Decoding Performance with Data

Charles Martin

Abstract

While some improvisations such as jazz solos or keyboard performances can be transcribed for analysis, free improvisation on computer instruments can resist traditional transcription. For our touch-screen ensemble performances we have taken a data-driven approach to transcription and analysis that focuses on the interaction data created by the performers' touches rather than the resultant sounds.

In our system, the musician's sound-creating gestures are coded over the course of a performance automatically by a computer program using machine-learning algorithms. The resulting sequence of gestural states can be further analysed using transition matrices, familiar from Markov processes, and statistical methods. When transition matrices for all ensemble members are summed, they represent whole-ensemble behaviours. We have developed two matrix measures, flux and entropy, that help us to understand ensemble interactions.

This system allows gestural scores of performances to be automatically produced and the structure and ensemble interactions of performances to be compared even if different software instruments are used. Since the analysis is entirely performed by a computer program it can either run after performances or during them in real-time. In the latter case, information about ensemble interactions can be used to trigger changes in the software instruments.

In this paper, we describe the theoretical foundation for classifying and analysing touch-gestures in this system and some of the applications this process can have in analysis and performance.

Introduction

This paper is about analysing free improvisations on touch-screen devices, specifically on iPads. In the past, I've analysed free-improvisations by coding against a video performance[1]. In my 2012 Masters project, I started to develop computer music apps for mobile device, like iPhones and iPads, and gave them to my percussion group, Ensemble Evolution. We recorded a series of improvisations and I coded each video with how we were using the devices. This analysis yielded a long list of techniques and patterns that we used in these improvisations.

[1]Martin, C. (2012). Mobile Computer Music for Percussionists. Masters thesis, Department of Arts, Communication and Education, Luleå University of Technology. http://pure.ltu.se/portal/files/37021660/LTU-EX-2012-36941424.pdf

Figure 1: Improvised iPhone Performances with Ensemble Evolution

Although coding videos is productive, it is a very time consuming process; you have to watch the video over and over. However, with computer music instruments, such as the iPad apps I've developed during my current research, we can access very detailed information about how performers are using them. Every interaction with the touch-screen can be saved as a stream of data, so there's a potential to use this data to help decode these performances. There's a possibility of actually automating this process of coding a live improvisation, rather than having to do it by watching the video.

There are two advantages to automating this process of analysis. First of all, you can apply an automated process to a lot more improvisations than would be possible to code by hand. Rather than just a couple, or tens of improvisations; we could analyse hundreds - or more. Secondly, this analysis could be applied in real-time, that is, during the performance that is being analysed! The information from the analysis can actually be sent back to the iPad instruments, and they can be designed to respond in interesting ways. This is particularly useful in an ensemble context, because the instruments can be aware of how the whole group is playing, not just the individual performers.

So in this paper, I will introduce the software tools I've built to do automatic coding of ensemble touch-screen performances, and the process of creating them. These tools let us see the structure of gestural improvisations visually. They also help us to identify segments in improvised performances by marking moments where the performers change gesture most rapidly. Excitingly, we've been able to apply these systems to a large corpus of around 100 performances, and to real-time situations.

Gesture in free-improvisation.

So let's look at what these touch-screen free-improvisations actually look like. Here's a video of some of our recent iPad performances[2]. In this video the ensemble are all tapping, swiping, and swirling on their screens in a free-form way, and you can see the performers' touch gestures in the animation.

Figure 2: Four of the iPad apps developed for Ensemble Metatone

These apps have been created by a process of percussionist-led design that has emphasised this kind of free-form interaction with the whole screen. Rather than using a visual representation of piano keys, or a little picture of a drumset, performers can touch almost anywhere on the screen to create different kinds of sounds. The basic mapping is that a tap creates a short percussion sound, and a swipe creates a long sound with the volume connected to the speed of movement. This mapping emphasises direct manipulation of sounds, similar to an acoustic instrument. The performers tap to create notes, not to trigger sequences or arrange repeating phrases.

Steve Schick has discussed how percussion is defined more by the gesture of making sounds than by the instruments that percussionists play[3]. Percussionists

[2] Metatone Video: https://youtu.be/S-LFpBbjrxs

[3] Schick, S. (2006). The Percussionist's Art: Same Bed, Different Dreams. Rochester, NY, USA: University of Rochester Press.

are defined by what they do: "strike, scrape, brush, rub, whack, crash", all kinds of different objects. This is an appropriate analogy for computer music because the gestural input to a computer instrument can be mapped to any kind of electronic or sampled sound.

Towards a vocabulary for touch-screen gestures

To code these improvisations, we need a vocabulary of gestures to look for. There are actually lots of touch-screen gestures out there, but the existing characterisations are not very useful in a musical setting. Apple give developers a vocabulary of touch gestures[4], and there is lots of work in computer science (e.g. Wobbrock's unistroke gestures[5]), in the field of gesture recognition. These gestures are things like: double tap, long press, draw a star. They're intended to trigger some kind of command on the device, not to communicate a musical idea.

Our process for learning about improvisation was to improvise - so I set about defining a vocabularly by improvising with a group. I put together an ensemble of percussionists, including me, called Ensemble Metatone. We used a prototype app in a series of rehearsals and performances and (again) I coded what was going on in performances by hand and interviewed the performers to understand how they were using gestures. This piece of work is published elsewhere[6], but we found that the group was using a small vocabularly of simple continuous percussive gestures in creative ways to perform.

I settled on a list of 9 very simple gestures as a basis for our performance coding:

Gesture Vocabulary

1. None
2. Slow taps
3. Fast taps
4. Fast swipes
5. Accelerating swipes
6. Small Swirls
7. Big Swirls
8. Very Slow Swirls
9. Combinations

[4] Apple (2015). Event Handling Guide for iOS.

[5] Wobbrock, J. O., Wilson, A. D., & Li, Y. (2007). Gestures Without Libraries, Toolkits or Training: A $1 Recognizer for User Interface Prototypes. In Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology, New York, NY, USA, 2007 (pp. 159–168). ACM. 10.1145/1294211.1294238

[6] Martin, C., Gardner, H., & Swift, B. (2014). Exploring Percussive Gesture on iPads with Ensemble Metatone. In Proc. CHI '14, New York, NY, USA, 2014 (pp. 1025–1028). ACM. 10.1145/2556288.2557226

Something that all of these gestures have in common is that they never "finish", you could have "accelerating swipes" that last a few seconds, or several minutes. So these are continuous musical gestures, not command gestures that have a defined start and end.

Classifying Touch-Screen Data

Now we need to look at the touch-screen data to see how we can automatically identify these gestures. Each touch on the screen of your smartphone or tablet has a life-cycle that the operating system tracks and sends to the app you're using. There's three phases that are important to our analysis: a touch-start, a touch movement, and a touch ending.

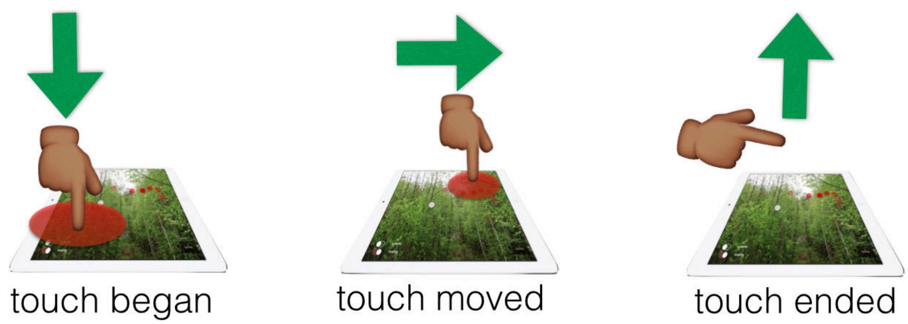

Figure 3: The touch lifecycle - touches begin, they might move, and they end.

My iPad apps use these messages to create sounds. For example, when they receive a "touchesBegan" message, they starts a sound. The apps also record these messages by sending the relevant data to a central server which gives each message a timestamp. Here's an example of some of this data that we collect:

```
2013-04-21T11:49:16.832236,christina,811.5,301.5,12
2013-04-21T11:49:16.835367,charles,136,43,0.5
2013-04-21T11:49:16.835645,yvonne,566.5,446.5,43.566040039062
2013-04-21T11:49:16.844141,christina,811.5,313,11.5
2013-04-21T11:49:16.854227,yvonne,525,449,41.575233459473
2013-04-21T11:49:16.863180,christina,808.5,323.5,10.920165061951
2013-04-21T11:49:16.865793,charles,136,44,1
2013-04-21T11:49:16.868442,yvonne,487,449,38
2013-04-21T11:49:16.880064,christina,803,334.5,12.298374176025
2013-04-21T11:49:16.882696,charles,136,44.5,0.5
2013-04-21T11:49:16.885074,yvonne,454,445.5,33.185085296631
2013-04-21T11:49:16.897053,christina,795.5,344.5,12.5
2013-04-21T11:49:16.902097,yvonne,426.5,435,29.436372756958
```

```
2013-04-21T11:49:16.904264,charles,136,45,0.5
2013-04-21T11:49:16.911449,christina,785.5,354.5,14.142135620117
2013-04-21T11:49:16.916264,yvonne,404.5,420,26.627054214478
2013-04-21T11:49:16.919006,charles,136,45.5,0.5
2013-04-21T11:49:16.928390,christina,774.5,362.5,13.601470947266
```

This is a bit dense, but you can see there's a date and time stamp at the start of each line. Next, we have the name of the performer that created that message. The three numbers at the end give us the X location of the touch, the Y location, and the current velocity. To simplify things, the format is the same for touchesBegan - the velocity is just zero. Even more simply, touchesEnded are just ignored. The timestamps tell us that this sample of data is less than 1/10th of a second so this data describes a microscopic amount of music.

Just one of these lines doesn't tell us much about what a performer is doing, we really need data over a couple of seconds to find out if a performer is tapping or swiping. So my analysis software takes 5-second windows of this data, divides up the contributions of each player in the group, and calculates some descriptive statistics about the data. These are things like:

- frequency of all touch messages
- frequency of touch began
- frequency of touch moved
- centroid in X
- centroid in Y
- standard deviation in X
- standard deviation in Y
- mean velocity

Now we need to automatically code these micro-gestures using the gestural vocabulary of macro-gestures. So to code this data, I used an approach from computer science called "Statistical Machine Learning", This technique has previously been used by Swift to identify which performer in a group is playing what instrument[7]. This is a group of algorithms that can be tuned to let you divide up data into multiple classes. To use these algorithms, you typically need to have a bunch of data with some known classifications, and you use this to "train" the parameters of the algorithm. Then you can send the algorithm new data and it will classify it for you automatically.

To collect the known data, my colleagues and I played each gesture into the app for a minute[8]. This data was used to train a well-known classifier algorithm

[7]Swift, B. (2012). The Design of a Smartphone-based Digital Musical Instrument for Jamming. Ph.D. dissertation, Research School of Computer Science, The Australian National University.

[8]We actually used a formal collection procedure where we played each gesture in a random order for a minute with 20-second breaks in between. The data at the edge of the gestures could be thrown out and we found that the accuracy of our classifier was comparable to other gesture recognition schemes (Martin 2015).

called Random Forests[9], which then uses the descriptive statistics to discriminate between the 9 touch gestures in our vocabulary.

Calculating Gesture Scores

So this trained classifier allows us to automatically code a log of the touch-data at regular intervals over a whole performance. The first interesting thing we can do with this data is to plot it - here's a plot of one of our coded performances.

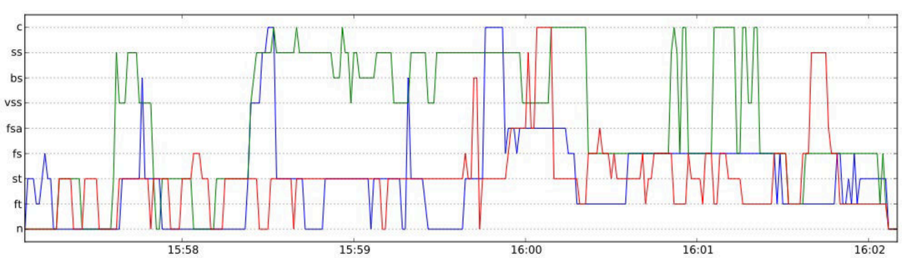

Figure 4: A gesture plot of an iPad performance. Each coloured line represents a different performer and each level on the y-axis is a different gesture.

Each level on the plot is a different gesture, and each coloured line is a different performer in the group. The codes are actually classified each second based on the previous five seconds of data in the performance. By presenting the data in this way, it is immediately apparent where the performers were staying together or apart, where one performer strikes out on their own, where someone lays out for a while, and where the whole the group suddenly seems to change gear and move into a different part of the gestural space. So just looking at this automatic coding can give us insights into the structure of the performance and the ensemble interactions going on.

Segmenting Performances

Authors like Jeff Pressing[10], Harald Stenström[11], and Tom Nunn[12], have described how free improvisation sometimes takes on a segmented form, where a group of musicians explore one particular sound world, and then spontaneously break off into another. Having acquired sequences of macro-gestures for each

[9] Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.

[10] Pressing, J. (1988) Improvisation: Methods and Models. In Generative Processes in Music. Oxford University Press. Oxford, UK.

[11] Stenström, H. (2009). Free Ensemble Improvisation. Gothenburg, Sweden: Konstnärliga fakultetskansliet, University of Gothenburg.

[12] Nunn, T. (1998). Wisdom of the Impulse: On the Nature of Musical Free Improvisation. self-published.

performer, we can start to search for moments in these performances that might correspond to the start of a new segment.

I've taken an approach of looking at the transitions between gestures, rather than the gestures themselves. Each pair of adjacent gestures defines a single transition. Over windows of 15 seconds in the performance, we can summarise all the transitions between the different gestures in a matrix. So in this matrix, the rows are the "from" gesture, and the columns are the "to" gesture.

	nothing	taps	swipes	swirls	combo
nothing				1	
taps					
swipes					
swirls					
combo					

Figure 5: A transition matrix for one transition. We reduce the nine-gesture classes to five simpler classes in transition calculations.

All the self-transitions, slow taps to slow taps, for example, are actually on this diagonal line. And the changing transitions are in the upper triangle and lower triangle. So we can see if the group is particularly static or a dynamic by looking at how much the data is centered on or off the diagonal.

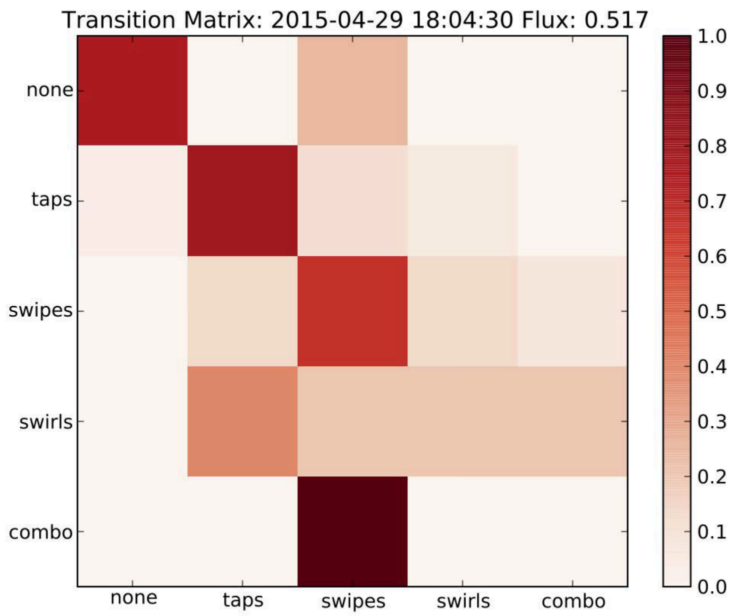

Figure 6: A 15-second transition matrix from a real performance, the darker red means a higher proportion of transitions

We've developed a simple measure on transition matrices called flux[13]. Flux is defined as the ratio of changed-transitions to self-transitions. It takes the value 0 if the group never changes gesture, and 1 if no two sequential gestures are the same. Another measure that we've been investigating is entropy[14], which is used in information theory. Entropy tells you how spread out data is on the matrix. So it is lower when the ensemble is using a few gestures, and higher when the group explores the whole space. We've noticed that flux, or gestural change, increases quickly at moments like these when the ensemble changes to a new segment of the improvisation. So with flux, we can actually annotate our gestural scores with moments where the group may have started a new musical idea.

Figure 7: A gesture plot of an iPad performance, annotated with flux-increase moments that segment the performance.

This figure shows a coded performance where the flux-increase moments have been annotated. You can see that at some points where the ensemble seems to change gear, the software has detected some moments, and since we have this moving window of data, it tends to detect them a few times in adjacent calculations.

[13]Martin, C., Gardner, H., & Swift, B. (2015). Tracking Ensemble Performance on Touch-Screens with Gesture Classification and Transition Matrices. In Proc. NIME '15, Baton Rouge, Louisiana, USA, 2015 (pp. 359–364). Louisiana State University.

[14]Shannon, C. E. (1948). A Mathematical Theory of Communcation. The Bells Systems Technical Journal, 27, 379–423.

Outcomes

This automatic process for coding performances allows us to interrogate a very large corpus of performances - I've recorded over 100. This is much larger than would be practical for me to code by hand by myself in a reasonable amount of time. Some questions that we're working on include whether these performances have particular kinds of structures in different situations, say between rehearsals and concerts.

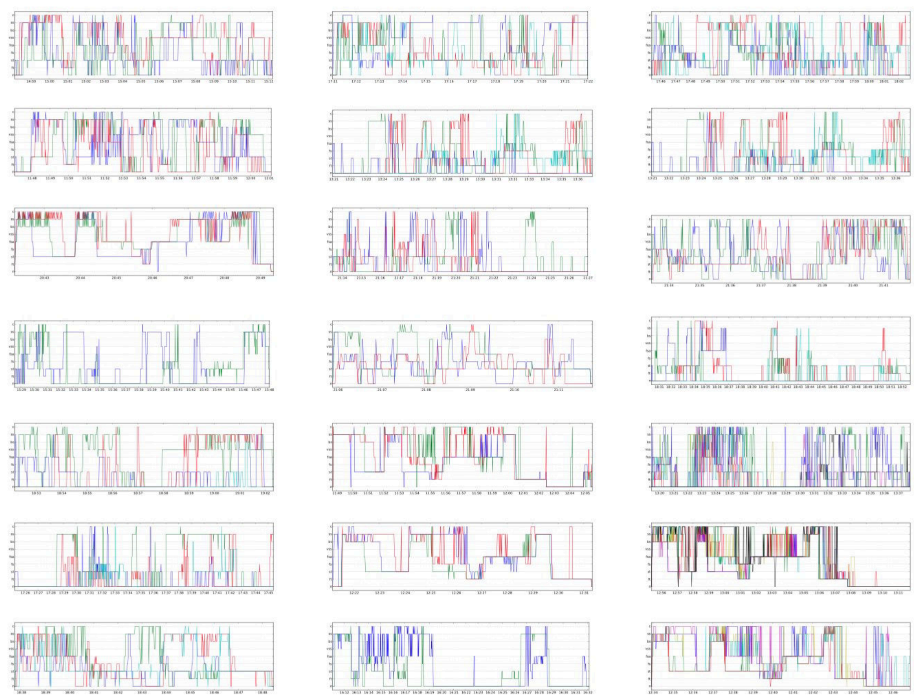

Figure 8: LOTs of coded performances.

We also use this process in real time during improvisations. The software, called Metatone Classifier[15], runs all the time on a server, and sends back gestures, flux, entropy and new idea messages to the iPads during a performance. We've designed a number of different apps that react to this information about the ensemble context[16], and present new notes and sounds to the performers, or encourage them to move onto a new idea.

So just to conclude, I'll say that free-improvisation can sometimes resist analysis,

[15] Metatone Classifier: Software for tracking touch-screen ensembles. Available on Github: http://github.com/cpmpercussion/MetatoneClassifier/

[16] Martin, C., Gardner, H., & Swift, B. (2015). Tracking Ensemble Performance on Touch-Screens with Gesture Classification and Transition Matrices. In Proc. NIME '15, Baton Rouge, Louisiana, USA, 2015 (pp. 359–364). Louisiana State University.

Figure 9: Metatone Classifier in real-time application during an iPad performance.

particularly because coding these performances is so time-consuming. This automatic process gives some immediate insight into these performances very quickly, and I think that this will let us continue to understand more and more about these highly interesting performances!