Pied de Pulse: Packing Embroidered Circles and Coil Actuators in Pied de Poule (Houndstooth)

Loe Feijs[1] and Marina Toeters[2]
Department of Industrial Design
[1]Eindhoven University of Technology
l.m.g.feijs@tue.nl
[2]by-wire.net, Utrecht

Abstract

We report on the integration of fractal-like structures inspired by Appolonian circles with pied de poule (houndstooth) fabric patterns, and on the use of electric actuators in garments made with digital embroidery techniques.

Algorithmic packing of random circles

This project has two aims: first is to study and implement a fractal-like structure of circles inspired by Apollonian circles, combined with pied de poule (houndstooth) [1]. The 2nd aim is to push the integration of electric actuators in garments, using the power of algorithmic design and digital manufacturing. The result is a unique electronically enabled garment. Embroidery-machine patterns are made in Tajima file format. Flat coils of copper with magnets work as vibration actuators in the garment (like rotary vibration motors, but are better integrated in the garment and matching fashion production methods).

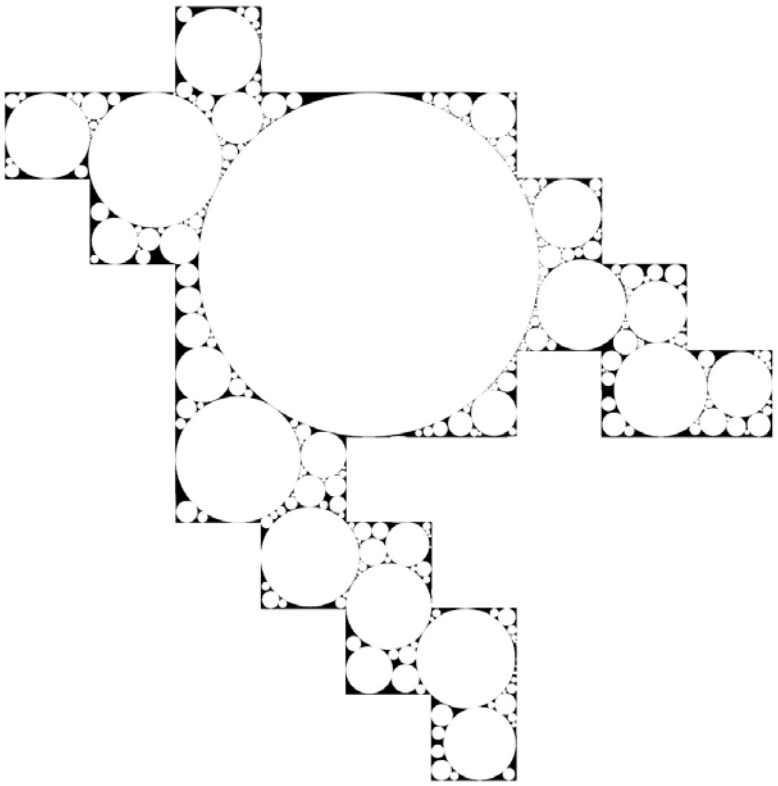

Figure 1: *Fractal like circle packing inside pied de poule figure.*

The center of the circles are chosen by a random generator. The packing is less regular than the circle fractal [2]. If the candidate point lies in an existing circle, it is rejected and a new candidate is generated. If the point falls outside the pied de poule figure [1] it is also rejected. The circle grows till it cannot grow further. The growth resembles algorithm 2.2.5 in [3], but is more sophisticated: the center is shifted over a small distance in different directions. Thus 10 candidate circles are developed: the largest is taken, 9 rejected. This is repeated until no more progress is obtained. It is not always the largest circle possible in the given situation (usually it has grown fat first and cannot tunnel anymore to a larger place). This is the algorithm in Processing (where the pair (`cx`,`cy`) is the circle center and `prev` means previous):

```
void expandStep(SegmentList segments, CircleList circles){
     float prevcx = cx;
     float prevcy = cy;
     final int NROPTIONS = 10;
     float[] dx = new float[NROPTIONS];
     float[] dy = new float[NROPTIONS];
     float dphi = radians(360)/(NROPTIONS-1);
     for (int i=0; i < NROPTIONS; i++){
          dx[i] = cos(i * dphi);
          dy[i] = sin(i * dphi);
     }    int aha = -1;
     float myMax = -1000;
     for (int i = 0; i < NROPTIONS; i++){
          r  = 0;
          cx = prevcx + dx[i];
          cy = prevcy + dy[i];
          float maxr = expandInSitu(segments,circles);
          if (maxr >= myMax){
              aha = i;
              myMax = maxr;
     }    }
     r = 0;
     cx = prevcx + dx[aha];
     cy = prevcy + dy[aha];
     expandInSitu(segments, circles);
}
```

What we embroider approximates a mathematical ideal (150 circles). For embedding electric coils in the circle packing, we had to decide on the number of electric coils. We also had to decide what would be meaningful connectivity points? From Heesch Kienzle theory [4] we know that a pied de poule pattern has type TTTTTT and therefore a network topology in which each cell figure has 6 connection points. We decided that these would be the connection points (e.g. for soldering). Each coil needs two connectors so we decided to go for three electric coils. Coding is in Processing, using Oogway [5].

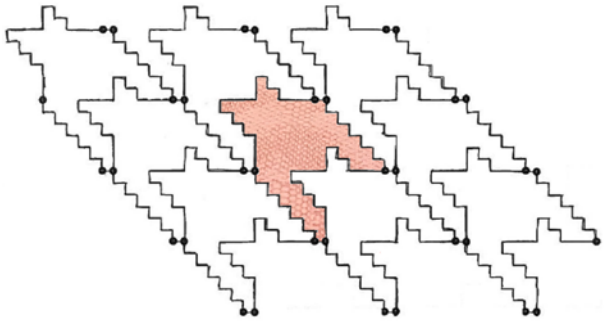

Figure 2: *Network structure of pied de poule in Heesch Kienzle theory*.

Embroidering

Roughly speaking the larger circles are generated first and then gradually the holes are filled by smaller and smaller circles. For embroidery however, the order is different. Three circles of average size are special: they are to made of conductive yarn and have to be embroidered last (to avoid the stitches of later loops to damage the conductive yarn). Avoiding unnecessary long jumps between circles is a travelling-salesman problem (approximation: greedy). Distance is Euclidean distance, except when the jumpstitch would cross one of the three special circles (in which case a penalty is added).

Figure 3: *The Brother Entrepreneur® PR655 embroidery machine in action.*

We experimented with Litze type yarn and enamel-coated copper wire. Litze type is easily embroidered (in the under-bobbin) but the needle sometimes cuts through adjacent loops and gradually interrupts the inner threads, increasing resistance (Ohms) or causing an open circuit. The experimentation process was very time-consuming. For the time being we are limited by the dimensions of the embroidery machine's area, which is 20 x 30cm. Pied de Pulse fits in 19 x 19cm. A typical pattern has 70,000 stitches.

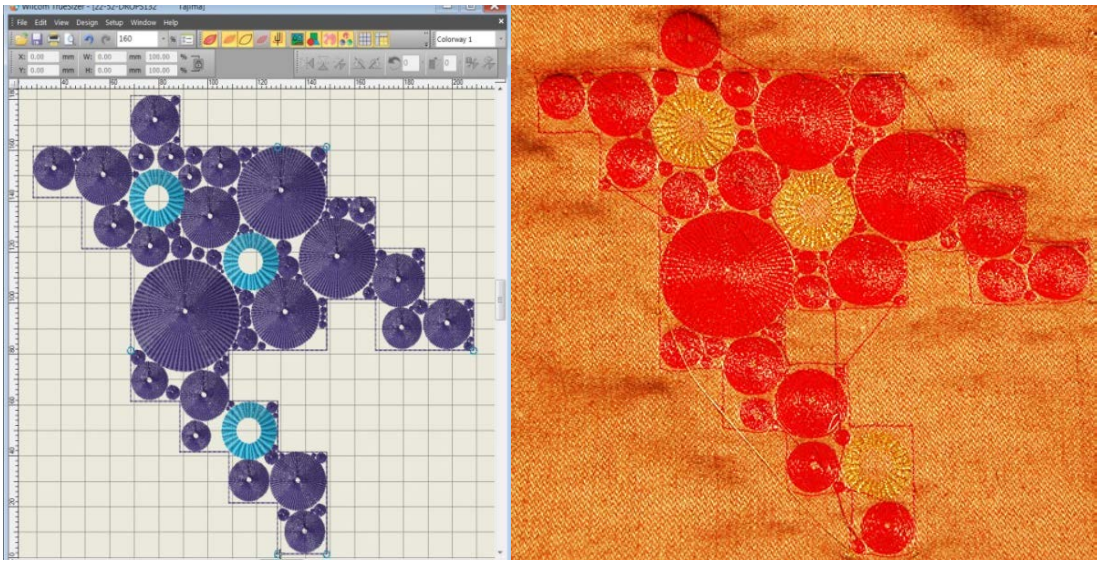

Figure 4: *Wilcom simulated and realised Brother embroidered circle packing (3 conductive-yarn coils).*

The jacket is shown in Figure 5. The coils are .5 - 1.0 Ω, actuated by an Arduino and H bridge circuit at 25 Hz (current 1.5A) . Typical operation is pulsed (1s pulse). This can be used for signalling events to the wearer of the jacket. We envision many possible applications: bodily parameters and speed indications for sports, messages from mobile networks, friendly notifications from your garments, and so on.

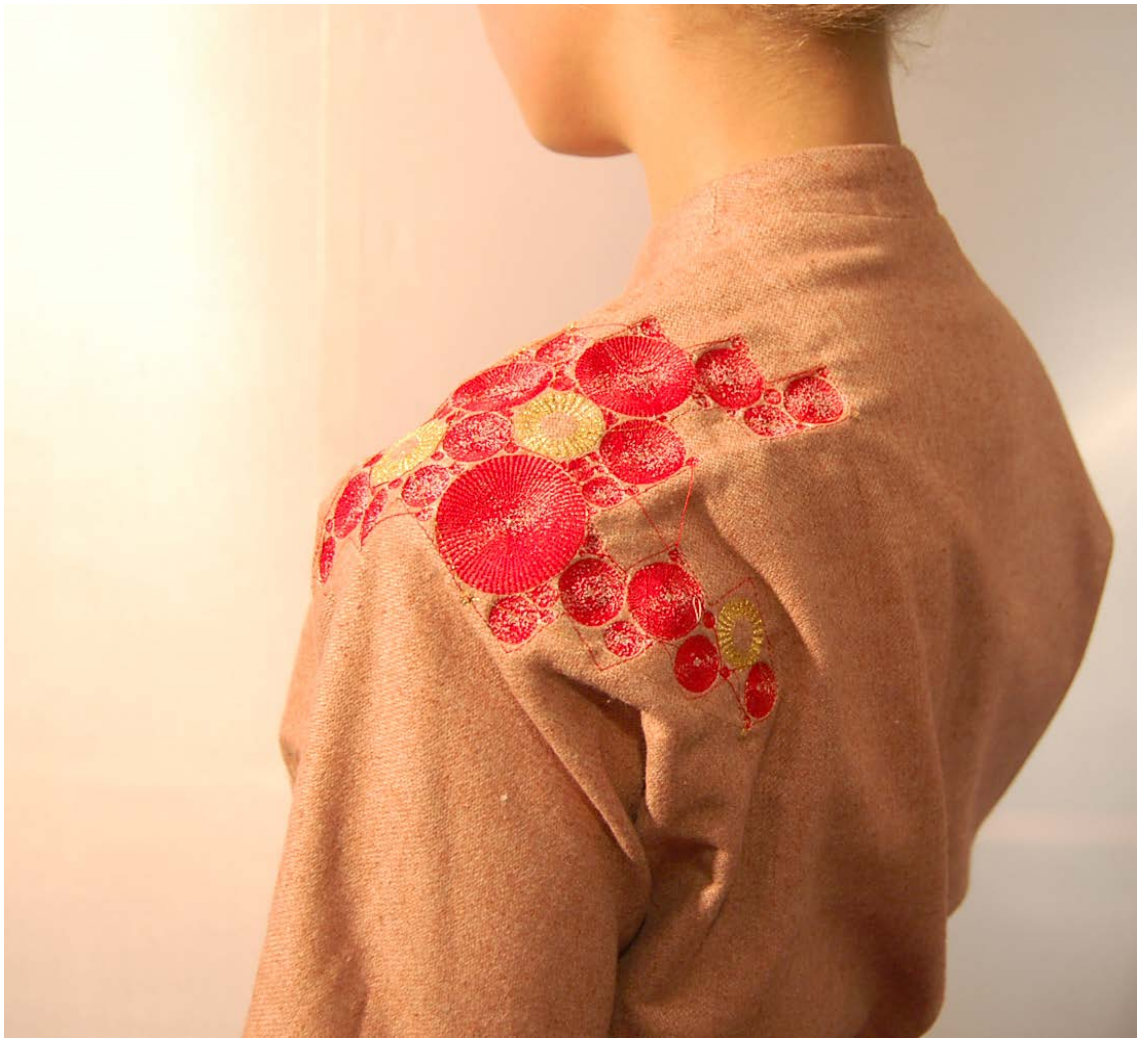

Figure 5: *Pied de Pulse jacket.*

Acknowledgements. We like to thank Lilian Admiraal, Geert van den Boomen, Giorgia Presti and Katinka Feijs for their great help and support during the development of this project.

References

[1] L.M.G. Feijs. *Geometry and Computation of Houndstooth (Pied-de-poule)*, In: Bosch. R., McKenna, D., and Sarhangi, R. (Eds.), Proc. of Bridges 2012, Baltimore, Maryland, pp. 299-306 (2012).
[2] D. Dunham and J. Shier. *Fractal wallpaper patterns.* Proc. of Bridges 2015, pp. 183-190.
[3] H. Bohnacker, B. Groß, J. Laub, & C. Lazzeroni. *Generative gestaltung*. Verlag Hermann Schmidt (2009).
[4] H. Heesch. and O. Kienzle. (1963). *Flächenschluss; System der Formen lückenlos aneinanderschliessender Flachteile.* Berlin, Springer.
[5] L.M.G. Feijs, J. Hu, *Turtles for tessellation*. Bridges 2013 (241–248). github.com/iddi/oogway-processing/