Základy počítačové grafiky

Vyplňování 2D oblastí

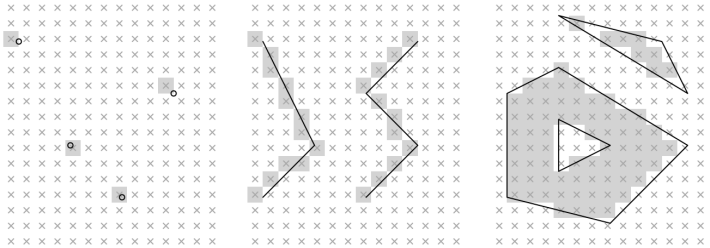
Michal Španěl
Tomáš Milet

Ústav počítačové grafiky a multimédií

Brno 2023

Cíl přednášky

Seznámit se s algoritmy pro vyplňování uzavřených rovinných oblastí s vektorově definovanou hranicí (trojúhelník, polygon).



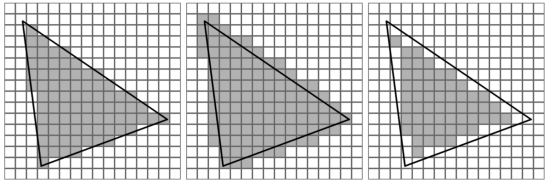
Obsah

Vyplňování oblastí

Definice

Proces nalezení a označení (obarvení) všech vnitřních bodů dané oblasti (= rasterizace).

- Vstupem je popis hranice oblasti.
- Výstupem je rastrový popis vyplněné oblasti.
- Jsou různé typy vyplňování...

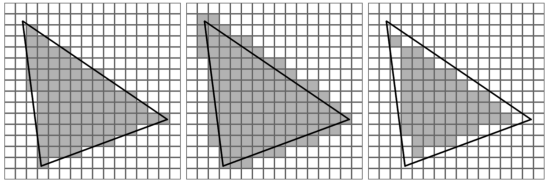


Vyplňování oblastí

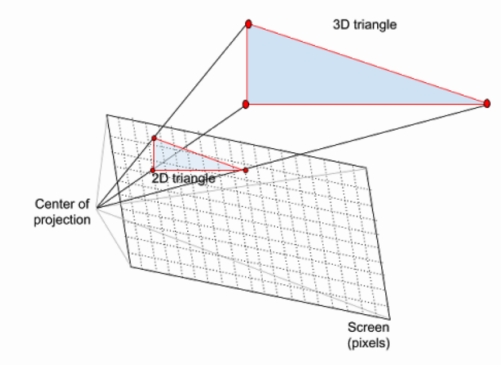
Definice

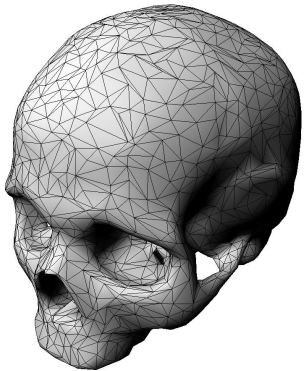
Proces nalezení a označení (obarvení) všech vnitřních bodů dané oblasti (= rasterizace).

- Vstupem je popis hranice oblasti.
- Výstupem je rastrový popis vyplněné oblasti.
- Jsou různé typy vyplňování...



Vyplňování oblastí najdeme i ve 3D grafice





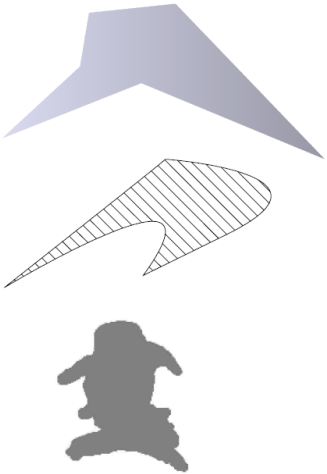
Popis a typy oblastí

Vektorové oblasti

Hranice popsána seznamem vektorových entit (úseček, kruhových oblouků, křivek, atd.).

Rastrové oblasti

Hranice popsána v rastrové matici hodnotou pixelů na hranici nebo barvou vyplňované oblasti.



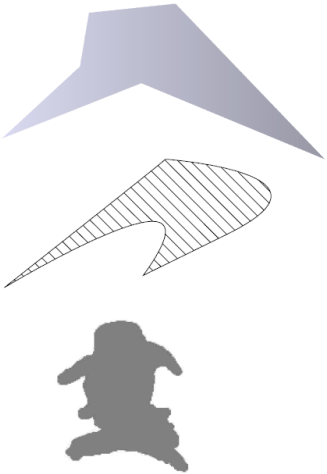
Popis a typy oblastí

Vektorové oblasti

Hranice popsána seznamem vektorových entit (úseček, kruhových oblouků, křivek, atd.).

Rastrové oblasti

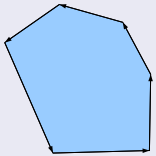
Hranice popsána v rastrové matici hodnotou pixelů na hranici nebo barvou vyplňované oblasti.



Dělení oblastí podle geometrie

Konvexní (vypouklé, vyduté)

Pro libovolné dva body oblasti platí, že jejich spojnice je součástí oblasti, neprotíná její hranice.



?

Jak určíme, kde je uvnitř a kde vně oblasti?

Konkávní (nekonvexní, prohnuté, duté)

Oblast, která není konvexní.

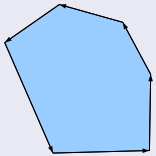
S vnitřními otvory

Hranice je tvořena více nezávislými smyčkami, obrysem a otvory.

Dělení oblastí podle geometrie

Konvexní (vypouklé, vyduté)

Pro libovolné dva body oblasti platí, že jejich spojnice je součástí oblasti, neprotíná její hranice.



Konkávní (nekonvexní, prohnuté, duté)

Oblast, která není konvexní.

S vnitřními otvory

Hranice je tvořena více nezávislými smyčkami, obrysem a otvory.

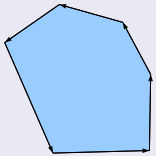
?

Jak určíme, kde je uvnitř a kde vně oblasti?

Dělení oblastí podle geometrie

Konvexní (vypouklé, vyduté)

Pro libovolné dva body oblasti platí, že jejich spojnice je součástí oblasti, neprotíná její hranice.



?

Jak určíme, kde je uvnitř a kde vně oblasti?

Konkávní (nekonvexní, prohnuté, duté)

Oblast, která není konvexní.

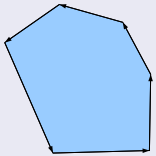
S vnitřními otvory

Hranice je tvořena více nezávislými smyčkami, obrysem a otvory.

Dělení oblastí podle geometrie

Konvexní (vypouklé, vyduté)

Pro libovolné dva body oblasti platí, že jejich spojnice je součástí oblasti, neprotíná její hranice.



?

Jak určíme, kde je uvnitř a kde vně oblasti?

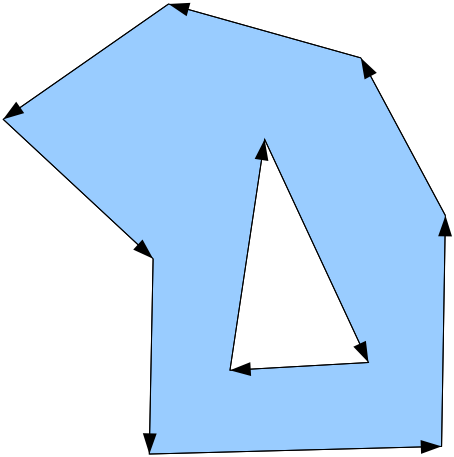
Konkávní (nekonvexní, prohnuté, duté)

Oblast, která není konvexní.

S vnitřními otvory

Hranice je tvořena více nezávislými smyčkami, obrysem a otvory.

Vektorový popis oblastí



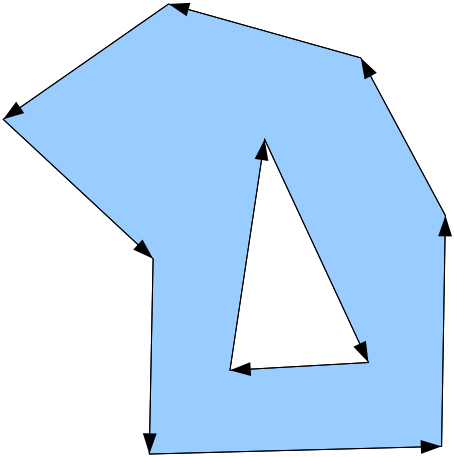
Korektní definice oblasti

- Seznam hraničních entit vyplňované oblasti musí být *orientovaný* a *spojitý*.
- Vnitřní díry mají vždy opačnou orientaci.

Jak otestovat orientaci?

- Vektorový součin dvou sousedních hran (pravidlo pravé ruky).
- V případě nekonvexních polygonů - suma přes celý polygon.

Vektorový popis oblastí



Korektní definice oblasti

- Seznam hraničních entit vyplňované oblasti musí být *orientovaný* a *spojitý*.
- Vnitřní díry mají vždy opačnou orientaci.

Jak otestovat orientaci?

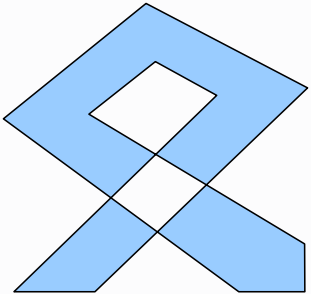
- Vektorový součin dvou sousedních hran (pravidlo pravé ruky).
- V případě nekonvexních polygonů - suma přes celý polygon.

Pravidla vyplňování složitých oblastí

- Řeší nejednoznačnost při vyplňování složitých oblastí (hranice protínají sami sebe).

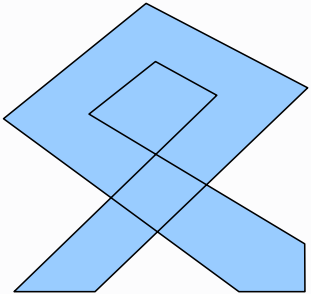
Paritní vyplňování

Hranice odděluje vyplněný a nevyplněný prostor, nejběžnější způsob.



Vnitřní vyplňování

Jsou vyplněny všechny body, které nejsou vně oblasti.

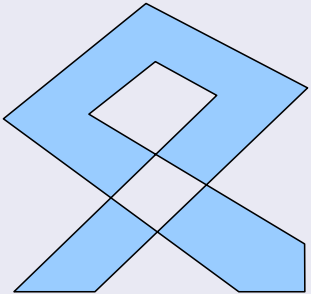


Pravidla vyplňování složitých oblastí

- Řeší nejednoznačnost při vyplňování složitých oblastí (hranice protínají sami sebe).

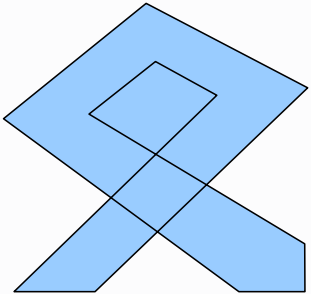
Paritní vyplňování

Hranice odděluje vyplněný a nevyplněný prostor, nejběžnější způsob.



Vnitřní vyplňování

Jsou vyplněny všechny body, které nejsou vně oblasti.

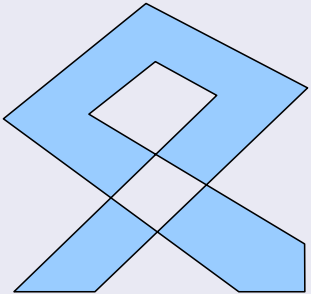


Pravidla vyplňování složitých oblastí

- Řeší nejednoznačnost při vyplňování složitých oblastí (hranice protínají sami sebe).

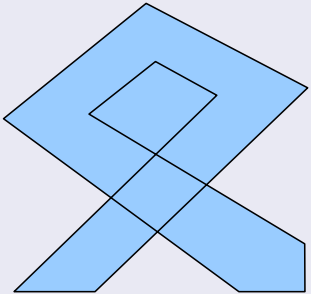
Paritní vyplňování

Hranice odděluje vyplněný a nevyplněný prostor, nejběžnější způsob.



Vnitřní vyplňování

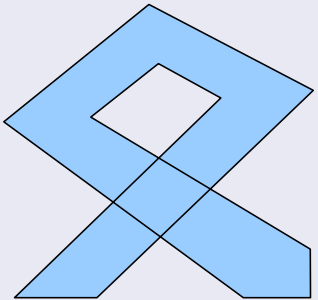
Jsou vyplněny všechny body, které nejsou vně oblasti.



Pravidla vyplňování složitých oblastí, pokr.

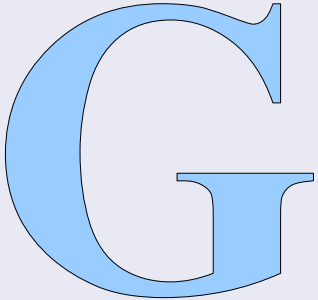
Nenulové vyplňování

Nenulové objetí bodu uzavřenou smyčkou při vyřešení sebeprotínání.

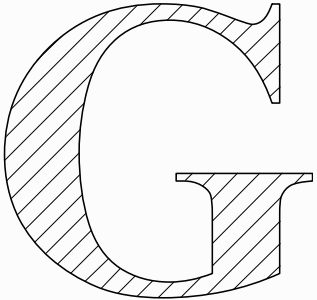


Druhy výplní

Konstantní barvou (*solid*)

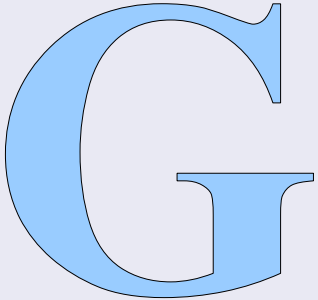


Jednoduchým čárovým vzorem (šrafou, angl. *hatch*)

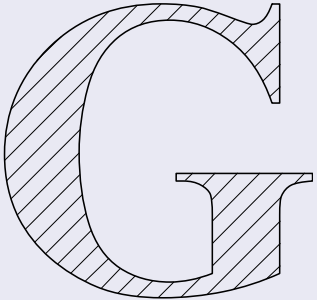


Druhy výplní

Konstantní barvou (*solid*)

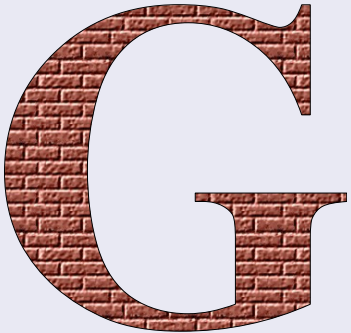


Jednoduchým čárovým vzorem (šrafou, angl. *hatch*)



Druhy výplní, pokr.

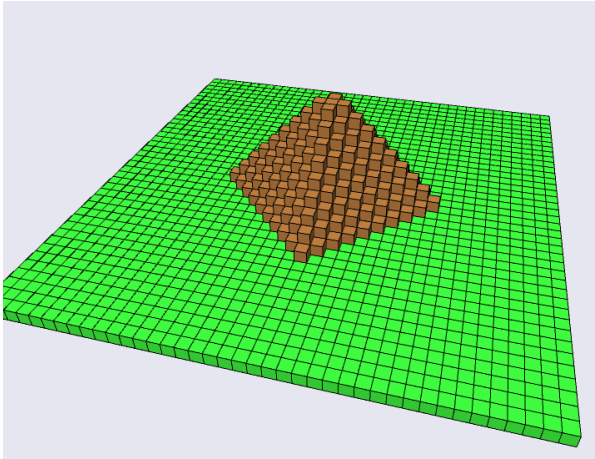
Složitějším obecným vzorem nebo texturou (*pattern*)



Obsah

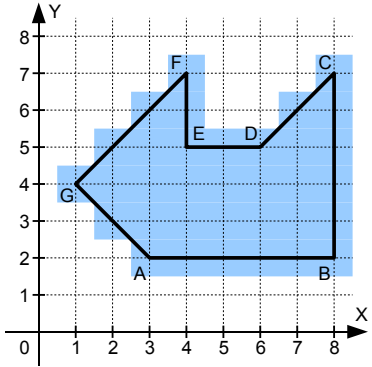
Stavíme pyramidu

- Rasterizace troj. v rovině XY
- Vrstvy zmenšujících se troj.



Řádkové vyplňování (angl. Scanline Fill)

- Základní algoritmus vyplňování obecných mnohoúhelníků.
- Seznam hraničních entit (hran) oblasti Ω.
- Orientovaný seznam vrcholů úseček.

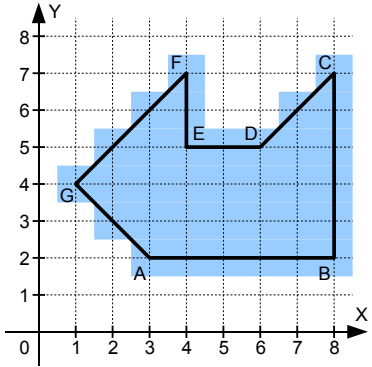


Základní myšlenka

- Procházení oblasti Ω po jednotlivých řádcích výsledného rastru.
- Úseky ležící uvnitř oblasti jsou vyplněny barvou nebo vzorem.

Řádkové vyplňování (angl. Scanline Fill)

- Základní algoritmus vyplňování obecných mnohoúhelníků.
- Seznam hraničních entit (hran) oblasti Ω.
- Orientovaný seznam vrcholů úseček.



Základní myšlenka

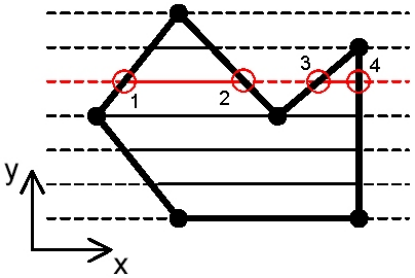
- Procházení oblasti Ω po jednotlivých řádcích výsledného rastru.
- Úseky ležící uvnitř oblasti jsou vyplněny barvou nebo vzorem.

Řádkové vyplňování, pokr.

Algoritmus (pro paritní vyplňování)

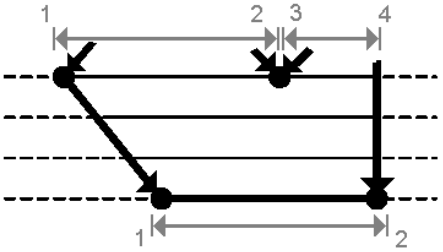
Pro každý řádek $Y_j \in \langle Y_{min}, Y_{max} \rangle$ oblasti Ω:

- Seznam souřadnic x_i průsečíků řádku Y_j se všemi hraničními úsečkami. *Vodorovné hrany se vynechávají*!
- Setřídění seznamu průsečíků podle souřadnic x_i.
- Vykreslení vodorovných úseků řádku Y_j mezi lichými a sudými průsečíky seznamu (dvojice tvoří úsek uvnitř oblasti).



Problém krajních bodů úseček (lichý počet průsečíků hran)

- Algoritmus předpokládá sudý počet průsečíků.
- Nemusí platit, pokud řádek protíná některý vrchol hranice!

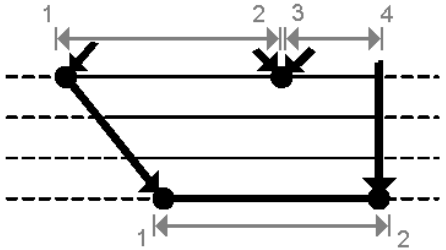


Obecné řešení

- Ve vrcholech, které jsou lokálním extrémem v ose Y generovat průsečíky obou hran.
- V ostatních vrcholech generovat průsečík pouze jednou.

Problém krajních bodů úseček (lichý počet průsečíků hran)

- Algoritmus předpokládá sudý počet průsečíků.
- Nemusí platit, pokud řádek protíná některý vrchol hranice!



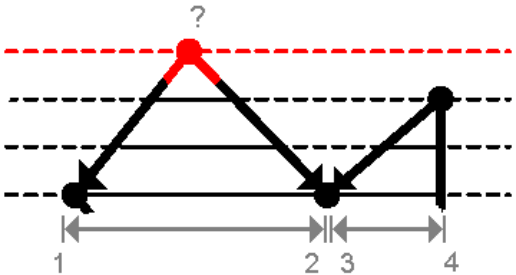
Obecné řešení

- Ve vrcholech, které jsou lokálním extrémem v ose Y generovat průsečíky obou hran.
- V ostatních vrcholech generovat průsečík pouze jednou.

Problém krajních bodů úseček, pokr.

Analýza typu vrcholů

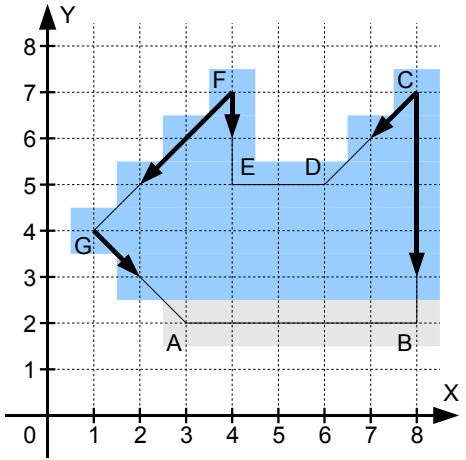
- Generování správného počtu průsečíků testováním extrému v ose Y.
- Implementace algoritmu se komplikuje a zpomaluje!



Problém krajních bodů úseček, pokr.

Zkrácení dolního okraje všech hran

- Zkrátit hranu ve směru osy Y o 1.
- Vodorovné hrany se vypouštějí.
- *Je nutné překreslit obrys oblasti!*



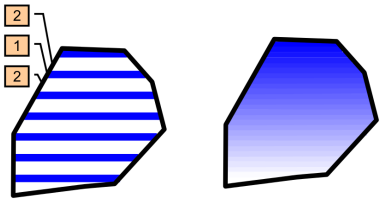
Šrafování, gradient

Vyplnění vodorovnou šrafou

- Jednoduchá úprava vyplňovacích algoritmů.
- Přeskakování řádků pomocí parametru.

Vyplnění gradientem

- Podobné šrafování.
- Parametr inkrementující se každý řádek.



Šrafování, gradient pod libovolným úhlem

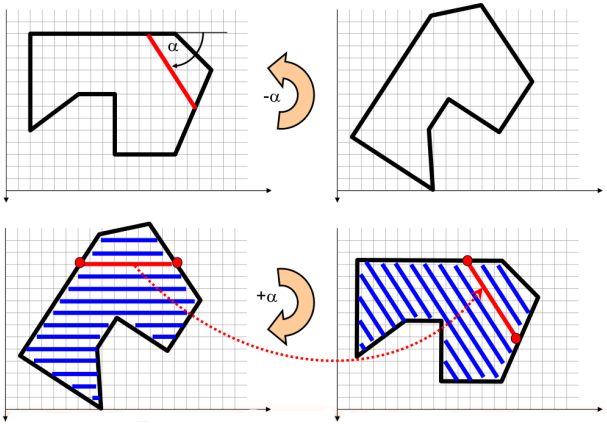
Algorimus pro šrafování pod libovolným úhlem

Modifikace vodorovného šrafování

- Pro šrafu/gradient pod úhlem α nutné nejprve otočit oblast o úhel $-\alpha$
- Šrafovat vodorovně
- Otočit oblast zpět o úhel α

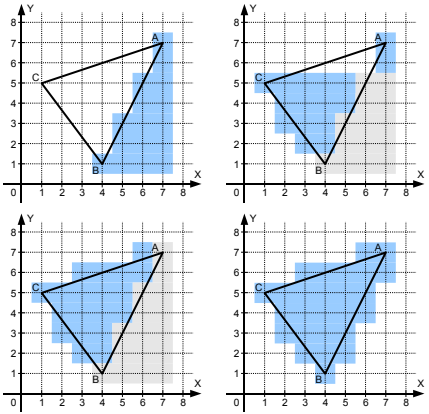
Otočení se řeší transformační maticí

Šrafování, gradient pod libovolným úhlem



Inverzní řádkové vyplňování

- Odstraňuje nutnost třídit průsečíky pro každý řádek.
- Mnohoúhelníky, seznam hran oblasti nebo orientovaný seznam vrcholů.

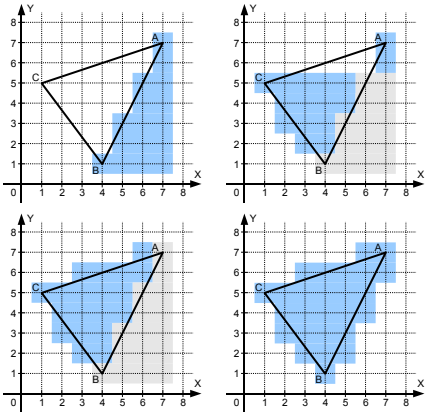


Základní myšlenka

- Procházení oblasti Ω po jednotlivých hranách $e_i \in \Omega$.
- Od každé hrany je napravo po řádcích provedeno vyplnění s inverzí hodnot pixelů.

Inverzní řádkové vyplňování

- Odstraňuje nutnost třídit průsečíky pro každý řádek.
- Mnohoúhelníky, seznam hran oblasti nebo orientovaný seznam vrcholů.



Základní myšlenka

- Procházení oblasti Ω po jednotlivých hranách $e_i \in \Omega$.
- Od každé hrany je napravo po řádcích provedeno vyplnění s inverzí hodnot pixelů.

Inverzní řádkové vyplňování, pokr.

Algoritmus (pro paritní vyplňování)

Nalezení maximální souřadnice X_{max} v ose X.

Každou hranu $e_i \in \Omega$ oblasti zpracuj následovně:

- Získání Y_{min} a Y_{max} pro danou hranu e_i.
- Pro každý řádek $Y_j \in \langle Y_{min}, Y_{max} \rangle$, kromě vodorovných hran, proveď:
  - Najdi průsečík $P_{ij} = (x_{ij}, y_{ij})$ řádku Y_j s aktuální hranou e_i.
  - Invertuj hodnoty pixelů v řádku Y_j od průsečíku P_{ij} po maximální souřadnici oblasti X_{max}.

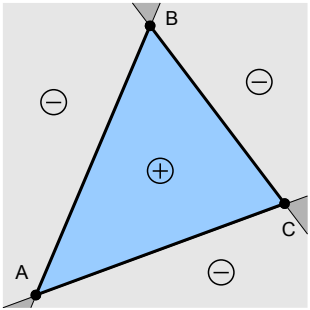
Inverzní řádkové vyplňování, pokr.

+/-

- Lineární časová složitost v závislosti na počtu hran.
- Po vyplnění je nutné překreslit obrys oblasti.
- Binární operace inverze pixelů $\rightarrow$ binární obraz.
- Ovlivňuje okolí oblasti $\rightarrow$ generování šablony v pomocném bufferu.

Pinedův algoritmus (J. Pineda, 1988)

- Pracuje pouze s *konvexními mnohoúhelníky*.
- Vyplňovaná oblast Ω je popsána seznamem hran $\Omega = \{e_1, e_2, \ldots, e_n\}$.
- Nejčastěji používán pro trojúhelníky (vždy konvexní).
- Snadná realizace v HW.

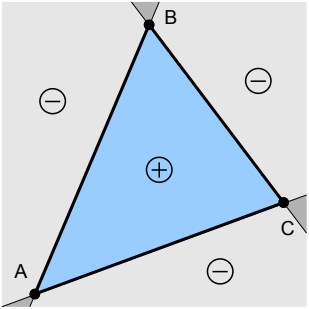


Základní myšlenka

- Rozdělení roviny oblasti Ω na poloroviny hran $e_i \in \Omega$.
- Body roviny, které leží na kladné straně všech polorovin hran e_i jsou uvnitř oblasti Ω.

Pinedův algoritmus (J. Pineda, 1988)

- Pracuje pouze s *konvexními mnohoúhelníky*.
- Vyplňovaná oblast Ω je popsána seznamem hran $\Omega = \{e_1, e_2, \ldots, e_n\}$.
- Nejčastěji používán pro trojúhelníky (vždy konvexní).
- Snadná realizace v HW.



Základní myšlenka

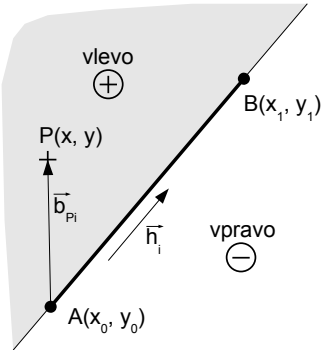
- Rozdělení roviny oblasti Ω na poloroviny hran $e_i \in \Omega$.
- Body roviny, které leží na kladné straně všech polorovin hran e_i jsou uvnitř oblasti Ω.

Test polohy bodu $P(x, y)$ k přímce

Hranová funkce $E_i(x, y)$

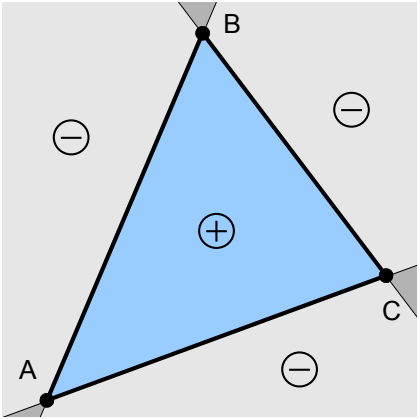
Vekt. součin vektoru $\vec{h_i}$ hrany a vektoru $\vec{b_{Pi}}$ z počátku hrany k testovanému bodu P.

$$\begin{aligned} \vec{h_i} &= (x_{i1} - x_{i0}, y_{i1} - y_{i0}) = (\Delta x_i, \Delta y_i) \\ \vec{b_{Pi}} &= (x - x_{i0}, y - y_{i0}) \\ E_i(x, y) &= \vec{h_i} \times \vec{b_{Pi}} \\ E_i(x, y) &= (x - x_{i0})\Delta y_i - (y - y_{i0})\Delta x_i \end{aligned}$$



Trojúhelníková oblast a její poloroviny

- Je-li hodnota všech hranových funkcí $E_i(x, y) \geq 0$, pak bod $P(x, y)$ leží uvnitř nebo na hranici oblasti.



Výpočet hranové funkce

- Není nutné vyhodnocovat hranové fce $E_i(x, y)$ pro každý bod!
- Hodnotu lze určit na základě sousedního bodu $P(x \pm 1, y)$ nebo $P(x, y \pm 1)$.

Odvození

$$
\begin{aligned}
E_i(x, y) &= (x - x_{i0})\Delta y_i - (y - y_{i0})\Delta x_i \\
E_i(x + 1, y) &= (x + 1 - x_{i0})\Delta y_i - (y - y_{i0})\Delta x_i \\
E_i(x + 1, y) &= E_i(x, y) + \Delta y_i
\end{aligned}
$$

$$
\begin{aligned}
E_i(x \pm 1, y) &= E_i(x, y) \pm \Delta y_i \\
E_i(x, y \pm 1) &= E_i(x, y) \pm \Delta x_i
\end{aligned}
$$

Výpočet hranové funkce

- Není nutné vyhodnocovat hranové fce $E_i(x, y)$ pro každý bod!
- Hodnotu lze určit na základě sousedního bodu $P(x \pm 1, y)$ nebo $P(x, y \pm 1)$.

Odvození

$$\begin{aligned} E_i(x, y) &= (x - x_{i0})\Delta y_i - (y - y_{i0})\Delta x_i \\ E_i(x + 1, y) &= (x + 1 - x_{i0})\Delta y_i - (y - y_{i0})\Delta x_i \\ E_i(x + 1, y) &= E_i(x, y) + \Delta y_i \end{aligned}$$

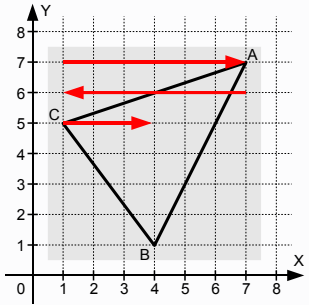
$$\begin{aligned} E_i(x \pm 1, y) &= E_i(x, y) \pm \Delta y_i \\ E_i(x, y \pm 1) &= E_i(x, y) \pm \Delta x_i \end{aligned}$$

Základní algoritmus

- Nalezení x_{min}, x_{max}, y_{min} a y_{max}.
- Pro každou hranu oblasti inicializuj $E_i(x_{min}, y_{min})$.
- Cyklus přes všechny body (x, y) v obdélníku (x_{min}, y_{min}), (x_{max}, y_{max}):
  - Je-li $E_i(x, y) \geq 0$ pro všechny hrany, pak nastav hodnotu pixelu.
  - Aktualizace $E_i(x, y)$.

Procházení opsaného obdélníka po řádcích

Zbytečný průchod $\sim 1/2$ obdélníka.

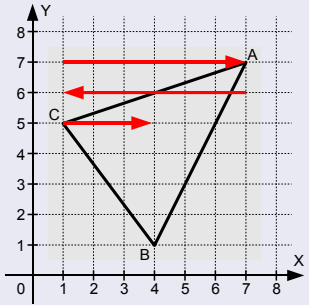


Základní algoritmus

- Nalezení x_{min}, x_{max}, y_{min} a y_{max}.
- Pro každou hranu oblasti inicializuj $E_i(x_{min}, y_{min})$.
- Cyklus přes všechny body (x, y) v obdélníku (x_{min}, y_{min}), (x_{max}, y_{max}):
  - Je-li $E_i(x, y) \geq 0$ pro všechny hrany, pak nastav hodnotu pixelu.
  - Aktualizace $E_i(x, y)$.

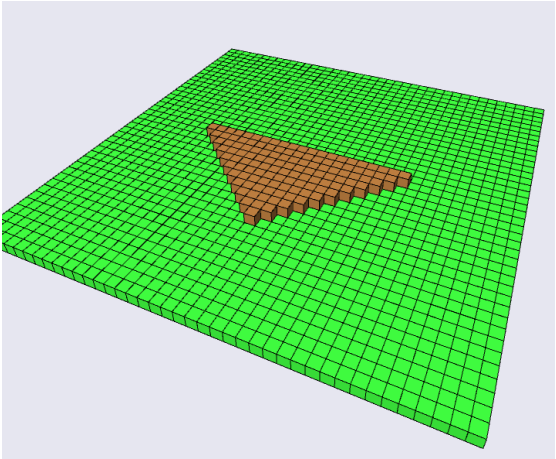
Procházení opsaného obdélníka po řádcích

Zbytečný průchod $\sim 1/2$ obdélníka.



Ukázka - Pinedův algoritmus v Processing

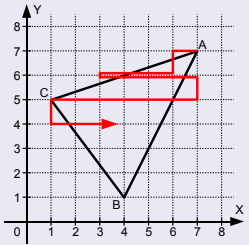
- Rasterizace troj. v rovině XY



Lepší algoritmus

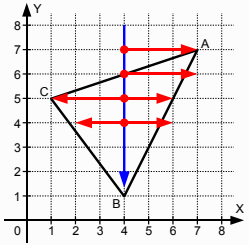
Procházení od maximálního vrcholu s obratem a přechodem na další řádek

Algoritmicky složitější.



Procházení po řádcích od svislého plotu

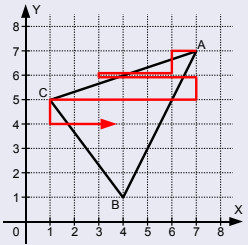
Plot ve středu opsaného obdélníka nebo skrz prostřední vrchol. Možnost paralelizace (strany nezávisle).



Lepší algoritmus

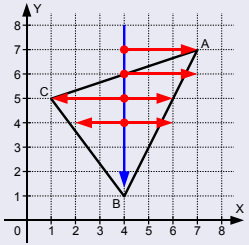
Procházení od maximálního vrcholu s obratem a přechodem na další řádek

Algoritmicky složitější.

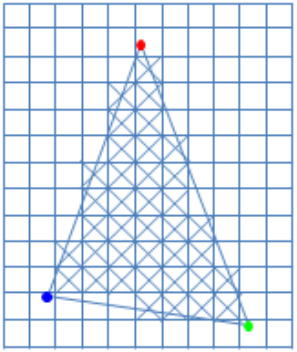


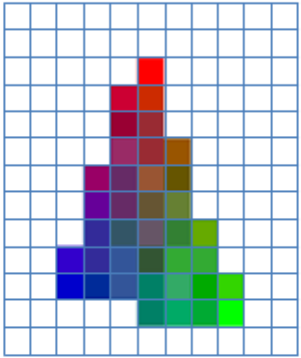
Procházení po řádcích od svislého plotu

Plot ve středu opsaného obdélníka nebo skrz prostřední vrchol. Možnost paralelizace (strany nezávisle).



Jak interpolovat barvu v ploše trojúhelníku?



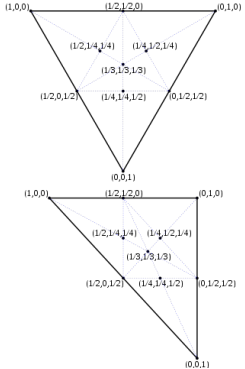


Barycentrické souřadnice

Hlavní myšlenka

- Souřadnice vztažené relativně k n zvoleným bodům.
- Pro $n-$úhelník lze jakýkoliv bod vyjádřit n souřadnicemi.
- Představují vliv ”hmotnosti” bodů N-úhelníku v konkrétním bodě (barycenter = těžiště).

Hodnoty bar. souřadnic v konkrétních bodech trojúhelníku:



Barycentrické souřadnice pro trojúhelník

Vlastnosti

- Mějme libovolný bod $r = (\lambda_1, \lambda_2, \lambda_3)$ v barycentrických souřadnicích
- Mějme p_1, p_2, p_3 body trojúhelníku
- Pak platí
  - $r = \lambda_1 p_1 + \lambda_2 p_2 + \lambda_3 p_3$
  - $\lambda_1 + \lambda_2 + \lambda_3 = 1$ pokud bod r náleží trojúhelníku (p_1, p_2, p_3)

Vhodné pro určení polohy bodu vůči trojúhelníku

- $\lambda_1 + \lambda_2 + \lambda_3 = 1$ pokud bod r náleží trojúhelníku (p_1, p_2, p_3)
- $\lambda_1 + \lambda_2 + \lambda_3 \neq 1$ pokud bod r leží mimo trojúhelník (p_1, p_2, p_3)
- $\lambda_1 = 0$ a $\lambda_2 + \lambda_3 = 1$ pokud bod r leží na hraně protilehlé bodu p_1

Barycentrické souřadnice pro trojúhelník

Vlastnosti

- Mějme libovolný bod $r = (\lambda_1, \lambda_2, \lambda_3)$ v barycentrických souřadnicích
- Mějme p_1, p_2, p_3 body trojúhelníku
- Pak platí
  - $r = \lambda_1 p_1 + \lambda_2 p_2 + \lambda_3 p_3$
  - $\lambda_1 + \lambda_2 + \lambda_3 = 1$ pokud bod r náleží trojúhelníku (p_1, p_2, p_3)

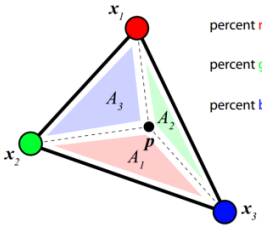
Vhodné pro určení polohy bodu vůči trojúhelníku

- $\lambda_1 + \lambda_2 + \lambda_3 = 1$ pokud bod r náleží trojúhelníku (p_1, p_2, p_3)
- $\lambda_1 + \lambda_2 + \lambda_3 \neq 1$ pokud bod r leží mimo trojúhelník (p_1, p_2, p_3)
- $\lambda_1 = 0$ a $\lambda_2 + \lambda_3 = 1$ pokud bod r leží na hraně protilehlé bodu p_1

Výpočet barycentrických souřadnic pro trojúhelník

Konverze z kartézských souřadnic

- $\lambda_1 = A(r, p_2, p_3)/A(p_1, p_2, p_3)$
  $\lambda_2 = A(r, p_1, p_3)/A(p_2, p_1, p_3)$
  $\lambda_3 = A(r, p_1, p_2)/A(p_3, p_1, p_2)$



Barycentrické souřadnice pro N-úhelník

Zobecnění souřadnic pro N-úhelník

- Pro N-úhelník je bod vyjádřen n souřadnicemi.
- Platí

$$\sum_{i=1..n} \lambda_i = 1$$

pro bod náležící n-úhelníku.

- Platí

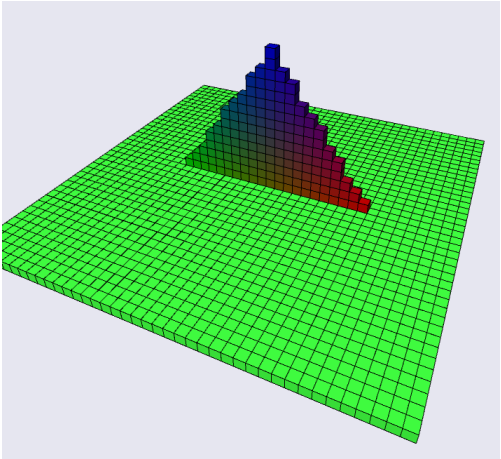
$$r = \sum_{i=1..n} \lambda_i r_i$$

pro libovolný bod.

Ukázka - Interpolace barev z vrcholů pomocí bar. souřadnic

- Vazba mezi výpočtem bar. souřadnic a hranovou fcí Pinedova algoritmu

https://youtu.be/eu6i7WJeinw?list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab



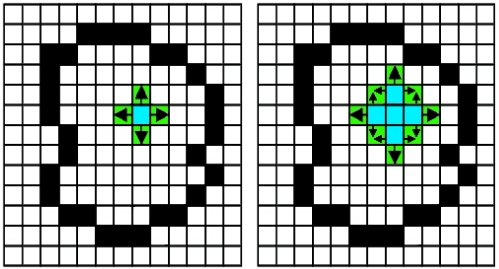
Obsah

Semínkové vyplňování (angl. Flood-fill)

- Vyplňování rastrových oblastí.
- Startovací bod (semínko) uvnitř oblasti.

Základní myšlenka

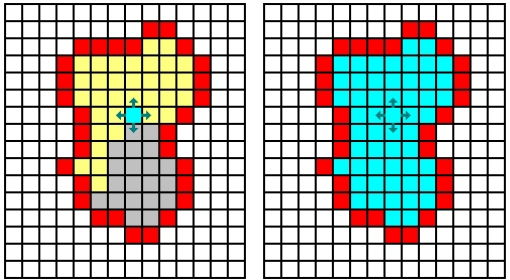
- Semínko uvnitř oblasti šíříme na sousedy (obarvování sousedních pixelů).
- Obarvené pixely se rekurzivně stávají semínky.



Definice hranice oblasti

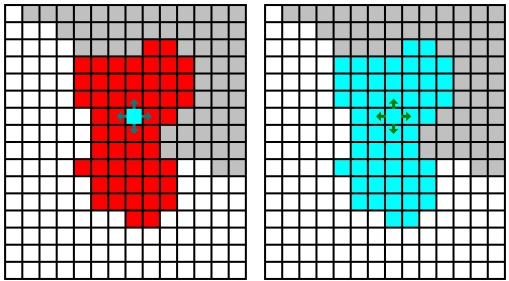
Hraniční

Oblast definována spojitou hranicí z pixelů dané barvy.



Záplavová

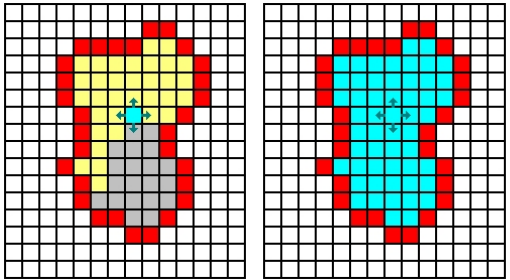
Oblast definována spojitou množinou vnitřních pixelů dané barvy.



Definice hranice oblasti

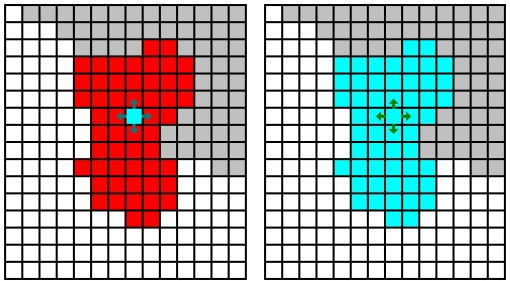
Hraniční

Oblast definována spojitou hranicí z pixelů dané barvy.



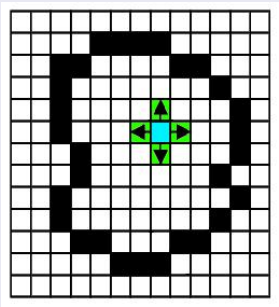
Záplavová

Oblast definována spojitou množinou vnitřních pixelů dané barvy.



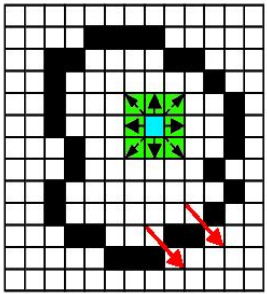
Způsob výběru sousedů

4-okolí



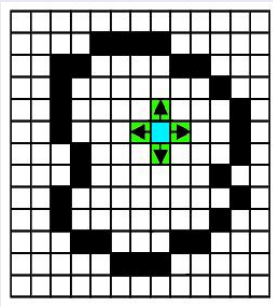
8-okolí

Vyžaduje ”silnější”hranice.



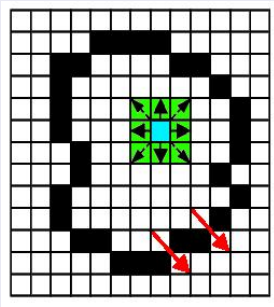
Způsob výběru sousedů

4-okolí



8-okolí

Vyžaduje ”silnější”hranice.



Příklady implementace

Rekurzivní implementace

```
FloodFill(int x, int y)
{
    if (get_pixel(x,y) == background_color)
    {   set_pixel(x, y, fill_color);
        FloodFill(x, y - 1);
        FloodFill(x, y + 1);
        FloodFill(x - 1, y);
        FloodFill(x + 1, y);
    }
}
```

- Hrozí nebezpečí přetečení zásobníku!

Implementace s využitím fronty/zásobníku

- Vlož počáteční semínko (x_0, y_0) do fronty.
- Dokud fronta není prázdná:
  - Vyber semínko (x_i, y_i) ze začátku fronty.
  - Je-li uvnitř oblasti, pak nastav hodnotu pixelu a vlož sousedy $(x_i + 1, y_i), \ldots$ do fronty.

Příklady implementace

Rekurzivní implementace

```
FloodFill(int x, int y)
{
    if (get_pixel(x,y) == background_color)
    {   set_pixel(x, y, fill_color);
        FloodFill(x, y - 1);
        FloodFill(x, y + 1);
        FloodFill(x - 1, y);
        FloodFill(x + 1, y);
    }
}
```

- Hrozí nebezpečí přetečení zásobníku!

Implementace s využitím fronty/zásobníku

- Vlož počáteční semínko (x_0, y_0) do fronty.
- Dokud fronta není prázdná:
  - Vyber semínko (x_i, y_i) ze začátku fronty.
  - Je-li uvnitř oblasti, pak nastav hodnotu pixelu a vlož sousedy $(x_i + 1, y_i), \ldots$ do fronty.

Příklady implementace

Optimalizace vyplňování

- Řádkové vyplňování se seznamem aktivních hran.
- Řádkové semínkové vyplňování (angl. scanline seed fill).

Druhy výplní

- Použití šablon pro šrafování.
- Vyplnění vzorkem.

Literatura

- Žára, J., Beneš, B., Felkel, P., *Moderní počítačová grafika*, ComputerPress, 1999.

Příklady implementace

Optimalizace vyplňování

- Řádkové vyplňování se seznamem aktivních hran.
- Řádkové semínkové vyplňování (angl. scanline seed fill).

Druhy výplní

- Použití šablon pro šrafování.
- Vyplnění vzorkem.

Literatura

- Žára, J., Beneš, B., Felkel, P., *Moderní počítačová grafika*, ComputerPress, 1999.

Příklady implementace

Optimalizace vyplňování

- Řádkové vyplňování se seznamem aktivních hran.
- Řádkové semínkové vyplňování (angl. scanline seed fill).

Druhy výplní

- Použití šablon pro šrafování.
- Vyplnění vzorkem.

Literatura

- Žára, J., Beneš, B., Felkel, P., *Moderní počítačová grafika*, ComputerPress, 1999.