A Quantitative Review of Mappings in Musical iOS Applications

Thor Kell
IDMIL / CIRMMT, McGill University
thor.kell@mail.mcgill.ca

Marcelo M. Wanderley
IDMIL / CIRMMT, McGill University
marcelo.wanderley@mcgill.ca

ABSTRACT

We present a quantitative review of the mappings and metaphors used across the most popular musical iOS applications. We examined 337 applications in terms of both the metaphor they present to the user (piano, guitar, etc), and the exact nature of their mappings (pitch mapped horizontally, time mapped vertically, etc). A special focus is given to applications that do not present a well-known interaction metaphor to the user. Potential reasons for the popularity of certain metaphors are given. We further suggest that this data could be used to help explore the iOS design space, and offer some examples.

1. INTRODUCTION

iOS is the dominant platform for touch-based musical applications [1] [2], and more and more musicians are using iOS devices and applications to perform, produce, and practice their music. In any sort of instrument or interface for making music, mappings are important [3], as are the metaphors for those mappings.

On iOS devices, the hardware inputs being mapped are very limited: a capacitive multi-touch surface, an accelerometer, and a microphone. Yet the software to capture these inputs is limitless - does the application capture each touch, complex gestures, or somewhere in between?. The sonic output generated by the application software is also essentially limitless: anything from simple sample playback to complex synthesis techniques can be used to create sound.

Hunt et al. have written about the value of mappings in mediating between these two layers [4], and Jacob et al. have also written about the value of mapping parameters that are related in an integral way [5]. On iOS, the integrality of parameters largely rests on the metaphor presented by the application.

Fels et al. have written about the value of metaphor in human-machine interactions, and how it can improve a performer's understanding of the mapping and the instrument [6]. On iOS devices, as will be seen, the metaphors tend to be exceedingly obvious: pianos and guitars abound. Some applications, however, have non-obvious mappings (tone control based on where a piano key is touched, for example) that a metaphorical piano does not have. Furthermore, the wide range of abstract applications make the question of metaphor (or lack thereof!) a key one. Wessel and Wright have also written about more abstract control metaphors, in terms of the relationship of gesture and metaphor to the acoustic results [7]. Finally, McGlynn et al. have written about the expressive possibilities of interfaces that are not modelled on existing metaphors [8]. Their paper does not explicitly mention mapping, but mapping choices are inherent in each interface they discuss.

We hope to provide real-world insight into how metaphors and mappings are used for music making on iOS devices. We examined the most popular iPhone and iPad music applications (as of February 2013), categorized them in terms of the metaphor used by the application, e.g. a piano keyboard or synthesizer console, and reviewed the exact mappings used, e.g. pitch mapped horizontally via discrete buttons, with low pitches on the left. Based on this overview, we offer suggestions as to how to best use this data to create effective iOS music applications (apps), in terms of both standard and non-standard mappings.

2. METHOD

From the approximately 800,000 apps on the iOS app store [9], 1,200 music apps were chosen for review. These were selected by examining the 'Top Paid', 'Top Free', and 'Top Grossing' subsections of the iOS music app page, Each of those subsections lists 200 apps and differs across iPhone and iPad, giving 1,200 applications. Of these music apps, 337 deal with music creation in some way. These 337 apps were looked at in detail. "Music creation" is given a broad scope here: any application that allows creative interaction with music, in real time or not, is counted. This includes karaoke applications, but does not include radio applications, simple sound recorders, fingerprinting apps, or artist themed apps.

A cursory overview of the apps indicated that they could be organized into categories based on overarching metaphor - the most obvious being piano apps. Each app was assigned a metaphor, and then the total number of apps for each metaphor were added up. The goal of this classification was to delimit categories that would have broadly similar mappings. As the numbers for each app were added, it became clear that there were ten main categories, and then a large number of varied, heterogenous apps. Indeed, outside of the ten categories (all of which had at least thirteen apps), the largest metaphor was that of a violin, with two apps.

The final list of categories was as follows: Piano, DJ, Digital Audio Workstation (DAW), Music Production Controller (MPC), Guitar, Drum Kit, Synthesizer, Sequencer, Karaoke, Amp Sim, and Other. For each category, the metaphor and the general mappings for the metaphor were examined. A number of apps from each category were looked at in detail in order to discover novel or additional mappings. All apps in the Other category were looked at in detail. Regardless of category, each app was analyzed in terms of the direction and layout of its mappings, giving an overview of how musical parameters are mapped regardless of metaphor.

Note that only a subsection of the applications with standard metaphors were downloaded and tested; their mappings are assumed to be consistent across the category. A larger subset of these applications were examined via their websites. However, *every* app in the Other category was looked at in detail. When an application could not be downloaded and tested by hand (due to hardware or price restrictions), it was examined via screenshots and video. Specifically, those applications are: *Korg iKaosillator, Rockmate, Ocarina 2*, and *Live FX*.

3. METAPHORS

Table 1 contains an overview of the number of applications in each category. Note that we have split the Other category into apps that represent known acoustic instruments (a trumpet, for example), and apps that have no acoustic referent. It must also be noted that apps that appeared on both the iPhone and iPad are counted twice.

Table 1. Metaphors

Metaphor	iPhone	iPad	Total
Piano	25	43	68
DJ	17	15	32
DAW	14	16	30
MPC	14	14	28
Guitar	12	13	25
Drum Kit	7	14	21
Synthesizer	4	16	20
Sequencer	6	13	19
Karaoke	9	9	18
Amp Sim	5	8	13
Other	21	34	55
Other (Acoustic Instruments)	4	4	8
Total	138	199	337

As can be seen, piano apps are the standout category, followed somewhat surprisingly by DJ apps. The other two acoustic instruments, Guitar and Drum Kit, are below DAWs and MPC apps. This primacy of the electronic is perhaps not surprising given that iOS is an electronic platform, but it is belied by the massive popularity of piano applications. The piano may simply be such a well-known metaphor that it transcends the limitations of the iOS platform (lack of easy volume and timbre control, etc).

Continuing down the list, we find Synthesizers, Sequencers, Karaoke apps, and then Amp Sims - applications that mimic guitar amplifiers and effects pedals. In the Other category, a small subsection of apps mimics other acoustic instruments, against suggesting that non-acoustic metaphors are more dominant. The rest of the Other apps present no consistent metaphor.

The following sub-sections detail each category in terms of its metaphor and mappings, and discuss some of the variations within each category.

3.1 Piano

Piano apps display a traditional keyboard that plays discrete pitches. Pitches are mapped from left to right, low to high, in steps of one semitone. The vast majority of apps display a keyboard, though some simply display abstract circles (*Smule Magic Piano*) Playback of multiple pitches is possible. Volume control is generally not possible, nor is timbre control, though some apps offer a 'pedal' button, for sustained notes (*Piano Infinity*), or give control over the amount of reverberation added (*Piano Complete*). Some apps provide a toggle to switch between instruments - piano, grand piano, harpsichord, cat, dog, and so on (*Real Piano HD, Piano Infinity, Cat Piano Concerto*). Exact tuning control (A440 vs. A442, for example) is also sometimes available (*Real Piano HD*), and some apps give access to a synthesizer-esq pitch bend wheel and a mod wheel for real-time volume control (*Pianist Pro*). Solutions for volume control include a 'force based' volume control (*Real Piano HD*), and a volume control based on where the user strikes each key - higher up the key is softer, near the bottom of the key is louder (*Pianist Pro*). Some programs include teaching modes where notes fall from the top of the screen to the bottom, and must be played as they hit the bottom (*Smule Magic Piano, Piano Infinity*).

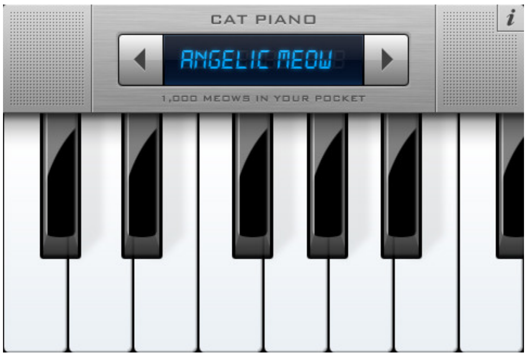

Figure 1. *Cat Piano Concerto*, a typical piano app

3.2 DJ

These apps provide two virtual turntables, with a virtual mixer. The volume of each turntable is controlled by a vertical fader, with louder being higher. The mix between turntables is controlled by a horizontal fader. Play, stop, and pause commands are controlled by buttons. The speed of each turntable is controlled by a pitch fader; faster is towards the user for some apps (*djay*), matching a traditional turntable, and away from the user for other apps (*DJ Rig Free*). This fader is generally in percent. 'Pitch bends', small corrections to the speed of each turntable, are controlled by buttons. The user can touch the 'turntable' to scratch or backspin, but not to change the speed (*DJ Rig Free*).

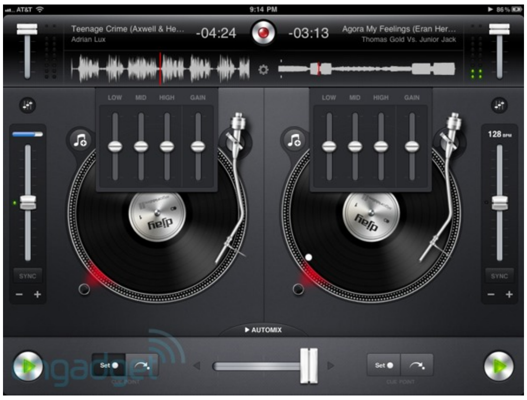
Teenage Crime (Axwell & He...
Adrian Lux
-04:24
-03:13
Agora My Feelings (Eran Her...
Thomas Gold Vs. Junior Jack
LOW MID HIGH GAIN
LOW MID HIGH GAIN
128 BPM
SYNC
SYNC
AUTOMIX
Set
Set

Figure 2. *djay*, a typical DJ app.

3.3 Digital Audio Workstation

DAW apps provide a complete solution for producing music and working with audio. They often include synthesizers, sequencers, and MPCs, as well as effect sections and mixers. Some go so far as to include auxiliary sends (*Auria*). The key distinction between a DAW app and a full-featured sequencer is that DAWs work with recorded audio: audio is recorded with a traditional red 'Record' button, and represented in clips wherein time moves from left to right, and amplitude is represented vertically (*FL Studio Mobile HD, Music Studio Lite*).

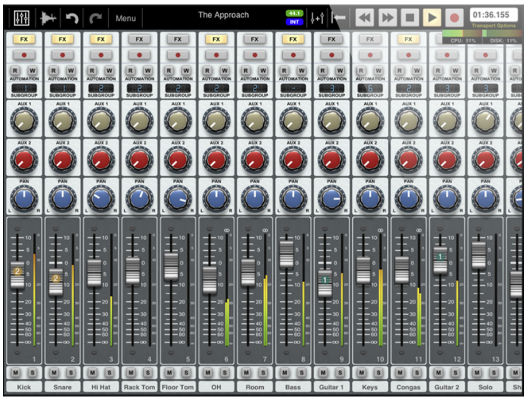

Figure 3. *Auria*, a typical DAW app.

3.4 MPC

These apps are based on the Akai **M**usic **P**roduction **C**enter line, a classic of hip-hop production. They have some number of trigger buttons in a grid - traditionally 16 buttons in a 4 x 4 grid. These buttons play a user-configurable sample when triggered. The user typically records one line, then loops it and records another line. Tempo can be tapped in (*iMPC*) or set with a slider (*BeatPad Lite*). The app may have a dedicated mixer (*iMPC*), or set volume via a slider on each pad (*Rhythm Pad*). There may be a separate FX section (*DJ Soundbox Pro*), or deep synthesis control of each drum sound (*Impaktor*). Finally, instead of the traditional 4x4 grid, some MPC apps have fewer buttons (*Rhythm Pad* has 8).

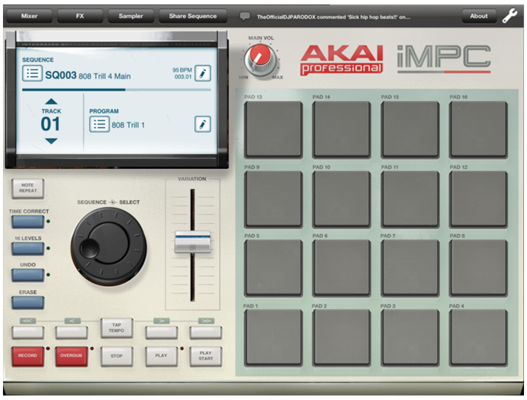

Figure 4. *iMPC*, a typical MPC app.

3.5 Guitar

A guitar, with 'strummable' strings and a fretboard. Frets are selected by holding down the appropriate area, and lower notes are placed to the left, as when holding a guitar. The lowest string is likewise placed closest to the user, and the strings are mapped vertically, again as when holding a guitar. Some apps provide direct access to complex chords via buttons (*Guitar!, Real Guitar Free*). Some apps provide vibrato by shaking the device (*Smule Magic Guitar*), and others allow effects via virtual pedals, with the timbre controlled by rotary knobs (*PocketGuitar*). Most apps do not provide timbral control or volume control.

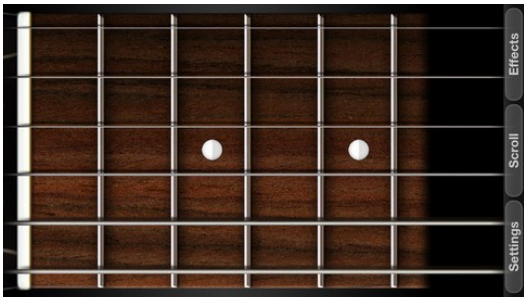

Figure 5. *Pocket Guitar*, a typical guitar app.

3.6 Drum Kit

A traditional drum kit, with some number of drums. Tapping each drum plays an appropriate sample, or one of a set of appropriate samples for that drum, and rolls can sometimes be performed by sliding a finger on a drum head; a faster slide leads to faster rolls (*Ratatap Drums Free*). As with the piano apps, volume and exact timbre control are generally not available. However, some applications provide force-based volume control (*Ratatap Drums Free*), and some play differing samples based on the exact location of the tap - playing the bell vs. the edge of a cymbal, for example (*Drums!*). Finally, the user can often switch between drum kits or drum kit layouts (*Drum Kit Pro, Drums!*)

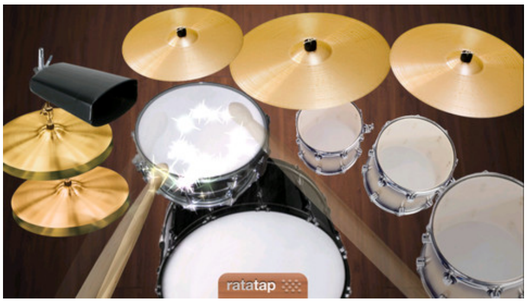

Figure 6. *Ratatap Drums*, a typical drum app.

3.7 Synthesizer

A synthesizer app exposes a selection of controls to a synthesis engine, and provides a piano-style keyboard for triggering the synthesized sounds. Control of the synthesis parameters is typically done with rotary knobs, but horizontal (*Alchemy*) or vertical (*Minisynth*) sliders, and XY pads (*Alchemy*) are also often used. Common parameters include:

- Wave type - sawtooth, sine, square, etc (*Magellan*)
- Filters - cutoff, type, resonance (*Alchemy*)
- Frequency modulation (*iMS20*)
- ADSR envelope control (*iMS20)*

In addition to triggering sounds with a piano keyboard, sequencers are included in some synthesizers (*Magellan, iMS20*), as are grids with volume mapped vertically (*Magellan*), and XY pads (*iMS20*). Indeed, some synthesizers can set the scale used by the keyboard or XY pad (*Animoog, iMS20*). In the case of the *Animoog*, this changes the layout of black and white keys! Finally, some synthesizers apps include extra effects, which are controlled with rotary knobs (*Magellan*) or with virtual patching environments (*iMS20, Audulus*).

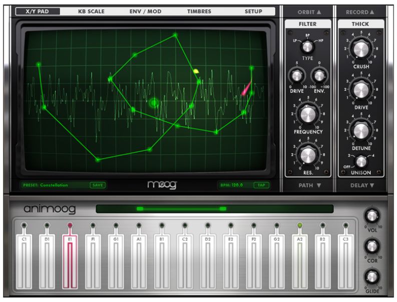

Figure 7. *Animoog*, a typical synthesizer app.

3.8 Sequencer

This category is inclusive of both drum machines and step sequencers. Time is divided into some number of discrete steps (16, 32, or 64), and time then moves step-by-step from left to right, according to a set tempo. One or more sounds or drum can be triggered on each step. Some sequencers model traditional drum machines (*Korg iElectribe*), and only allow access to a single track at a time, whereas others offer a grid with multiple tracks (*EasyBeats 2 Pro*). Some include DAW-style mixers with vertical sliders (KeyZ), some add effects sections with rotary control (*Molten Drum Machine*), and some have an MPC-style interface for adding events to the grid (*FunkBox Drum Machine*). The mapping of time also varies: some only display a single bar of time, whereas others allow a bar to be sequenced, and then allow the bar itself to be sequenced with other bars (*Genome MIDI Sequencer, DM1*). Zooming in time is occasionally provided by a rotary knob that controls the subdivision of a beat (*Molten Drum Machine*). Finally, volume per sound is sometimes controlled by the vertical position of the sound in the grid (*Looptastic Producer*).

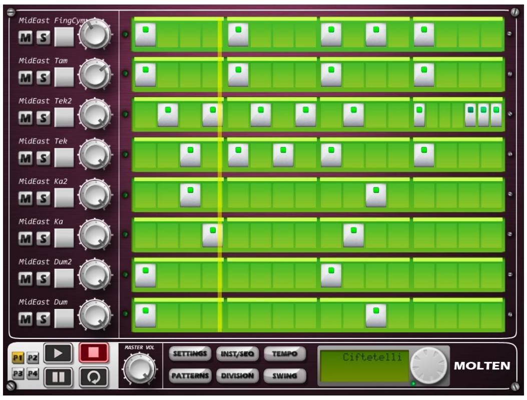

Figure 8. *Molten Drum Machine*, a typical sequencer app.

3.9 Karaoke

Karaoke apps allow the user to sing along to the instrumental track of a known song. At the very least, they present and somehow highlight the lyrics to be sung. Some provided visible pitch mapping, usually with pitch mapped vertically (higher notes are higher in pitch, lower notes are lower) and time moving from left to right (*StarMaker: Karaoke+*). Other options include additional reverb or echo (*Soulo Karaoke*), automating tuning effects that can be toggled on and off (*Sing! Karaoke, StarMaker: Karaoke+*), and toggles and level sliders for guide vocals (*StarMaker: Karaoke+*).

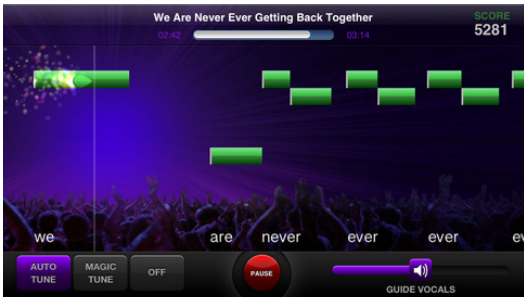

Figure 9. *StarMaker: Karaoke+*, a typical karakoae app.

3.10 Amp Sim

These apps provide some sort of model of a hardware FX box, usually a guitar pedal or guitar amplifier. Control of the effect is provided by rotary knobs (*AmpliTube*) horizontal faders (*AmpKit*), and on/off switches (*AmpliTube, AmpKit*). Some examples of the effects & parameters under control, from *AmpliTube*, are:

- Octave Pedal: direct level, octave level
- Delay: Delay time, feedback, delay level
- Phaser: speed

Some apps additionally allow the user to position a virtual microphone in front of the virtual amplifier, providing non-linear, two dimensional control of timbre (*Ultimate Guitar Amps and Effects*).

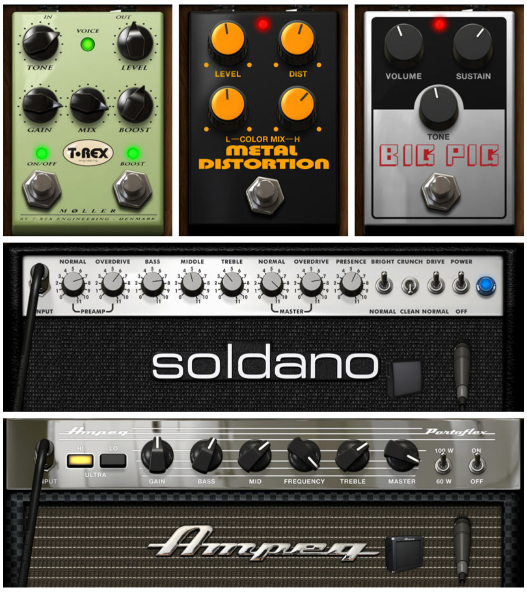

Figure 10. *AmpliTube*, a typical amp sim app.

3.11 Other

The Other category ranges from touch-based implementations of acoustic instruments to wildly abstracted music applications. Violin, harmonica, and trumpet applications were examined, along with gravity-based sequencers, isomorphic pitch-space controllers, and granular synthesizer experiments. In general, the most atypical mappings appeared in this category. For example, *Rework* maps pitch radially out from the centre, and *ThumbJam* allows the user to add vibrato and tremolo by shaking the device.

4. MAPPINGS

Beyond the metaphors listed above, we examined the raw mappings behind each app. For example, a standard piano application maps pitch horizontally from left to right (all directions given imply an increase), with discrete buttons. Likewise, a standard DAW application has a mixer that maps volume vertically, from bottom to top, continuously. Table 2 breaks down mappings in terms of pitch, trigger, time, volume, and timbre, across the ten metaphors listed above: Piano, DJ, DAW, MPC, Guitar, Drum Kit, Synthesizer, Sequencer, Karaoke, and Amp Sim.

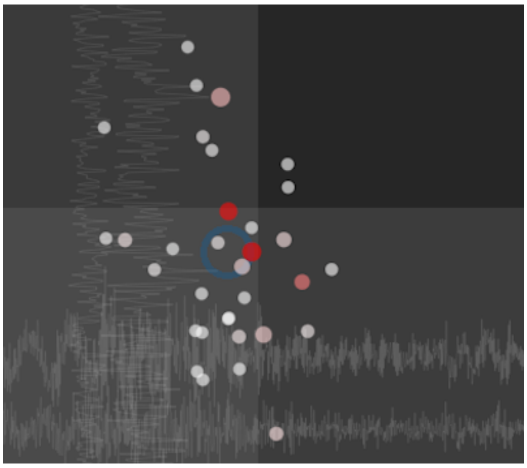

Figure 11. *Borderlands*, an app from the Other category.

It is important to note that some apps contain multiple mappings for a given parameter. Thus, the numbers in Table 2 will not add up to the total number of apps listed in Table 1. Secondly, despite the fact that many applications present rotary knobs or dials to control parameters (especially for timbral controls), these are not controlled in a rotary manner. They are in fact controlled as a vertical slider, and are notated here as such. Finally, some apps do not rotate when the device rotates. If the app presented a know metaphor (such as with guitar apps), the device was oriented to match the way the metaphorical instrument would be held. If the app presented no known metaphor, a best guess was taken, based on orientation of text, icons, and so on.

In Tables 2 and 3, each column refers to the parameter to be mapped. Pitch, Trigger, Volume, and Timbre should be self-explanatory. The Time column applies to applications like sequencers and DAWs that allow a user to queue or schedule events in time, and to tempo controls in DJ apps and sequencer apps. Each row refers to the mapping used. Most mappings should be self-explanatory. The Known Layout mapping is less clear: it refers to controlling a parameter through some visual layout that does not fit in a simple horizontal or vertical mapping, but is nevertheless clear to the user. For example, a drum kit app would control timbre via a known layout - that of a drum kit. Likewise, a trumpet app that mimics the valves of a trumpet would control pitch via a known layout.

4.1 Results

As can be seen from Table 2, the mappings for those standard categories do not cover a wide range of the possibilities. The runaway winner for pitch input, for example, is discrete pitches mapped left to right - almost certainly on a piano keyboard. It is important to note that mappings based on the keyboard are so common because users understand

Table 2. Mappings for standard categories

Mapping	Pitch	Trigger	Time	Volume	Timbre
Horizontal: Left-to-Right	143		67	32	
Horizontal: Right-to-Left				32	
Vertical: Top-to-Bottom	32				
Vertical: Bottom-to-Top	73			142	114
Continuous	50		48	174	114
Discrete	178		19	174	114
Known Layout					49
Toggle		45			50
Touch		243			
Gesture		43			
Microphone		18		18	

them instantly, without having to build up their own model for how an app maps pitch. Mapping pitch using a system of gestures would be interesting and novel, but would not be easy to use.

4.2 Other

In order to get a clearer view of potentially novel mappings, the raw mappings for each of the apps in the Other category (from Table 1) are listed in Table 3.

Most mappings listed in Table 3 should be self-explanatory. The Touch Area mapping refers to the width-times-height area touched, in terms of the size: a tap with a pinky finger covers a smaller area than a thumb, for example. The Physics mapping refers to some model of the physical world: virtual balls bouncing with pitch matched to their speed, for example. Finally, the Location mapping refers to placing a virtual object at a certain XY location in the app: Moving an virtual loudspeaker closer to a virtual microphone, for example.

4.3 Results

As can be seen from Table 3, these mappings are substantially more creative than the mappings for known metaphors. Indeed, many new mappings appear, and some of them are used for only single apps! Standard horizontal and vertical mappings remain very popular, but in general, these apps are more interesting - though they may also be correspondingly more difficult for an end user to grasp.

5. DISCUSSION

Our categorization of applications has shown that the majority of iOS music applications are based on known metaphors, and that piano applications are by far the most popular, followed by emulation of electronic music interfaces: DJ rigs, DAWs, and MPCs. Taken as a single class, the Other category would be the second most popular category, just behind piano apps. However, as these apps vary from simple percussion apps (*iMaracas*) to sophisticated isomorphic pitch controllers (*SoundPrism*), it would be disingenuous to group them together and point to their high number as evidence of the power of novel metaphors. Further investigation of this category would be needed in order to draw more accurate conclusions.

Table 3. Mappings for Other category

Mapping	Pitch	Trigger	Time	Volume	Timbre
Horizontal: Left-to-Right	22		15	4	11
Horizontal: Right-to-Left					1
Horizontal: Edge-to-Center				1	
Vertical: Top-to-Bottom	2		1		1
Vertical: Bottom-to-Top	16		6	12	16
Rotation: Clockwise	2		5		
Rotation: Counter-Clockwise	1				
Radial: Center-to-Edge	2		1	1	1
Radial: Edge-to-Center					
Diagonal: Bottom-Left-to-Top-Right	1				
Continuous	9		18	17	28
Discrete	40		9	2	2
Known Layout	3				4
Toggle	1	7			16
Touch		26			
Touch Area				1	
Gesture		1			
Microphone		9		3	
Shake	1	2		1	
Tilt	4	2		1	2
Physics	2	2			
Location		4	1		1
Colour	3				2

To the contrary, this research indicates that simple or known mappings and metaphors, such as the all-powerful piano keyboard, are the most popular. Even complex synthesis applications emulate physical synthesizers, with sundry dials and faders for timbral control. In the Other category, where apps lack a common metaphor, standard horizontal or vertical mappings still appear. However, numerous apps present novel mappings and novel inputs, indicating that there is more design space to be explored outside of keyboards and drum kits. Indeed, regardless of their lack of known metaphor, apps like *Figure, Borderlands* and *Samplr* show that successful applications can be made with novel mappings.

The importance of metaphor cannot be overstated. The massive popularity of piano apps, DJ apps, and so on, can be explained by Fels et al. [6] and their discussion of how a metaphor provides the user with a ”literature” of common knowledge about the interface. This leads to transparency between the mappings and the user, which makes the mappings more effective for beginners. Wessel and Wright [7] discuss the value of metaphors in terms of organizing musical material. They also discuss the value of more abstract and creative metaphors across parameters like pitch and timbre. As has been shown in the above tables, most iOS applications lack such a creative metaphor: only 55 out of 337, just less than a sixth of the examined apps, do not fit in to known categories. It may be possible to bring new categories to life, however. The lack of success of, say, iPhone violins could be because no app has made the correct set of mappings with which to emulate a violin.

In terms of mappings, Tables 2 and 3 could be used to aid the design of new iOS applications. While it seems premature to relate these mappings directly to profitability and financial success (especially as the App Store does not provide sales numbers for each app), the fact that the vast majority of applications map pitch from left to right indicates that an app aimed at widespread success should at least include such a mapping as an option. The same can be said for the mapping of volume vertically, and of time from left to right. Tables 2 and 3, however, could also be used to create spectacularly atypical iOS apps, simply by utilizing mappings that are under-represented. Such an app might map pitch from right to left, continuously, while controlling timbre via the microphone, and selecting rhythms via certain gestures. Or, the app might run time counter-clockwise, control pitch via the area of each touch, and map volume radially. These examples highlight the possibilities for deeply creative mapping solutions that exist on the iOS platform.

The most successful use of these tables, however, is probably in a combination of these two approaches. A scattergun, unfocused collection of novel mappings will probably result in a scattergun, unfocused app. However, an app with some traditional mappings and some novel mappings, especially in underutilized areas such shaking and tilting, or with underutilized parameters such as timbre, may be both more of a research success and more of a popular success

Finally, it is also important to note the limitations of the iPhone and iPad hardware, and how those limitations impact mappings. Though capable of exceptional capacitive multi-touch input, iOS devices lack the ability to easily tell how hard a user is tapping them, or any way of giving the user tactile feedback on their input. In some cases, this leads to creative mappings to work around these limitations. For instance, *Smule Magic Piano* maps the tone of each note vertically: touching higher up a key plays a darker sound. Likewise, *Ratatap Drums* uses data from the accelerometer to detect the force of a tap, and adjusts the volume accordingly.

6. CONCLUSION

We have summarized the most popular categories, mappings, and metaphors for musical iOS apps, as of February 2013. It must be noted that the iOS App Store is an ever-changing world: the top 200 apps of February 2013 are almost certainly not the top 200 apps of July 2013 - and are without question not the top 200 apps of 2015.

As of February 2013, however, we found a massive prevalence of piano apps, and of apps that show known metaphors to the user. We also found a subset of apps with no known metaphor, which were, as a rule, the applications with the most creative mappings. Across all apps, the vast majority used simple mappings: pitch from left to right, volume from top to bottom, and so on. Even within the Other subset of apps, these simple mappings were the most popular. However, this subset also included deeply creative mappings, making use of tilting, physics models, radial lines, and more. We then suggested that these lists of mappings could be used to explore underutilized designed spaces on iOS and similar platforms.

Touch applications for music, on iOS and on other platforms, will only become more popular as such technology becomes more and more available. It is hoped that this report has helped expose how mappings and metaphors are currently used on these devices, and helped shine a light on mappings that have not yet been investigated.

7. ACKNOWLEDGEMENTS

This work is partially funded by a Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant to the second author.

8. REFERENCES

[1] E. van Buskirk, "Developer Explains Why Android Sucks for Some Audio App," http://evolver.fm/2012/05/23/developer-explains-why-android-sucks-for-some-audio-apps/, 05 2012, accessed: 24/02/2013.

[2] M. R. Team, "The best Android music making apps in the world today," http://www.musicradar.com/news/tech/the-best-android-music-making-apps-in-the-world-today-276167, 02 2013, accessed: 24/02/2013.

[3] A. Hunt, M. M. Wanderley, and M. Paradis, "The importance of parameter mapping in electronic instrument design," *Journal of New Music Research*, vol. 32, no. 4, pp. 429–440, 2003.

[4] A. Hunt, M. Wanderley, and R. Kirk, "Towards a model for instrumental mapping in expert musical interaction," in *Proceedings of the 2000 International Computer Music Conference*, 2000, pp. 209–212.

[5] R. J. Jacob, L. E. Sibert, D. C. McFarlane, and M. P. Mullen Jr, "Integrality and separability of input devices," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 1, no. 1, pp. 3–26, 1994.

[6] S. Fels, A. Gadd, and A. Mulder, "Mapping transparency through metaphor: towards more expressive musical instruments," *Organised Sound*, vol. 7, no. 2, pp. 109–126, 2002.

[7] D. Wessel and M. Wright, "Problems and prospects for intimate musical control of computers," *Computer Music Journal*, vol. 26, no. 3, pp. 11–22, 2002.

[8] P. McGlynn, V. Lazzarini, G. Delap, and X. Chen, "Recontextualizing the multi-touch surface."

[9] Apple, "Apple Updates iOS to 6.1," http://www.apple.com/pr/library/2013/01/28Apple-Updates-iOS-to-6-1.html, 01 2013, accessed: 24/02/2013.