

VYPLŇOVANIE A TEXTÚROVANIE

doc. Ing. Branislav Sobota, PhD.

Katedra počítačov a informatiky, FEI TU v Košiciach

P 08

© 2024

VRSTVY VIZUALIZAČNÉHO PROCESU

1. Definovanie/spracovanie modelu
(reprezentácia, súradnicové systémy)
2. Transformácie nad objektami
3. Riešenie viditeľnosti
4. Tieňovanie
5. Osvetľovanie
6. Realistické zobrazovanie
7. Kompozícia a Vykresľovanie



TYPY VYPLŇOVANIA

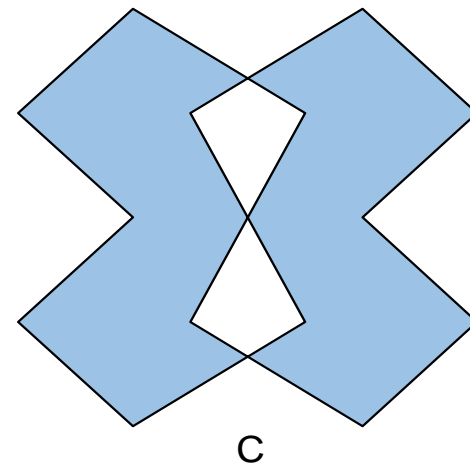
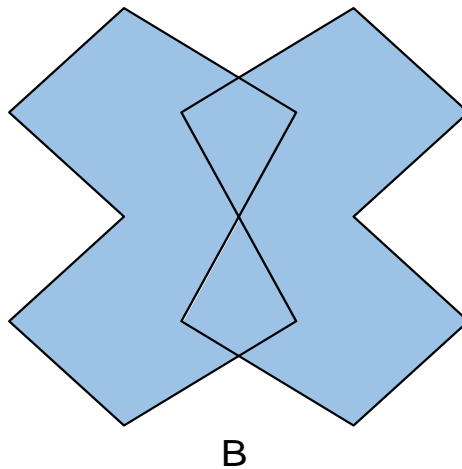
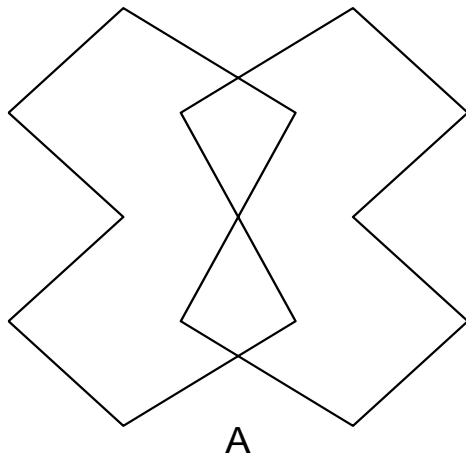
- vyplnenie oblasti jedinou farbou
- vyšrafovanie oblasti
- vyplnenie farebným vzorom (textúrovanie).



ROZDELENIE ALGORITMOV VYPLŇOVANIA

podľa toho akým spôsobom je zadaná hranica vyplňanej oblasti:

- hranica definovaná *geometricky*
- hranica *nakreslená na zobrazovači*



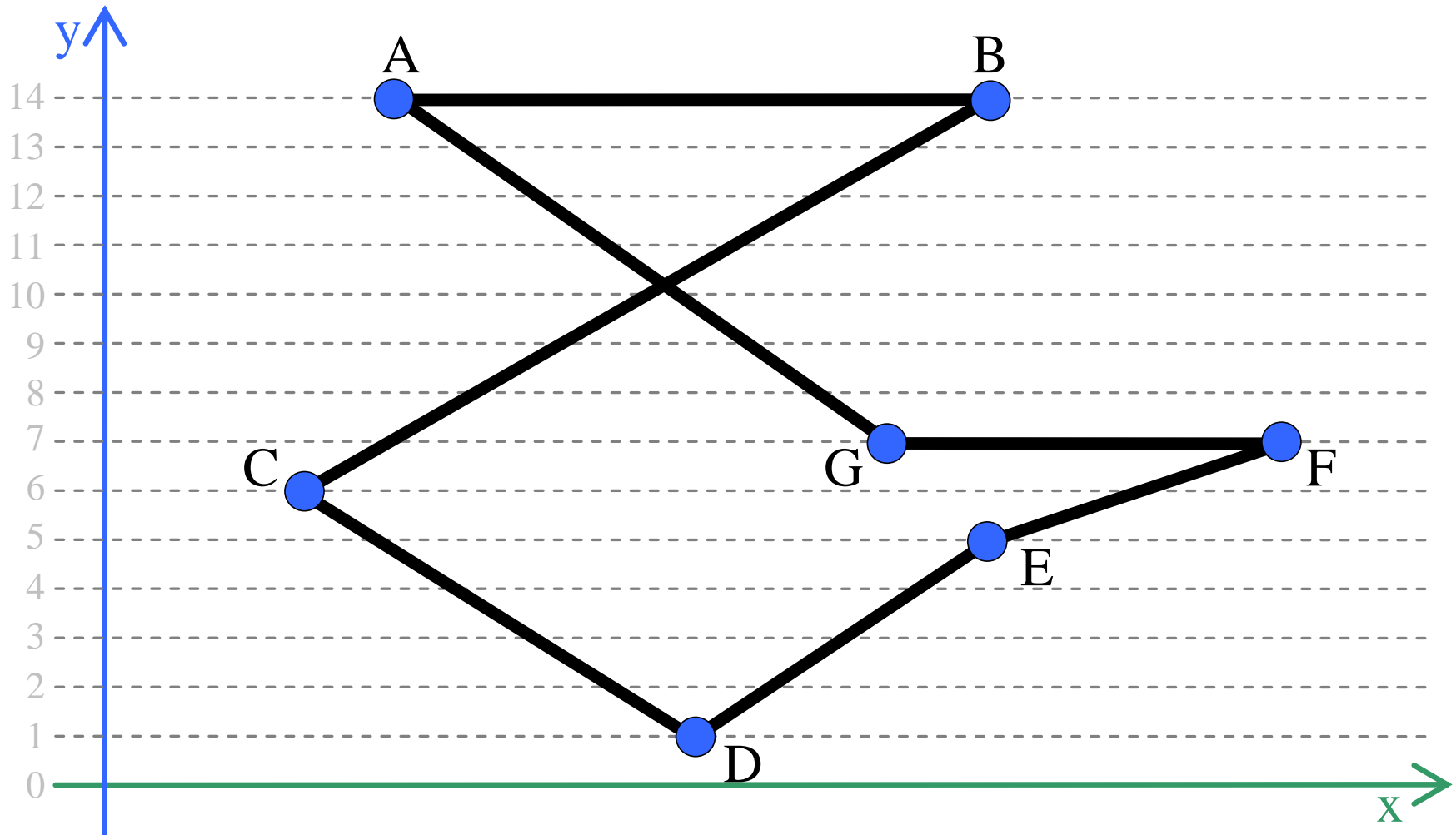
METÓDA RIADKOVÉHO ROZKLADU

SCANLINE

1. Postupuje sa od najvyššieho vrcholu oblasti k najnižšiemu, smerom zľava doprava v každom riadku.
2. Pre jednotlivé rozkladové riadky rovnobežné s osou X a s konštantnou súradnicou Y , klesajúcou s krokom -1 , sa nájdu priesečníky s hranicami vyplňanej oblasti.
3. Vo výslednom zozname sú usporiadané priesečníky zľava doprava a vyfarbené úseky medzi nepárnymi a párnymi priesečníkmi. Preto je nutné aby počet priesečníkov bolo párne číslo.

METÓDA RIADKOVÉHO ROZKLADU

SCANLINE

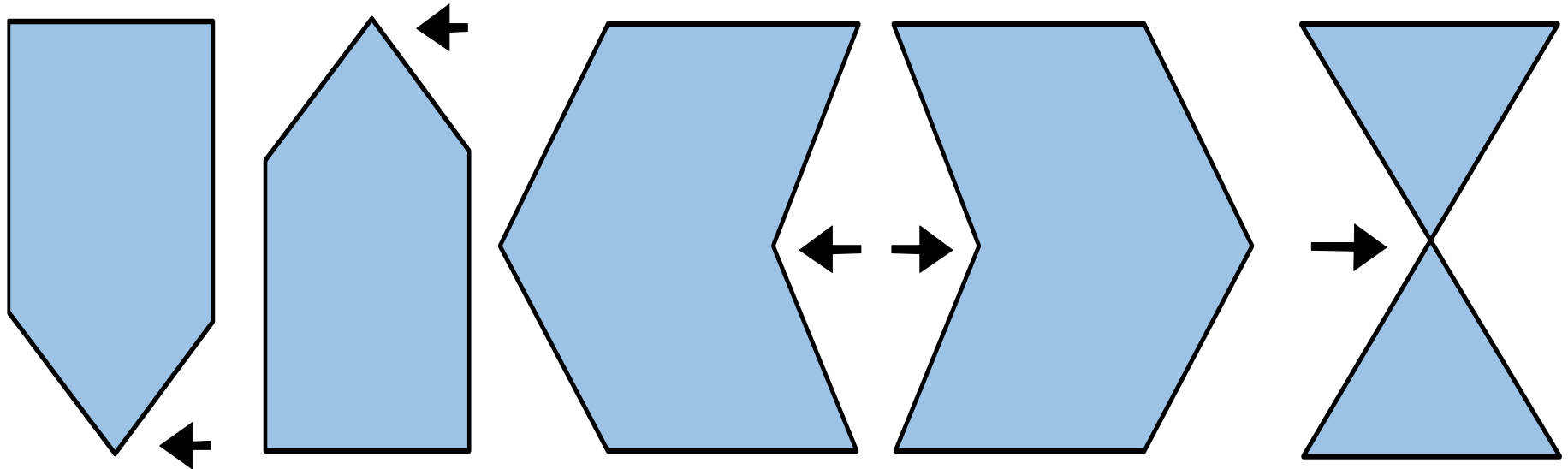


NÁJDENIE PRIESEČNÍKOV PRI METÓDE RIADKOVÉHO ROZKLADU

$$x = a_x + \frac{c_y - a_y}{b_y - a_y} \cdot (b_x - a_x)$$

$$y = c_2$$

EXTRÉMY PRI METÓDE RIADKOVÉHO ROZKLADU



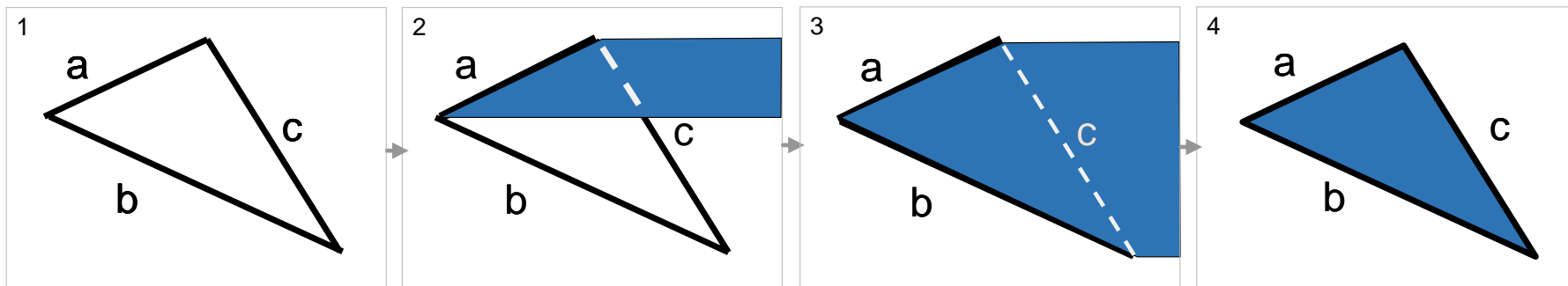
INVERZNÉ VYPLŇOVANIE

Využíva metódu XOR ($0 \vee 0 = 1 \vee 1 = 0$, $0 \vee 1 = 1 \vee 0 = 1$)

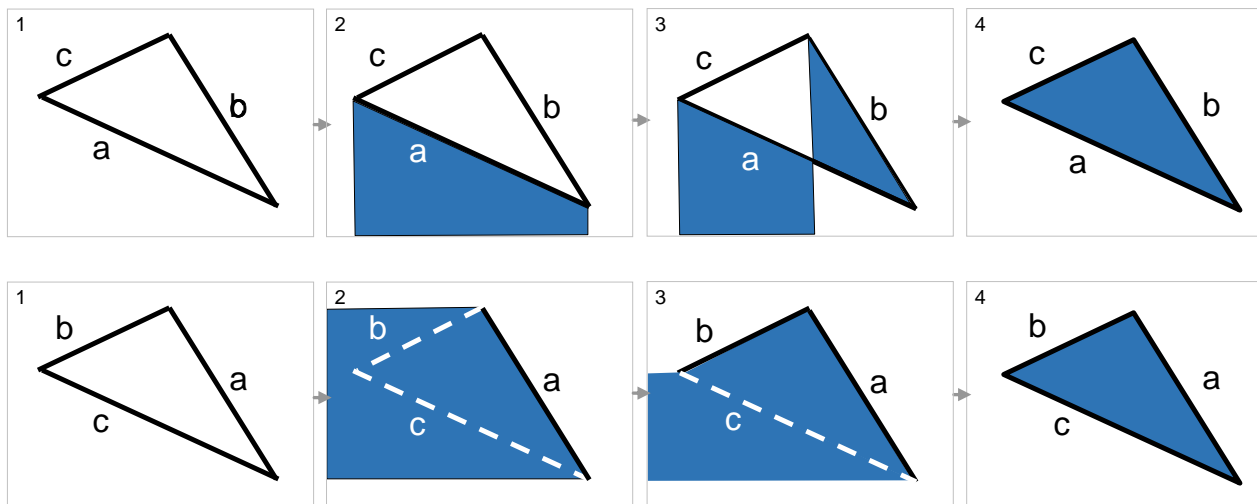
Dva typy:

- vodorovné (riadkové) vyplňovanie
- zvislé (plotové) vyplňovanie

INVERZNÉ VYPLŇOVANIE



alternatívy:



VYPLŇOVANIE SPEKTRUM (2 SPÔSOBY)

Prvý spôsob:

výplne spektrum využíva metódu riadkového rozkladu popísanú skôr.

- Na začiatku sa musí otočiť mnohouholník tak, aby smer vykresľovania priamok bol rovnobežný s osou x , t.j. o uhol β a metódou riadkového rozkladu sa nájdú priesečníky.
- Pri vykresľovaní úsekov medzi priesečníkmi sa musia úseky spätne otočiť o uhol β .
- Rozdiel oproti riadkovej výplni je v tom, že oblasť nie je vyplňaná jednou farbou, ale farba postupne plynule prechádza z prvej farby do druhej.

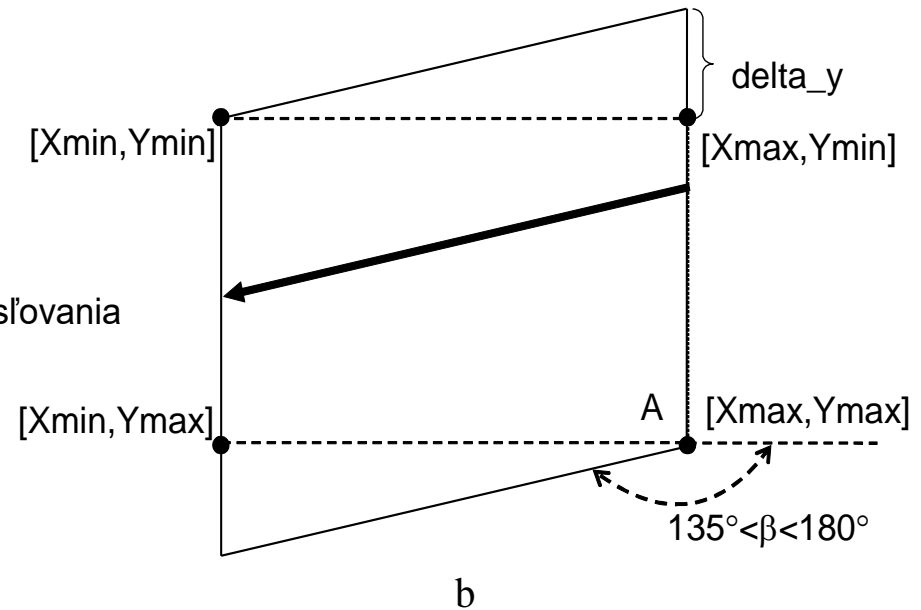
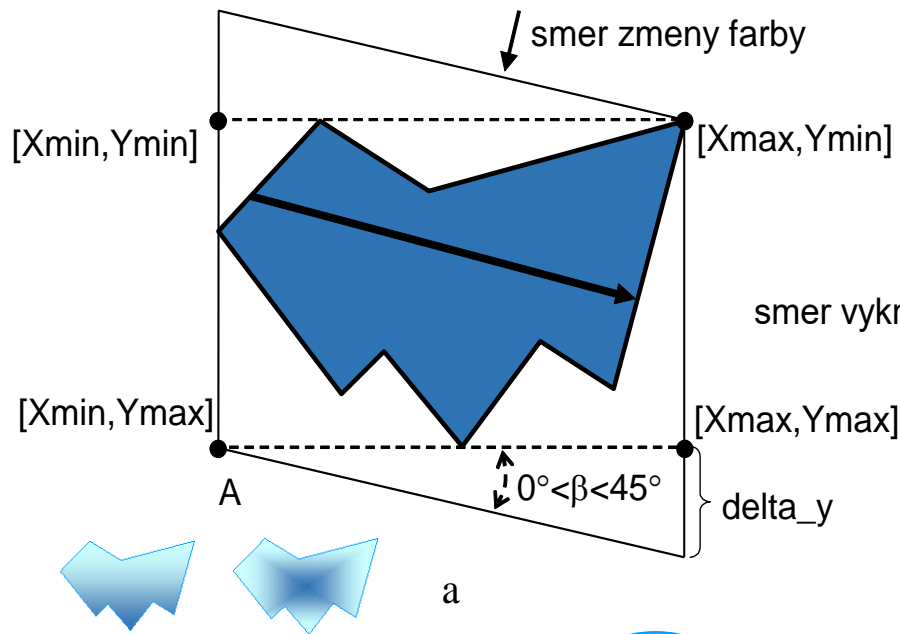
VYPLŇOVANIE SPEKTROM (2 SPÔSOBY)

Druhý spôsob:

využíva možnosť nastavenia orezávacej oblasti, ktorú poskytuje napr. grafické rozhranie MS Windows.

- Na začiatku sa nastaví orezávacia oblasť na celú oblasť mnohoúhelníka a vypočítajú sa súradnice $Xmin$, $Xmax$, $Ymin$, $Ymax$.
- Podľa veľkosti uhla β je vyplnený príslušný rovnobežník a orezávacia oblasť zabezpečí vyplnenie len v oblasti mnohoúhelníka.

VYPLŇOVANIE SPEKTRUM



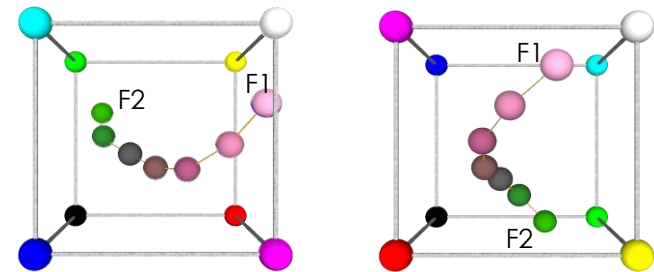
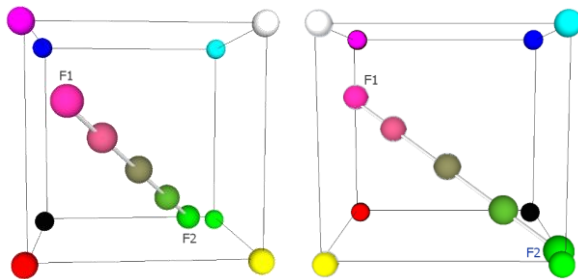
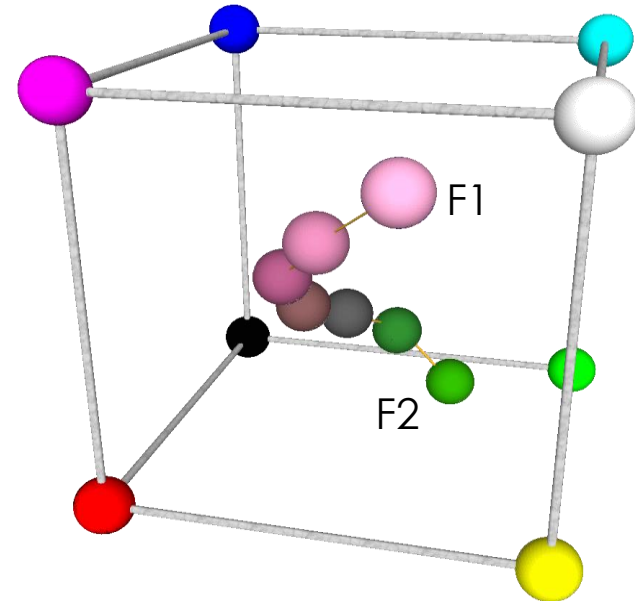
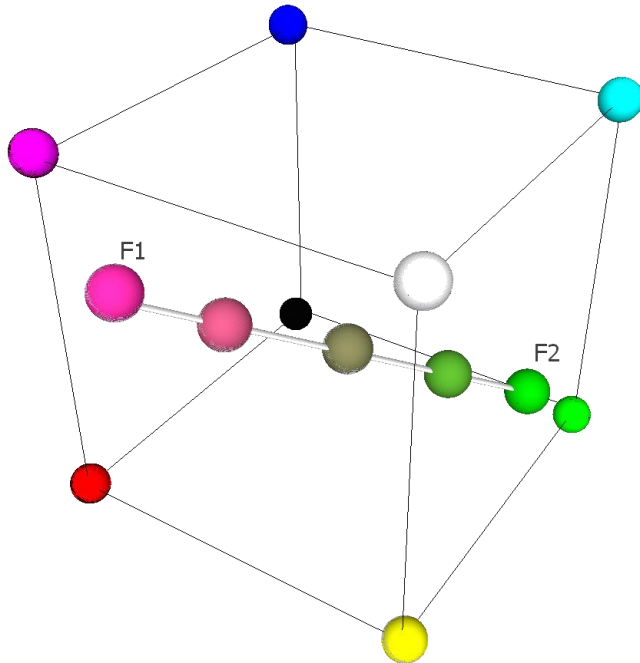
VYPLŇOVANIE SPEKTRUM

Plynulý prechod farieb:

- Zistenie rozdielu jednotlivých zložiek (napr. RGB) prvej a druhej farby.
- Vypočítanie počtu úsečiek v rovnobežníku.
- Vypočítanie prírastkov jednotlivých zložiek ako pomer rozdielu zložiek prvej a druhej farby a počtu úsečiek, napr. pre zložku R
- alebo prechod napr. v RGB modeli

VYPLŇOVANIE SPEKTROM (PRECHOD V RGB MODELI)

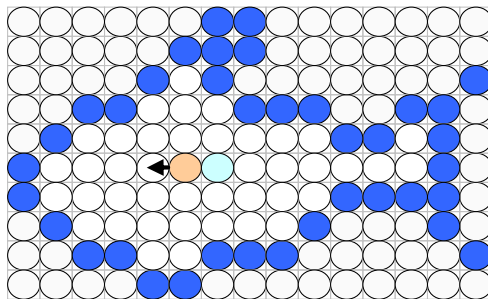
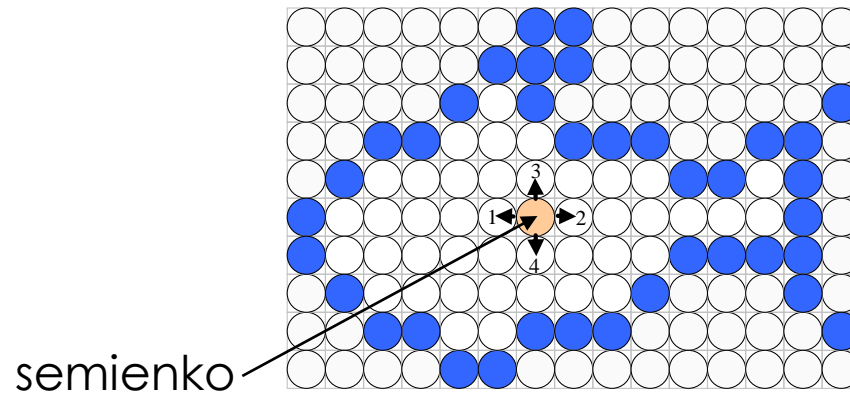
Ekvivalent alfa miešania (lineárne aj nelineárne, RGB priestor)



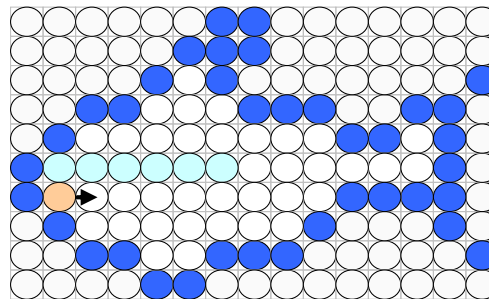
SEMIENKOVÉ VYPLŇOVANIE

```
SeedFill(x, y)
{
    SetPixel(sx, sy, BLUE);
    if(((GetPixel(sx-1, sy)) != HR_FARBA) &&
        ((GetPixel(sx-1, sy)) != BLUE)) SeedFill(sx-1, sy);
    if(((GetPixel(sx+1, sy)) != HR_FARBA) &&
        ((GetPixel(sx+1, sy)) != BLUE)) SeedFill(sx+1, sy);
    if(((GetPixel(sx, sy-1)) != HR_FARBA) &&
        ((GetPixel(sx, sy-1)) != BLUE)) SeedFill(sx, sy-1);
    if(((GetPixel(sx, sy+1)) != HR_FARBA) &&
        ((GetPixel(sx, sy+1)) != BLUE)) SeedFill(sx, sy+1);
}
```

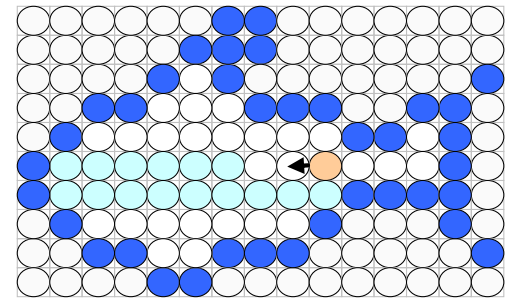

SEMIENKOVÉ VYPLŇOVANIE



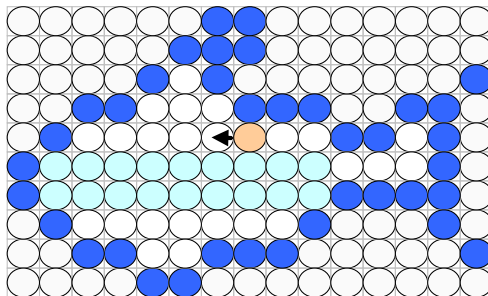
po 1. kroku



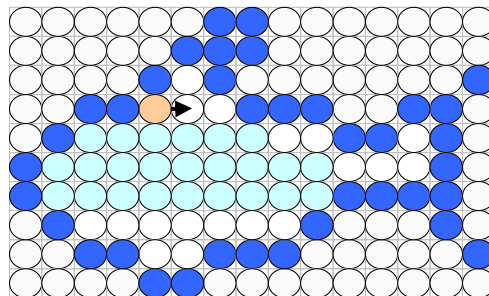
po 6. kroku



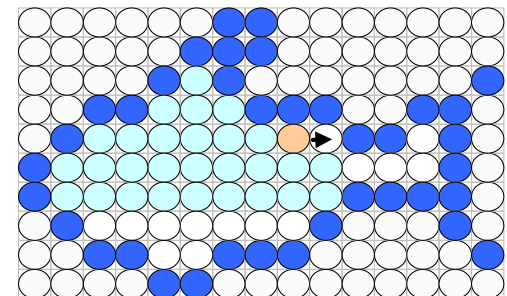
po 15. kroku



po 18. kroku

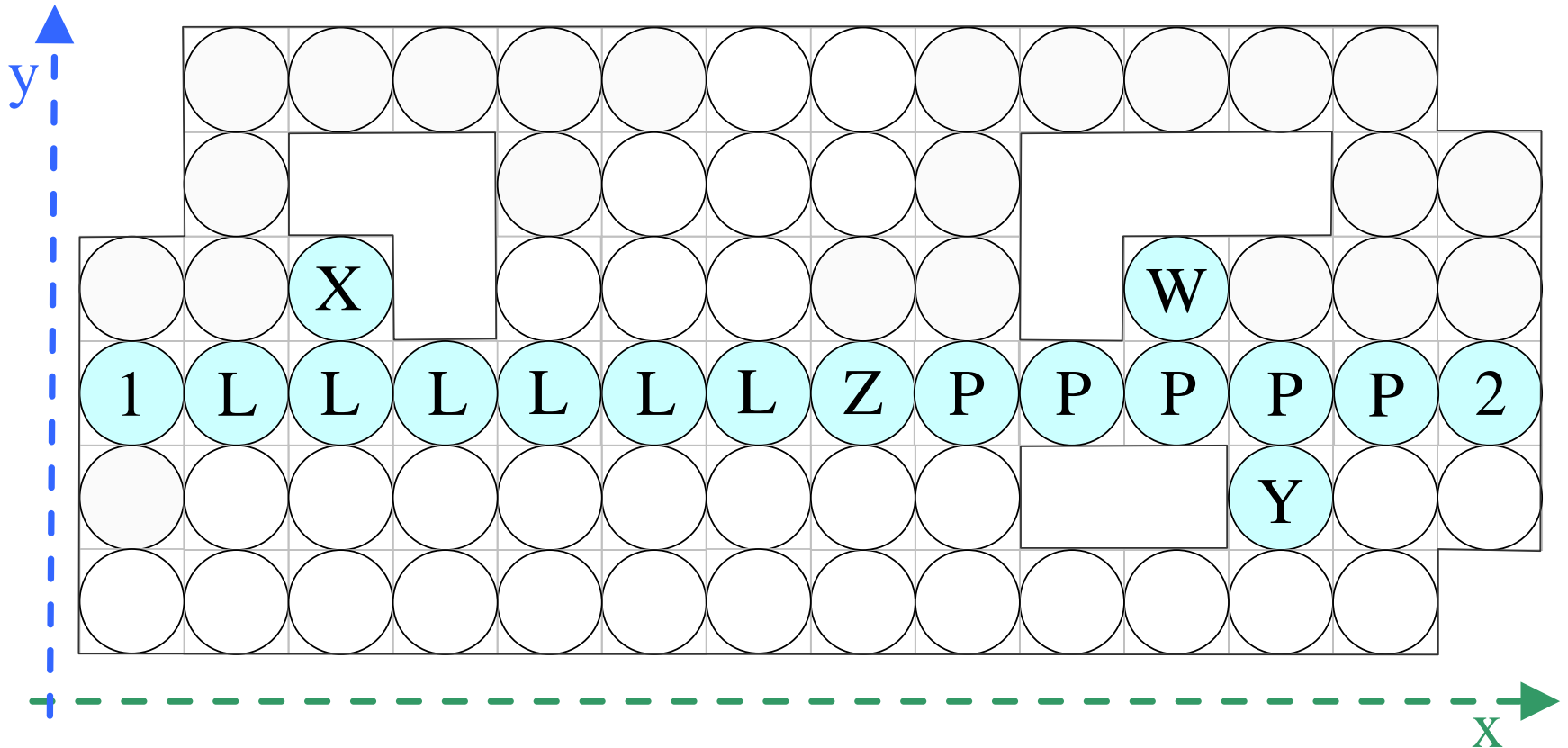


po 24. kroku



po 28. kroku

NEREKURZÍVNE SEMIENKOVÉ VYPLŇOVANIE



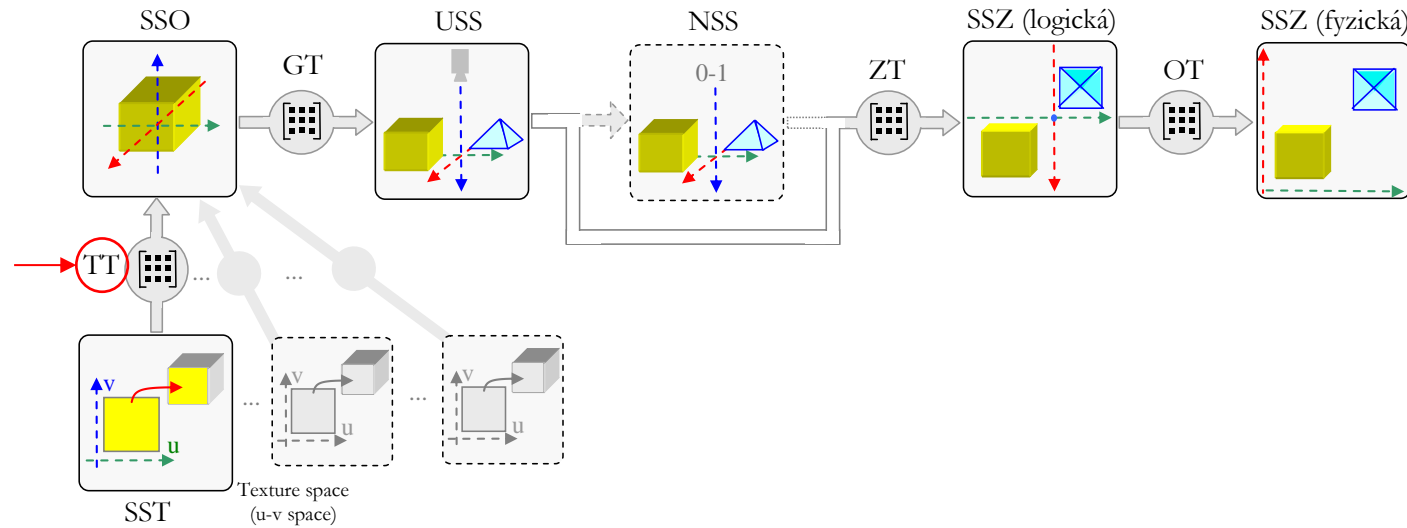
TEXTÚROVANIE

Proces (Textúrovacia transformácia, TT) nanášania obrazových vzoriek (textúr, tapiet) na povrch objektov za účelom získania vizuálneho dojmu, že objekt je z istého materiálu (napr. drevo, kameň, kov a pod.).

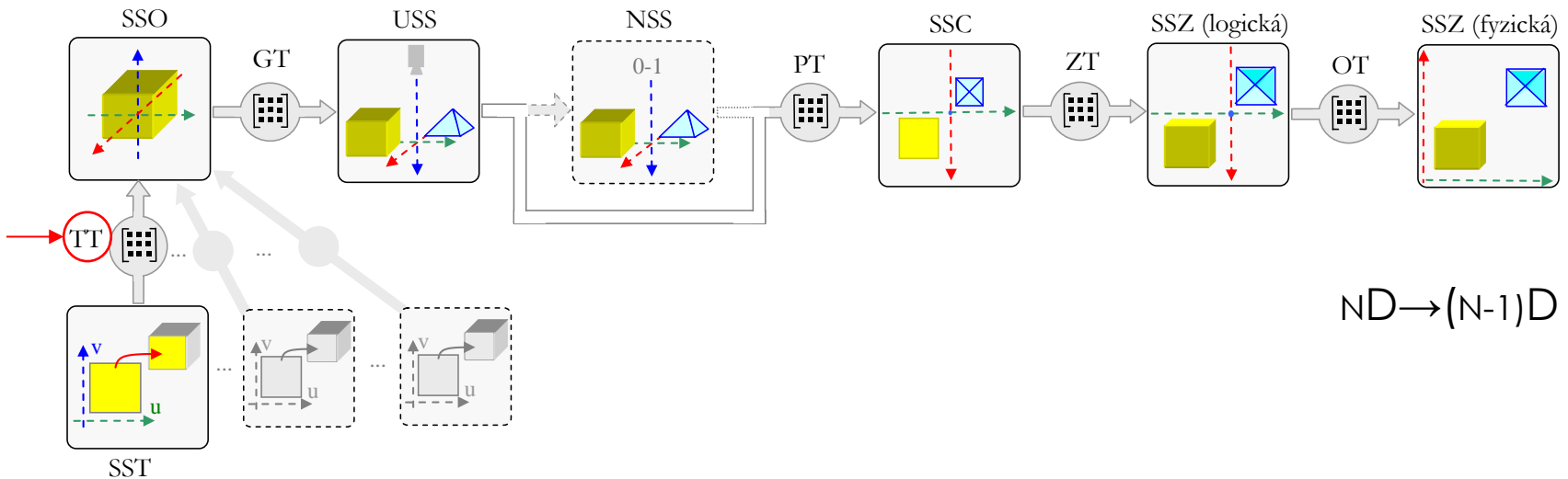
- Typy textúr podľa topologického rozmeru:
 - 1D, **2D**, 3D
- Typy textúr podľa spôsobu nanášania:
 - Statické
 - Dynamické
 - Procedurálne
 - Animačné



TEXTÚROVANIE A TRANSFORMAČNÉ REĹAZCE



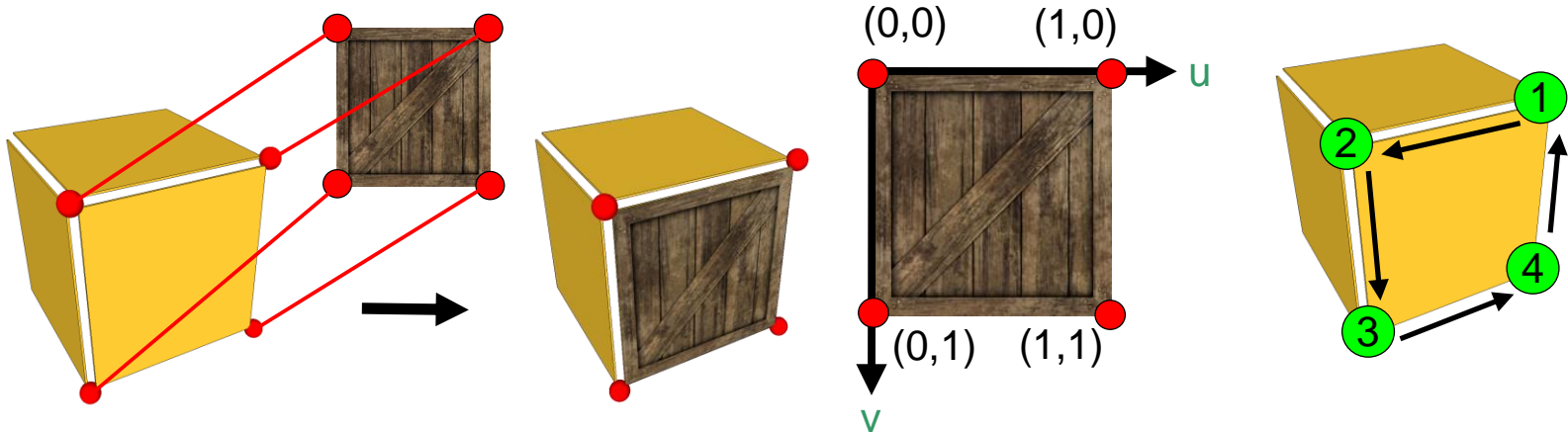
ND \rightarrow ND



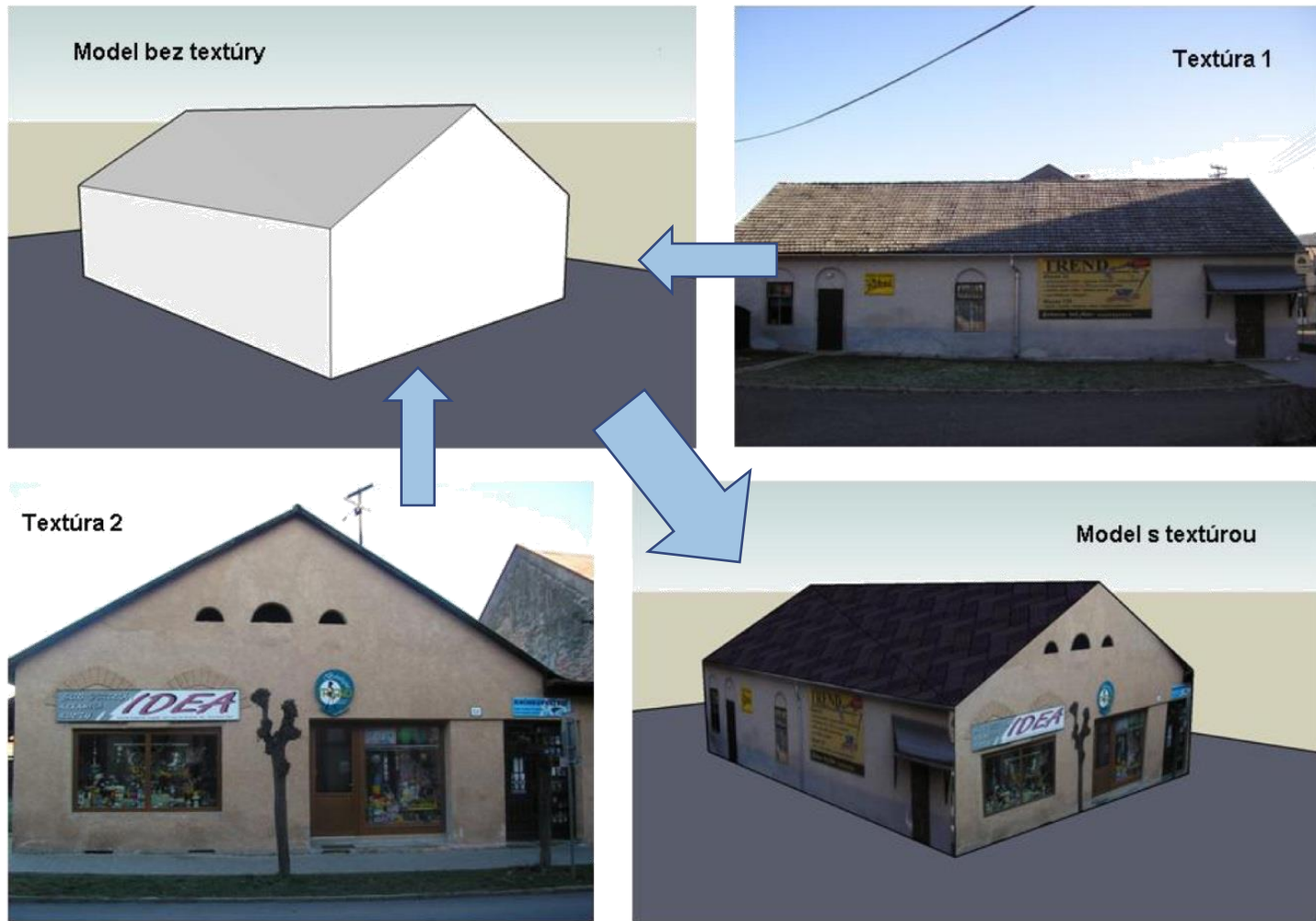
ND \rightarrow (N-1)D

TEXTÚROVANIE (BILINEÁRNE, UV)

- Pri tomto procese nanášania obrazového formátu na plochy trojrozmerného objektu sa najčastejšie používa ako vzor obrázok (textúra).
- Obrazový formát (obrázok, textúra) využíva dvojrozmernú súradnicovú sústavu $ST [u,v]$, ktorej hodnoty súradníc u,v zodpovedajú jednotkovej miere obrazového formátu.
- Súradnica „ u “ zodpovedá x-ovej osi obrázka (textúry),
- súradnica „ v “ zodpovedá y-ovej osi textúry.
- Jedna jednotka = dĺžka/šírka obrázku textúry. (u/v)

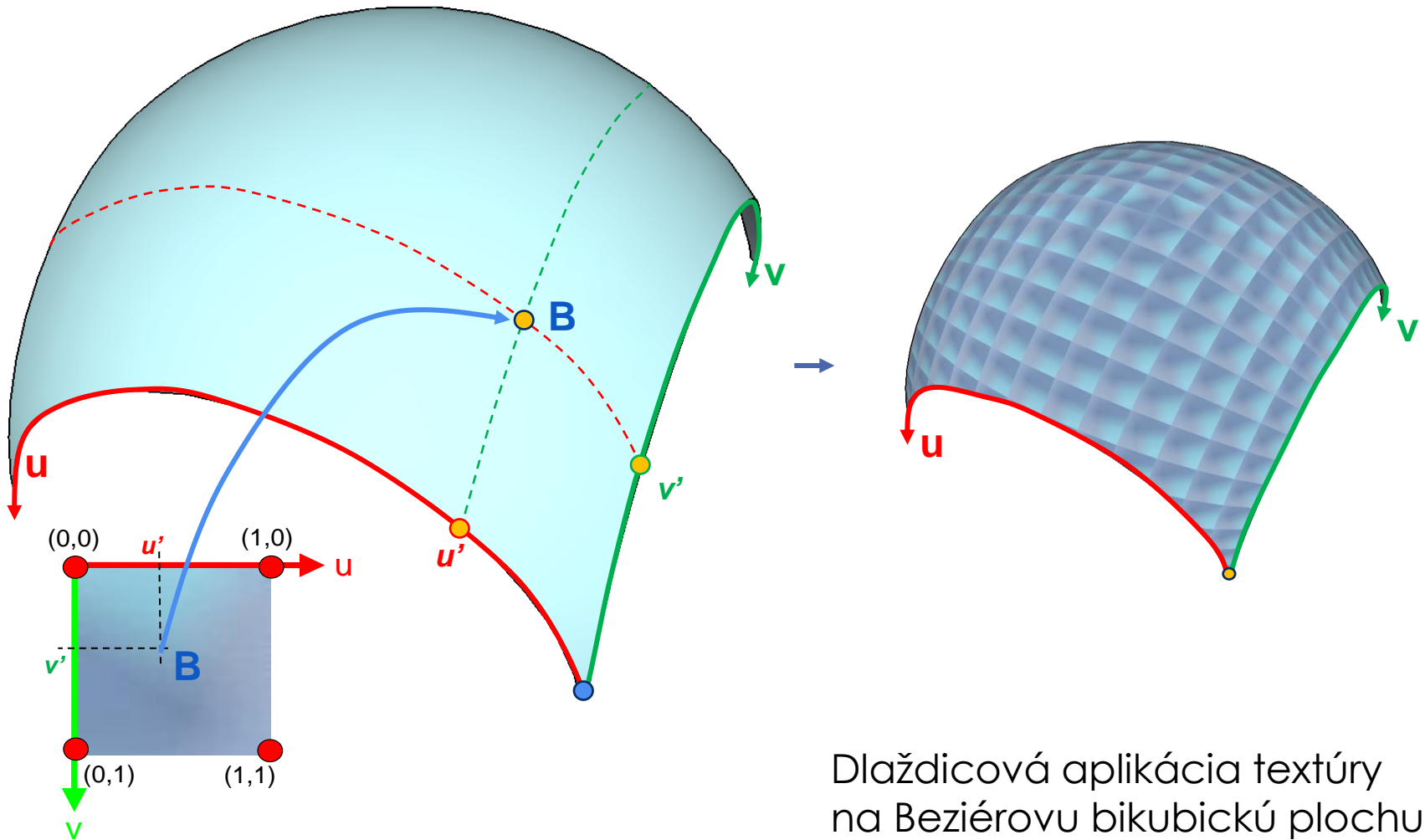


TEXTÚROVANIE (BILINEÁRNE, UV)



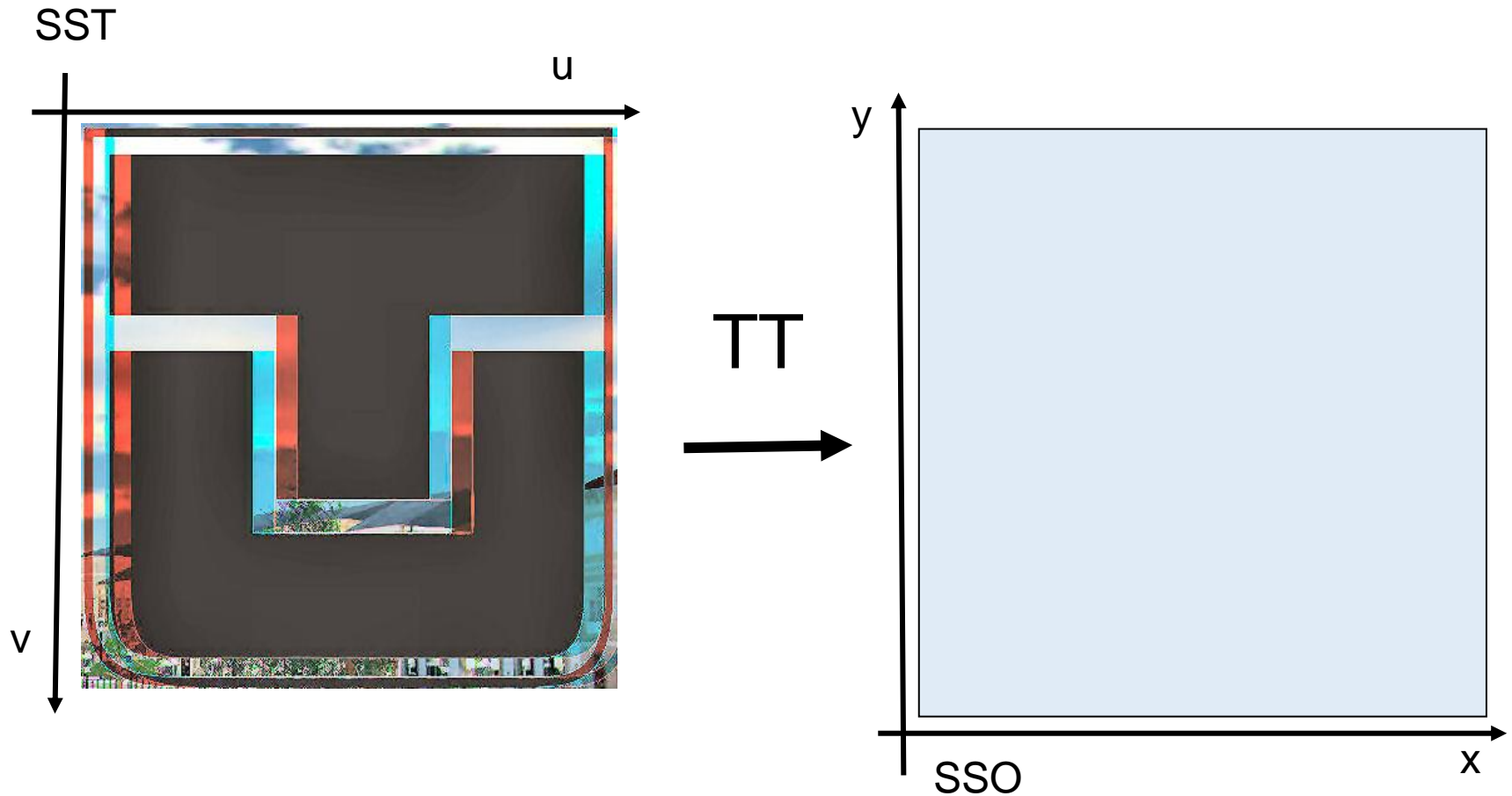
postup pri textúrovaní

TEXTÚROVANIE (BILINEÁRNE, uv)

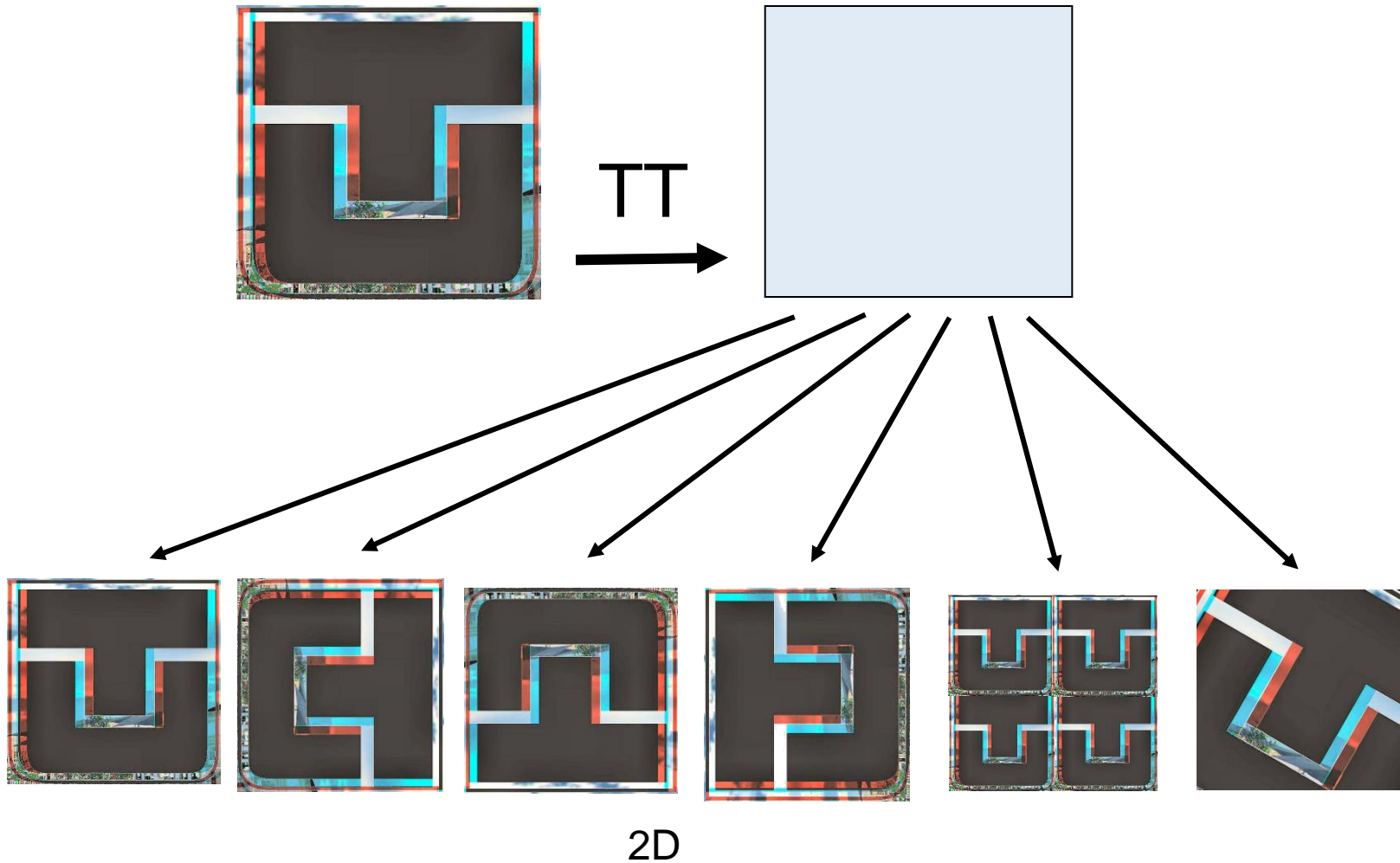


Dlaždicová aplikácia textúry
na Beziérovu bikubickú plochu

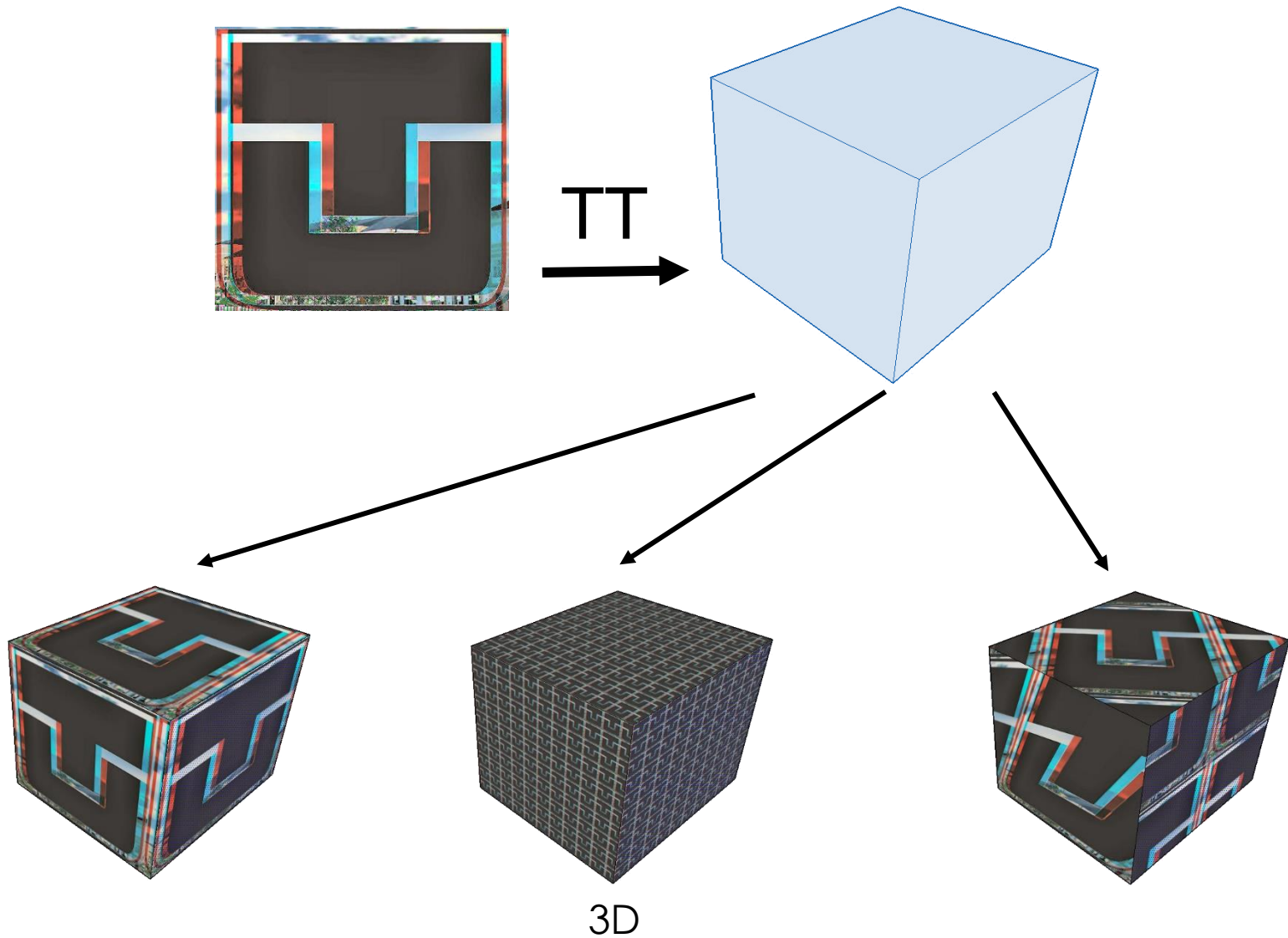
APLIKÁCIA (MAPOVANIE) TEXTÚRY



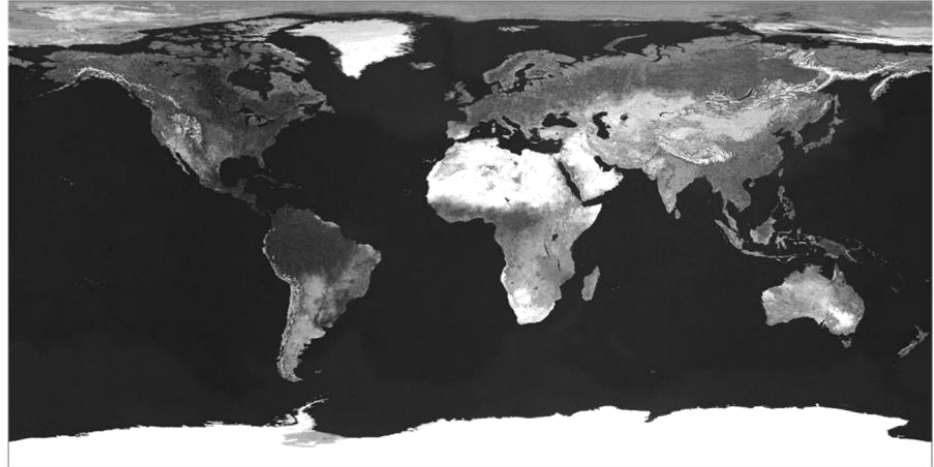
APLIKÁCIA (MAPOVANIE) TEXTÚRY



APLIKÁCIA (MAPOVANIE) TEXTÚRY



BUMP APLIKÁCIA (MAPOVANIE) TEXTÚRY



normal



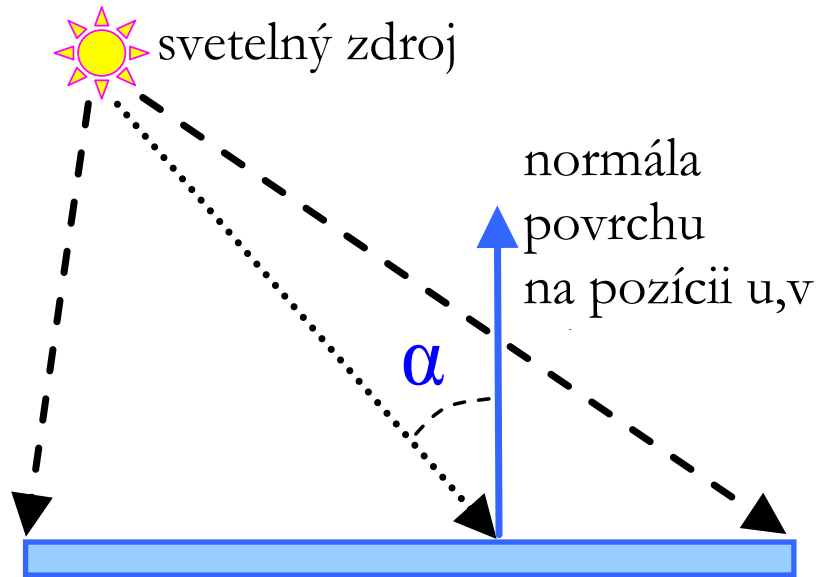
s tieňovaním



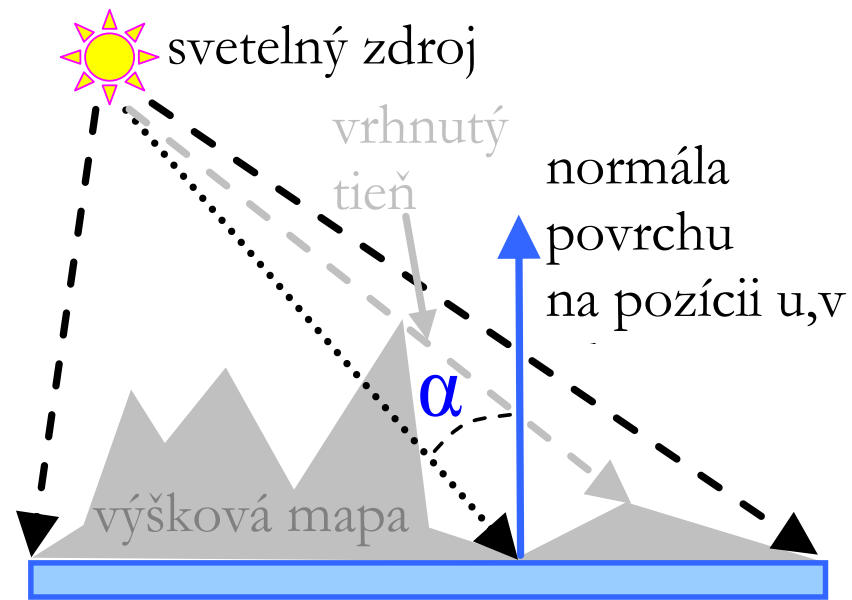
bump (s výškovou mapou)

IMPLEMENTÁCIA TIEŇOVANIA A OSVETĽOVANIA TEXTÚRY

$$\alpha = \arccos \left(\frac{a_x \cdot b_x + a_y \cdot b_y + a_z \cdot b_z}{\sqrt{a_x^2 + a_y^2 + a_z^2} \cdot \sqrt{b_x^2 + b_y^2 + b_z^2}} \right)$$



s tieňovaním

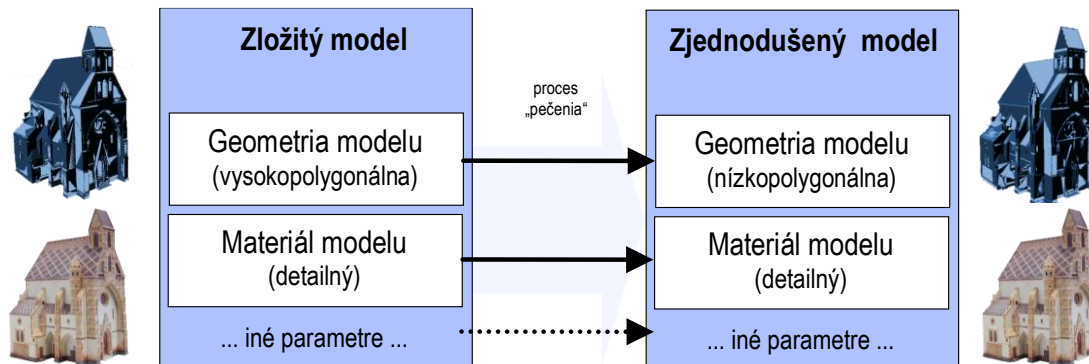


s výškovou mapou

upravuje sa farba resp. jas alebo saturácia bodov textúry, čím vzniká dojem tieňovania alebo hrboľatosti povrchu

„PEČENIE“ TEXTÚRY (TEXTURE BAKING)

- „Pečenie“ textúry je proces prenosu údajov o textúre (materiále) zo zložitejšieho (detailnejšieho, presnejšieho) 3D modelu na iný jednoduchší (menej presný, decimovaný) 3D modelu.
- Prenášané parametre/údaje pri tomto procese môžu byť napríklad: textúra, farba, osvetlenie, tieň, odrazy a podobne
- Vo všeobecnosti sa tento proces používa najmä pri snahe vziať model definovaný vysokým počtom polygónov (highpoly) a previesť ho na zjednodušený model s nízkym počtom polygónov (lowpoly).
Najčastejšie dôvody použitia:
 - nízkopolygonálny model má menej údajov, a preto má menšiu veľkosť súboru ako vysokopolygonálny model.
 - mnohé aplikácie najmä herné, webové prehliadače a napr. virtuálno-reálné systémy často zápasia so spracovaním vysokopolygonálnych modelov, najmä ak je potrebné ich spracovávať v reálnom čase.



Q & A

branislav.sobota@tuke.sk

Katedra počítačov a informatiky, FEI TU v Košiciach

© 2024