

TÖL105M TÖLVUGRAFÍK

Fyrirlestur 14: Víðsjón og skuggar

Hjálmtyr Hafsteinsson
Haust 2024



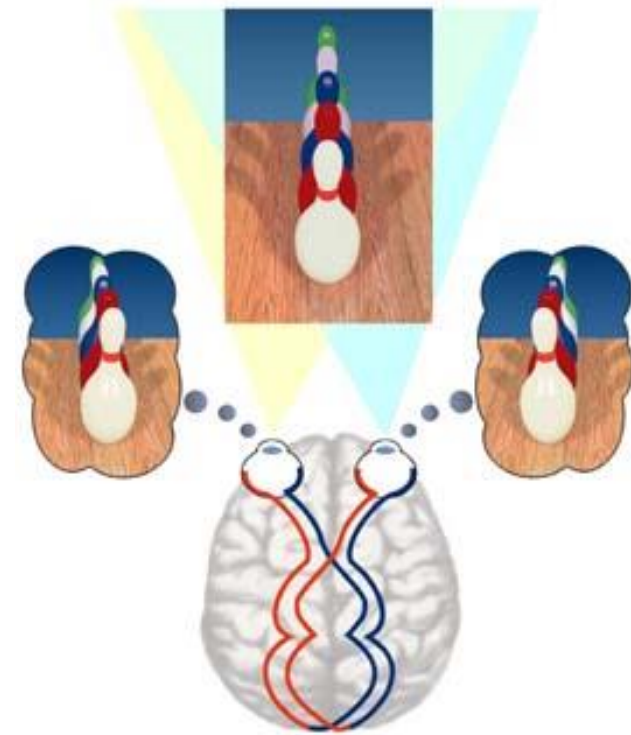
- Víðsjón (*stereoscopic vision*)
 - Dýptaráhrif með tveimur myndum
- Falin yfirborð
 - Bakhliðareyðing (*backface culling*)
- Skuggar (*shadows*)
 - Skuggar með ofanvarpi
 - Skuggakort (*shadow maps*)

aukaefni

5.8

5.10, 5.11

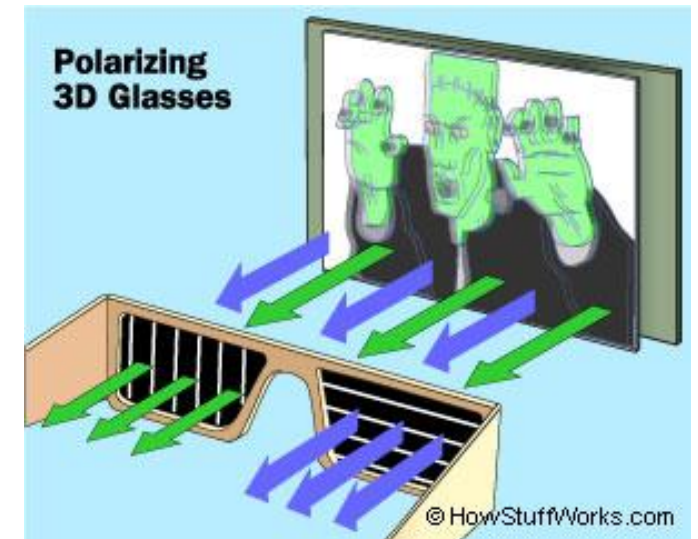
- Augu okkar sjá umhverfið frá örlítið mismunandi sjónarhorni
 - Að meðaltali 63.5 mm fjarlægð milli sjáaldra (*pupils*)
- Heili okkar vinnur fjarlægðarupplýsingar úr þessum mismun
 - Notar einnig aðrar upplýsingar
 - Vöðvaspennu við að finna fókus
 - Vöðvaspennu við að snúa augunum svo bæði horfi á sama hlut
 - Lýsing og mynstur hluta



- Búa til tvær tvívíðar myndir af sömu sviðsmynd frá mismunandi sjónarhornum
 - Hafa fjarlægð milli auganna "rétt" og lárétta í xz-sléttunni
- Koma myndunum í sitthvort augað
 - Hefur verið erfiðasta verkefnið hingað til
 - Ýmsir möguleikar:
 - Nota "prívíddargleraugu" (eins og í kvikmyndahúsum í dag)
 - Nota sýndarveruleikagleraugu (VR)
 - Nota mislit gler

Þrívíddargleraugu

- Nota skautað ljós (*polarized light*)
 - þ.e. ljós sem sem sveiflast aðeins í einni sléttu
- Önnur myndin er með einni skautun og hin myndin með annari
- Síur í gleraugunum hleypa aðeins í gegn ljósi með rétta skautun
 - Ef horft er á myndina án gleraugna, þá virðist hún frekar óskýr



Frá physics.org

Kostur: Hægt að ná nokkuð góðum aðskilnaði milli augna

Ókostur: Verðum að nota gleraugun

Sýndarveruleika"gleraugu"

- Sýna hvoru auga aðeins eina mynd
- Hægt að gera það á ýmsa vegu:



[View-Master](#)



[Google Cardboard](#)



[Meta Quest 3](#)



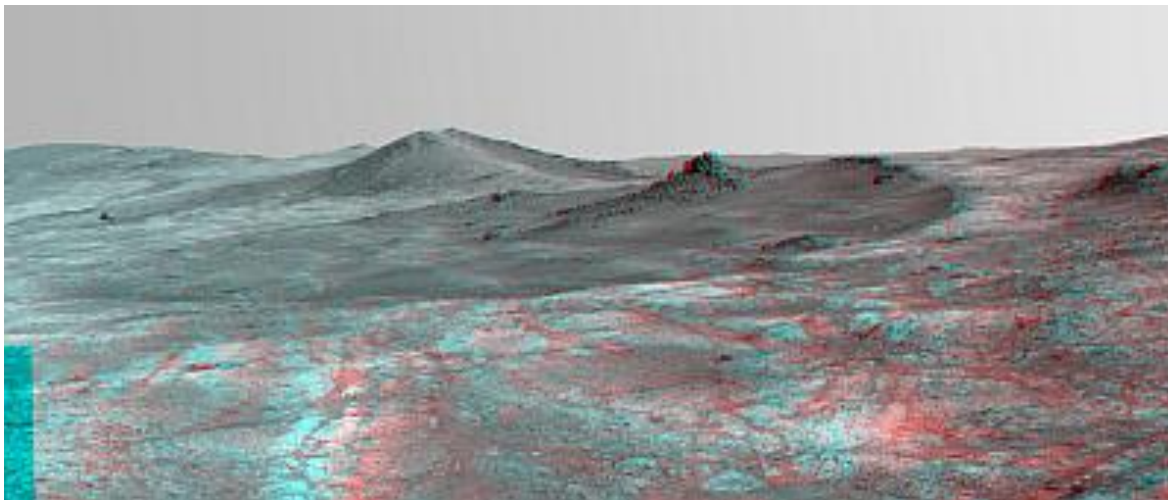
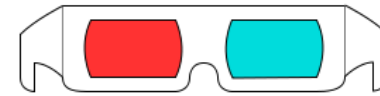
[Apple Vision Pro](#)

Kostur: Fullkominn aðskilnaður augna

Ókostur: Óþægileg í notkun til lengri tíma

Aðskilnaður með litum (*anaglyph*)

- Birta aðra myndina í rauðum lit og hina í blágrænum (*cyan*)
- Til ódýr gleraugu með linsum sem sía út þessa liti
 - Rauða linsan síar burt rauðan lit í myndinni
 - Sjáum þá aðeins blágræna litinn
 - Blágræna linsan síar burt blágræna litinn í myndinni
 - Sjáum þá aðeins rauða litinn



Sjá: [NASA](#)

[Sýniforrit](#) frá [Lee Stemkoski](#)

[Gamla myndir](#) frá [Brooklyn stereography](#)

- Víragrindarteningur teiknaður tvisvar
 - í mismunandi lit frá tveimur sjónarhornum

```
function render() {  
  ...  
  // Vinstra auga...  
  var mv = lookAt( vec3(0.0-eyesep/2.0, 0.0, zDist),  
                  vec3(0.0, 0.0, zDist+2.0),  
                  vec3(0.0, 1.0, 0.0) );  
  mv = mult( mv, mult( rotateX(spinX), rotateY(spinY) ) );  
  
  gl.uniform4fv( colorLoc, vec4(1.0, 0.0, 0.0, 1.0) );  
  gl.uniformMatrix4fv(mvLoc, false, flatten(mv));  
  gl.drawArrays( gl.LINES, 0, NumVertices );  
  
  // Hægra auga...  
  ...  
}
```

Breytan **eyesep** er
bilið milli augnanna

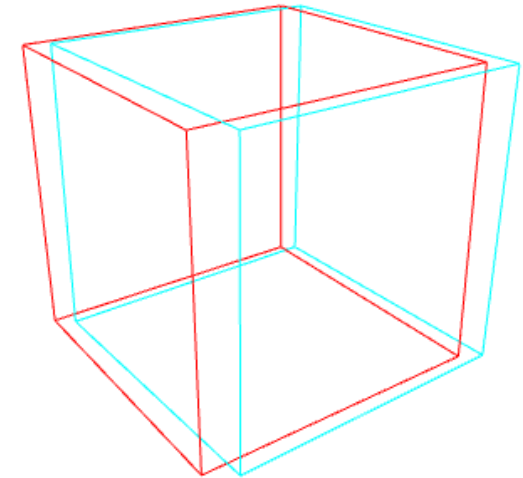
Teiknum í rauðu

Teiknum teninginn
sem 12 línur

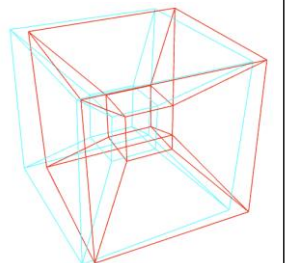
Sjá [stereocube](#)

Fyrir hægra augað:

- Breyta áhorfsstaðsetningu
- Breyta lit í blágrænan (cyan)



Enn flottara:
[4D-stereocube](#)



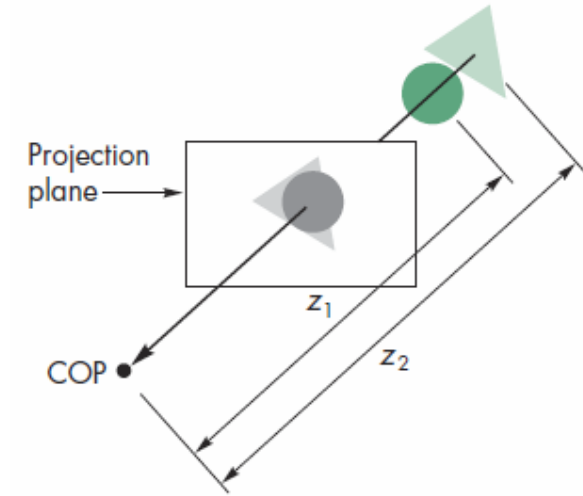
1. Linsurnar í 3-víddarglæraugum eru oftast rauð og blágræn (*cyan*). Af hverju eru þessir litir valdir?
2. Skiptir teikniröð þríhyrninga einhverju máli þegar dýptarminni (*z-buffer*) er notað?
3. Hálfskuggar (*penumbra*) koma vegna þess að ljósgjafi er (oftast) ekki óendalega lítill punktur. Hvernig væri hægt að útfæra þá með ofanvarpi?

Falin yfirborð (*hidden surfaces*)

- Viljum ekki birta þau yfirborð sem eiga ekki að sjást frá auganu
 - Gerist ekki af sjálfu sér!
- Tvær lausnir:
 - Dýptarminni (*z-buffer*)
 - Ákvarðar hvaða þríhyrningar sjást frá auganu, út frá z-hniti þeirra í sjónhnitum
 - Bakhliðareyðing (*backface culling*)
 - Hendir burt öllum þríhyrningum sem snúa frá auganu
 - Þeir eru ósýnilegir frá auganu (framhliðarnar fela þá)

Notum oftast
báðar aðferðirnar

- Dýptarminnið d er jafnstórt og skjáminnið (*frame buffer*) s
- Hólf $d[i, j]$ inniheldur dýpi (z-gildi) þess bútar sem er með lit í $s[i, j]$
- Dýptarminnið er upphafsstillt sem $+\infty$
- Förum í gegnum alla búta allra þríhyrninga
 - Ef z-gildi bútar (i, j) er minna en það sem er í $d[i, j]$, þá fer litur hans í $s[i, j]$, annars ekki



Í kóða

```
gl.enable(gl.DEPTH_TEST);  
...  
gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);  
...
```

1. Linsurnar í 3-víddargleraugum eru oftast rauð og blágræn (*cyan*). Af hverju eru þessir litir valdir?
2. Skiptir teikniröð þríhyrninga einhverju máli þegar dýptarminni (*z-buffer*) er notað?
3. Hálfskuggar (*penumbra*) koma vegna þess að ljósgjafi er (oftast) ekki óendalega lítill punktur. Hvernig væri hægt að útfæra þá með ofanvarpi?

- Flestir þrívíðir hlutir í tölvugrafík eru margflötungar (*polytope*)
 - Samsettir úr grind af þríhyrningum
- Þeir þríhyrningar hlutanna sem snúa að auganu sjást, en hinir ekki
 - Getum því hent þeim þríhyrningum í burtu
 - Losnum við um helming allra þríhyrninga!

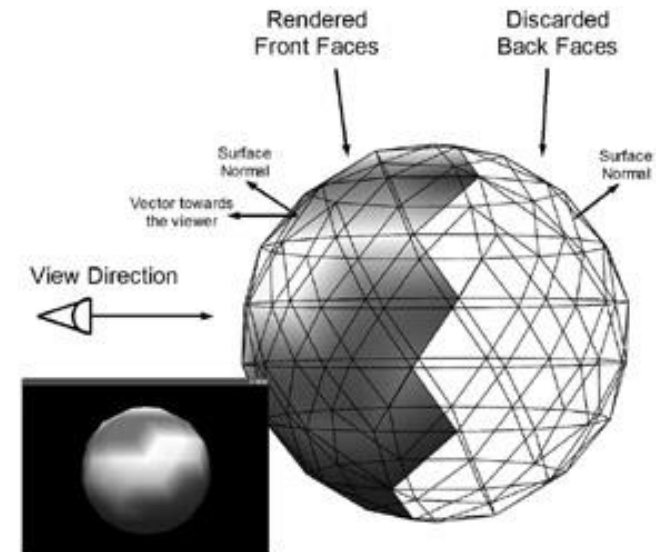
Kveikja á
bakhliðareyðingu

Eyða bakhliðum

Hnútaröð framhliðar

```
gl.enable(gl.CULL_FACE);  
gl.cullFace(gl.BACK);  
gl.frontFace(gl.CCW);  
...
```

Sjá [cube-cull](#)

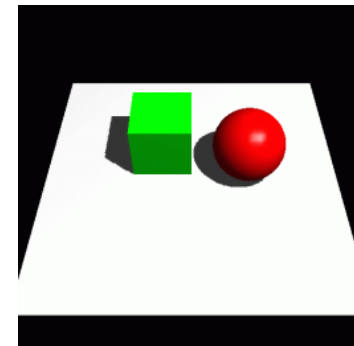
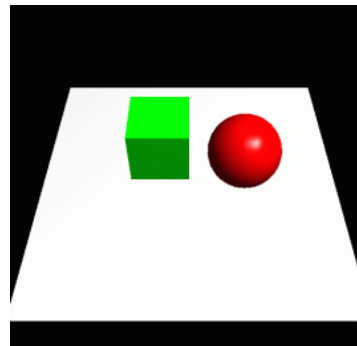


- Förum í ljós og lýsingu næst, en...
- Ein leið til að búa til skugga í tölvugrafík byggir á ofanvarpi
 - Við munum því skoða það hér
- Aðrar leiðir til að búa til skugga:
 - Skuggakort (*shadow maps*)
 - Skuggarúm (*shadow volumes*)

Sleppum þessari
aðferð hér

Gefur meiri raunveruleikatilfinningu

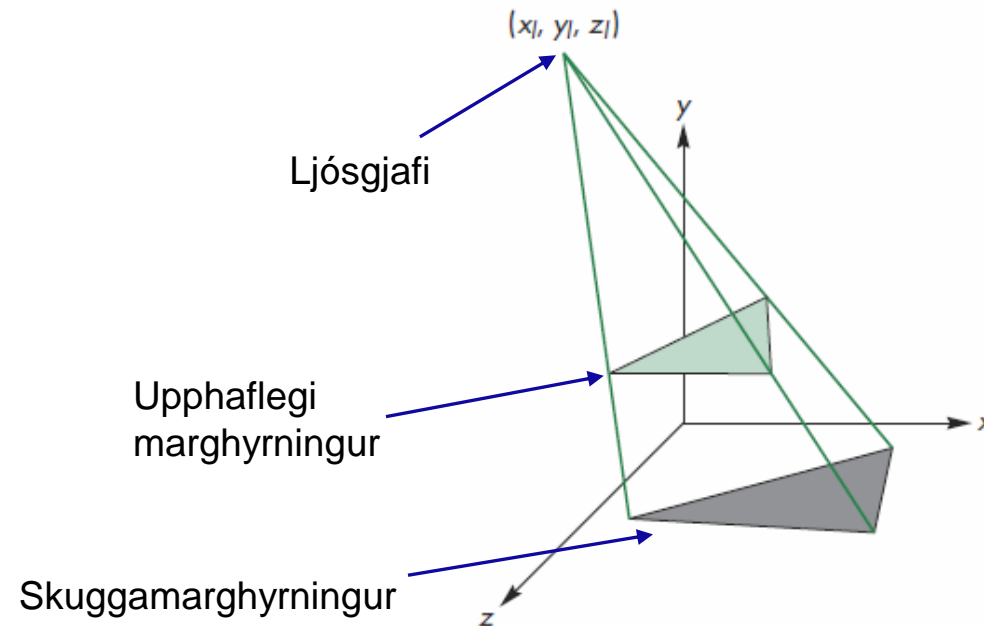
Hjálpar við að skilja staðsetningu



- Skuggar eru víðvært (*global*) fyrirbæri
 - Hlutur A varpar skugga á hlut B
 - Þegar B er teiknaður þarf að vita af staðsetningu A miðað ljósgjafann
 - Þetta passar illa inn í grafíkpípuna
 - Hún teiknar bara einn búa (skjápunktur) í einu
 - Myndi tefja grafíkpípuna mikið ef fyrir hvern einasta búa þyrfti að kanna alla aðra hluti í líkaninu!
 - Skuggar passa ágætlega inn í geislaöölun (*ray tracing*)
 - Þar er hver skjápunktur litaður út frá öllu líkaninu

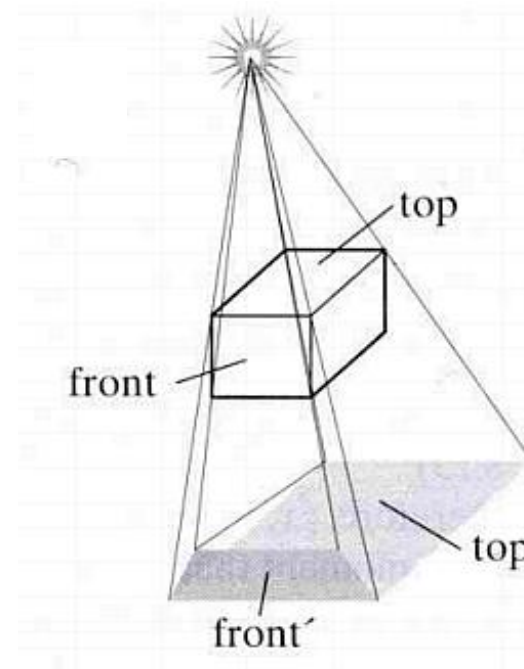
Enda er geislaöölun mjög dýr!

- Elsta aðferðin í tölvugrafík til að búa til skugga
 - Upphaflega notuð í flughermum til að gefa sjónrænar vísbendingar
- Ofanvarp marghyrnings á sléttu er annar marghyrningur
 - Kallast skuggamarghyrningur



- Virkar fyrir flöt yfirborð sem eru lýst með punktljósi
- Skuggi hlutar:
 - Fyrir hverja hlið hlutarins, finna skuggamarghyrning
 - Skuggi hlutarins er sammengi skuggamargyrninganna

Gerum ráð fyrir að skugginn komi á xz-sléttuna (þ.e. $y=0$)



Útreikningur á ofanvarpsskugga

- Ljósgjafi er í (x_l, y_l, z_l)
- Skugginn kemur á yfirborðið $(x_p, 0, z_p)$

Hliðra svo að ljósgjafi sé í núllpunkti:

Framkvæma ofanvarp á $y=0$ sléttuna:

$$\mathbf{q} = M\mathbf{p}$$

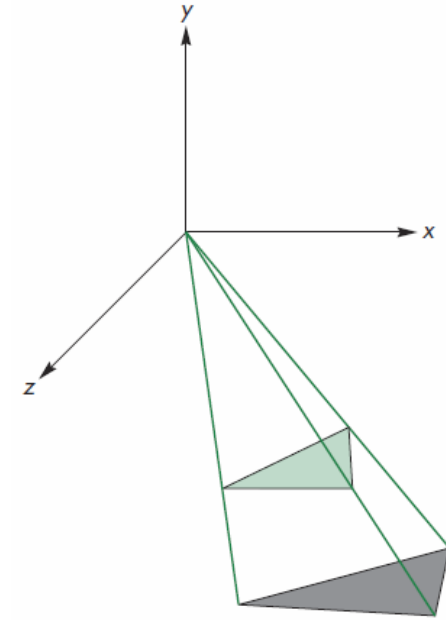
$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{-y_l} & 0 & 0 \end{bmatrix}$$

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\text{þá } \mathbf{q} = \begin{bmatrix} x \\ y \\ z \\ -y/y_l \end{bmatrix}$$

deila í gegn með
jafnþætta hnit:

$$\mathbf{q}' = \begin{bmatrix} x \\ -y/y_l \\ -y_l \\ z \\ -y/y_l \\ 1 \end{bmatrix}$$



1. Hliðra ljósgjafa í núllpunkt:
2. Sjónhornsofanvarp á $y=0$:

$$T(-x_l, -y_l, -z_l)$$

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{-y_l} & 0 & 0 \end{bmatrix}$$

3. Hliðra til baka:

$$T(x_l, y_l, z_l)$$

Heildarskuggavörpun

$$M = \begin{bmatrix} 1 & 0 & 0 & x_l \\ 0 & 1 & 0 & y_l \\ 0 & 0 & 1 & z_l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{-y_l} & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x_l \\ 0 & 1 & 0 & -y_l \\ 0 & 0 & 1 & -z_l \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Reikna út líkanavörpun fyrir flöt
- Senda líkanafylki á GPU og teikna flöt
- Reikna skuggaofanvarp fyrir flöt
- Senda ofanvarpsfylki á GPU og teikna flöt

} Eins og venjulega

Teiknum skuggann
sem sjálfstæðan hlut!

Búa til fylki fyrir
skuggaofanvarp:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{-y_i} & 0 & 0 \end{bmatrix}$$

```
var light = vec3(0.0, 2.0, 0.0);  
m = mat4();  
m[3][3] = 0;  
m[3][1] = -1/light[1];
```

```
function render() {
```

```
  ...
```

```
  mv = lookAt(eye, at, up);
```

```
  gl.uniformMatrix4fv( mvLoc, false, flatten(mv) );
```

```
  gl.uniform4fv(fColor, flatten(red));
```

```
  gl.drawArrays(gl.TRIANGLE_FAN, 0, 4);
```

```
  light[0] = Math.sin(theta);
```

```
  light[2] = Math.cos(theta);
```

```
  mv = mult(mv, translate(light[0], light[1], light[2]));
```

```
  mv = mult(mv, m);
```

```
  mv = mult(mv, translate(-light[0], -light[1], -light[2]));
```

```
  gl.uniformMatrix4fv( mvLoc, false, flatten(mv) );
```

```
  gl.uniform4fv(fColor, flatten(black));
```

```
  gl.drawArrays(gl.TRIANGLE_FAN, 0, 4);
```

```
  ...
```

```
}
```

Venjuleg líkanavörpun

Teikna rauðan fering

Snúa ljósgjafa í hringi
í y=2 sléttunni

Skugga-
ofanvarp

Teikna svartan fering

Sjá [shadow](#)

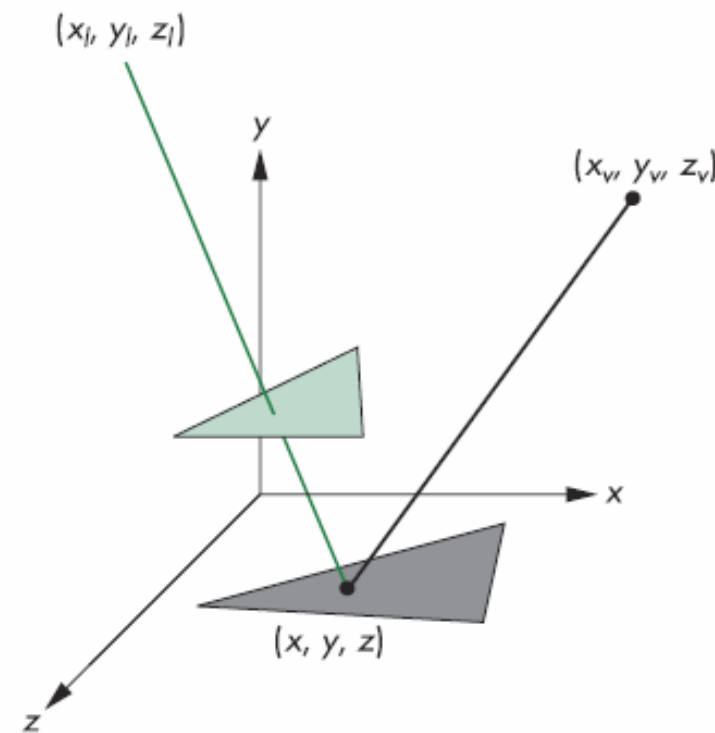
Sjá [shadowHH](#)



Skuggakort (*shadow maps*)

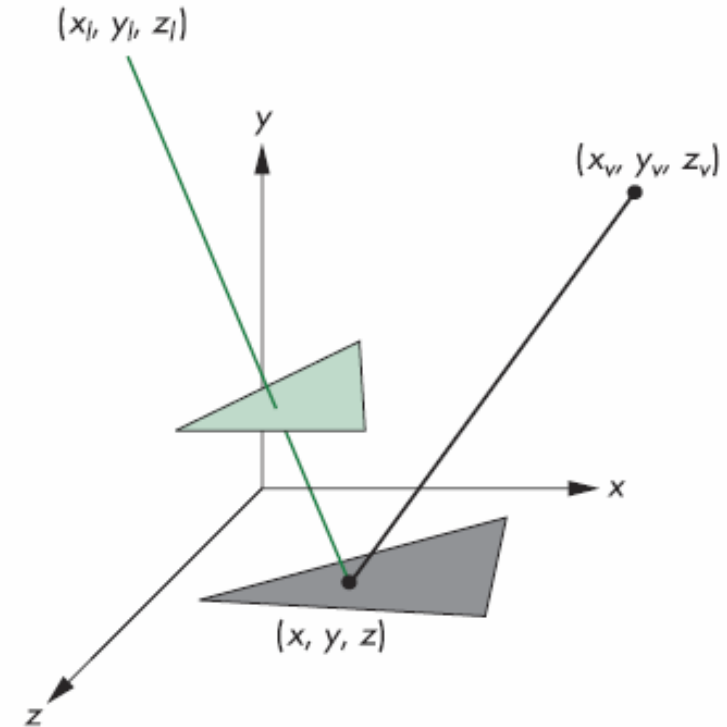
- Ef við teiknum allt líkanið frá sjónarhorni ljósgjafa vitum við hvaða yfirborð eru efst
 - Upplýsingar um það koma í dýptarminni (*z-buffer*)
- Köllum þessar upplýsingar skuggakort
 - Inniheldur reyndar upplýsingar um það hvaða bútar eru *lýstir* af ljósgjafanum
 - Skuggakortið er geymt sem mynstur (*texture*)

Förum síðar í mynstur!



Skuggakort, framh.

- Þegar litur á búi (x, y, z) er ákvarðaður berum við fjarlægð (x, y, z) frá ljósgjafa við gildið í skuggakortinu
- Ef fjarlægðin í skuggakortinu er minni þá er einhver hlutur á milli (x, y, z) og ljósgjafans
 - Ef svo er, þá er (x, y, z) í skugga og verður svartur (eða dökkur)
 - Annars er (x, y, z) litaður eins og venjulega



1. Linsurnar í 3-víddargleraugum eru oftast rauð og blágræn (*cyan*). Af hverju eru þessir litir valdir?
2. Skiptir teikniröð þríhyrninga einhverju máli þegar dýptarminni (*z-buffer*) er notað?
3. Hálfskuggar (*penumbra*) koma vegna þess að ljósgjafi er (oftast) ekki óendalega lítill punktur. Hvernig væri hægt að útfæra þá með ofanvarpi?