



Audio Engineering Society Convention Paper

Presented at the 130th Convention
2011 May 13–16 London, UK

The papers at this Convention have been selected on the basis of a submitted abstract and extended precis that have been peer reviewed by at least two qualified anonymous reviewers. This convention paper has been reproduced from the author's advance manuscript, without editing, corrections, or consideration by the Review Board. The AES takes no responsibility for the contents. Additional papers may be obtained by sending request and remittance to Audio Engineering Society, 60 East 42nd Street, New York, New York 10165-2520, USA; also see www.aes.org. All rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.

Gestures to Operate DAW Software

Wincent Balin¹ and Jörn Loviscach²

¹*Universität Oldenburg, D-26111-Oldenburg, Germany*

²*Fachhochschule Bielefeld (University of Applied Sciences), D-33602-Bielefeld, Germany*

Correspondence should be addressed to Jörn Loviscach (joern.loviscach@fh-bielefeld.de)

ABSTRACT

There is a noticeable absence of gestures—be they mouse-based or (multi-)touch-based—in mainstream digital audio workstation (DAW) software. As an example for such a gesture consider a clockwise O drawn with the finger to increase a value of a parameter. The increasing availability of devices such as smartphones, tablet computers and touchscreen displays raises the question in how far audio software can benefit from gestures. We describe design strategies to create a consistent set of gesture commands. The main part of this paper reports on a user survey on mappings between 22 DAW functions and 30 single-point as well as multi-point gestures. We discuss the findings and point out consequences for user-interface design.

1. INTRODUCTION

A major advantage of using gestures is the removal of visually obtrusive user interface elements (widgets) such as buttons, dials and context menus usually needed to invoke a function or control a parameter of the software. This helps to remove clutter; the graphical user interface can focus on the content. Gestures can be a blessing on a space-constrained screen, which explains their popularity on touch-enabled mobile devices. Since many of the current touch-enabled mobile devices lack a standard computer keyboard, shortcut keystrokes—as used regu-

larly with desktop computers—are not available. On these devices, gestures also benefit from the bimanual operation and from the direct interaction with the screen as opposed to an execution of gestural strokes with an attached computer mouse.

Gestures are common on mobile devices but can also be found in standard desktop software. In 2001, Opera Software enhanced their web browser by mouse gestures [21]. In the meantime, a range of gesture extensions has become available for Mozilla Firefox [18] and Microsoft Internet Explorer, see for instance [9]. Autodesk Maya, a 3D modeling and an-

imation software package, goes even further by using marking menus [14] by default, an input method related to abstract, non-metaphorical mouse gestures.

Classic digital audio workstation (DAW) software, however, seems to shun gestures altogether. Today, most music software with gesture support runs on Apple iOS. For instance, in StepPolyArp [6], a MIDI arpeggiator software, the user places his or her finger on the rectangle representing a trigger event and then swipes it down to delete it. In a similar fashion, Apple GarageBand for iPad [3] employs basic gestures for zooming and parameter control—in addition to using the touch screen as virtual piano keyboard, guitar, or drum set, which may not be considered a use of gestures.

Considering these developments and the users' growing awareness of gesture control through mobile devices, it seems only a matter of time until gestures are widely adopted in *desktop* DAW software as well. A quick swipe over a note accomplishes its function much faster than going to a tool panel, choosing an eraser tool, pointing the cursor to the note and doing a click—and then most probably switching back to a pointing tool on the tool panel to prevent further, unwanted deletions. Even without using quantification methods such as GOMS [5] it is evident that stepping through the list of actions mentioned above takes more time and mental effort than executing a single gesture.

An alternative to clicking icons in the toolbox would be using keyboard shortcuts to switch tools, keeping one hand on the mouse and the other on the computer keyboard. But this requires memorizing the shortcuts; it still incurs a sequence of three actions (key, mouse, key) rather than one gesture; and it requires the user to temporarily shift his or her focus from the screen to the keyboard and back, at least occasionally.

There is a technically-induced difference between mouse gestures, single-finger and multi-touch gestures. Some finger gestures may readily be converted to mouse gestures. It is, however, difficult to hold a mouse button pressed while performing a mouse gesture. Multi-touch gestures open a new realm of expression, but require specific display hardware. Assuming that an ecosystem [26] of consistent choices that extend from mobile devices to desktop comput-

ers is needed, this work looks at all these three kinds of gestures and treats them on the same footing.

Gestures need to be picked judiciously. One option is to create “intuitive” ones, that is: ones that leverage cultural, sensimotor or even innate knowledge [19]. For instance, the usual “pinch” gesture for zooming out with two fingers moving on the screen may stem from extending or compressing tissue in the real world. An entirely different approach is to create gestures that are entirely symbolic but can be learned progressively. Section 2 will elaborate on existing discussions of such issues.

The objective of this work is to examine which gestures are the most promising ones to be used in DAW software. We have built a list of common tasks and collected a repertoire of mouse/single-touch/multi-touch gestures, which we subjected to a first user survey. Section 3 introduces the survey-based approach we chose and presents the results of the initial survey that was conducted as a pre-test. This survey helped to filter the initial repertoire but also to add a small number of further gestures. This refined repertoire was subjected to a second survey, which is covered in Section 4, together with its results. Section 5 concludes this paper and provides and outlook on future work.

2. RELATED WORK

Gestures can be input by mouse, touch, visual recognition and other techniques. Research done on gestural input in the past 40 years [12] has emphasized vision-based techniques and “semaphoric” gestures, that is: gestures from a dictionary of abstract symbols. Already in their 1985 paper [4] on touch tablets, Buxton and his coauthors hinted at the possibilities of multi-touch and even of pressure-sensitive devices—while pointing out the increased importance of visual and audio feedback due to the lack of haptic feedback.

Modern developments encompass a range of software development frameworks for multi-touch interaction [10], not mentioning the existing support in operating systems such as Apple iOS, Microsoft Windows 7, and Google Android. Already Windows XP edition for Tablet PC contained a recognizer for gestures executed with the stylus [17]. Along with

the technology, critique has evolved. Norman [20] finds a lack of guidelines and an ignorance of conventions and established findings of human-computer interaction in current touch-based gestural interfaces. For instance, it is hard to guess the “magical” gesture that will perform a specific function; radio buttons and checkboxes no longer look different; it is easy to trigger unwanted actions but hard to undo accidental selections.

Thus, the design of such interfaces is still far from optimal. Wu et al. [25] propose to move away from ad-hoc designs to an approach based on the principles of gesture registration (e.g., switching the role of a stylus from a selection lasso to a handle used for dragging), gesture relaxation (e.g., choosing a less awkward pose to continue after the initial recognition), and gesture and tool reuse (e.g., using the same gesture for different commands depending on the context). Referring to gestures in particular, George and Blake introduce objects, containers, gestures, and manipulations as general concepts in human-computer interaction [7].

In a more concrete way, other authors have introduced taxonomies to describe the vast range of touch and multi-touch gestures. Wigdor [23] distinguishes 16 different types: Are gestures executed with a single finger, several fingers, a single shape (such as the palm of the hand) or with multiple shapes? Does the palm of the hand trace a path or not? Does the shape change or not? Kammer et al. [11] propose a formalization of multi-touch gestures through different parameters: the shape of the recognized blob, its motion, the “temporal progression” (parallel, successive, or asynchronous), the “focus” (how which target is identified), and “area constraints” (such as join and spread).

Instead of providing users with a fixed set of gestures, one can try to learn from observation [1]. The study by Wobbrock et al. [24] finds that users do not distinguish much between single- and multi-touch gestures, they tend to use one hand rather than both hands, are strongly influenced by what they have learned from desktop computers, and that some commands tend to be represented with a large range of gestures. For the latter commands, regular on-screen widgets may be suited better than gestures. The same work introduces another taxonomy: static or dynamic “form” concerning pose

and path; a “nature” ranging from physical to abstract; a “binding” ranging from object-centric to world-independent; and a “flow” ranging from discrete (response occurs after gesture) to continuous. This taxonomy seems to be the most promising for the application in DAWs. The “binding” can for instance distinguish if a gesture concerns a snippet of audio or a control knob or a general action such as saving the current project. The “flow” can indicate if a gesture invokes single action such as deleting a note or if it leads to continuous feedback such as when interactively setting a parameter.

Referring to the user’s intuition [19] is central to most efforts to a principled design of gestural interfaces. In a decidedly different approach, Kurtenbach and Buxton propose to use polygonal lines to cross a hierarchy of “marking menus” [13]. After a learning phase, the text of the menus can be removed from the screen so that only an (abstract) mouse gesture remains. In another work about non-mnemonic interfaces, Appert and Zhai [2] compare stroke shortcuts to keyboard shortcuts for commands that bear no mnemonic link to the shortcut. They find that stroke shortcuts are far easier to learn.

On the opposite side, Lee and Lee [16] attempt an even more prominent use of intuition through simulated physical behavior. Measuring the performance of users in handling a contact list that possesses a varying degree of inertia when scrolling, they find that inertia is detrimental for the operation with the mouse but beneficial for the operation by touch.

3. APPROACH AND INITIAL SURVEY

To examine if basic commands of a DAW offer an “intuitive” [19] mapping to gestures, we picked a set of commands, see Table 1, and conducted two iterations of a web-based survey among prospective users. In each survey, the users had to assign one of 30 fixed gestures to each of the 22 commands. Both the gestures and the commands were presented in randomized order. Some of the commands such as “split selected region” change a specific item; others such as the zoom commands target the window as a whole, compare the notion of “binding” [24]. The list of commands contains a majority of one-shot commands but also contains two commands that may

Go one bar forward
Go one bar back
Jump to beginning
Jump to end
Toggle metronome
Select all
Cut selected region
Copy selected region
Paste copied or cut region
Duplicate selected region
Delete selected region
Split selected region
Glue selected regions
Undo last action
Redo action undone last
Increase value of a control
Decrease value of a control
Fit selected track to window
Fit selected region to window
Fit all tracks to window: vertical
Fit all tracks to window: horizontal
Recall zoom preset

Table 1: The 22 commands examined in this study

suggest continuous control: “increase value” and “decrease value”, compare the notion of “flow” [24].

The initial survey served as a pre-test to improve the selection of gestures of the final survey. This turned out to be necessary as the number of possibly interesting gestures is far too high to be presented in one single pass. Even with two iterations, we had to limit the survey to one- and two-point gestures, despite the availability of three- and four-finger gestures on mobile devices.

Representing the gestures in an appropriate graphical form for the survey turned out to be an unexpected problem. Specialized gesture clipart libraries such as [8, 15] offer professional-quality graphics, which, however, contain too much detail to display them in randomized order on a 6×5 grid. Hence, we opted for a highly reduced style in which one or two red dots indicate the starting points and dark lines show the motion, see Tables 2 and 3. A small number of gestures requires further explanations, which were also given in writing to the surveys’ participants: \bullet denotes a tap, \odot a double tap, \mathfrak{r} a tempo-

	14		12		24
	2		10		16
	17		9		9
	5		8		32
	7		5		15
	17		17		7
	6		5		21
	2		9		15
	11		9		7
	13		5		6

Table 2: Choice frequency of the gestures in the initial survey (total: 335)

rally extended tap. \circ^{\bullet} represents any number drawn with the mouse or the finger; Δ^Z represents any alphabetic character.

The surveys were implemented in both English and German and publicized using personal contacts as well as postings on Facebook and Linked-in to attract professional users. In addition, we circulated the surveys among students and recent graduates of study programs related to audio engineering and to media technology. We received 18 submissions for the initial survey and 59 for the final one. Adding all participants, 16% were female; the self-reported expertise was distributed as follows: 43% regard themselves as hobbyists, 25% are students in a program related to audio or media technology, 16% produce audio on a professional basis, 13% are software developers, researchers or educators. Most users reported that they own a touch-enabled device; only 6% neither use a touch-enabled device nor gesture-enabled desktop software. Table 2 shows the frequency with which participants have picked each of the 30 gestures in the initial survey.

4. FINAL SURVEY AND RESULTS

For the second, final survey—see Tables 3, 4, and 5—we stayed with the number of 30 gestures, but removed the less popular multi-touch rotate and tridirectional types to free up slots for other gestures. We added some the “tap” types that had been requested by participants of the initial survey. To prevent the letter-A Δ gesture of the first test to be mistaken

	Go one bar forward	Go one bar backward	Jump to beginning	Jump to end	Toggle metronome	Select all	Cut selected region	Copy selected region	Paste copied or cut region	Duplicate selected region	Delete selected region	Split selected region	Glue selected regions	Undo last action	Redo action undone last	Increase value of a control	Decrease value of a control	Fit selected track to window	Fit selected region to window	Fit all tracks to window: vertical	Fit all tracks to window: horiz.	Recall zoom preset	
Number of answers	54	54	54	54	53	54	53	52	54	49	54	54	54	53	54	54	54	53	53	51	54	49	
	4	4	4	6	6	4	2	6	6	4	4	6	2	6	6	2	2	6	4	4	4	8	
			2	2	9	22	6	13	15	18	4		2					9	8			10	
			2	4	4	11	4	12	13	2	2			6	2			4	9	4	4	12	
					<u>44</u>	6	9	6	4	4		2	6		2	2		6	2	2		2	
	<u>50</u>	2	2	13				2		4	2	2			4			2				2	
	7						2			2		2	2			4						2	
	6			13		4			11		2	2			2		4	2					
			5		6	9	2	2		4			4	2	9			4	8	2	13	4	
		<u>44</u>	12	4			2				15			15	4								
	2	4			2															2		2	
		4	7		2			2	4		4	2		4	4		4						
			2	4	2	2	6	2		2	4	4	4	8	2			4	2	2	15		
			7				2	2			2	2				<u>65</u>		4				2	
	13			17			2		4	4	2				4		2	4		4	2		
		11	25				2				6			6	6							2	2
	2			2			2	4	2			4				2		2	4	10	2	2	
		2		4	2		6	2	9		2	9	2				<u>67</u>	2		4	2		
		<u>25</u>	<u>20</u>	6				2			4			2	2								
	11			20			2		6	2	2	6			4							2	
					2	2	2	4	4		2	15	4				2	4	2	10	4	2	
	2	4	4		2	9		4		6		2		11	<u>33</u>	19	2		2			6	
	2	4		2		9	2	2					2	<u>32</u>	9	2	17		2				
	2	2		2	4	2				6							2		8	2		6	
AZ			2		21	7	8	12	6	18	2	7	4		4			2	6	4	4	16	
		2		2		9	<u>30</u>		2	6	<u>33</u>	4	4	2	2	2		4	9		4	2	
		2	2	2	4	4	8	15	7	10	9	9	6	8	4			4	2	2	2		
									4		2		<u>50</u>				2	11	4	4	17	6	
			2			4			2				2					2	8	22	6		
						2		4	2	4	2	22	2			2		6	<u>26</u>	8	15	8	
					2					2		2						13		20	2	6	

Table 3: Results of the second survey (values for gestures in percent, rounded, summing to 100 % per column; percentages above 25 % underlined)

•	50	◦	55	↺	53
◉	63	△	45	↻	48
→	45	↑	46	↻ ^{o.g.}	18
↵	11	↶	30	⌘	63
↶	24	↷	32	⊗	62
↔	39	↕	19	⊔	52
←	52	↓	60	↔	53
↶	7	↷	28	⋮	20
↷	19	↶	29	↔	54
↔	33	↕	30	⋮	28

Table 4: Choice frequency of the gestures in the second survey (total number: 1168)

for a triangle, we explicitly differentiated between a triangle Δ and an alphabetic character Δ^Z .

The strongest agreements between the participants can be found in a number of complementary gestures for complementary function, which supports a similar finding of [24]: \uparrow and \downarrow are clear candidates for increasing and decreasing a parameter; \leftrightarrow and \leftarrow are prevalent choices for jumping by one bar; \circlearrowleft and \circlearrowright are preferred for undo/redo. One would have to study, though, if users also prefer linear gestures for controls that are graphically represented as knobs. The connection between the rotate gesture and undo/redo may indicate that the choice is influenced by the standard icons used in desktop software.

The gesture \otimes is assigned to both Cut and Delete, again as noted by [24] in a general context. It may be useful to combine Cut and Delete to a single command. The Δ gesture was connected to the metronome function, obviously due to the similarity in shape. The multi-touch gestures \leftrightarrow \vdots \leftrightarrow \vdots have been assigned in about 20% of the answers to zoom functions. In contrast, the “horizontal pinch” multi-touch gesture \leftrightarrow is assigned to the glue command in 50% of the answers. Giving the preponderance of multi-touch zoom gestures in mobile applications, this seems surprisingly and may be due to learning that has to take place.

Some general insights from our survey are: Simple, linear or rotational gestures are favored. Tap gestures are among the standard choices. Alphabetic

Linear gestures starting rightward	119
Linear gestures starting leftward	111
Linear gestures starting upward	127
Linear gestures starting downward	147
Unidirectional gestures	203
90° gestures	180
Bidirectional gestures	121
Rotate gestures	101
Letter gestures	63
Digit gestures	18
Tap gestures	168
Sign gestures (+, x, triangle)	159
Multi-touch pinch gestures	73
Multi-touch spread gestures	82
Simple single-touch gestures	672
Complex single-touch gestures	341
Multi-touch gestures	155

Table 5: Average frequency of gesture groups in the final survey (total number: 1168). Note that these groups are not mutually exclusive

characters Δ^Z are picked three times more often than number $o.g.$. Angled gestures do not offer a clear connection to commands.

5. CONCLUSIONS AND OUTLOOK

This paper has presented first results on which mouse and touch gestures are of interest to control DAW software. The answers of the participants in our surveys confirm former, general studies, but also indicate specific choices for DAW software concerning for instance commands for glueing and the metronome.

Rather than presenting a fixed choice of gestures, future investigations could try to learn from users [1], because their statements may not conform to what they actually like or understand. This would require training the participants of the tests so that they can fully exploit the range of possible gestures. An approach that is even more user-centered would enable users to assign *custom* gestures. However, this does not seem popular with current touch-enabled applications; it is even rare to see a user creating his or her custom keyboard shortcuts.

Much research has explored gestures and their mappings to commands. There is, however, still much to invent and examine when it comes to preventing fatigue and to show cues for gestures on the display (Which parts can be operated? With which gestures?) [26]. Touch screens pose a usability conundrum. If mounted vertically, they are easy to read but strenuous to touch—and vice versa, if mounted horizontally. Gestures executed with the mouse, on notebook computer’s mousepad or on a touch surface on the mouse may be of advantage here.

The development may not stop here, as Apple GarageBand for iPad demonstrates with its simulation of velocity sensing through the built-in accelerometer [3]. But the very same software also shows how persistent the use of button widgets in DAW software has become: Even in GarageBand, gestures have not rendered the classic VCR-style buttons “Jump to the beginning”, “Play”, and “Record” obsolete. In addition, the lack of haptic feedback [4] largely remains unsolved, despite first attempts for solutions through vibration or electric induction.

Besides a rapidly increasing number of devices with (multi-)touch interfaces we see a variety of novel human-computer interface devices used in games. For instance, the Wii remote controller offers itself as both a pointing device and as gesture input device, which may be used to control DAW software [22]. Microsoft Kinect, a 3D camera intended to be used as a game controller, currently is the basis for a huge number of experiments on gestural control, conducted by both researchers and hobbyists. One use of this devices could be to “conduct” a DAW in a highly expressive fashion—similar to how one would conduct an orchestra.

6. ACKNOWLEDGEMENTS

We thank the participants of our surveys for their time and effort.

7. REFERENCES

- [1] D. L. Akers. Observation-based design methods for gestural user interfaces. In *Proceedings of CHI Extended Abstracts*, CHI EA '07, pages 1625–1628, New York, NY, USA, 2007. ACM.
- [2] C. Appert and S. Zhai. Using strokes as command shortcuts: cognitive benefits and toolkit support. In *Proceedings of CHI*, pages 2289–2298, New York, NY, USA, 2009. ACM.
- [3] Apple Inc. GarageBand for iPad. Last checked March 2011. Available from: <http://itunes.apple.com/us/app/garageband/id408709785>.
- [4] W. Buxton, R. Hill, and P. Rowley. Issues and techniques in touch-sensitive tablet input. *SIGGRAPH Computer Graphics*, 19:215–224, July 1985.
- [5] S. K. Card, A. Newell, and T. P. Moran. *The Psychology of Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1983.
- [6] L. Colson. StepPolyArp – MIDI step polyphonic arpeggiator for iPad. Last checked March 2011. Available from: <http://laurentcolson.com/steppolyarp.html>.
- [7] R. George and J. Blake. Objects, containers, gestures, and manipulations: Universal foundational metaphors of natural user interfaces. SIGCHI 2010 Workshop on Natural User Interfaces, 2010.
- [8] GestureWorks. GestureWorks – open source multitouch gesture library and illustrations. Last checked March 2011. Available from: <http://gestureworks.com/features/open-source-gestures/>.
- [9] R. Hare. Mouse gestures for Internet Explorer. Last checked March 2011. Available from: <http://www.ysgyfarnog.co.uk/utilities/mousegestures/>.
- [10] D. Kammer, G. Freitag, M. Keck, and M. Wacker. Taxonomy and overview of multitouch frameworks: Architecture, scope and features. In *Workshop on Engineering Patterns for Multitouch Interfaces*, Berlin, 2010.
- [11] D. Kammer, J. Wojdziak, M. Keck, R. Groh, and S. Taranko. Towards a formalization of multi-touch gestures. In *Proceedings of ACM ITS*, pages 49–58, New York, NY, USA, 2010. ACM.

- [12] M. Karam and M. C. Schraefel. A taxonomy of gestures in human computer interactions. Technical Report ECSTR-IAM05-009, Electronics and Computer Science, University of Southampton, 2005.
- [13] G. Kurtenbach and W. Buxton. The limits of expert performance using hierarchic marking menus. In *Proceedings of INTERCHI*, pages 482–487, Amsterdam, The Netherlands, 1993. IOS Press.
- [14] G. Kurtenbach and W. Buxton. User learning and performance with marking menus. In *Proceedings of CHI*, pages 258–264, New York, NY, USA, 1994. ACM.
- [15] R. Lee. Gesturecons – icons for multi-touch interfaces. Last checked March 2011. Available from: <http://www.gesturecons.com/>.
- [16] S.-S. Lee and W. Lee. Exploring effectiveness of physical metaphor in interaction design. In *Proceedings of CHI Extended Abstracts*, CHI EA '09, pages 4363–4368, New York, NY, USA, 2009. ACM.
- [17] Microsoft. Edit text using gestures, 2002. Last checked March 2011. Available from: <http://www.microsoft.com/windowsxp/using/tablet/learnmore/gestures.msp>.
- [18] Mozilla Corp. Add-on search results for gesture: Add-ons for firefox. Last checked March 2011. Available from: <https://addons.mozilla.org/en-US/firefox/search/?q=gesture>.
- [19] A. Naumann, J. Hurtienne, J. H. Israel, C. Mohs, M. C. Kindsmüller, H. A. Meyer, and S. Hußlein. Intuitive use of user interfaces: defining a vague concept. In *Proceedings of EPCE*, pages 128–136, Berlin, Heidelberg, 2007. Springer-Verlag.
- [20] D. A. Norman and J. Nielsen. Gestural interfaces: a step backward in usability. *Interactions*, 17:46–49, September 2010.
- [21] Opera Software. Opera gestures tutorial. Last checked March 2011. Available from: <http://www.opera.com/browser/tutorials/gestures/>.
- [22] R. Selfridge and J. Reiss. Interactive mixing using Wii controller. In *Proceedings of the 130th Convention of the AES*, 2010. In print.
- [23] D. Wigdor. Architecting next-generation user interfaces. In *Proceedings of AVI*, pages 16–22, New York, NY, USA, 2010. ACM.
- [24] J. O. Wobbrock, M. R. Morris, and A. D. Wilson. User-defined gestures for surface computing. In *Proceedings of CHI*, pages 1083–1092, New York, NY, USA, 2009. ACM.
- [25] M. Wu, C. Shen, K. Ryall, C. Forlines, and R. Balakrishnan. Gesture registration, relaxation, and reuse for multi-point direct-touch surfaces. In *Proceedings of IEEE TABLETOP*, pages 185–192, Washington, DC, USA, 2006. IEEE Computer Society.
- [26] W. Yee. Potential limitations of multi-touch gesture vocabulary: Differentiation, adoption, fatigue. In *Proceedings of HCI International. Part II: Novel Interaction Methods and Techniques*, pages 291–300, Berlin, Heidelberg, 2009. Springer-Verlag.