The Perceptual Window: Head Motion as a new Input Stream

François Bérard

CLIPS-IMAG
BP 53
38041 Grenoble Cedex 9
FRANCE
francois.berard@imag.fr

ABSTRACT We introduce a novel interaction technique using head motions to control the location of a window viewpoint within its document space. Head motion is acquired by a non-intrusive head-tracker using computer vision. The tracking technique used in the system, namely correlation matching, is described in order to exhibit its strengths and weaknesses in the context of tightly coupled interaction. The output of the tracker is used in both a rate control interaction and a position control interaction. The benefit is demonstrated by two user studies built around two common GUI tasks: navigating in a two-dimensional document space, and moving an object from one place to another in a document. Our system, The Perceptual Window, allows significant improvements in task completion time after a short learning period.

KEYWORDS Input Devices, Computer Vision, Head Tracking, Multiple Streams of Spatial Input, Interaction Techniques, Perceptual User Interface, Scrolling, Drag and drop

INTRODUCTION

The idea of using head tracking as an input component for interactive systems has appeared in the literature for some time. Most publications emphasize techniques for head tracking rather than its benefit in human computer interaction (Yang 1998). Tomaya (1998) proposes the use of head posture to control the pointer of a GUI but the benefit is not clearly demonstrated. Gaver (1995) presents a promising application of face tracking, the Virtual Window, but the face tracker used in the system can not meet HCI requirements. The work presented here focuses on measuring the benefits of head motion as a general purpose spatial input stream in standard Graphical User Interfaces (GUI).

This paper is organized as follow: the work is motivated with a review of previous results on multiple streams of spatial input. Our face tracker is then presented, followed by a description of the global system. The Perceptual Window uses head motions to set the location of a window viewpoint within its document space. We present the results of two user studies demonstrating that the system is easily accepted by users and significantly improves performance for two common GUI tasks.

1. MOTIVATIONS

The benefit of using a second stream of input, in addition to the mouse, has long been known (Buxton 1986). Continuous and compound tasks are achieved with greater performance when their load is balanced over two hands, rather than with only one hand holding the mouse. The most intuitive source of the improvement is motion efficiency. However, more recent studies have shown that two streams of input can be worse than one if they are not appropriately designed (Kabbash 1994). It seems that the cognitive load has a greater impact on task completion time than the amount of motion involved. Therefore, parallelization of user actions should associate a simple, coarse-grained action with a more complex, accurate one. Guiard (1987) presents a model which characterizes the complementary use of two hands to achieve such parallelization. He describes the respective roles of the “dominant hand” and the “non-dominant hand”. Guiard’s model was successfully applied by Zhai (1997) in an experiment where the best combination was scrolling with the non-dominant hand and pointing with the dominant hand, compared to other multiple spatial input devices.

In our studies, the head plays the role of the non-dominant hand because we claim that it fulfills its three characteristics:

1. The head defines the frame of reference for the mouse: it sets the viewpoint of the window representing the mouse workspace.
2. Motion sequence is: head first (set the viewpoint), then the mouse (click in the window).
3. The head executes coarse-grained motions (window viewpoint does not have to be set accurately) compared to the mouse.

2. THE SYSTEM

Our system, called “Perceptual Window”, is based on a Computer Vision tracker measuring the head motions involved in the interaction. The system is “perceptual” in the sense that it reacts directly to user’s actions without need for separate physical contact. In this section, we describe the system from the low-level hardware set-up to the high-level interaction control.

2.1 Hardware setup

The input to the Perceptual Window is provided by a video camera (pan-tilt-zoom SONY EVID31) placed in front of the user’s head, on top of the monitor. The camera provides a video stream of images of the user to the frame grabber. The workstation is an Apple Power Macintosh 8600 equipped with a PowerPC 604 processor running at 350 Mhz and a built-in frame grabber. Note that this setup uses only off-the-shelf hardware, which is not the case for most new input devices.

2.2 Head Tracker

Head motion is detected by tracking a region of the face over time using *correlation matching* (Anandan 1989). The principles of correlation matching are:

1. memorize an image of the target at initialization,
2. search the target in a new image by *correlating* (measuring similarity with) the memorized target image to subparts of the new image. The subpart that *matches* (that most resembles) the target image is selected as the new target location. Details on the matching formula are available in the appendix. In

order to keep computation cost low, the search in the new image must be restricted to a small area, centered on the target location in the previous image.

In the context of tightly coupled interaction, correlation matching has two key features making it suitable for our system.

1. The system is fast. In our implementation, the processing of a new image takes less than 16 milliseconds (image size is 384 x 288 pixels, target image is 32 x 32 pixels, search area is 60 x 60 pixels). Adding the time necessary to generate the feedback, our system has a maximum latency of 65 milliseconds (translating to a running frequency above 15 Hz). Achieving low latency is critical for the usability of the system.
2. The system is stable: if the target does not move, the output of the tracking remains strictly constant. A magnetic tracker, for example, does not achieve such stability. Stability is equally a requirement for tightly coupled interaction.

None-the-less, we observe three main limitations of correlation based tracking.

1. Tracking fails when the target is too fast. This occurs when the target speed is such that the displacement between two frames is greater than the search area. In the context of this work, this was not a problem because the speed range of face motions are easily bearable by the tracker.
2. Tracking fails if the target undergoes significant rotations or large changes in distance. Such motions induce modifications of the *appearance* of the target in the image. The new target image fails to match the memorized target image, causing the tracking to fail. The phenomenon is aggravated if the target is chosen on an area of the face having variable depths (such as the nose or the frame of the eyeglasses). In our context, we carefully chose the target on a planar area of the face.
3. Tracking fails if the neighborhood of the target has a similar appearance to the one of the target itself. This occurs for example when choosing a target on a wide non-textured area of the face, such as the forehead.

In our experiment, the most annoying limitations were numbers 2 and 3, as we were forced to manually initialize the tracking to a suitable target. We chose the area between the two tips of the eyebrows because the appearance of this area would not change significantly with head rotations (limitation 2), and because it is dissimilar enough to any of its neighboring areas (limitation 3).

Manually initializing the target of the tracking would be a serious limitation for general use of our system. Making the tracking autonomous is one of our research efforts. Encouraging initial results have recently been obtained (Bérard 1997). In our user experiments, the lack of autonomy was not a limitation. The tracking was initialized by an operator only once at the beginning of the experiment. The tracking requires re-initialization only if the user turns his head away from the monitor, which typically did not occur during the experiments.

In order to maximize the resolution of the tracker output, the camera field of view was manually adjusted to maximize target translations corresponding to the widest comfortable head motions. However, there is a trade-off between maximizing the resolution and minimizing the risk that the target leaves of the camera field of view. To get an idea of the resolution achieved, typical target translations remained in a 260 x 200 pixel rectangle within a 384 x 288 image (thus about 70% of the maximum in both dimensions).

2.3 Interaction

The tracker outputs the two-dimensional coordinates (x, y) of its target in the image space. When integrated over time, this results in the translation parameters of the target (variation in x and y from some original location). The target being set on the face of the user, the system is informed about face motions. Note that the estimated translation is the translation of the *image* of the target, not of the *target* itself. The benefit is that users do not have to actually translate their face to operate the system, they rather rotate it, which is a much more comfortable motion to execute. The head being rotated, a target set on the face appears to translate in the image.

The head is allotted the task of navigating within the document space. Head motions are used to set the view location of a window showing a subpart of the document (on standard workstations, this task is typically allotted to the mouse and scroll-bars). The interaction is controlled with a *trigger* key. We used either “tab” or “space” for the trigger key but modifiers (“shift”, “control” or “alt”) would also be good candidates. The trigger key has two purposes. When the user first depresses it, the *origin* of the translation is set to the user’s current head location. Secondly, the system is switched into head control mode: the control occurs only while the trigger key is depressed; it stops immediately when the key is released.

We use this setup to implement two different kind of interactions: one is a rate control interaction, the other one is a position control interaction.

Rate control interaction

As the head is tilted upward outside of a *neutral* area, the window content is scrolled down. The more the head is tilted, the faster the scrolling. Upward scrolling is stopped by returning the head inside the neutral area. Tilting the head downward induces a symmetrical behavior. Rotating the head left or right causes scrolling to the right or to the left, respectively. Scrolling speed is governed by an exponential rather than a linear relationship to this movement, permitting both accurate adjustments and fast scrolling depending on the degree of head rotations. Figure 1 shows the transfer function for upward / downward head rotations. The transfer function for left / right head rotations has exactly the same shape. Diagonal scrolling is achieved by rotating the head both horizontally and vertically. To get a better feeling of the perceptual window behavior, movies of the running system are available on the web[1].

Considering that it is possible to initiate and stop the scrolling in two different ways (neutral area and trigger key), we observed the two following behaviors:

1. novice users sometimes found it difficult to return the head to within the neutral area in order to stop

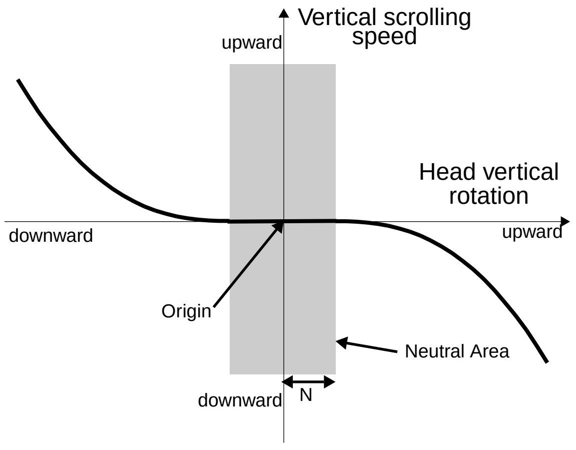


Figure 1: Transfer function for upward / downward head rotations.
N is the radius of the neutral area.

the scrolling. They preferred to press and release the trigger key for this purpose. The radius of the neutral area was set to 0 in order for the scrolling to start as soon as the trigger key was depressed.
2. some trained users performed a more continuous navigation by maintaining the trigger key in the depressed position. They used the neutral area to stop the scrolling when needed. In that case, the radius of the neutral area was set to 3% of the processed image height. The trigger key was only used as a higher level start / stop control, at the beginning and the end of the global task.

A usability study has been performed on this interaction (Bérard 1999). A summary is presented in section 3 (“Exploratory experiment”).

Position control interaction

This interaction is similar to the mouse interaction in the sense that translations performed by the user are directly reflected by the system. User head translations results in the same translation of the window viewpoint in the document. The only processing applied to the tracker output is amplification (multiplication of the translation by a constant).

Amplification is necessary because of the relative low resolution of the tracker (see end of section 2.2) compared to the mouse resolution. If amplification were not performed, many *repositioning* would be necessary for large viewpoint translations. A repositioning occurs when the head is rotated to its

1. http://iihm.imag.fr/demos/pwindow/

maximum, but a wider translation is necessary. The trigger key is released so that the head is able to return to its neutral posture without changing the window viewpoint. Then, the trigger key is depressed again to apply a further translation in the initial direction. This is similar to a mouse repositioning occurring when the mouse has reached the limit of a comfortable wrist motion (or the limit of the mousepad). The mouse must be raised in order to move it to a new location without sending inputs to the workstation. In our experiments, users almost never had to reposition their head because they were able to go from one border of the document to the opposite with a single head-rotation.

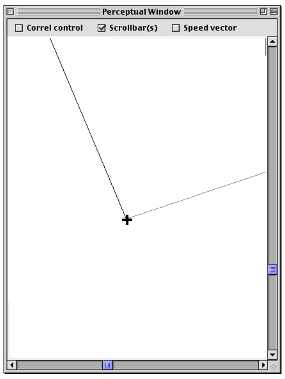
Applying a high *gain* (the amplification factor) to the tracker output has the inconvenience that it increases the minimal step of viewpoint translation. In our experiments, a 1-pixel translation of the head triggers a 31 pixel translation of the document (6.2% of the window height). The movies (see footnote 1 on previous page) provide an appreciation of the system behavior.

Usability of the perceptual window has been tested against users for two different families of navigational tasks. An initial experiment tested the rate control interaction in an exploratory task (users had to follow a path to the target). In the second experiment, position control interaction was used in a “drag and drop” task in which the destination of the navigation was made immediately available to the user.

3. EXPLORATORY EXPERIMENT

This experiment is reported in another publication (Bérard 1999), results are summarized in this section for the completeness of the paper.

The experiment task is based on real two-dimensional navigational tasks such as following lines and columns of a large spreadsheet, following the shape of a mechanical part on a high resolution technical sketch, or navigating via a mental representation of a large picture when only a small part of it is visible.



☐ Correl control ☑ Scrollbar(s) ☐ Speed vector

Figure 2: A target along the path.

Users were presented a succession of 50 targets. They had to click on a target, follow the line to the next target by scrolling the window, click on the next target and so on (figure 2). Subjects performed the task once while scrolling the window with the scrollbars and once with the rate control interaction.

Head motion interaction significantly outperformed scrollbars by an average improvement of 32% on task completion time.

This experiment showed that subjects were able to use this new modality surprisingly well: in less than a minute, they knew how to use it and could do so with skill. All users preferred head motion over the scrollbars. One user commented that scrolling control with the head was very *natural*: he simply had to orient his head towards what he wanted to see, and it just appeared in the middle of the window.

Rate control interaction is suitable for exploratory navigations because it allows a user to easily set the speed and direction of scrolling (which scrollbars can not do). However, in the case where the user knows where to go at the beginning of the navigation, we found that position control interaction was more appropriate.

4. “DRAG AND DROP” EXPERIMENT

In this experiment, we reproduce the steps necessary to move an object from one location to another in a large document displayed in a smaller window. This is a very common task when editing large texts, pictures or spreadsheets.

4.1 Motivations

Consider the two different situations occurring in standard GUI:

1. Both the object and its destination are visible in the window.
2. The object is visible in the window but not its destination.

In the first case, many applications allow a “drag and drop” operation: the object is “caught” by depressing the mouse button and maintaining it depressed while the mouse pointer is on top of the object. The object is then moved to its destination by moving the pointer while still maintaining the button depressed. The object is actually moved when the user releases the button after pointing to the destination.

In the second case, a drag and drop operation becomes difficult because the user has to set the viewpoint to destination while “holding” the object. The mouse button being depressed, it can not be used to operate the scrollbars or any kind of navigational means requiring button clicks. Most applications offer an alternative: when the pointer holding the object crosses one of the window borders, the content of the window starts to scroll towards the opposite border. This navigational means is very limited in the sense that it does not allow the user to set the scrolling speed. It is very unlikely that the arbitrary scrolling speed suits the needs of the navigation. The scrolling is either too fast (the drop location comes in and out of the window before the user has a chance to stop it) or too slow (the user has to wait a long time before the destination appears in the window). In practice, when the destination is far away from the object, drag and drop is abandoned for a more efficient cut, navigate and paste sequence.

By using the perceptual window the mouse is freed from the navigational task. Therefore, the user can perform a drag and drop operation and leave the navigation to head control. As drag and drop requires less articulatory tasks than the sequence “select, cut, navigate, select destination, paste”, we anticipate better performances of the perceptual window compared to standard GUI.

4.2 Task

Users are presented a black square that must be put in a succession of 50 black frames (the *targets*). Only one target is visible at any time. Target locations are randomly distributed within the 1600 x 2000 pixels document space. Once the black square has been put in a target, the target disappears and the black square is moved at a random position in the window. The next target is created at the same time in the document space.

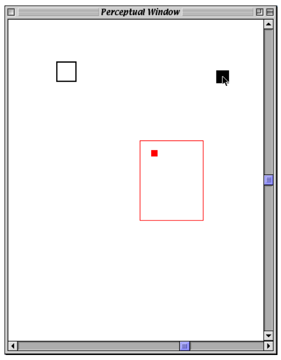


Figure 3: The radar view overlaid on the window. The big frame represents the current position of the window in the document space. The small square represents the target. The frame up-left is the target, the mouse is currently holding the black square.

At any time, only 6.2% of the document space is visible through a 400 x 500 pixels window. Users set the window viewpoint in the document space by using a *radar view*. The radar view represents a global view of the document space showing the current position of the window and the current target location in this space. When activated, the radar view is overlaid on top of the window: the window surface represents the document space, the current position of the window is represented by a red frame, the current target is represented by a small red square. Figure 3 shows an example of the radar view overlaid on the window. The radar view is activated when the space bar is depressed and remains active as long as the space bar is not released. The space bar is thus used as a trigger for both the radar view and the head control interaction.

Two different conditions were tested in this experiment. In the first condition, called *mouse condition*, the subject is asked to use the mouse and the keyboard to carry out the task. Subjects must select the black square (by clicking on it), depress the

"Command-X" key combination to cut it, call the radar view by maintaining the spacebar depressed, click near the target in the radar view. The window view-point is set to the click location in the radar view, the target is then apparent in the window. Subjects finish the operation by clicking on the target and depressing the "Command-V" key combination to paste the black square.

In the second condition, called the *head condition*, subjects carry out drag and drop operations by performing the following actions: depress and hold the mouse button on the black square, depress and hold the space bar to start the radar view and start head control interaction, rotate the head to let the window viewpoint include the target, release the space bar, drop the black square in the target.

Nine volunteer subjects performed the experiment twice for the two conditions. Four subjects started with the head condition, the other ones started with the mouse condition. Subjects had a minimal training by performing a test series of 50 targets before each of the two conditions.

4.3 Results

Data were analyzed with a paired samples t-test which revealed a highly significant difference between conditions (t(8) = 7.24; p =0.00008). As shown in figure 4, head condition outperformed mouse condition by an average improvement of 18% (mouse condition average completion time was 190 s. versus 155 s. for head control, standard errors were respectively 9.10 and 7.42).

Five users expressed their preference for the head condition, the 4 others preferred the mouse.

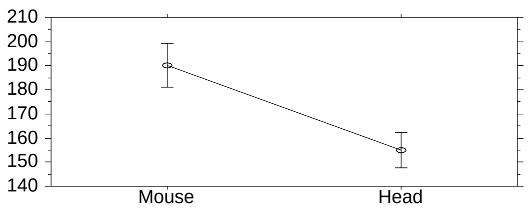


Figure 4: Completion time means (seconds) and standard errors by condition

4.4 Discussion

As with the exploratory experiment, the most striking results are the generality of the performance improvement and the ease of learning the system. All users where faster with the head condition, in spite of the fact that this was a completely new way for them to interact with a computer.

Post experiment interviews revealed a strong contrast between users that liked the head control and those that did not like it. This second group were impeded by the inability to stabilize the window location when controlling with the head. They incorrectly thought they had been faster with the mouse. The task of navigation did not require to accurately locate the window, navigation could stop as soon as the target entered in the window.

Replacing a cut and paste operation in a large space by a drag and drop improves user performance. The required number of user action is reduced, and furthermore, the operation seems to have a lower cognitive cost. How can a user know, for example, if the object has been cut when it is not visible? It may be cut, or it may be out of the window view. This does not occur with drag and drop as you can see if you are "holding" the object with the mouse.

5. CONCLUSION

This paper presents a novel means to provide spatial input to computers using head motions. The interaction is sufficiently natural that users can immediately improve their performances on common GUI tasks. The success of the experiments was due to the adequate allocation of tasks to the new input. Rather than replacing the mouse, head rotations can complement the mouse in a task at which the head naturally excels: setting the point of view.

It is noticeable that the hardware required for this input is already available on many workstations. As videoconferencing promotes the development of video-capable workstations, perceptual interfaces could be integrated as easily as a software component.

Another important point in favor of the acceptance of this kind of interaction is that it *complements* standard interaction rather than *replacing* it. The Perceptual Window features fully functional scrollbars. It simply offers new alternatives to allow a more natural interaction for some particular tasks.

Finally, we think this work opens a new path of possible GUI improvements. An interesting method to explore would be the suppression of the homing for the mouse in text editing tasks: wide cursor motions would be allocated to head motion while fine adjustments would still be accomplished by the cursor keys.

6. ACKNOWLEDGEMENTS

The author wish to thank James L. Crowley, Marie-Claude Frasson, and Leon Watts for their contribution to this work.

7. APPENDIX

Normalized Cross Correlation (NCC) formula:

Let $I(x, y)$ be the intensity of the pixel at coordinates (x, y) in image *I*. Let *T* be the image of the target, having size $n \times m$.

The Normalized Cross Correlation of target image *T* in image *I* at location (x, y) is given by:

$$NCC(x,y) = \frac{\sum_{u,v} I(x+u, y+v) \cdot T(u,v)}{\sqrt{\sum_{u,v} I^2(x+u, y+v) \cdot \sum_{u,v} T^2(u,v)}}$$

A publication (Crowley 1995) motivates the use of NCC and documents a fast implementation of a NCC based tracker in the context of finger tracking. The same implementation was used in the Perceptual Window.

REFERENCES

Anandan, P. (1989). A Computational Framework and an Algorithm for the Measurement of Visual Motion. *International Journal on Computer Vision*, 2(3),p. 283-310.

Bérard, F. Coutaz, J. and Crowley, J.L. (1997). Robust Computer Vision for Computer Mediated Communication. *In proceedings of INTERACT'97*. p. 581-582.

Bérard, F. (1999). A study on Two-Dimensional Scrolling with Head Motion. *CLIPS-IMAG Technical Report IMAG_CLIPS_IIHM_199901* http://iihm.imag.fr/

Buxton, W. and Myers, B. (1986). A Study in Two-Handed Input. *In proceedings of CHI'86*. p. 321-326.

Crowley, J.L. Bérard, F. and Coutaz, J. (1995) Finger Tracking as an Input Device for Augmented Reality. *In Proceedings of International Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland.

Gaver, W. Smets, G. and Overbeeke, K. (1995). A virtual window on mediaspace. *In proceedings of CHI'95*.

Guiard, Y. (1987). Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior* 19(4). p. 486-517.

Kabbash, P. Buxton, W. and Sellen, A. (1994). Two-Handed Input in a Compound Task. *In proceedings of CHI'94*. p.417-423.

Toyama, K. (1998)."Look, Ma – No Hands!" Hands-Free Cursor Control with Real-Time 3D Face Tracking. *In proceedings of the 1998 workshop on Perceptual User Interfaces*.

Yang, J. Stiefelhagen, R. and Waibel, A. (1998). Visual Tracking for Multimodal Human Computer Interaction. *In Proceedings of CHI'98*, p. 140-147.

Zhai, S. Smith, B. A. and Selker, T. (1997). Improving Browsing Performances: A study of four input devices for scrolling and pointing tasks. *Proceedings of INTERACT'97*. Chapman & Hall1997, p. 286-293.