

Přednáška 5

Ořezávání v rovině



Ořezávací oblast

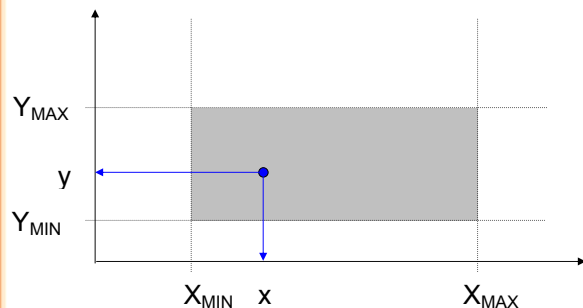
- Určení viditelné oblasti uživatelského souřadnicového systému USS
- Vykreslení pouze té části obrazu, která leží ve viditelné oblasti
- Obvykle je viditelnou (ořezávací) oblastí osově orientovaný pravoúhelník (většinou okno souřadnicového systému zařízení – SSZ).
- Cílem může být:
 - zrychlení grafického výstupu (v případě, že velké množství objektů leží mimo viditelnou oblast a zjištění této skutečnosti je rychlejší než samotné kreslení)
 - vykreslení obrazu pouze do určité části grafického výstupního zařízení
 - náhrada ořezávání na úrovni grafického výstupního zařízení

PŘEDNÁŠKA: 5 / 2

Test polohy bodu

- V případě ořezávání osově orientovaným pravoúhelníkem stačí vyhodnotit souřadnice vykreslovaného bodu $[x,y]$ vzhledem k hraničním pravoúhelníku.

$$(x > X_{\min}) \wedge (x < X_{\max}) \wedge (y > Y_{\min}) \wedge (y < Y_{\max})$$



PŘEDNÁŠKA: 5 / 3

V případě ořezávání konvexním mnohoúhelníkem je třeba vyhodnotit polohu bodu vůči všem hraničním úsečkám.

Polohu bodu P a úsečky AB lze určit pomocí velikosti

vektorového součinu $\vec{u}(\overline{AB})$ a $\vec{v}(\overline{AP})$

$$S = (b_x - a_x)(p_y - a_y) - (b_y - a_y)(p_x - a_x)$$

$$\vec{u} \times \vec{v} = \begin{vmatrix} i & j & k \\ u_x & u_y & u_z \\ v_x & v_y & v_z \end{vmatrix} = \begin{vmatrix} i & j \\ b_x - a_x & b_y - a_y \\ p_x - a_x & p_y - a_y \end{vmatrix} \begin{matrix} k \\ 0 \\ 0 \end{matrix}$$

⇒ $S < 0$ bod leží napravo od úsečky AB

⇒ $S = 0$ bod leží na úsečce AB

⇒ $S > 0$ bod leží nalevo od úsečky AB

Pokud je mnohoúhelník definován posloupností vrcholových uzlů

⇒ **proti směru** hodinových ručiček, potom vyšetřovaný bod leží uvnitř, pokud leží **vlevo** od všech úseček.

⇒ **po směru** hodinových ručiček, potom vyšetřovaný bod leží uvnitř, pokud leží **vpravo** od všech úseček.

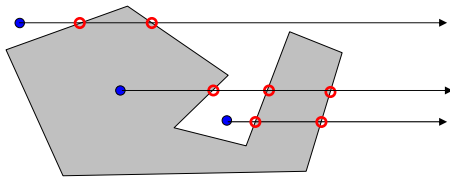
PŘEDNÁŠKA: 5 / 4



Ořezání úsečky (Cohen-Sutherland)

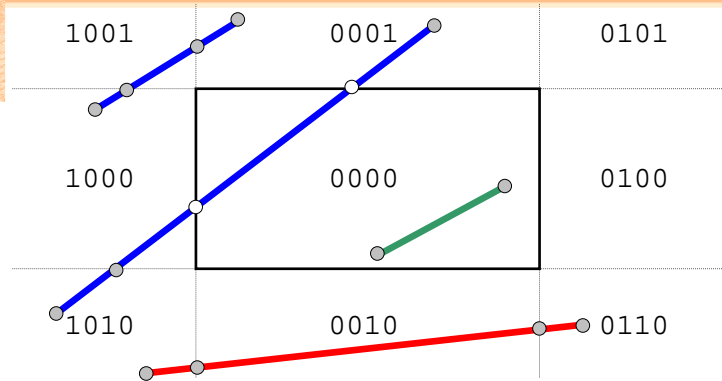
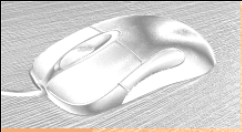
- V případě ořezávání nekonvexním mnohoúhelníkem je třeba vyhodnotit počet průsečíků polopřímky vedené z vyšetřovaného bodu P libovolným směrem s hranami mnohoúhelníku. Problémy nastanou s body na hranách a v některých speciálních případech.

Sudý počet průsečíků.....bod leží vně
Lichý počet průsečíků.....bod leží uvnitř



- Založeno na rozdělení celé plochy USS na jednotlivé oblasti
- Oblastem je přiřazen 4-bitový kód podle polohy vzhledem k poloze ořezávací oblasti (vlevo, vpravo, dole, nahoře)

1001	0001	0101
1000	0000	0100
1010	0010	0110



- Pro každý počáteční (Z) a koncový (K) bod úsečky určí kód.
- Pokud $kód(Z) \cup kód(K) = 0$ úsečka je celá uvnitř
- Pokud $kód(Z) \cap kód(K) \neq 0$ úsečka je celá mimo
- Pokud $kód(Z) \cap kód(K) = 0$ úsečka prochází více oblastmi a je třeba ji oříznout (ořezávání se provádí dle libovolné jedničky v kódech pro Z a K a testy s novými kódy se opakují).



Ořezání úsečky (Liang-Barsky)

- Založeno na parametrickém vyjádření úsečky
- Minimalizuje počet zbytečně určených nových hraničních bodů
- Ořezávání pravoúhelníkem, jehož hrany jsou rovnoběžné s osami SS
- Metoda Cyrus-Beck je zobecněním pro ořezávání obecným konvexním mnohoúhelníkem.

Princip Liang-Barsky metody (1)

- Úsečka je definována body se souřadnicemi (x_1, y_1) a (x_2, y_2) .
- Parametrické vyjádření úsečky:

$$x = x_1 + u \Delta x$$

$$y = y_1 + u \Delta y, \text{ kde}$$

- Parametr u nabývá hodnot $<0, 1>$.
- $\Delta x = (x_2 - x_1)$ a $\Delta y = (y_2 - y_1)$
- Pro každý bod úsečky, ležící uvnitř ořezávané oblasti platí:

$$X_{\min} \leq x_1 + u \Delta x$$

$$x_1 + u \Delta x \leq X_{\max}$$

$$Y_{\min} \leq y_1 + u \Delta y$$

$$y_1 + u \Delta y \leq Y_{\max}$$



Princip Liang-Barsky metody (2)

- Přepis uvedených nerovností do tvaru: $u p_k \leq q_k, \text{ pro } k = 1, 2, 3, 4$
- Definice hodnot p_k a q_k :

$$k = 1: \quad p_1 = -\Delta x \quad q_1 = x_1 - X_{\min}$$

$$k = 2: \quad p_2 = \Delta x \quad q_2 = X_{\max} - x_1$$

$$k = 3: \quad p_3 = -\Delta y \quad q_3 = y_1 - Y_{\min}$$

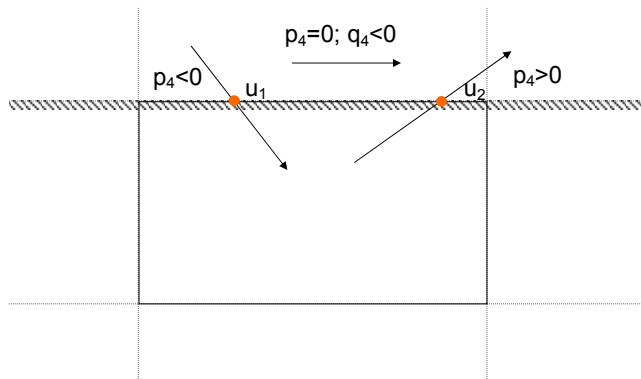
$$k = 4: \quad p_4 = \Delta y \quad q_4 = Y_{\max} - y_1$$

- Pokud některé $p_k=0$, potom úsečka je rovnoběžná s příslušnou hranou ořezávací oblasti.
- Pokud některé $p_k < 0$, potom úsečka (přímka) směřuje dovnitř oblasti (vzhledem na příslušnou hranici ořezávací oblasti).
- Pokud některé $p_k > 0$, potom úsečka (přímka) směřuje ven z oblasti (vzhledem na příslušnou hranici ořezávací oblasti).



Princip Liang-Barsky metody (3)

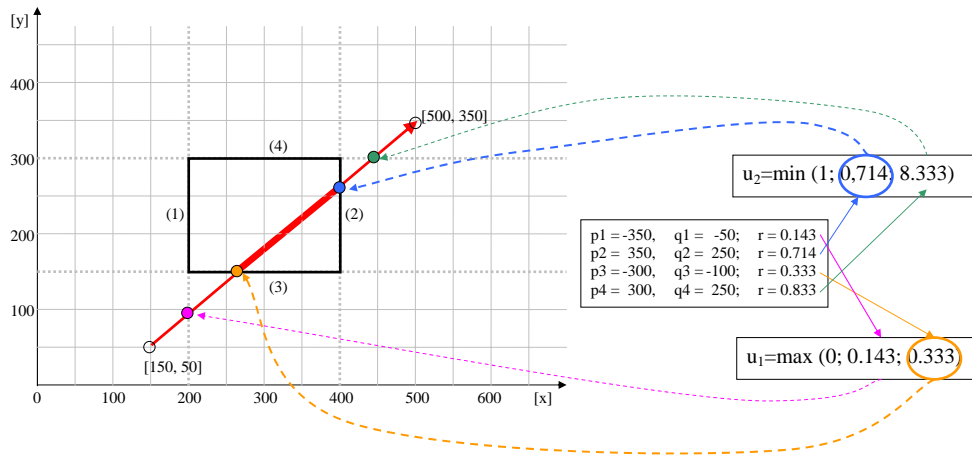
- Význam paramterů p_k a q_k pro jednu hranu ořezávací oblasti



Princip Liang-Barsky metody (4)

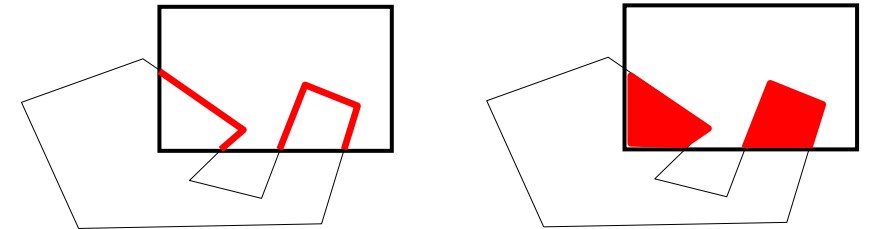
- V případě $p_k=0 \wedge q_k < 0$ leží rovnoběžná úsečka "před" hranicí a je možno ji vynechat.
- Pro každou podmínku, kde $p_k \neq 0$, je možno spočítat parametr $u = q_k / p_k$, který určuje bod průsečíku úsečky (přímky) s danou hranicí přezávací oblasti.
- Pro všechny podmínky, kde $p_k < 0$ (úsečka směřuje dovnitř) určíme parametr $r_k = q_k / p_k$ a **výsledný parametr u_1** (počátečního bodu **ořezané** úsečky) vybereme **maximum** z vypočítaných hodnot r_k a hodnoty **0 (nula)**.
- Pro všechny podmínky, kde $p_k > 0$ (úsečka směřuje ven) určíme parametr $r_k = q_k / p_k$ a **výsledný parametr u_2** (koncového bodu **ořezané** úsečky) vybereme **minimum** z vypočítaných hodnot r_k a hodnoty **1 (jedna)**.
- Ořezaná ús. je dána par. u_1 a u_2 , kde $u_1 < u_2$, u_1 a $u_2 \in <0, 1>$.

Příklad Liang-Barsky metody

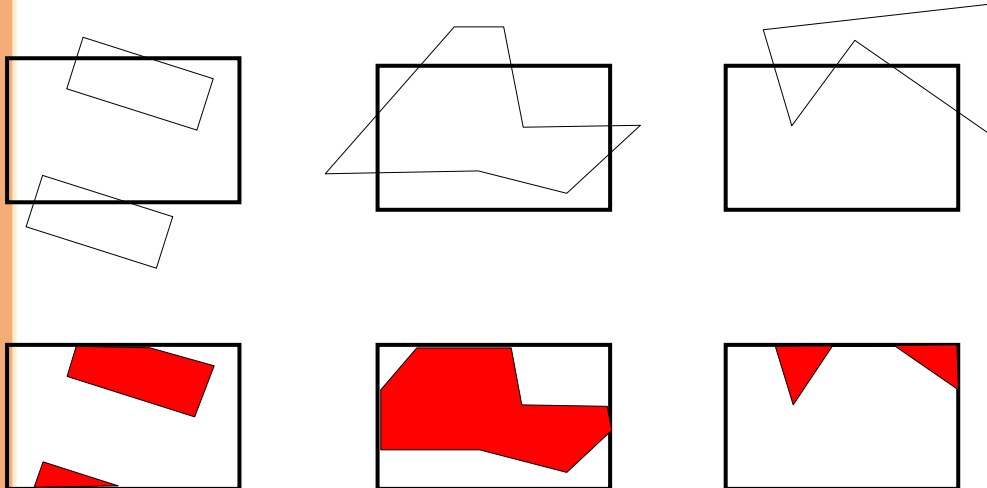


Ořezávání n-úhelníku

- Ořezávání jednotlivých úseček (tvořících hranici polygonu) pomocí předcházejících algoritmů – lze použít pouze pro vykreslení obrysu ořezaného polygonu.
- Pro vyplňování je nutno použít jiný algoritmus, například proudové ořezávání metodou Sutherland-Hodgman.



- Různé varianty, vznikající při ořezávání n-úhelníků.



Sutherland-Hodgman

- Proudové ořezávání
- Ořez se provádí 4x, vždy pro jednu hranici okna.
- Je nutno v případě potřeby vytvářet nové hraniční úsečky tak, aby oblast zůstala „spojitá“ a uzavřená.
- Při implementaci lze používat 4x stejný kód (modul) a ořezávání jednotlivými hranami okna realizovat pomocí otáčení o 90°.
- Modul probírá postupně všechny aktuální vrcholy polygonu.

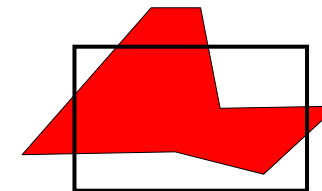
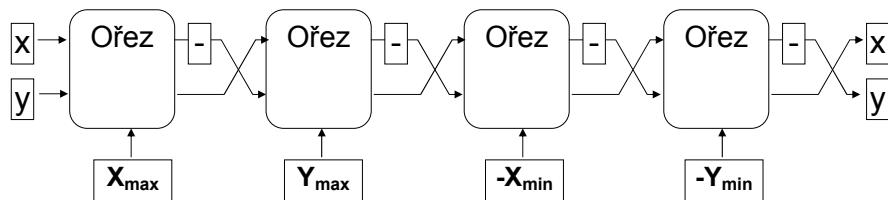
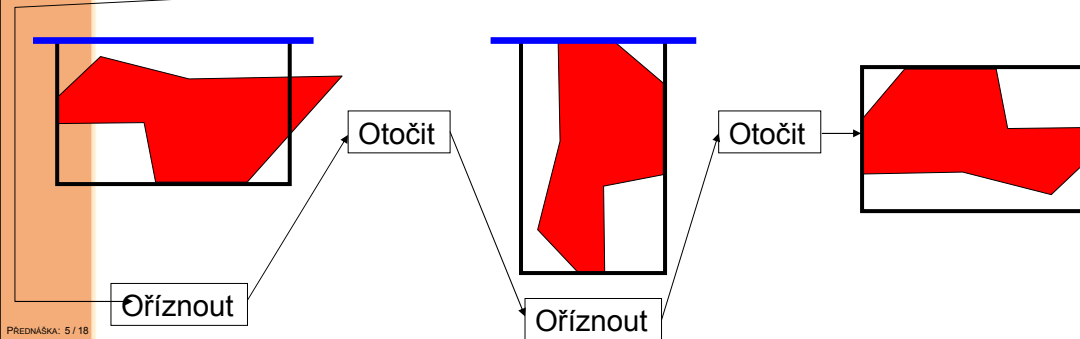
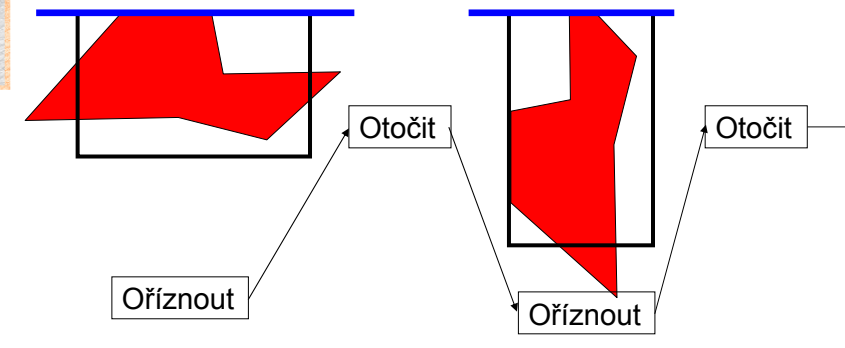


Schéma proudového ořezávání pomocí jednotného modulu

- Modul provede ořezání podle jedné hranice okna.
- Aktualizuje seznam vrcholů polygonu.
- Otáčení se provádí záměnou x a y-ových souřadnic současně se změnou znaménka.



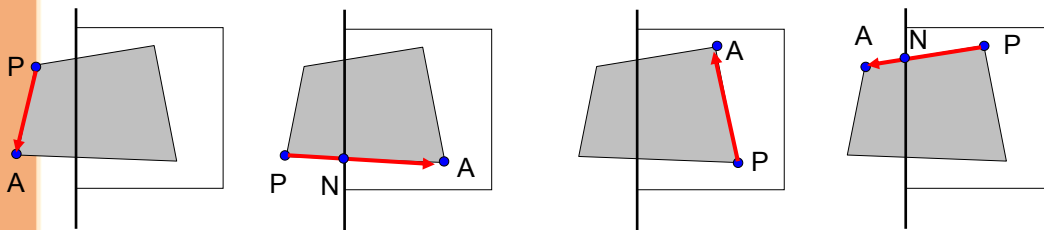
PŘEDNÁŠKA: 5 / 17



PŘEDNÁŠKA: 5 / 18

Aktualizace vrcholů

- Modul provádí vkládání nových vrcholů, popřípadě rušení vrcholů na základě polohy právě vyšetřovaného a předcházejícího vrcholu vůči hranici.
- 4 případy polohy A (aktuální) a P (předcházející)



a/ Vrchol A se ruší a nepostupuje do dalšího zpracování

b/ Vkládá se nový vrchol N, vrchol A postupuje nezměněn

c/ Vrchol A postupuje nezměněn

d/ Vrchol A je nahrazen novým vrcholem N

PŘEDNÁŠKA: 5 / 19

Stručný algoritmus modulu

- **vstup:** souřadnice vrcholů polygonu (pole, seznam), hranice
- **inicializace:**
 - Vynulovat pole *Aktualizovane_vrcholy*, *PocetAktVrcholu* := 0
 - *P* := poslední vrchol polygonu
- **postupně pro všechny vrcholy $V_i := 1..PočetVrcholu$ proved'**
 - **pokud hrana PV_i je celá vně,**
 - potom nic (pokračuj dalším vrcholem)
 - **pokud hrana PV_i směřuje z venku dovnitř,**
 - potom vypočítej průsečík *N* s hranicí, do pole přidej *N* a V_i ; *Pocet*: +2
 - **pokud hrana PV_i je celá uvnitř,**
 - potom přidej do pole V_i ; *Pocet*: +1
 - **pokud hrana PV_i směřuje zevnitř ven,**
 - potom vypočítej průsečík *N* s hranicí a přidej *N* do pole; *Pocet*: +1
 - **$P := V_i$**

PŘEDNÁŠKA: 5 / 20