Looking for Lovely Links in Lattices

Anton Bakker[1] and Tom Verhoeff[2]

[1]Norfolk, Virginia, USA; antonbakker.com, antonbakker30@gmail.com
[2]Department of Mathematics and Computer Science, Eindhoven University of Technology, Netherlands; T.Verhoeff@tue.nl

Abstract

We describe a semi-automated way to find nice links of two or more given 3D paths. These paths are typically generated in a particular lattice and are given a particular cross section. The goal is to find links where the paths are truly linked (entangled) and where they fit snugly together. A special use case is where the paths are all congruent. The search can be controlled by a number of parameters. We also show some links that were found this way.

Introduction

In the field of mathematics known as Knot Theory, a knot is a closed path in 3D space. The simplest knot, also known as the trivial knot or the unknot, is a path that can be transformed continuously (that is, without self intersection, or cutting and gluing) into a circle. Closed paths are relevant in mathematical art, because unlike open paths there is an additional challenge when dressing ('thickening') the path using a profile (cf. [4]). One can compare this to the challenge in music composition of ending in the opening key. Without the requirement of proper closure, the artist could just do anything. Mathematical artist Koos Verhoeff (1927–2018, [2][3]) often used piecewise linear paths in 3D space, where beams are connected by miter joints.

Figure 1: *Links by Koos Verhoeff: Trinity III, II, and I (left, wood, 14 cm); Four Unity (right, wood, 40 cm).*

Multiple knots can be combined into what is known as a link. Figure 1 shows four links by Koos Verhoeff. The three links on the left have an octahedron as convex hull and each consists of three identical copies of a planar path. The one on the right consists of four copies of the same non-planar hexagon. For Trinity I and Four Unity, see [2], and for Trinity II and III, see [7]. Other links by Koos are described in [6][8].

Both authors have collaborated with and have been inspired by Koos. This article addresses the problem of automatically finding nice[1] ways of entangling multiple closed 3D piecewise linear paths whose endpoints are in a lattice (such as the Simple, Face-Centered, and Body-Centered Cubic lattices, commonly known as SC, FCC, and BCC) reminiscent of Koos' designs.

[1]The article's title captures this by 'lovely', which is a nod to the Hopeless Love designs described in [8].

Search Process

Overall, the search process involves four consecutive phases, that we will detail below:

1. Generate path variations, restricting to the top N based on proximity.
2. Combine path variations into prospective links, filtering out path combinations that intersect.
3. Rank links based on, among others, a convex hull analysis.
4. Profile the paths in the links and render them for visual assessment.

Generate path variations The link-finding tool is fed with two (polyline) paths, which we name path A and path B. Path B can be identical to path A. Path A is fixed in location and orientation. For path B, the tool generates variations by translating, reflecting, and rotating it in all kinds of ways, constrained by the lattice. Each of these operations can be restricted in various ways. For instance, the translation distance can be capped and its direction can be constrained. Similarly, reflections and rotations can be constrained. Note that these transformations are discrete due to the underlying lattice. Variants of path B that intersect or overlap with path A are filtered out. Typically, this gives rise to thousands of variants of path B. Of these, we keep the top N based on proximity to path A, measured as distance between their centroids. Parameter N is user selectable, and typically is set between 100 and 800.

Combine path variations This phase starts with path A and repeatedly adds a variant of path B. Currently, the tool supports adding up to seven variants of path B, yielding links consisting of up to eight paths. This phase gives rise to another combinatorial explosion, which is 'managed' by sorting and capping after each addition. Of course, here also intersections are filtered out. No further filtering or capping is done at the end and all valid combinations are passed on to the next phase.

Rank links The goal of this phase is to find truly entangled paths that moreover are interesting in some sense. There are no standard algorithms for this. Instead, the tool can rank the links according to various heuristic computational criteria (no AI, yet), selectable by the user. The aim is to prioritize links that are compact or not, or that have regularities or not. We discuss a few of these criteria. The first set of criteria is based on the link's convex hull, i.e., the smallest polyhedron that encloses the (unprofiled) link. Ranking can be based on: largest volume, smallest volume, and smallest number of vertices. There are also three criteria involving the hull's face areas rounded to integer values:

- Collect them in a set (collapsing duplicate areas), and rank by smallest set size.
- Determine minimum and maximum face area, and rank by smallest difference.
- Rank by most unique face areas (corresponding to less symmetry).

The second criterion is based on the convex hulls of the individual paths, ranking more intersections among these hulls higher. The third criterion concerns the proximity of the paths' centroids.

Profile all paths Finally, the user selects a profile (polygon, rotation, scale) and some finishing parameters to get profiled and rendered output in Rhinoceros. The ranking helps to bring more interesting links to the attention. The ranking criteria can be updated interactively to quickly explore the vast space of link candidates and select the most beautiful and intriguing ones.

Implementation

The link-finding tool is implemented in Rhinoceros as a Grasshopper definition (network of interconnected blocks, including custom Python code). It serves as a post-processor for the path generator based on Anton's Path Language described in [1]. The user sets paths A and B, selects various parameters on the top-level Grasshopper blocks for the four phases, and triggers the search process. Ranked links can be laid out in an XY grid, or one specific link can be selected via its index. The rendered results are shown in Rhinoceros.

Examples

We present some links found via the tool and rendered with Rhinoceros in Figures 2, 3, 4, and 5.

Figure 2: *A link consisting of 2 paths in FCC.*

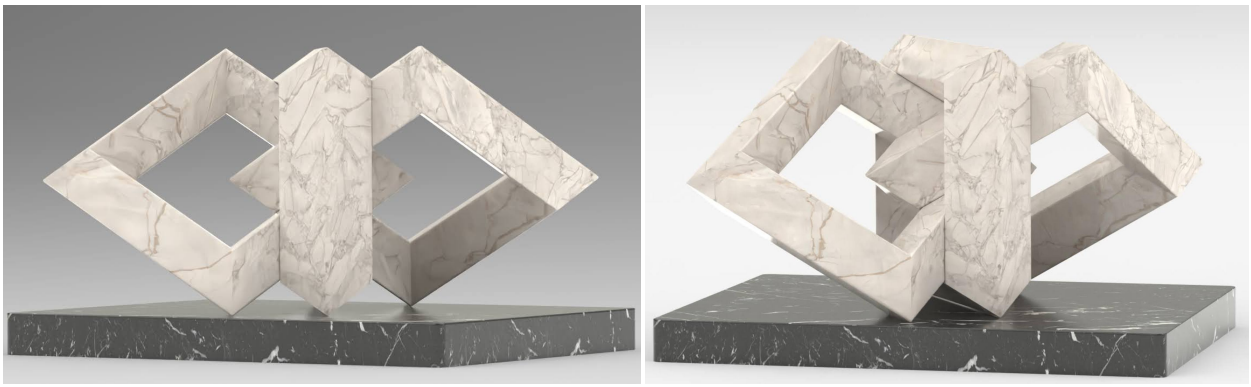

Figure 3: *A link consisting of 3 squares in FCC.*

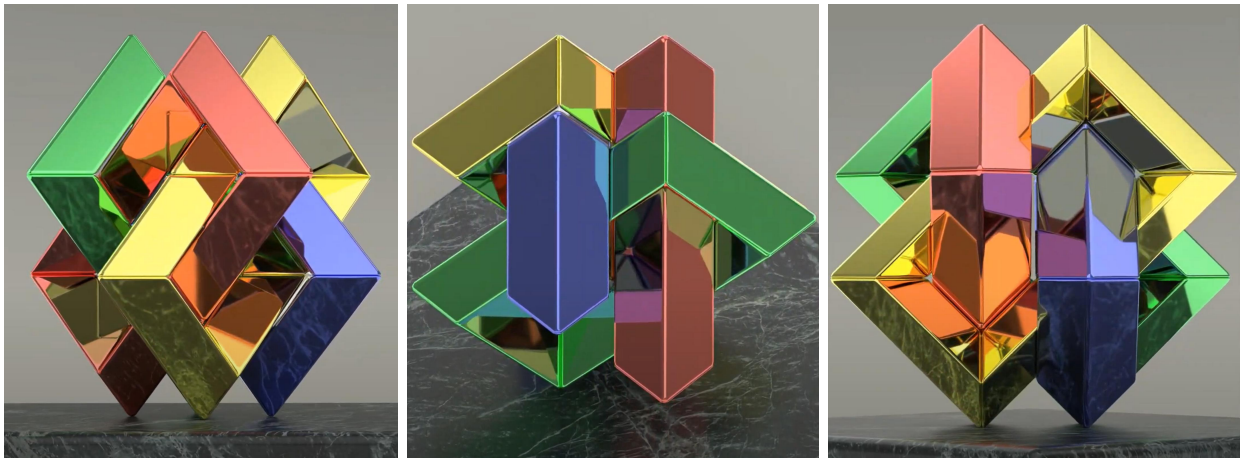

Figure 4: *A link with 4 rectangles in FCC.*

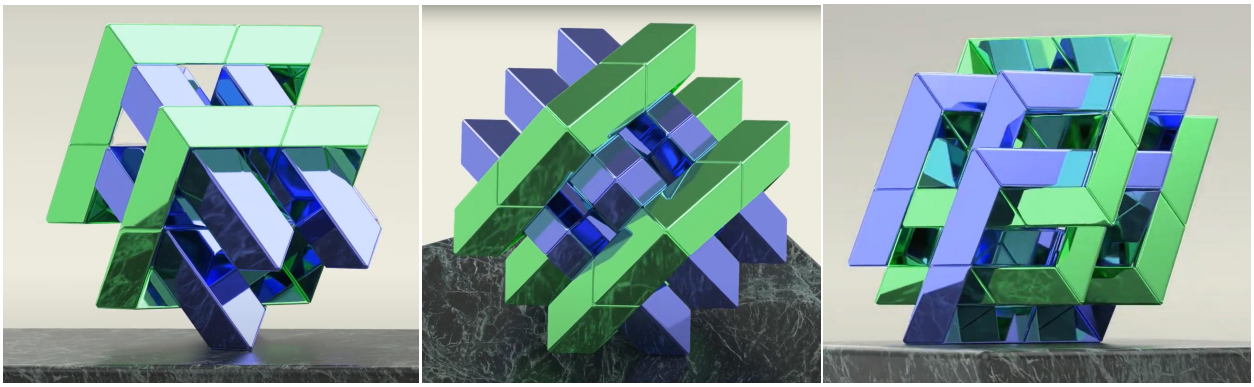

Figure 5: *A link consisting of 6 squares in FCC.*

Conclusion

We described a tool to help find interesting links involving some given polylinear paths in a lattice. This tool was implemented in Rhinoceros as a Grasshopper definition that serves as a post-processor for Anton's path generator described in [1]. This tool has been instrumental in finding many beautiful links. For more examples, see the supplementary material. The link-finding tool is still being refined. The most recent version also filters out symmetries, so that the same link will not be reported multiple times in different orientations. The ranking phase offers opportunities for further exploration of new criteria. Finally, there is room for performance improvements.

References

[1] A. Bakker and T. Verhoeff. "Domain-Specific Languages for Efficient Composition of Paths in 3D." *Bridges Conference Proceedings*, Halifax, Nova Scotia, Canada, July 27–31, 2023, pp. 259–266. http://archive.bridgesmathart.org/2023/bridges2023-259.html

[2] B. Ernst and R. Roelofs, Eds. *Koos Verhoeff – Jaarboek*. Ars et Mathesis, 2013. https://www.arsetmathesis.nl/jaarboek-2013-2

[3] T. Verhoeff. "Some Memories of Koos Verhoeff (1927–2018)." *Bridges Conference Proceedings*, Stockholm, Sweden, July 25–29, 2018, pp. 3–6. http://archive.bridgesmathart.org/2018/bridges2018-3.html

[4] T. Verhoeff and K. Verhoeff. "The Mathematics of Mitering and its Artful Application." *Bridges Conference Proceedings*, Leeuwarden, the Netherlands, July 24–29, 2008, pp. 225–234. http://archive.bridgesmathart.org/2008/bridges2008-225.html

[5] T. Verhoeff and K. Verhoeff. "Three Families of Mitered Borromean Ring Sculptures." *Bridges Conference Proceedings*. Coimbra, Portugal, Jul. 29–Aug. 2, 2011, pp. 73–80. http://archive.bridgesmathart.org/2011/bridges2011-73.html

[6] T. Verhoeff and K. Verhoeff. "Folded Strips of Rhombuses and a Plea for the $\sqrt{2}:1$ Rhombus." *Bridges Conference Proceedings*. Eschede, Netherlands, Jul. 27–31, 2013, pp. 71–78. http://archive.bridgesmathart.org/2011/bridges2013-71.html

[7] T. Verhoeff and K. Verhoeff. "Three Families of Mitered Borromean Ring Sculptures." *Bridges Conference Proceedings*. Baltimore, USA, Jul. 29–Aug. 2, 2015, pp. 53–60. http://archive.bridgesmathart.org/2014/bridges2015-53.html

[8] T. Verhoeff and K. Verhoeff. "Hopeless Love and Other Lattice Walks." *Bridges Conference Proceedings*. Waterloo, Ontario, Canada, July 27–31, 2017, pp. 197–204. http://archive.bridgesmathart.org/2017/bridges2017-197.html