Proceedings of Bridges 2013: Mathematics, Music, Art, Architecture, Culture

Math for Visualization, Visualizing Math

Jarke J. van Wijk
Dept. Mathematics & Computer Science
Eindhoven University of Technology
vanwijk@win.tue.nl

Abstract

I present an overview of our work in visualization, and reflect on the role of mathematics therein. First, mathematics can be used as a tool to produce visualizations, which is illustrated with examples from information visualization, flow visualization, and cartography. Second, mathematics itself provides challenges. Visualization of Seifert surfaces, regular maps, and the construction of Julia sets helps to understand these better.

1 Introduction

I am not an artist nor a mathematician. However, I do aim to produce images and animations that are interesting and useful, and also, I use mathematics, to solve problems as well as a challenge.

My topic is data visualization: the use of interactive computer graphics to get insight in large amounts of data. My main focus is information visualization, aiming at understanding abstract data. A typical example is our work on visualization of hierarchical data using treemaps [17, 1], which we used for instance to visualize the contents of hard disks. Fig. 1 shows an example, produced with SequoiaView [18]. Each rectangle denotes a file, the area shows its size, the color shows its type. Files in the same directory are shown close to each other, and we added hierarchical cushions to emphasize the directory structure. This tool has been used by many to understand why their disks are full.

Besides information visualization, I have explored other areas as well. In my talk I will show results of projects we have done, using demo's and videos. The two main themes are that math can help to produce visualizations, and that visualization can help to understand math.

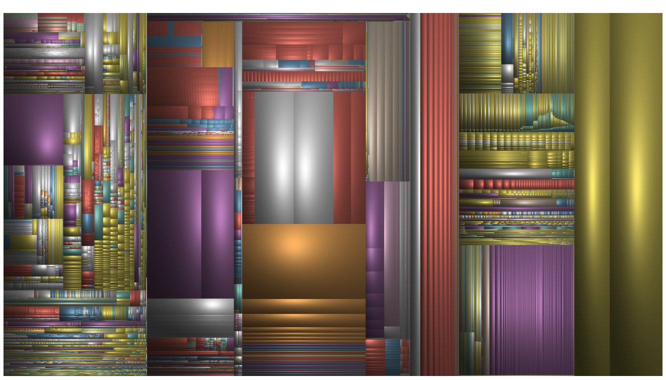


Figure 1: *SequoiaView: visualization of a hard disk*

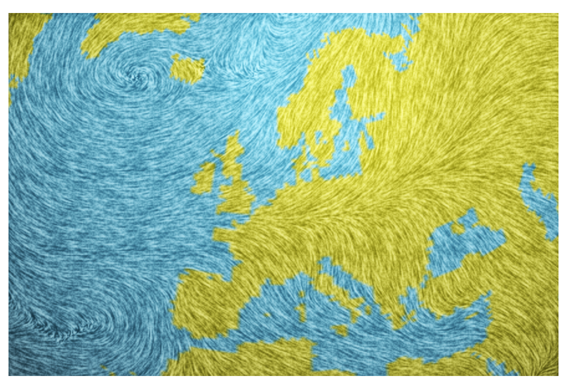


Figure 2: *Wind flow Europe in December*

2 Math for Visualization

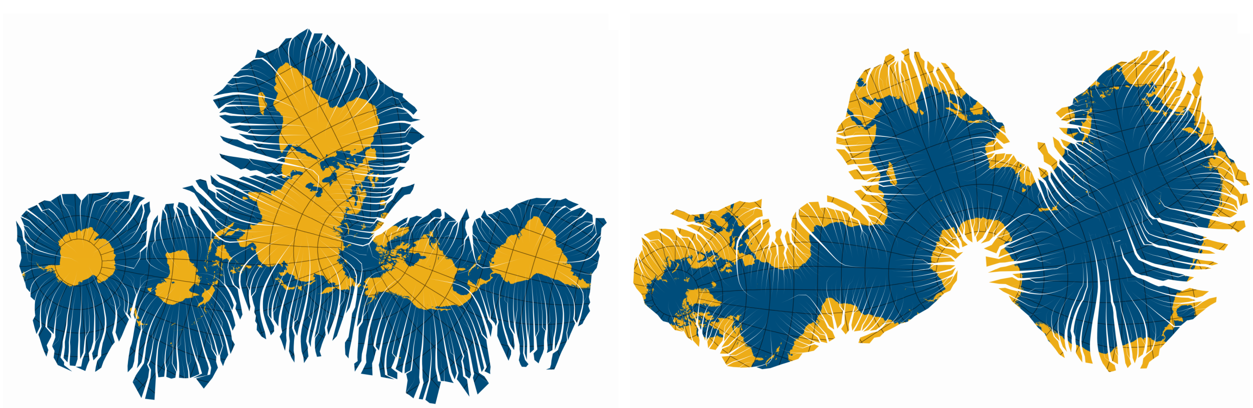
Mathematics provides a huge and powerful set of concepts, methods, and techniques to describe, understand, and manipulate complex phenomena in a concise way. This is most useful if you want to make images and animations to visualize data.

My favorite example how math can help to solve a problem concerns animation. Suppose, you look at a map of Enschede, and want to shift focus to Eindhoven. However, you want some nice animation, zooming out first, panning the camera next, zooming in again, to understand the relative position of these cities and not to loose context. The challenge now is how to define an optimal camera path and orientation. Mathematics provides perfect tools to model and describe this, and we finally found that geodesics in hyperbolic space provide a great solution [14, 15].

The key problem of visualization is to get insight into large and complex data sets. Given a problem, one approach I use is to try to imagine an optimal image first, using metaphors from the real world, art, and anything else. The next challenge is to translate such metaphors into mathematical models and algorithms to generate such images automatically, thereby trying to capture the essence in a compact and flexible model.

My work in flow visualization is an example. Flow visualization aims at showing vector fields, for instance to show the variation in wind direction and strength over a country. The flow of lava, the wear patterns on heavily used floors, and Van Gogh's starry night are examples that textures can vividly display flow, which inspired me to use texture as a metaphor for flow visualization. In the early nineties I developed a method to translate flow fields into textures, by splatting large amounts of small distorted shapes on a surface [8]. Theory from signal processing provided a sound base to understand and explain why and how this worked. Later on, I found that similar effects could be achieved much faster by repeatedly distorting images according to the flow and adding noise patterns, and again, math helped here greatly to describe and understand the effects [9]. This method turned out to be fast enough to process video input in real-time. Also, this method can be used to generate textures on surfaces in 3D, to visualize flow patterns on surfaces, such as wind flow patterns on the earth (Fig. 2) as well as to emphasize shapes or produce non-photorealistic renderings [10]. Finally, we used this method to generate animations of Iterated Function Systems [16] in real time, by copying, duplicating, scaling, and adding images.

Mapping the earth is a classical problem, which has intrigued cartographers and mathematicians for centuries. It is not possible to flatten a sphere in a perfect way. One can try to get the local angles right



(a) *Continent centered*

(b) *Ocean centered*

Figure 3: *Myriahedral projections of the earth*

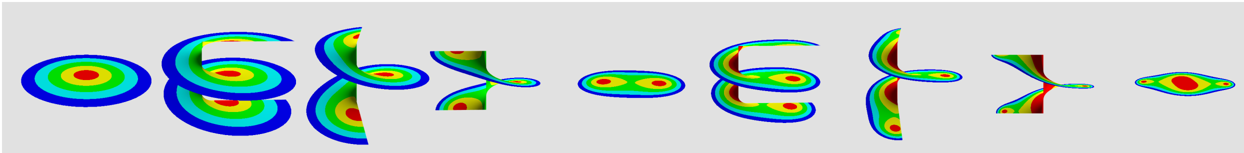
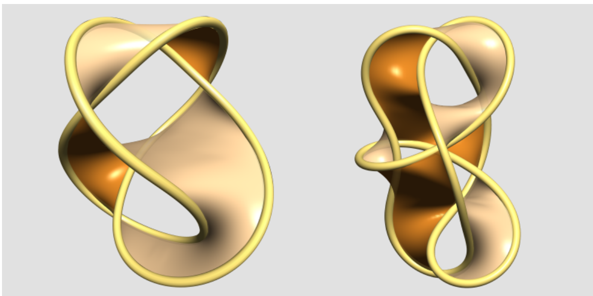
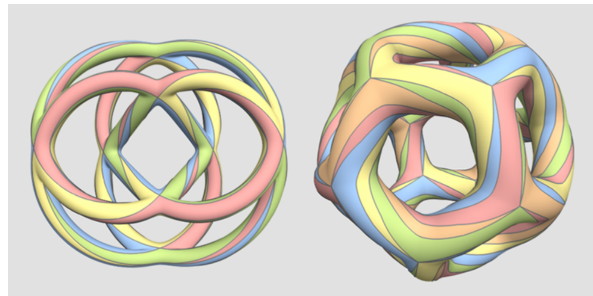


Figure 4: *Visualization of the generation of Julia fractals*



(a) *Seifert surfaces of knots* 4_1 *and* 7_7



(b) *Regular maps R9.4' and R11.1*

Figure 5: *Visualization of mathematical objects*

(like in the Mercator projection), one can get areas right (but at the expense of distortion), or opt for some compromise. However, if we do not care for interrupts in the map, things get a lot easier, and distortion free maps can be produced. The metaphor used here is the peel of an orange, which can be flattened easily and thereby produces an intricate shape. This led to Myriahedral Projections [11]. The globe is projected on a polyhedron with many faces (a myriahedron), next, this shape is folded out. By making different choices for the mesh and where to fold and cut, a wide variety of strange maps can be produced. For instance, one can try not to cut through continents, but also, we can try to keep the oceans together (see Fig. 3).

3 Visualizing Math

Besides as a tool, mathematics is also a challenge in itself: given some mathematical concept, try to visualize it to understand it better. Images of fractals are a classic example how mathematics can lead to fascinating imagery. But, it is hard to understand how the simple definition of for instance Julia fractals leads to such complex shapes. We recently worked on providing visual explanations [4]. Inspired by the sour dough metaphor, we show how Julia fractals can be produced by repeatedly duplicating, morphing, and adding shapes, leading to increasingly complex images. Fig. 4 shows key-frames from the first two steps in this process, repeating these steps leads to a fractal shape. Here the deforming surface is modeled as a kind of corkscrew, we also implemented three other solutions to show the deformation of the surfaces.

Some years ago, my colleague Arjeh Cohen, professor in discrete mathematics, asked me if I could make pictures of Seifert surfaces. Of course I did not know what these were, but soon became highly fascinated. A Seifert surface is an orientable surface that has a mathematical knot as its boundary. This sounds strange, and indeed, these surfaces are strange, though they can be embedded regularly in 3D space. We developed a method to produce images of Seifert surfaces [13], and very recently extended this method to obtain minimal genus variants [6]. The images (Fig. 5a) were made with SeifertView [7], which is freely available for download. Barbara Grossman used this tool to produce wonderful sculptures [3].

Regular maps can be viewed as a generalization of the classic Platonic solids. Each vertex of a cube is perfectly equivalent to all other vertices, and the same holds for all edges and faces. Regular maps are embeddings of such highly symmetric graphs on surfaces with a higher genus. For a torus these are trivial to generate, for objects with more holes it is a much bigger challenge. Mathematicians have calculated all solutions for objects up to genus 101 [2], but they only give the structure, and not the geometry of the shapes. One approach to find solutions is to use physical models, smart heuristics, insight in 3D structure, and a lot of patience and persistence, an approach where Carlo Séquin is the master [5]. I came up with an automatic method, and succeeded to find solutions for a number of cases [12]. Fig. 5b shows two examples: 32 hexagons on a genus 9 surface (R9.4') and 60 quads on a genus 11 surface (R11.1). However, there are still a lot of these regular maps for which no symmetrical visualization models have been constructed so far.

4 Conclusion

I hope you agree that the combination of mathematics and visualization leads to useful and intriguing results.

References

[1] D.M. Bruls, C. Huizing, and J.J. van Wijk. Squarified treemaps. In *Proc. Data Visualization 2000*, pages 33–42. Springer, Vienna, 2000.

[2] M. Conder and P. Dobcsányi. Determination of all regular maps of small genus. *J. Comb. Theory, Ser. B*, 81(2):224–242, 2001.

[3] B. Grossman. Borromean rings. `http://www.bathsheba.com/math/borromean`.

[4] O.J. Schrijvers and J.J. van Wijk. Visual explanation of the complexity in Julia sets. *Computers & Graphics*, 32(3), 2013.

[5] C.H. Séquin. My search for symmetrical embeddings of regular maps. In *Proc. Bridges 2010 Conference*, pages 85–94, Pécs, Hungary, July 24-28 2010.

[6] M. van Garderen and J.J. van Wijk. Seifert surfaces with minimal genus. In *Proc. Bridges 2013*, Enschede, The Netherlands, July 27-31 2013.

[7] J.J. van Wijk. Seifertview. `http://www.win.tue.nl/~vanwijk/seifertview`.

[8] J.J. van Wijk. Spot noise: texture synthesis for data visualization. In *Proc. ACM SIGGRAPH 1991*, pages 309–318, 1991.

[9] J.J. van Wijk. Image based flow visualization. *ACM Trans. Graph.*, 21(3):754, 2002.

[10] J.J. van Wijk. Image based flow visualization for curved surfaces. In *Proc. IEEE Visualization 2003*, pages 123–130, 2003.

[11] J.J. van Wijk. Unfolding the earth: Myriahedral projections. *The Cartographic Journal*, 45(1):32–42, 2008.

[12] J.J. van Wijk. Symmetric tiling of closed surfaces: visualization of regular maps. *ACM Trans. Graph.*, 28(3):12p., 2009.

[13] J.J. van Wijk and A.M. Cohen. Visualization of Seifert surfaces. *IEEE Trans. Vis. Comput. Graph.*, 12(4):485–496, 2006.

[14] J.J. van Wijk and W.A.A. Nuij. Smooth and efficient zooming and panning. In *Proc. IEEE InfoVis 2003*, pages 15–23, 2003.

[15] J.J. van Wijk and W.A.A. Nuij. A model for smooth viewing and navigation of large 2D information spaces. *IEEE Trans. Vis. Comput. Graph.*, 10(4):447–458, 2004.

[16] J.J. van Wijk and D. Saupe. Image based rendering of iterated function systems. *Computers & Graphics*, 28(6):937–943, 2004.

[17] J.J. van Wijk and H. van de Wetering. Cushion treemaps: Visualization of hierarchical information. In *Proc. IEEE INFOVIS'99*, pages 73–78, 1999.

[18] J.J. van Wijk et al. Sequoiaview. `http://www.win.tue.nl/sequoiaview`.