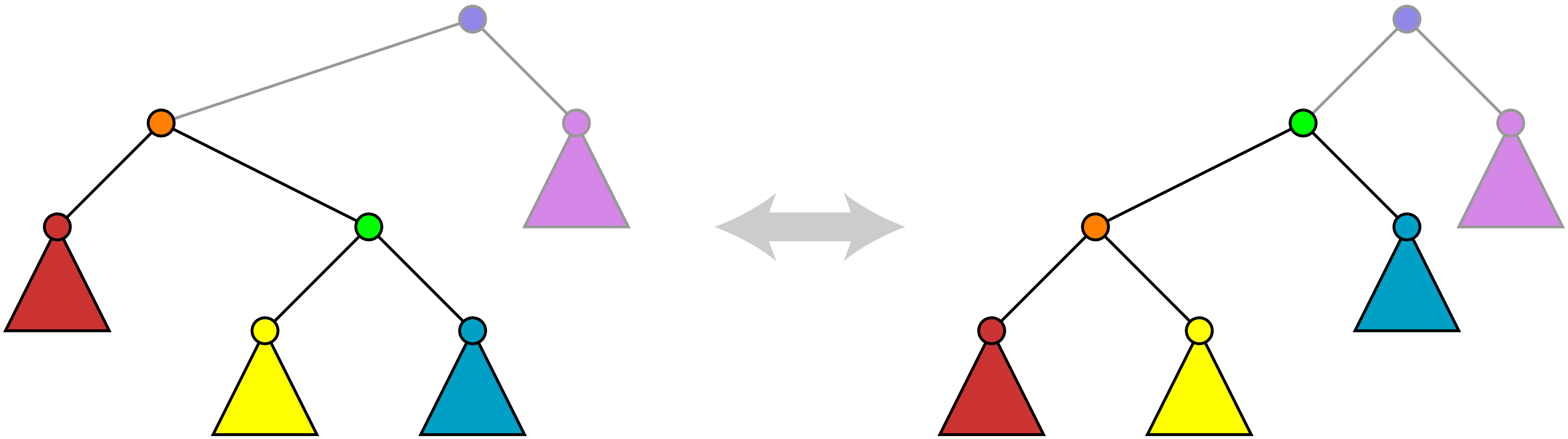
Happy Endings for Flip Graphs

David Eppstein
Univ. of California, Irvine
Computer Science Department

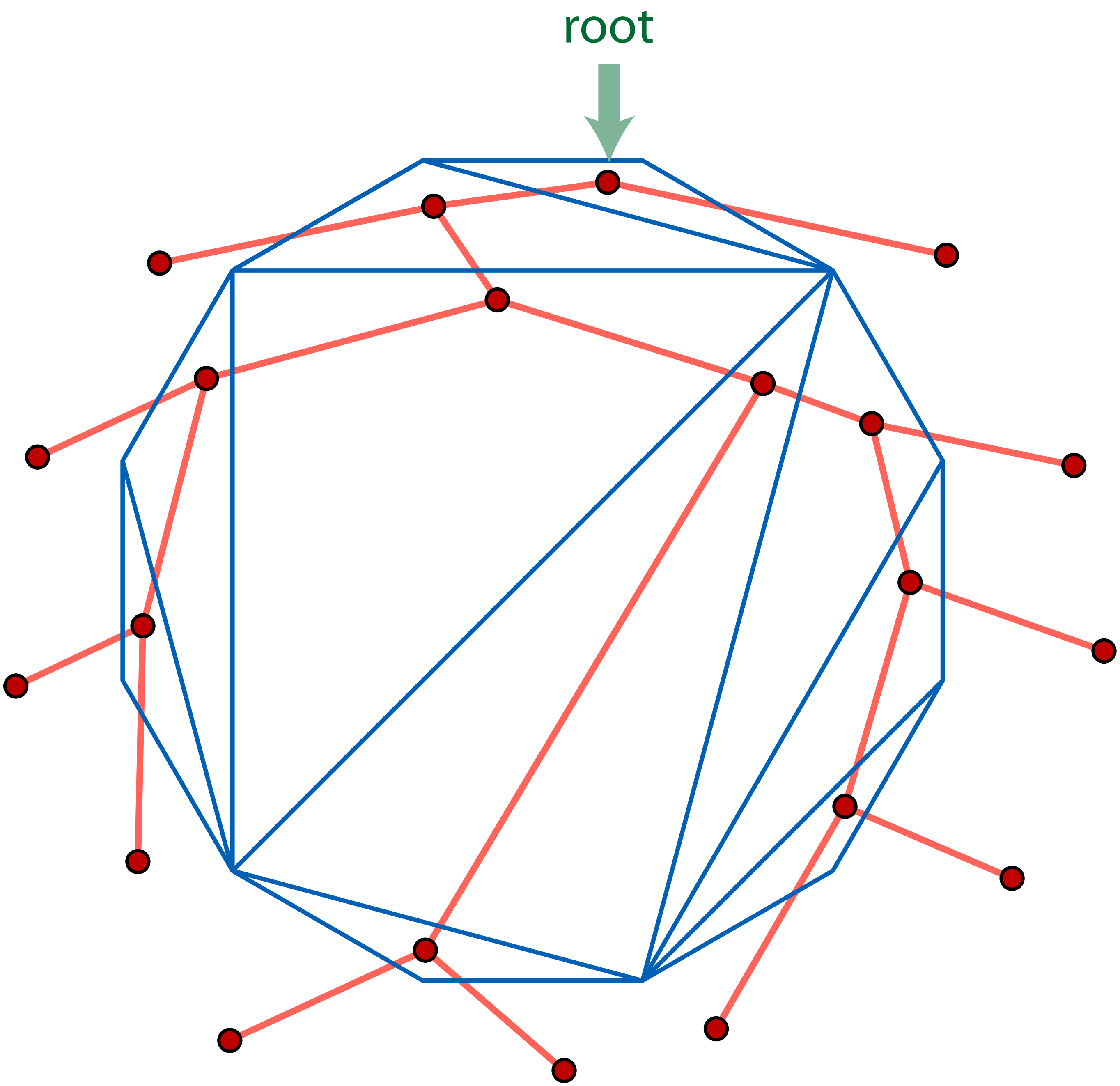
Rotation in binary trees



Rearrange links in and out of two adjacent nodes
while preserving in-order traversal ordering of all nodes

Fundamental to many balanced binary search tree data structures

Equivalence of binary trees and polygon triangulations



Tree with n leaves

⇔

polygon with n vertices

Root + leaves of tree

⇔

edges of polygon

Internal nodes of tree

⇔

triangles of triangulation

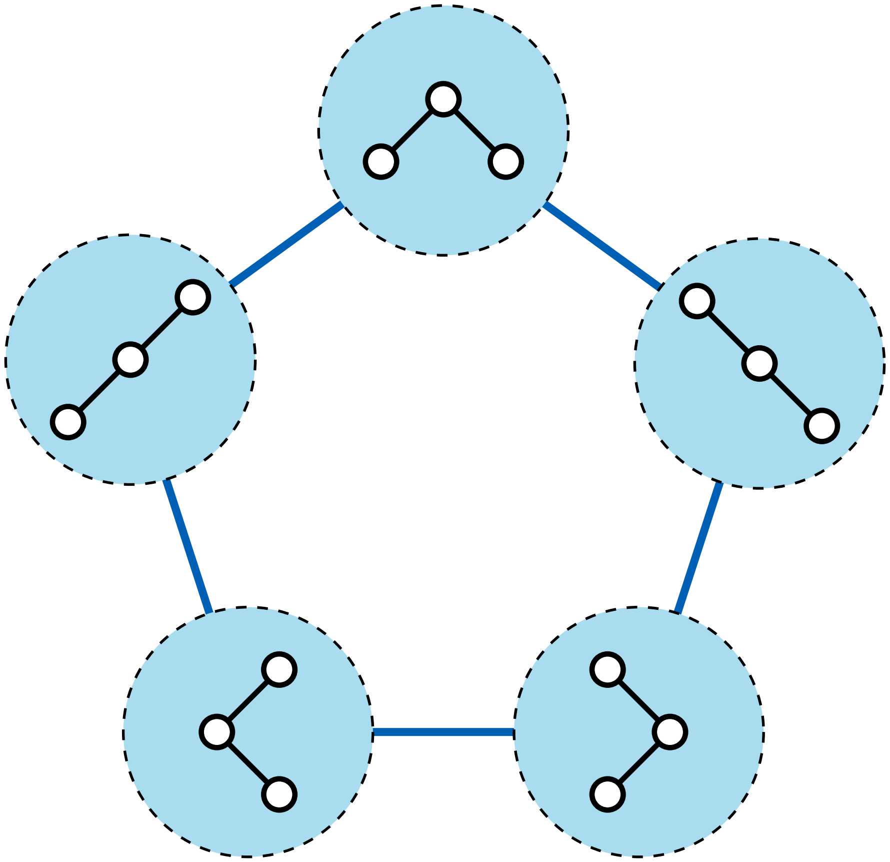
(a form of planar duality)

Rotation between trees

⇔

Flip between triangulations

Rotation graph and rotation distance

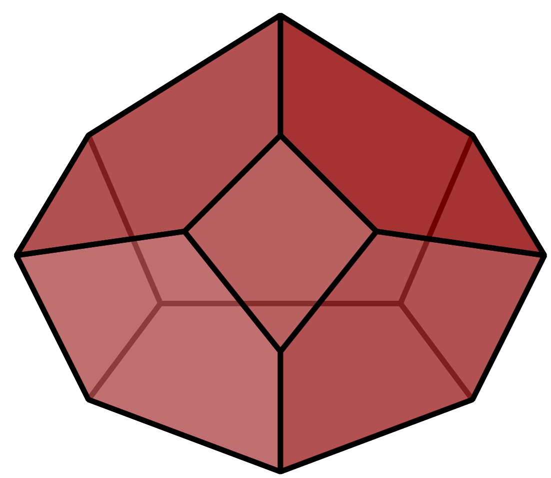


Vertices: n-node binary trees
Edges: rotations

Rotation distance:
minimum #rotations
to change one tree into another

(shortest path length in rotation graph)

Rotation graph is skeleton of
(n-1)-dimensional Stasheff polytope



Rotation distance and hyperbolic geometry

[Thurston, Sleator, Tarjan 1986]

Max distance among n-node trees is exactly 2n-6

- Use flipping of polygons instead of rotation of trees
- Form polyhedron with triangulations as its top and bottom
- Flip sequence = partition of polyhedron into tetrahedra
- Use hyperbolic geometry: all tetrahedra have volume $\leq \pi$
- Find polyhedra with large hyperbolic volume

Obvious open problem: how to compute flip distance?

Little progress since then...

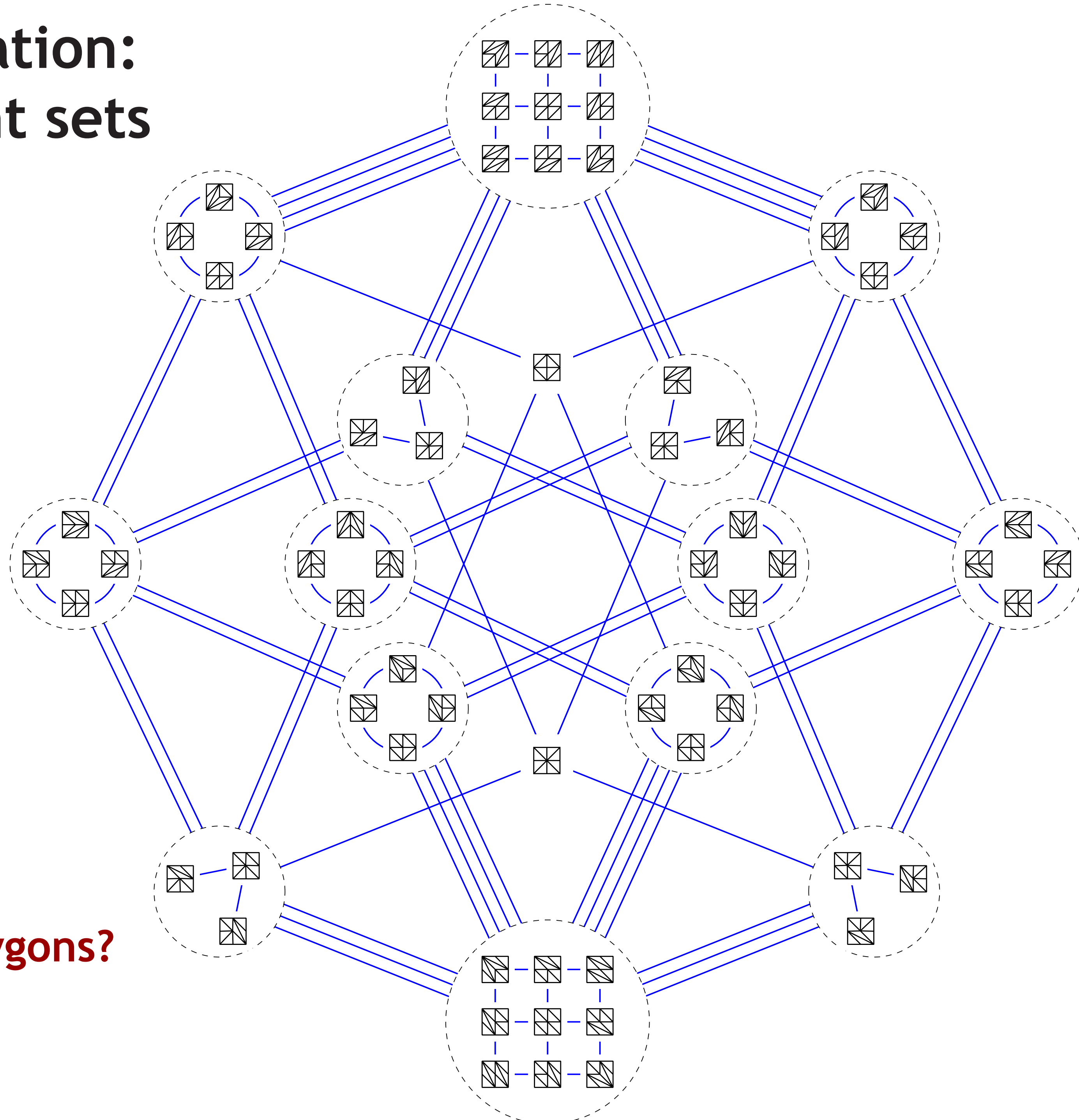
Natural generalization: flip graphs of point sets

(shown: all triangulations of a 9-point square grid)

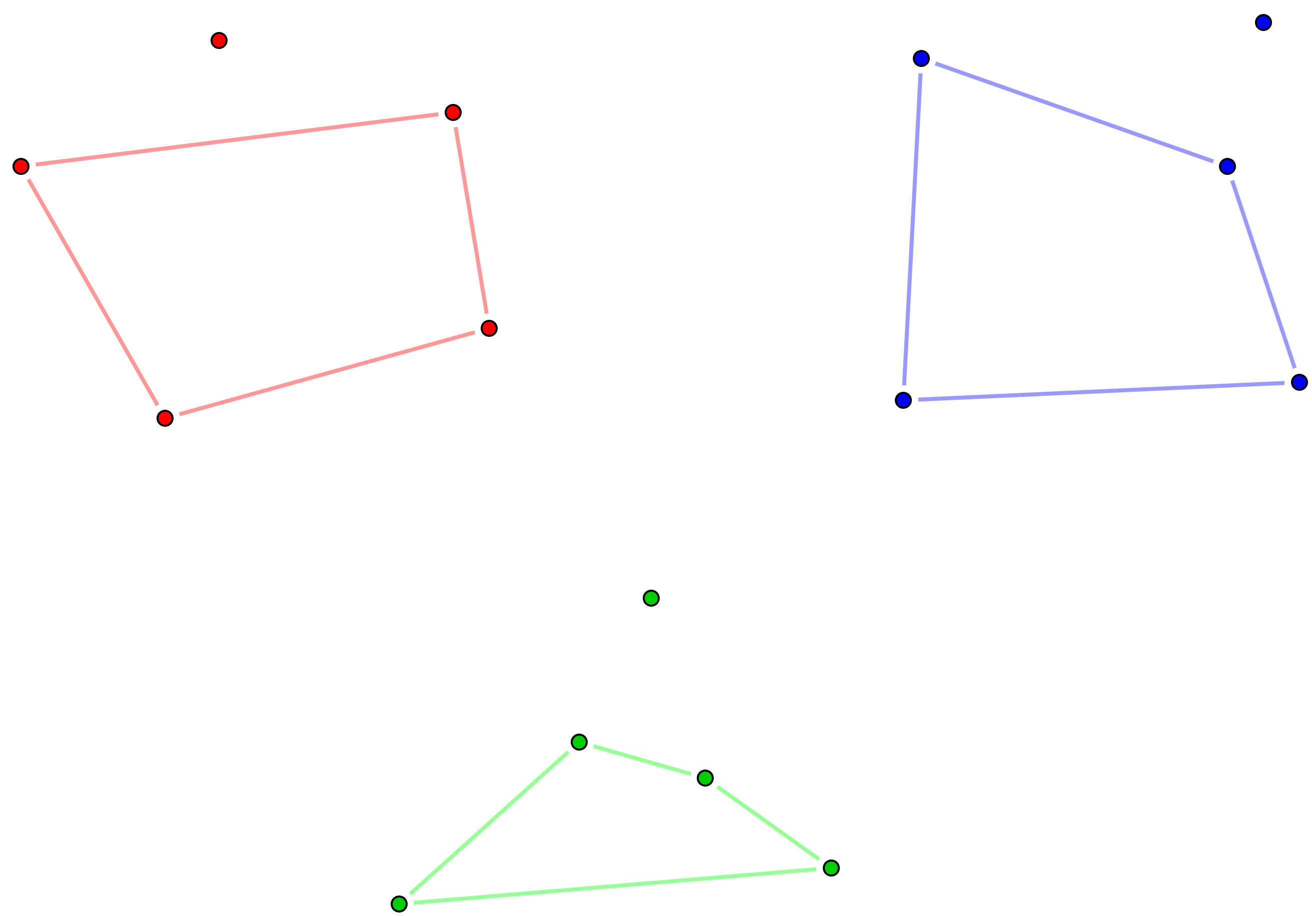
Flipping is important in mesh improvement

Flip distance can be quadratic [Lawson 1972]

More complex than polygons?



The Happy Ending theorem



Five points in general position have four forming a convex quadrilateral [E. Klein]

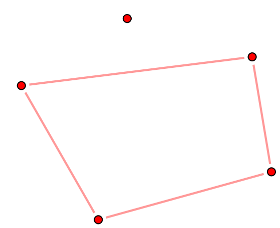
More generally, for any n, sufficiently many points (no three collinear) include the vertices of a convex n-gon

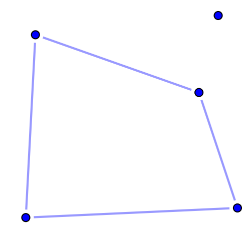
Erdős and Szekeres (1935); “happy ending” = Klein-Szekeres marriage

Empty convex polygons in point sets

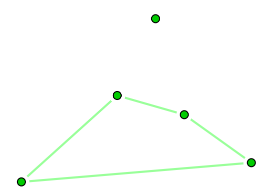
Five points in general position always contain an **empty** quadrilateral

Ten points in general position
always contain an empty convex pentagon
[Harborth 1978]





Sufficiently many points in general position
always contain an empty convex hexagon
[Gerken 2006; Nicolás 2006]

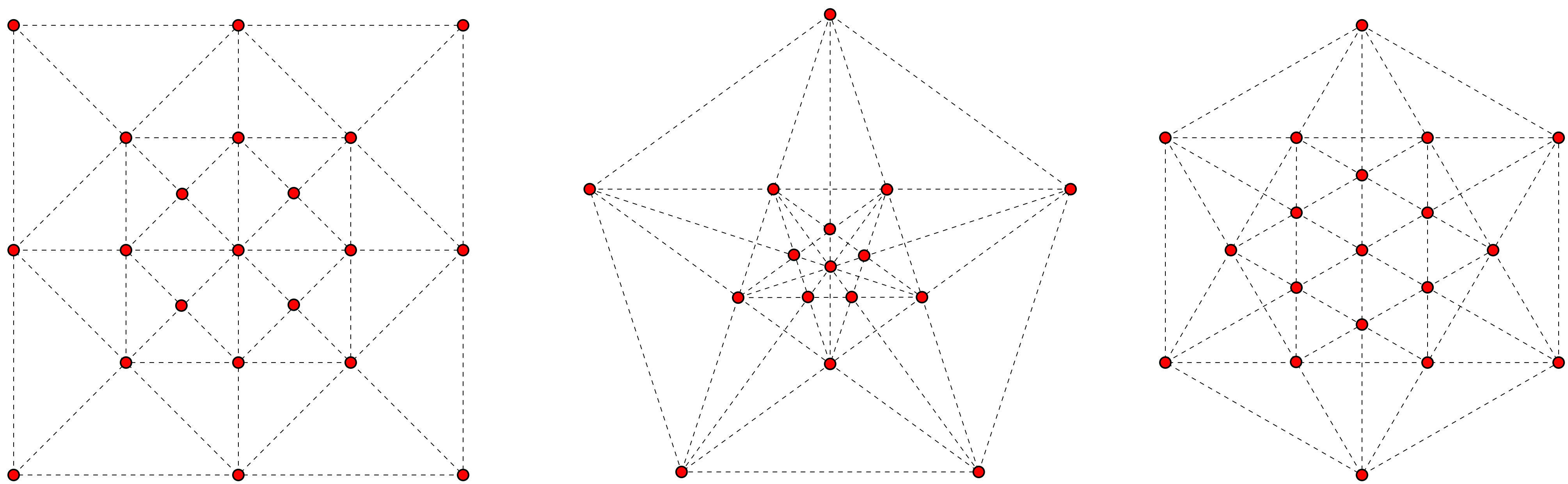


Arbitrarily many points in general position
do not always contain an empty convex heptagon
[Horton 1983]

But what if they're not in general position?

Point sets without empty convex quadrilaterals: highly constrained
(we'll see later: can describe them all simply)

Point sets without empty convex pentagons: many examples

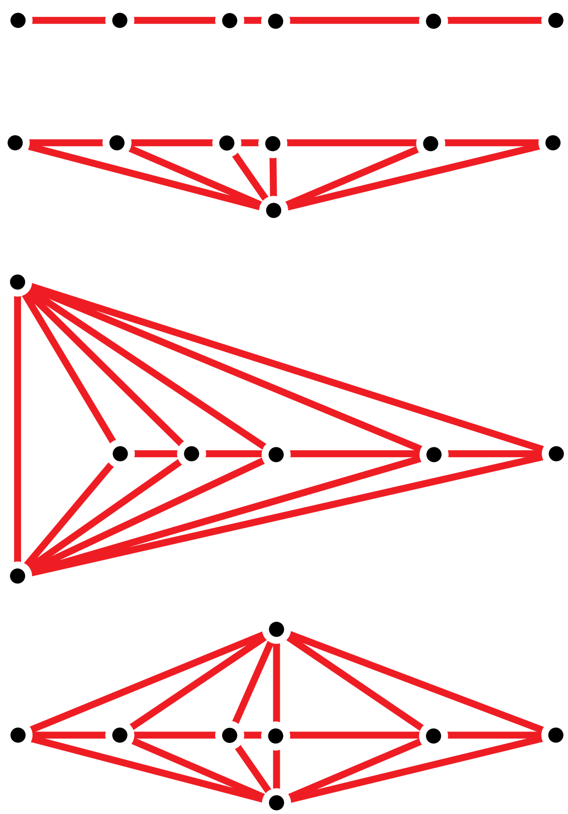


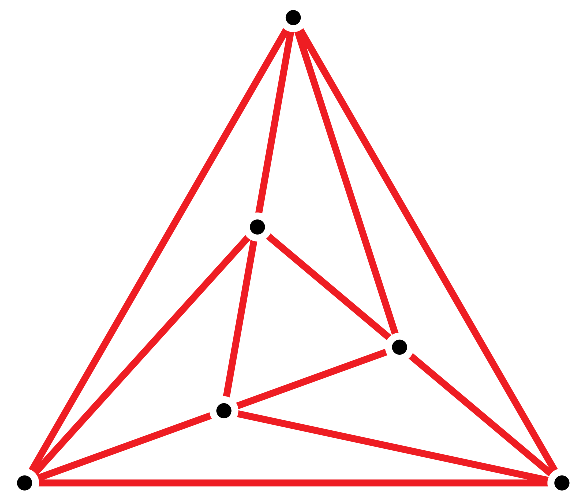
points on two lines, any convex subset of a lattice, ...

Point sets with no empty quadrilateral

= point sets with **exactly one triangulation**

= vertices of planar graphs connecting any two points by a straight path





Four infinite families
and one special case...

[E. 1997; Dujmovic, E., Suderman, Wood 2006]

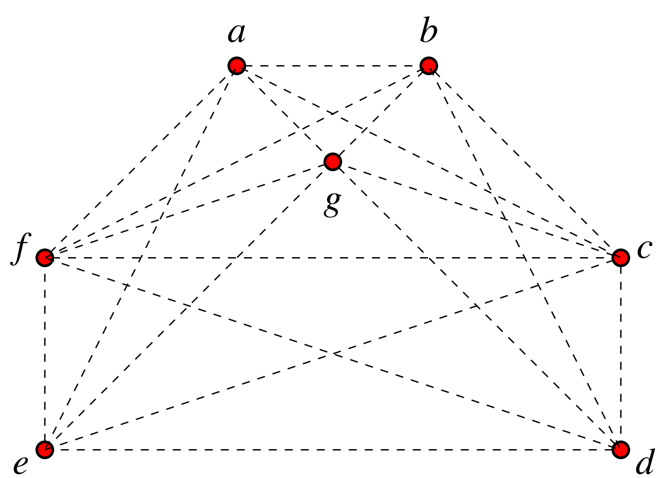
Flip distance is trivial!

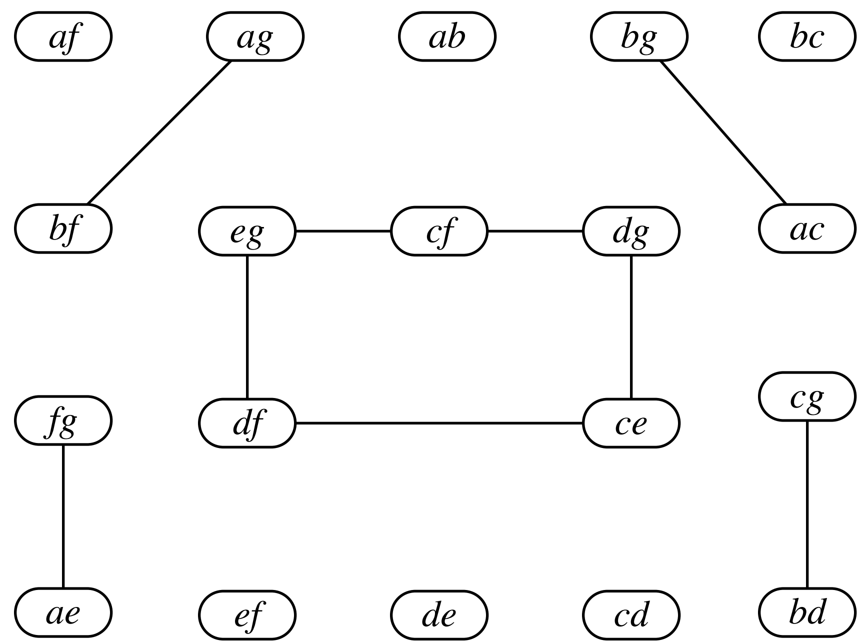
If there are empty quadrilaterals...

describe them all using the **quadrilateral graph**

vertices = line segments between input points

edges = pairs of diagonals of empty quadrilaterals

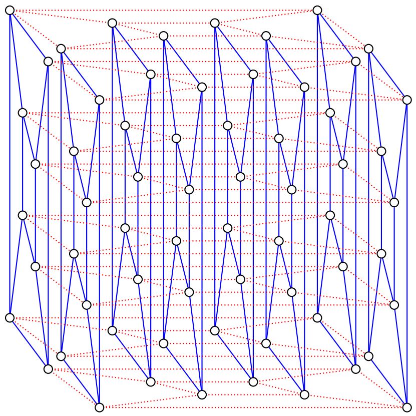




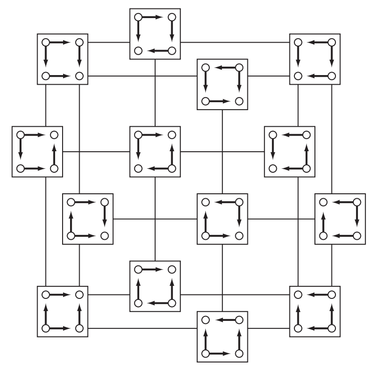
Partial cubes

Graphs that can be labeled by bitvectors so graph distance = Hamming distance

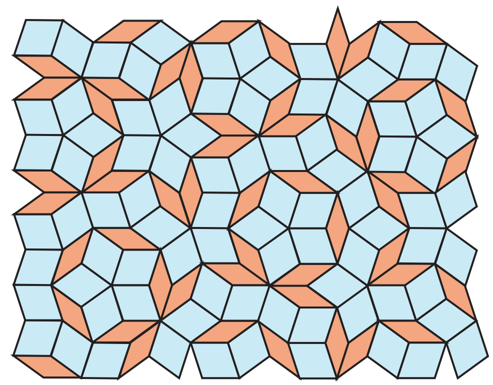
Examples:



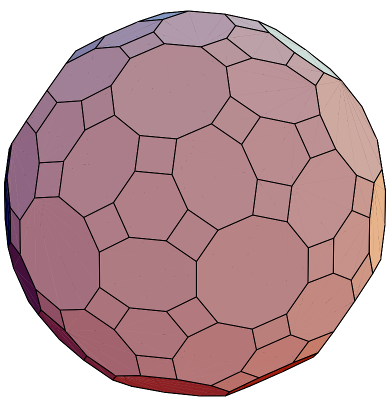
hypercube



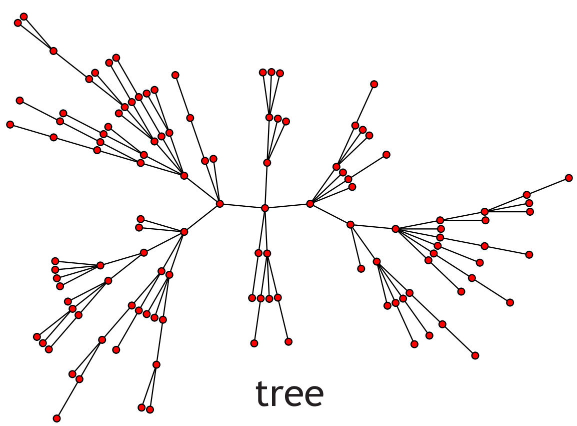
acyclic orientations of
an undirected graph



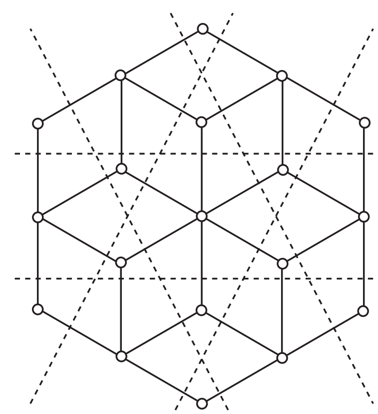
Penrose rhomb tiling



zonohedron



tree



adjacent regions of
line arrangement

Point sets with no empty pentagon

= point sets for which quadrilateral graph is a forest

= point sets for which flip graph is a partial cube

(our main results)

Idea for proof that no empty pentagon => forest:
- transform point set so Delaunay triangulation unique
- parent of edge = replacement edge of Delaunay flip
- there can be only one

Idea for proof that flip graph is partial cube:
- triangulation has one edge per tree in forest
- find short paths via constrained DT of shared edges

Algorithm for flip distance
(for point sets with no empty pentagon)

Flip both triangulations to Delaunay

Find edges that occur in one flip sequence but not both

Flip distance = number of such edges

Total time: $O(n^2)$

Condition that input has no empty pentagon can also be tested in $O(n^2)$

Estimating flip distance for more general point sets

Represent triangulation as subset of quad graph vertices

For each quad graph vertex in T1, find path connecting it to a corresponding quad graph vertex in T2

Minimize total length of paths (min weight matching)

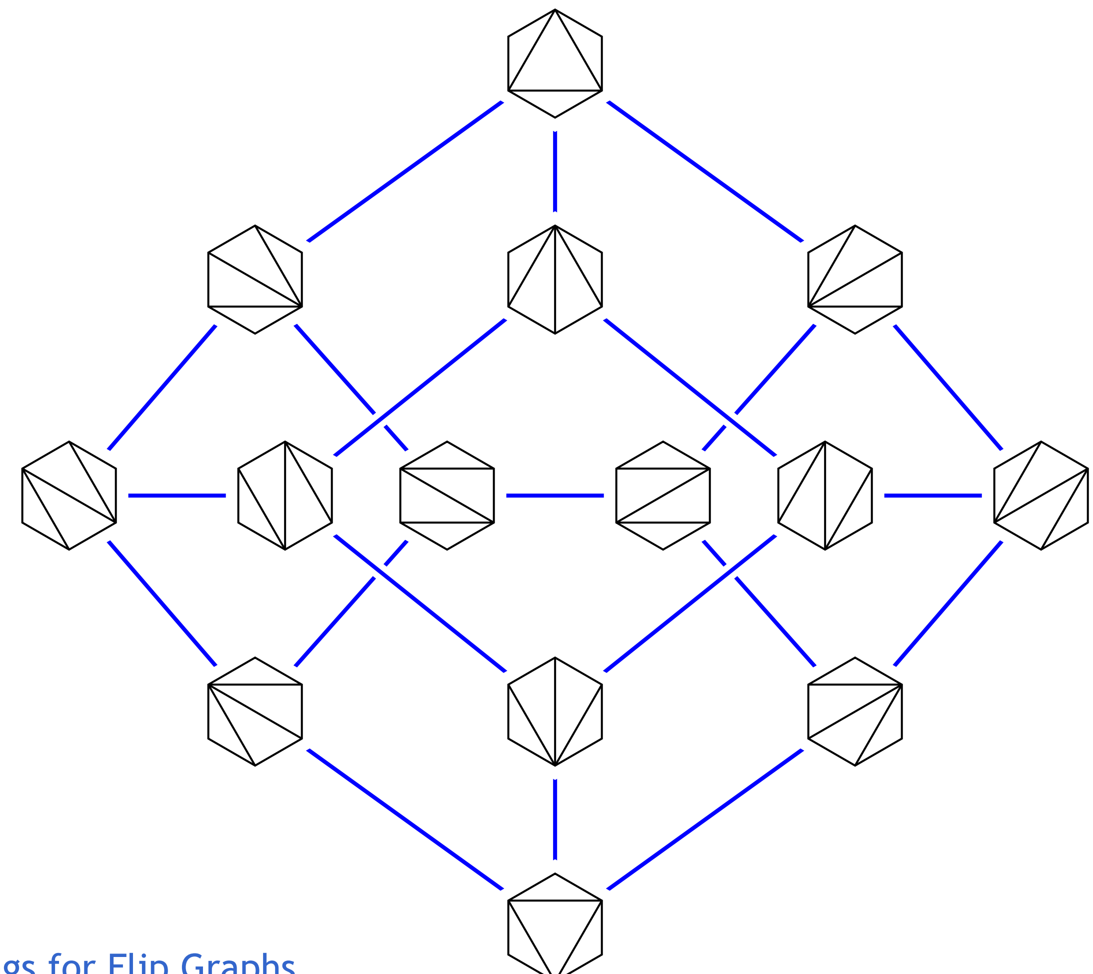
Flip distance ≥ total path length

Underestimate of true distance, suitable for A* algorithm

Point sets with no empty hexagon

necessary condition for estimated distance = flip distance

conjecture: also sufficient condition



flip graph of a hexagon

top, bottom triangulation
have flip distance = 4
estimated distance = 3

Conclusion

first progress on computing flip distance in nontrivial family of instances

complexity hierarchy for point sets?
(bigger empty polygons => more complex)

sometimes general-position assumptions hide interesting geometry