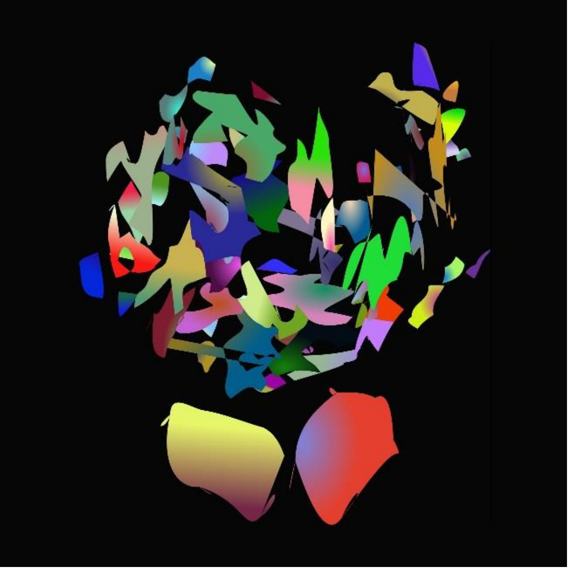


Algorithms for 3D Printing and Other Manufacturing Processes

The Width of a Polyhedron

Dan Halperin
School of Computer Science
Tel Aviv University

Spring 2017

Re-orient a heavy model to reduce height

- Heavy model, $\geq$ several 100,000s of triangles
- Find the width and re-orient
- The width algorithms needs to be robust and efficient
- We will also discuss approximation, allowing to report $(1+\varepsilon)w$, where w is the (minimal) width
- We start with an exact solution to the 3D width problem

Outline

- Quasi output-sensitive algorithm via Gaussian maps
- Improved algorithms
- Approximation
- Robustness issues
- Generalization: penetration depth
- Minkowski sums, take I

Width, reminder

- Input: A polyhedron P in R^3
- Output: The minimum distance between two parallel supporting planes to P, delimiting a slab containing P

The structure of the problem

The complexity of a convex polyhedron

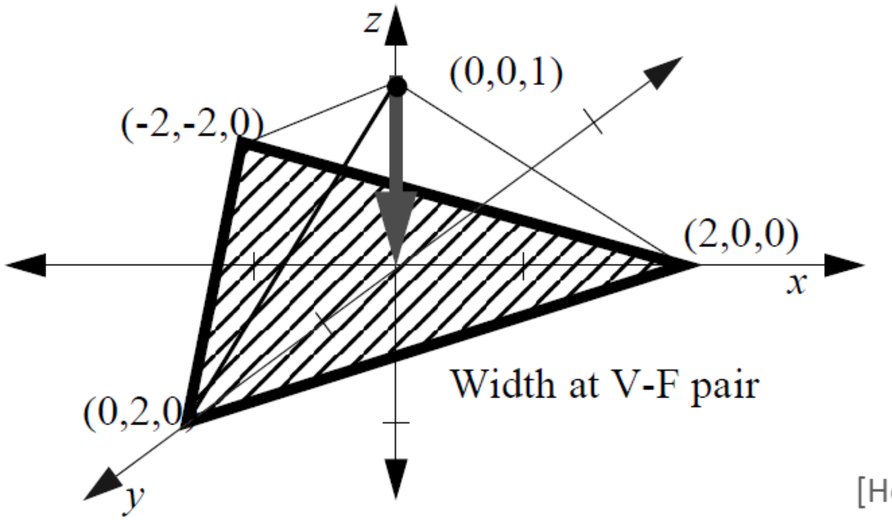
- The number of vertices is n
  - The number of edges is at most 3n-6
  - The number f faces is at most 2n-4
- If the facets are triangular then the bounds are tight

Relevant contact pairs of the supporting planes

- V-V
- V-E
- V-F
- E-E
- E-F
- F-F

The case V-F

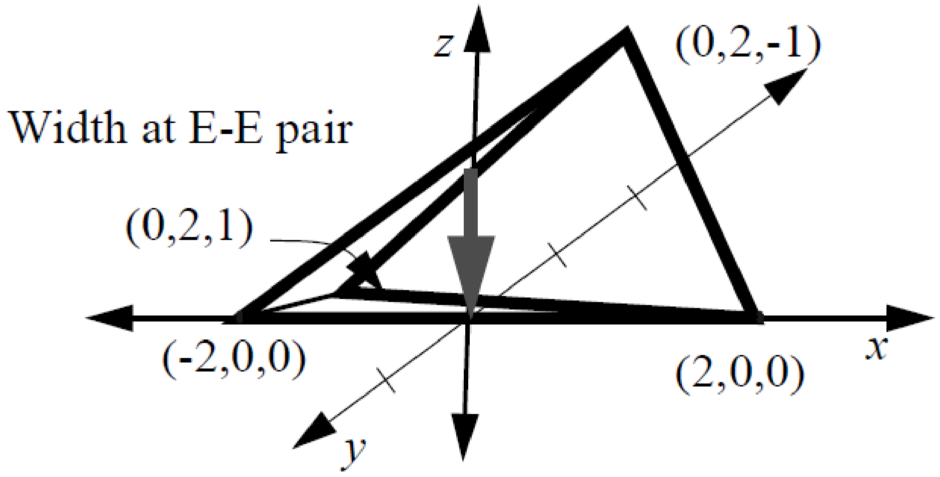
- O(n) pairs
- The distance between a plane and a point



[Houle-Toussaint]

The case E-E

- $O(n^2)$ pairs
- The distance between a pair of lines



[Houle-Toussaint]

Exact algorithms

The Gaussian map of a polytope P in R^3

- The external normal to a facet of P

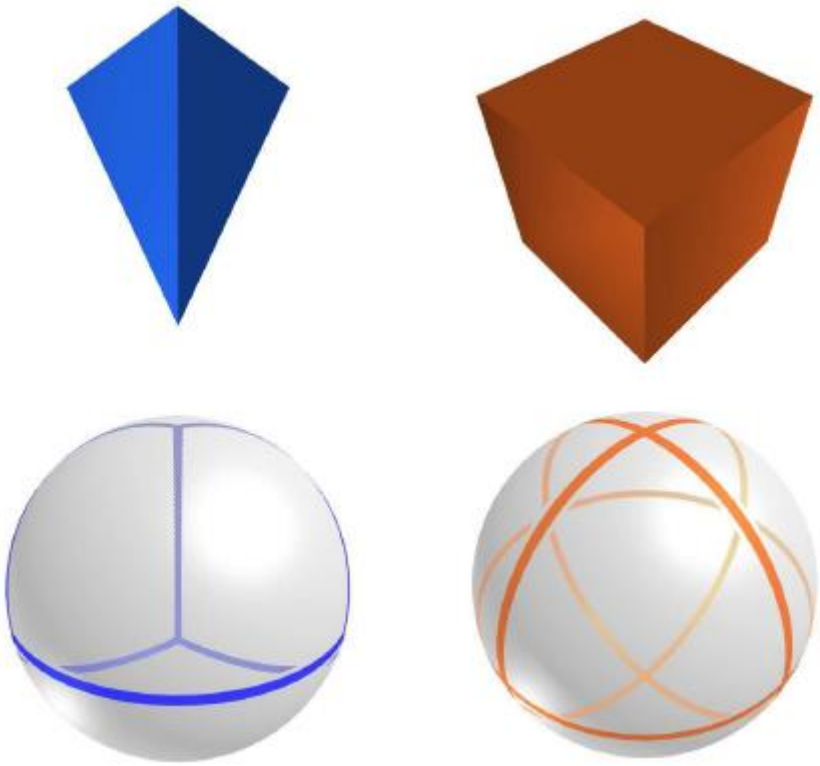
-> a vertex on S^2

- The external normals to

-> an arc on S^2

- The external normals to supporting planes of a vertex of P

-> a face on S^2

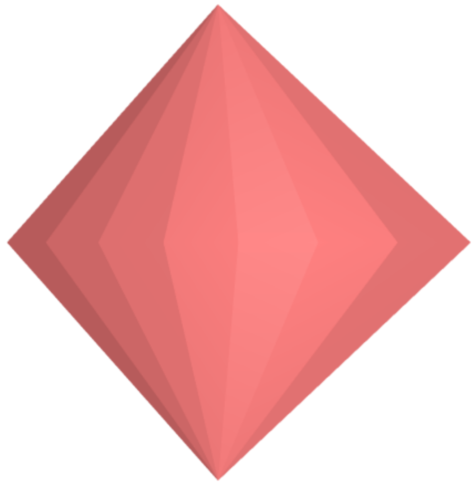


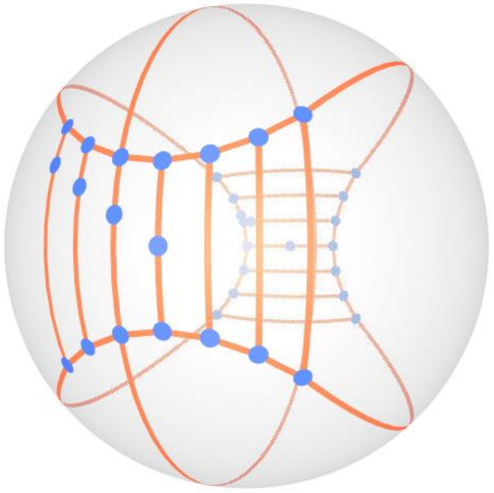
Gaussian map overlay

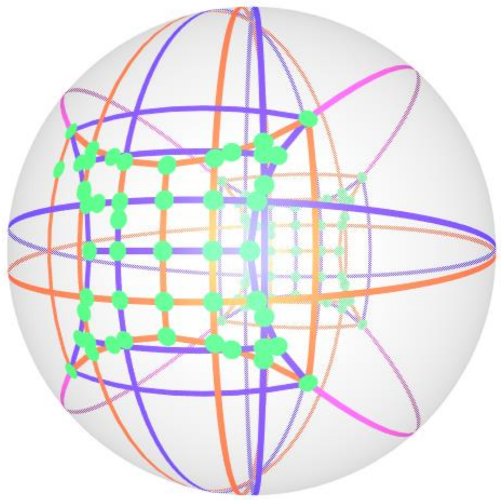
- Overlay of the map and a mirrored version through the origin
- Sufficient to look at the upper hemisphere
- Caution about the equator
- Can be transformed into an arrg on the plane z=1

The complexity of the overlay

- Two sets of n/2 points in R^3, each arranged along one of two skewed arcs
- Take the CH of these n points to yield the polyhedron
- The overlay has complexity $\Omega(n^2)$







Algorithms

- Quasi output-sensitive algorithm
  - Plane sweep, O((n+k) log n), k number of relevant EE pairs
  - Special convex-map overlay [Guibas-Seidel], O(n+k)
- Randomized
  - involved [Agarwal-Sharir], $O(n^{3/2+\varepsilon})$

Approximation

Strategies

- Grid of directions on S^2
  - requires some extra machinery(?)—see below
- Simplify the polytope
- Coresets

Robustness

What can go wrong when computing the CH

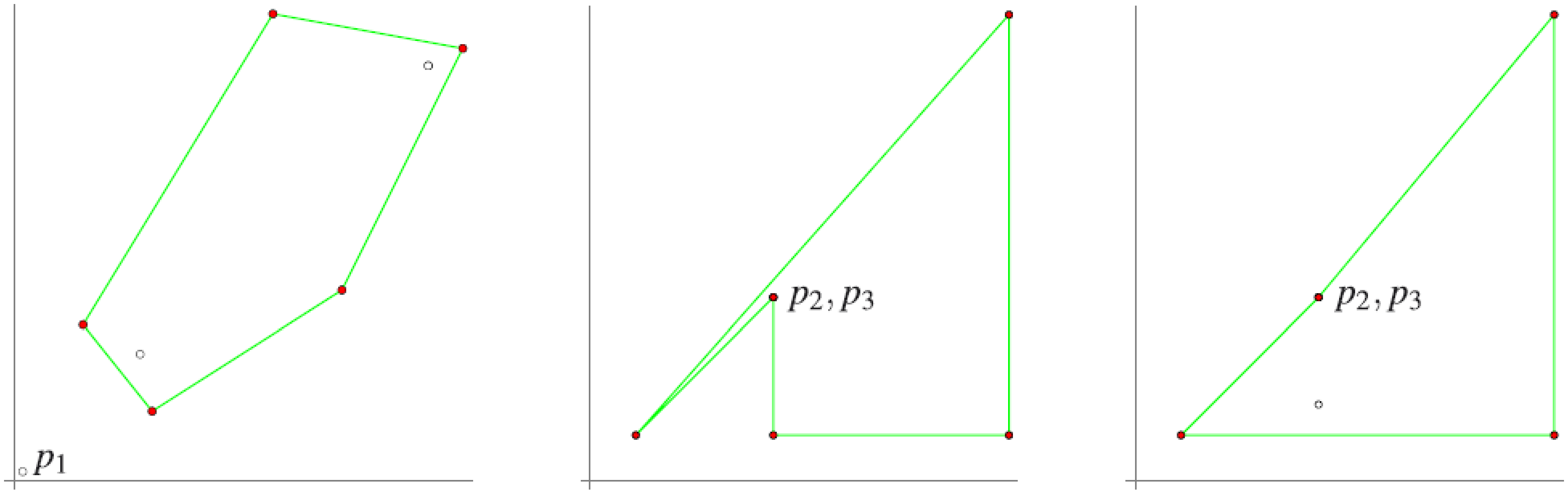
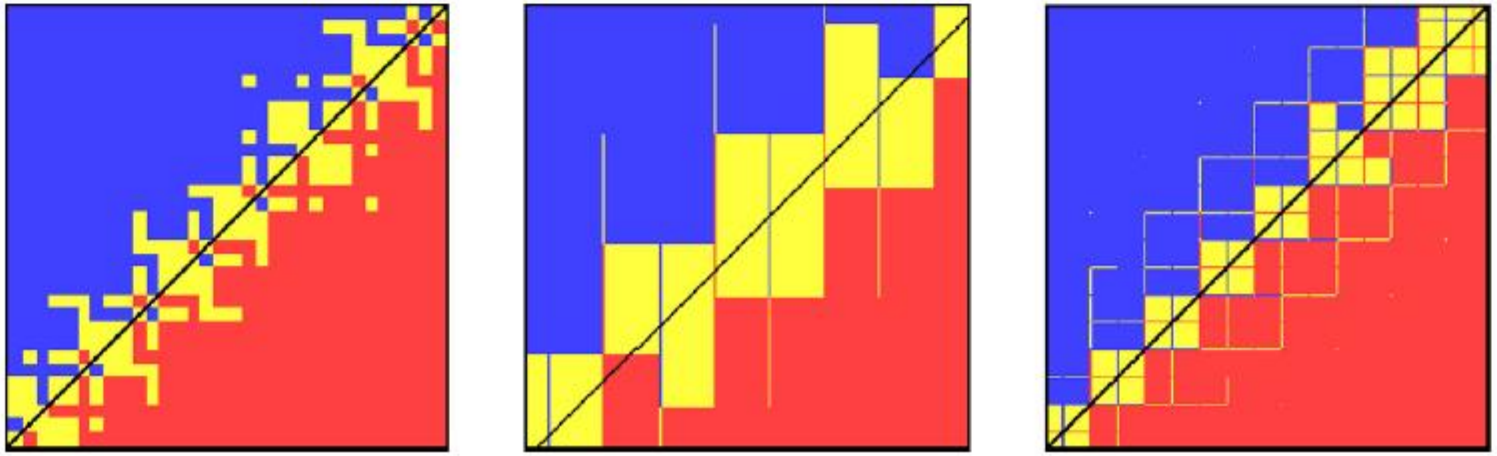


Fig. 1. Results of a convex hull algorithm using double-precision floating-point arithmetic with the coordinate axes drawn to give the reader a frame of reference. The algorithm makes gross mistakes (from left to right): The clearly extreme point p_1 is left out. The convex hull has a large concave corner with a (non-visible) self intersection near p_2 and p_3, which are close together. The convex hull has a clearly visible concave chain (and no self-intersection). Details on these examples are explained in Section 4.

[kettner et al]

What can go wrong, cont’d



(a) $p: \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$, $q: \begin{pmatrix} 12 \\ 12 \end{pmatrix}$, $r: \begin{pmatrix} 24 \\ 24 \end{pmatrix}$

(b) $\begin{pmatrix} 0.50000000000002531 \\ 0.5000000000000171 \end{pmatrix}$, $\begin{pmatrix} 17.300000000000001 \\ 17.300000000000001 \end{pmatrix}$, $\begin{pmatrix} 24.00000000000005 \\ 24.0000000000000517765 \end{pmatrix}$

(c) $\begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$, $\begin{pmatrix} 8.8000000000000007 \\ 8.8000000000000007 \end{pmatrix}$, $\begin{pmatrix} 12.1 \\ 12.1 \end{pmatrix}$

Fig. 2. The weird geometry of the float-orientation predicate: The figure shows the results of *float_orient*$(p_x + Xu_x, p_y + Yu_y, q, r)$ for $0 \leqslant X, Y \leqslant 255$, where $u_x = u_y = 2^{-53}$ is the increment between adjacent floating-point numbers in the considered range. The result is color coded: Yellow (red, blue, resp.) pixels represent collinear (negative, positive, resp.) orientation. The line through q and r is shown in black.

[kettner et al]

Exact predicates are necessary and sufficient

- For computing the convex hull
- Arbitrary precision rational numbers will do assuming the input vertex coordinates are rational
- Compute squared distance (squared width)

Rounding, why we may need it

- Example: vertical decomposition of arrgs of triangles

- The coordinates (x,y,z) of every triangle corner are each represented with a 16-bit over 16-bit rational

Complexity of numbers, input coordinates

```
Triangle 1:
     (-9661 / 499, 898 / 2689, -92949 / 3802),
     (-15034 / 1583, -8174 / 1759, -57116 / 3851),
     (13605 / 1261, -90590 / 3669, -11791 / 518)

Triangle 2:
     (-77665 / 4036, -130679 / 3347, -31167 / 1630),
     (-5851 / 297, 36471 / 893, -53137 / 2704),
     (132613 / 3310, 3 / 8, -21926 / 1111)

Triangle 3:
     (-37497 / 1939, -131078 / 3301, 591 / 3680),
     (-74461 / 3822, -28120 / 3397, 7607 / 346),
     (21622 / 1037, -12461 / 1441, 17957 / 827)

Triangle 4:
     (-10760 / 521, -58546 / 3057, 27619 / 1322),
     (-65262 / 3181, 74693 / 3622, 17898 / 863),
     (48898 / 2419, 1602 / 1627, 26390 / 1273)

Triangle 5:
     (-73482 / 3845, 88794 / 2203, 2720 / 3661),
     (-20591 / 1049, 9257 / 983, 57830 / 2693),
     (28590 / 1363, 38699 / 3957, 62390 / 2957)
```

Complexity of numbers, computed coordinates

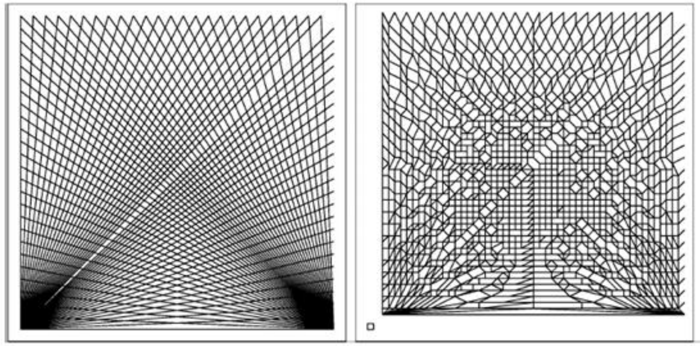
A normalized coordinate of the worst feature of the partial decomposition — 237 digits long:

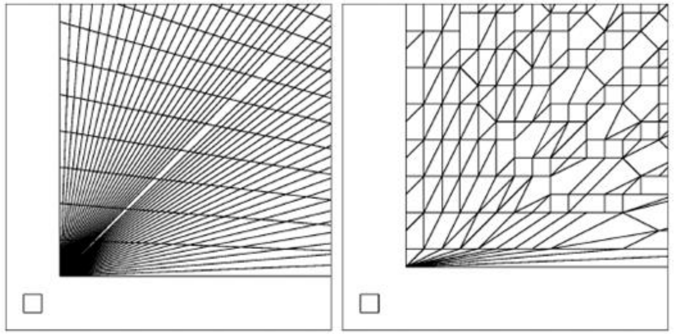
```
PD feature = 49799838826104887192775516219046994702
461828025059123646217485873346921099238939609590257
269896740240221692997023332971 / 50277907098591079 37
563103744532644005619919434042984323896243977724409
284407170688213486885149673158070430134598067 16
```

A normalized coordinate of the worst feature of the full decomposition — 559 digits long:

```
FD feature = 23279315243924676155798958688382904585
988203585590361740839519681254968145162747098072652
141858607502723046239367209776569259776678871640355
476703121623912558549584789123982974129958278704985
390744483577662104085231708340232525122368990013542
79996132937206816849552931288112 92981 / 22458231406
216094878202976126790054324698816432478447511802089
665363641250066501433769538474807742947270581109819
674675916341254734148663444090199254276142009850182
419444726060661342077926179045344110704705488623957
680809306210269199637837088757430354530277343135738
809521441456
```

Snap rounding arrangements of segments





Generalization: penetration depth

What is penetration depth

- Let A and B be two convex polyhedra in R^3. The penetration depth of A and B, denoted $\pi(A,B)$, is the minimum distance by which A has to be translated such that A and B do not intersect

$$\pi(A,B) = \min\{\|t\| \mid \operatorname{int}(A+t) \cap B = \emptyset,\ t \in \mathbb{R}^3\}$$

Width and penetration depth

Claim: For a convex polyhedron P, width(P)= $\pi(P,P)$

- Let w be the width, and v be the vector realizing it
- Let s be the minimum separation distance and u be the vector realizing it
- $s \leq \| v \|$
- $w \leq \| u \|$
- $s \leq \| v \| = w \leq \| u \| = s$

Computing the penetration depth, preliminaries

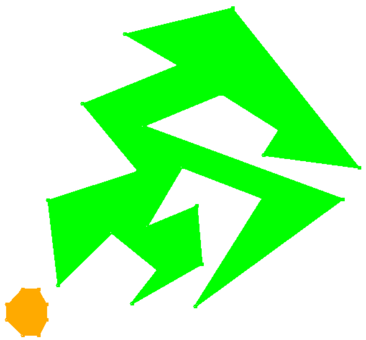
- Let A and B be two convex polyhedra in R^3 with m and n facets respectively
- We can determine in O(m+n) time whether they intersect (LP)
- If they do not, then $\pi(A,B)=0$ and we are done
- Otherwise, we move to a configuration-space formulation, where B is a static obstacle and A is translating
- Let P denote the Minkowski sum $B \oplus (-A)$
- Let O denote the origin of the coordinate system
- then $\pi(A,B)= \min\{d(O,x\} \mid x \in bd(P)\}$

Minkowski sums

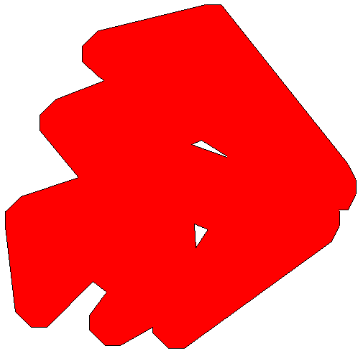
Take I

The Minkowski sum of two sets P and Q in Euclidean space is the result of adding every point in P to every point in Q

$$\{(x_1, y_1)\} \oplus \{(x_2, y_2)\} = \{(x_1 + x_2, y_1 + y_2)\}$$



P, Q

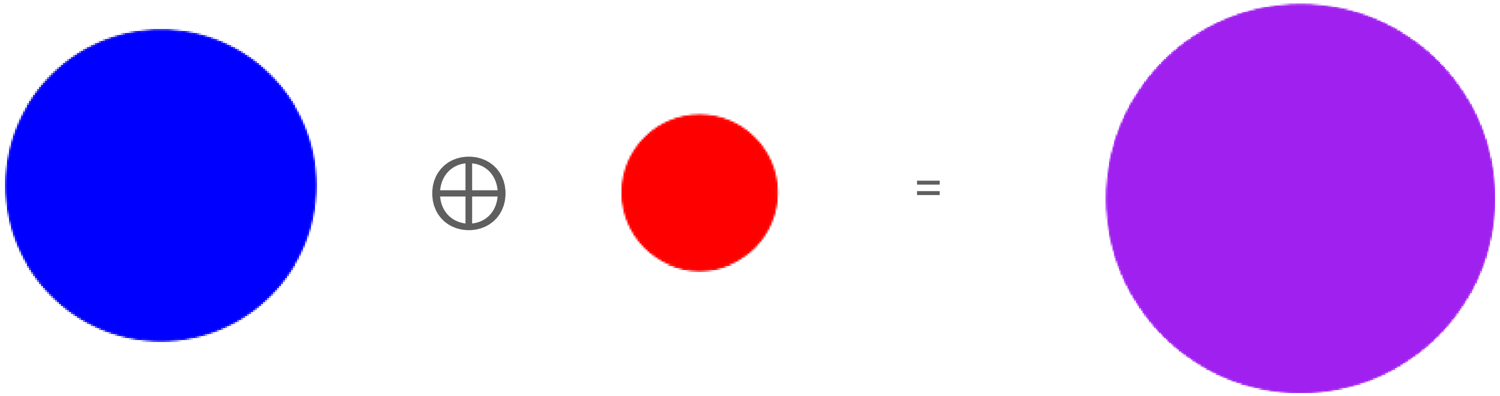


$P \oplus Q$

[wikipedia]

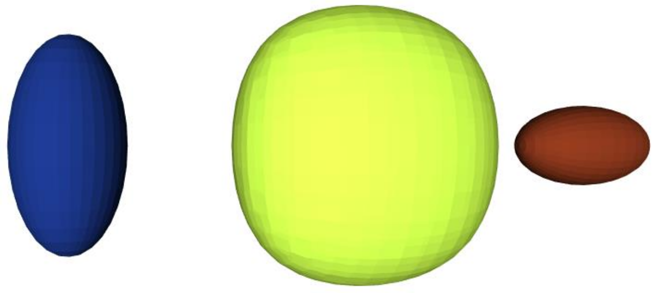
1864 - 1909

Warm-up



Convex polytopes

- The farthest point of the sum in any direction is the sum of the farthest points in that direction of the summands
- The sum of convex polytopes is a convex polytope
- For polygons with m and n vertices, the sum has at most $m + n$ vertices
- For polytopes (3D) with m and n vertices, the sum has $\Theta(mn)$ vertices; exact numbers [Fogel-H-Weibel '09]



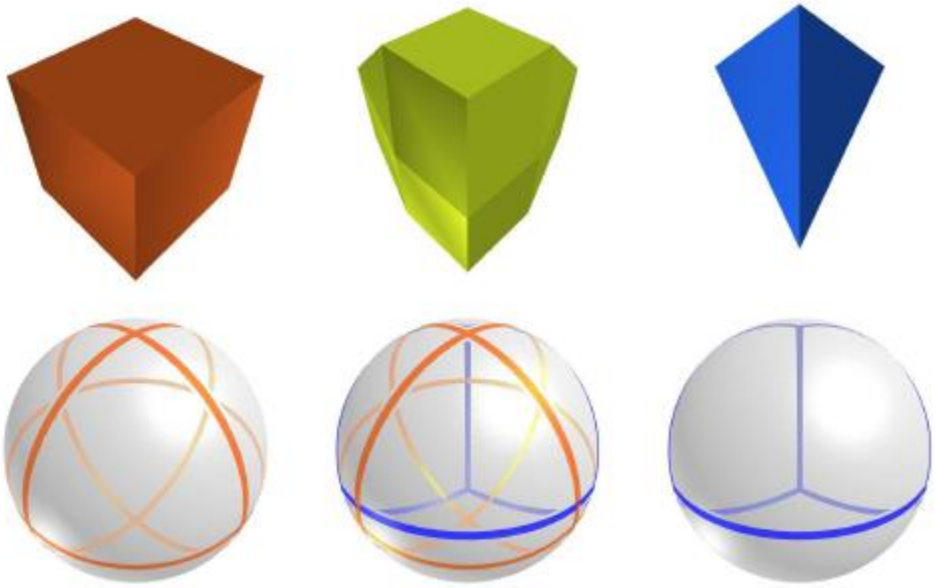
Minkowski sums and Gaussian maps

Observation

The overlay of the Gaussian maps of two convex polytopes P and Q is the Gaussian map of the Minkowski sum of P and Q.

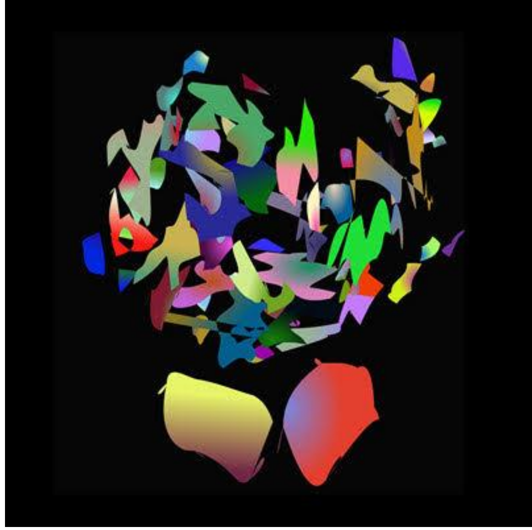
$$\text{overlay}(G(P), G(Q)) = G(P \oplus Q)$$

- The overlay identifies all the pairs of features of P and Q respectively that have common supporting planes.
- These common features occupy the same space on $\mathbb{S}^2$.
- They identify the pairwise features that contribute to $\partial(P \oplus Q)$.

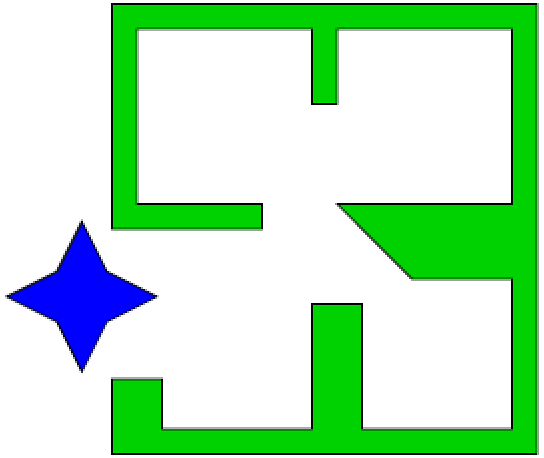


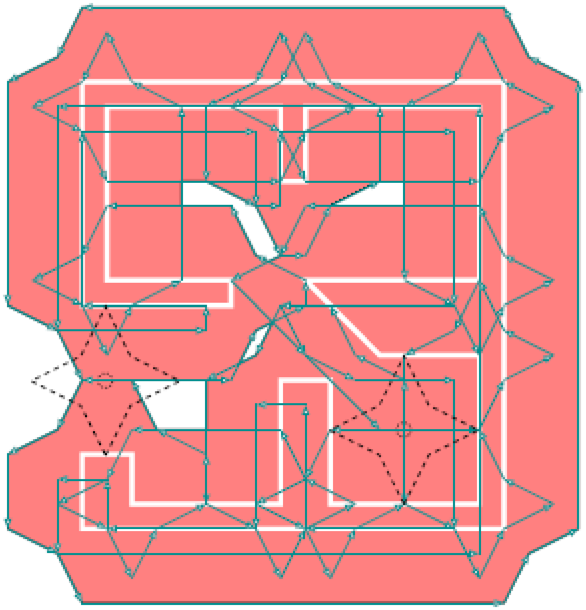
Cube Minkowski sum tetrahedron

How to represent Minkowski sums in general? The language of arrangements



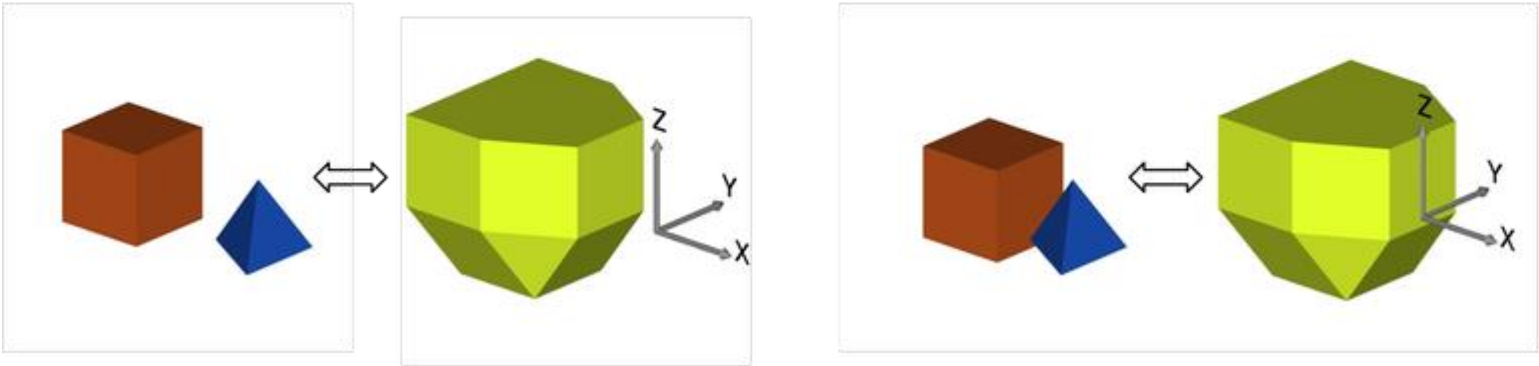
- Much more involved than the convex case
- Should allow for complex topology, holes of any dimension
- Arrangements of curves and surfaces do the job





Why are Minkowski sums so useful?
Here’s a major reason:

- Claim: Two sets A and B intersect if and only if the Minkowski sum $A \oplus -B$ contains the origin, where $-B$ is the set B reflected through the origin

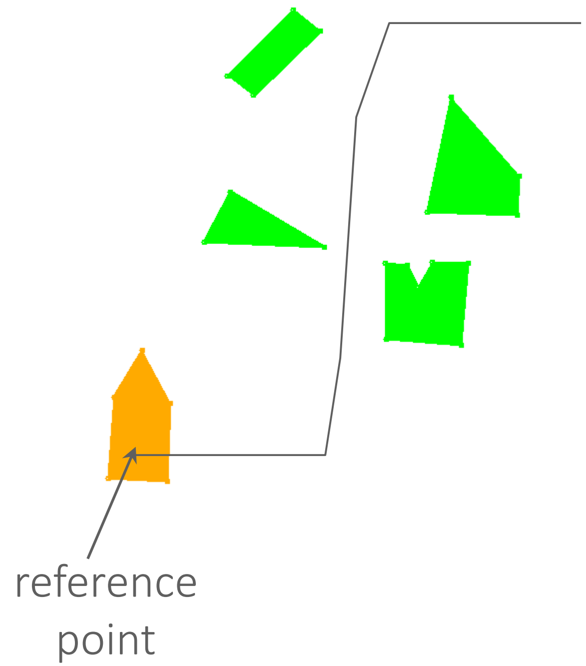


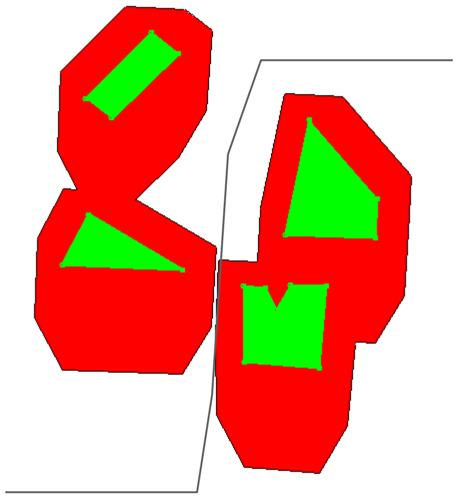
In the plane $-B$ is B rotated by π radians around the origin

Example

R - a polygonal object that moves by translation

P - a set of polygonal obstacles





Claim: When translating, R intersects P iff ref(R) is inside $P \oplus -R$

Back to penetration depth

Reminder, computing the penetration depth

- Let A and B be two convex polyhedra in R^3 with m and n facets respectively
- We can determine in O(m+n) time whether they intersect (LP)
- If they do not, then $\pi(A,B)=0$ and we are done
- Otherwise, we move to a configuration-space formulation, where B is a static obstacle and A is translating
- Let P denote the Minkowski sum $B \oplus (-A)$
- Let O denote the origin of the coordinate system
- then $\pi(A,B)= \min\{d(O,x\} \mid x \in bd(P)\}$

Computing the penetration depth, cont'd

- Find the shortest distance from O to the boundary of the Minkowski sum B⊕(-A)
- It is the distance between O and a face of B⊕(-A)
- Each face is the sum of a vertex of one and the face of another, or an edge of one and an edge of another
- All edges correspond to vertices of the overlay of the Gaussian maps of B and –A
- Maximum complexity of the overlap $\Theta(mn)$
- Notice the similarity with width computation

Approximating the penetration depth

- And hence the width w
- We allow to report (1+ε)w
- Divide the interval [0,π] into ceiling($c_1/\sqrt{\varepsilon}$) intervals for a constant c_1
- Create a grid of points on S^2 such that from any point on S^2 the distance to a grid point is at most $\sqrt{\varepsilon}$
- For each grid point p compute the distance between O and the intersection of the ray from O indirection p with the boundary of B⊕(-A)
- Output the smallest such distance as w’

Computing the directional penetration depth

- What is the minimum separation distance in direction p?
- Can we find it efficiently without computing the entire $B \oplus (-A)$?
- This can be done in $O(\log^2(m+n))$ using the hierarchical representation of each of B and –A [Dobkin et al]
- Why cannot we use the (much easier to compute) directional width?

Approximating the penetration depth, cont'd

Claim: w' ≤ (1+ε)w

- v: the vector that realizes the depth
- u: the computed vector (in the direction of a grid point)

$$||u|| \le \frac{||v||}{\cos\alpha} \le \frac{||v||}{1-\alpha^2/2} \le (1+\alpha^2)||v|| \le (1+\varepsilon)||v||$$

- Running time

$$O(m+n+(\log^2(m+n))/\varepsilon)$$

Computing the width in 3D: Bibliography

- Michael E. Houle, Godfried T. Toussaint: Computing the width of a set. Symposium on Computational Geometry 1985: 1-7

  Basics

- Pankaj K. Agarwal, Micha Sharir: Efficient Randomized Algorithms for Some Geometric Optimization Problems. Discrete & Computational Geometry 16(4): 317-337 (1996)

  $O(n^{3/2+\varepsilon})$ time algorithm

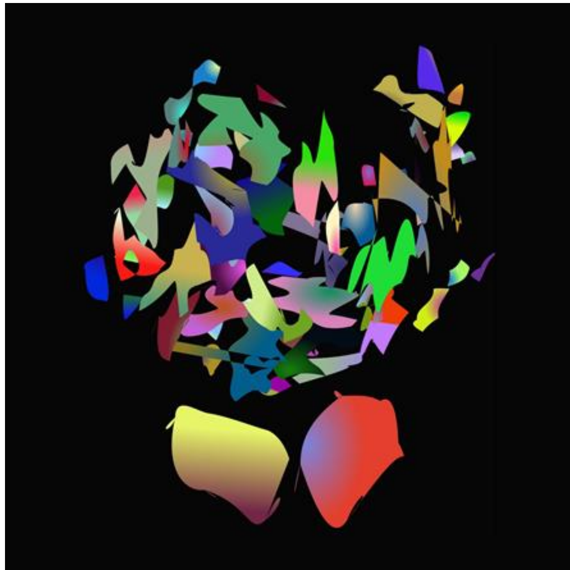
- Pankaj K. Agarwal, Leonidas J. Guibas, Sariel Har-Peled, Alexander Rabinovitch, Micha Sharir: Computing the Penetration Depth of Two Convex Polytopes in 3D. SWAT 2000: 328-338

  Includes the approximation algorithm via penetration depth

- David P. Dobkin, John Hershberger, David G. Kirkpatrick, Subhash Suri: Computing the Intersection-Depth of Polyhedra. Algorithmica 9(6): 518-533 (1993)

  Efficient computation of the directional penetration depth, needed in the approximation algorithm

THE END



[Gaither, ArtByAI, CGAL arrgs]