Large, Symmetric, “7-Around” Hyperbolic Disks

Sean Jeng Liu, Young Kim, Raymond Shiau, Carlo H. Séquin
EECS, Computer Science, University of California, Berkeley

Abstract

Maximal symmetry is used to reduce the computational complexity in the construction of the largest possible hyperbolic disk composed entirely of equilateral triangles meeting seven per vertex.

Modeling Hyperbolic Disks

Mathematicians as well as artists are fascinated by the hyperbolic plane. Hilbert proved that the infinite hyperbolic plane cannot be smoothly and isometrically embedded (without self-intersections) in 3-space. Artists have made approximate models of a finite portion of the hyperbolic plane from various materials, for instance by crocheting (Fig.1a) or by gluing together paper triangles (Fig.1b). They may start with a small hyperbolic patch (warped like a potato chip) placed at the origin, and then add consecutive annular ribbons around the perimeter. With each annulus the distance from the origin grows only linearly, but the length of the annular ribbon grows exponentially; thus the outer border becomes ever more convoluted, and needs to curl through space in ever tighter undulations. Soon the physical properties of the modeling material will reach some practical limit of packing density, and a uniform radial extension of such a hyperbolic disk is no longer possible.

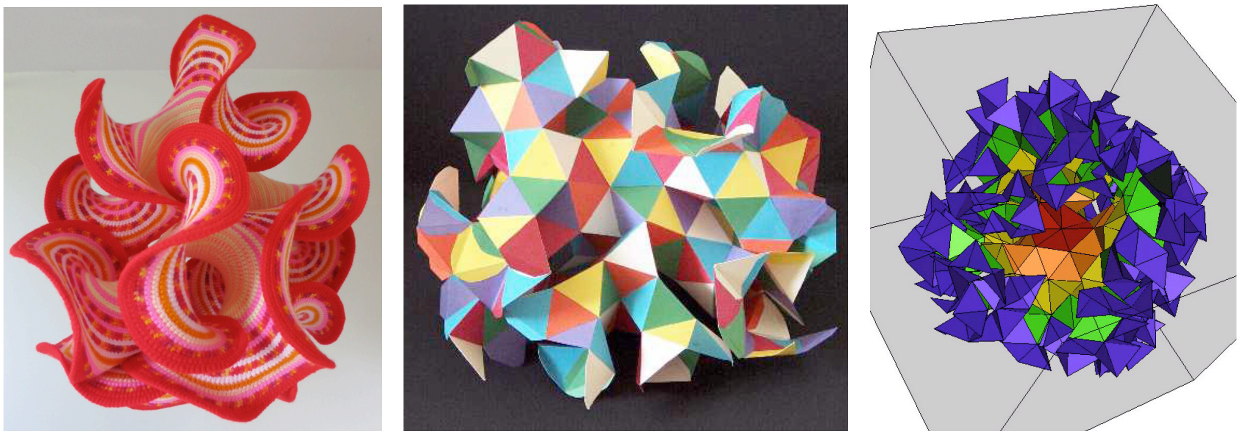

Figure 1: *Hyperbolic disks: (a) crochet model by G. Meyer [3], (b) paper model by D. A. Richter [4], (c) virtual CAD model by M. Howison [1].*

Alternatively, one may try to generate procedurally an approximate virtual model of a much larger hyperbolic disk. Because the geometrical surface is infinitely thin, layers of the curlicue undulations of the outer annuli might be stacked much more densely on top of one another. To make this an interesting and fair competition, some constraint must be imposed. In the current effort we are trying to construct a large hyperbolic disk from all equilateral triangles with unit edge-length – always packing exactly seven such triangles around each vertex. In order to prevent us from creating a trivial solution by running arbitrarily long strips of triangles off to infinity, we will always complete a full annular ribbon before proceeding radially outwards to construct the next annulus. The challenge then is to try to complete as many annuli as possible without encountering any self-intersections. We also disallow dihedral bends of 180° at the edge between two triangles – which would put them flush on top of one another.

In 2007 Mark Howison wrote such a program as a course project [1] in a graduate course on solid modeling (CS285). But that program quickly hit a barrier, where an annular ribbon is prevented from growing any further, since it always produces some self-intersections. The left shaded portion of Table 1 shows the exponential growth rate of the number of triangles in consecutive rings, when we first start with a single vertex surrounded by seven equilateral triangles. Howison's program maxed out at 810 triangles – it never was able to complete the 4th ring (blue); his best result, obtained after a run-time of several hours, is shown in Figure 1c. The program was quite sophisticated and employed a learning/training phase where some parameters were optimized to yield the most promising angular configurations between subsequent triangles added. Nevertheless, eventually the spatial crowding became so intense, that the program spent most of its time backtracking, trying to undo a bad configuration that preserves no space for new triangles and to establish a looser, more promising configuration. At 810 triangles, the search tree is very deep, and the number of possible branches to be explored becomes astronomical.

Table 1: *Growth of hyperbolic disks made from equilateral triangles, clustered 7-around each vertex.*

Ring	Faces (old)	Cumul.Faces	Color [1]	Ring	Faces (new)	Cumul.Faces
0	7	7	red	0	1	1
1	28	35	orange	1	15	16
2	77	112	yellow	2 = init.core	45	61
3	219	322	green	3 = swath #1	120	181
4	574	896	blue	4 = swath #2	315	496
5	1568	2464	purple	5 = swath #3	825	1321
6	4284	6748	white	6	2160	3481

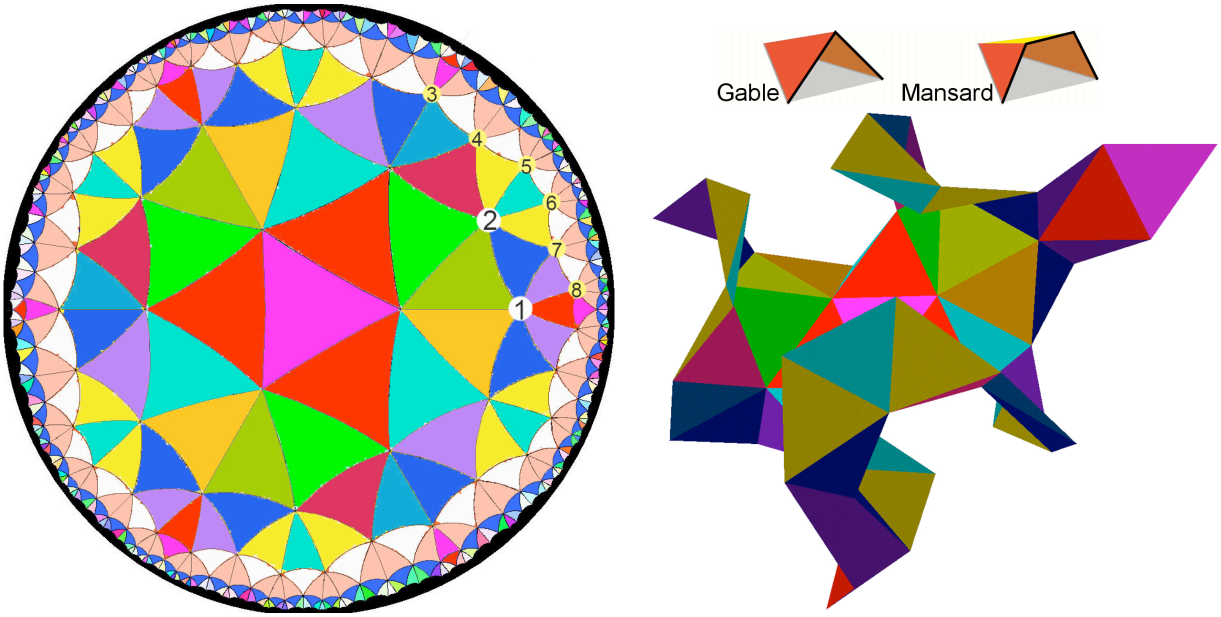

Figure 2: *The core of a symmetrical hyperbolic disk: (a) shown in the Poincaré disk, and (b) rendered perspectively in 3-space.*

Introducing Symmetry

Our novel idea is to construct a hyperbolic disk that is as symmetrical as possible. If the disk exhibits n-fold symmetry, then only $1/n^{th}$ of it has to be constructed, and the whole disk is obtained by instantiating the constructed segment *n* times. Maximal symmetry can be obtained by starting with a single triangle in the *xy*-plane, symmetrically straddling the origin. By enforcing the D_3-symmetry found in this triangle, we need to construct the sought-after hyperbolic surface in only one wedge of space comprising an angle of 60°. The central core of the disk is almost fully determined by the symmetry constraints: The three triangles surrounding the central triangle must also lie in the *xy*-plane. The two triangles (gold and olive) sharing vertex #1 (Fig.2a) each have an edge lying on a symmetry axis and must therefore be coplanar. The tilt against the *xy*-plane is determined by the condition that the two triangles (olive and green) sharing vertex #2 must form a symmetrical "gabled roof" above a right-angled wedge in the *xy*-plane; thus there are just two mirror solutions. Placing this "gabled roof" six times around the *z*-axis with D_3-symmetry completes ring #1 of this new configuration. The growth rate and the number of triangles in subsequent annuli for this new, symmetrical scheme are shown in the right shaded portion of Table 1.

We could now use this 16-triangle "core" as a starting base and let a random search algorithm similar to Howison's program calculate the next annulus sector, which would have as an outer border the vertex chain from #3 through #8. However, since there are still a substantial number of constraints in this ribbon domain, we decided to also complete this swath by hand. The two triangles (blue and teal) sharing vertex #3 also share an edge on a symmetry axis and thus must be coplanar; and there are again just two possible solutions for the position of vertex #4. We choose to give vertex #4 the same sign for its *z*-coordinate as for vertex #2; this produces a nice tall arch in the *z*-direction with ample room for the next annular ring to undulate around this looping border.

We make some further arbitrary decisions to complete this swath inside the vertex chain #3 through #8. First we choose the tilt around the C_2 symmetry axis for the (red/magenta) double triangle attached to vertex #1 so that it becomes coplanar with the (gold/olive) double triangle inside vertex #1. This particular choice leaves a 90° open angle in this shared tilted plane; this can again be spanned by a "gabled roof," which we choose to angle upwards. Finally we fill in three more triangles around vertex #2 (yellow/cyan/yellow) by letting them form an upward-pointing arch in the shape of a symmetrical 3-panel "Mansard roof." The result of these constructions, replicated six times around the origin, can be seen in Figure 2b. We always start from this fixed symmetrical "core" when we try to construct larger hyperbolic disks.

Triangle Placement Heuristics

Triangles are added to the disk individually or in pairs along the current open border forming a new annular ribbon. Individual triangles connect to that border with just a single edge, and thus there is a degree of freedom concerning the dihedral angle at the junction. We want to avoid forming acute dihedral angles, because they may make it difficult to place triangles in that region in the subsequent annulus; thus we choose a dihedral angle from a flat distribution (this seems to work better than a Gaussian distribution) in an interval that spreads $\delta = \pm 60$ degrees around a coplanar extension. Along the border, one vertex at a time, we complete the fan of seven triangles around every vertex. The last two triangles in each 7-element fan have to be placed as a pair, since they need to close up seamlessly with the first triangle in the fan; they form a "gabled roof" (Fig. 2b, top) in the remaining angular space around the current vertex, and there is the binary choice of angling the roof upwards or downwards.

For each triangle that we place, we must check that it does not intersect any other triangles. But just checking for current intersections is too shortsighted; near-misses will definitely cause problems in the next outer annulus. Thus we do a more conservative proximity check, measuring the overlap of the circumsphere of the new triangle against the circumspheres of all other triangles in the neighborhood. If the distance of the centers of these spheres falls below a certain threshold, we try a series of different dihedral angles, and if they all fail, we backtrack and reposition the previous triangle. More details about the various heuristics employed can be found in the Appendix of the electronic, extended and updated version of the paper [2].

Results

With the above heuristics in place, we now can let the program run many times to complete successive swathes in a 60° sector around the core's border. We visually inspect the results for a particular swath, and among several possible solutions pick the one that we think will form the most beneficial border for the subsequent swath. Such a good solution can then be "frozen" and added to an extended starting core. With this approach we have constructed good solutions for swaths #1, #2 and #3. On June 9, we were able, for the first time, to successfully complete swath #3 for a total of 1321 triangles. These results can be seen in the electronic version of the proceedings, including a brief dynamic file "clip-1321-compressed.mov" that conveys a better 3D understanding of this geometry.

By July 6, working from both ends of swath #4, we have been able to place up to 147 triangles into the subsequent annulus. This result is shown in Figure 3a using the following color coding: the central starting triangle is white, the manually constructed core of 60 triangles around it is dark red, swath #1 is light green, swath #2 is dark blue, swath #3 is magenta-colored, and the triangles in the outermost ring are gold. After instantiating the constructed sector six times, we then have a total of 2197 triangles in our hyperbolic disk. Figure 3b shows the full disk geometry with the same coloring for the triangles.

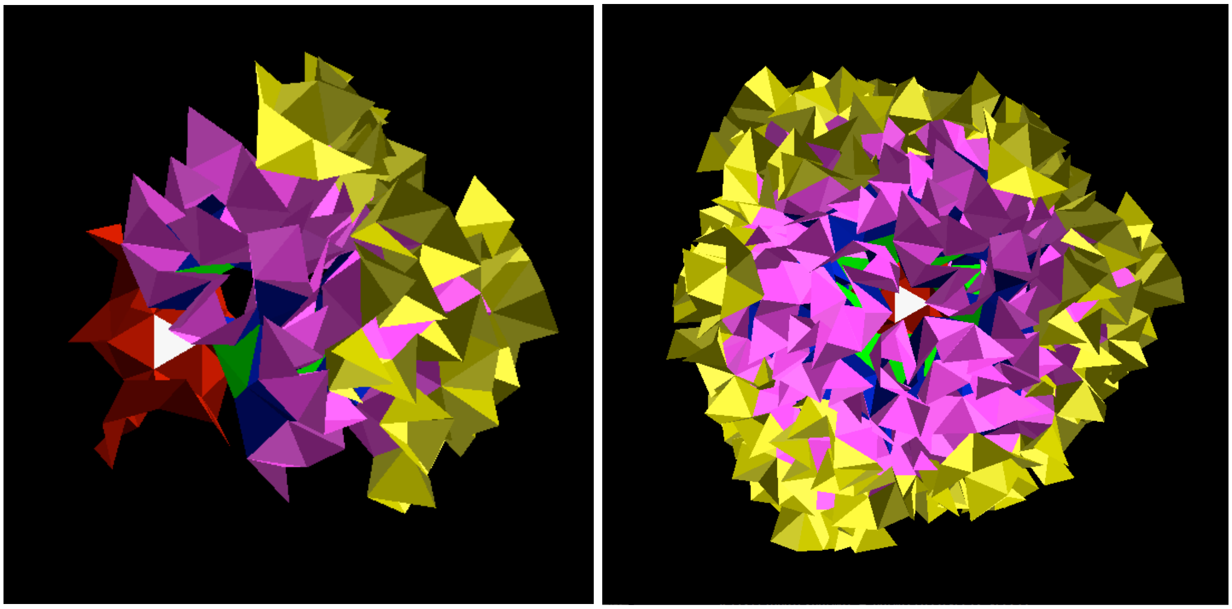

Figure 3: Best result to-date: (a) core_61 plus one 60° sector, (b) the assembled disk.

References

[1] M. Howison, *Constructing Models of the Seven-Around Surface*. Final Report for CS 285: Solid Modeling -- http://www.cs.berkeley.edu/~sequin/CS285/2007_REPORTS/Howison_7-around.pdf (April 2015).

[2] S. J. Liu, Y. Kim, R. Shiau, C. H. Séquin, *Large, Symmetric, "7-Around" Hyperbolic Disks*. Extended and updated version of this paper. -- http://www.cs.berkeley.edu/~sequin/PAPERS/2015_Bridges_7-around-ext.pdf

[3] G. Meyer, *Hyperbolic Disk with Red Rim*. -- http://gallery.bridgesmathart.org/exhibitions/2014-joint-mathematics-meetings/gabriele_meyer (April 2015).

[4] D. Richter, *A paper model of the seven-around surface.* -- http://homepages.wmich.edu/~drichter/hyperbolicimbed.htm (April 2015).

[5] T. Möller, *A Fast Triangle-Triangle Intersection Test*. Journal of Graphics Tools, A.K.Peters, Vol.2, Issue 2, (1997), pp 25-30.

Appendix for the Electronic Version of this Paper

Here we give some more details about the inner workings of our triangle-placement program and the current settings for the parameters in the various heuristic subroutines. Triangles are first placed with a randomly selected dihedral angle selected from a flat distribution in an interval that spreads $\delta = \pm 60$ degrees around a coplanar extension.

Placement Heuristics

At this stage we already introduce some heuristics to obtain a reasonably regular undulation of the seven triangles forming a fan around a vertex. We sum up the dihedral angles along the edges radiating out from the shared vertex for all the triangles added to the fan so far. For triangles #4 and #5 in each 7-member fan we check the sign of the current sum and then give the new triangle a dihedral angle of opposite sign against the last placed triangle.

The last two triangles in a fan are placed as a pair forming a "gabled roof." To make sure that this roof is feasible and not too narrow, we want the base angle that it spans to lie in the range between 60° and 120°. Thus when we place triangle #5 in a fan, we check that the remaining angular gap lies in this interval. If it does not, we try to fix the problem by making ten more random tries for placing #5. If this is not successful, the program backtracks (see below).

Once a new triangle has been placed by employing the above heuristics, it needs to be checked for any intersections with any previously placed triangles. But first a more conservative tests is performed to see whether it encroaches too closely to any nearby triangles, which may cause difficulties for the placement of any future adjacent triangles. For this purpose, the distances of the center of the new triangle from the centers of all the triangles in the neighborhood are calculated, and the triangles closer than two circum-radii are placed in a queue sorted by closeness. If the distance from the closest triangle (if it is not directly connected to the new one) is less than $dn = 1/\sqrt{3}$ (this corresponds to one circum-radius), the program tries to turn the new triangle away from this obstacle. For triangles that are directly connected to the new one, we allow a closer separation between the triangle centers of $dc = 0.408$ (this corresponds to a separation of the triangle centers for a dihedral angle of 90° between them). In either case we try to turn the triangle further away from the obstacle by adding steps of 20° to its dihedral angle until the center-to-center distance is larger than the specified threshold or the dihedral angle exceeds ±90°. If these attempts to obtain enough clearance from nearby obstacles are not successful, the program backtracks.

If the latest triangle is part of a gabled roof (and there is an encroachment problem), the whole roof is flipped to the opposite side of its base gap; if this does not yield an acceptable solution, the program backtracks.

If the new triangle is finally positioned to contribute to a nice border curve that has some potential for accommodating another swath of triangles, we still need to do an exact triangle-triangle intersection test [5]. Two triangles could just intersect with their corners while their centers are separated by just a little bit less than two circum-radii. We only have to apply this more expensive test to triangle pairs that had a center separation of less than two circum-radii.

Backtracking

If the triangle we want to backtrack is not part of a roof, then we simply invalidate it by readjusting pointers. If it is part of a roof, then we invalidate the entire roof plus triangle #5 in the fan. The number of times that a given triangle has been backtracked is stored as an attribute in the previous triangle. Non-roof triangles can backtrack up to $bn = 10$ times before the previous triangle gets backtracked. To speed up the program, the fifth triangle in a fan will only get backtracked $b5 = 3$ times before the previous triangle (#4) gets backtracked.

Visible Inspection and Interactive Guidance

The program is running with a tight coupling to an interactive graphical display. Any new triangles added as well as any backtracking steps are shown in real time. The program can be stopped at any time or stepped through individual triangle additions. This allowed us to develop an intuitive feeling for judging which heuristics are beneficial and what parameter setting may work best.

Recently we have introduced an editing feature where a specific triangle that seems to hold up progress because of poor placement can be turned manually to a more promising dihedral angle. And the gabled roof that completes any swath can be flipped manually to its mirror position.

This approach allows us to carefully develop a good hyperbolic disk one swath at a time. Starting from the core, we run the program many times (this only takes a few seconds) and visually select a swath #1 that seems to have the best potential for allowing the addition of a complete swath #2. This attractive swath #1 can then be frozen and added to the core; and for swath #2, this extended core now serves as the starting point. When we have found a good swath #2 (this may take a few minutes), it also gets added to the core, and we can tackle swath #3 from a good starting point.

On swaths #1 and #2, we apply the above heuristics in order to obtain a gracefully undulating border line, open enough to give us a reasonable chance of constructing the next outer swath. On swath #3 and #4, however, we no longer are so conservative; we accept any triangle that does not cause an outright intersection.

The symmetry constraints of the whole disk add some constraints for the two ends of each swath constructed in a 60° sector. This also gives us the opportunity of making some heuristic choices that reduce computational overhead and/or may lead to a better border curve. For instance, for swath #1, when placing the magenta/red triangle pair outside of vertex 3, we chose to make them coplanar to the central red triangle, i.e., place them also into the x-y-plane.

Status and Best Results at this Time

With the above approach, at the time of this submission we could complete the first two swaths within run times of a few minutes and some interactive adjustments of some crucial triangles. By June 9, we had completed swath #3 for the first time. By the time of the Bridges July 2015 conference, we have been able to place 147 triangles into swath #4. This gave us a grand total of 2197 triangles.

Eventually we also plan to make a physical model of this 7-around hyperbolic disk by using layered manufacturing; Figure 4 shows a suitably thickened model of the core plus a completed swath #1.

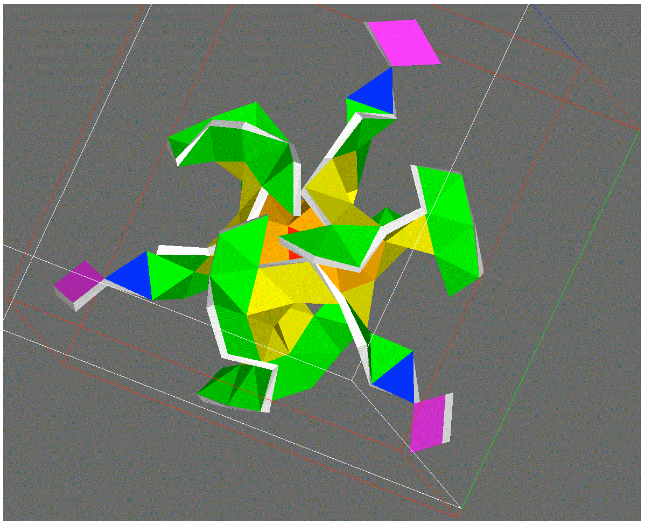

Figure 4: *Model of the core plus swath #1 made suitable for 3D-printing.*

Updates over Summer 2015

Over the summer, we made some small modifications to our heuristics in order to achieve this new record. In general, we rejected unpromising triangles in swath #1 earlier and allowed more flexibility in placing triangles in outer swaths.

For instance, we only applied the 60°-minimum rule to base angles of "gabled roofs" in swath #1. For outer swaths, we gave the program more freedom with the base angles due to space constraints.

When a triangle fails the conservative intersection test, the program only gives five attempts of adjusting the triangle in steps of 20°. If the triangle still falls below the *dn* or *dc* threshold and is in swath #1, then the program backtracks. Otherwise, if the triangle is not in swath #1, we apply the more expensive test to decide whether or not to backtrack.

Similarly, if a "gabled roof" in swath #2 or greater fails the conservative test, we apply the expensive test to see if there was actual intersection. If not, the program leaves it as it is instead of backtracking.

Finally, for swath #3, we disabled the ±90° constraint for manual tweaking and the immediate backtracking that occurred when an intersection was detected for the randomly placed triangle. This is because at such a tight space, randomly placed triangles almost always had intersections, so the backtracking feature was not as effective as manual tweaking on some of these triangles in outer swaths.