Stability of 3D Quasicrystals: Elementary

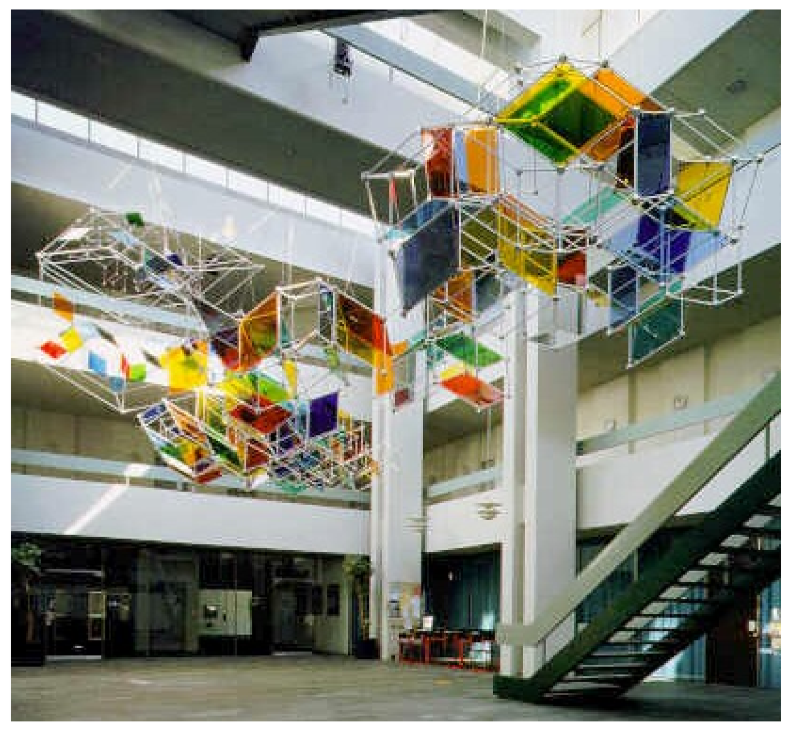
Manting Huang, Pranav Bhardwaj, Sung Jib Kim, Tejo Nutalapati, Yi "Lisa" Li, Eliana Duarte, Nima Rasekh, George Francis

Introduction

Motivation

The project is part of a greater effort to understand the stability of 3d quasicrystals. The goal is to figure out which parts of a given quasicrystal we have to make rigid such that the whole shape cannot be deformed. This was motivated by Tony Robbin, who needed this theory for his artwork (COAST) and in his case had to find the answer using experimentations.

COAST



In this 3D quasicrystal, Tony Robbin chose several rhombohedra and made them rigid by plating some of their faces. This had the effect of making the whole structure rigid. Rigidity means that the only possible motions of the structure are rotations or translations. It remains an open problem in mathematics to give criteria to decide when a 3D rod and pinion framework is rigid.

Figure 1: *Tony Robbin, COAST, Installed 1994 at Danish University, Destroyed!!!*

The 2-Dimensional Case: Wester's Theorem

Theorem 1: (Wester's Theorem, [2])
Let K be a rhombic carpet with associated Wester graph Γ and let Φ be a subgraph which is both spanning and connected. Then, bracing the rhombi corresponding to the edges of Φ makes K rigid.

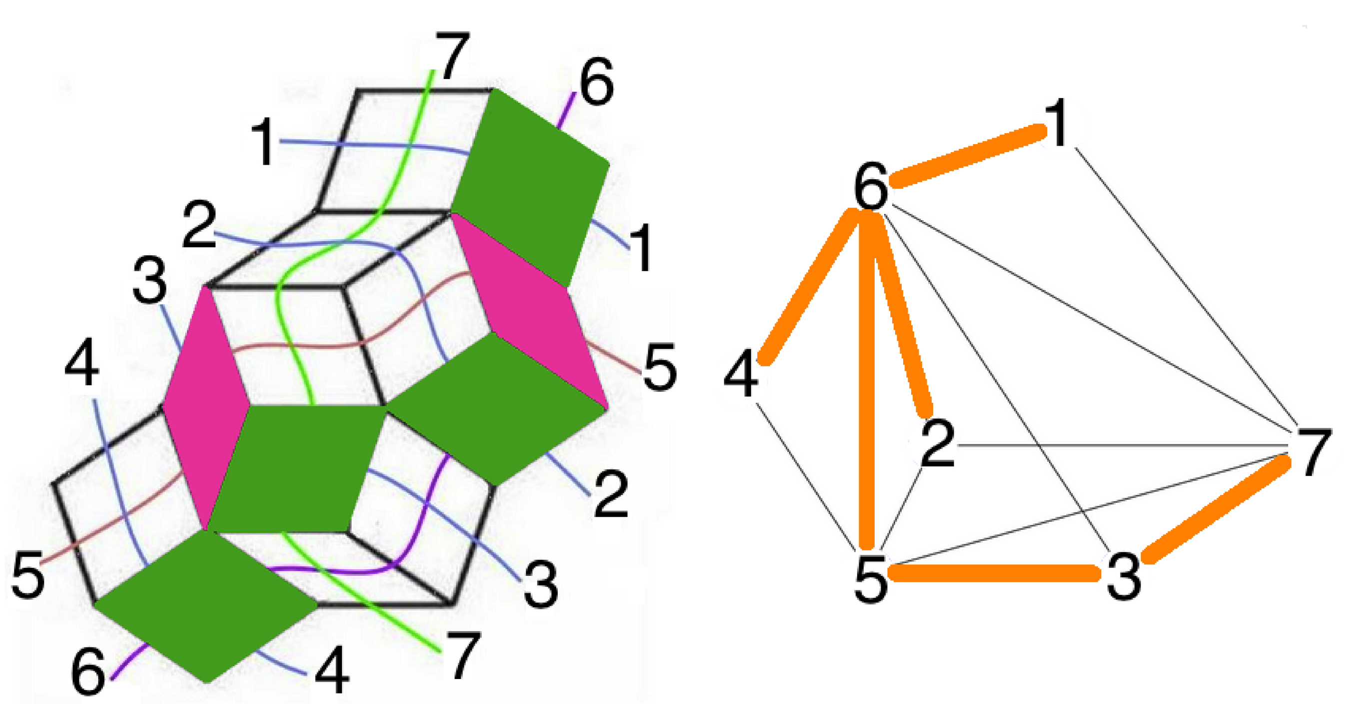


Figure 2: *A 2D quasicrystal rod and pinion framework is rigid if and only if the colored graph on the right is spanning and connected. Each edge on the graph corresponds to a plated face of the 2D framework on the left*

Goal

1. Gain the necessary coding abilities to be able to work on this very difficult problem.
2. Work on and improve programs that tackle this problem in an experimental manner.

Logistic Chaos Equation

The logistic chaos equation is the equation $f(x) = 4ax(1-x)$ where $0 < a < 1$

Question:

If we start with a value a and then look at the sequence of values $\{f(a), f(f(a)), f(f(f(a))), \ldots\}$, then does this sequence stabilize in some form?

In order to analyze this question we initially used the "Allerton code". It uses computer graphics to help us visualize the result.

Allerton Code Steps:

1. Pick a starting point on the curve $(x, f(x))$.
2. Plot a horizontal line from this point to $(f(x), f(x))$ on the diagonal line $y = x$.
3. Plot a vertical line from this point to $(f(x), f(f(x)))$ on the curve.
4. The next iteration begins from this point on the curve

The team members *Lisa Yi* and *Manting Huang* managed improve upon the code in the following ways:

Improvements

1. The original code was written in Python. Rewrote parts of the original code so that the whole program can now be run in javascript.
2. Added functionalities to the code. In particular, now the user can customize the color, height and altitude of the graph and witness different results.

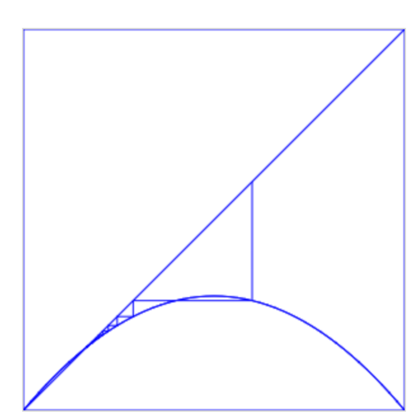


Figure 3: *Sequence with no repeating point that converges to 0*

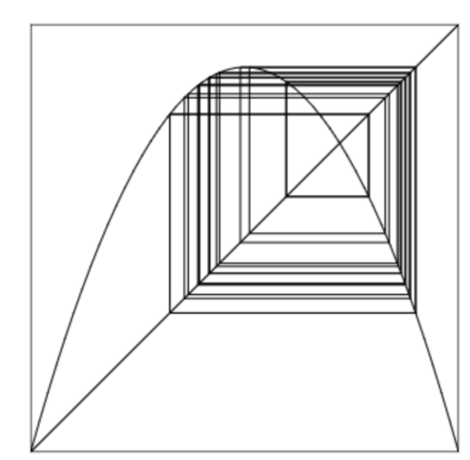


Figure 4: *Sequence with two repeating points (period 2)*

Visualization of Geometric Objects

The code *glsim.js* in [1] allows us to visualize geometric shapes such as *tori*, *octahedra* and *cubes*. Team member *Sung Jib Kim* rewrote part of the code which visualize tori.

Improvements

1. Added the ability to modify the size and color of the linked tori.
2. Added custom parametrization of the torus.

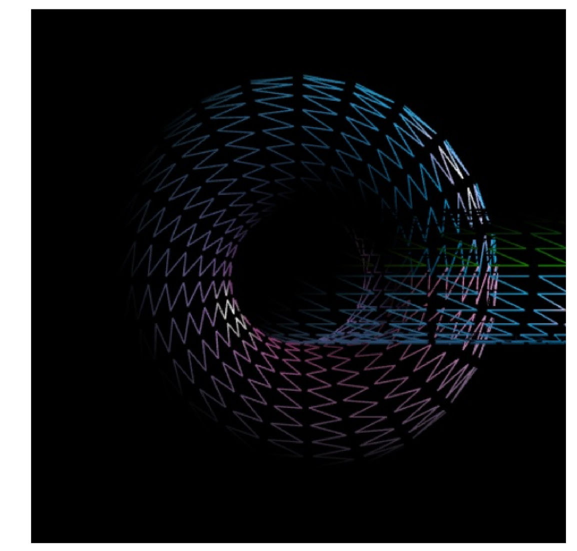


Figure 5: *Original linked Tori*

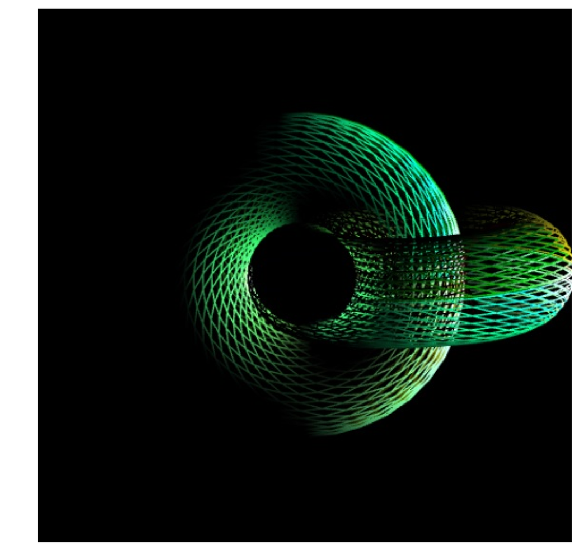


Figure 6: *Improved linked Tori*

Construction of Quasicrystal bricks

In [2], Eck introduces codes that allow us to build simple cubes, which has been heavily modified by team members *Pranav Bhardwaj* and *Tejo Nutalapati*:

Improvements

1. Changed the coordinates using *Gray coordinates for cubes* to change the shape from a cube to a quasicrystal brick
2. Used *triangle fans* to modify the coloring
3. Added functionalities such as rotation and translation.

The eventual goal of this code is to be able to move different bricks by rotation and translation so that we can build larger shapes out of them.

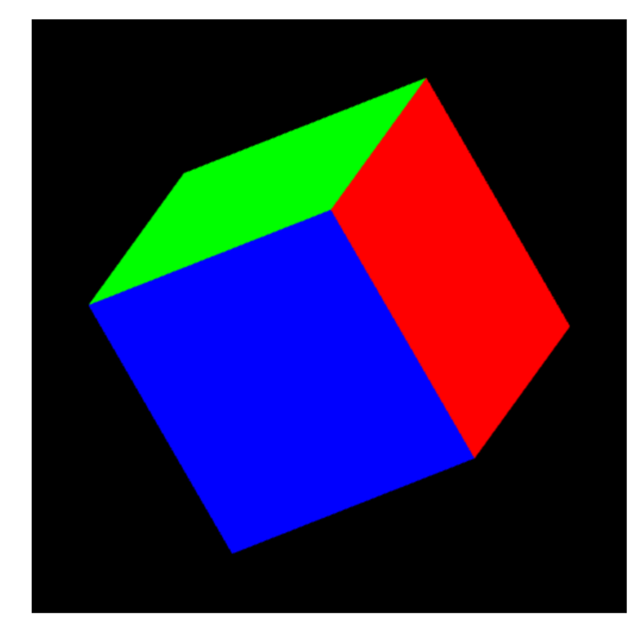


Figure 7: *Simple Cube*

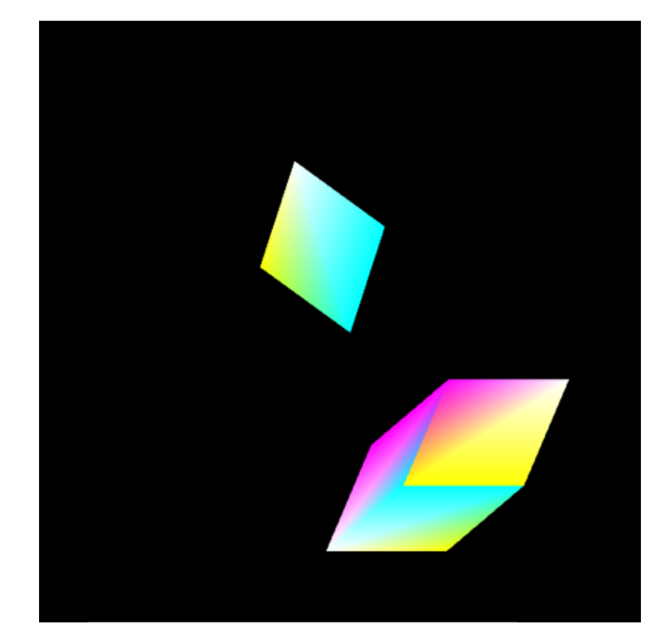


Figure 8: *Quasicrystal brick*

Future Directions

1. Formulate and prove a 3D version of Westers Theorem.
2. Learn the programming languages required to model this problem using real time interactive animations (RTICAs).
3. Use RTICAs to experiment with possible generalizations to Westers Theorem.

References

[1] David Eck. *Introduction to Computer Graphics.* Online: http://math.hws.edu/graphicsbook/ (2016)

[2] Gélvez, Eliana M. Duarte, and George K. Francis. *Stability of Quasicrystal Frameworks in 2D and 3D.*

Support for this project was provided by the Illinois Geometry Lab and the Department of Mathematics at the University of Illinois at Urbana-Champaign.