Washington, DC 2005

Automated Brick Layout

BrickFest 2005

David V. Winkler

Sunday 8/14/2005
10:30am-11:30am

Computer use with Legos

- Visualization
  - LDraw
  - MLCad
- Mosaic generation
  - GIMP
  - Pixelego
  - Lego Shop At Home
  - Lugnet
- Sculplture design
  - Legoland
- Sculpture layout
  - ?

Automated Brick Layout

- Model design
  - Use of free software
    - Pov-ray
    - Simple scripts to produce model outline
- Brick Layout
  - Software for taking outline and producing brick layout

Models into outlines

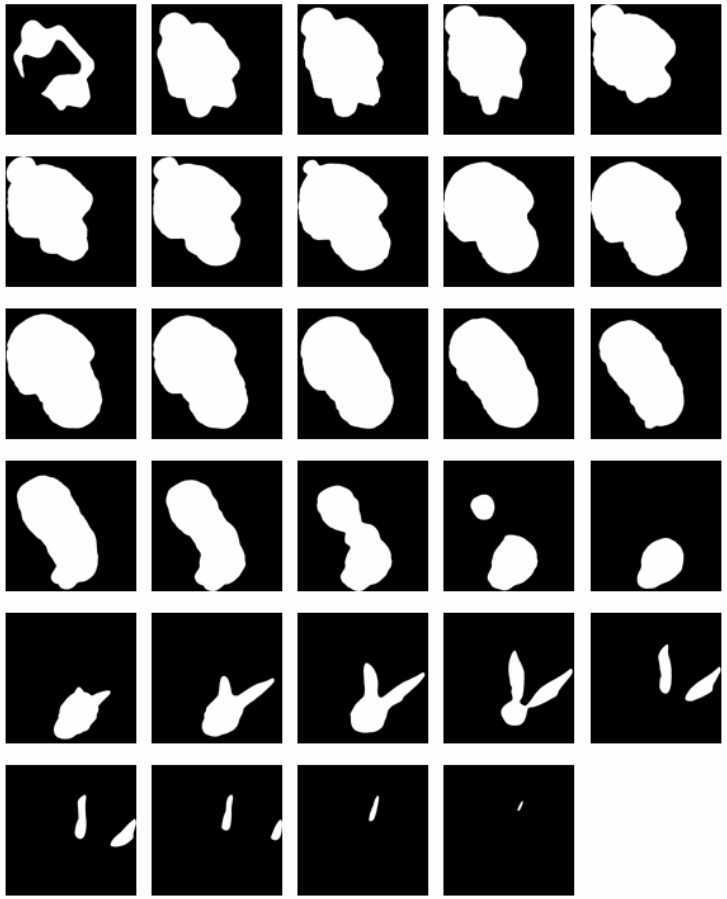
- Use modeling package to produce 3d object
  - Or grab one from the internet
  - Normally triangle mesh
- Turn this model into a pov-ray CSG object
  - Just add exterior vector
- Intersect model with two planes
  - Produce thin slice of model
- Animate planes moving vertically through model
  - Each frame is a Lego layer
  - Shrink frame resolution to desired Lego size
    - e.g. 48x48 for gray baseplate

Model

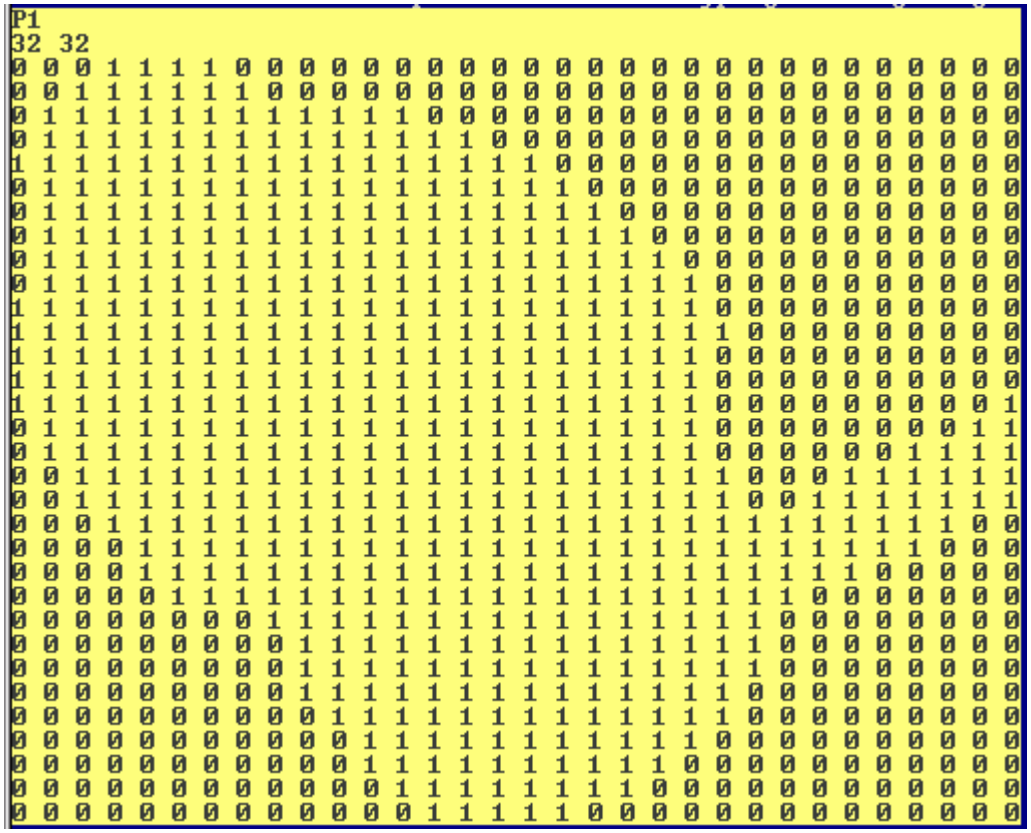
- Model courtesy of the Stanford Graphics Laboratory

Model from above (orthographic projection)

Frames of model



Shrink to correct size



Automated Brick Layout

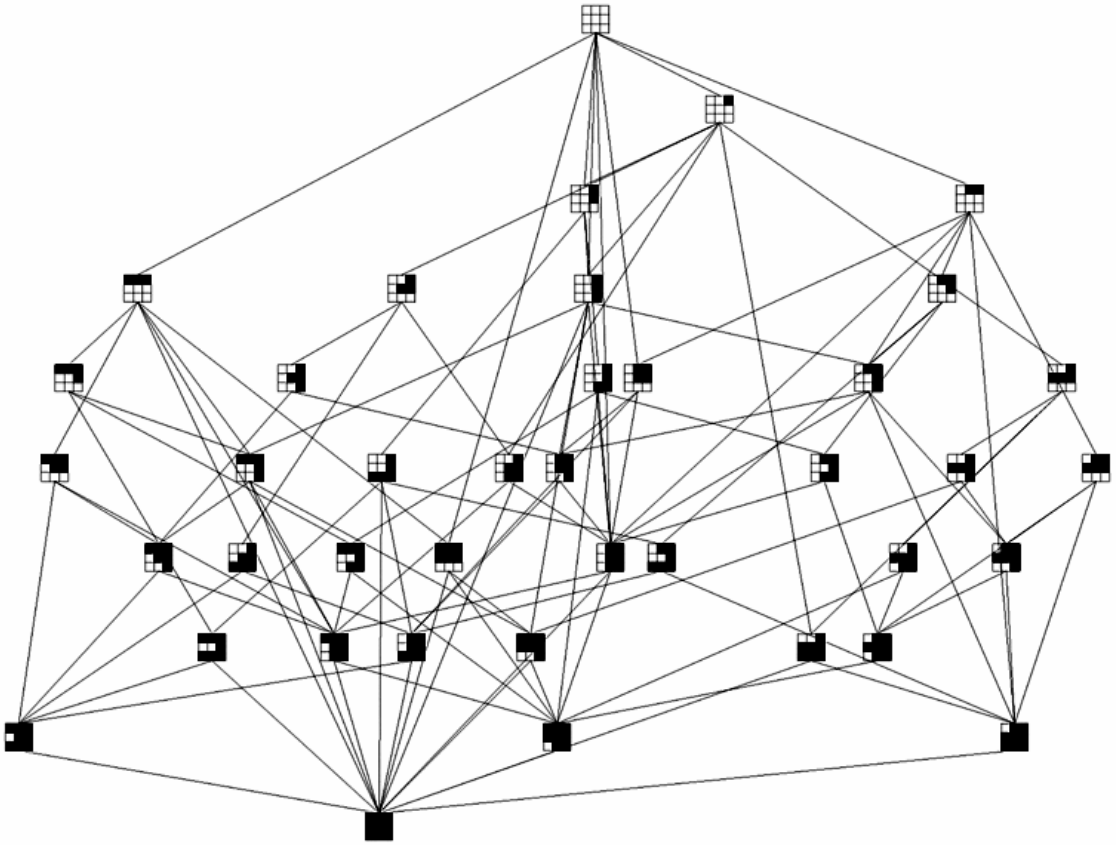
- General problem of filling area with a cost function is NP-complete
  - Like many problems special cases make it solvable
    - Small pieces
  - We will make some approximations
- Strategy
  - Try **lots** of possibilities
  - Apply a cost function
    - Piece price
    - Piece stability – how much that piece holds the model together
      - Try to keep 25% or less of piece over or under empty space
      - Better to bridge gap than cantilever
      - Better to cover more different pieces than match below layer

Automated Brick Layout

- How to try lots of possibilities
- Number each square to be filled
- At every step we try all possible pieces
  - At every step we will fill in lowest numbered unfilled square
    - Squares may not be filled in order
    - When all squares are filled we will have tried all possible fillings
  - We merge together states that fill the same squares
    - Use cost function to select a winner
  - Perform search in order of increasing number of filled squares
- This produces a lattice (directed acyclic graph with source and sink nodes)

Lattice of all fillings of 3x3 region

- Every path from top to bottom of this diagram is a unique filling.
- Only finished fillings of sorted piece order will appear
  - That's why center filled and center empty don't appear in the diagram

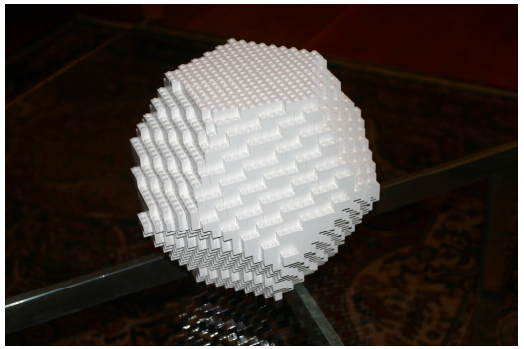


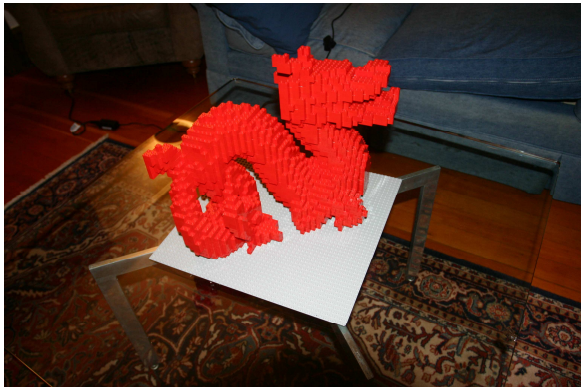
Making the lattice manageable

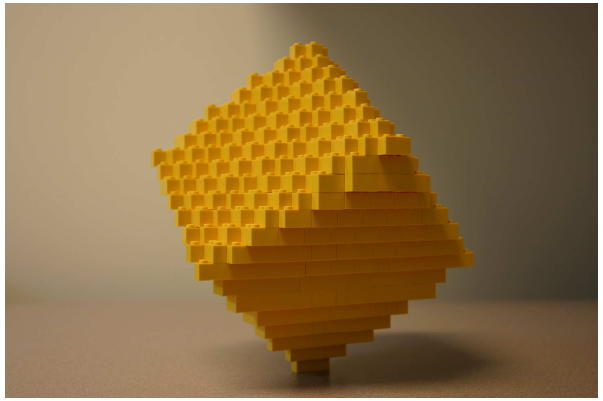
- Lossy techniques that work
  - Discard some high cost states
    - This is called a Beam Search
- Recovering from lossy techniques
  - Recursive overlapping search (10x speed improvement for same quality)
    - Fill region quickly, but poorly
    - Remove legos from top half and perform high quality filling
    - Remove legos from bottom half and perform high quality filling
    - Remove legos from middle half and perform high quality filling
    - Repeat until pieces stop changing

Results

- Large structurally sound models can be quickly constructed
- 3d meshes courtesy of the Stanford Graphics Laboratory and Cyberware, Inc.





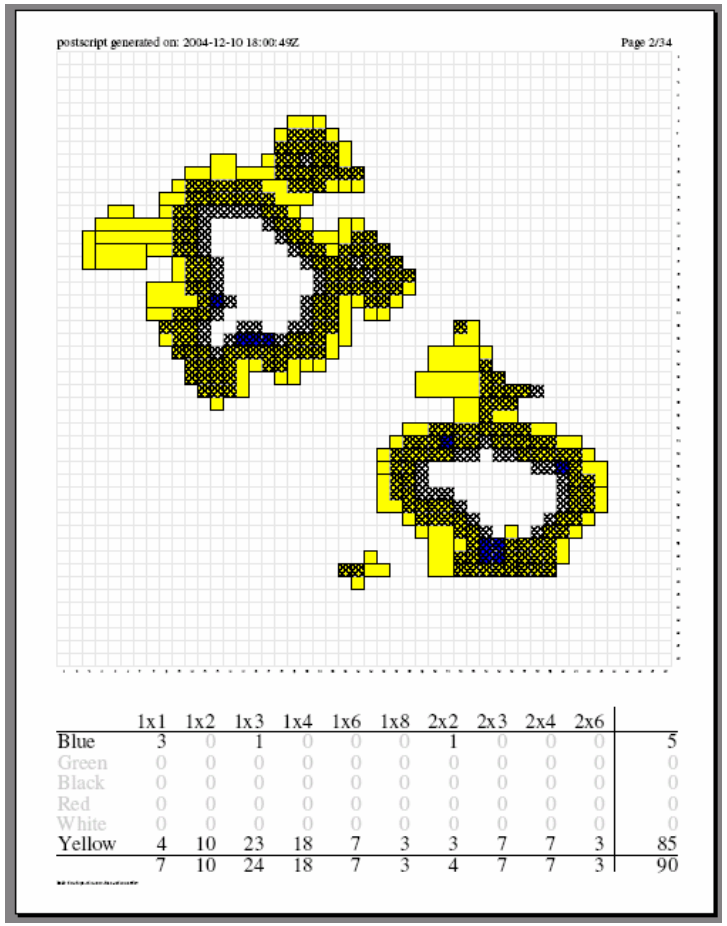


Making the lattice manageable

- Lossless techniques that work
  - Generate lattice on the fly – discard nodes deep in the past
    - Can discard any nodes deeper in the past than the area of the largest piece
  - Size of lattice is exponential in cross-sectional distance
    - Making the models hollow greatly reduces cross-sectional distance
    - Different numbering of squares to be filled
  - Remove common pieces to common list
  - Bit packing
    - Especially the list of pieces for a state

Offtopic: Instructions

- Hardest part of constructing model is following the instructions.
- Good instruction features:
  - Thick dark line surrounding each piece
  - Shadow of layer beneath
    - Allowing alignment
    - Check of previous layer
- What I did:
  - Parse ldr and generate postscript



	1x1	1x2	1x3	1x4	1x6	1x8	2x2	2x3	2x4	2x6	
Blue	3	0	1	0	0	0	1	0	0	0	5
Green	0	0	0	0	0	0	0	0	0	0	0
Black	0	0	0	0	0	0	0	0	0	0	0
Red	0	0	0	0	0	0	0	0	0	0	0
White	0	0	0	0	0	0	0	0	0	0	0
Yellow	4	10	23	18	7	3	3	7	7	3	85
	7	10	24	18	7	3	4	7	7	3	90

Commercialism

This technology is available for license from Microsoft™.

Please contact JoyMu@microsoft.com with licencing questions.

Thank you

Any Questions?

Please feel free to follow up by email:
happyfrosh@hotmail.com