



POLITÉCNICA

SIMULACIÓN Y PLANIFICACIÓN QUIRÚRGICA

Tema 5 – Interacción entre modelos



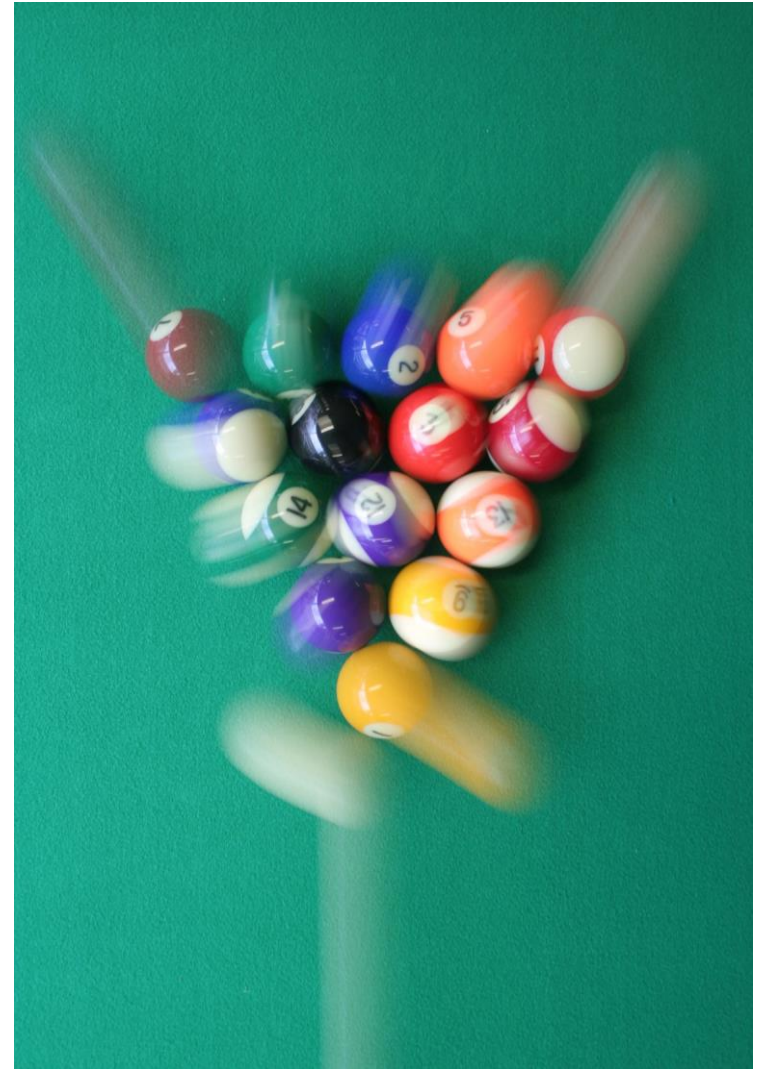
Francisco Morán Burgos
Grupo de Tratamiento de Imágenes

Dpto. Señales, Sistemas y Radiocomunicaciones
E.T.S.I. Telecomunicación
Universidad Politécnica de Madrid

Índice

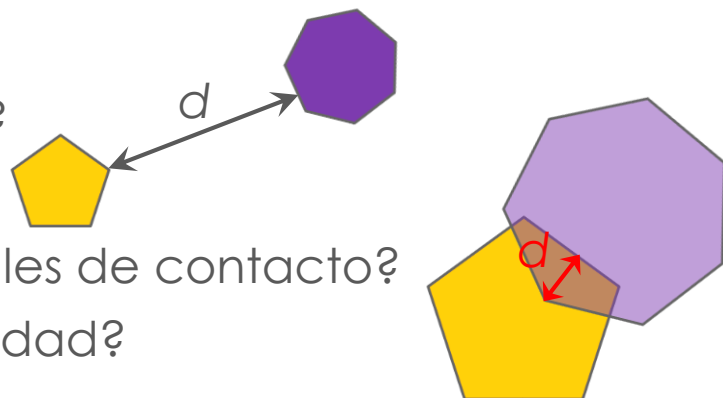
- Introducción
 - Conceptos fundamentales
 - Especificación del problema
 - Aplicaciones

- DCs (Detección de Colisiones) para SBF (Simulación Basada en Física)
 - Ecuaciones diferenciales
 - Más conceptos importantes
 - Historia: casi 50 años de investigación
 - *Pipelines* (cadenas de operaciones)
 - Optimizaciones
 - Objetos deformables y fracturables



Introducción: conceptos fundamentales

- Interacción de objetos de escena 4D ($x, y, z, \underline{t}$) con
 - usuario $\Rightarrow$ ya descrita (brevemente) en tema 2: eventos, rutas, etc.
 - otros objetos $\Rightarrow$ SBF (Simulación Basada en Física) de colisiones
- La colisión puede ser
 - instantánea (rebote) vs. permanente [para $t > t_0$] (contacto)
 - elástica vs. inelástica (fricción)
- DCs y problemas relacionados
 - Cálculos de proximidad/distancia: d ?
 - DCs propiamente dicha: $d = 0$?
 - Si hay colisión ($d = 0$), puntos y normales de contacto?
 - Si hay penetración ($d < 0$), profundidad?



Introducción: especificación del problema

- Tipo de modelado de
 - forma 3D
 - Superficial: poliedros, convexos o no
 - Volumétrico: modelos algebraicos...
 - propiedades físicas/dinámicas
 - Sólidos rígidos
 - Sólidos deformables/fracturables
 - Fluidos ($\Rightarrow$ sistemas de partículas)
- Tipo de estudio
 - Modelado del tiempo
 - Continuo: solución [exacta] de ecuaciones
 - Discreto: animación paso a paso con posible “marcha atrás”
 - Colisión simple (2 objetos) vs. múltiple (n objetos $\Rightarrow$ sist. partículas)
 - Cálculos de distancia/penetración vs. tiempo estimado para colisión

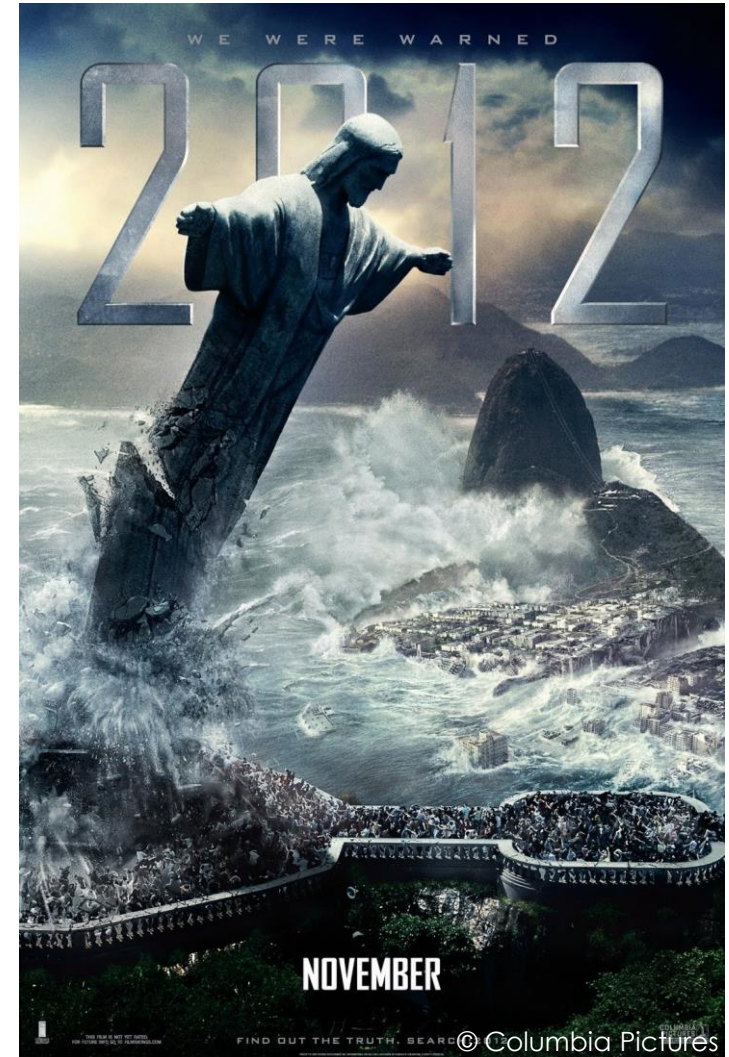
Introducción: aplicaciones

■ Intermedias

- SBF
- *Haptic rendering*

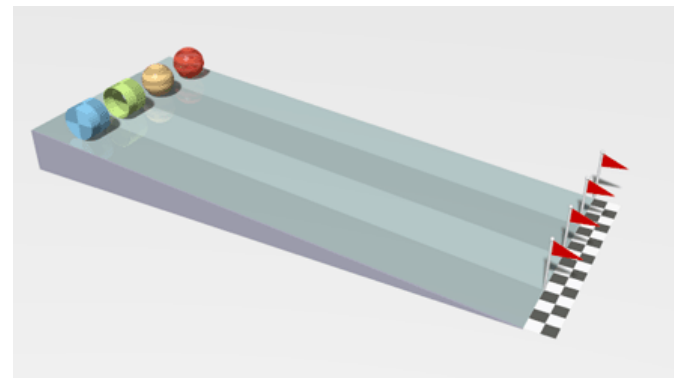
■ Finales

- Ocio
 - Videojuegos
 - Cine (fuego, explosiones, etc.)
- Ciencia e ingeniería
 - Simulación y planificación quirúrgica
 - Simulación científica [didáctica]
 - Planificación de movimiento de robots
 - Formación en tareas de [des]ensamblado
 - Análisis ergonómico
 - Verificación de tolerancias



SBF: ecuaciones diferenciales (1/2)

- Sólidos rígidos (lo menos que se despacha): leyes de Newton
 1. Inercia: Todo cuerpo persevera en su estado de reposo o movimiento uniforme y rectilíneo a no ser que sea obligado a cambiar su estado por fuerzas impresas sobre él
 2. Fuerza: El cambio de movimiento es proporcional a la fuerza motriz impresa y ocurre según la línea recta a lo largo de la cual aquella fuerza se imprime
 - Traslación: $\mathbf{F} = m \mathbf{a}$ ($= m \dot{\mathbf{v}} = m \ddot{\mathbf{x}}$)
 - Rotación: $\boldsymbol{\tau} = \mathbf{I} \boldsymbol{\alpha}$ ($= \mathbf{I} \dot{\boldsymbol{\omega}} = \mathbf{I} \ddot{\boldsymbol{\theta}}$)
 - $\boldsymbol{\tau}$ (3x1): momento rotacional (*torque*) aplicado
 - $\mathbf{I}$ (3x3): tensor/momento de inercia
 - $\boldsymbol{\alpha}$ (3x1): aceleración angular
 3. [Re]acción: Con toda acción ocurre siempre una reacción igual y contraria: quiere decir que las acciones mutuas de dos cuerpos siempre son iguales y dirigidas en sentido opuesto
 - Si hay colisión, son necesarios puntos y normales de contacto



SBF: ecuaciones diferenciales (2/2)

- Colisiones inelásticas: esto se complica MUCHO!
 - Fricción: la energía no se conserva (como cantidad de movimiento), porque parte de ella se disipa
 - Contacto: los solvers deben llegar a soluciones estables
- Sólidos deformables/fractuables: esto se complica mucho MÁS!!
 - Auto-colisiones
 - Ejemplos importantes
 - Para juegos y simulación en general: pelo, tejidos (textiles u orgánicos)
 - Para SIP: hilo de sutura

- Líquidos: bienvenido al INFIERNO!!!

- Ecuaciones de Navier-Stokes

$$\underbrace{\frac{\partial \mathbf{u}}{\partial t}}_{\text{Variation}} + \underbrace{\mathbf{u} \cdot \nabla \mathbf{u}}_{\text{Convection}} - \underbrace{\nu \nabla^2 \mathbf{u}}_{\text{Diffusion}} = \underbrace{-\nabla w}_{\text{Internal source}} + \underbrace{\mathbf{g}}_{\text{External source}}$$

Inertia (per volume) Divergence of stress

- En 3D, no hay prueba de que su solución siempre exista ni de que, si existe, sea derivable: es uno de los *Millennium Prize Problems* por cuya solución el *Clay Mathematics Institute* ofrece **1MUSD**

- NB: los gases (ej: vapor, humo) también se simulan así $\Rightarrow$ sist. partículas

DCs/SBF: más conceptos importantes (1/2)

■ Modelado del tiempo

■ Continuo (DCs a priori $\Rightarrow \nexists$ penetración)

- Trayectoria de objetos conocida a priori
- Análisis de contacto mediante solución [exacta] de ecuaciones

■ Discreto (DCs a posteriori $\Rightarrow \exists$ penetración)

- Animación paso a paso (Δt) $\Rightarrow$ problemas de muestreo temporal
- Posible “marcha atrás” para deshacer penetraciones a posteriori

■ Coherencia temporal

- En DC discreta (y también en DC continua, como se verá...), casi todo lo que era cierto para t lo es para $t + \Delta t \Rightarrow$ análisis diferencial

DCs/SBF: más conceptos importantes (2/2)

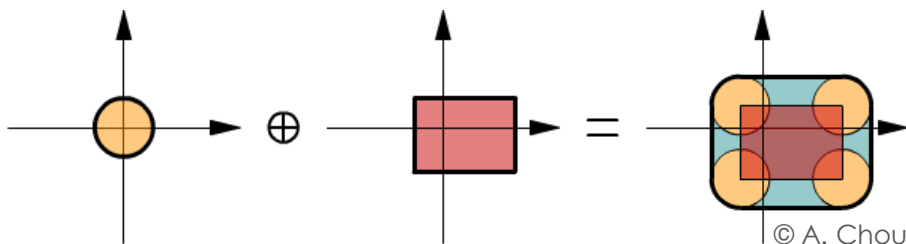
- ▣ Volúmenes limítrofes (*bounding volumes*)
 - ▣ AABBs (*Axis-Aligned Bounding Boxes*)
 - ▣ OBBs (*[object-]Oriented Bounding Boxes*)
 - ▣ Envolturas convexas (*convex hulls*)
 - ▣ Muy útiles para acelerar DCs mediante descarte (*culling*)

▣ Operadores de Minkowski

▣ Suma: $A \oplus B = \{\mathbf{a} + \mathbf{b} : \mathbf{a} \in A, \mathbf{b} \in B\}$

▣ Diferencia: $A \ominus B = \{\mathbf{a} - \mathbf{b} : \mathbf{a} \in A, \mathbf{b} \in B\}$, llamada CSO (*Configuration Space Obstacle*), tiene interesantes propiedades para DCs

- En general, la distancia entre A y B es $d(A, B) = \min\{\|\mathbf{x}\| : \mathbf{x} \in A \ominus B\}$
- En particular, $\mathbf{0} \in A \ominus B \Leftrightarrow d(A, B) = 0 \Leftrightarrow$ hay colisión entre A y B , y entonces la penetración entre A y B es $p(A, B) = \min\{\|\mathbf{x}\| : \mathbf{x} \notin A \ominus B\}$



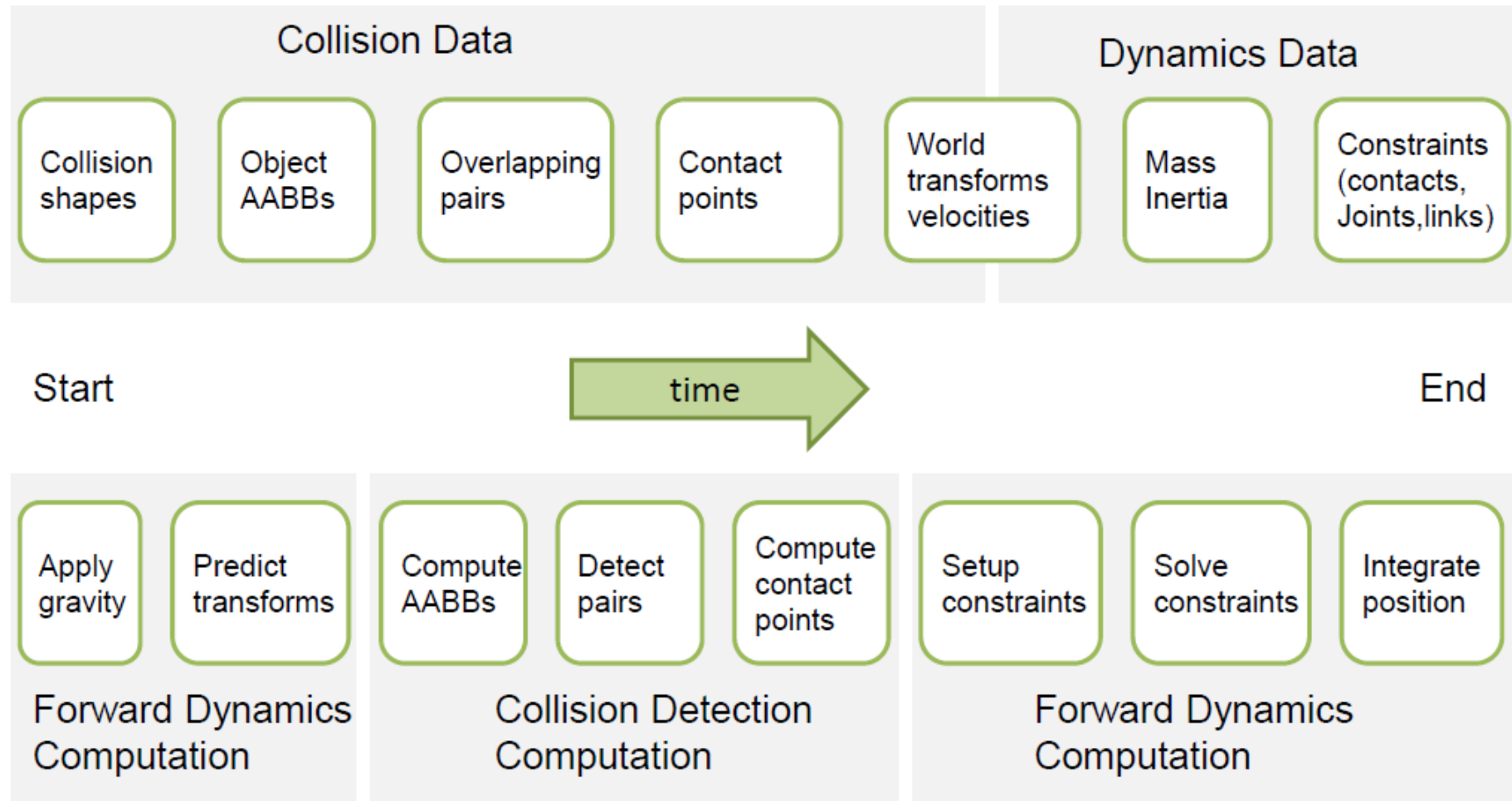
DCs/SBF: ~50 años de investigación (1/2)

- 1970's y 80's: primeros pasos
 - Algoritmos de cálculo de intersecciones 2D y 3D, sobre todo para *rendering* (ej: *ray tracing*)
 - Primeros intentos de DCs (ej: fronteras de juegos)
- 1990's: mucha actividad!
 - Problemas “sencillos”
 - Sólidos rígidos
 - DCs discreta
 - Cálculo de distancia entre politopos convexos
 - Estudio mediante volúmenes limítrofes [jerárquicos] y barridos
 - Algoritmo iterativo GJK (de E.G. Gilbert, D.W. Johnson y S.S. Keerthi) para estimar el punto del CSO más próximo al origen
 - Algunos nombres más: D. Baraff, J.F. Canny, A. Gregory, M.C. Lin...
 - Primeros algoritmos para optimización de DCs entre n objetos
 - Aplicación a juegos, robótica, realidad virtual, *haptic rendering*, etc.

DCs/SBF: ~50 años de investigación (2/2)

- 2000's y 10's: aún mucha actividad!
 - Problemas aun más complejos
 - Sólidos deformables y fracturables
 - Modelos articulados con auto-colisiones
 - Fluidos y gases
 - Progresos considerables en DCs continua
 - Conceptos fundamentales
 - Operadores de Minkowski y CSO
 - DCs entre objetos no convexos (por descomposición en convexos)
 - Sistemas de partículas
 - “Motores” físicos (como parte, p.ej., de “motores” de juegos)
 - Esfuerzo enorme en optimización
 - Determinación del mayor Δt “conservador” posible para DCs continua
 - Paralelización masiva gracias a GPUs
 - Más nombres: E. Coumans (autor principal de *Bullet Physics Library*)...

SBF: *pipeline* (cadena de operaciones)



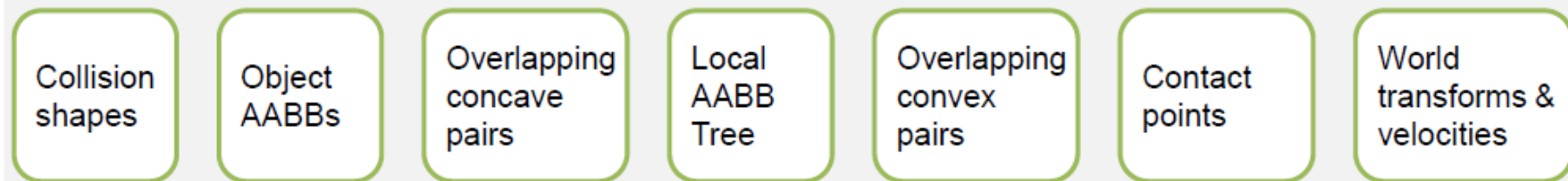
AABB = axis aligned bounding box

© E. Coumans

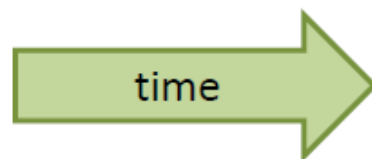


DCs: *pipeline*

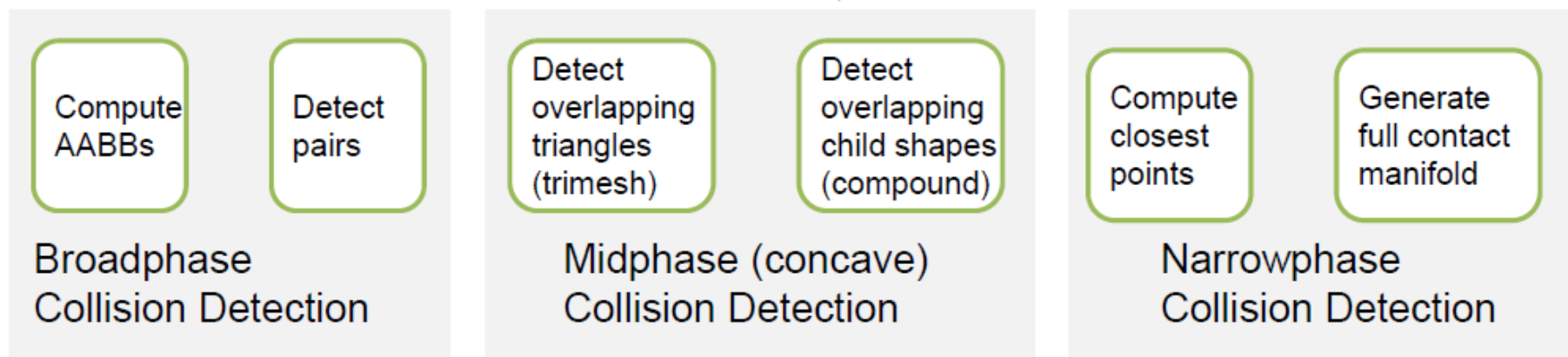
Collision Data



Start



End



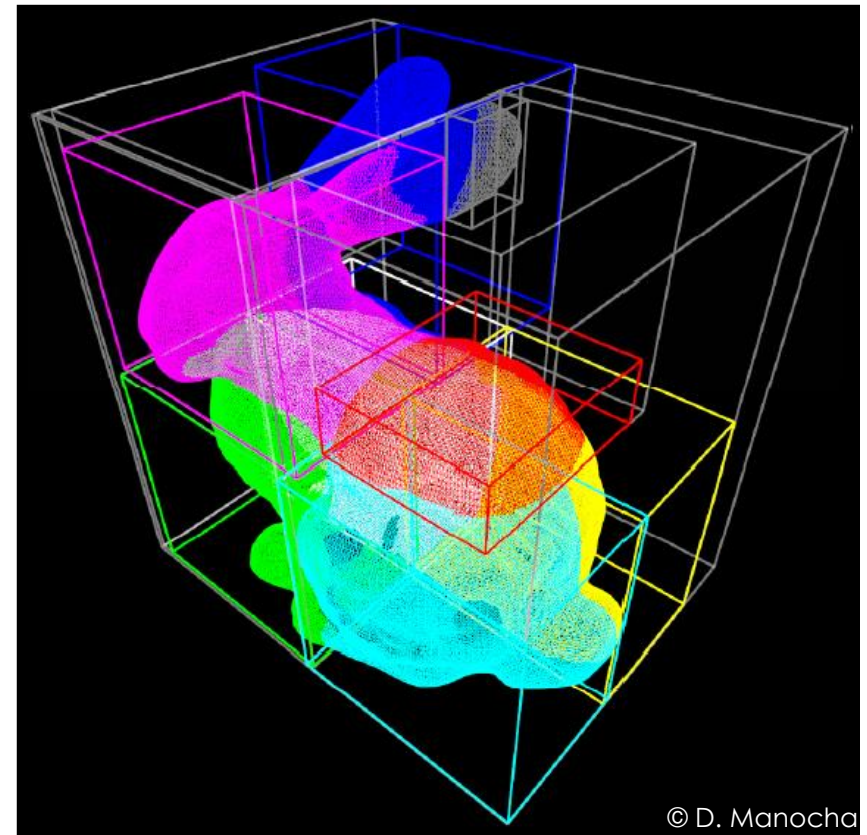
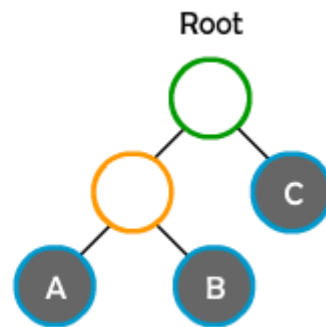
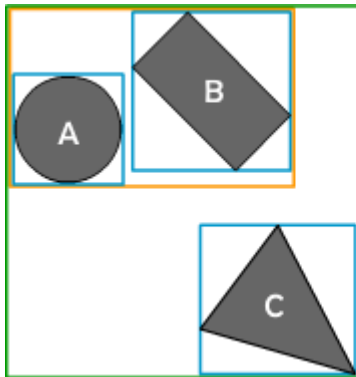
culling using acceleration structures

© E. Coumans



DCs: optimizaciones (1/4)

- Fase abierta (*broad phase*) (1/3)
 - DCs entre pares de AABBs para descartar colisiones imposibles
 - NB: las AABBs pueden ser jerárquicas



© D. Manocha

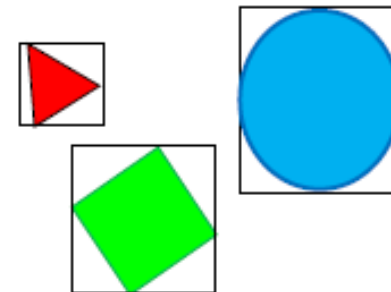
DCs: optimizaciones (2/4)

■ Fase abierta (2/3)

■ Objetivo: evitar análisis mediante fuerza bruta, de coste $O(N^2)$

```
bool TestAabbAgainstAabb ( __global const btAabbCL* aabb1, __global const btAabbCL* aabb2) {
    bool overlap = true;
    overlap = (aabb1->m_min.x > aabb2->m_max.x || aabb1->m_max.x < aabb2->m_min.x) ? false : overlap;
    overlap = (aabb1->m_min.z > aabb2->m_max.z || aabb1->m_max.z < aabb2->m_min.z) ? false : overlap;
    overlap = (aabb1->m_min.y > aabb2->m_max.y || aabb1->m_max.y < aabb2->m_min.y) ? false : overlap;
    return overlap;
}

__kernel void computePairsBruteForceKernel( __global const btAabbCL* aabbs, volatile __global int2* pairsOut,
                                             volatile __global int* pairCount, int numObjects, int axis, int maxPairs)
{
    int i = get_global_id(0);
    if (i >= numObjects)
        return;
    for (int j=i+1; j<numObjects; j++)
    {
        if (TestAabbAgainstAabb (&aabbs[i], &aabbs[j]))
        {
            int2 myPair;
            myPair.x = aabbs[i].m_minIndices[3];
            myPair.y = aabbs[j].m_minIndices[3];
            int curPair = atomic_inc (pairCount);
            if (curPair < maxPairs)
            {
                pairsOut[curPair] = myPair; //flush to main memory
            }
        }
    }
}
```

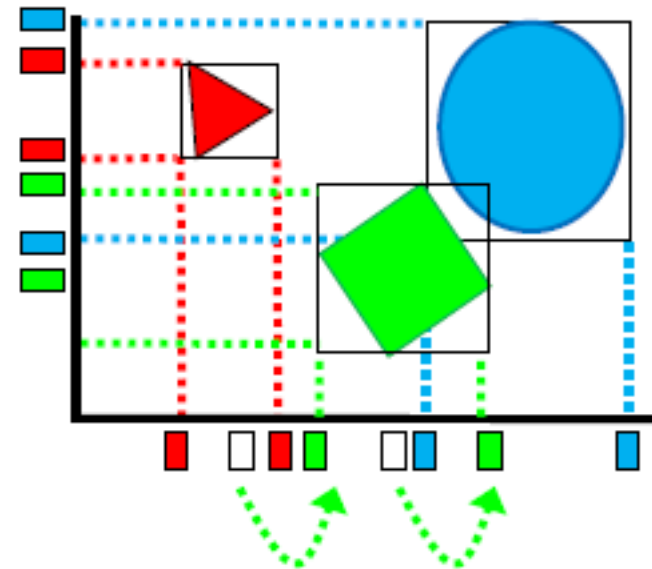
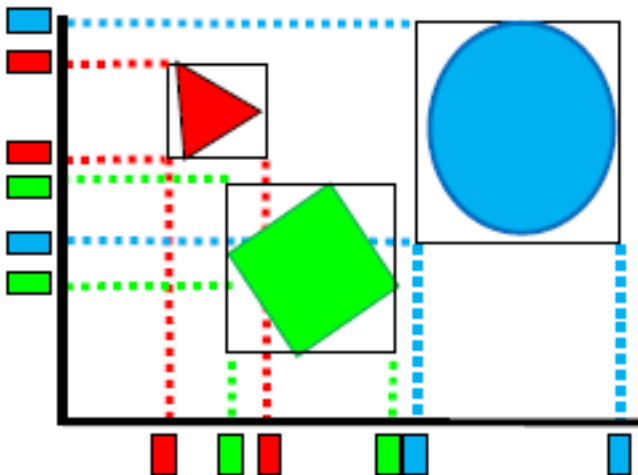


© E. Coumans

DCs: optimizaciones (3/4)

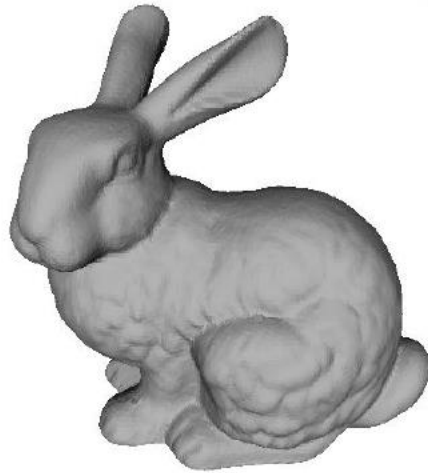
■ Fase abierta (3/3)

- Técnica de barrido y poda (*sweep & prune*) incremental
 - Ordena y actualiza los pares de AABBs que se solapan eje por eje
 - Funciona mejor cuanto mayor sea la coherencia temporal
 - Difícil de paralelizar

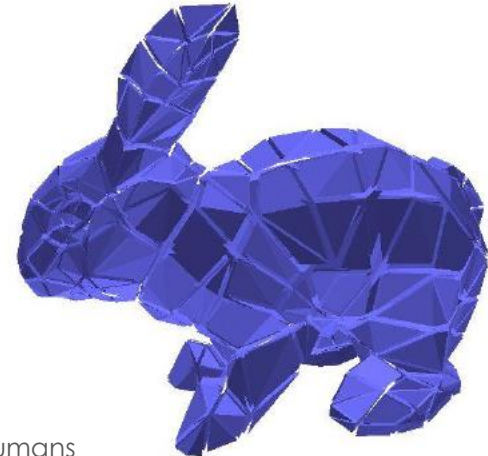


© E. Coumans

DCs: optimizaciones (4/4)

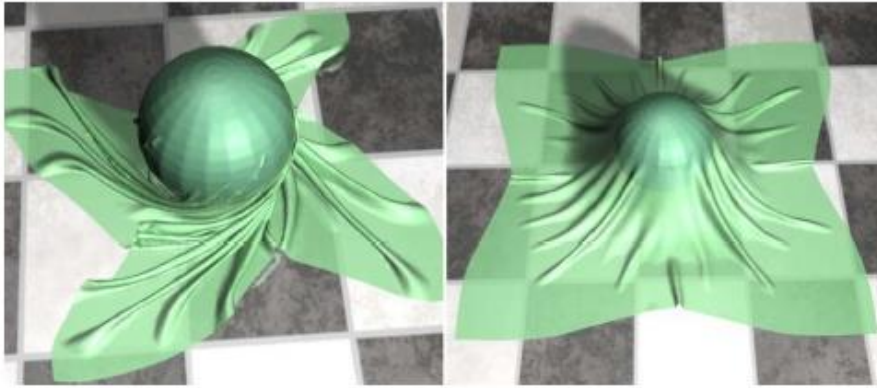


© E. Coumans



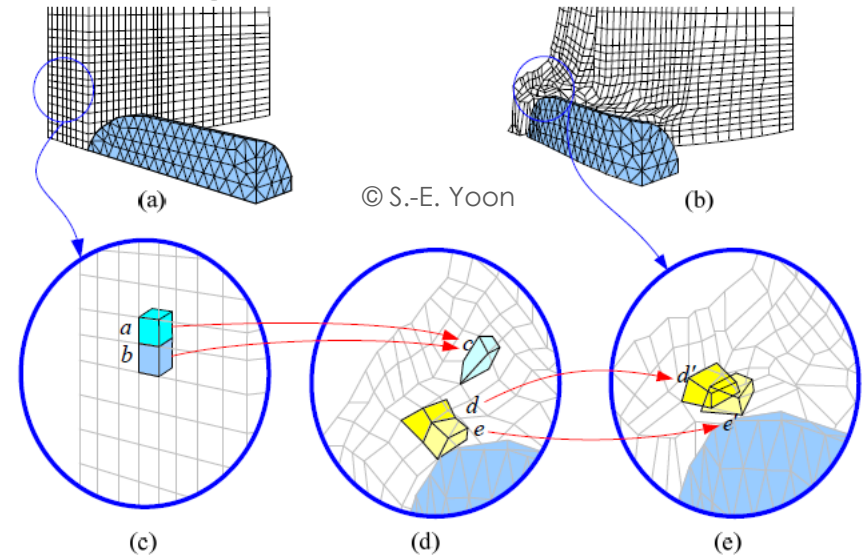
- Fase central (*mid phase*)
 - Mecanismo para descomponer un objeto complejo en sub-objetos y, de nuevo, descartar colisiones imposibles
 - NB: aquí cobra mucha importancia la jerarquización de AABBs
- Fase cerrada (*narrow phase*)
 - Por fin, ya no hay más remedio que trabajar...
 - Cálculo de puntos más cercanos, puntos/superficies/normales de contacto, profundidades de penetración, instantes de impacto, etc.

DCs: objetos deformables y fracturables



© S.-E. Yoon

- ▣ AABBs/OBBs dinámicas, actualizadas según deformación
- ▣ Análisis multi-resolución
- ▣ Para modelos volumétricos, se debe considerar las posibles auto-colisiones debidas a deformaciones y cambios topológicos



© S.-E. Yoon

Bibliografía

- T. Akenine-Möller, E. Haines, N. Hoffman: *Real-Time Rendering*, 3rd ed., A.K. Peters (ahora CRC Press), 2008

- E. Coumans [et al.]:
 - Cursos impartidos en ACM SIGGRAPH 2010 & 2011, GDC 2009-13...
 - *Bullet Physics Library* (<http://bulletphysics.org>)

- T. Mullen (prólogo de E. Coumans): *Bounce, Tumble, and Splash! Simulating the Physical World with Blender 3D*, Wiley, 2008
[NB: ∃ T. Mullen: *Mastering Blender*, 2nd ed., Wiley, 2012]

- Wikipedia, claro...