Rasterizácia úsečky a kružnice

V počítačovej grafike sa na zobrazovanie používajú rastrové zariadenia. Body obrazovky monitora aj body zobrazované tlačiarňou majú na rozdiel od bodov spojitej euklidovskej roviny iba celočíselné súradnice. Tieto body nazývame pixle (z anglického picture element). Pixle sú usporiadané tak, že každý vnútorný bod (štvorček, resp. obdĺžniček) má práve štyroch susedov, s ktorými má spoločnú stranu a ďalších štyroch, s ktorými ma spoločný iba vrchol. Takúto sieť pixlov nazývame raster a v tejto prednáške sa budeme venovať vykresľovaniu úsečiek a kružníc v rastri.

Základné a najviac používané krivky v počítačovej grafike sú úsečky, kružnice, kružnicové oblúky. Ďalšie používané krivky sú elipsy, polynomické prípadne racionálne krivky, krivky dané implicitne.

Kružnice a úsečky sú spravidla vykresľované tak často, že je nutné ich vykresľovať čo najrýchlejšie, preto uvedieme okrem základného algoritmu aj celočíselné algoritmy.

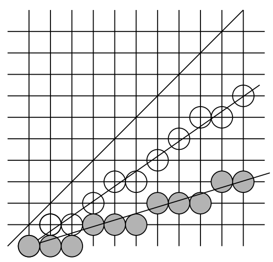
A ešte jedna poznámka o rastri. Štvorcovou sieťou, ktorú budeme v tejto prednáške používať v obrázkoch, nebudeme znázorňovať priamo pixle ale mrežové body s celočíselnými súradnicami, v ktorých ležia stredy pixlov.

Prírastkový algoritmus rasterizácie úsečky

Nech je daná úsečka AB, nenulovej dĺžky, ktorá nie je rovnobežná s osou y (úsečku rovnobežnú s osou y vykreslíme hravo aj v rastrovej grafike, preto sa ňou nebudeme ďalej zaoberať), a ktorej koncové body majú celočíselné súradnice: $A = [x_A, y_A], B = [x_B, y_B]$. Úsečka AB leží na priamke ktorej smernicová rovnica má tvar:

$$y = mx + b, \text{kde } m = \frac{y_B - y_A}{x_B - x_A}, b = \frac{x_B y_A - x_A y_B}{x_B - x_A} \tag{1}$$

Svoju pozornosť zúžime na úsečky, ktoré ležia na priamkach so smernicou $m \in \langle 0, 1\rangle$. (Algoritmus, ktorý ďalej uvedieme, je možné pre ostatné úsečky jednoducho modifikovať.)



Obrázok 1: Príklad dvoch úsečiek so smernicou z $\langle 0, 1\rangle$

Pre úsečky, ktorým sme sa rozhodli venovať platí, že pri vykresľovaní od bodu s menšou x -ovou súradnicou (nech je to bod A), treba ďalší bod úsečky vykresliť vždy v ďalšom stĺpci rastra. Treba sa v smere x posunúť o $\Delta_x = 1$. V smere y sa niekedy do susedného riadku rastra posunieme, niekedy nie (viď. Obrázok 1).

Skúsme zistiť, o koľko by sme mali posunúť v smere y, ak sa v smere x posunieme o 1. Inak povedané, aký je prírastok v smere y, ak prírastok v smere x je 1. Nech $[x, y]$ patrí úsečke, x, y vyhovujú rovnici (1). Nech $\acute{x} = x + 1$. Ak $[\acute{x}, \acute{y}]$ má tiež patriť úsečke, musí platiť:

$$\begin{aligned}
\acute{y} &= m\acute{x} + b \\
\acute{y} &= m(x + 1) + b \\
\acute{y} &= mx + b + m \\
\acute{y} &= y + m
\end{aligned}$$

Keďže $\Delta_y = m$ nie je celé číslo, musíme pred tým ako chceme vykresliť bod úsečky, jeho y -ovú súradnicu zaokrúhliť.

Zrejme teda najjednoduchší algoritmus pre vykreslenie úsečky bude vyzerať takto (funkcia $round$ zaokrúhli reálne číslo na celé):

1. vykresli bod $[x_A, y_A]$, $x := x_A$, $y := y_A$
2. $x := x + 1$, $y := y + m$, vykresli bod $[x, round(y)]$
3. ak $x < x_B$ tak opakuj krok 2

Nevýhodou tohoto algoritmu je to, že v jeho priebehu pracujeme s reálnymi číslami. Reálna aritmetika počítača je omnoho pomalšia ako celočíselná a preto by sme sa jej radi vyhli. Ak si uvedomíme, že rovnaký výsledok by sme dostali, ak by sme v kroku 2 pripočítavali iba celé čísla $0, 1$ (treba len rozhodnúť ktoré kedy), sme už len krôčik od vylepšenia algoritmu.

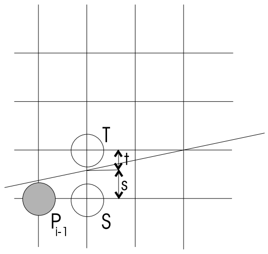
Bresenhaimov algoritmus na vykresľovanie úsečky

Rovnako ako v predošlej časti: Nech $A = [x_A, y_A], B = [x_B, y_B]$, s celočíselnými súradnicami, sú koncové body úsečky. Nech $x_A < x_B$, nech smernica priamky na ktorej úsečka AB leží je $m \in \langle 0, 1 \rangle$.

Predpokladajme, že sme v $i - 1$ kroku algoritmu vykreslili bod $P_{i-1} = [x_{i-1}, y_{i-1}]$. Rozhodujeme sa, či v i-tom kroku vykresliť bod $S = [x_{i-1}+1, y_{i-1}]$, alebo bod $T = [x_{i-1} + 1, y_{i-1} + 1]$ (viď. Obrázok 2).

Za P_i zvolíme bod S ak $s - t < 0$, a bod T ak $s - t \leq 0$. Pomocou rovnice (1) vyjadríme rozdiel $s - t$:

$$\begin{aligned}
s + y_{i-1} &= m(x_{i-1} + 1) + b \\
y_{i-1} + 1 - t &= m(x_{i-1} + 1) + b \\
s - t &= 2m(x_{i-1} + 1) - 2y_{i-1} + 2b - 1
\end{aligned}$$



Obrázok 2: Označenie použité v Bresenhaimovom algoritme pre úsečku

Na pravej strane rovnice je reálne číslo. Ak ju však vynásobíme číslom $d_x = x_B - x_A$, budeme mať na pravej strane celé číslo. Všimnime si, že $s - t$ má rovnaké znamienko ako $(s - t)d_x$. (Označme ešte $d_y = y_B - y_A$.)

Ďalej označíme

$$d(i) = 2x_{i-1}d_y - 2y_{i-1}d_x + 2d_y + 2bd_x - d_x \quad (2)$$

Podľa znamienka tohoto čísla zvolíme bod P_i. Aby sme ušetrili v ďalšom kroku numerické operácie, vyjadríme $d(i+1)$ pomocou $d(i)$.

$$\begin{aligned} d(i+1) &= 2x_i d_y - 2y_i d_x + 2d_y + 2bd_x - d_x \\ d(i+1) - d(i) &= 2d_y(x_i - x_{i-1}) - 2d_x(y_i - y_{i-1}) \end{aligned}$$

PRe úsečky, ktoré uvažujeme $x_i - x_{i-1} = 1$. Podľa toho, ktorý z bodov S, T vyberieme vieme aj hodnotu $y_i - y_{i-1}$.

$$\text{ak } d(i) < 0 \quad d(i+1) = d(i) + 2d_y \quad (3)$$

$$\text{ak } d(i) \geq 0 \quad d(i+1) = d(i) + 2d_y - 2d_x \quad (4)$$

Počiatočné $d(1)$ môžeme z (2) vyjadriť takto:

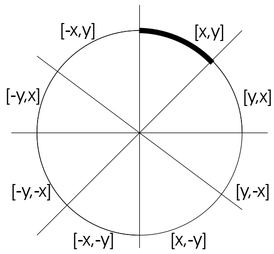
$$d(1) = 2x_A d_y - 2y_A d_x + 2d_y + 2bd_x - d_x \quad (5)$$

Bresenhaimov algoritmus pre nami uvažované úsečky vyzerá takto:

1. vykresli bod $[x_A, y_A]$, $x_0 := x_A$, $y_0 := y_A$, $i = 1$
2. z (5) vypočítaj $d(1)$
3.
   - ak $d(i) < 0$ tak $x_{i+1} := x_i + 1$, $y_{i+1} := y_i$, $d(i+1) := d(i) + 2d_y$
   - ak $d(i) \geq 0$ tak $x_{i+1} := x_i + 1$, $y_{i+1} := y_i + 1$, $d(i+1) := d(i) + 2d_y - 2d_x$
4. vykresli bod $[x_{i+1}, y_{i+1}]$
5. ak $x_{i+1} < x_B$ tak $i := i + 1$ a vráť sa na krok 3

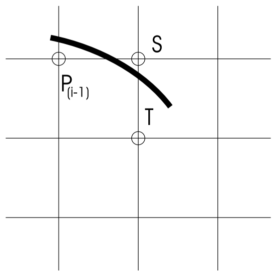
Bresenhaimov algoritmus na vykresľovanie kružnice

Uvedieme algoritmus pre vykreslenie jednej osminy (v druhom oktante) kružnice so stredom v bode $[0,0]$, polomerom r. Osminu preto, lebo ostatné časti je možné dostať zo symetrií (viď. Obrázok 3).



Obrázok 3: Symetrie kružnice

Predpokladajme, že sme v $i-1$ kroku algoritmu vykreslili bod $P_{i-1} = [x_{i-1}, y_{i-1}]$. Rozhodujeme sa, či v i-tom kroku vykresliť bod $S = [x_{i-1}+1, y_{i-1}]$, alebo bod $T = [x_{i-1}+1, y_{i-1}-1]$ (viď. Obrázok 4).



Obrázok 4: Označenie použité v Bresenhaimovom algoritme pre kružnicu

Pre bod $X = [x, y]$, označme $d(X) = x^2 + y^2 - r^2$. Táto hodnota je rovná nule, ak X leží na kružnici, je kladná ak leží zvonku kružnice a záporná ak leží vo vnútri kružnice.

Na rozhodnutie, ktorý bod vykresliť použijeme hodnotu $d(i) = d(S) + D(T)$. Ak $d(i) \geq 0$, vykreslíme bod T. Ak $d(i) < 0$, vykreslíme bod S.

Uvedieme ešte vyjadrenie $d(i+1)$ pomocou $d(i)$.

$$\begin{aligned}
d(i) &= (x_{i-1}+1)^2 + y_{i-1}^2 - r^2 + (x_{i-1}+1)^2 + (y_{i-1}-1)^2 - r^2 \\
d(i) &= 2x_{i-1}^2 + 4x_{i-1} + 2y_{i-1}^2 - 2y_{i-1} - 2r^2 + 3 \\
d(i+1) &= 2x_i^2 + 4x_i + 2y_i^2 - 2y_i - 2r^2 + 3 \\
d(i+1) - d(i) &= 2(x_i^2 - x_{i-1}^2) + 4(x_i - x_{i-1}) + 2(y_i^2 - y_{i-1}^2) - 2(y_i - y_{i-1})
\end{aligned}$$

$$\begin{array}{lll} \text{ak } d(i) < 0 & d(i+1) = & d(i) + 4x_{i-1} + 6 \\ \text{ak } d(i) \geq 0 & d(i+1) = & d(i) + 4x_{i-1} - 4y_{i-1} + 10 \end{array}$$

Úlohy

1. Odvoďte Δ_x, Δ_y z prírastkového algoritmu pre všetky typy úsečiek.
2. Vytvorte algoritmus pre kreslenie elipsy.

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