sampler/sampler - investigating Blackwork Embroidery through Live Coding performance

Sean Cotterill
Newcastle University
s.cotterill@newcastle.ac.uk

Toni Buckby
tonibuckby@tonibuckby.com

ABSTRACT

This paper discusses *sampler/sampler*, a networked performance environment investigating commonalities between sampling practices in Blackwork embroidery and electronic music using an embroidery emulator, live coding and live sonification. This paper details the influences and development of this work and both its place in and contribution to the canon of live coding practices, reflects on performances of the work, and suggests some potential further directions for this work.

1 INTRODUCTION

Originally developed for a residency for the AlgoMechanical festival in Sheffield, *sampler/sampler* was developed as a collaboration between the authors to use their practices to investigate commonalities between sampling practices in Blackwork embroidery and electronic music. The result of this collaboration was a networked performance environment which uses a digital Blackwork embroidery emulator developed by Buckby in Processing, coupled with a live coding engine for sampling and sonification developed in SuperCollider by Cotterill. The performance allowed the performativity of Blackwork embroidery to be explored digitally, as well as investigating the potential for Blackwork pattern data to be interrogated as a malleable sonic and visual material through live coding algorithms.

2 Background

2.1 Initial Ideas

The impetus for the development of this work was initially drawn from from Buckby's previous practice in Blackwork embroidery and digital media work (`http://tonibuckby.com`) and Cotterill's work in live coded electronic music performance and intermedia art (`http://seancotterill.xyz`). Development of the work was also reflective of a number of events and symposiums the authors attended in Sheffield focusing on the intersection of programming and pattern (notably Sonic Pattern and the Textility of Code (`http://sonicpattern.com/`)). Contemporary work by practitioners in the field surrounding programming and textiles also informed the development of the work, particularly the work within the 'Weaving Codes, Coding Weaves' project (Harlizius-Klück 2017) and the work of Rohrhuber, Griffiths and McLean (Rohrhuber and Griffiths 2017, Griffiths and McLean (2017)).

In addition to this, Sheffield, UK is home to a vibrant community of practitioners engaging with questions of pattern and code, and the direct impetus for the development of the work was provided by a short residency for the Festival of Algorithmic and Mechanical Movement, 2016 in Sheffield.

With *sampler/sampler* the intention was to produce an environment for live performance integrating physical blackwork embroidery practice and live coding, enmeshing the practice of the two artists across the digital and physical domains, however this plan had to be changed radically when the embroiderer suffered a broken wrist which made physical stitching unfeasible. The decision was made to translate their tacit knowledge (Polanyi 1966) into a digital emulator which would closely mimic the process of Blackwork embroidery stitching and could be interfaced with live coding - eschewing craft-based approaches in favour of digital ones.

2.2 Blackwork Embroidery

Traditionally defined by the application of black thread on a white background (although other colours can be used), Blackwork is a style of embroidery originating in North Africa during the 8th century and spreading via Spain to Western Europe, becoming popular in England during the 16th century (George Wingfield Digby 1963).

A counted thread technique usually worked in running stitches, Blackwork patterns are influenced by arabesque motifs and use simple straight stitches across the regular grid structure of woven fabric to create repeating geometric patterns. Basic Blackwork patterns (such as squares, crosses, chevrons and rhombi) are built from vertical, horizontal and diagonal lines worked on a woven grid. It is by repeating, combining and breaking apart the basic pattern shapes that the embroider can create an infinite palette of patterns of varying complexity.

There is a conceptual commonality between embroidery, computer code and sampling in electronic music - each is constructed from small repetitive components combined to create complex structures (Various 2010). The authors were interested in using Blackwork patterns to visually represent data and how traditional patterns as data can be interrogated through sonification and live coding.

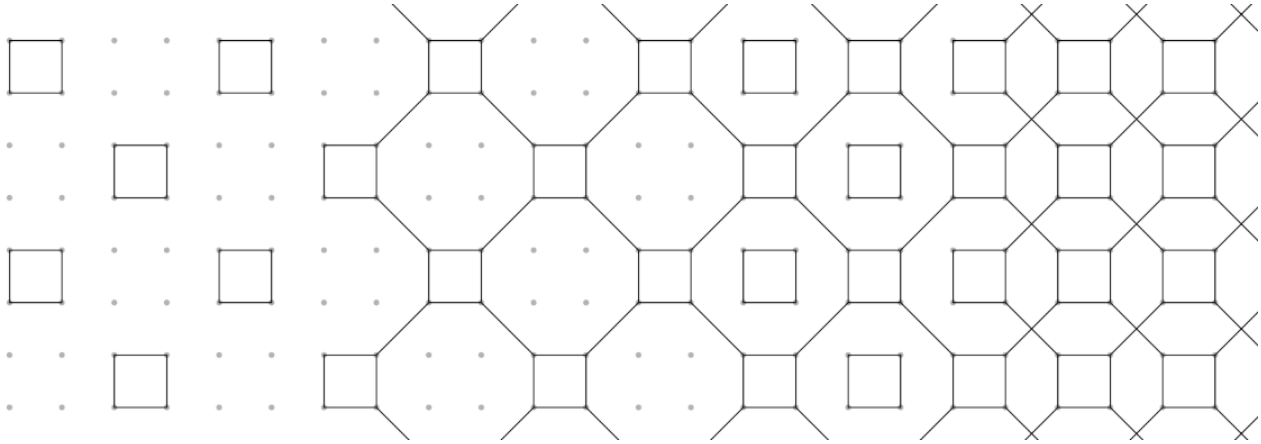

Figure 2: Blackwork patterns of varying complexity

3 DEVELOPMENT

3.1 Emulator

Built using Processing with the help of Alex McLean (`https://yaxu.org`), the emulator mimics the path a thread would take through a piece of even-weave fabric (Colton 1979). Beginning at the centre of a grid of evenly spaced points (the scale of which can be altered), the emulator draws a 'stitch' (line) when a keyboard key is pressed in one of 8 possible directions using the relative end position of the previous stitch as its start point. The stitches alternate between black and light grey to signify whether the thread is on the front or back of the 'fabric' (Processing canvas). There are functions to draw double length stitches and clear/reset the screen, as well as shortcuts for creating squares and crosses to speed up the creation of patterns. An embroiderer using the emulator to form traditional Blackwork embroidery patterns forms the basis of *sampler/sampler*.

3.2 Live Coding Engine

In order to facilitate a collaborative performance as well as draw on the second artist's live coding experience in SuperCollider, a communication protocol was set up between the emulator on the embroiderer's machine and a SuperCollider server on the live coder's machine. This allowed information about stitches entered using keystrokes to be received by SuperCollider, and in turn for information about sampled stitches to be sent from SuperCollider to the emulator. This communication was set up using SuperCollider's native `OSCdef` class and Andreas Schlegel's `oscP5` module for Processing.

This two-way communication between emulator and SuperCollider resulted in the decision to adopt a collaborative networked performance setup, with the two artists performing on separate screens using shared data in the form of traditional Blackwork patterns 'stitched' on the emulator live during the performance. The embroiderer would create traditional Blackwork patterns using the emulator, which would be sent to a dictionary in SuperCollider held by the live coder.

Once patterns are saved to SuperCollider's local dictionary, they are leveraged by a function which the live coder uses to play back patterns on their own machine and their own emulator. The function controls aspects such as the pattern played, the speed and delay of pattern repetition and additional stitches to be added at the end of a cycle. This function loops indefinitely, and can be supplemented with SuperCollider's native methods as well as being modified on the fly, allowing the algorithmic sequencing, transformation and repetition of patterns.

3.3 Sonification

Alongside live-coded pattern manipulation in SuperCollider, a simple sonification engine was also developed to expand the commonality between sampling in embroidery and electronic music. Informed by Theo Burt's 'Tiling Session' performance in Sheffield in 2015 (Burt 2013), the sonification of the pattern data is instant, directly reflecting the visual action of a performance. In order to differentiate the roles of the two performers, when stitches are places by the embroiderer a long-enveloped sine wave is triggered (with its frequency determined by direction), and when stitches are placed using the live coded sampling engine engine, a short-enveloped fixed frequency sound is triggered (with its wave type determined by direction).

In addition to these default settings, some parameters of this sonification can be specified in a function as part of the Task Reference Definition used to control sampled stitching, in particular the frequency of sounds, amplitude of sounds and the duration of envelopes.

The use of code in the performance was handled by the live coder using a transparent SuperCollider overlay on top of the emulator, which could be faded in over the developing patterns for the performer to make edits to the code. This approach was directly informed by TOPLAP's approach to live coding performance (TOPLAP 2010).

```
(
Tdef(\stitch,
    { loop ({
        ~loopPattern.(
            pattern:(0..5).choose,
            period:[0.1, 0.3, 0.5].choose,
            delay:1,
            dir1:(0..7).choose,
        );
        ~sampleResponder.(
            freq:[400,500,600,700].choose,
            amp:0.1,
            atk:0.001,
            rel:[0.1, 0.3].choose,
        );
    })}
)
)
Tdef(\stitch).play;
```

Figure 2: Example of a loop controlling patterns and sonification used in performance

3.4 Controller

To interface with the stitching emulator, a controller was designed for the host performer to create stitching patterns. The controller has a plate with an array of buttons for operating each stitch direction, as well as modifier and clear screen keys. The console uses an Arduino with Keyboard Library to trigger the relevant keystrokes to operate the Processing sketch when buttons are pressed.

4 Reflections on performance

Figure 3: Still from Performance at AlgoMechanical festival, showing Live Coder on left and Embroiderer on right

There have been two public performances of *sampler/sampler* to date, which followed the format outlined in this paper, and the documentation of the Algomechanical Festival (`https://www.youtube.com/watch?v=dY6oSwoRRho`) performance is broadly representative of our experience performing with *sampler/sampler* thus far.

Performances started from no existing pattern library, with the embroiderer building a library of increasingly complex patterns throughout the performance, giving the live coder a continually developing set of material to sample throughout. The live coder began with a very simple algorithm for repeating patterns which was improvised with, resulting in various degrees of sonic and visual complexity. The aspects for improvisation included iterative transformations, pattern looping, compound pattern sampling and live coding various aspects of pattern sonification. The sonic and visual potential within each pattern was easy to explore, and through live coding algorithms their character as sonic data as well as visual data could be illustrated and interrogated in many different ways.

The input of Buckby very much dictated the overall form of performances, starting with simple patterns resulting in repeated sonic and visual patterns, building up into a more complex set of actions by Cotterill as both the number and complexity of patterns available increased over the course of the performance - yeilding a multitude of both visual and sonic patterns.

Some interesting aspects of the system were discovered through (sometimes accidental) exploration of extreme values, and pushing the system to the point of error produced some interesting visual and sonic artifacts as a result of scheduling conflicts. These scheduling conflicts were a result of imperfect design, however they yielded unique characteristics that formed an intrinsic part of the performances.

The sonification engine revealed its simplicity rather quickly in performance, with the sine waves used to sonify the activity of the embroiderer quickly becoming tiring both to audiences and the performers, and while the sonification of sampled patterns can be modified through the performance, sonically it is very limited for use in performances beyond a relatively short duration. Despite this however, the intended goal of interrogating Blackwork Embroidery patterns through live coded performance was one the authors felt was achieved.

5 Further Development

Drawing on some of Jakob Nielsen's heuristics in evaluating the usability of the controller it must be noted that the action of pressing buttons bears no resemblance to the action of stitching or drawing patterns, and little relationship to the patterns being drawn on screen (Nielsen 1995). It would be desirable to develop a gestural interface to allow for more intuitive interactions. The sonification engine also needs significant development, as the sonic palette available is very limited.

Further development of the concept of the work can include re-integrating it with the craft processes behind Blackwork embroidery, and investigating how live coding and algorithmic pattern work can be re-integrated with the craft practices of a Blackwork embroiderer. This is an interesting area for further exploration given the instantaneous nature of the digital work and the time and effort needed to complete physical pieces of Blackwork stitching.

References

Burt, Theo. 2013. "Tiling Session." *Vimeo*. https://vimeo.com/114978130.

Colton, Virginia. 1979. *Complete Guide to Needlework*. Readers Digest Association.

George Wingfield Digby. 1963. *Elizabethan Embroidery*. London: Faber; Faber.

Griffiths, David, and Alex McLean. 2017. "Textility of Code: A Catalogue of Errors." *TEXTILE* 15 (2):198–214. https://doi.org/10.1080/14759756.2017.1298308.

Harlizius-Klück, Ellen. 2017. "Weaving as Binary Art and the Algebra of Patterns." *TEXTILE* 15 (2):176–97. https://doi.org/10.1080/14759756.2017.1298239.

Nielsen, Jakob. 1995. "10 Heuristics for User Interface Design: Article by Jakob Nielsen." https://www.nngroup.com/articles/ten-usability-heuristics/.

Polanyi, Michael. 1966. *The Tacit Dimension*. New York: Doubleday & Company Inc.

Rohrhuber, Julian, and David Griffiths. 2017. "Coding with Knots." *TEXTILE* 15 (2):142–57. https://doi.org/10.1080/14759756.2017.1298307.

TOPLAP. 2010. "ManifestoDraft - Toplap." https://toplap.org/wiki/ManifestoDraft.

Various. 2010. "Introduction to Open Source Embroidery." https://research.gold.ac.uk/6742/1/osembroidery.htm.