Rapid-Prototyping of Rapid-Prototyping Machines

The Inner Workings of the Software

Steve Leibman
sleibman@mit.edu
Server

**Design Engine**
- Factor Graph
- Solver
- Math string
- Octree

**Virtual Machine**
- CAM (generate toolpath)
- VM rules and capabilities
- Real Time Control

**Physical Machine**
- Axis 1 i0 Node
- Axis 2 i0 Node
- Axis 3 i0 Node
...
Thin Client UI

Design
Visualization

Server

Design Engine

Factor Graph
Math string
Solver

Virtual Machine

CAM
(generate toolpath)

Real Time Control

VM rules and capabilities

Physical Machine

Axis 1 i0 Node Axis 2 i0 Node Axis 3 i0 Node …
MATH STRING REPRESENTING A SQUARE

\(((X \geq 0.5) \& (X \leq 1.5)) \& \newline\ (Y \geq 0.5) \& (Y \leq 1.5))\)
SLIGHTLY MORE COMPLEX...

\[ ((X \geq 0) \& (X \leq 5) \& (Y \geq 0) \& (Y \leq 32)) \mid ((X \geq 9) \& (X \leq 14) \& (Y \geq 10) \& (Y \leq 32)) \mid ((X \geq 18) \& (X \leq 23) \& (Y \geq 0) \& (Y \leq 32)) \mid ((X \geq 27) \& (X \leq 32) \& (Y \geq 0) \& (Y \leq 23)) \mid ((X \geq 27) \& (X \leq 32) \& (Y \geq 27) \& (Y \leq 32)) \mid ((X \geq 36) \& (X \leq 41) \& (Y \geq 0) \& (Y \leq 23)) \mid ((X \geq 36) \& (X \leq 54) \& (Y \geq 27) \& (Y \leq 32)) \]
... WHICH REPRESENTS:
2D
TOO SLOW...

- Bearable for most practical 2D applications
- Hit the limit quickly for 3D objects
REVISIT THE MATH STRING FOR A SQUARE

\(((X \geq 0.5) \& (X \leq 1.5)) \&
(Y \geq 0.5) \& (Y \leq 1.5)\)
IMPROVEMENTS

- Don’t perform the evaluation everywhere
- Don’t evaluate the entire expression
OCTREE / QUADTREE

An object:
OCTREE / QUADTREE

Start chopping:
OCTREE / QUADTREE

Recursively:
But without evaluating the math string everywhere, we don’t yet know where the boundaries are...

...remind me to answer that question 3 slides from now.
EXPRESSION TREE

\[
\begin{align*}
& \land \\
& = \land \\
& X \quad 0.5 \\
& \leq \\
& X \quad 1.5 \\
& = \land \\
& \geq \\
& Y \quad 0.5 \\
& \leq \\
& Y \quad 1.5
\end{align*}
\]
Subregions that only contain space in the $X \geq 0.5$ portion of the plane do not need to be resolved any further.

Subregions that span the $X=0.5$ line may (but are not guaranteed to) contain boundaries between solid material and empty space, so they do need to be resolved further.

Subregions that only contain space in the $X>0.5$ portion of the plane may contain boundaries and need to be resolved further, but for these areas, the math string can be pruned to a shorter expression, eliminating the first condition.
Our quadtree/octree shows us large swaths of area that can be ignored.
We do not have to evaluate the entire math string anywhere. Significant chunks are pruned away.
WHAT IF OUR SQUARE GETS A LITTLE CURVY?

( ( X >= \sin(Y/(2*\pi)) ) & (X <= 1.5) & 
(Y >= 0.5) & (Y <= 1.5))
EXPRESSION INCLUDES A RANGE

( ( X >= some value in the range [-1,1] ) & (X <= 1.5) & (Y >= 0.5) & (Y <= 1.5))
IMPLICATIONS

- Use interval arithmetic to deal with compound operations on ranges
- Slightly wider area in which the octree must be finely resolved
GENERATING TOOLPATHS

- Was easy when we had a fully resolved grid of values indicating True/False for the presence of material
- Do something analogous on the octree
GENERATING TOOLPATHS

- Evaluate (pruned) math string in octree leaf node region
- Compare with neighbors, identify boundaries
- Fully resolve to specified resolution along the boundary and where contours are requested
- Ignore everything else
traverse_speed = 8
cutting_speed = 4.0
plunge_speed = 4.0
z_down = 0.0
z_up = 0.1
move(z=z_up, rate=plunge_speed)
move(0.0, 0.0, z_up, traverse_speed)
move(z=z_down, rate=plunge_speed)
move(0.0462962962963, 0.0, z_down, cutting_speed)
move(0.0462962962963, 0.296296296296, z_down, cutting_speed)

...
...and then the exciting part happens. Things start getting built.