Seifert Surfaces with Minimal Genus

Mereke van Garderen
Eindhoven University of Technology
m.v.garderen@student.tue.nl

Jarke J. van Wijk
Eindhoven University of Technology
vanwijk@win.tue.nl

Abstract

Seifert surfaces are orientable surfaces, bounded by a mathematical knot. These surfaces have an intriguing shape and can be used to produce fascinating images and sculptures. Van Wijk and Cohen have introduced a method to generate images of these surfaces, based on braids, but their approach often led to surfaces that were too complex, *i*.e., the genus of the surface was too high. Here we show how minimal genus Seifert surfaces can be produced, using an extension of standard braids and an algorithm to find such surfaces.

1 Introduction

Although generally undesirable in fishing lines and earphone cables, knots are a beloved subject of mathematical research. The simplest mathematical knot is the trefoil, shown in Figure 1. One puzzle related to knots is to find an orientable surface with a boundary that coincides with the knot. In 1934, the German mathematician Herbert Seifert described an algorithm to find such a surface for any knot [7], and these surfaces were named after him. A very clear introduction to Seifert surfaces can be found in *The Knot Book* of Adams [1]. Seifert surfaces are interesting for multiple reasons. They do not only result in fascinating sculptures and images, as made by Charles Perry, Robert Engman, and Robert Longhurst, they can also be used to define the *genus* of a knot, as an invariant of its topological structure. The genus of a Seifert surface is the topological genus of the closed, compact surface that results when a unit disk is glued along the knot. The genus of a knot is defined as the minimal genus of all possible Seifert surfaces of that knot. Our goal is to find and visualize Seifert surfaces that have this minimal genus.

Previous work Van Wijk and Cohen have presented a method to visualize Seifert surfaces [9, 10]. Their method is based on the use of braids, which describe the knot as a sequence of crossings on a set of lines. The endpoints of a braid are connected to those on the other side (see Fig. 1), but these connections are usually omitted from the drawing. For a large number of knots, braid representations with a minimal number of crossings are available, thanks to Gittings [4]. This information can be conveniently retrieved from KnotInfo [3], as well as a lot of other information, including the genus of the knot, evaluated by Rasmussen. Seifert surfaces can easily be derived from braids. Every level (horizontal row) in the braid gives a disk; every crossing in the braid corresponds to a twisted band connecting two disks. The collection of all disks and bands forms a Seifert surface. This simple structure can also be used to compute the genus g of the surface for a knot with one strand: $g = (b-d+1)/2$, where b is the number of bands and d is the number

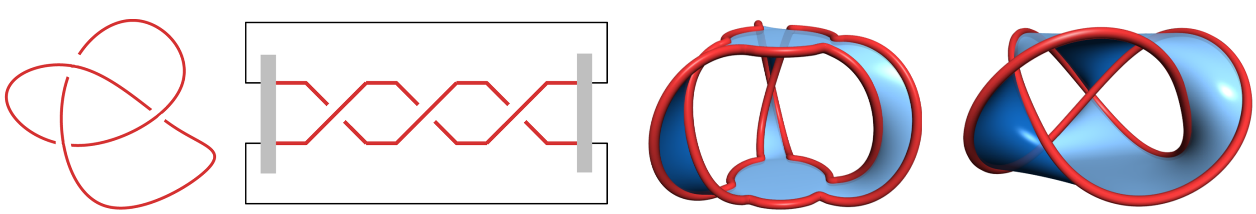

Figure 1 : *Knot diagram, braid diagram, Seifert surface and smoothed Seifert surface of the trefoil knot*

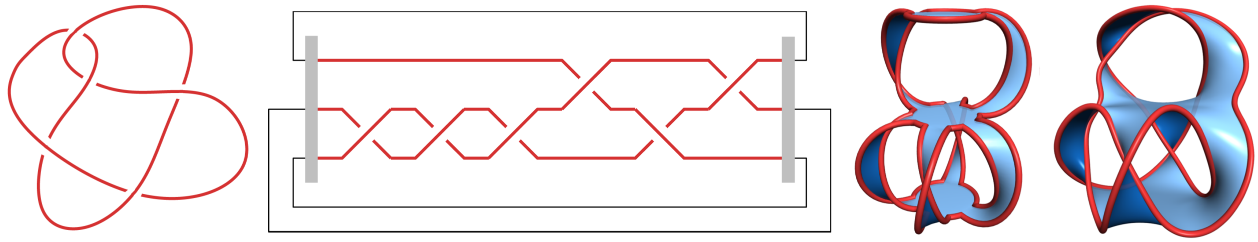

Figure 2 : *Knot diagram, braid diagram, Seifert surface and smoothed Seifert surface of knot 5_2*

of disks. Finally, the surface can be smoothed to generate a nicer visualization (see Fig. 1). The method is implemented in a tool called *SeifertView*, which is freely available online [8]. Inspired by the aesthetic appeal of Seifert surfaces, Bathsheba Grossman used this tool to produce wonderful sculptures [5].

Problem The classic braid notation is too limited to achieve braids with the minimal number of crossings and Seifert surfaces with the minimal genus for all knots. Some knots, such as the trefoil, are no problem, but difficulties arise with more complicated knots. Figure 2 shows knot 5_2 as an example. The minimal number of crossings for this knot is five, as can be seen from the diagram, but the braid notation gives six crossings. Also, the minimal genus of this knot is one, while the Seifert surface generated from its braid has genus two. To solve this problem, we need to find a braid representation that results in a Seifert surface with a lower genus. Because this is not always possible with classic braids, we use an extension: we allow crossings to skip over one or more lines. This results in longer bands in the Seifert surface, skipping over one or more disks. This extension has previously been described and used by Rudolph [6].

2 Solution

We produce minimal Seifert surfaces by first defining a set of moves to produce variations of braids, while keeping the knot constant, and applying these moves repeatedly to search for minimal genus Seifert surfaces.

Moves We defined five basic moves to manipulate a braid. The *cancel* move decreases the number of crossings in the braid and the genus of the resulting Seifert surface, the other moves are needed to facilitate this move. Figure 3 shows each move applied to a braid. If the orientation of a crossing is not indicated, this means both orientations are allowed for this crossing, as long as the orientation of such a crossing is the same before and after the move.

The *swap* move allows to change the order of two neighboring crossings on disjoint pairs of lines. These swaps are often necessary to enable other moves, as all moves are only defined on subsequent crossings. The *cancel* move can be applied when two neighboring crossings on the same pair of lines are in opposite orientations. In this case one line is consistently on top of the other, so both lines can simply be pulled straight, eliminating both crossings. We apply this move only in one direction (from two crossings to zero),

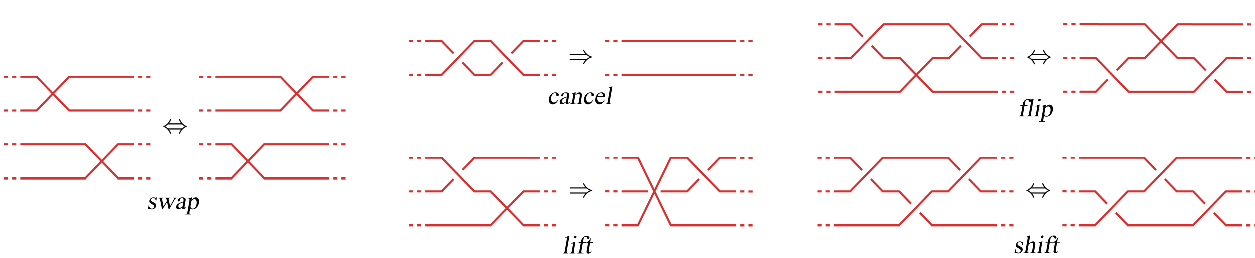

Figure 3 : *Five basic moves we applied to the braids*

to ensure termination of our algorithm, though it is an equivalence relation just like the other moves. The *shift* move is more complicated: one line is effectively shifted between the others to change the order of the crossings. These three moves are based directly on the theory of braid groups introduced by Artin in 1925, who showed that all possible equivalent classic braids can be produced with these three equivalence relations [2]. Interestingly, we needed a fourth move, the *flip* move, which is conceptually similar to the shift move. Both manually and experimentally we could not show that it could be deduced from the first three, possibly because we only allow the *cancel* move in one direction. In the *flip* move, the middle line is consistently on top of the others, and can therefore be flipped over to the other side. Finally, we defined the *lift* move, which introduces longer crossings. As the name suggests, this move lifts one crossing over the other, changing the order and making the lifted crossing longer. This does not affect the genus, but can be useful if *cancel* can be applied afterward. The *lift* move is also applied in one direction only. Since the original braid does not contain longer crossings, applying it in the other direction would just be undoing a previous move. All moves can also be applied if there are lines in between the upper and lower lines drawn here that are not involved in the crossings.

Algorithm The algorithm we designed is straightforward; all possible moves are tried out until either a braid of the correct genus (according to [3]) is found or all possible options have been explored. Variations of the knot are explored using breadth-first search. We first try all our moves on the initial knot, possibly multiple times at different locations, and we store each new braid we find. When we found all braids that can be generated by applying a single move to the initial knot, we try our moves on those variants to find all braids that are two moves away from the original, and so on. We stop if we find a braid with the minimal genus. Because none of the moves can be repeated infinitely often without producing braids already seen before, the process will terminate even if we do not find a braid of the desired genus.

3 Results

Figure 4 shows a step-by-step manipulation of knot 8_1. The original braid has genus three. After applying *lift*, *cancel*, *lift* and *cancel*, two long bands have been introduced and the genus of the surface is reduced to one, which is the minimal genus of this knot. Note that the algorithm produced many more variants during its execution, here we only show a trace that led to a Seifert surface with minimal genus.

Test set We applied our algorithm to a total of 798 knots with three up to eleven crossings. For the 249 knots with up to ten crossings, we found a Seifert surface with minimal genus for all except one. For knot 9_{35}, we could reduce the genus from five to two, but its minimal genus is one. We could solve 530 of the 549 knots with eleven crossings. For three of them, our moves were not sufficient to generate a braid with minimal genus. For the remaining 16, our program ran out of memory and we have not been able to fix this yet, due to time limitations. Running time ranged from seconds for easy cases to multiple hours for cases where exhaustive exploration of the search space was unsuccessful.

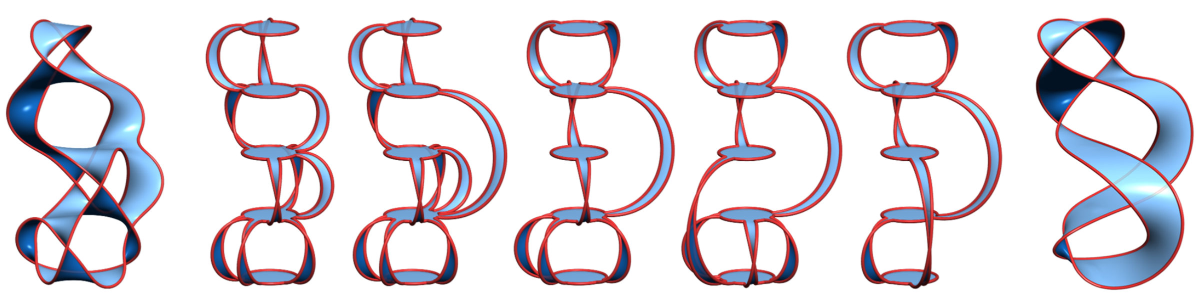

Figure 4 : *Step-by-step manipulation of knot* 8_1

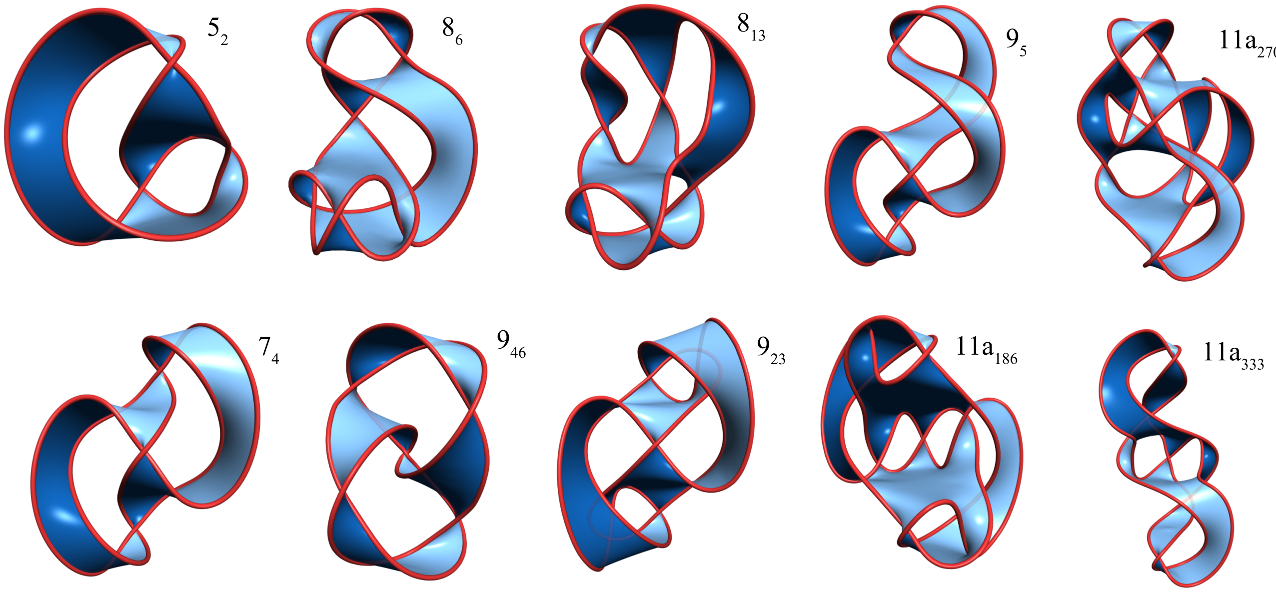

Figure 5 : *Examples of Seifert surfaces found by our algorithm*

Figure 5 shows examples of the resulting surfaces. There is a large diversity in the shapes of the Seifert surfaces, and although all are of minimal genus, they can still look quite complex. Judging hundreds of Seifert surfaces separately is somewhat tedious. We found symmetric configurations to be attractive, and implemented a scanning algorithm to detect these automatically. Some examples found in this way are 7_4, 9_{23}, 9_{46}, $11a_{186}$, and $11a_{333}$ (using knot names of KnotInfo [3]), which are shown in the bottom row of Figure 5.

Future work We aim to apply our approach to a larger sets of knots, with 12 crossings or more. Also, we want to understand why our approach fails in some rare cases and reconsider our set of moves. Currently, just one solution is produced per knot, we aim to generate all solutions per knot, and pick optimal ones, using different criteria. Furthermore, the braid representation can be extended further, for instance by allowing for multiple twists per crossing, such that more compact representations can be achieved.

References

[1] C.C. Adams. *The Knot Book - An Elementary Introduction to the Mathematical Theory of Knots*. American Mathematical Society (originally published in 1994 by W.H. Freeman and Company), 2004.

[2] E. Artin. Theorie der Zöpfe. *Abh. Math. Sem. Univ. Hamburg*, 4(1):47–72, 1925.

[3] J.C. Cha and C. Livingston. Knotinfo: Table of knot invariants. `http://www.indiana.edu/knotinfo`.

[4] T.A. Gittings. Minimum braids: a complete invariant of knots and links. *arXiv preprint math/0401051*, 2004.

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

[6] L. Rudolph. Braided surfaces and Seifert ribbons for closed braids. *Comm. Math. Helvetici*, 58(1):1–37, 1983.

[7] H. Seifert. Über das Geschlecht von Knoten. *Mathematische Annalen*, 110(1):571–592, 1935.

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

[9] J.J. van Wijk and A.M. Cohen. Visualization of the genus of knots. In *Proc. IEEE Vis.*, pages 567–574, 2005.

[10] J.J. van Wijk and A.M. Cohen. Visualization of Seifert surfaces. *IEEE TVCG*, 12(4):485–496, 2006.