

Základy počítačové grafiky

Ořezávání objektů

Michal Španěl
Tomáš Milet



Ústav počítačové grafiky a multimédií

Brno 2023

Cíl přednášky

Zobrazovat to, co stejně není vidět je očividně zbytečné!
Jak však poznat, co není vidět?

Seznámit se s algoritmy pro rychlé ořezávání vektorových primitiv (úsečka, polygon) ve 2D a principy ořezávání ve 3D.

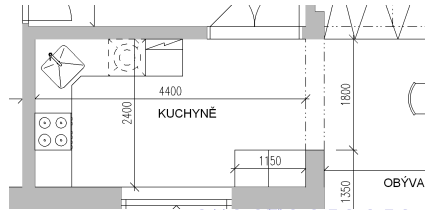
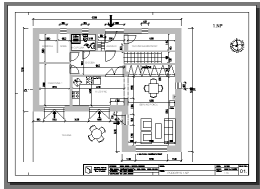
Obsah

- 1 Úvod
 - Základní přístupy
- 2 Ořezání úsečky
 - Cohen-Sutherland
 - Liang-Barsky
- 3 Ořezávání polygonů
 - Sutherland-Hodgman
 - Weiler-Atherton
- 4 Ořezání v zobrazovacím řetězci
 - Clipping a clip space
 - Plné projekční matice

Ořezávání objektů ve 2D

Výběr zobrazovaných objektů vzhledem k aktuální poloze a rozměru zobrazovacího okna.

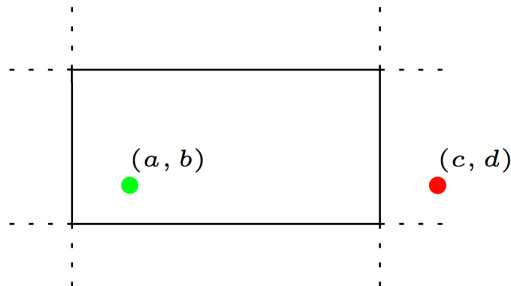
- Objekty mimo okno, budou vyloučeny.
- Objekty, které hranice okna protínají, budou rozděleny (oříznuty).
- Výsledkem vždy objekt stejného typu (úsečka, polygon, atd.), jako původní.



Příliš snadný "přístup"

Test polohy rasterizovaných bodů

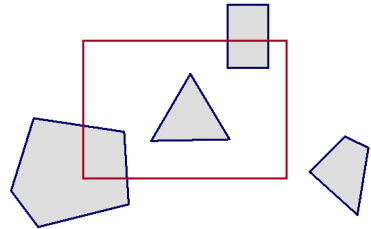
- Souřadnice pixelů získané rasterizací se porovnávají se zobrazovacím oknem a vykreslují se pouze vnitřní body.
- Nejprimitivnější způsob, neefektivní.



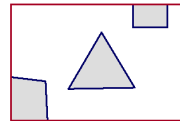
Lepší přístup

Ořezávání vektorových entit

- Nejčastěji úsečky a mnohoúhelníky definované po částech lineární hranicí - polyline, polygon.
- Zobrazovací okno je obecně definováno polygonem.
- Pro další výklad předpokládáme okno ve tvaru obdélníku se stranami rovnoběžnými se souřadnými osami.



Before Clipping

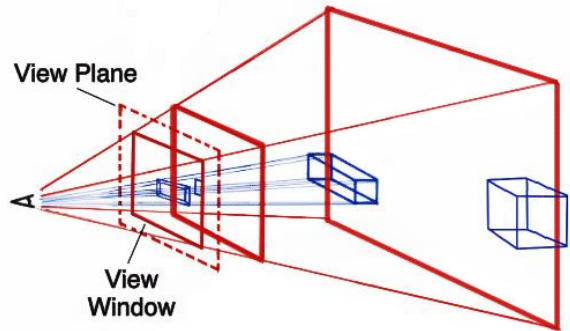


After Clipping

Ořezání ve 2D nebo 3D?

Ořezání ve 2D

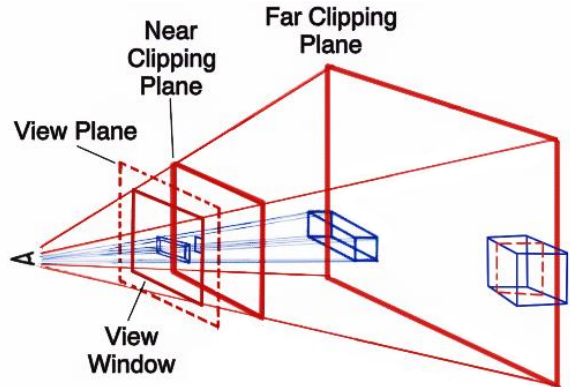
- 3D objekty jsou nejprve transformovány do roviny zobrazovacího okna a poté ořezány.
- Výrazně jednodušší a rychlejší algoritmy.
- Realizováno v HW grafické karty.



Ořezání ve 2D nebo 3D?

Ořezání ve 3D

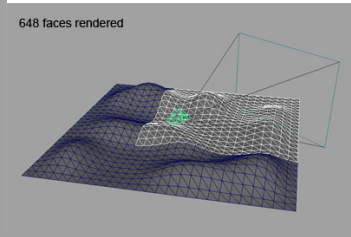
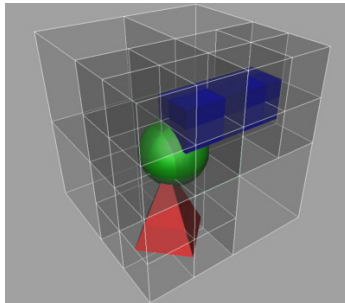
- Nejprve ořezání vzhledem k tzv. *pohledovému objemu* (angl. view frustum) přímo ve 3D, následně transformace do roviny pohledu a vykreslení.
- Při implementaci na CPU Výrazná redukce počtu entit přenášovaných do grafické karty.
- Ale implementace je náročnější...



Ořezávání vs. viditelnost objektů

Kombinace obou přístupů

- Ve 3D řešíme *viditelnost*. Entity rozdělíme na neviditelné a potencionálně viditelné (dělení prostoru scény, Octree, BSP stromy, occludery, apod.).
- Ořezávání provádíme ve 2D.
- V praxi nejčastěji používaný přístup.



Obsah

- 1 Úvod
 - Základní přístupy
- 2 Ořezání úsečky
 - Cohen-Sutherland
 - Liang-Barsky
- 3 Ořezávání polygonů
 - Sutherland-Hodgman
 - Weiler-Atherton
- 4 Ořezání v zobrazovacím řetězci
 - Clipping a clip space
 - Plné projekční matice

Ořezávání úsečky

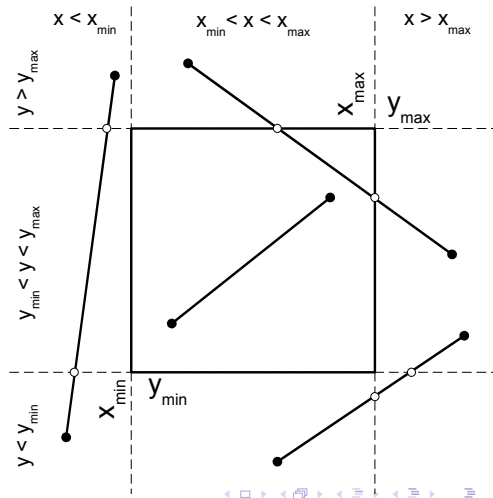
- Nejjednodušší vektorová entita.
- Definována souřadnicemi koncových bodů $P_1 = (x_1, y_1)$ a $P_2 = (x_2, y_2)$.

Test polohy úsečky vůči oknu

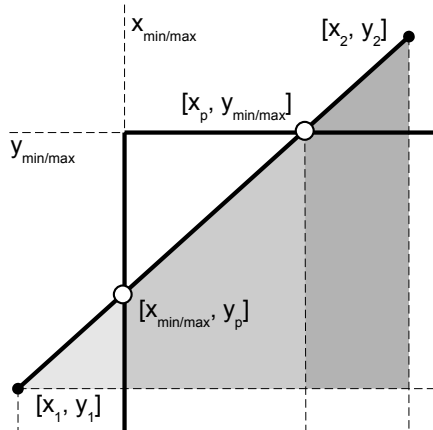
- Úsečka leží zcela mimo okno → *vyloučit* ze zpracování.
- Celá úsečka uvnitř okna → *vykreslit*.
- Protíná-li úsečka hranice okna → *rozdělit* na hranicích okna na části a znovu testovat.

Primitivní test polohy úsečky - testování koncových bodů

- Oba koncové body úsečky uvnitř okna
→ celá úsečka uvnitř okna.
- Oba koncové body na vnější straně stejné hranice → úsečka leží vně okna.
- Zbývající případy → analyzovat průsečíky, rozdělit na části a opětovně testovat.



Výpočet průsečíku úsečky s hranicemi okna



- Vztah pro výpočet průsečíku $[x_p, y_p]$ odvodíme z podobnosti trojúhelníků.

$$\frac{x_2 - x_1}{y_2 - y_1} = \frac{x_p - x_1}{y_p - y_1}$$

Výpočet průsečíku úsečky s hranicemi okna, pokr.

- Dosadíme polohy hranice okna $x_p = x_{min/max}$ pro svislé nebo $y_p = y_{min/max}$ pro vodorovné hranice okna.

$$x_p = x_{min/max}$$

$$y_p = \frac{(x_{min/max} - x_1)(y_2 - y_1)}{x_2 - x_1} + y_1$$

$$y_p = y_{min/max}$$

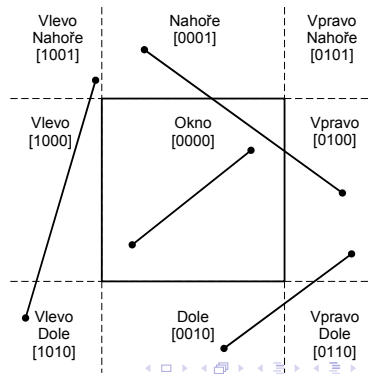
$$x_p = \frac{(x_2 - x_1)(y_{min/max} - y_1)}{y_2 - y_1} + x_1$$

Algoritmus Cohen-Sutherland

- Chytřejší testování polohy úsečky vůči oknu.
- Koncové body P_1, P_2 označeny binárními kódy C_1, C_2 oblastí.
- Devět kódů oblastí $\rightarrow$ 4-bity.

Rozhodování

- $C_1 = C_2 = 0 \rightarrow$ celá úsečka uvnitř okna.
- C_1 and $C_2 \neq 0 \rightarrow$ celá úsečka mimo okno.
- C_1 and $C_2 = 0 \rightarrow$ úsečka potenciálně protíná hranice.

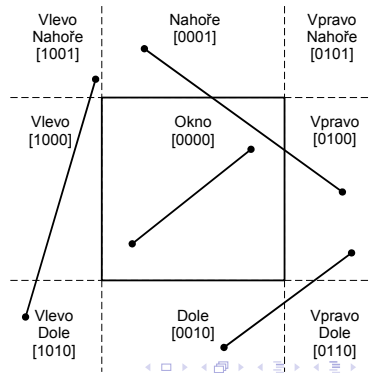


Algoritmus Cohen-Sutherland

- Chytřejší testování polohy úsečky vůči oknu.
- Koncové body P_1, P_2 označeny binárními kódy C_1, C_2 oblastí.
- Devět kódů oblastí $\rightarrow$ 4-bity.

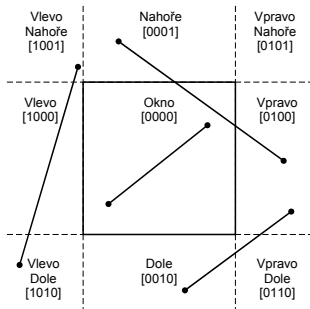
Rozhodování

- $C_1 = C_2 = 0 \rightarrow$ celá úsečka uvnitř okna.
- C_1 and $C_2 \neq 0 \rightarrow$ celá úsečka mimo okno.
- C_1 and $C_2 = 0 \rightarrow$ úsečka potenciálně protíná hranice.



Úsečka potencionálně protíná hranice

- Určit průsečíky, rozdělit úsečku na části a znovu testovat polohu.
- Nastavené bity v hodnotě binárního výrazu C_1 or C_2 odpovídají zasaženým hranicím okna.



Liang-Barsky

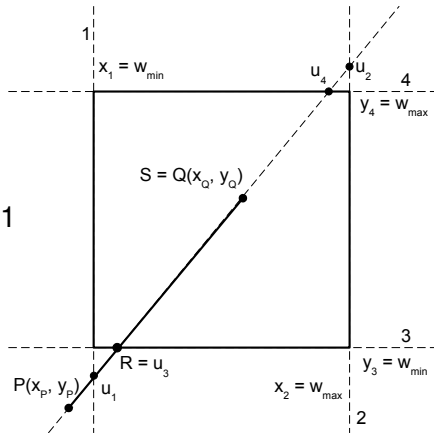
- Řešení v *parametrické oblasti*.
- Úsečka dána body $P[x_P, y_P]$, $Q[x_Q, y_Q]$ a rovnicemi:

$$x = x_P + (x_Q - x_P) \cdot u = x_P + \Delta x \cdot u$$

$$y = y_P + (y_Q - y_P) \cdot u = y_P + \Delta y \cdot u, \quad 0 \leq u \leq 1$$

Základní myšlenka

- Hledáme část úsečky ležící uvnitř okna, nebo-li hodnoty parametru u .



Liang-Barsky

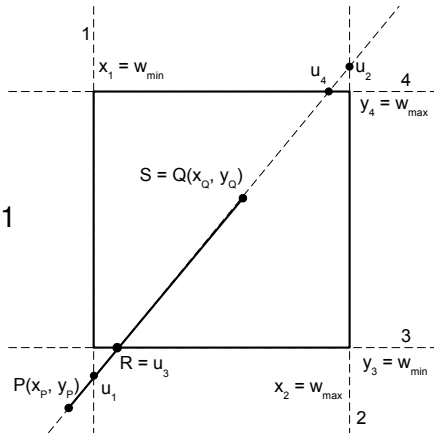
- Řešení v *parametrické oblasti*.
- Úsečka dána body $P[x_P, y_P]$, $Q[x_Q, y_Q]$ a rovnicemi:

$$x = x_P + (x_Q - x_P) \cdot u = x_P + \Delta x \cdot u$$

$$y = y_P + (y_Q - y_P) \cdot u = y_P + \Delta y \cdot u, \quad 0 \leq u \leq 1$$

Základní myšlenka

- Hledáme část úsečky ležící uvnitř okna, nebo-li hodnoty parametru u .



Ořezání úsečky v parametrickém prostoru

- Uvnitř okna platí nerovnice

$$x_{wmin} \leq x_P + \Delta x \cdot u \leq x_{wmax}$$

$$y_{wmin} \leq y_P + \Delta y \cdot u \leq y_{wmax}$$

- Přepíšeme do tvaru

$$u \leq \frac{q_k}{p_k}, \quad k = 1, 2, 3, 4$$

$$p_1 = -\Delta x, \quad q_1 = x_P - x_{wmin}$$

$$p_2 = \Delta x, \quad q_2 = x_{wmax} - x_P$$

$$p_3 = -\Delta y, \quad q_3 = y_P - y_{wmin}$$

$$p_4 = \Delta y, \quad q_4 = y_{wmax} - y_P$$

Ořezání úsečky v parametrickém prostoru

- Uvnitř okna platí nerovnice

$$x_{wmin} \leq x_P + \Delta x \cdot u \leq x_{wmax}$$

$$y_{wmin} \leq y_P + \Delta y \cdot u \leq y_{wmax}$$

- Přepíšeme do tvaru

$$u \leq \frac{q_k}{p_k}, \quad k = 1, 2, 3, 4$$

$$p_1 = -\Delta x, \quad q_1 = x_P - x_{wmin}$$

$$p_2 = \Delta x, \quad q_2 = x_{wmax} - x_P$$

$$p_3 = -\Delta y, \quad q_3 = y_P - y_{wmin}$$

$$p_4 = \Delta y, \quad q_4 = y_{wmax} - y_P$$

Pravidla určování polohy úsečky vůči oknu

$$p_k = 0$$

- Úsečka rovnoběžná s hranicí k .

$$p_k = 0 \text{ a } q_k < 0$$

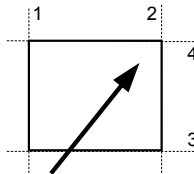
- Úsečka zcela mimo okno.

$$p_k < 0$$

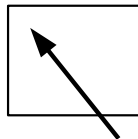
- Přímka "vniká" do okna přes hranici k .

$$p_k > 0$$

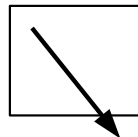
- Přímka "opouští" okno do bodu Q .



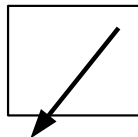
$$p_1 < 0, p_2 > 0 \\ p_3 < 0, p_4 > 0$$



$$p_1 > 0, p_2 < 0 \\ p_3 < 0, p_4 > 0$$



$$p_1 > 0, p_2 < 0 \\ p_3 > 0, p_4 < 0$$



$$p_1 < 0, p_2 > 0 \\ p_3 > 0, p_4 < 0$$

Pravidla určování polohy úsečky vůči oknu

$$p_k = 0$$

- Úsečka rovnoběžná s hranicí k .

$$p_k = 0 \text{ a } q_k < 0$$

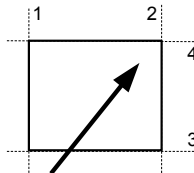
- Úsečka zcela mimo okno.

$$p_k < 0$$

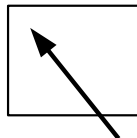
- Přímka "vniká" do okna přes hranici k .

$$p_k > 0$$

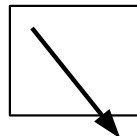
- Přímka "opouští" okno do bodu Q .



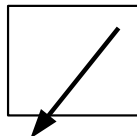
$$p_1 < 0, p_2 > 0 \\ p_3 < 0, p_4 > 0$$



$$p_1 > 0, p_2 < 0 \\ p_3 < 0, p_4 > 0$$



$$p_1 < 0, p_2 > 0 \\ p_3 > 0, p_4 < 0$$



$$p_1 > 0, p_2 < 0 \\ p_3 > 0, p_4 < 0$$

Pravidla určování polohy úsečky vůči oknu

$$p_k = 0$$

- Úsečka rovnoběžná s hranicí k .

$$p_k = 0 \text{ a } q_k < 0$$

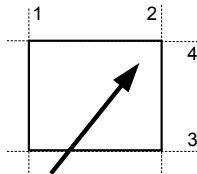
- Úsečka zcela mimo okno.

$$p_k < 0$$

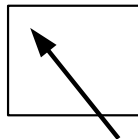
- Přímka "vniká" do okna přes hranici k .

$$p_k > 0$$

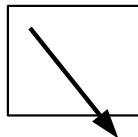
- Přímka "opouští" okno do bodu Q .



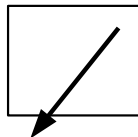
$$p_1 < 0, p_2 > 0 \\ p_3 < 0, p_4 > 0$$



$$p_1 > 0, p_2 < 0 \\ p_3 < 0, p_4 > 0$$



$$p_1 < 0, p_2 > 0 \\ p_3 > 0, p_4 < 0$$



$$p_1 > 0, p_2 < 0 \\ p_3 > 0, p_4 < 0$$

Pravidla určování polohy úsečky vůči oknu

$$p_k = 0$$

- Úsečka rovnoběžná s hranicí k .

$$p_k = 0 \text{ a } q_k < 0$$

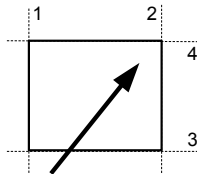
- Úsečka zcela mimo okno.

$$p_k < 0$$

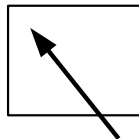
- Přímka "vniká" do okna přes hranici k .

$$p_k > 0$$

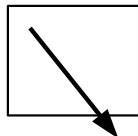
- Přímka "opouští" okno do bodu Q .



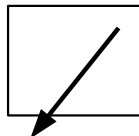
$$p_1 < 0, p_2 > 0 \\ p_3 < 0, p_4 > 0$$



$$p_1 > 0, p_2 < 0 \\ p_3 < 0, p_4 > 0$$



$$p_1 < 0, p_2 > 0 \\ p_3 > 0, p_4 < 0$$



$$p_1 > 0, p_2 < 0 \\ p_3 > 0, p_4 < 0$$

Ořezání úsečky v parametrickém prostoru, pokr.

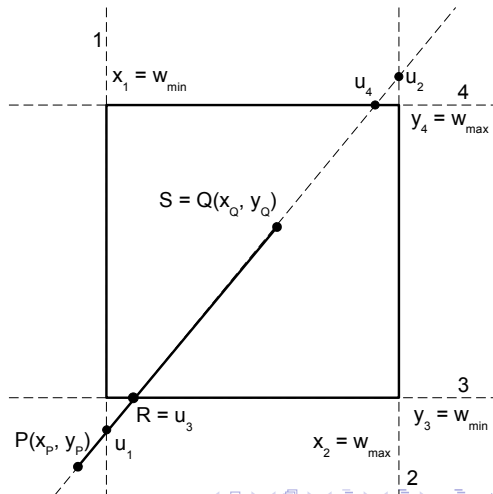
- Uvnitř okna musí platit $0 \leq u_{R,S} \leq 1$.

Nalezení parametrů u_R a u_S

$$u_R = \max (u_k (p_k < 0), 0)$$

$$u_S = \min (u_k (p_k > 0), 1)$$

$$u_k = \frac{q_k}{p_k}, \quad k = 1, 2, 3, 4$$



Ořezání úsečky v parametrickém prostoru, pokr.

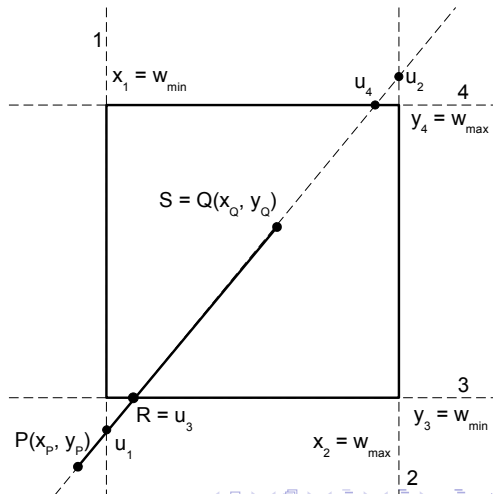
- Uvnitř okna musí platit $0 \leq u_{R,S} \leq 1$.

Nalezení parametrů u_R a u_S

$$u_R = \max (u_k (p_k < 0), 0)$$

$$u_S = \min (u_k (p_k > 0), 1)$$

$$u_k = \frac{q_k}{p_k}, \quad k = 1, 2, 3, 4$$

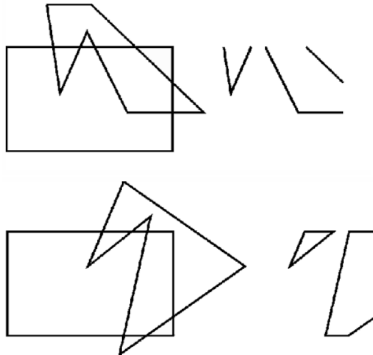


Obsah

- 1 Úvod
 - Základní přístupy
- 2 Ořezání úsečky
 - Cohen-Sutherland
 - Liang-Barsky
- 3 Ořezávání polygonů
 - Sutherland-Hodgman
 - Weiler-Atherton
- 4 Ořezání v zobrazovacím řetězci
 - Clipping a clip space
 - Plné projekční matice

Ořezávání oblastí

- Ořezávají se uzavřené oblasti, polygony (mnohoúhelníky s po částech lineární hranicí = hranice z úseček).
- Musí se zachovat uzavřené oblasti → nelze pouze ořezat hranice oblasti.

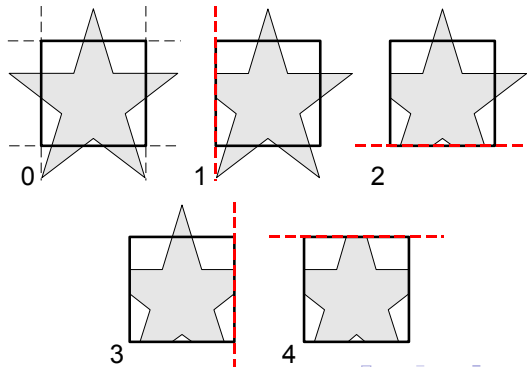


Sutherland-Hodgman

- Ořezání polygonů obdélníkovým oknem se stranami rovnoběžnými se osami X a Y.
- Jednoduchý a snadno implementovatelný algoritmus.

Princip

- Ořezání celým oknem zjednodušeno na ořezání pouze *jednou stranou*.
- Po oříznutí jednou stranou provedeme oříznutí druhou, třetí a čtvrtou stranou.

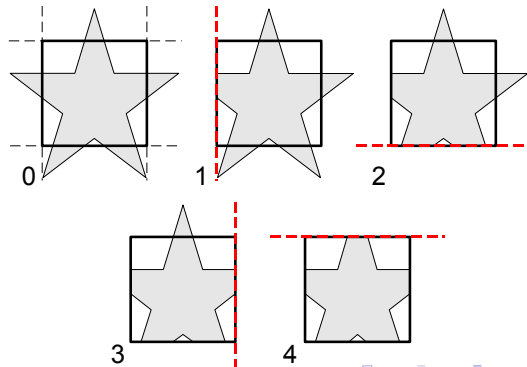


Sutherland-Hodgman

- Ořezání polygonů obdélníkovým oknem se stranami rovnoběžnými se osami X a Y.
- Jednoduchý a snadno implementovatelný algoritmus.

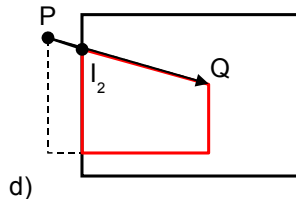
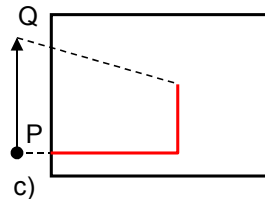
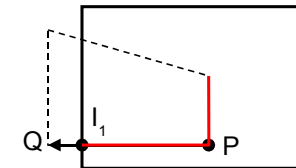
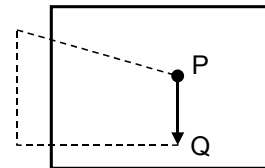
Princip

- Ořezání celým oknem zjednodušeno na ořezání pouze *jednou stranou*.
- Po oříznutí jednou stranou provedeme oříznutí druhou, třetí a čtvrtou stranou.



Postup ořezání polygonu jednou stranou okna

- Seřazení vrcholů $\rightarrow$ orientovaný seznam.
- Postupně pro každou hranu
 - Otestuj polohu úsečky (koncové body).
 - Je-li celá uvnitř $\rightarrow$ ulož do výsledného seznamu oba vrcholy.
 - Protíná stranu $\rightarrow$ vypočítej průsečík a ulož do seznamu.
 - Vnější hrany zahazujeme.



Častý způsob implementace

- Ořezáváme přímo jednou souřadnou osou.
- Postupně otáčíme a posouváme ořezávaný polygon.

Otočení polygonu o 90

- Prohození souřadnic: $x' = y, y' = -x$.

Častý způsob implementace

- Ořezáváme přímo jednou souřadnou osou.
- Postupně otáčíme a posouváme ořezávaný polygon.

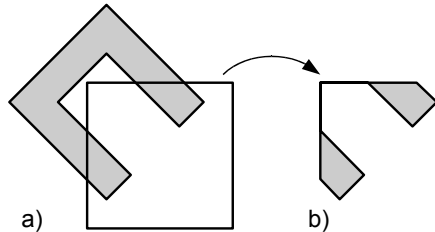
Otočení polygonu o 90

- Prohození souřadnic: $x' = y$, $y' = -x$.

Sutherland-Hodgman, pokr.

+/-

- Generování hran ořezaného polygonu podél hranic a rohů okna.
- Komplikovaná realizace polygonů s vnitřními otvory.
- Ořezávaný polygon může být definován pouze jednou uzavřenou smyčkou hraničních úsečků.

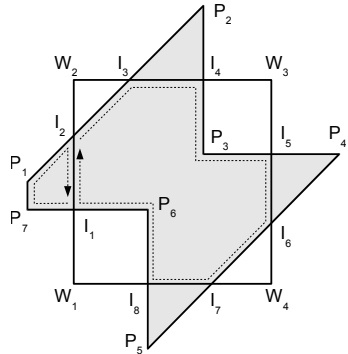


Weiler-Atherton

- Ořezání *obecných* polygonů *obecným* oknem definovaným polygonem.

Základní myšlenka

- Výpočet průsečíků s ořezávacím oknem (alg. Liang-Barsky).
- Pomocné orientované seznamy z vrcholů a průsečíků.
- Vhodný průchod seznamy a nalezení vnitřních nebo vnějších částí polygonu.

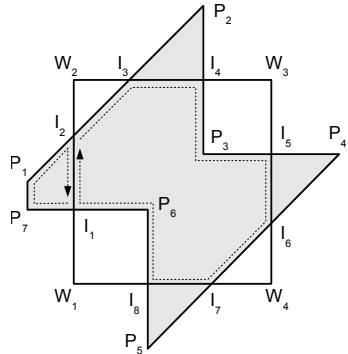


Weiler-Atherton

- Ořezání *obecných polygonů obecným oknem* definovaným polygonem.

Základní myšlenka

- Výpočet průsečíků s ořezávacím oknem (alg. Liang-Barsky).
- Pomocné orientované seznamy z vrcholů a průsečíků.
- Vhodný průchod seznamy a nalezení vnitřních nebo vnějších částí polygonu.



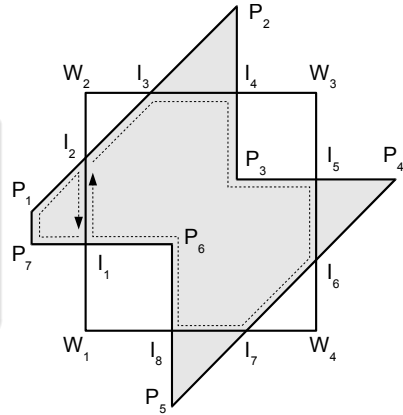
Pomocné seznamy

- P – seznam vrcholů a průsečíků po ořezávaném polygonu.
- W – seznam vrcholů a průsečíků po ořezávacím okně.
- I – seznam průsečíků na vstupu do okna.
- O – seznam průsečíků na výstupu z okna.
- C – seznam ořezaných polygonů uvnitř okna.
- R – seznam odřezků vně okna.

Příklad

Příprava seznamů

- $P - P_1, I_2, I_3, P_2, I_4, P_3, I_5, P_4, I_6, I_7, P_5, I_8, P_6, I_1, P_7, P_1$
- $W - W_1, I_1, I_2, W_2, I_3, I_4, W_3, I_5, I_6, W_4, I_7, I_8, W_1$
- $I - I_2, I_4, I_6, I_8$
- $O - I_1, I_3, I_5, I_7$



Postup ořezání polygonu

Polygony uvnitř okna, ukládáme do C

- Start v P na prvním vrcholu z I.
- Jsme-li v P nebo W na vrcholu z I, pokračujeme po P.
- Jsme-li v P nebo W na vrcholu z O, pokračujeme po W.
- Pokračujeme do uzavření obrazce.

Odřezky vně okna, ukládáme do R

- Start v P na prvním vrcholu z O.
- Jsme-li v P nebo W na vrcholu z O, pokračujeme po P.
- Jsme-li v P nebo W na vrcholu z I, pokračujeme po W.
- Pokračujeme do uzavření obrazce.

Postup ořezání polygonu

Polygony uvnitř okna, ukládáme do C

- Start v P na prvním vrcholu z I.
- Jsme-li v P nebo W na vrcholu z I, pokračujeme po P.
- Jsme-li v P nebo W na vrcholu z O, pokračujeme po W.
- Pokračujeme do uzavření obrazce.

Odřezky vně okna, ukládáme do R

- Start v P na prvním vrcholu z O.
- Jsme-li v P nebo W na vrcholu z O, pokračujeme po P.
- Jsme-li v P nebo W na vrcholu z I, pokračujeme po W.
- Pokračujeme do uzavření obrazce.

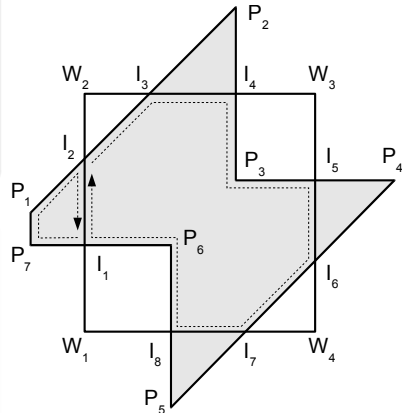
Příklad, pokr.

Orientované seznamy

- $P - P_1, I_2, I_3, P_2, I_4, P_3, I_5, P_4, I_6, I_7, P_5, I_8, P_6, I_1, P_7, P_1$
- $W - W_1, I_1, I_2, W_2, I_3, I_4, W_3, I_5, I_6, W_4, I_7, I_8, W_1$
- $I - I_2, I_4, I_6, I_8$
- $O - I_1, I_3, I_5, I_7$

Výsledek

- $C - I_2, I_3, I_4, P_3, I_5, I_6, I_7, I_8, P_6, I_1, I_2$
- $R_1 - I_1, P_7, P_1, I_2, I_1$
- $R_2 - I_3, P_2, I_4, I_3$
- $R_3 - I_5, P_4, I_6, I_5$



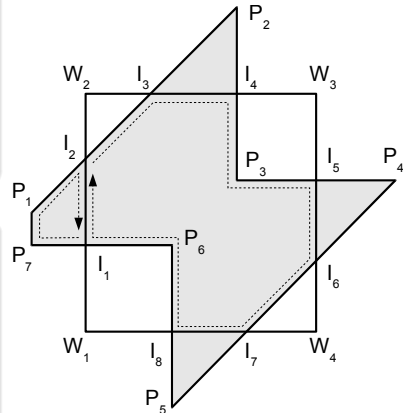
Příklad, pokr.

Orientované seznamy

- $P - P_1, I_2, I_3, P_2, I_4, P_3, I_5, P_4, I_6, I_7, P_5, I_8, P_6, I_1, P_7, P_1$
- $W - W_1, I_1, I_2, W_2, I_3, I_4, W_3, I_5, I_6, W_4, I_7, I_8, W_1$
- $I - I_2, I_4, I_6, I_8$
- $O - I_1, I_3, I_5, I_7$

Výsledek

- $C - I_2, I_3, I_4, P_3, I_5, I_6, I_7, I_8, P_6, I_1, I_2$
- $R_1 - I_1, P_7, P_1, I_2, I_1$
- $R_2 - I_3, P_2, I_4, I_3$
- $R_3 - I_5, P_4, I_6, I_5$

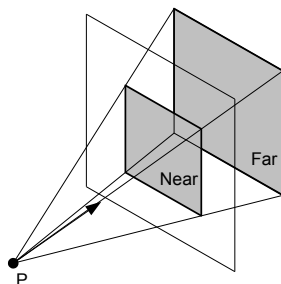
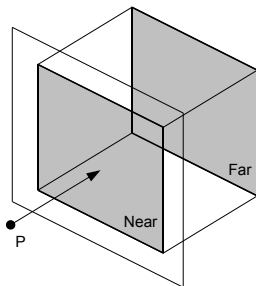


Obsah

- 1 Úvod
 - Základní přístupy
- 2 Ořezání úsečky
 - Cohen-Sutherland
 - Liang-Barsky
- 3 Ořezávání polygonů
 - Sutherland-Hodgman
 - Weiler-Atherton
- 4 Ořezání v zobrazovacím řetězci
 - Clipping a clip space
 - Plné projekční matice

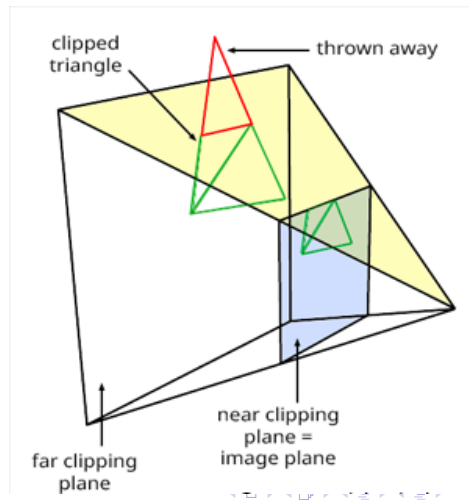
Pohledový objem

- Oblast prostoru obsahující zobrazované objekty.
- Přední a zadní ořezávací stěna.
- *Hranol* u paralelní projekce.
- *Komolý jehlan* u perspektivní projekce.



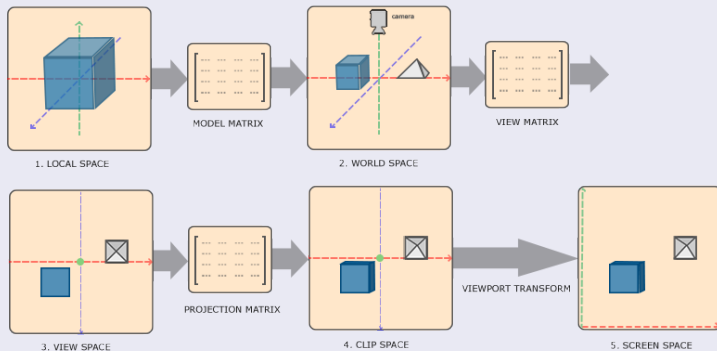
Ořezání pohledovým objemem

- Oříznutí objektů mimo hranice pohledového objemu → *urychlení procesu zobrazení.*



Clip space

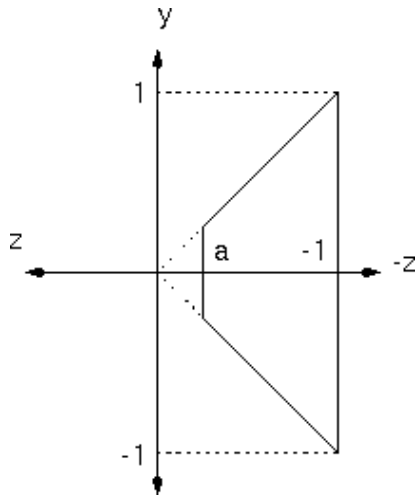
Připomenutí z pohledu transformačních matic



Clip space

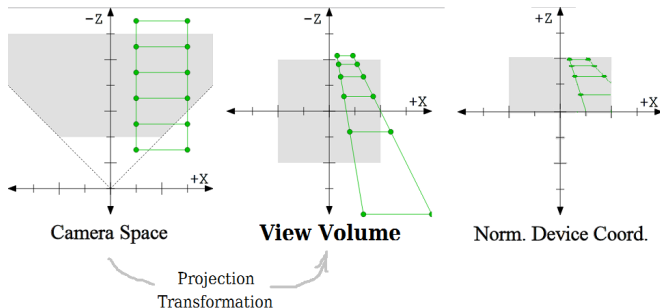
Optimalizace

- Transformace na jednotkovou krychli nebo jehlan pro snadnější ořezání.



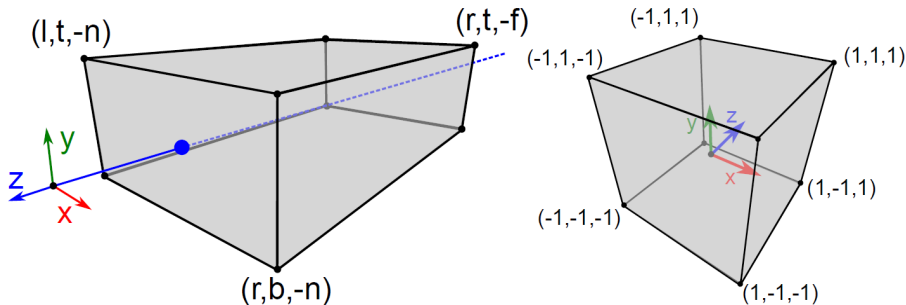
Clipping a perspektivní projekce

- Vrcholy transformujeme z *camera space* do *clip space* (projekční matice) → vrcholy v homogenních souřadnicích
- Proběhne ořezání
- Proveďte se normalizace - dělení w-souřadnicí

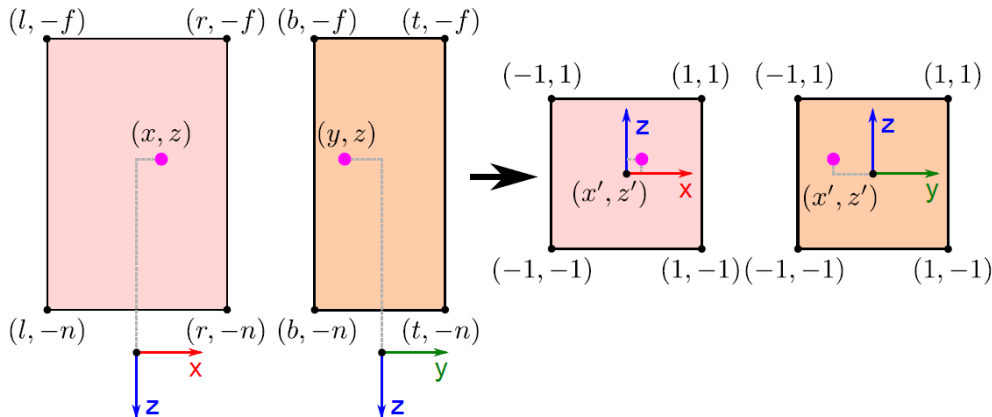


Ortogonalní projekce

- Kvádr má šest parametrů: l (left), r (right), t (top), b (bottom), n (near), f (far)
- Kvádr je umístěn relativně ke středu souřadného systému



Ortogonalní projekce, pokr.



Ortogonalní projekce - odvození rovnic pro x' , y' a z'

$x \in [l, r]$	$y \in [b, t]$	$z \in [-n, -f]$
$x - l \in [0, r - l]$	$y - b \in [0, t - b]$	$z + n \in [0, n - f]$
$\frac{x - l}{r - l} \in [0, 1]$	$\frac{y - b}{t - b} \in [0, 1]$	$\frac{z + n}{n - f} \in [0, 1]$
$2\frac{x - l}{r - l} \in [0, 2]$	$2\frac{y - b}{t - b} \in [0, 2]$	$2\frac{z + n}{n - f} \in [0, 2]$
$2\frac{x - l}{r - l} - 1 \in [-1, 1]$	$2\frac{y - b}{t - b} - 1 \in [-1, 1]$	$2\frac{z + n}{n - f} - 1 \in [-1, 1]$
$2\frac{x - l}{r - l} - \frac{r - l}{r - l} = x'$	$2\frac{y - b}{t - b} - \frac{t - b}{t - b} = y'$	$2\frac{z + n}{n - f} - \frac{n - f}{n - f} = z'$
$\frac{2}{r - l}x - \frac{r + l}{r - l} = x'$	$\frac{2}{t - b}y - \frac{t + b}{t - b} = y'$	$\frac{-2}{f - n}z + \frac{f + n}{f - n} = z'$

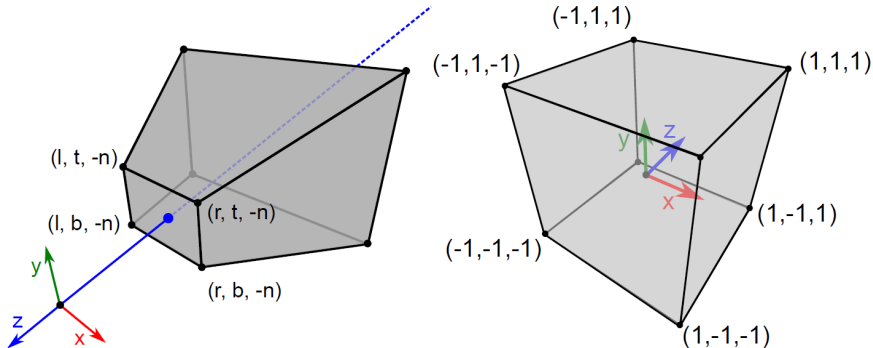
Ortogonalní projekce - finální matice

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{2}{r-l}x - \frac{r+l}{r-l} \\ \frac{2}{t-b}y - \frac{t+b}{t-b} \\ \frac{-2}{f-n}z + \frac{f+n}{f-n} \\ 1 \end{bmatrix}$$

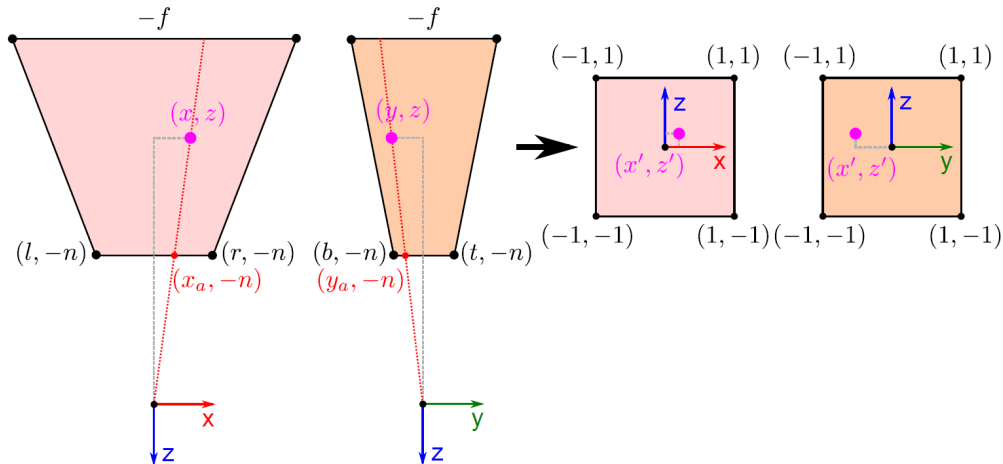
$$\begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{-2}{f-n} & \frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

Perspektivní projekce

- Jehlan má opět šest parametrů: l (left), r (right), t (top), b (bottom), n (near), f (far)
- Je umístěn relativně ke středu souřadného systému



Perspektivní projekce



Perspektivní projekce - odvození rovnic pro x' , y'

$$\frac{x_a}{-n} = \frac{x}{z}$$

$$x_a = \frac{-nx}{z}$$

$$x_a \in [l, r]$$

$$x_a - l \in [0, r - l]$$

$$\frac{x_a - l}{r - l} \in [0, 1]$$

$$2 \frac{x_a - l}{r - l} - 1 \in [-1, 1]$$

$$\frac{2}{r - l} x_a - \frac{r + l}{r - l} = x'$$

$$\frac{-2n}{r - l} \frac{x}{z} - \frac{r + l}{r - l} = x'$$

$$\frac{-2n}{r - l} \frac{x}{z} - \frac{r + l}{r - l} = x' \quad (1)$$

$$\frac{y_a}{-n} = \frac{y}{z}$$

$$y_a = \frac{-ny}{z}$$

$$y_a \in [b, t]$$

$$y_a - b \in [0, t - b]$$

$$\frac{y_a - b}{t - b} \in [0, 1]$$

$$2 \frac{y_a - b}{t - b} - 1 \in [-1, 1]$$

$$\frac{2}{t - b} y_a - \frac{t + b}{t - b} = y'$$

$$\frac{-2n}{t - b} \frac{y}{z} - \frac{t + b}{t - b} = y'$$

$$\frac{-2n}{t - b} \frac{y}{z} - \frac{t + b}{t - b} = y' \quad (2)$$

Perspektivní projekce - problém se z souřadnicí

$$z \in [-n, -f]$$

$$z + n \in [0, n - f]$$

$$\frac{z + n}{n - f} \in [0, 1]$$

$$2 \frac{z + n}{n - f} - 1 \in [-1, 1]$$

$$\frac{2}{n - f} z + \frac{n + f}{n - f} = z'$$

$$\frac{2}{n - f} z + \frac{n + f}{n - f} = z' \quad (3)$$

$$\frac{1}{z} \in \left[-\frac{1}{n}, -\frac{1}{f}\right]$$

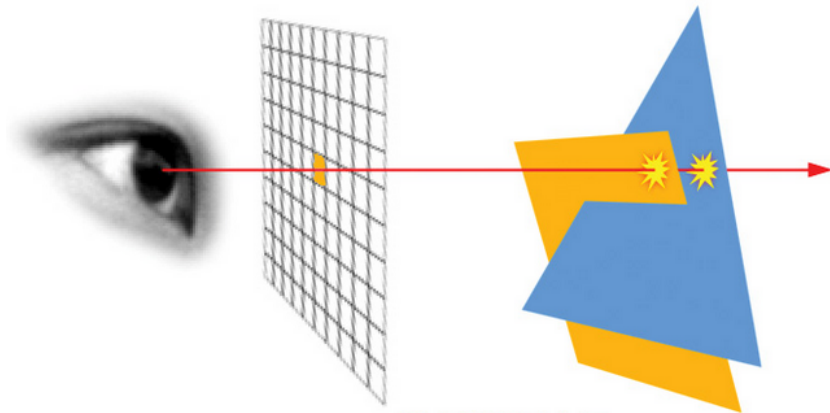
$$\frac{1}{z} + \frac{1}{n} \in \left[0, \frac{f - n}{nf}\right]$$

$$\frac{nf}{f - n} \frac{1}{z} + \frac{f}{f - n} \in [0, 1]$$

$$\frac{2nf}{f - n} \frac{1}{z} + \frac{2f}{f - n} - 1 \in [-1, 1]$$

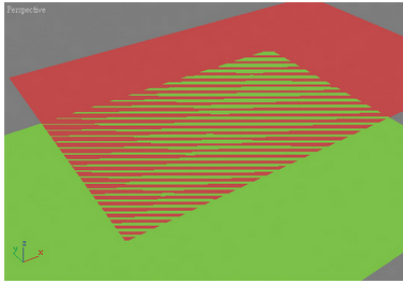
$$\frac{2nf}{f - n} \frac{1}{z} + \frac{f + n}{f - n} = z' \quad (4)$$

Z-fighting - při zobrazování potřebujeme rozlišit hloubku



Z-fighting

- Rozsah near a far plane ($-n$, $-f$) je mapován na *bitový rozsah* float čísla = přesnost je omezená.
- Pro velmi blízké plochy nelze rozlišit hloubku!



- Mapování není lineární (viz dělení z) - větší přesnost blíže ke kameře.

Perspektivní projekce - prohození z a w souřadnice

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \\ z \end{bmatrix}$$

$$\begin{bmatrix} -\frac{2n}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & -\frac{2n}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{2nf}{f-n} & \frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \\ z \end{bmatrix} = \begin{bmatrix} x' \cdot z \\ y' \cdot z \\ z' \cdot z \\ z \end{bmatrix}$$

Perspektivní projekce - finální matice

$$\begin{bmatrix} -\frac{2n}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & -\frac{2n}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{2nf}{f-n} & \frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} -\frac{2n}{r-l} & 0 & -\frac{r+l}{r-l} & 0 \\ 0 & -\frac{2n}{t-b} & -\frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{f+n}{f-n} & \frac{2nf}{f-n} \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2nf}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} -x' \cdot z \\ -y' \cdot z \\ -z' \cdot z \\ -z \end{bmatrix}$$