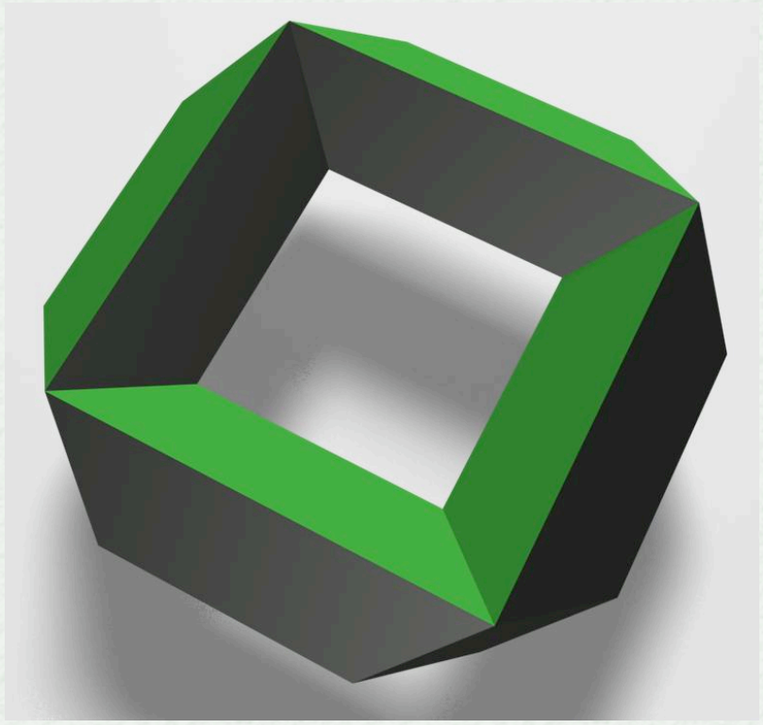
Creating Optimized Cut-Out Sheets for Paper Models from Meshes

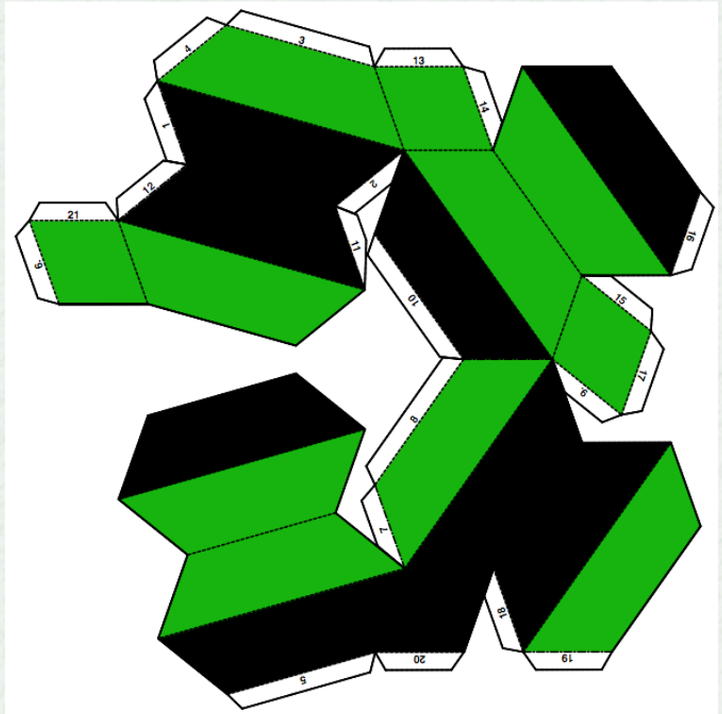
Raphael Straub
University of Karlsruhe, Germany

Universität Karlsruhe (TH)
Research University · founded 1825

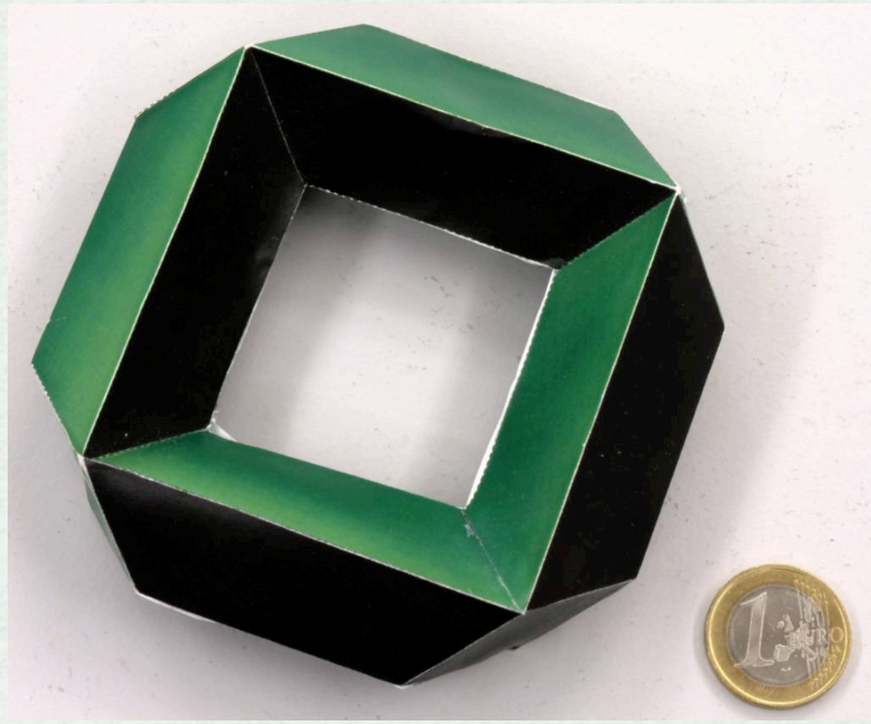
Goal



mesh



cut-out sheet



paper model

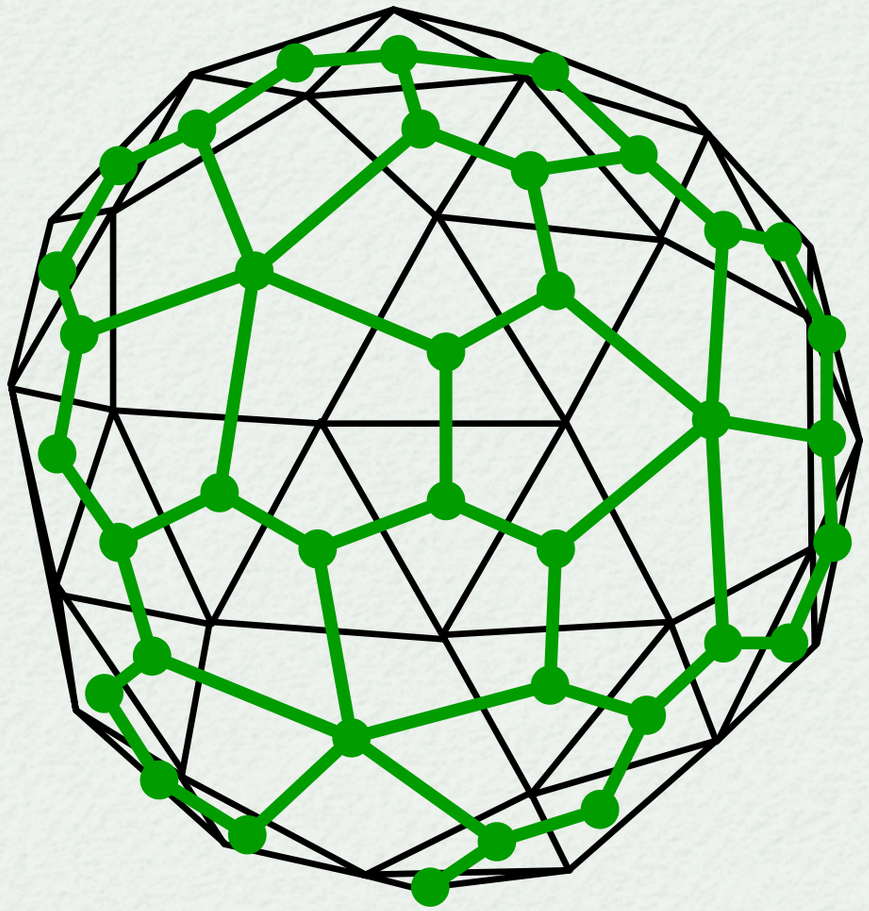
Algorithm

- Input: VRML file containing a 2-manifold textured polygonal mesh
- Output: optimized PDF file regarding
  - cutting, bending and glueing time
  - stability of glue joints
  - use of paper

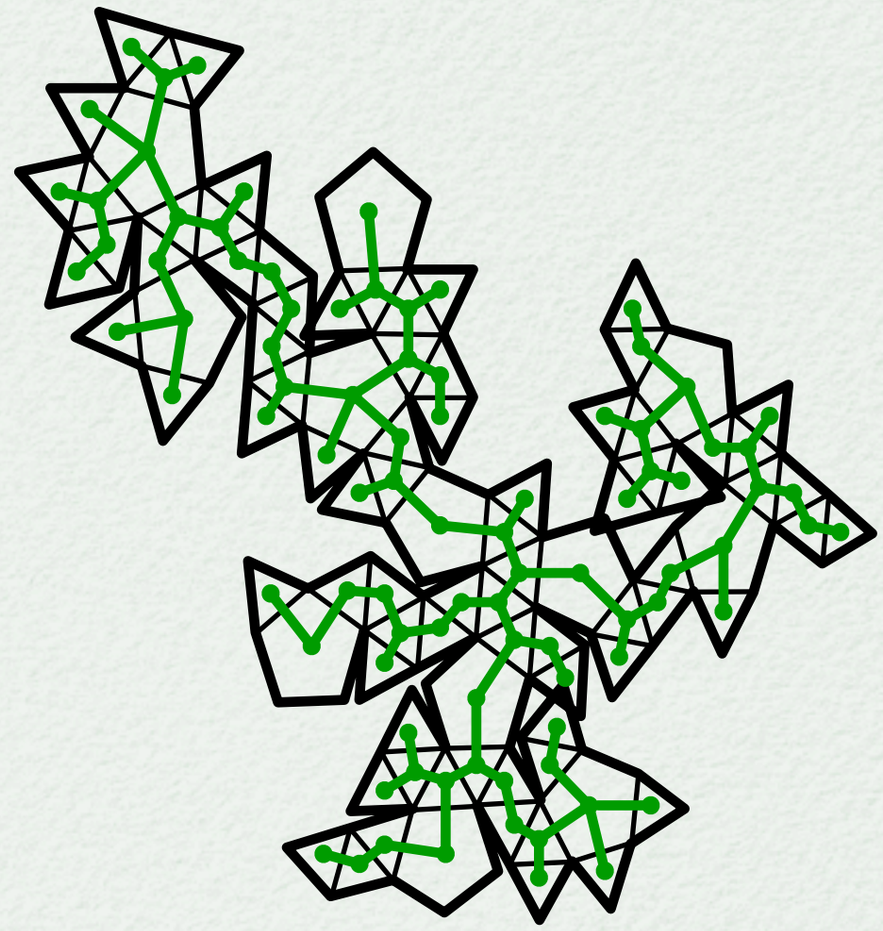
Algorithm Overview

1. Unfold mesh into the plane by cutting edges
2. Divide into parts to remove overlaps
3. Subdivide parts that are larger than a paper sheet
4. Pack parts on paper sheets

Unfolding the Mesh



Dual graph
of a mesh



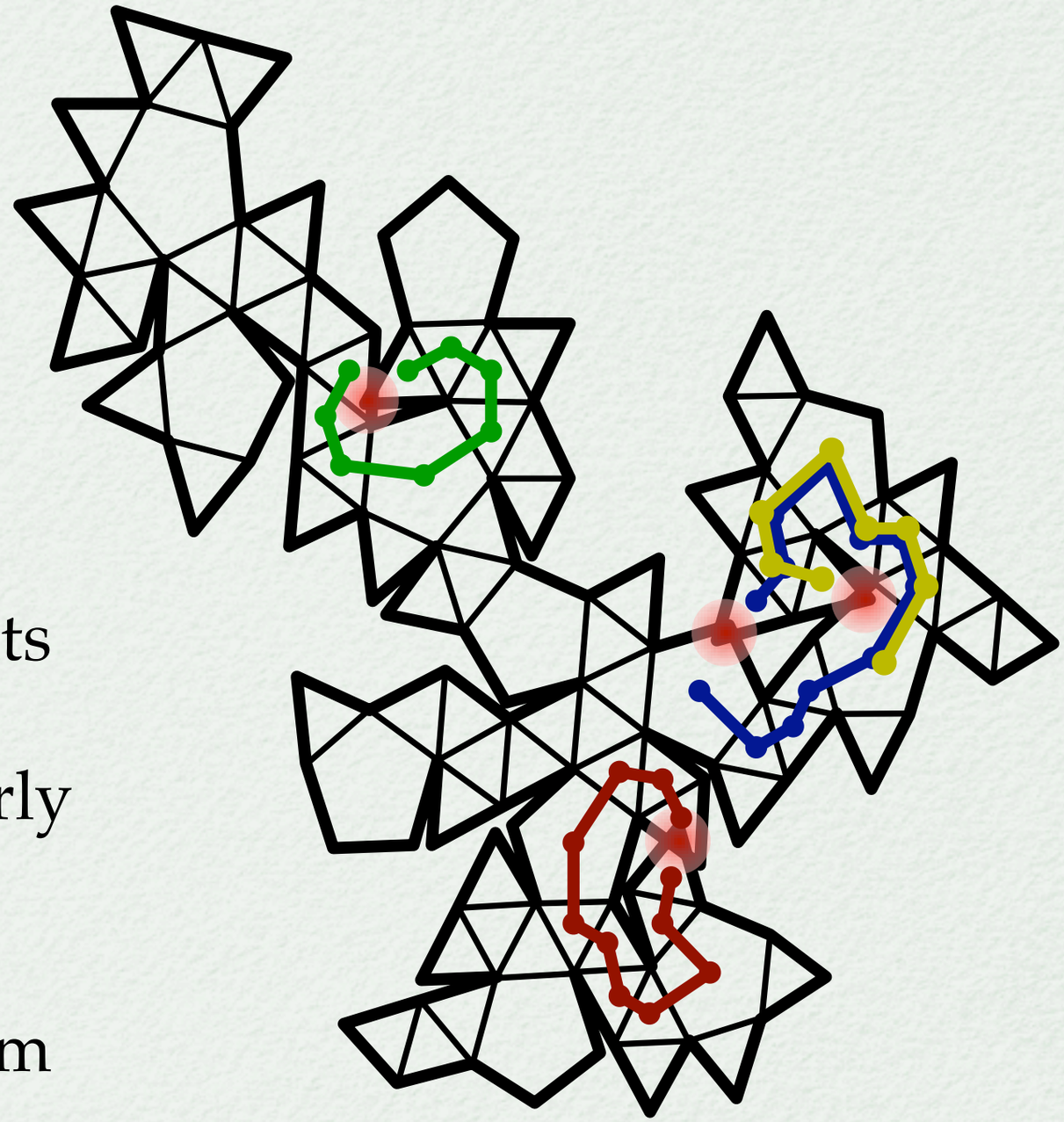
Spanning trees
correspond to
unfoldings

Unfolding Using a MST

- Assign weights for each edge of the dual graph, which are a weighted sum of
  - heuristics
  - user defined weights
- Compute initial unfolding as a minimum spanning tree (MST)

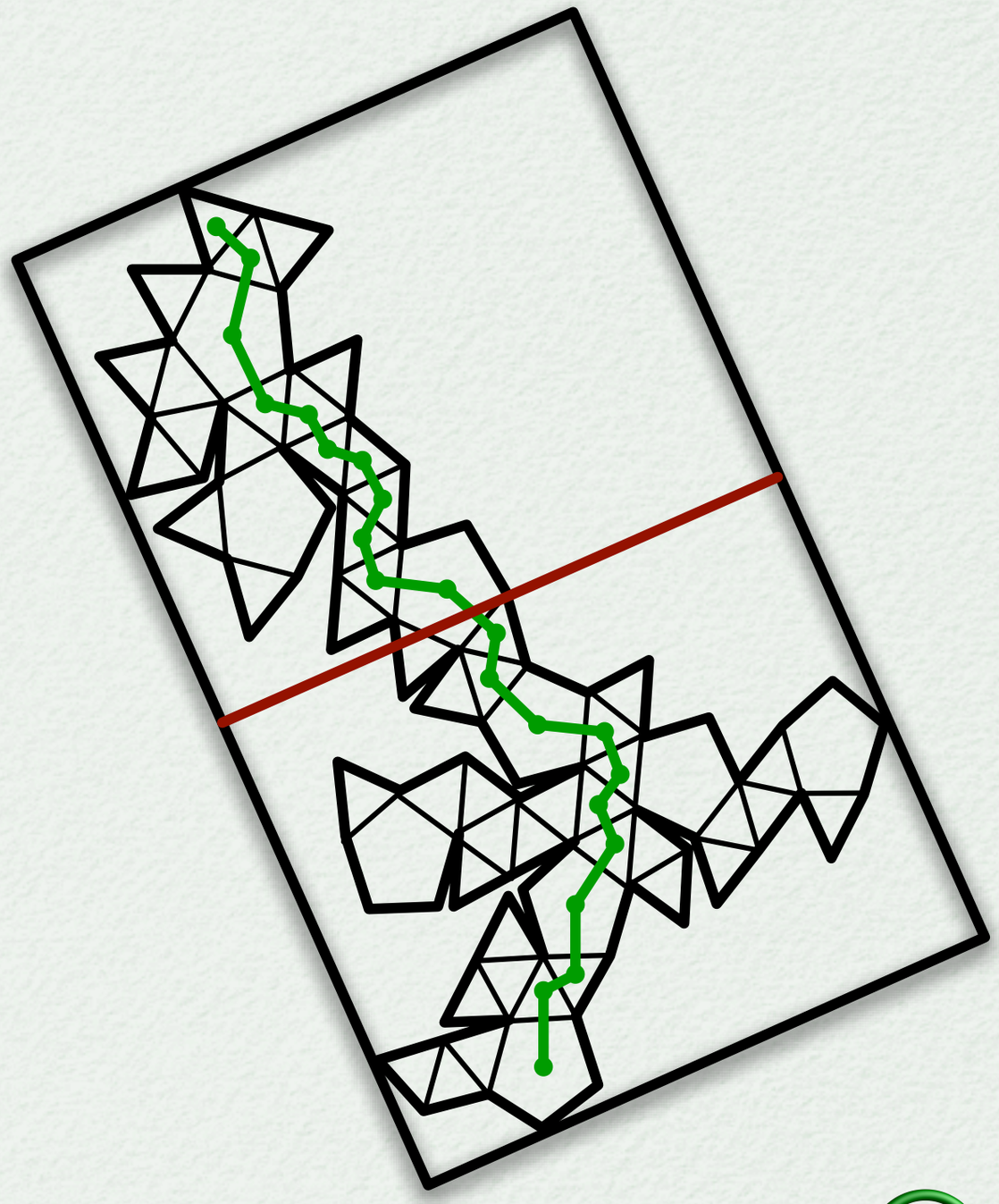
Removing Overlaps

1. Detect overlaps
2. Cut all paths in the dual graph between overlapping faces
   - minimize the number of cuts
   - prefer cuts resulting in nearly equally sized parts
   - minimum set cover problem



Subdividing Large Parts

- Compute minimum area bounding box of each part
- Iteratively subdivide parts along x- or y-axis of bounding box

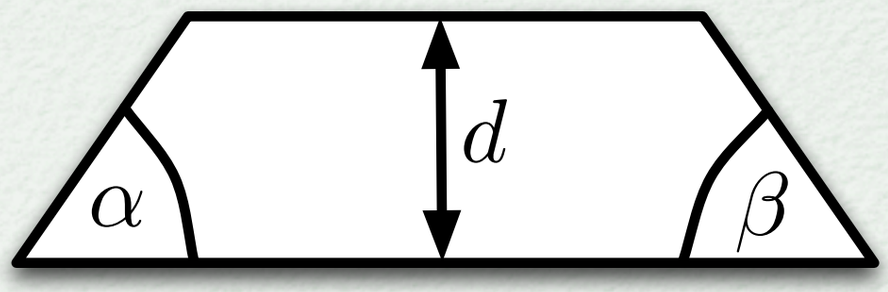


Packing on Paper Sheets

- Pack only bounding boxes of the parts
- 2D bin packing problem is NP-complete
- Solution: use approximation algorithm

Glue Tabs

- Trapezoidal shape
- Tab size
  - stable glue joints
    ⇒ minimum size
  - minimize use of paper and overlaps
    ⇒ maximum size



Tab Arrangement Formula

- Every edge has two potential tabs and every tab corresponds to a variable x_i.

$$A(x_i) = \begin{cases} \text{true}, & \text{if tab } i \text{ is present} \\ \text{false}, & \text{else} \end{cases}$$

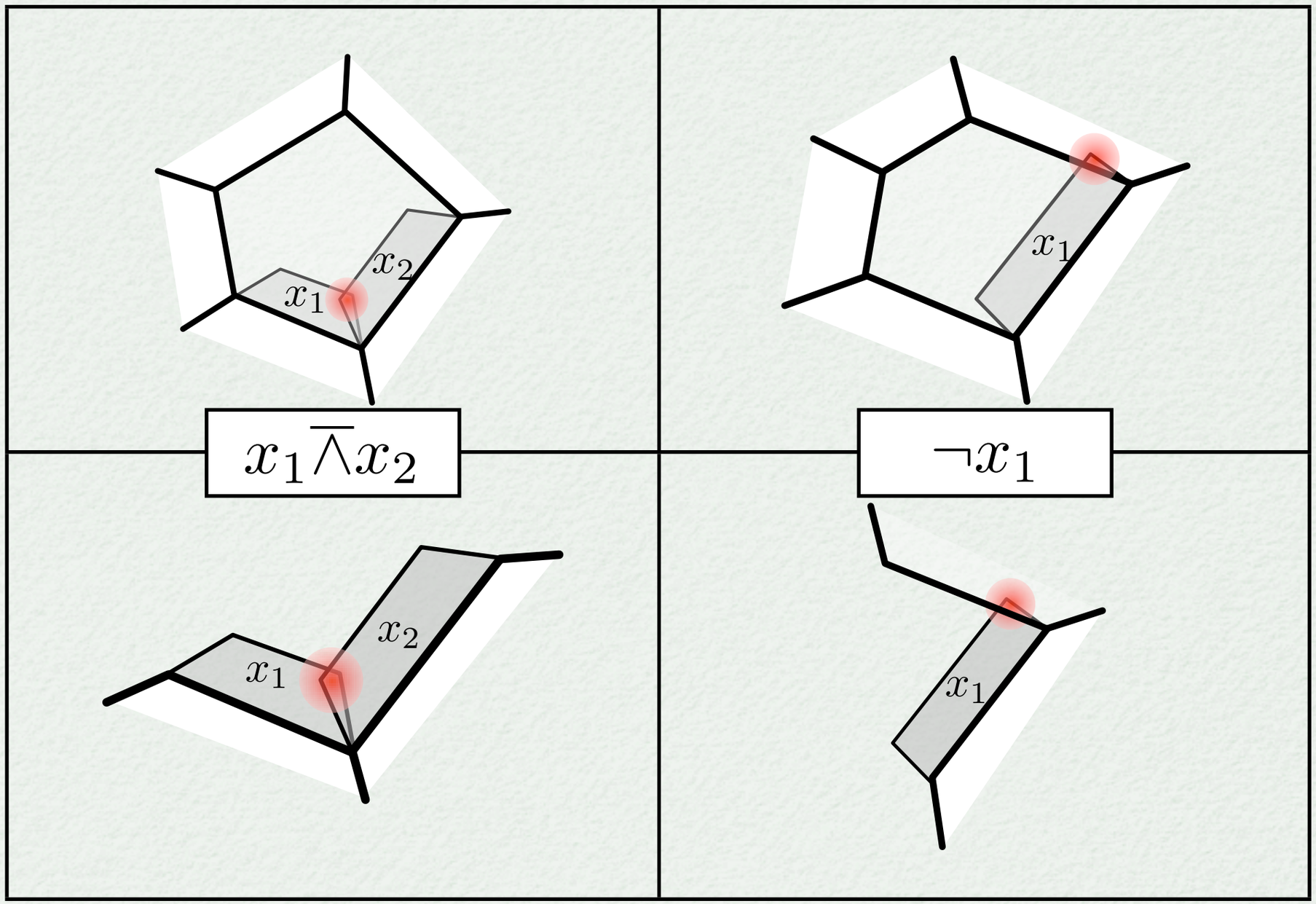
- Contraints on tab positions are expressed by a logical formula.

- A glue tab arrangement is valid iff the corresponding assignment satisfies the formula.

Glue Tab Algorithm

1. Determine glue tab arrangement with minimally sized tabs
2. Post-optimization to maximize tab sizes

Conflicts



tab / tab

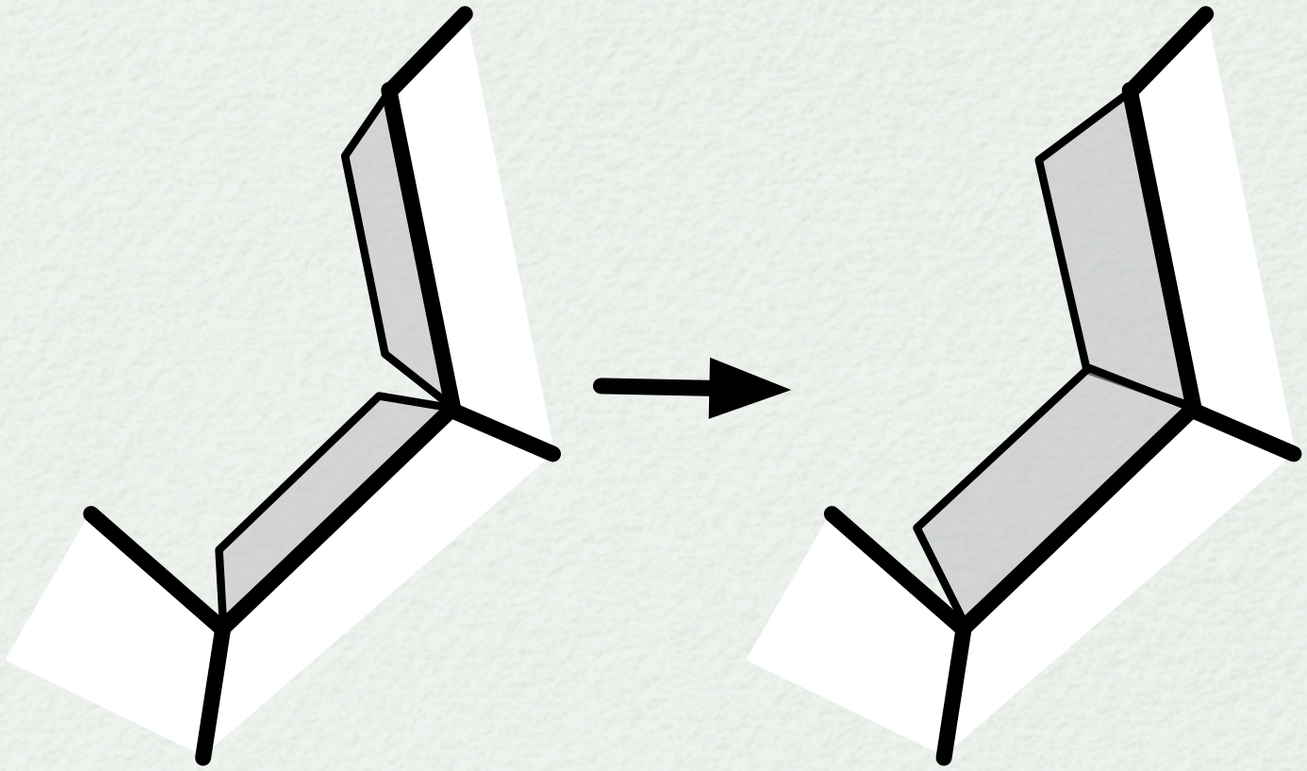
tab / face

Glue Tab Arrangement

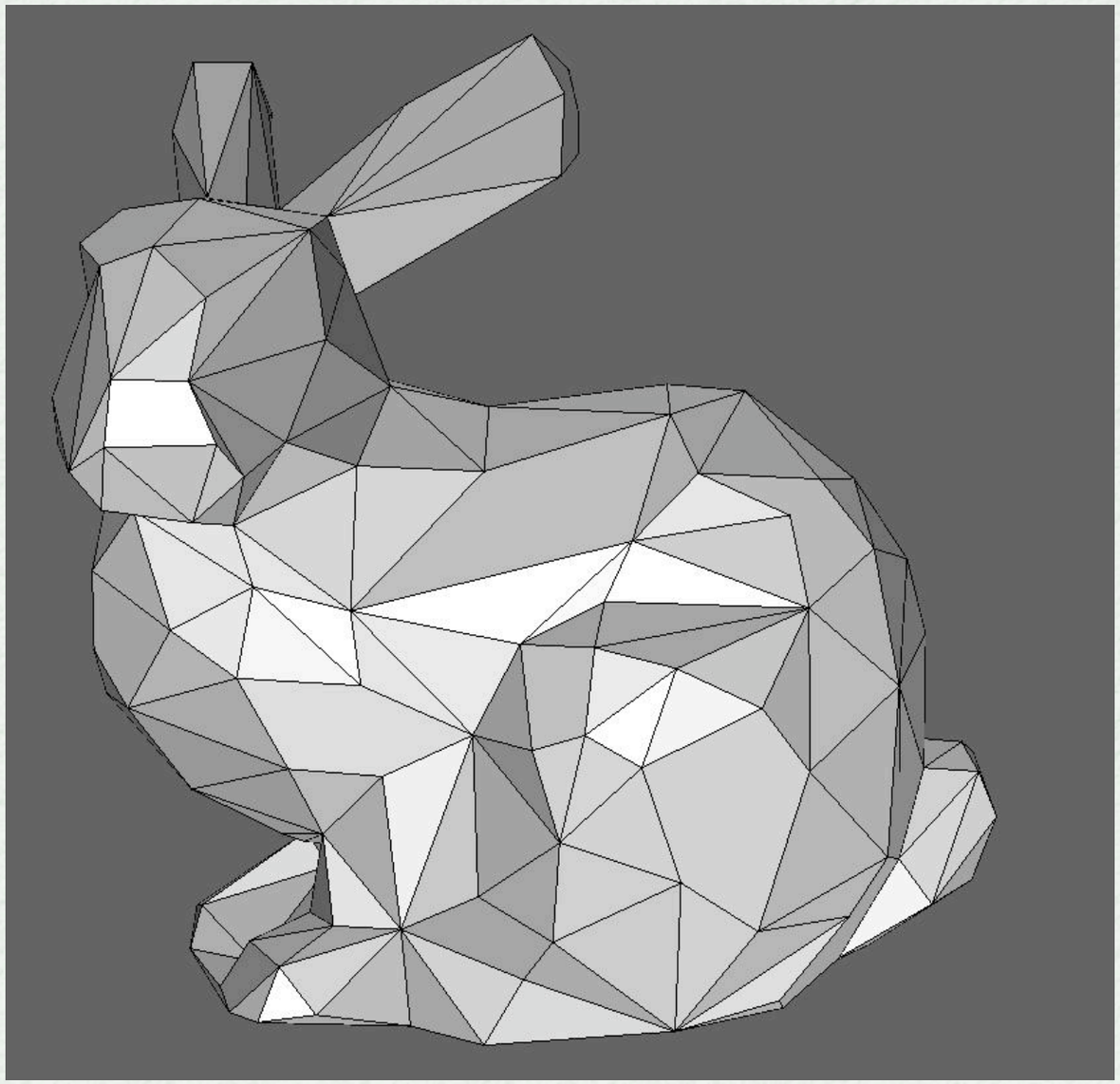
- Before unfolding:
  conflicts in 3D → NAND and NOT clauses
- During unfolding:
  new cuts → XOR clauses
- After unfolding:
  conflicts in 2D → NAND and NOT clauses
- Solve 2-SAT problem to get valid tab arrangement

Post-Optimization of Tabs

Iteratively grow all tabs until they have maximum size or touch



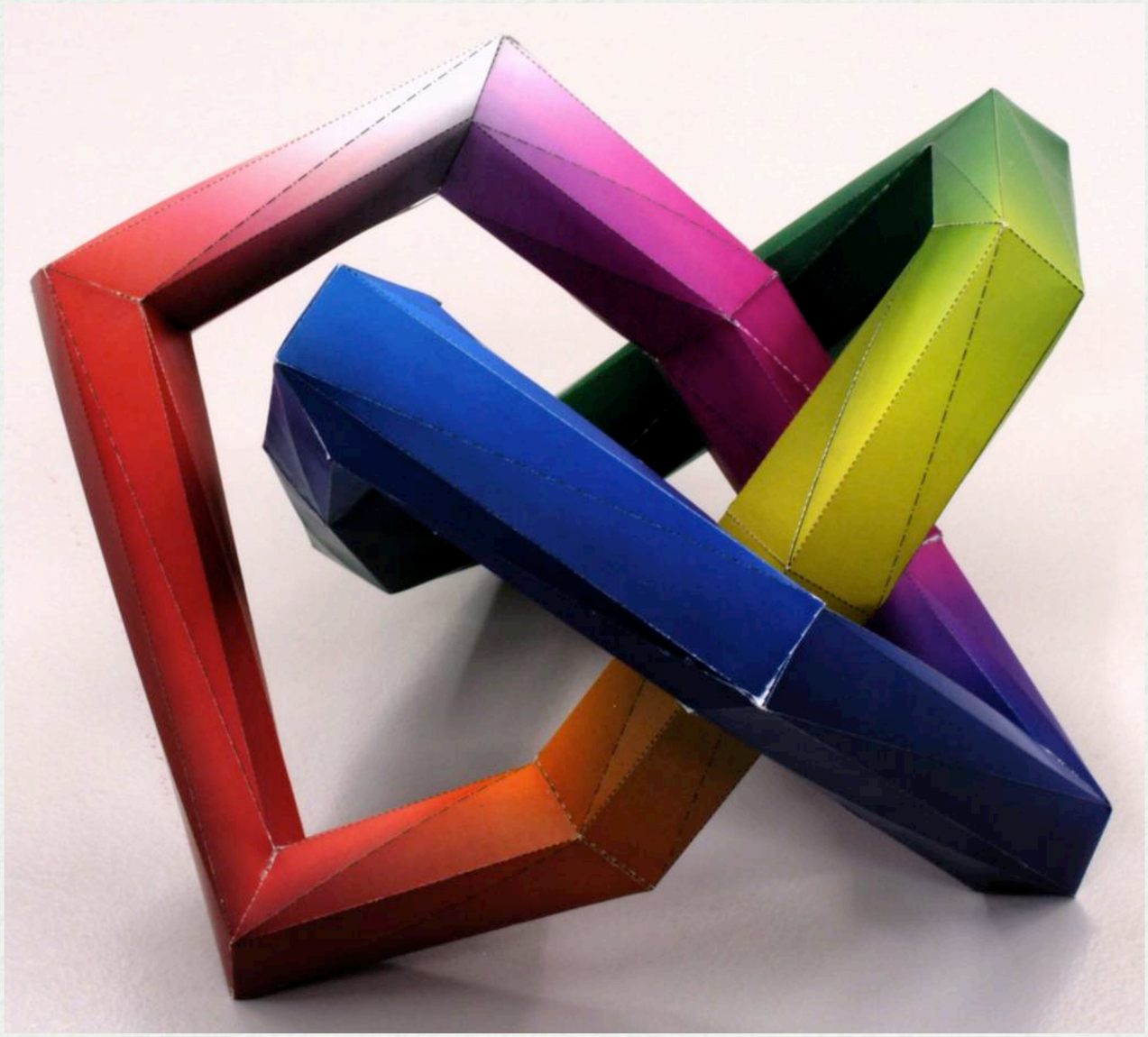
Stanford Bunny



348 polygons

12 h crafting time

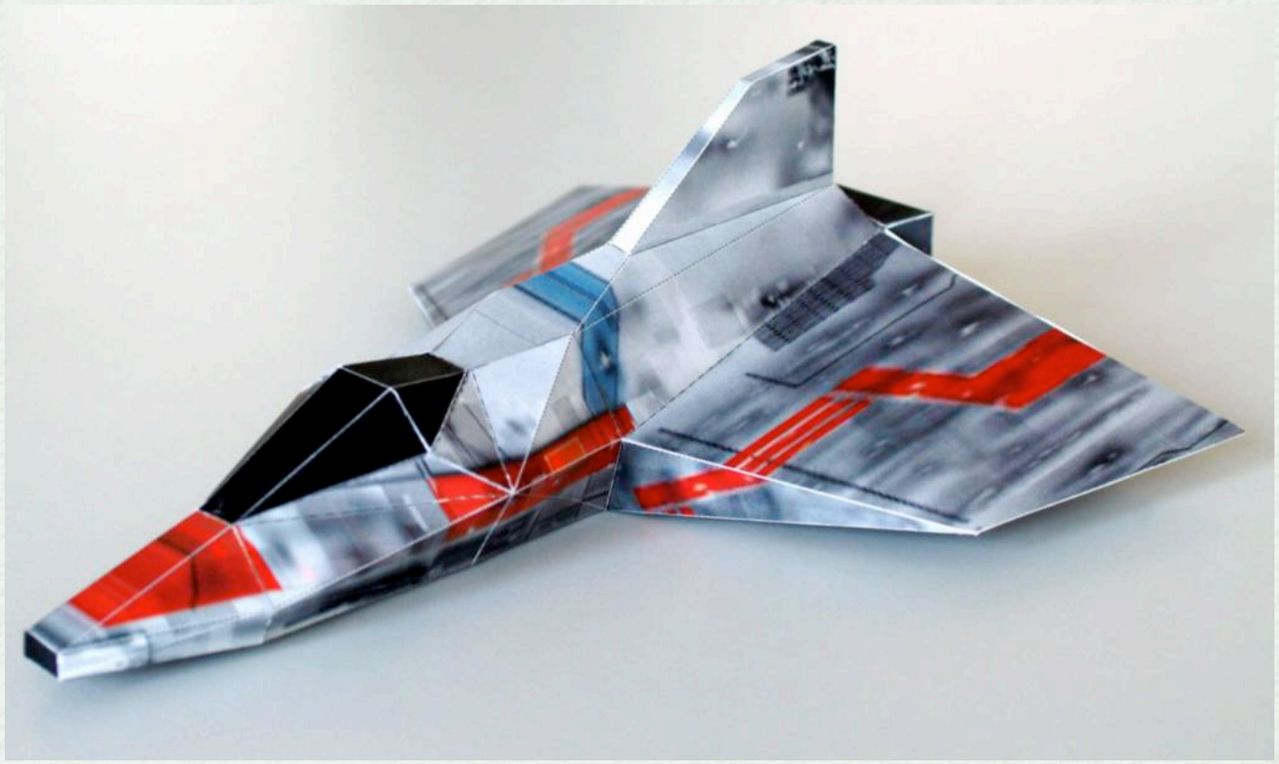
Torus Knot



143 polygons

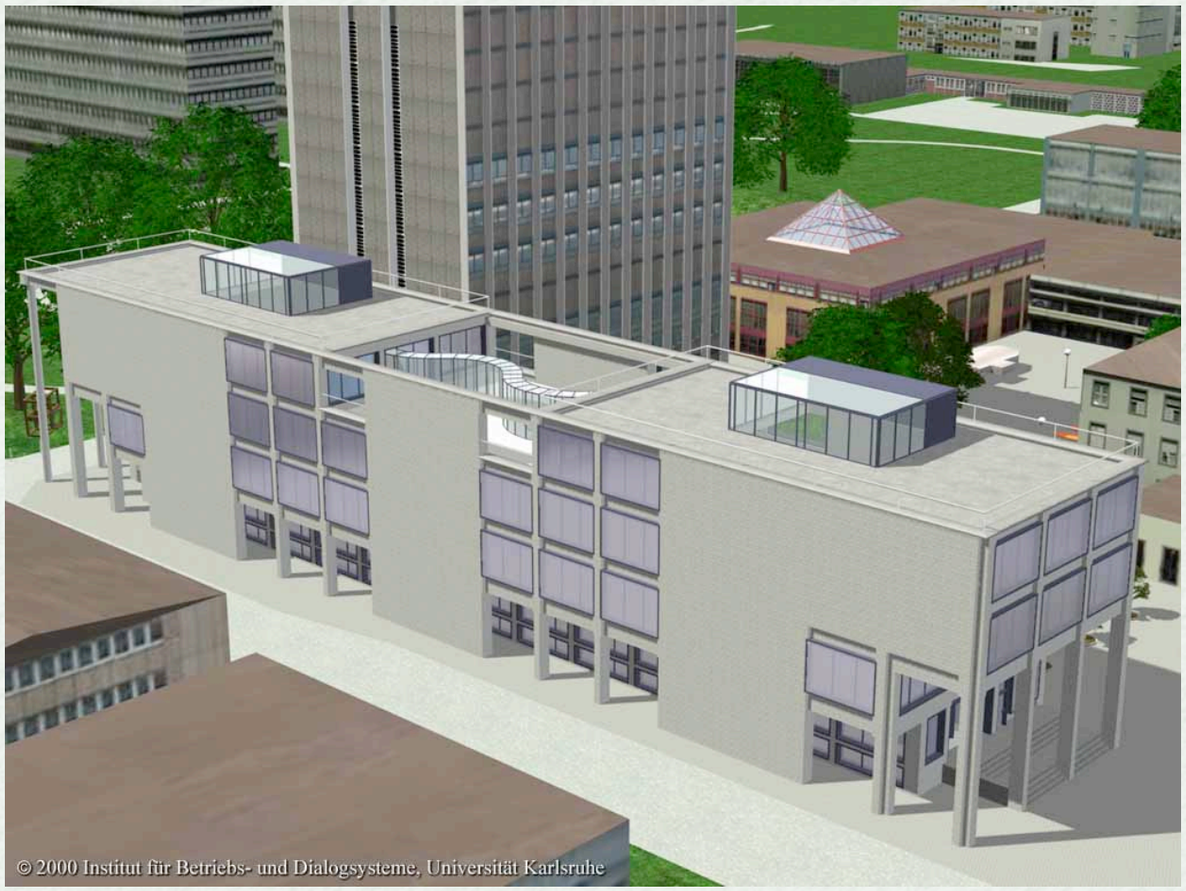
Space Ship

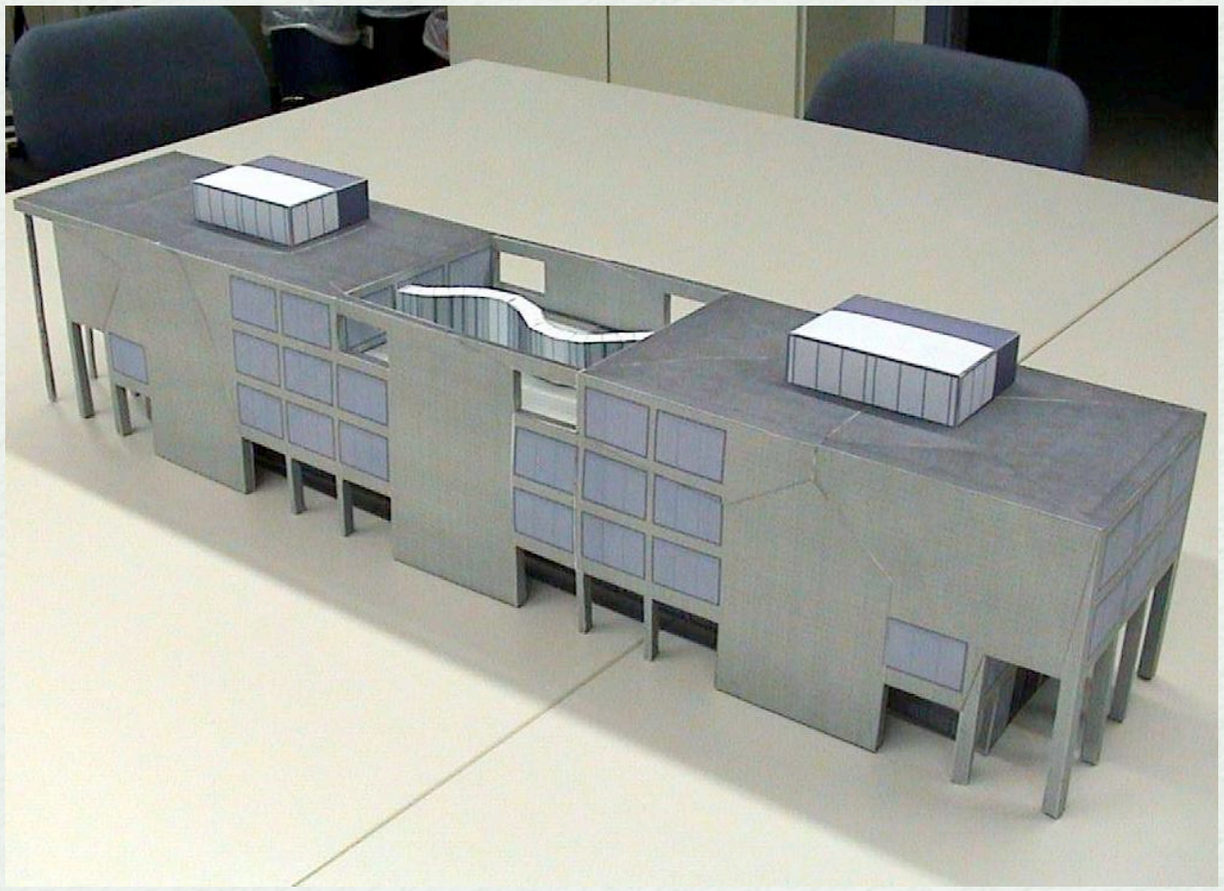
62 polygons



2 h crafting time

University Library





347 polygons

25 h crafting time

?

raphael@ira.uka.de
http://i33www.ira.uka.de/