Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

05: BROAD PHASE COLLISION DETECTION

01/02/2016

RECAP: TWO-PHASE COLLISION DETECTION

```
detect(t_curr, {O_0, ..., O_N-1})
    for t ← t_prev to t_curr in steps of Δt_d
        for each object O_i ∈ {O_0, ..., O_N-1}
            move O_i to its position at time t
        for each object O_i ∈ {O_0, ..., O_N-1}
            for each object O_j ∈ {O_i+1, ..., O_N-1}
            if (O_i penetrates O_j)
                collision occurs at simulation time t
    t_prev ← t_curr
```

1. All-pairs weakness
2. Pair Processing weakness
3. Fixed timestep weakness

Naïve collision detection problems [Hubbard 1993]

In order to address the above issues, collision detection is often broken down further (two phase collision detection)

[Hubbard93] **Interactive Collision Detection.** P. Hubbard, in Proceedings of IEEE Symposium on Research Frontiers in Virtual Reality, 1993

BROAD-PHASE COLLISION DETECTION

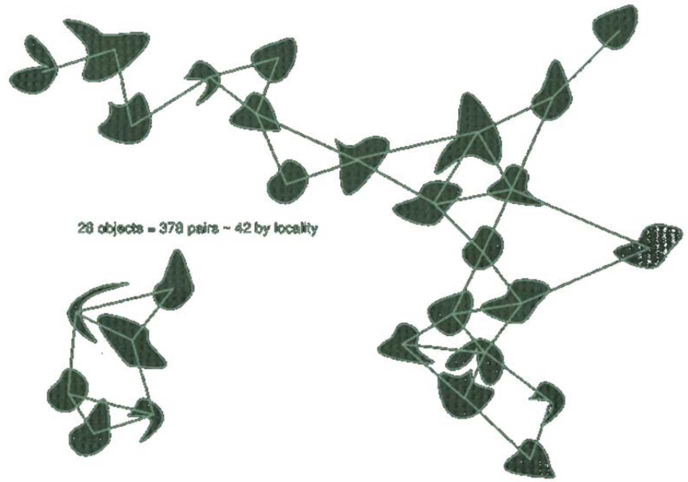
Main Purpose: quickly prune away pairs of objects from more detailed collision/contact processing

Faster than $O(N^2)$ performance achieved by exploiting certain features typically present in animation data

- Locality
- Coherency
- Kinematic knowledge

Typical solutions

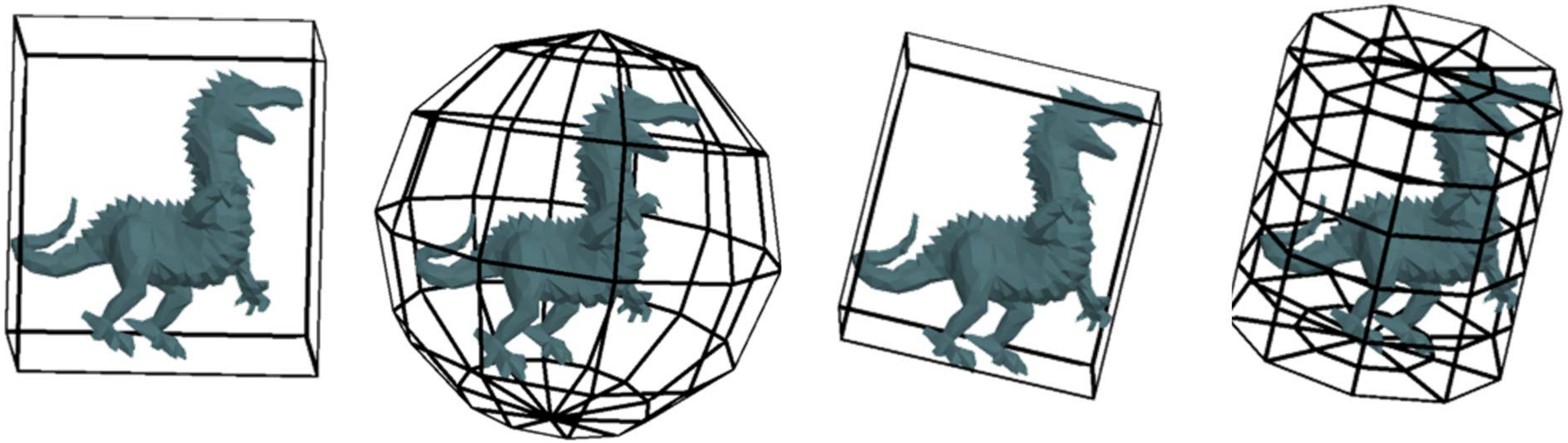
I. Bounding volumes
II. Spatial subdivision
III. Sweep and sort / Sweep and Prune
IV. Space-Time Bounds



Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

I. BOUNDING VOLUMES

BOUNDING VOLUMES



Key issues: representation | construction | update | collision detection

AXIS-ALIGNED BOUNDING BOX (AABB)

Definition: Box with edges that always align to the major axes of the coordinate system

Representation: 6 floats (limits in each dimension)

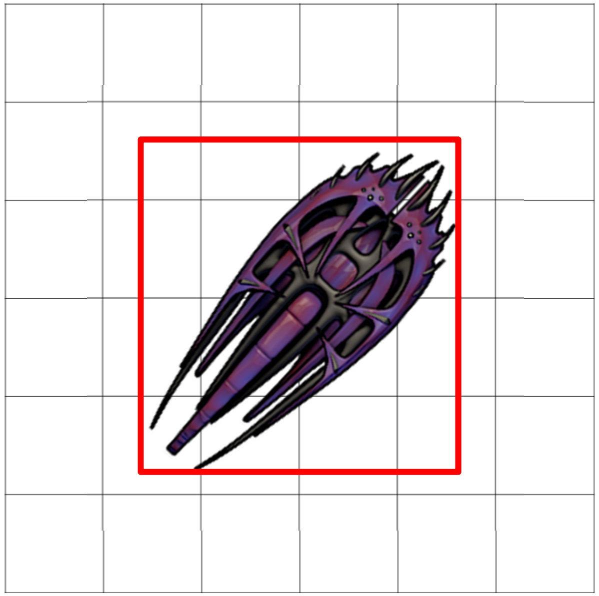
Creation: Find min/max in x, y, z

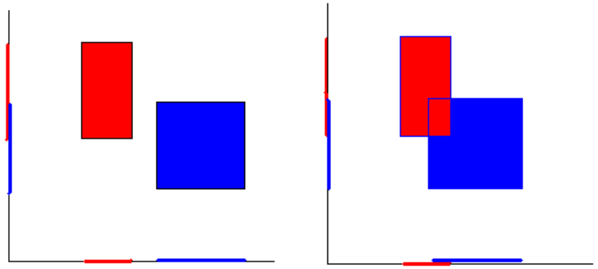
Collision check: Get intervals of projections on each coordinate axis for every body and check if these overlap; 3 x 1d Interval testing

Advantages: Computationally efficient

Disadvantages:

- Unsatisfying fill efficiency
- Not invariant to rotation:
  - *requires dynamic update of AABB OR use very large boxes enclosing object in all orientations*

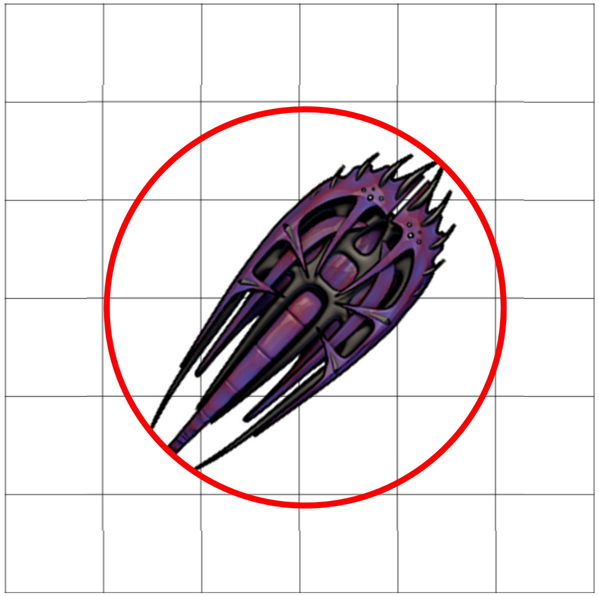




Spaceship image: free sprite by millionthvector (http://millionthvector.blogspot.ie/2013/07/free-alien-top-down-spaceship-sprites.html)

BOUNDING SPHERE

Definition: the smallest sphere that encloses object



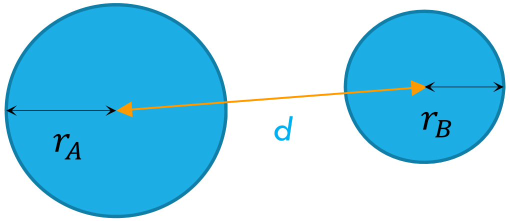
Representation: 4 floats (centre, radius)

Creation: Find minimum enclosing sphere

- min+max (xyz); center; min radius

Collision check

- Let $d = distance\ between\ centres\ of\ 2\ spheres$
- Then spheres are colliding $IF\left(d < (r_A + r_B)\right)$



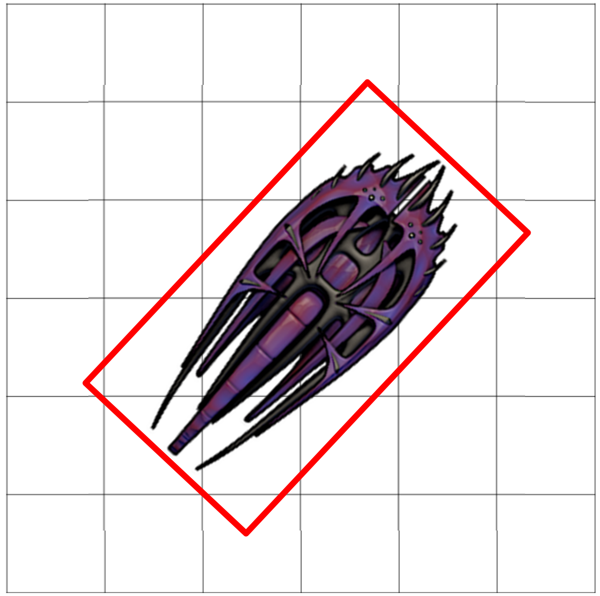
Advantages:

- Invariant to rotation; computationally efficient
- Update is simply a translation

Disadvantages: Not good fit for long/flat objects

ORIENTED BOUNDING BOX (OBB)

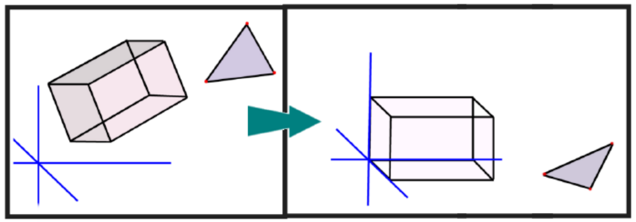
Definition: Box with edges that align to object such that it fits optimally in terms of fill efficiency



Representation: 15 floats (position, orientation, extents)

Creation: Manual or by PCA based fitting (see later)

Collision test: Map to box reference coords OR Use separating axis theorem



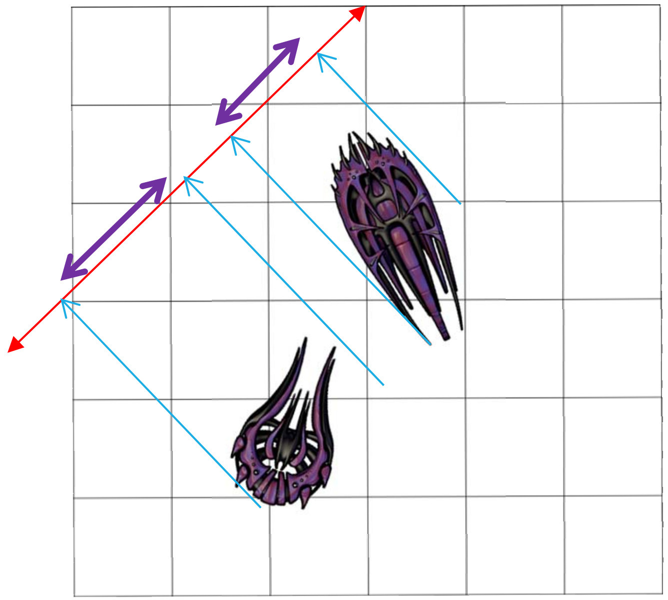
Advantages:

- Invariant to rotation
- Tighter bounds than AABB and spheres

Disadvantages:

- Computationally more expensive to generate
- Harder to implement

SEPARATING AXIS THEOREM



Separating Axis:

- An axis on which the projections of two polytopes don't overlap.
  - But there are an infinite number of potential axes; we can't try them all

Separating Axis Theorem:

- Two convex polytopes are disjoint **if and only if** there exists a separating axis orthogonal to a face of either polytope or orthogonal to an edge from each polytope.
  - Reduces number of potential axes we need to test

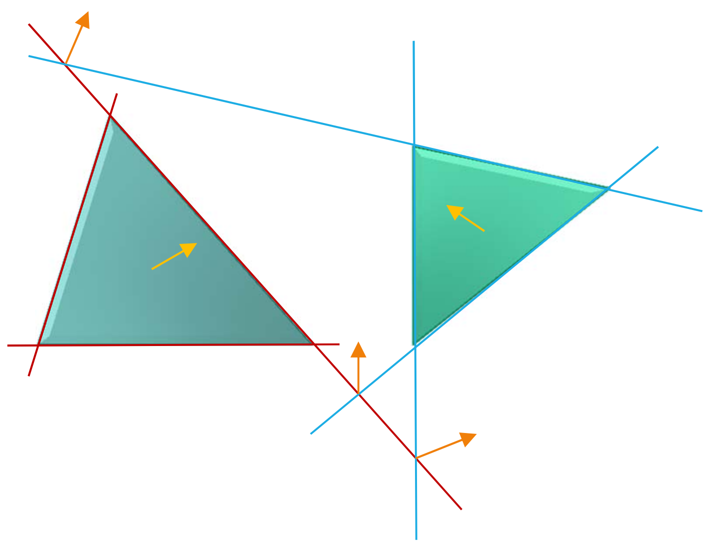
SEPARATING AXIS THEOREM

Two convex polytopes are disjoint **IFF** there exists a separating axis

- orthogonal to a face of either polytope

OR

- orthogonal to an edge from each polytope.



IMPLICATIONS OF THEOREM

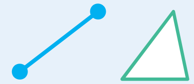
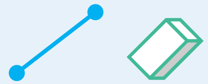
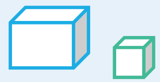
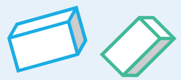
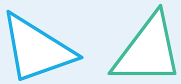
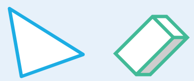
- Given two generic polytopes, each with E edges and F faces, number of candidate axes to test is:

$$2F + E^2$$

- OBBs have only $E = 3$ distinct edge directions, and only $F = 3$ distinct face normals. OBBs need at most 15 axis tests.

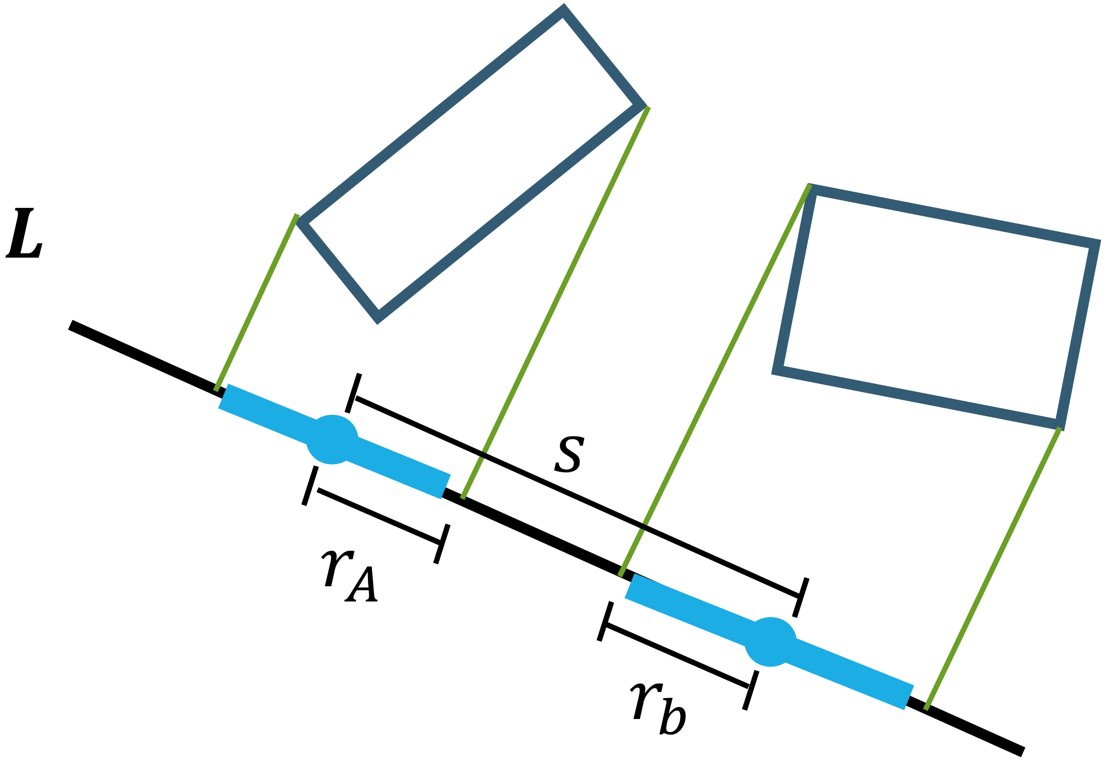
- Because edge directions and normals each form orthogonal frames, the axis tests are rather simple.

NUMBER OF AXES TO TEST

3D Objects	Face dirs (A)	Face dirs (B)	Edge dirs (AxB)	Total	
Segment–Tri	0	1	1x3	4	
Segment–OBB	0	3	1x3	6	
AABB–AABB	3	0(3)	0(3x0)	3	
OBB–OBB	3	3	3x3	15	
Tri–Tri	1	1	3x3	11	
Tri–OBB	1	3	3x3	13	

From Collisions using Separating Axis Test. C. Ericsson. GDC 07.
http://realtimecollisiondetection.net/pubs/

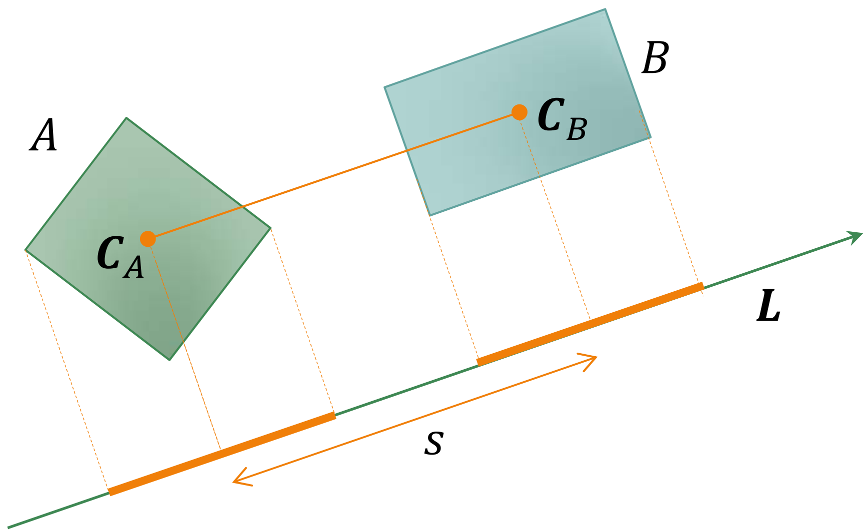
OBB OVERLAP TEST: AXIS TEST



- L is a separating axis IFF $s > r_A + r_B$
- r_A and r_B are the the half-length of interval spanned by the projection of the box onto L (somewhat like the radius)

OBB OVERLAP TEST: AXIS TEST DETAILS

- Box centers project to interval midpoints, so midpoint separation s is length of vector T's image.

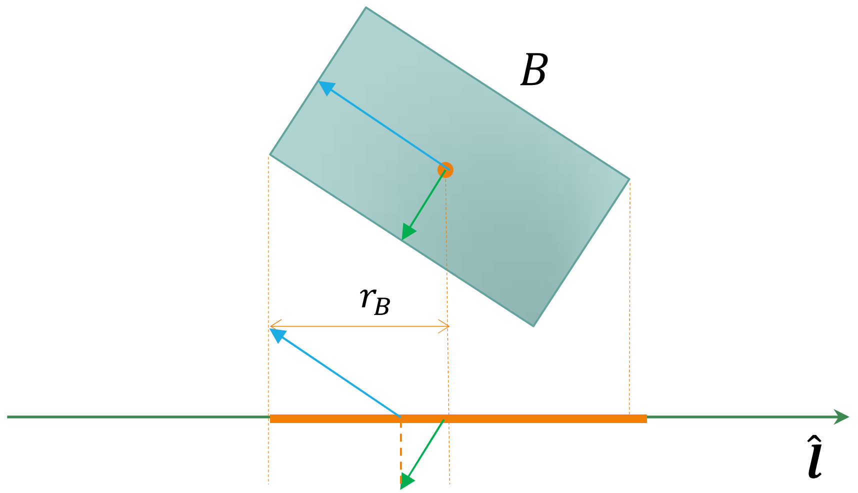


$$s = \left|(\boldsymbol{C}_A - \boldsymbol{C}_B) \cdot \hat{\boldsymbol{l}}\right|$$

$\hat{\boldsymbol{l}}$ is a normalised vector in direction of candidate axis $\boldsymbol{L}$
$\boldsymbol{C}_A$ and $\boldsymbol{C}_B$ are the centres of respective OBB's

OBB OVERLAP TEST: AXIS TEST DETAILS

Half-length of interval is sum of box axis images.



OBB is typical stored as orientation matrix $\boldsymbol{R}$ and radius along each orthogonal axis e_1, e_2, e_3

$$r_B = e_1^B\left|\boldsymbol{R}_1^B \cdot \hat{\boldsymbol{l}}\right| + e_2^B\left|\boldsymbol{R}_2^B \cdot \hat{\boldsymbol{l}}\right| + e_3^B\left|\boldsymbol{R}_3^B \cdot \hat{\boldsymbol{l}}\right|$$

Note that $\boldsymbol{R}_i$ are the column vectors of $\boldsymbol{R}$

The superscript B indicates that this is an element of body B

SEPARATING AXIS TESTS

Table 4.1 The 15 separating axis tests needed to determine OBB-OBB intersection. Superscripts indicate which OBB the value comes from.

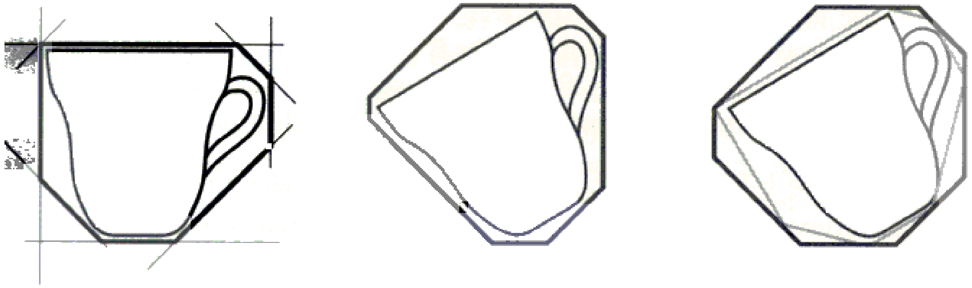
L	$\lvert T \cdot L \rvert$	r_A	r_B
$\mathbf{u}_0^A$	$\lvert \mathbf{t}_0 \rvert$	$\mathbf{e}_0^A$	$\mathbf{e}_0^B \lvert r_{00} \rvert + \mathbf{e}_1^B \lvert r_{01} \rvert + \mathbf{e}_2^B \lvert r_{02} \rvert$
$\mathbf{u}_1^A$	$\lvert \mathbf{t}_1 \rvert$	$\mathbf{e}_1^A$	$\mathbf{e}_0^B \lvert r_{10} \rvert + \mathbf{e}_1^B \lvert r_{11} \rvert + \mathbf{e}_2^B \lvert r_{12} \rvert$
$\mathbf{u}_2^A$	$\lvert \mathbf{t}_2 \rvert$	$\mathbf{e}_2^A$	$\mathbf{e}_0^B \lvert r_{20} \rvert + \mathbf{e}_1^B \lvert r_{21} \rvert + \mathbf{e}_2^B \lvert r_{22} \rvert$
$\mathbf{u}_0^B$	$\lvert \mathbf{t}_0 r_{00} + \mathbf{t}_1 r_{10} + \mathbf{t}_2 r_{20} \rvert$	$\mathbf{e}_0^A \lvert r_{00} \rvert + \mathbf{e}_1^A \lvert r_{10} \rvert + \mathbf{e}_2^A \lvert r_{20} \rvert$	$\mathbf{e}_0^B$
$\mathbf{u}_1^B$	$\lvert \mathbf{t}_0 r_{01} + \mathbf{t}_1 r_{11} + \mathbf{t}_2 r_{21} \rvert$	$\mathbf{e}_0^A \lvert r_{01} \rvert + \mathbf{e}_1^A \lvert r_{11} \rvert + \mathbf{e}_2^A \lvert r_{21} \rvert$	$\mathbf{e}_1^B$
$\mathbf{u}_2^B$	$\lvert \mathbf{t}_0 r_{02} + \mathbf{t}_1 r_{12} + \mathbf{t}_2 r_{22} \rvert$	$\mathbf{e}_0^A \lvert r_{02} \rvert + \mathbf{e}_1^A \lvert r_{12} \rvert + \mathbf{e}_2^A \lvert r_{22} \rvert$	$\mathbf{e}_2^B$
$\mathbf{u}_0^A \times \mathbf{u}_0^B$	$\lvert \mathbf{t}_2 r_{10} - \mathbf{t}_1 r_{20} \rvert$	$\mathbf{e}_1^A \lvert r_{20} \rvert + \mathbf{e}_2^A \lvert r_{10} \rvert$	$\mathbf{e}_1^B \lvert r_{02} \rvert + \mathbf{e}_2^B \lvert r_{01} \rvert$
$\mathbf{u}_0^A \times \mathbf{u}_1^B$	$\lvert \mathbf{t}_2 r_{11} - \mathbf{t}_1 r_{21} \rvert$	$\mathbf{e}_1^A \lvert r_{21} \rvert + \mathbf{e}_2^A \lvert r_{11} \rvert$	$\mathbf{e}_0^B \lvert r_{02} \rvert + \mathbf{e}_2^B \lvert r_{00} \rvert$
$\mathbf{u}_0^A \times \mathbf{u}_2^B$	$\lvert \mathbf{t}_2 r_{12} - \mathbf{t}_1 r_{22} \rvert$	$\mathbf{e}_1^A \lvert r_{22} \rvert + \mathbf{e}_2^A \lvert r_{12} \rvert$	$\mathbf{e}_0^B \lvert r_{01} \rvert + \mathbf{e}_1^B \lvert r_{00} \rvert$
$\mathbf{u}_1^A \times \mathbf{u}_0^B$	$\lvert \mathbf{t}_0 r_{20} - \mathbf{t}_2 r_{00} \rvert$	$\mathbf{e}_0^A \lvert r_{20} \rvert + \mathbf{e}_2^A \lvert r_{00} \rvert$	$\mathbf{e}_1^B \lvert r_{12} \rvert + \mathbf{e}_2^B \lvert r_{11} \rvert$
$\mathbf{u}_1^A \times \mathbf{u}_1^B$	$\lvert \mathbf{t}_0 r_{21} - \mathbf{t}_2 r_{01} \rvert$	$\mathbf{e}_0^A \lvert r_{21} \rvert + \mathbf{e}_2^A \lvert r_{01} \rvert$	$\mathbf{e}_0^B \lvert r_{12} \rvert + \mathbf{e}_2^B \lvert r_{10} \rvert$
$\mathbf{u}_1^A \times \mathbf{u}_2^B$	$\lvert \mathbf{t}_0 r_{22} - \mathbf{t}_2 r_{02} \rvert$	$\mathbf{e}_0^A \lvert r_{22} \rvert + \mathbf{e}_2^A \lvert r_{02} \rvert$	$\mathbf{e}_0^B \lvert r_{11} \rvert + \mathbf{e}_1^B \lvert r_{10} \rvert$
$\mathbf{u}_2^A \times \mathbf{u}_0^B$	$\lvert \mathbf{t}_1 r_{00} - \mathbf{t}_0 r_{10} \rvert$	$\mathbf{e}_0^A \lvert r_{10} \rvert + \mathbf{e}_1^A \lvert r_{00} \rvert$	$\mathbf{e}_1^B \lvert r_{22} \rvert + \mathbf{e}_2^B \lvert r_{21} \rvert$
$\mathbf{u}_2^A \times \mathbf{u}_1^B$	$\lvert \mathbf{t}_1 r_{01} - \mathbf{t}_0 r_{11} \rvert$	$\mathbf{e}_0^A \lvert r_{11} \rvert + \mathbf{e}_1^A \lvert r_{01} \rvert$	$\mathbf{e}_0^B \lvert r_{22} \rvert + \mathbf{e}_2^B \lvert r_{20} \rvert$
$\mathbf{u}_2^A \times \mathbf{u}_2^B$	$\lvert \mathbf{t}_1 r_{02} - \mathbf{t}_0 r_{12} \rvert$	$\mathbf{e}_0^A \lvert r_{12} \rvert + \mathbf{e}_1^A \lvert r_{02} \rvert$	$\mathbf{e}_0^B \lvert r_{21} \rvert + \mathbf{e}_1^B \lvert r_{20} \rvert$

N.B. the notation in this figure is from Ericson's book and is slightly different from the previous slide

Real-Time Collision Detection by Christer Ericson (Morgan Kaufmann, 2005)

K-DOP

Discrete oriented polytopes:



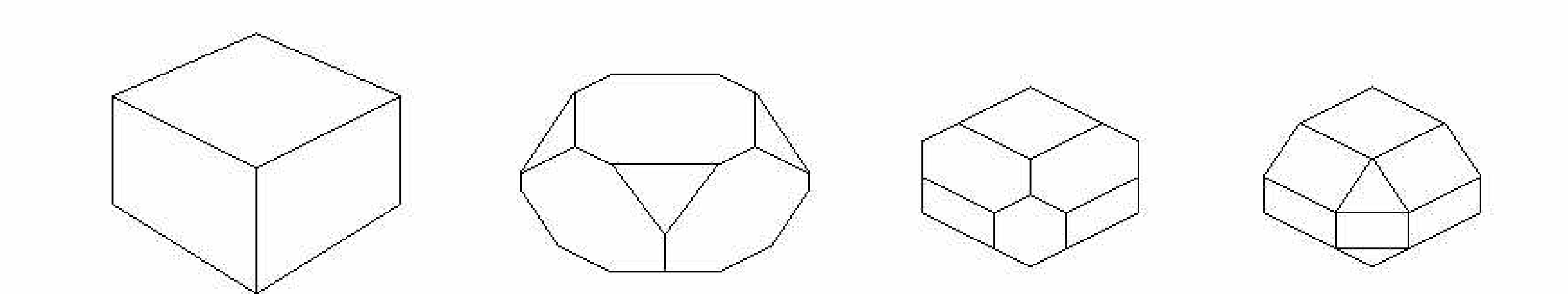
Essentially an extension of AABB/OBBs (Boxes are 6-dops)

- Similar to clipping corners of bounding box

Advantages

- Easy to compute
- Good fill efficiency
- Simple overlap test
  - generalisation of separating axis test for relevant value of k

K-DOP

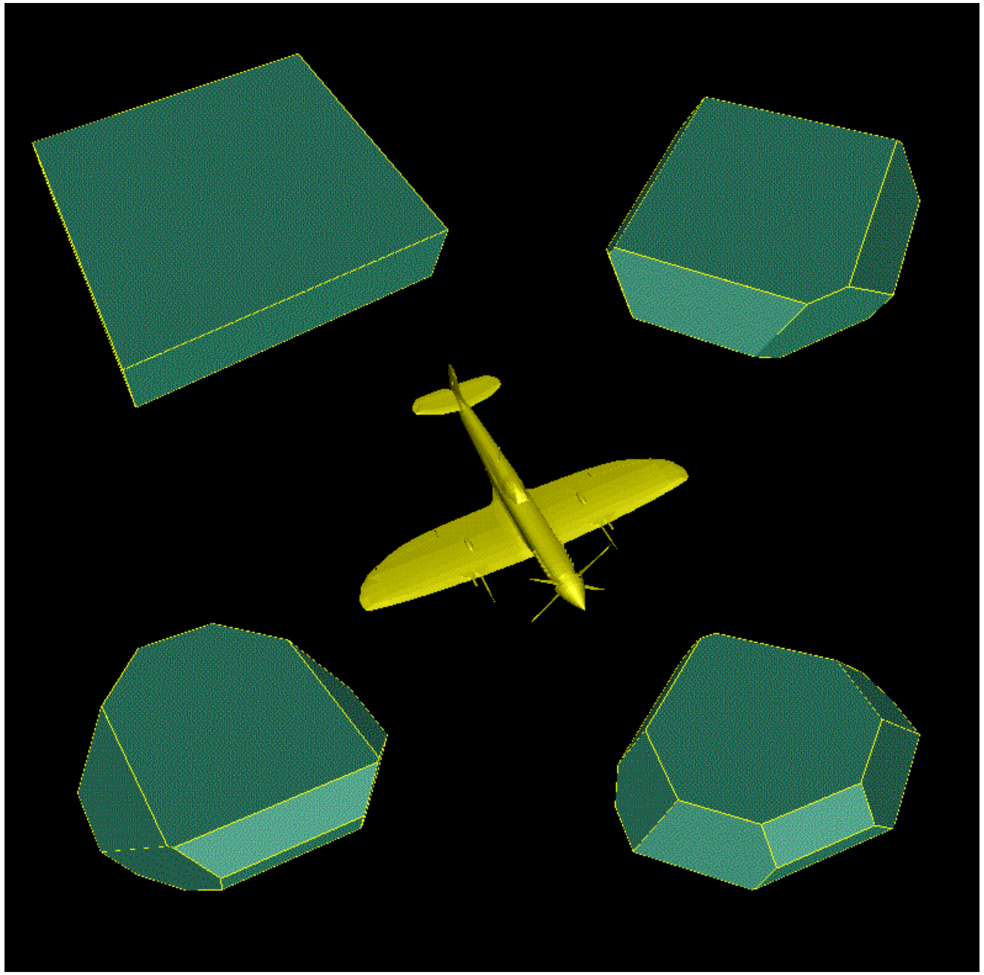


6-DOP	**14-DOP**	**18-DOP**	**26-DOP**
=AABB	(corners)	(edges)	(edges and corners)

K-DOP

6-DOP

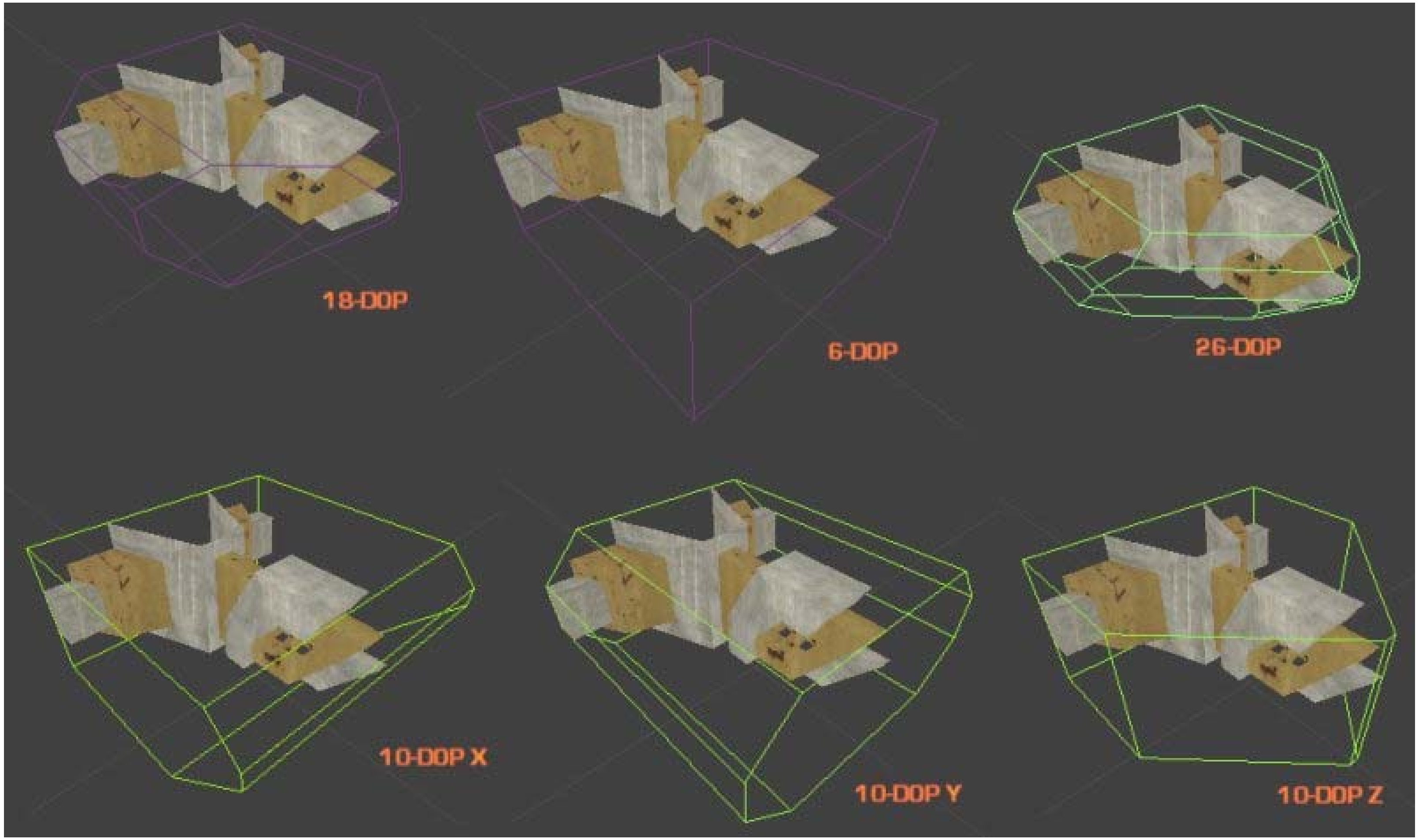
14-DOP



18-DOP

26-DOP

UNREAL 3 ENGINE EXAMPLE



http://udn.epicgames.com/Three/CollisionReference.html#K-DOP

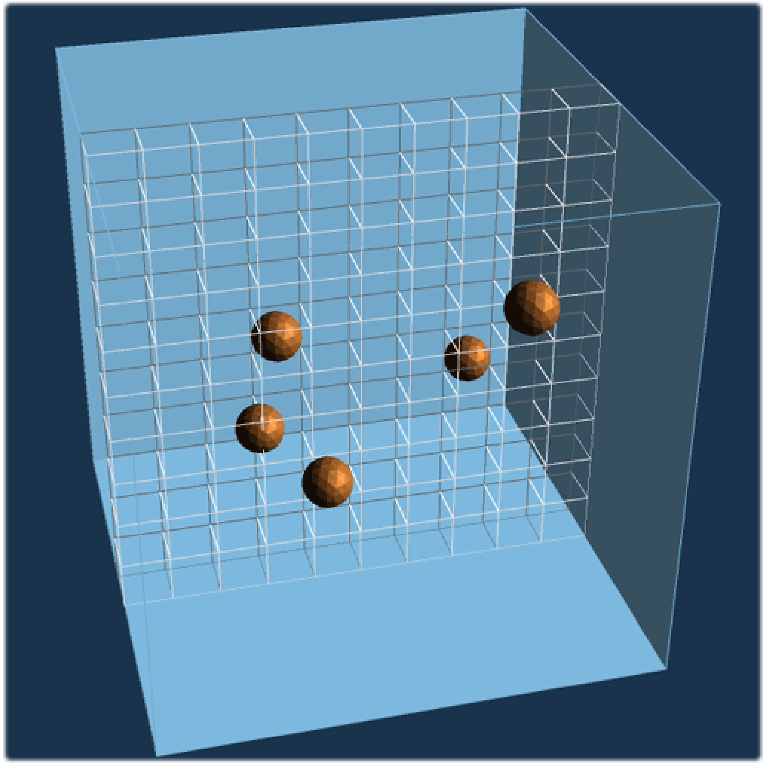
Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

II. SPATIAL SUBDIVISION

SPATIAL SUBDIVISION

e.g. Grid method

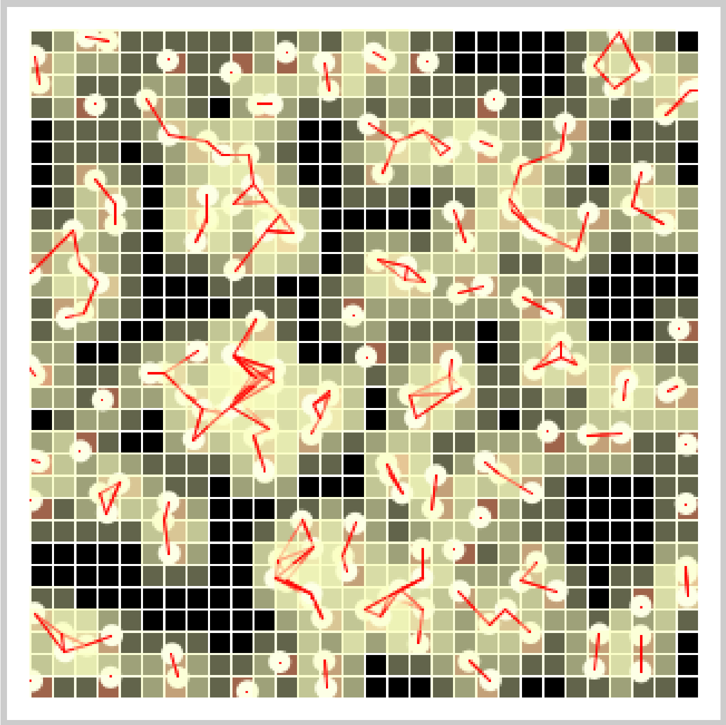
- Split scene up into uniform cells
- Each object keeps a record of grid cells that it overlaps with
- Only perform pair-wise collision test with objects in own cell or cell-neighbours on grid



SPATIAL HASHING

Project 2D/3D domain into 1D hashtable

- 2D/3D points used as keys
- Exploit locality

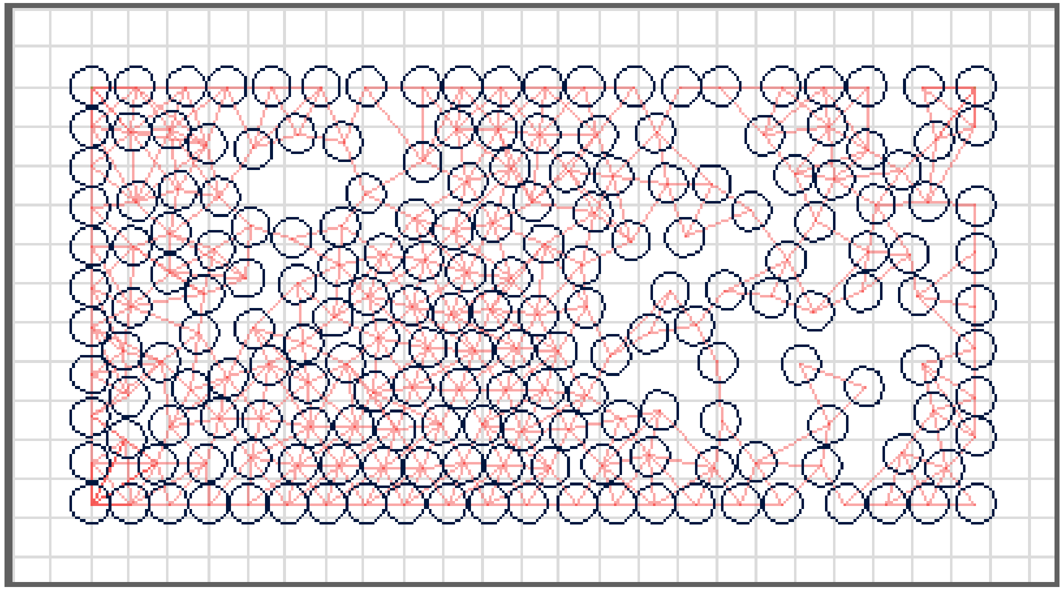


© 2012 Taura J. Greig
http://openprocessing.org/sketch/60117

Optimization of Large-Scale, Real-Time Simulations by Spatial Hashing - E. Hastings, J. Mesit, R.K. Guha
http://www.cs.ucf.edu/~jmesit/publications/scsc%202005.pdf

Perfect Spatial Hashing – S. Lefebvre, H. Hoppe (http://research.microsoft.com/en-us/um/people/hoppe/perfecthash.pdf)

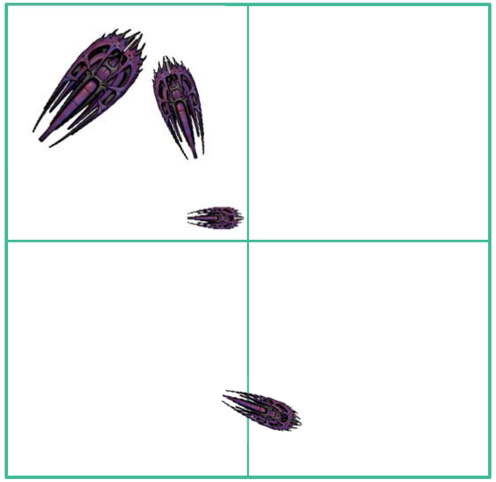
SPATIAL HASHING



http://users.design.ucla.edu/~mflux/p5/hashcollision2/applet/

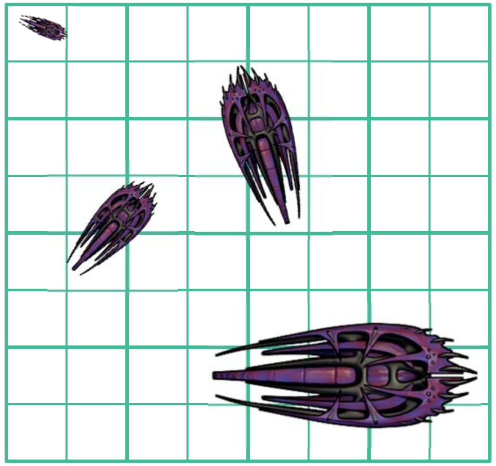
SPATIAL SUBDIVISION

If grid size is too big in relation to objects: too many false positives in narrow phase



If grid size too small: too much work in broad phase

- single object could cover more than 1 grid position



HIERARCHICAL SPATIAL SUBDIVISION

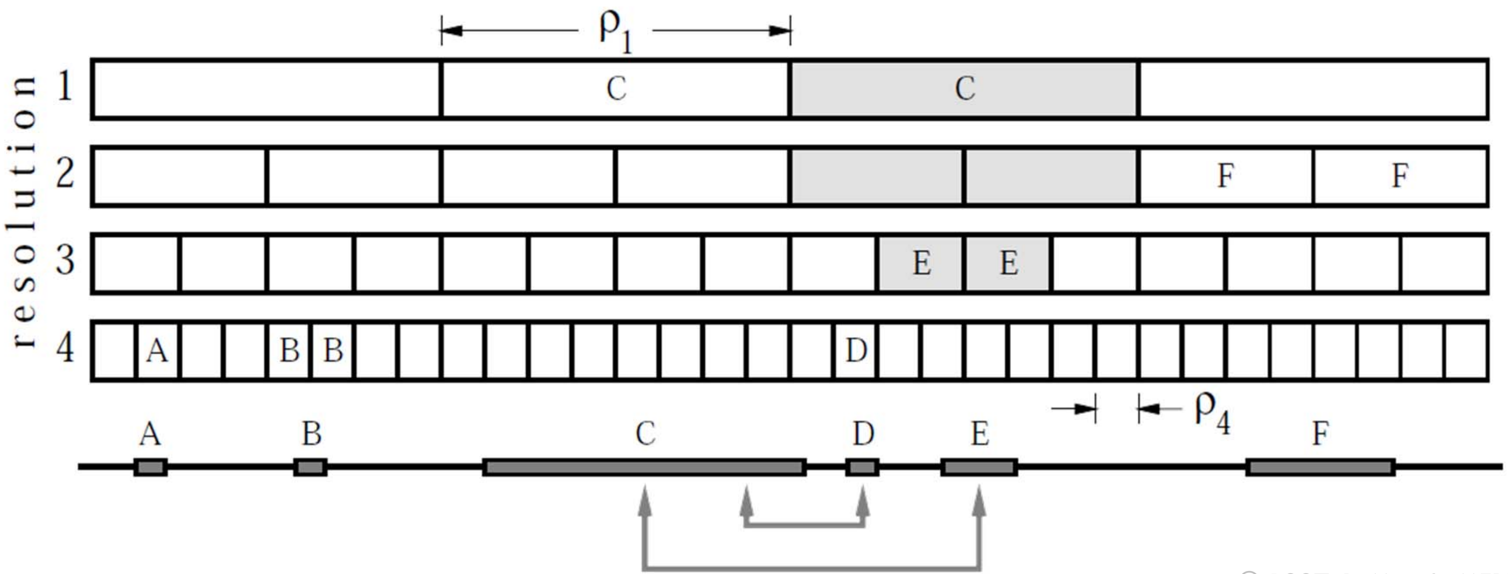


Image © 1997, B. Mirtich, MERL

Adaptive Grid Sizes:

- Boxes are tiled at the minimum resolution where the cells are enough to cover the bounding volume of the object
- If res(X) and res(Y) are the tiling resolutions for X and Y, then Boxes X and Y are close iff they overlap a common cell at resolution min(res(X), res(Y))

For details see "Efficient Algorithms for Two-Phase Collision Detection" Brian Mirtich – TR9723 MERL, Dec, 1997. http://www.merl.com/reports/docs/TR97-23.pdf

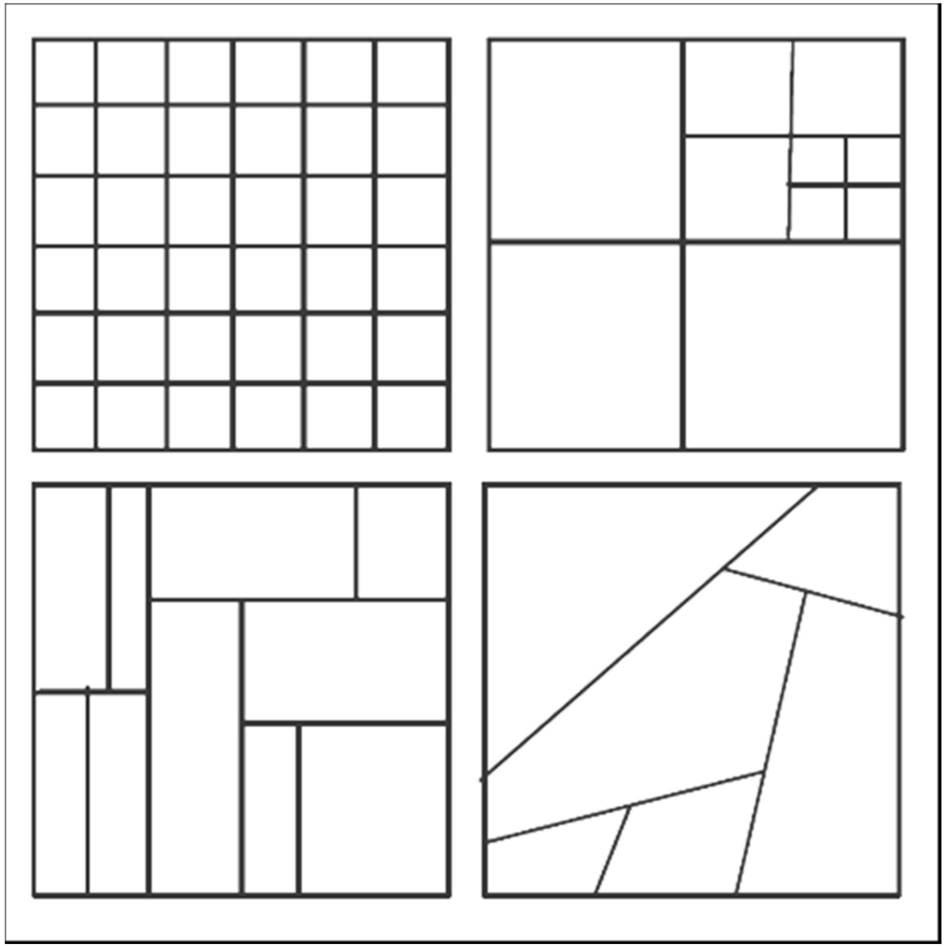
SPATIAL SUBDIVISION SCHEMES

Object independent

- Grid

Object dependent

- Quadtree/Octree
- Kd-tree
- BSP Tree



Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

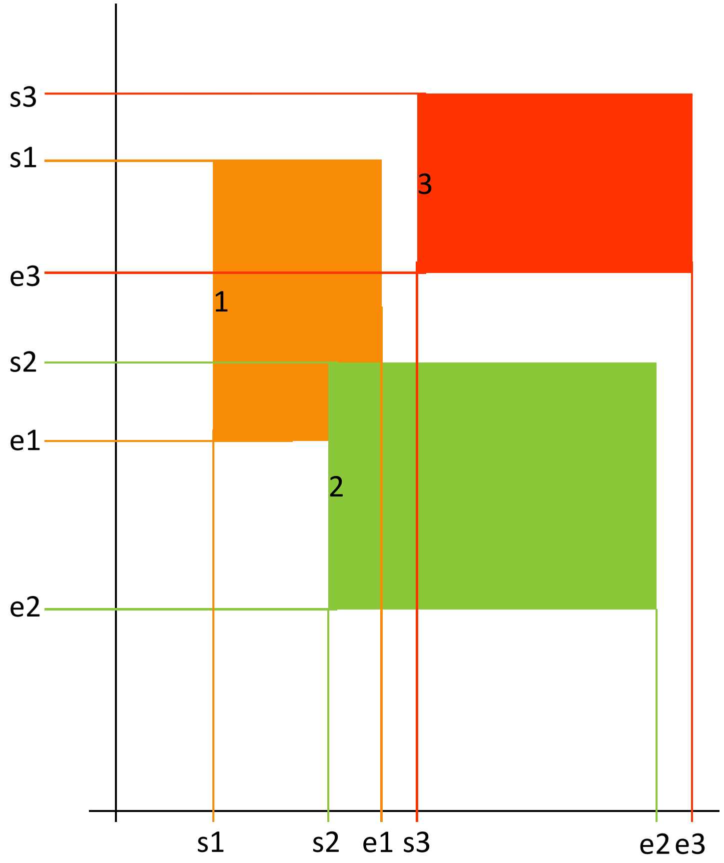
III. SWEEP AND PRUNE

SWEEP AND PRUNE

Create ABBB for each object

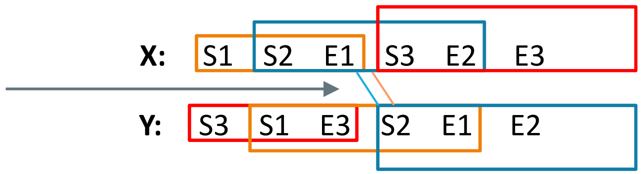
- For each axis,
  - Create sorted list of start and end points of intervals for each box
- Traverse each list
  - Each time a startpoint is reached insert into active list
  - If endpoint is hit remove it from active list
  - If 2 or more objects are active at the same time they overlap in the specified dimension
  - Potential colliding pairs must overlap in all 3 lists

"Early out" if not overlapping in any direction



Exploit coherency e.g. insertion-sort to get log(N) performance

OVERLAP 1,2

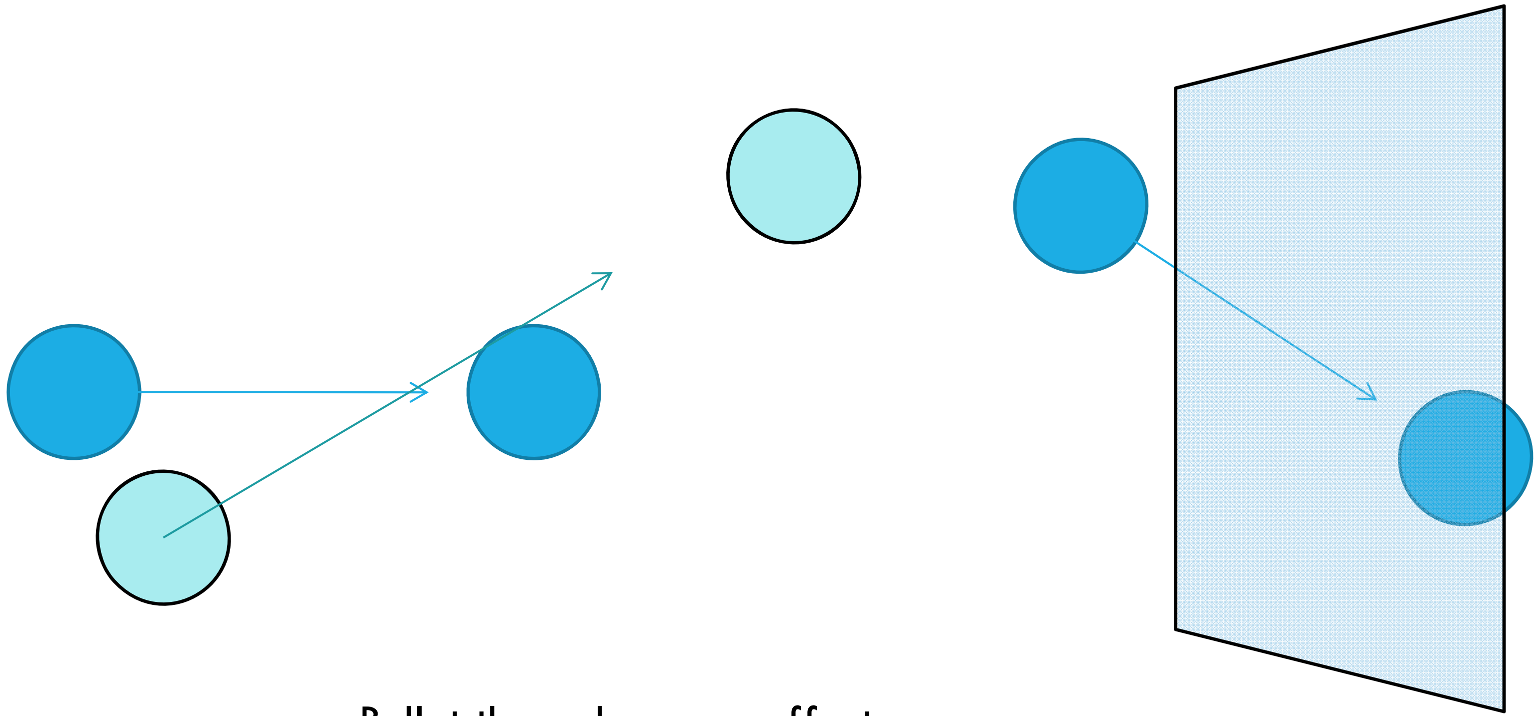


Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

IV. SPACE TIME BOUNDS

RECALL: FIXED TIME-STEP WEAKNESS

Have objects collided?

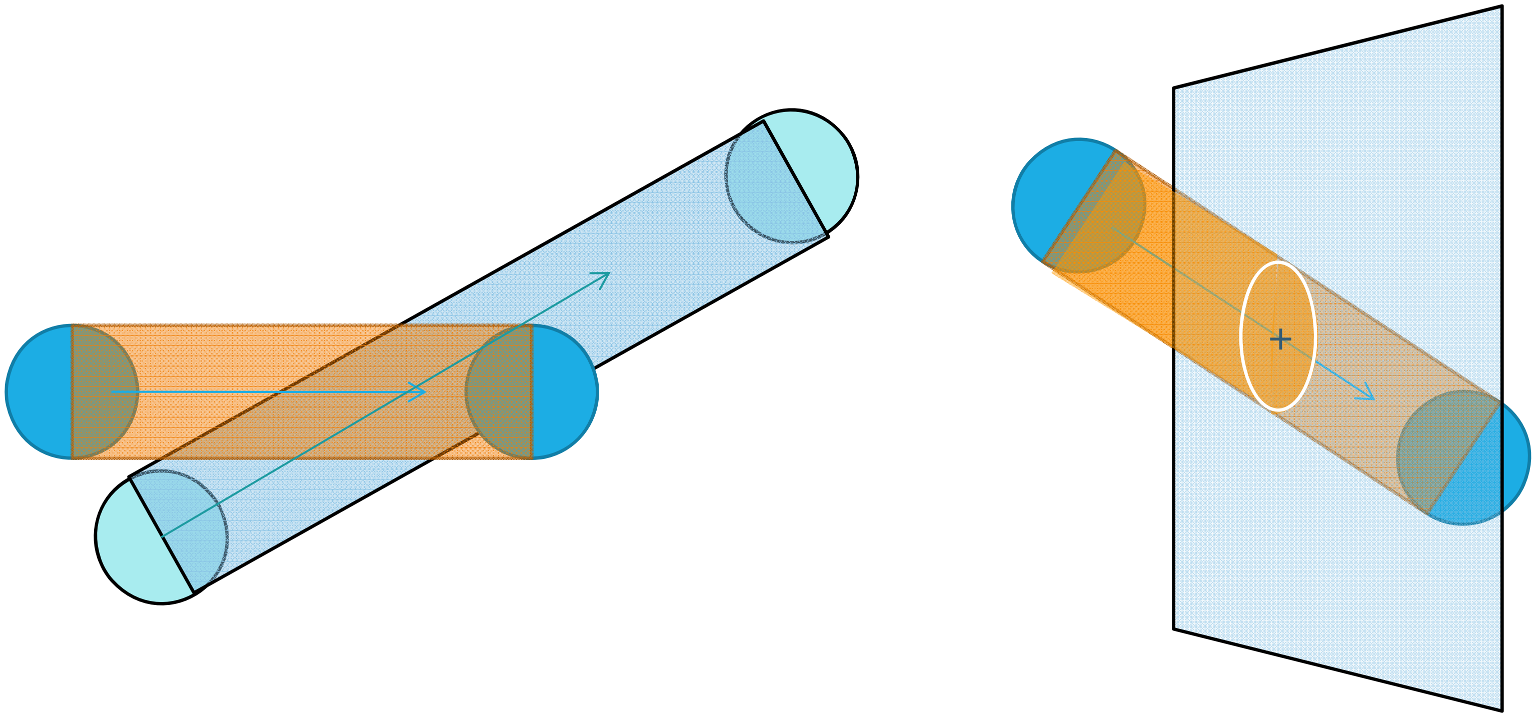


Bullet through paper effect

SOLUTION: SPACE TIME BOUNDS

Continuous collision detection

Sweep tests

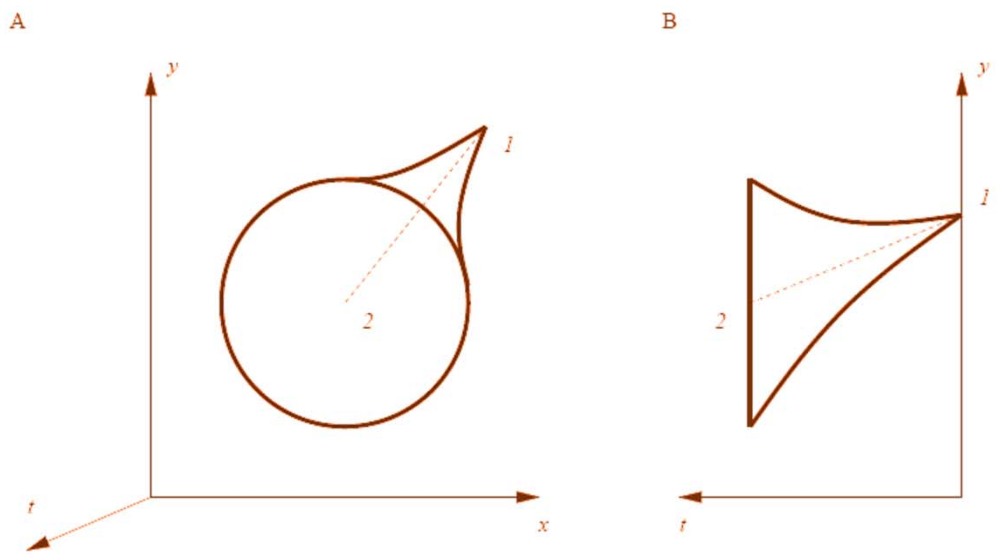


But this assumes constant linear velocity within the timestep

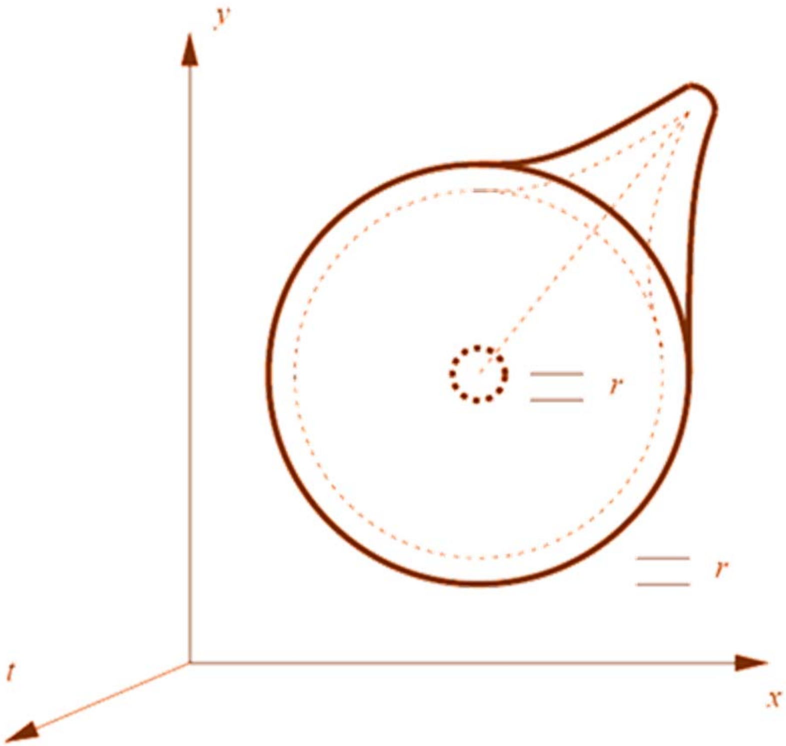
PARABOLIC HORN

A volume encompassing all positions that a particle could be within after a given timestep Δt

- Central axis is calculated from velocity $\boldsymbol{x_2} = \boldsymbol{x_1} + \boldsymbol{v}\Delta t$
- Radius is calculated from acceleration $R(t + \Delta t) \geq |\boldsymbol{a}\Delta t|$



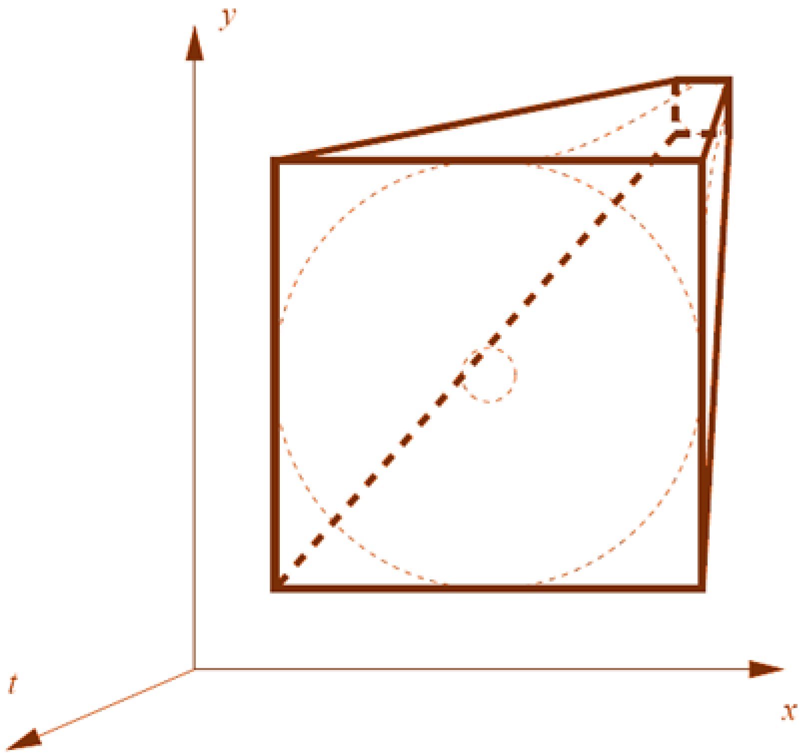
PARABOLIC HORN



For a solid object

Parabolic Horn is grown by r, the radius of bounding sphere

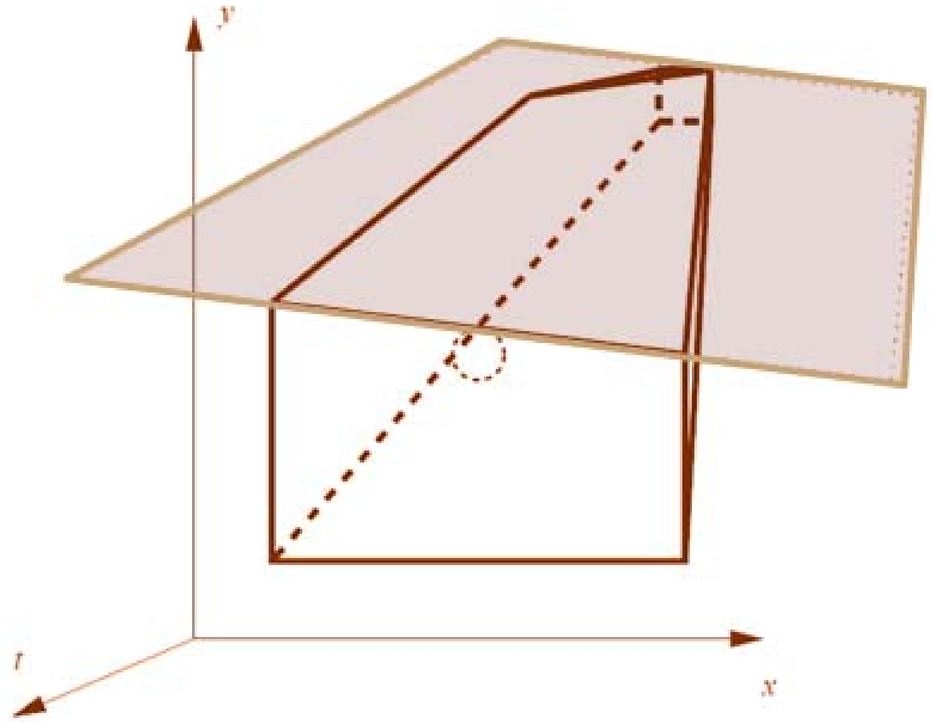
HYPER-TRAPEZOID



Hyper-trapezoid approximation of the parabolic horn

More wasted space but quicker collision check

HYPER-TRAPEZOID

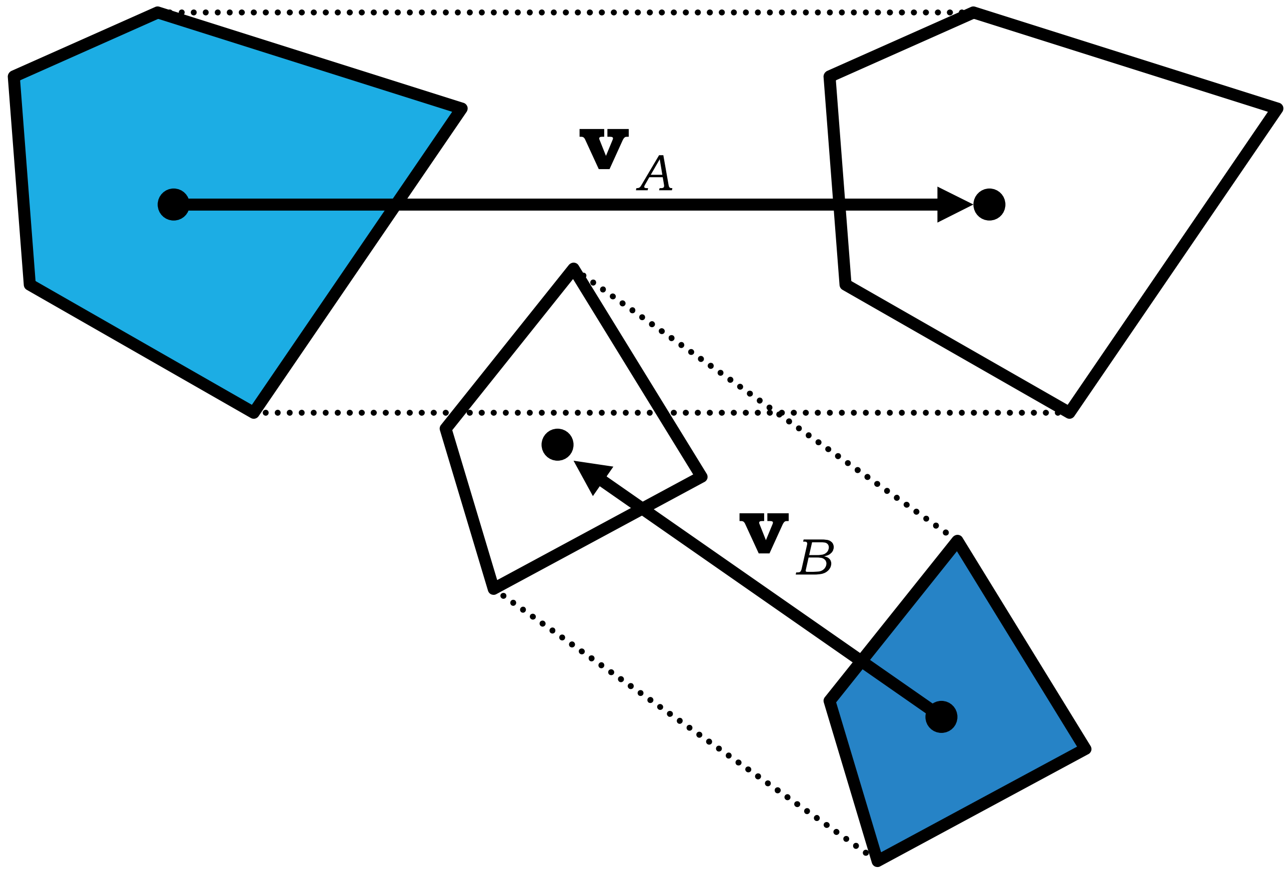


Further culling is possible if we take into account the direction of a

- An example of exploiting kinematic information

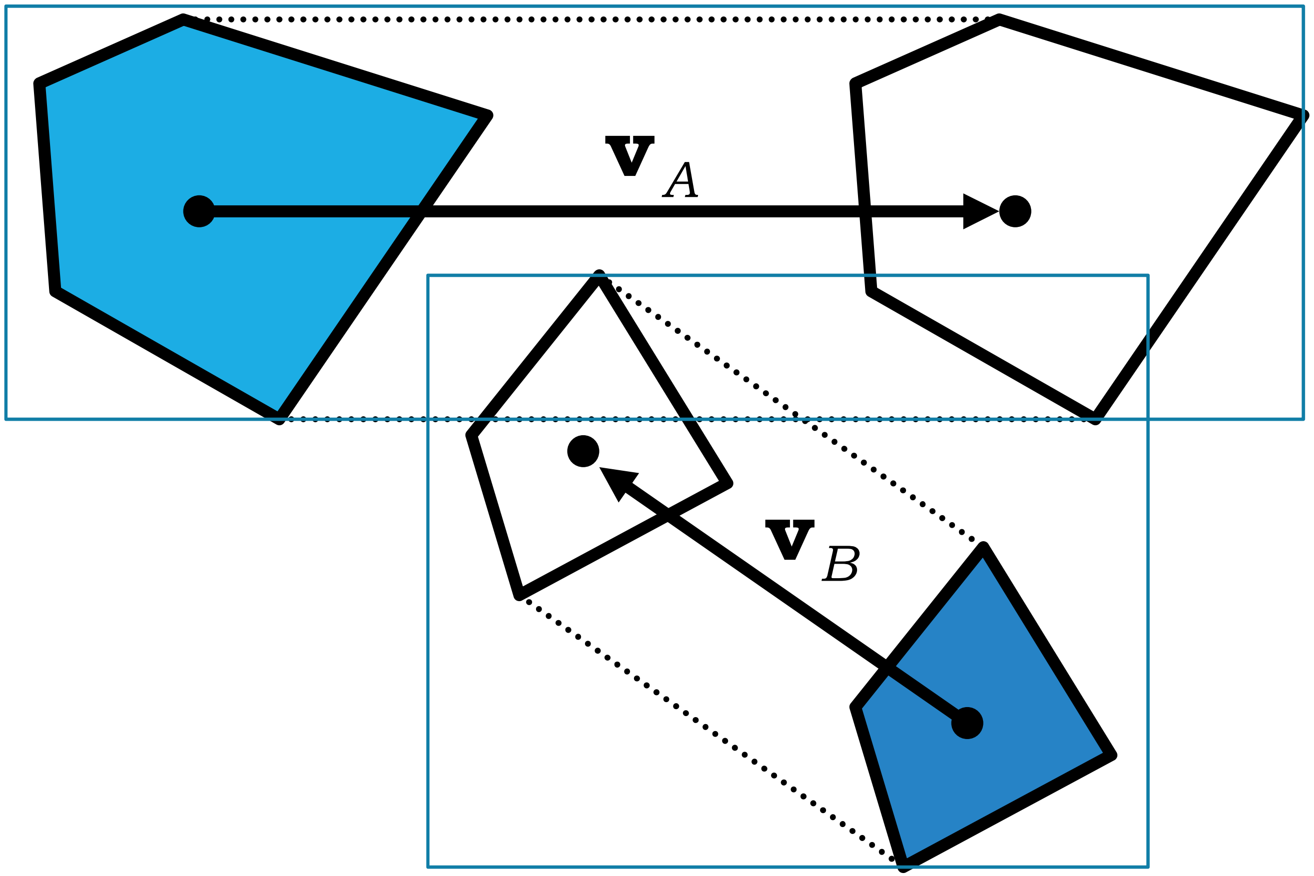
GJK FOR MOVING OBJECTS

FROM ERICSSON SIGGRAPH 04 TALK



GJK FOR MOVING OBJECTS

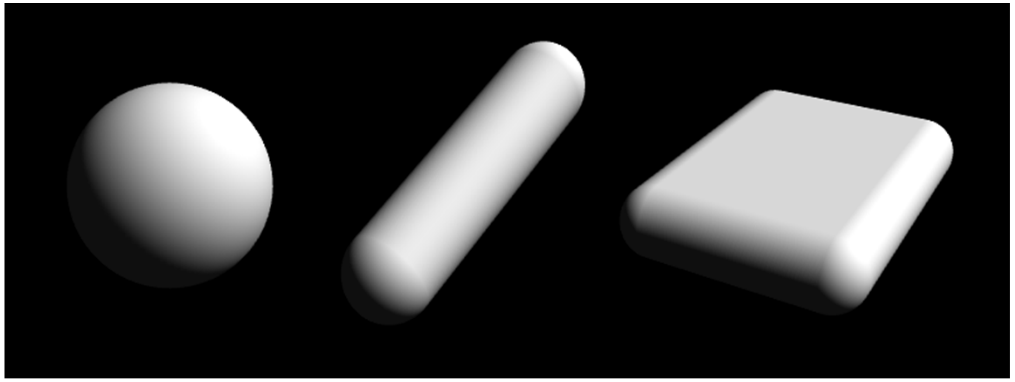
FROM ERICSSON SIGGRAPH 04 TALK



SPHERE-SWEPT VOLUMES

The volume generated by sweeping a sphere over a primitive

- Effectively: Growing a primitive outward by distance R

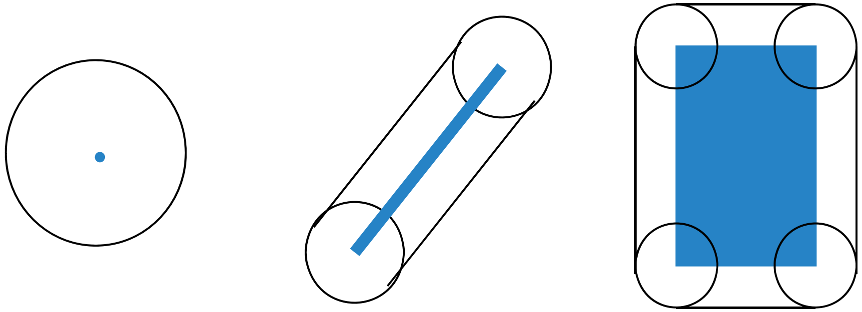


- More formally: the minkowski sum of a sphere and relevant primitive

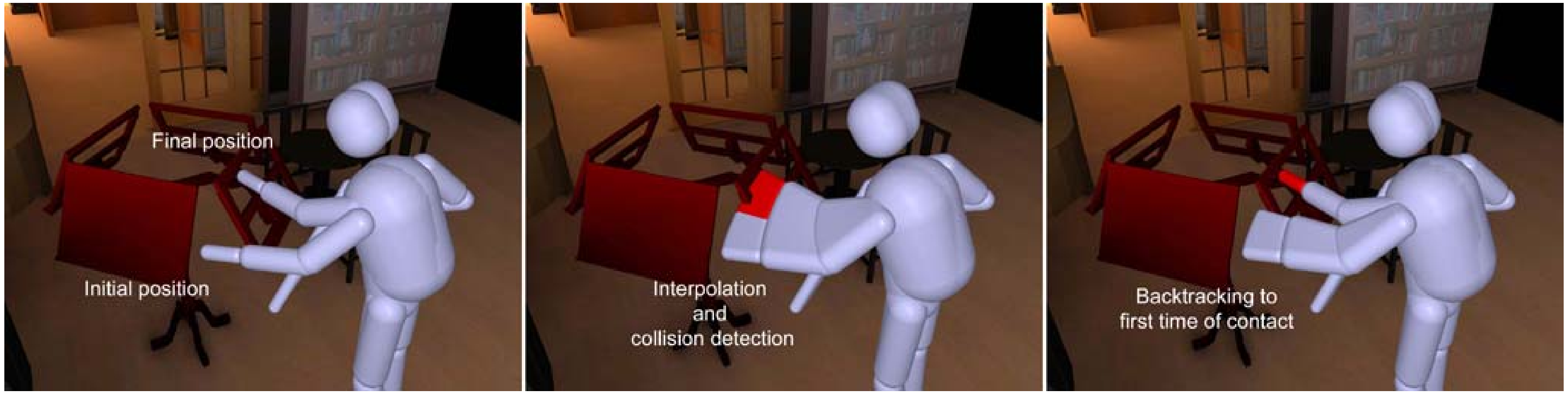
Point-swept sphere (PSS)

Line-swept sphere (LSS)

Rectangle-swept sphere (RSS)



CONTINUOUS COLLISION DETECTION



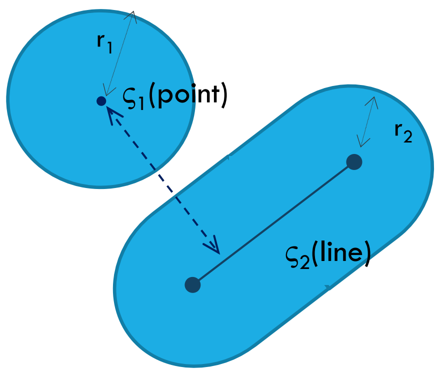
[Redon 2004]

SSV DETAILS

Update standard: orientation + position update

Collision checks

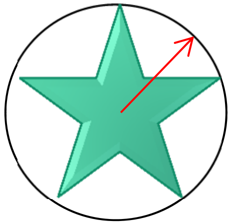
- Similar to the distance checking with the sweep primitive associated with the SSV (i.e. point, line-segment, quad) ....
- Is distance (V1, V2) < r1+r2? (where Vi is the sweep primitive i.e. point, line, rectangle; and ri the respective radii);



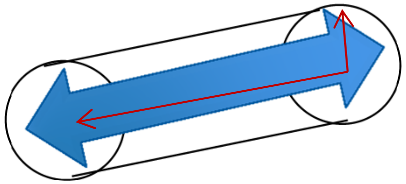
- Minkowski sum can be optimised on GPU

SSV GENERATION

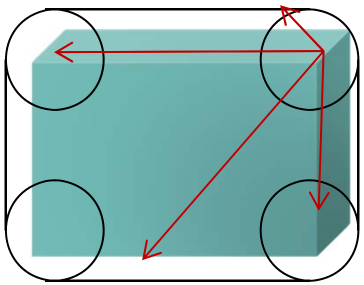
Similar to OBB based upon *Principle Component Analysis*



- For PSS, use the largest dimension as the radius



- For LSS, use the two largest dimensions as the length and radius



- For RSS, use all three dimensions

RELATED READING (OPTIONAL)

Miguel Gomez – "Simple Intersection Tests For Games" – Gamasutra, Oct 18, 1999

- http://www.gamasutra.com/view/feature/3383/simple_intersection_tests_for_games.php

Kenny Erleben et al "Physics Based Animation" (Book) Chapter 12: Broad Phase Collision Detection

Brian Mirtich – "Efficient Algorithms for Two-Phase Collision Detection" TR9723 MERL, Dec, 1997. (http://www.merl.com/reports/docs/TR97-23.pdf)

Philip Hubbard – "Space time bounds for collision detection" TR CS-93-04 Computer Science, Brown University, 1993. (ftp://ftp.cs.brown.edu/pub/techreports/93/cs93-04.pdf)

Demo of Spatial hashing: http://users.design.ucla.edu/~mflux/p5/hashcollision2/applet/

SSV's: PQP Collision detection package: http://www.cs.unc.edu/~geom/SSV/

SSV Paper: Eric Larsen, Stefan Gottschalk, Ming C. Lin, Dinesh Manocha, "Fast Proximity Queries with Swept Sphere Volumes", Proc. of Int. Conf. on Robotics and Automation, pp. 3719-3726, April 2000.