# An Algorithm For Drawing Cubes

Robey Holderith

Denison University - CS 271

October 18, 2005

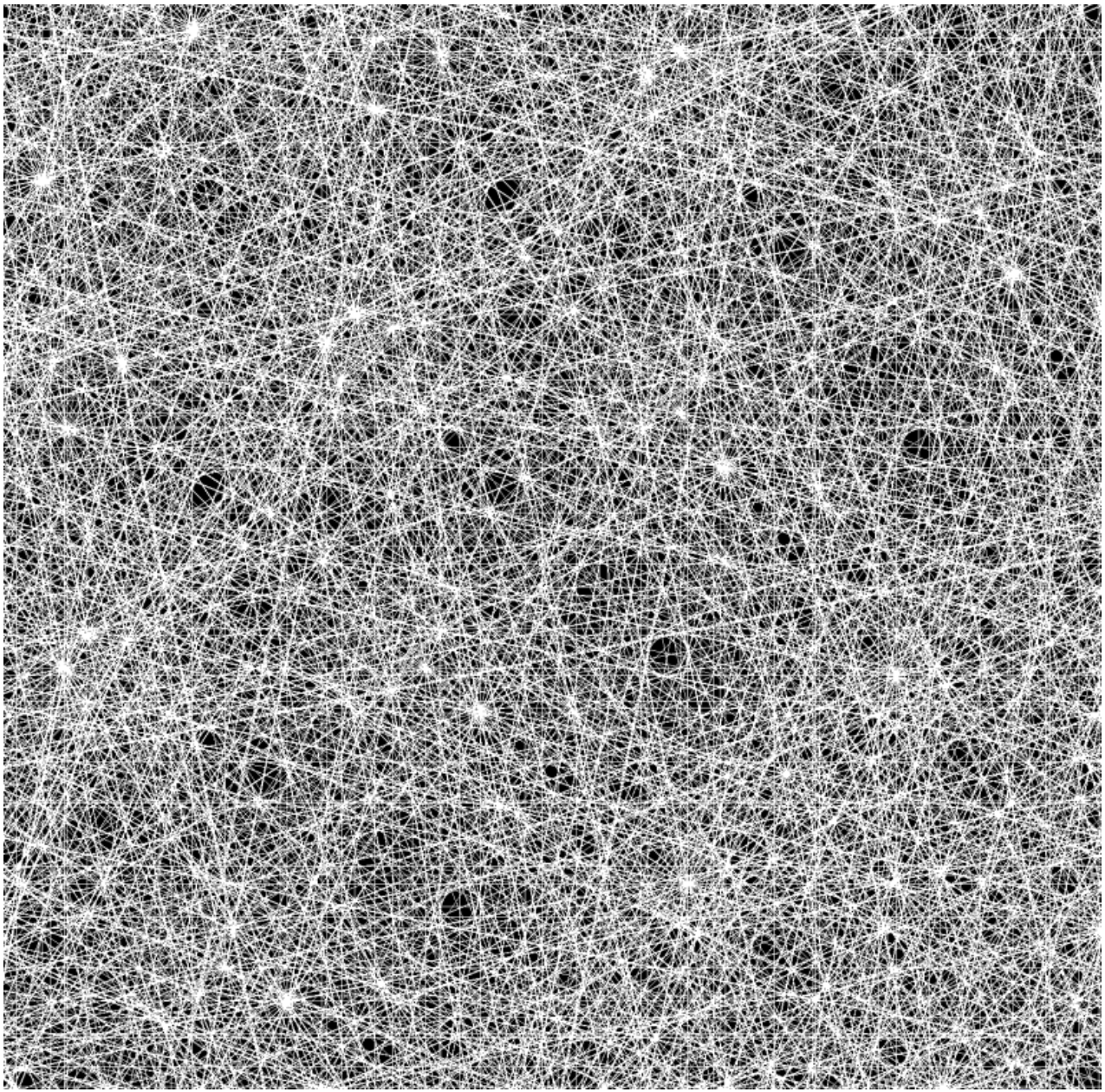
# What is an Algorithm?

An algorithm is a step by step process to do some task. Some examples include:

- A Recipe
- Accomplish Wealth in 3 Easy Steps
- Newton's Algorithm
- Method Presented Last Presentation

# Computer Algorithms

- A sequence of steps that can be done by a computer.
- Minimize time.
- Minimize memory use.
- Efficiency is often measured in O notation.



# How Many Lines?

- Define $F(n)$ = number of lines where n is dimension.

By examining the algorithm presented last time:

- $F(n) = 2F(n-1)+$vertices
- $F(n) = 2F(n-1) + 2^{n-1}$

Through induction you can find the closed form:

- $F(n) = n2^{n-1}$

# Drawing Lines

Actually drawing the lines is the simple part.

- Each point is split into $n$ components.
- Each component is assigned an $x$ and $y$ value.
- Sum up the values of each component to get an $x$ and $y$ value for each point.
- Draw a line between the translated points.

# Drawing Lines

- The amount of time to draw each line increases linearly with the dimension of the object being drawn.
- $O(n)$

# Generating Lines

Generating the list of lines to draw is a bit more complex.

- A lower bound can be found by multiplying the efficiency of drawing a line by the number of lines drawn.
- A lower bound is $O(n * n2^{n-1})$ or $O(n^2 2^{n-1})$.

How can we generate all (and only all) necessary lines?

# A Naive Algorithm

- Go through all possible points.
- Draw a line from each point to each of its neighbors.

# A Naive Algorithm

- This algorithm draws each line $n$ times.
- $O(n^3 2^{n-1})$
- Doesn't even address how to go through each possible point.

# A Recursive Approach

- Start at the origin.
- Draw a line only to upward neighbors.
- Run again for each upward neighbor.

# A Recursive Approach

- Many points have multiple downward neighbors.
- This works fine for lower dimensions.
- For $n \geq 3$ this begins to draw more lines than necessary.

# A Second Recursive Approach

- Start at the origin.
- Draw a line only to upward neighbors.
- Run again for each upward neighbor that doesn't have a lower neighbor after this point.

# A Second Recursive Approach

- Every point will have at most one path leading to it.
- Every point will have at least one path leading to it.
- Every line will be drawn exactly once.

Thank You.