

Master 1 Informatique
UE M1if37 - Animation en synthèse d'image
Partie - Simulation par modèles physiques
Cours 4 - Détection et réponse aux collisions

Florence Zara

LIRIS - Université Lyon 1

<http://liris.cnrs.fr/florence.zara>

E-mail: florence.zara@liris.cnrs.fr

Détection et traitement des collisions

Souvent plusieurs objets en interaction dans la scène

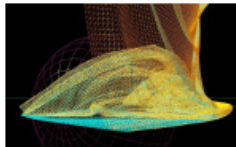
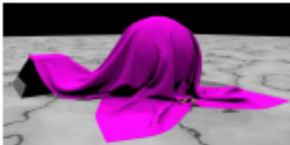
Détection et traitement des collisions : exemples (1)

Exemple de l'animation de textiles (objets déformables) :

- Collisions avec un autre objet de la scène, un avatar
- Auto-collisions également à considérer



[Teschner, et. al. 2004]



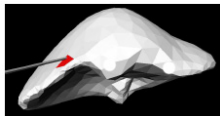
[RobertBridson]

→ Utilisation d'algorithmes dédiés aux objets déformables

Détection et traitement des collisions : exemples (2)

Exemple de la simulation de gestes médicaux :

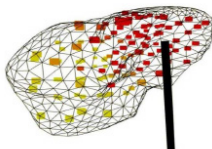
- Collisions entre différents organes
- Collisions entre un organe et un objet chirurgical
- Auto-collisions dans le cas de changement topologique



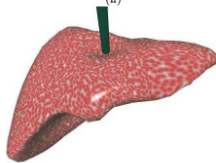
(i)



(ii)



(iii)



(iv)



[Debunne] [SOFA]

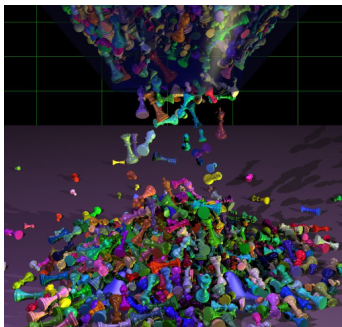
Détection et traitement des collisions : exemples (3)

Exemple de la simulation d'objets rigides :

- Collisions à traiter entre les solides en interaction - vidéo

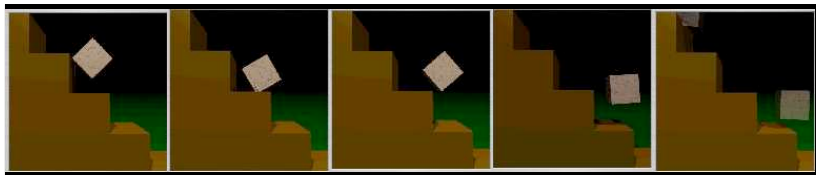


[Kaufman, et. al. 2005]



Détection et traitement des collisions

Souvent plusieurs objets en interaction dans la scène



Deux problèmes à résoudre :

- Détection des collisions dans la scène
- Traitement de ces collisions

Gestion de plusieurs objets dans la scène

Boucle de simulation - Mouvement sans contrainte

- Pas d'interaction entre les objets de la scène
- Calcul des états des objets de la scène à chaque pas de temps

Boucle de simulation - Mouvement avec contrainte

- Interactions entre les objets de la scène
- Contraintes de non-pénétration des objets entre eux
- Calcul des états des objets de la scène en dehors de ces contraintes
- Boucle de simulation interrompue pour répondre à la collision

Gestion de plusieurs objets dans la scène

Boucle de simulation - Mouvement avec contrainte

- Calcul des états des objets de la scène
- Détection des collisions
- Détermination des points de contacts
- Calcul des réponses aux collisions
- Détermination des forces de contraintes
- Calcul des nouvelles vitesses

Plan du cours

Traitements des collisions

- Déterminer le type de la collision : contact en collision, contact établi, rupture de contact
- Répondre en fonction du type de collision : force d'impulsion, force de contact

Détection des collisions

- Détection exacte
- Méthode des axes séparateurs
- Boîtes englobantes

Comment on traite les collisions ?

Collision entre deux objets **rigides** A et B au point de contact P
On notera pour la suite **N** la normale au point P

- Etape 1 : déterminer le type de la collision
 - Point de collision ou contact en collision (*colliding contact*)
 - Contact établi (*resting contact*)
 - Rupture de contact (*separation*)
- Etape 2 : répondre à la collision en fonction de son type
 - Force d'impulsion
↔ permet de modifier la vitesse de manière discontinue
 - Force de contact
↔ permet de maintenir le contact

Comment on traite les collisions ?

Considère uniquement les collisions entre deux **polyèdres convexes**

Un polyèdre est dit **convexe** si sa frontière (incluant ses faces et ses arêtes) ne se coupe pas elle-même et si un segment joignant deux points quelconques du polyèdre fait partie de celui-ci ou de son intérieur

Autrement dit, un polyèdre est convexe si toutes ses diagonales sont entièrement contenues dans son intérieur

Rappel - Produit scalaire

Soient deux vecteurs \mathbf{u} et \mathbf{v}

Leur produit scalaire est défini par :

$$\mathbf{u} \cdot \mathbf{v} = (u_1, u_2, \dots, u_n) \cdot (v_1, v_2, \dots, v_n) = u_1 v_1 + u_2 v_2 + \dots + u_n v_n$$

Le produit scalaire d'un vecteur par lui-même représente sa norme au carré :

$$\mathbf{v} \cdot \mathbf{v} = v_1^2 + v_2^2 + \dots + v_n^2 = \|\mathbf{v}\|^2$$

Résultat d'un produit scalaire est un scalaire (non un vecteur)

Rappel - Produit scalaire

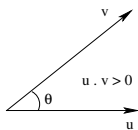
Soit θ le plus petit angle formé par les vecteurs \mathbf{u} et \mathbf{v} :

$$\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\| \|\mathbf{v}\| \cos(\theta) \Rightarrow \theta = \cos^{-1} \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$$

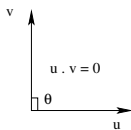
Pour des vecteurs \mathbf{u} et \mathbf{v} non nuls, si leur produit scalaire est positif $\Rightarrow \theta$ est un angle aigu ($<$ à 90 degrés)

nul $\Rightarrow \theta$ est un angle droit (= à 90 degrés)

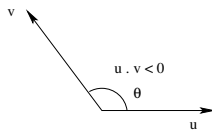
négatif $\Rightarrow \theta$ est un angle obtus ($>$ à 90 degrés)



Angle aigu



Angle droit

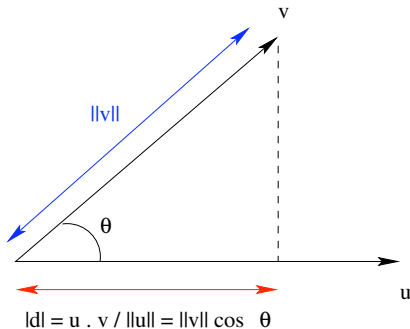


Angle obtus

Rappel - Produit scalaire

Produit scalaire représente la projection de \mathbf{v} sur \mathbf{u}

Soit d la distance signée de \mathbf{v} sur \mathbf{u} : $d = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|} = \|\mathbf{v}\| \cos(\theta)$



Comment on traite les collisions ?

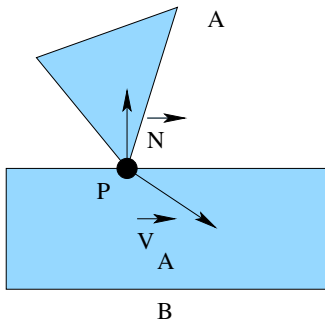
Etape 1 : déterminer le type de la collision

Quel est le type de la collision ? - Cas 1

- *Colliding contact* - Point de collision ou contact en collision

La collision vient juste de se produire

Pénétration de l'objet A dans l'objet B : $\mathbf{N} \cdot \mathbf{V}_A < 0$



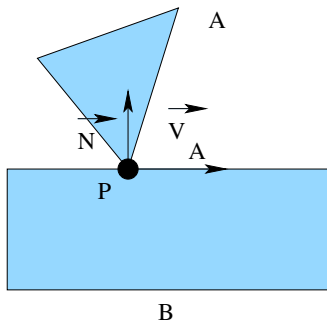
Quel est le type de la collision ? - Cas 2

- *Resting contact* - Contact établi

Il y avait déjà un contact à l'instant précédent

Cas de glissement ou de frottement entre les deux objets

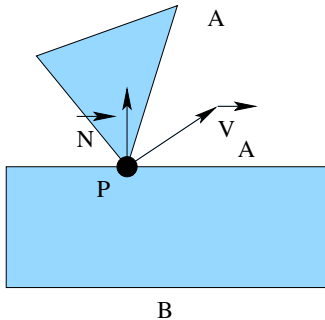
\vec{V}_A est nulle en P ou perpendiculaire à B en P : $\mathbf{N} \cdot \mathbf{V}_A = 0$



Quel est le type de la collision ? - Cas 3

- *Séparation* - Rupture de collision

Les deux objets se séparent après une collision : $\mathbf{N} \cdot \mathbf{V}_A > 0$



Dans quel cas sommes-nous ?

Il faut déterminer le type de la collision pour répondre à la collision
↔ besoin de connaître le signe de $\mathbf{N} \cdot \mathbf{V}_A$

Pour cela, nous allons observer (avant la collision) la trajectoire des deux points qui ont engendré la collision

Soit t_0 le temps du premier contact établi entre A et B.

Soit P_0 le point de contact entre ces deux objets.

Soit N_0 longueur unitaire de la normale sortante à la face. Pour intersection arête-arête, N_0 représente le produit scalaire entre les directions des arêtes.

Vecteur choisi de façon à pointer en dehors du deuxième objet.

Dans quel cas sommes-nous ?

Soient $P_A(t)$ et $P_B(t)$ les trajectoires des points impliqués dans la collision avant celle-ci *i. e.* pour $t \leq t_0$ avec $P_A(t_0) = P_B(t_0) = P_0$.

Soit $\mathbf{N}(t)$ la normale du point de B impliqué dans la collision avec $\mathbf{N}(t_0) = N_0$.

La distance signée entre les points des objets contribuant à l'intersection (mesurée dans la direction de la normale) est :

$$d(t) = \mathbf{N}(t) \cdot (P_A(t) - P_B(t))$$

La composante vitesse dans la direction de la normale est :

$$\dot{d}(t) = \mathbf{N}(t) \cdot (\dot{P}_A(t) - \dot{P}_B(t)) + \dot{\mathbf{N}}(t) \cdot (P_A(t) - P_B(t))$$

Dans quel cas sommes-nous ?

A l'instant du contact t_0 , on a :

$$d(t_0) = 0 \text{ et } \dot{d}(t_0) = \mathbf{N}_0 \cdot (\dot{P}_A(t_0) - \dot{P}_B(t_0))$$

$\dot{d}(t_0)$ permet de qualifier le type de collision au point de contact P_0 .

Au moment de la détection de la collision, il est alors possible de connaître les différents types de collision qui se sont produits aux différents points de contact.

Comment on traite les collisions ?

Etape 2 : répondre à la collision en fonction de son type

Etape 2 : répondre à la collision en fonction de son type

Il y a trois types de collision possibles :

- la collision vient juste de se produire
- la collision continue dans le temps
- la collision est rompue

Nous nous intéressons à la réponse à donner dans le cas :

- où la **collision vient de se produire** (*colliding contact*), avec l'objectif d'éviter l'inter-pénétration des deux objets : introduction d'une **force d'impulsion**

Réponse à la collision - *Colliding contact* - Impulsion

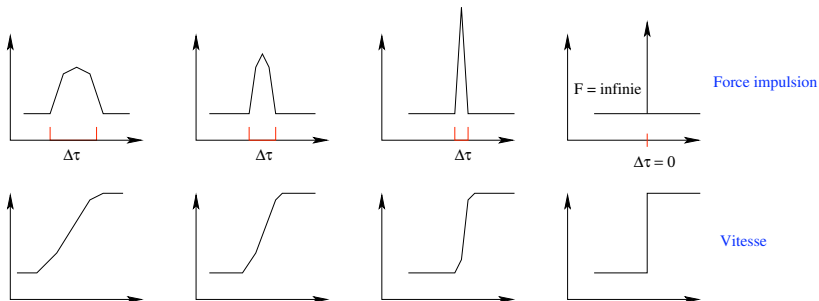
En cas de *colliding contact* (cas où $\dot{d}(t_0) < 0$), une réponse à la collision doit être donnée afin d'éviter l'inter-pénétration des deux objets en contact.

↔ cette réponse correspond à l'application d'une **impulsion** au point de contact.

↔ la vitesse relative $\dot{P}_A(t) - \dot{P}_B(t)$ va ainsi être changée de manière discontinue.

Réponse à la collision - *Colliding contact* - Impulsion

Lien entre le changement de vitesse et la force d'impulsion
 (collision d'intensité de plus en plus importante) :

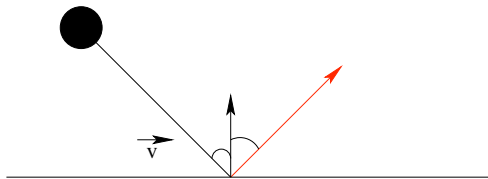


Réponse à la collision - *Colliding contact* - Impulsion

Exemple de la balle rigide :

↔ vitesses des objets font en sorte que les objets tentent de se pénétrer l'un dans l'autre

↔ il faut changer le vecteur vitesse pour que la balle rebondisse



Colliding contact - Changement de vitesse

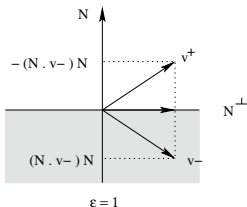
Pour répondre à la collision, la vitesse souhaitée **après** l'application de l'impulsion au point de contact est sélectionnée.

Vitesse choisie = réflexion par la rapport à la normale de la vitesse de l'objet au point d'impact.

Vitesse **avant** l'impulsion : $\mathbf{v}^- = \mathbf{N}^\perp + (\mathbf{N} \cdot \mathbf{v}^-)\mathbf{N}$

Vitesse **après** l'impulsion : $\mathbf{v}^+ = \mathbf{N}^\perp - (\mathbf{N} \cdot \mathbf{v}^-)\mathbf{N}$

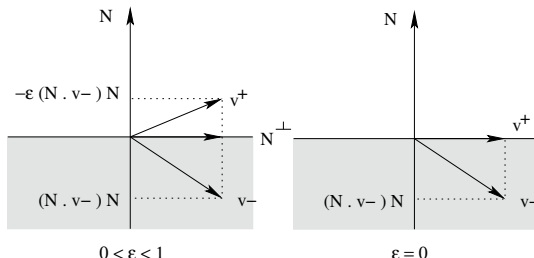
↪ cas de la réflexion sans perte d'énergie durant la collision



Colliding contact - Changement de vitesse

Mais pour obtenir une réponse à la collision plus réaliste, il est possible de rajouter une perte d'énergie

Introduction d'un **coefficient de restitution** $\epsilon \in [0, 1]$ avec impulsion : $\mathbf{v}^+ = \mathbf{N}^\perp - \epsilon (\mathbf{N} \cdot \mathbf{v}^-) \mathbf{N}$

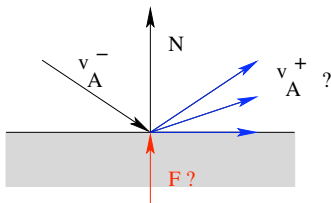


Si $\epsilon = 1$, pas de perte d'énergie (réflexion parfaite, choc élastique)
Si $\epsilon = 0$, point de contact = un contact établi (choc absorbant)

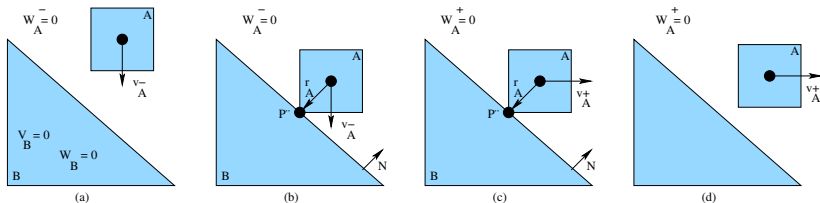
Réponse à la collision - *Colliding contact* - Impulsion

Supposons l'impulsion $\mathbf{F} = f \mathbf{N}_0$ appliquée sur l'objet A au moment de la collision.

f est l'**intensité de l'impulsion** (scalaire) qui doit être calculée de manière à obtenir la vitesse désirée après impulsion.



Réponse à la collision - *Colliding contact* - Exemple



- Gravité pas prise en compte
- Coefficient de restitution $\epsilon = 1$

Comment on détecte la collision ?

Avant tout, il faut savoir détecter les collisions qui se produisent dans la scène

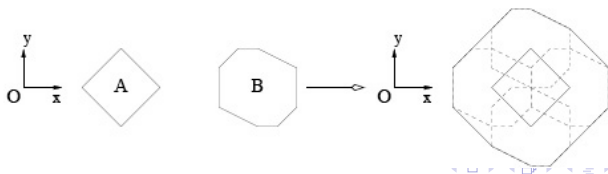
Il faut savoir déterminer si les objets se touchent ou s'inter-pénètrent (pour revenir en arrière et éviter cette collision)

Détection exacte : collision entre 2 objets

Comment déterminer si 2 polyèdres A et B sont en collision ?

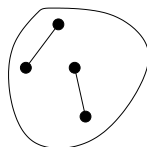
- 1 Regarde si il existe un plan partitionnant l'espace en 2 demi-espaces
→ l'un contient A, l'autre B : pas de collision
- 2 Calcul de la distance entre ces 2 objets
→ si distance ≤ 0 : collision
→ utilise la distance de Minkowski

$$\mathcal{M} = A - B = \{x - y / x \in A, y \in B\}$$

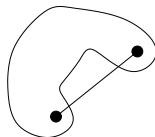


Détection des collisions entre polyèdres convexes

Considère uniquement des polyèdres convexes = polytopes
Élément convexe : tout axe reliant deux points de l'élément se trouve entièrement dans celui-ci



Convexe



Non convexe

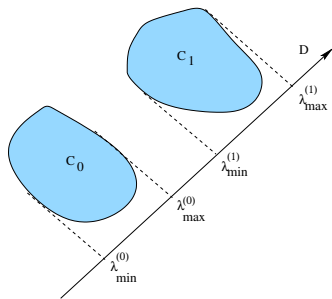
Souhaite que les polytopes ne s'inter-pénètrent pas

But du système de détection de collisions : calculer tous les points de contact entre chaque paire de polytopes

Méthode des axes séparateurs

Deux éléments convexes sont **séparés** ssi il existe une direction D telle que les intervalles des projections de ces éléments ne se chevauchent pas :

$$\lambda_{min}^{(0)}(D) > \lambda_{max}^{(1)}(D) \text{ ou } \lambda_{max}^{(0)}(D) < \lambda_{min}^{(1)}(D)$$



Polygones convexes (2D)

Considère deux polygones convexes notés C_j avec $j = 0, 1$

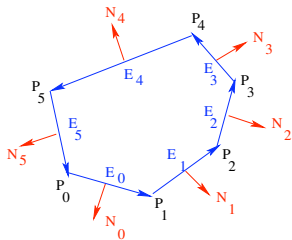
Soit N_j le nombre de sommets du polygone C_j

Soient P_i^j pour $0 \leq i \leq N_j$ les sommets du polygone C_j

Arêtes des polygones notées $\mathbf{E}_i^{(j)} = P_{i+1}^{(j)} - P_i^{(j)}$ pour $0 \leq i \leq N_j$

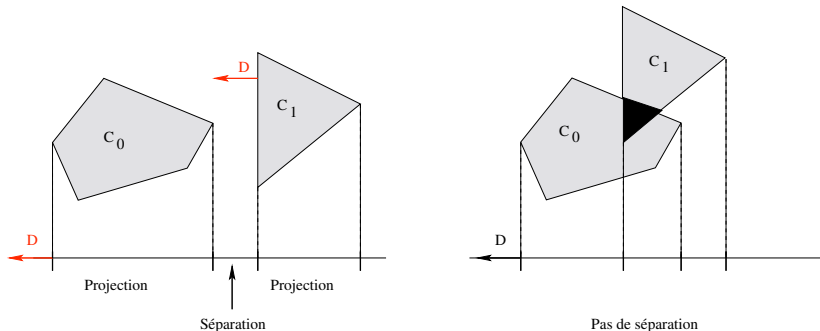
Soient $N_i^{(j)}$ les vecteurs sortants normaux des arêtes avec :

$$(N_i^{(j)})^\perp \cdot \mathbf{E}_i^{(j)} > 0, \text{ avec } (x, y)^\perp = (-y, x)$$



Polygones convexes

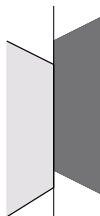
Test de séparation effectué sur un ensemble fini S de directions
Ensemble correspond aux normales des arêtes des polygones



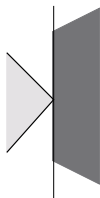
Polygones convexes

Seulement trois intersections possibles sans inter-pénétrations entre deux polygones convexes :

- Intersection arête-arête
- Intersection sommet-arête
- Intersection sommet-sommet



arête-arête



sommet-arête



sommet-sommet

Polygones convexes : test des intersections

Soient C_0 et C_1 deux polygones convexes ayant N_0 et N_1 sommets
Algorithme naïf de la méthode des axes séparateurs :

- Sélectionne une direction potentielle D
- Calcul des intervalles de projection en projetant les sommets des deux polygones sur cette ligne
- Test si ces intervalles sont séparés

↔ nécessite le calcul de N_i projections pour le polygone C_i et la conservation de la projection minimale et maximale de chaque polygone

↔ le pire cas (intersection entre les polygones) :

N_0 directions testées pour C_0 , nécessitant chacune $N_0 + N_1$ projections, N_1 directions testées pour C_1 , nécessitant chacune $N_0 + N_1$ projections : $(N_0 + N_1)^2$ projections nécessaires

Polygones convexes : test des intersections

Algorithme plus judicieux de la méthode des axes séparateurs :

Evite de tester l'ensemble des projections, en utilisant uniquement le plus grand intervalle de C_0 et le plus petit intervalle de C_1

Soit N_j le nombre de sommets de C_j

Soient $P_i^{(j)}$ pour $0 \leq i \leq N_j$ les sommets de C_j

Polygones convexes : test des intersections

\mathbf{D} normale à l'arête $P_{i+1}^{(0)} - P_i^{(0)}$ de C_0

Projection de C_0 sur axe séparateur $P_i^{(0)} + t\mathbf{D}$ est $[-\mu, 0]$ avec $\mu > 0$

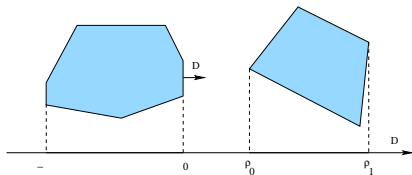
Projection de C_1 sur cette ligne est $[\rho_0, \rho_1]$

⇒ il suffit de tester si $\rho_0 > 0$ pour le test de séparation

⇒ ne nécessite pas la connaissance de μ

⇒ pas besoin d'effectuer les projections des sommets de C_0

⇒ sommets de C_1 projetés : si les valeurs sont négatives, D n'est plus considéré comme une séparation ; si toutes les valeurs sont positives, D est un axe séparateur



Polygones convexes : test des intersections

Dans le pire cas (pas d'intersection entre les polygones) :

N_0 arêtes normales de C_0 projetées, chacune engendrant N_1 projections pour C_1

N_1 arêtes normales de C_1 projetées, chacune engendrant N_0 projections pour C_0

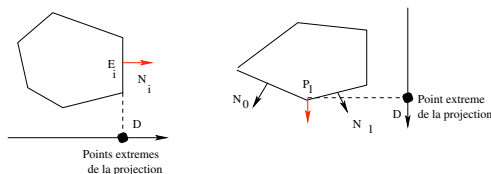
Nombre total de projections : $2N_0N_1 \ll (N_0 + N_1)^2$

↔ algorithme pourrait être amélioré en connaissant un point extrême de la projection des polygones

Polygones convexes : test des intersections

Si $\mathbf{D} = \mathbf{N}_i$; alors tous les points de l'arête \mathbf{E}_i sont des points extrêmes de la projection du polygone par rapport à cette direction

Si \mathbf{D} se trouve strictement entre \mathbf{N}_0 et \mathbf{N}_1 , P_1 correspond à l'unique point extrême dans cette direction



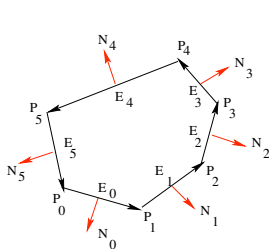
↔ applique ces arguments de manière similaire à toutes directions \mathbf{D} comprises entre deux normales consécutives

Polygones convexes : test des intersections

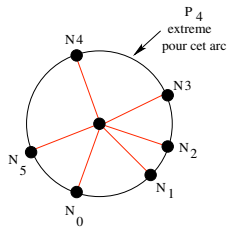
Cercle composé de points correspondants aux normales

Ces points décomposent le cercle en arcs, chaque arc correspondant à un sommet extrême du polygone

L'extrémité d'un arc correspondant à une arête extrême



Polygone convexe



Dual polaire du polygone

Polygones convexes : test des intersections

La recherche de l'ensemble des points extrêmes en testant les directions \mathbf{D} peut se résumer ainsi :

- le sommet P_i est optimal quand $\mathbf{N}_{i-1}^\perp \cdot \mathbf{D} > 0$ et $\mathbf{N}_i^\perp \cdot \mathbf{D} < 0$
- l'arête E_i est optimale quand $\mathbf{N}_{i-1}^\perp \cdot \mathbf{D} = 0$ et $\mathbf{N}_i^\perp \cdot \mathbf{D} < 0$

Ce qui peut encore être résumé par :

- le sommet P_i est optimal quand $\mathbf{N}_{i-1}^\perp \cdot \mathbf{D} \geq 0$ et $\mathbf{N}_i^\perp \cdot \mathbf{D} < 0$

avec $(x, y)^\perp = (-y, x)$

↔ obtention de N arcs correspondants aux N côtés du polygone

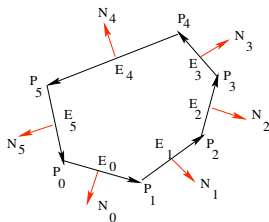
↔ si on effectue d'abord la recherche de ces arcs, puis le test pour savoir si \mathbf{D} se trouve sur un arc, mais engendrerait une recherche en $O(N)$: on va utiliser un arbre binaire de partitionnement de l'espace (*BSP tree*) avec recherche en $O(\log N)$

Arbres BSP

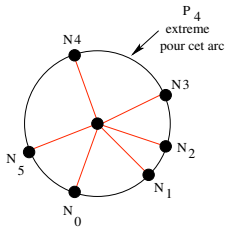
Pour construire l'arbre BSP, le disque est récursivement divisé en secteurs, chaque division correspondant au test $\mathbf{N}_j^\perp \cdot \mathbf{D} \geq 0$

Les sommets du cercle sont N_i pour $0 \leq i \leq 5$

Les arcs sont notés $A_{ij} = \langle N_i, N_j \rangle$ avec $j = (i + 1) \bmod 6$



Polygone convexe



Dual polaire du polygone

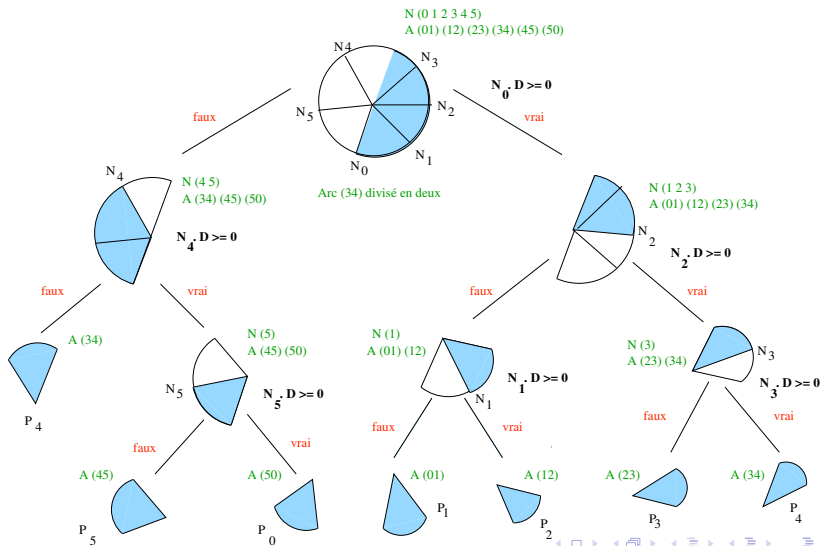
Etapes de la construction de l'arbre BSP

- Initialisation de la racine avec l'ensemble des sommets et arcs
- Considère la première normale de la liste : N_0
- Calcul de $\mathbf{N}_0^\perp \cdot \mathbf{D} \geq 0$ avec \mathbf{D} vecteur pris dans la liste restantes des normales :
 - si $\mathbf{N}_0^\perp \cdot \mathbf{N}_i \geq 0$, \mathbf{N}_i placée dans l'ensemble des vecteurs utilisés pour créer le fils droit de la racine ;
 - sinon \mathbf{N}_i placée dans l'ensemble des vecteurs utilisés pour créer le fils gauche de la racine.

Etapas de la construction de l'arbre BSP

- L'ensemble des arcs est également mis à jour pour chacun des noeuds créés. Les extrémités de l'arc A_{ij} correspondent aux normales \mathbf{N}_i et \mathbf{N}_j pour lesquelles les calculs de $d_i = \mathbf{N}_0^\perp \cdot \mathbf{N}_i$ et $d_j = \mathbf{N}_0^\perp \cdot \mathbf{N}_j$ viennent d'être faits :
 - si $d_i \geq 0$ et $d_j \geq 0$, arc A_{ij} placé dans le noeud du fils droit ;
 - si $d_i \leq 0$ et $d_j \leq 0$, arc A_{ij} placé dans le noeud du fils gauche ;
 - si $d_i d_j < 0$, arc A_{ij} placé dans les deux noeuds enfants.
- Puis le processus est appliqué récursivement.
- Au final, chaque noeud intérieur de l'arbre correspond au test d'une normale ; et chaque feuille correspond au sommet j de l'arête A_{ij} .

Etapas de la construction de l'arbre BSP - Exemple



Arbres BSP

Ensuite, il suffit de parcourir l'arbre pour trouver le noeud extrême correspondant à un potentiel axe séparateur **D**

↔ parcours d'un arbre binaire en $O(\log N)$

Polyèdres convexes (3D)

Problématique de la séparation des polygones convexes est étendue aux polyèdres convexes

Pour deux polyèdres convexes, le test de séparation ne doit être effectué que pour un nombre fini de directions

Seulement quelques contacts à considérer (non-interpénétration) :

- face-face, face-arête, face-sommet
- arête-arête, arête-sommet
- sommet-sommet

↔ il faut considérer comme directions potentielles, les vecteurs normaux aux faces, et vecteurs issus du produit vectoriel de deux arêtes (une de chacun des polyèdres)

Objets en mouvement à vitesse linéaire constante

Nous avons considéré le cas de polygones ou polyèdres immobiles

Considérons le cas d'objets en mouvement à vitesse linéaire constante et de vitesse angulaire nulle

- ↪ on va prédire le moment de l'impact entre les deux objets
- ↪ extension de la méthode de séparation des axes

Séparation entre polygones convexes

Soient C_0 et C_1 deux polygones convexes

Vitesses linéaires \mathbf{V}_0 et \mathbf{V}_1

Par projection, il est possible de savoir si il y aura une collision au temps $T \geq 0$

Si c'est le cas, il est possible de connaître le moment de l'impact

Séparation entre polygones convexes

Considérons pour simplifier que

- C_0 est immobile
- C_1 a une vitesse \mathbf{V} (rq : $\mathbf{V} = \mathbf{V}_1 - \mathbf{V}_0$)

Considérons que l'impact a lieu pour $T = 0$

Directions potentielles à considérer : normales aux arêtes des polygones (pour les polyèdres il faut rajouter comme direction potentielles, les vecteurs issus du produit vectoriel de deux arêtes)

Projection de C_1 sur la direction \mathbf{D} (non perpendiculaire à \mathbf{V})
bouge également avec une vitesse définie par $\sigma = (\mathbf{V} \cdot \mathbf{D}) / \|\mathbf{D}\|^2$

↪ si la projection de C_1 s'éloigne de celle de C_0 , il n'y aura jamais d'intersection

↪ intersection ne se produit que si la projection de C_1 se rapproche de celle de C_0

Séparation entre polygones convexes - Idée

Si les deux polygones **s'intersectent pour la première fois au temps** $T_{\text{first}} > 0$, alors leurs projections ne seront pas séparées

Un instant avant ce contact, les polygones sont séparés, c'est-à-dire qu'il existe au moins un **axe de séparation au temps** $T_{\text{first}} - \epsilon$ avec $\epsilon > 0$

Si deux polygones **s'intersectent pour la dernière fois au temps** $T_{\text{last}} > 0$, alors leurs projections ne seront pas séparées mais le seront à l'instant d'après

Il existe au moins un **axe de séparation pour les polygones au temps** $T_{\text{last}} + \epsilon$ avec $\epsilon > 0$

Séparation entre polygones convexes - Idée

T_{first} et T_{last} peuvent être traqués pour chacune des directions potentielles

⇒ si $T_{\text{first}} \leq T_{\text{last}}$: première intersection à T_{first}

⇒ si $T_{\text{first}} > T_{\text{last}}$: pas d'intersection

Séparation entre polygones convexes - Algorithme

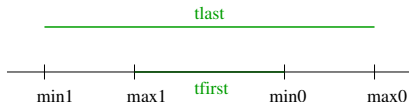
Test d'intersection entre C_0 et C_1

- Soient $T_{\text{first}} = 0$ et $T_{\text{last}} = \infty$
- Considère les normales des arêtes de C_0 comme directions potentielles
 - Calcul de la projection de C_0 : \min_0, \max_0
 - Calcul de la projection de C_1 : \min_1, \max_1
 - Calcul de la vitesse $(\mathbf{V} \cdot \mathbf{D}) / \|\mathbf{D}\|^2$
 - Regarde si il y a une intersection ou non
 - mise à jour de T_{first} et T_{last}
- Même processus effectué avec les normales des arêtes de C_1

Séparation entre polygones convexes - Algorithme

Test de non-intersection

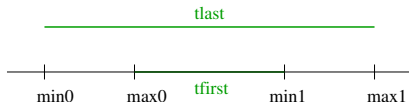
- Si $max_1 < max_0$ (intervalle de C_1 à gauche de celui de C_0)
 - Si vitesse ≤ 0 : pas d'intersection
 - $t = (min_0 - max_1) / vitesse$
 - Si $t > t_{first}$, $t_{first} = t$
 - Si $t_{first} > t_{max}$: pas d'intersection
 - $t = (max_0 - min_1) / vitesse$
 - Si $t < t_{last}$, $t_{last} = t$
 - Si $t_{first} > t_{last}$: pas d'intersection



Séparation entre polygones convexes - Algorithme

Test de non-intersection

- Sinon si $max_0 < min_1$ (intervalle de C_1 à droite de celui de C_0)
 - Si vitesse ≥ 0 : pas d'intersection
 - $t = (max_0 - min_1) / vitesse$
 - Si $t > t_{first}$, $t_{first} = t$
 - Si $t_{first} > t_{max}$: pas d'intersection
 - $t = (min_0 - max_1) / vitesse$
 - Si $t < t_{last}$, $t_{last} = t$
 - Si $t_{first} > t_{last}$: pas d'intersection



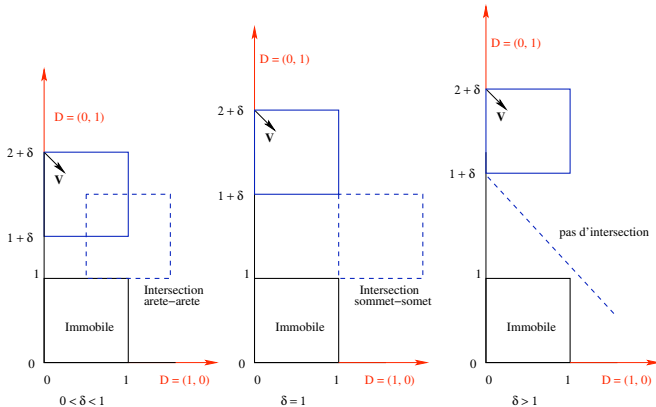
Séparation entre polygones convexes - Algorithme

Test de non-intersection

- Sinon (intervalles se chevauchent)
 - Si la vitesse > 0
 - $t = (max_0 - min_1) / vitesse$
 - Si $t < t_{last}$, $t_{last} = t$
 - Si $t_{first} > t_{last}$: pas d'intersection
 - Sinon si la vitesse < 0
 - $t = (min_0 - max_1) / vitesse$
 - Si $t < t_{last}$, $t_{last} = t$
 - Si $t_{first} > t_{last}$: pas d'intersection
- Dans tous les autres cas : intersection

Séparation entre polygones convexes - Exemple

Soient 2 cubes : noir (immobile), bleu (mobile avec $\mathbf{V} = (1, -1)$)
 Intersection entre ces deux cubes dépend de la valeur de δ
 Deux directions potentielles : $(0, 1)$ et $(1, 0)$



Séparation entre polygones convexes - Exemple

Soient $T_{\text{first}} = 0$ et $T_{\text{last}} = \infty$

Considère la direction $D = (1, 0)$:

- Pour C_0 : $min_0 = 0$, $max_0 = 1$
- Pour C_1 : $min_1 = 0$, $max_1 = 1$
- Vitesse = $(\mathbf{V} \cdot \mathbf{D}) / \|\mathbf{D}\|^2 = ((1, -1) \cdot (1, 0)) / \|(1, 0)\|^2 = 1$
- Cas où les intervalles se chevauchent
 - Cas où la vitesse est positive
 - $t = (max_0 - min_1) / vitesse = 1$
 - $t = 1 < T_{\text{last}} = \infty \Rightarrow T_{\text{last}} = t = 1$
 - $T_{\text{first}} = 0 < T_{\text{last}} = 1 \Rightarrow$ ne peut rien conclure

Séparation entre polygones convexes - Exemple

Considère la direction $D = (0, 1)$:

- Pour C_0 : $min_0 = 0$, $max_0 = 1$
- Pour C_1 : $min_1 = 1 + \delta$, $max_1 = 2 + \delta$
- Vitesse = $(\mathbf{V} \cdot \mathbf{D}) / \|\mathbf{D}\|^2 = ((1, -1) \cdot (0, 1)) / \|(0, 1)\|^2 = -1$
- Cas où intervalle de C_1 à droite de celui de C_0
 - Cas où la vitesse est négative
 - $t = (max_0 - min_1) / vitesse = \delta$
 - $t = \delta > T_{first} = 0 \Rightarrow T_{first} = t = \delta$
 - $t = (min_0 - max_1) / vitesse = 2 + \delta$
 - $t = 2 + \delta > T_{last} = 1$ (avec $\delta > 0$)
 - Intersection ssi $T_{first} \leq T_{last} \Rightarrow \delta \leq 1$

↔ correspond aux trois exemples de la figure

Points de contacts entre polygones convexes

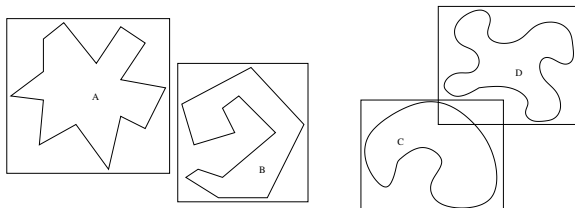
L'algorithme est ensuite légèrement modifié pour conserver la trace des arêtes ou des sommets dont la projection correspondait aux min ou max des intervalles de projection

On obtient ainsi l'ensemble des points de contact entre les polygones

Boîtes englobantes

Détection des collisions

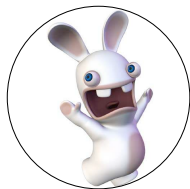
- Détection exacte : détection au niveau de l'objet
↳ extrêmement coûteux si objets formés de plusieurs centaines de polygones
- Détection entre volumes englobants (*Bounding Volume - BV*)
↳ objets encapsulés dans des boîtes
↳ permet de réduire le nombre de tests d'intersection



↳ pas besoin de tester plus précisément intersection entre A et B

Différents types de boîtes englobantes

AABB, sphère, DOP, OBB, spherical shell, coque convexe, prisme, cylindre, intersection d'autres BVs, ...



Sphere



AABB



OBB



6-DOP



Enveloppe convexe

Les volumes englobant sont calculés avant la boucle de simulation n'affectant pas les ressources de calcul en cours d'exécution

↪ stockés au même titre que la géométrie des objets de la scène

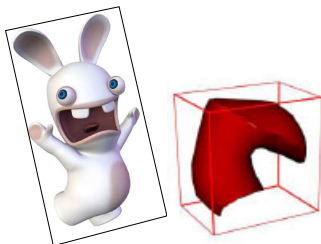
↪ ne doivent pas nécessiter beaucoup d'espace mémoire

Boîtes orientées englobantes (OBB)

Oriented Bounding Boxes (OBB) : représente la partie de l'espace occupée par l'objet

Terme orienté pour dire que les axes de la boîtes ne correspondent pas forcément à ceux de l'objet

Utilisation de ces boîtes englobantes pour optimiser la détection des collisions entre les objets de la scène



Hiérarchies de volumes englobants (BVHs)

Structure de données efficace pour la détection de collisions

Construit pour chaque objet de la scène

→ phase de pré-traitement

Décomposition récursive de l'ens des primitives d'un objet
(primitives = entités de l'objet graphiques : polygones, NURBS)

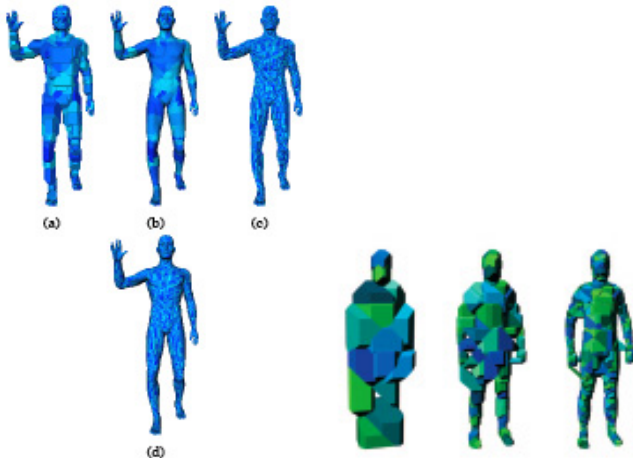
→ critère d'arrêt : reste 1 seule primitive (feuille)

A chaque noeud de l'arbre est associé

- un sous-ens. de primitives de l'objet
- un BV englobant le sous-ens (le plus petit par rapport à un type de BV donné)

Exemple de BVHs

Trois niveaux hiérarchiques : 18-DOP



Construction des BVHs hiérarchiques

Décomposition récursive de l'ensemble des primitives d'un objet jusqu'à un certain critère

- Peut utiliser une heuristique simple :
 - Chaque polygone est représenté par son centre
→ ensemble de points noté B
 - Calcul des composantes principales de cet ensemble
→ vecteurs propres de la matrice de covariance
 - Choix du plus grand (celui ayant la variance la plus importante)
 - Place un plan orthogonal à cet axe principal en passant par le barycentre de B
→ B est alors divisé en deux sous-ensembles

Algorithme de détection pour 2 objets avec BVHs A et B

Fonction Traverse(A, B)

Si A et B ne se chevauchent pas **Alors**

return

FinSi

Si A et B sont des feuilles **Alors**

return intersection primitives incluses par A et B

Sinon

Pour chaque enfant A[i] **Faire**

traverse(A[i], B)

FinPour

FinSi

Boîtes orientées englobantes (OBB)

OBB possède 6 faces, 12 arêtes et 8 sommets

OBB défini par son centre \mathcal{C} (origine du système de coordonnées)

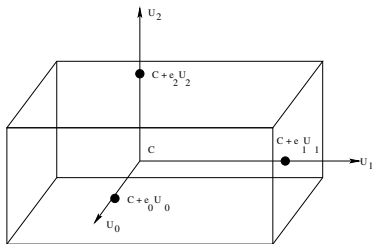
Axes orthogonaux U_i pour $i = 0, 1, 2$

Directions = vecteurs normaux aux faces de l'OBB

8 sommets de l'OBB définis par :

$$\mathcal{P} = \mathcal{C} + \sigma_0 e_0 \mathbf{U}_0 + \sigma_1 e_1 \mathbf{U}_1 + \sigma_2 e_2 \mathbf{U}_2$$

avec $|\sigma_i| = 1$ pour $i = 0, 1, 2$ (8 choix pour le signe de σ_i)



Boîtes orientées englobantes (OBB) - Intersection

Souhaite savoir quand deux OBBs se coupent

Il y a 156 axes séparateurs potentiels : 6 faces normales pour le premier, 6 faces pour le second et $144 = 12 \times 12$ arête-arête

On peut réduire ce nombre de tests en utilisant la symétrie

Il y a que 3 directions différentes pour les arêtes

On a 3 paires de faces parallèles

↪ besoin de considérer uniquement 3 faces normales pour un OBB

On a donc **15 tests d'intersections** : 3 faces pour le premier, 3 faces pour le second et $9 = 3 \times 3$ arête-arête

Boîtes orientées englobantes (OBB) - Intersection

Doit projeter un OBB sur un axe séparateur potentiel $\mathcal{Q} + t\mathbf{D}$

Symétrie permet de déterminer les intervalles de projection

Pour trouver un point extrême, il suffit de déterminer un sommet \mathcal{P} qui maximise :

$$\mathbf{D} \cdot (\mathcal{P} - \mathcal{Q}) = \mathbf{D} \cdot (\mathcal{C} - \mathcal{Q}) + \sigma_0 e_0 \mathbf{D} \cdot \mathbf{U}_0 + \sigma_1 e_1 \mathbf{D} \cdot \mathbf{U}_1 + \sigma_2 e_2 \mathbf{D} \cdot \mathbf{U}_2$$

Boîtes orientées englobantes (OBB) - Intersection

$\sigma_i = -1$ ou 1 et on souhaite maximiser $\sigma_i \mathbf{D} \cdot \mathbf{U}_i$:

$$\text{si } \mathbf{D} \cdot \mathbf{U}_i > 0 \Rightarrow \sigma_i = 1$$

$$\text{si } \mathbf{D} \cdot \mathbf{U}_i < 0 \Rightarrow \sigma_i = -1$$

$$\text{si } \mathbf{D} \cdot \mathbf{U}_i = 0 \Rightarrow \sigma_i = -1 \text{ ou } 1$$

$$\max \mathbf{D} \cdot (\mathcal{P} - \mathcal{Q}) = \mathbf{D} \cdot (\mathcal{C} - \mathcal{Q}) + e_0 |\mathbf{D} \cdot \mathbf{U}_0| + e_1 |\mathbf{D} \cdot \mathbf{U}_1| + e_2 |\mathbf{D} \cdot \mathbf{U}_2|$$

$$\min \mathbf{D} \cdot (\mathcal{P} - \mathcal{Q}) = \mathbf{D} \cdot (\mathcal{C} - \mathcal{Q}) - e_0 |\mathbf{D} \cdot \mathbf{U}_0| - e_1 |\mathbf{D} \cdot \mathbf{U}_1| - e_2 |\mathbf{D} \cdot \mathbf{U}_2|$$

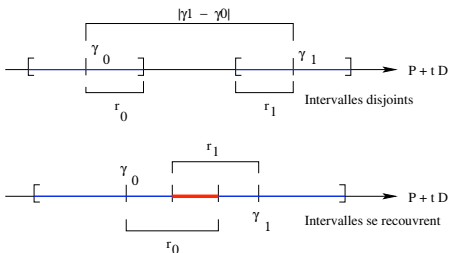
Projection sur l'intervalle $[\gamma - r, \gamma + r]$ avec :

$$\gamma = \mathbf{D} \cdot (\mathcal{C} - \mathcal{Q}) \text{ et } r = \sum_{i=0}^2 e_i |\mathbf{D} \cdot \mathbf{U}_i|$$

Boîtes orientées englobantes (OBB) - Intersection

Soient deux OBBs de centres C_0 et C_1 , d'axes A_i et B_i
Projection sur la ligne $Q + tD$ donne les intervalles

$$[\gamma_0 - r_0, \gamma_0 + r_0] \text{ et } [\gamma_1 - r_1, \gamma_1 + r_1]$$



Si $|\gamma_1 - \gamma_0| > r_0 + r_1 \Rightarrow$ intervalles disjoints

Si $|\gamma_1 - \gamma_0| < r_0 + r_1 \Rightarrow$ intervalles se chevauchent

Si $|\gamma_1 - \gamma_0| = r_0 + r_1 \Rightarrow$ intervalles se touchent

Boîtes orientées englobantes (OBB) - Intersection

Soit $r = |\gamma_1 - \gamma_0| = |\mathbf{D} \cdot (\mathcal{C}_1 - \mathcal{Q}) - (\mathcal{C}_0 - \mathcal{Q})| = |\mathbf{D} \cdot (\mathcal{C}_1 - \mathcal{C}_0)|$

↔ besoin de spécifier uniquement direction \mathbf{D}

↔ pas besoin de spécifier un point \mathcal{Q} sur la ligne

$$r_0 = \sum_{i=0}^2 a_i |\mathbf{D} \cdot \mathbf{A}_i|, r_1 = \sum_{i=0}^2 b_i |\mathbf{D} \cdot \mathbf{B}_i|$$

Intervalles disjoints ssi :

$$|\mathbf{D} \cdot (\mathcal{C}_1 - \mathcal{C}_0)| = |\mathbf{D} \cdot \Delta| > r_0 + r_1 = \sum_{i=0}^2 a_i |\mathbf{D} \cdot \mathbf{A}_i| + \sum_{i=0}^2 b_i |\mathbf{D} \cdot \mathbf{B}_i|$$

Axe séparateur potentiel : $\mathbf{D} = \mathbf{A}_i \times \mathbf{B}_j$ en prenant une arête de chacun des deux OBBs

A noter que si $\mathbf{D} = \mathbf{A}_0$, alors $r_0 = a_0$

Intersection entre OBBs

15 tests d'axes séparateurs pour les intersections OBB-OBB

Soient $\Delta = \mathcal{C}_1 - \mathcal{C}_0$ $\alpha_i = \Delta \cdot \mathbf{A}_i$, $\beta_i = \Delta \cdot \mathbf{B}_i$, $c_{ij} = \mathbf{A}_i \cdot \mathbf{B}_j$

Intervalles disjoints ssi $r > r_0 + r_1$

D	r_0	r_1	r
A_0	a_0	$b_0 c_{00} + b_1 c_{01} + b_2 c_{02} $	$ \alpha_0 $
A_1	a_1	$b_0 c_{10} + b_1 c_{11} + b_2 c_{12} $	$ \alpha_1 $
A_2	a_2	$b_0 c_{20} + b_1 c_{21} + b_2 c_{22} $	$ \alpha_2 $
B_0	$a_0 c_{00} + a_1 c_{10} + a_2 c_{20} $	b_0	$ \beta_0 $
B_1	$a_0 c_{01} + a_1 c_{11} + a_2 c_{21} $	b_1	$ \beta_1 $
B_2	$a_0 c_{02} + a_1 c_{12} + a_2 c_{22} $	b_2	$ \beta_2 $
$A_0 \times B_0$	$a_1 c_{20} + a_2 c_{10} $	$b_1 c_{02} + b_2 c_{01} $	$ c_{10}\alpha_2 - c_{20}\alpha_1 $
$A_0 \times B_1$	$a_1 c_{21} + a_2 c_{11} $	$b_0 c_{02} + b_2 c_{00} $	$ c_{11}\alpha_2 - c_{21}\alpha_1 $
$A_0 \times B_2$	$a_1 c_{22} + a_2 c_{12} $	$b_0 c_{01} + b_1 c_{00} $	$ c_{12}\alpha_2 - c_{22}\alpha_1 $
$A_1 \times B_0$	$a_0 c_{20} + a_2 c_{00} $	$b_1 c_{12} + b_2 c_{11} $	$ c_{20}\alpha_0 - c_{00}\alpha_2 $
$A_1 \times B_1$	$a_0 c_{21} + a_2 c_{01} $	$b_0 c_{12} + b_2 c_{10} $	$ c_{21}\alpha_0 - c_{01}\alpha_2 $
$A_1 \times B_2$	$a_0 c_{22} + a_2 c_{02} $	$b_0 c_{11} + b_1 c_{10} $	$ c_{22}\alpha_0 - c_{02}\alpha_2 $
$A_2 \times B_0$	$a_0 c_{10} + a_1 c_{00} $	$b_1 c_{22} + b_2 c_{21} $	$ c_{00}\alpha_1 - c_{10}\alpha_0 $
$A_2 \times B_1$	$a_0 c_{11} + a_1 c_{01} $	$b_0 c_{22} + b_2 c_{20} $	$ c_{01}\alpha_1 - c_{11}\alpha_0 $
$A_2 \times B_2$	$a_0 c_{12} + a_1 c_{02} $	$b_0 c_{21} + b_1 c_{20} $	$ c_{02}\alpha_1 - c_{12}\alpha_0 $

Conclusion - Traitement des collisions

Trois types de collision déterminés en fonction du signe de $\mathbf{N} \cdot \mathbf{V}_A$:

- *Colliding contact* - contact en collision
- *Resting contact* - contact établi
- *Separation contact* - rupture de contact

Deux types de réponse à la collision :

- Force d'impulsion pour un contact en collision
- Force de contact pour un contact établi

Conclusion - Détection des collisions

Utilisation de la méthode des axes séparateurs

Utilisation de structure de données de type BSP

Prédiction du moment de l'impact pour des polyèdres en mouvement

Utilisation de volumes englobants pour optimiser la détection des collisions