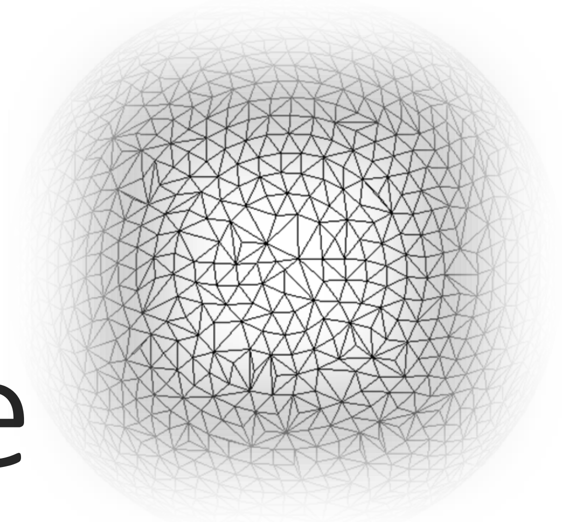
fhwedel
UNIVERSITY OF APPLIED SCIENCES

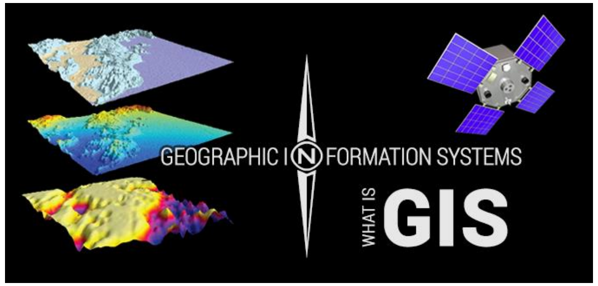


Computational Geometry
Convex Hull in 3-Space

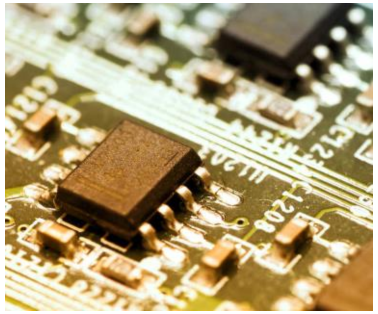
JONAS SORGENFREI

Introduction

Computational Geometry

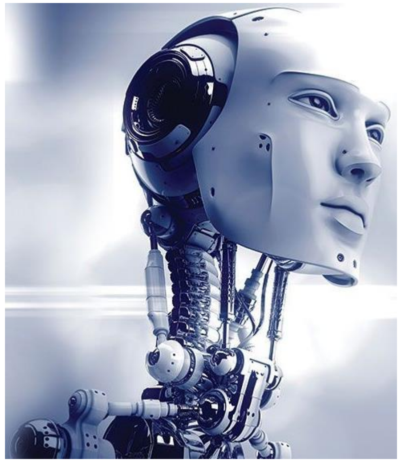


[2] – Geographic Information Systems

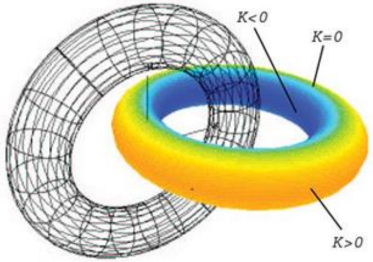


[3] – Integrated Circuit design

[6] – Computer Vision

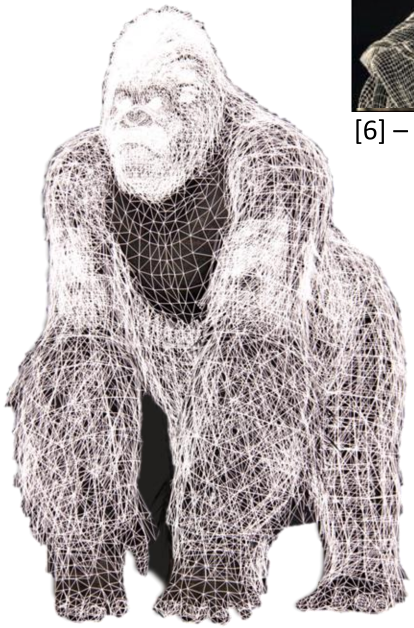


[1] - Robotics



[4] – Computer-Aided Engineering

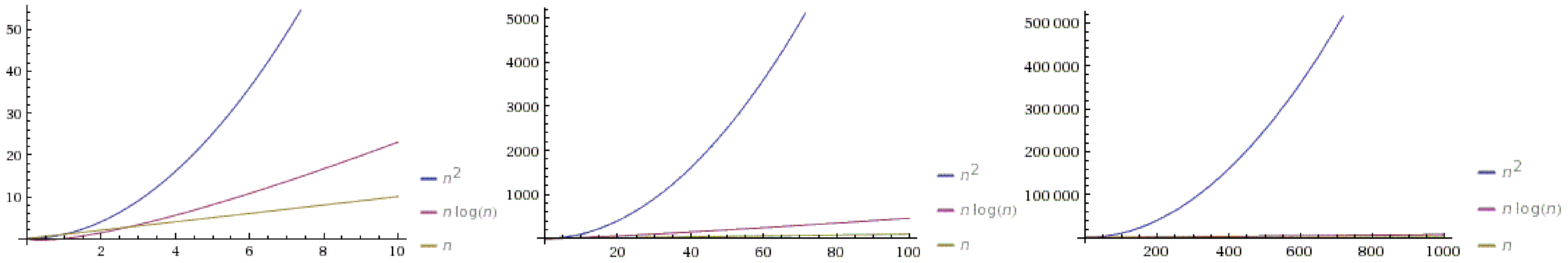
[5] - Databases



[7] – Computer Vision

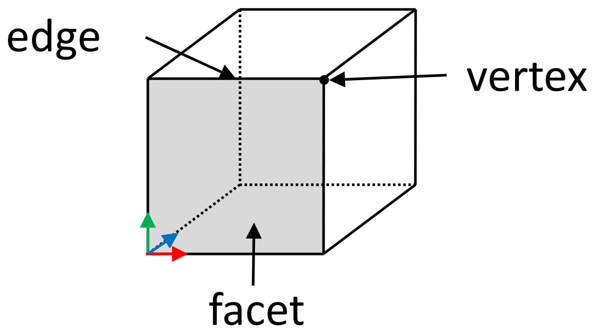
Introduction

Why efficients matters.



Plots of linear, squared and logarithmic running time

Introduction

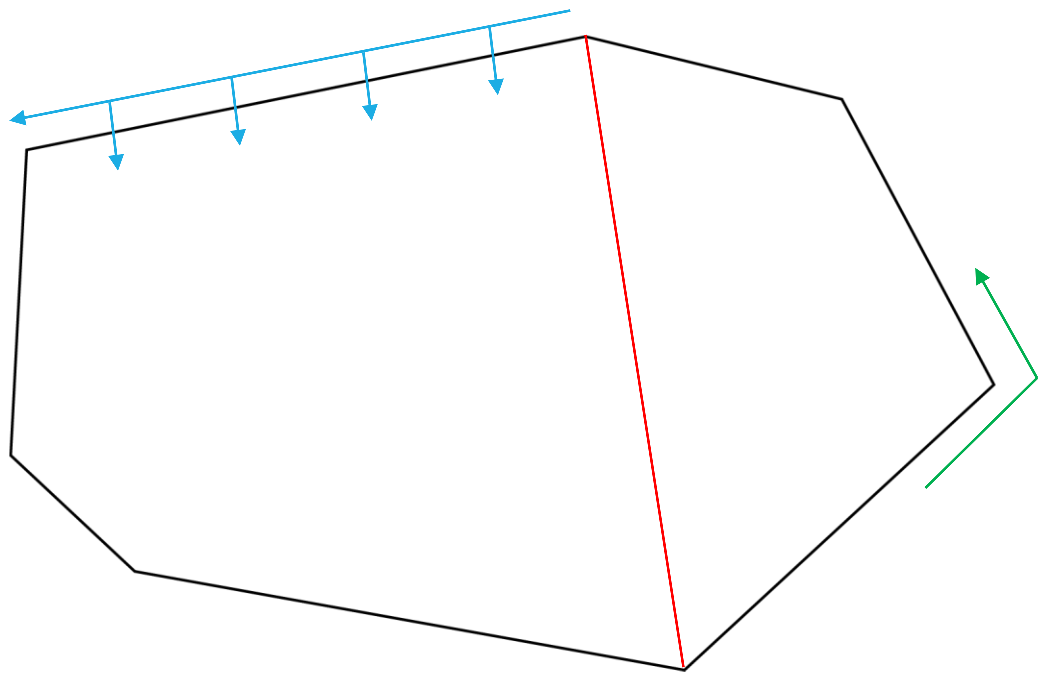


Convex Hull

Problem:

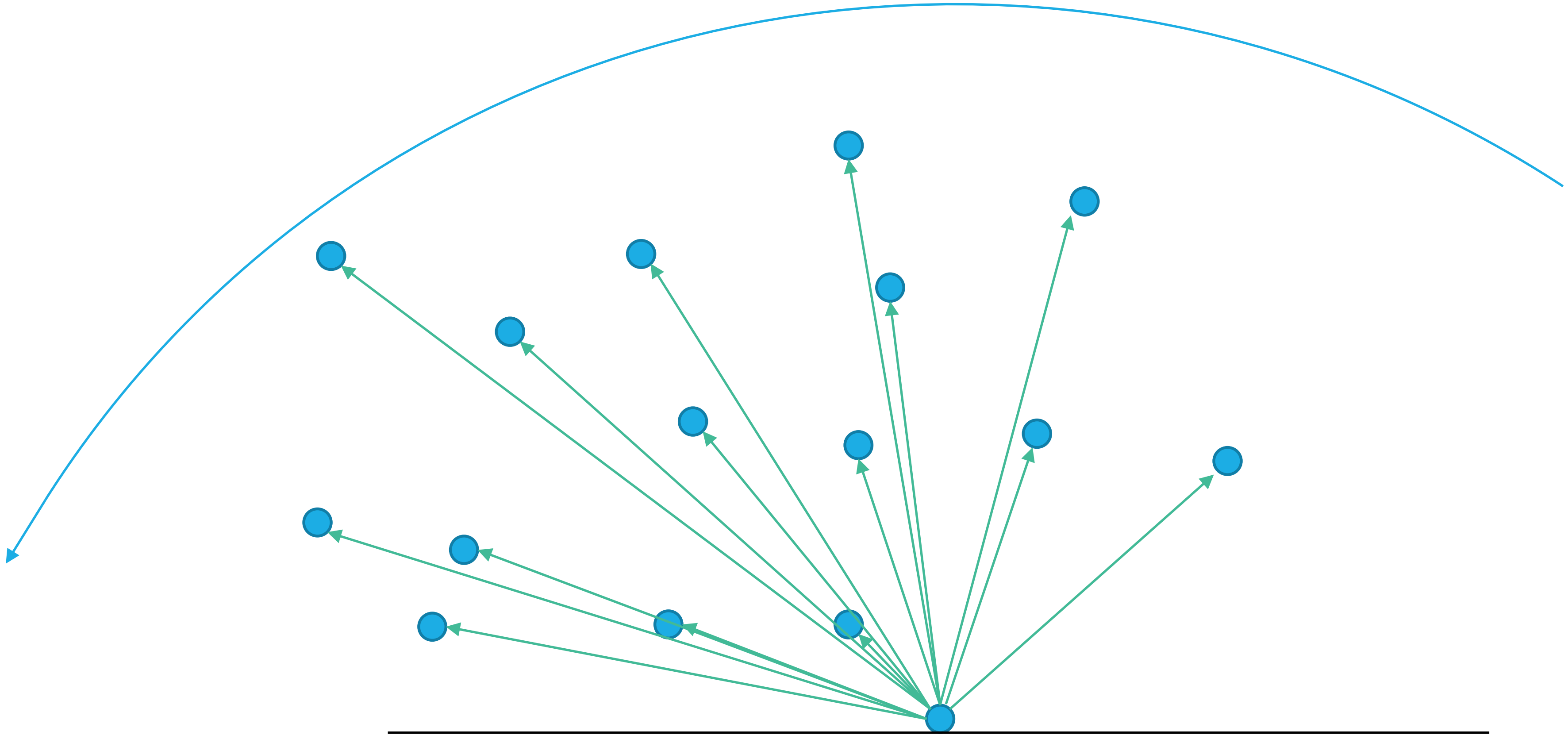
Given a set of n points, find the minimal convex polygon that contains all the points.

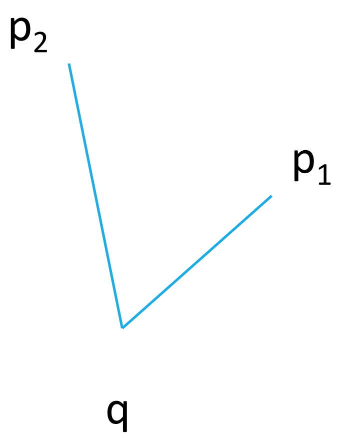
Convex Hull - Properties

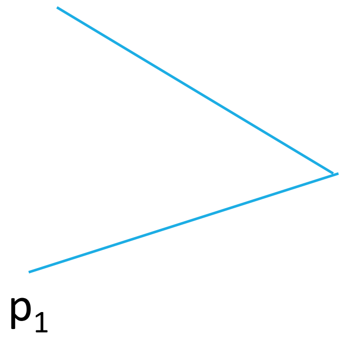


Graham Scan Algorithm

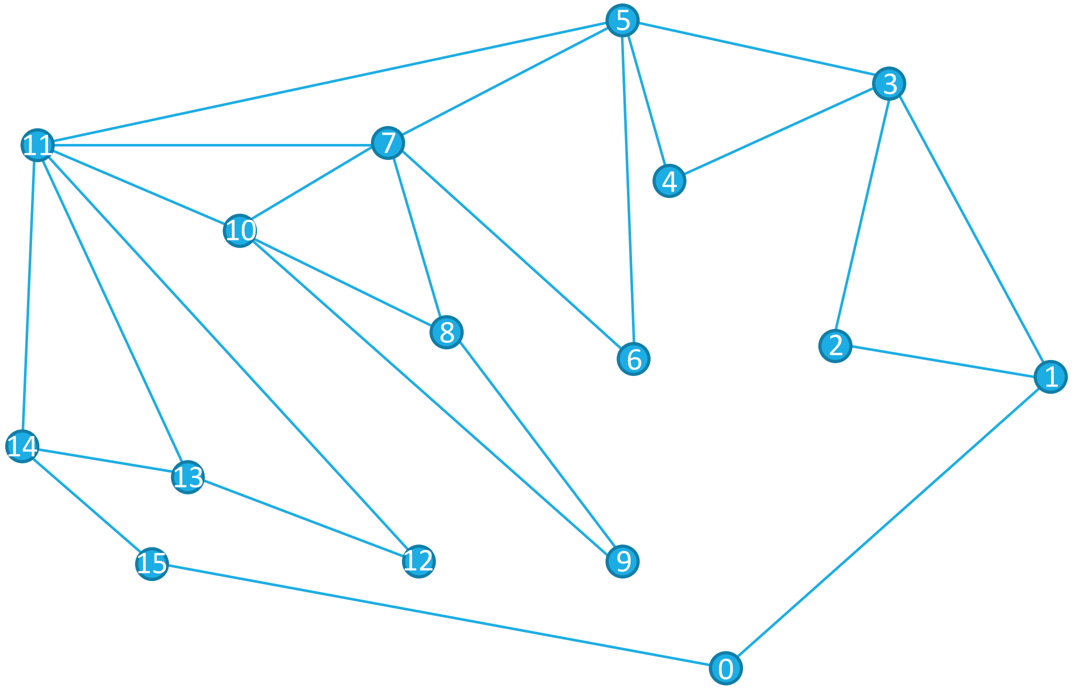
Convex Hull - Algorithms




p_2
p_1
q


p_1

Convex Hull - Algorithms



Convex-Hull Stack
15
14
5
3
1
0
0

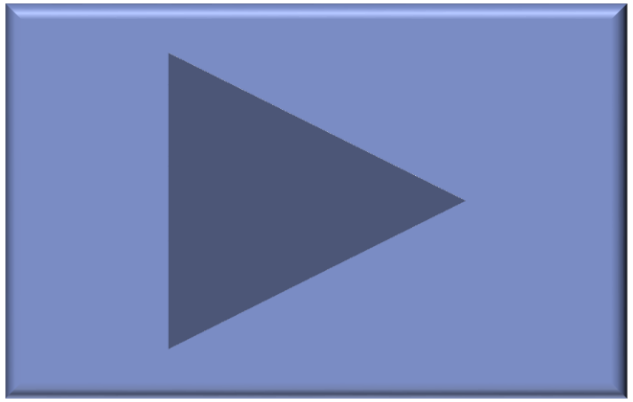
Convex Hull - Algorithms

Pseudo Code – Graham Scan Algorithm

- Select the point with the minimum y and swap with point at [0]
- Sort all points in CCW order $\{p_0, p_1, \ldots, p_n\}$
- $S = \{p_0, p_1, p_2\}$
- for i = 3 to n
  - While $|S| > 2$ && p_i is to the right of S_{-2}, S_{-1}
    - S.pop
  - S.push(p_i)

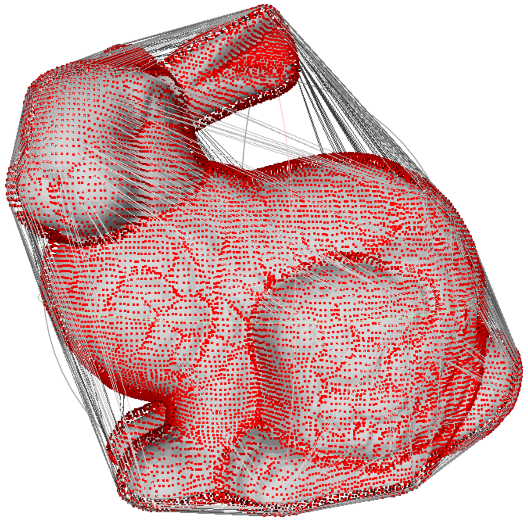
fhwedel UNIVERSITY OF APPLIED SCIENCES

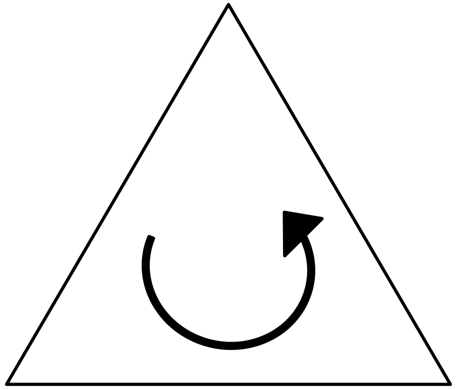
Convex Hull – Algorithms DEMO



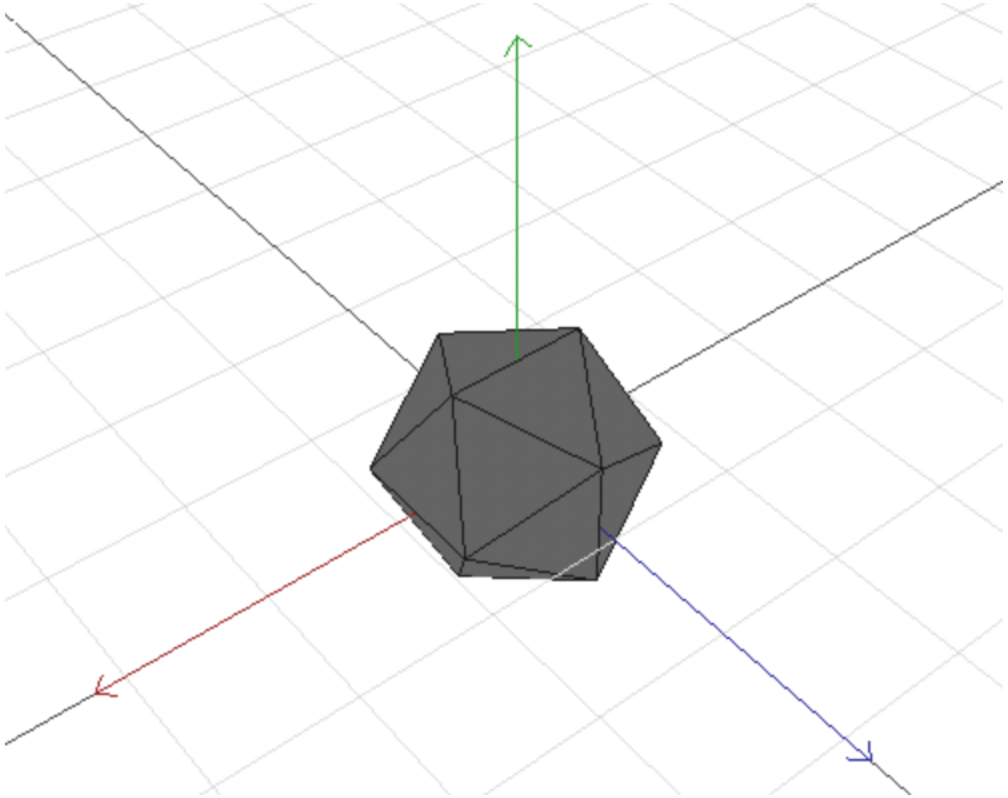
Convex Hull 3D

- polyhedron (convex 3-polytop) containing points
- for every facet of CH(P)
  - all other points of P lie in only one half space(back side)



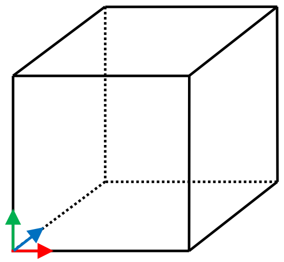


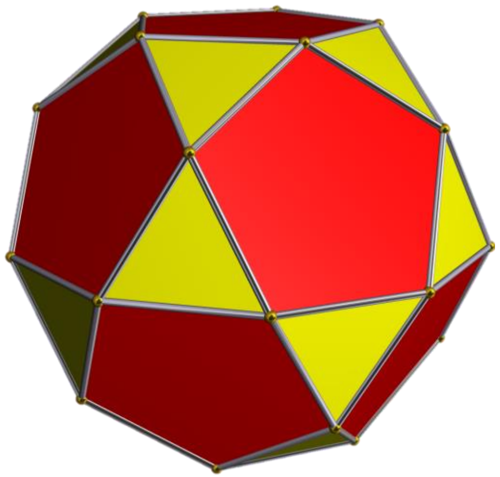
Frontface – CCW-Order



Convex Hull 3D - Complexity

- number of edges can be higher than the number of vertices
- facet = convex polygon





Icosidodecahedron

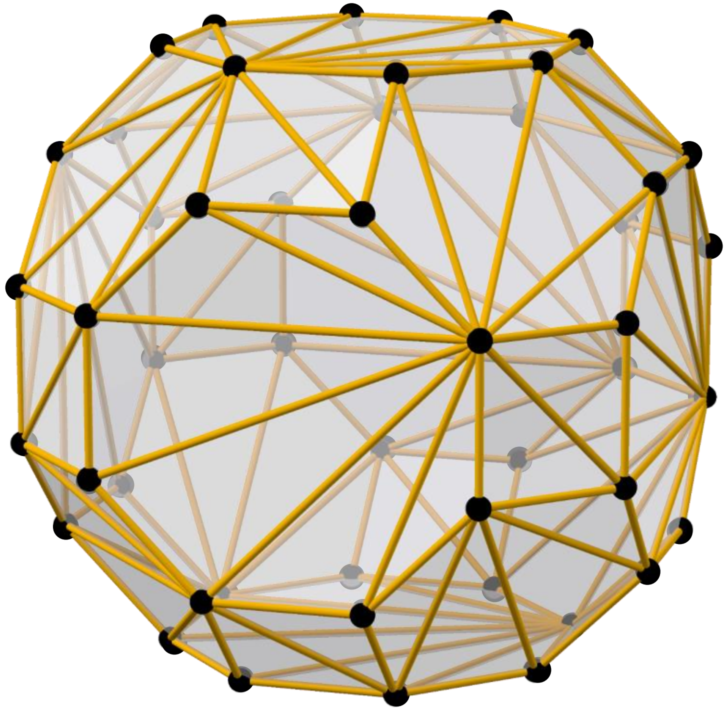
Convex Hull 3D - Computation

2 main problems

- finding extreme points
- finding connectivity

Saving CH in a Data Structure

- Saving Points
- Saving Facets (Connection of points)



fhwedel UNIVERSITY OF APPLIED SCIENCES

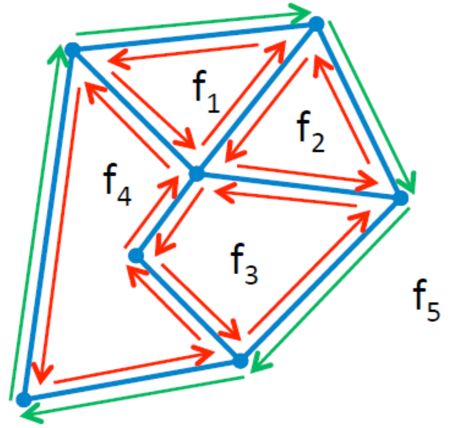
Doubly Connected Edge List

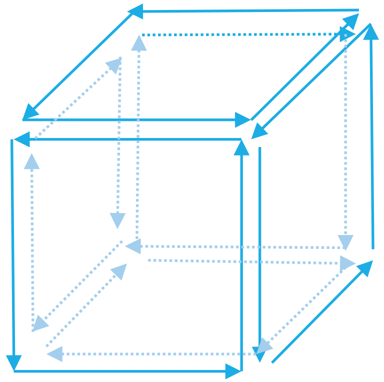
- data structure
- represents an embedding of a planar graph in the plane and polytopes in 3D
- needs functions to work on abstract data structure

links together:

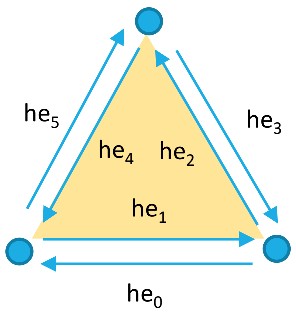
- **Vertex**
- **Halfedge**
    - Edge Connecting P1 → P2
    - (Twin) Edge Connecting P2 → P1
- **Face**

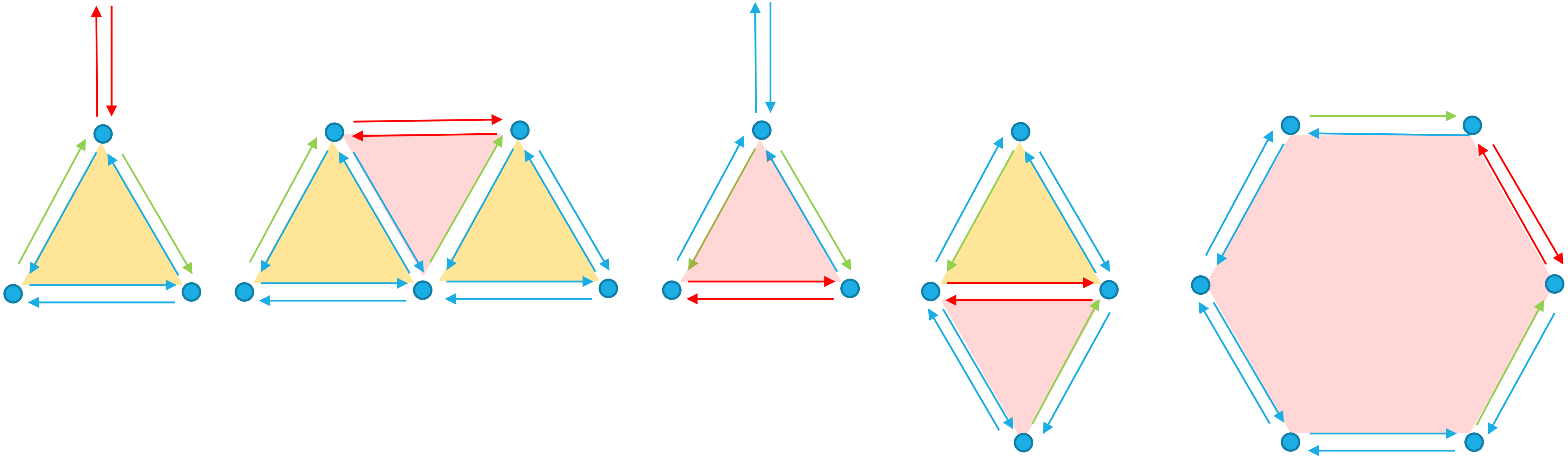
complex & complicated structure/programming!





fhwedel UNIVERSITY OF APPLIED SCIENCES


he5
he4
he2
he3
he1
he0



Convex Hull 3D - Computation

Randomized incremental algorithm

Input

$P = Set(point_1, point_2, \dots point_n)$

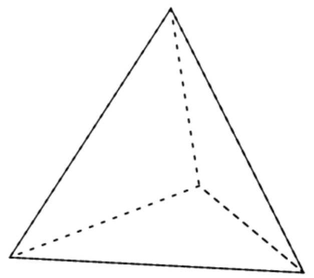
Output

CH(P) *(Convex Hull of P)*

Convex Hull 3D - Computation

1) choosing 4 Points, that **don't** lie in a common plane (tetrahedron)

- p1 and p2 be 2 Points in P
- p3 a point in P that does not lie on the line of p1 and p2
- p4 a point in P that does not lie in the plane p1,p2,p3
- (can't find 4 points → all point in P in a plane → 2D CH algorithm)



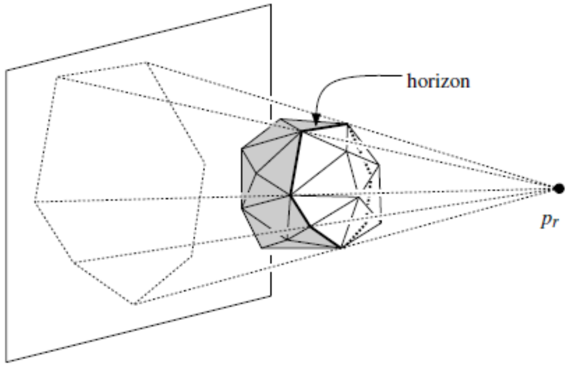
Tetrahedron

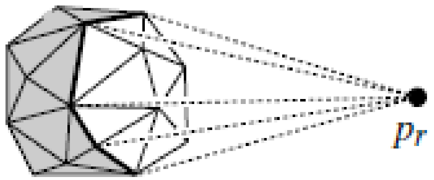
2) compute random permutation p_5, … p_n of remaining points

3) for all points $p_5,\ldots,p_n$

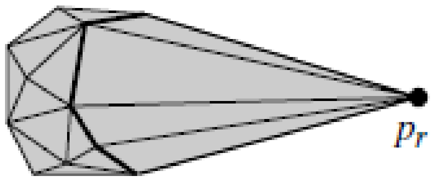
- In a generic step of the algorithm we have to add the point p_r to the convex hull of $CH(P_{r-1})$
- → transform $CH(P_{r-1})$ to $CH(P_r)$
  - →if p_r in $CH(P_{r-1})$ or on it's boundary => $CH(P_r) = CH(P_{r-1})$
  - →else p_r outside of $CH(P_{r-1})$ adding new point deleting old ones

Convex Hull 3D - Computation





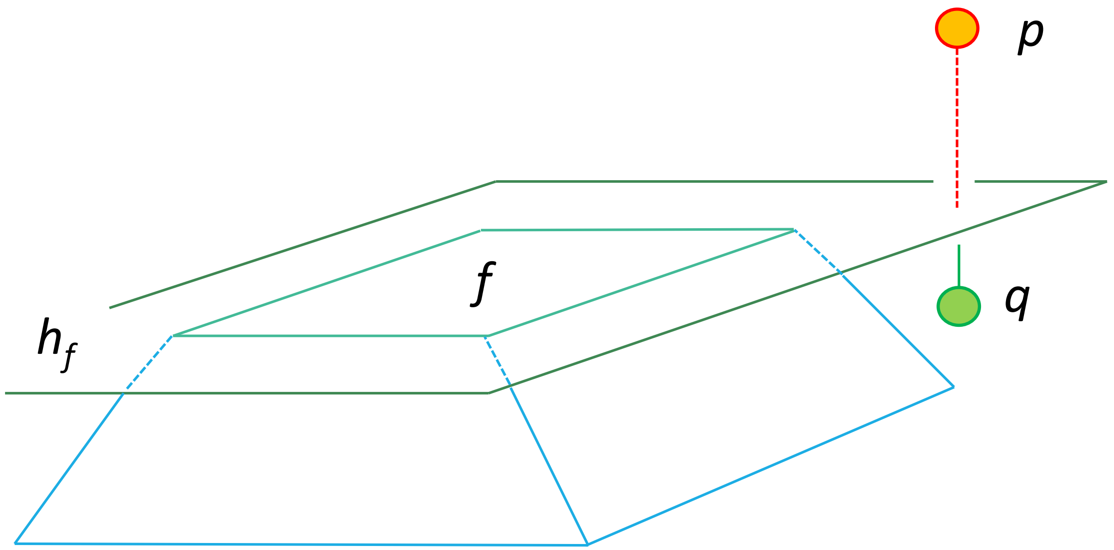
$CH(P_{r-1})$



$CH(P_r)$

Convex Hull 3D - Computation

f is visible from p,
but not from q

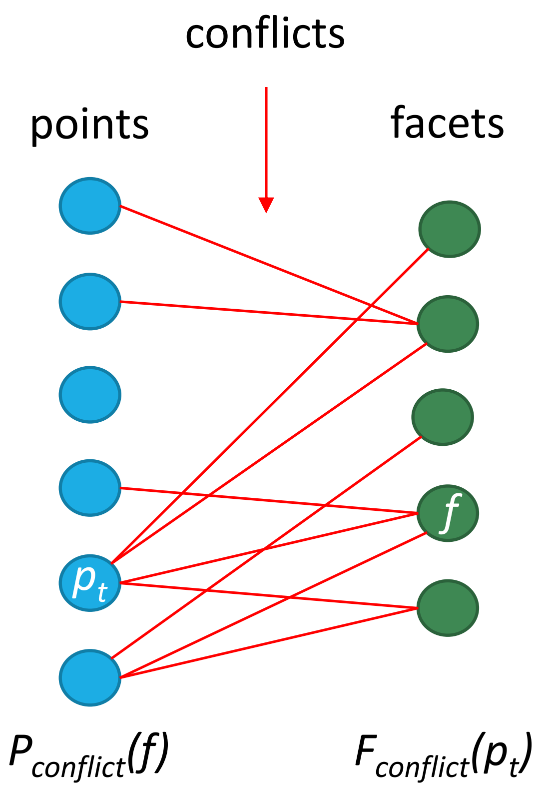


fhwedel UNIVERSITY OF APPLIED SCIENCES

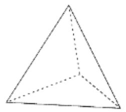
Conflict Graph

Data structure to store

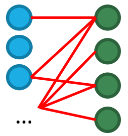
- visible facets $\{f_n, f_m, ..\}$ from point p_t
  - as conflicts
- visible points $\{p_n, p_m, ..\}$ from point f_t
  - as conflicts
- methods for finding Conflicts-Partners
  - $P_{conflict}(f)$ – List of Points in Conflict with Face f
  - $F_{conflict}(p_t)$ – List of Faces in Conflict with Point p_t



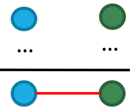
Convex Hull 3D – Algorithmen Overview



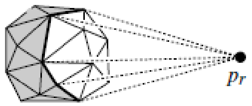
- Starting with a Tetrahedron
  - 4 non-Coplanar points of the Set P
- Compute random permutation for remaining points



- Initialize the Conflict-Graph with Tetrahedron & remaining points
- Loop over all remaining points
  - Check if Conflict Graph of current point is Empty



    - True: continue with next point (point is inside current C. Hull)
    - False:



      - Delete Faces & Build up the Horizon
      - Go along the Horizon & Create new Faces with the current Point
        - Check for new Conflicts between new Face & remaining Points (not c. Point)
  - Delete Old Conflicts

Sample Program

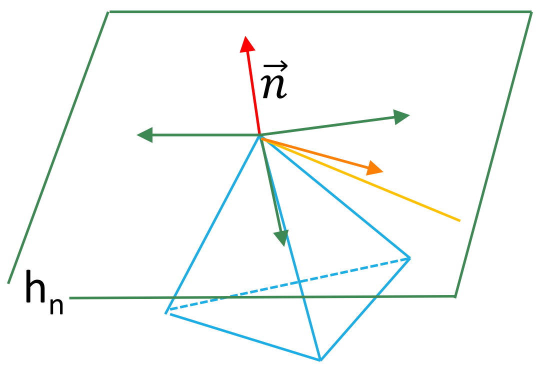
Implementation of a 3D convex hull program

→ 2D & 3D convex hull

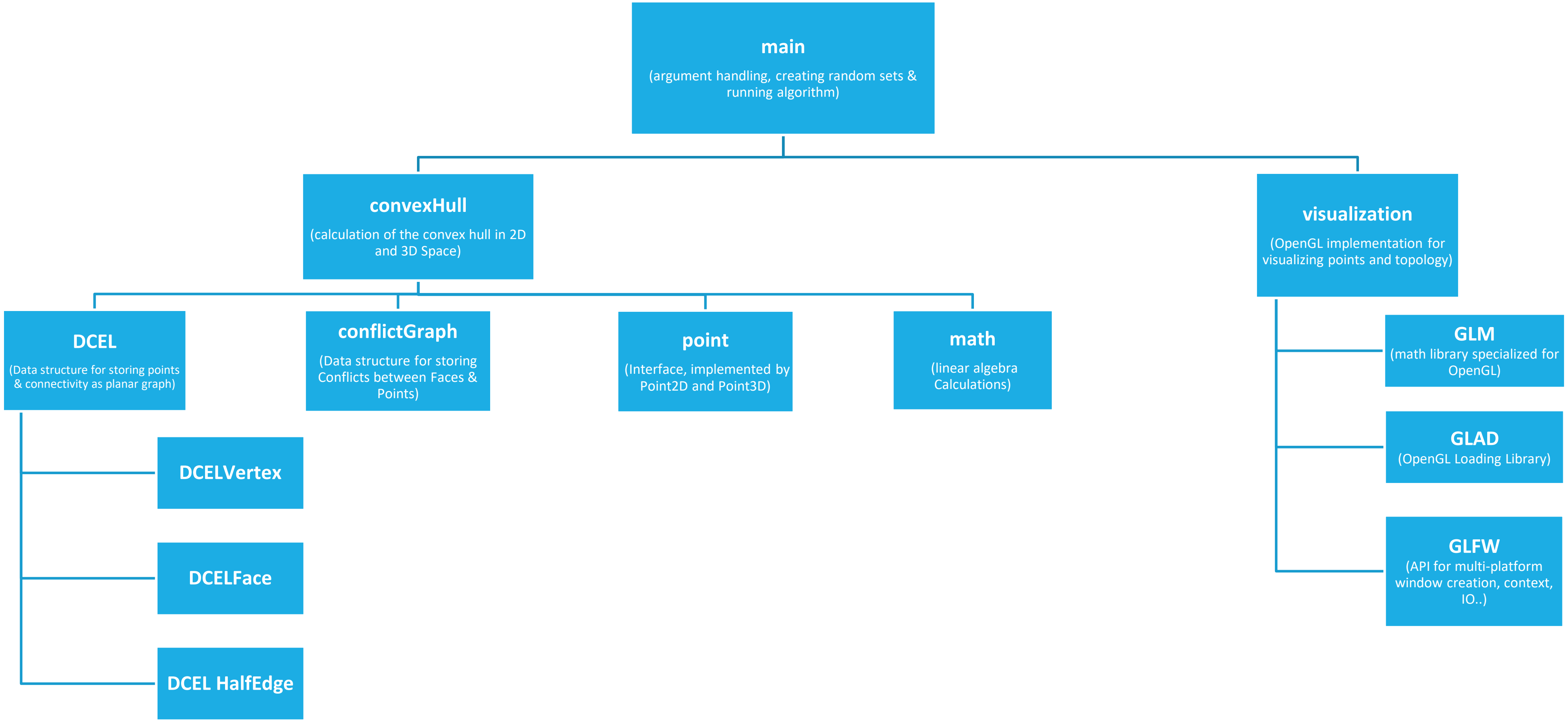
→ parameter arguments as runtime-arguments (-h → help shows all possible arguments)

→ using OpenGL for the visualization

https://github.com/jonassorgenfrei/convexHull3D


$\vec{n}$
h_n

Sample Program



Convex Hull 3D - Conclusion

- running time (expected) **O(n log n)**
- generalizes to **higher dimensions** (optimal in the worst case)
- best **deterministic** convex hull algorithm for odd-dimensional spaces is based on a (quite complicated) de-randomization of this algorithm

- improvements for applications
  - parallelize algorithm → use on Hardware (GPU)

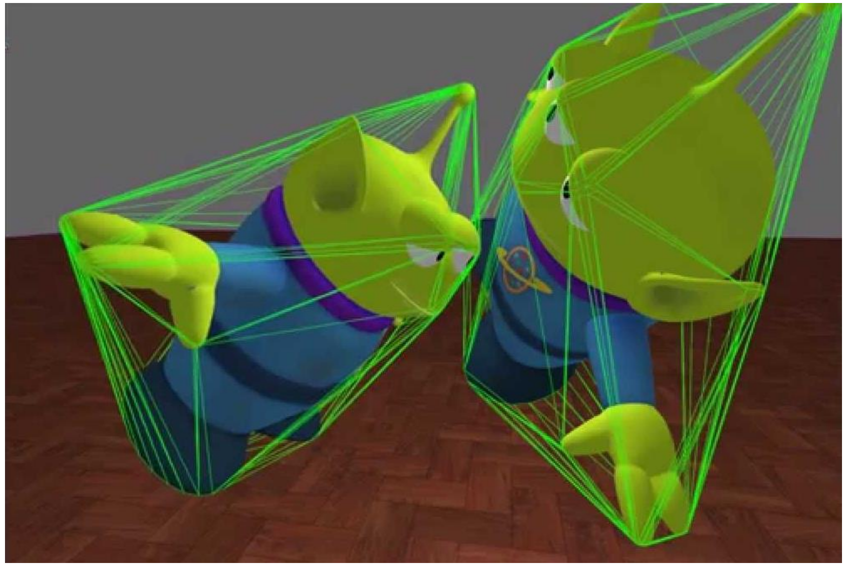
Computer Graphic Applications

Convex Hull 3D

Computer Graphic Applications

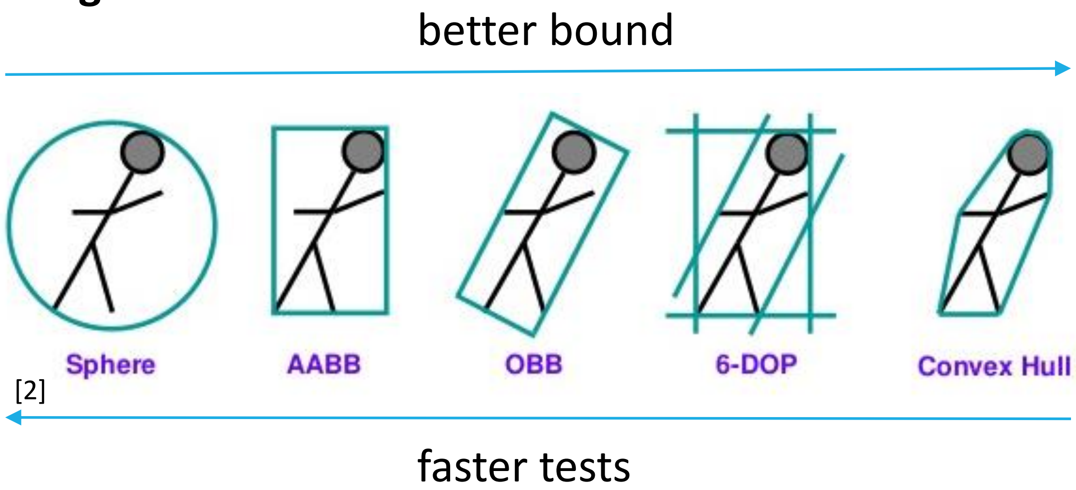
Collision Detection in Computer Animation / Computer Games

- speed up (when result is negative most of the time)
- less costly then actual polyhedron
- better approximation



[1]

Bounding Volumes



[2]

References

Mark de Berg / Otfried Cheong / Marc van Kreveld / Mark Overmars: Computational Geometry, Algorithms and Applications Springer 2008 (3. Aufl.), ISBN 978-3-540-77973-5

Convex Hull in 3D Dimensions – Peter Felkel Fel CTU Prague – Version 23.10.2014

GPU accelerated Convex Hull Computation - Min Tanga, Jie-yi Zhaoa, Ruo-feng Tonga, Dinesh Manochab

http://www.cs.ucr.edu/~eldawy/18SCS133/

https://www.sidefx.com/community/h17-launch/

http://www.cs.sfu.ca/~binay/813.2011/DCEL.pdf

Additional Ref

D. R. Chand and S. S. Kapur. An algorithm for convex polytopes. *J. ACM*,17:78–86, 1970.

B. Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete Comput. Geom.*, 10:377–409, 1993

K. L. Clarkson and P.W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989

M. Kallay. The complexity of incremental convex hull algorithms in *Rd*. *Inform. Process. Lett.*, 19:197, 1984

C. O' 'Du'nlaing and C. K. Yap. A "retraction" method for planning the motion of a disk. *J. Algorithms*, 6:104–111, 1985

F. P. Preparata and S. J. Hong. Convex hulls of finite sets of points in two and three dimensions. *Commun. ACM*, 20:87–93, 1977.

F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, 1985

R. Seidel. *Output-Size Sensitive Algorithms for Constructive Problems in Computational Geometry*. Ph.D. thesis, Dept. Comput. Sci., Cornell Univ., Ithaca, NY, 1986. Technical Report TR 86-784

External Graphics

Slide 3

[1] https://www.techiexpert.com/future-proof-clients-portfolios-robotics-ai/

[2] https://gisgeography.com/what-gis-geographic-information-systems/

[3] http://www.imperial.ac.uk/continuing-professional-development/short-courses/eng/electrical/cmos/

[4] https://ocw.mit.edu/courses/mechanical-engineering/2-158j-computational-geometry-spring-2003/

[5] https://www.cbronline.com/what-is/what-is-a-database-4917209/

[6] https://techterms.com/definition/rendering

[7] http://www.zbrushcentral.com/printthread.php?t=168180&pp=15&page=16

Slide 13

http://diskhkme.blogspot.com/2015/10/convex-hull-algorithm-in-unity-2-3d.html

Slide 14

https://upload.wikimedia.org/wikipedia/commons/0/02/Icosidodecahedron.png

Slide 17

https://commons.wikimedia.org/wiki/File:Truncated_cuboctahedron,_ball-and-stick,_triangles.png

External Graphics

Slide 33

http://www.cgrecord.net/2016/02/deadpool-vfx-breakdown.html

Slide 35

[1] https://www.youtube.com/watch?v=O4FC5rxEqGk

[2] https://www.slideshare.net/oiotoxt/collision-detection-ooxv0p2

Slide 36

https://www.sidefx.com/company/press/