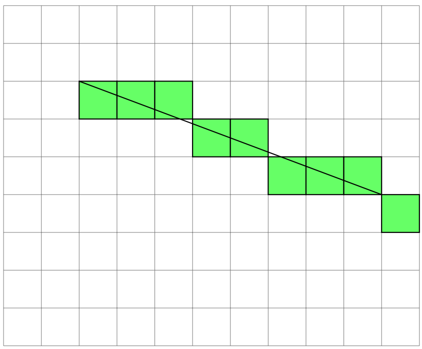
Počítačová grafika, 5. díl

EDUARD BARTL

Přírodovědecká fakulta UP, Olomouc

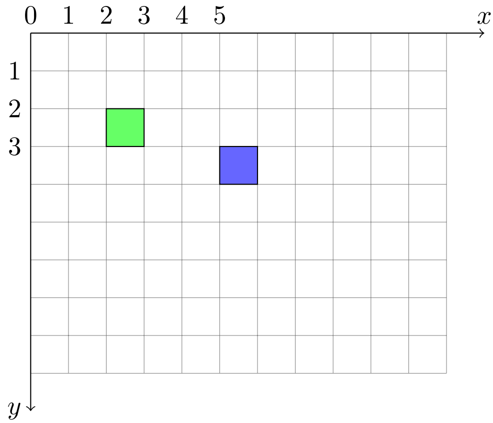
Článek navazuje na předchozí díly věnující se počítačové grafice. Stejně jako minulý díl se zabývá rasterizací úsečky, konkrétně je vysvětlen takzvaný Bresenhamův algoritmus. Článek může sloužit jako pomůcka pro středoškolské učitele informatiky a výpočetní techniky.

Na úvod připomeňme, že rasterizací objektů rozumíme kreslení těchto objektů do rastru zobrazovacího zařízení. K dispozici máme matematický popis daného geometrického objektu, v našem případě se jedná o úsečku. Tento objekt je tvořen nekonečným množstvím bezrozměrných geometrických bodů a naším úkolem je ho co nejvěrněji zobrazit v rastru, který je však tvořen pouze konečným množstvím fyzických pixelů (obr. 1). Rasterizační algoritmy by měly navíc pracovat co nejefektivněji, tedy co nejrychleji a s využitím co nejmenšího množství paměti.



Obr. 1 Rasterizace úsečky

Připomeňme dále, že počátek souřadnicového systému umísťujeme do levého horního rohu zobrazovacího zařízení a že kladná polovina osy x směřuje zleva doprava a kladná polovina osy y shora dolů. Předpokládáme, že pixely mají jednotkovou velikost. Logický pixel (viz [1]) umístíme do levého horního vrcholu pixelu – souřadnice daného fyzického pixelu jsou proto dány souřadnicemi jeho levého horního vrcholu, jak ukazuje obr. 2.



Obr. 2 Zelený pixel má souřadnice $\langle 2, 2\rangle$, modrý pixel má souřadnice $\langle 5, 3\rangle$.

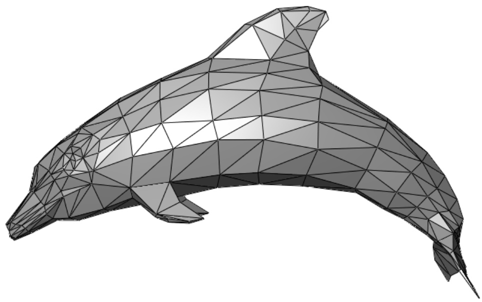
Algoritmus DDA, představený v předchozím dílu [2] seriálu, je velmi jednoduchý. Tato jednoduchost je jistě výhodná – algoritmus je snadno pochopitelný a je snadné ho naprogramovat. Má však jednu podstatnou nevýhodu, nepracuje totiž s celými čísly. Výpočty s celými čísly probíhají na počítači velmi rychle. Vysoká rychlost je důležitá v situacích, kdy se modelují složité scény (zejména trojrozměrné) ve vysokém rozlišení. Dokonce i z jednoduchého obr. 3 je patrné, že při jeho tvorbě bylo rasterizováno velké množství úseček. Využívání celočíselné aritmetiky je tedy jedním ze základních požadavků na kvalitní rasterizační algoritmy.

V tomto díle seriálu si představíme takzvaný *Bresenhamův algoritmus*, který celočíselnou aritmetiku používá. Autorem tohoto algoritmu je americký informatik Jack Elton Bresenham (narozen roku 1937).

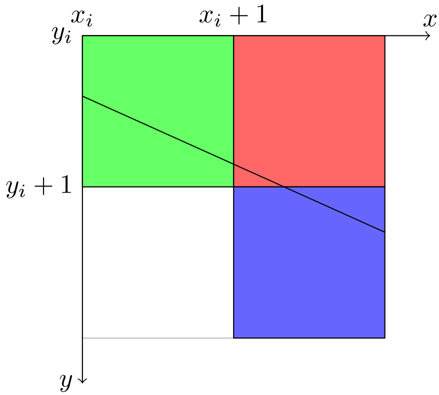
Stejně jako v případě algoritmu DDA budeme pracovat se směrnicovou rovnicí přímky:

$$y = kx + q,$$

kde k je takzvaná *směrnice* určující sklon přímky a q určuje průsečík přímky s osou y. Opět se omezíme na případ, kdy $k < 1$; to znamená, řídící osou bude osa x. Předpokládejme, že v i-tém kroku výpočtu obarvíme pixel o souřadnicích $\langle x_i, y_i\rangle$; tento pixel je znázorněn na obr. 4 zelenou barvou.



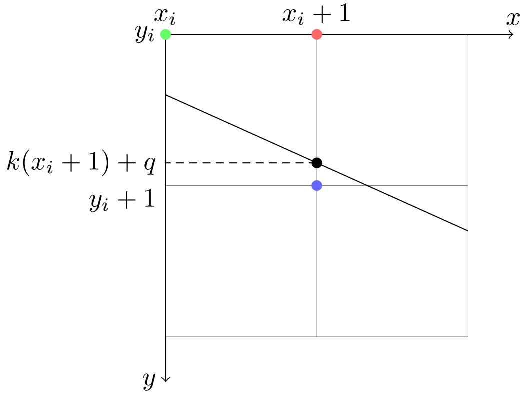
Obr. 3 Model delfína reprezentovaný pomocí trojúhelníkové sítě (zdroj Wikipedia)



Obr. 4 Výběr pixelů v Bresenhamově algoritmu (barevné čtverce znázorňují fyzické pixely)

Z obrázku je zřejmé, že v následujícím kroku obarvíme buď pixel, který leží bezprostředně napravo (označen je červenou barvou), nebo pixel, který leží napravo dole (označený modrou barvou). Základní myšlenka Bresenhamova algoritmu spočívá v tom, že v jednotlivých krocích ve skutečnosti nepotřebujeme znát přesné souřadnice bodů na přímce (jako tomu bylo u algoritmu DDA), které jsou obecně neceločíselné. Místo toho budeme neustále zvyšovat x-ovou souřadnici obarvovaných pixelů o jedničku a vhodným způsobem se budeme rozhodovat, jestli y-ovou souřadnici necháme nezměněnou (vybraným bude červený pixel), nebo ji zvýšíme o jedničku (vybereme modrý pixel). Ve zbytku této sekce se tedy budeme zabývat především tím, jak vybrat správný pixel.

Z obr. 4 je zřejmé, že červený pixel má souřadnice $\langle x_i + 1, y_i \rangle$, modrý pixel má souřadnice $\langle x_i + 1, y_i + 1 \rangle$. Příslušné logické pixely jsou ukázány na obr. 5 jako červený a modrý bod.



Obr. 5 Výběr pixelů v Bresenhamově algoritmu (barevné body znázorňují logické pixely)

Výběr pixelu v následujícím kroku provedeme podle toho, jestli je odpovídajícímu bodu na přímce (to znamená bodu na přímce, který má x-ovou souřadnici rovnou $x_i + 1$; na obr. 5 je tento bod zobrazen černou barvou) blíže červený nebo modrý bod. Vzdálenost červeného bodu od černého budeme značit jako d_1, vzdálenost modrého bodu od černého pak budeme značit jako d_2, viz opět obr. 5. Pro tyto vzdálenosti platí:

$$
\begin{aligned}
d_1 &= k(x_i + 1) + q - y_i, \\
d_2 &= y_i + 1 - k(x_i + 1) - q.
\end{aligned}
$$

Rozhodnutí, zdali je černému bodu blíže červený nebo modrý, učiníme na základě rozdílu těchto vzdáleností $\Delta d = d_1 - d_2$. Pro tento rozdíl můžeme psát

$$\Delta d = d_1 - d_2 = 2k(x_i + 1) - 2y_i + 2q - 1. \tag{1}$$

Pokud bude platit $\Delta d < 0$, pak bude černému bodu blíže červený. Jestliže však bude platit $\Delta d > 0$, bude černému bodu blíže modrý. K rozhodnutí tedy bude stačit znát pouze znaménko hodnoty Δd, velikost této hodnoty nás v konečném důsledku zajímat nebude, což představuje značné zjednodušení.

V dalším výkladu budeme pracovat se součinem $\Delta x \cdot \Delta d$, který se zpravidla nazývá *predikce* a budeme ho značit symbolem p_i:

$$p_i = \Delta x \cdot \Delta d.$$

Za hodnotu Δd můžeme dosadit výraz (1) a provést několik jednoduchých úprav:

$$\begin{aligned}
p_i &= \Delta x \cdot (2k(x_i + 1) - 2y_i + 2q - 1) \\
&= \Delta x \cdot 2k(x_i + 1) - \Delta x \cdot 2y_i + \Delta x \cdot 2q - \Delta x \\
&= \Delta x \cdot 2\frac{\Delta y}{\Delta x}(x_i + 1) - \Delta x \cdot 2y_i + \Delta x \cdot 2q - \Delta x \\
&= 2\Delta y x_i - 2\Delta x y_i + 2\Delta y + \Delta x(2q - 1).
\end{aligned}$$

Poznamenejme, že součet posledních dvou členů

$$2\Delta y + \Delta x(2q - 1)$$

je pro danou přímku neměnný; je stejný ve všech krocích výpočtu, protože závisí pouze na hodnotách Δy, Δx a q. Označíme ho písmenem c, můžeme tedy psát

$$p_i = 2\Delta y x_i - 2\Delta x y_i + c. \tag{2}$$

Predikce p_{i+1} v následujícím kroku odvodíme z predikce p_i jednoduše tak, že každý výskyt indexu i ve výrazu (2) nahradíme indexem $i + 1$:

$$p_{i+1} = 2\Delta y x_{i+1} - 2\Delta x y_{i+1} + c.$$

Dále vypočteme rozdíl predikcí $p_{i+1} - p_i$, konstantní člen c se samozřejmě odečte. Po několika úpravách tak získáme:

$$\begin{aligned}
p_{i+1} - p_i &= 2\Delta y x_{i+1} - 2\Delta x y_{i+1} - 2\Delta y x_i + 2\Delta x y_i \\
&= 2\Delta y(x_{i+1} - x_i) - 2\Delta x(y_{i+1} - y_i) \\
&= 2\Delta y - 2\Delta x(y_{i+1} - y_i).
\end{aligned}$$

Predikci p_{i+1} tak můžeme vypočítat z předchozí predikce p_i pomocí následujího vzorce:

$$p_{i+1} = p_i + 2\Delta y - 2\Delta x(y_{i+1} - y_i). \tag{3}$$

Vraťme se nyní k problému rozhodnutí, který pixel vybereme. Ze vztahů, které jsme výše odvodili, vidíme, že existují dvě možnosti:

1. Jestliže pro predikci v daném kroku platí

$$p_i \leq 0,$$

pak musí být i $\Delta d \leq 0$.[1] To znamená, že černému bodu je blíže bod červený,[2] v dalším kroku tedy y-ovou souřadnici nezvýšíme. Platí tedy

$$y_{i+1} = y_i.$$

Tuto skutečnost pak zaneseme do vzorce (3) pro výpočet predikce p_{i+1}, který se tak výrazně zjednoduší:

$$\begin{aligned} p_{i+1} &= p_i + 2\Delta y - 2\Delta x(y_i - y_i) \\ &= p_i + 2\Delta y. \end{aligned}$$

2. Jestliže je naopak

$$p_i > 0,$$

pak je také $\Delta d > 0$. To znamená, černému bodu je blíže modrý a musíme tedy zvýšit y-ovou souřadnici:

$$y_{i+1} = y_i + 1.$$

Dosazením této rovnosti do vztahu (3) pak získáme vzorec pro výpočet predikce p_{i+1}:

$$p_{i+1} = p_i + 2\Delta y - 2\Delta x.$$

Vidíme tedy, že rozhodnutí, který pixel obarvíme, činíme pouze na základě znaménka predikce p_i. Úpravu této predikce provádíme tak, že k ní přičteme hodnotu $2\Delta y$ a v případě, že $p_i > 0$, od takto upravené hodnoty odečteme $2\Delta x$. Algoritmus 2 tedy pracuje pouze s celými čísly a je proto výrazně rychlejší než algoritmus DDA. Poznamenejme, že počáteční hodnota predikce se nastaví na $2\Delta y - \Delta x$; není obtížné odvodit, proč je to právě tato hodnota.

[1] Připomeňme, že predikce p_i je definována jako součin $\Delta x \Delta d$; navíc $\Delta x = x_B - x_A$ je v našem nastavení vždy kladné číslo.

[2] Přesněji řečeno, černému bodu je blíže červený *nebo* jsou oba body, jak červený tak modrý, od černého stejně vzdálené.

Algoritmus 1 Bresenhamův algoritmus

Vstup: Body $A = \langle x_A, y_A \rangle$ a $B = \langle x_B, y_B \rangle$; všechny souřadnice jsou celočíselné.
Výstup: Souřadnice pixelů, které rasterizují úsečku danou koncovými body A a B.

$c_1 = 2\Delta y$
$c_2 = -2\Delta x$
$p = 2\Delta y - \Delta x$
$\langle x, y \rangle = \langle x_A, y_A \rangle$
while $x \leq x_B$ **do**
  obarvi pixel o souřadnicích $\langle x, y \rangle$
  $x = x + 1$
  $p = p + c_1$
  if $p > 0$ **then**
    $y = y + 1$
    $p = p + c_2$
  end if
end while

Další díl seriálu budeme naposledy věnovat rasterizačním algoritmům. Ukážeme si, jakým způsobem efektivně rasterizovat složitější geometrické objekty než je úsečka. Konkrétně se zaměříme na vykreslování kružnice a elipsy.

Literatura

[1] *Bartl, E.*: Počítačová grafika I. Matematika-fyzika–informatika, roč. 29 (2020), č. 2, s. 138–148.

[2] *Bartl, E.*: Počítačová grafika IV. Matematika-fyzika–informatika, roč. 33 (2020), č. 2, s. 144–150.

[3] *Felkel, P., Sochor, J., Žára, J., Beneš, B.*: Moderní počítačová grafika. 2. vydání. Computer Press, 2005.

[4] *Gonzalez, R. C., Woods, R. E.*: Digital Image Processing. 4. vydání. Pearson Prentice Hall, 2018.

[5] *Huges, J. F. a kol.*: Computer Graphics. Principles and Practice. 3. vydání. Addison-Wesley, 2014.

[6] *Martíšek, D.*: Matematické principy grafických systémů. Littera, 2002.