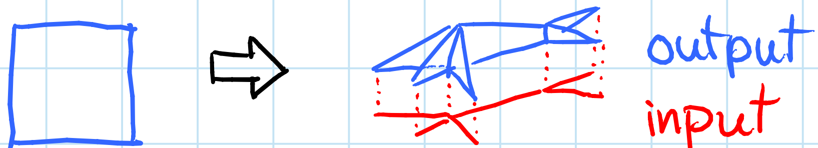
6.849	Lecture 4	Sept. 20, 2010

Tree method: [Lang 1994–2003; Lang & Demaine 2004–]
algorithm to find folding of smallest square into "uniaxial" origami base whose projection is a desired metric tree



But: – optimization is difficult: exponential time, as hard as disk packing, but good heuristics
– non-self-intersection is only conjectured (we're working on it)

Uniaxial base:
1. in $z \geq 0$ half-space
2. intersection with $z = 0$ plane = projection onto that plane
3. partition of faces into *flaps*, each projecting to a line segment ($\Rightarrow$ all faces vertical)
4. *hinge crease* shared by two flaps projects to a point: common endpoint of flap projections
5. graph of flap projections as edges, connected when flaps share a hinge crease, is a tree (*shadow tree*). Hinge creases projecting to a vertex form a *hinge*
6. only one point of paper folds to each leaf

Tree method: (cont'd)

Key lemma: in any uniaxial base from convex paper,
distance between two points on shadow tree
$\le$ distance between corresponding points on paper

Proof: latter = length of line segment on paper
- folds to path in uniaxial
- projects to shorter path on shadow tree
- shortest path in tree is only shorter $\square$

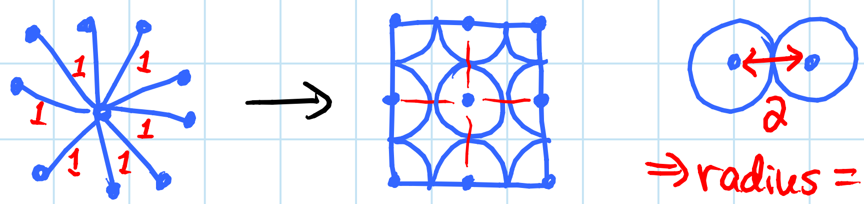
Scale optimization: focus on shadow leaves i
& placement as points p_i on paper:

λ → scale factor for tree

$$\begin{cases} \text{maximize } \lambda \\ \text{subject to } \underbrace{d(p_i, p_j)}_{\text{distance on paper}} \geq \lambda \cdot \underbrace{d(i,j)}_{\text{fixed distance in tree}} \text{ for leaves } i,j \end{cases}$$

— quadratic constraint

Example:



star → disk packing, centers in square

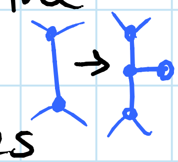
⇒ with $2n \times 2n$ piece of paper, get $(n+1)^2$ arms in star; can flatten to perimeter $\Theta(n^2)$

→ MARGULIS NAPKIN PROBLEM [Lang 2003]

Tree method (cont'd)

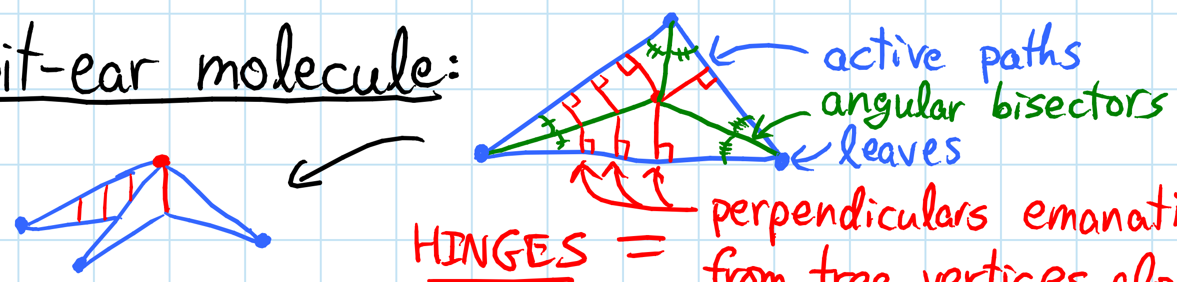
Active path = path between two shadow leaves of length = distance in piece of paper
- never cross each other [GFALOP Lem. 16.4.2]

Triangulation: can add artificial leaf edges to the shadow tree to make the active paths partition the piece of paper into triangles (without changing scale factor) [GFALOP Lem. 16.6.2]



- later these leaf edges can be "folded away"
- some triangle edges are paper boundary, not active

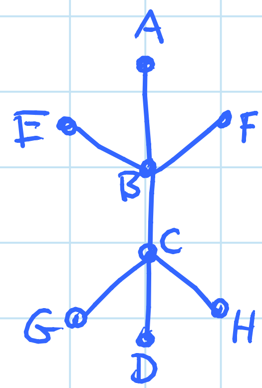
Rabbit-ear molecule:

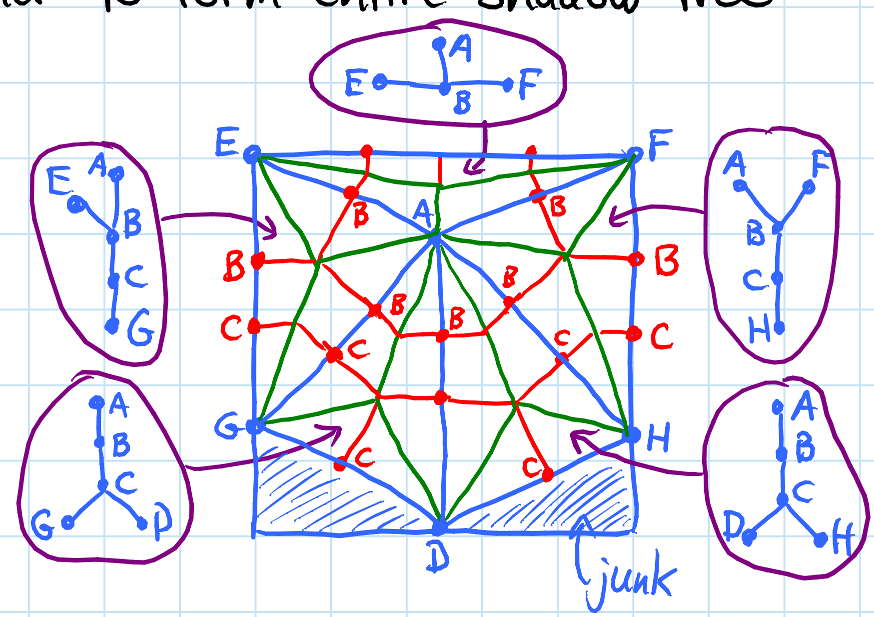


HINGES = perpendiculars emanating from tree vertices along path

- put them together to form entire shadow tree

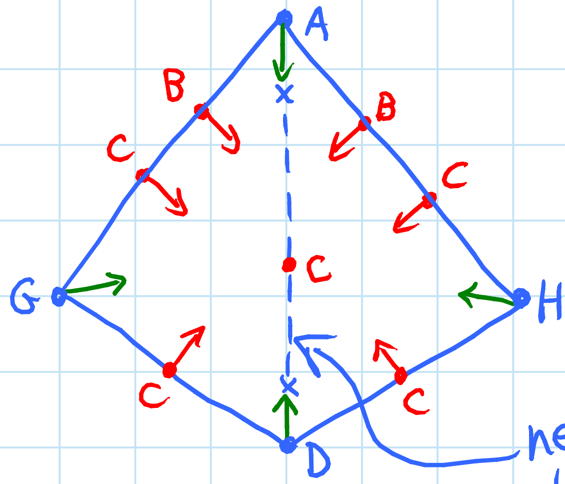
Example:





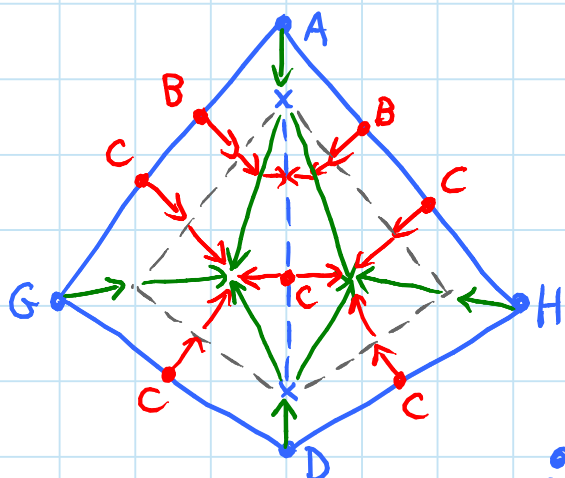
More practically:

- use convex decomposition instead of triangulation (in practice by letting tree edge lengths vary a bit)
- Lang Universal Molecule folds convex polygon

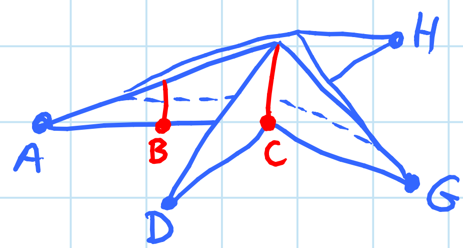


angular bisectors
perpendiculars

GUSSET:
new active path at some point



angular bisectors
perpendiculars



2 kinds of events:

1. gusset: new active path → split shrunken polygon
2. two vertices meeting → continue along new angular bisector

Cube wrapping: [Catalano-Johnson, Loeb, Beebee - Monthly 2001]

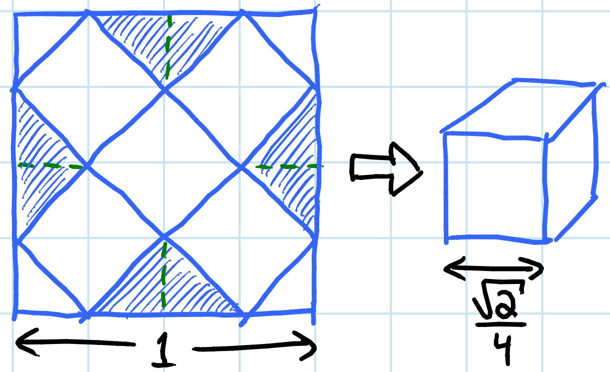
- consider 1×1 square
- in $x \times x \times x$ cube, every point has an antipodal point $\geq 2x$ away

$\Rightarrow$ center of square must be $\geq 2x$ away from corner (points only get closer by folding)

$\Rightarrow$ opposite corners have distance $\geq 4x$

$\Rightarrow$ side length $\geq 2\sqrt{2}\,x$

$\Rightarrow x \leq 1/2\sqrt{2} = \sqrt{2}/4$ & this is possible



$\boxed{\text{OPEN}}$: optimal square $\to$ regular tetrahedron? & other simple shapes [Grabarchuk 2005]

$\boxed{\text{OPEN}}$: $x \times y$ rectangle $\to$ largest cube?
- strip method efficient as $x/y \to \infty$

Checkerboard folding: square bicolor paper $\to n \times n$
- standard approaches route paper boundary along color reversals $\Rightarrow$ perimeter $\geq 2n^2$
- new approach achieves perimeter $n^2 + O(n)$ [Demaine, Demaine, Konjevod, Lang 2009]
  - visit squares instead of square boundaries
  - seamless too
  - win for $n > 16$; even 8×8 better for seamless
- $\boxed{\text{OPEN}}$: optimal? any nontrivial lower bound?

Origamizer: [Tachi 2006; Demaine & Tachi 2010]
a practical algorithm to fold any polyhedron
↳ efficient in practice (formalism?)

Seamless: each convex face of polyhedron is a polygon of paper on the surface
(possible by strip method too [L3])

A little extra: in addition to polyhedron, fold tiny flaps attached to edges & vertices
- within ε of polyhedron (Hausdorff dist.)

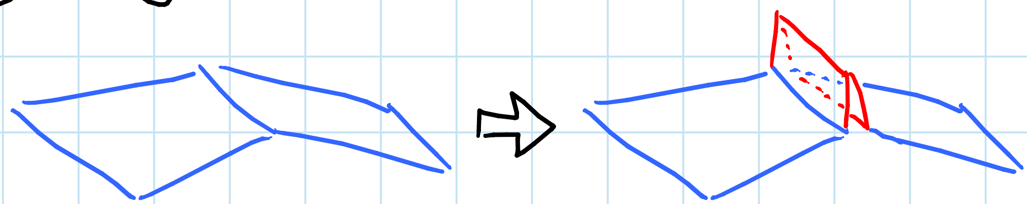
Watertight: boundary of paper maps to within ε of boundary of polyhedron (Haus.)
- here assume cut up so polyhedron is topologically a disk (homeomorphic)

⇒ traveling from one side of the disk to the other, while remaining in contact with the folding, requires coming within ε of disk boundary

Idea: place polyhedron faces on paper
fold ("tuck") away rest of paper
optimize scale factor
↳ hard, use heuristics, and even not clearly optimal

Tuck proxy:

- attach narrow tab to each polyhedron edge along angular bisector, all same side



- join & add tabs at vertices so that:
  - all on one side of disk (no intersection)
  - all dihedral angles convex
  - all faces convex
  - $\leq 360°$ of material at each vertex
  - still topological disk

Embedding:

- place polyhedron faces arbitrarily, scaled down
- connect corresponding edges with "tunnels"
- connect corresp. vertices according to dual of tuck proxy

Tuck gadgets:

- pleat tunnels to fit in narrow tab
- add dots to fill region around vertex, taking care at dual vertices from
- fold Voronoi diagram
- crimp angles down to match tuck proxy