

# Virtualna okruženja

Igor S. Pandžić, Tomislav Pejša

“Specijalni” efekti

# Što je specijalno?

- Većina prirodnih vizualnih efekata može se simulirati na računalu
  - ako imate dovoljno vremena
- Sve više efekata je moguće realizirati u stvarnom vremenu
- Specijalni efekt – sve što nije rutinski dio grafičkog protočnog sustava
- Dolaskom *shadera* pojam sve više gubi smisao – svi efekti su sada „specijalni“

# Pregled predavanja

- Metode teksturiranja
- Tehnike panoa
- Zrcaljenje
- Sjene

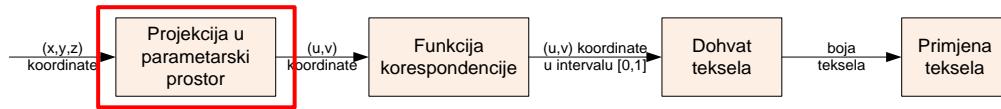
# Metode preslikavanja teksture

- Poopćeno teksturiranje
- Animacija teksture
- Preslikavanje materijala
- Preslikavanje prozirnosti (alpha mapping)
- Preslikavanje svjetlosti (light mapping)
- Preslikavanje okoline (environment mapping)
- Preslikavanje neravnina (bump mapping)

# Poopćeno teksturiranje, projekcija

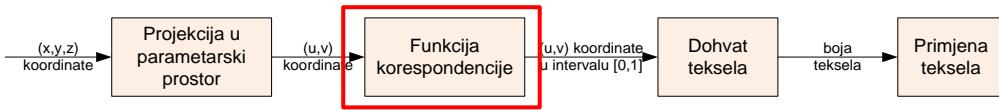


- Teksturiranje (texture mapping) = preslikavanje *uzorka* na geometriju predmeta
- Uzorak može doći iz bilo kakvog izvora, uklj. slike
- Uzorak se može koristiti na puno različitim načina, uklj. jednostavno preslikavanje uzorka na boju



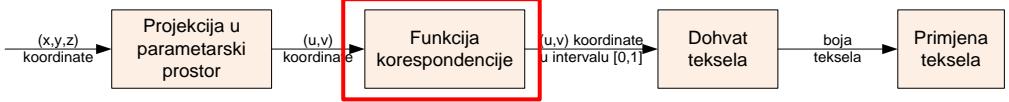
# Projekcija u parametarski prostor

- $u,v$  koordinate obično zadane za svaki vrh (prilikom modeliranja), te se interpoliraju
- Više tekstura na predmetu (multitexturing) = više parova  $u,v$  koordinata po vrhu
- 3D teksture  $(u,v,w)$  – npr. mramor, drvo
- 1D teksture – npr. bojenje terena po visini

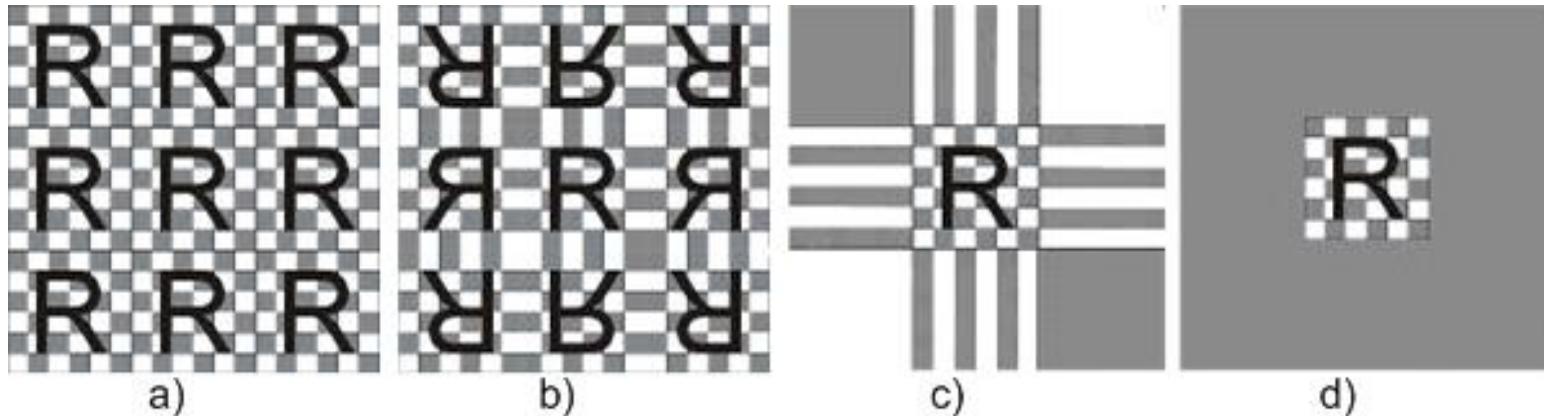


# Funkcija korespondencije (1/2)

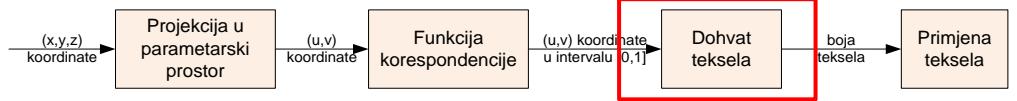
- Preslikavanje  $u,v$  koordinata u prostor tekstuure (vrijednost ograničena na  $[0,1]$ )
- 2D slika –  $(0,0)$  donji lijevi kut;  $(1,1)$  gornji desni kut
- Koordinate neovisne o dimenzijama – lako zamijeniti sliku (npr. manja razlučivost radi štednje memorije)



# Funkcija korespondencije (2/2)

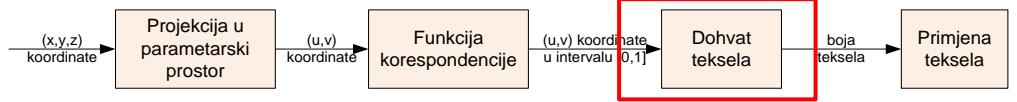


- Funkcije korespondencije:
  - Ponavljanje (wrap, repeat, tile)
  - Zrcaljenje (mirror)
  - Ponavljanje ruba (clamp)
  - Ograničavanje (border)
- Na  $u,v$  koordinate možemo primijeniti matričnu transformaciju u shaderu (npr. translacija ili rotacija teksture)



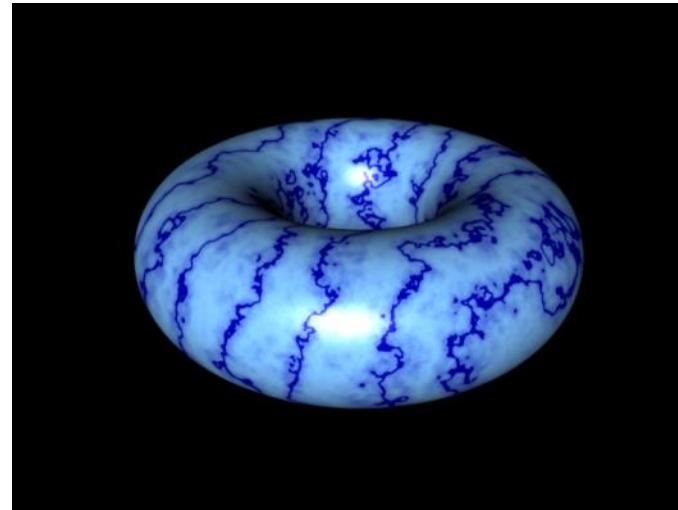
# Dohvat teksela (1/2)

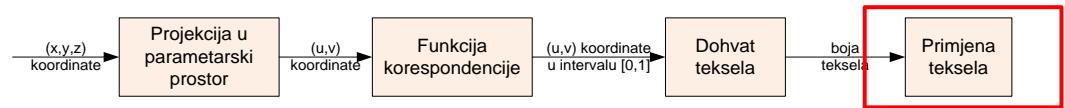
- = uzorkovanje teksture
- Teksel – točka teksture na u,v koordinatama:
  - RGB – 24-bit (8-bit po komponenti boje)
  - RGBA – 24-bit + 8-bit  $\alpha$  kanal
  - HDR – visoki dinamički raspon; više od 8-bit po komponenti
- (Ne mora se koristiti kao boja, nego npr. normala, parametar materijala...)



# Dohvat teksela (2/2)

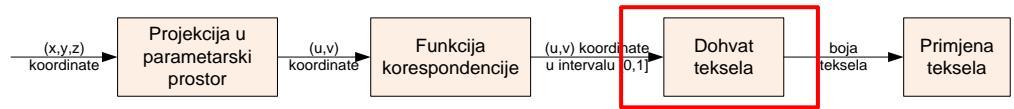
- Uzorkovanje = čitanje memorijске lokacije
- Moguće koristiti filtriranje – čitanje susjednih lokacija i interpolacija rezultata
- No, teksel se može i izračunati, npr. matematičkom operacijom (proceduralne teksture)





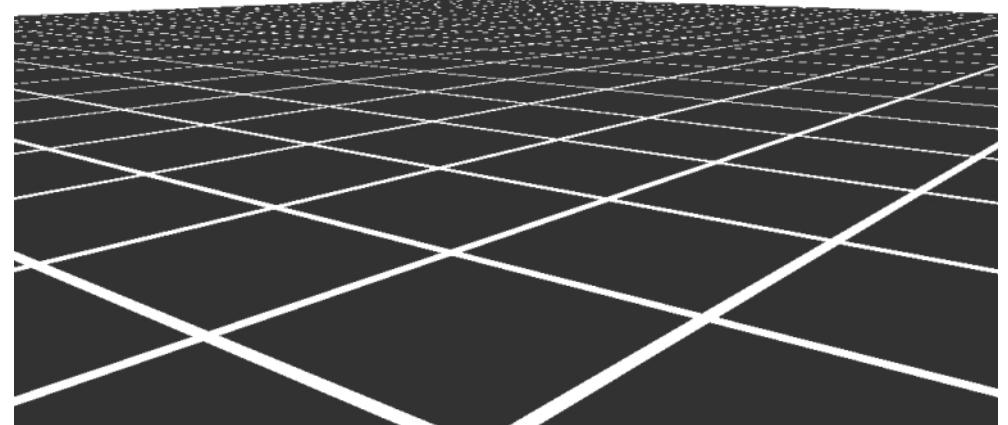
# Primjena tekscela

- Kako vrijednost tekscela utječe na konačnu boju točke koja se sjenča?
  - Izravna primjena kao boje
  - Težinsko miješanje s bojom osvjetljenja
  - Parametar materijala (difuzna ili spekularna tekstura, tekstura sjaja)
  - Normala (v. preslikavanje normala)
  - ...



# Filtriranje tekture

- Ako je tekstura = slika, javljaju se problemi poduzorkovanja ili naduzorkovanja (ovisno o udaljenosti od kamere)
- Idealno – korespondencija piksel-teksel 1:1
  - ◆ Slika uvećana (predmet bliže kamери) → efekt blokova
  - ◆ Slika umanjena (predmet dalje od kamere) → aliasing



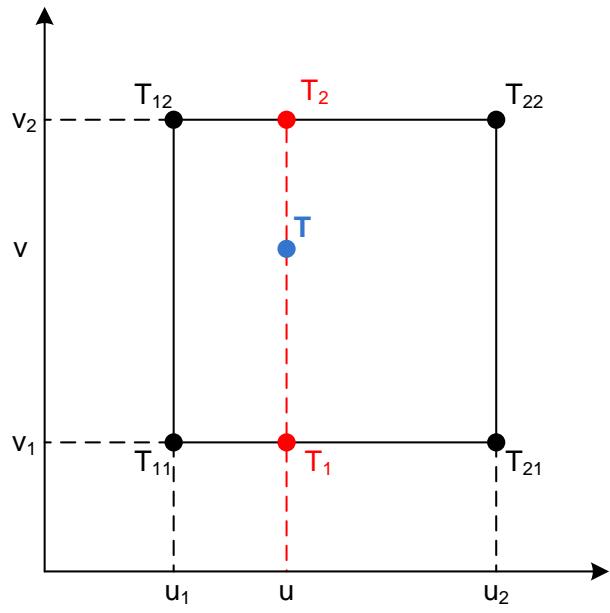
# Metoda najbližeg susjeda

- Dohvaća se teksel najbliži u,v koordinatama
- Naglašen efekt blokova

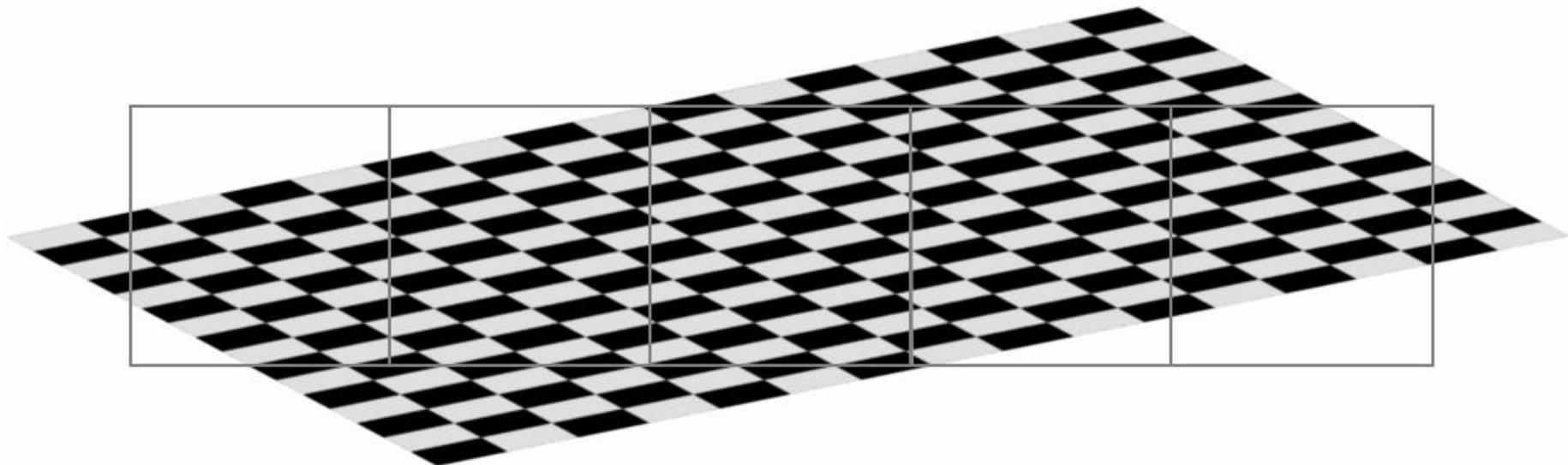


# Bilinearna interpolacija

- Interpolacija 2x2 najbližih texsela
- 2 uzastopne linearne interpolacije
- Zamućena slika, ublažen efekt blokova
- Sklopopovski podržana



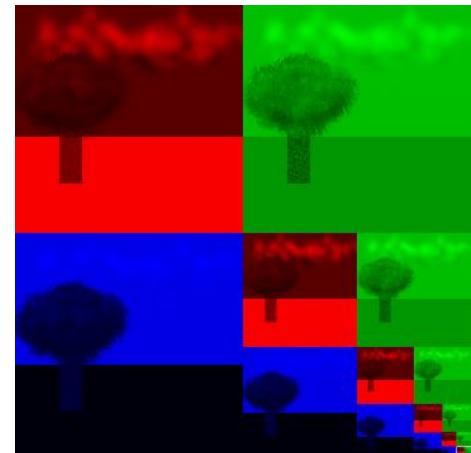
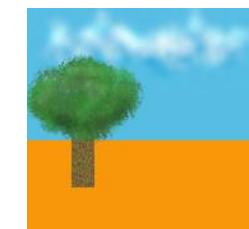
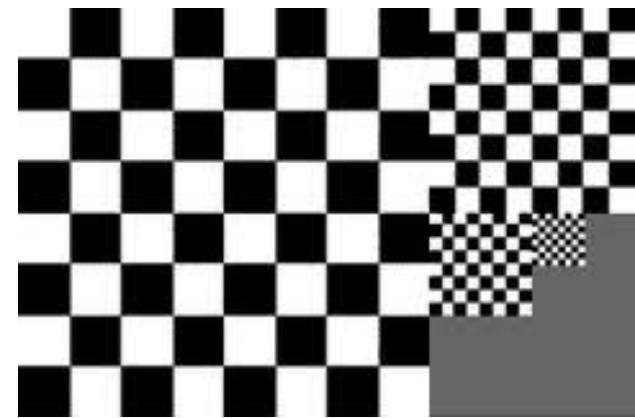
# Mipmapiranje (1/3)



- 1 piksel prekriva više teksele
- Ispravno bi bilo zbrojiti njihove utjecaje, ali ne možemo u stvarnom vremenu

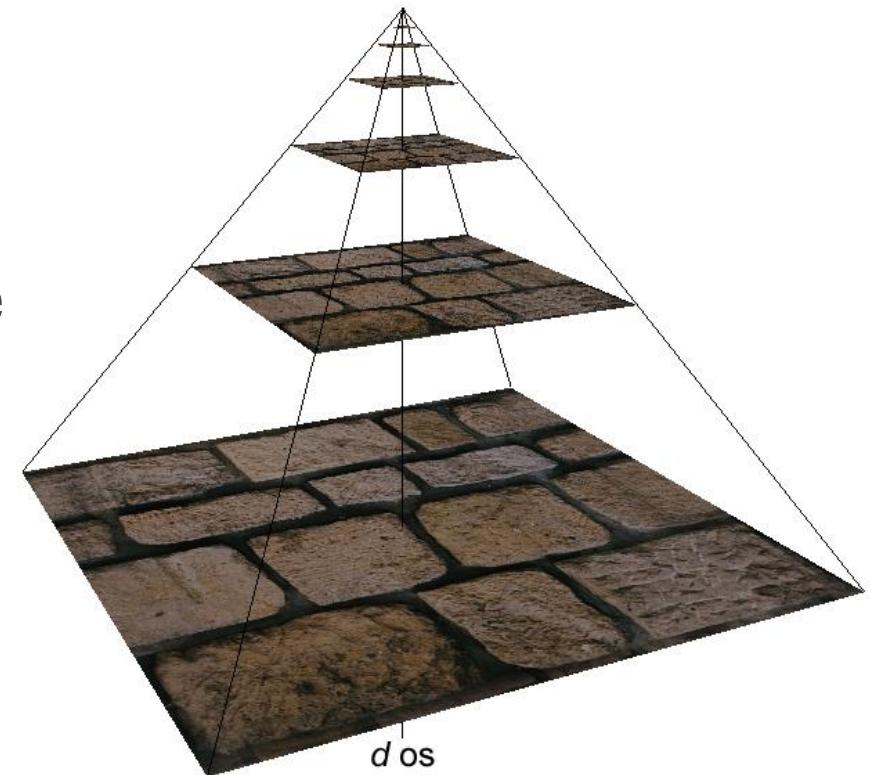
# Mipmapiranje (2/3)

- Mipmap – originalna slika + lanac umanjenih, filtriranih slika (svaka 2x manja od prethodne)
- Uzorkuju se texseli one slike čija veličina najbolje odgovara veličini piksela
  - ◆ Mipmap slike moguće pohraniti zajedno (datoteka veća za samo 1/3 jer  $\frac{1}{4} + \frac{1}{16} + \frac{1}{64} + \frac{1}{256} + \dots \rightarrow \frac{1}{3}$ )
  - ◆ Može ih se generirati dinamički (ugrađena sklopovska podrška)



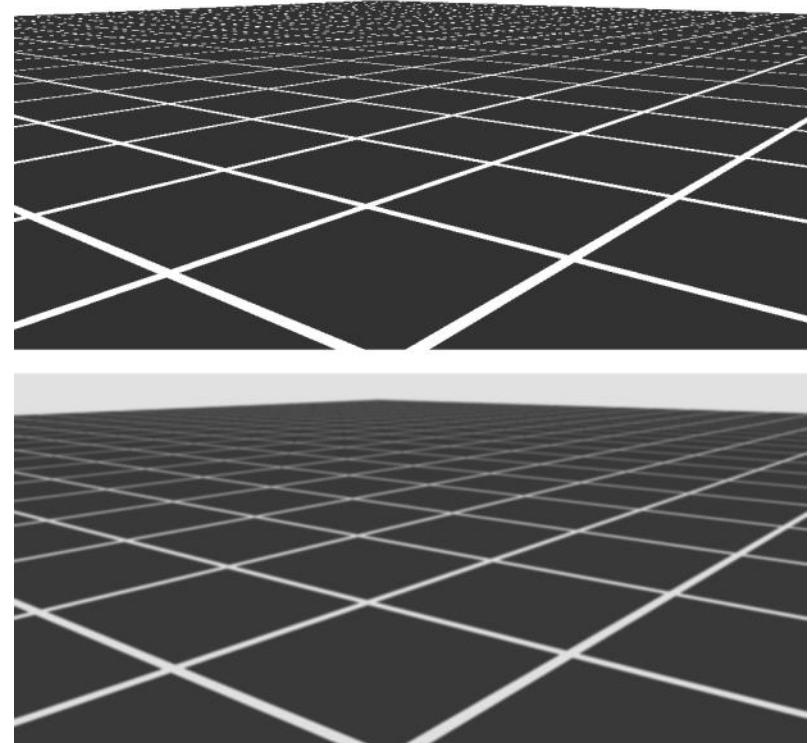
# Mipmapiranje (3/3)

- Parametar  $d$  (razina detalja, level od detail) – mjera prekrivanja bloka teksela pikselom
- Za uzorkovanje mipmap teksture trebamo  $u, v$  i  $d$  (veći  $d \rightarrow$  odabire se manja slika)
- 2 načina izračuna  $d$ :
  1. Najdulja stranica četverokutaprojekcije piksela na teksturu
  2. Iz diferencijala  $u$  i  $v$  duž osi zaslona  $x$  i  $y$  ( $du$  i  $dv$ )



# Trilinearna interpolacija

1. Odredi vrijednost  $d$
2. Pomoću  $d$ , odredi dvije susjedne mipmap slike
3. Uzorkuj obje slike na  $u, v$  (uz korištenje bilinearne interpolacije)
4. Linearno interpoliraj rezultat po  $d$



# Animacija tekstuра

- Promjena slike tekstuра u vremenu
1. Vremenski niz slika koje se izmjenjuju:
    - Efekti vatre, dima, eksplozija...
    - Slike pohranjene kao 3D tekstura (w koord. = vrijeme) ili polje tekstuра (SM4.0+)
    - Gladak prijelaz između tekstuра težinskim miješanjem (npr. mrštenje lika)
  2. Animiranje teksturnih koordinata:
    - Translacija, rotacija, skaliranje i smicanje koordinata u programu za sjenčanje
    - Tok duž površine (slap, rijeka, kapi vode), efekti dima i vatre...



# Preslikavanje materijala

- Umjesto boje, teksture pohranjuju parametre materijala
- Parametri definirani za svaku točku predmeta umjesto konst. za čitav predmet
- Teksture materijala (za Phongov model osv.):
  - Tekstura difuzne boje (diffuse color map) –  $k_d$
  - Tekstura spekularne boje (specular color map) –  $k_s$
  - Tekstura sjaja (gloss map) –  $n$

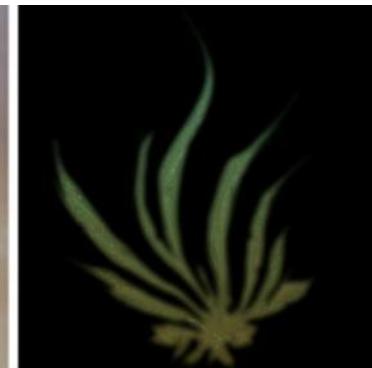
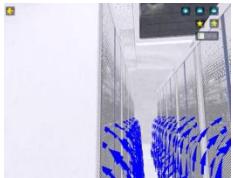
# Preslikavanje prozirnosti (1/2)

- Alpha mapping
- $\alpha$  kanal teksture definira prozirnost u svakoj točki predmeta
- Efekt naljepnice:
  - Slika je neki nepravokutan oblik (npr. krug), ostali dijelovi prozirni
  - Moguće dodavati u scenu dinamički rupe od metaka, mrlje od krvi itd.)



# Preslikavanje prozirnosti (2/2)

- Primjena prozirnosti na sam predmet:
  - Predmet proziran gdje je  $\alpha = 0$
  - Prikaz složenog predmeta jednim poligonom (npr. vegetacija)
  - Uz animaciju teksture – plamen, dim, eksplozije...



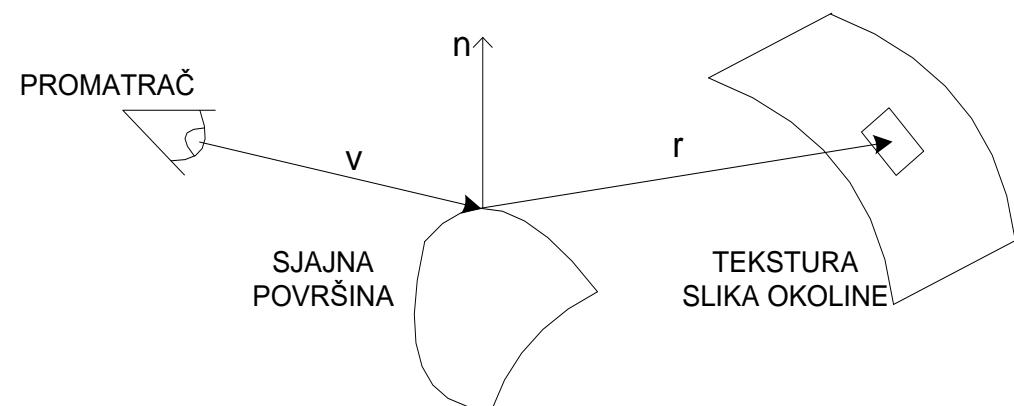
# Preslikavanje svjetlosti (light mapping)

- Simulacija efekata difuzne svjetlosti
- Osvjetljenje se računa unaprijed (npr. isijavanje) i pohranjuje u teksture svjetlosti (light map)
  - ◆ (Moguće zato što difuzna svjetlost ne ovisi o položaju promatrača)
  - ◆ Teksture svjetlosti koriste se u realnom vremenu



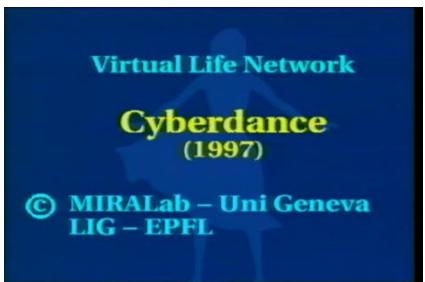
# Preslikavanje okoline (environment mapping)

- Simuliranje zrcaljenja na oblim površinama
- Ideja – slika okoline (gledano iz pozicije predmeta) pohrani se u teksturu, koja se „lijepi“ na predmet u realnom vremenu
- u,v koordinate ne možemo izračunati unaprijed, zrcaljenje ovisi o položaju promatrača (sjetite se spekularnog odsjaja!)



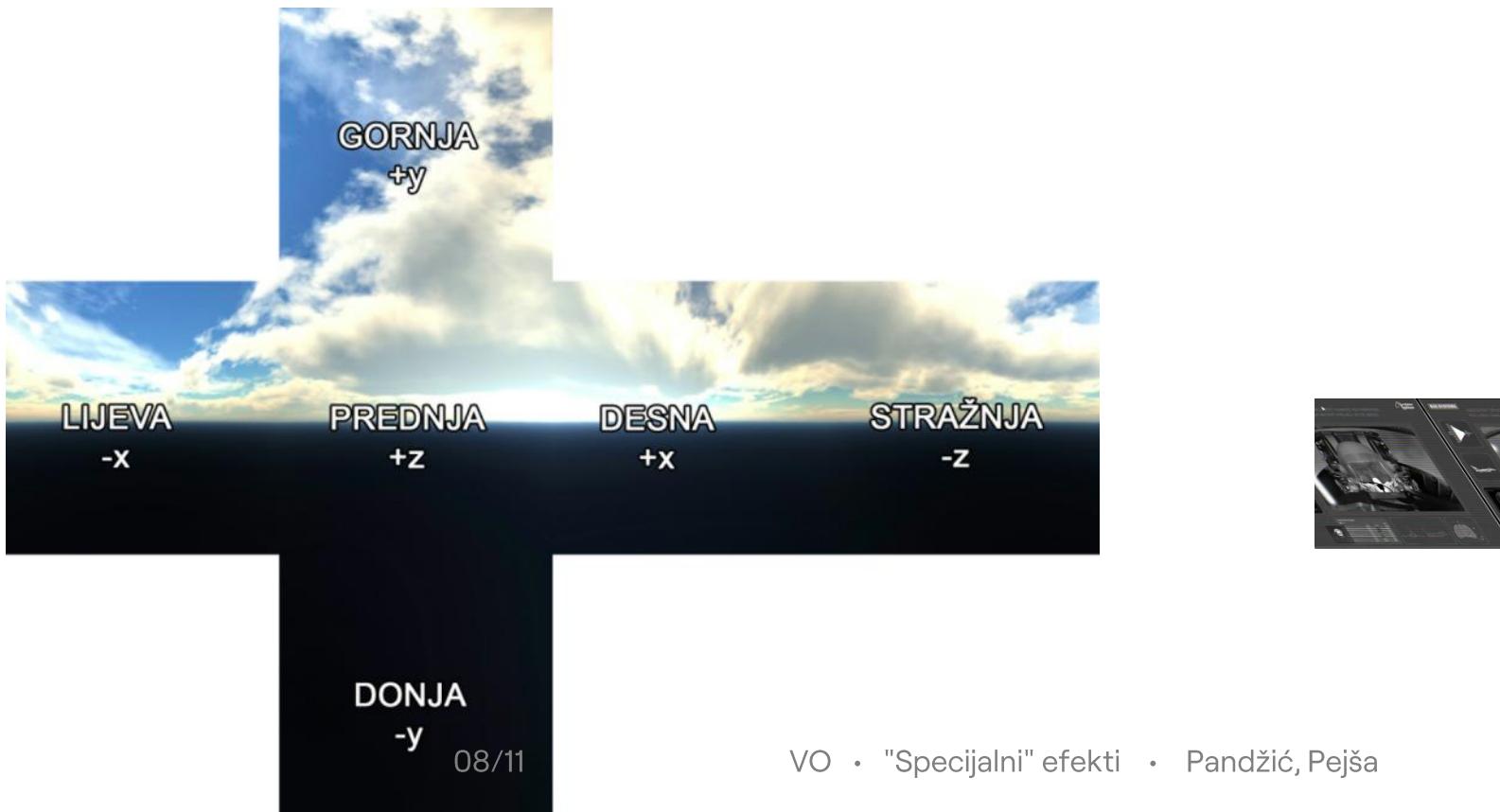
# Kuglasto preslikavanje

- Projekcija okolina na kuglu oko predmeta



# Kockasto preslikavanje (1/2)

- Projekcija okoline na 6 strana kocke oko predmeta
- Rezultat – tekstura od 6 dijelova (cube map)

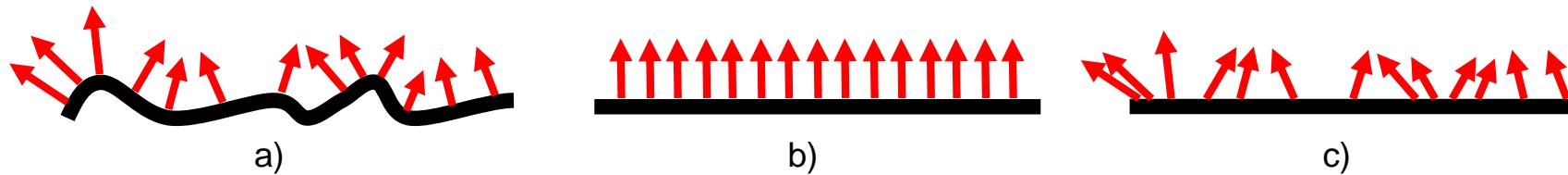


# Kockasto preslikavanje (2/2)

- ◆ Postupak iscrtavanja:
  1. Vektor r se normalizira
  2. Određuje se najveća komponenta r (x, y ili z), te njen predznak; iz njih se određuje koju stranu uzorkovati
  3. Uzorkuj odgovarajuću stranu (u,v koordinate – 2 manje komponente r)
    - (U jezicima za sjenčanje – texCUBE, textureCube)
- ◆ Prije SM4.0 – 6 prolaza za iscrtavanje kockaste teksture
- ◆ SM4.0+ – dovoljan jedan prolaz, ako umnožimo geometriju 6x u procesoru geometrije

# Preslikavanje neravnina (bump mapping)

- Skup tehnika za simuliranje neravnih, hrapavih površina
- Phongov model osvjetljenja – difuzna i spekularna komponenta ovise o normali
- Poremećajem normale dobivamo privid neravnina
- Tekstura neravnina – pohranjuje podatke potrebne za poremećaj normala
- (Iluzija se gubi pod oštrim kutem)



# Preslikavanje normala (normal mapping)

- Tekstura neravnina pohranjuje poremećene normale – tekstura normala
- Uzorkuje se u procesoru točaka, dobivena normala se koristi za proračun osvjetljenja



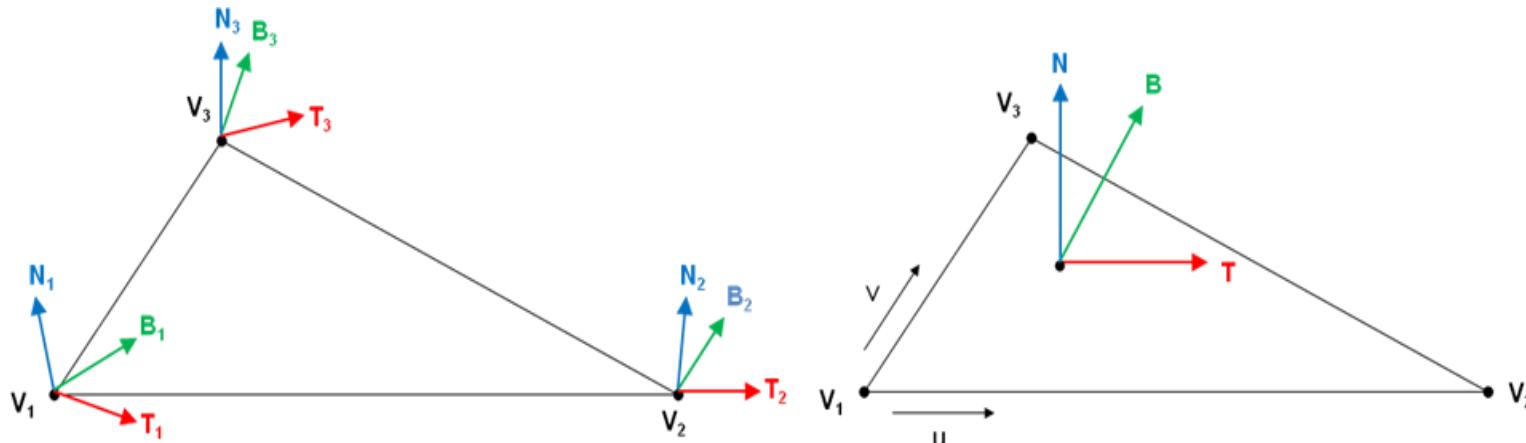
# Koordinatni sustav za normale

- U kojem koord. sustavu zadati normale?
- Globalni koordinatni sustav:
  - Nepraktičan, ne smijemo pomicati niti deformirati predmet
- Koordinatni sustavi predmeta:
  - Moguće rigidne transformacije
  - Nije moguća deformacija predmeta (npr. skinning kod likova)
- Najbolje rješenje – koordinatni sustav tangente!

# Koordinatni sustav tangente (1/2)

- Zadan trima vektorima:
  - Vektor tangente T (na površinu predmeta)
  - Vektor bitangente B (na površinu predmeta)
  - Vektor geometrijske normale N
- N već znamo odrediti (okomit na poligon)
- T i B – vektori u smjeru rastućih u,v koordinata:
  - a. Najčešće zadani u svakom vrhu (kao i sama normala), te se interpoliraju duž površine
  - b. Mogu se računati dinamički u proc. točaka iz diferencijala u i v (funkcije ddx i ddy)

# Koordinatni sustav tangente (2/2)



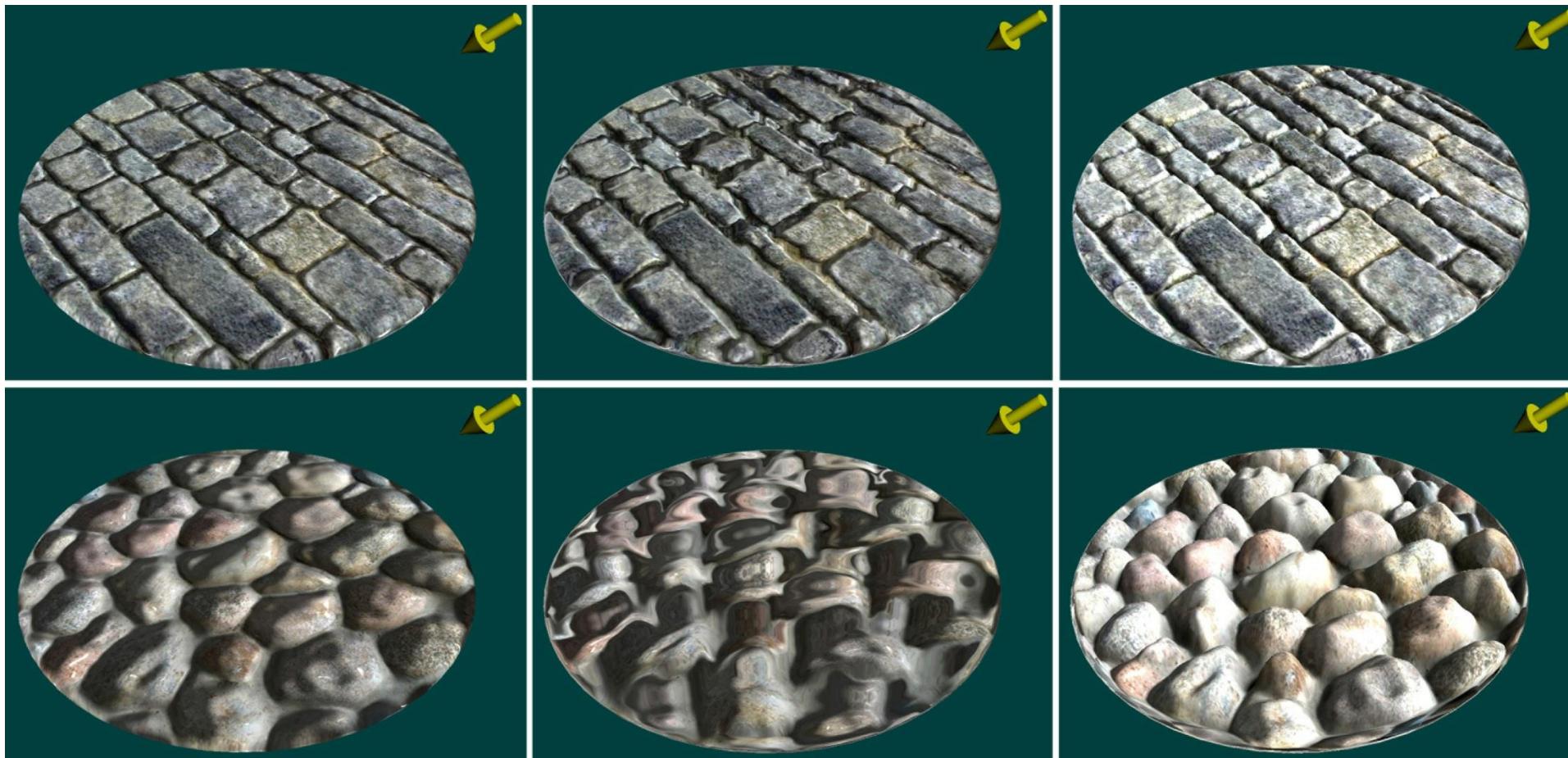
- Koord. sustav tangente varira duž površine – normale ostaju ispravne prilikom deformacija
- $T$  i  $B$  općenito nisu okomiti  $\rightarrow$  distorzija normale (no ne predstavlja problem)
- Pitanje: Zašto je tekstura normala plava?

# Preslikavanje neravnina i okoline

- Environment bump mapping
- u,v koordinate za uzorkovanje teksture okoline odredimo iz poremećene normale
  - ◆ Rezultat – zrcaljenje na neravnoj površini (npr. vode)
  - ◆ Povijesna važnost (1999.) – prvi primjer zavisnog uzorkovanja tekstura u GPU



# Preslikavanje paralakse, reljefa

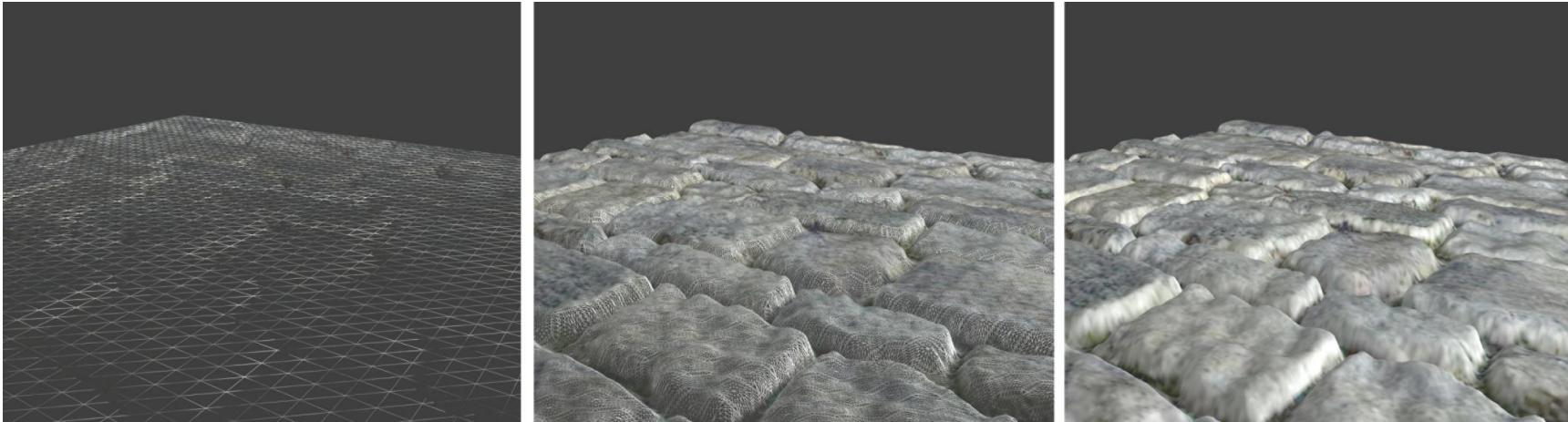


# Preslikavanje pomaka (displacement mapping) (1/2)

- Prethodne tehnike – iluzija neravnina, trikovi sa sjenčanjem
- Preslikavanje pomaka – tekstura visina se koristi za deformiranje same geometrije predmeta
- Jednostavnija izvedba (SM3.0):
  - Ulaz – „ravna“ geometrija
  - (Vertex shader) Za svaki vrh se uzorkuje tekstura visina, te se vrh pomiče
  - Nije „pravi“ bump mapping (ne povećava detaljnost geometrije)
  - Korisno za animaciju neravnina (npr. valovi na moru)

# Preslikavanje pomaka (displacement mapping) (2/2)

- „Pravo“ preslikavanje pomaka:
  - (Geometry shader) Uzorkuje se tekstura visina, te se stvaraju dodatni trokuti za neravnine
  - Potrebna sklopovska podrška za teselaciju (SM5.0)
- Odlični vizualni rezultati
- Detekcija presjeka i sudara postaje problem – vrši se u aplikaciji, prije unošenja pomaka

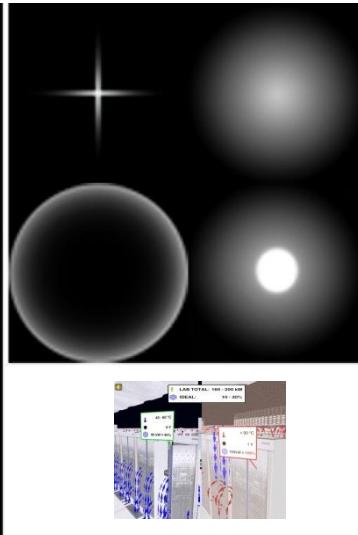
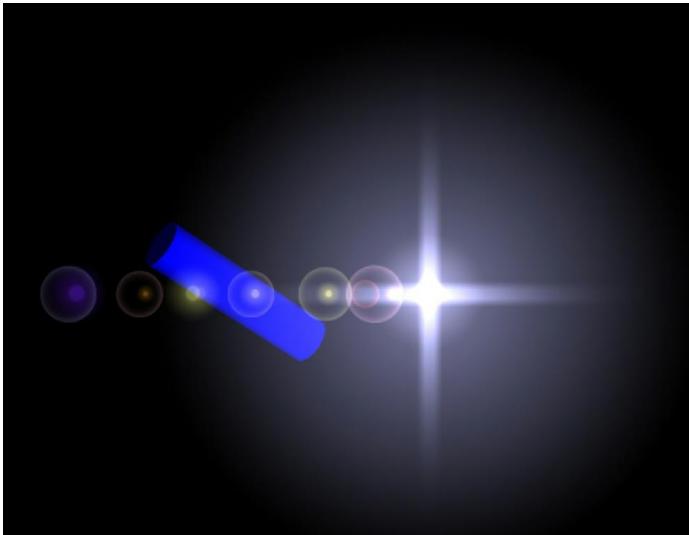


# Tehnike panoa (billboard)

- Za automatsku orijentaciju predmeta prema kameri
- Predmet se uvijek vidi iz istog kuta
- Npr. tekst, vegetacija, oblaci...
- Na pravokutnik (pano) se lijepi 2D slika predmeta (sprite)
- Često se koriste preslikavanje prozirnosti i/ili animacija tekstu

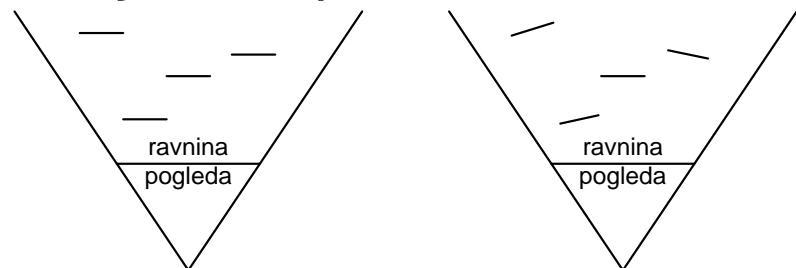
# Pano poravnat sa zaslonom

- Uvijek ista orijentacija prema zaslonu
- Normala uvijek okomita na ravninu projekcije
- Vertikalna os (up axis) = vertikalna os kamere
- Primjeri – tekst, odbljesak na objektivu (lens flare), sustavi čestica...



# Globalno orijentirani pano

- Vertikalna os djelomično globalno orijentirana – predmet ne može rotirati oko osi pogleda
- Za predmete koji su fizički prisutni u sceni, a nisu radijalno simetrični – oblaci, dim, eksplozije...
- Normala na pano:
  - a. Okomita na ravninu proj.  
– rubni panoi izgledaju „iskriviljeno”
  - a. Usmjerena prema kameri



# Osni pano

- Predmet prema kameri rotira oko fiksne osi
- Primjer – simulacija drveća

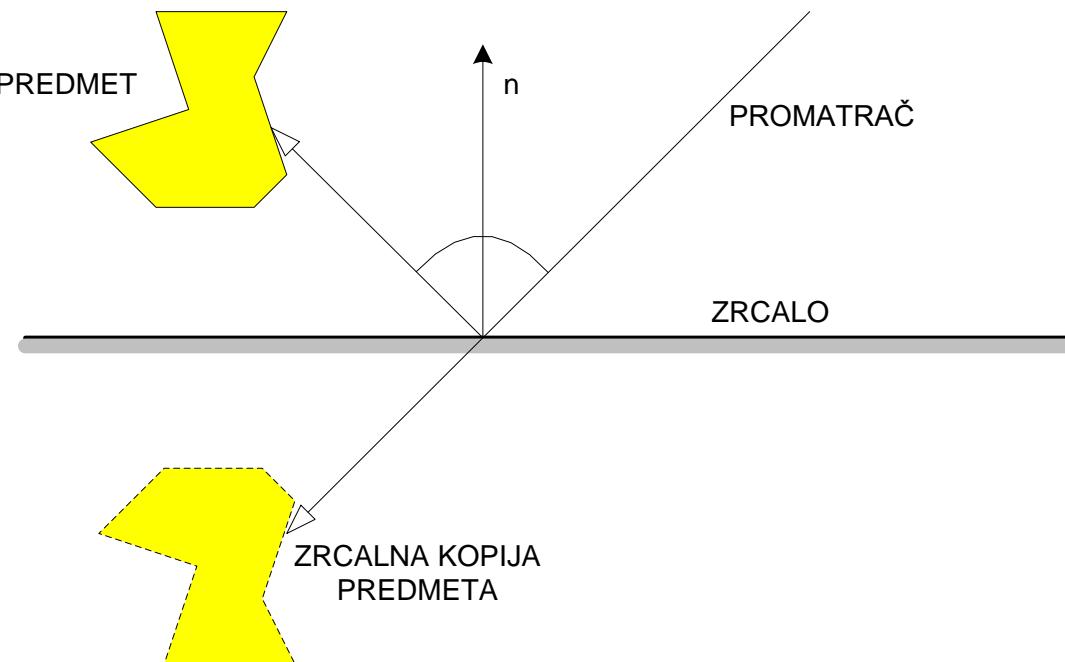


# Zrcaljenje

- Jednostavno – zraka se odbija simetrično na normalu (idealno zrcalo)
- Praćenje zrake (ray tracing) daje dobro rješenje, ali skupo
- Preslikavanje okoline prikladno samo za udaljene predmete

# Zrcaljenje na ravnini (1/3)

- Pojednostavljen ali često koristan slučaj: jedno zrcalo u sceni, i to ravno
- Ideja: konstruiramo kompletну kopiju scene zrcaljenjem oko ravnine zrcala, zatim crtamo scenu i kopiju

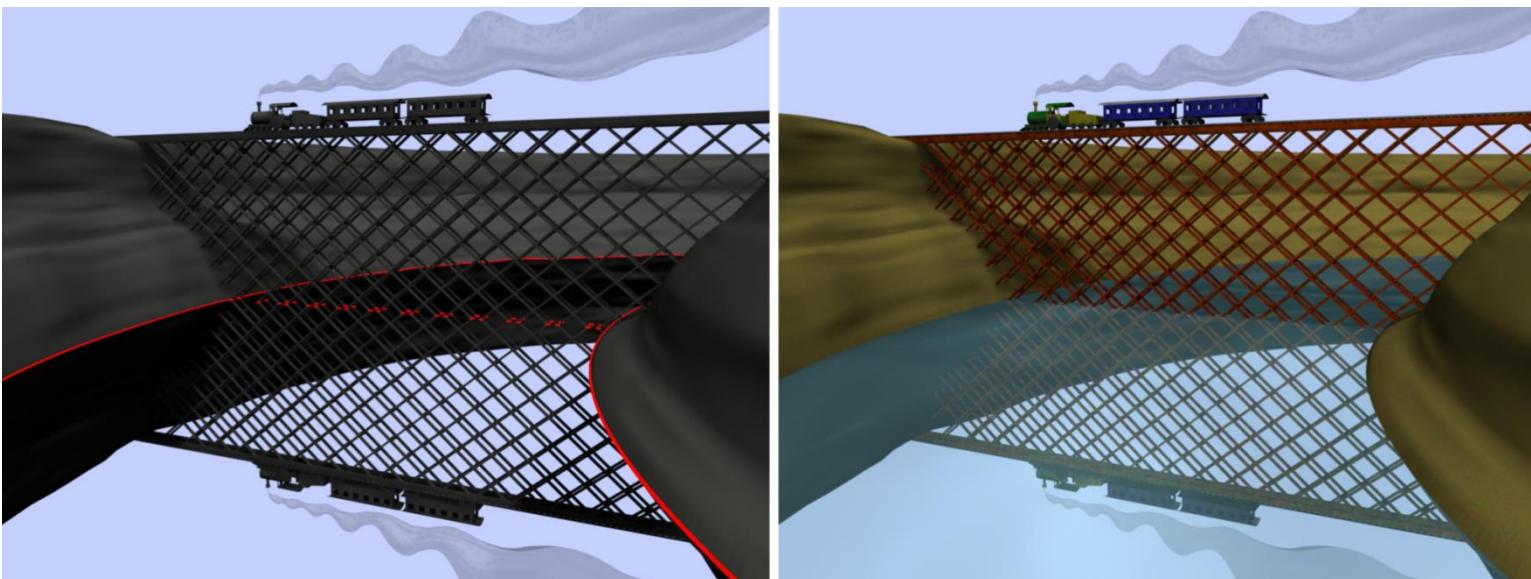


# Zrcaljenje na ravnini (2/3)

- Najjednostavniji slučaj: ravnina  $y=0$ 
  - Zrcaljenje izraženo matricom  $S(1, -1, 1)$
- Za općenu zrcalnu plohu zadalu točkom  $P$  i normalom  $N$ 
  - $F = T(-P) R(N, (0, 1, 0))$  preslikava plohu na  $y=0$
  - Konačna zrcalna matrica je  $M = F S(1, -1, 1) F^{-1}$

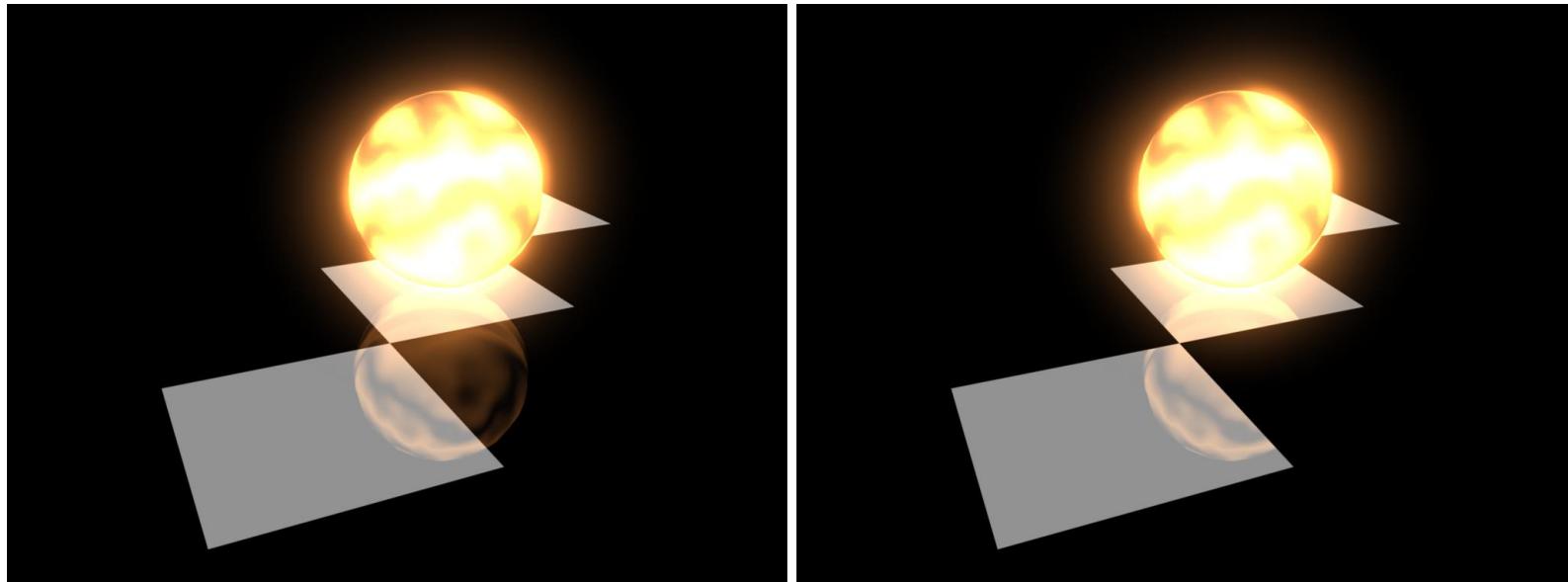
# Zrcaljenje na ravnini (3/3)

1. IsCRTavanje zrcaljene geometrije (transformirane matricom M)
2. IsCRTavanje izvorne geometrije
3. IsCRTavanje zrcalne plohe
  - ◆ Faktor prozirnosti djeluje kao faktor zrcaljenja



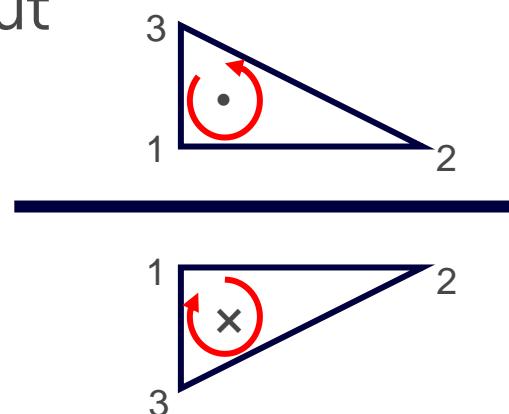
# Ograničenje zrcaljenja na dio ravnine

- IsCRTavanje zrcalne plohe u spremnik predloška
- IsCRTavanje zrcaljene geometrije samo po predlošku
- Ostatak postupka isti



# Ostali problemi

- Geometrija ispod plohe se ne smije zrcaliti
  - Sve ispod zrcala treba odrezati, direktno ili korištenjem proizvoljne odrežujuće plohe (arbitrary clipping plane)
- Odbacivanje stražnjih poligona
  - Zrcaljenje obrće redoslijed vrhova poligona → odbacivanje daje pogrešan rezultat!
  - Isključiti odbacivanje, ili zadati obrnut redoslijed vrhova



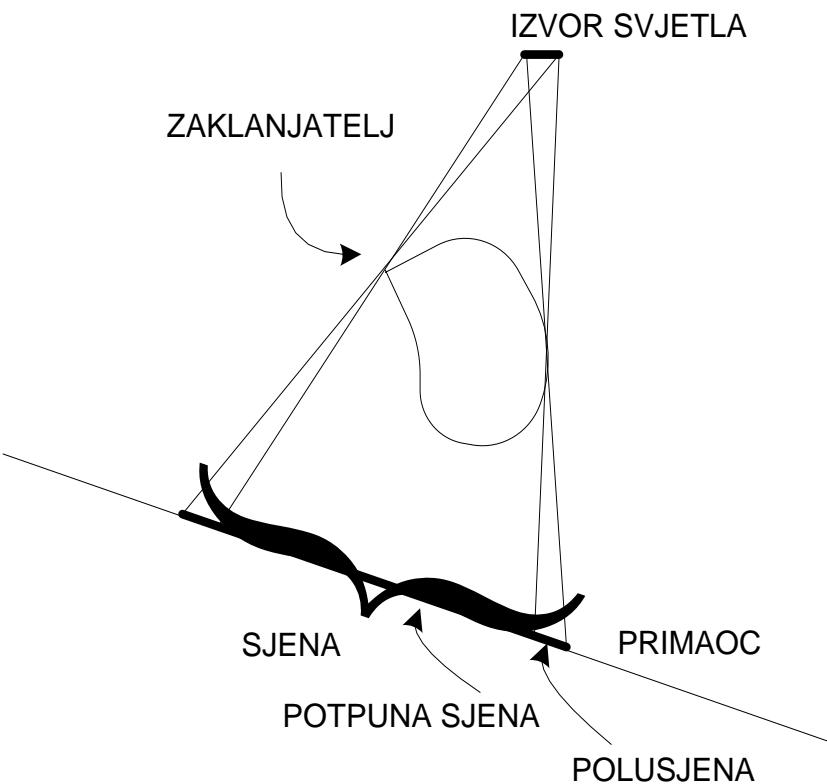
# Sjene

- Daju osjećaj o položaju predmeta (čak i kada su jako pojednostavljene)
- Sjena nastaje jer predmet zaklanja svjetlo
- Jednostavno, ali:
  - U sceni postoji velik broj potencijalnih zaklanjatelja
  - Zaklonjenost treba utvrditi u svakoj točki – skupo
  - Može biti više svjetala



# Nomenklatura sjena

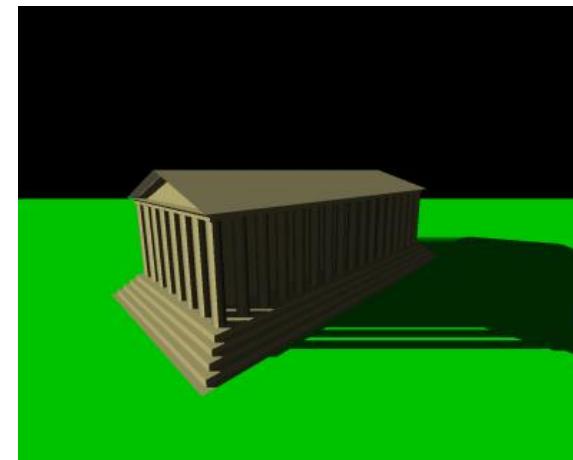
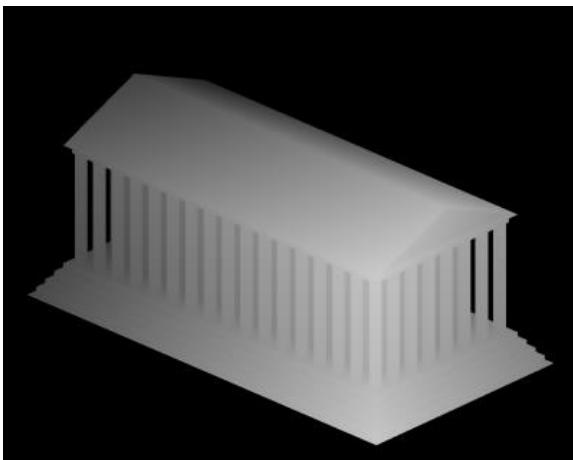
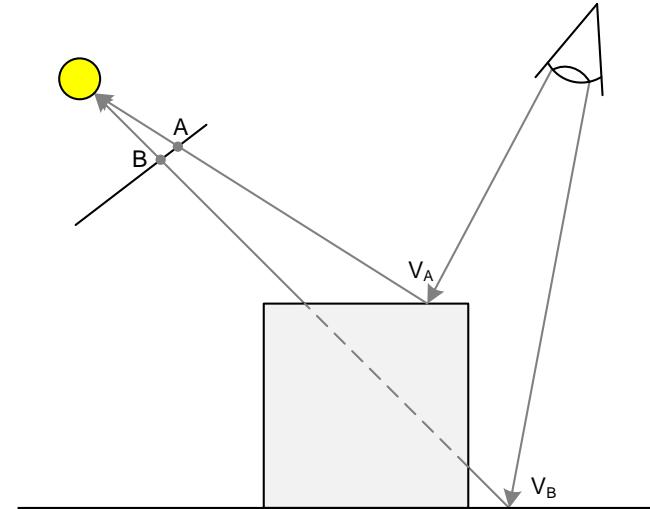
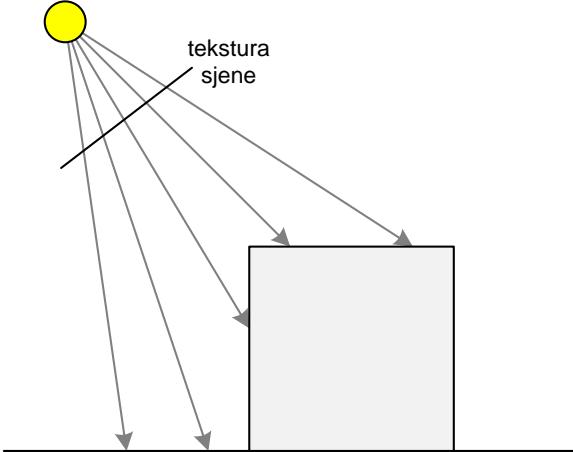
- Sjene su mekane jer izvori svjetla nisu točkasti
- Oštре sjene u prirodi rjeđe, no u grafici se često koriste (bolje išta nego ništa)



# Metoda teksture sjena (shadow map) (1/3)

- Ideja – samo točke vidljive iz perspektive svjetla trebaju biti osvijetljene
- Bacanje zrake prema svjetlu i provjera presjeka s geometrijom? – skupo, nepraktično!
- Umjesto toga koristimo *teksturu sjena*

# Metoda teksture sjena (2/3)



# Metoda teksture sjena (3/3)

1. Iscrtati scenu iz perspektive svjetla u Z-spremnik (samo dubine) → tekstura sjena (TS)
2. Iscrtati scenu iz perspektive kamere:
  - U svakoj točki uzorkovati TS
  - Ako je  $z > z_{TS}$  → točka je u sjeni, samo ambijentalno osvj.
  - (Prethodno točku transformirati u KS svjetla)
  - Samo 1 projekcijska ravnina za TS:
    - Prikladno za usmjereni svjetla (npr. Sunce)
    - Za točkasta svjetla potrebna kockasta TS (generira se u 6 prolaza)
  - Dobra skalabilnost – vrijeme izvođenja linearno s brojem poligona