

Posicionament de la Càmera: opció Moure Objectes

Moviment de l'objecte (una possibilitat):

- 1) $\text{Tr}(-\text{VRP})$
- 2) $\text{Gy}(-90)$
- 3) $\text{Tr}(0,0,-d)$

$$\text{TC} = \text{T}(0,0,-d) \text{G}_Y(-90) \text{T}(-\text{VRP})$$

(angles positius => girs anti-horari)

La podem calcular amb OpenGL

```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glTranslatef(0,0,-d);
glRotate3f(-90,0,1,0);
glTranslatef(-VRP.x,-VRP.y,-VRP.z);
```

$P_O = \text{TC} P_A$

GL_MODELVIEW

IDI 2014-2015 1Q 7

gluLookAt(...) → càlcul TC

$$\text{TC} = \begin{bmatrix} s.x & s.y & s.z & 0 \\ w.x & w.y & w.z & 0 \\ F.x & F.y & F.z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \text{Trans}(-\text{OBS})$$

$\mathbf{F} = \text{OBS} - \text{VRP} = (F.x, F.y, F.z) \quad \mathbf{F} = \mathbf{F} / \|\mathbf{F}\|$

$\mathbf{s} = \mathbf{up} \times \mathbf{F} \quad \mathbf{s} = \mathbf{s} / \|\mathbf{s}\|$

$\mathbf{w} = \mathbf{F} \times \mathbf{s}$

OBS = Observador

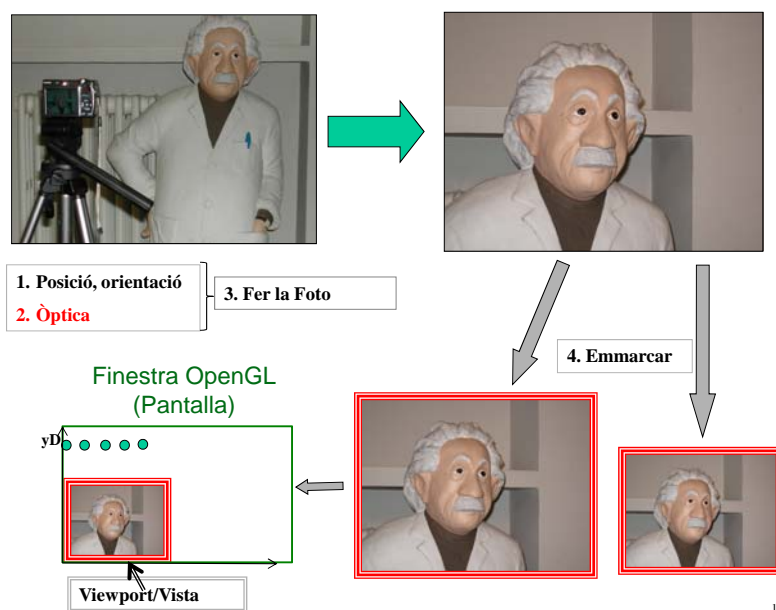
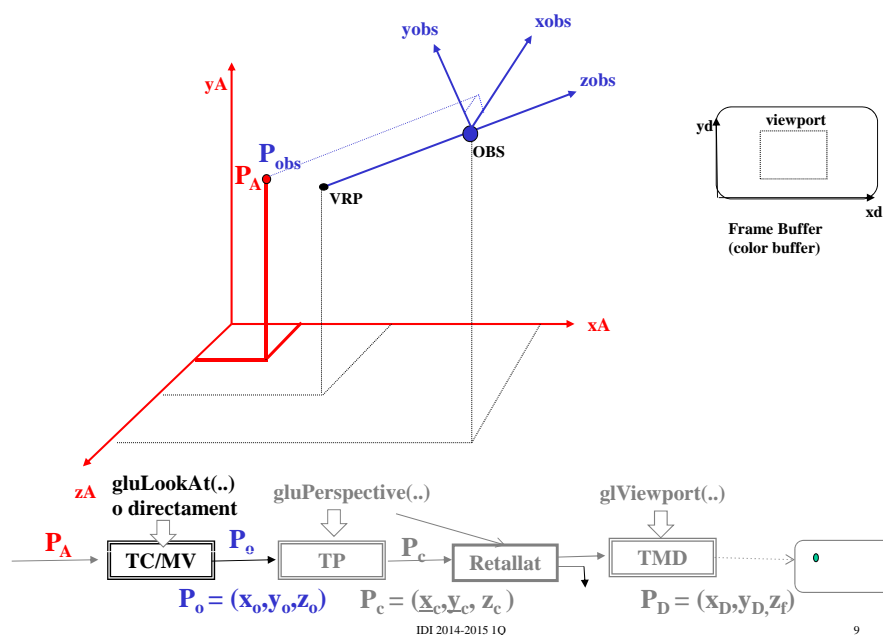
VRP = View Reference Point

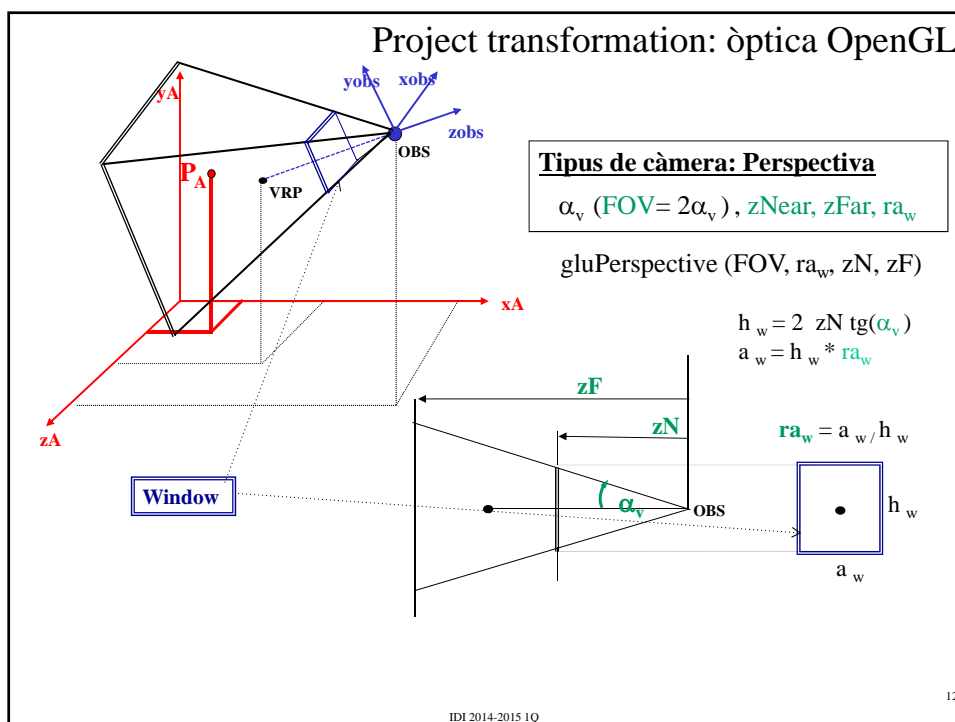
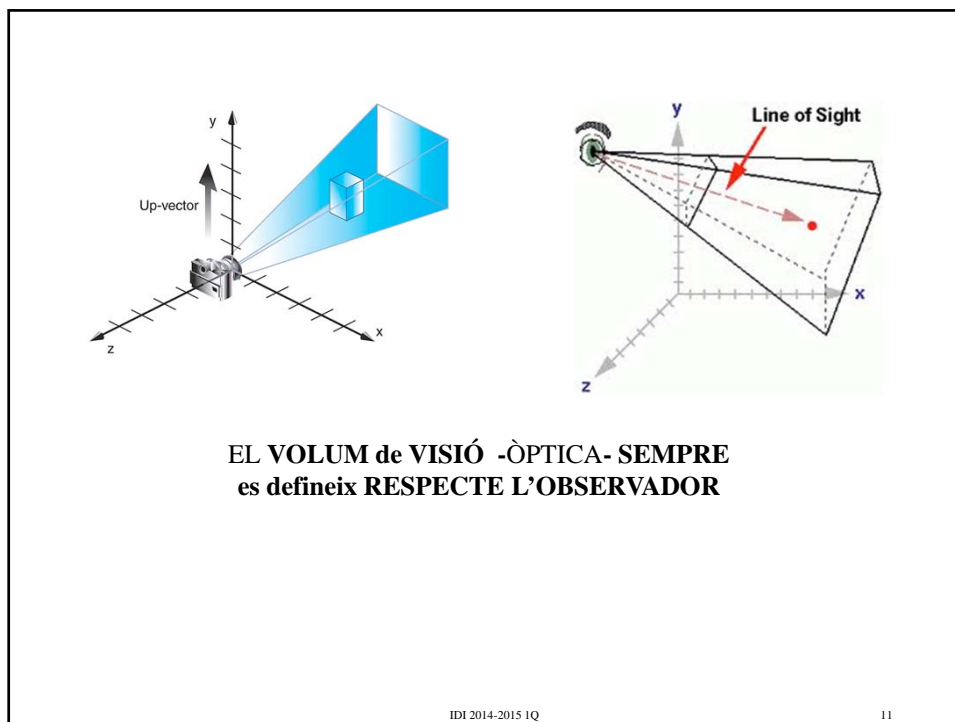
up = View Up Vector

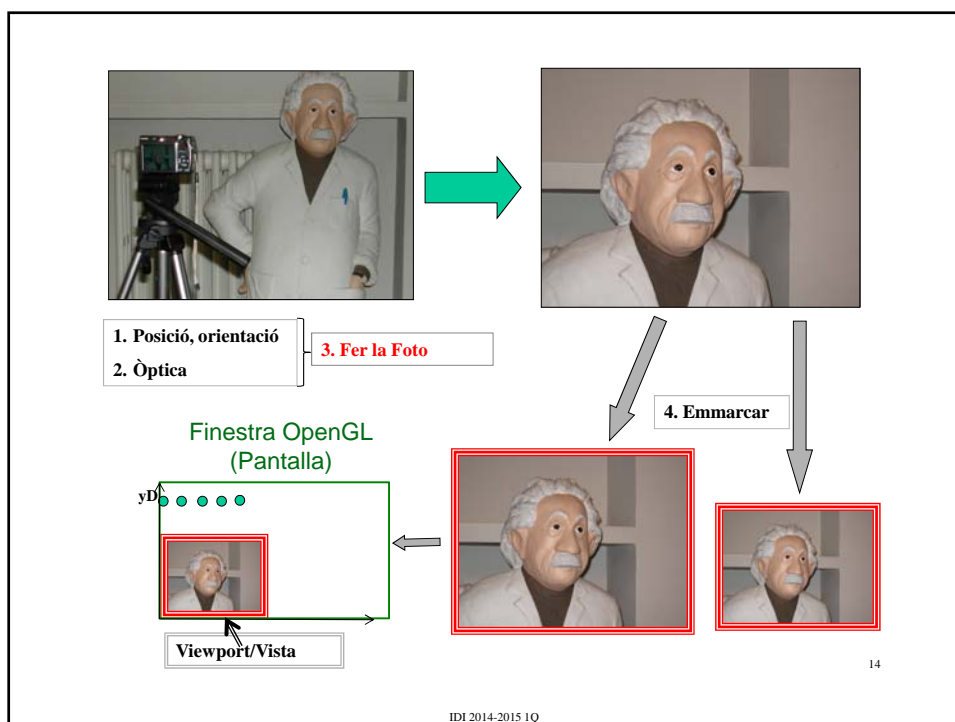
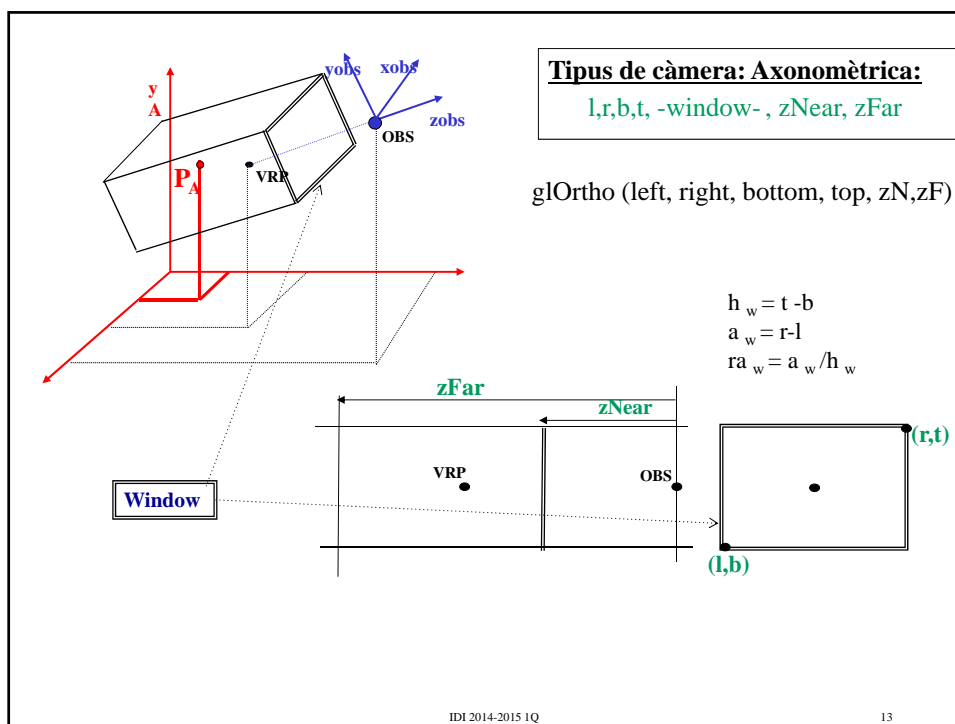
up "indica" la direcció de l'eix vertical de la Càmera (inclinació)

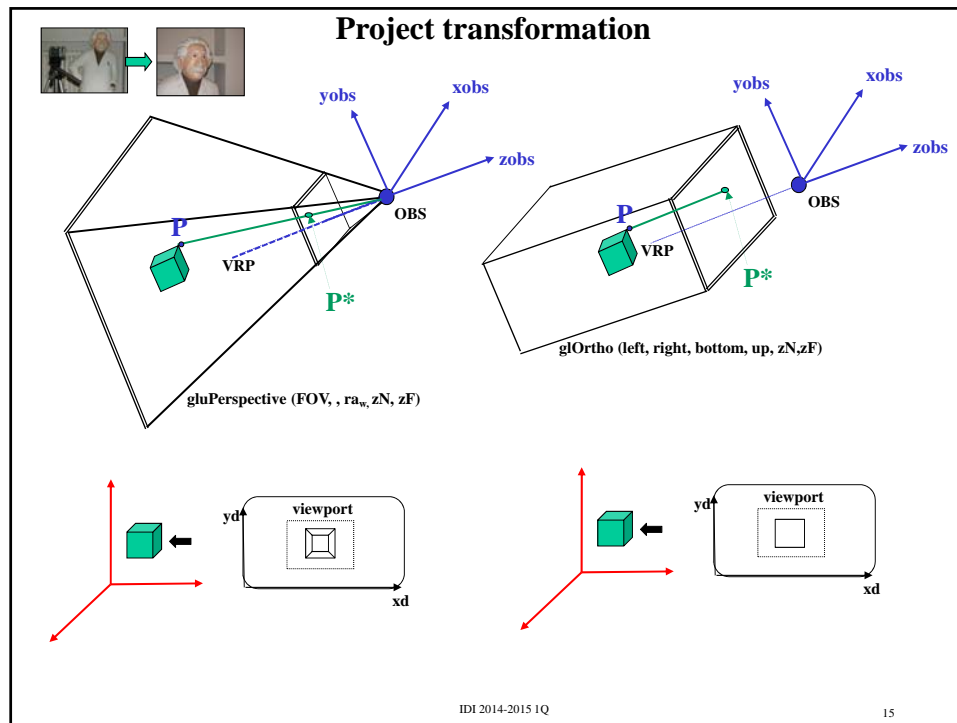
IDI 2014-2015 1Q 8

Procès Visualització OpenGL (1)

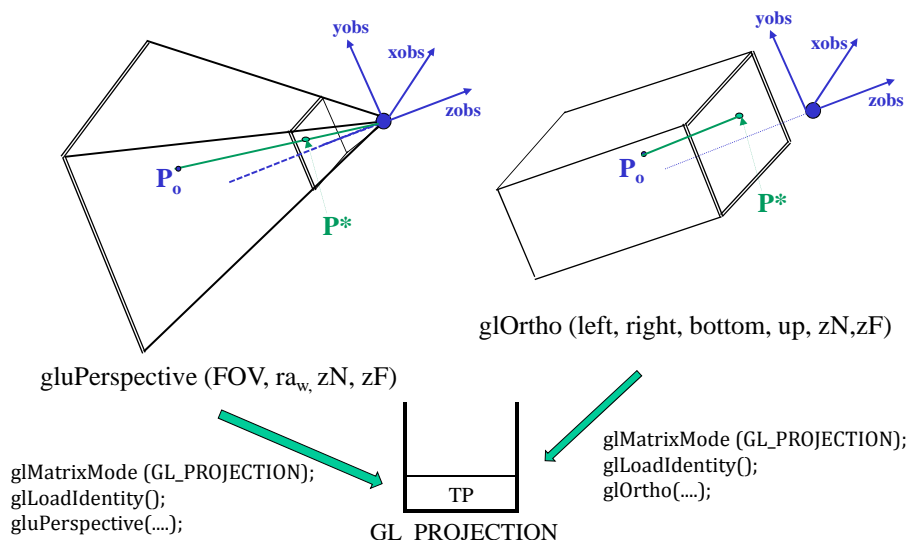








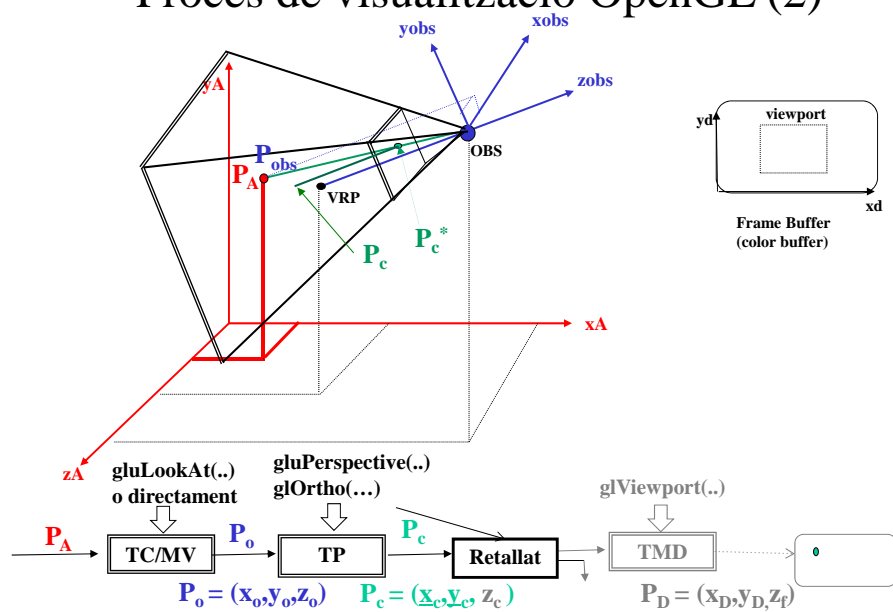
Project transformations: càmera OpenGL



IDI 2014-2015 1Q

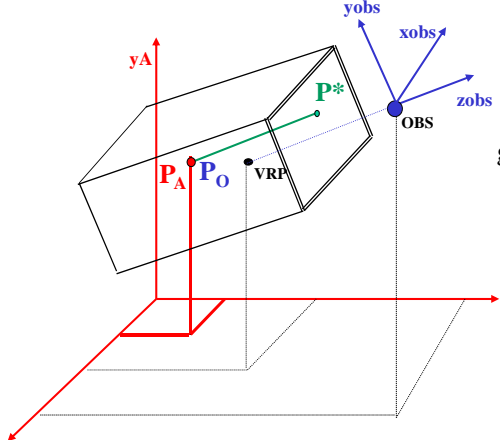
17

Procés de visualització OpenGL (2)



IDI 2014-2015 1Q

18



Tipus de càmera: Axonèmtrica:
left,right,bottom, top, zNear, zFar

glOrtho (left, right, bottom, top, zN,zF)

TP

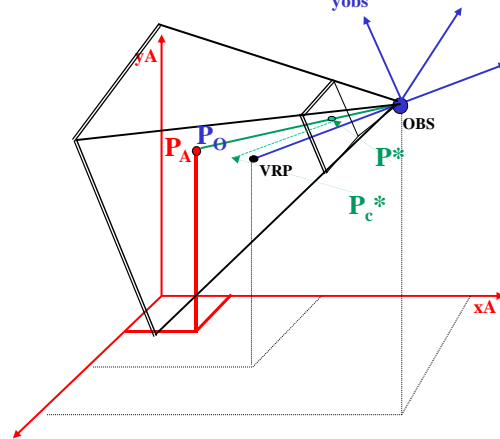
GL_PROJECTION

$P_c^* = TP P_O$

$$TP_o = \begin{pmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & d \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$a=2/(r-l) \quad b=2/(t-b)$
 $c=2/(f-n)$
 $d=(n+f)/(f-n)$

IDI 2014-2015 1Q 19



Tipus de càmera: Perspectiva:
 α_v (FOV= $2\alpha_v$), zNear, zFar, ra_w

gluPerspective (FOV, ra_w, zN, zF)

TP

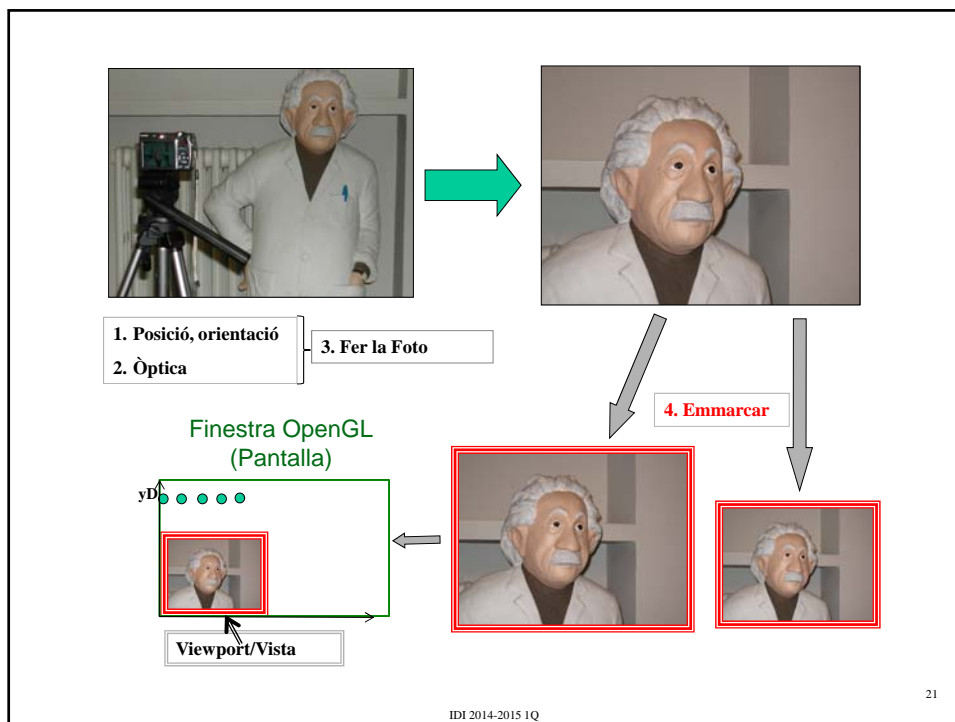
GL_Projection

$P_c^* = TP P_O$

$$TP = \begin{pmatrix} 1/ra^*a & 0 & 0 & 0 \\ 0 & 1/a & 0 & 0 \\ 0 & 0 & c & d \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

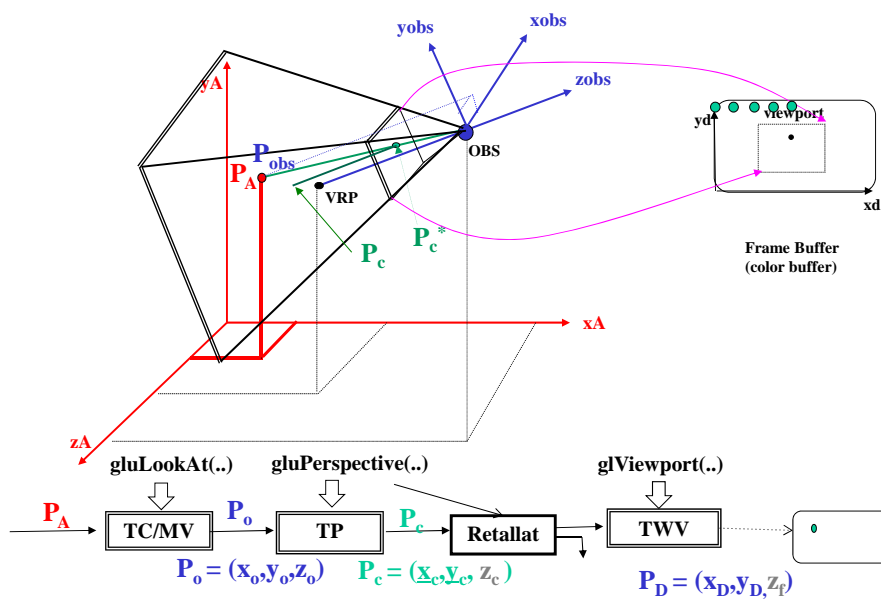
$a = \tan(FOV/2)$
 $c = (f+n)/(n-f)$
 $d = 2nf/(n-f)$

IDI 2014-2015 1Q 20



21

Procés de visualització OpenGL (3)



IDI 2014-2015 1Q

22

Solutions:

- $ra_v = ra_w$ (Bloc 1 de Lab)
- $ra_w = ra_v$ (Bloc 3 de Lab)

IDI 2014-2015 1Q 23

Project transformations: càmera OpenGL

$gluPerspective(FOV, ra_w, zN, zF)$

$ra_w = a_w / h_w$

$h_w = 2 \cdot zN \cdot \tan(\alpha_v)$

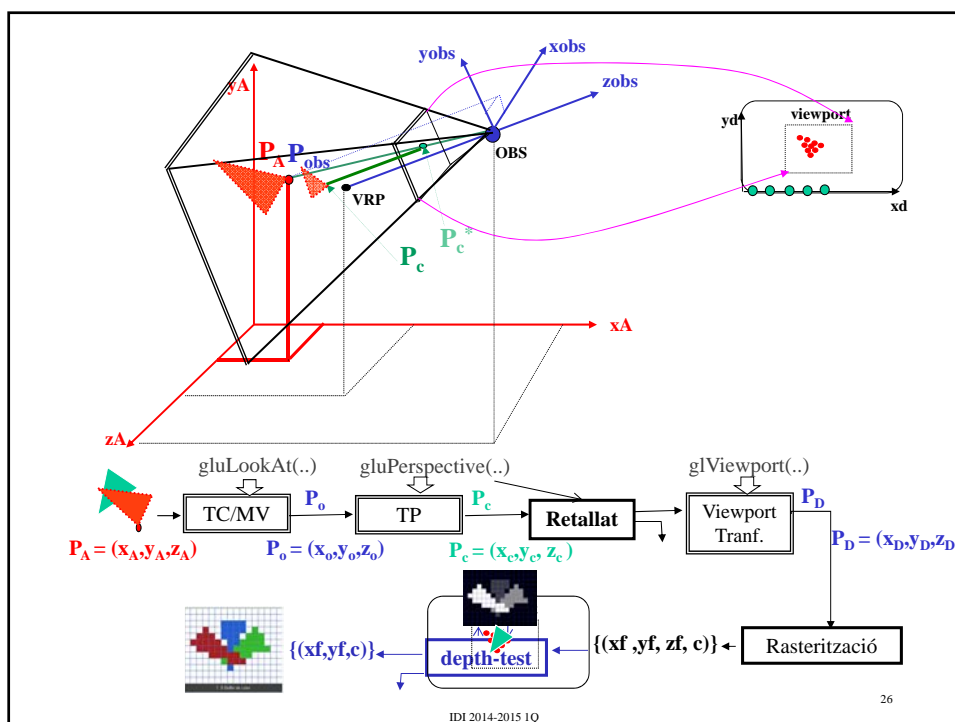
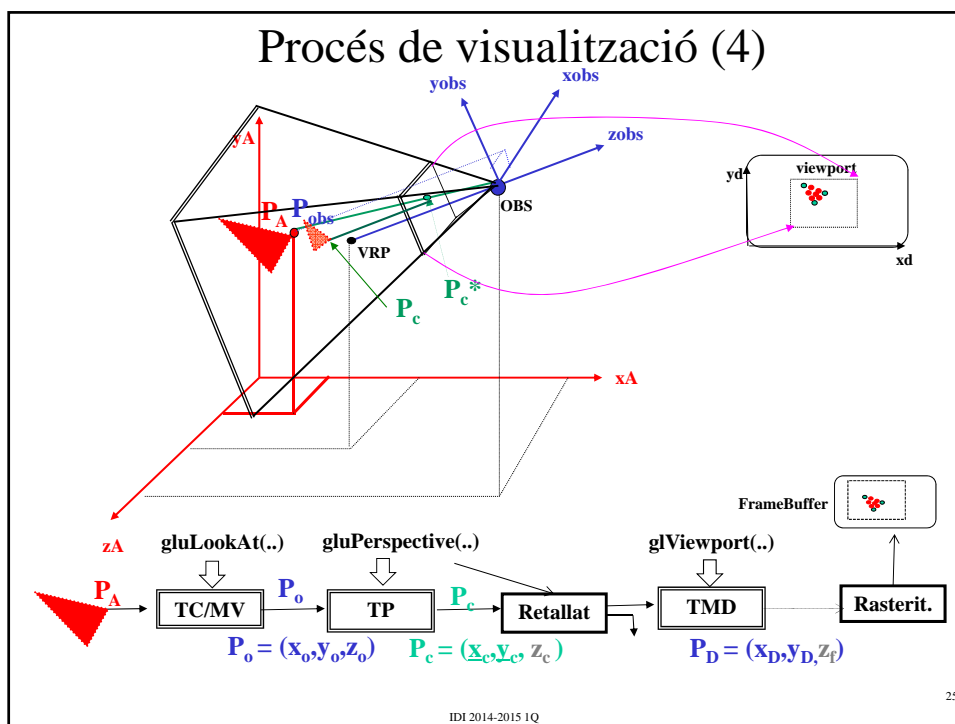
$a_w = h_w \cdot ra_w$

$glOrtho(left, right, bottom, up, zN, zF)$

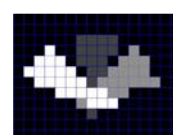
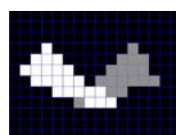
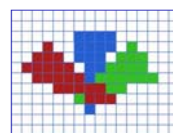
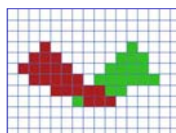
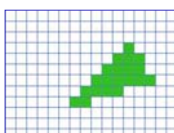
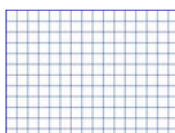
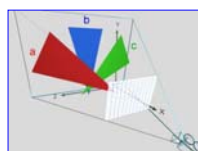
TP

GL_PROJECTION

IDI 2014-2015 1Q 24



Depth Buffer (z-buffer)



IDI 2014-2015 1Q

27

Depth Buffer (z-buffer)

- Dos buffers de la mateixa resolució que la pantalla

Buffer color (frame_buffer)

$(r, g, b) \in [0, 2^n - 1]$

Buffer profunditats (depth_buffer)

$z \in [0, 2^{nz} - 1]$

- Inicialitzar al color de fons



- Inicialitzar al més lluny possible



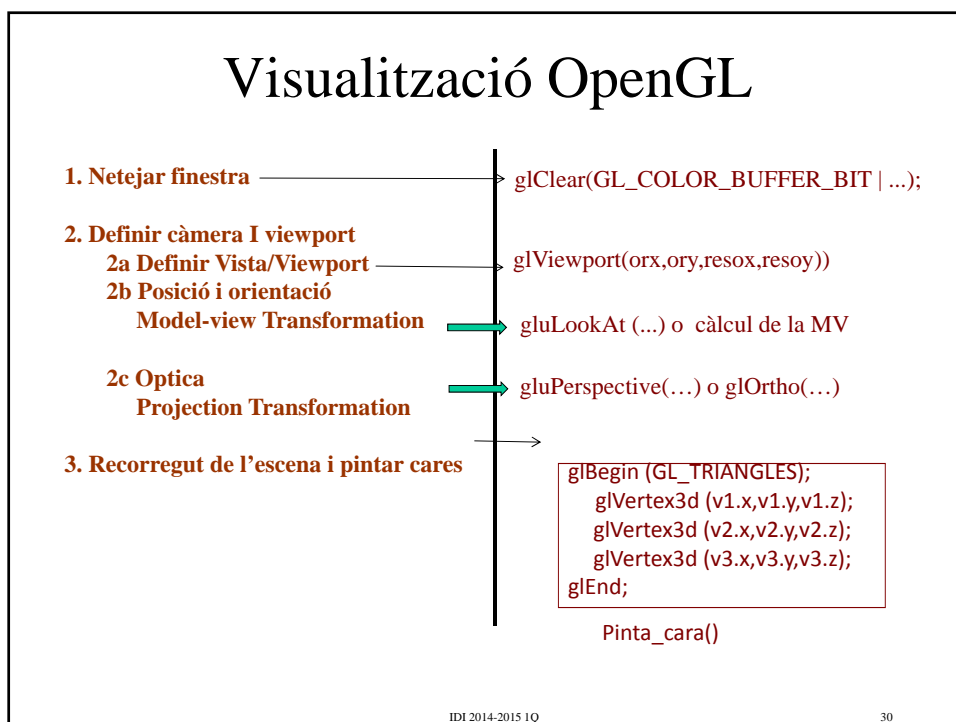
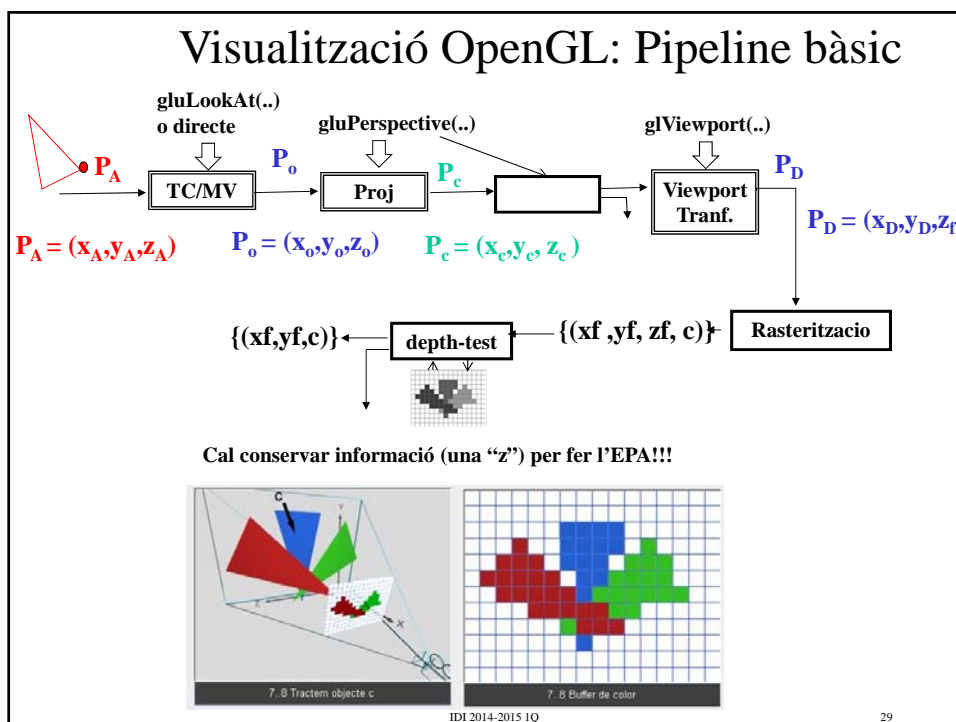
```

si zf < depth_buffer(xf,yf) llavors
    depth_buffer(xf,yf) := zf
    color_buffer(xf,yf) := c
fisi

```

IDI 2014-2015 1Q

28



Per pensar...

Quins podrien ser uns **paràmetres de posició, orientació i òptica** per a una càmera que, donada una escena amb caps mínima contenidora

$(x_{\min}, y_{\min}, z_{\min}) - (x_{\max}, y_{\max}, z_{\max})$,

visualitzi una imatge que inclogui totalment l'escena, ocupant el màxim de la vista (viewport) i sense deformació?

IDI 2014-2015 1Q

31

IDI 2014-2015 1Q

32