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.

- Irudi formatoak

  1. Videoa: 720 x 512 x 24

  2. lor daitezkenak 1024 x 1024 x 96 edo gehiago

- Animazioan gutxienez 20 irudi segunduko.

HARDWAREA

Bete beharreko baldintzak:

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

SOFTWAREA

- Estandarren porrota.
- Behe mailari ekin zaio. X11R5
- 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.
  - Zehatzak.
  - Hurbilketak.
  - Gailu bidez lortutakoak.
  - sortutakoak. Sortzeko aukerak.
- Tratamendua. Aldaketak eragin ereduari.
- Ikuste prozesuak.
  - Marra eta kurbak marraztea.
  - Ikuste leihoaren barnekaldekoa mugatzea.
  - Proiekzioak kalkulatzea.

- Azalerak betetzea.
- 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

1. Adierazpide batean adierazi eta aldatu.

2. Kamara definitu eta bere erreferentzi sistema 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

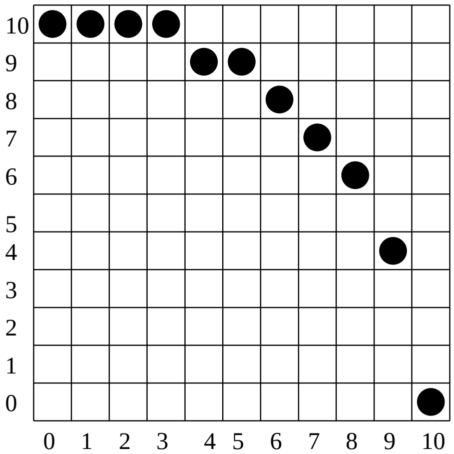
DISKRETIZATZEA

- 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,



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

3. Aliasing. Garunak antzekotasunak bilatzen ditu.

4. Kalkulu kopuru haundia. Zirkuito integratuak.

MARRAK ETA KURBAK

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 borobilketa egin behar da.

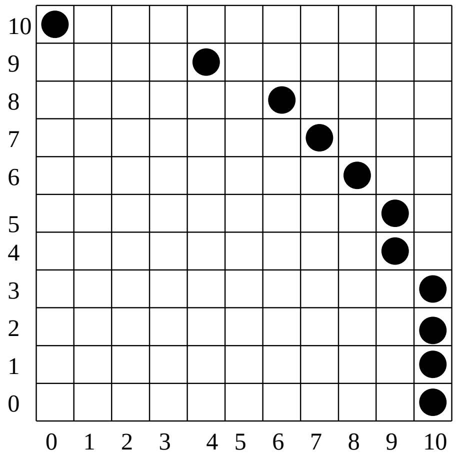
- asko erabiliko da eta eraginkorra behar da.

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

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

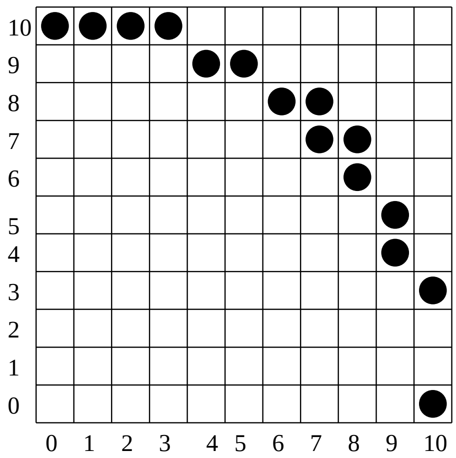
$$y(x_i) + m$$

- 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
borobilduz	10	10	10	10	9	9	8	7	6	4	0

- Zirkunferentzientzat ez digu balio. Nahiz eta pixel gehiago kalkulatu.



Taula honetako balioak aurreko irudikoarekin osotu behar dira.

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
borobilduz	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:

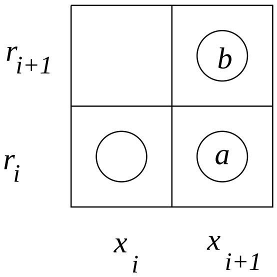
$$y^{'} = \frac{\Delta y_i}{\Delta x_i}$$

- 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



- a kasurako

$$r_{i+1} = r_i \Longleftrightarrow \text{int}(y_{i+1} + 0.5) = r_i$$

$$r_i \leq y_{i+1} + 0.5 < r_i + 1$$

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:

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

$$= e_i + m$$

- 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 ($E_i \geq 0$);

$\{ r_i ++$; $E_i -= \Delta x$; $\}$

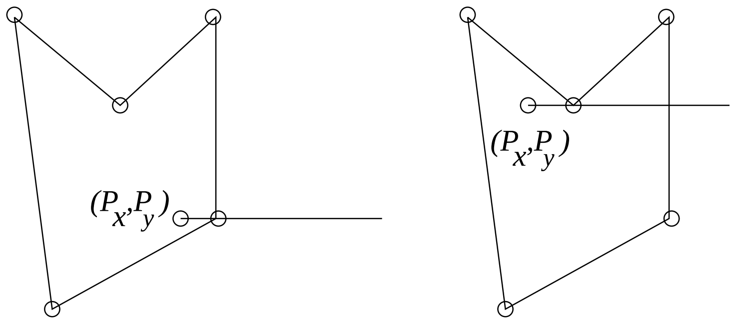
marraztu(x_i, r_i;) $\}$

$\}$

Poligonoak betetzea.

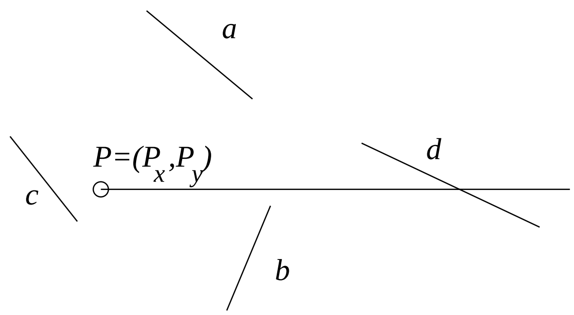
- Mugak marraztea erraza.
- Barnekaldea betetzea ez hain erraza.
- Barruko puntoa? Jordanen teorema.

  Puntutik infinitorako zuzenak kopuru bakoitia aldiz mozen badu orduan barnean dago.
- Praktikan arazoak:



a kasua   b kasua

- Praktikan gehienetako kasuak:



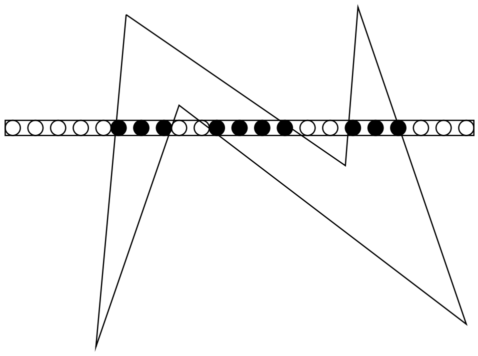
- Arazoak konpontzeko: goikoa bada erpina kontuan hartu bestela ez.

- Lerroz lerro.

Lerroz lerrokako algoritmoa.

- Y_{max} eta Y_{min} lerroak kontuan hartu.

- Lerroaren koherentzia. Ilarak.



- Marrazterakoan ebaketak jaso eta binaka erabili.

- 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)

{

*$zbkitzailea$ += Δx;

while (*$zbkitzailea >= \Delta y$)

{/* $\Delta x > \Delta y$ gerta daiteke */

*x ++;

*$zbkitzailea -= \Delta y$;

}

}

Aliasing, ekiditeko bideak

- 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.

- 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, urrats hauek ondorengoak dira:

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.

- Zenbat eta azpipixel gehiago orduan eta kalkulu gehiago

- Kalkulu gehikuntza n^2 ordenekoa da.

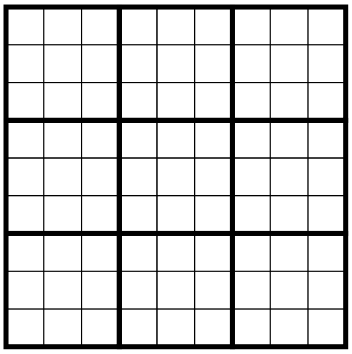
- Pixel errealetara egokitzeko pisudun filtroa erabili daiteke.

- 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.

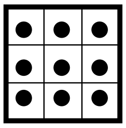
- Memoria asko eska dezake.

Bereizmen haundiarekin sortutako irudia



Iragazkia. Azpipixel bakoitzari
pisua egokitzen dio

1	2	1
2	4	2
1	2	1



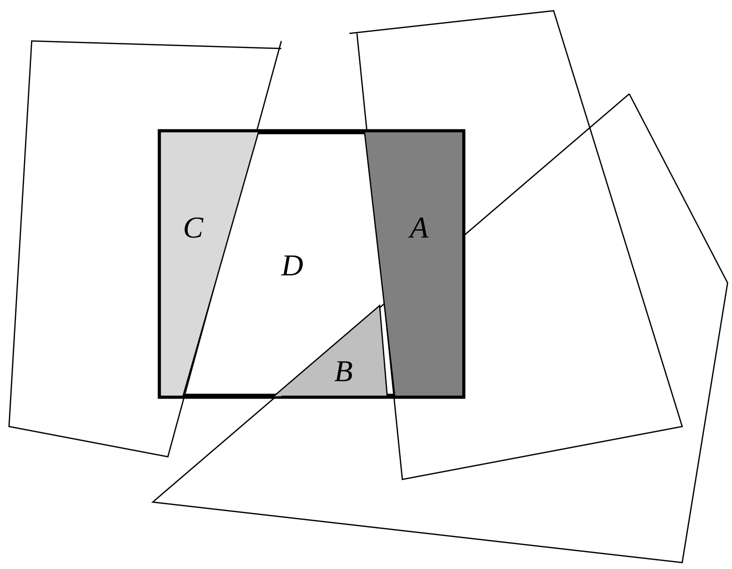
Iragazkiak azpipixelei
egokitutako pisuekin lortutako
pixelen balioak

Azalera laginketa

- Oinarrian pixelaren barne azaleraren geometria dago.

- Bertan ikus daitezken azpi azalerak kontuan hartzen ditu.

- Guztien eragina kalkulatuz pixelari intentsitatea egokitzen dio.



$$I = AI_A + BI_B + CI_C + DI_D$$

- 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.

- pausoak:

  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.

Geometri aldaketak

- Aurpegiz (erpinez eta ertzez) osatutako objektuak.

- Ikusiko ditugun aldaketak:

  1. Leku aldaketa.
  2. Tamainu aldaketa.
  3. Biraketa.
  4. Erreflexua.

- 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.

Tamainu aldaketa

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

- Matrize bidez adierazi daiteke:
$$\begin{pmatrix} x^{'} \\ y^{'} \end{pmatrix} = \begin{pmatrix} p & 0 \\ 0 & p \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

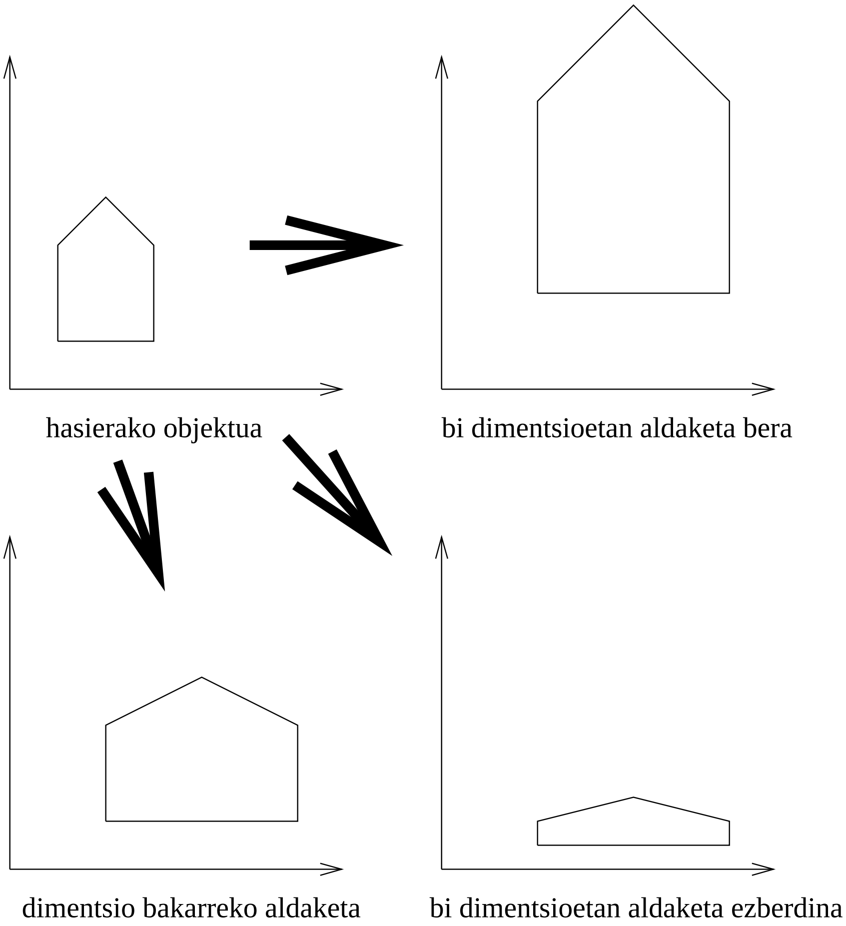
- Horrela bi dimentsioetan aldaketa bera lortzen da, batzutan proportzioa galdu nahi izango da, orduan $\begin{pmatrix} p & 0 \\ 0 & q \end{pmatrix}$ matrizea erabili daiteke.

- $p, q > 0$ izan behar dute.

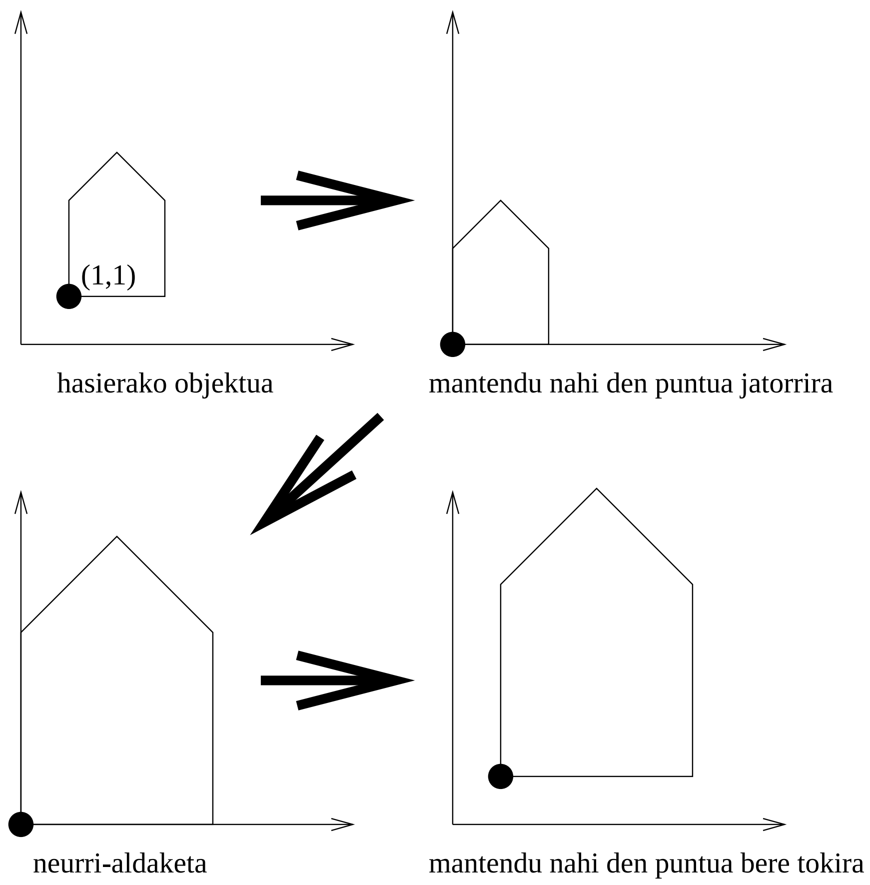
- $p, q > 1$ baldin badira objektua handitzen ari gara.

- $p, q > 1$ baldin badira txikitzen ari gara.

- Tamainu aldaketak objektuaren kokagune aldaketa sor lezake:



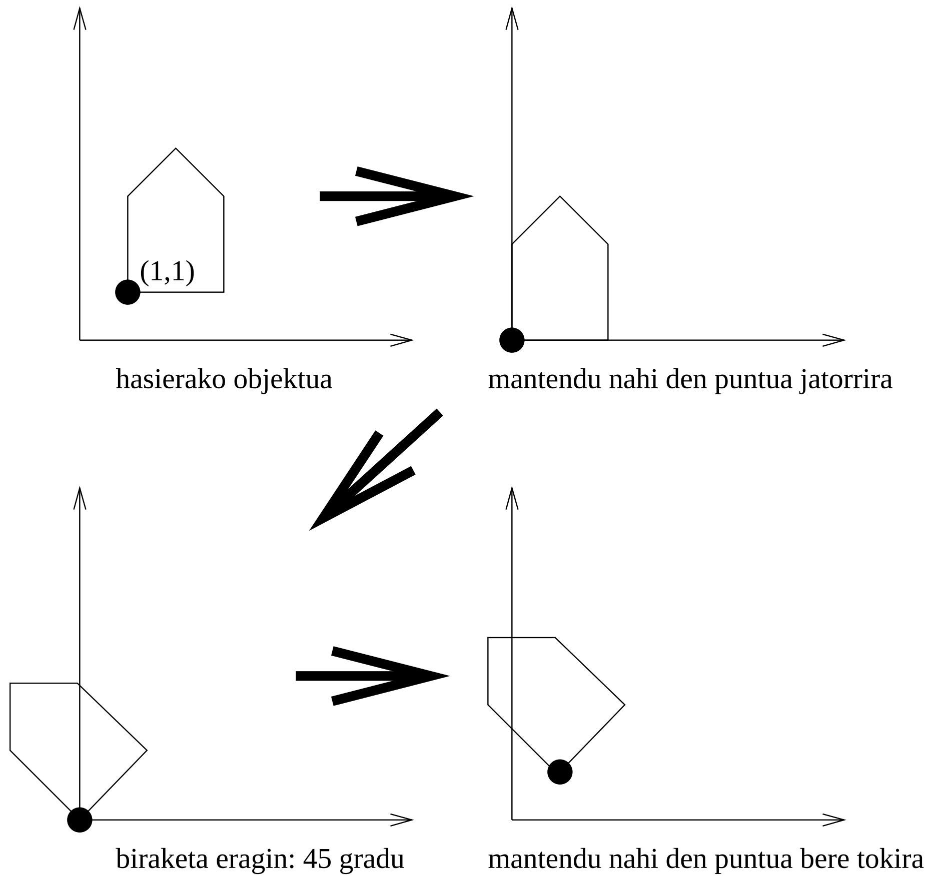
- Puntu jakin bat bere lekuan geratu arazteko hiru aldaketa kateatu behar dira:


(1,1)
hasierako objektua
mantendu nahi den puntua jatorrira
neurri-aldaketa
mantendu nahi den puntua bere tokira

Biraketa

- Puntu jakin baten inguruan: jatorria.

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



- 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:

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

- Ondorioz puntuari eragindako aldaketa:

$$x^{`} = L(\cos\alpha\cos\beta - \sin\alpha\sin\beta)^{*}$$

$$y^{`} = L(\cos\alpha\sin\beta + \sin\alpha\cos\beta)^{\dagger}$$

$^{*}\cos(\alpha+\beta) = \cos\alpha\cos\beta - \sin\alpha\sin\beta$
$^{\dagger}\sin(\alpha+\beta) = \sin\alpha\cos\beta + \cos\alpha\sin\beta$

- Beste era batera:

$$x^{'} = x\cos\beta - y\sin\beta$$

$$y^{'} = x\sin\beta + y\cos\beta$$

- Eta matrize bidez:

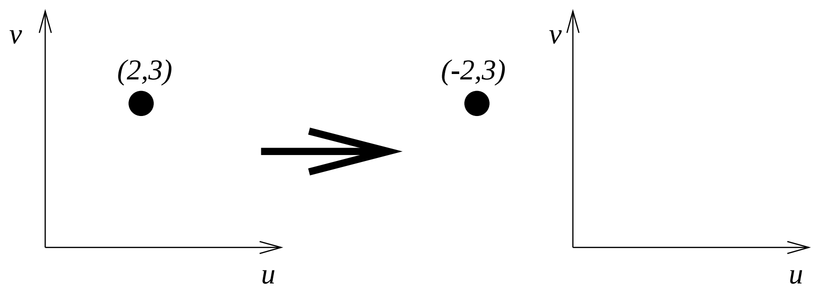
$$\begin{pmatrix} x^{'} \\ y^{'} \end{pmatrix} = \begin{pmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

- 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.



- Householderren matrize erreflexutzaileak erabili 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.

-

- Edozein ardatzekiko erreflexua: x bektorearekiko α gradu baditu $\begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix}$ izango da.

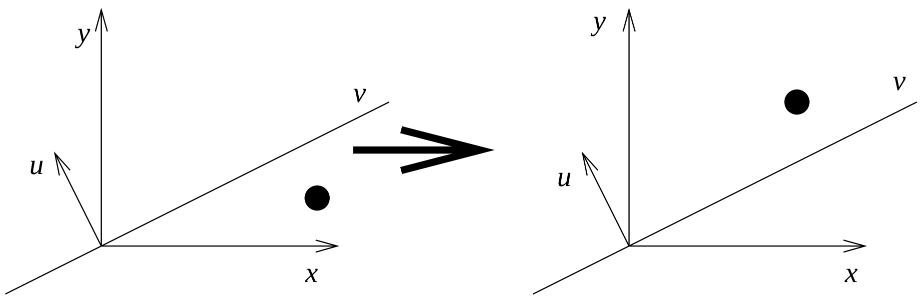
- Berekiko elkarzuta den u behar da: $\begin{pmatrix} -\sin\alpha \\ \cos\alpha \end{pmatrix}$.

- 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:

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



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.

- Koordenatu homogeneoetan puntua adierazteko $\begin{pmatrix} hx \\ hy \\ h \end{pmatrix}$ erabiliko dugu ($h = 1$ gehienetan) Bektorea berriz $\begin{pmatrix} x \\ y \\ 0 \end{pmatrix}$ izango da.

- 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

- 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.

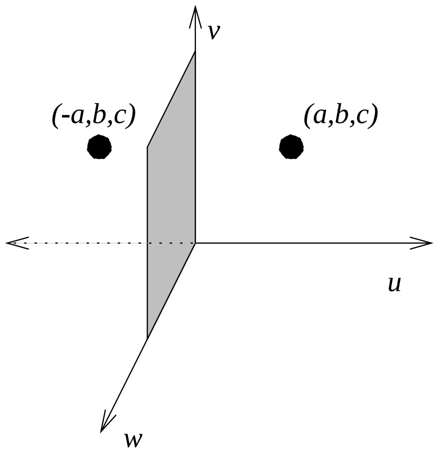
- Erreflexua

  Housenholderren matrize erreflektatzaileak, baina plano batekiko:

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

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

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


v
(-a,b,c)
(a,b,c)
u
w

Biraketa

- Biraketa ardatza. Erreferentzi sistemakoa edo edozein.

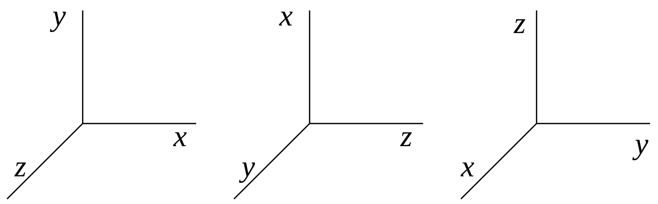
  w ardatzarekikorekiko:

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

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

$$w^{'} = w$$

  w gure hiru ardatzetako edozein izan daiteke:



- 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:

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

- x ardatzarekiko biraketa

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

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

$$x^{'} = x$$

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

- y ardatzarekiko biraketa

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

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

$$y^{'} = y$$

edo beste era batera:

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

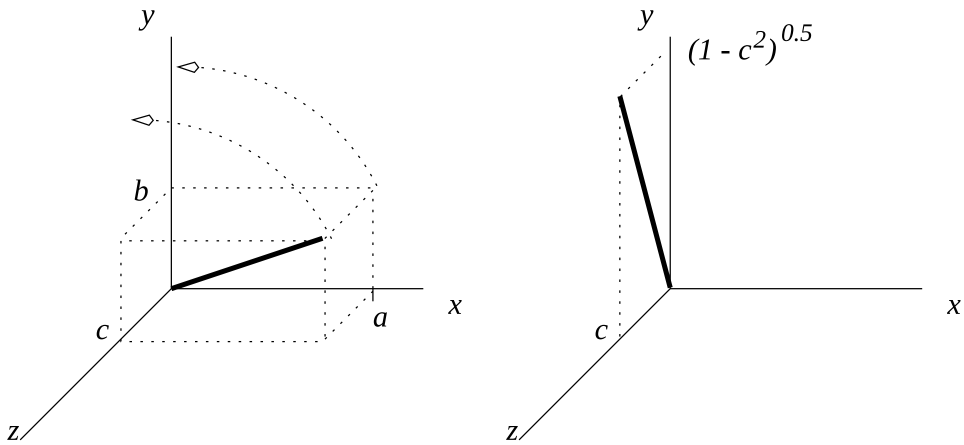
$$y^{'} = y$$

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

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

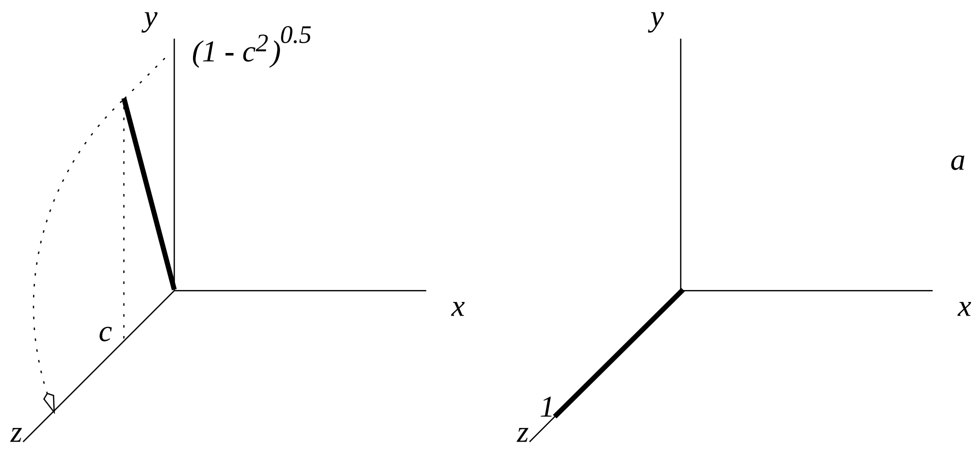
- Edozein ardatzekiko biraketa Orain arte ikusitako biraketetan oinarritu gaitezke, horretarako biraketa ardatza erreferentzi sistemaren ardatz bat izatea lortu behar da.

  1. Lehenengo biraketa:



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 itzultzeazkenik haserako bi biraketak desegin behar dira:

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.

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.**

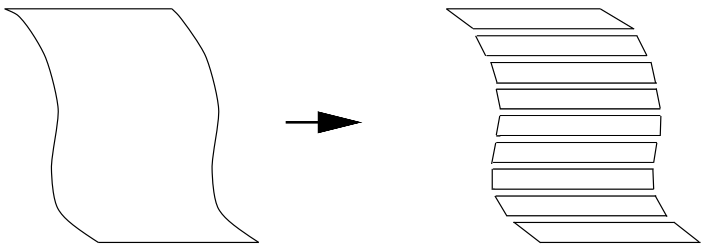
5. funtzio matematikoak ere erabili daitezke. Adibidez:

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

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.



- 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. Horretaz gain informazio gehiago jaso daiteke.

- Barne antolaketa egin daiteke tratamendua hobetzeko.

- Poligonoz poligono ihardun beharrean ertzetan oinarritu gaitezke. Horrela ertz bakoitza behin bakarrik landuka da.

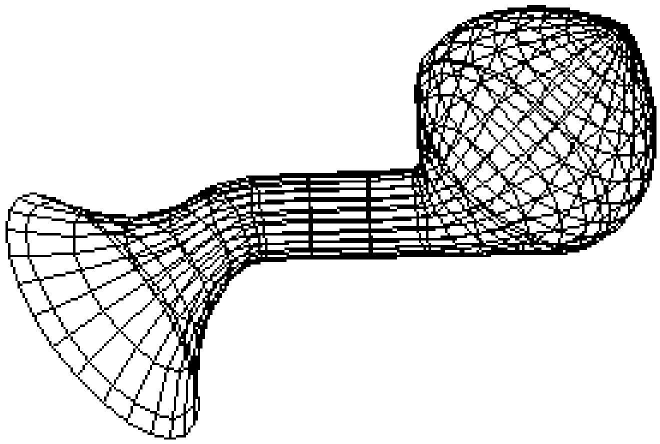
objektuen eraiketa

- 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 eskatzen duten 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 ez poligonoen araberakoak, poligonoak bereizmen haundirako badaude kalkulu gehiegi suposatzen dute, eta alderantziz gertatzen bada irudi kaskarra lortuko dugu.

Adierazpidearen desabantailak:

- Datu egitura eraikitzea zailagoa da.

- Txatalen 16 kontrol puntuek osotasuna mantentzeko jarraitasun baldintzak bete behar dituzte.

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

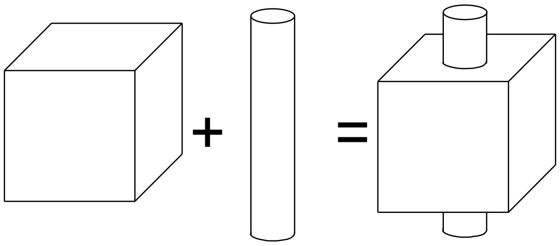
- Objektua zuhaitz bidez adierazten da.

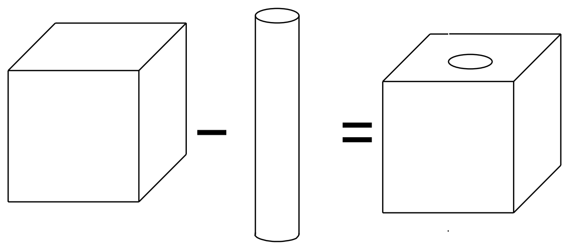
- Hostoek oinarrizko objektuak adierzten dituzte.

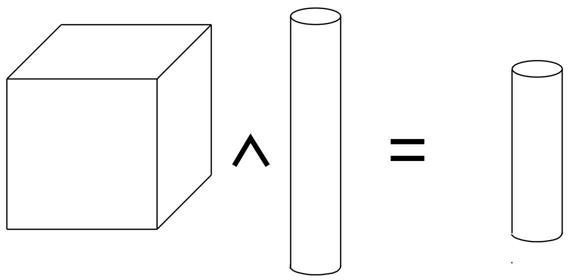
- barneko erpinek objektu arteko eragiketak adierazten dituzte.

- oinarrizko objektuak esferak, konoak, kuboak . . . izan daitezke.

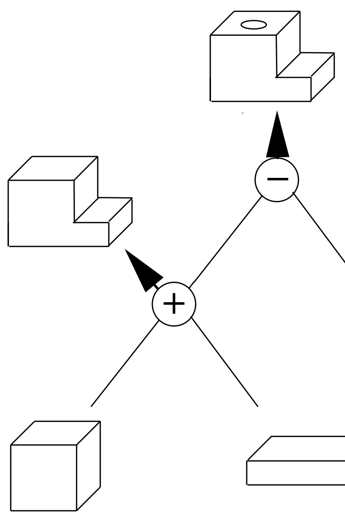
- eragiketak aldaketa linealak edo eragiketa bulearrak izan daitezke.







- adibide bezala:



CSGren desabantailak

- Irudia sortzeko denbora asko behar da

- Eragiketak globalak dira, ezin dira zati batengan eragin.

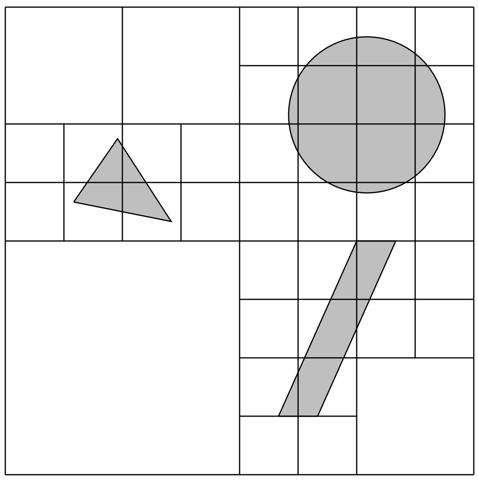
Hala ere Objektuak sortzeko erabilerrazak dira.

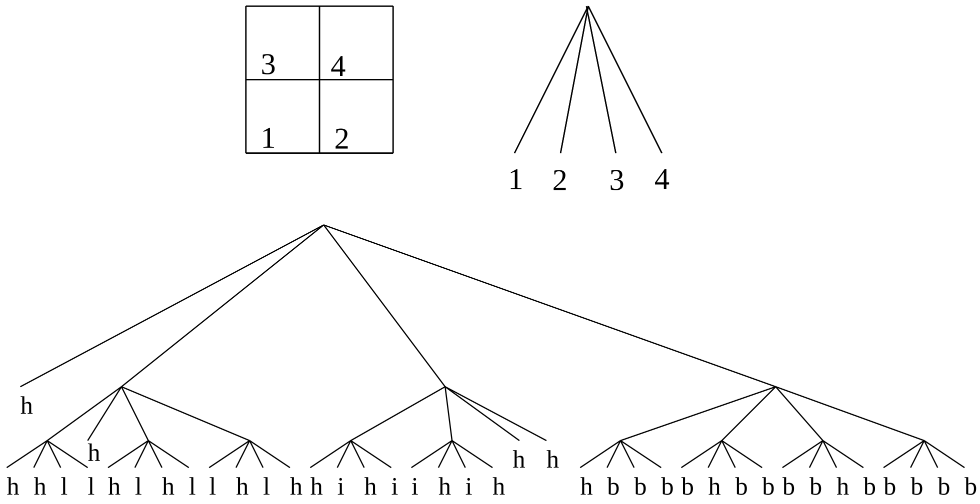
Espazio partiketa

- 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.





h = hutsik

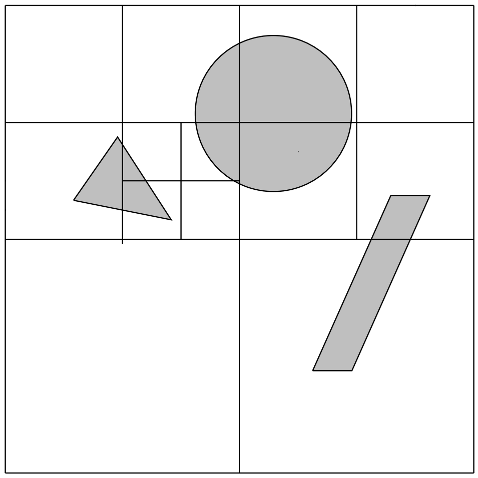
i = hirukia

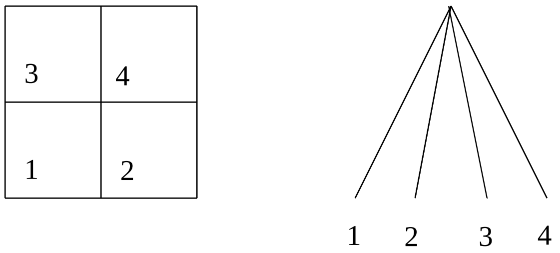
b = borobila

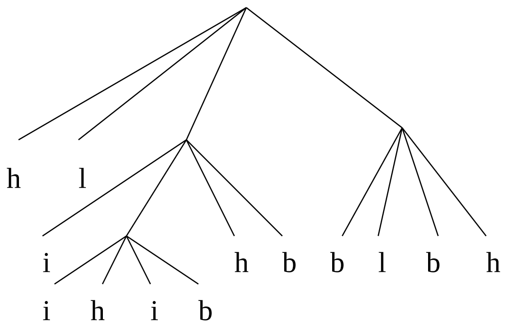
l = laukizuzen okertua

- Espazioaren zatiketa bereizmenaren arabera finkatu behar da.

- Hutsik dagoen espazio zatia ez dago zatitu beharrik.







h = hutsik

i = hirukia

b = borobila

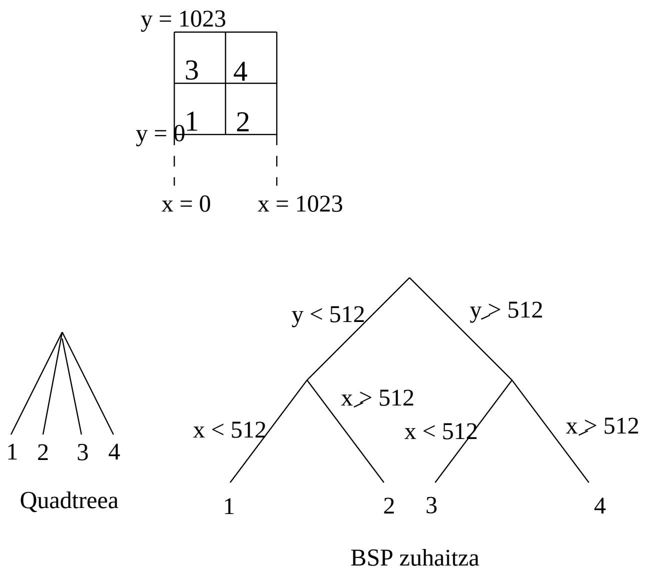
l = laukizuzen okertua

- Borneak bakarrik har ditzake kontuan.

- Orokorrean, hosto batek poligono lista izango du.

BSP zuhaitzak

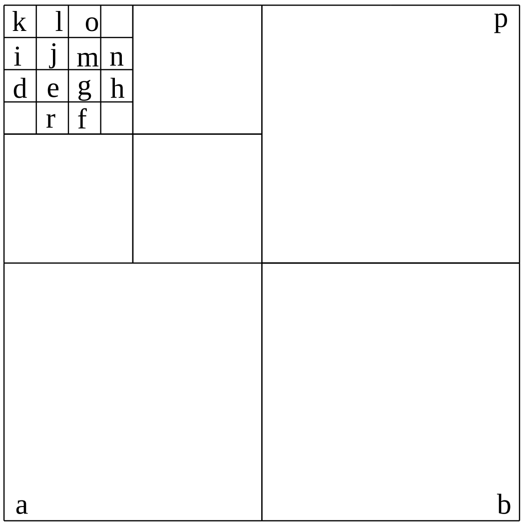
- Hau ere zuhaitz egitura da.
- Espazioa beti bitan banatzen da.
- Zatitzeko planoak erabiltzen dira.

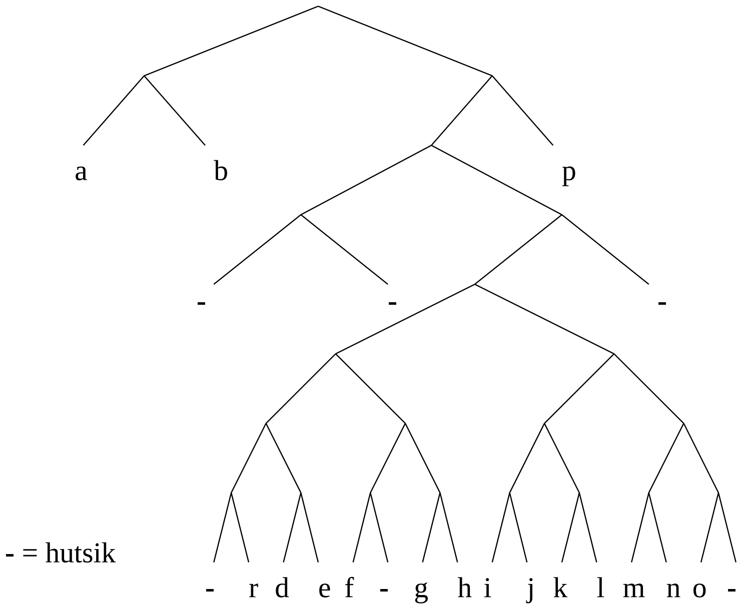


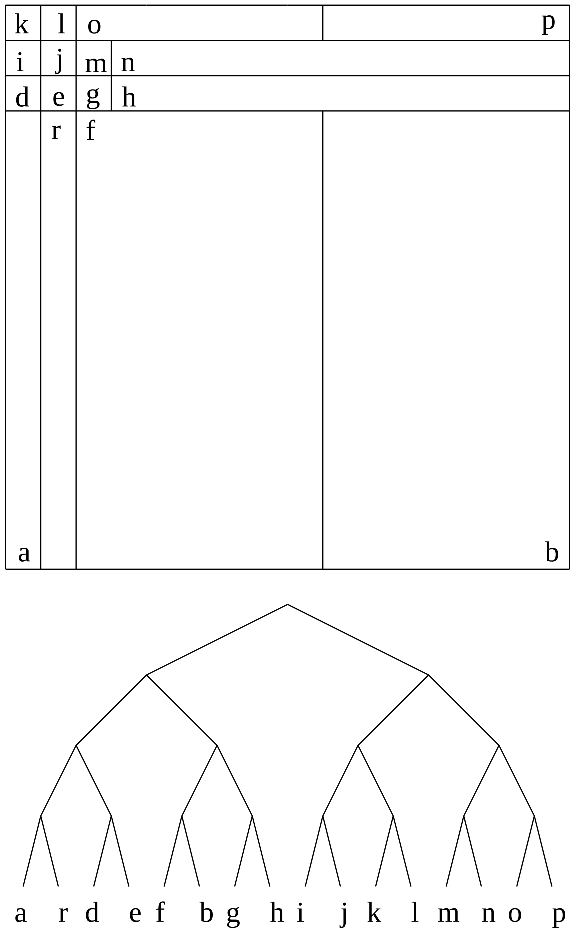
- Izpi hedaketarako egokia. Izpiaren ibilbidea zuhaitzean mugitzearen parekoa da eta bertako objektuekin ebaketa testak egitea.

moldaerazko zatiketa

- Zatiketarako planoa edozein izan daiteke.





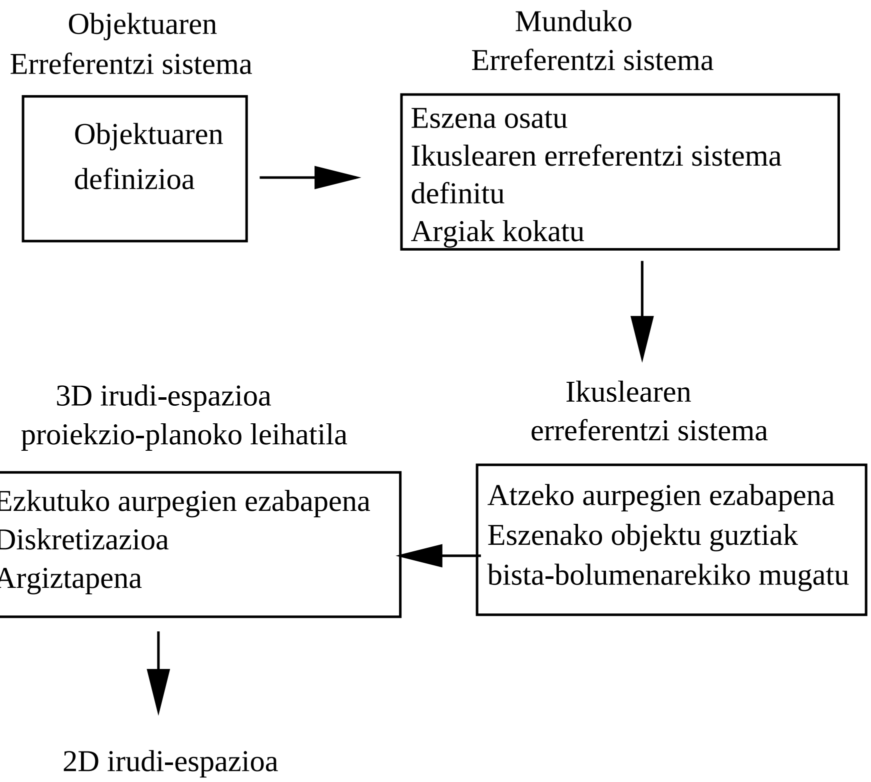


- Zuhaitz orekatuagoak lortzen dira, sakonera txikituz.

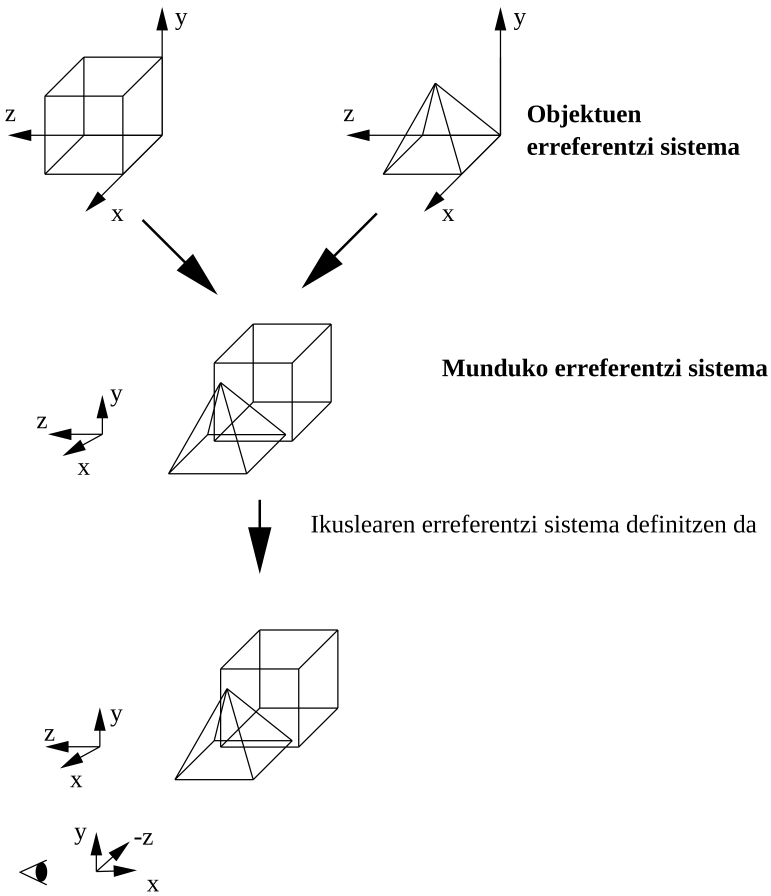
B-rep bidez irudia sortuz

- Sarrera poligono lista da, irteera koloredun pixel multzoa.

- Koordenatu espazio desberdinak igaroko ditugu.

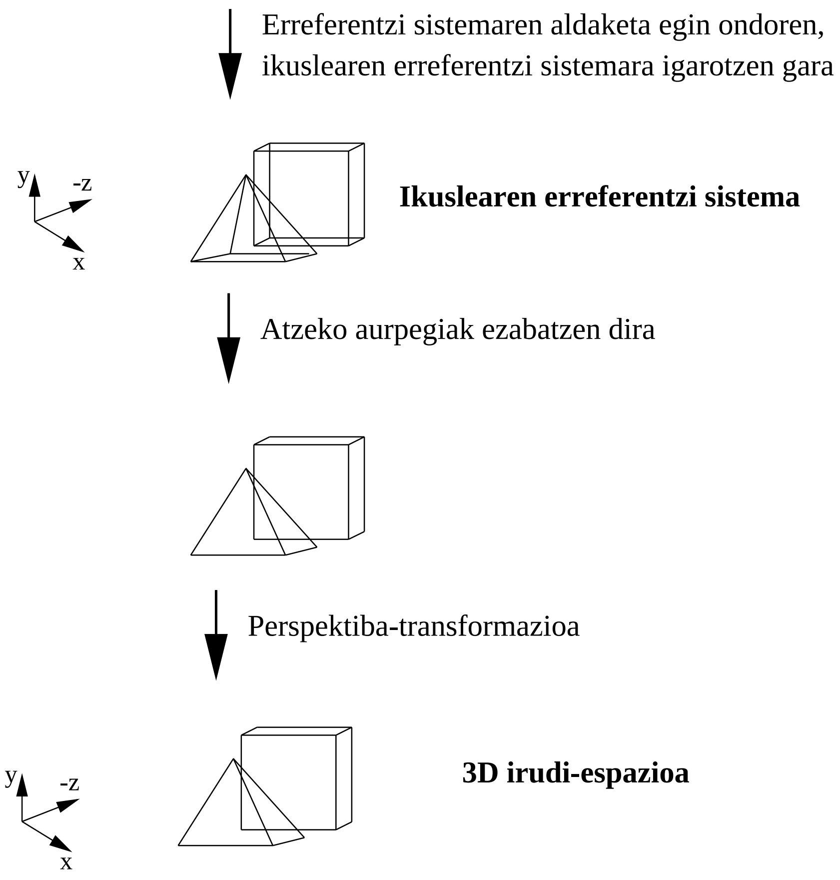


- Objektuek bere koordenatu sistema daukate.
- Guztiak mundu bakar batean sartzen dira, bertan koordenatu sistema jakina izango dugu eta objektu bakoitza hor adierazi beharko da.
- Argien eta ikuslearen kokapena sistema horretan adieraziko dira, ikuste direkzioa ere bai.



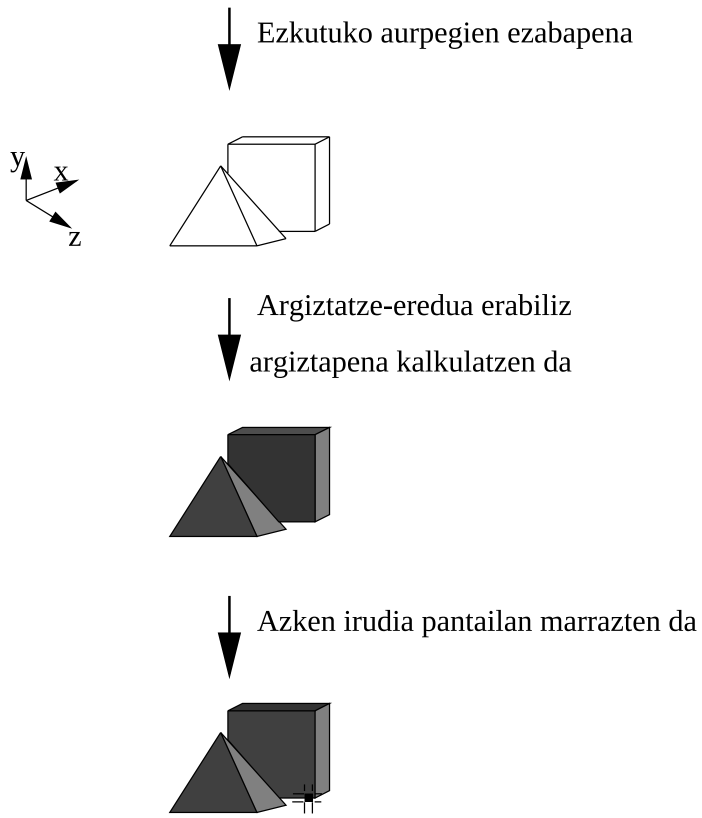
- Horren arabera ikulsearen erreferentzi sistema kalkulatuko da.

- Objektu guztiak sistema horretara pasa, atze-aurpegiak ezabatu eta mugaketa burutu.



- Ezkutuko aurpegiak kendu eta itzaleztatu.

- Azkenik Pixeletara pasa behar dugu irudia.



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

Txatal bidez irudia sortuz

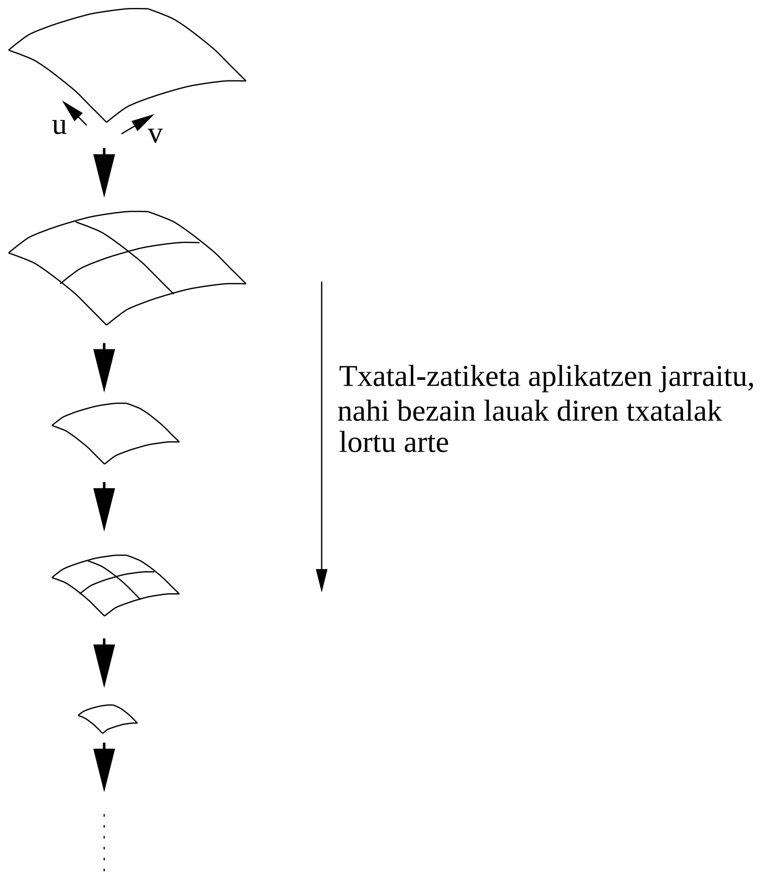
- Errazena txatal sarea poligono adierazpidera pasatzea.

- Txatalak edozein bereizmen eduki dezake, baina azpi-txataletan zatitu dezakegu, nahi bezalako txatalak lortu arte, horrela guk finkatutako bereizmena lortu ahal izango dugu.

- Objektu hauek erabiltzailek era elkarreragilean sortzen ditu, baina irudia lortzeko itzulpena egin eta B-rep adierazpidearekin lan egin daiteke.

- Poligono kopurua txatal kopurua baino handiagoa izango da.

- itzulpena bi urratsetan egin daiteke: Bezier motako txatalera itzuli, eta Bezier txatalen zatiketa.



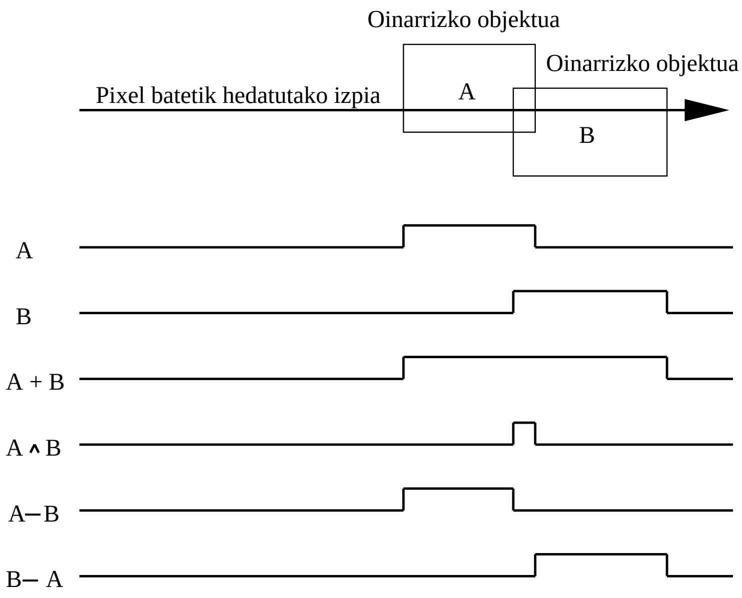
- Zatiketa bukatzeko lautasun baldintzak jarri behar dira.

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.

Ikuste Sistemak

- 3D-tik 2D-ra pasa behar dugu.

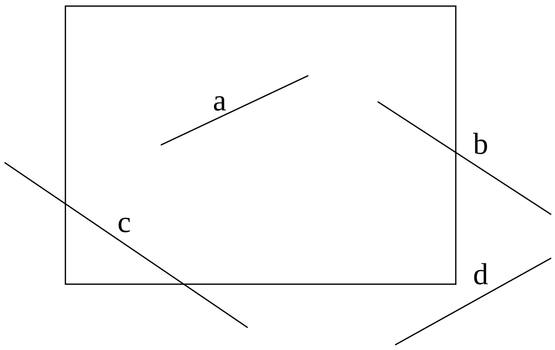
- 2D-tik pantalilara. Agertu nahi den zatia zehaztu eta horren irudia azaldu behar da.

  1. Mugaketa: agertu nahi duguna mugatu.

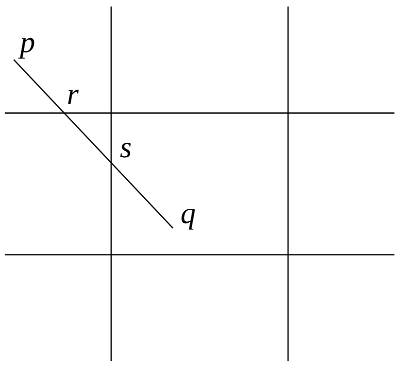
  2. Mugatutakoa pantailaren erreferentzi sistemara pasa.

Mugaketa

- Marren mugaketa: Cohen-Sutherland algoritmoa

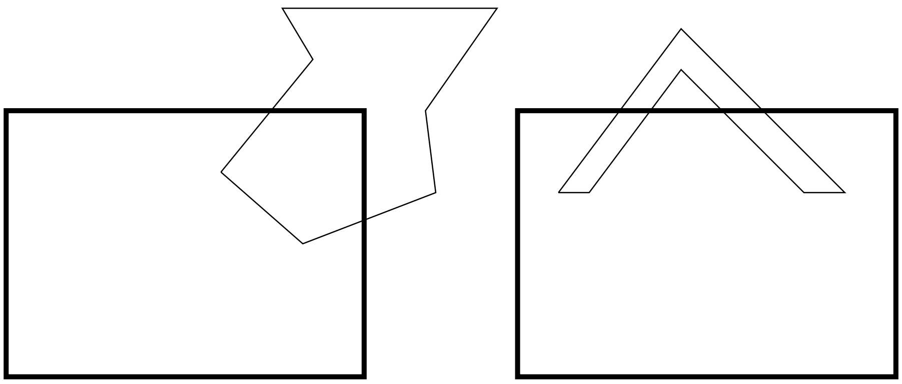


Espazioa 9 zatitan banatzen du eta kode batzuren bidez lana errazten da.



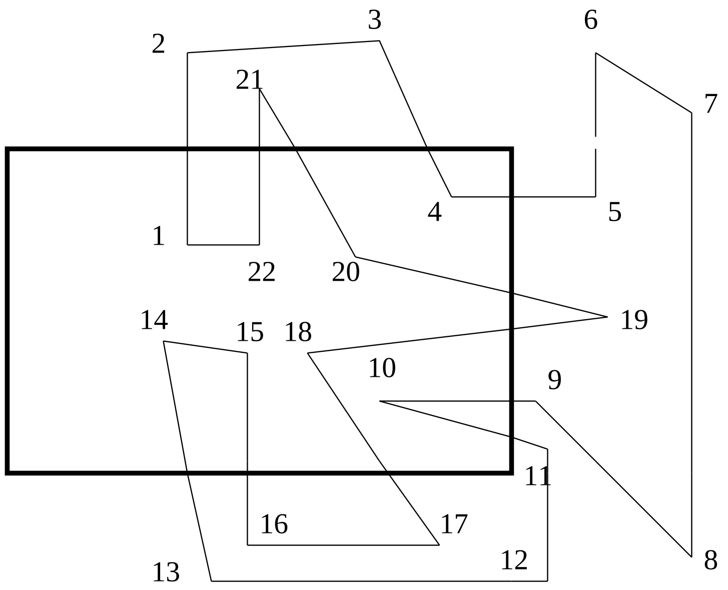
- Poligono mugaketa: Sutherland-Hodgman algoritmoa.

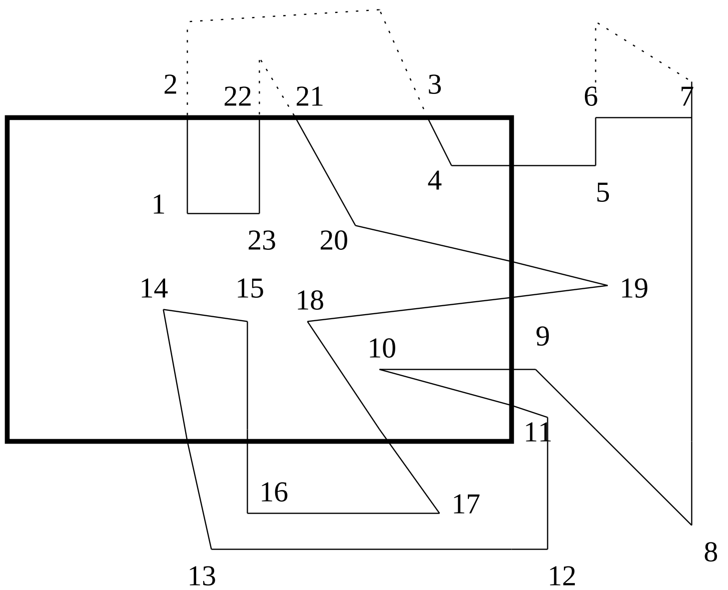
  Poligonoa bete egin behar bada bere mugetako ertzak mugatzea ez da nahikoa, poligono izaera galdu egin baitaiteke.

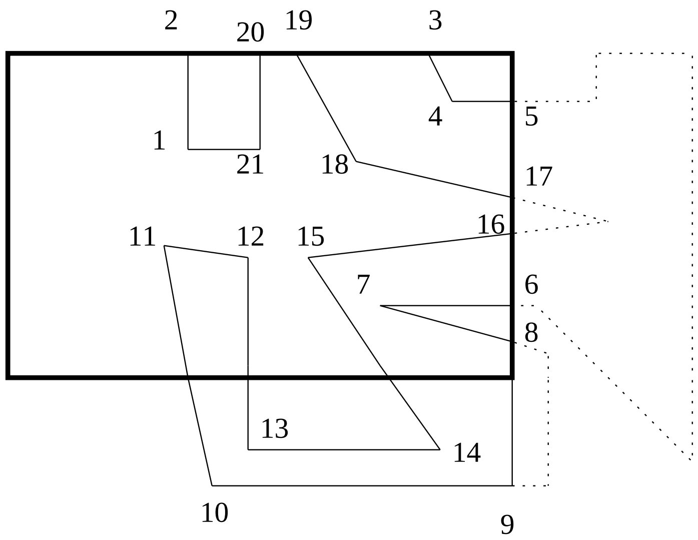


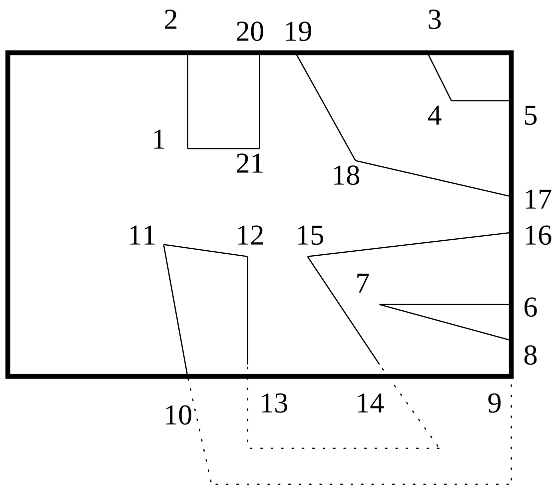
(a) (b)

Algoritmoak poligono bat hartu eta mugatu ondoren poligono bat itzultzen du.


3
6
2
21
7
4
5
1
22
20
14
15
18
19
10
9
11
16
17
12
13
8

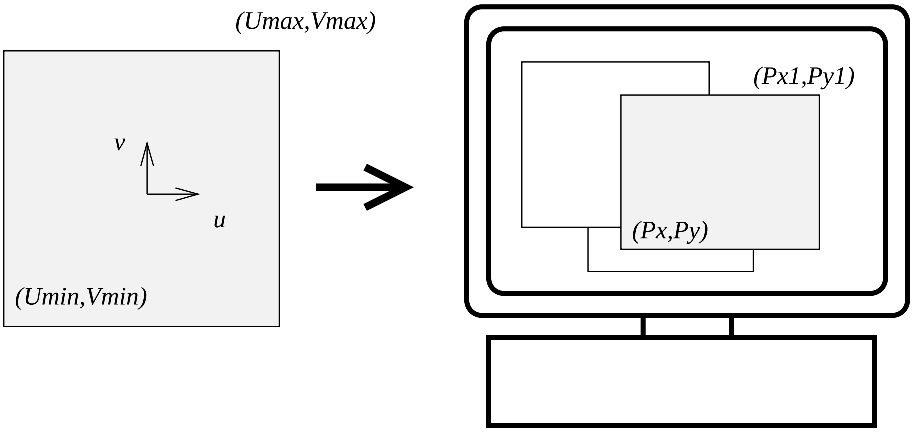

2
22
21
3
6
7
4
5
1
23
20
14
15
18
19
9
10
11
16
17
8
13
12


2
20
19
3
1
21
4
5
18
17
11
12
15
16
7
6
8
13
14
10
9


2
20
19
3
1
21
4
5
18
17
11
12
15
16
7
6
8
10
13
14
9

laukiratzea

- Erreferentzi sistema aldaketa besterik ez da.



- Azken finean $\overline{u} = Au + T$ motatako aldaketa.

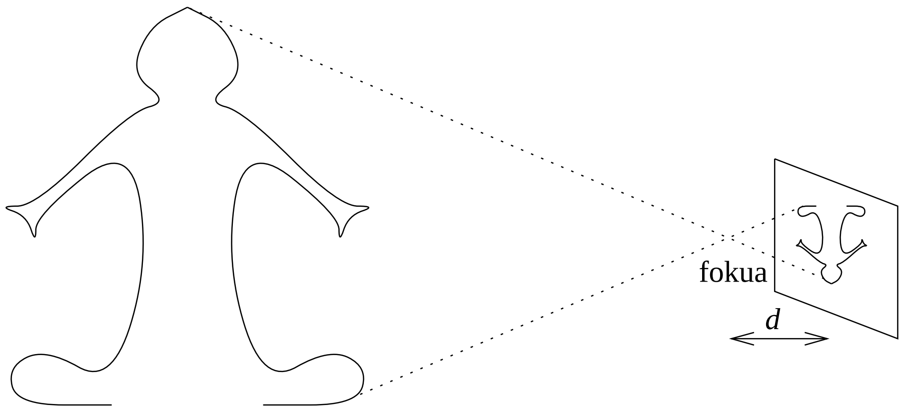
$$A = \begin{pmatrix} \frac{P_{x1}-P_x}{U_{max}-U_{min}} & 0 \\ 0 & \frac{P_{y1}-P_y}{V_{max}-V_{min}} \end{pmatrix}$$

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

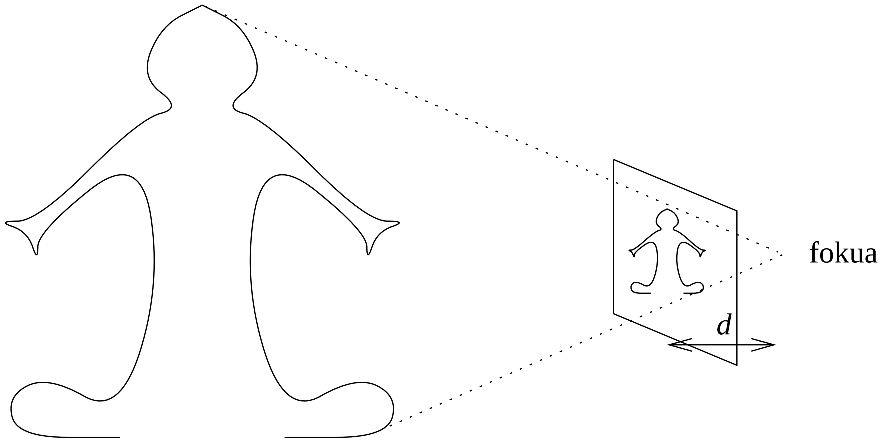
Kokapen eta neurri aldaketa da.

3D-tik 2D-ra

- Gure begiek edo argazki kamerak:

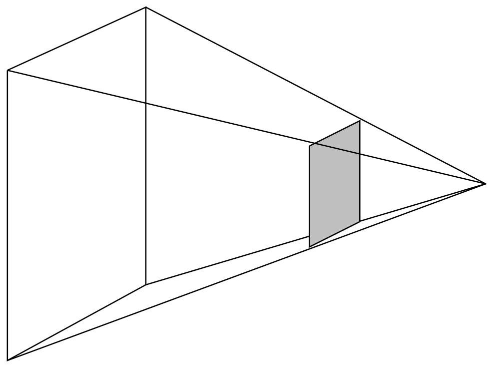


- Alderantziz egon ez dadin proiekzio planoa aurrean jar daiteke.



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

- 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**.

- Eman beharreko urratsak:

  - Erreferentzi sistema aldaketa: kamararen erreferentzi sistemara pasa behar da.

  - Proiektatzea.

- Hori egin ahal izateko behar den informazioa:

1. Kamarak nola ikusten duen adierazi behar du, hau da, kamararen erreferentzi sistema.

2. Kamarak 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

- 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 edozein puntutarantz proiektatuko den adieraziko duen puntua, PRP. Honen koordenatuak erreferentzi sistema berrian adierazten dira.

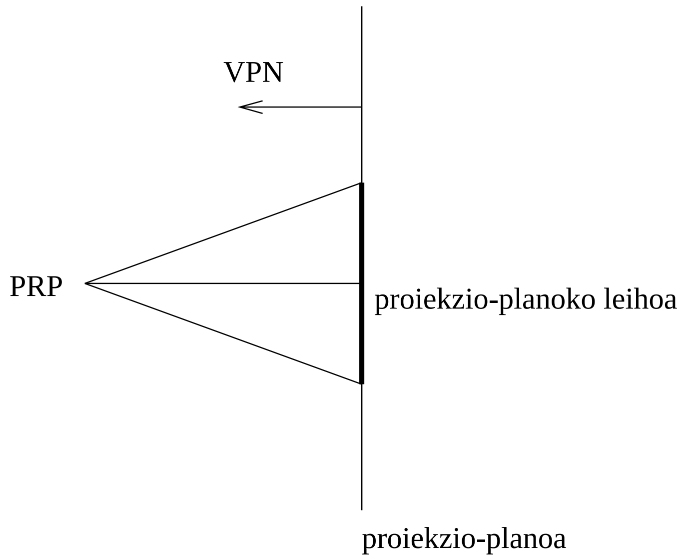
6. Plano urrunerako distantzia, b, distantzia honetaz harantzagoko objektuak ikusezinak bihurtzen dira.

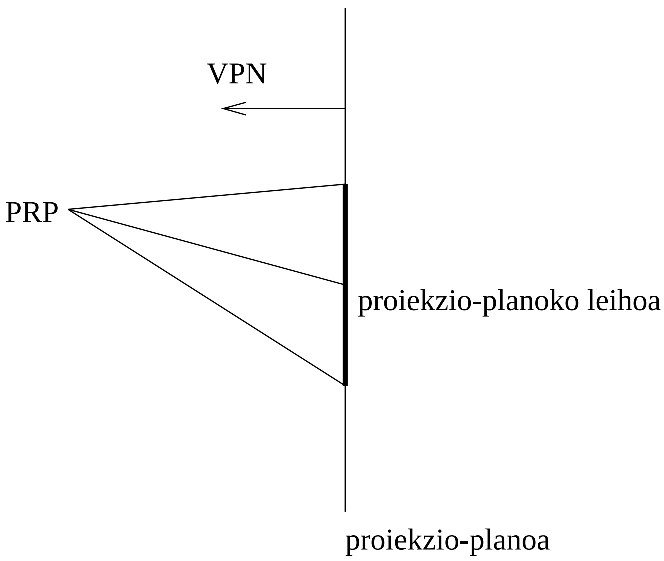
7. Plano hurbilerako distantzia, f. Honen eta aurrekoaren arteko objektuak besterik ez dira ikusiko.

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:



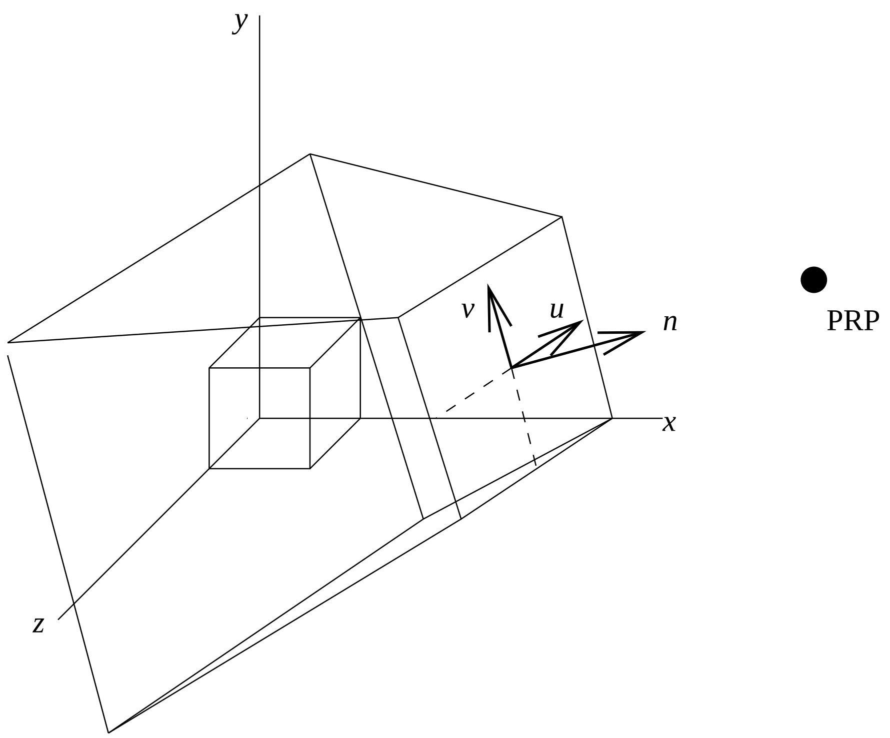


- Kamarak begiratzen duen direkzioa edozein izan daiteke.

- Proiekzio planoak edozein bektore dauka goranzko bektore bezala.

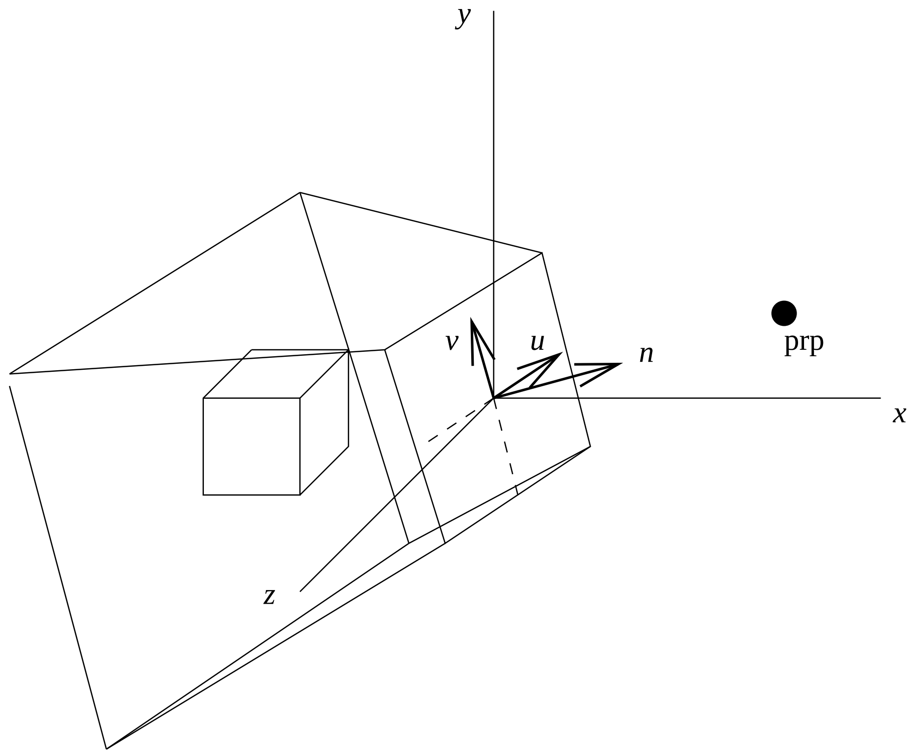
- Ikuste bolumena ematen da, horretarako proiekzio planoko leihatila, urruneko planoa eta hurbileko planoa ematen dizkigute.

Irudia sortzeko urratsak



- Munduaren erreferentzi sistematik kamararenera pasa behar da: VRP, VPN eta VUP parametroak erabili behar dira. Erreferentzi sistema berriaren oinarria eta jatorria zehaztu behar dira:

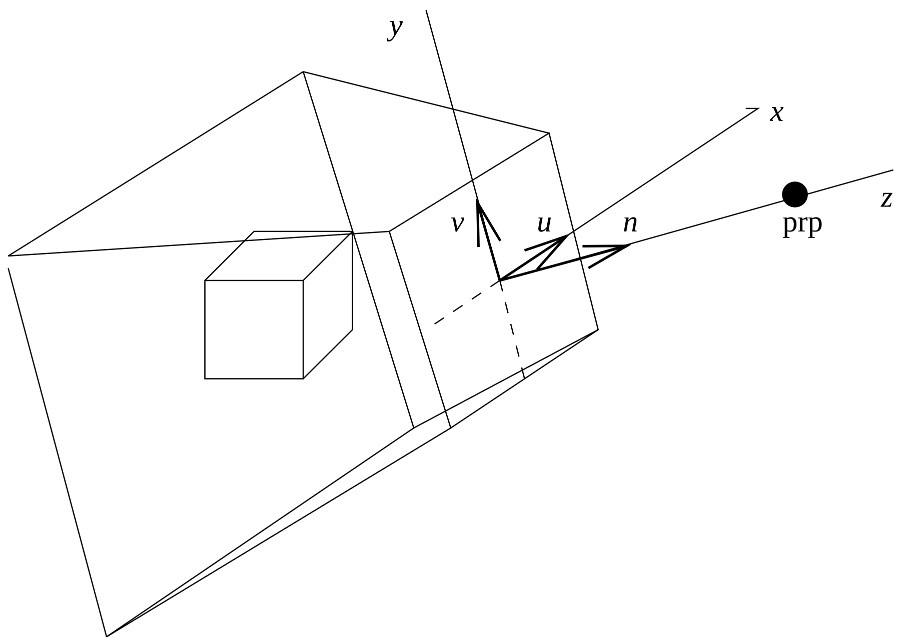
  1. jatorria VRP da.



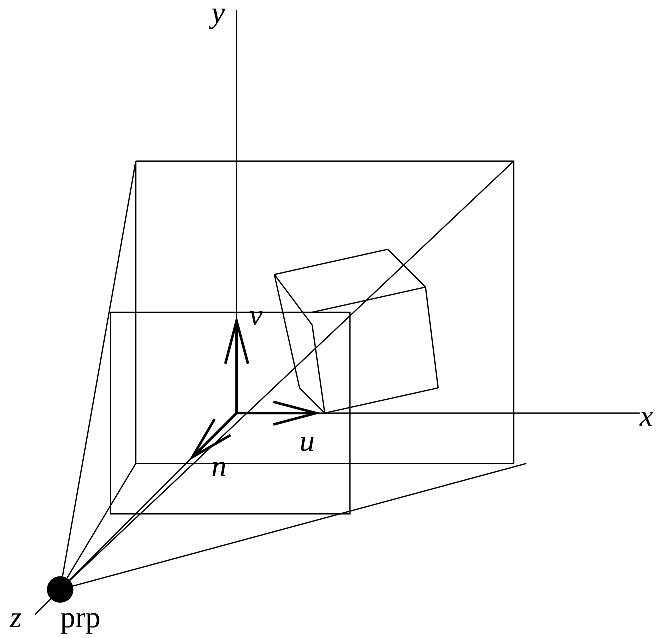
2. z_k: Hurbiltasuna eta urruntasuna VPN bektoreak adierazten ditu.

3. x_k: Goranzko direkzioa eta z_k ardatzek zehaztutako planoarekiko elkarzuta denez bi horien arteko bektore biderkaketa bidez lor daiteke.

4. y_k: beste bien bektore-biderkaketa bidez.



- kamerak ikusten duen bezala adierazi eszena:



- Proiekzioa lortu, baina zeiharra bada lehenengo proiekzio direkzioa eta proiekzio leihotik fokorako, PRP-rako, direkzioa bateratu behar dira.

- 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:

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