Benefits of the use of natural user interfaces in water simulations

Gennadii Donchyts\textsuperscript{a,b}, Fedor Baart\textsuperscript{a,b}, Arthur van Dam\textsuperscript{a}, Bert Jagers\textsuperscript{a}

\textsuperscript{a} Deltares, P.O. Box 177, 2600 MH, Delft, the Netherlands
\textsuperscript{b} Civil Engineering and Geosciences, Delft University of Technology, Stevinweg 1, 2628 CN, Delft, the Netherlands

gennadii.donchyts@deltares.nl

Abstract: The use of natural user interfaces instead of conventional ones has become a reality with the emergence of 3D motion sensing technologies. However, some problems are still unsolved (for example, no haptic or tactile feedback); so this technology requires careful evaluation before the users can benefit from it. We argue, that the best benefits can be achieved when these natural user interface technologies are combined with classical computer interaction devices such as mouse and keyboard. In our demonstration, we will show how the LEAP Motion controller can be applied in environmental modeling when combined with the shallow water flow model engine D-Flow Flexible Mesh and a 3D scientific visualization library. We will analyze where the new approach provides benefits compared to the classical computer input devices such as mouse and keyboard. We will also demonstrate a number of visualization and interaction techniques used during manipulation of model input data (bathymetry, roughness, etc.) or during exploration of the results of a running model.

Keywords: scientific visualization; user-computer interaction; natural user interfaces; user interface design; hydrodynamics; environmental modeling

1. INTRODUCTION

Although geospatial visualization is becoming more and more widespread, there are still a lot of serious challenges in designing and developing the next generation of geovisualization software components (MacEachren and Kraak, 2001). The challenges increase further when we consider the integration of visualization and computation, for example interaction with a hydrodynamic numerical model from within a 2D/3D GIS environment. Effective 3D visualization of water bodies is already a challenging task by itself since these water bodies usually have large differences in the spatial scales, e.g., the water depth is usually much smaller than the length/width. Additional challenges arise when model input and output data change in time. Last but not least, we need to solve these visualization challenges in a context of ever increasing resolution and size of numerical models and supplementary environmental geospatial data due to various technological improvements.

New motion sensing devices that were introduced in recent years, such as the LEAP motion controller (http://leapmotion.com) and Kinect (http://www.microsoft.com/en-us/kinectforwindows), offer new possibilities for interacting with your software. This raises the question whether and how they can help to simplify the daily operation of water simulation models. Will they result in faster and more intuitive user interfaces for these models? How to effectively integrate these new motion sensor devices with the current or next generation graphical user interfaces? These are the questions that we focus on in this paper.

Developments that go beyond the regular human-computer interaction devices are classified as post-WIMP (window, icon, mouse, and pointer) (van Dam, 1997). Section 2 provides an overview of the main differences between WIMP and post-WIMP devices and their main limitations. It mainly focuses on the analysis of the advantages and disadvantages of the use of LEAP motion controller. Subsequently, we will define the concept of an “interactive model” and summarize the main differences between
conventional and interactive modeling. The new interactive modeling paradigm is demonstrated in Section 6 by means of a software prototype based on the D-Flow FM shallow water numerical model engine (Kernkamp et.al, 2011) integrated with the LEAP motion controller and a 3D graphical user interface built using VTK 3D visualization library (Schroeder and Martin, 2005).

2. NATURAL USER INTERFACES AND 3D MOTION SENSING DEVICES

For decades, the classical computer input devices such as mouse and keyboard have been successfully used in computing. People have (after some initial learning) become accustomed to operating computers using these devices via alphanumeric input and 2D navigation. However, at the same time, they have limited the development of the more intuitive and direct user interfaces, especially when it comes to 3D interaction. (Mann, 2002) uses the term Natural User Interfaces or NUIs as an alternative to Command Line Interfaces (CLI) and Graphical User Interfaces (GUI). The main characteristics of such NUIs are that they are intuitive and direct. Examples of the natural user interfaces include smart boards, tablets, and smartphones. These news types of the graphical user environments and input devices are also referred to as post-WIMP, in a contrast to the classical WIMP (windows, icons, menus, and pointer) graphical user interfaces.

![Figure 1: Hand tracking using LEAP motion controller](http://mashable.com/2013/06/24/leap-motion-airspace/)

Technological developments in the last decade have brought many affordable motion-sensing devices to the market, such as Microsoft Kinect or LEAP motion controller (Figure 1). Still, these devices are used predominantly in the entertainment industry and very little for daily computing activities. Of these devices, the LEAP motion looks especially attractive because of its relatively modest price ($80) combined with micrometer level precision and low computing requirements. So, what are the main stopping factors preventing it from being used daily to operate software applications such as GIS, CAD systems or numerical models? What user interface design principles should we use to make it a success? We agree with (Bret, 2011) that one of the main difficulties of such new motion sensing devices is the problem of not being able to feel what we manipulate (no haptic or tactile feedback). On the other hand, these new devices enable much more degrees of freedom compared to the classical mouse. This can provide many usability benefits, if they are used properly. On the other hand, too many degrees of freedom may also introduce complexity, this is the so-called “degrees of freedom problem” (“Wikipedia: Degrees of freedom problem,” n.d.). As indicated by (Rosenbaum, 2009) this term is used by motor-control researchers for the problem of identifying those few degrees of freedom that characterize a task to be performed in the context of the many degrees of freedom that are in general possible for a physical action. Similar conclusions were also mentioned in the August, 2013 issue of the MIT Technology Review on their experiences with the LEAP motion controller (Metz, 2013).

---

1 Image courtesy of [http://mashable.com/2013/06/24/leap-motion-airspace/](http://mashable.com/2013/06/24/leap-motion-airspace/)
Before we can benefit from these new input devices, it is important to understand how we interact with existing input devices and how they have been integrated in our workflow using the current generation of graphical user interfaces. We may argue that these classical input devices, such as mouse and keyboard are not intuitive and direct. For example, the computer mouse does not provide sufficient level of directness (there is a gap between the horizontal motion of the mouse and the on-screen effect). Although the computer mouse feels quite natural when we use it to operate 2D graphical user interfaces and move around flat maps, it becomes less natural when we have to use it to interact with a 3D graphical user interface or to navigate through 3D space. In these latter cases, the interaction transformation is more complex, and as a result, it becomes less intuitive and more difficult to learn.

A software system feels natural when it is possible to learn its operations through exploration. This is easier to do when most of the actions are made visible; systems that don't follow this principle suffer (Norman, 2010). The fact that kids can successfully use iPad at a very young age (around their first birthday) shows that it provides a very natural and direct interface. You won't see them using mouse at that age (Connell, 2012). This is the result of the seamless integration of hardware and software: users operate the graphical interface using direct interaction on (2D) touch screen, while the software takes full advantage of the multi-touch capabilities of the hardware.

3. INTERACTIVE MODELING

According to (Oreskes et. al., 1994) the primary value of models is heuristic. In other words: models are useful for exploration purposes but not susceptible to proof. We suggest to use the term Interactive Modeling as the way to operate numerical models where the user is in control of the model state at every time step (Donchyts et. al. 2013). To this end, direct access to the internal state of the model is required (by means of get/set operations on variable values).

![Figure 2: The main elements of the interactive models.](image)

Furthermore, the model should provide access to general run-time operations (such as initialize, compute next time step, and finalize). The interactive modeling approach assumes that the model is combined with a graphical user interface, which preferably involves 3D visualization. Via this user interface, the user has full control over the model state and settings, which enables the exploration of the simulated physical processes, see Figure 2.
Turning an existing (conventional) model engine into an interactive one does not need to be very evasive and starts from a few basic principles. First, the interactive engine does not run the model from start to end, but it stays always available in memory. Second, while in memory, it can be fed by new state change requests at any time, preferably using one of existing model API conventions, such as BMI (Peckham, 2010). These changes may include new boundary conditions, adjustments of process settings and state variables, such as bathymetry, roughness or even adjustments to the computational domain grid.

Additionally, the Hollywood principle (“Don’t call us, we call you”) is applied, which allows instructions to pass from the ‘outside’ into the model engine (modular approach), instead of the model engine taking all necessary actions on its own initiative (monolithic integration of engine and user interaction). This enables new ways to use numerical models, for instance, by implementing advanced computer graphics libraries on top of a simulation engine with a well-defined API, one can generate photorealistic visualization of simulation results even during a model run.

4. REDUCE DEGREES OF FREEDOM TO IMPROVE USABILITY

One of the techniques that we incorporated into the design of the graphical user interface to be used with the LEAP motion controller is the reduction of DOFs for performing an action on virtual 3D objects. This resembles the "learning and optimal control" hypotheses used in the motion control sciences ("Wikipedia: Degrees of freedom problem," n.d.): first reduce the DOFs by stiffening the musculature in order to have tight control, then gradually "loosen up" and explore the available DOFs as the task becomes more comfortable, and from there find an optimal solution. An example of this principle is the use of Shift + drag in drawing programs. After Shift is pressed - the movements are allowed only in vertical and horizontal directions.
The same principle is used in our demonstration software. For example, in order to visualize cross-sectional slices through water bodies, the DOFs are reduced to only the horizontal positions of two fingers: the finger locations are projected onto the water surface or ground surface (Figure 4) and the line through these points defines the line of the cross-section. This technique can be further fine-tuned by combining it with other software engineering tricks like snapping of the objects, drawing of additional elements such as projected locations of the fingers to the objects and so on. While experimenting with the LEAP motion we found that it is very important to always draw the location of all visible fingers in the 3D scene, even when they are not used to perform the real action. This improves contextual association between the finger movement by the user and the action performed on the screen.

5. MIXING KEYBOARD, MOUSE AND MOTION SENSOR DEVICES

A few “show stopper” factors make the use of devices such as LEAP motion unnatural. It is quite difficult and unnatural to simulate discrete actions such as mouse click or key press using such continuous motion sensing device. In addition, it can be quite tiring to keep your hands in the air for a long time. Therefore, we suggest using such controllers in combination with traditional discrete input devices such as mouse buttons and keyboard keys. This combination is also necessary to reduce the “gorilla arm” effect mentioned in (Saffer, 2008), namely humans weren’t meant to do many tasks with hands up in front of their bodies for a long period of time. User interface action where keyboard or mouse would be preferable are: grabbing and dragging objects in 2D user interface, and turning on/off specific action (e.g. translate, rotate, scale (Brouet et.al., 2013)). User interface actions where the use of the LEAP motion controller would be preferable are: rotating or navigating a 3D scene, selecting object(s) in 3D space, performing continuous actions involving vertical coordinates (e.g. tilt up or down), and performing actions that involve multiple fingers (e.g. pinch-to-zoom).

6. EXAMPLE: INTERACTIVE D-FLOW FM MODEL

To test the LEAP motion controller we developed a prototype interactive 3D user interface for the new unstructured Delt3D-FLOW kernel. The prototype software uses the new D-FLOW Flexible Mesh numerical model (Kernkamp et.al., 2011), being developed at Deltares in frame of the Next Generation Hydro Software Project. The model can currently simulate hydrodynamics including salinity transport. To achieve an interactive modeling experience, the model engine was turned from a stand-alone executable into a library. Additionally, a very thin Application Programming Interface was implemented following the concepts of the basic model interface (BMI) (Peckham, 2010), which enables introspection. The simple prototype graphical user interface was developed to integrate the model
engine with a scientific visualization tool kit. The user interface includes integration with the LEAP motion controller to perform different actions in 3D. The user can make changes to the model state variables during the simulation. This could for instance be used in the design process of dikes and harbors, such that the effects of the bathymetry and geometry changes can be immediately seen in the model results.

The user interface can be used with or without the LEAP motion controller. When LEAP motion controller is detected a number of additional features become automatically available to the user (Figure 5). The current version of the prototype includes the following features where motion tracking is used: visualize the position of hands and fingers; visualize streamlines originating from the fingertips; draw a cross-section between two fingertips; select grid cells between two fingers; change bathymetry; change water level.

A typical workflow involving changes in bathymetry includes the following steps: 1) switch to the cell selection mode using keyboard (G) 2) make selection using 2 fingers, projected to the 2D model grid 3) switch to the bathymetry change mode using keyboard (D) 4) change the elevation of the selected cells by moving the hand up (note that only changes along Z axis are observed in this mode), and 5) complete changes using keyboard (D).

It is worth mentioning that the LEAP SDK provides a very intuitive, well-documented API. This makes the use of the device very easy for software developers. It was possible to connect it to our software within a matter of a few hours. Most of the development time went in learning the visualization libraries (VTK) and the parallelization of the code. The later was necessary to minimize the impact of the visualization routines on the physical simulation. We were able to decrease the visualization overhead to almost zero on common multi-core CPU hardware.

7. CONCLUSIONS AND DISCUSSION

In this paper, we have looked at how motion sensor input devices can be used in a combination with an interactive water and salinity transport simulation model. An overview of the trends in post-WIMP or NUI design has been given and we discussed the main advantages and disadvantages of the motion sensor controllers.

Since the LEAP controller is a relatively new device it should be used with caution: use it only where it is beneficial; in all other cases, the continued use of keyboard and mouse is preferable. Some of the user actions may be duplicated such that the LEAP controller can be used interchangeably with the mouse or keyboard. This is actually similar to how keyboard and mouse are used together (e.g. one
can select a word in Microsoft Word by double-clicking with the mouse or by pressing Ctrl+Shift+Right/Left on the keyword).

The new interactive modeling paradigm enables explorative use of numerical models. A clear benefit of the new approach is that it greatly reduces the feedback time to the user. This allows for a much better understanding of the processes simulated by the model, and the associated model limitations. The user focuses on the physical processes that are simulated instead of the file formats and post-processing tools - what you see is what you model (WYSIWYM).

In this paper, we highlighted a number of methods that can be used to improve the overall usability when working with a combination of classical and new input devices. We conclude that the users can greatly benefit from the new motion sensing devices in their daily computing tasks, especially to operate 3D graphical user interfaces. Depending on the user action in the user interface, we should analyze if it is beneficial to perform this action using motion-sensing device only or in a combination with mouse and keyboard.

These techniques have been implemented in an interactive user interface for D-FLOW Flexible Mesh. A video of this prototype of an interactive motion-controlled user interface is available on YouTube: [http://bit.ly/interactive_modeling](http://bit.ly/interactive_modeling). The binaries and demo data of the prototype are available upon request. Note that the prototype requires a LEAP controller to be connected to the computer to experience all features discussed in this paper.

8. REFERENCES


Norman, D.A., 2010. The way I see it: Natural user interfaces are not natural. interactions. doi:10.1145/1744161.1744163


