Using Indexed-Sequential Geometric Glyphs to Explore Visual Patterns

Jim Morey and Kamran Sedig

Cognitive Engineering Laboratory
Department of Computer Science
The University of Western Ontario, Canada
{jmorey,sedig}@uwo.ca

Abstract. This paper presents a visualization tool called PolygonR&D for exploring visual tiling patterns. To facilitate the exploration process, PolygonR&D uses dynamically-generated, interactive geometric glyph visualizations that intermediate reasoning between the sequential textual code and the parallel visual structure of the tilings. Sequential textual code generates indexed-sequential geometric glyphs. Not only does each glyph represent one procedure in the sequential code, but also a constituent element of the visual pattern. Users can reason with a sequence of glyphs to explore how tiling patterns are constructed. Alternatively, they can interact with glyphs to semantically unpack them. Glyphs also contain symbolic referents to other glyphs helping users see how all procedures work together to generate a tiling pattern. Experimenting with indexed-sequential glyphs in tools such as PolygonR&D can help us understand how to design interactive cognitive tools that support reciprocal reasoning between sentential and visual structures.

1 Introduction and Background

Mathematics has been described as the science of patterns [1]. Visual tilings are an example of mathematical patterns that are all around us [1, 2]. One of the best ways to investigate many mathematical concepts is to interact with their representations using computational cognitive tools—interactive tools that support and enhance cognition in the process of reasoning and experimentation [3]. Gaining insight into many ideas involves reasoning with multiple forms of representations of those ideas and interacting with those representations using different interaction styles and methods [3, 4, 5]. This is true of mathematical patterns. Due to their flexibility, malleable form, and dynamic nature, computational tools can easily present users with different representational forms of mathematical ideas and various interaction styles, allowing for different reasoning [3, 5]. In this paper, we are interested in investigating how to explore geometric tiling patterns using different representational forms.

Two forms of representation of geometric tilings include descriptive and visual [1, 2]. The first form is textual, sentential, and language-like; it is linear and sequential in nature; it linguistically describes how the tiling can be constructed. The second form is visual or diagrammatic; it is spatial and parallel in nature; it visually shows the structure of the tiling. These two forms of representation are informationally

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3038, pp. 996–1003, 2004.

equivalent, that is, they convey the same information. However, they are computationally non-equivalent, that is, they require different amounts and types of cognitive effort and can lead to different forms of knowledge [6]. These alternative forms of representation are complementary since they direct our attention to different aspects of their objects which we may otherwise overlook [4, 7]. The textual form communicates information about the logical sequence and ordering of the tilings, and the visual form conveys information about the geometry and structure of the tilings [6]. The textual form is cognitively processed in a sequential manner, whereas the visual form is cognitively processed in a more parallel and holistic manner.

This paper presents a visualization modeling tool called PolygonR&D, a programming environment that allows users to use a textual, procedural language to generate complex visual tiling patterns made of different types of polygons. Since tiling patterns are parallel visual structures, users can often find it difficult to reason about the constituent building blocks of such structures. Furthermore, reasoning back and forth between the sequential textual code and the parallel visual structure is not straightforward. Glyph, or iconic, visualizations can be used to facilitate comprehension of multi-dimensional data [8, 9]. These visualizations encode several dimensions of information, such as shape, size, and color, in a compact form to make these dimensions easily accessible to the users at a more perceptual level. Multidimensional glyphs have been used to encode different aspects of textual information. Such glyphs have proven to be easier and more efficient for users to process [10].

PolygonR&D uses dynamically-generated, interactive glyph visualizations to intermediate reasoning between the sequential textual code and the parallel visual structure of the tilings. Sequential textual code generates indexed-sequential geometric glyphs. Each glyph represents one procedure in the sequential code, as well as a constituent element of the visual pattern. Users can reason with a sequence of glyphs to explore how tiling patterns are constructed. Alternatively, they can interact with glyphs to semantically unpack them. Glyphs contain symbolic referents to other glyphs helping users see how all procedures combine to generate the tiling pattern.

2 Exploring Visual Tiling Patterns with PolygonR&D

PolygonR&D incorporates all the above representations: a sequential, textual representation, an intermediary, iconic representation, and a parallel, visuospatial representation. Fig. 1 shows how the environment of PolygonR&D[1] is separated into three panels each containing one of the representations of the tiling: Programming Panel (left) contains the sequential representation, Glyph Panel (centre) contains the intermediary representation, and Polygon Landscape Panel (right) contains the parallel spatial representation. Each of the panels is interactive: the Programming Panel allows for the standard text manipulations, the Glyph Panel allows for a semantic unpacking of the informationally-dense glyphs, and the Polygon Landscape Panel allows for interactive exploration of the execution of the programs.

[1] PolygonR&D is a java application that can be run in a browser with the Java plug-in http://www.csd.uwo.ca/~morey/CogEng/PolygonRnD.html

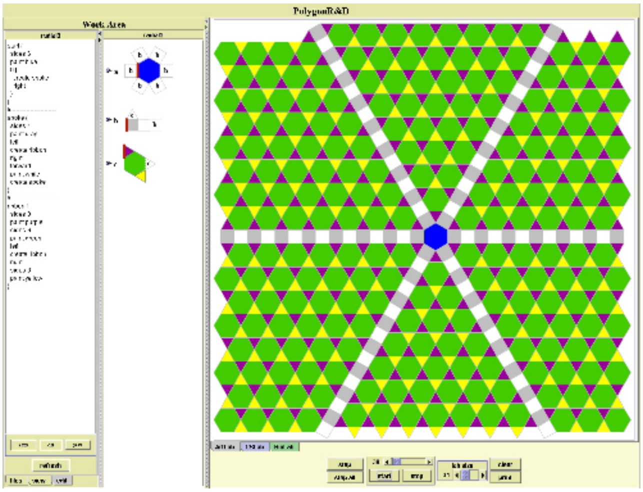


Fig. 1. A screen capture of the PolygonR&D

The sequential description of tilings is similar to the description of turtle movements in Logo except that the focus is on regular polygons rather than lines and the description allows for multiple turtle-like agents to perform the drawing[2]. The description is segmented into labeled procedures, paralleling the building blocks of the tiling. Table 1 shows the procedures that generate the tiling in Fig. 1, which is also present in its Programming Panel. The labels are used in the construction of new agents to reference the procedure the agent will perform. For instance, in the procedure labeled **start**, there is an instruction **create spoke**, which creates a new agent that performs the procedure labeled **spoke**.

Table 1. Example procedures associated with Fig. 1.

```
start {
  sides 6
  paint blue
  6 {
    create spoke
    right
   }
}
```

```
spoke {
  sides 4
  paint gray
  left
  create ribbon
  right
  forward
  paint white
  create spoke
}
```

```
ribbon {
  sides 3
  paint purple
  sides 6
  paint green
  left
  create ribbon
  right
  sides 3
  paint yellow }
```

[2] The environment can be described as a 2D-Turing machine with multiple polygonal heads that write coloured polygon symbols.

2.1 Exploring Textual and Glyph Representations

A glyph for each procedure is constructed and put into the Glyph Panel. The glyph visually represents the effects of executing its corresponding procedure. It is constructed by executing the procedure with the exception that the **create** instructions are not executed, but instead tokens are placed to visually indicate where the unexecuted instruction occurs. For instance, the upper left corner of Fig. 2 shows the glyph associated with the procedure **start**. The hexagon represents a portion of the tiling while the six dotted squares represent six tokens each indicating spatial places where the unexecuted **create** instructions are called or connected. The tokens reference the procedure **spoke** in two ways: visually, by depicting its starting shape (a square), and symbolically, by referring to its index (**b**). The square is the first shape that appears in the procedure **spoke**. The index **b** is the index given to **spoke**'s glyph. The glyphs are indexed according to the order that they are used in the construction of the tiling. In the above example, **start** is the first procedure used, hence indexed as **a**; **spoke** is the second procedure used, hence indexed as **b**; and **ribbon** is the third procedure used, hence indexed as **c.** The short indices help make the glyphs visually compact.

The creation of the list of glyphs from the procedures is not difficult for the computer but the reasoning back and forth between the glyphs and code can be difficult for a programmer. To aid this reasoning process, the glyphs are made interactive. Each glyph can be unpacked so that the details of its construction can be viewed. The details are shown as a labeled sequence of icons. The label is the original name of the procedure. The list of icons corresponds to the instructions of the textual code. The arrow icons represent motion, the polygons represent the introduction of a new shape, and the coloured ovals represent the placement of a coloured landmark. The icon for the **create** instruction is a miniature of the glyph without the indices. Fig. 2 shows the unpacked versions of the glyphs from Fig. 1. The icons' instructions provide spatial cues that help visualize the construction of the glyph. The inclusion of the miniature glyphs allows for the linear scanning of the icons to be uninterrupted with a look up of a glyph on the list.

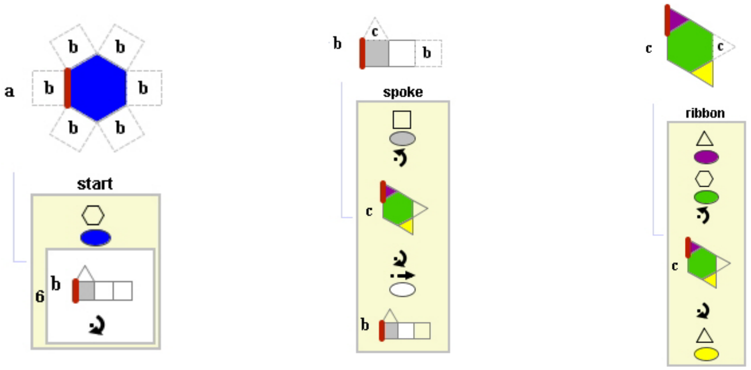


Fig. 2. Unpacked glyphs created from the code in Table 1.

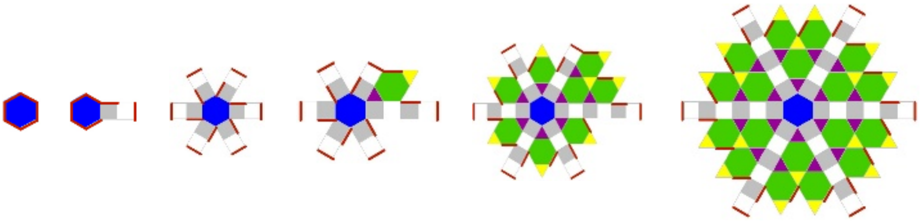


Fig. 3. A sequence of images created by the Polygon Landscape Panel during the execution of the code from Table 1.

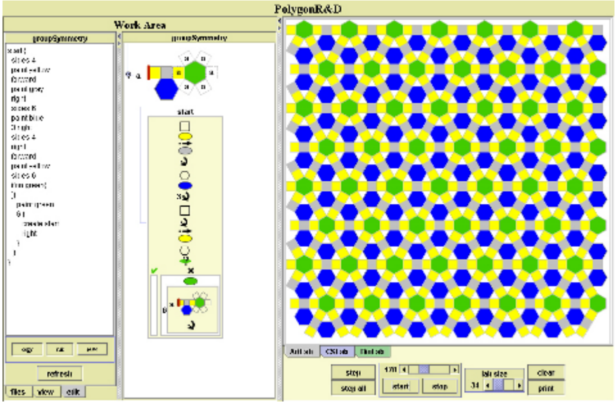
2.2 Exploring Glyph and Visuospatial Tiling Representations

Reasoning between the list of glyphs and the tiling takes place in the spatial domain. The act of creating the tiling is a matter of starting with the first glyph and replacing the indexed tokens with their referenced glyphs. Mentally visualizing the tiling using the glyphs is a matter of visually chunking copies of the glyphs into one mental image. This task can sometimes be difficult due to the mental manipulation of a large number of glyphs. The key to comprehending the tiling is to recognize replacement patterns so that large numbers of glyphs can be chunked together. For instance, the above example has a replacement pattern that can be noted in glyph **b**. This glyph creates another **b** in such a way that a line of squares is created. Mentally visualizing this pattern can be difficult, but it is easy to see after viewing the animated construction of the tiling in the Polygon Landscape Panel. Fig. 3 shows a sequence of images depicting the gradual construction of the tiling in the polygon landscape. The line-of-squares feature in the tiling is prominent and relating this feature back to the glyphs is a matter of scanning the list for squares. Unlike the squares, which are only present in glyph **b**, the hexagons in the tiling could potentially come from either glyph **a** or glyph **b**. The colouring of the glyphs can aid in disambiguating the origins of polygons in the tiling. For instance, the central hexagon must come from glyph **a** (not glyph **c**) since it is the only glyph that contains a hexagon of that colour.

2.2.1 Single Glyph Tilings

By their constructions, glyphs represent local neighbourhoods of the tilings. When tilings are defined by one local neighbourhood, one glyph can represent the entire tiling. For instance, Fig. 4 shows an example with a single unpacked glyph that creates a tiling made of squares, hexagons, and triangles. Visualizing this tiling using the glyph can be supported by watching the gradual construction of the tiling in the polygon landscape as in Fig. 3. Creating a glyph to produce the tiling is a matter of constructing a neighbourhood, which can act as a building block of the tiling, and then relating it to its identical nearby neighbourhoods. The difficulty of predicting the outcome of a glyph can be offloaded to the Polygon Landscape Panel.

There is a many-to-one relationship between glyphs and tilings. The left image of Fig. 5 shows an alternative glyph that produces the tiling from Fig. 4. This alternative construction demonstrates another way of understanding the tiling as a network of paths between with hexagons that branch into three paths (Fig. 5) instead of six paths (Fig. 4). Exploring alternative constructions can aid in developing to deeper understandings of the tiling can be achieved. Concretely, an elaboration of a glyph



```
start {
  sides 4
  paint yellow
  forward
  paint grey
  right
  sides 6
  paint blue
  3 right
  sides 4
  right
  forward
  paint yellow
  sides 6
  trim green {
  }
    paint green
    6 {
      create start
      right
    }
  }
}
```

Fig. 4. An example of a single glyph tiling.

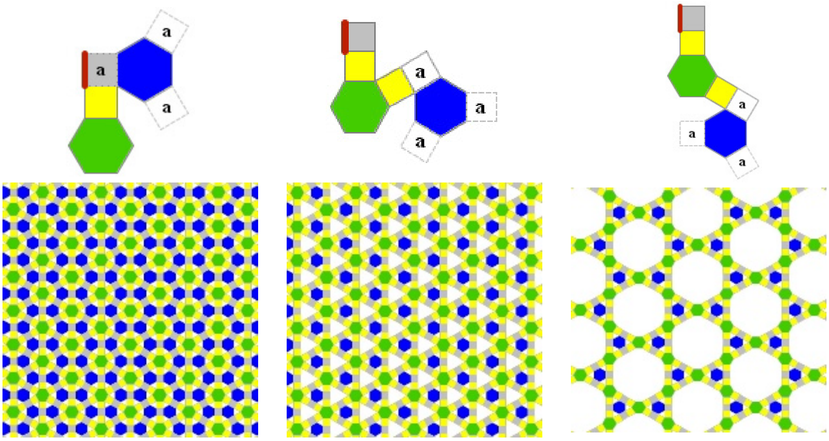
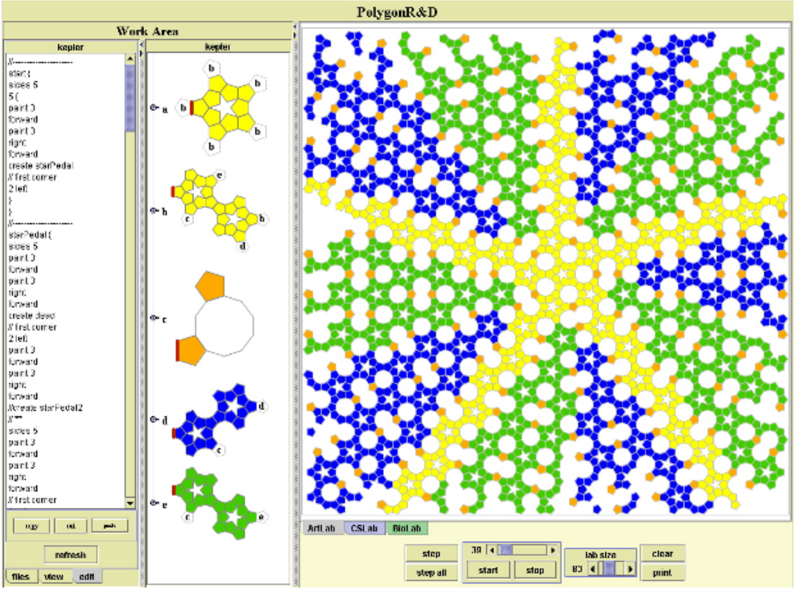


Fig. 5. Certain elaborations of the glyph on the left produce subsets on the tiling from Fig. 4

can expose the differences between alternative constructions of the same tiling. For instance, the second and third glyphs of Fig. 5 are elaborations of the first glyph. They result in tilings that difficult to create by elaborating the glyph from Fig. 4.

2.2.2 Negative Space

The tiling examples from Fig. 5 have features that may not be seen in the glyphs. For instance, the third example has large white rounded hexagons. PolygonR&D is restricted to using regular polygons as its basic shapes but the negative space, the gaps between the regular polygons, does not have this restriction. The creation of negative spaces often requires more experimentation in which the focus goes back and forth between the glyph construction and the resulting tiling. For example, Fig. 6 shows a complicated tiling designed by Kepler that incorporates five point stars and shapes with the outline of two fused decagons. The construction of this tiling was achieved in a piece-wise fashion where incomplete glyphs were used to partially construct the tiling. The partial tiling then helped to incrementally build and finish the glyphs.



```
//------------------
start{
sides 5
5{
part 3
forward
part 3
right
forward
create starPedal
// first corner
2 left
}
}
//------------------
starPedal{
sides 5
part 3
forward
part 3
right
forward
create deca
// first corner
2 left
part 3
forward
part 3
right
forward
//create starPedal2
//**
sides 5
part 3
forward
part 3
right
forward
// first corner
```

Fig. 6. A complicated tiling design by Kepler.

2.2.3 Multiple Glyph Tilings

Many tilings cannot be constructed from one local neighbourhood. Both the tilings from Fig. 1 and Fig. 6 are examples of this. These tilings are constructed by integrating a number of local neighbourhoods, which is reflected by their associated list of glyphs. Understanding how these local neighbourhoods are integrated seems to be facilitated by the glyph representation of the tiling's description. The glyph's visual description of the tiling often takes up less space than the textual description of the tiling. Notice in Fig. 6 that the complete set of glyphs that constructs Kepler's tiling is shown. At the same time, only a small portion of the textual description of the tiling is shown (one sixth, judging by the scroll bar). In this case, the amount of scrolling required when reasoning back and forth between the tiling and its glyphs is eliminated; this is not the case for the textual description.

Multiple glyphs can also aid in the construction of tilings that have only one local neighbourhood. One example would be if the defining neighbourhood were very large, in which case breaking down the neighbourhood into manageable sections may help organize its creation. A more important case occurs when the defining neighbourhood contains some smaller neighbourhoods that also repeat but do not define the tiling. In this way, repeating a section in the description is unnecessary.

3 Conclusions

PolygonR&D integrates index-sequential geometric glyphs to aid in bridging the text of the procedural description and the visuospatial pattern of the tiling. The transformation from a textual procedure to a glyph is supported by the glyph's interactive ability to be semantically unpacked into a list of icons. Each glyph is related to a section of the tiling. The sections connect and combine to form the tiling. The visual description of the list of glyph's interconnectedness is achieved through one-letter indices. The transformation from the glyph is supported by the interactive execution of the glyphs. Reasoning back and forth between the representations can aid in developing insight into the structure of tilings and how to construct them.

Experimenting with indexed-sequential glyphs in tools such as PolygonR&D can help us understand how to design interactive cognitive tools that support reciprocal reasoning between sentential and visual structures.

References

1. Devlin, K: Mathematics: The Science of Patterns. NY, Freeman and Company (2000)
2. Grunbaum, B, Shephard, GC: Tilings and Patterns: An Introduction. W.H. Freeman, New York (1989)
3. Sedig, K, Morey, J, Mercer, R, Wilson, W: Visualizing, interacting and experimenting with lattices using a diagrammatic representation. In G. Malcolm (Ed.), Multidisciplinary Approaches to Visual Representations and Interpretations. Elsevier Science (In press)
4. Peterson, D (ed.): Forms of representation. Intellect Books, Exeter, UK (1996)
5. Sedig, K, Rowhani, S, Morey, J, Liang, H: Application of information visualization techniques to the design of a mathematical mindtool: A usability study. Palgrave Macmillan Ltd. Information Visualization, (2003) 2(3): 142-16
6. Larkin, J, Simon, H: Why a diagram is (sometimes) worth 10000 words. In J. Glasgow, N. H. Narayanan, and B. Chandrasekaran (eds.), Diagrammatic reasoning: Cognitive and computational perspectives, The MIT Press (1995)
7. Myers, K, Konolige, K: Reasoning with analogical representations. In J. Glasgow, N. H. Narayanan, and B. Chandrasekaran (eds.), Diagrammatic reasoning: Cognitive and computational perspectives, The MIT Press (1995)
8. Post, FJ, van Walsum, T, Post, FH: Iconic techniques for feature visualization. In Proc. Visualization, IEEE Comp. Soc. Press (1995) 288-295
9. Ribarsky, W, Ayers, E, Eble, J, Mukherjea, S: Glyphmaker: Creating customized visualization of complex data. IEEE Computer (1994) 27(7):57-64
10. Spence R: Information Visualization. Addison-Wesley (2001)