Konputagailu bidezko grafikoak

Joseba Makazaga   Ander Murua

Computer graphics

Grafikoak, imajinak edo marrazkiak erabiltzen edo eta sortzen dituen aplikazioak sortzeaz arduratzen da.

1. Grafikoak eta irudiak manipulatzen dituzten aplikazioak.

   Informazio grafikoa darabiltenak

2. Emaitzak adierazteko grafikoak erabiltzen dituztenak.

   Informazioaren erabilera grafikoa

Garapena

- Hard garestia dela eta sorrera geldoa.
- Pantaila grafikoen sorrerak bultzada haundia eman dio.
- Hauen merketzeak hedapen zabala bultza du.
- Estandarrek porrota.
  1. GKS, CORE
  2. PHIGS, PHIGS+
  3. Open GL?
- Hardware bidezko prozesu asko sortu da.

Beharrak

- Irudi formatoek memoria handia behar dute
  1. Videoa: 720 x 512 x 24
  2. Pantaila arruntetan 1024 x 1024 x 96 edo gehiago
- Animazioan gutxienez 20 irudi segunduko, beraz abiadurak garrantzia dauka

Hardwarea

Bete beharreko baldintzak:

- Memoria haundia behar dugu.
- Kalkuluetarako gaitasun haundia.
- Erakuste abiadura haundia.

Softwarea

- Estandarren porrota.
- Behe mailari ekin zaio. Xorg
- Liburutegiak:
  1. Marrazteko funtzioak.
  2. Sarrera irteerakoak.
  3. Egoeraren berri ematen duten funtzioak.
  4. Hardwarearekiko askea izan behar du.
  5. Erreferentzi sistema ezberdinak.

Ezaugarriak

- Ereduekin lan egin beharra (eredu zehatzak edota hurbilketak, sortzeko eta aldatzeko aukerak...)
- Tratamendua. Aldaketak eragin ereduari.
- Ikuste prozesuak.
  - Marra eta kurbak marraztea.
  - Ikuste leihoaren barnekaldekoa mugatzea.
  - Proiekzioak kalkulatzea.
  - Azalerak betetzea eta argiztatzea
  - Ikustezinak diren zatiak kentzea.
  - Animatzea.
- Erabiltzailearekin elkarrekintza.
  - Erabiltzen erraza.
  - Azkarra.
  - Hutsegiteak konpontzeko bidea emango duena.
  - Adimentsua?

Liburutegiak

- Abstrakzio maila haundiagoa.
- Iturburu programak eramankorrak.
- Estandarrak komeni dira.
- Sortzekotan parametrizatua:
  1. Marrazteko eta betetzeko oinarrizkoak.
  2. Sarrera eta irteerakoak.
  3. Egoeraren berri emateko funtzioak.

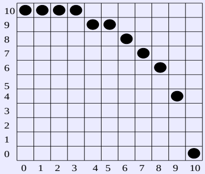
Irudia sortzeko prozesua

1. Adierazpide batean adierazi eta aldatu.
2. Kamara definitu eta bere erreferentzi sisteman adierazi dena.
3. Mugatu eta perspektiba lortu.
4. Atze-aurpegi eta ezkutuko aurpegiak kendu.
5. Argiak definitu eta objektuekin duten eragina kalkulatu.
6. Gainazalak argiztatu eta itzaleztatu.
7. Marraztu

Edukia

Adierazpen diskretua

- CRT edo pantaila. Pixelez osatua
    1. Piztuta-itzalita.
    2. Gris mailak.
    3. Errealitaterako kolorea.
- Matrizea da, arazoak:
    1. Informazio galera.
    2. Hutsuneak edo jarraitasun ezak eduki ditzake,
- Aliasing. Garunak antzekotasunak bilatzen ditu.
- Kalkulu kopuru haundia. Zirkuito integratuak.



x	0	1	2	3	4	5	6	7	8	9	10
y	10	9.9	9.7	9.5	9.1	8.6	8.0	7.1	6.0	4.3	0
biribilduz	10	10	10	10	9	9	8	7	6	4	0

Marra eta kurben adierazpena

Algoritmo inkrementalak

- Marra zuzenentzat. Ekuazio explizitoa. Malda 0 eta 1 artean.
- Aurreko pixelean oinarrituz

$$x_i = x_{i-1} + 1$$

$$y(x_i) = mx_i + k$$

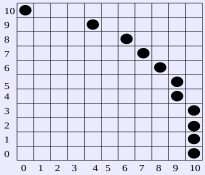
- biderkaketa, batuketa eta biribilketa egin behar da.
- asko erabiliko da eta eraginkorra behar da.

$$y(x_{i+1}) = mx_{i+1} + k =$$

$$m(x_i + 1) + k = mx_i + k + m =$$

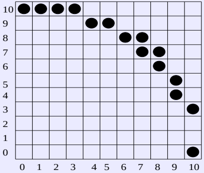
$$y(x_i) + m$$

- Malda 1 baino txikiagoa!



y	0	1	2	3	4	5	6	7	8	9	10
x	10	9.9	9.7	9.5	9.1	8.6	8.0	7.1	6.0	4.3	0
biribilduz	10	10	10	10	9	9	8	7	6	4	0

Zirkunferentzientzat ez digu balio. Nahiz eta pixel gehiago kalkulatu.



x	0.5	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5
y	9.9	9.8	9.6	9.3	8.9	8.3	7.6	6.6	5.2	4.3
biribilduz	10	10	10	9	9	8	8	7	5	4

DDA algoritmoak

- 1. mailako ekuazio diferentzialean oinarritua.
- Ebazpena zenbakizko metodo bidez.
- $y' = F(x, y)$ modukoa eta $y(x_0) = y_0$ ezaguna da.
- Adibide bezala Euler:

$$y' = \frac{\Delta y_i}{\Delta x_i}$$

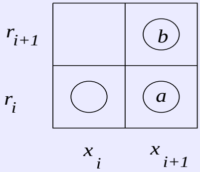
- Ondorioz:

$$y_{i+1} = y_i + F(x_i, y_i)\Delta x_i$$

$$x_{i+1} = x_i + \Delta x_i$$

Brasenham edo tarteko balioaren algoritmoa

Zuzenaren malda 0 eta 1 artekoa bada, pixel bakoitzaren hurrengoa bi pixelen artean aukeratu behar da:



eskuineko pixela bada (a kasurako)

$$r_{i+1} = r_i \Longleftrightarrow \mathrm{int}(y_{i+1} + 0{,}5) = r_i$$

$$r_i \leq y_{i+1} + 0{,}5 < r_i + 1$$

Baina $r_i = y_i - e_i$ eta $y_{i+1} = y_i + m$

$$y_i - e_i \leq y_i + m + 0{,}5 < y_i - e_i + 1$$

$$-e_i \leq m + 0{,}5 < -e_i + 1$$

$$-0{,}5 \leq m + e_i < 0{,}5$$

$$m + e_i < 0{,}5$$

ondorioz:

$$e_{i+1} = y_{i+1} - r_{i+1} = y_i + m - r_i =$$

$$= e_i + m$$

Goiko pixela bada (b kasurako)

$$r_{i+1} = r_i + 1 \Longleftrightarrow \mathrm{int}(y_{i+1} + 0{,}5) = r_i + 1$$

$$r_i + 1 \leq y_{i+1} + 0{,}5 < r_i + 2$$

Baina $r_i = y_i - e_i$ eta $y_{i+1} = y_i + m$

$$y_i - e_i + 1 \leq y_i + m + 0{,}5 < y_i - e_i + 2$$

$$0{,}5 \leq m + e_i < 1{,}5$$

$$0{,}5 \leq m + e_i$$

ondorioz:

$$e_{i+1} = y_{i+1} - r_{i+1} = y_i + m - r_i - 1$$

$$= e_i + m - 1$$

Brasenham. Koma higikorra

zuzena-marraztu(int x_0,int y_0,int x_1,int y_1)
{ int x_i,r_i; double e_i,m;
$m = (y_1 - y_0) \ / \ (x_1 - x_0)$;
$e_i = 0$; $r_i = y_0$;
marraztu(x_0, r_i;)
for($x_i = x_0 + 1$;$x_i < x_1$;x_i++)
{ $e_i += m$;
if ($e_i \geq 0{,}5$)
{ r_i++; e_i--; }
marraztu(x_i, r_i;)
}
}

Brasenham. Zenbaki osoak.

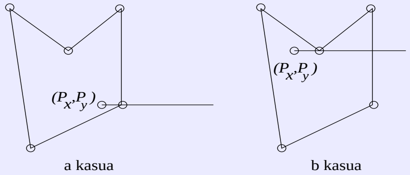
zuzena-marraztu(int x_0,int y_0,int x_1,int y_1)
{ int x_i,r_i,Δx,Δy,E_i;
$\Delta x = (x_1 - x_0)$; $\Delta y = (y_1 - y_0)$;
$E_i = -\Delta x$; $r_i = y_0$;
marraztu(x_0, r_i;)
for($x_i = x_0 + 1$;$x_i < x_1$;x_i++)
  { E_i+ = $2\Delta y$;
  if ($E_i \geq 0$);
    { r_i + +; E_i− = $2\Delta x$; }
  marraztu(x_i, r_i;) }
}

Poligonoak betetzea.

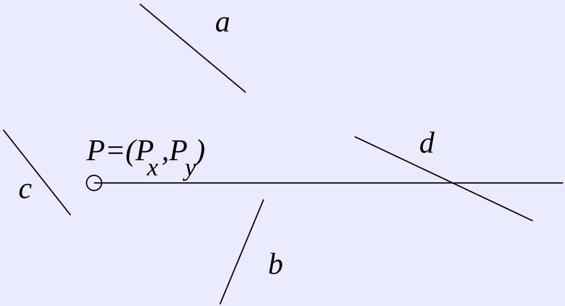
- Mugak marraztea erraza.
- Barnekaldea betetzea ez hain erraza.
- Barruko puntoa? Jordanen teorema.

  Puntutik infinitorako zuzenak kopuru bakoitia aldiz moztten badu orduan barnean dago.

- Praktikan arazoak:



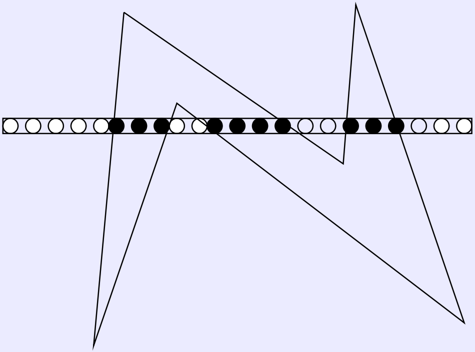
- Gehienetako kasuak:



- Arazoak konpontzeko: goikoa bada erpina kontuan hartu bestela ez.
- Lerroz lerro.

Lerroz lerrokako algoritmoa.

- Y_{max} eta Y_{min} lerroak kontuan hartu.
- Lerroaren koherentzia. Ilarak.



- Marrazterakoan ebaketak jaso eta binaka erabili.

Lerroz lerrokako algoritmoa (II).

- Algoritmoa:
  1. Lerroak poligonoarekin dituen ebaketak lortu.
  2. x ardatzarekiko ordenatu.
  3. bikoteka erabili sartu eta ateratzeko.
- Lerrotik lerrora informazioa bererabili?

Spanning Scan Line

void ebaketa_berria(int *x,int *$zbkitzailea$,int Δx int Δy)
{
*$zbkitzailea += \Delta x$;
while (*$zbkitzailea >= \Delta y$)
{/* $\Delta x > \Delta y$ gerta daiteke */
*x ++;
*$zbkitzailea - = \Delta y$;
}
}

Aliasing

- Objektu baten diskretizatzean oinarrituz, beste objekturen bat ikustea.
- Arrazoi nagusia espazio eremuaren laginketa bidezko irudien adierazpidea da.
- Pixelaren azaleraren zentroari dagokion intentsitatea pixel osoari jartzen diogu.
- Animazioan dir-dirka agertzen diren pixelak.
- ondoko kasuetan erraz gertatzen da:
    1. Argitasun aldaketa dagoenean.
    2. Kolore aldaketa askoko azaleretan.
    3. Objektu txikien eraginez animazioan sor daitezken dirdirak.

Aliasing, ekiditeko bideak

- Antialiasing metodo nagusiak:
  1. Gain-laginketa edo supersampling.
  2. Bi dimentsiotako anti-aliasing filtroen hurbilketa egin eta goi frekuentzien ezabaketa egitea.
  3. Laginketa estokastikoa.
- Aliasing efektuaren jatorria laginketa denez ahalik eta jarraiena izatea komeni da. Hala ere teknologia muga da.

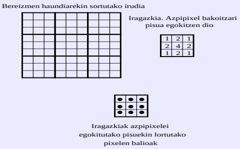
Gain-laginketa

Hiru urratsetan egiten den prozesua da:

1. Irudi jarraiari ikuste gailuaren bereizmena baino haundiagoko laginketa eragiten zaio, berezko pixel bakoitzeko *nxn* kalkulatuz.
2. Laginketa horri filtroren bat eragiten diogu, eta
3. horrela lortutako irudi diskretuarekin ikuste gailuari dagokion bereizmena duen irudi diskretua lortzen da.

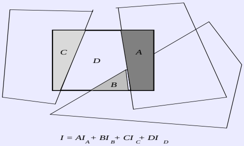
Gain-laginketaren ezaugarriak

- Zenbat eta azpipixel gehiago orduan eta kalkulu gehiago
- Kalkulu eta memoria gehikuntza n^2 ordenekoa da.
- Pixel errealetara egokitzeko pisudun filtroa erabil daiteke.
- Pixelaren zentroaren inguruko azpipixelei pisu handiagoa emanaz emaitza hobeak lortzen dira.
- Adibidez 512 x 512 pixeleko irudia kalkulatzeko eta pixel bakoitzarentzat 5x5 azpipixel erabili badira 512x512x25 biderketa eta beste horrenbeste batuketa egin behar dira.



Azalera laginketa

- Oinarrian pixelaren barne azaleraren geometria dago.
- Bertan ikus daitezkeen azpi azalerak kontuan hartzen ditu.
- Guztien eragina kalkulatuz pixelari intentsitatea egokitzen dio.



- Kalkuluak:
  1. Pixelari dagokion azaleran barnean dagoena kalkulatu.
  2. Hauetatik zein ikusten den eta zein ez erabaki.
  3. Benetan ikus daitezkenak kontuan hartuz intentsitatea kalkulatu.

Laginketa estokastikoa

.

- Begietako photoerrezetoreak ez daude uniformeki banatuta.
- Horretan oinarrituz lagineko puntuak ausaz aldatzen ditu:
    1. Irudiaren lagina lortu, baina lagineko puntu bakoitzaren kokapenari ausazko aldaketa egokituz.
    2. Aurreko urratsean lortutako laginarekin filtroren bat erabiliz aldaketarik gabeko lagineko pixel bakoitzaren intentsitatea kalkulatu.
- Aldaketak irudian zarata sartzen du, baina era berean aliasing efektua gutxitzen du.
- Ray-tracing edo izpi hedaketara ondo egokitu daiteke.
- Lerroz lerrokako eta z-bufferez dira hain ondo egokitzen, hala ere azalerak mikroazaleratan banatuz laginketa estokastikoa eragiten dutenak sortu izan dira.

Edukia

Geometria aldaketak

- Aurpegiz (erpinez eta ertzez) osatutako objektuak.
- Ikusiko ditugun aldaketak:
  1. Leku aldaketa.
  2. Tamainu aldaketa.
  3. Biraketa.
  4. Erreflexua.

Matrizeen erabilera

- Matrizeen erabilpena:

$$x^{'} = ax_1 + by_1$$

$$y^{'} = cx_1 + dy_1$$

- Linealtasuna:

$$A(v_1 + v_2) = Av_1 + Av_2$$

$$A(rv) = r(Av)$$

- Ondorioz marra zuzena aldatzean marra zuzena lortzen da, eta poligono bat aldatzean beste poligono bat lortuko dugu.
- Jatorria ez du aldatzen.

Leku aldaketa

- Objektuaren puntu bakoitzari leku aldaketari dagokion bektorea gehitu behar zaio.
- Mugimendua $\begin{pmatrix} m \\ n \end{pmatrix}$ izanik aldaketa:

$$v^{'} = v + \begin{pmatrix} m \\ n \end{pmatrix}$$

- Jatorria aldatu egiten du, beraz ezin dugu matrize bidez adierazi.
- Puntuen arteko distantzia mantentzen du, itxura aldaketarik ez du eragiten.

Neurri aldaketa

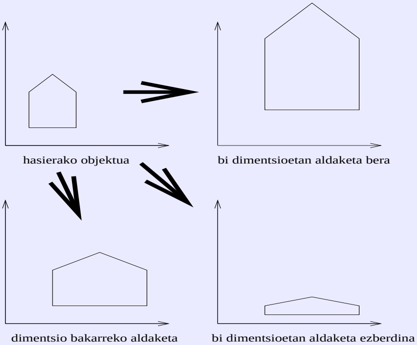
- Objektua p aldiz handiagoa egiteko nahikoa da erpinak $p$gatik biderkatzea. $x^{'} = px$ eta $y^{'} = py$
- Matrize bidez adieraz daiteke:

$$\begin{pmatrix} x^{'} \\ y^{'} \end{pmatrix} = \begin{pmatrix} p & 0 \\ 0 & p \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

- Horrela bi dimentsioetan aldaketa bera lortzen da, batzutan proportzioa galdu nahi izango da, orduan $\begin{pmatrix} p & 0 \\ 0 & q \end{pmatrix}$ matrizea erabili daiteke.
- $p, q > 0$ izan behar dute.
- $p, q > 1$ baldin badira objektua handitzen ari gara.
- $p, q < 1$ baldin badira txikitzen ari gara.

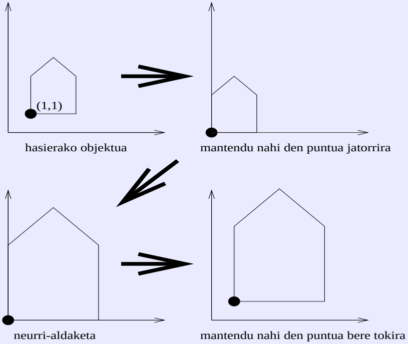
Neurri aldaketak eragindako leku aldaketa I

Neurri aldaketak objektuaren kokagune aldaketa sor lezake:



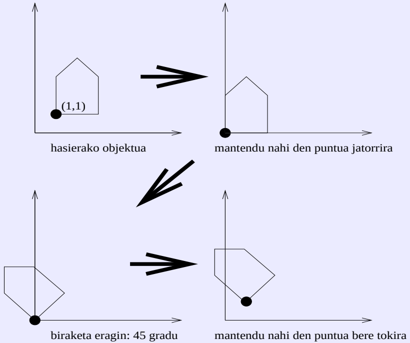
Mugitu gabe neurri aldaketa

Puntu jakin bat bere lekuan geratu arazteko hiru aldaketa kateatu behar dira:



Biraketa

- Puntu jakin baten inguruan: jatorria.
- Jatorriaren inguruan ez bada puntu hori jatorrira eraman, biraketa egin eta berriz bere lekura itzuli behar da puntua.



Biraketa: koordenatu polarren bidez

- koordenatu polarrekin errazago ulertzen da: $P = \begin{pmatrix} L \\ \alpha \end{pmatrix}$
- Ohiko koordenatuetara pasatzeko:

$$P = \begin{pmatrix} L\cos\alpha \\ L\sin\alpha \end{pmatrix}$$

- Puntua β gradu biratuta:

$$\begin{pmatrix} L\cos(\alpha+\beta) \\ L\sin(\alpha+\beta) \end{pmatrix}$$

- Koordenatu polarrak kartesiarretara itzuliz:

$$x^{'} = L(\cos\alpha\cos\beta - \sin\alpha\sin\beta)^{1}$$

$$y^{'} = L(\cos\alpha\sin\beta + \sin\alpha\cos\beta)^{2}$$

Biraketa: matrize bidezko adierazpena

- Puntuak hasieran zeuzkan koordenatuak kontuan hartuz:

$$x' = x\cos\beta - y\sin\beta$$

$$y' = x\sin\beta + y\cos\beta$$

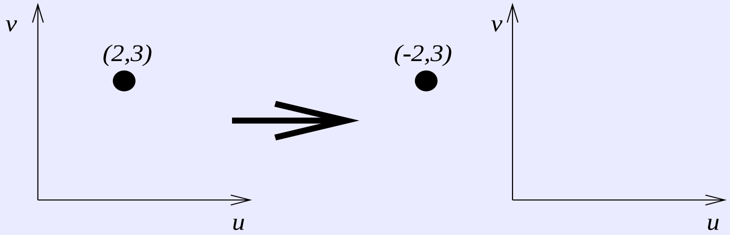
- Eta matrize bidez:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

- Matrize ortonormala, bere ezaugarria $A^{-1} = A^{i}$. Horretaz gain puntuen arteko distantziak errespetatzen ditu.

Erreflexua

- Helburua marra zuzen batekiko simetrikoa den irudia lortzea.
- Jatorritik pasatzen dela suposatuko dugu.



Erreflexua: matrize bidezko adierazpena

- Householderren matrize erreflexutzaileak erabil ditzakegu:

$$A = I - 2uu^{i}$$

- Bi propietate dituzte:
    1. $Au = -u$
    2. v bektorea $u$rekiko elkarzuta izanik $Av = v$
- u eta v gure erreferentzia sistemakoak badira A matrizeak v ardatzarekiko erreflexua lortzen du.

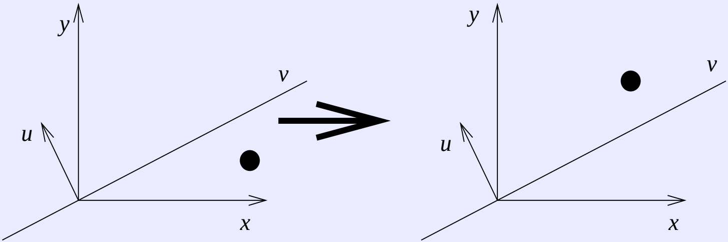
Erreflexua: edozein ardatz erabiliz

- Edozein ardatzekiko erreflexua: x bektorearekiko α gradu baditu $\begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix}$ izango da.
- Berekiko elkarzuta den u behar da: $\begin{pmatrix} -\sin\alpha \\ \cos\alpha \end{pmatrix}$.
- Zuzenean matrize erreflektatzailea lor daiteke.

$$A = I - 2uu^{i} = \begin{pmatrix} 1-2\sin^2\alpha & 2\sin\alpha\cos\alpha \\ 2\sin\alpha\cos\alpha & 1-2\cos^2\alpha \end{pmatrix}$$

Eta $\sin 2\alpha$ eta $\cos 2\alpha$ ordezkatuz:

$$A = \begin{pmatrix} \cos 2\alpha & \sin 2\alpha \\ \sin 2\alpha & -\cos 2\alpha \end{pmatrix}$$


y
v
u
x
y
v
u
x

aldaketen kateaketa

- Aldaketa ezberdinek, bata bestearen atzetik egitean, haserako puntuari aldaketa jakina eragingo diote. Ordenak bere garrantzia du.
  $\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} + \begin{pmatrix} m \\ n \end{pmatrix}$ modura adieraz daiteke aldaketa guztien eragina.
- Aldaketa guzti hauek marra zuzena beste marra zuzen batetara aldatzen dute. Gainera moztu egiten diren marrak aldaketa ondoren ere moztu egingo dira, eta paraleloak paralelo izaten jarraituko dute.

koordenatu homogeneoak

- Espazio afinean puntu eta bektore arteko ezberdintasuna ez da argia.
- Koord. homogeneoetan puntua: $\begin{pmatrix} hx \\ hy \\ h \end{pmatrix}$ eta bektorea: $\begin{pmatrix} x \\ y \\ 0 \end{pmatrix}$.
- Puntua eta bektorea bereizteaz gain leku aldaketa egiterakoan gertatzen zen tratamendu bereizketa gainditzen laguntzen digute. matrize bat biderkatuz leku aldaketa egin dezakegu:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & m \\ 0 & 1 & n \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Hiru dimentsiotan aldaketak

- 2D-ren hedapena
- Afinitatea: planoa planoan bihurtzen dute
- Koordenatu homegeneoak erabiliz

Leku aldaketa

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & m \\ 0 & 1 & 0 & n \\ 0 & 0 & 1 & o \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Neurri aldaketa

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} p & 0 & 0 & 0 \\ 0 & q & 0 & 0 \\ 0 & 0 & r & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Erreferentzia sistemaren jatorria mantendu egiten du, baina beste puntu guztiak aldatu egin ditzake. Neurriz aldatzean objektuak mugitu egin daitezke, eta hori gerta ez dadin mugitu nahi ez den puntua zehaztu beharko da.

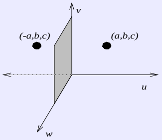
Erreflexua

Housenholderren matrize erreflektatzaileak, baina plano batekiko:

$$A = I - 2uu^{t}$$

u bektorea planoarekiko elkarzuta. $u = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}$ balitz

$$A = \begin{pmatrix} 1-2u_x^2 & -2u_xu_y & -2u_xu_z \\ -2u_xu_y & 1-2u_y^2 & -2u_yu_z \\ -2u_xu_z & -2u_yu_z & 1-2u_z^2 \end{pmatrix}$$



Biraketa I

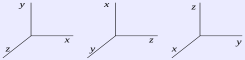
- Biraketa ardatza. Erreferentzi sistemakoa edo edozein. w ardatzarekiko:

$$u^{'} = u\cos\alpha - v\sin\alpha$$

$$v^{'} = u\sin\alpha + v\cos\alpha$$

$$w^{'} = w$$

w gure hiru ardatzetako edozein izan daiteke:



Biraketa II

- z ardatzarekiko biraketa

$$x' = x\cos\alpha - y\sin\alpha$$
$$y' = x\sin\alpha + y\cos\alpha$$
$$z' = z$$

Hori matrize bidez adierazi nahi izanez gero honela adierazi genezake:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{1}$$

Biraketa III

- x ardatzarekiko biraketa

$$y^{'} = y\cos\alpha - z\sin\alpha$$
$$z^{'} = y\sin\alpha + z\cos\alpha$$
$$x^{'} = x$$

$$\begin{pmatrix} x^{'} \\ y^{'} \\ z^{'} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{2}$$

Biraketa IV

- y ardatzarekiko biraketa

$$z^{'} = z\cos\alpha - x\sin\alpha$$
$$x^{'} = z\sin\alpha + x\cos\alpha$$
$$y^{'} = y$$

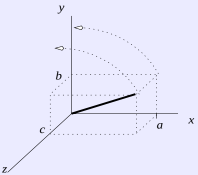
edo beste era batera:

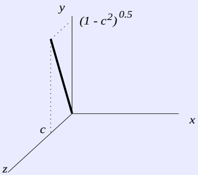
$$x^{'} = x\cos\alpha + z\sin\alpha$$
$$y^{'} = y$$
$$z^{'} = -x\sin\alpha + z\cos\alpha$$

Biraketa V

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

- Edozein ardatzekiko biraketa Orain arte ikusitako biraketetan oinarritu gaitezke, horretarako biraketa ardatza erreferentzi sistemaren ardatz bat izatea lortu behar da.
  1. Lehenengo biraketa:

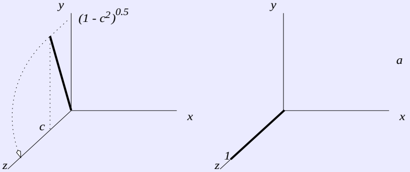




Biraketa VI

Horrela ardatza YZ planora eramango dugu (z-rekiko biraketa).

2. Erreferentzi sistemako ardatz bihurtzea.



x ardatzarekiko biraketa da.

Ondoren z ardatzarekiko biratu behar da, biraketa ardatza orain z ardatza baita.
Ardatzarekiko biraketa burutu ondoren, haserako erreferentzi sistemara itzultzeko, eragin ditugun bi biraketak desegin behar dira:

Biraketa VII

1. x ardatzarekiko alderantzizko biraketa.
2. z ardatzarekiko aldarantzizkoa.

Orokorrean P puntua edozein ardatzekiko biratzeko bost biraketa beharko ditugu:

$$P' = A^i B^i CBAP$$

Bost matrizeen biderkaketa kalkulatu eta puntu bakoitzari biderkatu beharko zaio.

Edukia

Adierazpideak

Adierazpide ugari dago, egokiena aukeratzeko begiratu beharrekoak:

- hardwarea, algoritmoak, datu egitura.
- irudia sortzeko prozesuaren kostua.
- irudiak behar duen zehaztasuna.
- objektuak editatzeko erreztasuna.

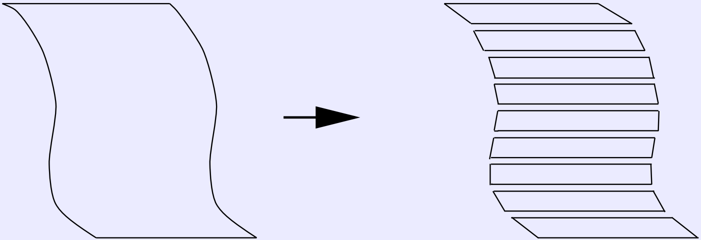
MOTAK

1. **Adierazpide poligonala.**
2. **Txatal bikubiko parametrikoen adierazpidea.**
3. **Solidoen geometria eraikitzailearen adierazpidea edo CSG.**
4. **Espazio partiketaren adierazpidea.**
5. funtzio matematikoak ere erabili daitezke. Adibidez:

$$x^2 + y^2 + z^2 = r^2$$

Adierazpide poligonala

- Poligono sarea. Borneen deskribapen geometriko eta topologikoa.
- Objektu kurbatuak adierazteko hurbilpenak egiten dira. Itxurak kalitatea gal ez dezan poligono asko erabili behar dira.

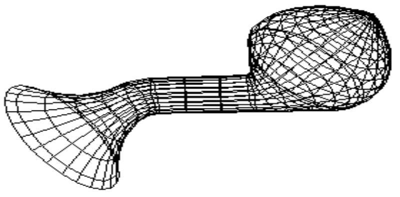


poligono bidezko adierazpenaren ezaugarriak

- Itzaleztatze algoritmo askok adierazpide hau bultza dute. Borobiltzeko interpolazioa erabilitzen dute.
- Beste adierazpideetako objektuak honetara itzultzen dira askotan, irudia sortzeko sinpleago eta emaitza onargarriak lortzen direlako.
- Poligono bakoitza banaka tratatu daiteke, Z-buffererako egokia.
- kasu errazena: erpinez osatutako poligonoak. Horrez gain informazio gehiago jaso daiteke: Bektore normala, auzokideak...
- Barne antolaketa egin daiteke tratamendua hobetzeko.
- Poligonoz poligono ihardun beharrean ertzetan oinarritu gaitezke. Horrela ertz bakoitza behin bakarrik landuko da.

objektuen eraiketa

- Eskuz edo digitalizatzaile bidez. Erpinak objektukoak izango dira baina poligonoak objektuaren hurbilketak dira.
- Automatikoki: Laser aztertzaile bidez.
- Matematikoki: Objektuaren sekzioa ibilbide batean zehar mugituz. Bereizmen arazoak sor daitezke. Toroidea.
- Sekzioak mugituz eta aldatuz: Sekzioak kurba bat egiten duen bezala itxura ere alda dezake.



Txatal bikubikoen sareak

- Txatalak gainazal kurbatu bateko puntu guztiak definitzen ditu.
- $Q(u, v)$ funtzioa polinomikoa da eta $0 \leq u, v \leq 1$
- Koefizienteak lortzeko 16 kontrol puntu erabiltzen dira. Hauek txatalaren itxura definitzen dute.
- Txatal auzokideek objektuaren osotasuna mantentzeko jarraitasuna behar dute.
- Poligonoek baino memoria baino gutxiago behar dute.
- Txatalak behar adinako bereizmena lortu arte zatitu daitezke, algoritmo egokiak erabiliz. Horrela irudia sortzeko bereizmenaren araberako kalkuluak egiten dira eta nahi adinako detailearekin adierazi ahal izango dira objektuak.

Txatal bikubikoen sareen desabantailak

- Datu egitura eraikitzea zailagoa da.
- Txatalen 16 kontrol puntuek osotasuna mantentzeko jarraitasun baldintzak bete behar dituzte.

Txatal bikubikoen sareen abantailak

CAD sistemetako abantailak:

- Definituta dauden objektuak editatzeko era intuitiboa. Bere zailtasunekin.
- Objektuaren itxura zehatza deskribatzen du eta objektuaren zenbait propietate kalkula daitezke: masa, bolumena, azalera, inertzi momentua, . . .
- Memoria gutxiago eskatzen du.

Txatal sareen eraiketa

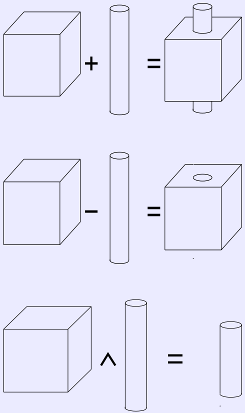
Bi teknika dira nagusi:

- **Gainazalen egokitzea:** interpolazio metodoa da. Objektuaren hurbilketa izango da emaitza, baina poligono bidez lortzen ez den jarraitasuna lortzen du.
- **Sekzioen mugimendua:** Kurba kubikoa espazioan mugitu araziz lortutako gainazala da. Mugimendua bera ere kurba kubiko batek adieraziko du. Pausu bakoitzean sortutako gainazalarentzat txatal bikubikoa kalkulatu behar da.

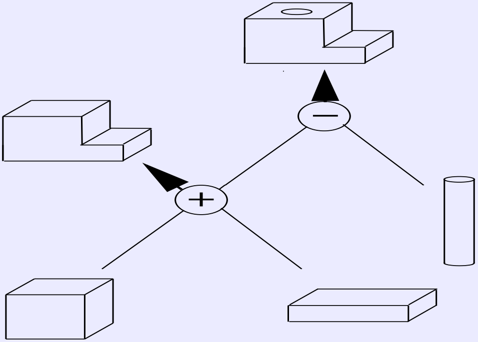
CSG

- Objektua zuhaitz bidez adierazten da.
- Hostoek oinarrizko objektuak adierzten dituzte.
- barneko erpinek objektu arteko eragiketak adierazten dituzte.
- oinarrizko objektuak esferak, konoak, kuboak . . . izan daitezke.
- eragiketak aldaketa linealak edo eragiketa bulearrak izan daitezke.

CSG: eragiketak



CSG adibidea



CSGren desabantailak

- Irudia sortzeko denbora asko behar da
- Eragiketak globalak dira, ezin dira zati batengan eragin.

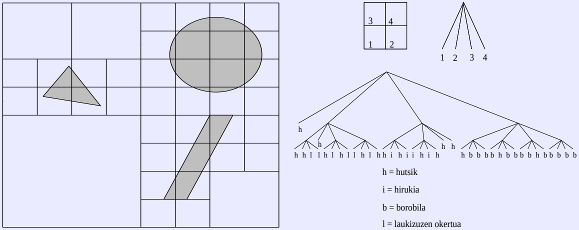
Hala ere objektuak sortzeko erabilerrazak dira.

Espazio partiketa

- Espazioa voxeletan banatzen da.
- Voxel bakoitzean zein objektu dagoen adieraziz.
- Objektu bakoitzak voxel batzuk beteko ditu.
- Memoria asko behar da.
- Egitura ezberdinak daude, eta laguntza bezala erabiltzen dira.
- Izpi hedaketarako egokia da.

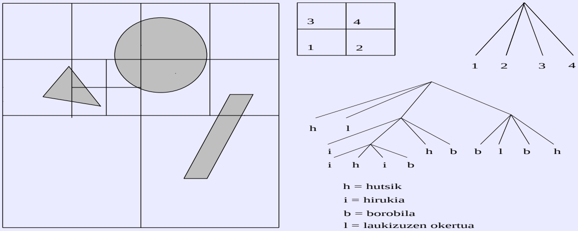
Octreeak

- Zuhaitz egitura da.
- Espazio zati bakoitza 8 zatitan egiten da, erpin bakoitzak 8 seme.
- Zati bakoitza oktante bati dagokio.
- Bi dimentsiotan quadtree izango litzateke.



Octreeak: zatiketaren ezaugarriak

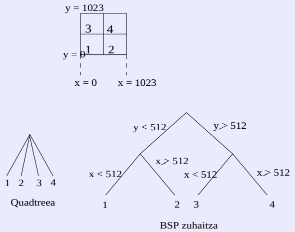
- Espazioaren zatiketa bereizmenaren arabera finkatu behar da.
- Hutsik dagoen espazio zatia ez dago zatitu beharrik.



- Borneak bakarrik har ditzake kontuan.
- Orokorrean, hosto batek poligono lista izango du.

BSP zuhaitzak

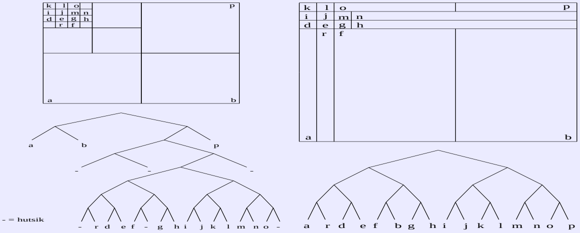
- Hau ere zuhaitz egitura da.
- Espazioa beti bitan banatzen da.
- Zatitzeko planoak erabiltzen dira.



- Izpi hedaketarako egokia. Izpiaren ibilbidea zuhaitzean mugitzearen parekoa da eta bertako objektuekin ebaketa testak egitea.

moldaerazko zatiketa

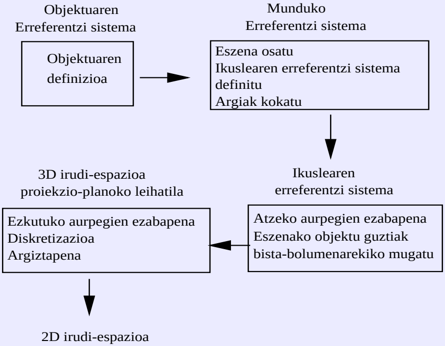
- Zatiketarako planoa edozein izan daiteke.



- Zuhaitz orekatuagoak lortzen dira, sakonera txikituz.

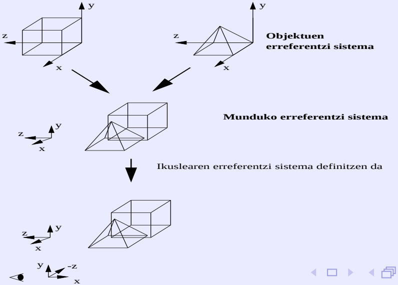
B-rep bidez irudia sortuz

- Sarrera poligono lista da, irteera koloredun pixel multzoa.
- Koordenatu espazio desberdinak igaroko ditugu.



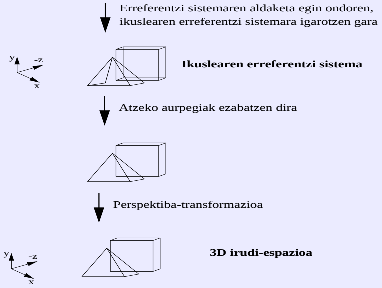
poligono bidez irudia I

- Objektuek bere koordenatu sistema daukate.
- Guztiak mundu bakar batean sartzen dira, bertan koordenatu sistema jakina izango dugu eta objektu bakoitza hor adierazi beharko da.
- Argien eta ikuslearen kokapena sistema horretan adieraziko dira, ikuste direkzioa ere bai.



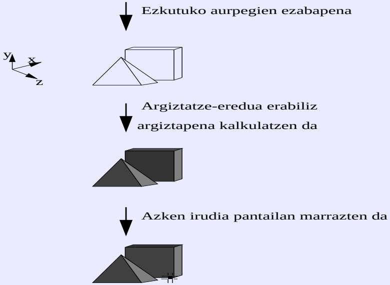
poligono bidez irudia II

- Munduan ikulsearen erreferentzi sistema zehaztu behar da.
- Objektu guztiak sistema horretara pasa, atze-aurpegiak ezabatu eta mugaketa burutu.



poligono bidez irudia III

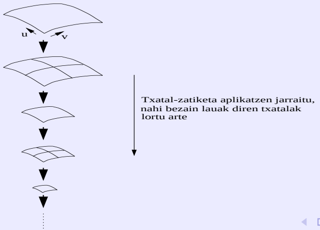
- Ezkutuko aurpegiak kendu eta itzaleztatu.
- Azkenik Pixeletara pasa behar dugu irudia.



- Azken urratsak era askotan egin daitezke, algoritmo asko dago.

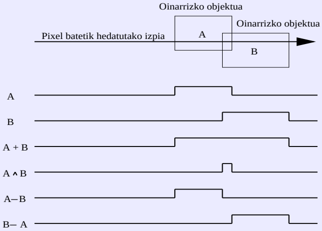
Txatal bidez irudia sortuz

- Errazena: txatal sarea poligono adierazpidera pasatzea.
- Txatalak edozein bereizmen eduki dezake eta guk finkatutako bereizmena lortu arte zatitu daiztezke.
- Irudia lortzeko, zatitutako txatalak poligono moduan har daitezke.
- Poligono kopurua txatal kopurua baino handiagoa izango da.
- Zatiketa bukatzeko lautasun baldintzak jarri behar dira.



CSG bidez irudia sortuz

- objektuak sortzea erraza izan arren irudia lortzea konplexua da.
- Arazoa borneak lortzean dago. Hiru teknika:
  1. CSG izpi hedaketa.



  2. CSG adierazpidetik, voxel adierazpidera itzuli eta bolumen renderizazioa burutzea.
  3. Z-buffer algoritmoa erabiltzea.

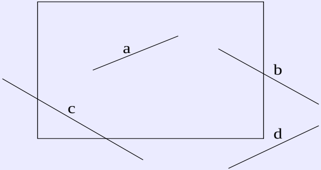
Edukia

Ikuste Sistemak

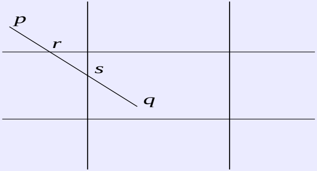
- 3D-tik 2D-ra pasa behar dugu.
- 2D-tik pantailara. Agertu nahi den zatia zehaztu eta horren irudia azaldu behar da.
  1. Mugaketa: agertu nahi duguna mugatu.
  2. Mugatutakoa pantailaren erreferentzi sistemara pasa.

Marren mugaketa

- Marren mugaketa: Cohen-Sutherland algoritmoa

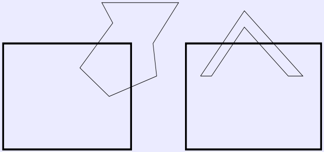


Espazioa 9 zatitan banatzen du eta kode batzuren bidez lana errazten da.



Poligonoen mugaketa

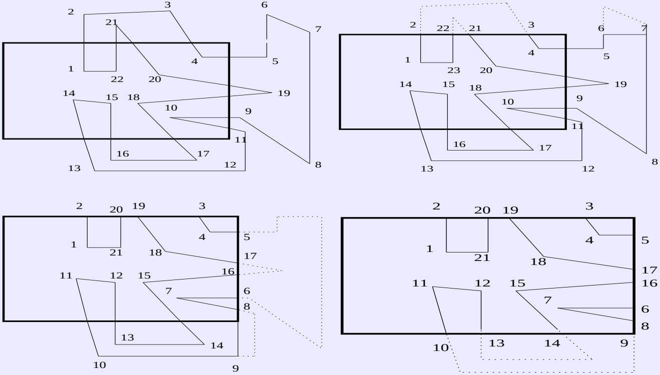
- Poligono mugaketa: Sutherland-Hodgman algoritmoa. Poligonoa bete egin behar bada, bere mugetako ertzak mugatzea ez da nahikoa, poligono izaera galdu egin baitaiteke. Barnealdea eta kanpoaldea bereizten jakin behar da beti:



(a) (b)

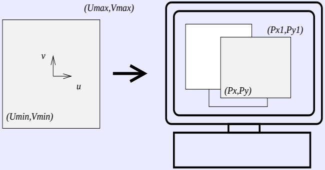
Poligonoen mugaketa

Algoritmoak poligono bat hartu eta mugatu ondoren poligono bat itzultzen du. Urrats bakoitzean muga bat hartuz.



laukiratzea

- Erreferentzi sistema aldaketa besterik ez da.



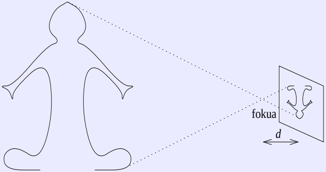
- Azken finean $\overline{u} = Au + T$ motatakoa: leku eta neurri aldaketa.

$$A = \begin{pmatrix} \frac{P_{x1} - P_x}{U_{max} - U_{min}} & 0 \\ 0 & \frac{P_{y1} - P_y}{V_{max} - V_{min}} \end{pmatrix}$$

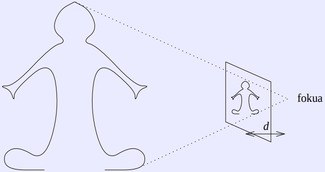
$$T = \begin{pmatrix} \frac{P_x . U_{max} - P_{x1} . U_{min}}{U_{max} - U_{min}} \\ \frac{P_y . V_{max} - P_{y1} . V_{min}}{V_{max} - V_{min}} \end{pmatrix}$$

3D-tik 2D-ra

- Gure begiek edo argazki kamerak:



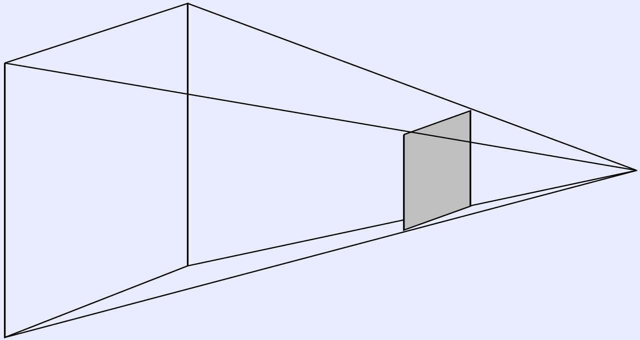
- Alderantziz egon ez dadin proiekzio planoa aurrean jar daiteke.



- Puntu bakoitzetik fokura izpia bota eta proiekzio planoa ebakitzen duen puntua izango da 2D-n dagokion puntua.

proiekzioa

- proiekzio planoaren mugak kontuan hartuz:



Ikuslearen erreferentzi sistema

- Piramidea zehatz mehatz adierazi behar da.
- Aldaketak egin ahal izateko parametrizatua behar du.
- Estandarretara jotzea komeni da: **phigs** edota **OpenGL**.
- Eman beharreko urratsak:
  - Erreferentzi sistema aldaketa: kamararen erreferentzi sistemara pasa behar da.
  - Proiektatzea.
- Hori egin ahal izateko behar den informazioa:
  1. Kamerak nola ikusten duen adierazi behar du, hau da, kameraren erreferentzi sistema.
  2. Kamerak ikus lezakeen guztiaren artean zer ikusten duen, hau da, zerren irudia lortu behar den ere agertu behar da.
  3. Azkenik, pantailaren zein zatitan azalduko den irudia.

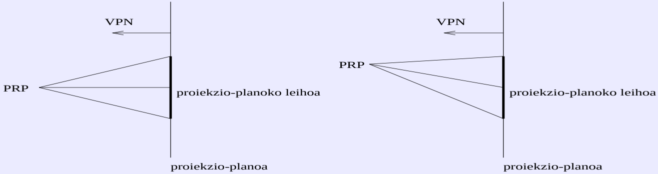
PHIGS kamera I

- Parametroak:
  1. Hiru dimentsiotako puntua, VRP, puntu hau erreferentzi sistema berriaren jatorria izango da.
  2. Begiratze direkzioa adieraziko duen bektorea, VPN.
  3. Goia zein direkziotan dagoen adieraziko duen bektorea, VUP.
  4. Proiekzio mota, perspektiba ala paraleloa.
  5. Proiekzioaren fokua edo zein puntutarantz proiektatuko den adieraziko duen puntua, PRP. Honen koordenatuak erreferentzi sistema berrian adierazten dira.
  6. Plano urrunerako distantzia, b, distantzia honetaz harantzagoko objektuak ikusezinak bihurtzen dira.
  7. Plano hurbilerako distantzia, f. Honen eta aurrekoaren arteko objektuak besterik ez dira ikusiko.

PHIGS kamera II

8. Proiekzio planorako distantzia, d. Hiru plano hauek erreferentzi sistema berriaren x_k, y_k bektoreek osatzen duten planoaren paraleloak dira, eta b, f eta d zenbakiek plano hauen kokapena adierazten dute, kokapen hori erreferentzi sistema berriaren jatorriarekiko distantzia adieraziz agertzen da, jakina z_k bektorea non mozten duten agertzen dute.
9. proiekzio planoan leihatilaren neurria, hau, x_{min}, y_{min}, x_{max} eta y_{max} parametroekin adierazten da. PRPtik leiho honen lau ertzetara doazen marrek mugatutako piramidea izango da ikuste-bolumena. Kontutan eduki beharrekoa da leiho honen erditik PRPra doan bektoreak ikuste direkzioaren paraleloa izan beharrik ez duela, beraz, proiekzio zeiharrak lor daitezke:

PHIGS kamera III



- Kamerak begiratzen duen direkzioa edozein izan daiteke.
- Proiekzio planoak edozein bektore dauka goranzko bektore bezala.
- Ikuste bolumena ematen da, horretarako proiekzio planoko leihatila, urruneko planoa eta hurbileko planoa ematen dizkigute.

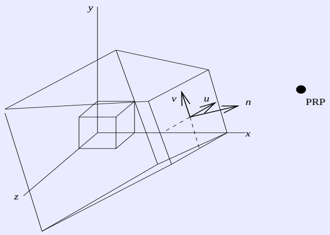
OpenGL kamera

- Kameraren kokapena. Hau jatorri berria izango da.
- Begiratze puntua. Bi datu hauekin begiratze direkzioa lor daiteke.
- Vup. Goranzko direkzioa.
- proiekzio planorako distantzia eta urruneko planoa. Proiekzio planoa gertuko planoaren berdina da kasu honetan.
- Proiekzio planoko leihoa (xmin,ymin) (xmax,ymax).
- Proiektatzeko era: paraleloa ala perspektiba.

Kameraren erreferentzi-sistema munduan

Munduko erreferentzi-sisteman kamera definitu behar da:

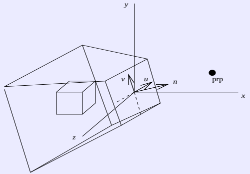
- jatorria (OpenGL: kameraren kokapena. PHIGS: VRP)
- oinarria (u, v, w) edo oinarriko bektoreak lortzeko informazioa: OpenGL: kokapena, begiratze puntua eta Vup. PHIGS: VPN eta Vup
- Fokua (kameraren kokapena edo PRP)



Kameraren erreferentzi-sistema

Munduaren erreferentzi sistematik kamararenera pasa behar da: kameraren parametroak erabiliz erreferentzi-sistema berriaren oinarria eta jatorria zehaztu behar dira:

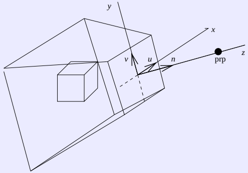
1. jatorria VRP edota kameraren kokapena da.



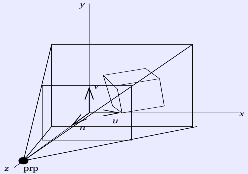
2. z_k: VPN bektoreak (PHIGS) edo kokapena - begiratze puntuak (OpenGL) adierazten du.
3. x_k: Vup eta z_k ardatzek zehaztutako planoarekiko elkarzuta denez, bi horien arteko bektore biderkaketa bidez lor daiteke.
4. y_k: beste bien bektore-biderkaketa bidez.

Erreferentzi-sistema aldaketak

- Munduko sistematik kamerarenera aldatu:



- kamerak ikusten duena:



- Proiekzioa lortu (proiekzio zeiharra kontuan hartuz).
- Proiekzio leihotik pantailara pasa: laukiraketa.

Atze-aurpegien ezabapena

- Poligono bakoitza plano batean dago:

$$Ax + By + Cz + D < 0$$

  A,B eta C parametroek, planoarekiko bektore normala definitzen dute.
- Bektore honen eta ikuslearengana doan bektorearen arteko angelua -90 eta 90 artekoa bada ikusleak ikusi egin dezake baina bestela atzerantz begira dago.
- Baldintza biderkaketa eskalarraren bidez lor daiteke:

$$VN = |V||N| \cos(\alpha) > 0$$

Edukia

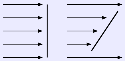
Argiztatze-ereduak

- Errealitate itxura lortzeko argiaren eragina kontuan hartu behar da.
- Hori lortzeko beharrezkoak dira:
    1. Objektuen adierazpen zehatza.
    2. Argiaren efektu fisikoen deskribapen ona.
- Lortu beharreko efektuak:
    1. Argi isladak
    2. Gardentasuna.
    3. Ehundura.
    4. Itzalak

Oinarrizko ereduak

- Kalkuluak gainazalen propietateen eta argi iturrien arabera egiten dira.
- Argi orokorra: Ingurune-argia.
  I_a toki guztietara berdin iristen da.
- Direkzioa daukaten argiak:
  1. Direkzioa besterik ez daukatenak: eguzkia
  2. Kokapena ere badaukatenak: bonbila.
  3. Kokapena daukatenak baina direkzio jakin batean besterik argitzen ez dutenak: fokoak

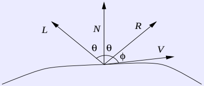
Argi isla motak

- Isla barreiatua.
  Objektuak jasotako argi kopuruaren arabera:
  1. Ingurune-argia beti bera da:
  $$I_{ingbarr} = K_d I_a$$
  2. Beste argiei dagokiena aldatu egiten da:
  
  Erasotze angeluaren araberakoa da, argi bakoitzeko:
  $$I_{l,barr} = K_d I_l \cos(\alpha) = K_d I_l\ (\overline{N}\ \overline{L})$$

  Guztira:
  $$I_{barr} = K_a I_a + K_d I_l\ (\overline{N}\ \overline{L})$$

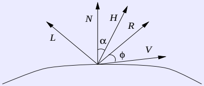
Ispilu isla. Phong eredua

Argia jasotzen duen objektuko zenbait puntutan, ikuspuntuaren arabera, argia ispiluan bezala isla daiteke, inguru distiratsuak sortuz.



$$I_{isp} = W(\theta) I_l \cos^{ns}(\phi) \text{ edota } I_{isp} = K_s I_l \, (\overline{V} \; \overline{R})^{ns}$$

Erdibideko bektorea erabiliz:



$$I_{isp} = K_s I_l \, (\overline{N} \; \overline{H})^{ns}$$

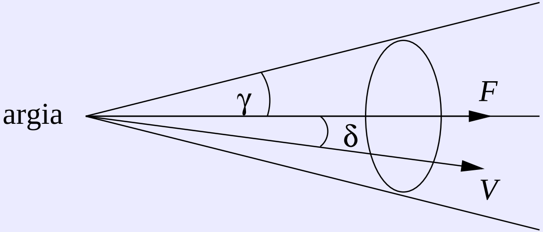
Argi efektuen kalkulu orokorra

Orokorrean argi-iturri askorekin:

$$I = K_a I_a + \sum_{i=1}^{n} I_{li}[K_d(\overline{N}\ \overline{L_i}) + K_s(\overline{N}\ \overline{H_i})^{ns}]$$

Warnen eredua: Fokoak

- Norabide jakinean argiztatzen duten argientzat.



$$I_{konoa} = I_l(-\overline{V}\ \overline{F})^p$$

- Eremu mugatuak argiztatzeko.

Intentsitate ahuldura

- Argia zenbat eta urrutiago orduan eta indar gutxiago.

$$f(d) = \frac{1}{(a_0 + a_1 d + a_2 d^2)}$$

gertu daudenak argi gehiegi jaso ez dezaten:

$$f(d) = \text{mín}(1, \frac{1}{(a_0 + a_1 d + a_2 d^2)})$$

- Argiztatze ereduari gehituz gero:

$$I = K_a I_a +$$

$$+ \sum_{i=1}^{n} f(d_i) I_{li} [K_d (NL_i) + K_s (NH_i)^{ns}]$$

Koloreari dagozkion zuzenketak

- Argi bakoitzak 3 osagai dauzka.
- Urdinarentzat ondorengoa izango da:

$$I_B = K_{aB} I_{aB} +$$

$$\sum_{i=1}^{n} f(d_i) I_{lBi} [K_{dB}(NL_i) + K_{sB}(NH_i)^{ns}]$$

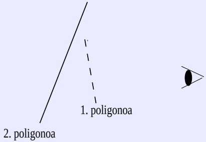
- beste aukera konstanteak ere bektore bihurtzea:

$$I_B = K_a S_{dB} I_{aB} +$$

$$\sum_{i=1}^{n} f(d_i) I_{lBi} [K_d S_{dB}(NL_i) + K_s S_{sB}(NH_i)^{ns}]$$

Gardentasuna

- Errefrakziorik gabeko gardentasuna.
  1. Gardentasun interpolatua.



$$I = (1 - K_t)I_1 + K_t I_2$$

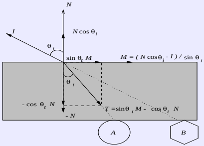
  2. Gardentasun iragazia.

$$I = I_1 + K_t O_{t\lambda} I_2$$

Errefrakziodun gardentasuna

- Errefrakziodun gardentasuna. Izpi erasotzailearen eta errefraktatuaren arteko angeluaren arteko erlazioa:

$$\frac{\sin(\theta_i)}{\sin(\theta_t)} = \frac{\eta_t}{\eta_i}$$

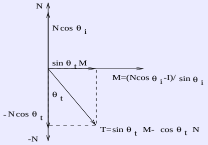


Errefrakzio bektorearen kalkulua

Errefrakzio bektorea:

$$\overline{T} = \left(\ \sin(\theta_t)\overline{M}, -\cos(\theta_t)\overline{N} \ \right)$$

$$\overline{T} = \left(\ \eta_r(\overline{N}\ \overline{I}) - \sqrt{1 - \eta_t^2(1 - (\overline{N}\ \overline{I})^2)} \ \right) \overline{N} - \eta_t \overline{I}$$

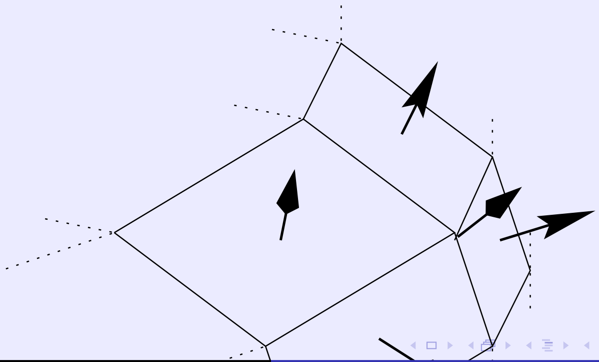


Itzalak

- Ezkutuko gainazalak ezabatzeko erabiltzen den edozein algoritmo erabil daiteke. Argiak zer ikusten duen algoritmo horrek adieraziko digu.
- Kalkuluan behin egin behar dira, argiaren kokapena aldatzen ez bada behintzat.
- Argi-iturriak ikusten duen puntuetan argi horri dagokion intentsitatea kalkulatu behar da, bestela ez.

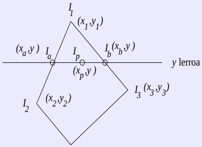
Poligonoak argiztatzeko teknikak

- Poligono barneko puntu bakoitzean argiaren kalkuluak egin behar dira.
- Poligono bakoitza leundu nahi da, ingurukoekin bat eginaz.
- Bektore normala erpinetan da ezaguna.



Poligonoak argiztatzeko teknikak: Gouraud

- Barneko puntuetarako bi teknika:
  1. Gouraud.
     Intentsitatea erpinetan kalkulatzen du eta barneko pixeletan interpolatu.



$$I_a = \frac{1}{y_1 - y_2}(I_1(y - y_2) + I_2(y_1 - y))$$

$$I_b = \frac{1}{y_1 - y_3}(I_1(y - y_3) + I_3(y_1 - y))$$

$$I_p = \frac{1}{x_b - x_a}(I_a(x_b - x) + I_b(x - x_a))$$

Phong eta Gouraud arteko diferentzia

- Gouraud azkarragoa da, kalkulu gutxiago dauzka.
- Phongen teknikan ispilu islada ongi lor daiteke, Gouraudenean ez.
- Hala ere test baten ondoren ispilu islada egon daitekeen jakin daiteke, eta horren arabera bata edo bestea erabiltzea erabaki daiteke poligono bakoitzerako.