

Základy počítačovej grafiky a spracovania obrazu

Rasterizácia, Alias, Anti-alias

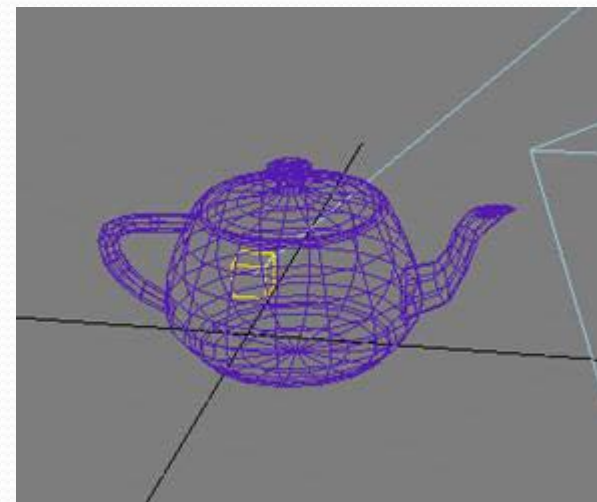
Viditeľnosť

Materiály, Textúry

Júlia Kučerová

Zobrazovací kanál

- Modelové transformácie
 - Lokálne → globálne
- Pohľadové transformácie
 - Globálne → kamerové
- Projekčné transformácie
 - Kamerové (3D) → obrazovka (2D)
- Orezávanie, Rasterizácia, Textúrovanie & Osvetlenie

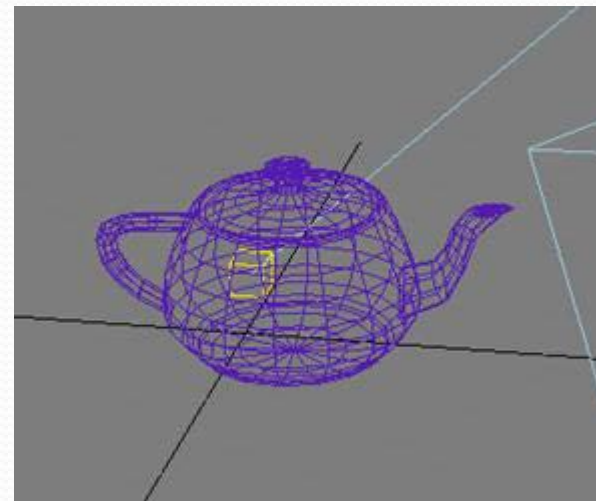


Transformácie

- lokálne \rightarrow globálne
 - posunutie, rotácia, škálovanie, posunutie
- globálne \rightarrow kamerové
 - posunutie, rotácia, rotácia, projekcia
- kamerové \rightarrow obrazovka
 - posunutie, škálovanie, posunutie
- kombinácia transformácií = násobenie matíc

Zobrazovací kanál

- Modelové transformácie
 - Lokálne → globálne
- Pohľadové transformácie
 - Globálne → kamerové
- Projekčné transformácie
 - Kamerové (3D) → obrazovka (2D)
- Orezávanie, **Rasterizácia**,
Textúrovanie & Osvetlenie

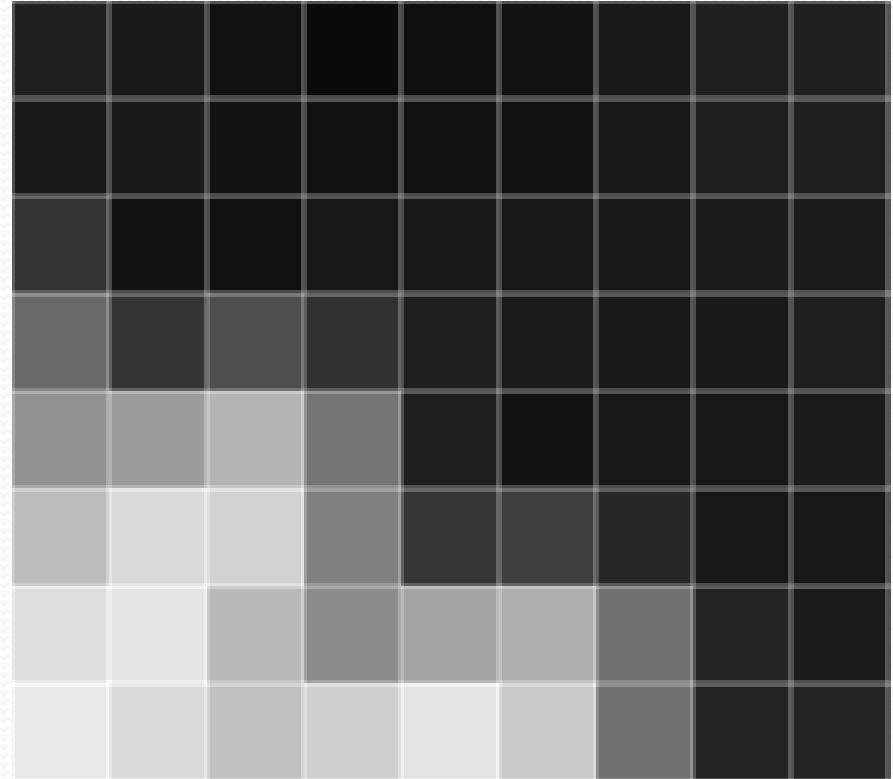




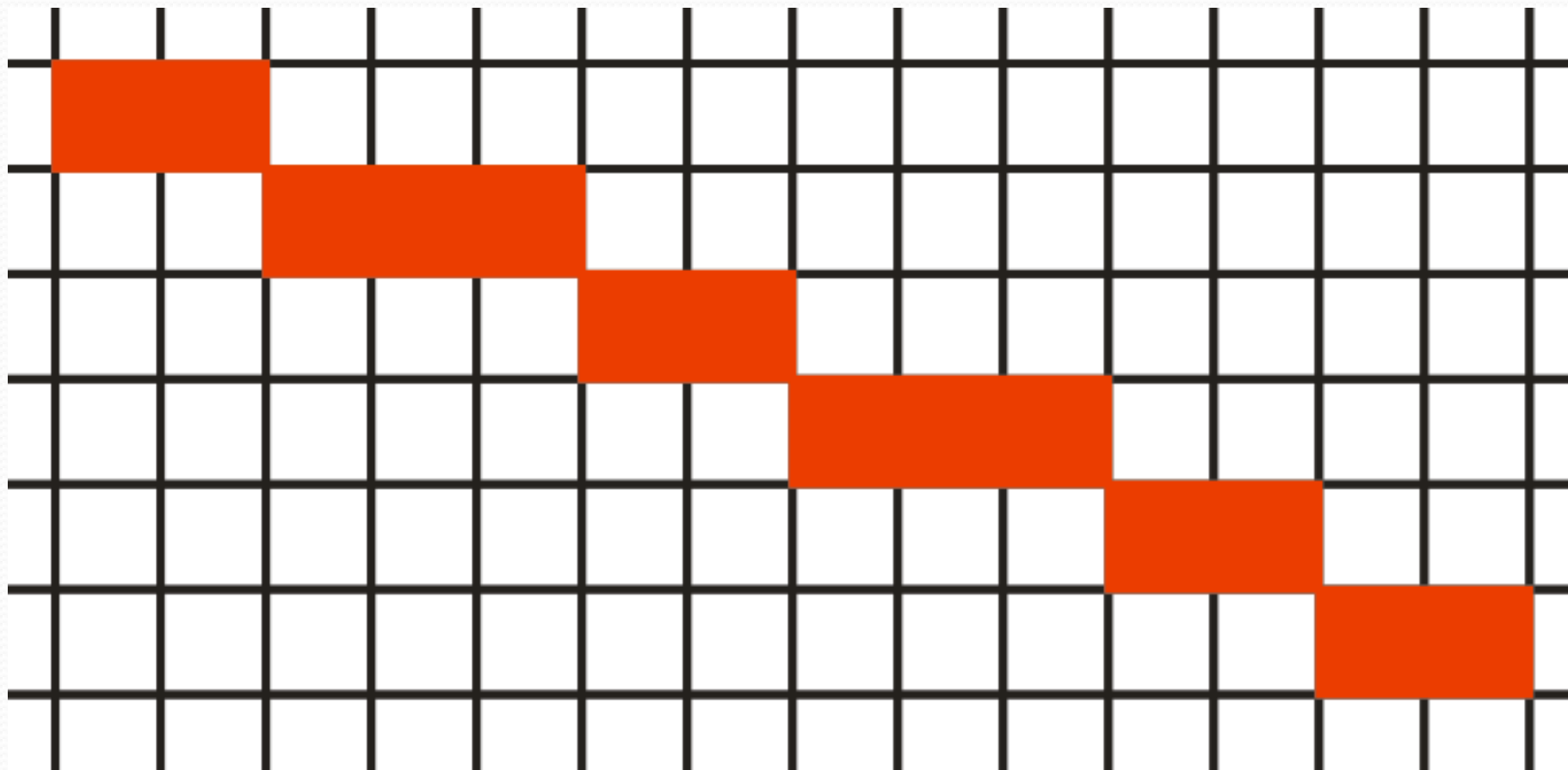
Rasterizácia

Všeobecný problém

- Daná spojitá geometrická reprezentácia objektu
- Rozhodnutie, ktorý pixel patrí objektu

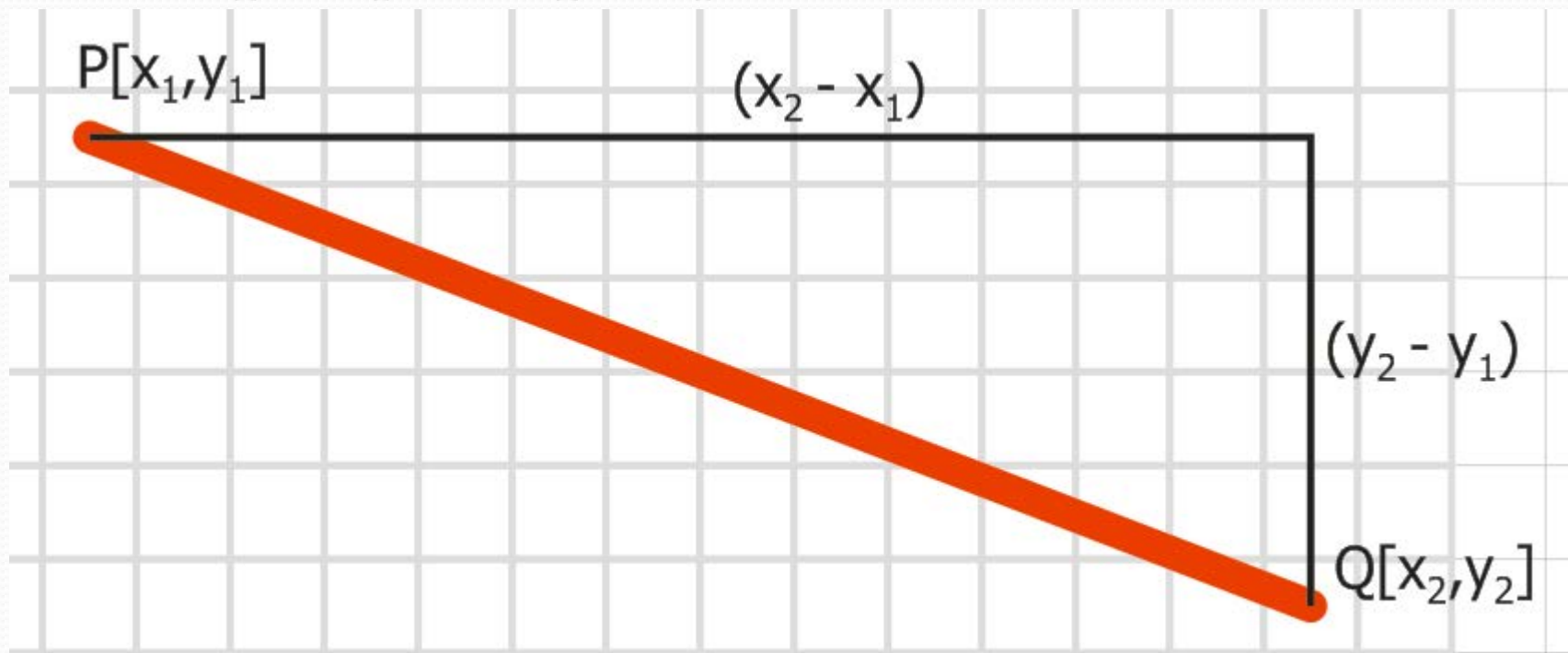


Rasterizácia úsečky



DDA

- Digital Differential Analyzer
- $m = (y_2 - y_1) / (x_2 - x_1)$ (smernica, desatinnné číslo)

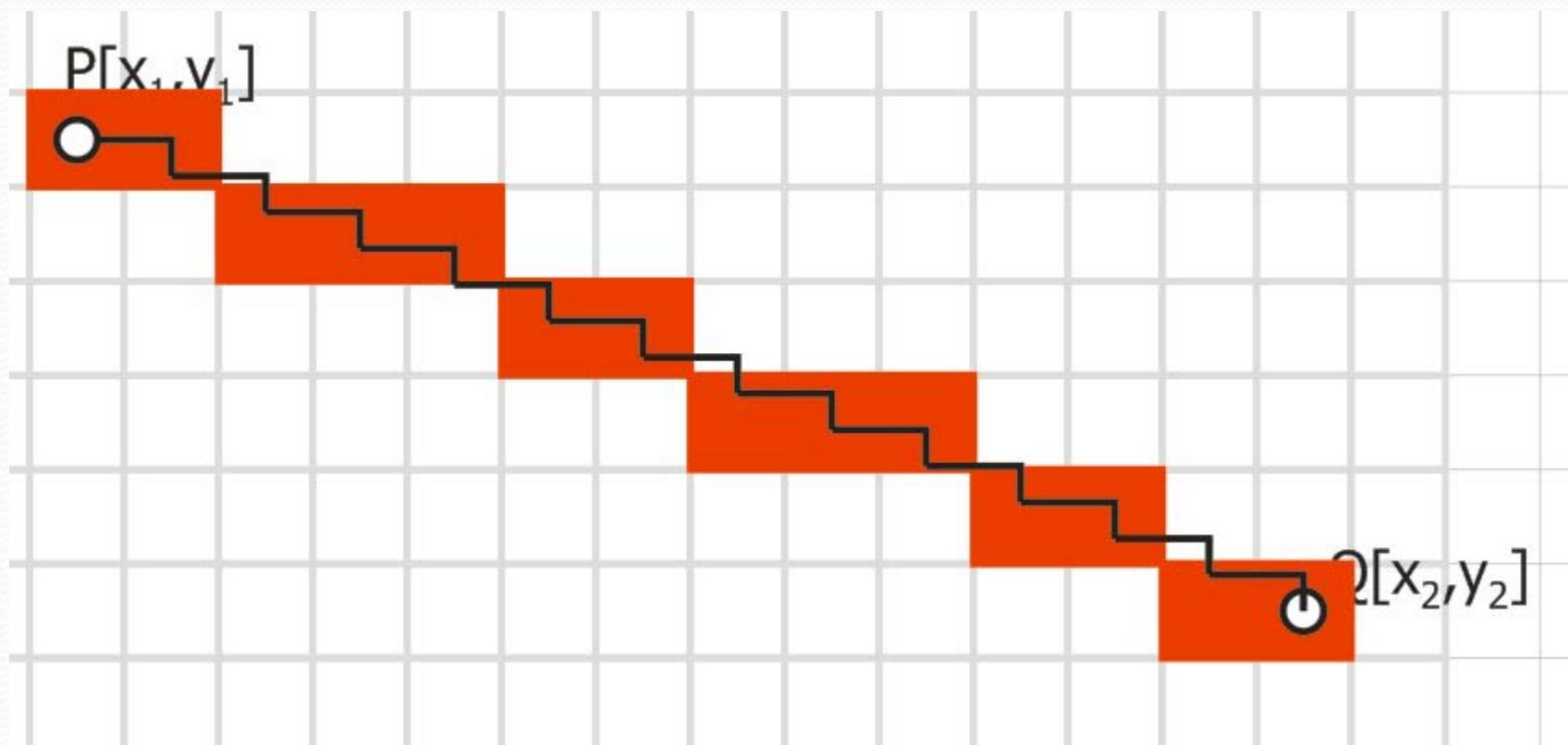


DDA

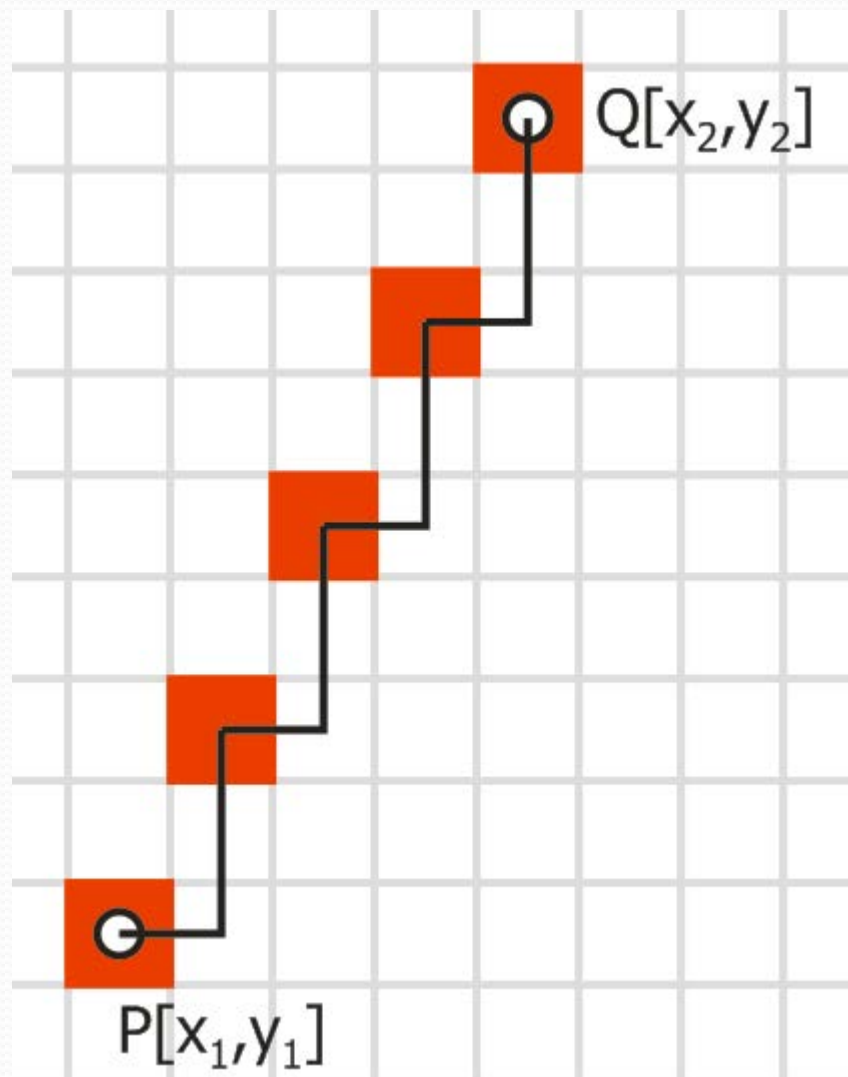
- **Algoritmus:**

1. Z koncových bodov $[x_1, y_1]$ a $[x_2, y_2]$ urči smernicu m
2. Inicializuj bod $[x, y]$ hodnotou $[x_1, y_1]$
3. Pokiaľ $x \leq x_2$ opakuj:
 - a) Vykresli bod $[x, \text{zaokrúhlené}(y)]$
 - b) $x = x+1$
 - c) $y = y+m$

DDA

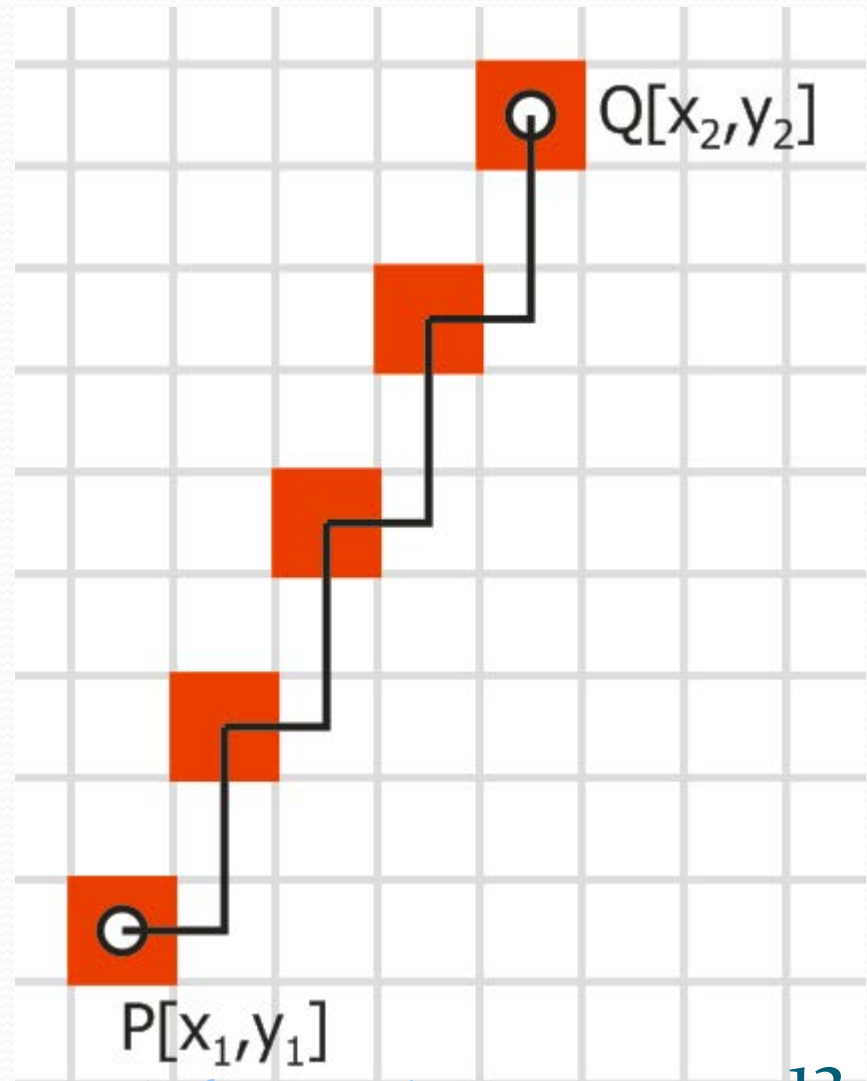
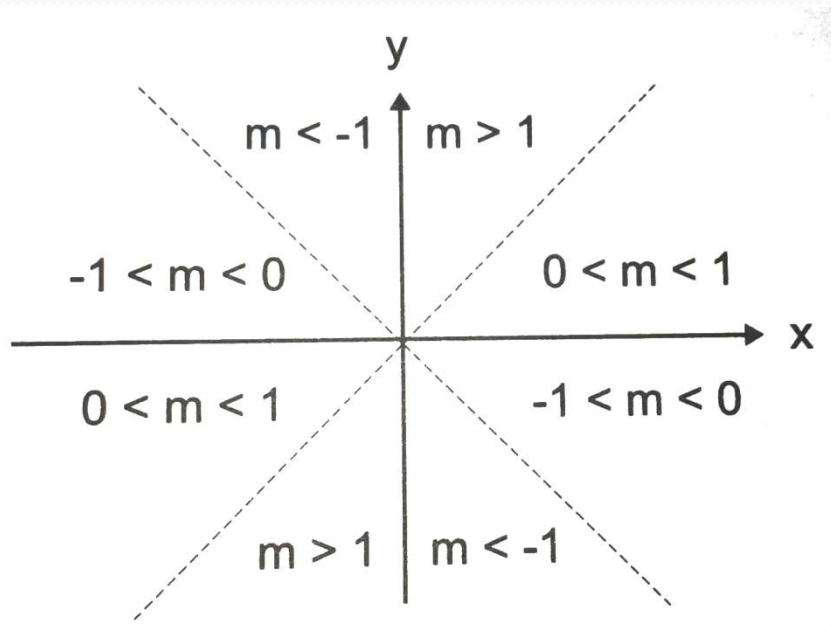


DDA



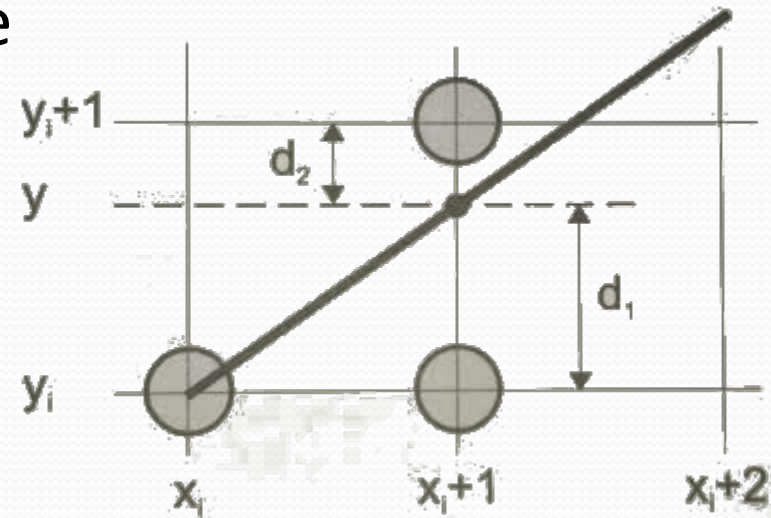
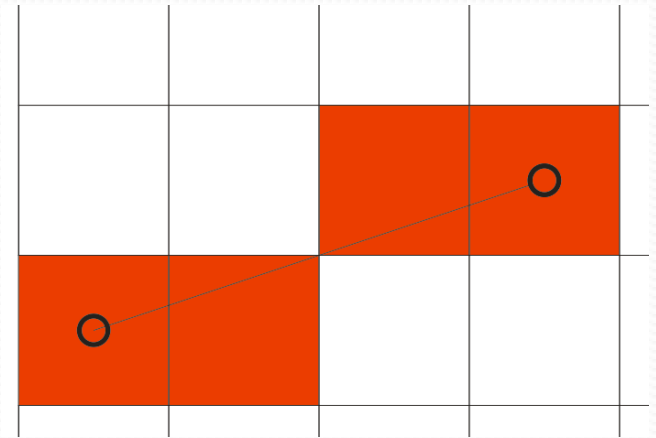
DDA

- ak $\text{abs}(m) > 1$
 - zameň $x \leftrightarrow y$ v algoritme



Bresenhamov algoritmus

- DDA – desatinné čísla
- Bresenham – celé čísla
- Hlavná myšlienka:
 - Pre každé x existujú iba 2 možné hodnoty y , kde sa vyberie hodnota s menšou chybou



Bresenhamov algoritmus

- Algoritmus:

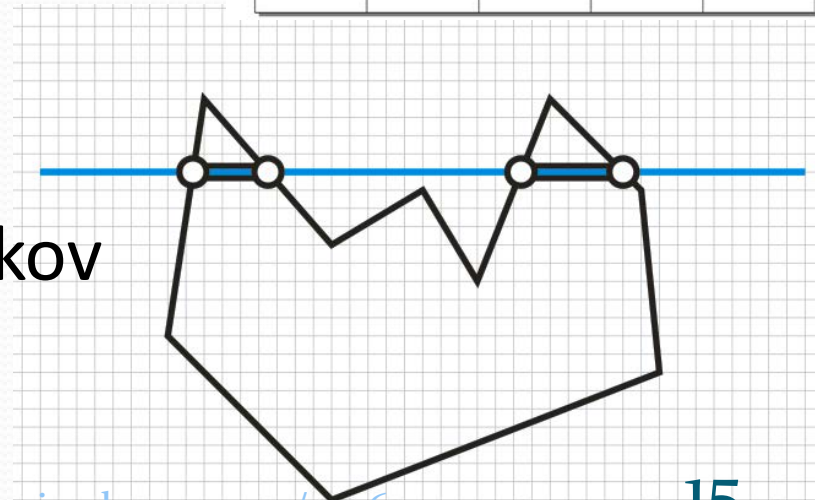
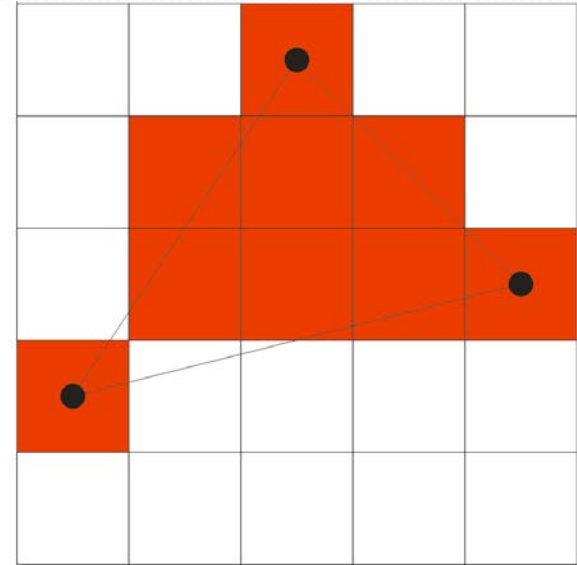
1. Z koncových bodov $[x_1, y_1]$ a $[x_2, y_2]$ urči konštanty $k_1 = 2\Delta y$, $k_2 = 2(\Delta y - \Delta x)$
2. Inicializuj rozhodovací člen p na hodnotu $2\Delta y - \Delta x$
3. Inicializuj $[x, y]$ ako $[x_1, y_1]$
4. Vykresli bod $[x, y]$
5. Pokiaľ $x \leq x_2$ opakuj:
 - a) $x = x + 1$
 - b) ak je p kladné $y = y + 1$ a $p = p + k_2$
 - c) ak p nie je kladné, tak $p = p + k_1$
 - d) vykresli bod $[x, y]$

Rasterizácia polygónov

- **Scaneline**

- Pre každú čiaru:

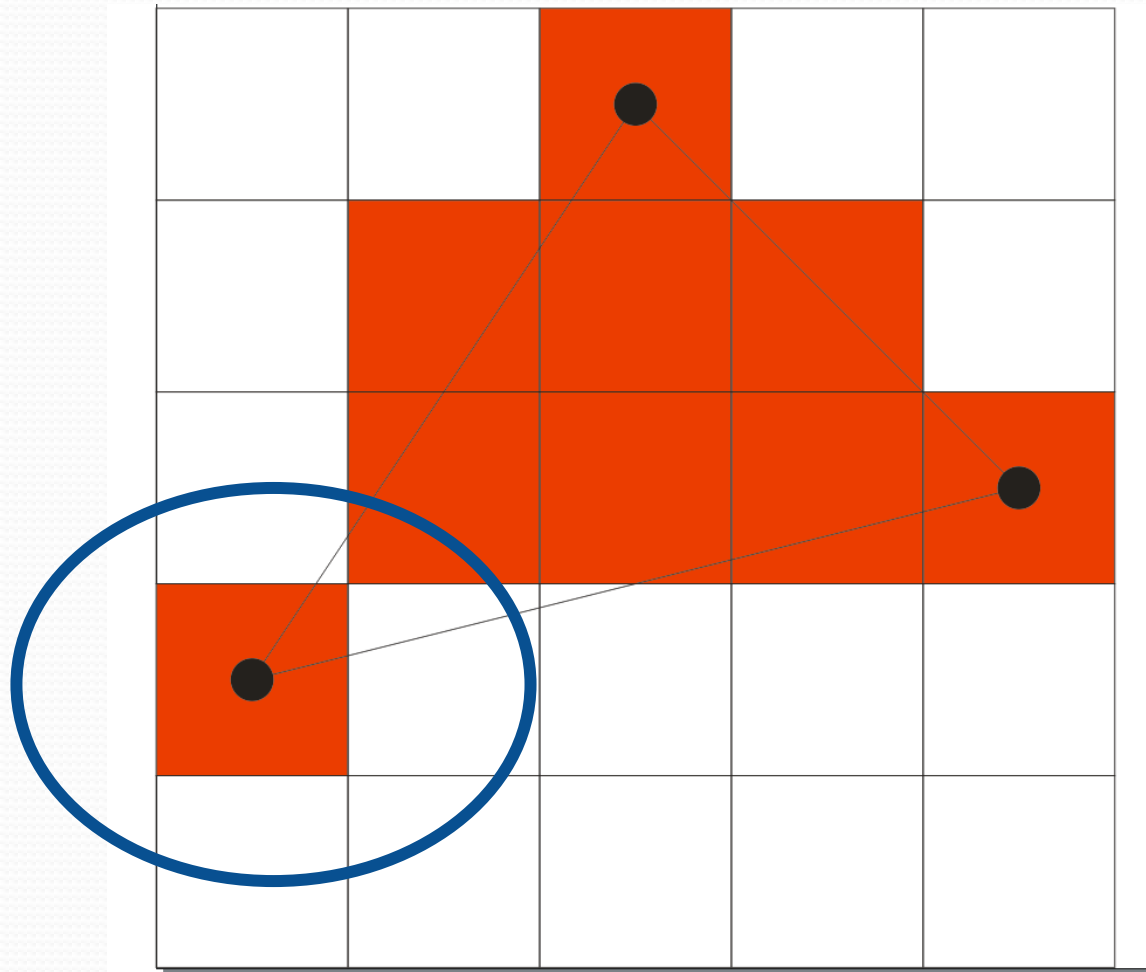
- 1) Nájdi prienik polygónu s čiarou
- 2) Utried' prieniky podľa x-ovej súradnice
- 3) Vyplň pixle medzi nasledujúcimi párami prienikov





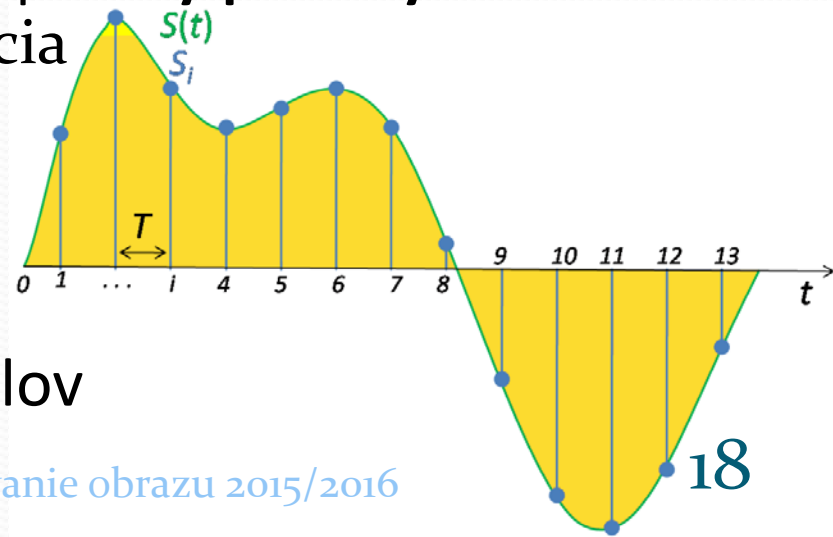
Alias, anti-alias

Alias



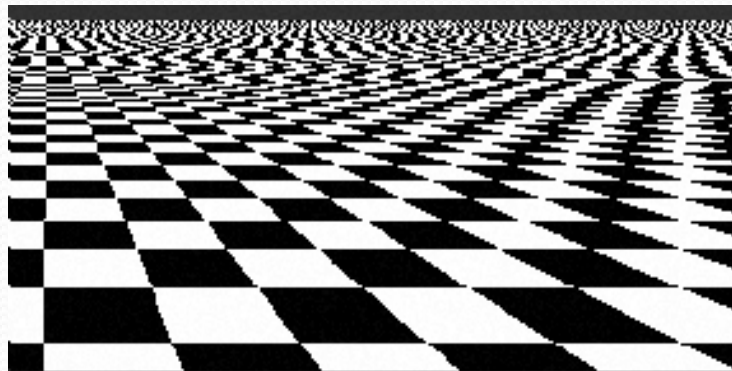
Alias

- Nežiadúci jav
spojité \rightarrow diskkrétne, môžu sa objaviť artefakty
- Vznik
 - ak je pôvodná funkcia frekvenčne neobmedzená (neexistuje žiadna maximálna frekvencia)
 - rekonštrukcia signálu vzorkovaného pod **Nyquistovým limitom** ($2f_{\max}$), f_{\max} je maximálna frekvencia
- vzorkovanie
 - Pozorovania spojitej funkcie v diskrétnych intervaloch
- vzorkovacia frekvencia - hustota pixlov



Formy aliasu

- priestorový alias
 - Zubaté hrany/zubatice (jaggy edges)
 - moiré (pi vzorkovaní textúry s pravidelným vzorom)
 - Skreslenie textúry
- časový
 - obrazovka snímaná kamerou
 - “wagon wheel”



Anti-aliasing

- Odstránenie / zmenšenie aliasu
 - Alias zvyčajne nemôžeme úplne odstrániť, môžeme ho čiastočne potlačiť
- najjednoduchšie:
 - Odstránenie vysokých frekvencií
- Anti-aliasing
 - Posunutie aliasu k vyšším hodnotám (vzorkovanie)
 - Prevedenie aliasu na šum (náhodné čísla)

Anti-aliasing

- všeobecný (globálny) anti-aliasing -
nadvzorkovanie (supersampling)
 - pravidelné
 - stochastické (jittering)
- Objektový (lokálny) anti-aliasing
 - čiarový anti-aliasing
 - anti-aliasing siluety
 - textúrny anti-aliasing

Anti-aliasing

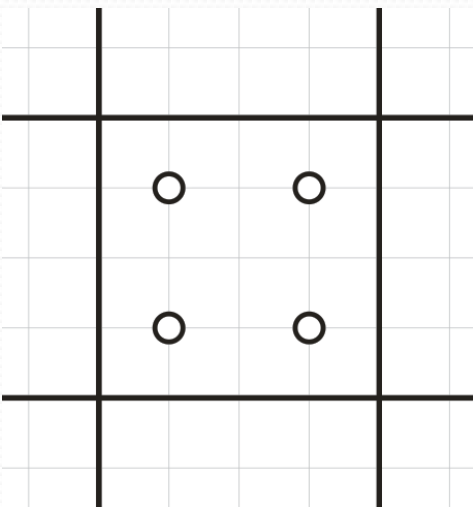
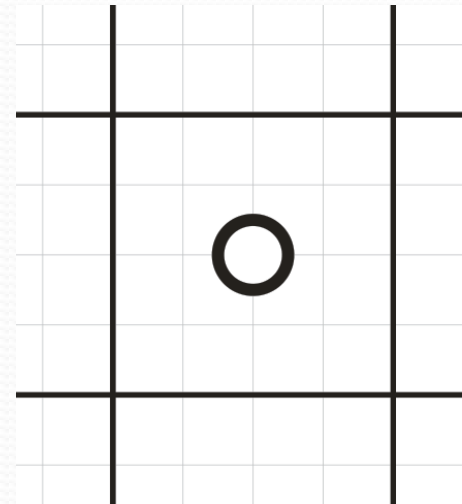
- Supersampling
- všeobecný (globálny) anti-aliasing
- vzorkovanie s viac ako jednou vzorkou na pixel
 - Previdelné
 - FSAA (Full Screen AntiAliasing)
 - Stochstické
 - Poisson disc
 - Roztrasenie (jittering)

Anti-aliasing

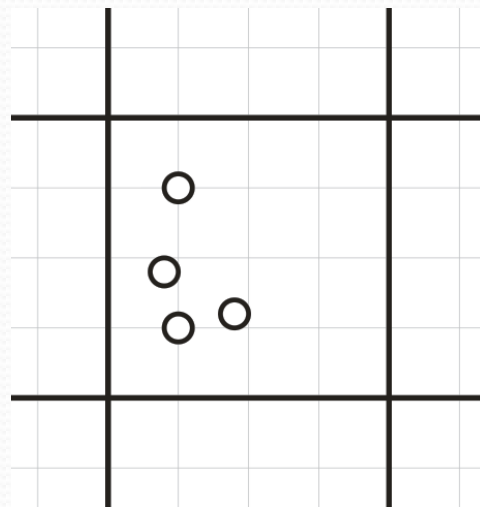
- Poisson disc
 - Na základe rozloženia tyčíniek a čapíkov v ľudskom oku
 - Výpočtovo náročné
- Roztrasenie (jittering)
 - Najčastejšie používaný stochastický anti-aliasing
 - Aproximácia optimálneho rozloženia voriek
 - Algoritmus: generovanie vzoriek do superpixlov a ich umiestnenie do stredu (rovnako ako pri pravidelnom vzorkovaní s vyššou frekvenciou). Vzorky sa náhodne posunú z okolia stredu pixlu, tak, aby žiaden neopustil hranicu svojho superpixla
 - Pridanie šumu do pravidelného vzorkovania

Super-sampling

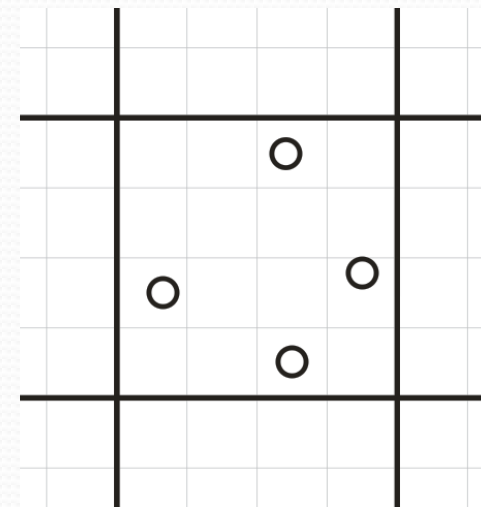
- Pre každý pixel vytvor viacero sub-pixlov a skombinuj ich



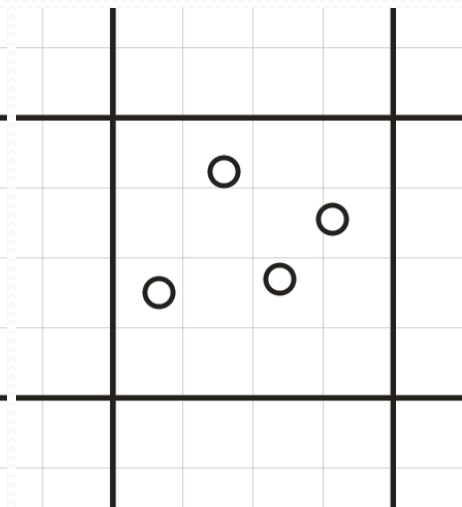
Pravidelné
(mriežka)



Náhodné
(stochastické)



Poisson



Jitter



Viditeľnosť

Viditeľnosť

- projekcia 3D objektov do 2D je nedostatočná
- Musíme brať do úvahy aj hĺbku (tretie D)



Viditeľnosť v rôznych metódach

- lokálne → svetové → kamerové
- Každý objekt je spracovávaný samostatne (lokálne)
 - potrebujeme spracovávať hĺbku samostatne
 - jeden objekt má veľa samostatných plôch

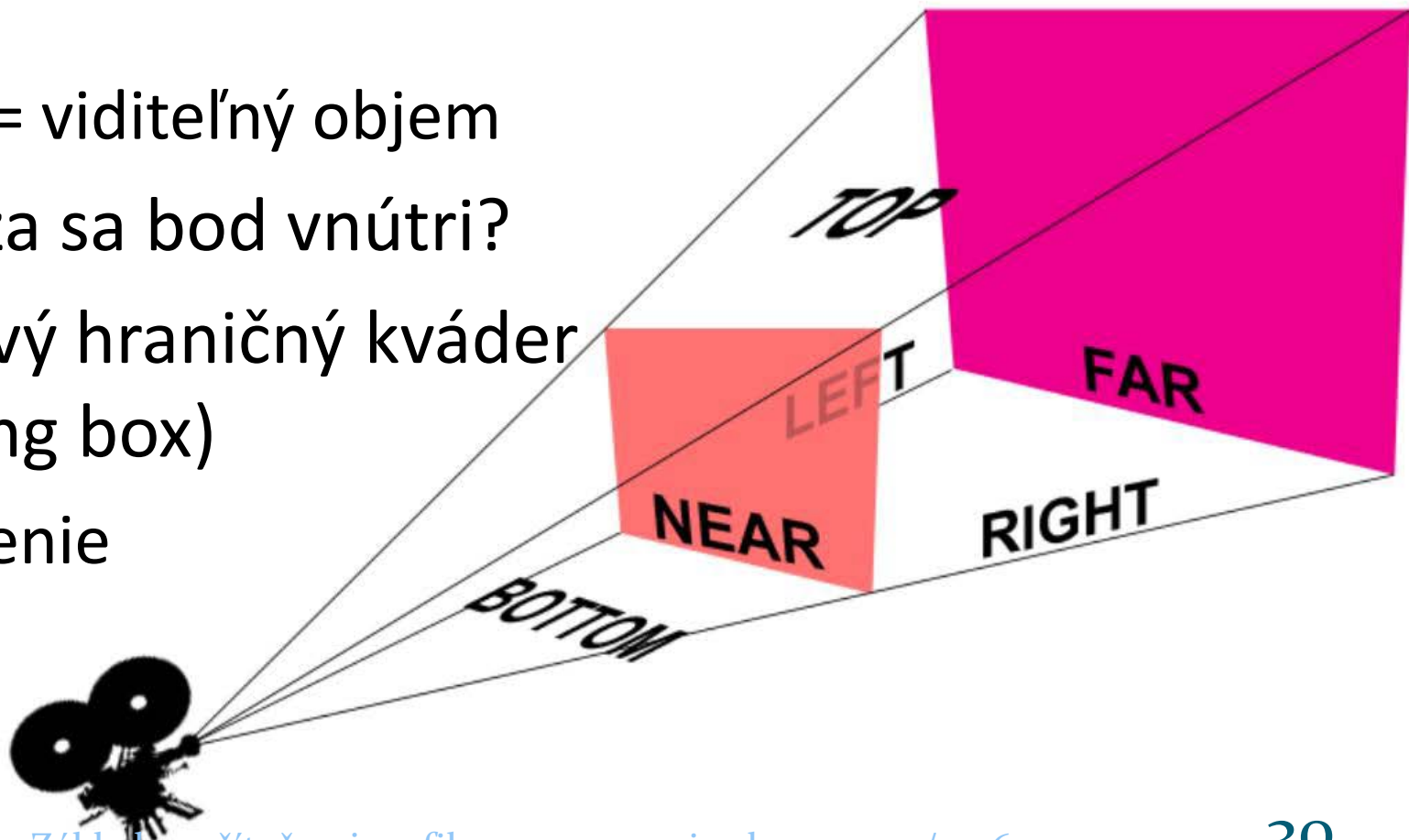


Optimalizovanie viditeľnosti

- Zbaviť sa všetkých objektov, ktoré určite nevidíme
- Orezanie viditeľného objemu (Frustum culling)
- Orezávanie zadných stien (Backface culling)
- Occlusion culling
- Portal culling

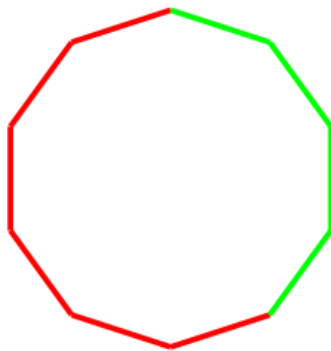
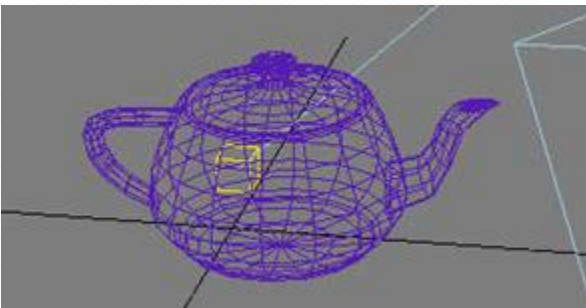
Orezanie viditeľného objemu (Frustum culling)

- 6 plôch
 - Vnútri= viditeľný objem
- Nachádza sa bod vnútri?
- Objektový hraničný kváder (bounding box)
 - Zrýchlenie



Backface culling

- Ktoré steny objektu sú viditeľné?
- Normálový vektor (orientácia steny objektu)



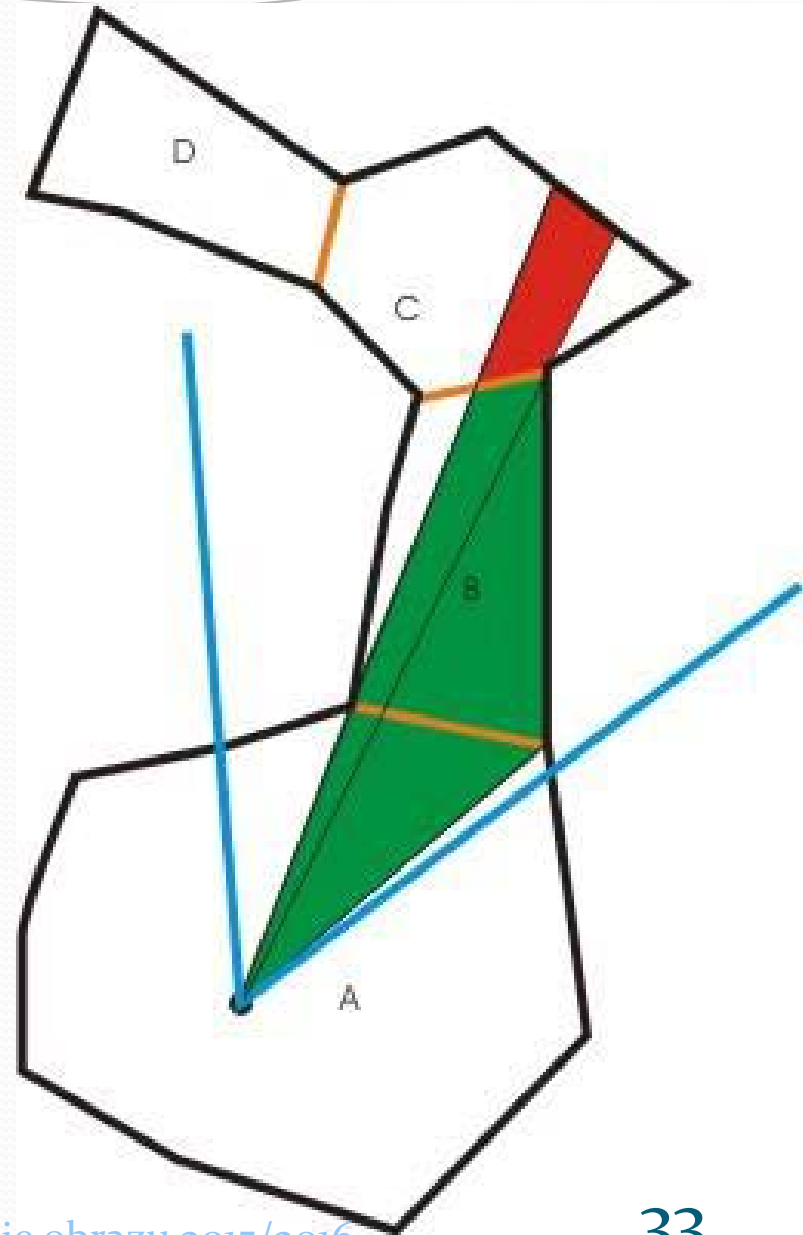
Occlusion culling

- Niektoré objekty sú plne zakryté inými
- Priestorové vzťahy medzi objektami
- Renderovanie v reálnom čase



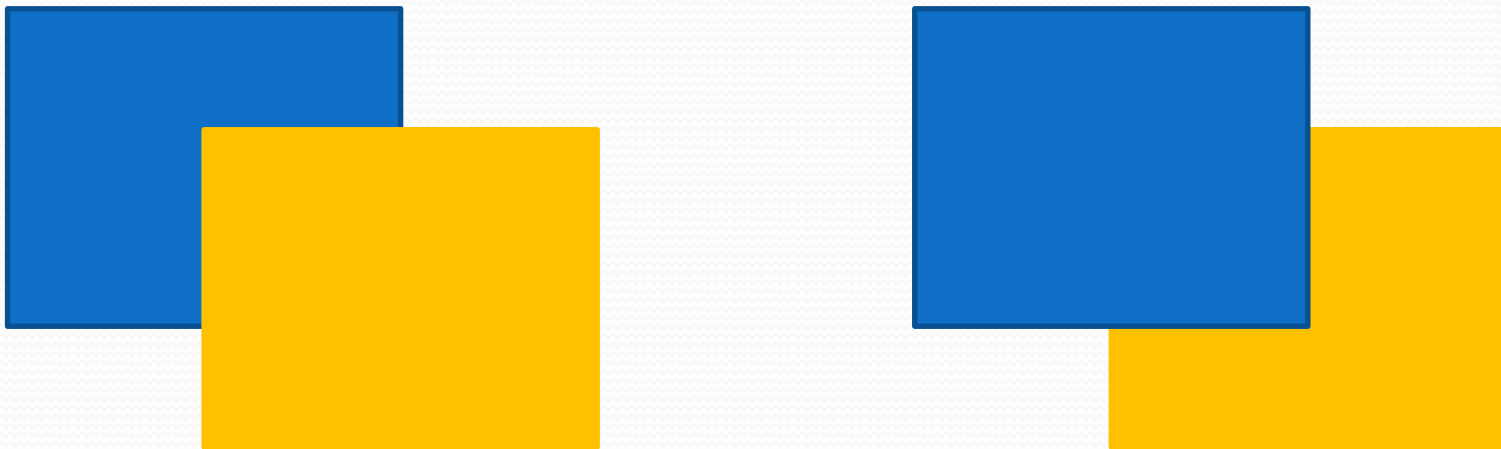
Portal culling

- Niektoré časti scény nie sú viditeľné z iných častí scény



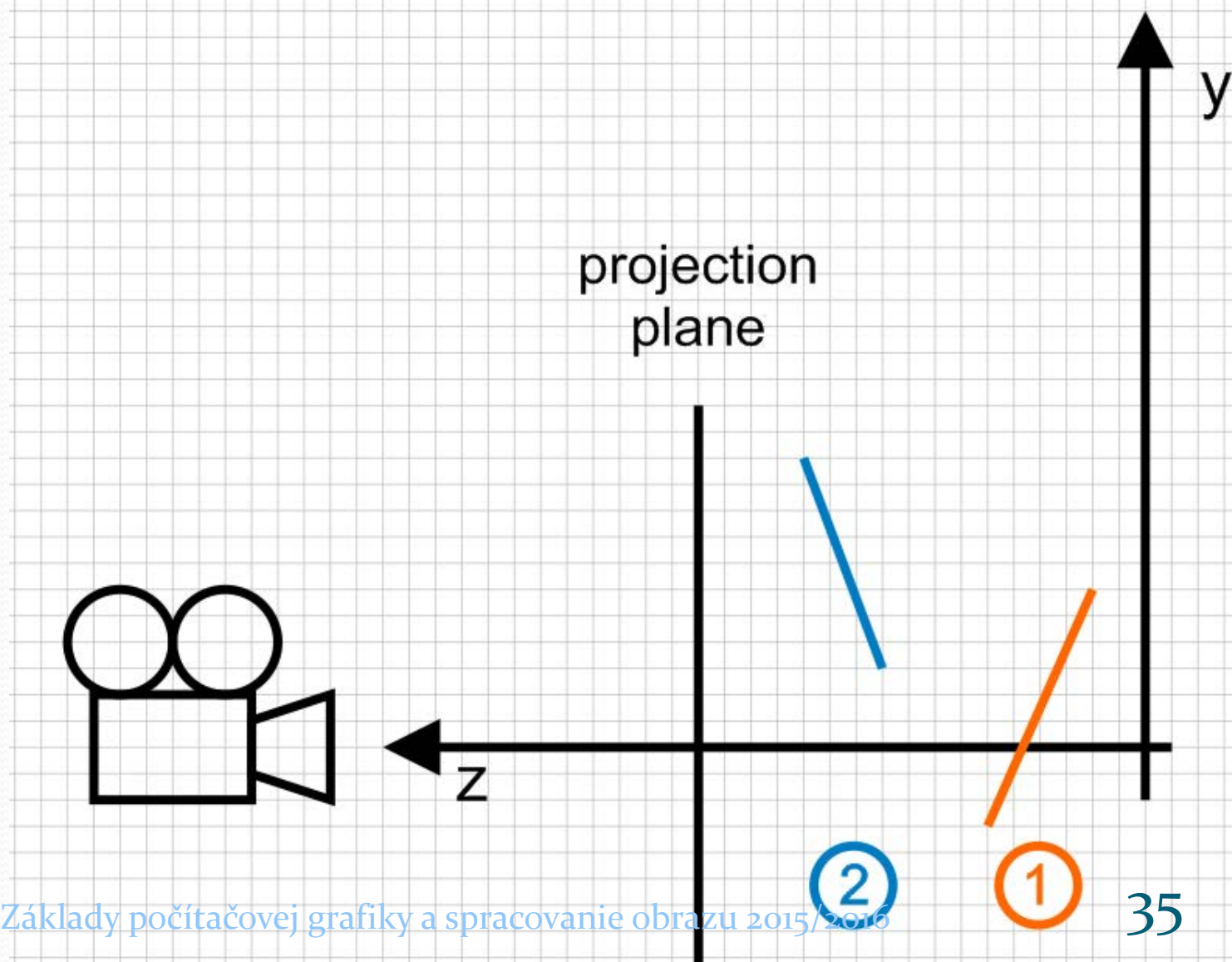
Viaceré objekty

- Uvažujme polygonálne objekty => zredukovanie viditeľnosti objektov na viditeľnosť stien



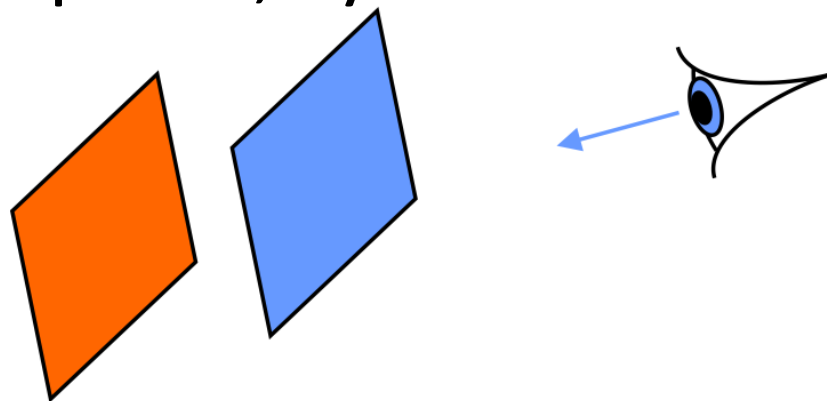
- Hlavná myšlienka – uvažujeme hĺbku objektu (z)

Svet -> Kamera

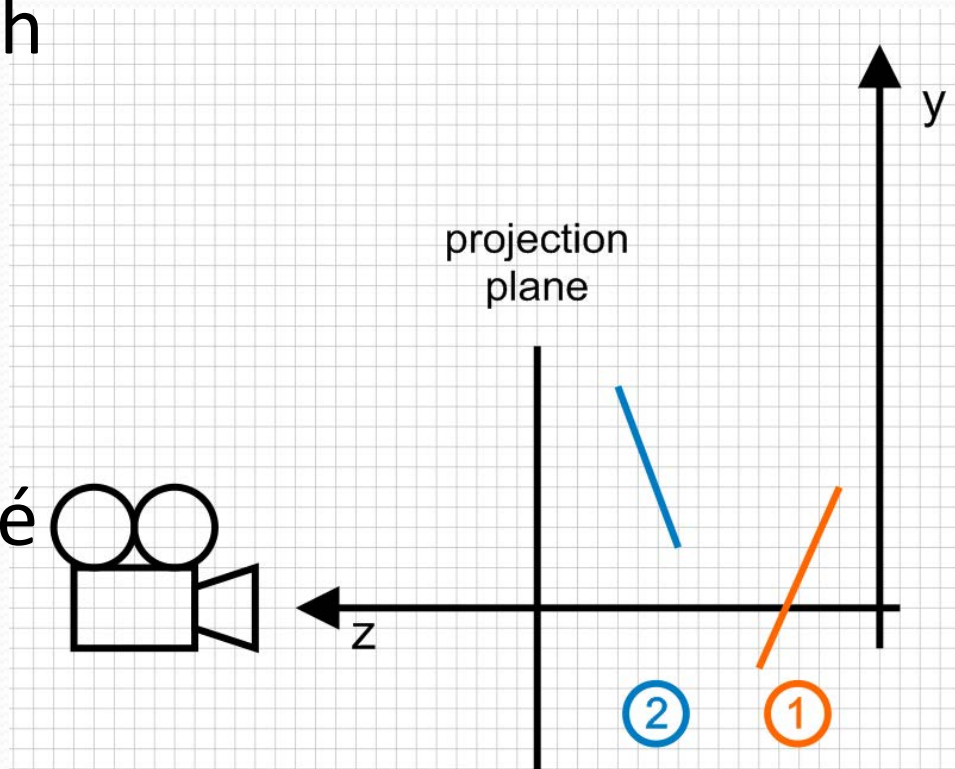
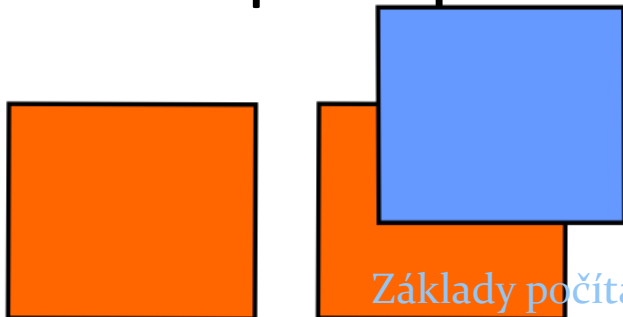


Maliarov algoritmus

- Usporiadať steny objektu do poradia zozadu-dopredu, vyrenderovať ich

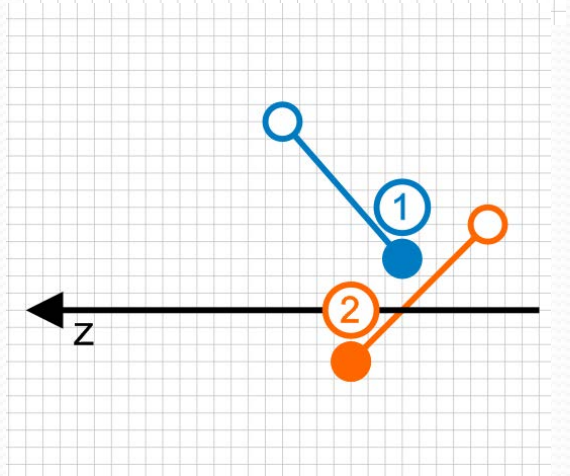
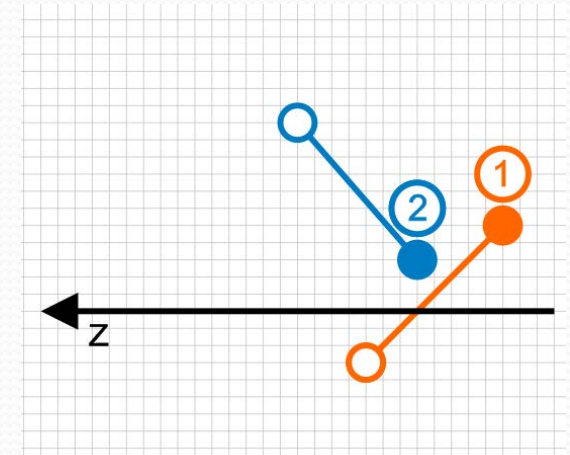
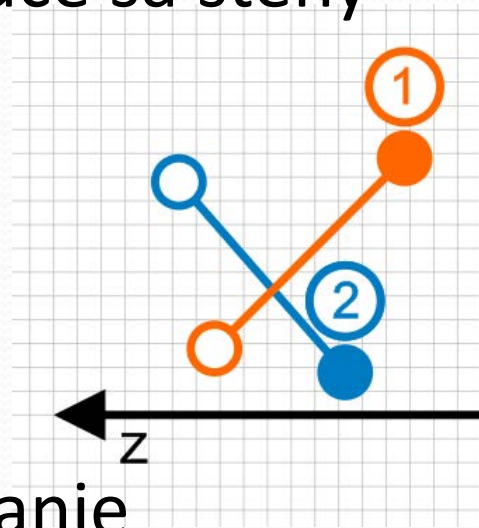
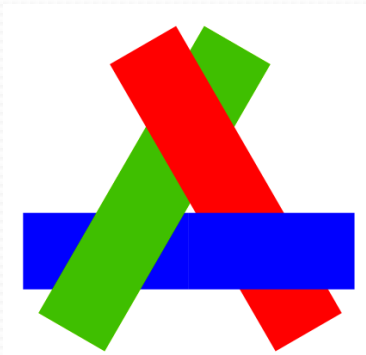


- Nové pixle prekreslia staré



Maliarov algoritmus – problém

- Ktorý pixel reprezentuje stenu?
- Prienik plôch
- Cyklicky sa prekrývajúce sa steny



- prebytočné renderovanie
- Vhodný pre jednoduché scény

Iné algoritmy

- **Warnockov algorimus**
 - Prerozdelenie scény pomocou quadtree kým nie je celá bunka vnútri jedného polygónu
- **Obrátený maliarov algoritmus**
 - Vykresľovať spredu-dozaďu a vymaľovávať len prázdne oblasti
- **Z-buffer**
 - Zapamätá si z-ovú hodnotu pre každý pixel a vykreslí iba, keď je nová hodnota vyššia

Z-buffer (Pamäť hĺbky)

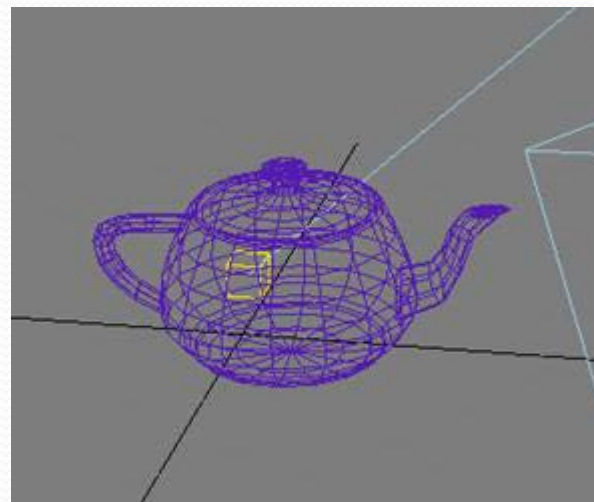
- Pracuje v priestore obrazovky
- obrazovka $\text{š} \times \text{v} \leftrightarrow \text{z-buffer } \text{š} \times \text{v}$
- Algoritmus:
 1. vyplň obrazovú pamäť farbou pozadia
 2. Vyplň pamäť hĺbky hodnotu $-\infty$
 3. Pre každú plochu nájdi pixely, do ktorých sa plocha premietne. Pre každý pixel $[x_i, y_i]$ urči hĺbku z_i
 4. Ak má z väčšiu hodnotu ako $[x_i, y_i]$ v pamäti hĺbky, tak
 - a) zafarbi pixel $[x_i, y_i]$ v obrazovej pamäti farbou tejto plochy
 - b) položku $[x_i, y_i]$ v pamäti hĺbky aktualizuj na hodnotu z_i

Z-buffer – výhody a nevýhody

- GPU
- Testovanie pred osvetľovaním/tieňovaním ušetrí nadbytočnú prácu
- Môžu sa vyskytnúť nepresnosti
- Pamäťová náročnosť (šírka x výška x presnosť)
- Môže byť zrýchlený použitím scanline algoritmu (šírka x 1 x presnosť)

Zobrazovací kanál

- Modelové transformácie
 - Lokálne → globálne
- Pohľadové transformácie
 - Globálne → kamerové
- Projekčné transformácie
 - Kamerové → obrazovka
- Orezávanie, Rasterizácia,
Textúrovanie & Osvetlenie





Textúry a mapovanie textúr

Materiály

- Vizuálne rozlíšia 2 objekty s identickou geometriou
- Zatiaľ sa zameriame na farbu objektu
- Iné materiály, vlastnosti preberieme neskôr



Farba objektu

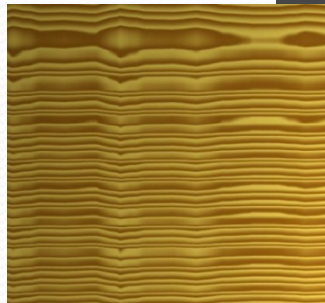
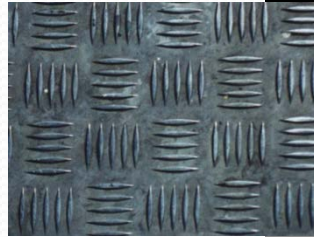
- Triviálny prípad
 - Konštantná farba

- Zvyčajný prípad
 - Farba sa mení



Textúra

- Používa sa na definovanie zmien vo farbe objektu
- 2D bitmapa
- 3D bitmapa
- Texel
- Procedurálne textúry



Rozdelenie textúr

- Podľa vlastnosti
 - Farba povrchu, odraz svetla, zmena normálového vektora, drsnosť, priehľadnosť
- Podľa rozmeru
 - 1D, 2D, 3D (objemové), 4D (animácia 3D textúr)
- Podľa reprezentácie
 - Tabuľkové, procedurálne

Mapovanie textúr

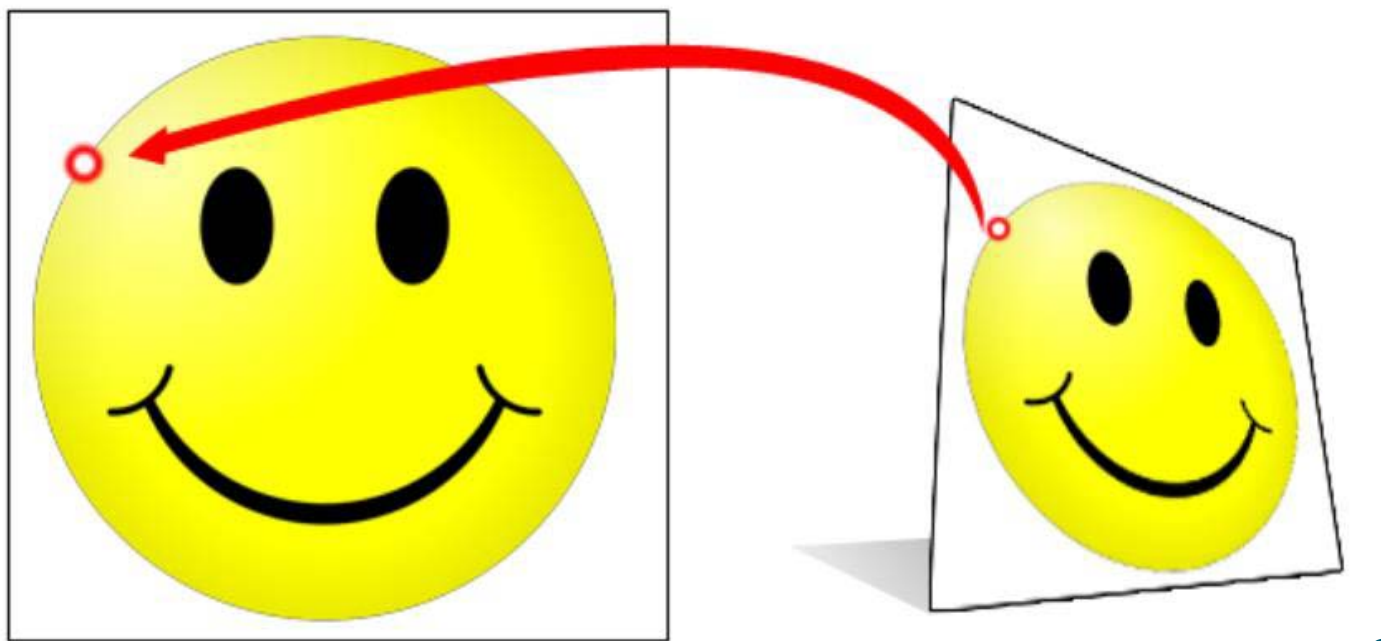
- Nanášanie textúry na povrch objektov
- Obobjektový priestor \leftrightarrow 2D textúrny priestor



- Nový súradnicový priestor: Textúrne súradnice

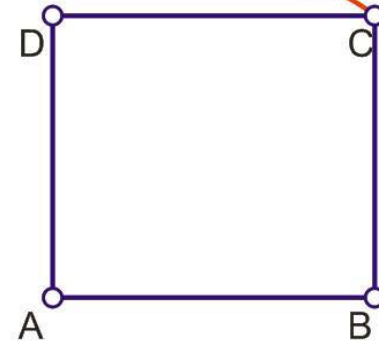
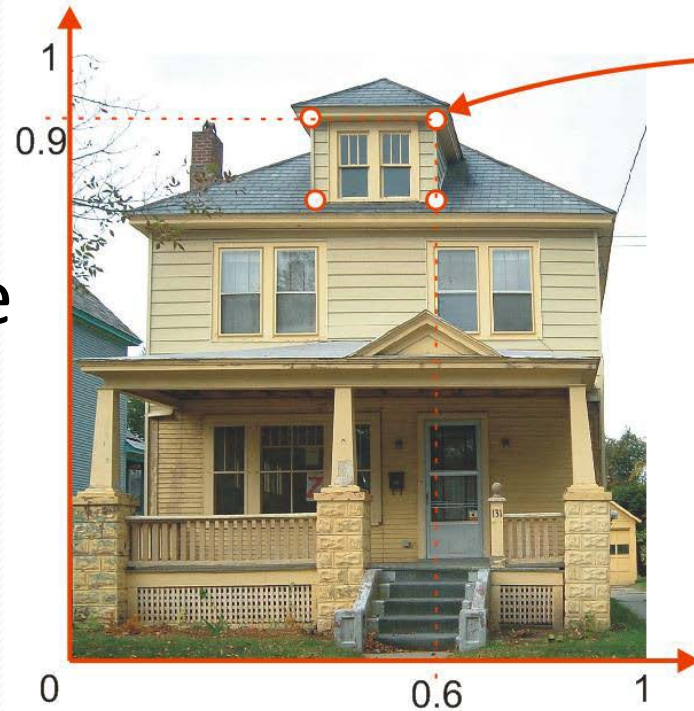
Mapovanie textúr na polygóny

- Všeobecný problém:
 - Ktorý textúrny pixel zobrazíme ako obrazový pixel na texturovanom polygóne?



Textúrne súradnice

- Pozícia kontrolných bodov polygónu v textúrnom priestore
- Textúrne súradnice pre C sú $[0.6, 0.9]$

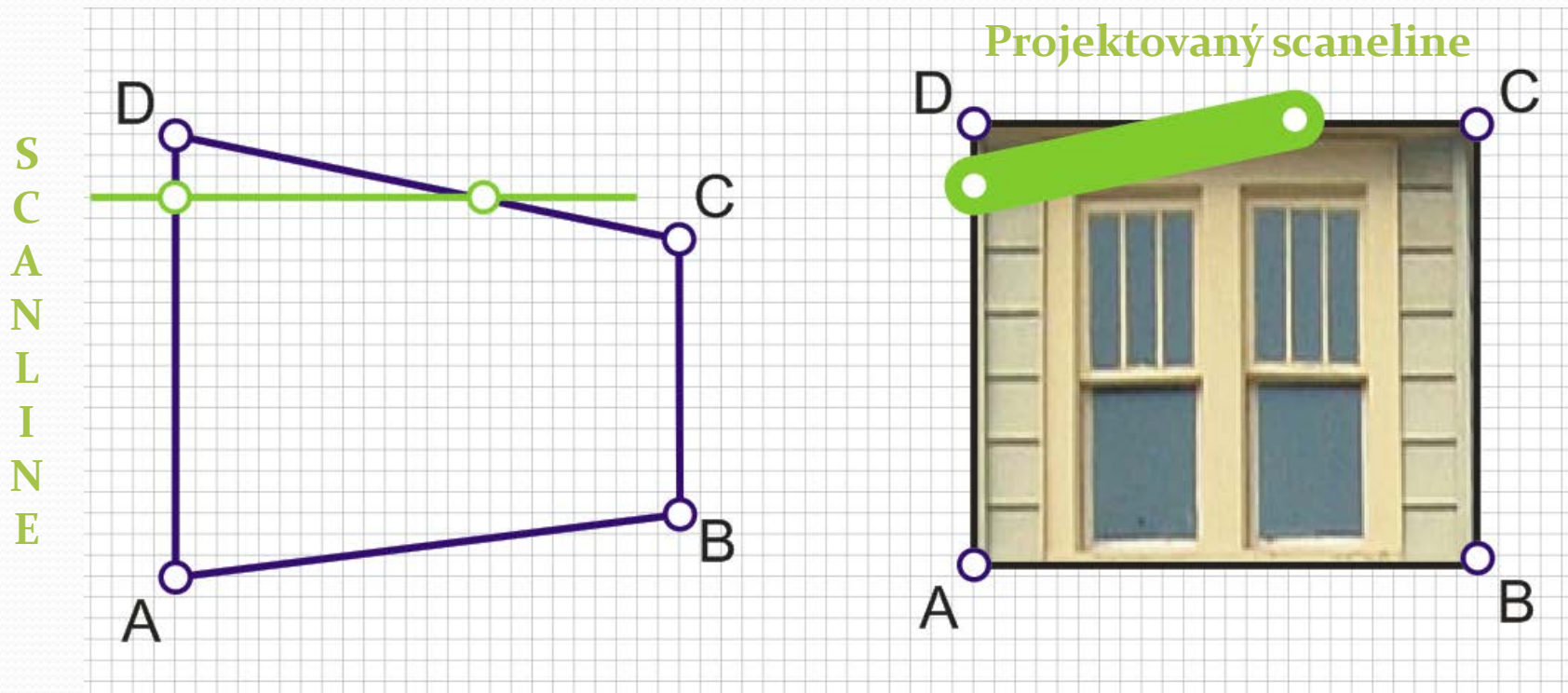


Textured polygon:



Určenie pixlov textúry

- Rasterizácia polygónu

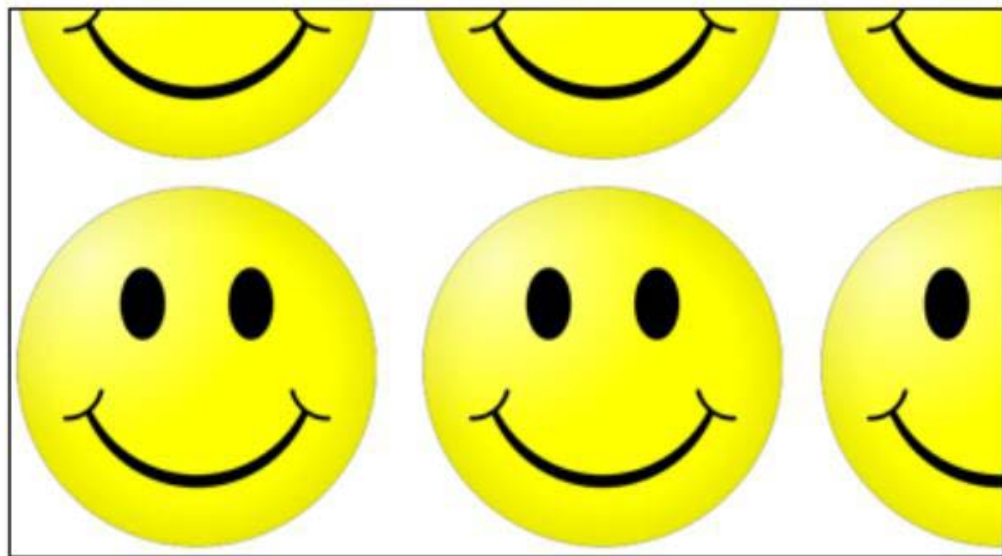


Priestor obrazovky

Priestor textúry

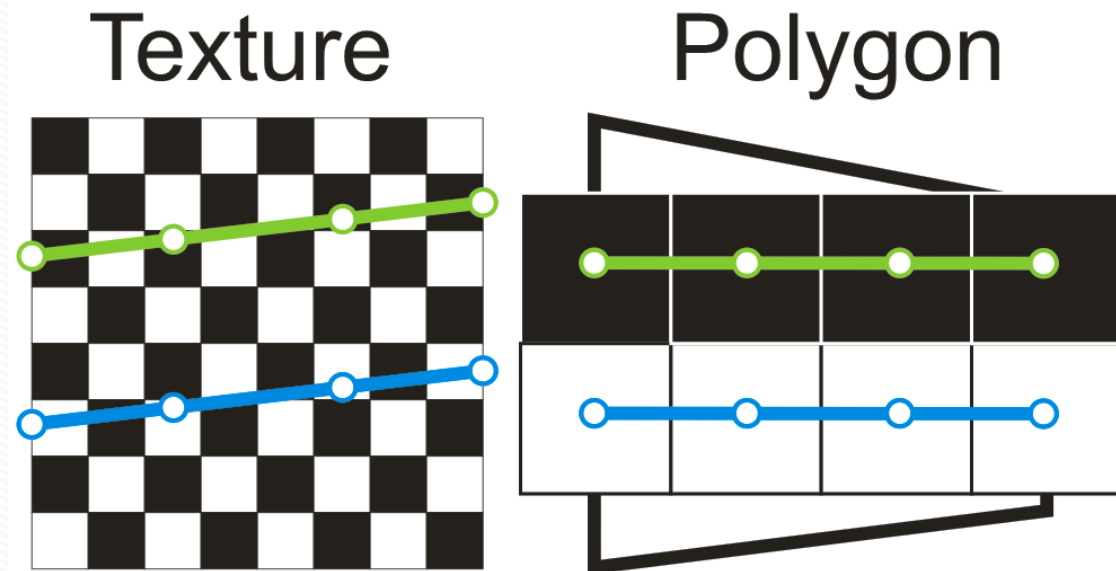
Opakovanie textúry

- Texture tiling
- Čo sa stane, ak sú súradnice textúry > 1 ?
- Napr. $[0, 0]$ $[2.5, 0]$ $[2.5, 1.5]$, $[0, 1.5]$

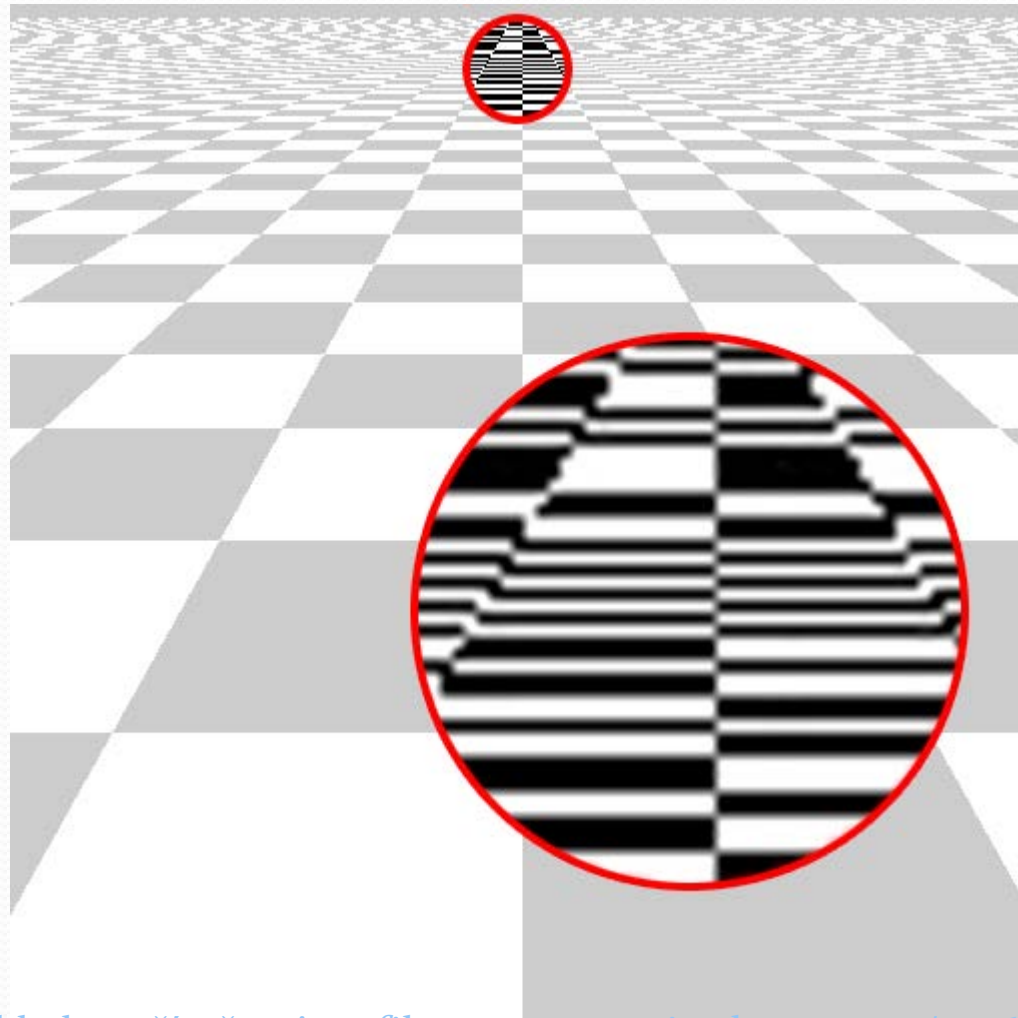


Problémy texturovania

- Veľkosť textúry \leftrightarrow veľkosť polygónu
 - 1-pixelový krok na obrazovke \leftrightarrow 1-pixelový krok v textúre
- Čítanie pixelov z textúry je vzorkovanie
- Artefakty
- Alias

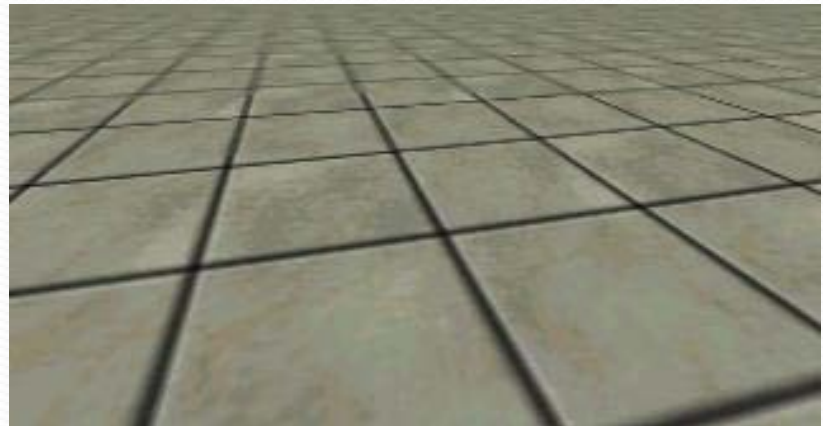


Príklad



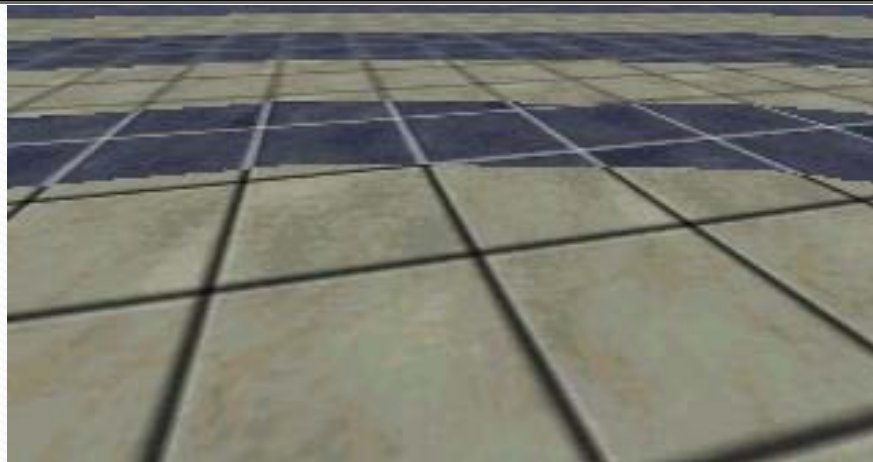
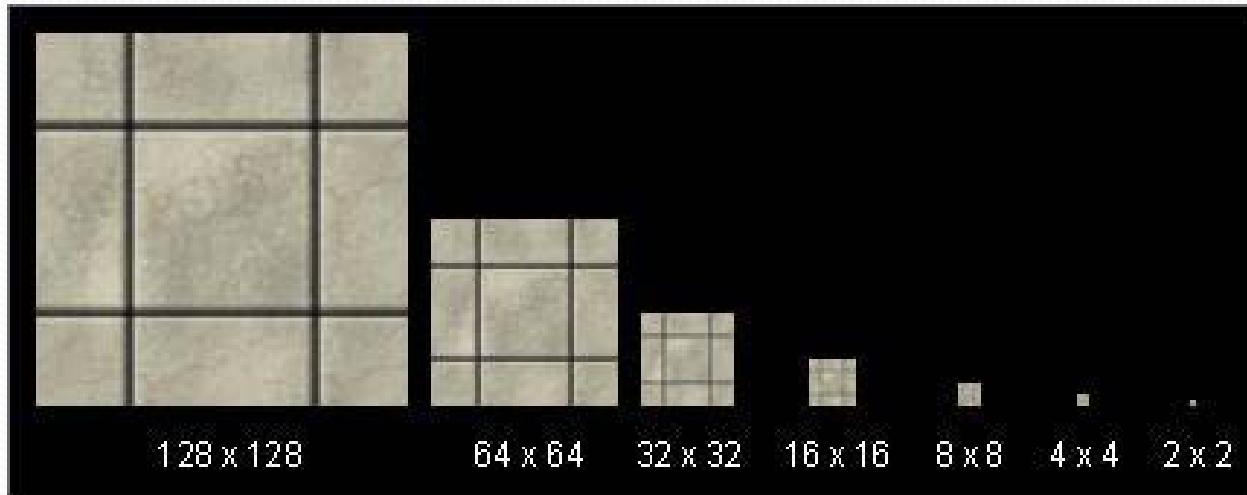
Vylepšenia

- Použitie redukovanej textúry na vzdialené polygóny
- MIP-mapy (Multum in Pravo)



- Drawbacks – “waves”
- <http://www.gamedev.net/reference/articles/article1233.asp>

Vylepšenia – MIP mapy

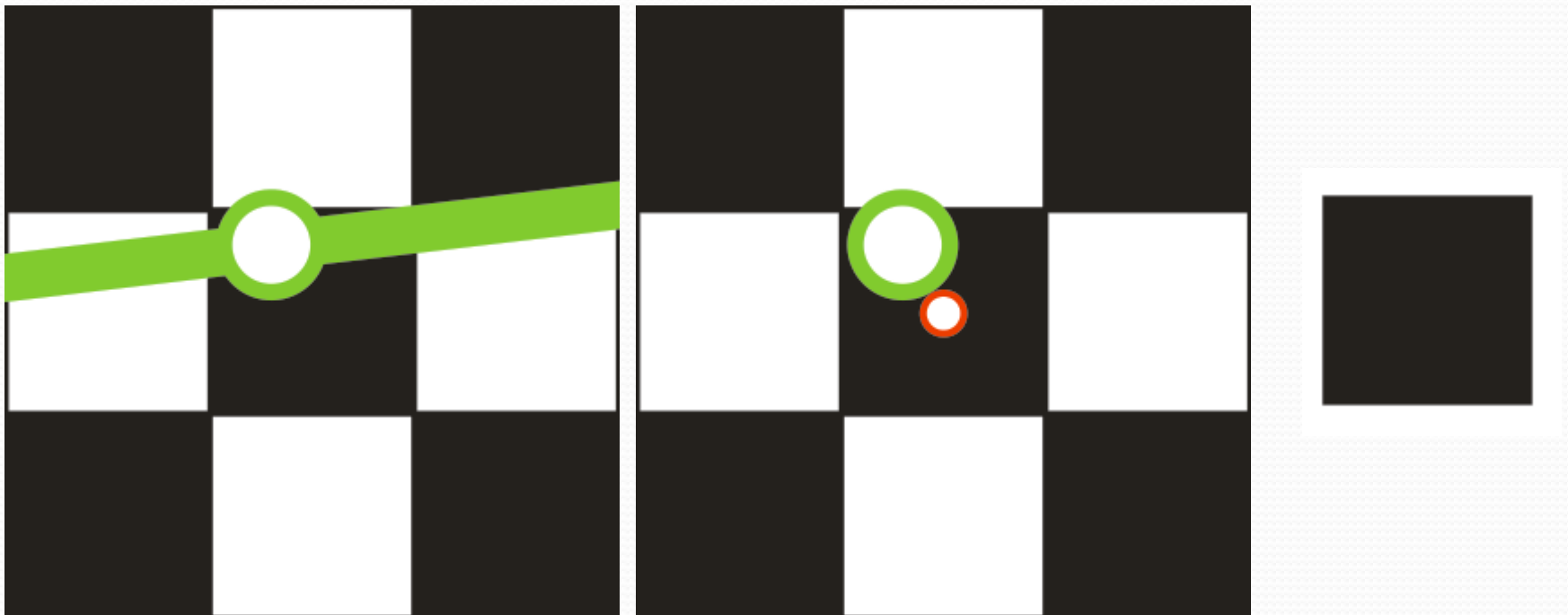


<http://www.gamedev.net/reference/articles/article1233.asp>

Základy počítačovej grafiky a spracovanie obrazu 2015/2016

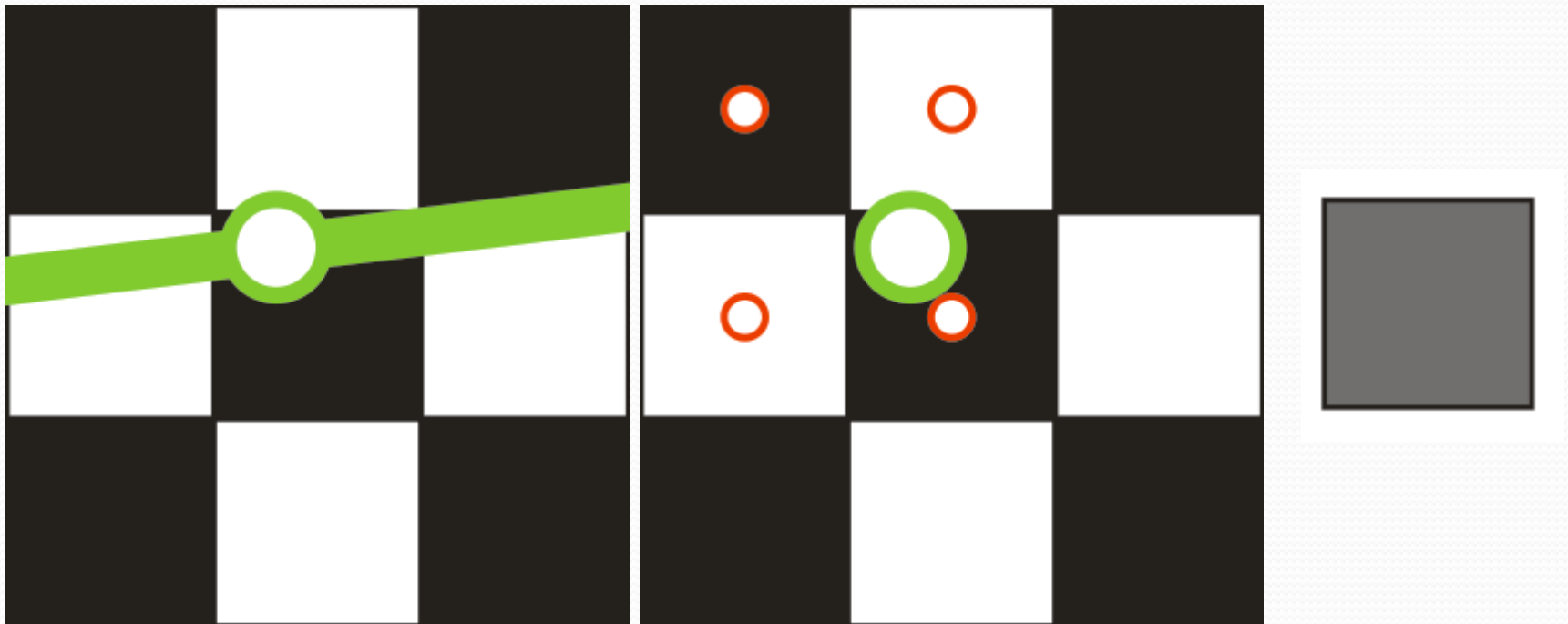
Vylepšenia

- Najbližší sused – nie filtrovanie



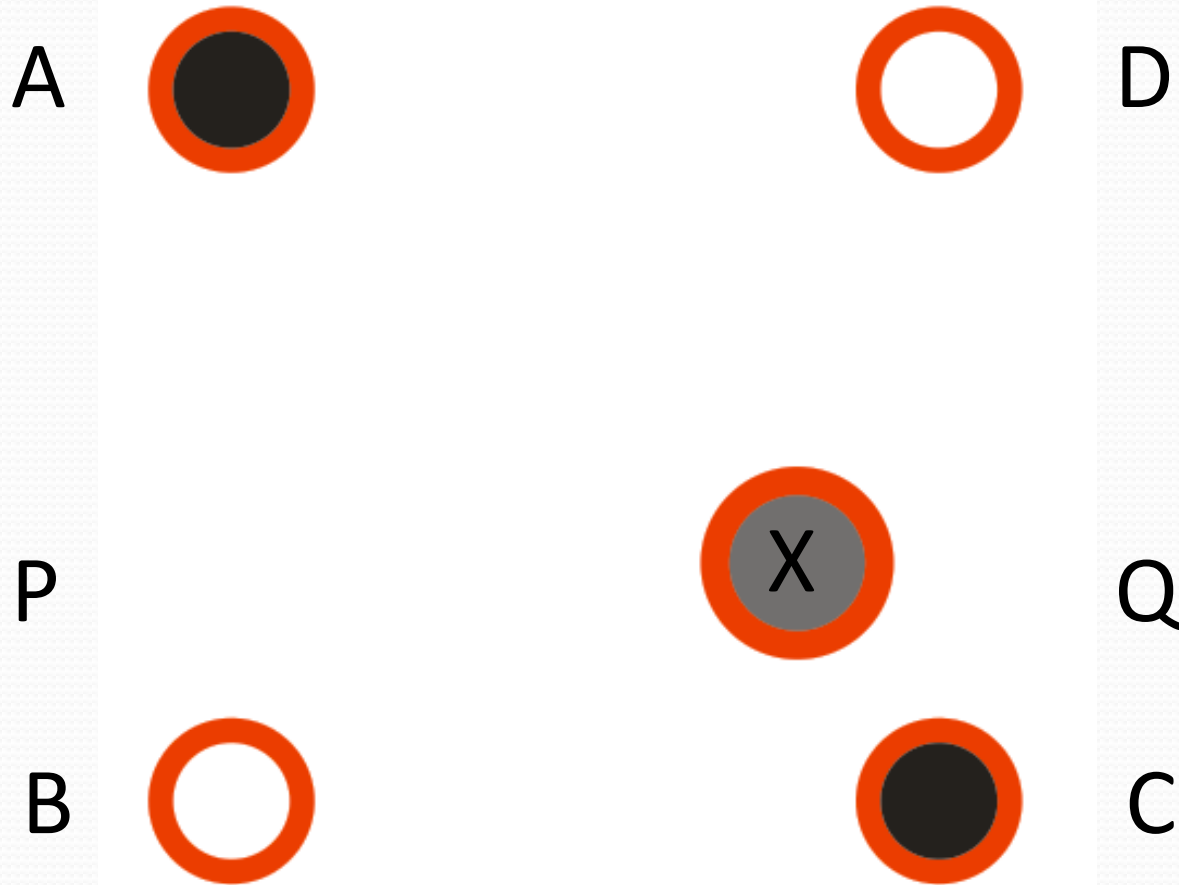
Vylepšenia - Filtrovanie

- Bilineárne filtrovanie



- Drawbacks – artefakty stále vznikajú

Bilineárna interpolácia



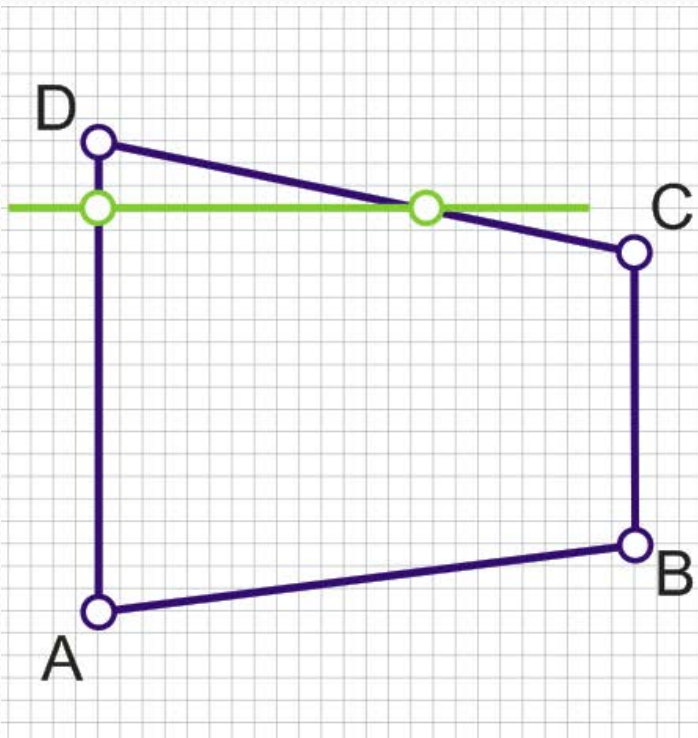
Bilineárna interpolácia

- 4 rohové body A, B, C, D so známymi hodnotami
- 1 vútorný bod X s neznáou hodnotou
- $P = A + u(B-A)$, $Q = D + u(C-D)$
- $X = P + v(Q-P)$
- Maticový zápis

$$X = (1-u, u) \begin{pmatrix} A & D \\ B & C \end{pmatrix} \begin{pmatrix} 1-v \\ v \end{pmatrix} \quad u \in \langle 0, 1 \rangle, v \in \langle 0, 1 \rangle$$

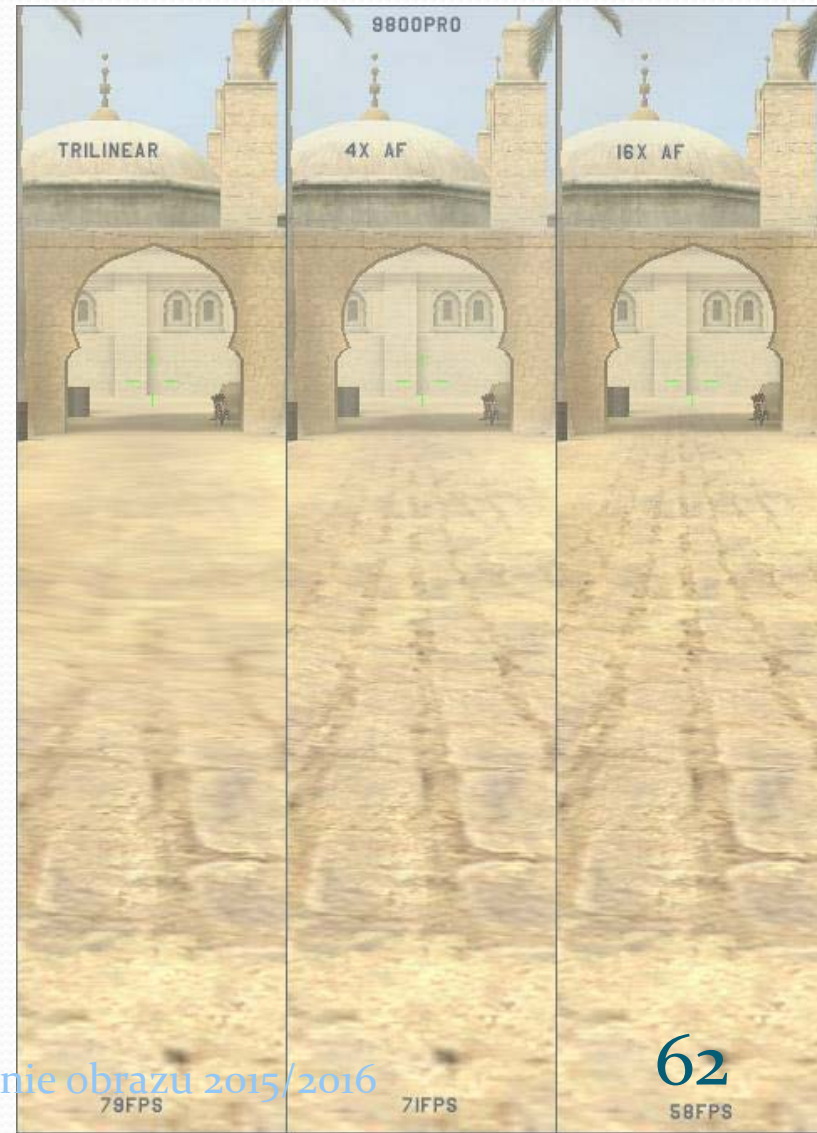
Aplikácia: mapovanie textúr

- Interpolácia $D \leftrightarrow A = P$, $D \leftrightarrow C = Q$, $P \leftrightarrow Q = X$



Ďalšie vylepšenia

- Trilineárna interpolácia
 - Bilineárna interpolácia v MIP mapách+ lineárna interpolácia
- Anizotropné filtrovanie
 - Rôzne vzorky textúry v rôznych smeroch
- Procedurálne textúry
 - Parametrické
 - Menší aliasing

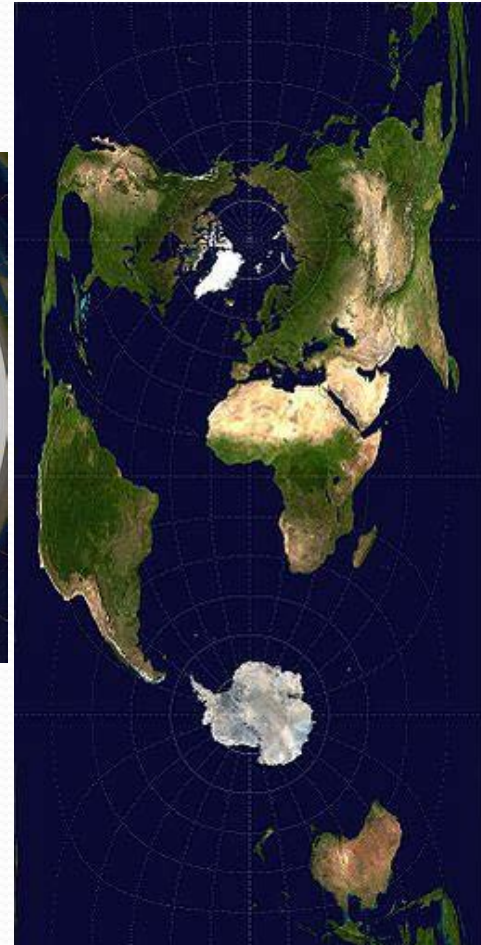
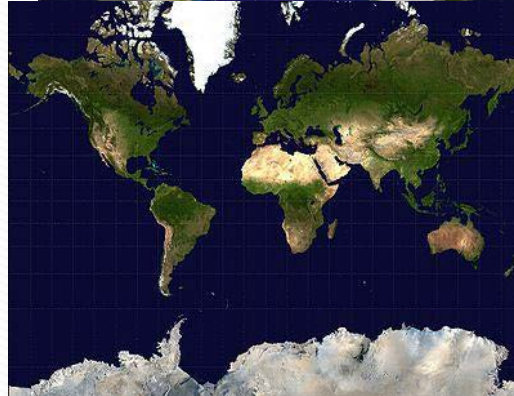
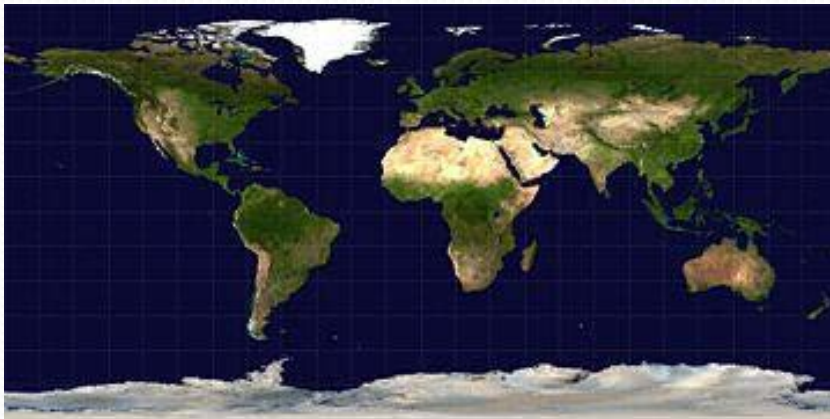
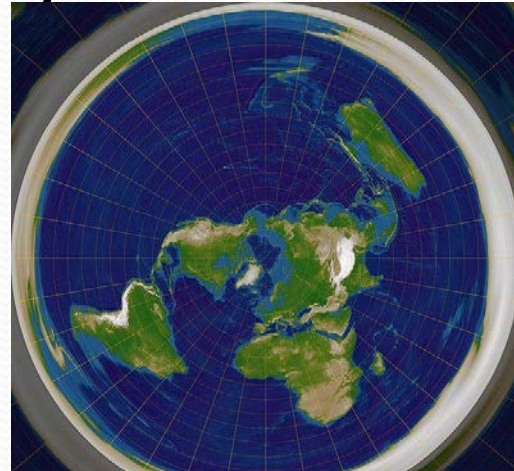


Textúrovanie objektov

- Objekty zložené z polygónov
- Vieme, ako textúrovať polygóny použitím textúrnych súradníc ich vrcholov
- Ale ako nastavíme textúrne súradnice?
 - Okrem manuálneho nastavenia
- Rozloženie objektu do roviny

Textúrovanie objektov - príklad

- Rozloženie zeme do roviny

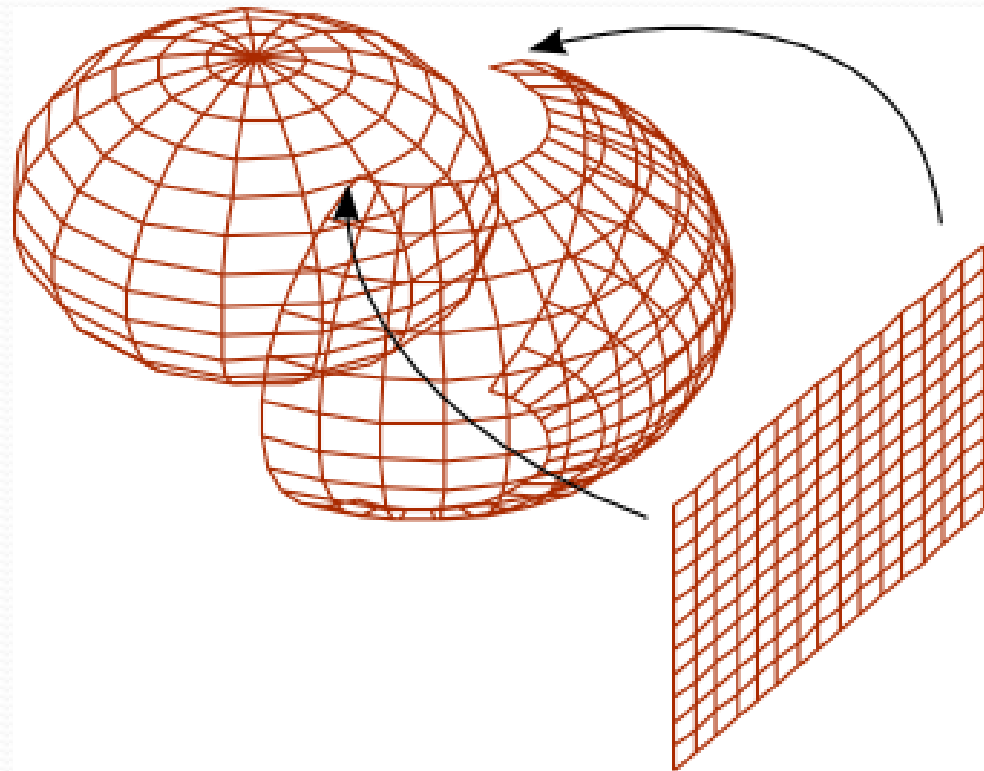


UV mapovanie

- XYZ \leftrightarrow UV
- guľa:

$$u = \frac{x}{\sqrt{x^2 + y^2 + z^2}}$$

$$v = \frac{y}{\sqrt{x^2 + y^2 + z^2}}$$

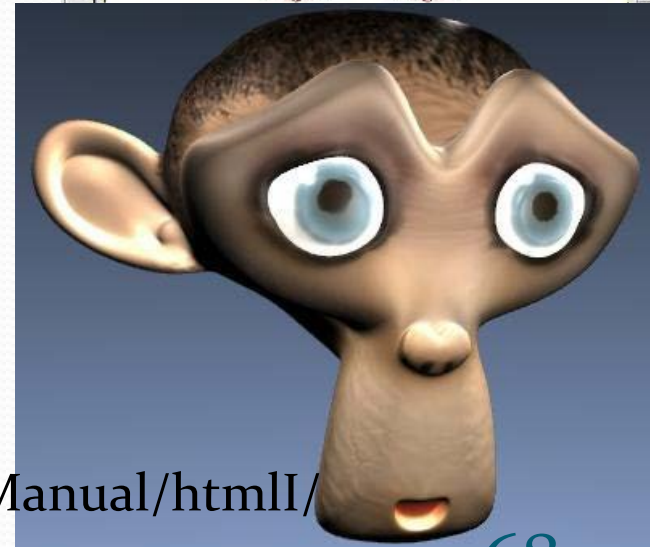
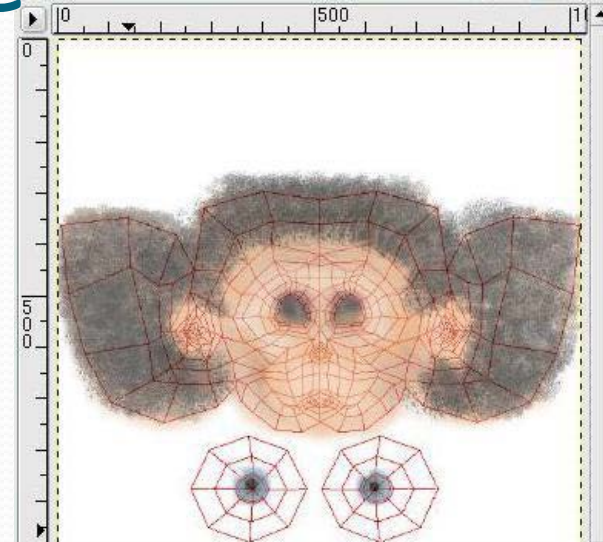
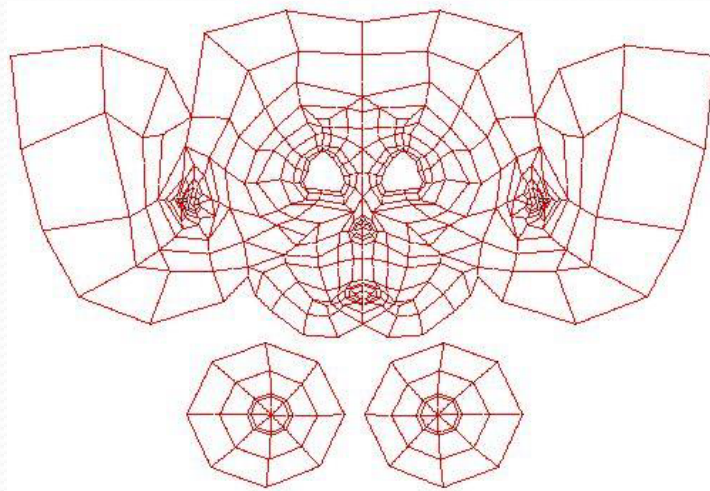


Rôzne UV mapovanie



UV mapovanie

- 3DS MAX:
 - Planar
 - Face
 - Cylindrical
 - Box
 - Shrink wrap
 - Spherical



- <http://www.ru.is/kennarar/hannes/useful/BlenderManual/htmlI/>