1 Detekcie kolízie

- V systémoch častíc a objektov môžu prirodzene vznikať kolízie. Detekcia kolízií sa zaujíma o to kde, kedy a ktoré objekty sú v kolízií, nerieši už teda následnú reakciu (odrazenie, rozbitie atď.).
- Detekcia kolízie sa delí na tri štádia broad, mid a narrow phase. Mid phase je pomocné štádium, ktoré spresňuje výsledky broad phase. To poskytuje iba hrubý odhad objektov, ktoré môžu byť v kolízií. Mid phase uľačí prácu záverečného štádia narrow phase, ktoré podáva finálny verdikt.
- V každom štádiu sa používajú iné algoritmy a postupy. Algoritmy vieme deliť aj podľa toho, či scénu vnímajú len staticky v nejakom časovom momente, alebo sú pseudo-dynamické až dynamické, ktoré využívajú časovú informáciu, teda berú do úvahy, či boli objekty v kolízií v predchádzajúcom časovom bode, ako boli ďaleko a aká bola ich rýchlosť.
- Algoritmy využívajú stratégie ako rozdelenie priestoru na podpriestory (spatial partitioning), obalenie zložítých objektov jednoduchšími obálkami, ktoré môžu mať prípadne viacero úrovní/hierarchiu (bounding volume hierarchies), projekcie objektov na osi/roviny, ich triedenie a následná kontrola (coordinate sorting), či sledovanie príznakov.

2 Broad Phase

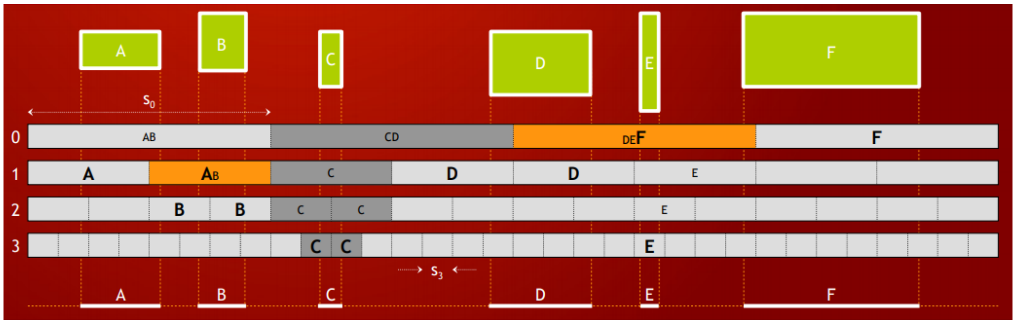
Hrubý odhad objektov, ktoré môžu byť v kolízií. Na vstupe dostane zoznam objektov v scéne a ďalej posiela zoznam dvojíc. Eliminuje objekty, ktoré sú príliš ďaleko od seba a určite nie sú v kolízií, takže už nemusia byť v ďalších štádiach zvažované. Okrem techník využívajúcich mriežku (uniformnú, hierarchickú), poznáme aj metódy s komplexnejším delením priestoru (dynamic BSP, kd trees, octrees) a techniky, ktoré okraje objektov projektujú na osi a projekcie následne triedia a porovnávajú (sweep and prune, range search).

(a) **Hierarchická mriežka** - Vylepšenie uniformnej mriežky, v ktorej sa priestor jednoducho rozdelil na mriežku s konštantnou veľkosťou bunky a ak AABB obálky dvoch objektov zasahovali do jednej bunky, vyhlásila sa možná kolízia. Problémoch tohto prístupu je, že nevieme nájsť vhodnú veľkosť bunky tak, aby sa dobre vyhodnocovali kolízie veľkých aj malých objektov.

Teraz máme viacero mriežok s veľkosťou bunky, ktorá sa zmenšuje faktorom 2. Majme napríklad 4 mriežky s rozlíšeniami $\frac{1}{2^k}; 0 <= k < 4$. Veľkosti buniek mriežok sú potom $s_0 = \frac{1}{2^0} = 1.0; \ldots; s_3 = \frac{1}{2^3} = 0.125$.

Pri vkladaní objektu **C** do mriežky najskôr nájdeme jeho rozlíšenie $Res(\mathbf{C})$. Objekt má obálku $AABB(\mathbf{C}) = (C_{x-}, C_{y-}, C_{z-}, C_{x+}, C_{y+}, C_{z+})$. Nájdeme veľkosť najdlhšej strany obálky $Size(\mathbf{C}) = max(C_{x+} - C_{x-}, C_{y+} - C_{y-}, C_{z+} - C_{z-})$. Rozlíšením objektu potom bude $Res(\mathbf{C})$ práve vtedy keď $a \leq (Size(\mathbf{C}/s_i)) \leq b$, kde typicky $a = 0.5; b = 1$. Zvolíme teda úroveň mriežky takú, že rozdieľ veľkosti medzi najdlhšiu stranou obálky a veľkosťou bunky danej mriežky je rozumne malý. Objekt **C** zaznačíme do tejto mriežky a aj všetkých mriežok, ktoré sú hierarchicky vyššie. **Špeciálne** ho zaznačíme v tej úrovni, ktorá prislúcha jeho rozlíšeniu.

V možnej kolízií sú tie dvojice objektov, ktorých **ID** sa pri pridávaní vyskytuje v jednej bunke niektorej z mriežok v kombinácií **špeciálne**×*normálne*, alebo **špeciálne**×**špeciálne**.



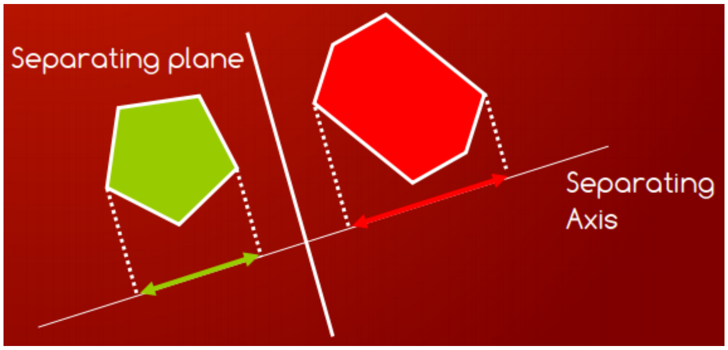
Ak by boli v kolízií objekty v kombinácií *normálne*$\times$*normálne*, boli by nahlásené už niekde na nižšej úrovni mriežok s menšími bunkami, ktoré prislúchajú ich rozlíšeniu. Pritom ak by sme používali iba uniformú mriežku s najväčšími bunkami, nevedeli by sme takýto prípad jednoznačne vyhodnotiť a aj takáto dvojica by sa posielala do mid-phase. V tom spočíva výhoda hierarchických mriežok.

Nevýhodami tohto prístupu sú väčšie pamaťové nároky, keďže musíme mať uložených viacero mriežok a pri pridávani/odobernaí objektov ich musíme zaznačiť na všetkých potrebných úrovniach.

Nech $R = (s^+/s^-)$, kde s^+ je veľkosť obálky najväčšieho a s^- najmenšieho objektu v scéne. Potrebujeme potom $j = log(R)$ mriežok, aby sa pre každý objekt dalo nájsť vhodné rozlíšenie. Keďže j je konštanta známa na začiatku, pridávanie/odoberanie objektov potom priebieha v konštantom čase $O(j)$. Časová zložitosť je teda lineárna od počtu objektov $O(n)$.

(b) **Nutná a postačujúca podmienka, kedy sú objekty v kolízií** - Seperating Plane Theorem. Ide o definíciu, ktorú využíva metóda Sweep-And-Prune v broad phase, ale aj ďalšie metódy v neskorších štádiach. Tvrdí, že dva konvexné objekty nie sú v kolízií *ak a iba ak* medzi nimi existuje rovina, ktorá ich oddeĺuje. Každý objekt je teda na opačných stranách tejto roviny.

Ekvivalentným (Separating Duality Principle), no možno trocha menej intuitívnym tvrdením je Separating Axis Theorem ktorá hovorí, že dva objektý sú separované *ak a iba ak* existuje os, na ktorej sú projekcie obálok týchto objektov separované.



3 Mid Phase

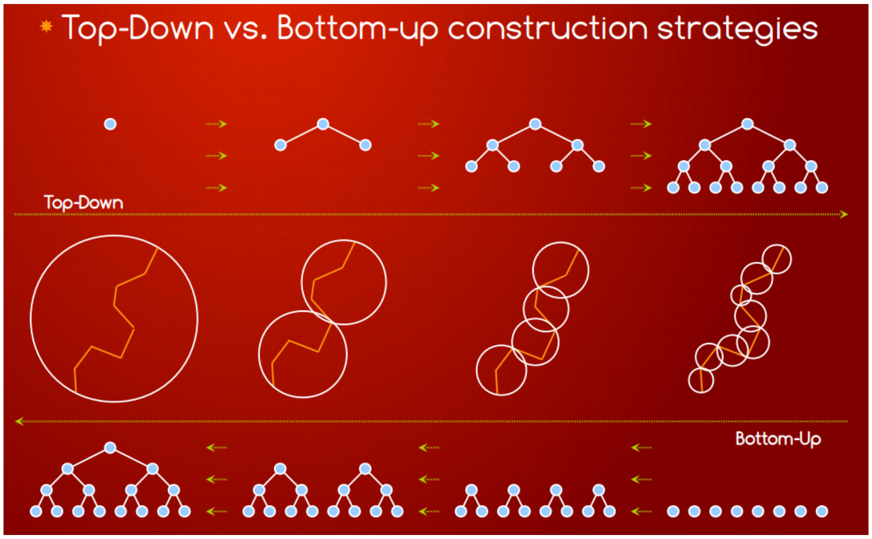
(a) **Hierarchie obálok** - mid phase štádium sa snaží upresniť zoznam dvojíc (možno) kolidujúcich objektov, jednou z techník je zaobalenie zložitej geometrie do hierarchie jednoduchších obálok. Hierarchia obálok (Bouding Volume Hierarchy), známa aj ako strom obálok (Bounding Volume Tree) je m-nárny strom $T = T_1, \ldots, T_m, BV, G$, ktorého vrcholy T_i obsahujú obálku BV, ktorá pokrýva nejakú časť geometrie G. Vlastnosti stromu:

- každá nižšia úroveň hierarchie by mala reprezentovať presnejší odhad geometrie
- detské vrcholy by mali spoločne pokrývať rovnakú časť geometrie, ako ich rodič
- obálky by mali byť invariantné na rigidný pohyb - transláciu/rotáciu, zachovávajúcu vzdialenosti medzi bodmi. To umožňuje aktualizáciu v dynamických systémoch, nemusíme vytvárať strom vždy nanovo.

Tvar obálky je špecifický, určuje sa na začiatku. Obálka by:

- mala byť jednoduchá, konvexná geometria
- tesne obaliť aj objekty, ktoré nemajú sférický/guľový tvar
- umožnovať rýchli výpočet prekryvu, kolízie
- vedieť sa posúvať a rotovať spoločne s objektom

Najjednoduchšou možnosťou je *sféra/guľa*, ktorá ale nespĺňa bod 2 a je nepresná na podlhovasté predmety (napr. šíp). *OBB* má už o niečo lepšie pokrytie a výhodou je, že sa môže rotovať spolu s objektom. O ešte lepší odhad tvaru geometrie sa snaží *kDop*, ktorý však treba po každej rotácií počítať nanovo, pretože smery hrán sú fixne dané. Najtesnejšou konvexnou obálkou je *Convex hull*, ktorý je ale náročný na výpočet.



Konštrukcia hierarchie by mala prebiehať automaticky. Na vstupe je geometria, ktorú chceme zaobaliť, poznáme dve stratégie.

- **zhora nadol** Zvolíme vetviaci faktor m (+ ďalšie prípadné parametre). Následne obalíme celú geometriu koreňovou BV obálkou vybraného tvaru. Tento vrchol následne vhodnou stratégiou rozdelíme do m podvrcholov. Mali by sme rozdeliť oblasti, ktoré majú medzi sebou najväčší rozptyl vrcholov a vzniknuté podobálky by mali mať podobný objem. Postup rekurzívne opakujeme, až dokým nedosiahneme zastavovacie kritérium (objem vrchola je už malý).

  Výhodou tohto prístupu je jednoduchá myšlienka a implementácia, no nevýhodou sú citlivosť na vetviaci faktor m a zastavovacie kritérium.

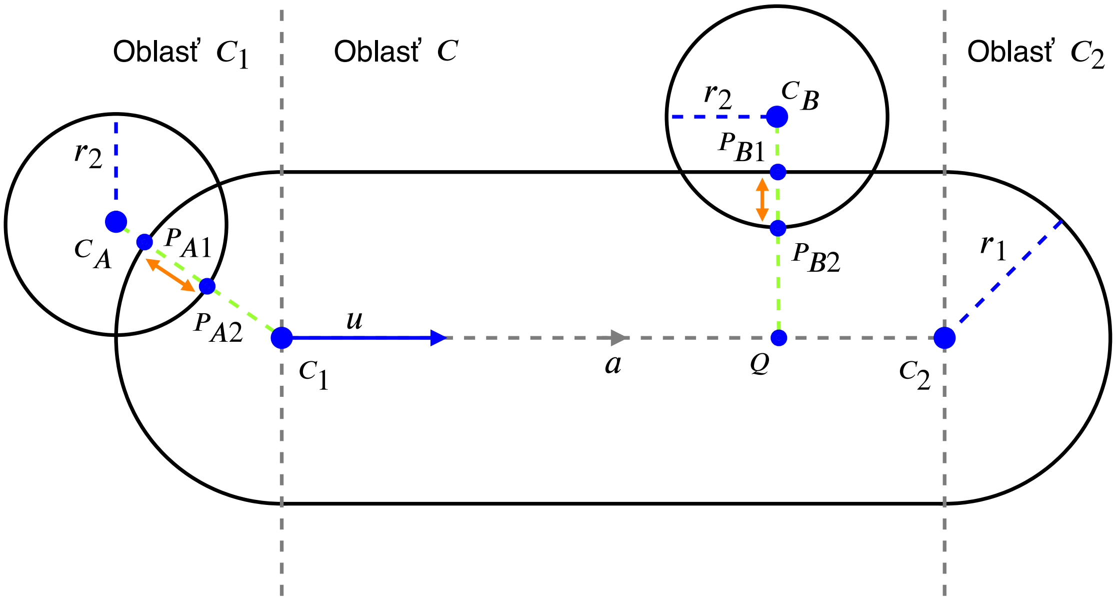
- **zdola nahor** Postup využívajúci unsupervised learning algoritmus - clustering. Začneme definovaním clustering faktoru m (+ ďalšie prípadné parametre). Najmenšie podčasti geometrie pokryjeme BVs. Pomocou clusteringu nájdeme m najbližších BVs. To zahŕňa výpočet vzdialeností stredov týchto obálok, aj vzdialenosti ich povrchov. Následne nájdené zhluky spojíme do jedného rodičovského BV tak, aby pokrýval všetky vrcholy detských BVs, prípadne geometrie, ktorú pokrývali. Tento proces opakujeme, až pokým sa nenájde jeden spoločný koreňový vrchol.

  Výhodou je, že tento proces väčšinou nájde optimálne vyvážený strom/hierarchiu obálok. Nevýhodou je, že iteračný clustering algoritmus môže byť pomalý.

Keď už máme pre všetky objekty vytvorené hierarchické stromy, na kontrolu kolízií sa použije algoritmus *Tandem Traversal.*

(b) Voronoiove oblasti v kolízii (vysvetlenie na kolízií guľa a kapsula)

V Mid Phase štádiu riešime kolízie konvexných obálok a jedným z možných prípadov je kolízia kruhu a kapsuly (gule a kapsuly v 3D). Projekciou stredu sféry (na obrázku C_A, C_B) na stredovú os kapsuly $a = C_2 - C_1$ zistíme, v ktorej voroinovej oblasti sa guľa nachádza. Ďalej získame vektor jednotkovej dĺžky $u = norm(a)$, smerový vektor $v = C - C_1$, a z toho už vieme pomocou skalárneho súčinu spočítať dĺžku projekcie $q = u \cdot v$.



Môžu nastať tri prípady:

- $q < 0$: V tomto prípade sa kruh nachádza vo voroinovej oblasti C_1 a tento prípad riešime ako kolíziu kruhu s kruhom. Na obrázku tento prípad nastane, keď vo výpočte smerového vektoru za C dosadíme stred kruhu C_A. Vtedy zoberieme vektor jednotkovej dĺžky $h = norm(v)$ a vypočítame body na povrchoch oboch gúl $P_{A1} = C_1 + r_1 h; P_{A2} = C_A - r_2 h$. Kontaktnú normálu sme už vypočítali, je ňou vektor h.
- $0 <= q <= |a|$: Prípad, keď kruh nie je v oblasti ani jedného z krajných kruhov kapsuly riešime ako kolíziu nekonečného cylindru a kruhu. Znova vysvetlíme na situácií z obrázku. Vypočítame bod na stredovej osi $Q = C_1 + qu$, smerový vektor $m = C_B - Q; b = norm(m)$. Z toho už vieme body na povrchoch obálok $P_{B1} = Q + r_1 b; P_{B2} = C_B - r_2 b$.
  Kontaktnú normálu vypočítame ako $n = norm(P_2 - P_1) = norm(Q - C_B) = -b$.
- $q > |a|$: Prípad, keď je kruh v oblasti C_2. Riešime analogicky ako prvý prípad, teda kolíziou dvoch kruhov, akurat teraz uvažujeme z kapsuly kruh so stredom v C_2.

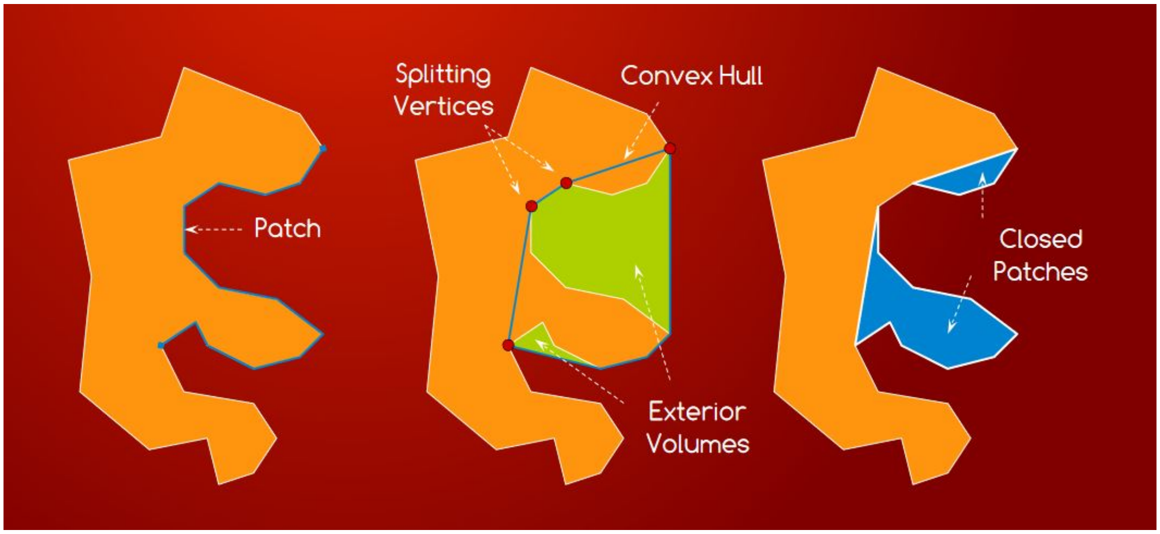
  Kontaktný bod je vo všetkých troch prípadoch priemer bodov na povrchoch obálok $p = 0.5(P_1 + P_2)$ a penetračná hĺbka $d = -|P_2 - P_1|$.

(c) Dekompozícia telesa na konvexné časti

Existuje viacero algoritmov, ktoré rozdeľujú nekonvexný útvar na malý počet "takmer konvexných" podčastí. Pre každú z týchto podčasti potom platí, že rozdiel medzi jej objemom a objemom jej konvexného obalu musí byť pod nejakou danou hranicou. Delenie sa zvyčajne vykonáva iba raz pred simuláciou. Pozrime sa na *ACD - Relaxation strategy*:

- Vyberieme deliacu stratégiu s delím zhora-nadol (napríkald OBB).
- Útvar rekurzívne delíme, až kým sa nedostaneme k listom stromu. Použijeme objemovú hranicu a zastavovacie kritérium.
- Získali sme tak veľký počet malých (takmer) konvexných častí.
- Vložíme ich do prioritného frontu vzhľadom na ich objem.
- Vyberáme prvú časť a snažíme sa ju spojiť s inou malou časťou tak, aby vzniknutá časť bola taktiež takmer konvexná.

Približná konvexná dekompozícia: Vyber "patch", vytvor konvexný obal, označ deliace vrcholy → vytvor podčasti. Vonkajší objem $\longrightarrow 0$.



4 Narrow Phase

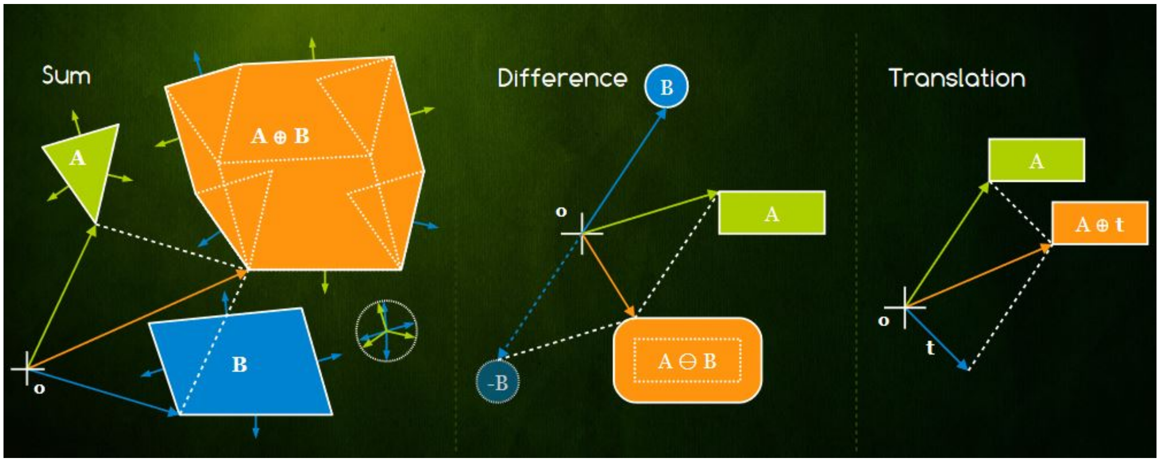
V Narrow Phase zisťujeme, ktoré z objektov vo vstupnom zozname párov potenciálne kolidujúcich objektov sú skutočne v kolízii a nekolidujúce páry odstránime. Ďalej určíme blízkosť/kontaktné informácie pre naše objekty (napríklad body, v ktorých sa objekty pretínajú, normálu kolízie, hĺbku penetrácie, atď) a nájdeme perzistetné kontakty. Vo výsledku dostaneme zoznam oblastí kontaktu s potrebnými informáciami o blízkosti kolidujúcich objektov.

(a) Minkowskeho priestor

Množina S bodov $p \in R^n$ je nazývaná konvexnou a ohraničenou, ak pre každé dva body a a b platí, že úsečka ab leží vo vnútri S a vzdialenosť $|a - b|$ je konečná. Teda platí: $a \in S \wedge b \in S \wedge t \in (0, 1) \implies (1 - t)a + tb \in S \wedge |a - b| \leq \beta$, kde S musí byť spojitá, ale nie hladká.

Majme dva konvexné objekty A a B. Definujme Minkowskeho súčet, rozdiel a preklad nasledovne:

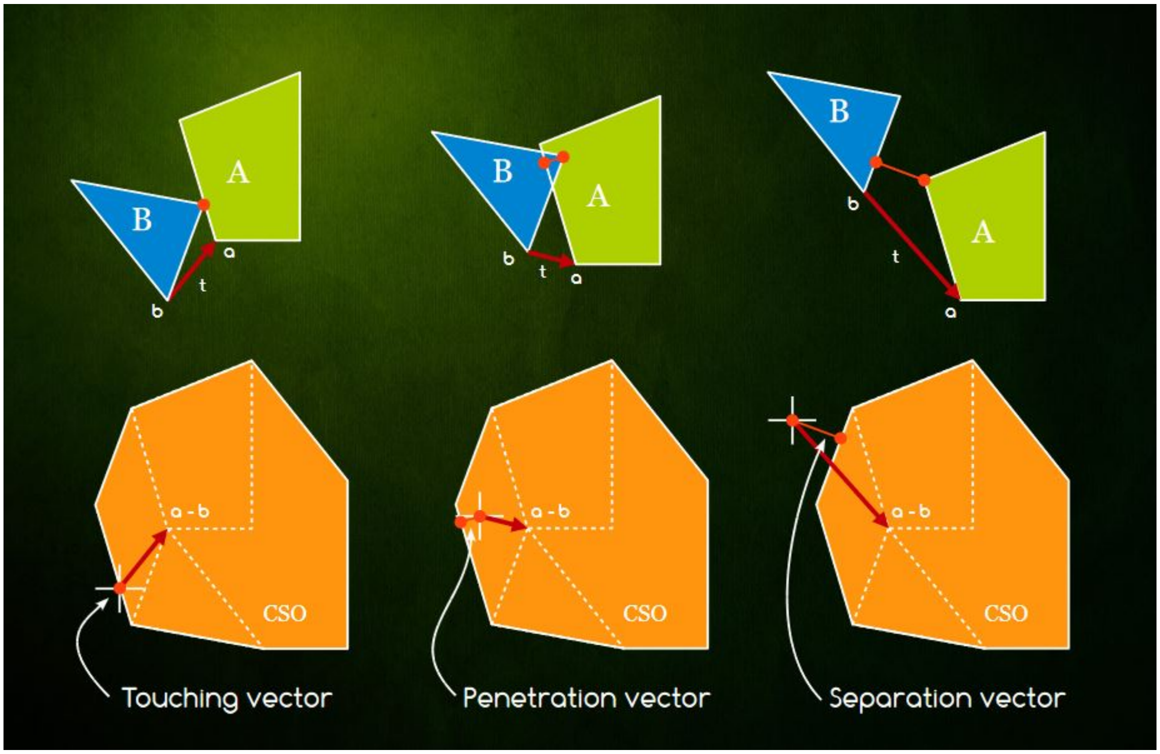
- Minkowskeho súčet: $A \oplus B = \{a + b \mid a \in A \wedge b \in B\}$
- Minkowskeho rozdiel: $A \ominus B = A \oplus (-B) = \{a - b \mid a \in A \wedge b \in B\}$
- Mikowskeho preklad: $A \oplus t = A \oplus \{t\} = \{a + t \mid a \in A\}$



(b) Blízkosť konvexných telies

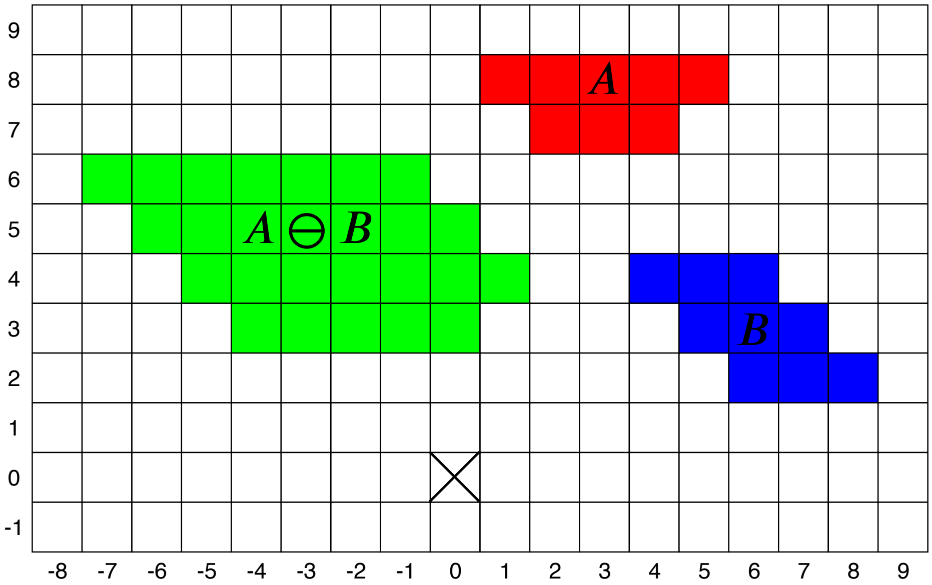
Pre skúmanie blízkosti konvexných objektov A a B sú dôležité tieto pojmy:

- **Kontakt dotyku**: Objekt A je v kontakte dotyku s objektom B, ak ich prienik (množina bodov) je podmnožinou roviny β. Teda $A \cap B \subset \beta$.
- **Vektor dotyku**: Vektor dotyku t_{AB} medzi objektami A a B je najkratší translačný vektor t posúvajúci objekty do kontaktu dotyku. Teda $t_{AB} \in t \mid A \cap (B \oplus t) \subset \beta \wedge t \in R^3 \wedge |t| = \delta_{AB}$.
- **Vzdialenosť dotyku**: Vzdialenosť dotyku δ_{AB} je dĺžka vektory dotyku t_{AB}. Teda $\delta_{AB} = min\{|t| \mid A \cap (B \oplus t) \in \beta \wedge t \in R^3\}$.



Objekty A, B sú v **tesnej blízkosti**, ak je vzdialenosť ich dotyku menšia ako stanovená hranica.
Ak sa objekty **nepretínajú**, ich vektor (vzdialenosť) dotyku sa nazýva deliaci vaktor (deliaca vzdialenosť).
Ak sa objekty **pretínajú**, ich vektor (vzdialenosť) dotyku sa nazýva penetračný vektor (hĺbka).
Deliaci vektor je jedinečný, zatiaľ čo penetračný vektor zväčša jedinečný nie je.

Teraz si ukážeme, ako môžeme použiť *minkowskeho rozdiel* (ktorého výsledok je známy aj ako **CSO**) na upresnenie definície **vektora** a **vzdialenosti dotyku**. Pre vizualizáciu je výsledok *minkowskeho rozdielu* zobrazený na nasledujúcom obrázku, ktorý je podrobnejší, aby sme v mriežke videli samotné body množín $A, B, A \ominus B$.

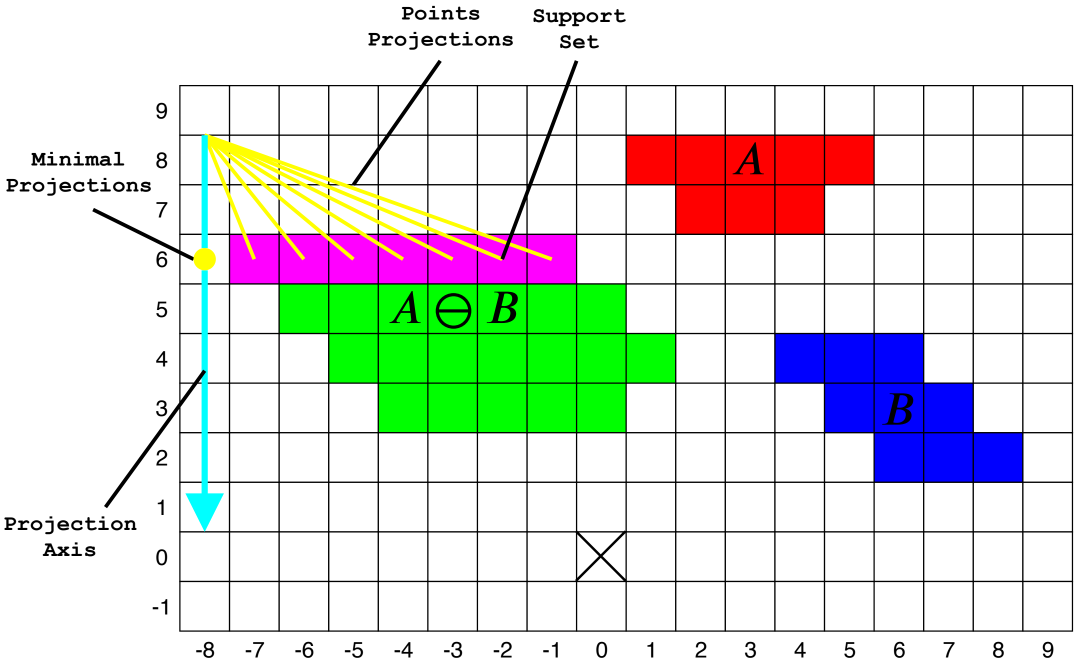


(c) **Podporná množina a hranica**

Množina bodov konvexného objektu C, ktorá má minimálnu veľkosť projekcie na nejakú smerovú os d je podpornou množinou C:

$$S_C^d = \{p \mid p \in C \wedge d^\top p = min\{d^\top c \mid c \in C\}\}$$

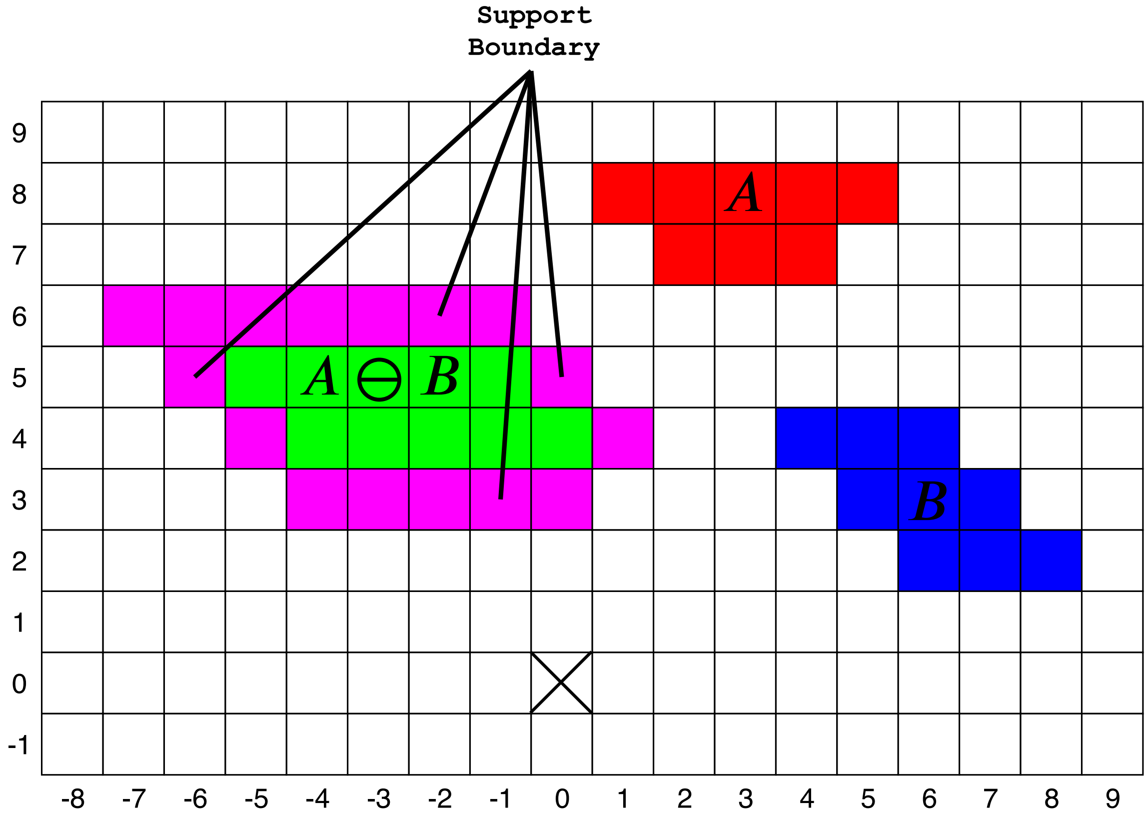
Na obrázku môžeme vidieť množinu podporných bodov objektu $A \ominus B$, ktorá je určená minimálnymi projekciami na zobrazenú smerovú os.



Množina všetkých podporných bodov konvexného objektu C vzhľadom na ľubovoľnú smerovú os d je hranicou objektu C:

$$\partial(C) = \{p \mid \in S_C^d \wedge d \in \mathbb{R}^3\}$$

Vizualizácia hranice objektu $A \ominus B$ je intuitívna:



(d) **Určenie vektoru dotyku**

Použitím **CSO** a podpornej hranice môžeme upresniť predchádzajúce definície. Ľubovoľný translačný vektor t presunie dva konvexné objekty A a B do dotykového kontaktu, vtedy a len vtedy, keď t leží na hranici ich **CSO**:

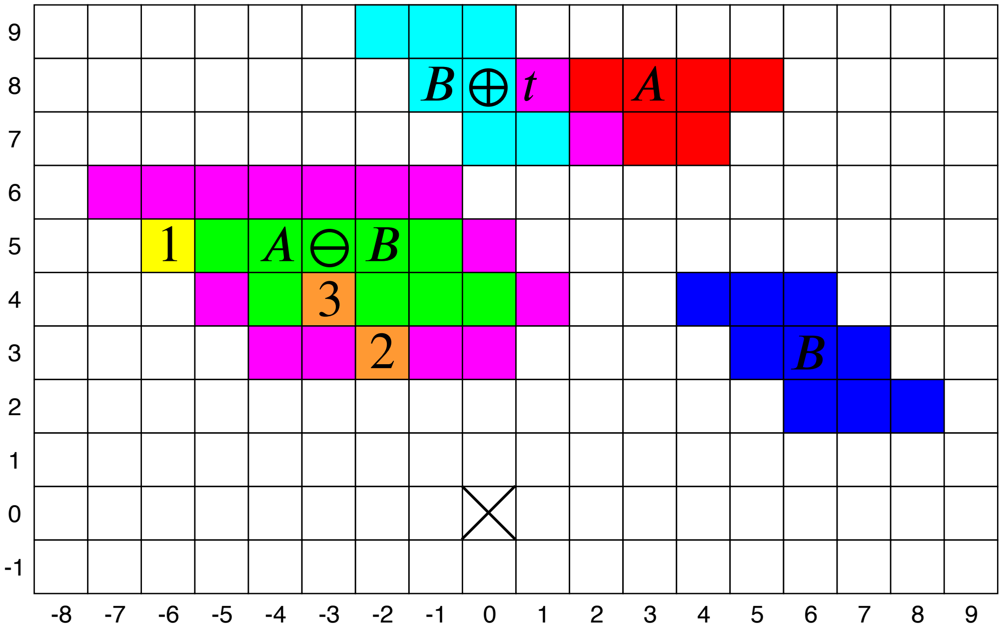
$$A \cap (B \oplus t) \subset \beta \Leftrightarrow t \in \partial(A \ominus B)$$

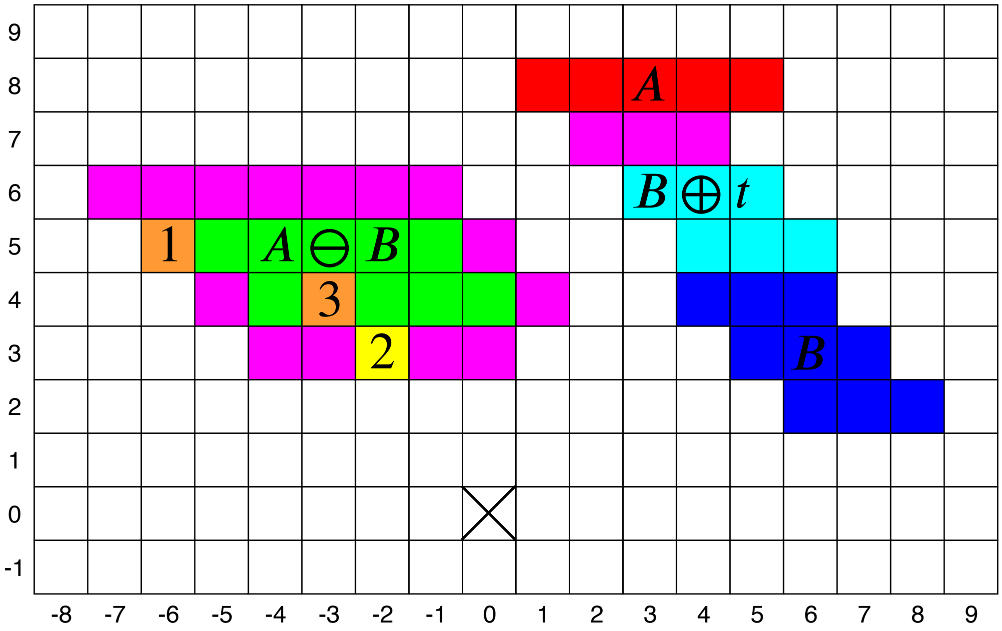
Z pôvodnej definície nie je jasné, ako nájsť translačný vektor t a kontaktnú rovinu β, **CSO** nám to umožňuje. Tvrdenie ďalej zjednodušuje definície vektora dotyku a vzdialenosti dotyku, nahradením $A \cap (B \oplus t) \subset \beta$ s $t \in \partial(A \ominus B)$:

$$\delta_{AB} = min\{|t| \mid t \in \partial(A \ominus B)\}$$

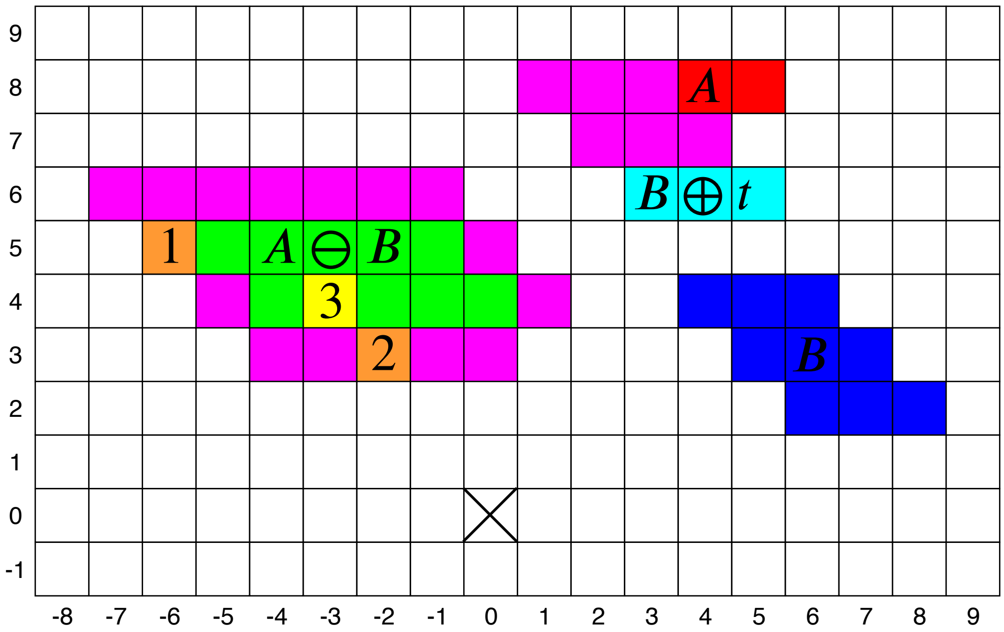
$$t_{AB} \in \{t \mid t \in \partial(A \ominus B) \wedge |t| = \delta_{AB}\}$$

Na nasledujúcich obrázkoch je niekoľko vizualizácií. Ak použijeme napríklad vybrané body $1 = (-6, 5); 2 = (-2, 3)$ z $\partial(A \ominus B)$ ako translačný vektor t na posun objektu B, spôsobí to, že sa A, B dostanú do **dotykového kontaktu**.





Vektory zvnútra CSO $\partial(A \ominus B)$, ako napríklad súradnice bodu $3 = (-3, 4)$, sú **penetračnými vektormi**, takže po aplikovaní posunu na objekt B sa objekty pretínajú.



Všetky ostatné translačné vektory mimo **CSO** $\partial(A \ominus B)$ sú **separačnými vektormi**, ktoré nespôsobia žiadnu kolíziu.

(e) **Dotyková rovina**

Ak sú objekty v dotykovom kontakte (δ_{AB} je nula), ich samotný prienik tvorí kontaktný región.

Ak sa objekty pretínajú alebo sú rozdelené, skonštruujeme kontaktný región nasledovne:

- Vypočítame dve podporné množiny $S_A^{+t_{AB}}$ a $S_A^{-t_{AB}}$ pre A, B, vzhľadom na t_{AB} získané napríklad pomocou **CSO**.
- Premietneme obe množiny na vektor dotyku t_{AB} a zoberieme medián.
- Kontaktná rovina bude mať počiatok v nájdenom mediáne a normálu t_{AB}
- Premietneme obe podporné množiny na kontaktnú rovinu. Prienik projekcií je kontaktným regiónom.