Finding 3D Lattice Paths Exhibiting Knotted Optical Illusions

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

Human perception of three-dimensional shapes is hindered by the fact that we can only see one projection of such a shape at a time. When projections from different perspectives seem to be irreconcilable, we call this an optical illusion. The artistic use of optical illusions can teach the viewer the dangers of making judgments based on a single viewpoint. This lesson is valuable not only for judging spatial shapes, but for many aspects of life in general. We describe the techniques that we have developed to help find lattice paths in 3D space that exhibit knotted optical illusions, and illustrate this with a number of artworks.

Introduction

Both authors have worked together with Koos Verhoeff for many years and are still inspired by his creations. Consider his sculpture *Right-angled Braiding* in Figure 1. The side view is what one normally sees, and the various side views from different angles look quite consistent. Koos would then draw attention to the top view, and take delight in seeing the look of surprise on people's faces. Anton sees it as his mission to make

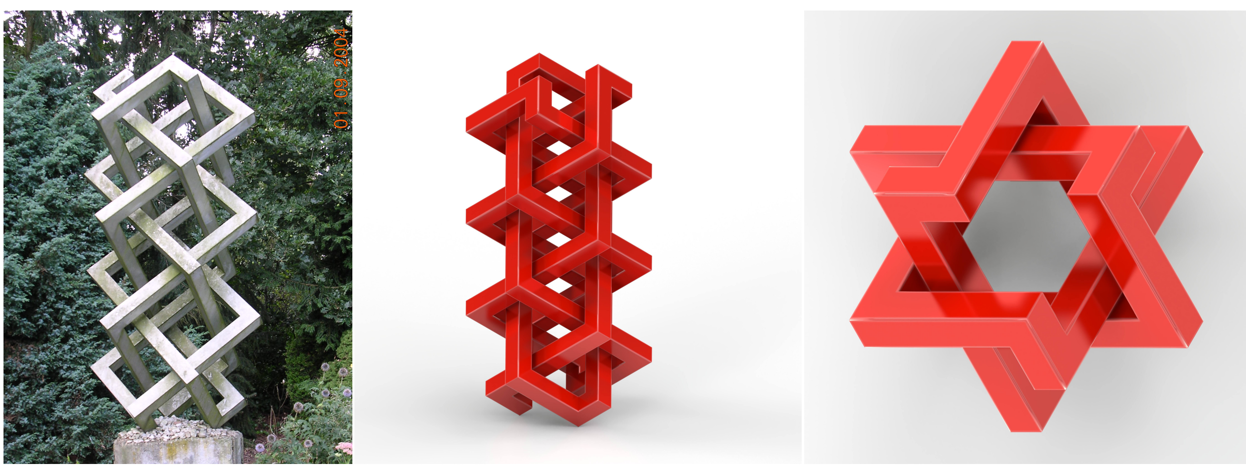


Figure 1: Right-angled Braiding*, design by Koos Verhoeff, side views and surprising top view (right).*

people more aware of the dangers of limited perspectives. It is all too tempting to believe that a single object should provide consistent projections when viewed from different angles. Objects with seemingly inconsistent projections are known as *optical illusions*. Real life is full of similar illusions, where multiple points of view are needed for a fuller understanding. Such art can serve as inoculation against the one-viewpoint trap.

In [1], we described *Anton's Path Language* to define families of closed 3D lattice paths. Through a *path expression* in the language, one controls structural properties of the paths generated from that expression. A single expression can generate thousands of paths, all having similar structural properties. Subsequently, we apply automated *beauty filters* (so named by Koos) to narrow down the search. Here, we focus on filtering for knots and for optical illusions. In our experience, knots lead to richer optical illusions.

Knot Filter

It is well-known [2] that algorithmic recognition of (un)knots is hard. The heuristic knot filter that Anton developed aims to be fast and avoid false negatives (false positives are unlikely because they are contrived).

The idea behind the knot filter is to tighten the closed path iteratively until it gets stuck. As an example, consider the trefoil knot in the Simple Cubic (SC) lattice in Figure 2, whose path consists of 12 vertices and edges. At each step three consecutive vertices, say P, Q, R, are considered (see Fig 2 (a, c)). If the filled triangle PQR does not intersect with other segments of the path (Figure 2 (a)), then vertex Q is eliminated from the path (Figure 2 (b)), so that the updated path goes straight from P to R, keeping the knot's topological structure invariant. If there is an intersection, then removing Q could affect the knot's structure (Figure 2 (c)). When no more vertex removals are possible, the iteration terminates. If the final path consists of just three vertices, then it's definitely unknotted; otherwise, it is flagged as a knot. Even if it is not a true knot, it still has features that could strengthen the optical illusion. In practice, this heuristic knot filter has never returned a false positive for the many lattice paths that were generated from expressions in Anton's Path Language.

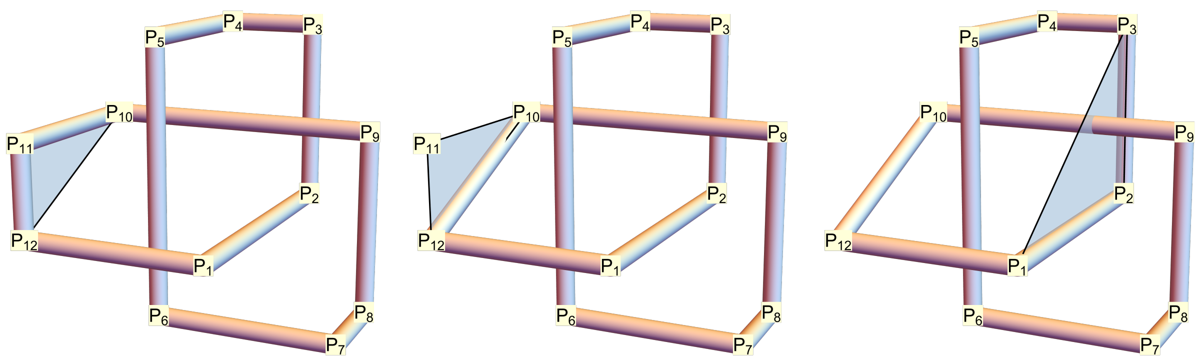


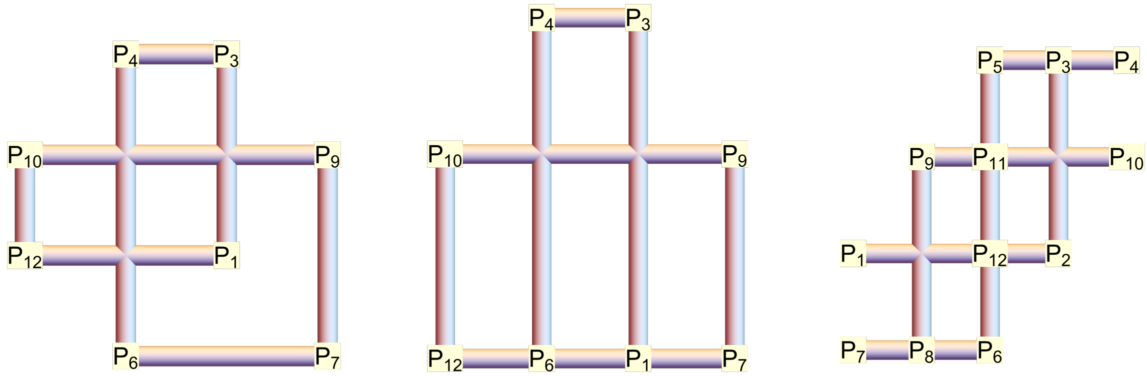
Figure 2: *Trefoil knot in SC: okay to remove P_{11} (a); P_{11} removed (b); not okay to remove P_2 (c).*

Optical Illusion Filter

Filtering for optical illusions is not as straightforward as filtering for knots. For one thing, there is no formal definition of an optical illusion. Anton's heuristic optical illusion filter aims to recognize specific features that are characteristic of certain optical illusions. Consider the sculpture *Eternal Links* in Figure 5 (a, b). In the leftmost view, it looks like two linked squares. That particular projection has the characteristic that there are exactly two intersection points when projecting the path's segments onto the viewing plane. The main goal of this filter is to help zoom in quickly on good candidates for optical illusions among thousands of paths.

In general, filtering for optical illusions currently is done in three stages, where each stage involves *projecting* the path in various directions, then *simplifying* the projected path (details are given below), and finally detecting certain features in the simplified projected path. These are the three stages:

1. Determine the number of self-intersections of the simplified projected path. This *crossing count* is determined automatically and can be filtered on (e.g., select only paths with crossing count 2 in some direction) or ranked by (e.g., lower crossing count ranks higher). Three projections of the SC trefoil knot of Figure 2 are shown in Figure 3. They have crossing count 3 in (a); 4 in (b), where P_1 and P_6 also count; and 6 in (c), where P_3 and P_8 also count. In our experience, more symmetry tends to strengthen



(a) (b) (c)

Figure 3: *Simplified projections of SC trefoil knot: 3 crossings (a); 4 crossings (b); 6 crossings, 4 ends (c).*

the effect of an optical illusion. This stage reduces the number of remaining candidate paths the most.

2. Determine (rotational and reflective) *symmetries* of the simplified projected path. In Figure 3, projections (a) and (b) have reflective but no rotational symmetry, whereas (c) has order-2 rotational but no reflective symmetry. This stage is still done manually by visual inspection.
3. Count *open ends* in the simplified projected path. An open end is where the projected path doubles back onto itself and loses another dimension, since it became one dimensional. Of the projections in Figure 3, (a) and (b) have no open ends, but (c) has four. Currently, this stage is also done visually.

Figure 4 shows some simplified projected paths for *Eternal Links* in Figure 5. It was selected because of its two crossings in the projection on the left.

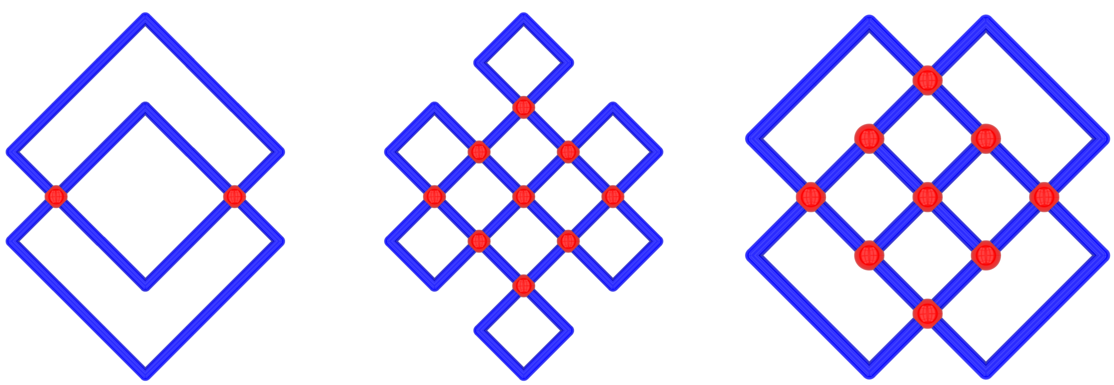


Figure 4: *Various simplified projections of* Eternal Links *in Figure 5, with crossings highlighted.*

Just like Anton's Path Language, all filters have been implemented in *Grasshopper*, which is a scripting environment integrated in the *Rhino* 3D-modeling software. Grasshopper has primitives for projecting curves onto planes, simplifying curves, and finding self-intersections for curves. Curve simplification removes redundant points, like P_2 in Figure 3 (a), since it coincides with P_1.

As projection directions, we use (i) lattice-related directions, in particular the lattice's edge directions (like X, Y, and Z in the SC lattice) and secondary directions (face and body diagonals), and (ii) path-related directions based on its *convex hull*, in particular, its face normals, and directions from its centroid to its vertices and to its edge midpoints.

Artwork

Figures 5 and 6 show some optical illusions that were discovered using the filters discussed above.

(a) (b) (c) (d)

Figure 5: *Optical illusions by Anton Bakker:* Eternal Links *a.k.a.* Two Squares*, patinated bronze height 25 cm (a, b);* Knot of Perception *a.k.a.* Ode to M.C. Escher*, patinated bronze height 25 cm (c, d).*

(a) (b) (c) (d)

Figure 6: *Optical illusions by Anton Bakker:* Hexagon Mirage*, mirror polished stainless steel, height 25 cm (a, b);* Deceptive Duo*, digital rendering (c, d).*

Conclusion

Programmatic, projection-based checks of 3D paths from specific lattice or convex hull viewpoints reveal visually intriguing phenomena. These include closed loops appearing open or varying self-intersection counts. Filtering on these features helps finding optical illusions, which mirror phenomena we encounter in daily life. Future work includes finding links with optical illusions and automating more of the manual steps.

References

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

[2] J. Hass, J.C. Lagarias, and N. Pippenger. "The computational complexity of knot and link problems." *Journal of the ACM*, vol. 46, no. 2, 1999, pp. 185–211. https://doi.org/10.1145/301970.301971.