Multiple Clothing Part Placement: Direct Representation of Curves vs. Polygonal Approximation

Victor J. Milenkovic

Department of Computer Science
University of Miami

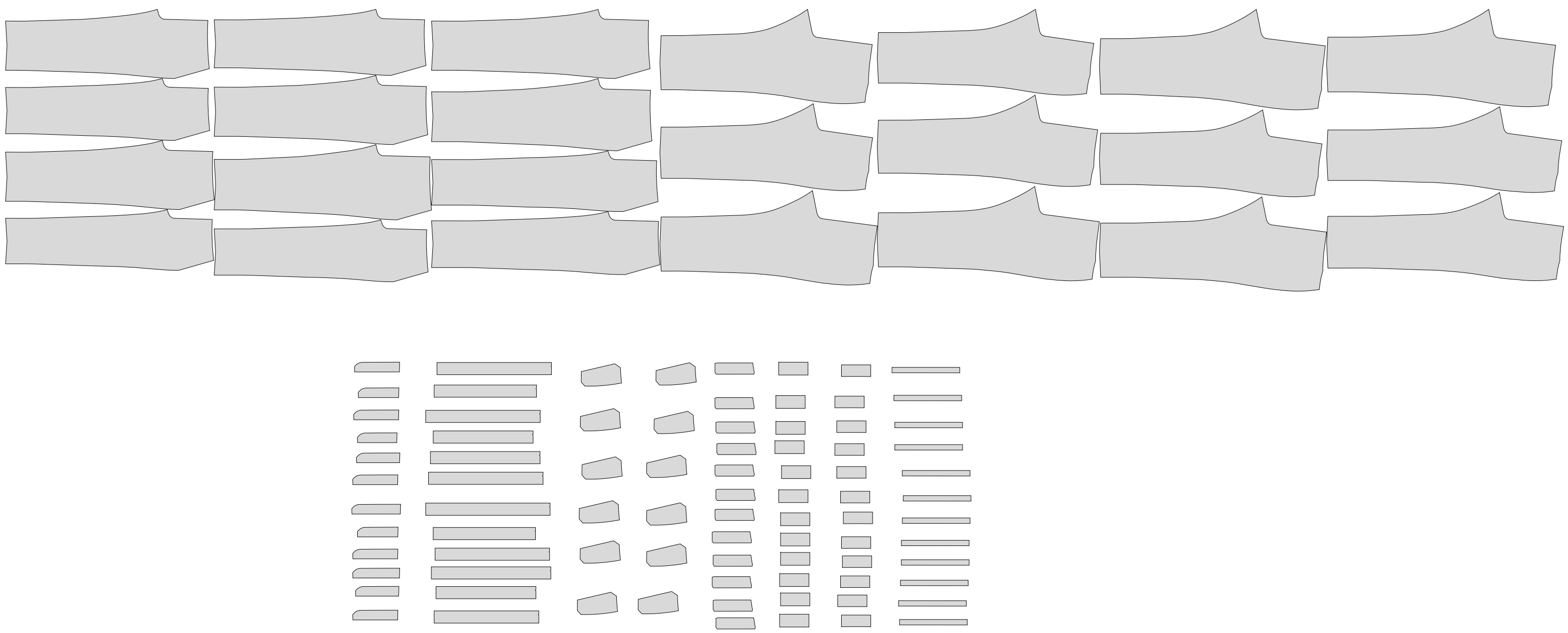
Joint work with Elisha Sacks, Purdue University.

Supported by NSF-CCR-0304955 and 0306214.

Old Research Project: LAYOUT FOR CLOTHING INDUSTRY

For a number of years I have been working on the following research problem:

Given a set of parts...

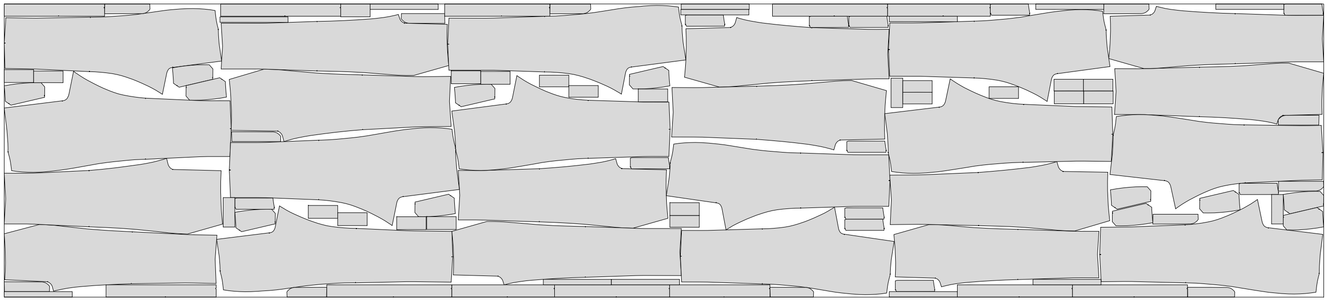


Old Research Project: LAYOUT

And a sheet of cloth...

Old Research Project: LAYOUT

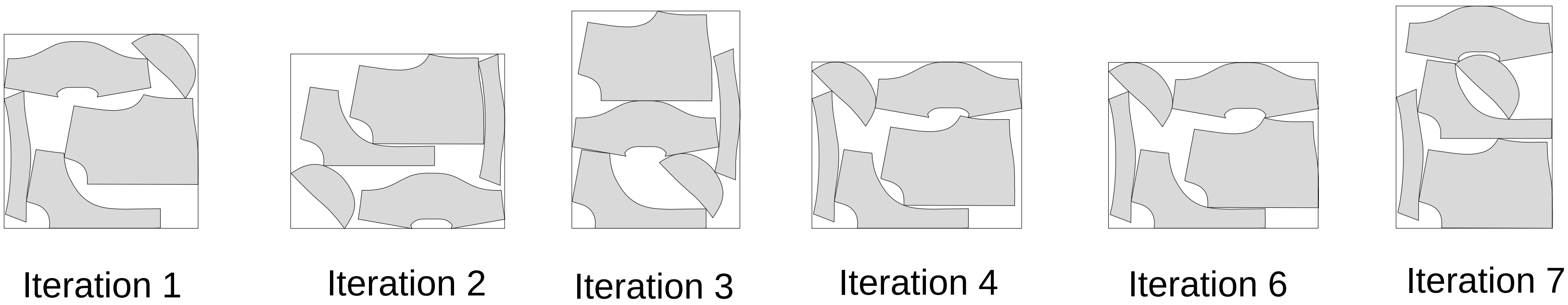
Figure out if one can place those parts into that piece of cloth.



Name: 37457c
Width: 59.75 in
Length: 269.04 in
Pieces: 108
Efficiency: 89.54%

Translational Containment & Minimum Enclosure

Joint Work with Karen Daniels (Ph.D. Harvard '95)



Minimal enclosing rectangle (within 0.01% of optimal) of five polygons with 55,61,66,65, and 72 vertices. The algorithm converges in seven iterations. Each feasible iteration corresponds to a smaller area. Iterations 5 and 8 (not shown) were infeasible: target area chosen smaller than optimum.

Limitations!

1. Translation, not rotation.

2. Polygons, not (poly) curves.

New Goals: Curves and Rotation

Joint work with Elisha Sacks (Purdue University).

1. New algorithms: semi-output-sensitive construction of 3D configuration spaces.

2. New approaches to geometric robustness: inconsistency-sensitive construction of arrangements in 2D and 3D.

3. New forms of approximation to keep the algebraic degree from blowing up too.

THIS TALK

Circular polygons: regions bounded by line segments and circular arcs.

Still only translation.

Needs iterated intersection and Minkowski sum of circular polygons.

Review: Multiple Polygon Placement

Also known as CONTAINMENT and MINIMUM ENCLOSURE.

Joint work with Karen Daniels.

Builds on work by Avnaim and Boissonnat, Chazelle, Devillers, and others, including, of course, Minkowski.

Minkowski Sum

review

Intersection: $A \cap B = \{p \,|\, p \in A \text{ and } p \in B\}$.

Translation: $A + t = \{a + t \,|\, a \in A\}$.

Opposite: $-A = \{-a \,|\, a \in A\}$.

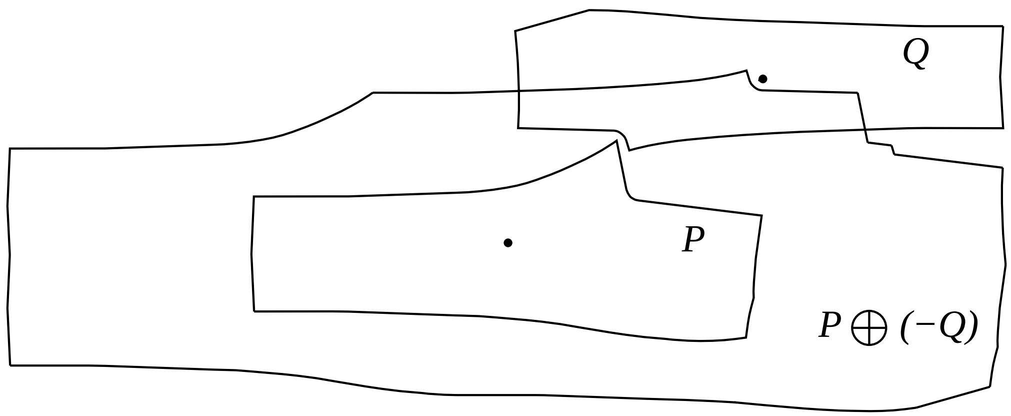
Minkowski Sum: $A \oplus B = \{a + b \,|\, a \in A \text{ and } b \in B\}$.

Claim: $A + s$ overlaps $B + t$ if and only if $t - s \in A \oplus -B$.

Proof: $a + s = b + t$ if and only if $t - s = a - b$.

Minkowski Sum

Convert polygon/polygon problem to point/polygon problem.



Polygon Placement

review

Input: container C and parts $P_1, P_2, \ldots, P_n$.

Output: $t_1, \ldots, t_n$ such that $P + t_i \subset C$ and $P_i + t_i$ does not overlap $P_j + t_j$.

Define: $P_0 = \overline{C}$, $t_0 = (0,0)$, $U_{ij} = \overline{P_i \oplus -P_j}$.

Goal: $t_0 = (0,0), t_1, \ldots, t_n$ such that $t_j - t_i \in U_{ij}$.

Define: U^*_{ij} as the set of all $t_j - t_i$ that belong to a solution $t_0 = (0,0), t_1, \ldots, t_n$ to the goal.

Restriction

review

Claim: $U_{ik}^* \subseteq U_{ij}^* \oplus U_{jk}^*$.

Proof: $t_k - t_i = (t_j - t_i) + (t_k - t_j)$.

Restriction: replace U_{ik} by $U_{ik} \cap (U_{ij} \oplus U_{jk})$.

Invariant: $U_{ij}^* \subseteq U_{ij}$.

Repeat restriction for all i, j, k until some U_{ij} becomes null or the area stops diminishing much (by more than 1%?).

Null means no solution.

Restriction is POWERFUL

Avnaim and Boissonnat's result can be rephrased as: for $n \leq 3$ and rectangular C, restriction generates U^*_{ij} in one iteration.

Daniels and Milenkovic: add evaluation and subdivision. Practical solution for $n = 5$, maybe more.

Evaluation

review

Evaluation is attempt to find solution in non-null set of U_{ij}.

Pick t_i in smallest U_{0i}.
Replace U_{0i} by $\{t_i\}$. Restrict. If not null, recurse.

If null, undo replacement (blind alley).

Subdivision

review

Pick some U_{ij} and partition it into U'_{ij} and U''_{ij}.

(Pick the smallest one with multiple components and partition based on components. In none have components, cut with some dividing line.)

Recurse on sub-problems with ONE U_{ij} replaced by either U'_{ij} or U''_{ij}.

Robustness Issues

$U_{ij} \leftarrow U_{ij} \cap (U_{ik} \oplus U_{kj})$

Restriction is iterated intersection and Minkowski sum of planar regions.

VERY ROUGH on the numerics.

Probably exponential growth in bit-complexity for any exact approach.

Robust Polygon Operations

Robust operations on polygons is a solved problem.

Rounding on integer grid: snap rounding (Greene, Guibas, and others).

Rounding on floating point grid: shortest path rounding (Milenkovic).

Purely floating point: nearest pair rounding (Milenkovic).

Robust Curve Operations

"Controlled perturbation" of circles: Halperin and Leiserowitz.

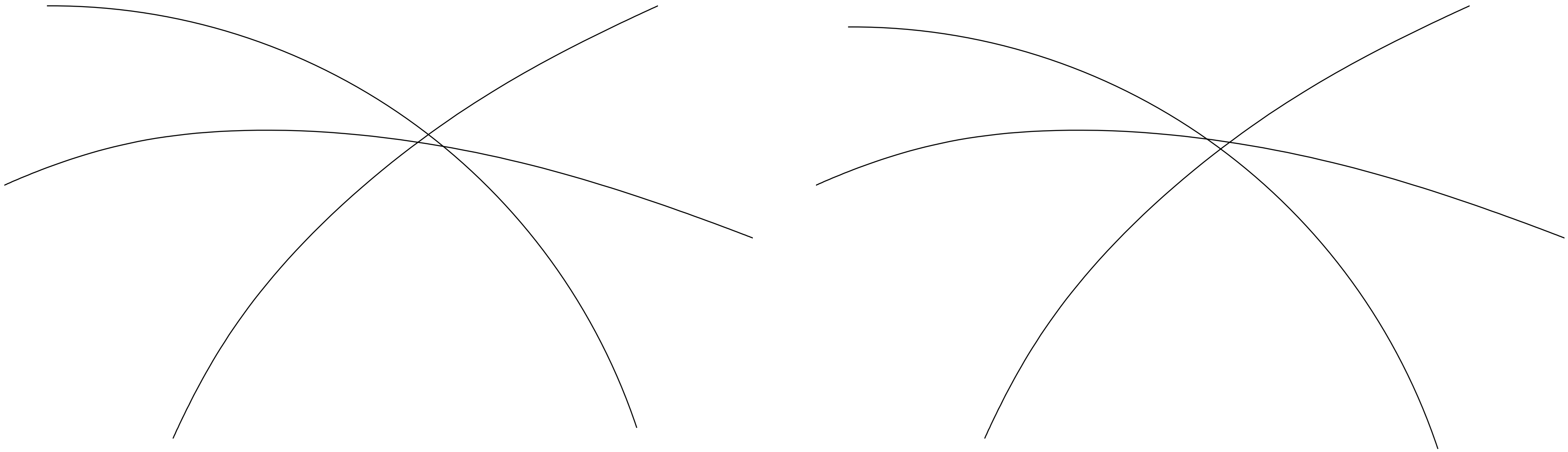
1. Deliberately AVOIDS degeneracy, which is not necessarily a good thing.

2. Doesn't necessarily keep things water-tight.

Inconsistency-Sensitive arrangement algorithm: Milenkovic and Sacks.

"If it ain't broke, don't fix it."

Tiny Triangles



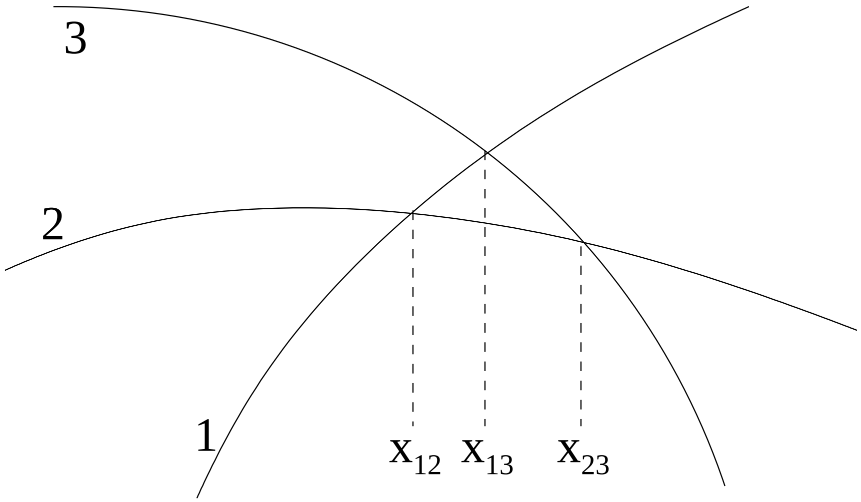
If you have a very tiny triangle:

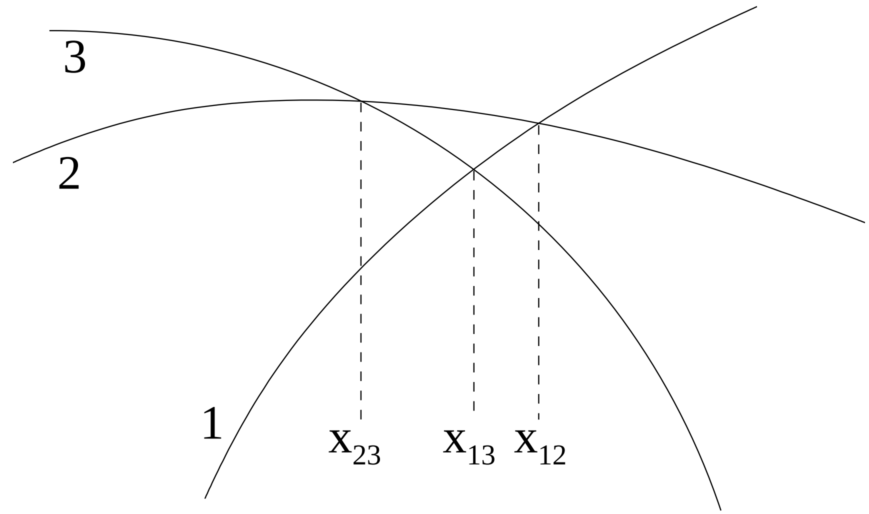
1. Floating point could be wrong.

2. Exact arithmetic might require a lot of bits and time and space.

But who cares if you are wrong?

The real problem: INCONSISTENCY





Either $x_{12} < x_{13} < x_{23}$ or $x_{23} < x_{13} < x_{12}$.

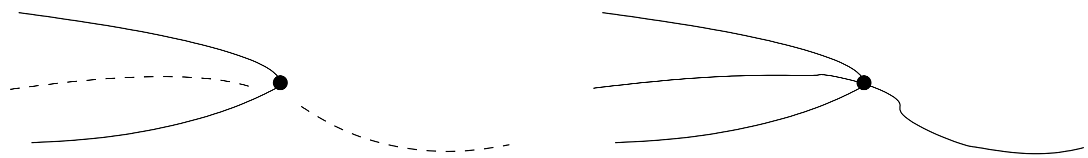
$\leq$ is o.k. (just another kind of degeneracy).

Any of the four orders is INCONSISTENT: $x_{13} < x_{12} < x_{23}$, etc.

UNAVOIDABLE INCONSISTENCY is VERY RARE and CHEAPLY REPAIRED.

Pinching

If a segment is caught between segments with a common endpoint,

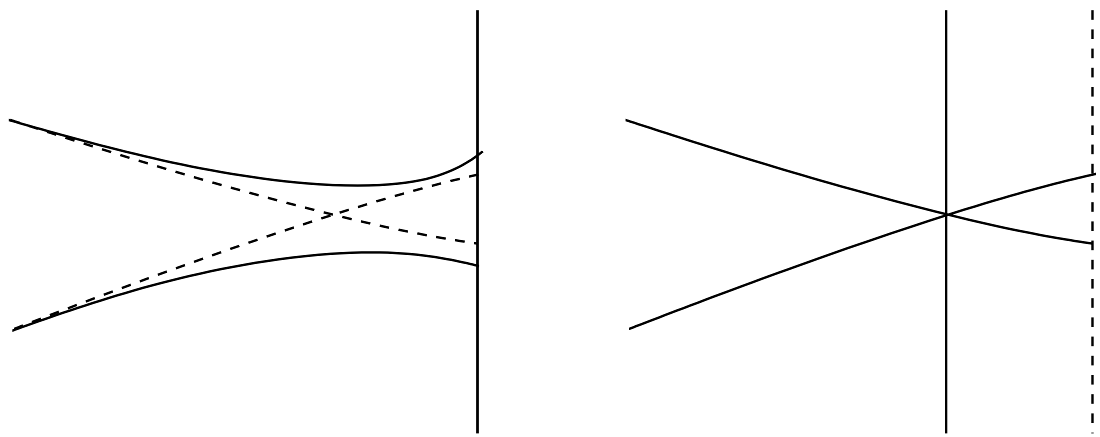


cut it at that vertex.

Missing Intersection

inconsistency-sensitive algorithm

If two segments see each other, and out of order because the sweep missed an intersection,

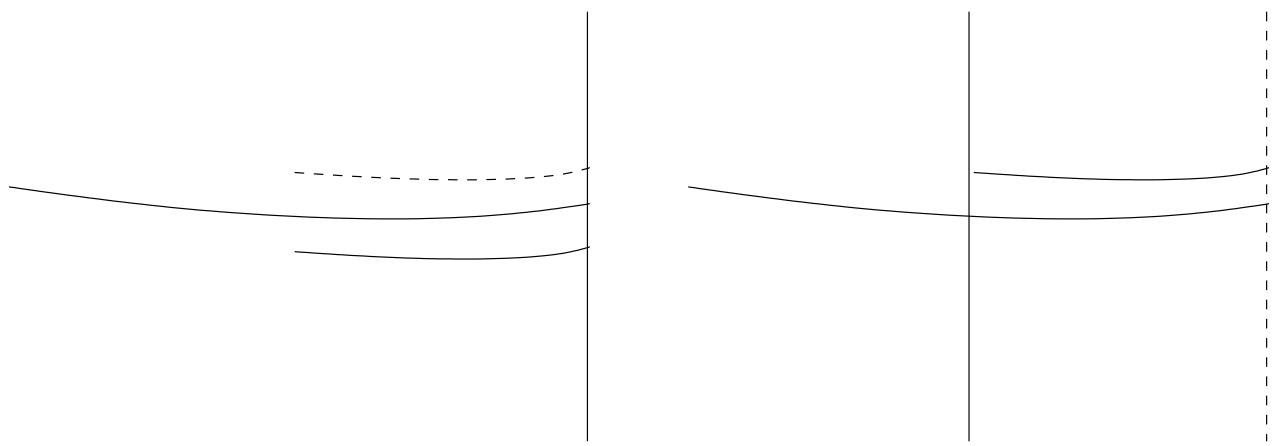


roll back the sweep to the intersection.

Wrong Initial Order

inconsistency-sensitive algorithm

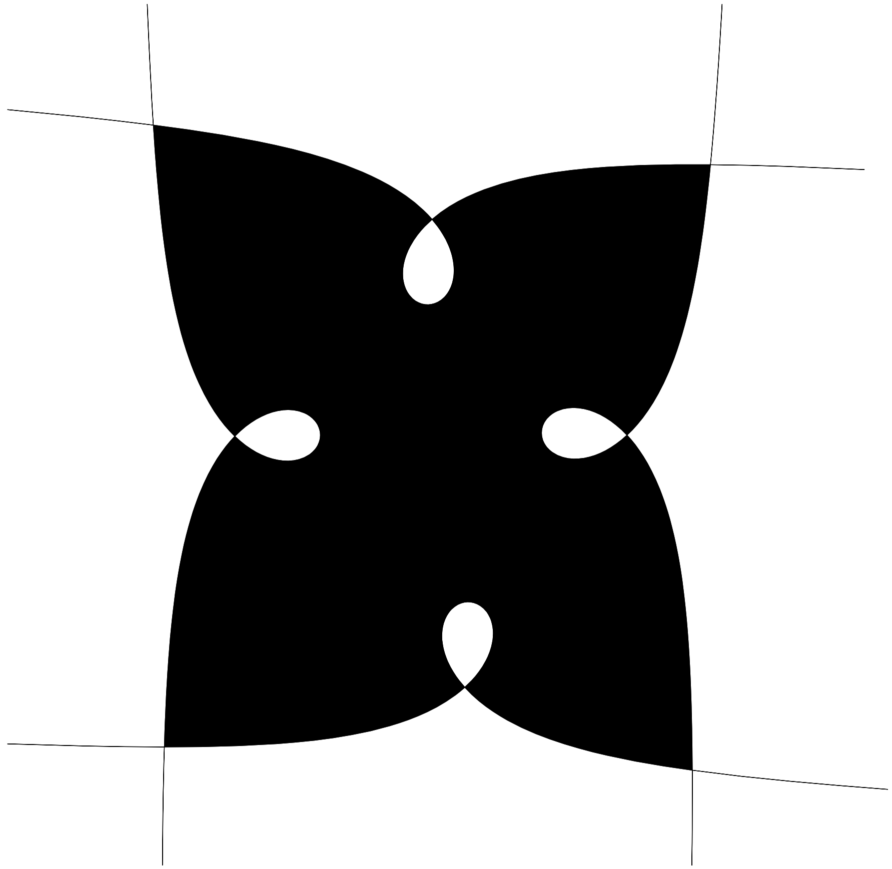
If two segments see each other, and out of order because the most recent insertion was in the wrong order,



roll back to the most recent insertion.

Experiment 1

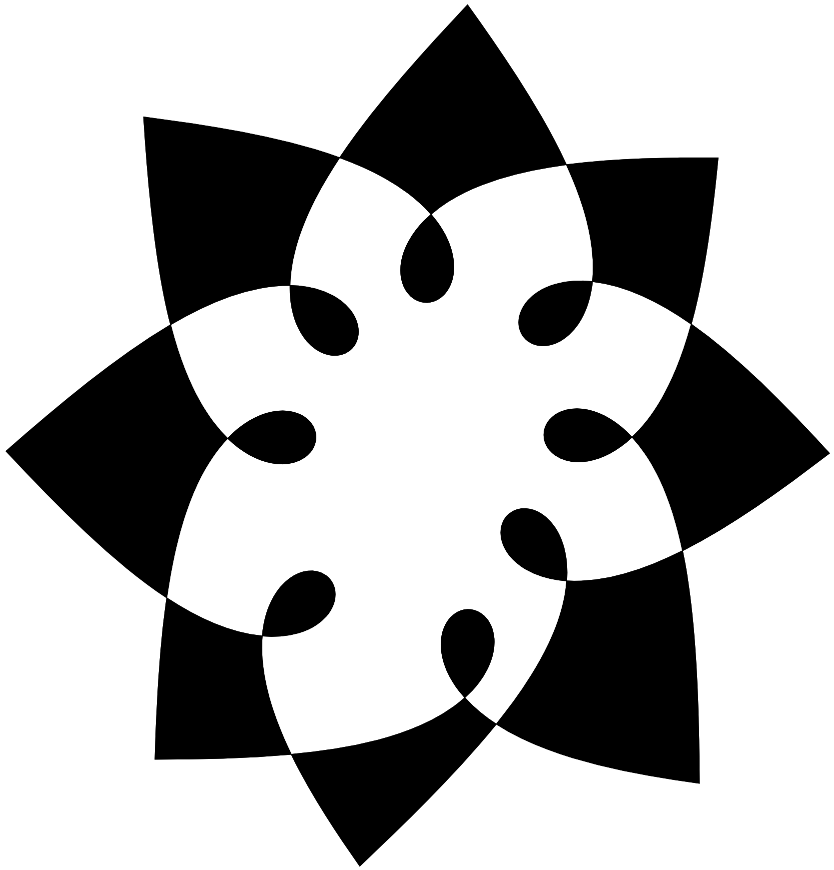
inconsistency-sensitive algorithm



Start with a "square" make of cubic curves:

$$-y^2 + xy^2 + x^2 + x^3 = 0.$$

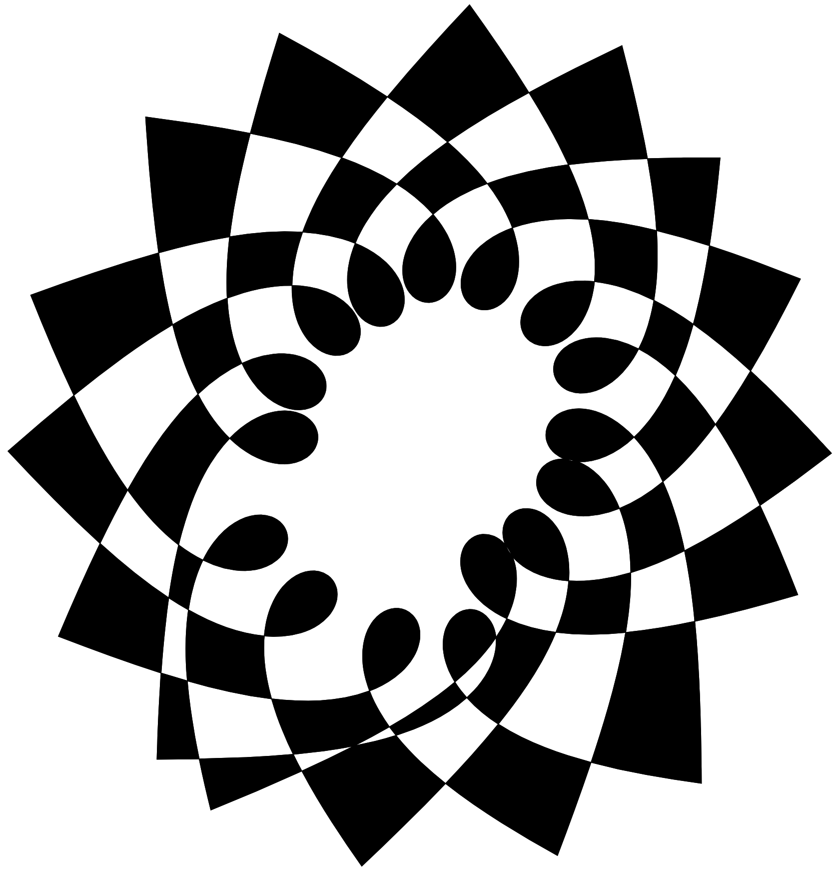
Experiment 1



Rotate by 47 degrees and XOR with itself.

Experiment 1

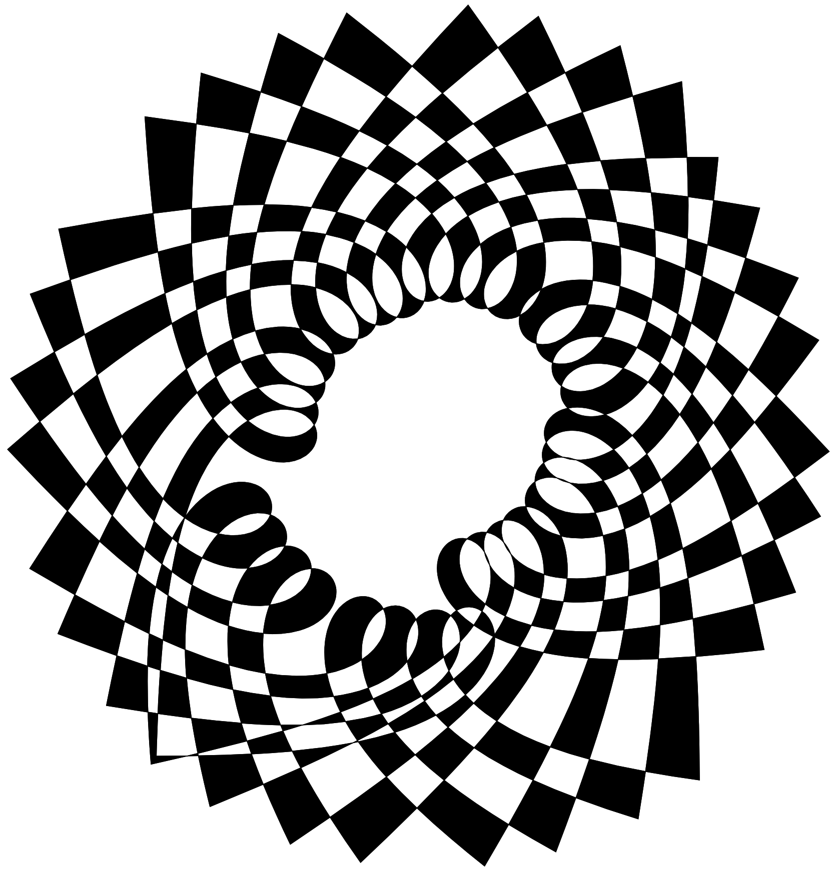
inconsistency-sensitive algorithm



Rotate by 47/2 degrees and XOR with itself.

Experiment 1

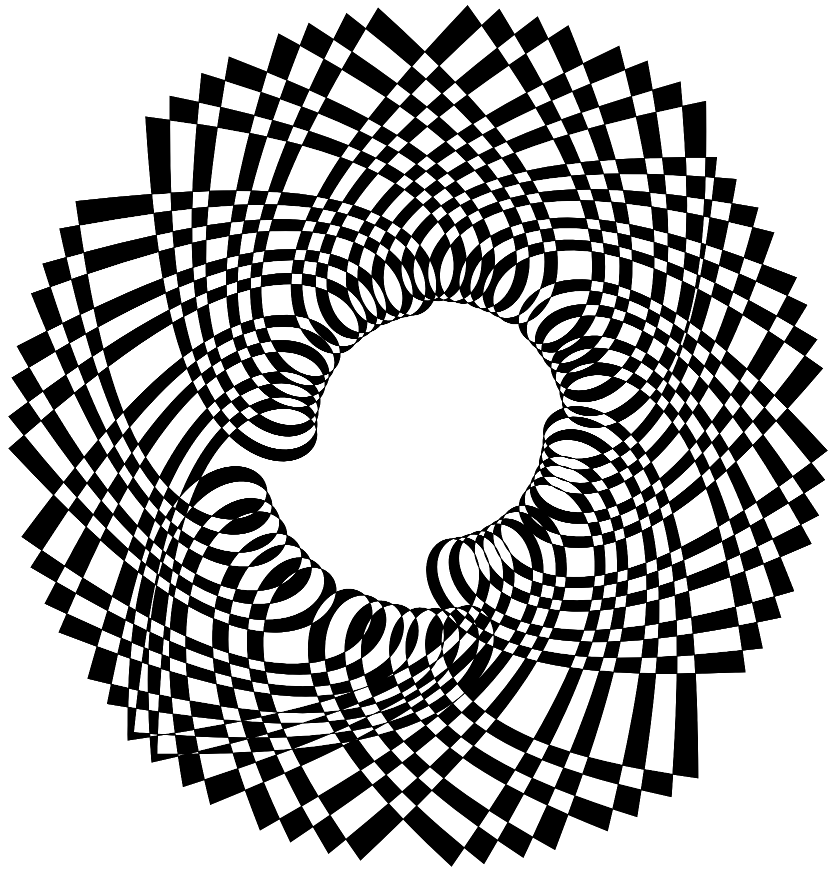
inconsistency-sensitive algorithm



Rotate by 47/4 degrees and XOR with itself.

Experiment 1

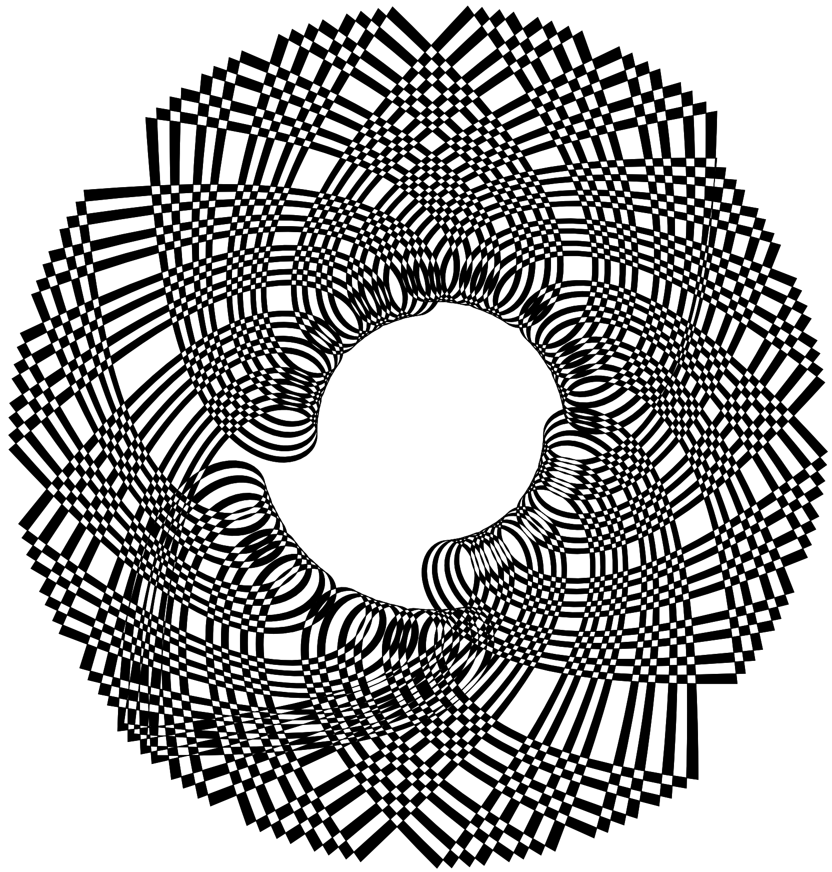
inconsistency-sensitive algorithm



Rotate by 47/8 degrees and XOR with itself.

Experiment 1

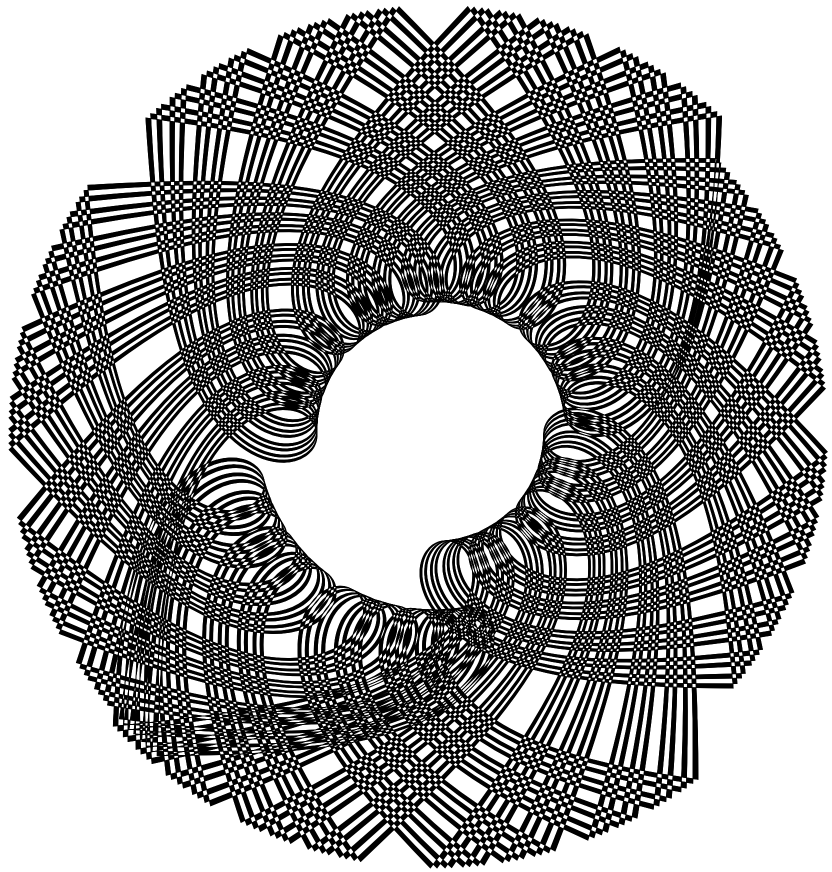
inconsistency-sensitive algorithm



Rotate by 47/16 degrees and XOR with itself.

Experiment 1

inconsistency-sensitive algorithm



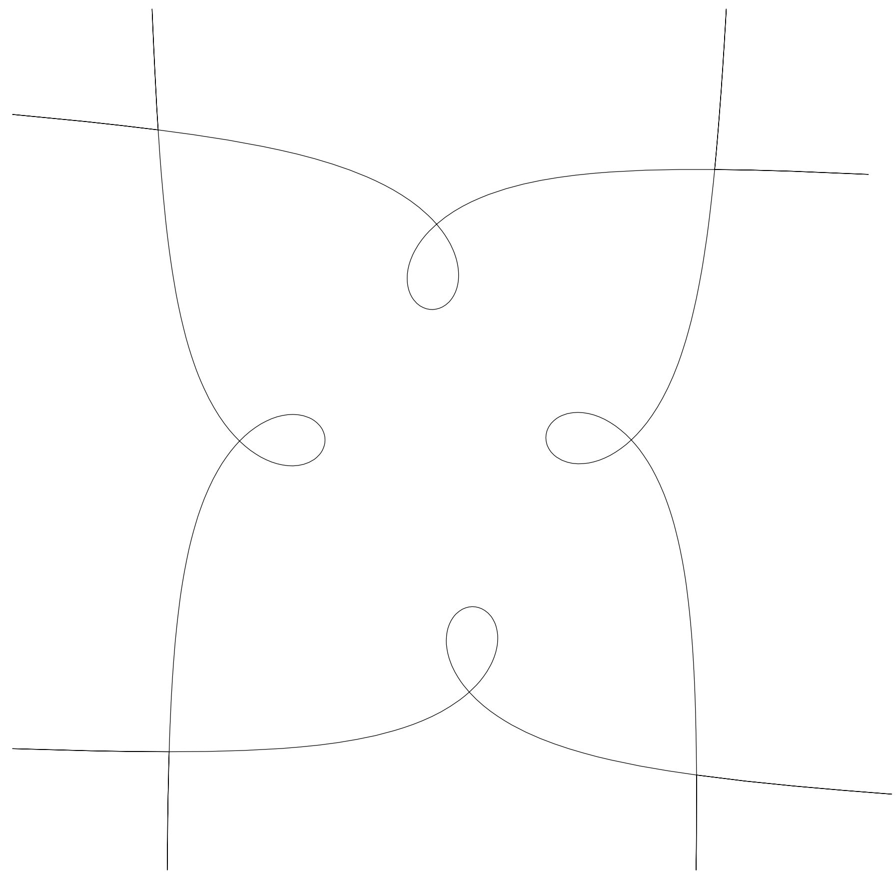
Rotate by 47/32 degrees and XOR with itself.

Number of vertices = 23971, edges = 47185, and cells = 23216.

NO INCONSISTENCIES.

Experiment 2

inconsistency-sensitive algorithm



Same, except 0.000001 degree instead of 47 degree.
Number of vertices = 11878, edges = 45817, and cells = 33949.
INCONSISTENCIES: 2 incorrect insertions.
Rollback: $59 + 20 = 79$ events.

Curve Arrangements: Who cares?

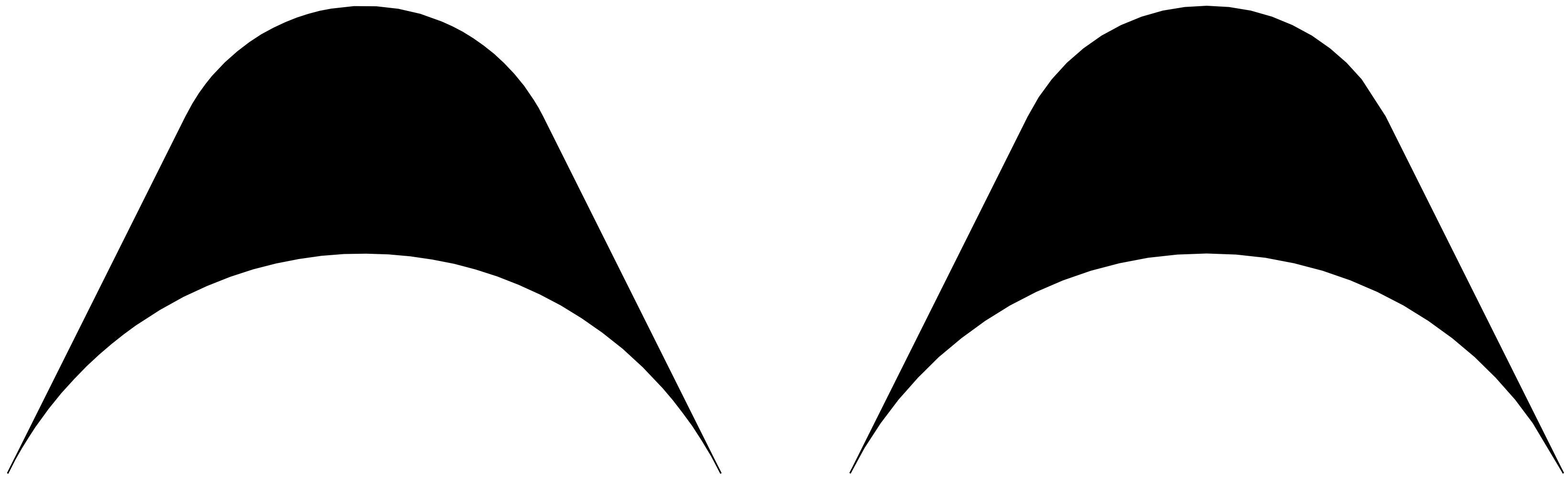
1. Most clothing part representations are greatly simplified by the introduction of circular arcs.

2. Ditto mechanical parts.

3. Sanity check: are the new techniques faster than just using polygons with a lot of small edges?

Curved Parts

sanity check



LEFT: part made of two circular arcs and two line segments.

RIGHT: polygonal approximation. Top arc has 20 points. Bottom arc has 30 points. This corresponds to typical numbers for actual parts.

Five orientations chosen to increase common lines (stress-test degeneracy code).

Packing

sanity check

SUCCESS: Packing of parts using restriction until no U_{ij} shrinks by 10%.

FAILURE: Make square 3% smaller.

Note: no subdivision, just restriction and evaluation.

Running Times

sanity check

SUCCESSFUL PACKING

	circular	polygonal
number of restrictions	556	556
running time seconds	29	68

FAILED PACKING

	circular	polygonal
number of restrictions	315	315
running time seconds	69	143

Number of inconsistencies: 1 (in failed circular packing)

Number of rolled back events: 2

Conclusions

Inconsistencies are indeed rare.

Rollbacks are few and of negligible cost.

Use of circular arcs is at least twice as fast as polygonal approximation.

Algorithms for curves are "worth it".

Future Work

The use of circles keeps the algebraic degree bounded.

(The Minkowski sum of two circles is a circle!)

Everything generalizes to rotational packing, except that the algebraic degree blows up!

Piano-movers problem was solved 25 years ago...in theory.
Robust arrangements in 6D?
Packing is a lot harder!