Geo-wrapping

Vivian Auduong

December 2016

1 Abstract

Inspired by the iconic ”Spider and Fly” problem in mathematics, this paper attempts to answer the question, ”What happens if the spider kept crawling?” The question can be visualized as string wrapping around 3-dimensional objects, and was therefore given the name ”Geo-wrapping”. Properties of line-continuation on a net were observed, a python turtle program was written to model the problem for a cube, and a generalization of the problem for different solids was attempted.

2 Introduction

2.1 Polyhedra and nets

Polyhedra (meaning ”many faces”) are defined as solids composed of flat **faces** called polygons, the **vertices** of such polygons, and the **edges** where the sides of the polygons intersect. (2)

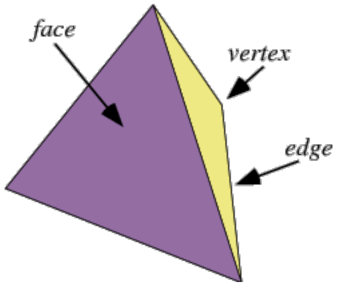


Figure 1: A diagram of a tetrahedron’s 4 triangular faces, 4 vertices, and 6 edges. (4)

A **net** of a polyhedron is defined as a 2-dimensional representation which contains all the faces of a polyhedron, some of them separated by angular gaps.

(6) Note that for any given polyhedron, there may be multiple accurate orientations of nets for that polyhedron. In addition, for any given polyhedron net, there may be multiple ways of joining that net to yield distinct polyhedra. (1)

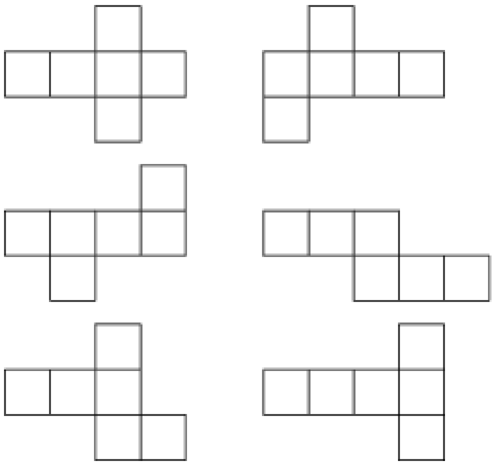


Figure 2: The 11 different possible cube nets (1)

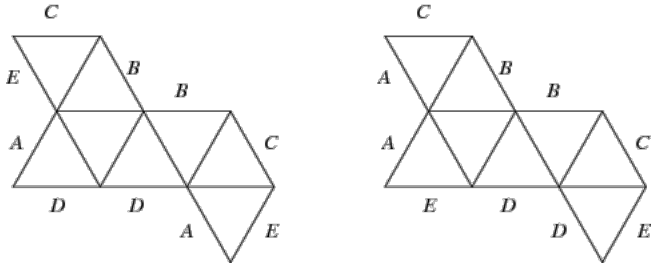


Figure 3: Two different foldings of an identical octahedron net (1)

2.2 Spider and Fly Problem

The iconic Spider and Fly Problem was first introduced in an English newspaper by puzzle-creator Henry Ernest Dudeney in 1903. The problem involves a rectangular room, or rectangular prism, and two points on the prism designated the spider and fly. The solution seeks to find the shortest distance the spider can travel on the rectangular prism to reach the fly.(3)

While an initial guess may involve finding the shortest distance between edges, the solution can be found by flattening the rectangular prism into a net. Because the shortest distance between any two points is a straight line, the shortest distance between two points on a polyhedron is a straight line segment on the net.

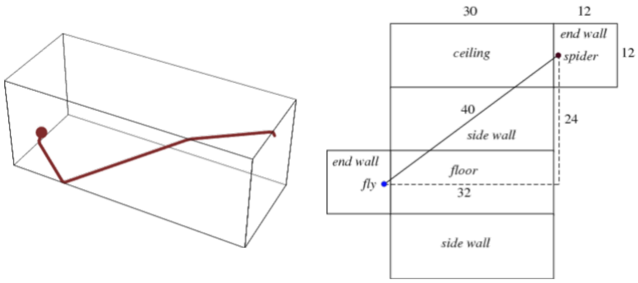


Figure 4: A diagram demonstrating the "flattening" of a net in determining the shortest distance between two points in a room of dimensions 12'x12'x30'. The spider is located in the middle of one 12'12' wall one foot away from the ceiling. The fly is in the middle of the opposite wall one foot away from the floor. (3)

Such a solution of connecting a line segment between two points on a cube provoked the questions: What if the spider continued on the line? "What are the properties of lines drawn on polyhedra? In addition to being a contribution to mathematical research, the properties could potentially be useful in determining string used for wrapping objects or in the structural integrity of buildings.

3 Methods

Consider the idea of a spider continuing on a line as a single line defined by a starting point and angle. When the line reaches an edge of the net, it jumps to the corresponding edge, rotates to preserve angle, and continues until reaching another edge. When the next edge is reached, there is a new starting point and angle, and the process repeats. Therefore this process can be written as a **recursive function**, a function which calls itself.

3.1 Flattening of the Polyhedron Net and Recognizing Corresponding Edges

Just as different nets are possible, so can the faces of a net be moved to visualize how a line on a net changes with identical polyhedra, but different net orientations. In figure 5 , it can be seen that the red square is identical in all images and B'C has the same orientation relative to the red square. In addition, every point on the edge containing B corresponds to a point on the edge containing B', regardless of different orientations of the net. Therefore, edges in which points correspond to each other in a one-to-one manner are defined as **corresponding edges**. Each and every edge has a coresonding edge, regardless of whether or not the edge is separated (requires jumping off the net) or continuous (can be reached with a straight line on the surface of the net).

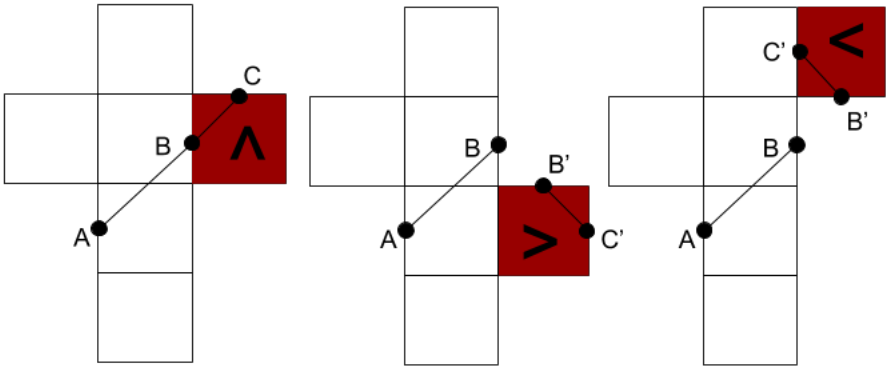
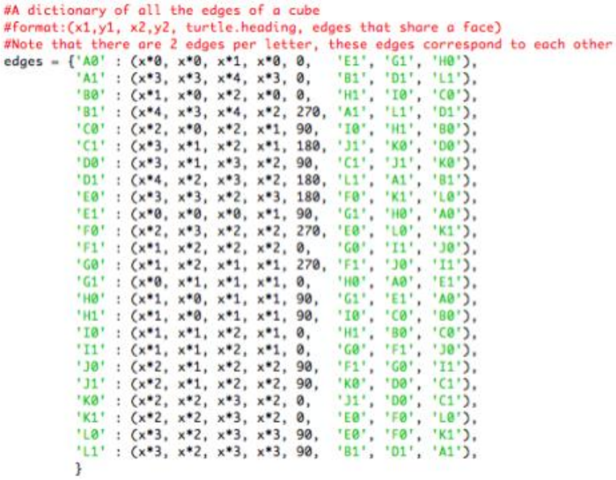


Figure 5: A diagram showing three different possibilities for the orientation of a red square. Notice that despite the nets being drawn differently, the resultant cubes are identical



```
#A dictionary of all the edges of a cube
#format:(x1,y1, x2,y2, turtle.heading, edges that share a face)
#Note that there are 2 edges per letter, these edges correspond to each other
edges = {'A0' : (x*0, x*0, x*1, x*0, 0,   'E1', 'G1', 'H0'),
         'A1' : (x*3, x*3, x*4, x*3, 0,   'B1', 'D1', 'L1'),
         'B0' : (x*1, x*0, x*2, x*0, 0,   'H1', 'I0', 'C0'),
         'B1' : (x*4, x*3, x*4, x*2, 270, 'A1', 'L1', 'D1'),
         'C0' : (x*2, x*0, x*2, x*1, 90,  'I0', 'H1', 'B0'),
         'C1' : (x*3, x*1, x*2, x*1, 180, 'J1', 'K0', 'D0'),
         'D0' : (x*3, x*1, x*3, x*2, 90,  'C1', 'J1', 'K0'),
         'D1' : (x*4, x*2, x*3, x*2, 180, 'L1', 'A1', 'B1'),
         'E0' : (x*3, x*3, x*2, x*3, 180, 'F0', 'K1', 'L0'),
         'E1' : (x*0, x*0, x*0, x*1, 90,  'G1', 'H0', 'A0'),
         'F0' : (x*2, x*3, x*2, x*2, 270, 'E0', 'L0', 'K1'),
         'F1' : (x*1, x*2, x*2, x*2, 0,   'G0', 'I1', 'J0'),
         'G0' : (x*1, x*2, x*1, x*1, 270, 'F1', 'J0', 'I1'),
         'G1' : (x*0, x*1, x*1, x*1, 0,   'H0', 'A0', 'E1'),
         'H0' : (x*1, x*0, x*1, x*1, 90,  'G1', 'E1', 'A0'),
         'H1' : (x*1, x*0, x*1, x*1, 90,  'I0', 'C0', 'B0'),
         'I0' : (x*1, x*1, x*2, x*1, 0,   'H1', 'B0', 'C0'),
         'I1' : (x*1, x*1, x*2, x*1, 0,   'G0', 'F1', 'J0'),
         'J0' : (x*2, x*1, x*2, x*2, 90,  'F1', 'G0', 'I1'),
         'J1' : (x*2, x*1, x*2, x*2, 90,  'K0', 'D0', 'C1'),
         'K0' : (x*2, x*2, x*3, x*2, 0,   'J1', 'D0', 'C1'),
         'K1' : (x*2, x*2, x*3, x*2, 0,   'E0', 'F0', 'L0'),
         'L0' : (x*3, x*2, x*3, x*3, 90,  'E0', 'F0', 'K1'),
         'L1' : (x*3, x*2, x*3, x*3, 90,  'B1', 'D1', 'A1'),
         }
```

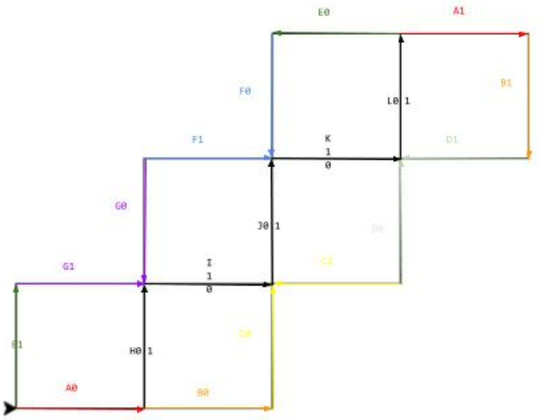


Figure 6: An example of a dictionary for a cube and a diagram of the resultant net.

Because of the properties of nets having multiple accurate orientations and also multiple ways of being joined to form a polyhedron, it can be concluded that modeling such a problem requires every edge endpoint, relation to other edges, and shared faces to be defined. This is a powerful conclusion, as it means that any polyhedra can be defined as a specific set of faces, vertices, and edges, and the process described to geo-wrap applies to any polyhedron. In python, this led to the decision to create a dictionary of edges as seen in figure 6, with each edge defined by the coordinates of its endpoints, heading, and edges with which it shares a face.

3.2 Coordinate Geometry and Establishing the Next Edge on a Shared Face

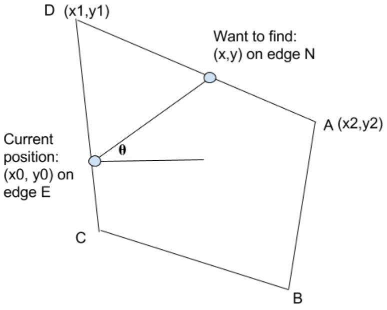


Figure 7: A diagram showing a possible face ABCD, point (x_0,y_0), angle θ, and proposed (x,y) on some edge N

Consider a face ABCD, where it is known that on some edge E on this face, a point (x0,y0) has continued from another edge, and the current angle of the line is θ. Constructing a line from (x_0,y_0) with angle θ, we wish to find the edge N that is intersected by this line and the intersection point on the edge.

Then, the line constructed from (x_0,y_0) and angle θ has the equation:

$$y - y_0 = tan(\theta) * (x - x_0)$$

$$y = tan(\theta) * x - tan(\theta) * x_0 + y_0$$

Without loss of generality, the line constructed by any edge with endpoints (x1,y1) and (x2,y2) has the equation:

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1} * (x - x_1)$$

Let

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

Then,

$$y = m * (x - x_1) + y_1$$

Setting equations equal to each other and solving for x, x=x1=x2 when the edge N is vertical. Otherwise:

$$x = \frac{tan(\theta) * x_0 - x_1 * m + y_1 - y_0}{tan(\theta) - m}$$

This result can be plugged back into the original equation to find y.

$$y = tan(\theta) * x - tan(\theta) * x_0 + y_0$$

The calculated value of (x,y) signifying edge N as the next edge is correct if and only if (x,y) lies on the segment defined by endpoints (x_1,y_1) and (x_2,y_2) of edge N. Otherwise, this signifies that another edge on the shared face ABCD is the correct edge, and the process should be repeated on a different edge to find the correct next edge.

3.3 Heading

Heading is a term used in navigation that refers to the direction a vehicle is pointing.

In Geo-wrapping, the **heading** of an edge is defined as the direction, or angle, of the edge. Every edge has a corresponding edge, and in order to move to that next edge while preserving the relationship between the line and the edge, it is necessary to change the angle of the line based on the headings of the current edge and corresponding edge.

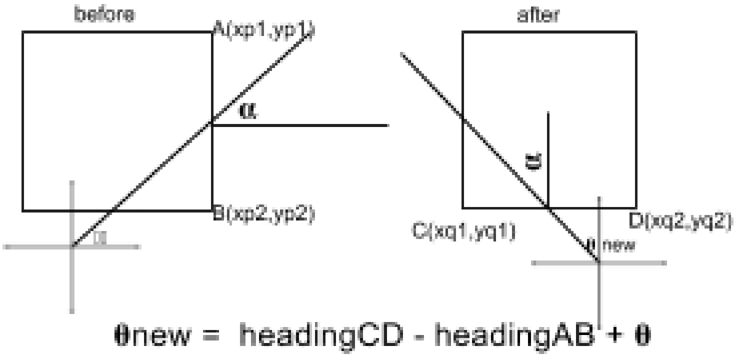


Figure 8: A diagram showing how corresponding edges and angles relate

If an angle exits AB at an angle α relative to AB, it enters CD at an angle α relative to CD.

As seen in figure 8 Consider edge AB with coordinates (x_{p1},y_{p1}) and (x_{p2},y_{p2}) and heading $heading_{AB}$, meaning (x_{p2},y_{p2}) is in direction $heading_{AB}$ from

(x_{p1},y_{p1}). Also consider edge CD with coordinates (x_{q1},y_{q1}) and (x_{q2},y_{q2}) and heading $heading_{CD}$. Let edges AB and CD be defined in a polyhedron as corresponding edges, with each point on AB corresponding to a point on CD.

As a result, the difference in the new angle and old angle is given by:

$$\Delta angle = heading_{CD} - heading_{AB}$$

and if the angle relative to the coordinate plane is θ,

$$\theta new = heading_{CD} - heading_{AB} + \theta$$

3.4 Ambiguous Case: Intersecting a Vertex

Although the process for jumping to a corresponding edge, preserving angle, and continuing until the next edge has been established, there is one ambiguous case to be wary of. When the line intersects the vertex of an edge, the next edge and heading are ambiguous due to the line being able to extend into more than one face.

4 Conclusion

This exploration of drawing straight lines on the surface of a polyhedron yielded useful rules. In particular, every polyhedron can be defined as a relationship of faces, vertices, and edges; the change in heading when jumping to a corresponding edge is equal to the difference between the headings of those corresponding edges, and subsequent edges can be calculated using coordinate geometry. These rules were found as a result of creating a python program to model different polyhedra, starting lengths, and angles, which are included in the appendix. Future exploration can be done through defining more complex polyhedra edge dictionaries, as described in the appendix. In addition, it would be a good exercise to consider other functions, perhaps conics, sinusoidal curves, or exponential functions on the surfaces of polyhedra and other geometric solids.

References

[1] Weisstein, Eric W. "Net." From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/Net.html

[2] Rhoad, Richard, George Milauskas, and Robert Whipple. "Geometry for Enjoyment and Challenge." Evanston, IL: McDougal, Littell, 1991. Print.

[3] Weisstein, Eric W. "Spider and Fly Problem." From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/SpiderandFlyProblem.html

[4] Weisstein, Eric W. "Polyhedron Vertex." From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/PolyhedronVertex.html

[5] Hyde, Dan. ”Notes on Using Python’s Turtle Built-in Commands.” Department of Computer Science. Bucknell University, 26 July 2011. Web. 25 Oct. 2016.

[6] Woodward, Ernest, and Rebecca Brown. ”POLYDRONS AND THREE-DIMENSIONAL GEOMETRY.” The Arithmetic Teacher 41.8 (1994): 451-58. Web.