Orezávanie v 2D

Pri orezávaní v počítačovej grafike ide o to, rozhodnúť, či dané objekty ležia v zadanom okne a môžu sa celé vykresliť, alebo nie a prípadne určiť ich prienik s oknom. Pojem okna, ako obdĺžnika so stranami rovnobežnými so súradnicovými osami je používaný nielen v obrazovom ale aj modelovom priestore.

Nech je okno, do ktorého budeme orezávať objekty, dané súradnicami ľavého dolného vrcholu x_{min}, y_{min} a pravého horného vrcholu x_{max}, y_{max}.

Orezávanie bodov

Orezávanie bodov do daného okna je triviálne: bod so súradnicami $[x, y]$ leží v okne vtedy a len vtedy, keď sú súčasne splnené nerovnosti $x_{min} \leq x \leq x_{max}$ a $y_{min} \leq y \leq y_{max}$.

Orezávanie úsečiek

Pri orezávaní úsečiek môže nastať niekoľko jednoducho rozhodnuteľných situácií:

- oba koncové body úsečky ležia vnútri okna, úsečku je možné celú vykresliť
- oba koncové body ležia nad oknom, úsečka nemá s oknom prienik a vykresľovať sa nebude
- oba koncové body ležia pod oknom, úsečka nemá s oknom prienik a vykresľovať sa nebude
- oba koncové body ležia vľavo od okna, úsečka nemá s oknom prienik a vykresľovať sa nebude
- oba koncové body ležia vpravo od okna, úsečka nemá s oknom prienik a vykresľovať sa nebude

Ak nenastane ani jeden z týchto prípadov (ich overenie je možné jednoduchými nerovnosťami), je treba zistiť prienik úsečky s niektorou stranou okna (prípadne s priamkou na ktorej strana leží) a dve vzniknuté úsečky znovu preskúmať podľa podmienok uvedených vyššie.

Ako príklad určenia prieniku úsečky so stranou okna uveďme vzorec pre orezávanie zhora. Z podobnosti trojuholníkov vyplýva, že x-ová súradnica prieniku má hodnotu $x = x_1 + \frac{x_2 - x_1}{y_2 - y_1}(y_{max} - y_1)$, kde $[x_1, y_1], [x_2, y_2]$ sú koncové body úsečky.

Cohen-Sutherlandovo kódovanie

Pre scénu, ktorá sa skladá z veľkého množstva úsečiek môže byť výhodné nastaviť pre koncové body úsečiek kódy a podľa týchto kódov rýchlejšie rozhodovať či je alebo nie je nutné počítať prienik úsečky so stranami okna.

Pre každý koncový bod (so súradnicami [x,y]) úsečky sa nastaví štvorbitový kód.

Prvý bit sa nastaví rovný 1, ak bod leží *nad* oknom ($y > y_{max}$), alebo 0 ak neleží nad oknom.

Druhý bit sa nastaví rovný 1, ak bod leží *pod* oknom ($y < y_{min}$), alebo 0 ak neleží pod oknom.

Tretí bit sa nastaví rovný 1, ak bod leží *vpravo* od okna ($x > x_{max}$), alebo 0 ak neleží vpravo od okna.

Štvrtý bit sa nastaví rovný 1, ak bod leží *vľavo* od okna ($x < x_{min}$), alebo 0 ak neleží vľavo od okna.

O úsečke je možné povedať, že leží nad, alebo pod, alebo vľavo, alebo vpravo od okna aklogický súčin kódov jej koncových bodov je rovný 0000 a zároveň koncové body úsečky nemajú kódy 0000.

Algoritmus postupného delenia

Pretože numericky najnáročnejšou operáciou predošlého postupu je počítanie prieniku, má praktický význam takáto modifikácia algoritmu.

Úsečku, ktorá neleží celá ani mimo okna ani celá v okne, rozdeľ na dve polovice (to je numericky rýchle) a ďalej uvažuj tieto dve úsečky. Keďže pracujeme v diskrétnej rovine je tento algoritmus konečný a pomerne efektívny.

Úlohy

1. Zistite koľko najviac prienikov je nutné počítať pri orezávaní úsečky.
2. Rozhodnite či je možné, že pri rozdelení úsečky v algoritme postupného delenia je niekedy nutné ďalej deliť obe vzniknuté úsečky.
3. A čo v ďalšom kroku? Mohol by nastať prípad, keď by nebolo možné vylúčiť ani jednu zo štyroch vniknutých úsečiek?

Vylepšenie algoritmu pomocou funkcie d

Nevýhodu základného postupu pre orezávanie úsečky je, že ju orezávame aj vzhľadom na tie strany okna, s ktorými prienik nemá a má prienik iba s priamkou, na ktorej strana leží. Zavedieme teraz reálnu funkciu $d(A, B, C)$, ktorá nadobúda nulovú hodnotu ak bod A leží na priamke BC, kladné znamienko ak A leží v jednej polrovine určenej priamkou BC a záporné znamienko ak leží v druhej polrovine.

Nech $d(A, B, C) = x_a(y_b - y_c) + y_a(x_c - x_b) + x_b y_c - y_b x_c$. Prípadne môžeme použiť aj výpočet, v ktorom je menej násobení: $d(A, B, C) = (x_a - x_b)(y_b - y_c) + (y_a - y_b)(x_c - x_b)$. Všimnime si, že funkcia d dostaneme vlastne dosadením bodu A do všeobecnej rovnice priamky BC.

Pomocou tejto funkcie vieme zistiť, kedy majú dve úsečky prienik a to je možné využiť na zefiktívnenie algoritmu. Podrobnosti pozri v Ružický E., Úvod do počítačovej grafiky.

Odvodenie funkcie d pomocou vektorového súčinu

Teraz odvodíme funkciu d takým spôsobom aby sme dokázali platnosť nasledovného tvrdenia:

$d(A, P, Q) < 0$ ak A leží vľavo od orientovanej priamky PQ a $d(A, P, Q) > 0$ ak A leží vpravo od orientovanej priamky PQ.

Uvažujme vektory $\vec{u} = (x_q - x_p, y_q - y_p, 0), \vec{v} = (x_a - x_p, y_a - y_p, 0)$.

Vektorový súčin $\vec{u} \times \vec{v}$ má nenulovú iba tretiu súradnicu. Z vlastnosti vektorového súčinu, ktorá hovorí o tom, že $\vec{w} = \vec{u} \times \vec{v}$ má taký smer, aby $\vec{u}, \vec{v}, \vec{w}$ tvorili kladne orientovanú bázu vyplýva, práve to čo chceme dokázať. Tretia súradnica vektora $\vec{w}$ má totiž presne hodnotu funkcie d.

Orezávanie mnohouholníka

Pri orezávaní mnohouholníka sa využíva funkacia d a problém sa redukuje na zistenie prieniku daného mnohouholníka s polrovinou. Okno je totiž zjednotením štyroch polrovín. Problému určenia prieniku mnohouholníka a polroviny sa budeme venovať v ďalšej časti kurzu.

Použitá literatúra

1. Ružický, E. - Ferko, A. 1995. Počítačová grafika a spracovanie obrazu. 1995. ISBN 80-967180-2-9. Bratislava: SAPIENTIA 1995.

2. Žára, J. a kol. 1998. Moderní počítačová grafika. ISBN 80-7226-049-9. Computer Press 1998

3. Juraj Štugel, www.pg.miesto.sk