphere. The transition between 2D and 3D interaction is therefore observable as well as implicit and smooth based on the application mode activated by the position of the cursor and more importantly based on the same AirMouse technique.

7 CONCLUSION

In this paper, we have introduced and studied a new technique, namely AirMouse, for 2D and 3D interaction using finger gestures above the keyboard. The controlled experiment of a vision-based implementation of AirMouse shows the promising pointing performance of the technique compared with existing and commonly used devices for a pointing task. Subjects pointed out the intuitive, easy to learn and comfortable aspects of AirMouse that does not require additional surface for interaction using a laptop. In addition to our current studies of other tasks than pointing using AirMouse, a longitudinal evaluation of the 2D pointing is under investigation. We plan to test AirMouse with three regular computer users (scientists in the lab) in their everyday work, replacing the mouse by AirMouse.We hope to observe an improvement, which will make the technique comparable with the mouse in terms of time performance.

Acknowledgments. The authors would like to thank R. Blanch for extensive comments, A. Demeure for his help in the code for evaluation and all the participants in the experiments. Special thanks to N. Mandran for her precious help and advise on statistics for the evaluation and G. Serghiou for reviewing the paper. The work presented in the article is partly funded by the European Commission under contract OpenInterface (FP6-35182), www.oi-project.org .

References

1. Smith, R. Bolt. Put-that-here: Voice and gesture at the graphics interface. Computer Graphics, 14:pp. 262–270, 1980.
2. Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. 3D User Interfaces: Theory and Practice. Addison-Wesley Professional, 2005.
3. S. K. Card, W. K. English, and B. J. Burr. Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys, for text selection on a crt. In Human-Computer Interaction, pages pp. 386–392. Morgan Kaufmann Publishers Inc., 1978.
4. Stuart K. Card, Thomas P. Morans, and Allen Newell. The Psychology of Human Computer Interaction. Lawrence Erlbaum Associates, Inc., 1983.
5. 3D Connexion. Space navigator, http://www.3dconnexion.com.

6. D. Dementhon and L.S. Davis. Model-based object pose in 25 lines of code. In International Journal on Computer Vision, 1995.
7. Sarah A. Douglas and Anant K. Mithal. The effect of reducing homing time on the speed of a finger-controlled isometric pointing device. In Computer-Human Interaction, pages 411–416. ACM, ACM Press, April 1994.
8. Fitts P. M.: The information capacity of the human motor system in controlling the amplitude of movement. Journal of Experimental Psychology, 47(6):pp. 381–391, 1954.
9. Grossman T., Widgor D., and Balakrishnan R.: Multi-finger gestural interaction with 3d volumetric displays. In User Interface Software and Technology. ACM Press, October 2004.
10. J. Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In Symposium on User Interface Software and Technology. ACM Press, 2005.
11. Hancock M., Carpendale S., and Cockburn A.: Shallow-depth 3d interaction: Design and evaluation of one-, two- and three-touch techniques. In Computer-Human Interaction, 2007.
12. Natural Point http://www.naturalpoint.com. Trackir http://www.naturalpoint.com/trackir.
13. G. J. Iddan and G. Yahav. 3d imaging in the studio. SPIE, www.3dvsystems.com.il, 2001.
14. Robert J. K. Jacob and Linda E. Sibert. The perceptual structure of multidimensional input device selection. In SIGCHI conference on Human factors in computing systems, 1992.
15. Kabbash P. and Buxton W.: The "prince" technique: Fitts' law and selection using area cursors. In Computer-Human Interaction, pages pp. 273–279, 1995.
16. MacKenzie I. S., Sellen A., and Buxton W.. A comparison of input devices in elemental pointing and dragging tasks. In Computer-Human Interaction, pages pp. 161–166, 1991.
17. Malik S. and Laszlo J.: Visual touchpad: A two-handed gestural input device. ICMI, 2004.
18. Thomas H. Massie and J. K. Salisbury. The phantom haptic interface: A device for probing virtual objects. In Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. ASME winter Meeting, 1994.
19. Thomas A. Mysliwiec. Fingermouse: A freehand computer pointing interface. Technical report for the degree of Masters of Science, October 1994.
20. A. Paljic, J. Burkhardt, and S. Coquillart. A study of distance of manipulation on the responsive workbench, 2002.
21. Jeffrey S. Pierce, Brian C. Stearns, and Randy Pausch. Voodoo dolls: Seamless interaction at multiple scales in virtual environments. In Symposium on Interactive 3D graphics, 1999.
22. Qhueck F. K.H.: Unencumbered gestural interaction. In IEEE Multimedia, Winter 1996.
23. Schneiderman B.: Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison-Wesley Longman Publishing Co., Inc., 1986.
24. Shneiderman B.: The future of interactive systems and the emergence of direct manipulation. Human Factors in Interactive Computer Systems, 1983.
25. Soukoreff R. W. and MacKenzie I. S.: Towards a standard for pointing device evaluation, perspectives on 27 years of fitts' law research in hci. International Journal of Human-Computer Studies, 61:pp. 751–789, November 2004.
26. Tory M., Moller T., Atkins M. S. and Kirkpatrick A. E.: Combining 2d and 3d views for orientation and relative position tasks. In CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems, pages 73–80, New York, NY, USA, 2004. ACM.
27. Daniel Vogel and Ravin Balakrishnan. Distant freehand pointing and clicking on very large, high resolution displays. In User Interface Software and Technology, 2005.
28. Wilson. A. D. : Robust computer vision-based detection of pinching for one and twohanded gesture input. In User Interface Software and Technology. ACM Press, October 2006.
29. Wilson A. D. and Cutrell E.: Flowmouse: A computer vision-based pointing and gesture input device. In INTERACT : human-computer interaction, 2005.
30. Zhai S.: Human Performance in 6dof Input Control. PhD Thesis, 1995.
31. Zhai S. and Milgram P.: Quantifying coordination in multiple dof movement and its application to evaluating 6 dof input devices. In Computer-Human Interaction, pages 320–327. ACM Press, April 1998.