



ШУТИС, Мэдээлэл Холбооны Технологийн Сургууль

F.CS209 Компьютерийн график

Лекц 10 – Clipping 2
Боловсруулсан багш: Х.Хулан

2022 он

Cohen-Sutherland pseudocode

```
int clipSegment(Point2& p1, Point2& p2, RealRect W)
{
do{
if(trivial accept) return 1; // whole line survives
if(trivial reject) return 0; // no portion survives
// chop
if(p1 is outside)
// find surviving segment
{
if(p1 is to the left) chop against left edge
else if(p1 is to the right) chop against right edge
else if(p1 is below) chop against the bottom edge
else if(p1 is above) chop against the top edge
}
```



```
else // p2 is outside
// find surviving segment
{
if(p2 is to the left) chop against left edge
else if(p2 is to right) chop against right edge
else if(p2 is below) chop against the bottom edge
else if(p2 is above) chop against the top edge
}
}while(1);
}
```



Үр дүнг ашиглан харьцуулатыг хурдасгах

► Startpoint болон Endpoint бүрийг 4 битээр кодчилон доорх үр дүнтэй харьцуулна.

$b_0b_1b_2b_3$

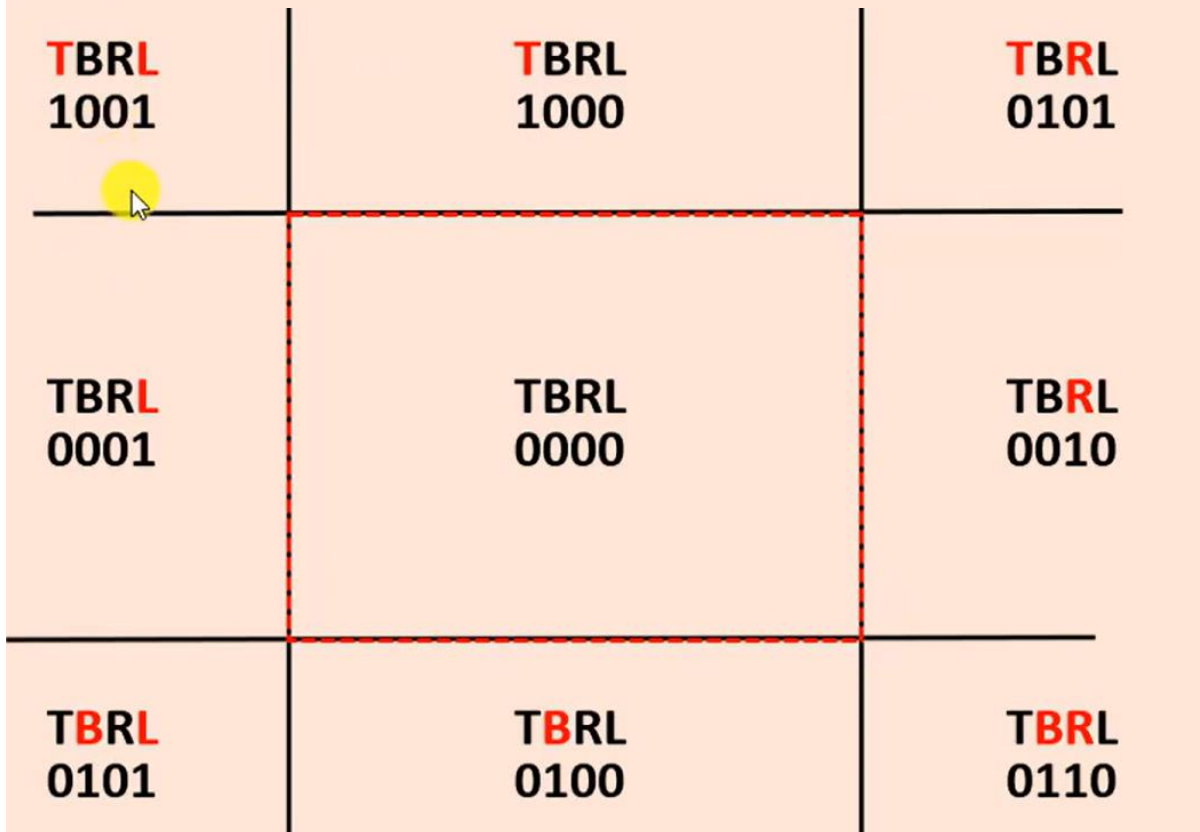
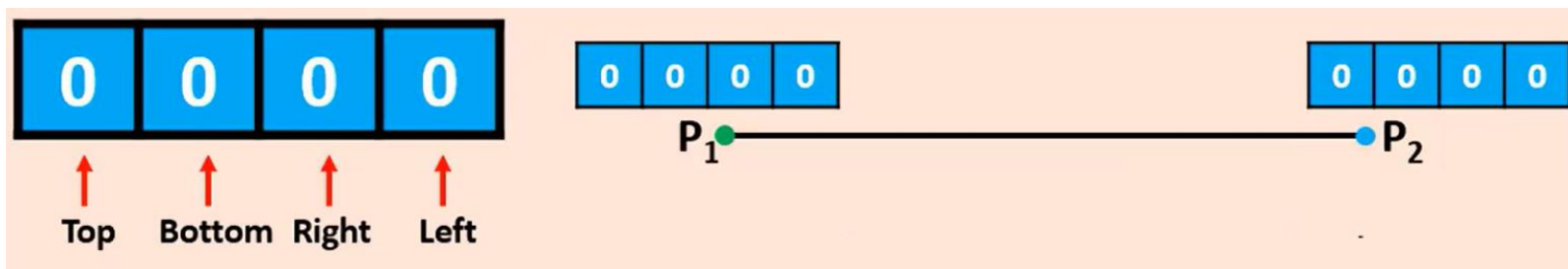
$b_0 = 1$ if $y > y_{\max}$, 0 otherwise

$b_1 = 1$ if $y < y_{\min}$, 0 otherwise

$b_2 = 1$ if $x > x_{\max}$, 0 otherwise

$b_3 = 1$ if $x < x_{\min}$, 0 otherwise

1001	1000	1010	$y = y_{\max}$
0001	0000	0010	
0101	0100	0110	$y = y_{\min}$
$x = x_{\min}$		$x = x_{\max}$	

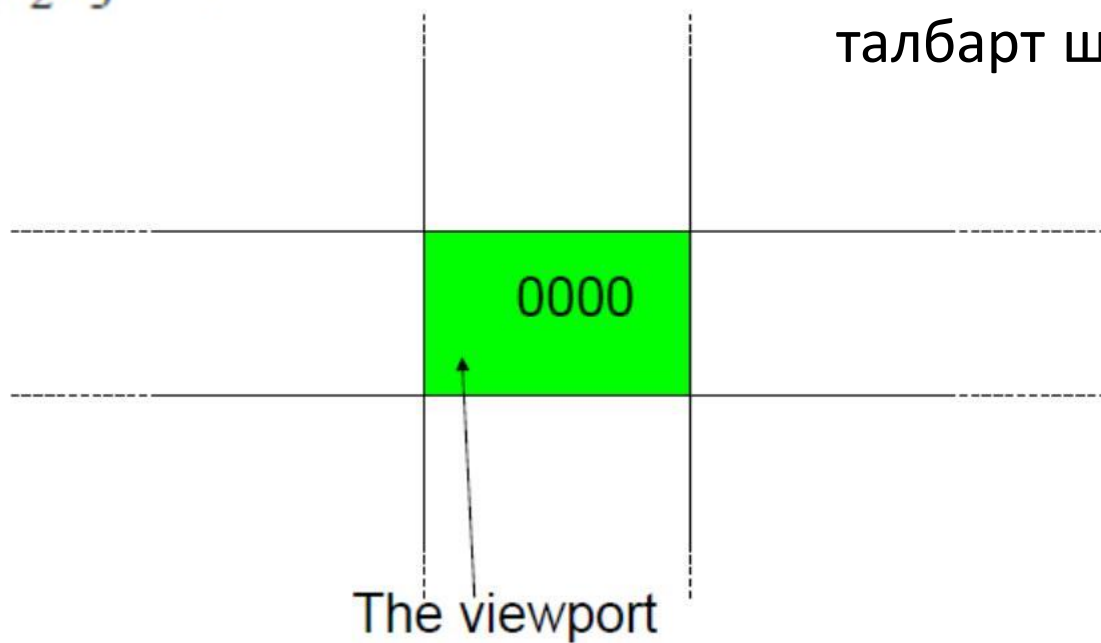


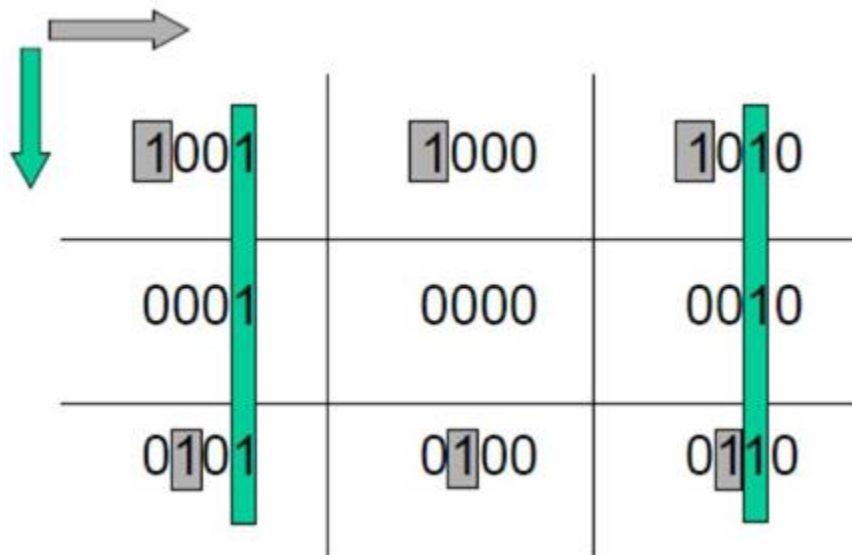
1001	1000	1010	$y = y_{\max}$
0001	0000	0010	
0101	0100	0110	$y = y_{\min}$
$x = x_{\min}$		$x = x_{\max}$	

Cohen Sutherland 2D хэмжээст

$$b_0b_1b_2b_3 = 0$$

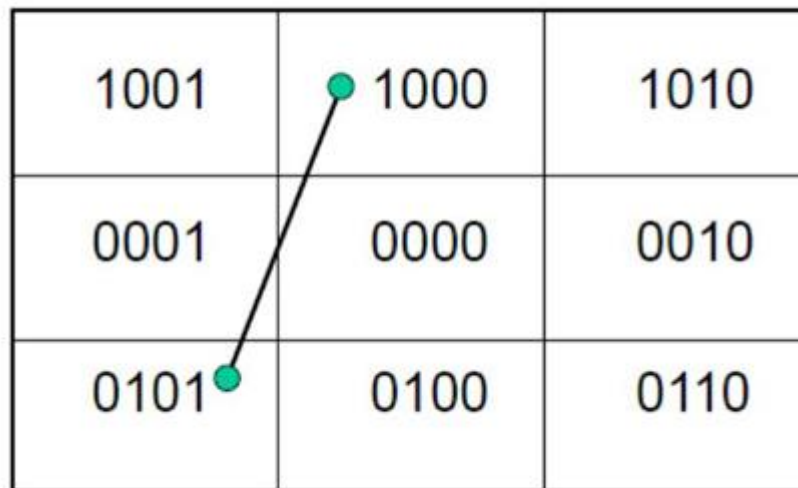
2D талбарын хувьд
4 бит бүхий 9
талбарт шалгана.





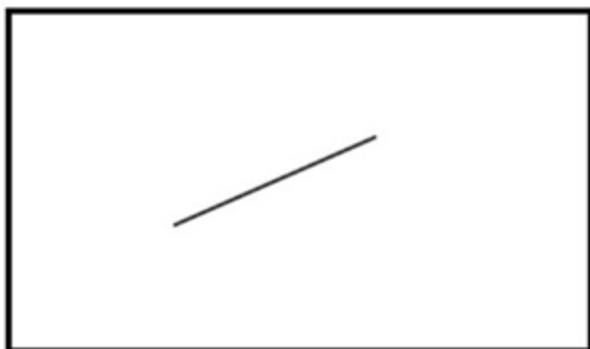
9 талбарыг 4 битээр кодчилох ба шулууны эхлэл ба төгсгөлийн цэгүүд уг талбаруудын хаана байгааг шалгана.

0000 буй хэсэг бол бидэнд харагдах Viewport билээ.



Жишээ:

1001	1000	1010	$y = y_{\max}$
0001	0000	0010	
0101	0100	0110	$y = y_{\min}$
$x = x_{\min}$		$x = x_{\max}$	

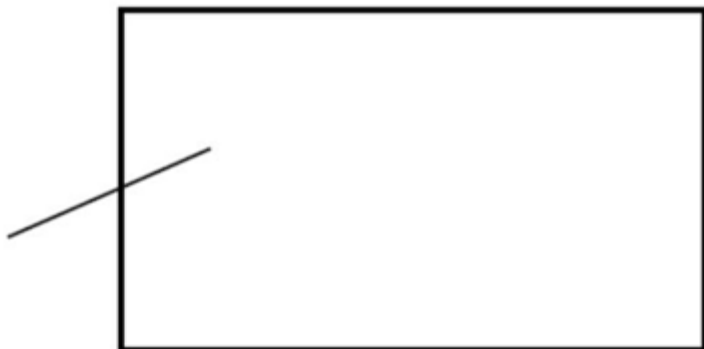


$p1 = p2 = 0000$

Accept

Жишээ:

1001	1000	1010	$y = y_{\max}$
0001	0000	0010	
0101	0100	0110	$y = y_{\min}$
$x = x_{\min}$		$x = x_{\max}$	



$p1 \neq 0000$

$p2 = 0000$

Clipping

Жишээ:

1001	1000	1010	$y = y_{\max}$
0001	0000	0010	
0101	0100	0110	$y = y_{\min}$
$x = x_{\min}$		$x = x_{\max}$	

/



$p1 \ \& \ p2 \neq 0000$
Trivial **Reject**

Жишээ:

1001	1000	1010
0001	0000	0010
0101	0100	0110

$p1 = 1001, p2 = 0101$

$1001 \& 0101 = 0001 \neq 0000 \rightarrow \text{Reject}$

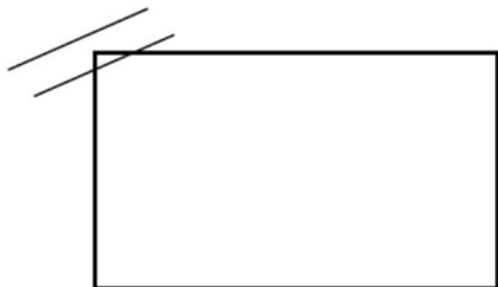
Жишээ:

1001	1000	1010
0001	0000	0010
0101	0100	0110

$p1 = 0101$ $p2 = 0110$

$0101 \& 0110 = 0100 \neq 0000 \rightarrow \text{Reject}$

Жишээ:



1001	1000	1010
0001	0000	0010
0101	0100	0110

$p1 \neq 0000$ $p2 \neq 0000$

$p1=0101$ $p2=1000$ $0101 \& 1000 = 0000$

Төгсгөлийн цэгүүд хоёулаа гадна байна.

Гэсэн хэдий ч хоёр өөр талбарт байна. Иймд цаашид нарийвчлан шалгах хэрэгтэй.

Огтлолын тооцоолол хийх

Хилтэй огтлолцож буй хэсгийг
хялбархан тооцоолж болно.

Параметр бүхий шулуун буюу энэ
нь шугаман интерполяци юм.

$$p(\alpha) = (1 - \alpha)p_1 + \alpha p_2$$

$$0 \leq \alpha \leq 1$$



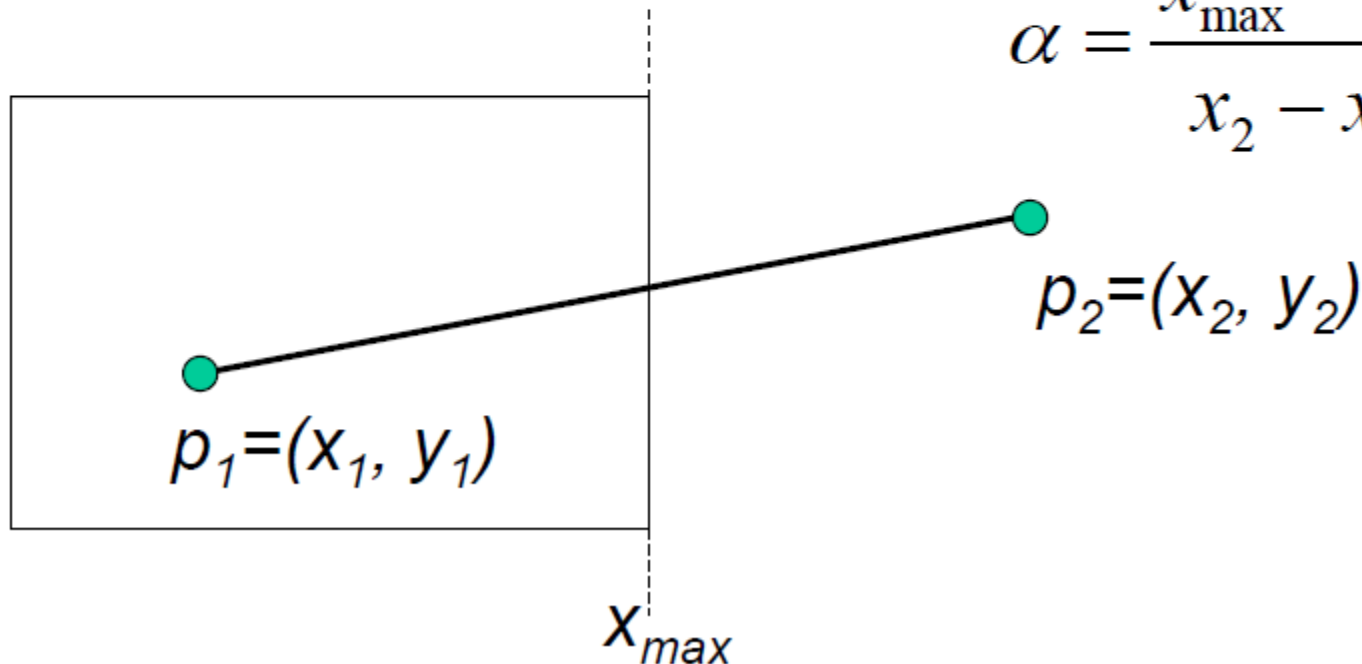
Баруун хилийн x_{\max} -тай
огтлолцох α параметрийг
хялбархан тооцоолж болно.

$$x_{\max} = (1 - \alpha)x_1 + \alpha x_2$$

$$x_{\max} = x_1 + \alpha(x_2 - x_1)$$

$$x_{\max} - x_1 = \alpha(x_2 - x_1)$$

$$\alpha = \frac{x_{\max} - x_1}{x_2 - x_1}$$



Эцэст нь y -координатыг
 α -г шугамын тэгшитгэлд
тавих замаар оруулан
тооцоолно.

$$\alpha = \frac{x_{\max} - x_1}{x_2 - x_1}$$

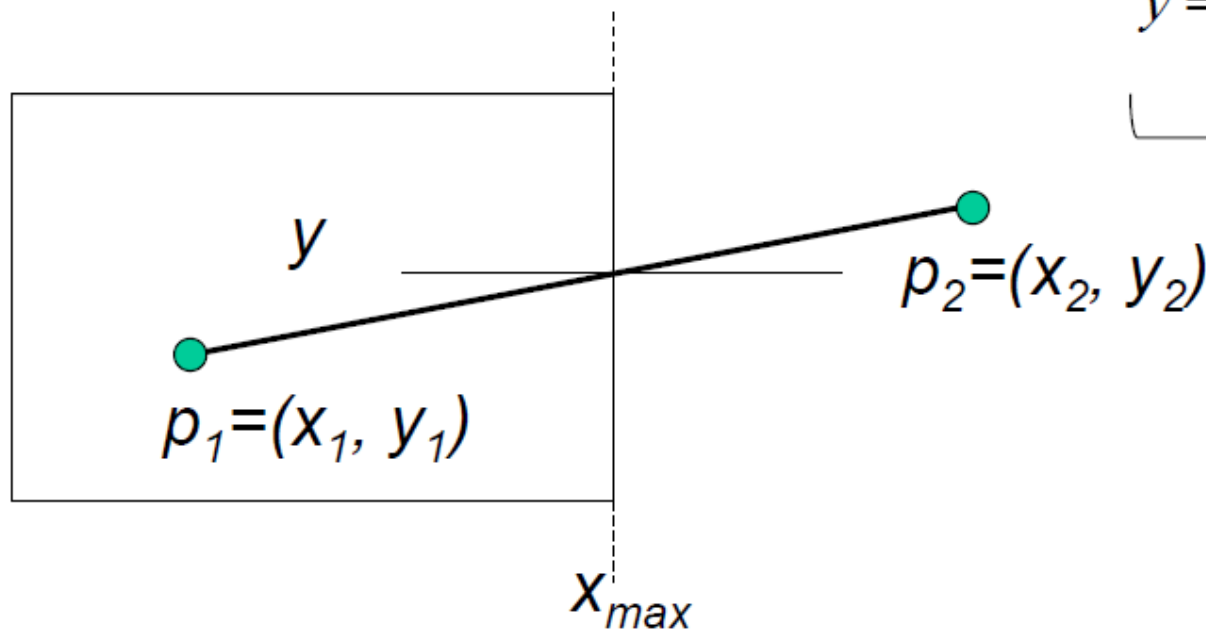
$$y = (1 - \alpha)y_1 + \alpha y_2$$

$$y = y_1 + \alpha(y_2 - y_1)$$

$$y = y_1 + \frac{x_{\max} - x_1}{x_2 - x_1}(y_2 - y_1)$$

$$y = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x_{\max} - x_1)$$

Хоёр цэгийн томъёолол



Огтлолцол

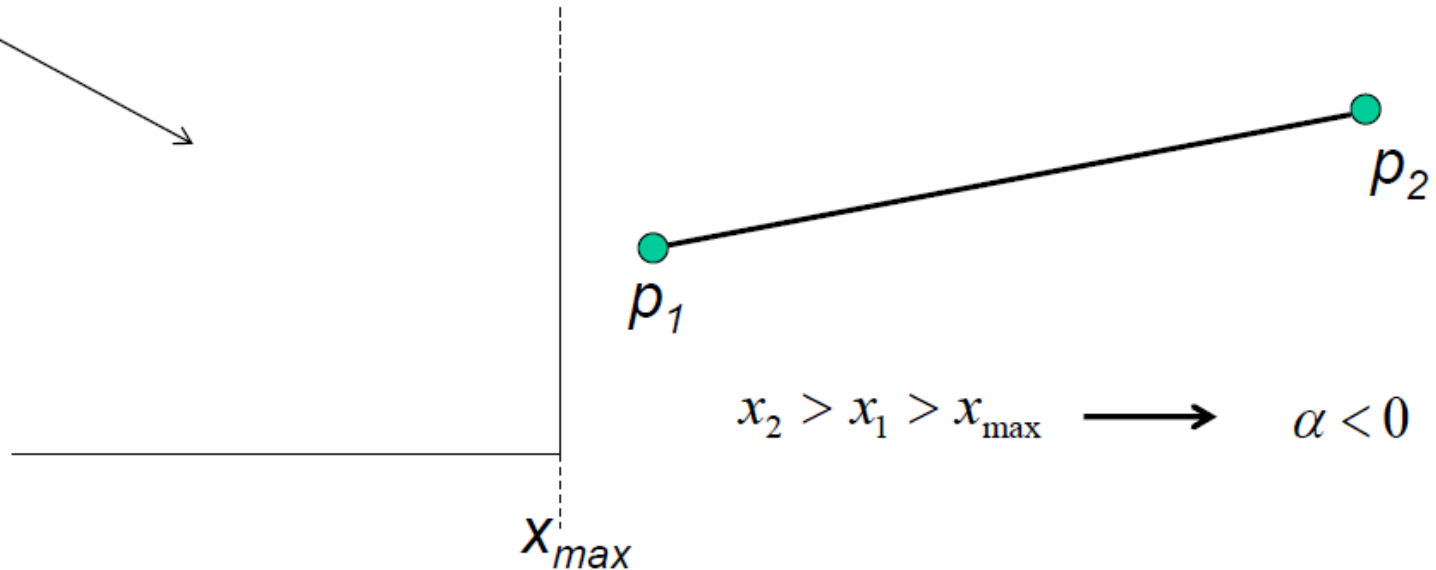
Хилийн огтлолцлыг хоёр цэгийн томъёогоор тооцоолж болно.

Хэрэв хилтэй огтлолцохгүй бол яах вэ?

- α параметр нь $[0, 1]$ утгаас гадуур байна.
- хэрэв шугам нь огтлолцсон эсэхийг шалгадаг хилтэй параллель байвал α нь ∞ байна.

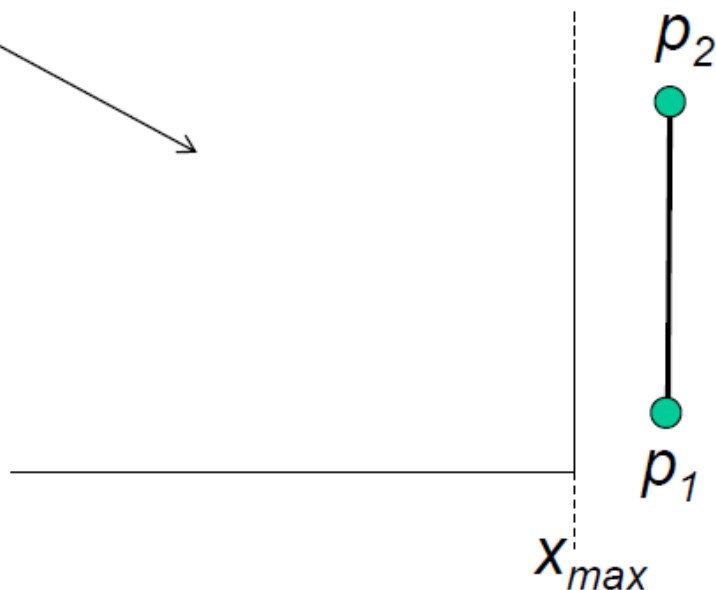
α параметр нь $[0, 1]$ утгад харьялагдаагүй
Жишээ 1:

Viewport



α параметр нь $[0, 1]$ утгад харьялагдаагүй
Жишээ 2:

Viewport



$$\alpha = \frac{x_{\max} - x_1}{x_2 - x_1}$$

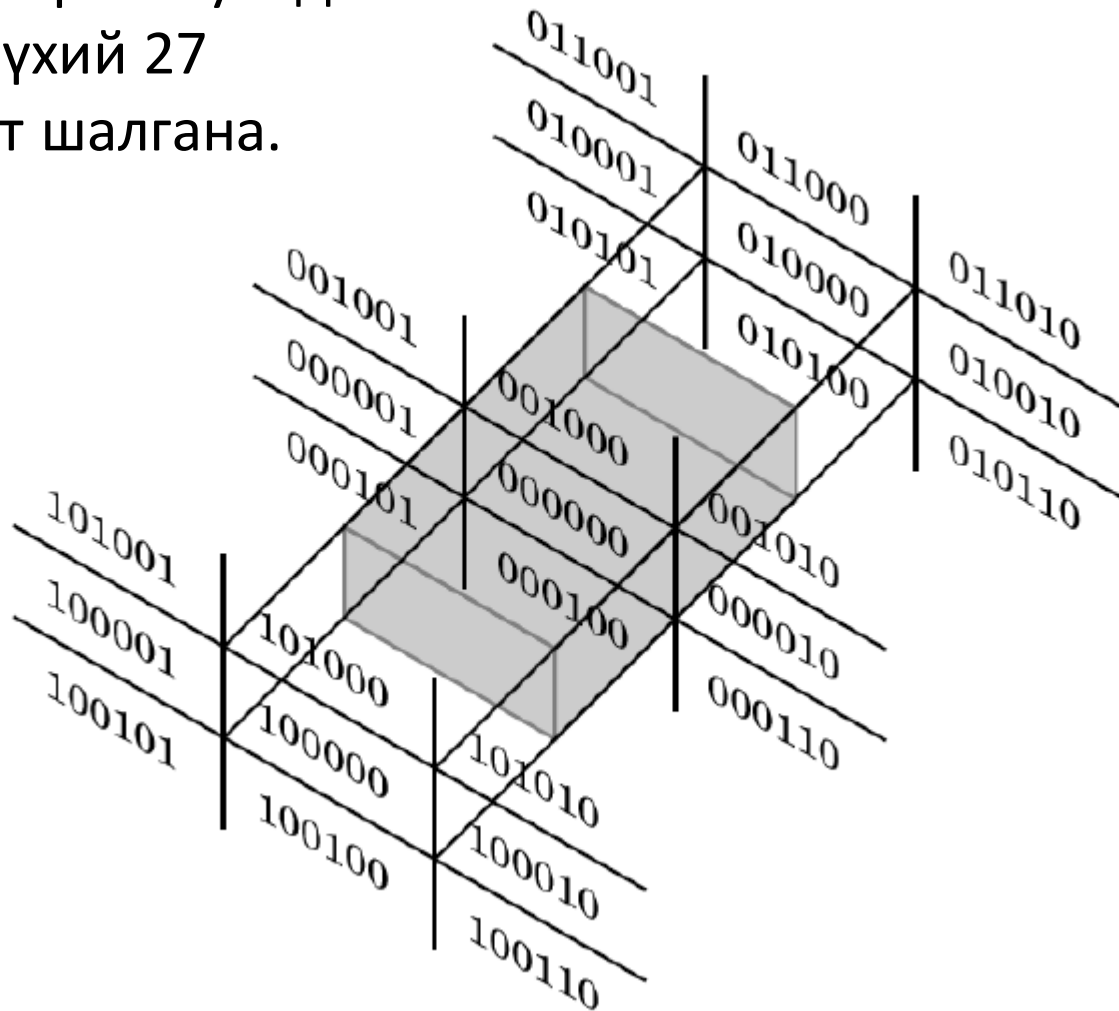
$$x_2 = x_1$$



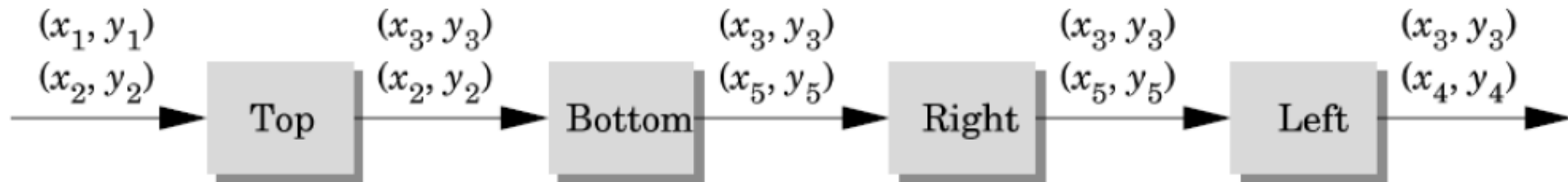
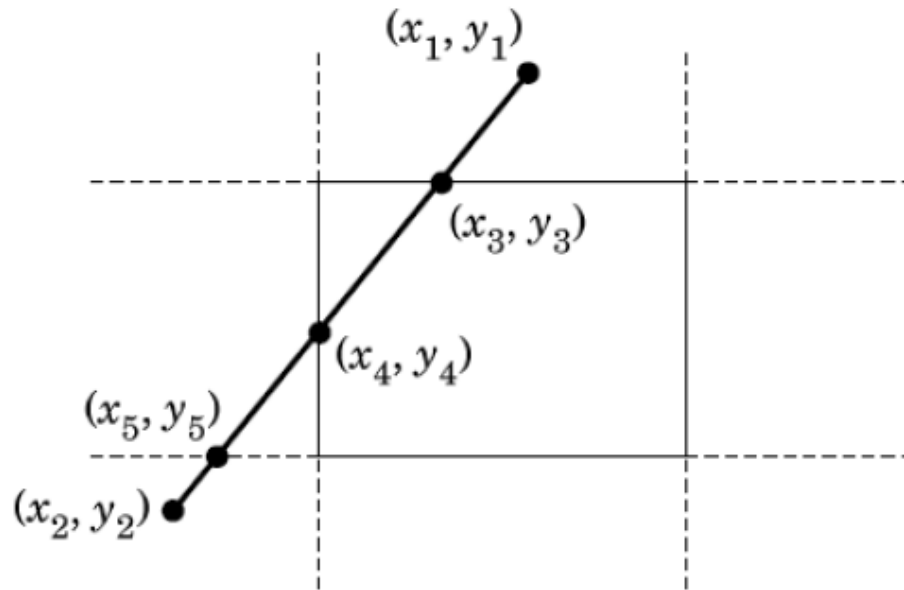
$$\alpha = \infty$$

Cohen Sutherland 3D

3D талбарын хувьд
6 бит бүхий 27
талбарт шалгана.

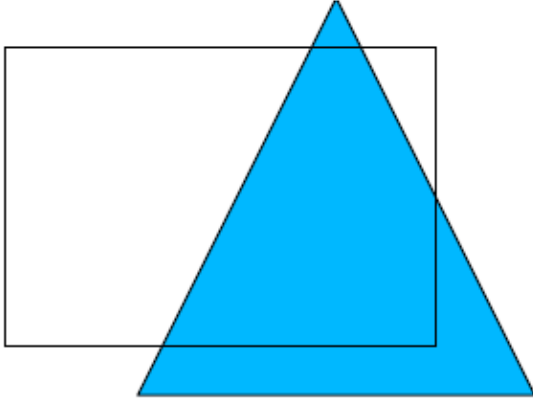


Sutherland Hodgeman

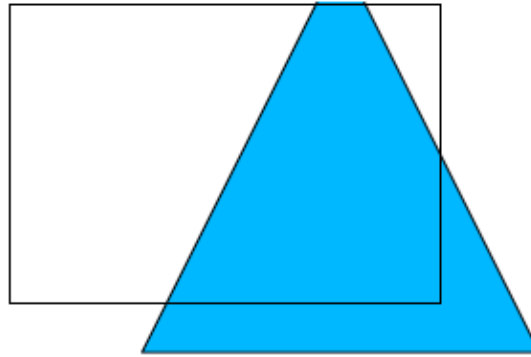


Sutherland Hodgeman

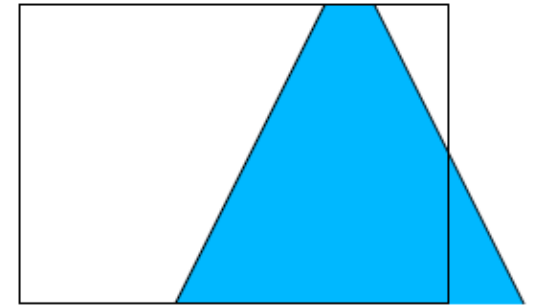
Original polygon



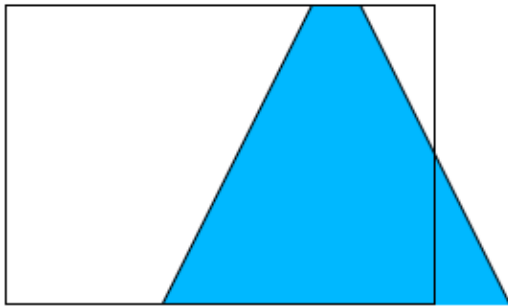
Clip top...



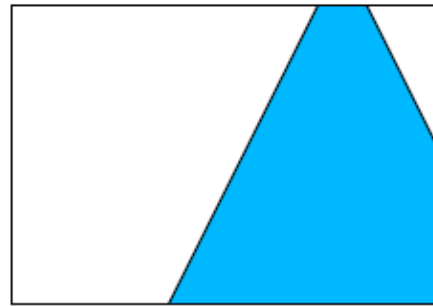
Clip bottom...



Clip left...

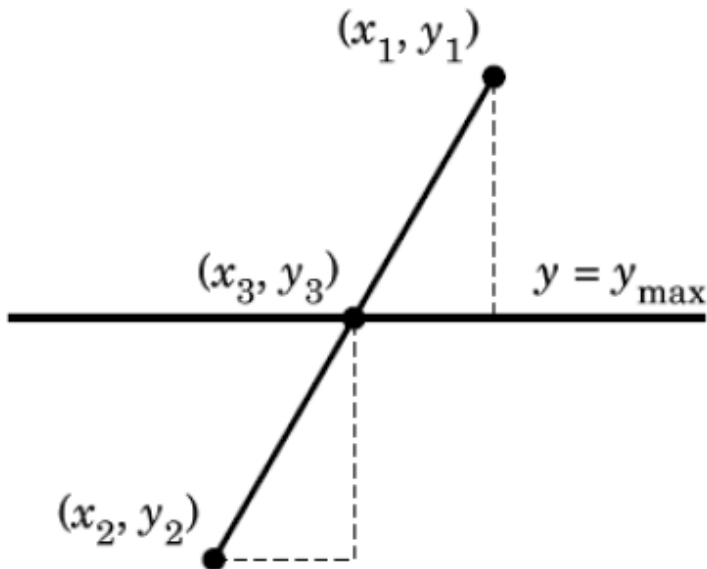


Clip right



Sutherland Hodgeman

Хоёр цэгийн томъёолол



(Top clipping)

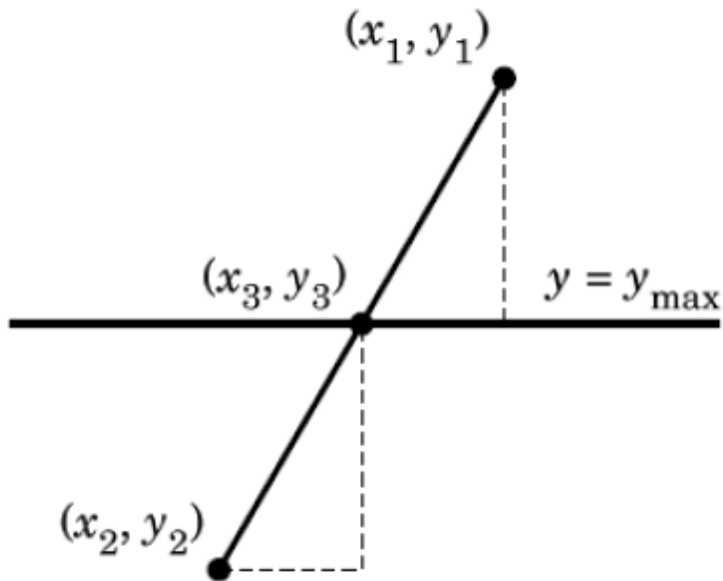
$$y_3 - y_1 = \frac{y_2 - y_1}{x_2 - x_1} (x_3 - x_1),$$

$$y_3 = y_{\max} \Rightarrow$$

$$x_3 = (y_{\max} - y_1) \frac{x_2 - x_1}{y_2 - y_1} + x_1$$

Top clipping

Хоёр цэгийн томъёог ашиглан
огтлолцлын тооцоолол хийх:



$$y_3 - y_1 = \frac{y_2 - y_1}{x_2 - x_1} (x_3 - x_1),$$

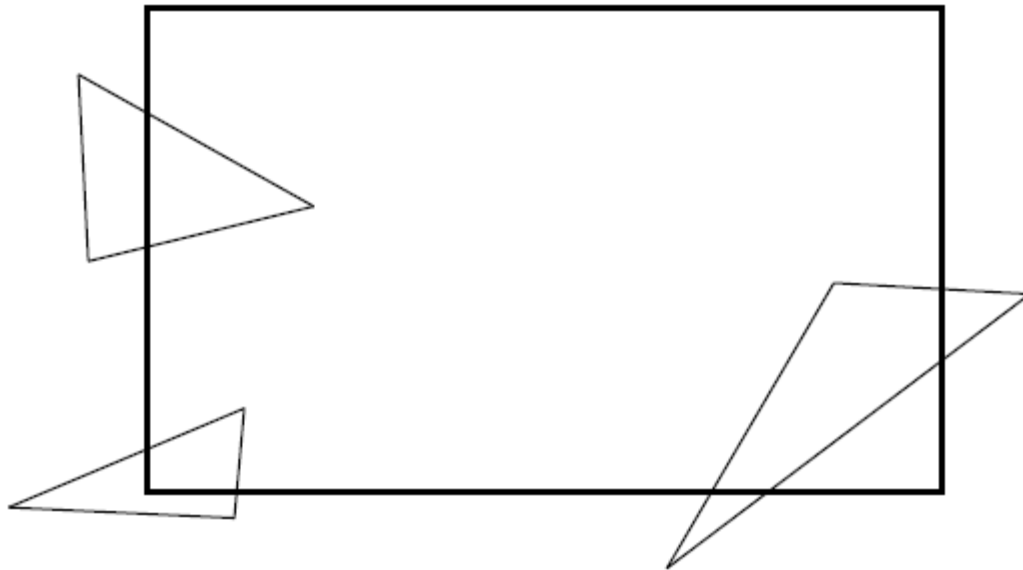
$$y_3 = y_{\max} \Rightarrow$$

$$x_3 = (y_{\max} - y_1) \frac{x_2 - x_1}{y_2 - y_1} + x_1$$

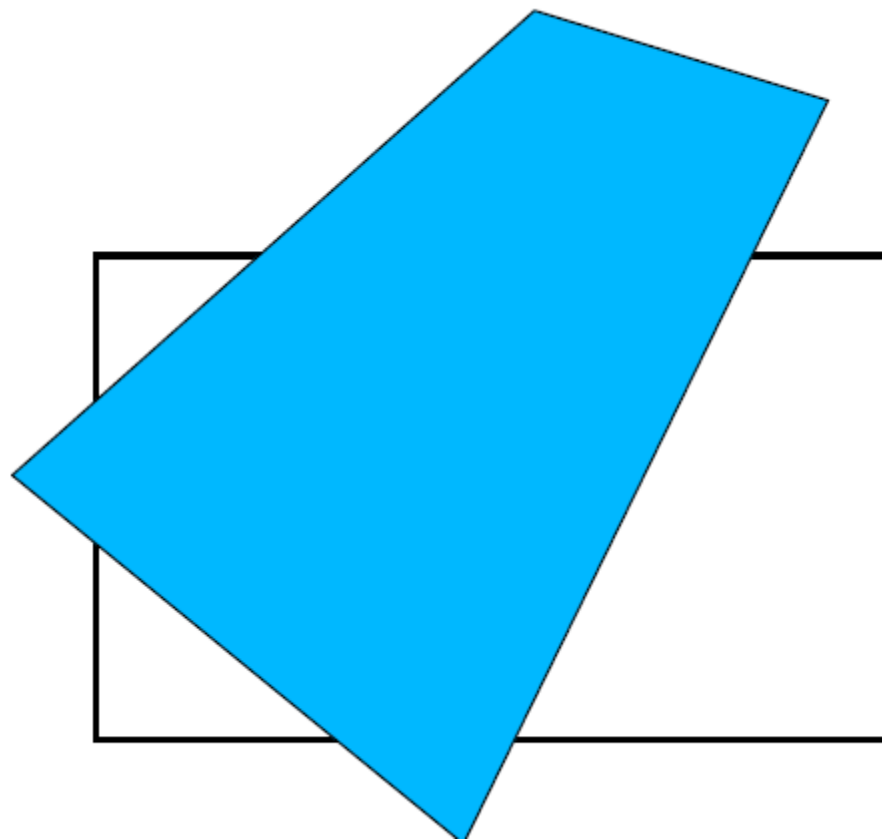
Polygon Clipping

Өмнө нь тайлбарласан аргуудын зарим өөрчлөлтөөр полигонуудыг хайчлахад ашиглаж болно.

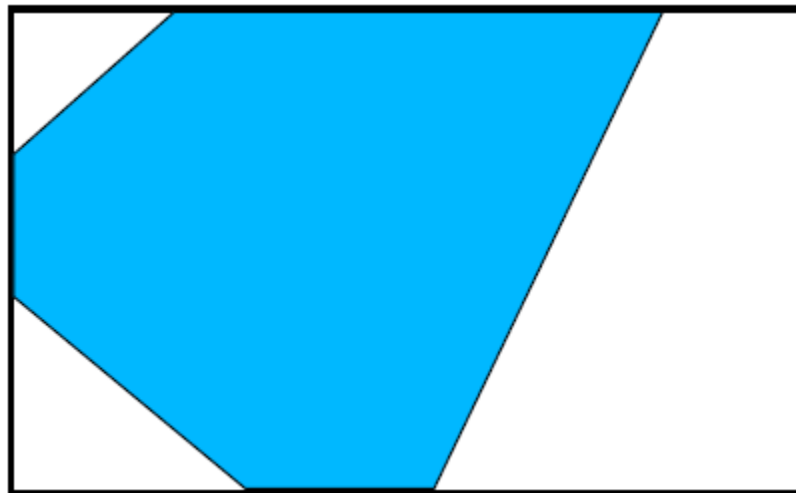
Гурвалжин/полигон хайчилсны дараа илүү олон оройтой үлдэж болохыг анхаарна уу!



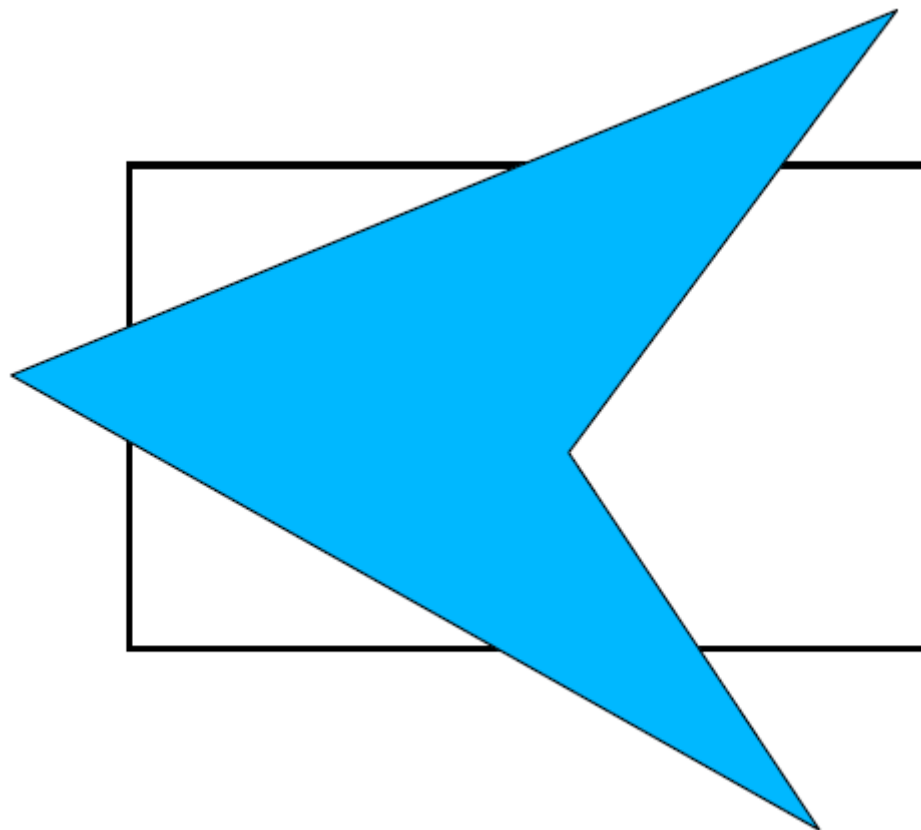
Гүдгэр олон өнцөгт...
4 орой



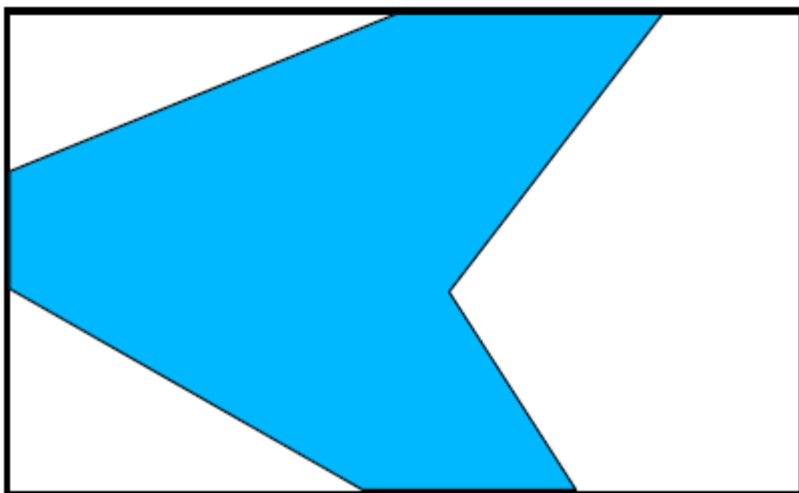
Гүдгэр олон өнцөгт...
Зүссэний дараа 6 орой.



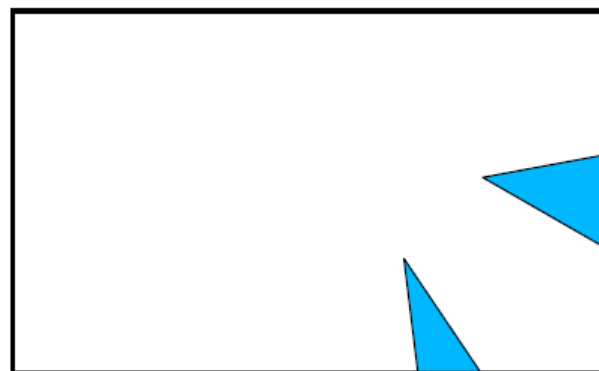
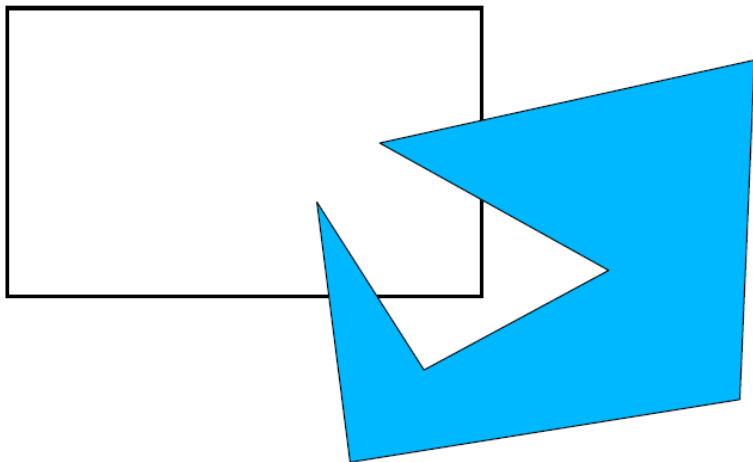
Хотгор олон өнцөгт...
Зүссэний дараа 7 орой.



Хотгор олон өнцөгт...
Зүссэний дараа 7 орой.



Хотгор полигоныг хайчилсны дараа
олон өнцөгтүүд тус тус үүсч болно!



Хурдасгах техник

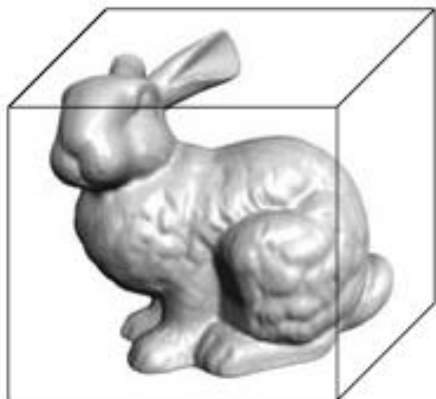


69451 ширхэг полигон



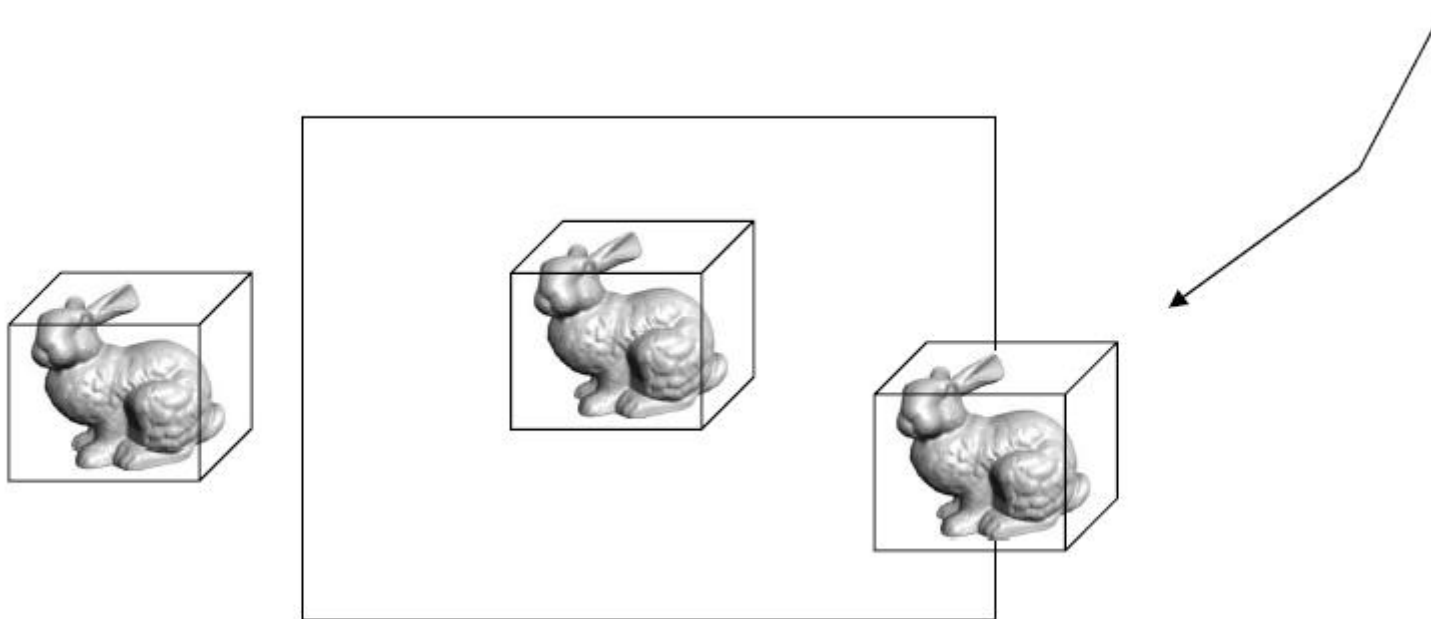
Хязгаарлах хэмжээ

- Bunny агуулах хамгийн жижиг хайрцгийг бий болгоно.
- Бид хязгаарлах хайрцгийг тэнхлэгүүдтэй зэрэгцүүлэн (илүү хялбар тооцоолол) хязгаарлаж болно, эсвэл хайрцгийг дур мэдэн чиглүүлж (ингэснээр илүү хайрцагт сайн тохирох боломжтой) болно.



Тэнхлэгүүдэд зэрэгцүүлсэн
хязгаарлах хайрцаг

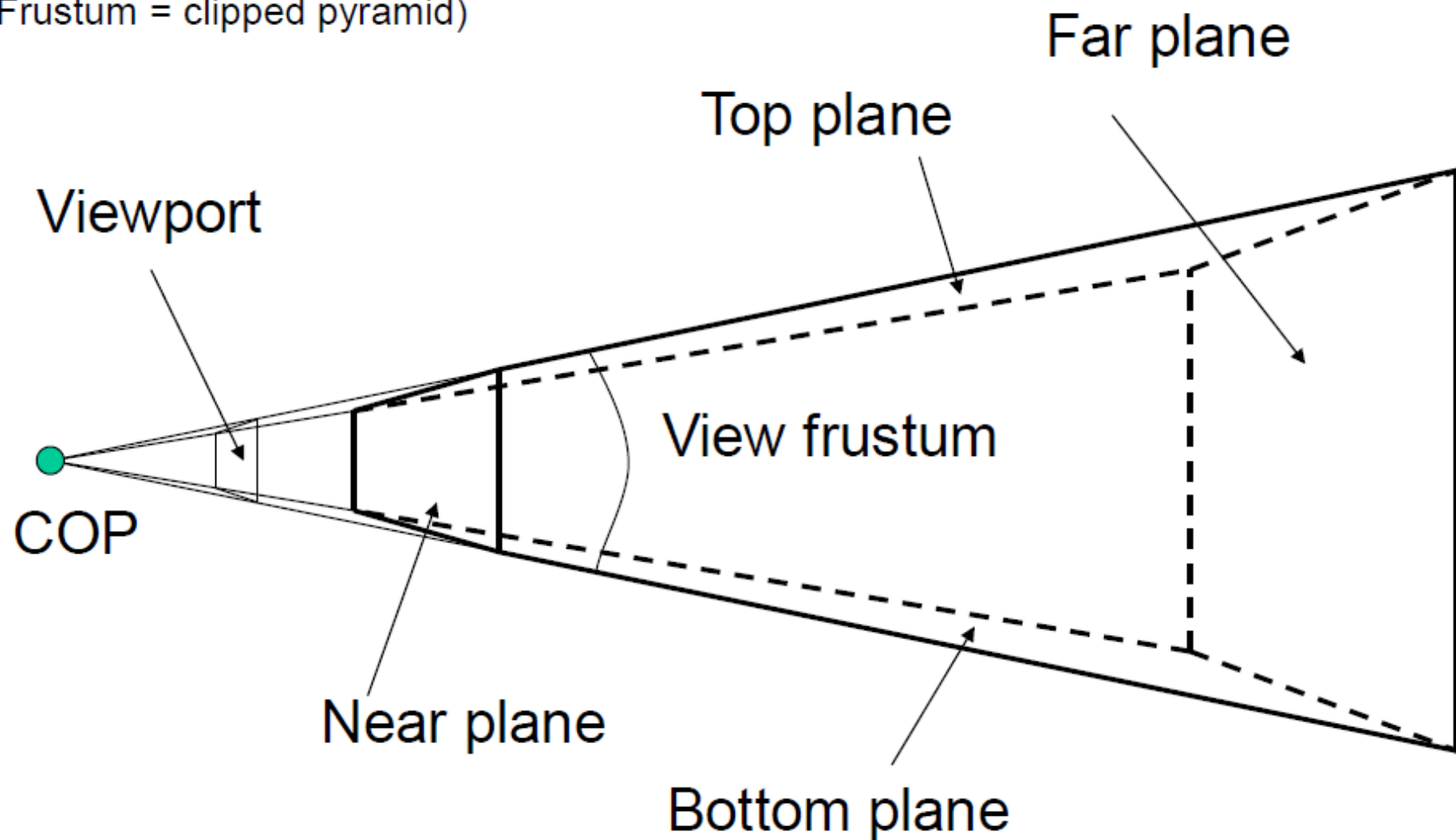
Энд өргөн хүрээний тооцоолол хийх
шаардлагатай хэвээр байна.



Frustum -ийн view

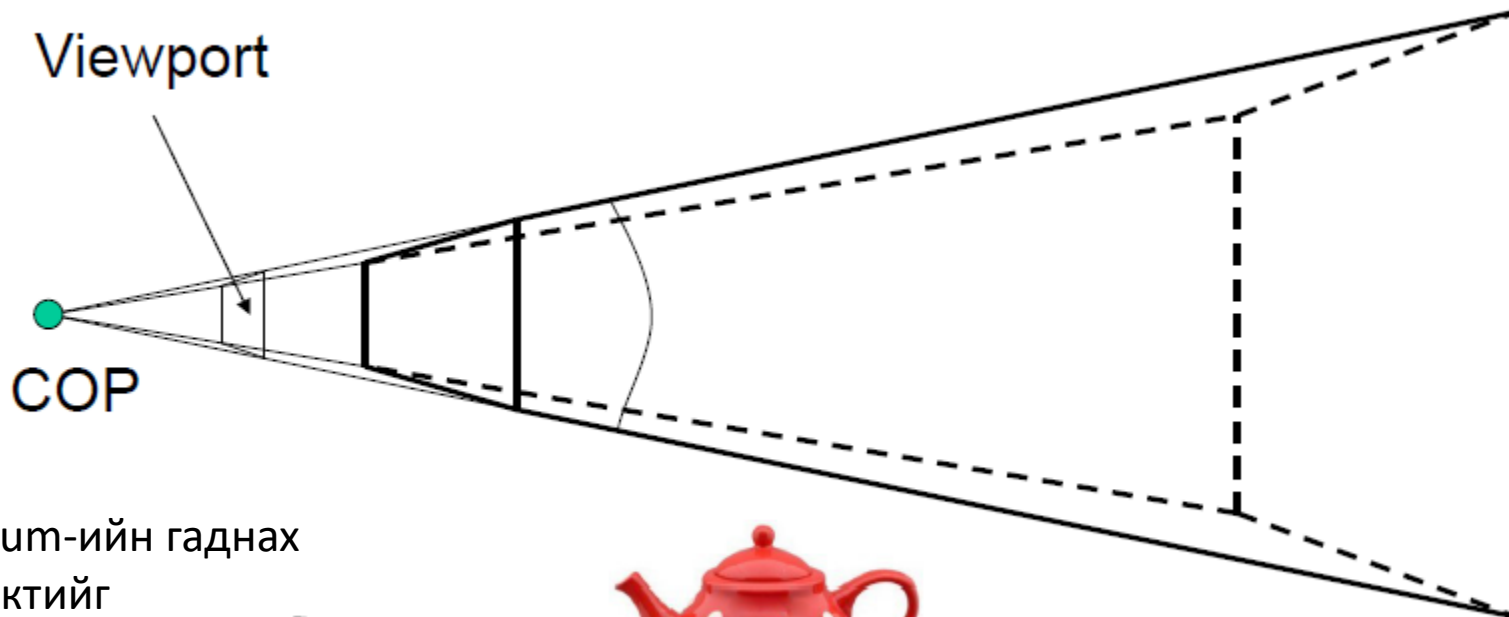
Frustum view нь 6 хэсгээс бүрдэнэ.

(Frustum = clipped pyramid)

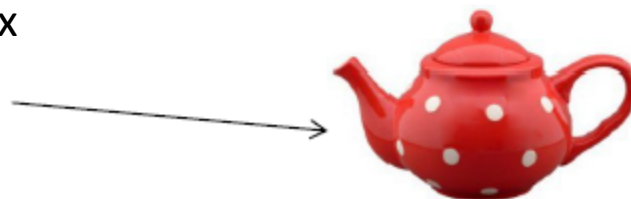


Frustum -ийн view

Frustum view нь 6 хэсгээс бүрдэнэ.

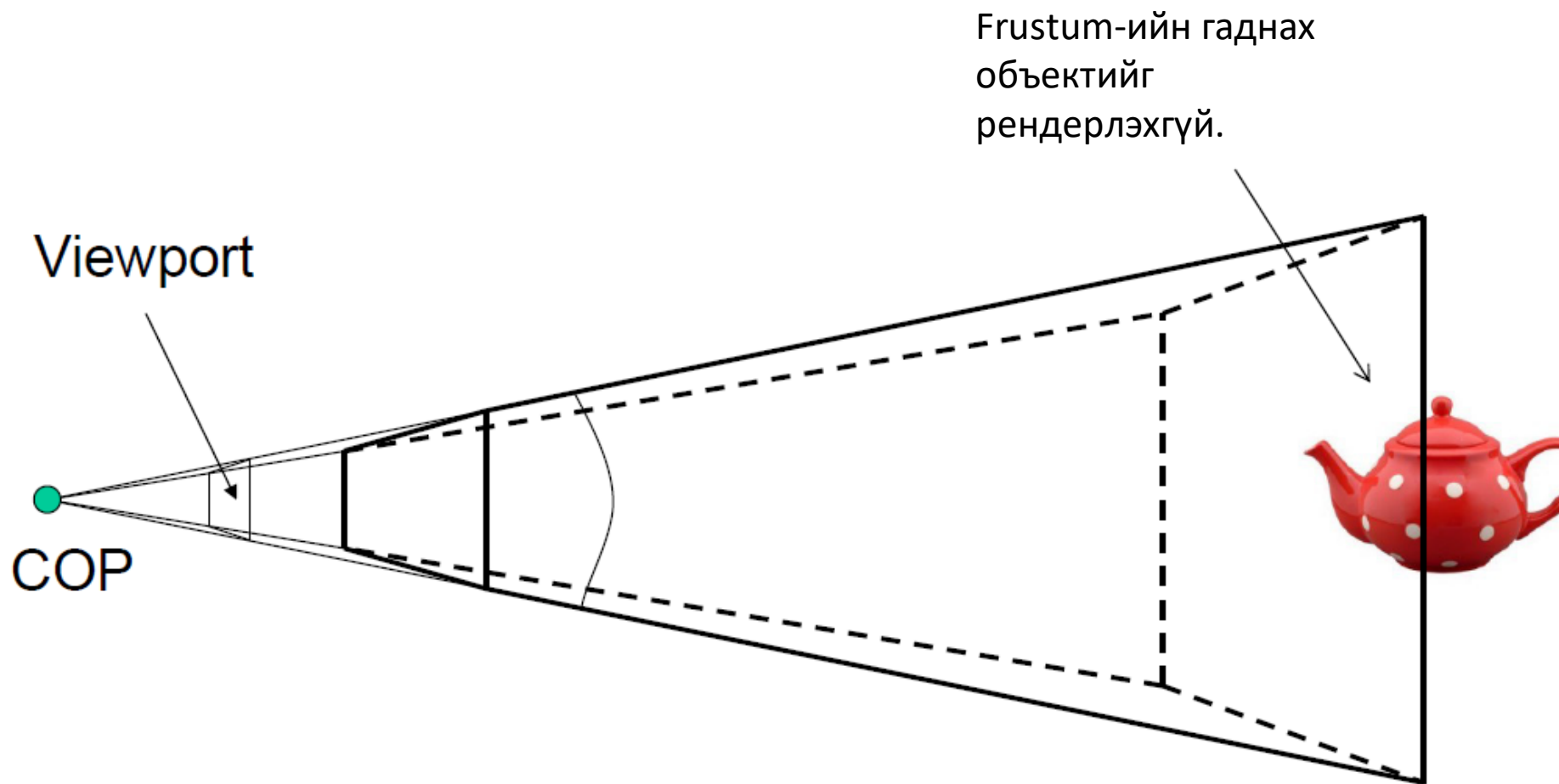


Frustum-ийн гаднах
объектийг
рендерлэхгүй.



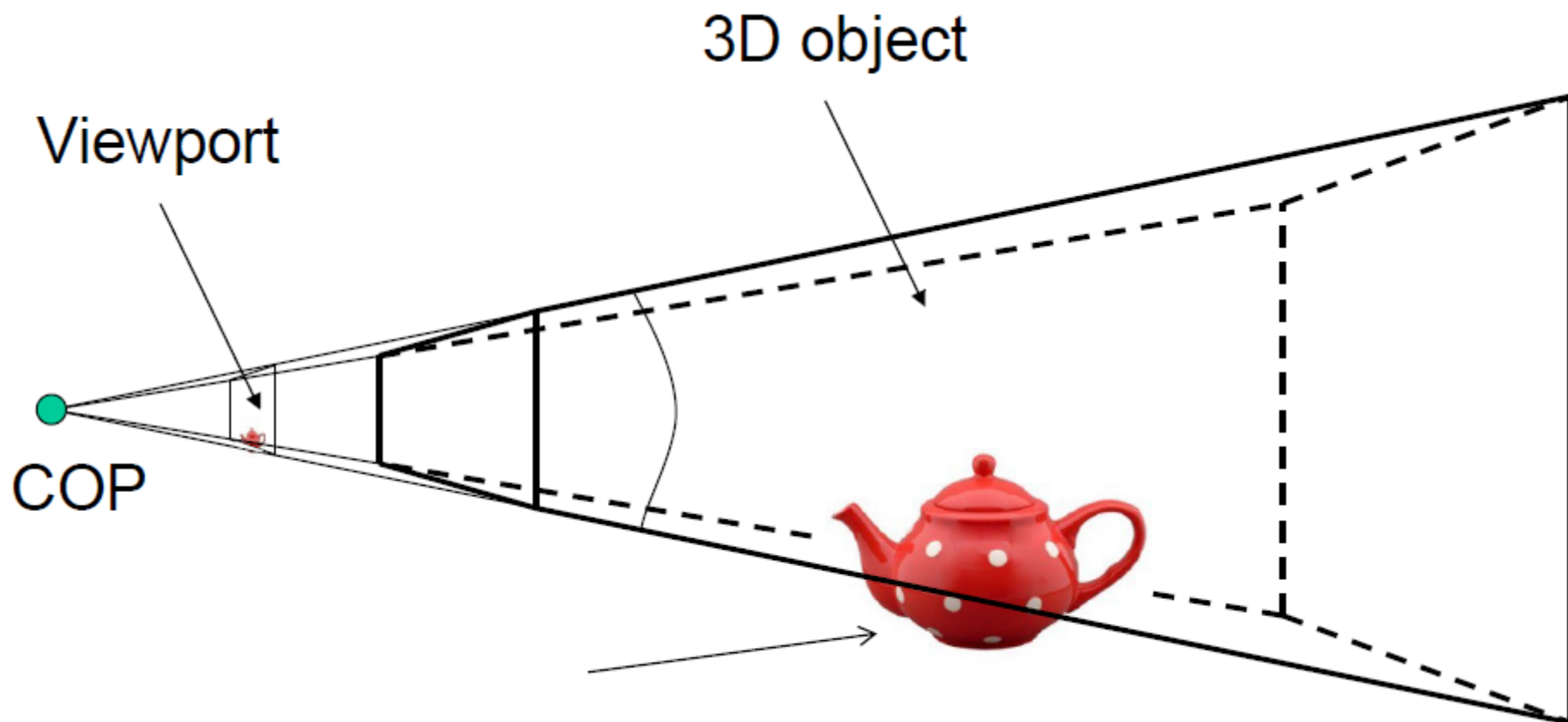
Frustum -ийн view

Frustum view нь 6 хэсгээс бүрдэнэ.



Frustum -ийн view

Frustum view нь 6 хэсгээс бүрдэнэ.



Frustum-ийн гаднах объектийн зарим хэсэг байгаа тул clipping хэрэгтэй.

View port-oor clipping хайчилж авсан
3D объект.

Viewport



Clipped 3D object

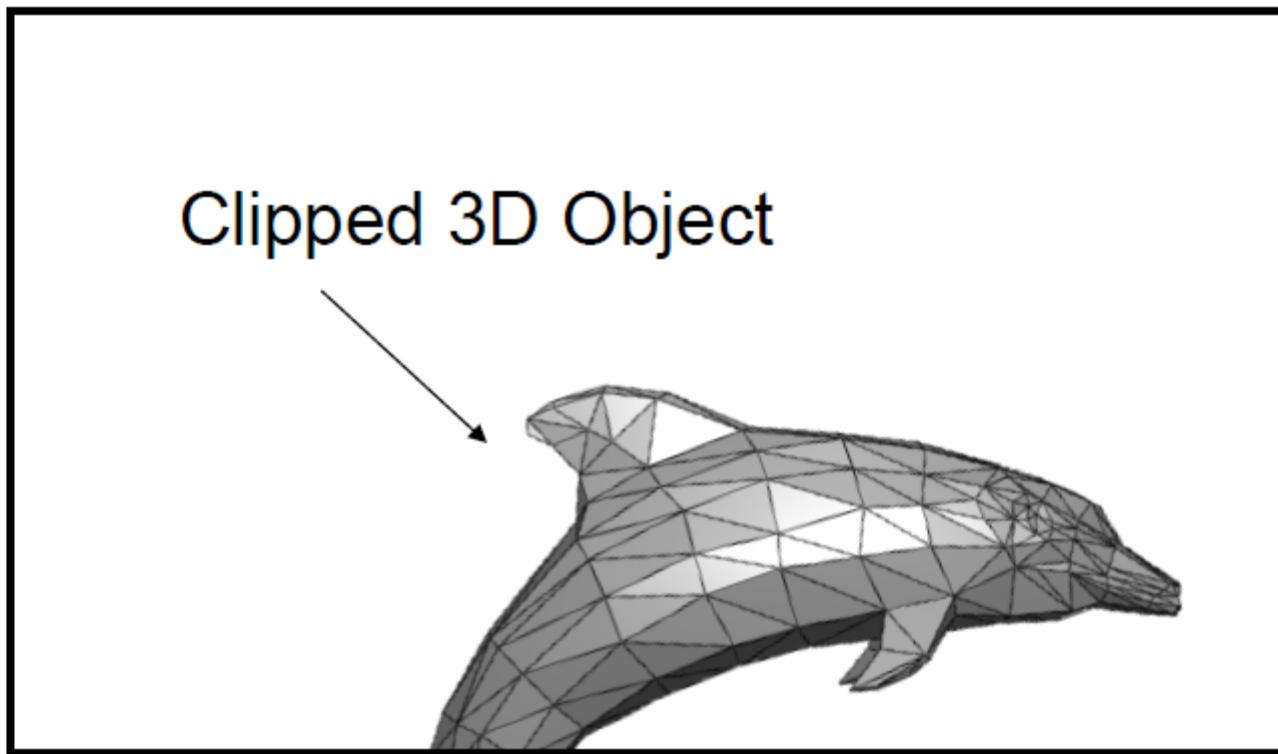


2d, 3d объектууд нь бүгд полигуунаас бүрддэгийг санаарай.

Viewport



Clipped 3D Object

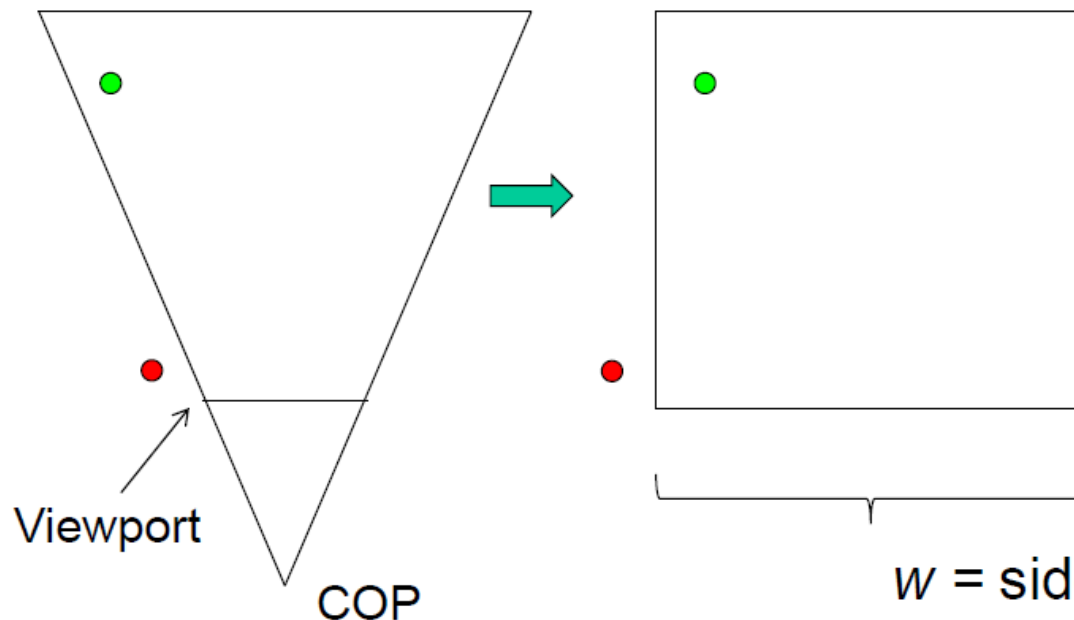
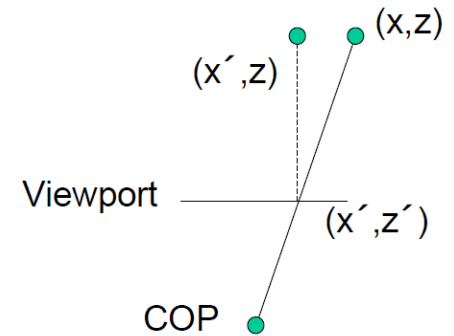


Normalization

Бид frustum -ийг шоо болгон харуулдаг (normalization хийсэн координатууд)

Шоог хайчлах нь илүү хялбар болно!

Цэгүүдийн x координатыг ажиглаарай! COP



Range

$$-w/2 \leq x \leq w/2$$

$$-w/2 \leq y \leq w/2$$

$$-w/2 \leq z \leq w/2$$

$w = \text{side of the cube}$

OpenGL

```
gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble zNear,  
GLdouble zFar);
```

```
glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top,  
GLdouble zNear, GLdouble zFar);
```

