

7. Prikaz detalja površine

- podjela tehnika ovisno o promjeni
 - uzorka teksture (*engl. image-based rendering*) - prividno povećava složenost površine bez povećavanja geometrijske složenosti
 - utjecaj na vektor normale - (difuzna, zrcalna komponenta)
 - pomicanje elemenata teksture
 - geometrije (*engl. geometry-based rendering*)

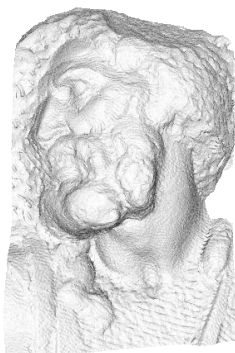
Primjer preslikavanja izbočina

$$I = k_a I_a + k_d I_d \frac{\vec{l} \cdot \vec{n}}{d_L + k} + k_z I_z \frac{(\vec{v} \cdot \vec{r})^n}{d_L + k}$$

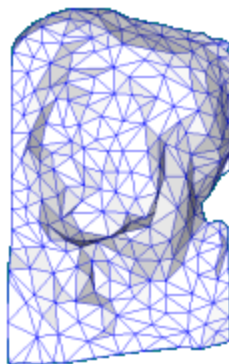
- na silueti se ne vide izbočine

<http://www.realtimerendering.com/teapot/> (Flat)

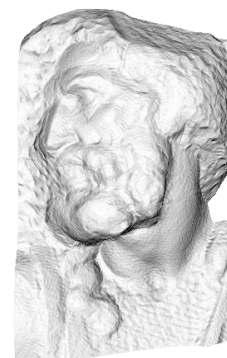
<https://www.shadertoy.com/view/4t2SzR>



manja
složenost

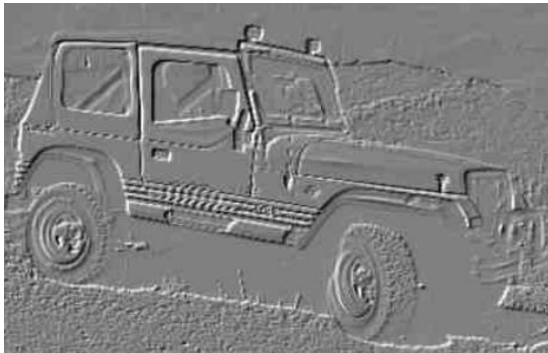


preslikavanje
izbočina



7.1 Tehnika istaknutosti (engl. emboss)

- motivacija - osnovna ideja prividnog povećanja složenosti površine potiče od stvaranja lažnih istaknutosti
- dodavanje lažne sjene već će stvoriti dojam da je nešto izbočeno <http://jsfiddle.net/3ehe58ru/3//>
- lažne izbočine
 - pomicanje teksture u smjeru/suprotno vektoru prema izvoru <https://math.hws.edu/graphicsbook/demos/c2/image-filters.html>
 - kombinacija s početnom teksturom (filter u domeni slike) <https://webglfundamentals.org/webgl/webgl-2d-image-processing.html>
- veličina pomaka je ograničena i ovisi o teksturi, može doći do pojave aliasa
- proširenje tehnike na 3D prostor
 1. određivanje vektora prema izvoru u tangentnom prostoru
 2. određivanje projekcije vektora prema izvoru na ravninu teksture

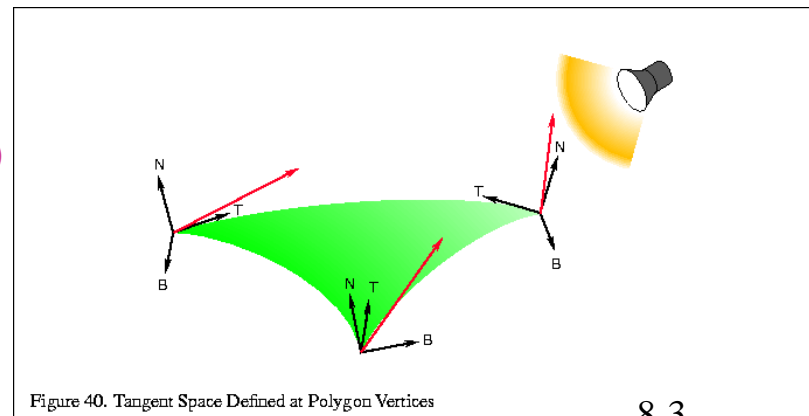
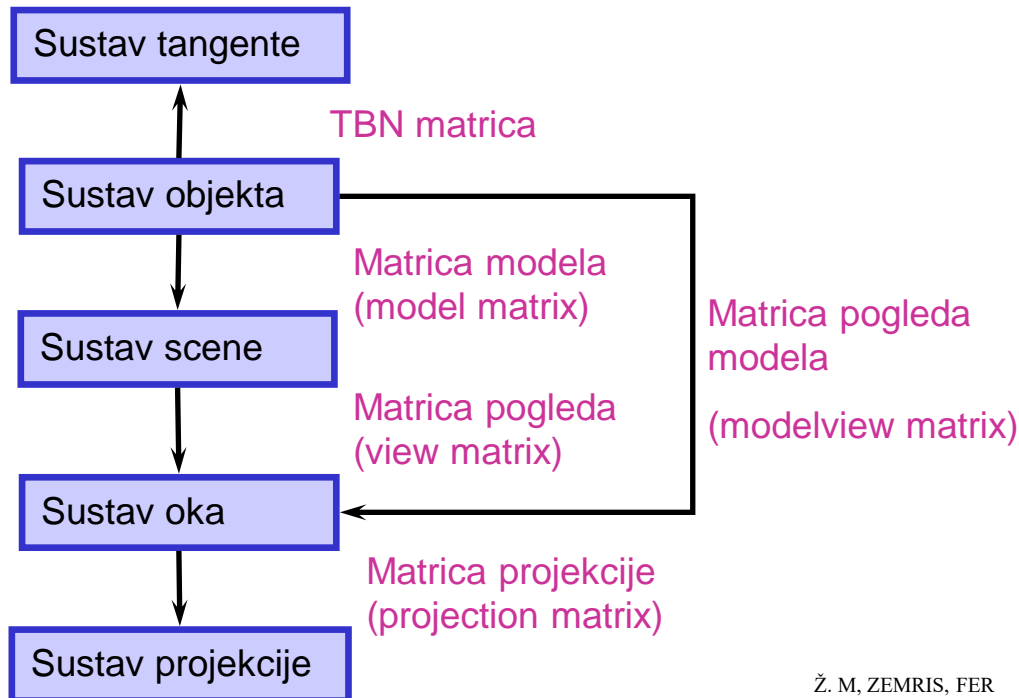


Postupke možemo podijeliti u 2 koraka

- koordinatnu sustavi – određivanje vektora (\mathbf{l} , \mathbf{v}) u koordinatnom sustavu TBN (tangenta, normala, binormala) sjenčar vrhova
- preslikavanje izbočina (uz paralaksu) – sjenčar fragmenata

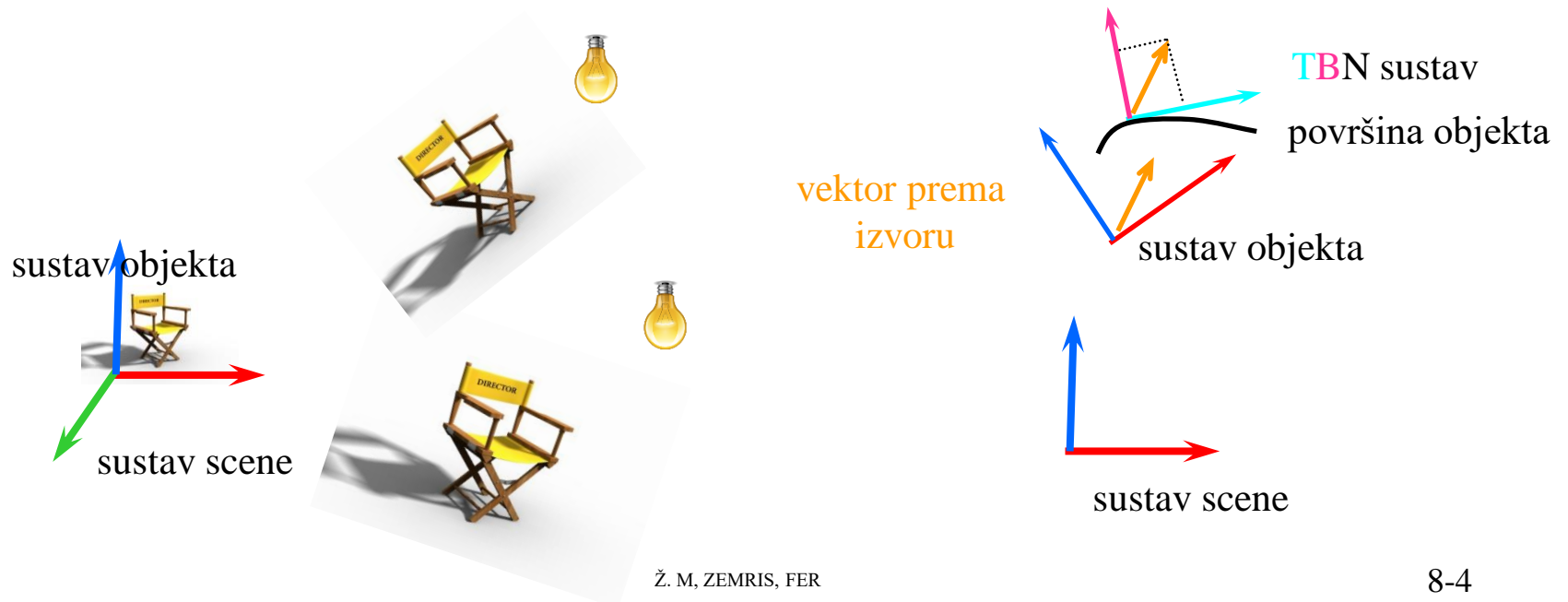
Koordinatni sustavi - sustav tangente (engl. tangent space)

- u koordinatnom sustavu površine definiramo sustav tangente
- izračunavanje osvjetljenja - vektori moraju biti u istom sustavu
- **TBN (Tangenta Binormala Normala)** matrica određuje preslikavanje iz sustava objekta u sustav tangente – vezano uz koordinatni sustav teksture



Koordinatni sustavi

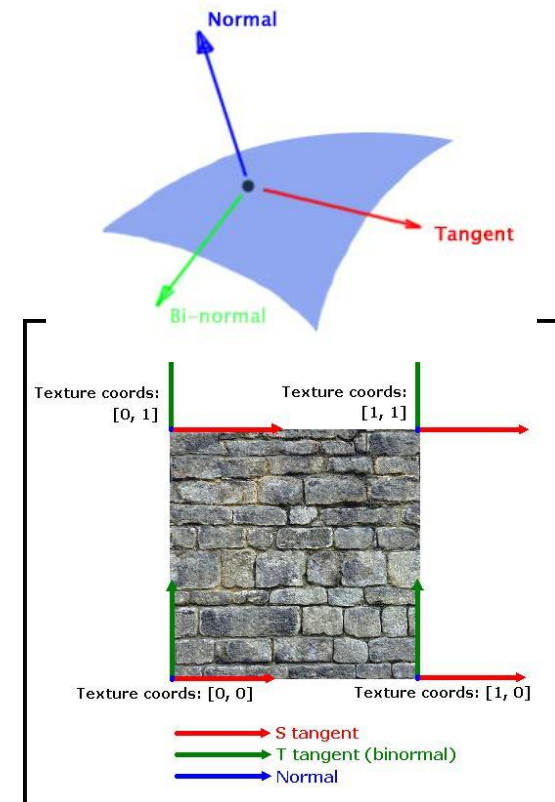
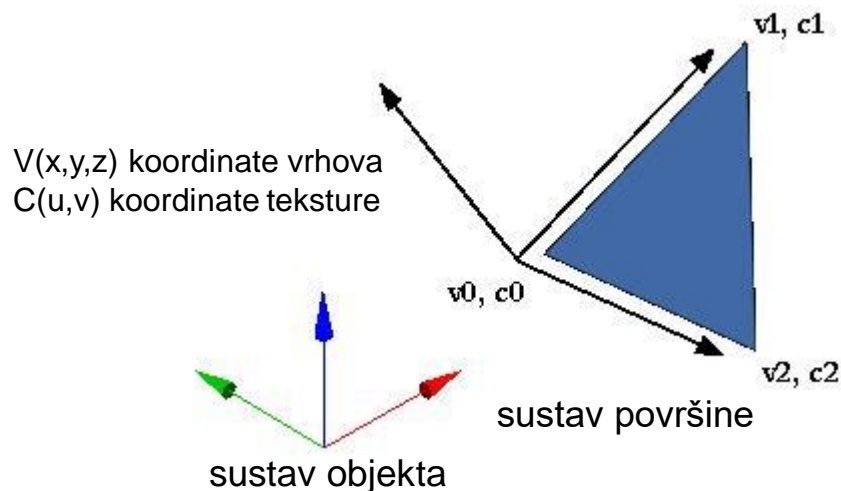
- u složenim scenama učitavamo objekte čije su koordinate u sustavu objekta
- isti objekt možemo transformirati na različite pozicije u sceni,
- pri određivanju osvjetljenja treba odrediti produkt $\mathbf{n}\mathbf{l}$
- normala \mathbf{n} je obično u koordinatnom sustavu objekta (TBN je određeno za svaku točku površine)
- izvore možemo imati zadane u koordinatnom sustavu scene ili koordinatnom sustavu objekta, vektor prema izvoru \mathbf{l} mora biti u istom koordinatnom sustavu kao \mathbf{n} (moramo \mathbf{l} transformirati u sustav objekta – odnosno površine - tangente)



Sustav tangente TBN (engl. tangent space)

- matrica za transformaciju vektora iz koordinatnog *sustava objekta* u *sustav površine tj. sustav tangente* – rotacija vektora
- **T** Tangenta, **B** Binormala, **N** Normala - čine matricu TBN
- **O** vektor u sustavu objekta, **S** transformirani (rotirani) vektor (npr. osi koordinatnog sustava $\mathbf{x} = (1\ 0\ 0)$, $\mathbf{y} = (0\ 1\ 0)$, \mathbf{z} se preslikavaju u **T, B, N**)
- vrijedi samo za ortonormiranu matricu $M=TBN$ (ako nije tada vrijedi $S = O (M^{-1})^T$)

$$\begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} = \begin{bmatrix} T_x & T_y & T_z \\ B_x & B_y & B_z \\ N_x & N_y & N_z \end{bmatrix} \cdot \begin{bmatrix} O_x \\ O_y \\ O_z \end{bmatrix}$$



7.2 Preslikavanje izbočina (eng. bump mapping)

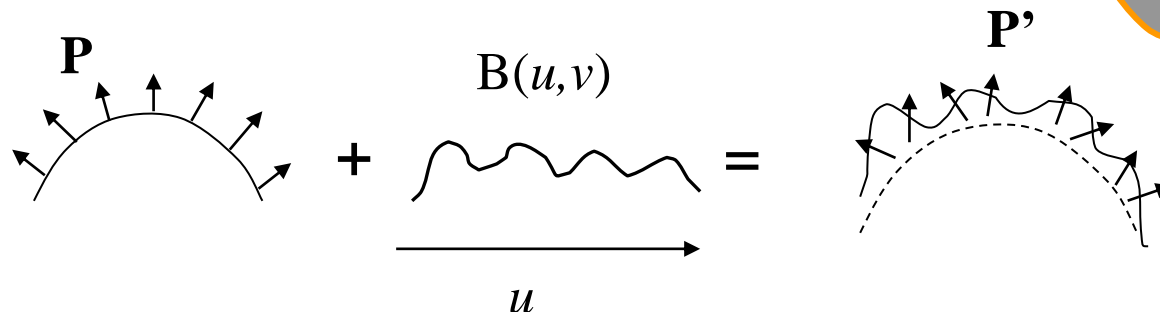
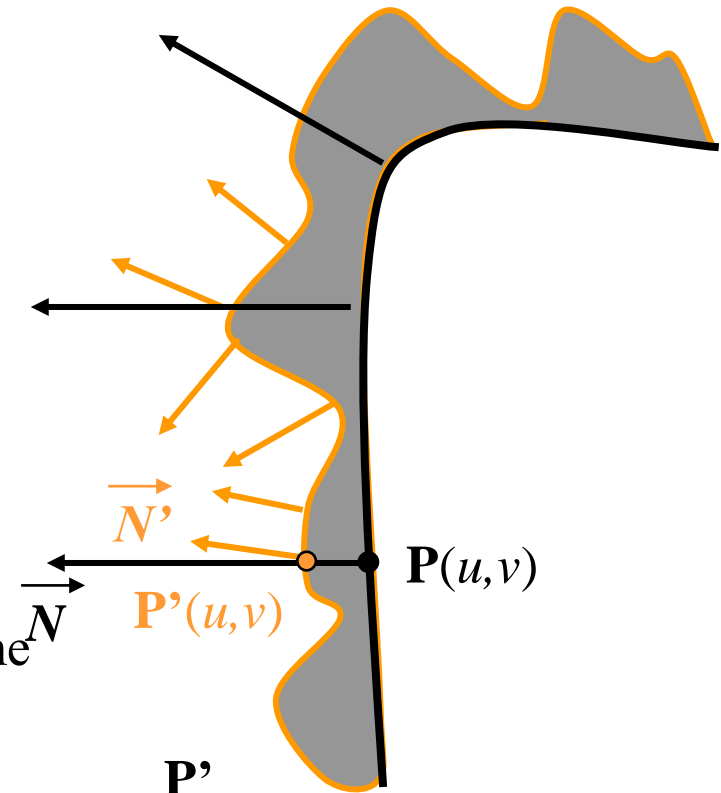
- određivanje normale na izbočinu
(engl. normal mapping [Blin 78])
- određivanje novoga vektora normale

$\mathbf{P}(u, v)$ početna površina

$\mathbf{B}(u, v)$ izbočine <http://cpetry.github.io/NormalMap-Online/>

$\mathbf{P}'(u, v)$ nova površina

- potrebno je odrediti normale nove površine



- točka površine i normala, P_u, P_v su parcijalne derivacije

$$\vec{P} = [x(u, v) \quad y(u, v) \quad z(u, v)]^T, \vec{N} = \vec{P}_u \times \vec{P}_v$$

- nova točka površine i nova normala

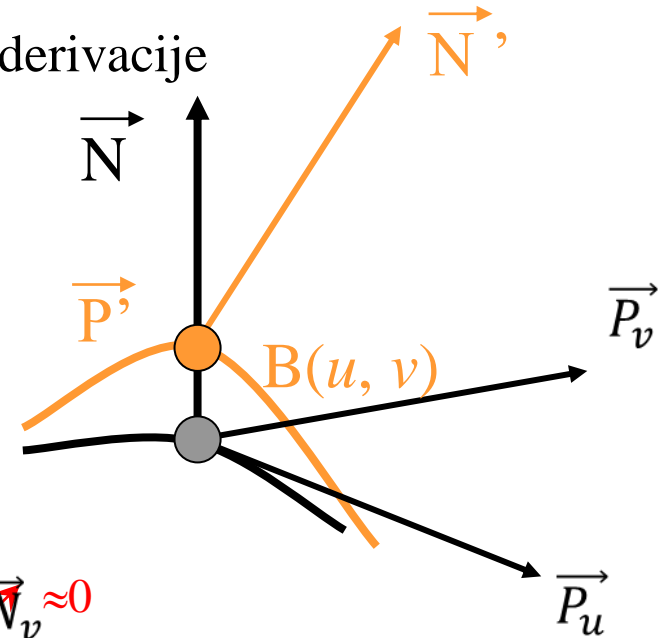
$$\vec{P}' = \vec{P} + B(u, v)\vec{N}/\|\vec{N}\|, \quad \vec{N}' = \vec{P}'_u \times \vec{P}'_v$$

- parcijalne derivacije u novoj točki površine

$$\vec{P}'_u = \vec{P}_u + \frac{B_u \vec{N}}{\|\vec{N}\|} + \frac{B \vec{N}_u}{\|\vec{N}\|} \approx 0, \quad \vec{P}'_v = \vec{P}_v + \frac{B_v \vec{N}}{\|\vec{N}\|} + \frac{B \vec{N}_v}{\|\vec{N}\|} \approx 0$$

- tražena normala

$$\vec{N}' \approx \vec{P}_u \times \vec{P}_v + \frac{B_u(\vec{N} \times \vec{P}_v)}{\|\vec{N}\|} + \frac{B_v(\vec{P}_u \times \vec{N})}{\|\vec{N}\|} + \frac{B_u B_v(\vec{N} \times \vec{N})}{\|\vec{N}\|^2}$$



- tražena normala

$$\vec{N}' \approx \vec{P}_u \times \vec{P}_v + \frac{B_u(\vec{N} \times \vec{P}_v)}{\|\vec{N}\|} + \frac{B_v(\vec{P}_u \times \vec{N})}{\|\vec{N}\|} + \frac{B_u B_v(\vec{N} \times \vec{N})}{\|\vec{N}\|^2}$$

- vrijedi

$$\vec{P}_u \times \vec{P}_v = \vec{N}, \vec{P}_u \times \vec{N} = -\vec{N} \times \vec{P}_u, \vec{N} \times \vec{N} = 0$$

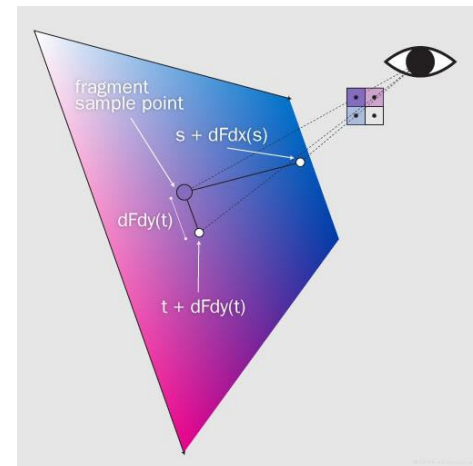
- pa slijedi

$$\vec{N}' \approx \vec{N} + \frac{B_u(\vec{N} \times \vec{P}_v)}{\|\vec{N}\|} - \frac{B_v(\vec{N} \times \vec{P}_u)}{\|\vec{N}\|}$$

- $B(u, v)$ je mapa izbočina, B_u, B_v su parcijalne derivacije

$$B_u = \frac{B(u - \Delta, v) - B(u + \Delta, v)}{2\Delta} \quad B_v = \frac{B(u, v - \Delta) - B(u, v + \Delta)}{2\Delta}$$

- <http://codepen.io/Mombasa/pen/ivdyC>
<http://29a.ch/2010/3/24/normal-mapping-with-javascript-and-canvas-tag>
<https://www.clicktorelease.com/code/spherical-normal-mapping/#>



- u postupku preslikavanja izbočina trebamo izračunati normalu N' ovisno o izbočini

za ravnu površinu u koordinatnom sustavu te površine možemo računati:

- polje teksture određuje visinu izbočine $B(u, v)$ -visinska mapa (*height map*) obično se uz teksturu RGB pohranjuje u α kanalu
- računamo parcijalne derivacije u smjeru u i v - $B_u(u, v)$, $B_v(u, v)$ aproksimacija je:

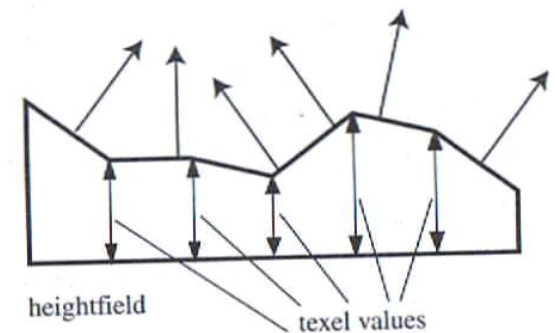
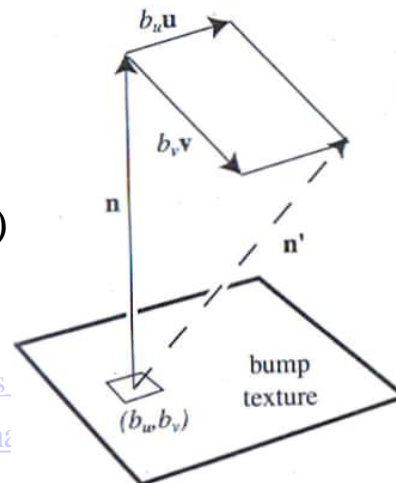
$$B_u(u, v) = B(u+1, v) - B(u, v)$$

$$B_v(u, v) = B(u, v+1) - B(u, v)$$

- vektorski produkt $B_u(u, v)$, $B_v(u, v)$ daje vektor normale N' na traženom mjestu (ako promatramo u koordinatnom sustavu površine) <http://lo-th.github.io/labs/index.html> (Terrain)
- slika u boji – također može poslužiti kao mapa izbočina – odredimo sivu sliku – visine

- u GLSL $dFdx(s)$ i $dFdy(s)$ - nisu u domeni teksture (s, t) no mogu se iskoristiti

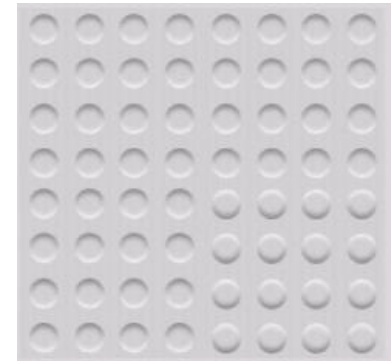
- http://threejs.org/examples/webgl_materials
https://threejs.org/examples/webgl_m



- u prostor za teksturu umjesto RGB komponenti upišu vrijednosti vektora normale N (engl. normal map)
 - <http://jyunming-chen.github.io/tutsplus/tutsplus28.html>
 - skalarni umnožak dobivene normale s vektorom prema izvoru l za difuznu komponentu i s vektorom r za zrcalnu komponentu (sjenčar fragmenata)
 - važno je da su vektori transformirani u isti prostor (sjenčar vrhova)
 - ova tehnika posebno dolazi do izražaja kada je izvor pomičan i blizu poligona
 - <http://neilwallis.com/projects/java/bump/bump1.php>
 - http://xeogl.org/examples/#materials_phong_textures_video_normalMap

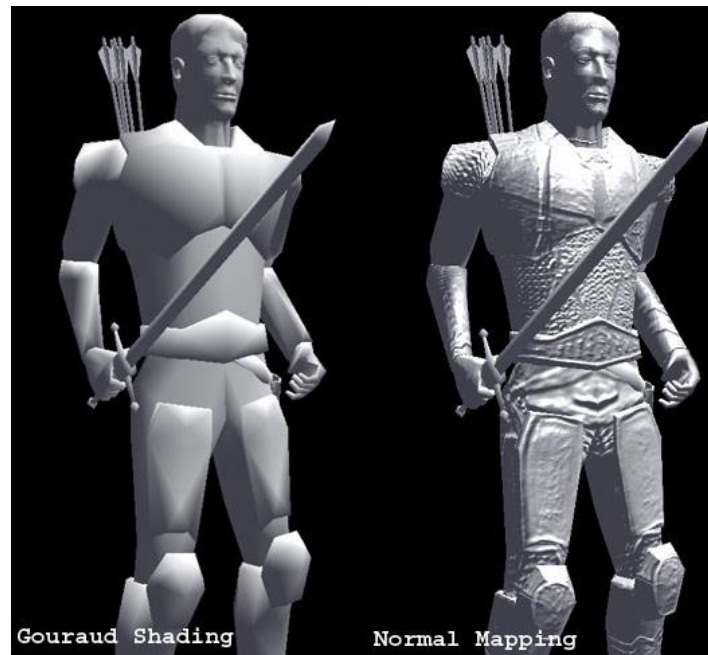
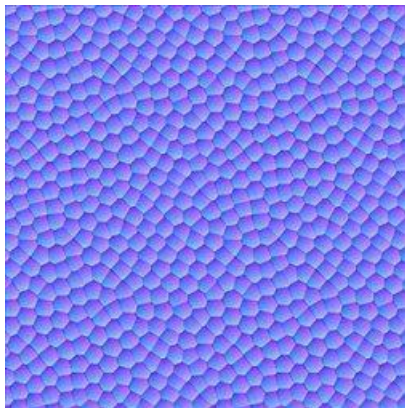
- nedostaci
 - silueta objekta nije ispravno prikazana
 - samozaklanjanje teksture nije ispravno
 - promjene uslijed paralakse
 (pomaci uslijed promjene pogleda)

- <http://www.echalk.co.uk/amusements/OpticalIllusions/illusions.htm>
- <https://sketchfab.com/models/8718c651666c4c4a9972a2dd393b759f>



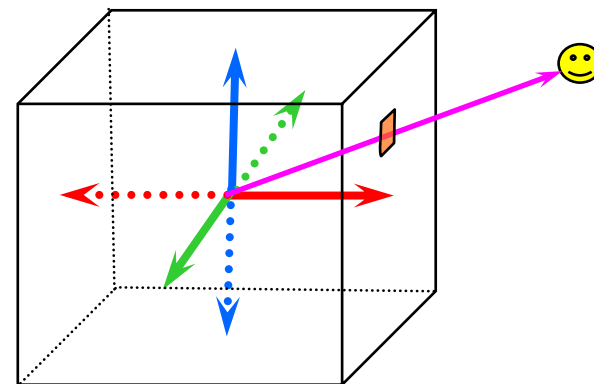
Mapa normala

- RGB komponente teksture se koriste za pohranjivanje normale
 - vrijednosti x, y, z komponenti normiranog vektora su $[-1,1]$, a želimo koristiti RGB komponente koje su $[0,1]$, pa moramo skalirati na zadani opseg $r_x = (x+1)/2$, $r_y = (y+1)/2$, $r_z = (z+1)/2$
 - npr. vektor $n = (0 0 1)$ se prikazuje kao $r_n = (0.5 0.5 1)$,
 - općenito dominira z komponenta, pa zato dominira plava boja
<http://lo-th.github.io/loth/shader/index.html#>
- što smo dobili
 - normala će se interpolirati između zadanih vrijednosti kao što se interpoliraju RGB komp. (sklopovski podržano)
 - neće nužno ostati ortonormirana baza



Normalizacija vektora

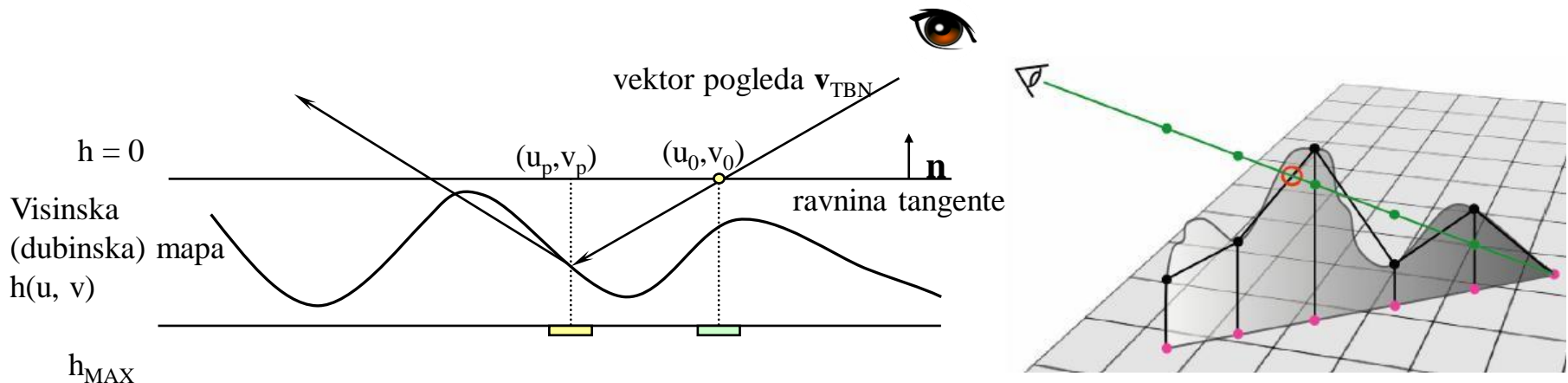
- koristimo preslikavanje s kocke (engl. cube maps)
(npr. funkcija `sin`, `cos` često se pohranjuje u tablice kako bi se ubrzalo izračunavanje)
- želimo brzo izračunati normirani vektor $n_n = n / \|n\|$
- koristimo preslikavanje na 6 ravnina kocke (sklopovski podržano)
pristupamo s tri koordinate tipa `float glTexCoord3f` umjesto `glTexCoord2f`
- u središtu kocke je ishodište
- imamo 6 ravnina teksture u koje pohranimo vrijednosti normiranih vektora koje unaprijed izračunamo za sve stranice (teksture) kocke
- za vektor (ljubičasti) čiju normalu tražimo, preslikavanje s kocke će nam vratiti normiranu vrijednost (narančasti element teksture)
- npr. probodište vektora (1 2 1) isto je kao za vektor (2 4 2), odnosno općenito (k 2k k)
i daje normiranu vrijednost koju smo pohranili u teksturu
- GLSL – `normalize()`



7.3 Preslikavanje izbočina uz zaklanjanja uslijed paralakse

(*engl. parallax occlusion mapping*)

- paralaksa (grč. promjena) – kada neki predmet gledamo iz dva različita kuta vidimo ga različito – neki dijelovi će biti zaklonjeni ovisno o pogledu
- vidljiva srednje-gruba svojstva površine
- ostvareno pomacima slikovnih elemenata teksture (*fragment shader*)
- dubinska mapa (*depth map*) određuje veličinu pomaka pojedinog slikovnog elementa
- projekcija vektora prema promatraču određuje smjer pomaka slikovnog elementa
- određivanje zasjenjenih dijelova <http://www.babylonjs-playground.com/#10I31V#23>
- http://callumprentice.github.io/apps/webgl_terrain/index.html
- <https://www.awwwards.com/inspiration/webgl-3d-parallax-effect-missing>



Preslikavanje uz zaklanjanje

Pomicanje elemenata teksture određeno dubinskom mapom

- jednostavna aproksimacija (nije egzaktno)
- h_0 je određen dubinskom mapom h $[0, 1]$ za (u_0, v_0)
- T'_0 koordinate izvornog elementa na poziciji (u_0, v_0) - iscrtat ćemo teksturu elementa $T'_p(u'_p, v'_p)$
- T'_p koordinate elementa kojeg pomičemo – pomiče se u smjeru projiciranog vektora prema promatraču na ravninu poligona za iznos
- mali kutovi su problematični http://ming4883.github.io/exp3/examples/demo_derivative_maps.html
(texture- pyramids, bumpness)

Egzaktno :

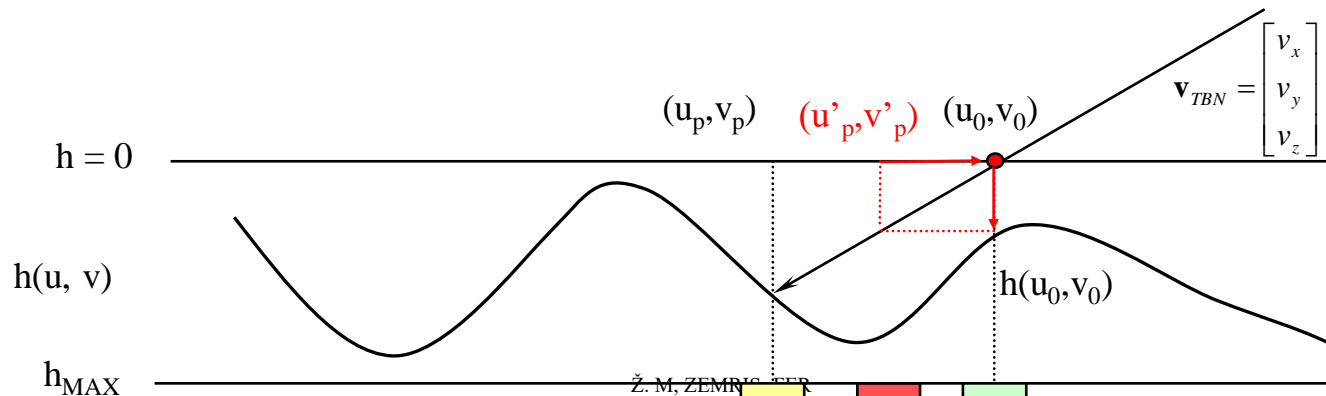
$$\begin{bmatrix} u_p \\ v_p \\ -h_p \end{bmatrix} = \begin{bmatrix} u_0 \\ v_0 \\ 0 \end{bmatrix} + t \cdot \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$$

Aproksimacija :

$$\begin{bmatrix} u'_p \\ v'_p \\ -h_0 \end{bmatrix} = \begin{bmatrix} u_0 \\ v_0 \\ 0 \end{bmatrix} + t \cdot \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \Rightarrow t = \frac{-h_0}{v_z} \Rightarrow$$

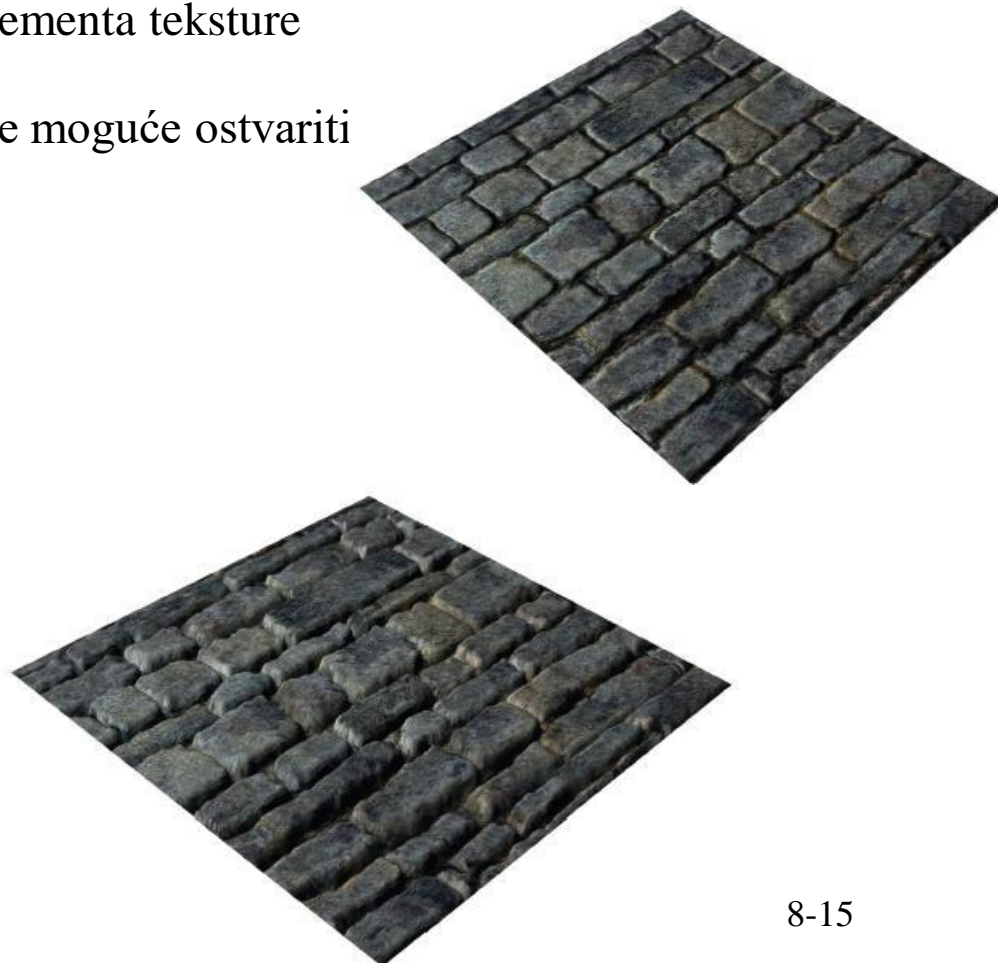
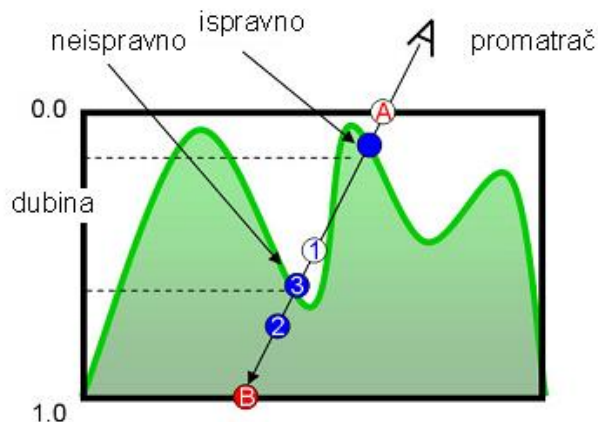
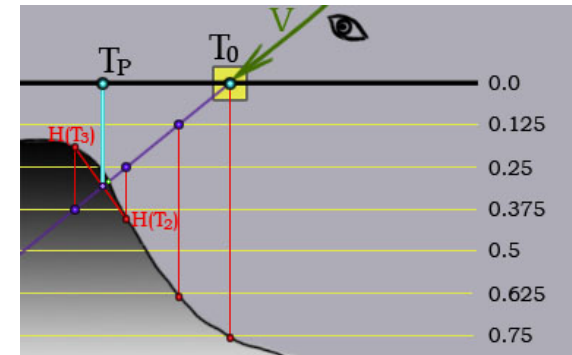
Na mjestu u_0, v_0 iscrtamo teksturu u'_p, v'_p :

$$\begin{bmatrix} u'_p \\ v'_p \end{bmatrix} = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} - \frac{h_0}{v_z} \cdot \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$



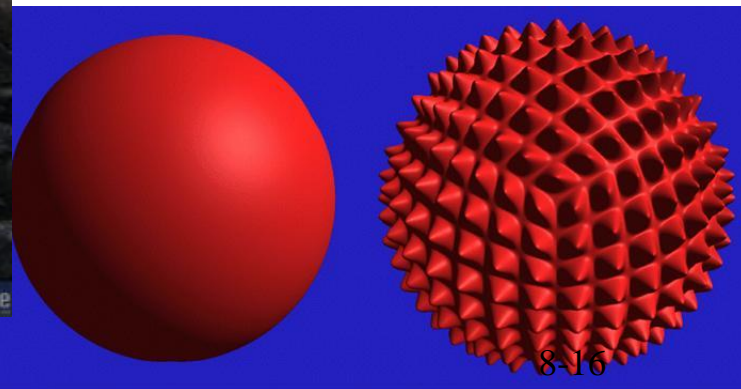
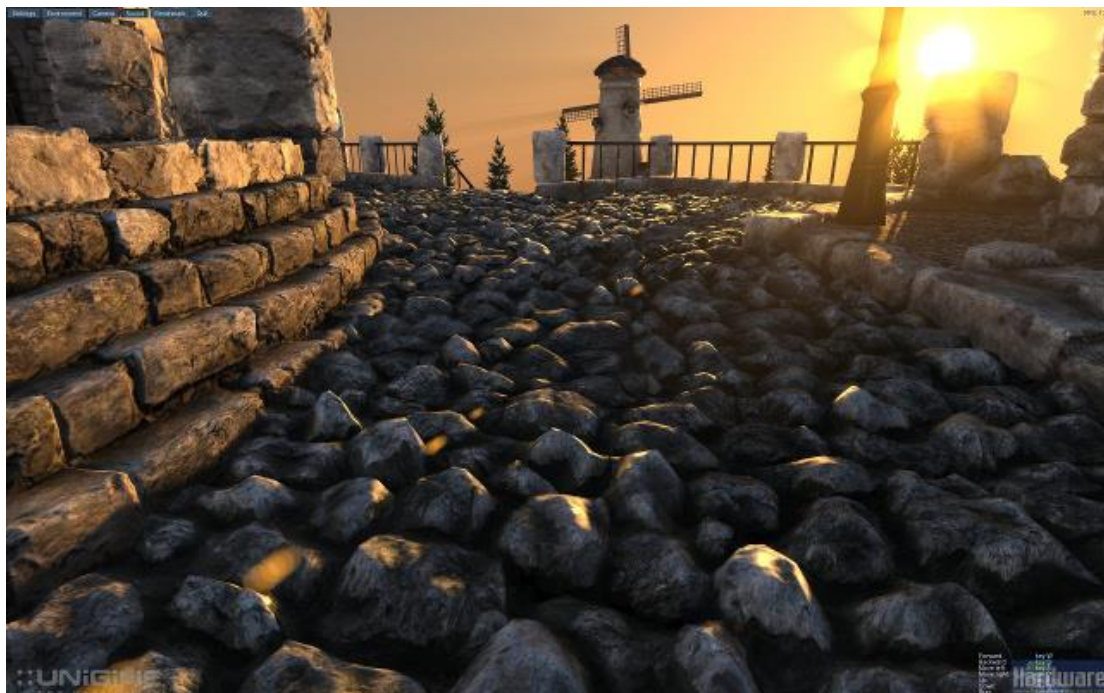
Reljefna tehnika (*relief mapping*)

- određivanje sjecišta praćenjem zrake u dubinskoj mapi
 - jednolika podjela duž zrake (Steep Parallax Mapping),
 - + binarna podjela <https://playground.babylonjs.com/#10131V#408>
 - kombinirano
- perspektivno ispravna udaljenost elementa teksture
- <http://lo-th.github.io/labs/index.html> (crveno-parallax/plane)
- ispravne sjene na teksturi i siluetu je moguće ostvariti
- dvostruke visinske mape [Lanci](#)



7.4 Preslikavanje pomaka (*engl. displacement mapping*)

- promjena geometrije, olakšava posao dizajnera
<http://felixpalmer.github.io/webgl-tombstone/> (zoom/toggle vertices /Load depth map) <http://www.babylonjs.com/Demos/DisplacementMap/>
- generiranje izbočina objekta (može biti i teren)
http://www.realtimerendering.com/erich/udacity/exercises/unit6_normal_vs_displacement.html
- https://threejs.org/examples/#webgl_materials_displacementmap (displ.)
- promjene broja poligona unutar protočnog sustava (*geometry shader*) <http://david.li/waves/>
- https://www.youtube.com/watch?v=1T8_kvD5X8

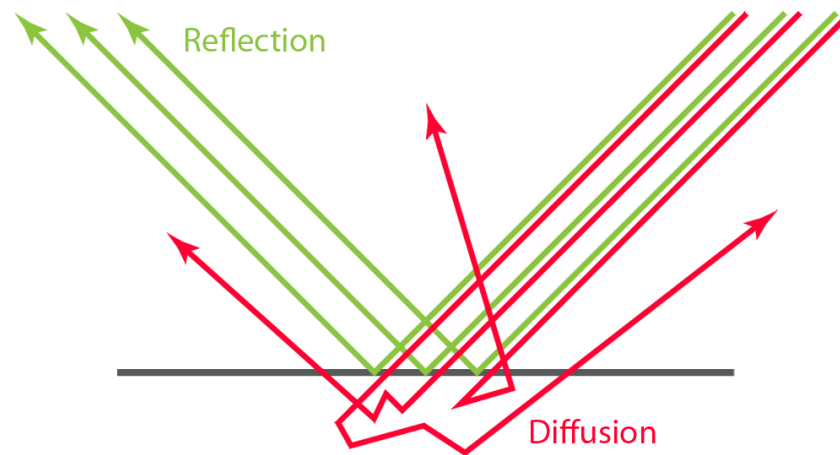


Ispod površinsko raspršenje (engl. subsurface scattering)

- karakteristično za led, mramor, ljudsku kožu
- dio svjetlosti je apsorbiran u površini (ovisno o valnim duljinama)

https://phet.colorado.edu/sims/html/beers-law-lab/latest/beers-law-lab_en.html

https://threejs.org/examples/webgl_materials_subsurface_scattering



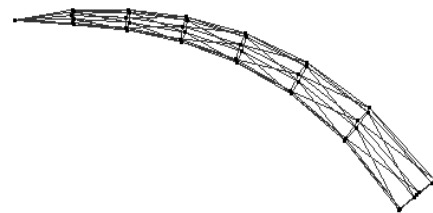
- <https://www.youtube.com/watch?v=tuZMMZ8vbNk>

- [http://fractalfantasy.net/#/4/uncanny valley](http://fractalfantasy.net/#/4/uncanny_valley)

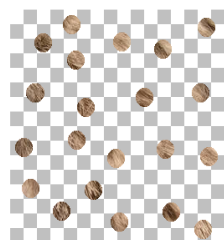


7.5 Volumne teksture - krzno

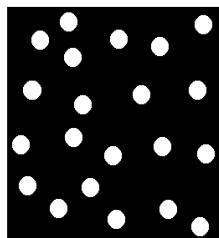
- simulacija krzna linijama
<http://www.ibiblio.org/e-notes/webgl/models/fur.html>
https://oosmoxiecode.com/archive/js_webgl/lines_particles_sphere/
- modeliranje pojedine travke poligonima
<https://codepen.io/zadvorsky/full/xPKBBJ//>
- modeliranje „zgužvanim” poligonima <https://www.babylonjs-playground.com/#EUCNP#6>
- simulacija krzna, trave, tepiha, oblaka – nizom paralelnih tekstura
http://jeromeetienne.github.io/threejs_grass/examples/demo.html
- krzno uzorkujemo nizom presjeka
- iznad poligona složimo niz tekstura (ljuske – shells) koje su djelomično prozirne
<https://keaukraine.github.io/webgl-fur/> Layers



originalna tekstura modela



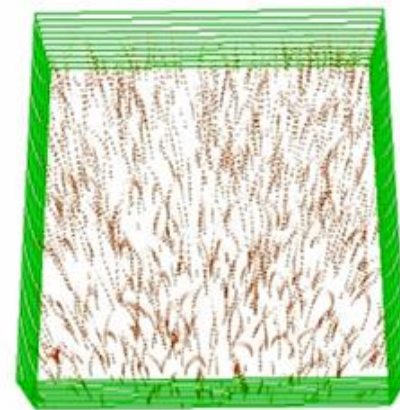
tekstura ljuske



maska



volumni uzorak krzna

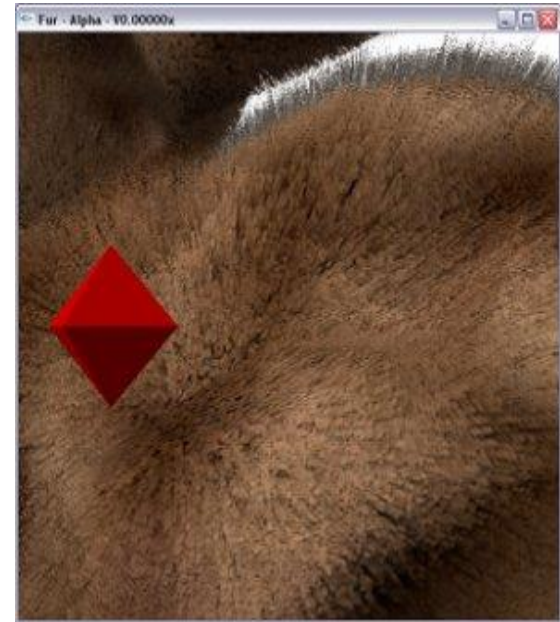
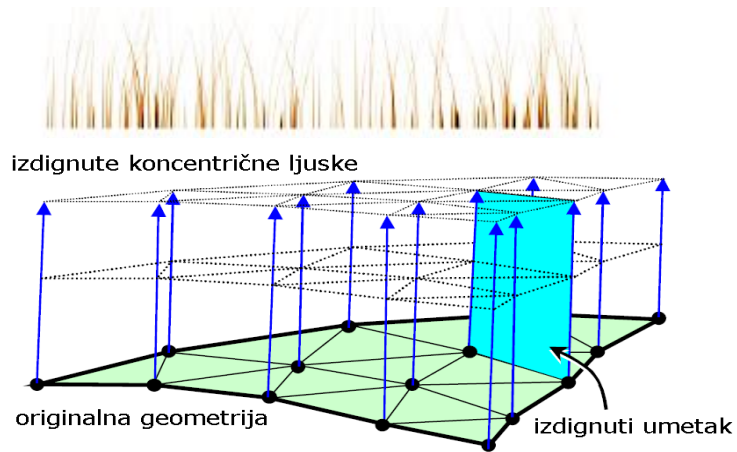


uzorkovanje tekstura ljusaka

- na silueti objekta dodajemo umetke s rubnom teksturom kako bi poboljšali rubna područja

https://oosmoxiecode.com/archive/js_webgl/fur/index_petting.html

https://oosmoxiecode.com/archive/js_webgl/fur/

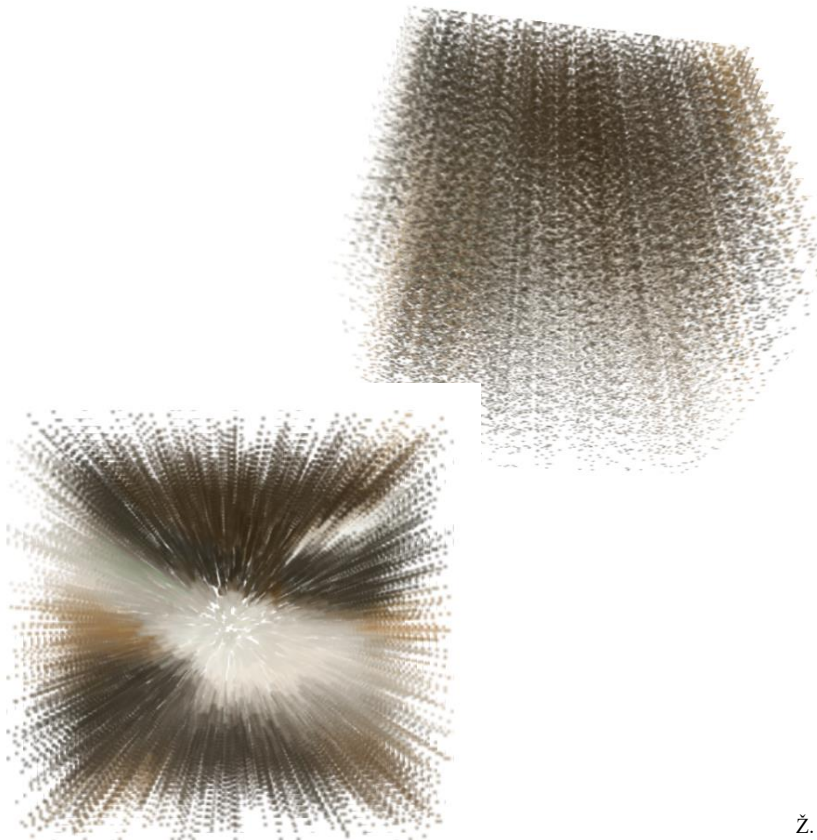


- utjecaj na boju i normalu teksture,



- utjecaj na normale poligona

- prednosti postupka – prikaz mekih objekata u stvarnom vremenu, sklopovska podrška, impresivni rezultati
- nedostaci postupka – pod malim kutovima alias artefakti
- [Zeko](#)



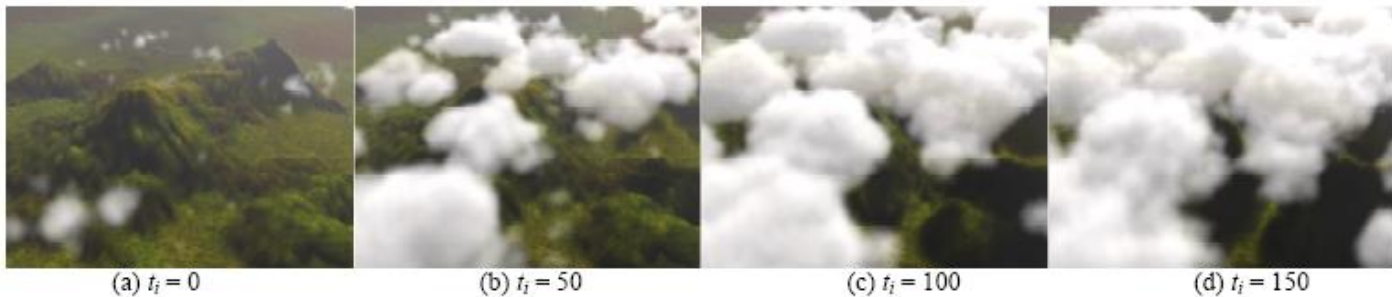
dim, oblaci ostvareni volumnom teksturom

<http://www.babylonjs.com/Demos/VertexData/>

<https://www.script-tutorials.com/demos/177/index.html>

Animirajuće teksture

- datotečni zapis .gif podržava animirajući slijed slika
 - npr. vatra, dim kao film <https://stemkoski.github.io/Three.js/Texture-Animation.html>
<https://cedricpinson.github.io/osgjs-website/examples/texture-video/>
- transformacije koje djeluju na preslikavanje teksture mogu se primijeniti i na video [Kino](https://stemkoski.github.io/Three.js/Video.html) <https://stemkoski.github.io/Three.js/Video.html> P
http://alteredqualia.com/three/examples/webgl_deferred_arealights_texture.html
- https://threejs.org/examples/#webgl_materials_video
- 360° video (cubemaps, skybox) <https://www.youtube.com/watch?v=sPyAQQklc1s>
• https://threejs.org/examples/#webgl_video_panorama_equirectangular
- animirajuće volumne teksture
 - simulacija stvaranja oblaka [Oblaci](https://www.clicktorelease.com/code/css3dclouds/#) <https://www.clicktorelease.com/code/css3dclouds/#>
 - https://threejs.org/examples/webgl2_volume_cloud.html
https://threejs.org/examples/#webgl2_materials_texture3d_partialupdate



- tekstura kao izvor sustava čestica

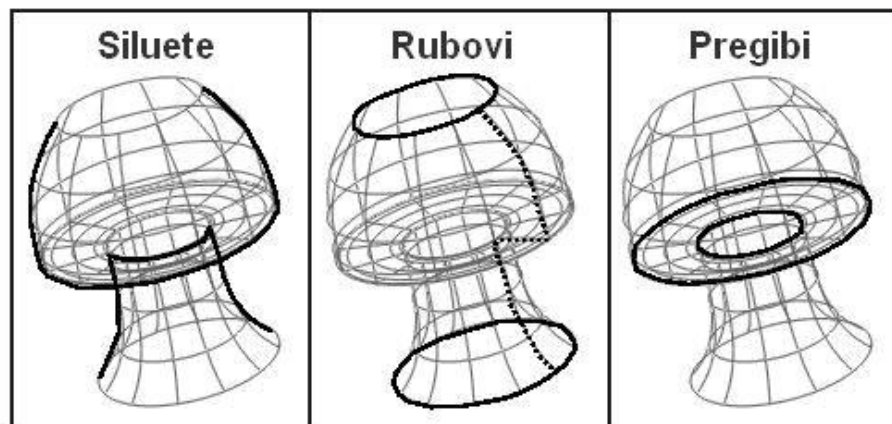
- <https://nme.babylonjs.com/#345ATT#4>

7.6 Nefotorealistične tehnike prikaza (*Non Photorealistic Rendering NPR*)

- Fotorealistične tehnike – cilj je predmete prokazati što vjernije
- NPR tehnike – naglašavanje bitnih informacija za raspoznavanje predmeta – pregibi i rubovi – naglašavanje oblika (skice strojarskih elemenata)
- ciljano oponašanje estetskog dojma tradicionalnih tehnika – slično stripu, kistom crtanih objekata, crtića http://www.realtimerendering.com/erich/udacity/exercises/unit3_toon_solution3.html

- tehnike prikaza

- rezbarenje,
- sjenčanje točkama,
- crtanje kistom, ugljenom,
- <http://david.li/paint/>
- kolaž, crtež <https://threejs.org/examples/>
- tonske mape <http://takahirox.github.io/mmd->



- analiza tijela prije iscrtavanja

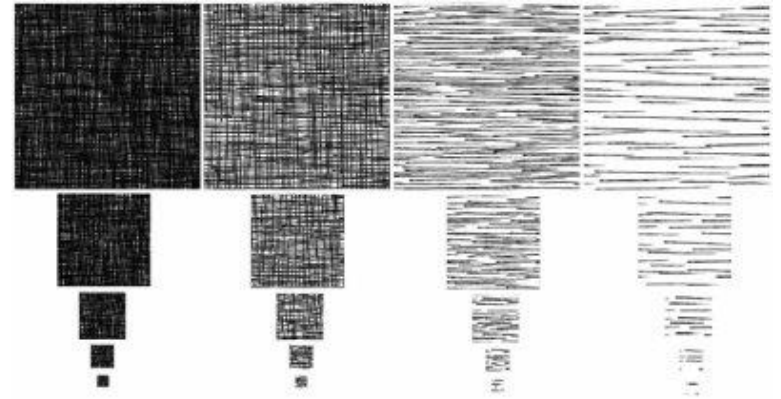
- obris (kontura, silueta) http://arefin86.github.io/edges_silhouettes.html
<http://stemkoski.github.io/Three.js/Outline.html>

3D - granica vidljivih i skrivenih poligona ili projekcija - Z spremnik

- rubovi – poligoni koji nemaju susjednih poligona imaju rubne bridove
- pregibi – nagla promjena normale, kut između normala veći od nekog unaprijed zadanog
- analiza zakrivljenosti

Prikaz tonskim mapama

- priprema tekstura – MipMape tekstura se unaprijed pripreme
 - dočarano osvjetljenje i razina detalja
 - toroidne sličice – moraju omogućavati popločavanje bez šavova gore-dolje lijevo-desno
- <https://www.clicktorelease.com/code/cross-hatching/>
- [Medo](#)
- primjena tekstura ovisno o zadanim pravilima
 - prostorna koherentnost – ostvarivanje prijelaza tekstura između različitih nijansi pri prijelazu sa svjetlijeg na tamniji vrh i detaljnosti teksture (prijelazi moraju biti kontinuirani)
 - potrebno anizotropno filtriranje
 - najveća lokalna zakrivljenost površine (diferencijalna geometrija) određuje smjer poteza na teksturi



<http://tangrams.github.io/tangram-sandbox/tangram.html?styles/tilt-ikeda#12.13511021767367/45.8244/15.9619>

– vremenska koherentnost

- u animaciji, teksture tamnijih nijansi sadrže svjetlije, u novoj nijansi dodaju se potezi na već postojeće,
- veće teksture su nadskup manjih
- stapanje više tekstura kako bi se umanjila kvantizacija osvjetljenja, zasebne teksture 3 vrha poligona Baricentrično se kombiniraju i stapaju u jednu teksturu (*stapanjem a b i c dobije se konačna tekstura d*)

