

# Random 3D Polyforms

Dhruv Gramopadhye, James Serrano, Khoi Tran

George Mason University, MEGL

May 6, 2022



# The Research Question

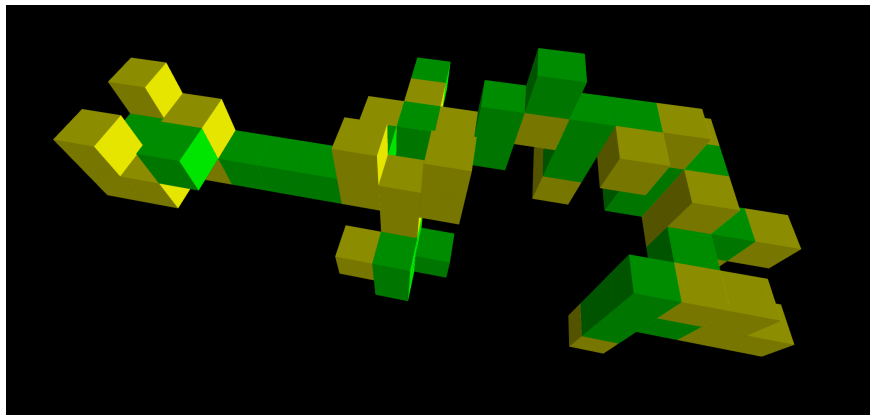
## Definition

A polyform is a figure constructed by joining together identical polytopes connected by at least one of their faces.

## Random 3D Polyforms

Specifically, we are studying random polyforms composed of strongly connected cubes. Every cube is connected by a path of cubes sharing two dimensional square faces. We're interested in the geometric characteristics of polyforms, such as the perimeter, radius, and number of holes of such generated shapes. Polyforms have applications in statistical physics, chemistry, and mathematics.

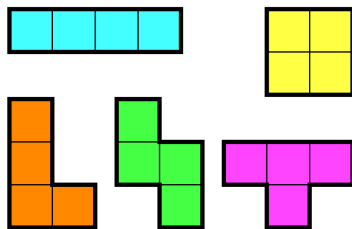
# Polyform of length 64, shuffled 9999 times



# The 2D Case: Polyominoes

## Definition

A polyomino is a plane figure constructed by joining together identical squares by at least one of their edges.

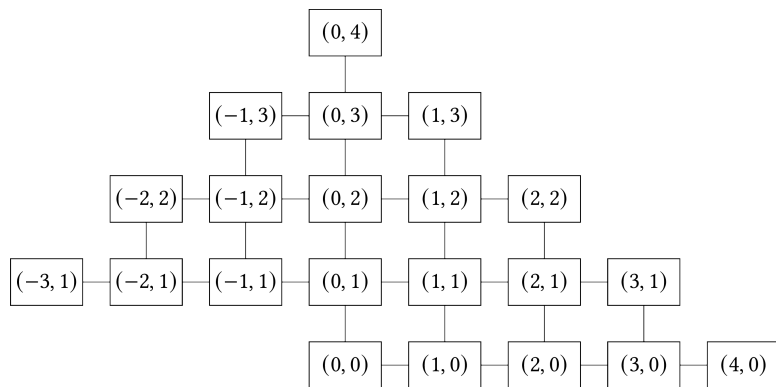


The video game Tetris uses tetrominos, a polyomino of size 4 (comprised of 4 squares). In a case like this, its relatively trivial to count all the possible polyominoes. But what if the size of the polyomino is  $\gg 4$ ?

# Redelmeier's Algorithm

- 1 Create an "empty" polyomino as the parent, with an "untried" set of nodes with only the origin.
- 2 Remove a random entry in the untried set.
- 3 Add a node to the graph.
- 4 Count the new polyomino.
- 5 If the size is less than  $n$ :
  - 1 Add new neighbor nodes to the "untried".
  - 2 Recursively call the algorithm, with the new polyomino becoming the parent.
  - 3 Remove the new neighbor nodes from "untried".
- 6 Remove the newest node.

# Candidate graph for pentominos(5)



# Pick's Theorem and the Euler Characteristic

## Theorem (Pick)

*Suppose that a polygon has integer coordinates for all of its vertices. Let  $i$  be the number of integer points that are interior to the polygon and  $b$  be the number of integer points on its boundary. The area  $A$  of the polygon is:*

$$A = i + \frac{b}{2} - 1$$

## Euler Characteristic

$$\chi = V - E + F = C - H$$

# Metropolis Hastings Algorithm

General process for sampling from a target distribution using a transition kernel:

- 1 Choose an arbitrary point  $x_t$  to be the first observation  $x_0$ .
- 2 For  $t = 1, 2, \dots$  pick a candidate  $y$  for the next sample by picking from a selected probability distribution.
- 3 Compute the acceptance probability,  $\alpha$ .
- 4 Generate a uniform random number  $u \in [0, 1]$ .
- 5 If  $u \leq \alpha$ , then accept the candidate,  $x_{t+1} = y$ . If  $u > \alpha$ , then reject the candidate and set,  $x_{t+1} = x_t$ .



# Metropolis Hastings Algorithm

Understanding the MH Algorithm with 3d polyforms:

- 1 Arbitrarily pick one of the cubes in the polyform.
- 2 Remove the cube from the polyform.
- 3 Select a neighbor of the remaining cubes uniformly at random.
- 4 Place the cube at the positions of one of the neighbors.
- 5 If the neighbor selected creates a valid connected polyform, keep the cube where it is. If not, put the original cube back as it was in the step 1.

# Optimizations

After a shuffle, we must check that the polyomino is still strongly connected. This operation takes notably longer than the shuffle itself.

- For 2D polyominoes, the estimated total shuffle time is between  $n^2$  and  $n^3$ , where  $n$  is the number of squares in the polyomino.
- We use a HashSet for near-instant lookups for pieces known to be strongly connected.
  - We don't need to check that the whole polyomino is strongly connected after a single shuffle operation; only pieces that were disconnected have to be checked for strong-connectedness, allowing an early stopping condition.
- Further algorithmic speedups are possible by applying the early stopping condition more aggressively and by porting this to a lower-level programming language and using multiple threads.

Much of the work in this semester has been to create and optimize the code base required to generate and study random polyominoes and polyforms. In the Fall, we intend to use the existing code:

- Sample from other distributions other than the uniform distribution
- Further study the homology of our random 3D polyforms
- Look for new and more efficient ways of shuffling polyforms, whilst still maintaining the desired distributions

# Acknowledgements

We'd like to thank our project mentors, Dr. Schweinhart and Dr. Roldan. We would also like to thank Aleyah Dawkins, our graduate research assistant, and MEGL for supporting our work. We're excited to continue in the fall of next semester.

## References:

- Kahle, Matthew, and Erika Roldan. "Polyominoes with Maximally Many Holes." *Geombinatorics Quarterly*, XXIX, no. 1, July 2019, pp. 5–15.
- Aleksandrowicz, Gadi, and Gill Barequet. "Redelmeier's Algorithm for Counting Lattice Animals." *Proceedings of the 27th Annual ACM Symposium on Computational Geometry - SoCG '11*, June 2011, pp. 283–284., <https://doi.org/10.1145/1998196.1998238>.