Gestures vs. Postures: 'Gestural' Touch Interaction in 3D Environments

Tobias Isenberg
DIGITEO/CNRS/INRIA, France
University of Groningen, the Netherlands
isenberg@cs.rug.nl

Mark Hancock
University of Waterloo, Canada
mark.hancock@uwaterloo.ca

CHI'12, May 5–10, 2012, Austin, Texas, USA.
ACM 978-1-4503-1016-1/12/05.

Abstract
We revisit the concept of a 'gesture' and discuss it, in particular, in the context of 3D interaction. We argue that one should distinguish between postures, quasi-postures, and *gestures* and that the first two should be (and are currently) preferred in 3D interfaces for directly-manipulative interaction with the environment in user-maintained modes. In scientific visualization we saw use cases of quasi-postures, suggesting that they are one way to re-use the same input space in order to facilitate several means of exploration.

Author Keywords
Gestures, (quasi-)postures, direct-touch interaction, direct manipulation, 3D interaction, scientific visualization.

Introduction
Multi-touch interaction has become one of the most prevalent metaphors for interacting with a variety of different devices including smart phones, tablet computers, tabletops, and wall displays. It also has been applied to numerous different domains, including gaming, museum exhibits, psychotherapy [18], 3D scene design [22], and interactive scientific visualization. Not only in this latter domain (our own background and area of interest) but also more generally in most touch-interactive environments we observed a frequent naming of (multi-)touch 'gestures' as one of the reasons that makes direct-touch interaction intuitive and natural. The term 'gesture' and its use in the context of touch

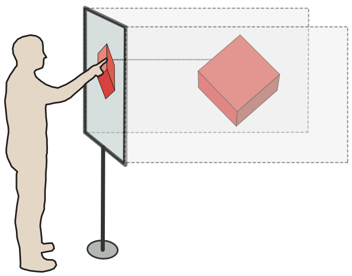

Figure 1: In touch-driven scientific visualization settings, the input provided on the 2D touch-sensitive display has to be mapped to 3D manipulations. Moreover, multiple different exploration techniques such as data navigation, data manipulation, data selection, cutting plane placement and manipulation, data probing, seed particle placement, etc. have to be accessible from within the same input space and researchers need to be able to quickly and intuitively switch between these different interaction techniques [20].

interaction, however, is often not clearly defined or well understood which can lead to confusion when discussing direct-touch interaction techniques. We found this confusion, in particular, to be relevant in 3D interaction due to its focus on the manipulation of objects or data in 3D space from the 2D touch surface (Fig. 1). In this paper we thus attempt a definition of terms related to 'gestures' to guide the further discussion of their role, properties, and applicability in 3D interaction. We discuss *gestures*,[1] related terms, and interaction properties in the context of general touch-based interaction with 3D environments and, specifically, environments for scientific visualization to facilitate the creation of well-designed 3D data manipulation techniques.

'Gestural' Interaction

In touch interfaces we face the challenge that the vocabulary of the primary means of input is limited: touch surfaces typically register touch down, touch move, and touch release events. In this respect they resemble mouse-based interfaces where a similar limitation of a few fundamental event types exists. In both cases, thus, we need to specify additional modes to clearly state what effect a given interaction event should cause and, ultimately, to be able to create flexible interfaces. While mouse-based interfaces can rely on, in particular, a number of mouse or keyboard buttons to specify such modes, such buttons are not readily available in touch-based interaction. Therefore, interface designers often rely on 'gestures' to specify interaction modalities.

'Gestural' interaction is often employed in touch-based (and related) interfaces. For example, researchers have established gesture catalogs (e. g., [33]) that support a variety of applications, have discussed object-dependent gestures (e. g., [26]), or distinguish between the shape or size of the touching finger/hand/object (e. g., [21, 37]). Others have extracted intuitive gestures through demonstration [8, 10, 27, 36], have examined cooperative gestures performed by groups of people [25], or have focused on multi-modal interaction with gestures (e. g., [32]). However, it seems that most authors employ an implicit notion of 'gestures'.

A notable exception are Baudel and Beaudouin-Lafon [1] who state that gestures have a start position, a dynamic phase, and an end position. The start and end positions do not only refer to positions of a hand in space but also include the notion of finger/wrist *configurations* and their concept of a 'gesture' clearly contains the notion of a dynamic motion. This notion is seen in contrast to "steady positions" that other systems recognize which are considered to be less powerful than gestures. Interestingly, this gesture concept is thought to specify (a series of) discrete *lexical entities* and the use of gestures as *commands*, i. e., the discrete change into different statuses or modes.

A related concept is discussed by Wu et al. [38] who emphasize the design principles of *registration* and *relaxation*. Gesture registration relies on a *posture* to start the recognition, as a "functional transition of a tool from one interaction style to another," similar to Baudel and Beaudouin-Lafon's [1] notion of gestures as commands. The second concept, the *relaxation* phase, relaxes the posture requirements to allow people to perform a gesture more freely. It also emphasizes that motion is an integral part of the gesture, a notion also emphasized by others [5]. However, Wu et al. also mention a distinction between *continuous* and *discrete* gestures, probably referring to their *postures* being part of the 'gesture' concept. In fact, they hint at "a more sophisticated" gesture definition which "considers both the hand posture and dynamic actions that occur immediately after the posture is recognized, within a predefined time window."

[1] To distinguish the different meanings, we use 'gesture' for the traditional (general) concept of interaction specification, while *gesture* refers to more narrow definition given later in this paper. If gesture is used without quotation marks and in normal roman type, then the classification is either not clear or we discuss related work which does not make this distinction.

We base our own definitions on a similar notion and discuss why such extended postures can be advantageous.

Wobbrock et al. [36] explore the 'gesture' concept by describing a taxonomy which classifies gestures according to their *form*, their *nature*, their *binding*, and their *flow*: The form describes whether a hand *pose* is static or dynamic both in its location and appearance. The nature describes whether a gesture visually depicts a symbol, acts physically on objects, indicates a metaphor, or has no related connection to the object it references. A gesture's binding refers to a gestures location with respect to object or world features. Finally, the flow can be continuous or discrete, i. e. the invoked response can occur *during* or *after* the gesture. This taxonomy adds several important aspects of gestures. The form dimension both emphasizes the notion of a configuration or posture as the basis of a gesture as well as the temporal aspect—gestures are defined as an action that takes a given time during which the finger(s)/hand(s) can be moved. An aspect added by Wobbrock et al. [36] is the dependence of gestures on a frame of reference—a specific object or the world in general. Another important aspect added in Wobbrock et al.'s [36] taxonomy is that gestures can already have specified an action (mode) while they are still being performed—leaving the continuing motion as a means to provide further input.

One other observation by Wobbrock et al. [36] that is important for our discussion is that people, when proposing gestures for certain actions, think about the gestures using metaphors such as the computer desktop and/or program windows. This notion, one could speculate, may have been prompted by the experiment's design which used two-dimensional "blocks" that were manipulated by "commands" on a two-dimensional surface. This design places much emphasis on abstract and discrete events that are common in the computer desktop such as opening, activating, and closing windows, undo/redo, or minimize/maximize. In fact, more than 70% of Wobbrock et al.'s [36] commands can be seen as being such discrete tasks. In the context of interacting with 3D objects or spaces, however, we tend to focus, e. g., on the continuous direct manipulation of objects. Thus, we now take a closer look at 'gestural' interaction with 3D environments and the 'gesture' concept.

Postures vs. *Gestures*

To start this discussion let us first precisely define what we understand a 'gesture' to be. Based on the notions by Baudel and Beaudouin-Lafon [1], Wu et al. [38], and Wobbrock et al. [36], we consider a touch ***gesture*** to be

> *a way to invoke **manipulations** in a direct-touch environment that is **started** by touching the surface in an **well-defined initial configuration** and that is **continued** for some time in a **well-defined motion pattern** (incl. the null motion) during which the configuration may change.*

This definition captures the previously mentioned notions of *configurations* and a *dynamic phase* [1, 36] as well as *gesture registration* and *relaxation* [38]. It is well suited to specify commands such as described by Wobbrock et al. [36] for 2D settings. In most 3D environments for which touch interfaces have been developed so far, however, the interaction focuses on the change of the camera view or the direct manipulation of objects. For example, in a 3D visualization of a brain the main operations are translations, re-orientations, resizing, or the translation of cutting planes. This concentration on the directly manipulative interaction brings with it an exploratory nature and the need for mental focus on the object manipulation: one primarily thinks about the object/data/scene that one is manipulating while the interaction to enable this manipulation has to become second nature. Therefore, requiring people to remember and employ well-defined motion patterns (which also delay the invoking of the intended manipulations) can be prohibitive to an intuitive interaction. Moreover, the *gesture* as in the definition can only specify discrete manipulations due to its reliance on a well-defined motion pattern.

Therefore, we need to ask if the well-defined motion pattern is essential to specify the intended manipulation. Instead, we could relax this constraint which would free up a potential motion for the specification of input parameters, to support a continuous and direct interaction with 3D objects or the 3D space. We thus define a touch **posture** to be

> *a way to invoke* ***manipulations*** *in a direct-touch environment that is* ***characterized*** *by touching the surface in an* ***well-defined initial configuration*** *whose effect can be* ***parametrized*** *by a subsequent dynamic action.*

The essential difference between *gestures* and postures is whether the dynamic action (e. g., a motion or a change of the initial configuration) characterizes the manipulation or not. The notion of a posture has the advantage for 3D interaction that directly-manipulative interactions can be specified while the posture is active. With *gestures*, in contrast, a direct manipulation of the system or objects has to be specified *after* the *gesture* has been fully and uniquely recognized. Both definitions thus differ in the type of modes that they encourage. Postures tend to specify *user-maintained modes* since they are only active as long as (a part of) the posture/configuration is maintained (often the dominant hand that specifies the motion). *Gestures* often specify *system-maintained modes* (or status changes) because a motion pattern for specifying parameters can only be started *after* the *gesture* is completed, while the system maintains the interaction mode. Consequently, *gestures* are also more prone to modal errors [4, 29].

For both definitions it is important to clarify the term *configuration*. We understand this concept relatively broadly to include the number of fingers (single- or bi-manually), their (relative) positions, the touching size or shape, touch IDs, and even marker properties such as 2D barcodes. In this notion we include the touched element on the surface such as a button or a screen region, as specified by Wobbrock et al.'s [36] *binding* category. Thus it is possible to use the same configurations on different screen regions/ bindings to define different postures or *gestures*. For both *gestures* and postures, however, the configurations have to be learned by users, but a configuration's binding can also serve as a reminder of a *gesture's*/posture's function. In contrast, a *gesture's* motion pattern has to be learned unless it resembles a shape associated with the action (e. g., scribble to delete)—the motion in postures directly affects object parameters whose change is typically visible.

As noticed before by Wu et al. [38], sometimes it is not only the static configuration that characterizes a posture but in some cases people also use "dynamic actions that occur immediately after the posture is recognized." We thus extend the posture concept to define a **quasi-posture** as a

> *a* ***posture*** *whose initial configuration is augmented with a* ***brief initial dynamic action*** *but where this action's continuation is also used to parameterize the effect.*

The brief initial dynamic action can be captured in the form of a few samples of an initial motion such as the initial direction of motion, which it typically directly tied to the effect that is intended with the quasi-posture. This connection and the briefness makes quasi-postures different from *gestures* whose motion paths are typically complex and not tied to the *gesture's* effect. Quasi-postures are similar to postures in that the configurations have to be learned and in that they can be used to define user-controlled interaction modes due to the briefness of the dynamic action.

'Gestural' 3D Direct-Touch Interaction

The previous discussion of the 'gesture' concept arose from our work with 3D environments and, specifically, from touch interaction with scientific visualizations. Next, we explore

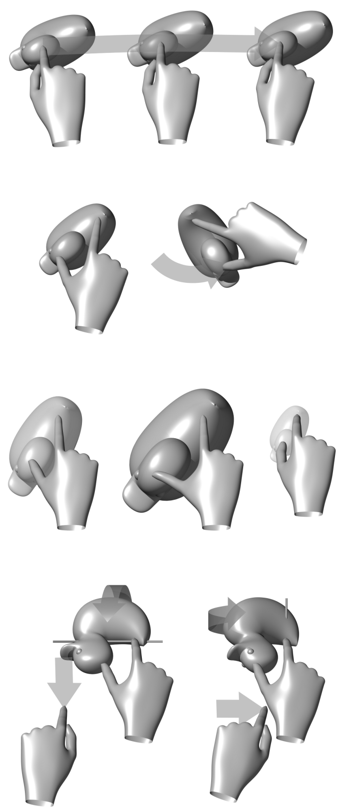

Figure 2: Example for posture-controlled direct manipulation of 3D objects: sticky tools [17, 18]. The one-, two-, or three-finger configurations uniquely determine the interaction modalities, and the following motion directly parametrizes the object manipulation. Images from [18], © ACM 2010, used with permission.

the three different 'gesture' notions by first analyzing a number of general 3D touch interaction techniques and later discussing examples from SciVis where a continuous and direct exploration is essential. We specifically examine which concept is used primarily and for what purpose.

General 3D Interaction Techniques
Many of the published 3D touch interaction techniques do not use *gestures* but, in fact, rely entirely on finger configurations to specify the interaction modality so that they should be seen as postures. Typically, their configurations consist of one or a few contact points that touch the objects or widget elements to be manipulated, without requiring motion patterns (Fig. 2). This is true for 3D RST interaction [28] (and, in fact, also the common 2D pinching technique [19]), shallow-depth interaction [16], sticky tools [17, 18], surface physics [34, 35], DabR [9], and *z*-positioning [24].

A further group of techniques also largely relies on (quasi-)postures but adds one or two *gestures* for very specific purposes. For example, Benko et al.'s [2] balloon selection uses postural interaction for the entire 3D placement process and only uses a tap *gesture* to for the final selection action. Similarly, the tBox manipulation [8] relies almost exclusively on postures and one quasi-posture that distinguishes between two-finger RST (both fingers moving) and pivoted rotation (one finger moving; static finger is pivot point). The only *gestures* in tBox are a double-tap for (un-)selecting objects and a double-tap on one of the widget's edges to invoke translation in the plane as opposed to along the initially selected second axis. Toucheo's [14] widget also mostly employs postures, its only *gesture* is a double-tap to raise objects to a pre-defined level.

Most of Eden's [22] 3D manipulation 'gestures' are also postures according to our definition. The only exceptions are the one-finger taps (null motion *gestures* in zero time) to place a pivot point for camera rotation or to add new objects to the scene. However, adding objects to the scene can also be seen as a posture without a continuing direct manipulation which could potentially be enhanced by interpreting a subsequent motion as a translation on the surface of the model (seemingly not done by Kin et al. [22]).

A nice example of postural interaction with true volumetric displays in which the postures are not only defined on the display surface but in 3D space was described by Grossman et al. [12]. Similar to Eden [22], the vast majority of 'gestures' are postures, the only exceptions being tapping and a trigger *gesture* for certain selection events.

We found one example that employs *gestures* to a larger degree: Navidget [15]. While its selection interaction has both *gestural* (completing a circle) and postural (drawing a unique selection region) aspects, the widget itself is invoked with a tap-and-hold *gesture*. *Gestures* are also used to control the widget including touching the peripheral ribbon (to switch between back and front views) and moving horizontally/vertically (to exit the resize mode). However, the interaction also has postural (direct-manipulative) aspects in how the view direction is continually controlled.

Touch Interaction with 3D Scientific Visualization
In addition to the issue of mapping 2D input to 3D manipulations, touch-based interaction with scientific visualization (SciVis) adds several further challenges such as the integration of multiple interaction techniques in the same input space as well as precise control [20]. We examine several tools w.r.t. their use of *gestures*, postures, and quasi-postures to understand how they employ these concepts and if it differs from the basic 3D techniques discussed above.

Fu et al.'s [11] astronomy visualization system tackles the interaction with spatial and scale space. They use postures to specify interaction modes based on numbers of contact points and, for two simultaneous contact points, their rel-

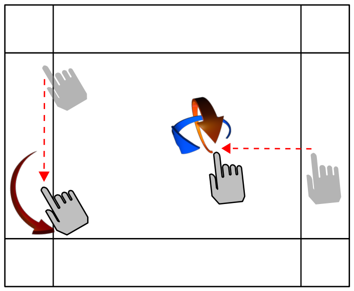

Figure 3: Example from FI3D for quasi-posture-controlled interaction [39]: depending on the *initial* motion along or perpendicular to the widget frame, a rotation around the *z*-axis or a trackball rotation is initiated, respectively. Image from [39], © IEEE Computer Society 2010, used with permission.

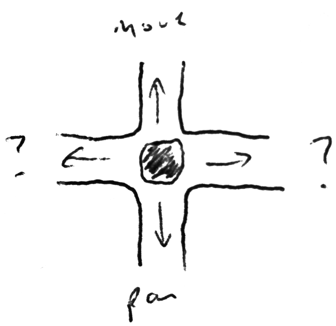

Figure 4: This example shows a low-fidelity prototype of how a directional slider could be used to provide visual feedback for a quasi-posture to enable several modes of interaction.

ative position w.r.t. each other. Tapping is the only *gesture* that is used—for selecting items (stars). A related approach used for astronomical data is FI3D [39] which also provides means to interact with the space—rather than with single objects. Here, one-/two-finger postures and their location w.r.t. a widget are used to specify modes, some of these being quasi-postures that differentiate according to the direction of movement w.r.t. the widget frame (Fig. 3).

Sultanum et al. [30, 31] described two systems for exploring geology. In their initial system [30] they use one-finger orbiting, two-finger RST, and axis-aligned cutting with two fingers on the data volume's corners which are all postures. The interaction in an extended system [31] are also all postural (including the use of tangibles). Interesting is the distinction between the four-finger splitting and the four-finger peeling which is done using a quasi-posture that looks at the (initial) relative directions of the points in a cluster.

The systems mentioned so far use regular displays for depicting the 3D data. Two more systems, however, rely on stereoscopic displays. Butkiewicz and Ware's [3] oceanographic visualization combines touch sensing with a depth camera for touch identification and can distinguish two-handed and single-handed-multi-finger postures for 3D positioning. Slice WIM [6, 7], in contrast, relies on touch interaction on a projected surface in its stereoscopic environment and uses postures which are specified w.r.t. a 2D widget.

In summary, in the scientific visualization systems we examined we found only postures and no *gestures* being used for directly-manipulative interaction—the only exception being tap *gestures* for selection. However, it is interesting that two of the six systems also used quasi-postures. The need for using quasi-postures in addition to regular postures may arise due to SciVis' requirement to combine multiple interaction techniques in the same input space and the need for data analysis incl. filtering or clustering.

Implications for Future Interactive Systems

As we have seen in our analysis of both general 3D direct-touch interaction techniques as well as touch-based scientific visualization systems, postures have already been used rather than *gestures* for directly-manipulative control in most cases. This may be due to the implied benefit of user-maintained modes as opposed to system-maintained modes. However, the real challenge—in particular for scientific visualization—is how to combine and integrate many of these interaction techniques such that they can be used within the same 2D input space [20]. In fact, many of the postures we surveyed for specifying 3D interactions conflict with each other by nature, so posture-based systems have to be designed such that those conflicts are avoided.

Quasi-postures are one way to extend the interaction vocabulary without requiring new (and potentially conflicting) configurations, and these should thus be explored to a larger extent. However, we will not be able to cover all necessary interactions with intuitive single-handed quasi-postures, so additional means need to be explored. One option is to use postural bi-manual interaction in which a 'button' is held down with one hand as part of the configuration to specify the main interaction type, while the other (dominant) hand performs a precise posture followed by the direct manipulation. In fact, the button only needs to be held down until the parameter specification has started because at that point the manipulation causes the visualization to change so that visual feedback is provided.

This use of visual feedback can also be developed a step further for one-handed quasi-postures, so that separate modes of interaction could be specified using something akin to a FlowMenu [13] and continued using a posture. For example, an *n*-directional slider widget could be used to initiate *n* different modes of interaction. At $n = 4$ (Fig. 4), this widget would allow different interaction when starting

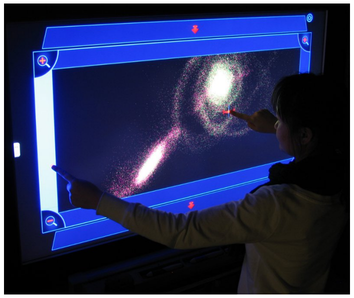

Figure 5: Person interacting with a particle simulation from astronomy using the FI3D widget [39], performing a directly-manipulative interaction. Image from [39], © IEEE Computer Society 2010, used with permission.

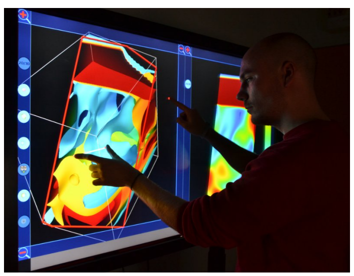

Figure 6: Person interacting with a visualization system for fluid mechanics [23] in which several different postural and quasi-postural interaction techniques are integrated and can be accessed from the same input space. This integration facilitates the exploration of complex 3D data.

the gesture in the up, down, left, and right directions, and once the slider's end is reached the posture would begin.

In addition to the challenge of creating an integrated set of interaction techniques, more work is also needed to understand cases where the distinction between postures and *gestures* is not entirely clear. One such case is single-tapping: it can be a posture to define a location [22] or a *gesture* to define an action [2]. One question is, for example, if such an interaction can only be either a posture or a *gesture* in an integrated interactive system, or if there are also intuitive ways to use both concepts simultaneously.

Conclusion

In this paper we argued that using the term 'gesture' in a generalizing way to describe direct-touch interaction can be problematic, in particular (but not only) in the context of 3D interaction. The reason for this is that 'gestures' can be and often are confused with real *gestures* that require motion patterns that have to be learned, that are typically used to specify status changes or system-controlled modes, and that thus can be prone to modal errors caused by these modes. Instead, (quasi-)postures are often better suited (and typically used) for directly-manipulative interaction in user-controlled modes, in particular, in 3D environments because here the motion that follows a posture can directly control the parameters one is adjusting (e. g., Fig. 5).

Our survey of direct-touch 3D interactions and touch-based SciVis tools showed that postures are indeed overwhelmingly used for direct manipulation—*gestures*, instead, are used for discrete tasks, e. g., selections. We argued, however, that more work is needed to be able to control the many different parameters in a scientific visualization and, thus, we need to find ways that allow us to integrate different posture-based control within the same 2D input space. We are currently in the process of exploring such possibilities for specific scientific domains such as flow visualization [23] and hope that this work (Fig. 6) as well as our present position paper on 'gestural' interaction foster research into direct-touch interaction with 3D scientific data.

Acknowledgments

We thank Björn Hartmann, Petra Isenberg, and Frédéric Vernier for inspiring discussions on the topic.

References

[1] T. Baudel and M. Beaudouin-Lafon. Charade: Remote Control of Objects Using Free-Hand Gestures. *Communications of the ACM*, 36(7):28–35, July 1993. doi> 10.1145/159544.159562

[2] H. Benko and S. Feiner. Balloon Selection: A Multi-Finger Technique for Accurate Low-Fatigue 3D Selection. In *Proc. 3DUI*, pp. 79–86, Los Alamitos, 2007. IEEE. doi> 10.1109/3DUI.2007.340778

[3] T. Butkiewicz and C. Ware. Exploratory Analysis of Ocean Flow Models with Stereoscopic Multi-Touch. In *IEEE Visualization Posters*, 2011.

[4] W. Buxton. Chunking and Phrasing and the Design of Human-Computer Dialogues. In *Proc. IFIP World Computer Congress*, pp. 475–480, 1986.

[5] W. Cleveringa, M. van Veen, A. de Vries, A. de Jong, and T. Isenberg. Assisting Gesture Interaction on Multi-Touch Screens. In *Proc. CHI Workshop on Multi-touch and Surface Computing*, New York, 2009. ACM.

[6] D. Coffey, N. Malbraaten, T. Le, I. Borazjani, F. Sotiropoulos, A. G. Erdman, and D. F. Keefe. Interactive Slice WIM: Navigating and Interrogating Volume Datasets Using a Multi-Surface, Multi-Touch VR Interface. *IEEE Transactions on Visualization and Computer Graphics*, 2012. To appear. doi> 10.1109/TVCG.2011.283

[7] D. Coffey, N. Malbraaten, T. Le, I. Borazjani, F. Sotiropoulos, and D. F. Keefe. Slice WIM: A Multi-

Surface, Multi-Touch Interface for Overview+Detail Exploration of Volume Datasets in Virtual Reality. In *Proc. I3D*, pp. 191–198, New York, 2011. ACM. doi> 10.1145/1944745.1944777

[8] A. Cohé, F. Dècle, and M. Hachet. tBox: A 3D Transformation Widget Designed for Touch-Screens. In *Proc. CHI*, pp. 3005–3008, New York, 2011. ACM. doi> 10.1145/1978942.1979387

[9] J. Edelmann, S. Fleck, and A. Schilling. The DabR – A Multitouch System for Intuitive 3D Scene Navigation. In *Proc. 3DTV*, Los Alamitos, 2009. IEEE Computer Society. doi> 10.1109/3DTV.2009.5069671

[10] M. Frisch, J. Heydekorn, and R. Dachselt. Investigating Multi-Touch and Pen Gestures for Diagram Editing on Interactive Surfaces. In *Proc. ITS*, pp. 149–156, New York, 2009. ACM. doi> 10.1145/1731903.1731933

[11] C.-W. Fu, W.-B. Goh, and J. A. Ng. Multi-Touch Techniques for Exploring Large-Scale 3D Astrophysical Simulations. In *Proc. CHI*, pp. 2213–2222, New York, 2010. ACM. doi> 10.1145/1753326.1753661

[12] T. Grossman, D. Wigdor, and R. Balakrishnan. Multi-Finger Gestural Interaction with 3D Volumetric Displays. In *Proc. UIST*, pp. 61–70, New York, 2004. ACM. doi> 10.1145/1029632.1029644

[13] F. Guimbretiére and T. Winograd. FlowMenu: Combining Command, Text, and Data Entry. In *Proc. UIST*, pp. 213–216, New York, 2000. ACM. doi> 10.1145/354401.354778

[14] M. Hachet, B. Bossavit, A. Cohé, and J.-B. de la Rivière. Toucheo: Multitouch and Stereo Combined in a Seamless Workspace. In *Proc. UIST*, pp. 587–592, New York, 2011. ACM. doi> 10.1145/2047196.2047273

[15] M. Hachet, F. Decle, S. Knödel, and P. Guitton. Navidget for Easy 3D Camera Positioning from 2D Inputs. In *Proc. 3DUI*, pp. 83–89, Los Alamitos, 2008. IEEE Computer Society. doi> 10.1109/3DUI.2008.4476596

[16] M. Hancock, S. Carpendale, and A. Cockburn. Shallow-Depth 3D Interaction: Design and Evaluation of One-, Two- and Three-Touch Techniques. In *Proc. CHI*, pp. 1147–1156, New York, 2007. ACM. doi> 10.1145/1240624.1240798

[17] M. Hancock, T. ten Cate, and S. Carpendale. Sticky Tools: Full 6DOF Force-Based Interaction for Multi-Touch Tables. In *Proc. ITS*, pp. 145–152, New York, 2009. ACM. doi> 10.1145/1731903.1731930

[18] M. Hancock, T. ten Cate, S. Carpendale, and T. Isenberg. Supporting Sandtray Therapy on an Interactive Tabletop. In *Proc. CHI*, pp. 2133–2142, New York, 2010. ACM. doi> 10.1145/1753326.1753651

[19] M. S. Hancock, S. Carpendale, F. D. Vernier, D. Wigdor, and C. Shen. Rotation and Translation Mechanisms for Tabletop Interaction. In *Proc. Tabletop*, pp. 79–88, Los Alamitos, 2006. IEEE Computer Society. doi> 10.1109/TABLETOP.2006.26

[20] T. Isenberg. Position Paper: Touch Interaction in Scientific Visualization. In *Proc. ITS Workshop on Data Exploration on Interactive Surfaces (DEXIS)*, 2011.

[21] T. Isenberg, M. Everts, J. Grubert, and S. Carpendale. Interactive Exploratory Visualization of 2D Vector Fields. *Computer Graphics Forum*, 27(3):983–990, May 2008. doi> 10.1111/j.1467-8659.2008.01233.x

[22] K. Kin, T. Miller, B. Bollensdorff, T. DeRose, B. Hartmann, and M. Agrawala. Eden: A Professional Multitouch Tool for Constructing Virtual Organic Environments. In *Proc. CHI*, pp. 1343–1352, New York, 2011. ACM. doi> 10.1145/1978942.1979141

[23] T. Klein, F. Guéniat, L. Pastur, F. Vernier, and T. Isenberg. A Design Study of Direct-Touch Interaction for Exploratory 3D Scientific Visualization. *Computer Graphics Forum*, 31(3), June 2012. To appear.

[24] A. Martinet, G. Casiez, and L. Grisoni. 3D Positioning Techniques for Multi-Touch Displays. In *Proc. VRST*, pp. 227–228, New York, 2009. ACM. doi> 10.1145/1643928.1643978

[25] M. R. Morris, A. Huang, A. Paepcke, and T. Winograd. Cooperative Gestures: Multi-User Gestural Interactions for Co-Located Groupware. In *Proc. CHI*, pp. 1201–1210, New York, 2006. ACM. doi> 10.1145/1124772.1124952

[26] M. Nijboer, M. Gerl, and T. Isenberg. Exploring Frame Gestures for Fluid Freehand Sketching. In *Proc. SBIM*, pp. 57–62, Goslar, Germany, 2010. Eurographics. doi> 10.2312/SBM/SBM10/057-062

[27] C. North, T. Dwyer, B. Lee, D. Fisher, P. Isenberg, K. Inkpen, and G. Robertson. Understanding Multi-touch Manipulation for Surface Computing. In *Proc. Interact*, pp. 236–249, Berlin, 2009. Springer-Verlag. doi> 10.1007/978-3-642-03658-3_31

[28] J. L. Reisman, P. L. Davidson, and J. Y. Han. A Screen-Space Formulation for 2D and 3D Direct Manipulation. In *Proc. UIST*, pp. 69–78, New York, 2009. ACM. doi> 10.1145/1622176.1622190

[29] A. J. Sellen, G. P. Kurtenbach, and W. A. S. Buxton. The Prevention of Mode Errors Through Sensory Feedback. *Human Computer Interaction*, 7:141–164, June 1992. doi> 10.1207/s15327051hci0702_1

[30] N. Sultanum, E. Sharlin, M. C. Sousa, D. N. Miranda-Filho, and R. Eastick. Touching the Depths: Introducing Tabletop Interaction to Reservoir Engineering. In *Proc. ITS*, pp. 105–108, New York, 2010. ACM. doi> 10.1145/1936652.1936671

[31] N. Sultanum, S. Somanath, E. Sharlin, and M. C. Sousa. “Point it, Split it, Peel it, View it”: Techniques for Interactive Reservoir Visualization on Tabletops. In *Proc. ITS*, pp. 192–201, New York, 2011. ACM. doi> 10.1145/2076354.2076390

[32] E. Tse, S. Greenberg, C. Shen, and C. Forlines. Multimodal Multiplayer Tabletop Gaming. *ACM Computers in Entertainment*, 5(2):12/1–12/12, Apr.–June 2007. doi> 10.1145/1279540.1279552

[33] C. Villamor, D. Willis, and L. Wroblewski. Touch Gesture Reference Guide. Published online, available from http://www.lukew.com/touch/, 2010.

[34] A. D. Wilson. Simulating Grasping Behavior on an Imaging Interactive Surface. In *Proc. ITS*, pp. 137–144, New York, 2009. ACM. doi> 10.1145/1731903.1731929

[35] A. D. Wilson, S. Izadi, O. Hilliges, A. Garcia-Mendoza, and D. Kirk. Bringing Physics to the Surface. In *Proc. UIST*, pp. 67–76, New York, 2008. ACM. doi> 10.1145/1449715.1449728

[36] J. O. Wobbrock, M. R. Morris, and A. D. Wilson. User-Defined Gestures for Surface Computing. In *Proc. CHI*, pp. 1083–1092, New York, 2009. ACM. doi> 10.1145/1518701.1518866

[37] M. Wu and R. Balakrishnan. Multi-Finger and Whole Hand Gestural Interaction Techniques for Multi-User Tabletop Displays. In *Proc. UIST*, pp. 193–202, New York, 2003. ACM. doi> 10.1145/964696.964718

[38] M. Wu, C. Shen, K. Ryall, C. Forlines, and R. Balakrishnan. Gesture Registration, Relaxation, and Reuse for Multi-Point Direct-Touch Surfaces. In *Proc. TABLETOP*, pp. 185–192, Los Alamitos, 2006. IEEE Computer Society. doi> 10.1109/TABLETOP.2006.19

[39] L. Yu, P. Svetachov, P. Isenberg, M. H. Everts, and T. Isenberg. FI3D: Direct-Touch Interaction for the Exploration of 3D Scientific Visualization Spaces. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1613–1622, Nov./Dec. 2010. doi> 10.1109/TVCG.2010.157