Manchester Metropolitan University

e-space
Manchester Metropolitan University's Research Repository

Please cite the Published Version

Russell, AP (2015) Repeatless: innovating print and pattern design with generative systems. In: Innovation: 8th MMU Postgraduate Research Conference, 05 November 2015 - 05 November 2015, Manchester Metropolitan University. (Unpublished)

Version: Presentation

Downloaded from: https://e-space.mmu.ac.uk/618039/

Usage rights: © In Copyright

Enquiries:
If you have questions about this document, contact openresearch@mmu.ac.uk. Please include the URL of the record in e-space. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from https://www.mmu.ac.uk/library/using-the-library/policies-and-guidelines)

Repeatless: innovating print and pattern design with generative systems

Alex Russell
MIRIAD/Manchester School of Art,
Manchester Metropolitan University
(U.K.)

a.russell@mmu.ac.uk

Cloth Of Gold (Russell)

Interdisciplinary, design practice-led PhD

Design, generative systems, computer programming and complexity

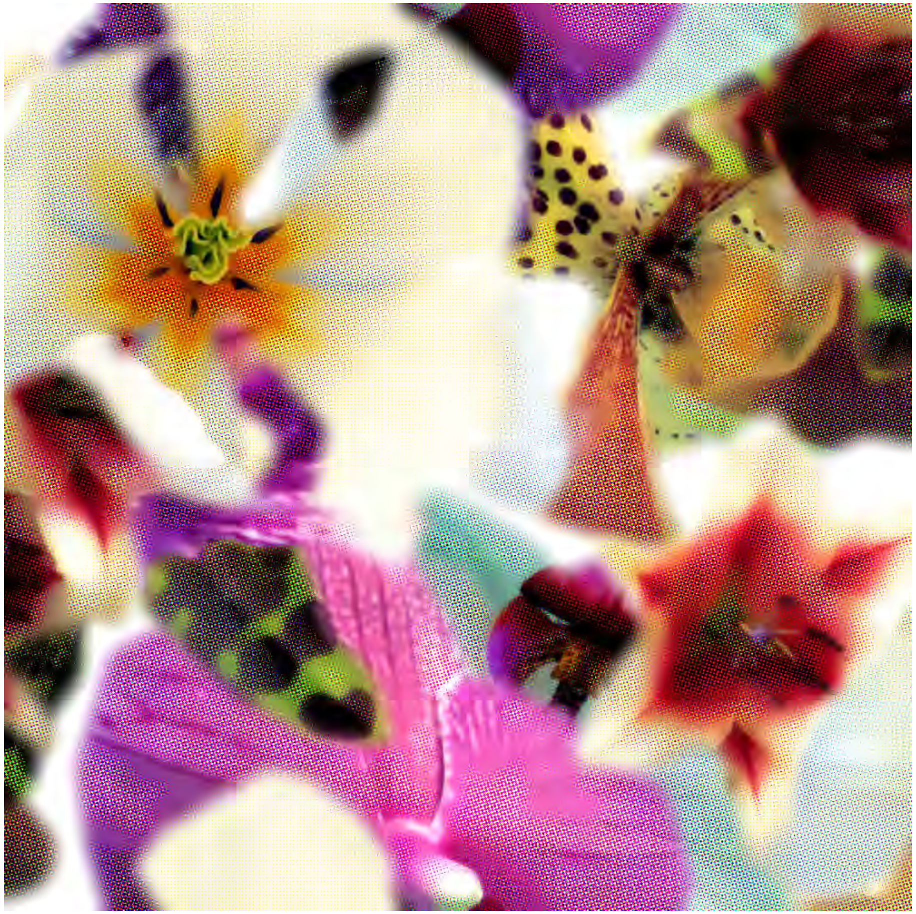
Patterandom Worn Stripe 002 (Russell)

Print and pattern design, printed textiles, surface pattern

Pattern that is printed onto fabric or any other substrate

Wallpaper/furnishings, clothes, wrapping paper...

...surrounded by it, but never heard of it



aram 1059 (Russell)

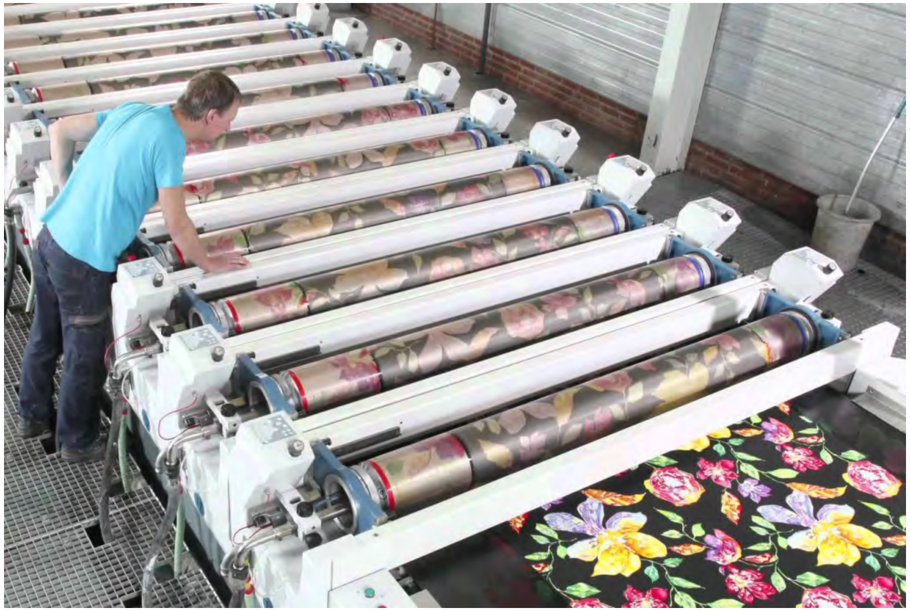
Pre-digital textile printing technology

Same design transferred repeatedly down fabric

Mechanised in industrial revolution

Copper roller, rotary screen

Design is physical part of printer mechanics



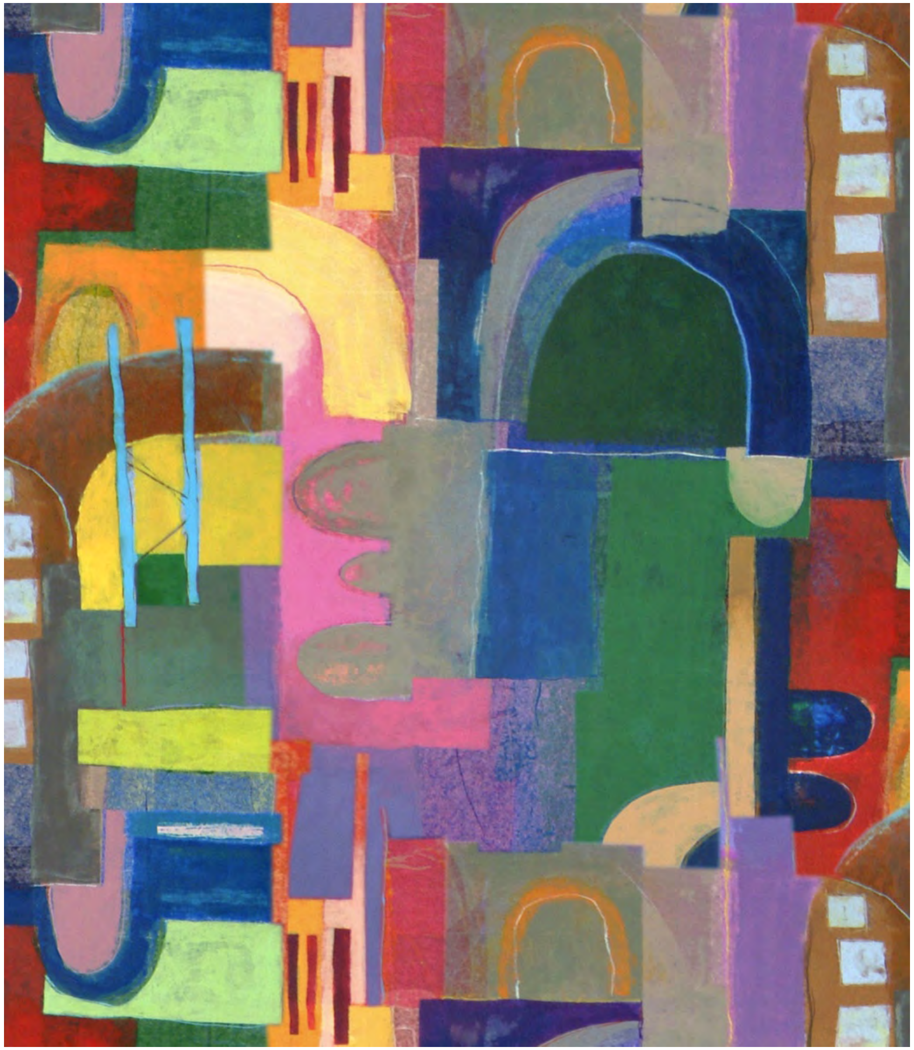
SPG Prints Rotary Screen Printer (SPG Prints)

Technology affects design

Designs have to create seamless pattern

Appears to flow down fabric

No element stands out



AVA Bridge (Russell)

Digital fabric printing also mechanical (jet of colour)

Design not physical part of printer

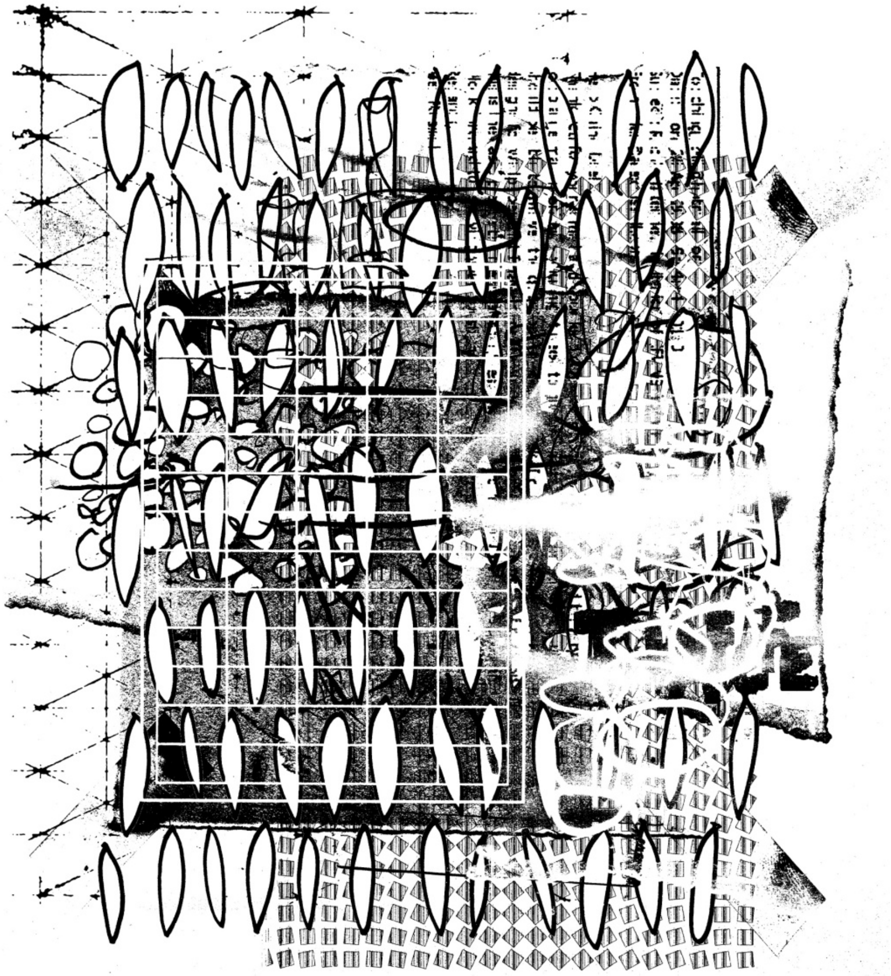
Print head only needs to know small part of a design at a time

Pattern could change as it's being printed

Existing research/practice* has identified potential for non-repeating pattern

Design completed, then sent to print

*Carlisle, 2002; Richardson, 2009; Häberle, 2011 and 2013; Schofield, 2012; McDonald, 2013; Paramanik, 2013 et al.



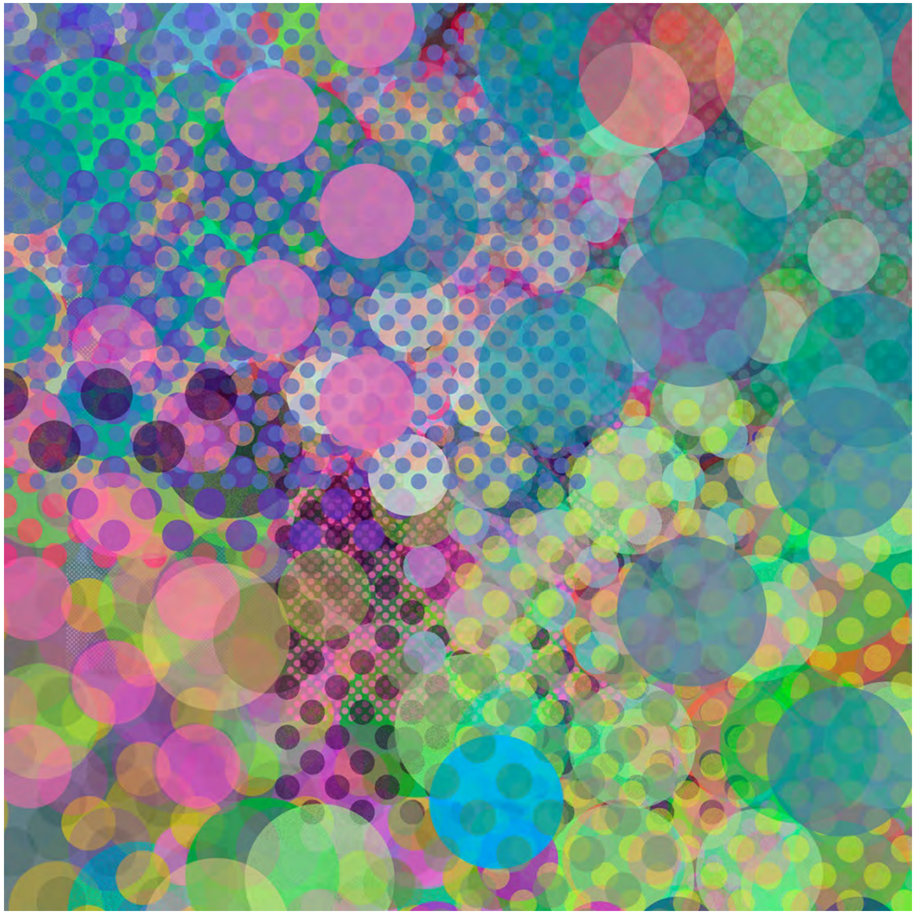
aram 1591 (Russell)

If data changed all the time, so would the pattern

If possible to create dynamically evolving design, could stream to digital printer

Rather than completed, fixed dimension design being sent to print, design that changed in real time and of any (potentially infinite) length could be digitally printed

Repeatless: dynamic, non-repeating pattern



Patterandom Worn Spot 002b (Russell)

Exploit this potential

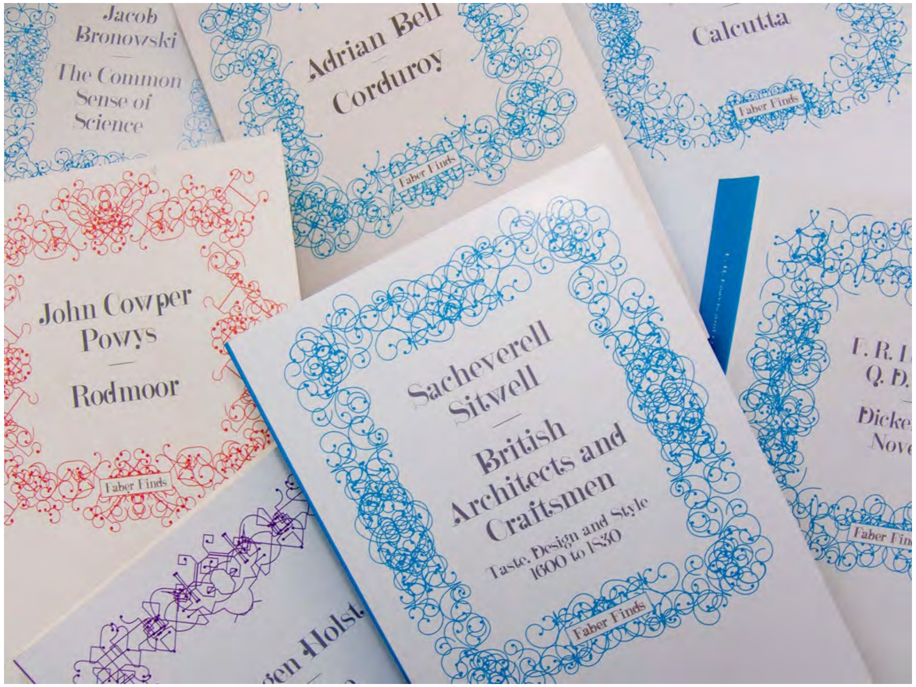
Generative design:

> the generation of designs by a set of rules or an algorithm, usually using computers*

Not as established in textiles as in other design fields (architecture, graphics)

Faber Finds**

Model for generative design needed



Faber Finds covers (Schmidt and Bantjes; Faber and Faber)

*Bruton and Radford, 2012: 166
**Schmidt and Bantjes, 2008

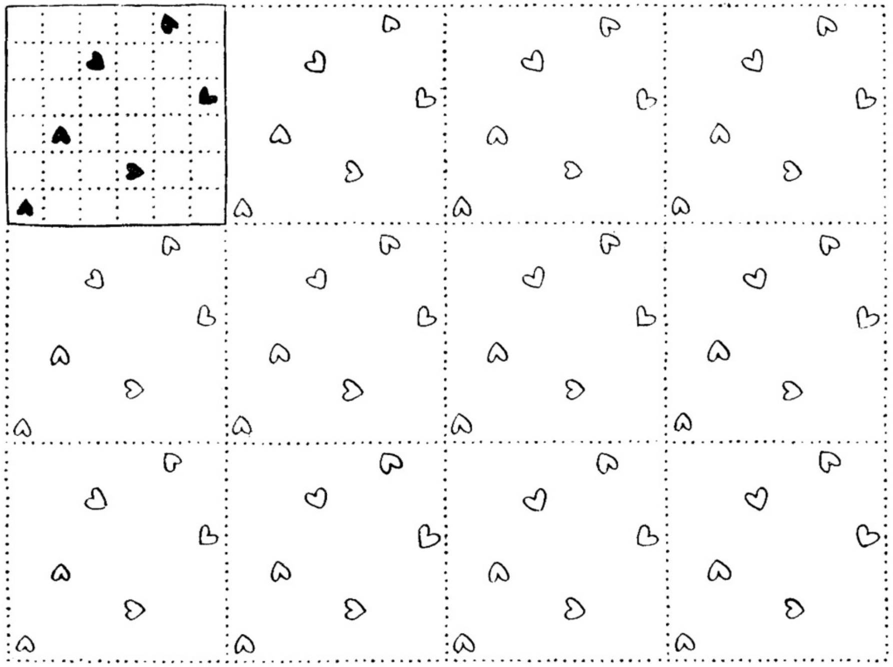
Idea that pattern could be assembled from a series of components

Day* outlines method of designing small-scale patterns

Method not dependant on imagery, but on structure by which it is arranged

Summarising de Quincy (1755-1849), McCullough** notes:

> [he] was one of the first to suggest a universal grammar in place of a lexicon or taxonomy of forms in architecture. ... artefacts are related by an underlying essence, even if their outward appearance is quite different.



Sateen repeat method (Day)

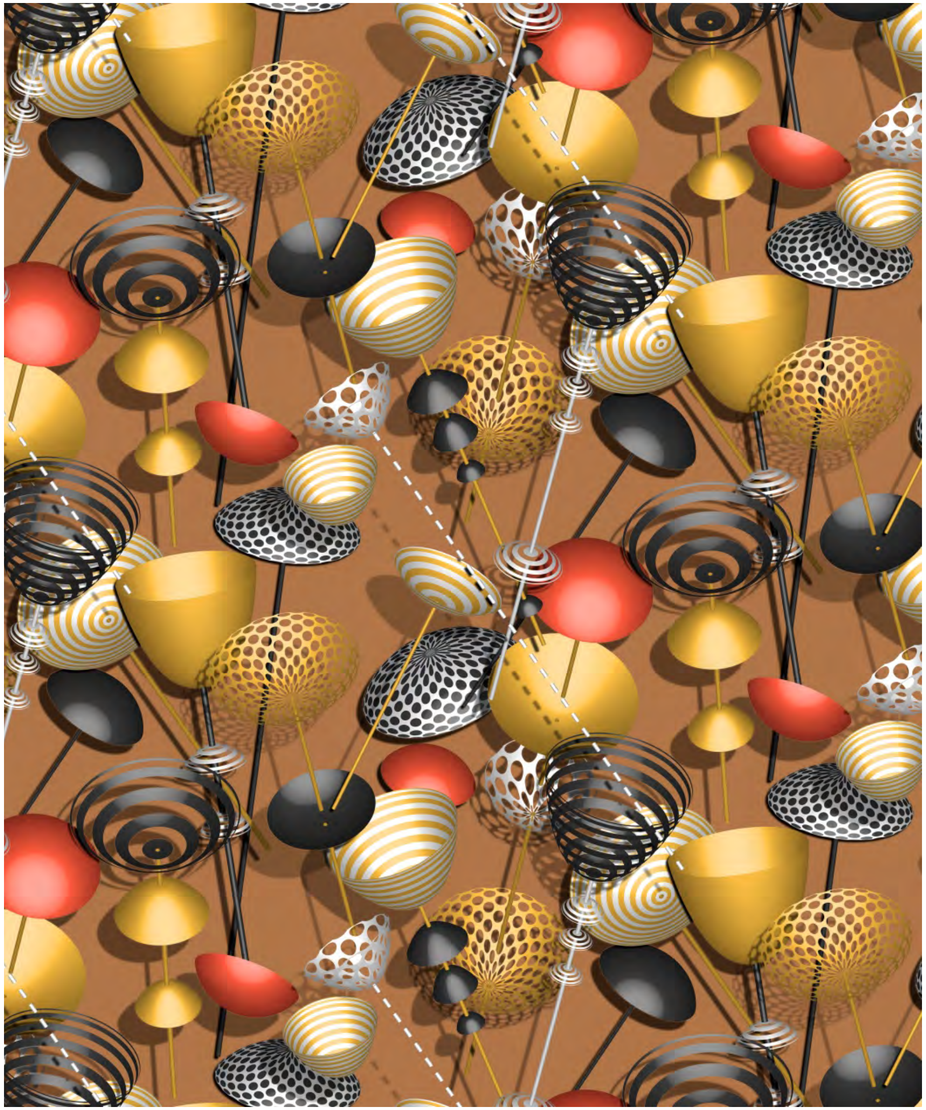
*Day,1903: 128-138
**McCullough, 1998: 181

How to develop a grammar for print/ pattern design

Traditional processes such as Day's sateen repeat could be abstracted

Paradox of creating non-repeating pattern from repeating grammar

Quality of traditional pattern design into repeatless pattern



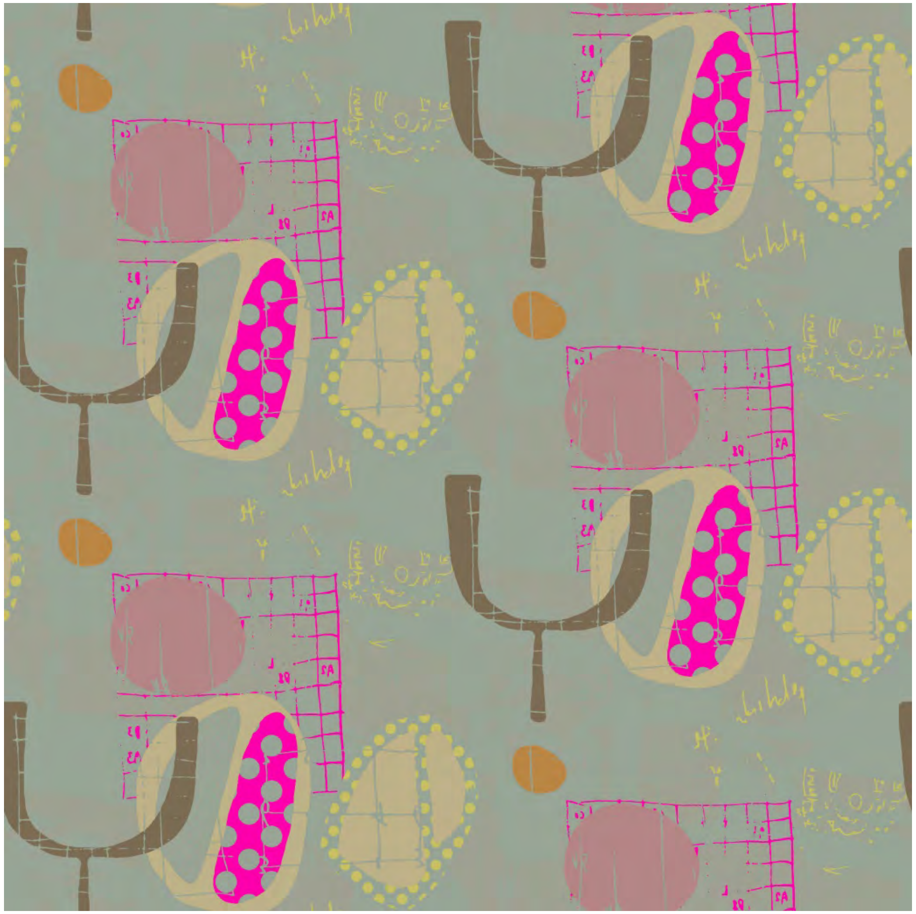
Rmx Caylx (Russell)

Balancing design

Avoiding tracking; no one element stands out

Design might appear random, but isn't

If actually random, elements will stand out



Tracking example (Russell)

System that allows elements within pattern to interact

Dynamic arrangement with underlying grammar

System could be a complex system:

> … research that seeks to explain how large numbers of relatively simple entities organize themselves … into a collective whole that creates patterns, uses information*

*Mitchell, 2009: 4

Noah Sateen (Russell)

Cellular automata

Grid where each cell's state evolves in time to those around it via set of rules

Example: Game of life*

Parallel's with Day's system

Previous (MA) study

*Conway, 1970

Game of life (Conway)

Rules of CA could be grammar of pattern design

Pattern a grid of motifs

Motifs = values of sites

Arrangment = deterministic rules

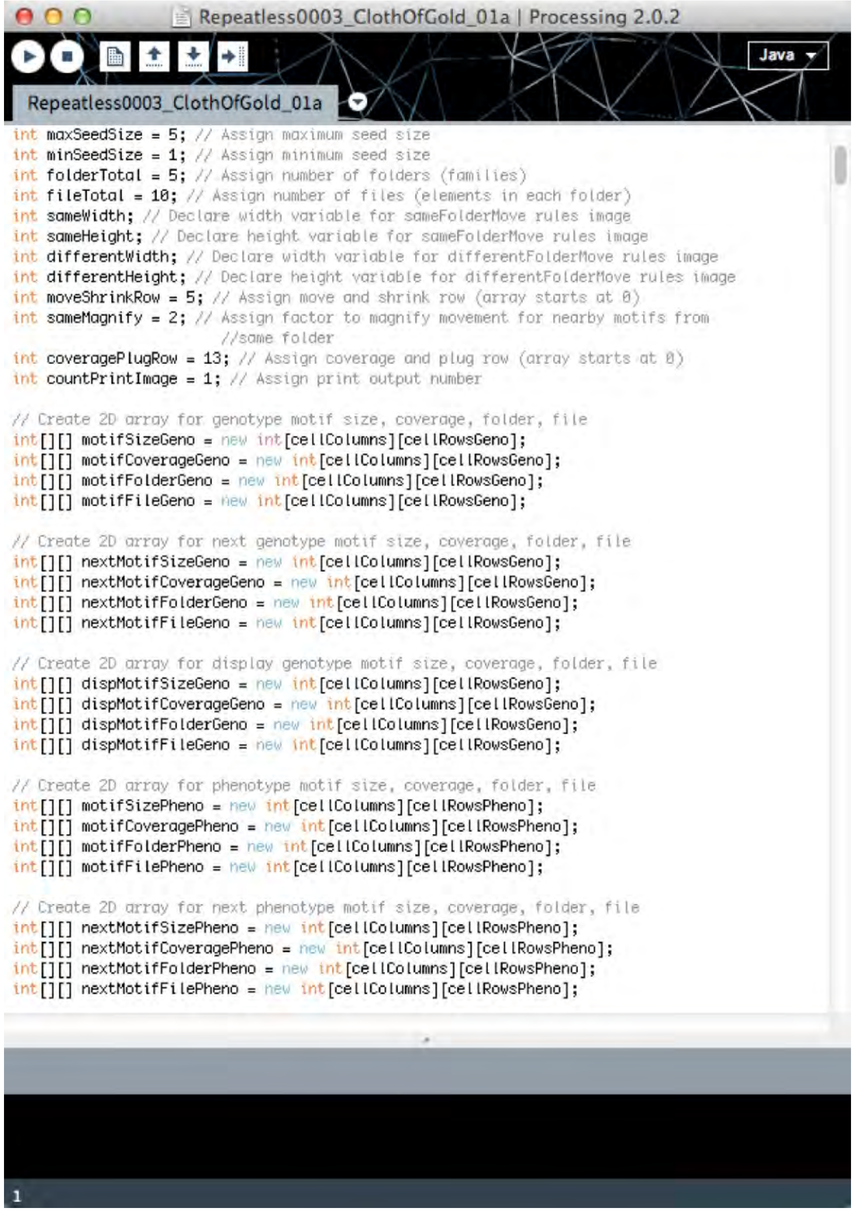
Motif arrangement (Russell)

Developed algorithm

Wrote generative software application in Processing*

Process in two biological taxonomy in some studies of complex systems**

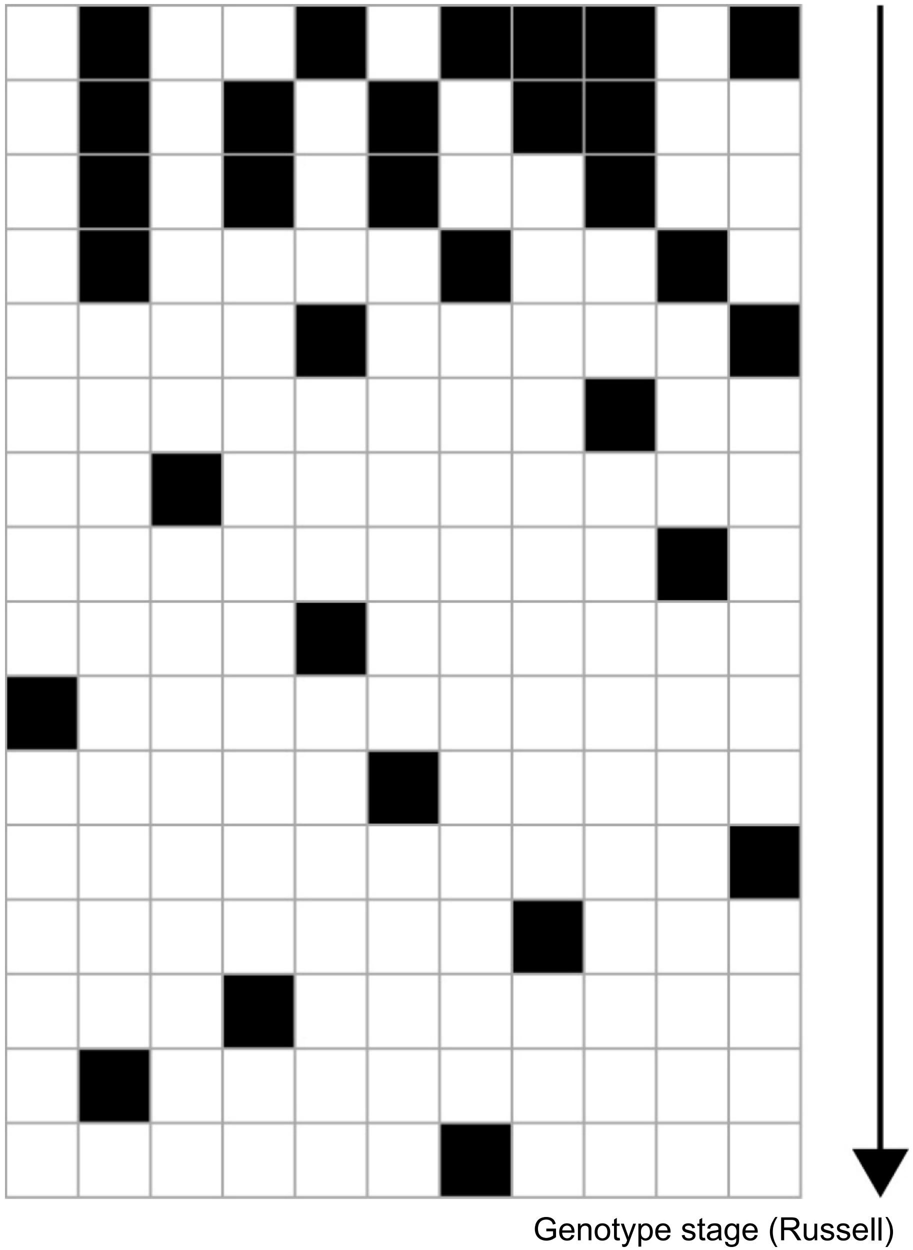
*Reas and Fry, 2007
**Bentley, 1999: 8; As and Schodek, 2008: 173


Repeatless0003_ClothOfGold_01a | Processing 2.0.2
Java
Repeatless0003_ClothOfGold_01a
1

```
int maxSeedSize = 5; // Assign maximum seed size
int minSeedSize = 1; // Assign minimum seed size
int folderTotal = 5; // Assign number of folders (families)
int fileTotal = 10; // Assign number of files (elements in each folder)
int sameWidth; // Declare width variable for sameFolderMove rules image
int sameHeight; // Declare height variable for sameFolderMove rules image
int differentWidth; // Declare width variable for differentFolderMove rules image
int differentHeight; // Declare height variable for differentFolderMove rules image
int moveShrinkRow = 5; // Assign move and shrink row (array starts at 0)
int sameMagnify = 2; // Assign factor to magnify movement for nearby motifs from
                        //same folder
int coveragePlugRow = 13; // Assign coverage and plug row (array starts at 0)
int countPrintImage = 1; // Assign print output number

// Create 2D array for genotype motif size, coverage, folder, file
int[][] motifSizeGeno = new int[cellColumns][cellRowsGeno];
int[][] motifCoverageGeno = new int[cellColumns][cellRowsGeno];
int[][] motifFolderGeno = new int[cellColumns][cellRowsGeno];
int[][] motifFileGeno = new int[cellColumns][cellRowsGeno];

// Create 2D array for next genotype motif size, coverage, folder, file
int[][] nextMotifSizeGeno = new int[cellColumns][cellRowsGeno];
int[][] nextMotifCoverageGeno = new int[cellColumns][cellRowsGeno];
int[][] nextMotifFolderGeno = new int[cellColumns][cellRowsGeno];
int[][] nextMotifFileGeno = new int[cellColumns][cellRowsGeno];

// Create 2D array for display genotype motif size, coverage, folder, file
int[][] dispMotifSizeGeno = new int[cellColumns][cellRowsGeno];
int[][] dispMotifCoverageGeno = new int[cellColumns][cellRowsGeno];
int[][] dispMotifFolderGeno = new int[cellColumns][cellRowsGeno];
int[][] dispMotifFileGeno = new int[cellColumns][cellRowsGeno];

// Create 2D array for phenotype motif size, coverage, folder, file
int[][] motifSizePheno = new int[cellColumns][cellRowsPheno];
int[][] motifCoveragePheno = new int[cellColumns][cellRowsPheno];
int[][] motifFolderPheno = new int[cellColumns][cellRowsPheno];
int[][] motifFilePheno = new int[cellColumns][cellRowsPheno];

// Create 2D array for next phenotype motif size, coverage, folder, file
int[][] nextMotifSizePheno = new int[cellColumns][cellRowsPheno];
int[][] nextMotifCoveragePheno = new int[cellColumns][cellRowsPheno];
int[][] nextMotifFolderPheno = new int[cellColumns][cellRowsPheno];
int[][] nextMotifFilePheno = new int[cellColumns][cellRowsPheno];
```

Cloth Of Gold code (Russell)

Part one: genotype, CA

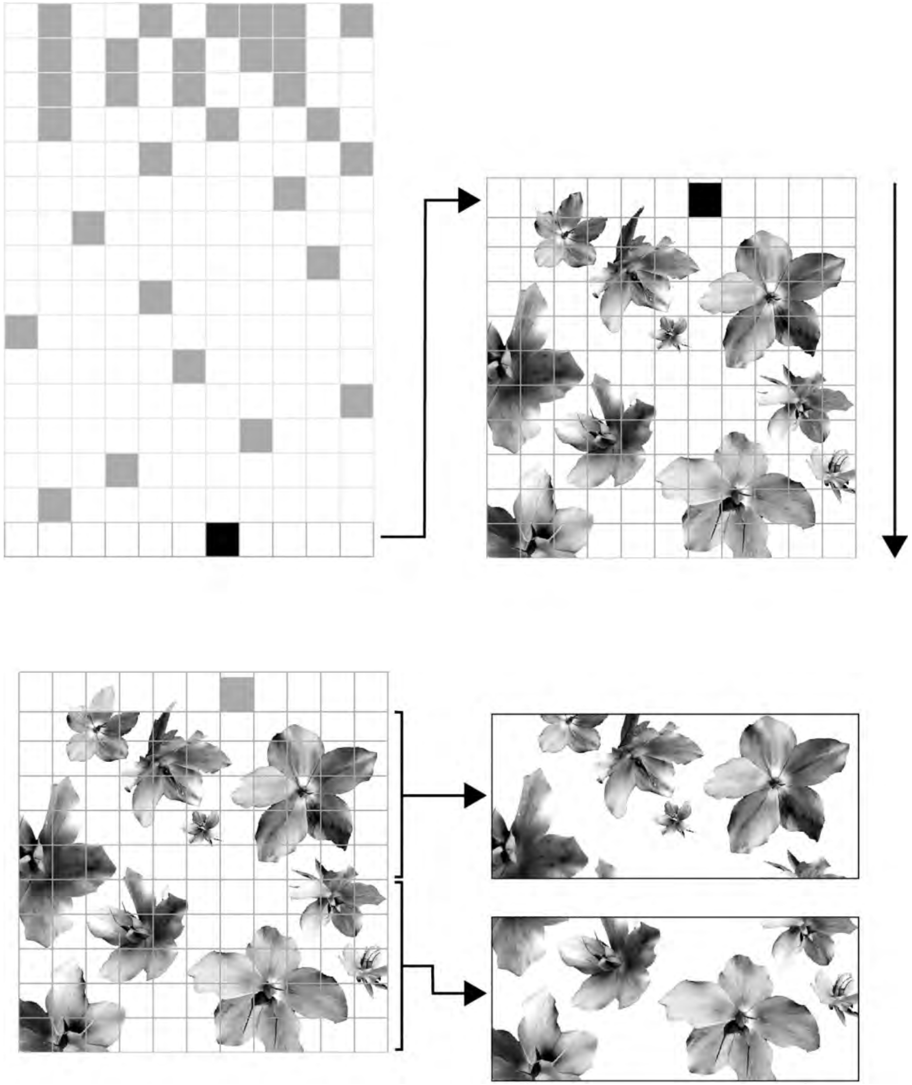


Part two: phenotype, library of motifs mapped onto grid

Saved section by section to be printed

Library can be any imagery

Important for commercial/industrial application



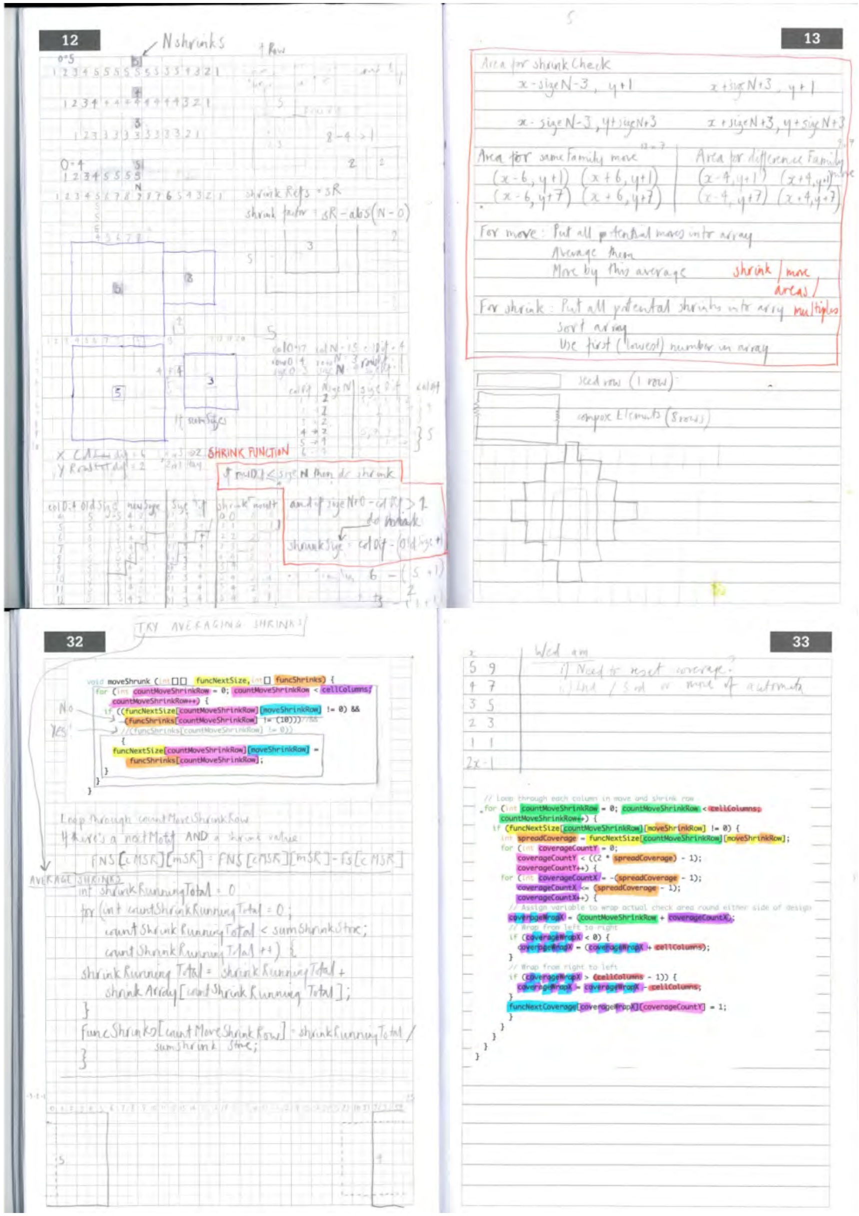
Phenotype stage (Russell)

To date:

Prototype developed

Proof of principle

Positioning papers to assert claim for originality



12

Nshrinks

1 Row

0=5

1 2 3 4 5 5 5 5 5 5 3 3 3 3 4 3 2 1

1 2 3 4 4 4 4 4 4 4 4 4 3 2 1

1 2 3 3 3 3 3 3 3 3 3 3 2 1

0=4

1 2 3 4 5 5 5 5

N

1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1

8 - 4 > 1

shrink Refs = sR

shrink factor = sR - abs(N - O)

col0 = 17 colN = 15 colDif = 4

row0 4 rowN 3

X ColDif = 4

Y RowDif = 2

SHRINK FUNCTION

if rowDif ≤ size N then do shrink

and if size NrO - col Dif > 1 do shrink

shrunkSize = colDif - (Old size +1)

6 - (5 + 1)

13

Area for shrink Check

x - sizeN-3 , y+1 x + sizeN+3 , y+1

x - sizeN-3 , y+sizeN+3 x + sizeN+3 , y+sizeN+3

Area for same Family move

(x-6, y+1) (x+6, y+1)

(x-6, y+7) (x+6, y+7)

Area for difference Family move

(x-4, y+1) (x+4, y+1)

(x-4, y+7) (x+4, y+7)

For move: Put all potential moves into array

Average them

Move by this average

shrink / move areas / multiples

For shrink: Put all potential shrinks into array

Sort array

Use first (lowest) number in array

seed row (1 row)

compose Elements (8 rows)

32

TRY AVERAGING SHRINKS

```
void moveShrunk (int[][] funcNextSize, int[] funcShrinks) {
  for (int countMoveShrinkRow = 0; countMoveShrinkRow < cellColumns;
    countMoveShrinkRow++) {
    if ((funcNextSize[countMoveShrinkRow][moveShrinkRow] != 0) &&
       (funcShrinks[countMoveShrinkRow] != (10)))//&&
       //(funcShrinks[countMoveShrinkRow] != 0))
       {
      funcNextSize[countMoveShrinkRow][moveShrinkRow] =
        funcShrinks[countMoveShrinkRow];
    }
  }
}
```

No

YES

Loop through countMoveShrinkRow

If there's a nextMotif AND a shrink value

FNS[cMSR][mSR] = FNS[cMSR][mSR] - FS[cMSR]

AVERAGE SHRINKS

```
int shrinkRunningTotal = 0
for (int countShrinkRunningTotal = 0;
    countShrinkRunningTotal < sumShrinkStore;
    countShrinkRunningTotal ++) {
  shrinkRunningTotal = shrinkRunningTotal +
    shrinkArray[countShrinkRunningTotal];
}
funcShrinks[countMoveShrinkRow] = shrinkRunningTotal /
              sumShrinkStore;
}
```

33

x	Wed am
5 9	i) Need to reset coverage?
4 7	ii) 2nd / 3rd or more of automata
3 5	
2 3	
1 1	
2x-1	

```
// Loop through each column in move and shrink row
for (int countMoveShrinkRow = 0; countMoveShrinkRow < cellColumns;
    countMoveShrinkRow++) {
  if (funcNextSize[countMoveShrinkRow][moveShrinkRow] != 0) {
    int spreadCoverage = funcNextSize[countMoveShrinkRow][moveShrinkRow];
    for (int coverageCountY = 0;
        coverageCountY < ((2 * spreadCoverage) - 1);
        coverageCountY++) {
    for (int coverageCountX = -(spreadCoverage - 1);
        coverageCountX <= (spreadCoverage - 1);
        coverageCountX++) {
      // Assign variable to wrap actual check area round either side of design
      coverageWrapX = (countMoveShrinkRow + coverageCountX);
      // Wrap from left to right
      if (coverageWrapX < 0) {
        coverageWrapX = (coverageWrapX + cellColumns);
      }
      // Wrap from right to left
      if (coverageWrapX > (cellColumns - 1)) {
        coverageWrapX = coverageWrapX - cellColumns;
      }
      funcNextCoverage[coverageWrapX][coverageCountY] = 1;
      }
     }
    }
  }
}
```

Cloth Of Gold sketchbook (Russell)

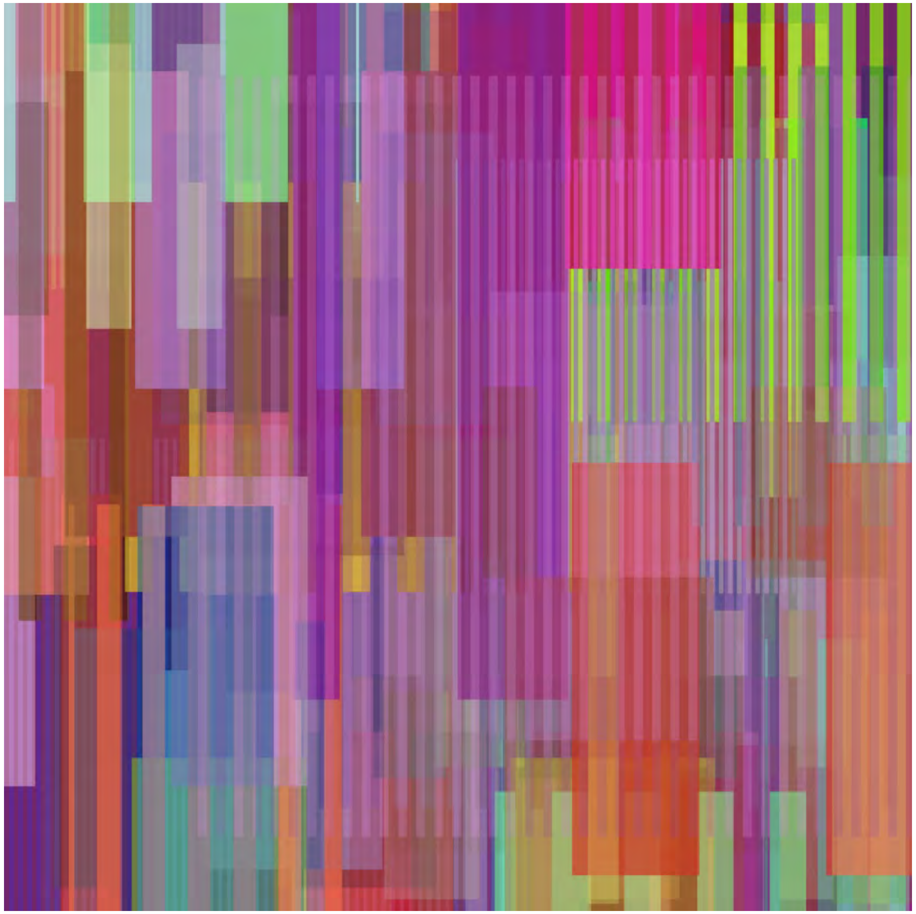
Next:

Develop new software application

Test design outputs via peer review

Assess quality of designs

Iterative approach, reflect on results, feedback into system



Patterandom Clean Stripe A 003 (Russell)

Innovation 01:

Pre-digital printing on industrial scale means large areas of fabric quickly covered with pattern

Trade-off that design does same thing over and over again

Digital hardware allows printing to be re-imagined; non-repeating designs printed

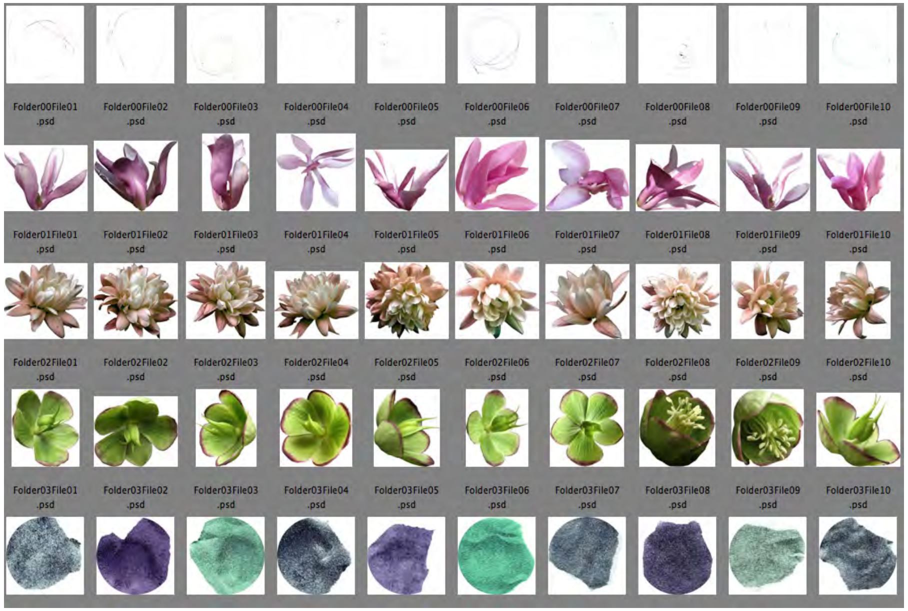
Cloth Of Gold Folder02 File02 (Russell)

Innovation 02:

Generative design produces ever-changing pattern in real time, streamed to a digital printer

CA, complex system governed by simple rules, ever-changing behaviour

Re-think pattern, any combination of predetermined motifs dynamically arranged design



Cloth Of Gold Library (Russell)

Innovation 03:

Existing methods of repeat design quantified for CA rules

Processes used by practitioners to conceal repeat used to develop algorithms

Allow the quality of repeatless design to be maintained at a consistently high level

CA outputs complex motif arrangements; obey guidelines yet never do same thing twice

Cloth Of Gold (Russell)

Interdisciplinary research across science, technology and design

Method of re-thinking pattern

Need for repeat is eradicated without losing any of rich legacy of print/pattern

Combination of complexity, digital systems, traditional design techniques

Particular synthesis of elements from the different areas that make this work innovative, shifting the paradigms of what pattern is and the way it can be reproduced

Cloth Of Gold (Russell)