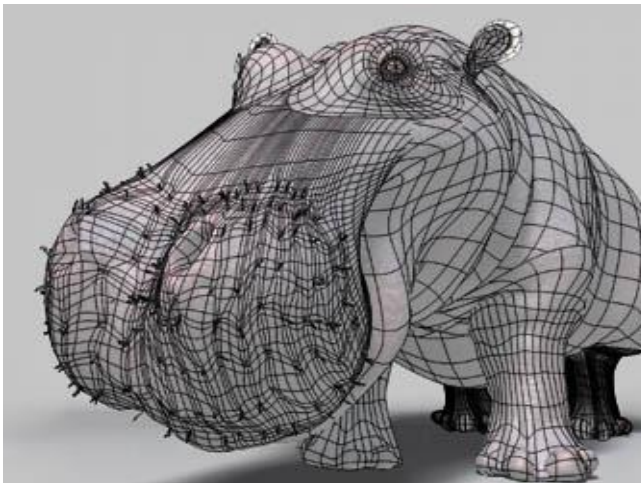


10 Preslikavanje i generiranje teksture

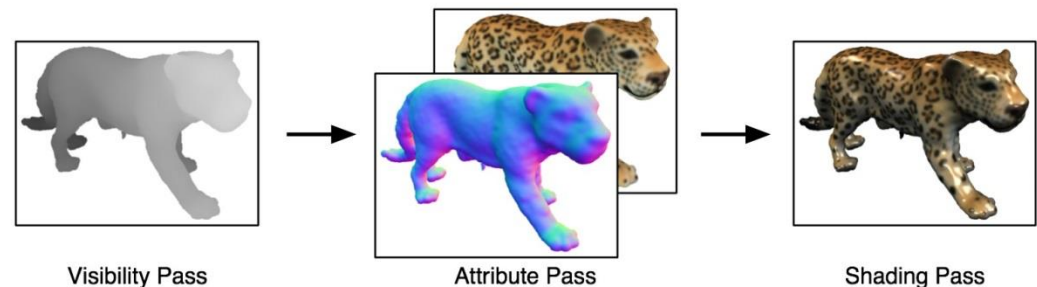
- povećavanje količine informacije bez povećavanja broja poligona
- možemo odvojiti postupak siječanja od postupka preslikavanja teksture i stopiti rezultat ili uvrstiti u formulu za osvjetljenje kao boju slikovnog elementa

- 1974 Ed Catmull

$$I_R = k_a O_{dR} I_{aR} + k_d O_{dR} I_{dR} \frac{\vec{l} \cdot \vec{n}}{d_L + k}$$



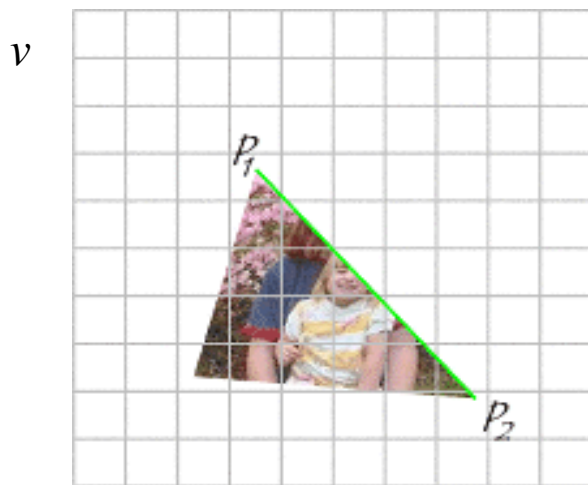
- postupak preslikavanja teksture
 - na poligon
 - na krpicu površine
- preslikavanje teksture na mrežu poligona
 - foto teksture
 - projekcijsko preslikavanje
 - ravnina, kocka, cilindar, sfera
 - preslikavanje okoliša
- generiranje teksture
 - proceduralno generirane teksture
- teksture za posebne namjene
 - mape svjetla
 - volumne tekstura
 - animirajuće teksture



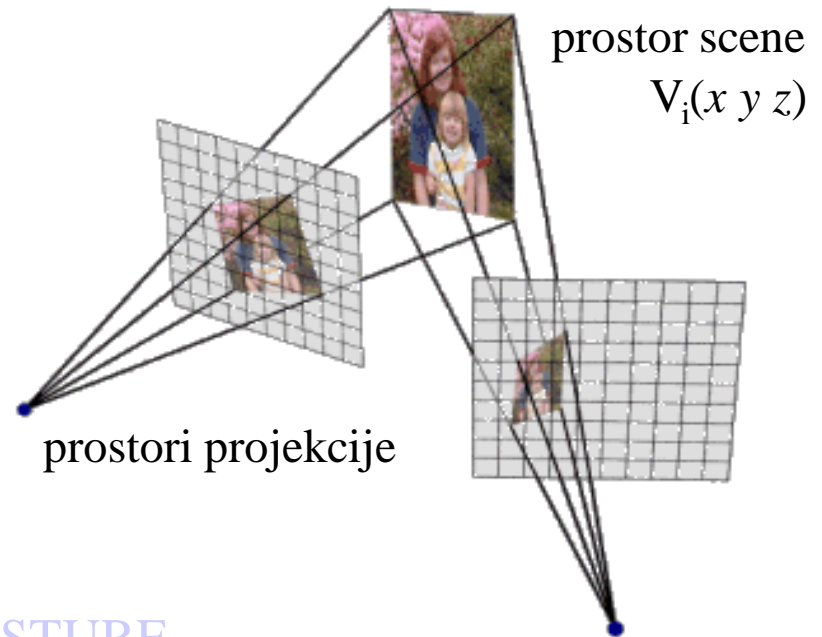
10.1 Postupak preslikavanja teksture

10.1.1 Postupak preslikavanja na poligon

- za svaki vrh poligona $V_i(x, y, z)$ potrebno je odrediti (u, v) koordinate u prostoru teksture, na taj način je određeno preslikavanje teksture 2D na poligon 3D
- poligon 3D se projicira u prostor projekcije

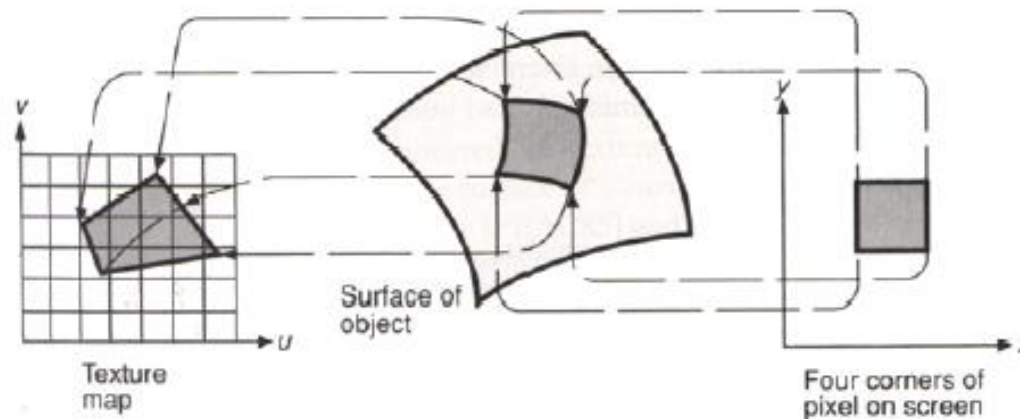


prostor teksture
texel - element teksture



• TEKSTURE

- za svaki slikovni element potrebno je odrediti područje na 3D objektu koje se preslikava i pripadni dio u prostoru tekstone
 - nije dobro jednostavno uzeti samo središnji element iz područja tekstone
- dio tekstone nije zahvaćen, dohvaćanje nepostojeće tekstone – proširivanje tekstone (zrcalno, popločavanje ...)



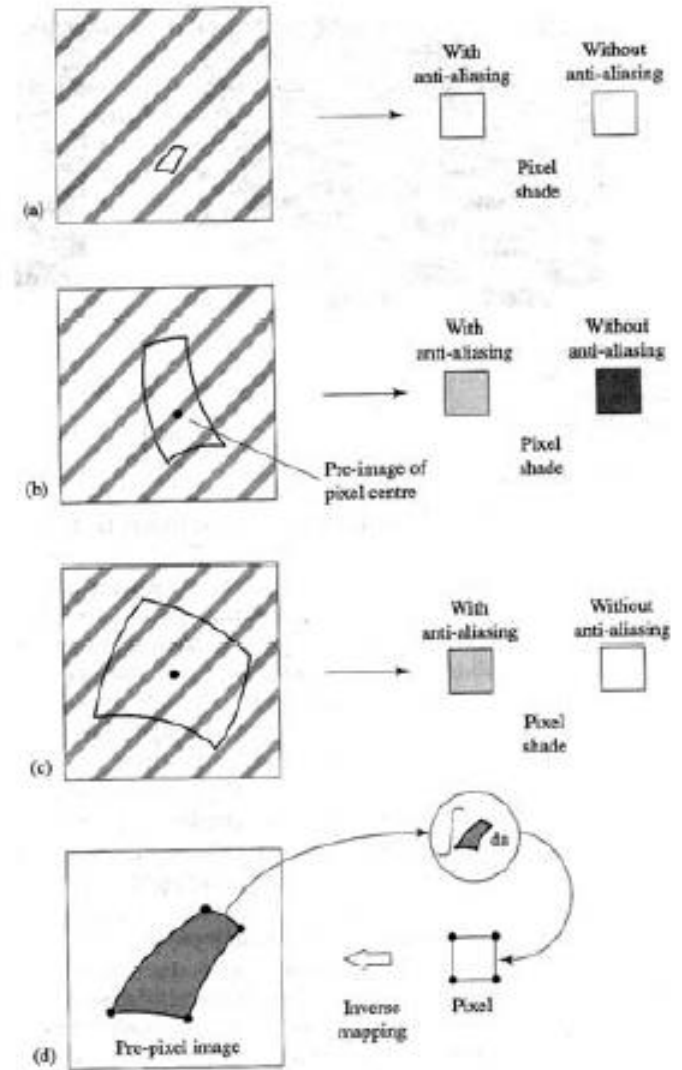
prostor tekstone
” pixel’s footprint”

prostor scene

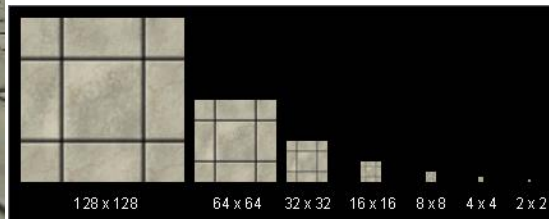
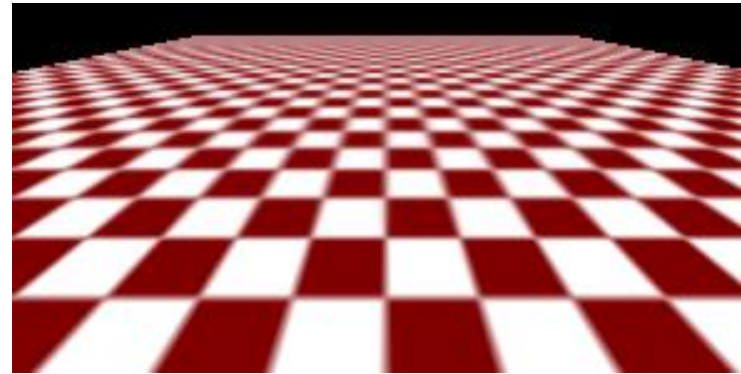
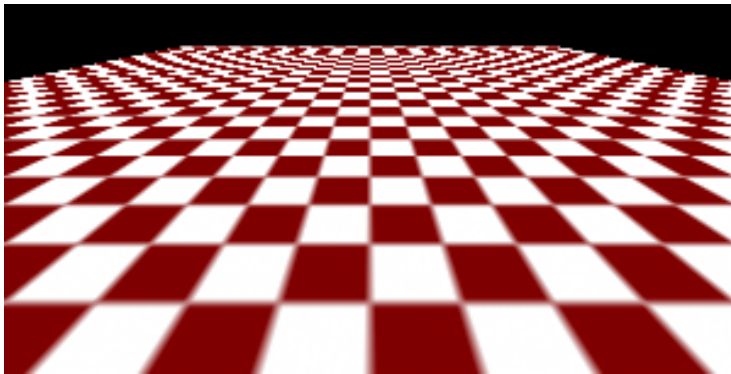
prostori projekcije

- dio u prostoru teksture može obuhvatiti
 - “nijedan”
 - nekoliko texel-a
 (engl. sub-sampling super-sampling)

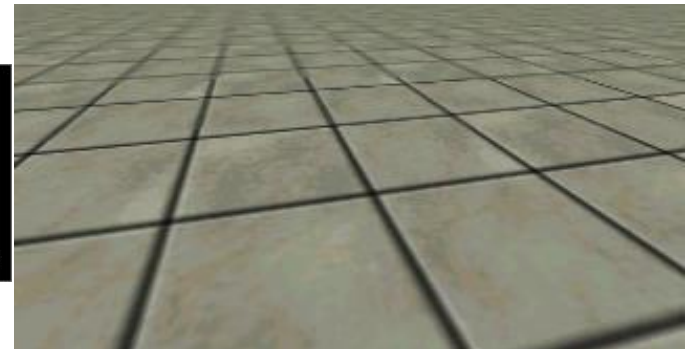
http://threejs.org/examples/#webgl_materials_texture_filters



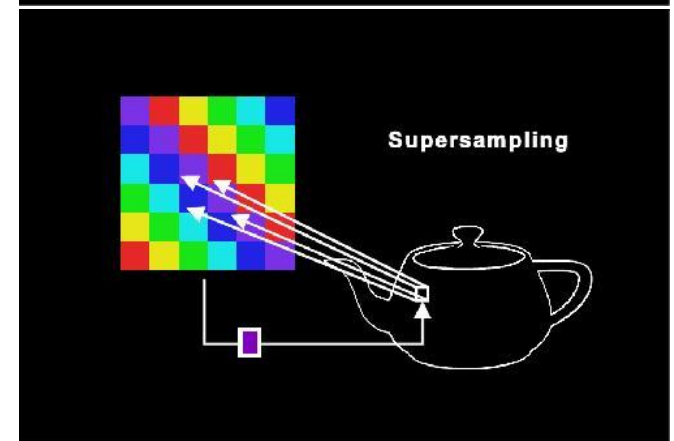
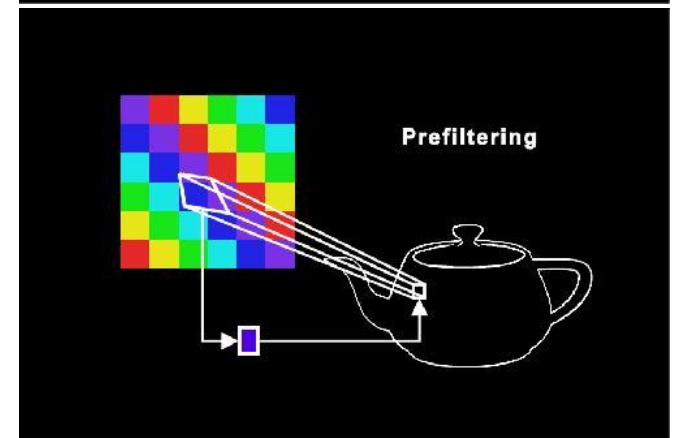
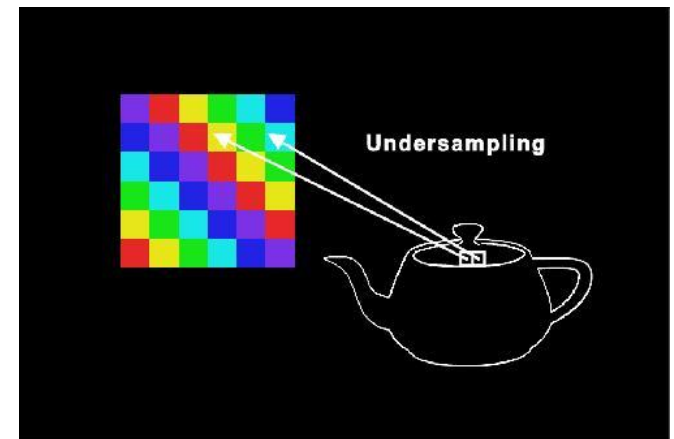
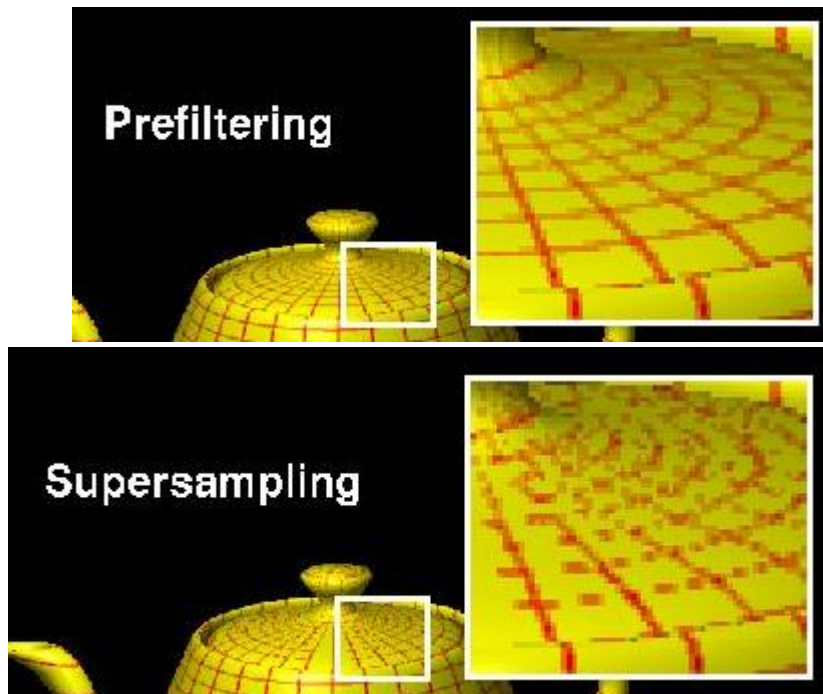
- utjecaj alias učinaka na teksturu
 - dio poligona blizu promatrača – jedan slikovni element – između texel-a
 - dio poligona daleko od promatrača – jedan sl. element – puno texel-a
 - <http://www.realtimerendering.com/udacity/?load=demo/unit8-mipmapping.js>
- postupci filtriranja, mip-map teksture
- anizotropni učinak <https://math.hws.edu/graphicsbook/source/webgl/anisotropic-filtering.html>
http://pages.cs.wisc.edu/~lizy/mrdoob-three.js-ef5f05d/examples/webgl_materials_texture_anisotropy.html
 - <http://mikolajysenko.github.io/voxel-mipmap-demo/> (Teapot, mali kut)
 - <http://www.realtimerendering.com/udacity/?load=demo/unit8-anisotropy.js>



Ž. M. ZEMRIS, FER
Mip-mape



- prilikom popunjavanja poligona u ravnini projekcije teksturom možemo “promašiti” neke elemente tekstuure
 - pre-filtriranje područja koje se preslika u element tekstuure, računa se prosječna vrijednost slikovnih elemenata tekstuure
 - povećano uzorkovanje (*supersampling*) po svakom slikovnom elementu u projekciji dohvaća 4 (ili više) elemenata tekstuure (opet možemo ‘promašiti’)

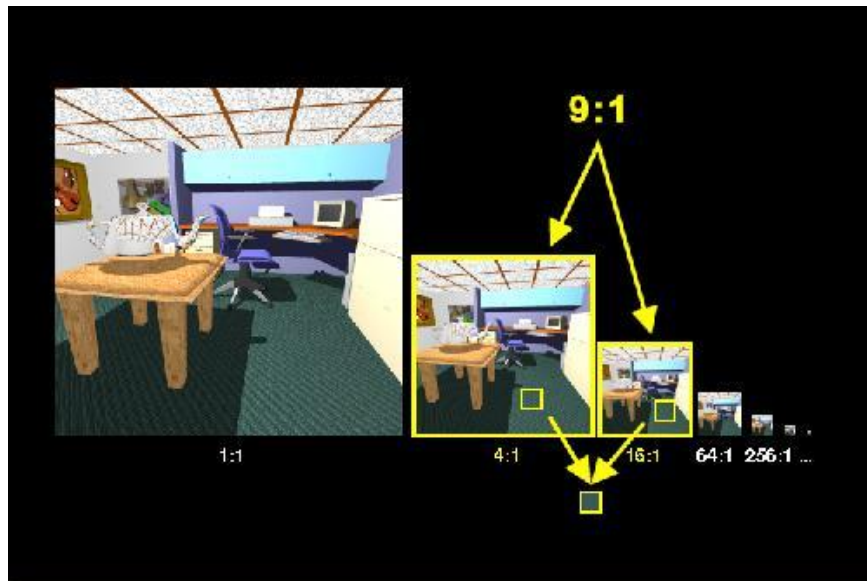


- Mip-mape (potencije broja 2)

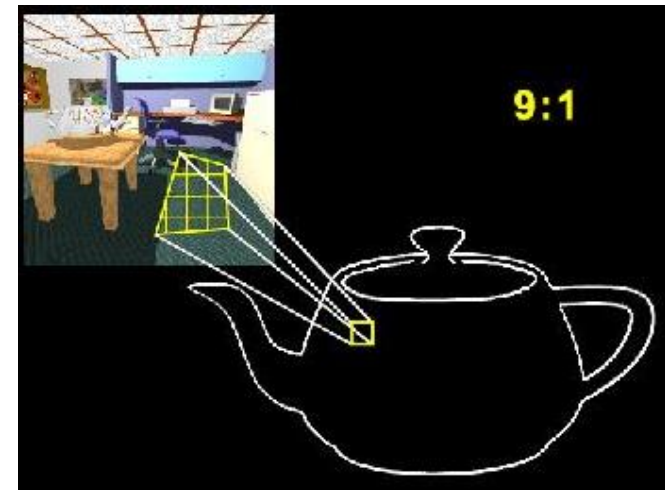
- unaprijed filtriramo – izračunamo 4 puta manje slike tako da u manjoj slici imamo prosječne vrijednosti 4 slikovna elementa veće slike (ili neki bolji filter)
- kada pridjeljujemo boju elementu slike odredimo koliko slikovnih elemenata u prostoru teksture pokriva npr. 9 se preslikava u 1 (footprint)
- ovaj omjer određuje iz kojih mip-mapa određujemo konačnu boju $4:1 < 9:1 < 16:1$
- na ovaj način kombiniramo dobra svojstva pre-filtriranja i povećanog uzorkovanja u sklopovski brzo izvedivu operaciju

<http://www.realtimerendering.com/udacity/?load=demo/unit8-mipmapping.js>

https://cs1230.graphics/demos/crawlies/crawlie_demo.html



IS, FER

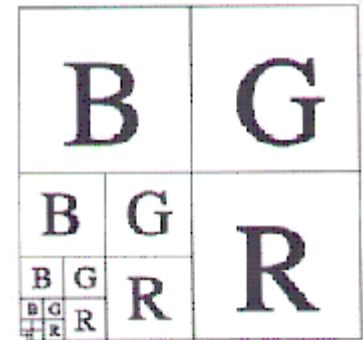
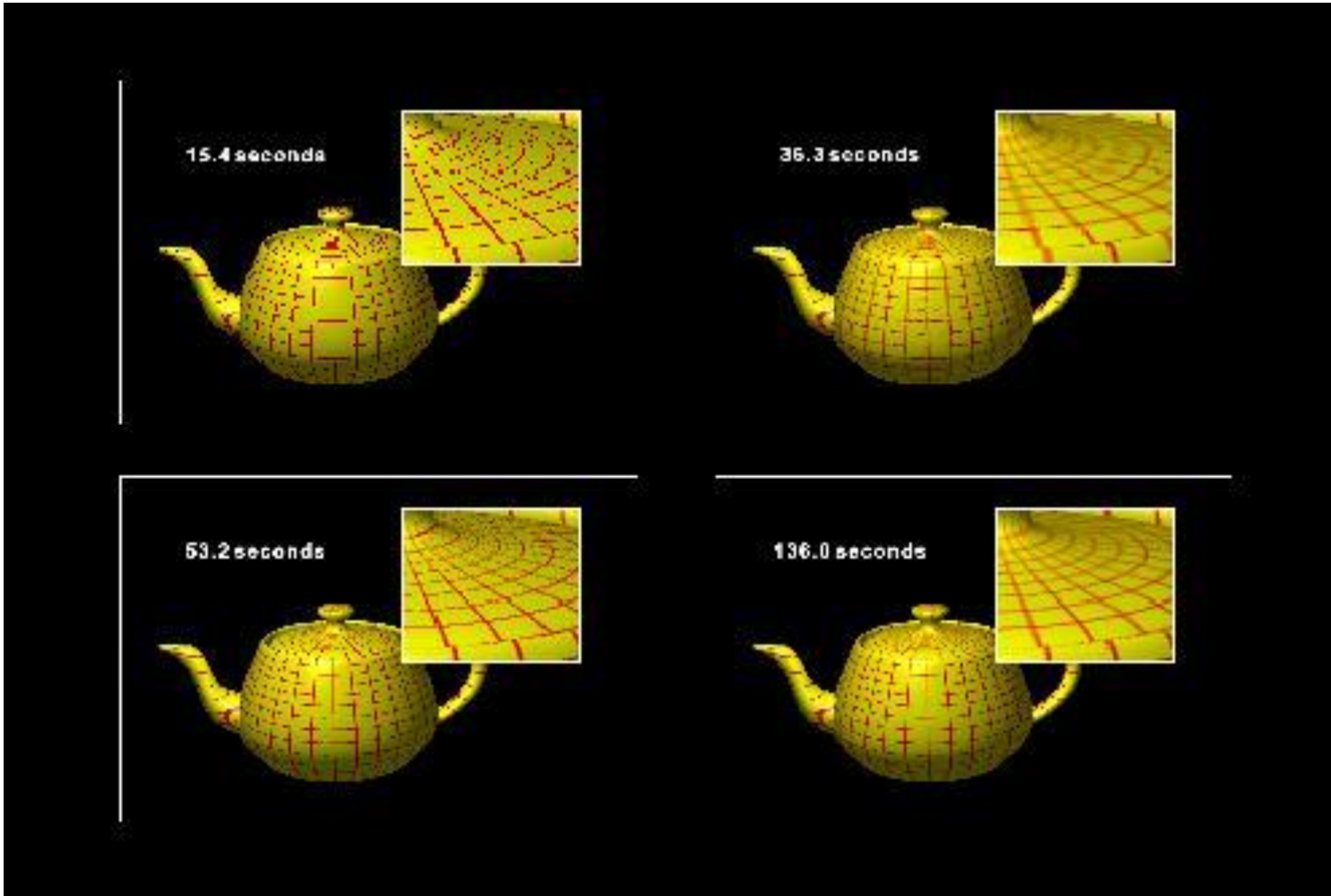


10-8

- mip-map teksture

bez antialiasinga

mip map



supersampling 9 x po sl. elementu kombinacija mip-map i supersampling 9 x

Problemi kod preslikavanja teksture

- problem perspektivno-ispravne projekcije
 - neispravno
 - usitnjavanje poligona dati će nešto bolji rezultat

<http://groups.csail.mit.edu/graphics/classes/6.837/F01/Lecture18/Subdvision-Applet.html>

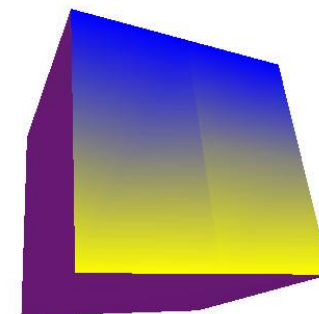
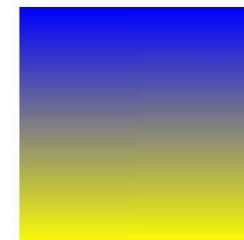
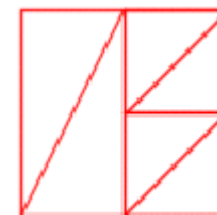
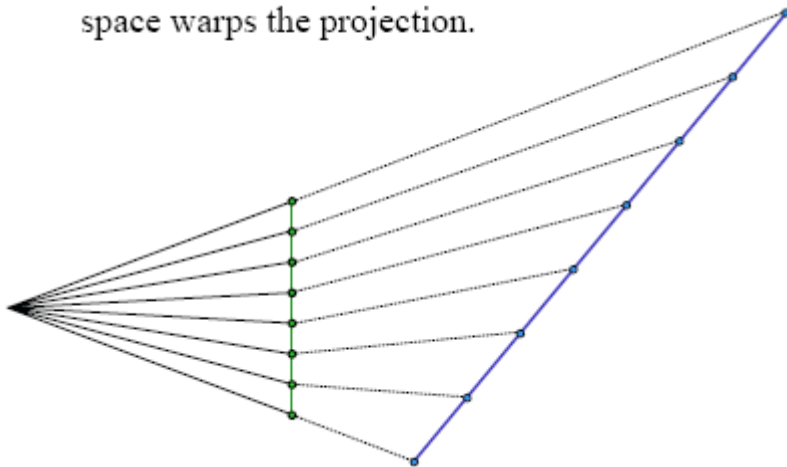
- ispravno <http://groups.csail.mit.edu/graphics/classes/6.837/F01/Lecture18/Perspective-Correct-Applet.html>

- T-spoj – Gouraud-ovo sjenčanje daje neispravan rezultat

<http://groups.csail.mit.edu/graphics/classes/6.837/F01/Lecture18/Gouraud-Applet.html>

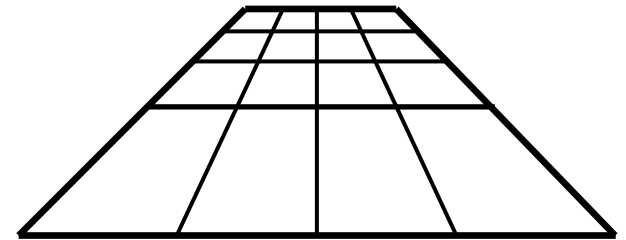
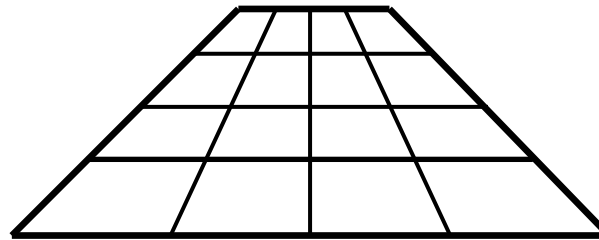
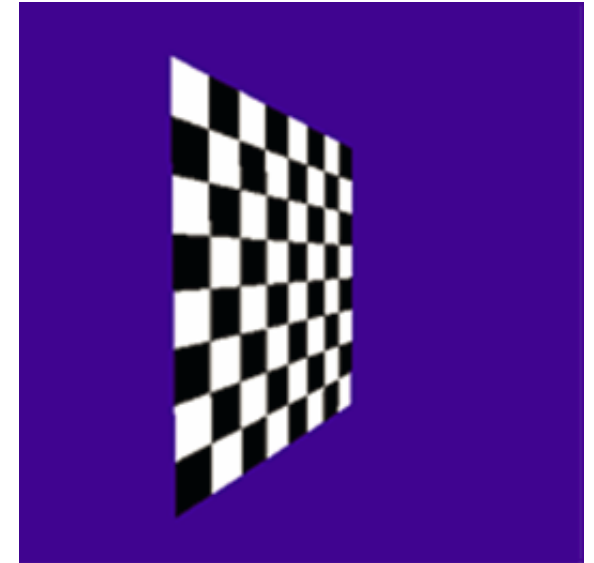
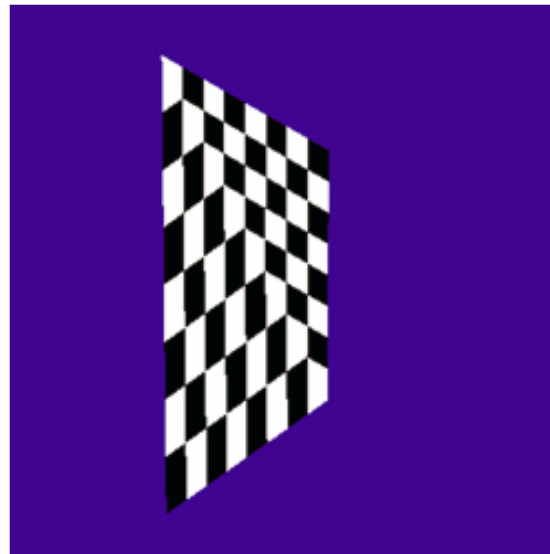
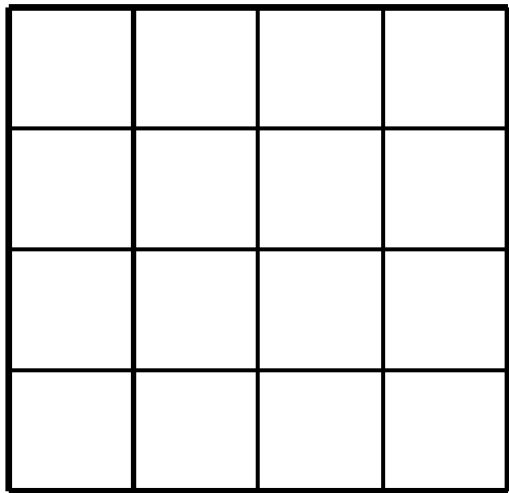
- problem aliasa uslijed uzorkovanja

space warps the projection.



Perspektivno ispravno preslikavanje teksture

- distorzija teksture <http://www.realtimerendering.com/udacity/?load=demo/unit8-texture-distortion.js>



Tehnike uklanjanja neželjenih učinaka uzorkovanja (*Anti-Aliasing*)



Usporedba bez AA
i s uključenim AA
visoka frekvencija je na žicama,
antenama



Bilinearna interpolacija
vide se prijelazi između mip mapa,
pod je mutan,

Trilinearna
ne vide se prijelazi
i dalje mutno,

Anizotropno filtriranje AF
Mip mape različitih rezolucija x, y
npr.
mali kut prema podlozi – različite
frekvencije po x i y
značajno oštrije

Tehnike uklanjanja neželjenih učinaka uzorkovanja (*Anti-Aliasing*)

- AA – npr. povećavamo 4× sliku koju prikazujemo, po svakoj osi 2×, 1024x768 → 2048x1536
 - MSAA (*Multi Sampling AA*) za svaki slikovni element (1024) dohvaća se 4 Z i 1 elementa teksture, zapisuje se 4 elementa
 - SSAA (*Super Sampled AA*) za svaki slikovni element (1024) dohvaća se 4 Z i 4 elementa teksture, zapisuje se 4 elementa (velika mem. propusnost)
- Bolje rezultate daje SSAA od MSAA ali bitno povećava zahtjev na memorijsku propusnost
- TrAA (*Transparent anti-aliasing*) uključuje prozirnost (proširuje MSAA, ne zamjenjuje) dolazi do izražaja kod prikaza polu-prozirnih tekstura, žičana ograda je primjer takve teksture (ATI istu funkciju zove AdAA *Adaptive anti-aliasing*)
 - CSAA (*Coverage Sampled AA*) pri većim povećanjima 16× smanjuje broj Z/boja na 4
 - TAA (*Temporal Anti-Aliasing*) aliasing u vremenu https://threejs.org/examples/#webgl_postprocessing_taa



MSAA bez prozirnosti MSAA s prozirnosti tj. TrAA

Antialiasing Modes Comparison

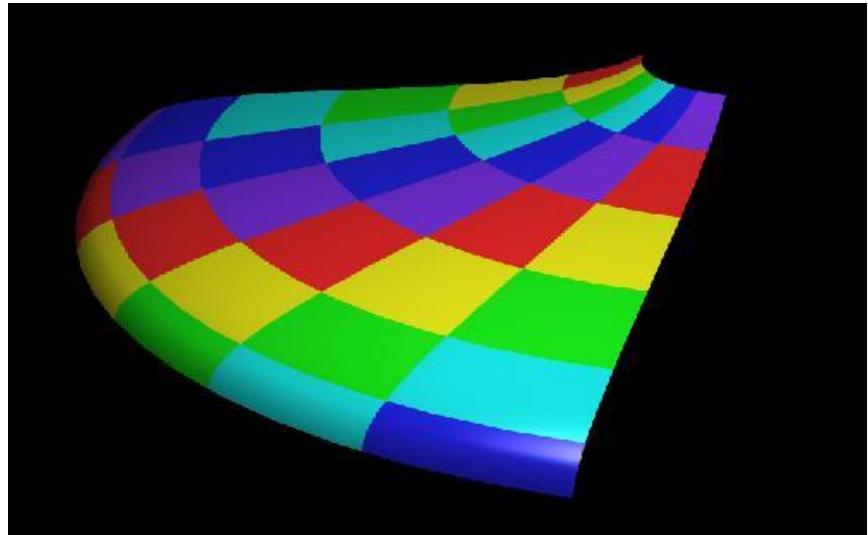
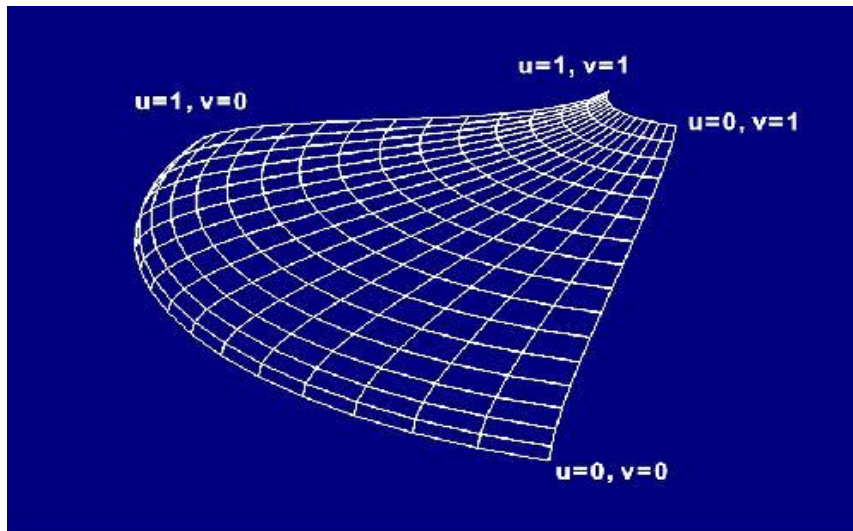
AA Mode:	Supersampling			Multisampling			Coverage Sampling		
Quality level:	1x	4x	16x	1x	4x	16x	1x	4x	16x
Texture/Shader Samples	1	4	16	1	1	1	1	1	1
Stored Color/Z Samples	1	4	16	1	4	16	1	4	4
Coverage Samples	1	4	16	1	4	16	1	4	16

Multisampling reduces texture & shader work

Coverage Sampling reduces color & Z footprint & bandwidth

10.1.2 Postupak preslikavanja na krpicu površine

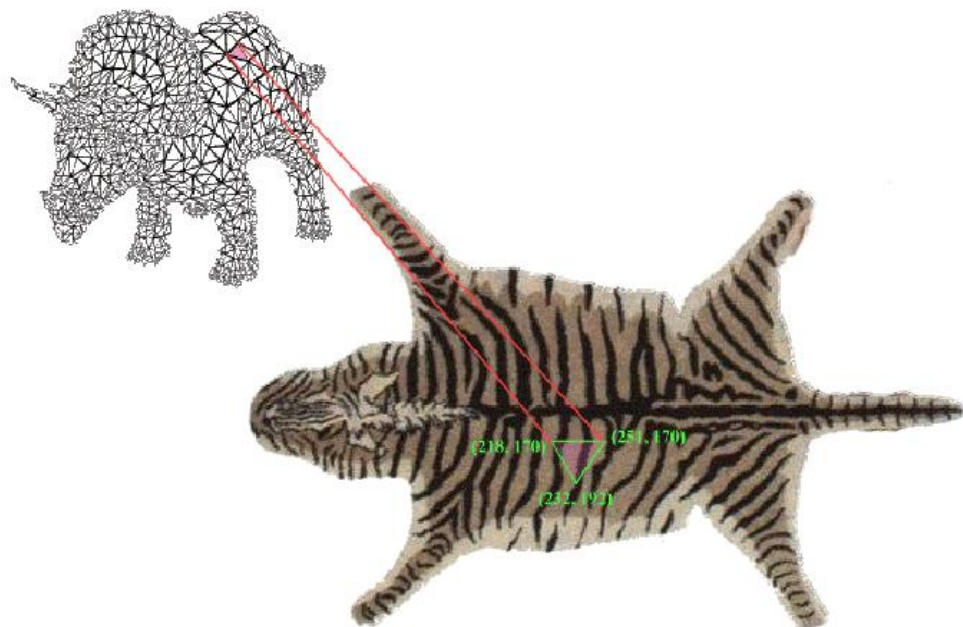
- parametarski definirano preslikavanje
 - kod Bezier-ovih ili B-krpica parametri koji određuju samu površinu jednoznačno određuju i preslikavanje texture na tu površinu



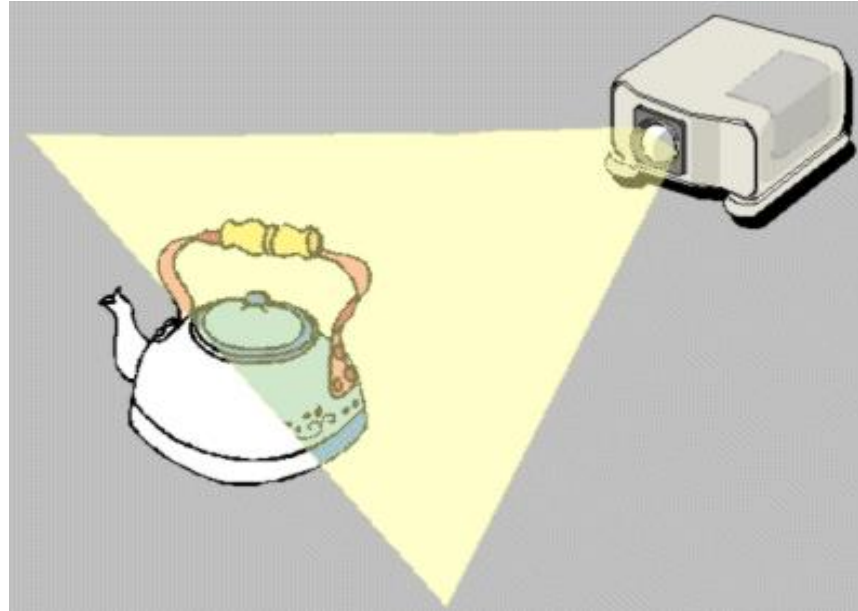
10.2 Preslikavanje teksture na mrežu poligona

10.2.1 UV razmatanje teksture – atlas tekstura

- za svaki trokut modela određuju se pripadno područje teksture – pripadne 2D koordinate u prostoru teksture (atlas teksture)
 - potrebno je definirati „šavove” – bridovi na kojima se tekstura spaja u 3D
- <https://tomtung.github.io/chameleon.js/> Fur/Export - Texture UnWrap

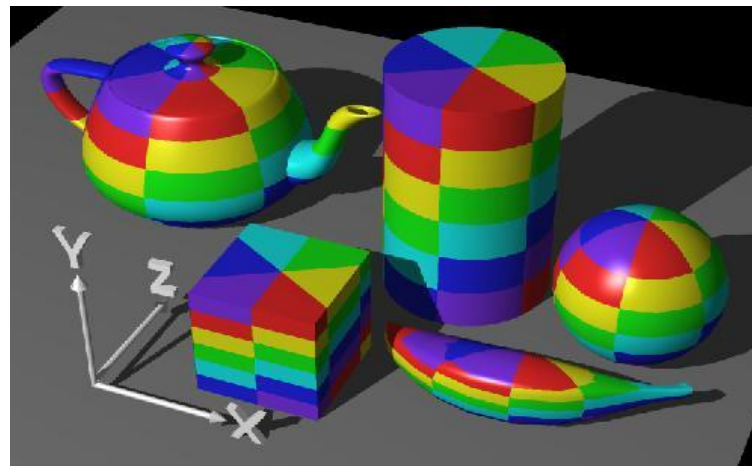
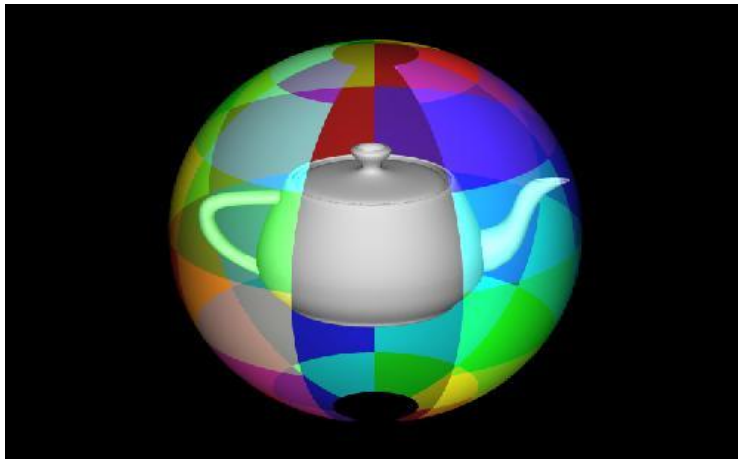


10.2.2 Projekcijske teksture

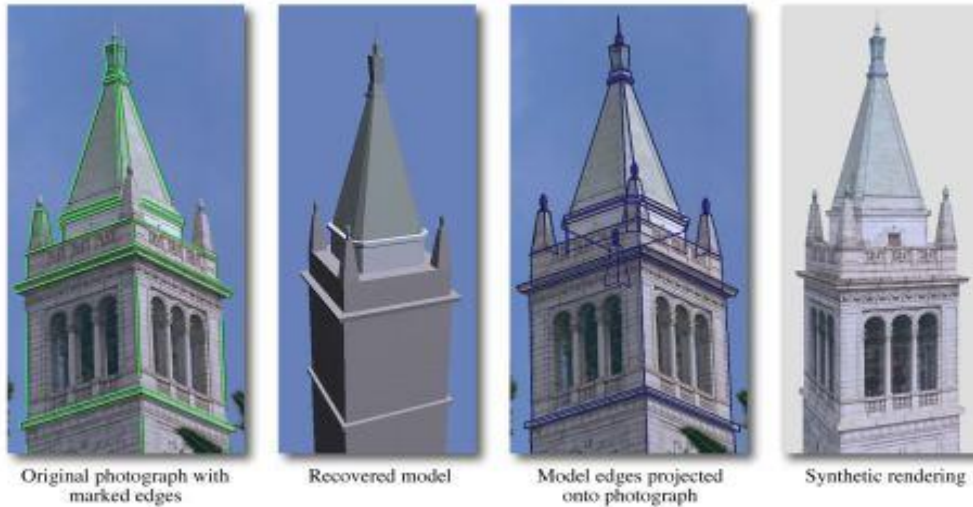


- zbog lakšeg određivanja (u, v) koordinata tekstura se odredi na
 - ravnini, kocki, cilindru, sferi
 - zatim se projicira na zadani objekt
 - moguća su „razvlačenja” teksture na površinama
- https://threejs.org/examples/#webgl_lights_spotlight
- <https://webglfundamentals.org/webgl/webgl-planar-projection-with-lines.html>
- <https://webglfundamentals.org/webgl/webgl-planar-projection-with-projection-matrix-0-to-1.html>

- projiciranje teksture sa sfere i cilindra <http://stemkoski.github.io/Three.js/Sphere-Unwrapping.html>
- <https://stemkoski.github.io/Three.js/Sphere-Project.html>
- iz zadanog smjera (kamera/normala) http://threejs.org/examples/webgl_decals
- <https://math.hws.edu/graphicsbook/demos/c7/generated-texcoords.html>
- <https://www.clicktorelease.com/code/decal-splatter/>



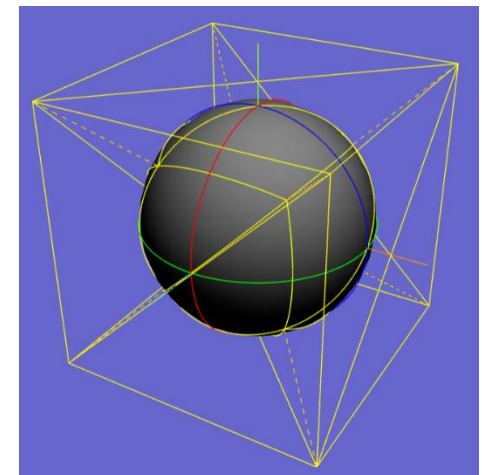
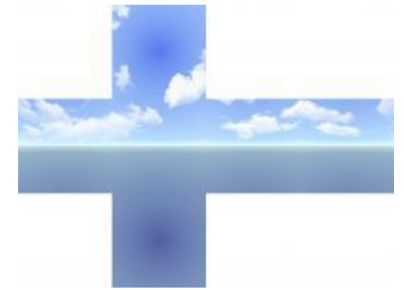
Projekcijsko preslikavanje teksture



- preslikavanje na tijela – ravnina, cilindar, sfera
- SIGGRAPH -
- <http://www.debevec.org/Movies/debevec-campanile.mov> 40MB

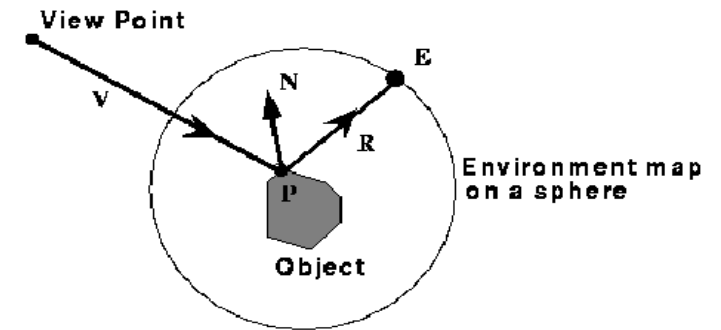
10.2.3 Preslikavanje okoliša (engl. environment mapping)

- kako bi scena ljepše izgledala obično postavljamo nebo scene (engl. *SkyBox*) – kupolu koje okružuje našu scenu i predstavlja udaljene objekte teksturom
- obično se koristi preslikavanje sa šest stranica kocke (engl. *CubeMap*)
- <https://stemkoski.github.io/Three.js/Skybox.html>
- preslikavanje između kocke i kupole koja okružuje scenu može se dobro definirati
- učinkovitiji način pohrane teksture je u jednu teksturu (HDR) umjesto na šest stranica kocke
- mapa okoliša se zatim može koristiti za
 - zrcaljenje okoliša na objekt <http://demo.dzsd.net/Viewer3DV4Preview/CAR.html>
https://threejs.org/examples/#webgl_materials_envmaps
 - postizanje prozirnosti objekta
http://threejs.org/examples/#webgl_materials_cubemap_refraction
 - osvjetljenje temeljeno na slici IBL (engl. *Image-based lighting*)
<http://playcanv.as/p/SA7hVBLt/>
- mapa okoliša može biti slika no može biti i film



- pretpostavlja se da je okoliš dovoljno daleko i da se objekt ne zrcali sam na sebe
- vektor reflektirane zrake određuje zrcaljenje okoliša
 1. određivanje 2D mape okoliša
 2. određivanje normale i reflektirane zrake
 3. reflektirana zraka daje indeks u mapi okoliša

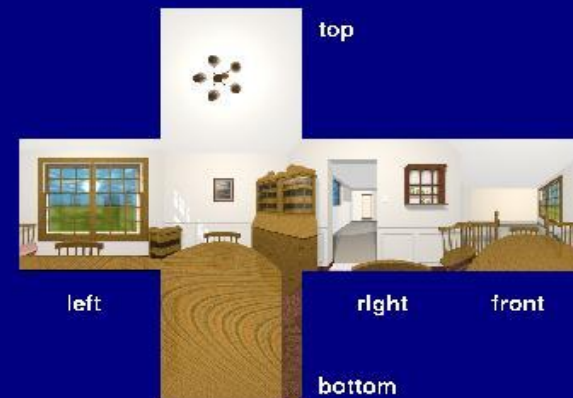
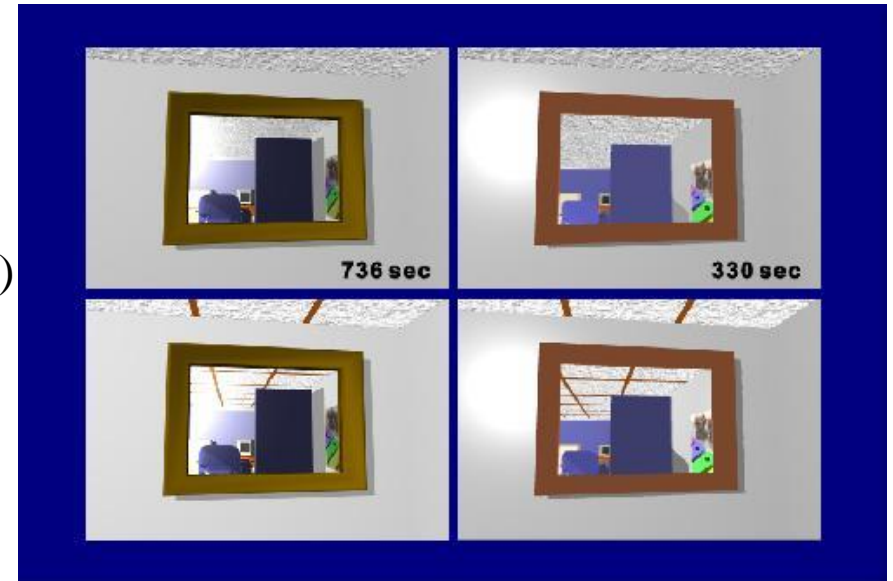
Sliku okoliša možemo promatrati tako da obuhvaća cijelu scenu no možemo promatrati i tako da obuhvaća samo zadani objekt



Preslikavanje okoliša

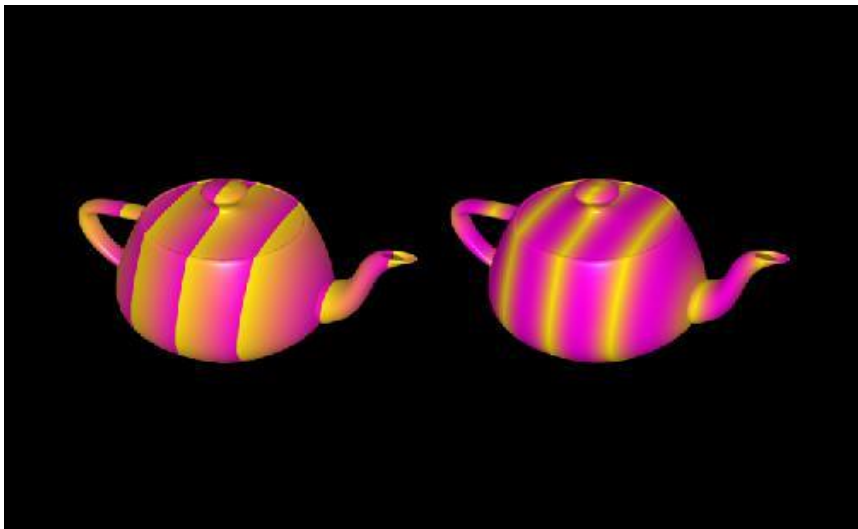
- određivanje 6 projekcija na ravnine kocke koja okružuje objekt s reflektivnim svojstvom
- sklopovski podržano (engl. cube map reflection)

https://threejs.org/examples/#webgl_materials_cubemap_dynamic

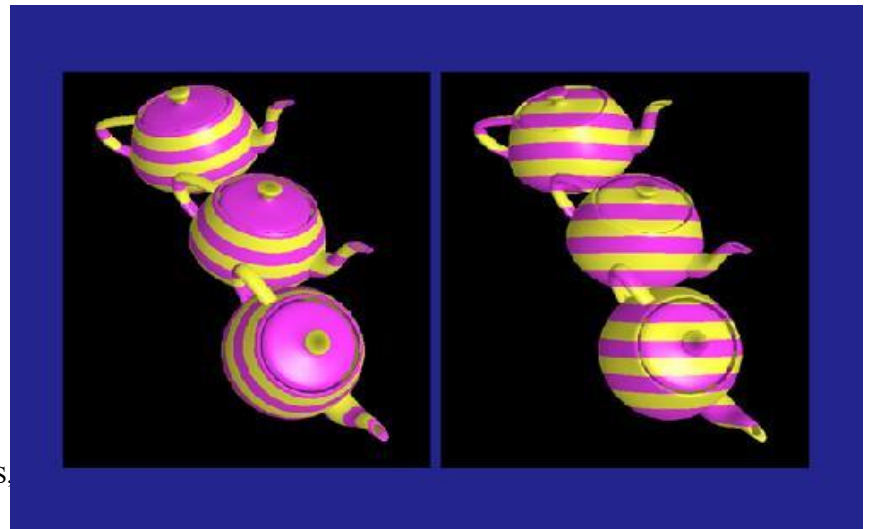


10.3 Proceduralne teksture - hiperteksture

- 2D hipertekstura – funkcijski opisana (problem AA kod preslikavanja)
- <https://www.shadertoy.com/view/MdjGR1>
- 3D hipertekstura – funkcija boja $=f(x,y,z)$
 - Npr. boja = if (x paran) žuta inače roza;
- generiranje teksture unutar volumena kocke 3D
 - nije potrebno preslikavanje u 2D
 - mramor, drvo
 - Marble = $\sin(n * (x + A * \text{Turb}(x,y,z)))$ <https://www.shadertoy.com/view/MtX3Ws>
 - n - broj širokih pruga, A - turbulencija pruga
- <http://legakis.net/justin/MarbleApplet/>



IRIS.

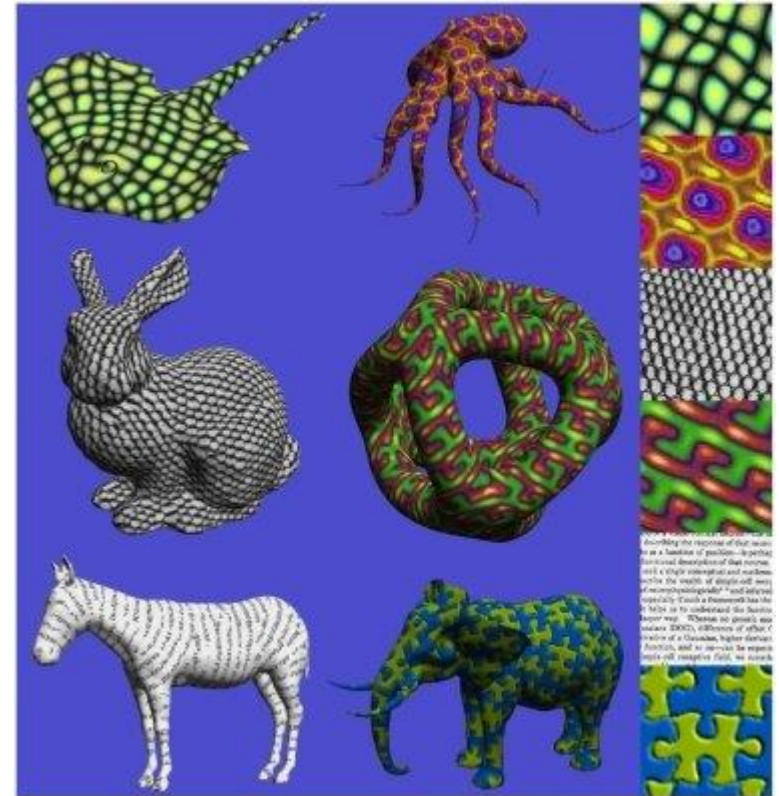


- Perlinov šum – za proceduralno generiranje tekstura, terena, oblaka ...

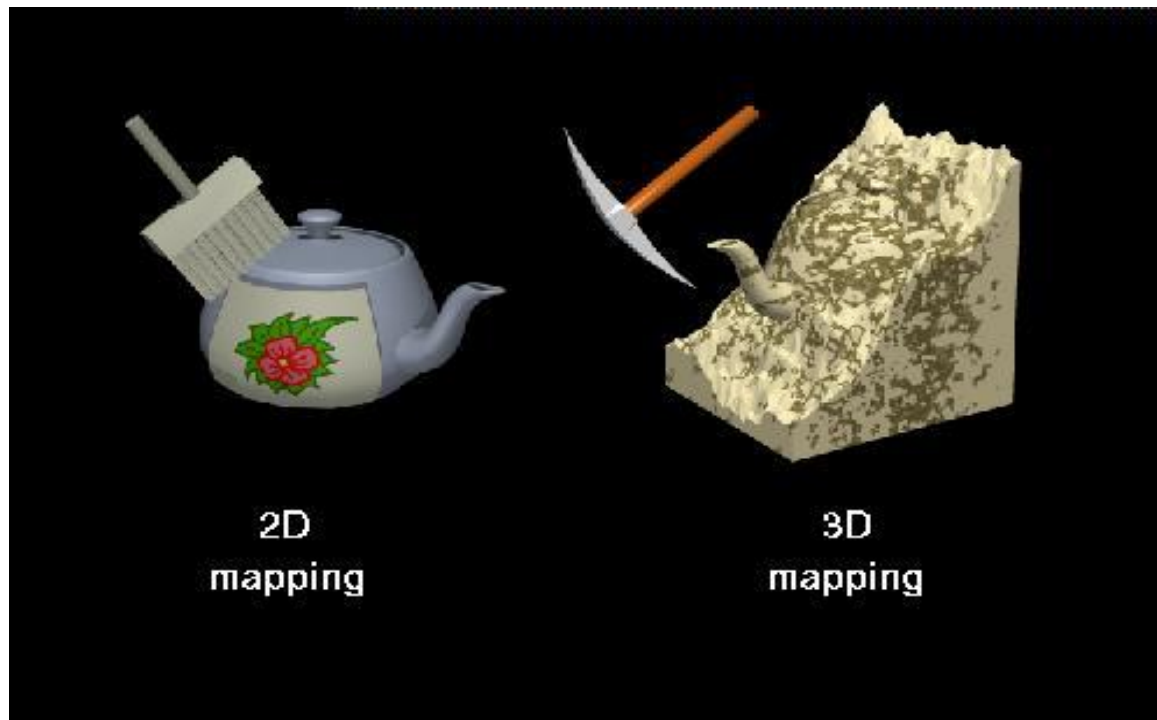
<https://codepen.io/jhnsnc/pen/KNyOqV>

https://threejs.org/examples/webgpu_volume_perlin

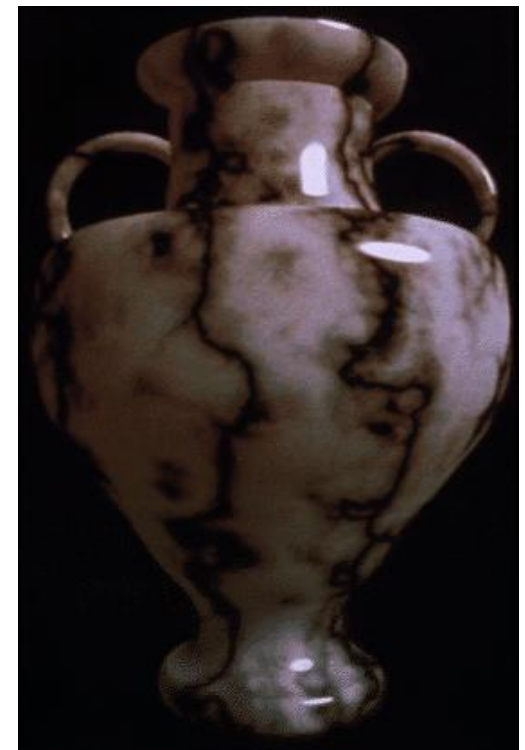
- sinteza tekstura – iz poznatog uzorka generira se proizvoljna veličina teksture (statističke metode)
 - generiranje proizvoljno velike teksture iz zadanog uzorka tako da se nastavlja na prethodni uzorak i prati zakrivljenost objekta
 - generiranje volumnog uzorka (3D teksture)



- 3D teksture definirane funkcijski, niz 2D tekstura čine 3D teksturu
- problem pomicanja objekta kroz prostor teksture (engl. solid texture)
 - teksturu možemo fiksirati uz objekt <http://inear.se/lathe/>
 - objekt možemo pomicati kroz prostor teksture



razlika preslikavanje 2D i 3D teksture

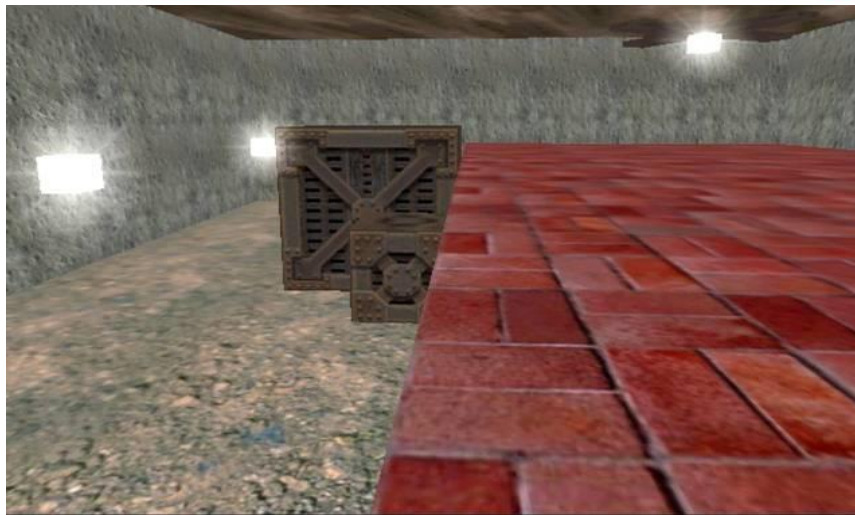


10.4 Teksture za posebne namjene

10.4.1 Mape svjetla (engl. light mapping)

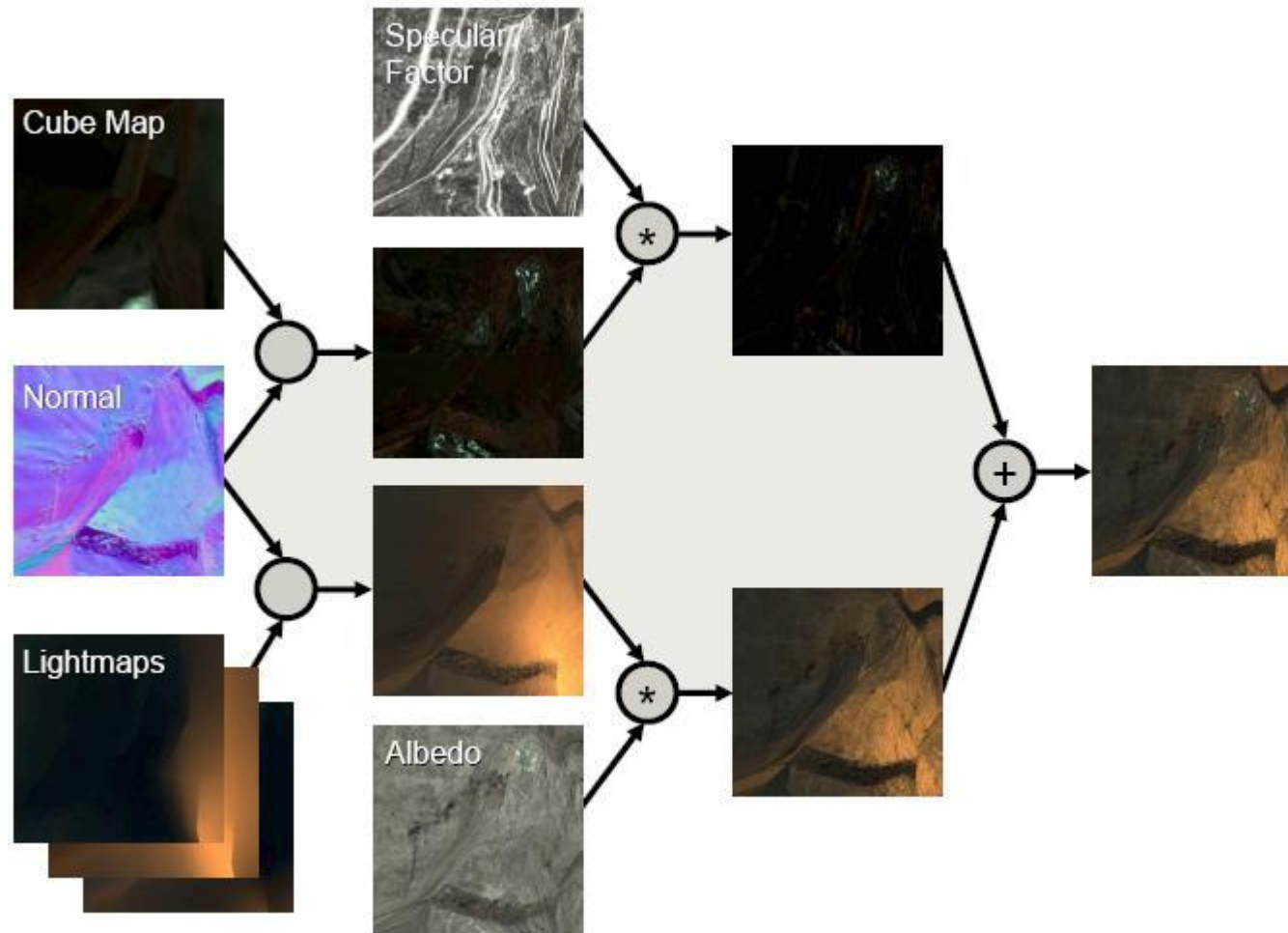
<http://jyunming-chen.github.io/tutsplus/tutsplusS-2.html>

https://threejs.org/examples/?q=mater#webgpu_materials_lightmap

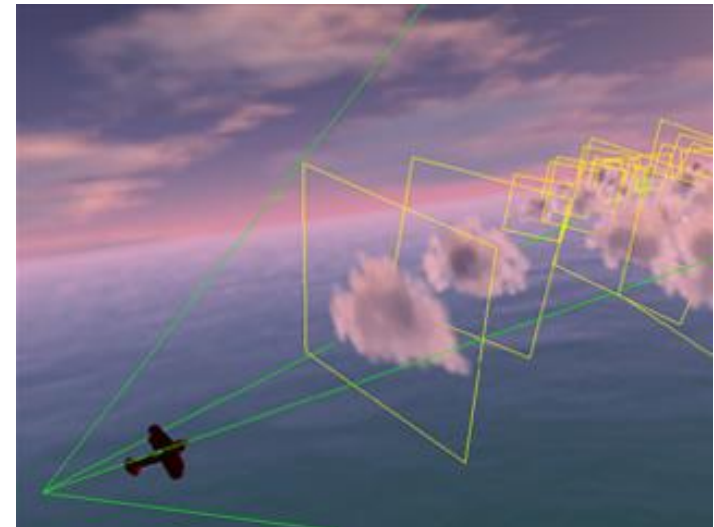


Kombiniranje slojeva prikaza scene (tekstura) u ostvarivanju konačne slike

- iscrtavanje s odgodom (*deferred rendering*)
- <https://experiments.withgoogle.com/webgl-deferred-irradiance-volumes>



- prozirne teksture – ploča za plakate (engl. billboards)
 - po uzoru na 2D tehnologiju pomičnih slika (sprite)
 - preslikavanje teksture na poligon koji je uvijek okrenut prema promatraču (ili više statičnih poligona) <https://stemkoski.github.io/Three.js/Sprites.html>
obično se koriste za prikaz drveća ili općenito vegetacije ($\alpha=0$ prozirno, $\alpha=1$ zeleno)
 - važan je redoslijed iscrtavanja BTF kod prozirnih objekata
<http://jeromeetienne.github.io/threex.grass/examples/demo.html>
- animirajuće teksture
<https://stemkoski.github.io/Three.js/Texture-Animation.html>
- video-teksture
<https://stemkoski.github.io/Three.js/Video.html/>
- biomske mape (stanište)
 - proceduralno generiranje terena
<https://icecreamyou.github.io/THREE.Terrain/>



Stapanje tekstura uz zadane prozirnosti (engl. alpha blending)

https://threejs.org/examples/#webgl_materials_blending_custom

- više slojeva koji imaju definiranu prozirnost <https://stemkoski.github.io/Three.js/Translucence.html>
- <http://cs.wellesley.edu/~cs307/threejs/demos/Transparency/transparencyTutor.html>
- konveksna kombinacija boja

- važan je redoslijed prekrivanja – operator **over**

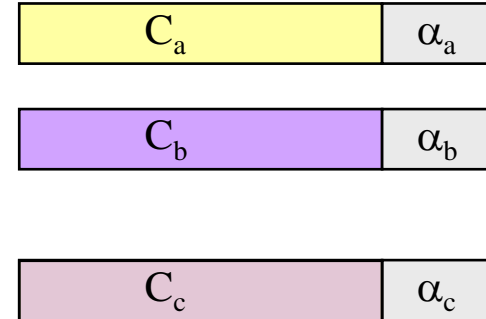
C boja - $C_{a \text{ over } b} = (C_a \alpha_a + C_b \alpha_b (1 - \alpha_a)) / \alpha_{a \text{ over } b}$

α prozirnost (0 prozirno, 1 neprozirno)

- nakupljanje (ne)prozirnosti

$$\alpha_{a \text{ over } b} = \alpha_a + \alpha_b (1 - \alpha_a)$$

a over b



http://www.realtimerendering.com/udacity/?load=demo/unit3-over_operator.js

Kombiniranje tekstura (engl. texture splatting)

