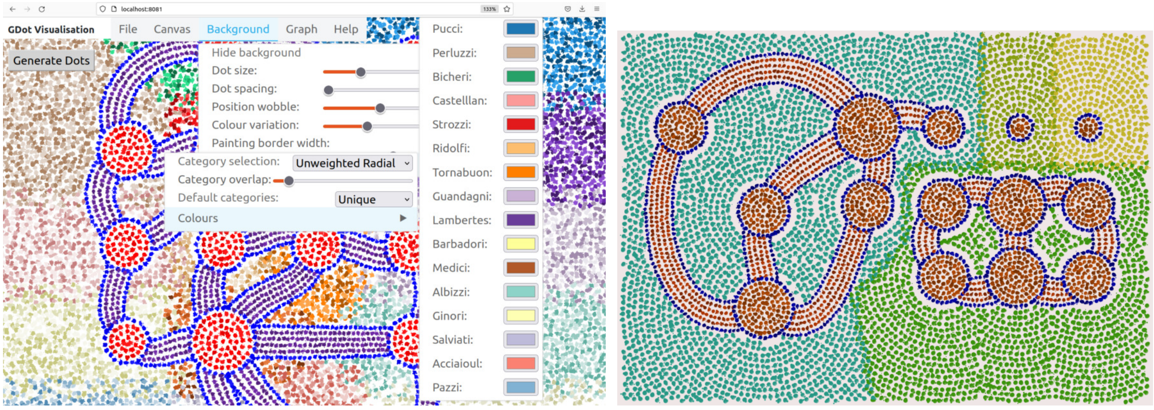
GDot-i: Interactive System for Dot Paintings of Graphs

Peter Eades, Seok-Hee Hong, Martin McGrane, and Amyra Meidiana

The University of Sydney, Australia



(a) *GDot-i interface* **(b)** *GDot-i artwork*

Figure 1: *(a)* ***GDot-i*** *user interface. (b)* ***GDot-i*** *artwork for* Aesthetic Experience Network *from the 2022 Graph Drawing Competition.*

Abstract

*This poster presents **GDot-i**, an interactive system visualizing graphs and networks as dot paintings, inspired by the dot painting style of Central Australia. We describe the implementation of **GDot-i**, a web-based interactive system, including the user interface and typical use cases.*

CCS Concepts

• ***Human-centered computing*** → ***Graph drawings;***

1. Introduction

This poster presents **GDot-i**, an *interactive system* for creating artworks for graphs and networks as dot paintings, inspired by the dot painting of the APY (Anangu-Pitjantjatjara-Yankunytjatjara) style of Central Australia. The dot painting style is established as a powerful medium for communicating information with abstraction, and has a long history of supporting storytelling. Specifically, **GDot-i** system implements the GDot framework [HET21], which visualizes graphs and networks in dot painting styles, using the following two layers:

The background layer consists of an array of dots across the whole canvas. Each dot can be rendered as a simple colored disk, or a more complex "splat" approximating a paintbrush dab. The dots can be colored to form a Voronoi tessellation of the plane, which shows clusters of nodes; see Fig. 1b.

The foreground layer consists of dot representations of nodes and edges. The nodes are concentric circles of dots. The edges are parallel Bezier curves of dots. The Bezier curves can be straight-lines, quadratic, cubic, or sequences of smoothly joined cubic curves. The control points for the Bezier curves can be chosen in a variety of ways.

For detailed algorithms and techniques for GDot such as specialized Bezier curves and Voronoi tessellation algorithms, see [HET21]. In this poster, we mainly describe the implementation of **GDot-i**, a *web-based interactive system*, including the interactive user interface and a typical use case.

2. Implementation of GDot-i

We have developed **GDot-i**, an interactive web application of GDot, see http://gdot.australiaeast.cloudapp.azure.com:8081/. The system consists of the following three main components:

DOI: 10.2312/evp.20221106

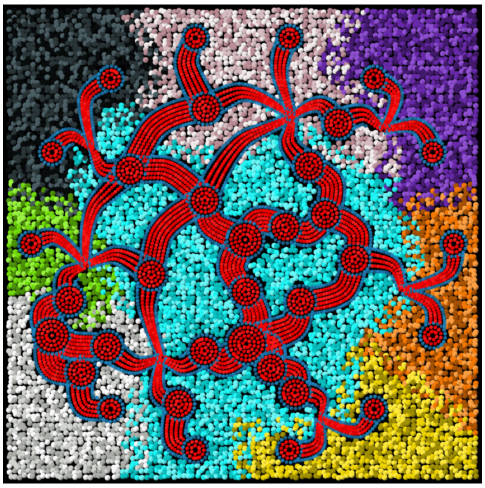


Figure 2: *Graph of relationships between European royal families.*

Figure 3: *A **GDot-i** picture in a gallery. This network is from Moreno's data on relationships between primary school children.*

- The *layer computation component* implements all the technical functions to compute dots for the background and foreground layers. It takes a graph in GML format as input and then computes locations for the dots.
- The *dot rendering component* implements the functions to render the appearance of the dots, drawing the dots and applying functions to control the visual variables such as color and shape.
- The *interactive user interface component* displays the output of the dot rendering functions and supports interactions that control the layer computation and dot rendering components.

3. Interactive User Interface of **GDot-i**

Fig. 1a shows the interactive user interface of **GDot-i**, which support extensive customization of the artistic aspects of the GDot painting. The rendering of the dots, such as size, position, color and spacing can be modified by the user interface. Colors for nodes and edges can be included in the input GML, and colors of the nodes, edges and Voronoi regions can be modified using the interface.

Users can control the number of bends per edges, from zero bends (i.e., straight-lines), a single bend (a quadratic Bezier curve), to up to eight bends (a chain of cubic Bezier curves). Inserted bends can be further adjusted by applying a force-directed algorithm, which can be constrained on the magnitude of bends, and the locations of bends within the Voronoi region.

4. Use Case of **GDot-i**

A typical use case for **GDot-i** is as follows:

1. An artist selects a graph G of interest, and computes a drawing D of G, using a graph layout tool. They may select colors and sizes for nodes and edges using the tool, which saves the resulting graph drawing as a GML file.
2. The artist opens the GML file, and creates a dot painting of G using **GDot-i**. Initially, the painting uses either the colors and sizes from the input GML file, or uses default colors and sizes.
3. The artist applies their creative artistic expression by adjusting the rendering of the background and foreground layers, such as:
   - the node size and edge thicknesses;
   - the edge bends (i.e, style of the Bezier curves);
   - dot colors, dot shapes and the spacing between the dots;
   - the background tessellation style and colors.
4. When satisfied, the artist saves the dot painting as SVG or PNG file for printing of the art work.

5. Art Gallery and Work in Progress

Fig. 2 shows the familial relationships between European royal families, where nodes represent monarchs and their siblings, and edges represent marriage or parent-child/ancestor-descendent/sibling relationships. The background is colored by country, where the central blue region with many nodes shows English monarchs, highly connected to most of Europe. Another example, printed and placed into a gallery, is in Fig. 3.

GDot-i aims to produce artworks of graphs and networks, and evaluation of such artworks are more subtle, and inherently subjective. We have received positive informal feedback to our artworks so far, and plan to hold an exhibition to further improve the dot paintings computed by the system.

Acknowledgement

We acknowledge and pay respect to the Gadigal people of the Eora Nation, custodians of the land where this research was undertaken. This research was supported from the Australian Research Council (ARC) Linkage Project LP160100935 with Oracle Labs Australia, and ARC Discovery Project DP190103301.

References

[HET21] Hong S.-H., Eades P., Torkel M.: Gdot: Drawing graphs with dots and circles. In *2021 IEEE 14th Pacific Visualization Symposium (PacificVis)* (2021), IEEE, pp. 156–165. 1