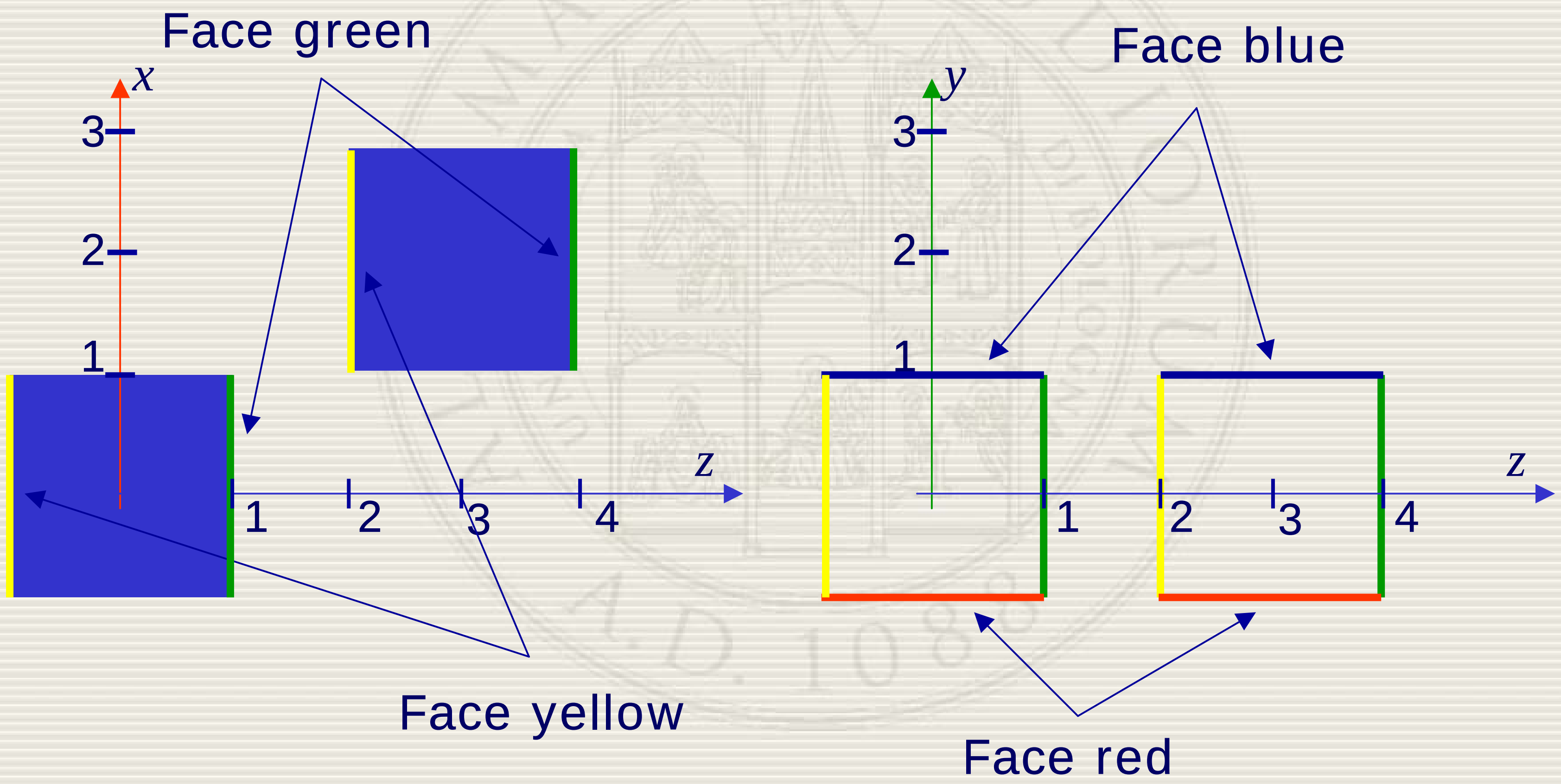
Proiezione Prospettica con Profondità in OpenGL

(opengl_1516/glstart/glutsimple_mat/glutsimple_mult.c)

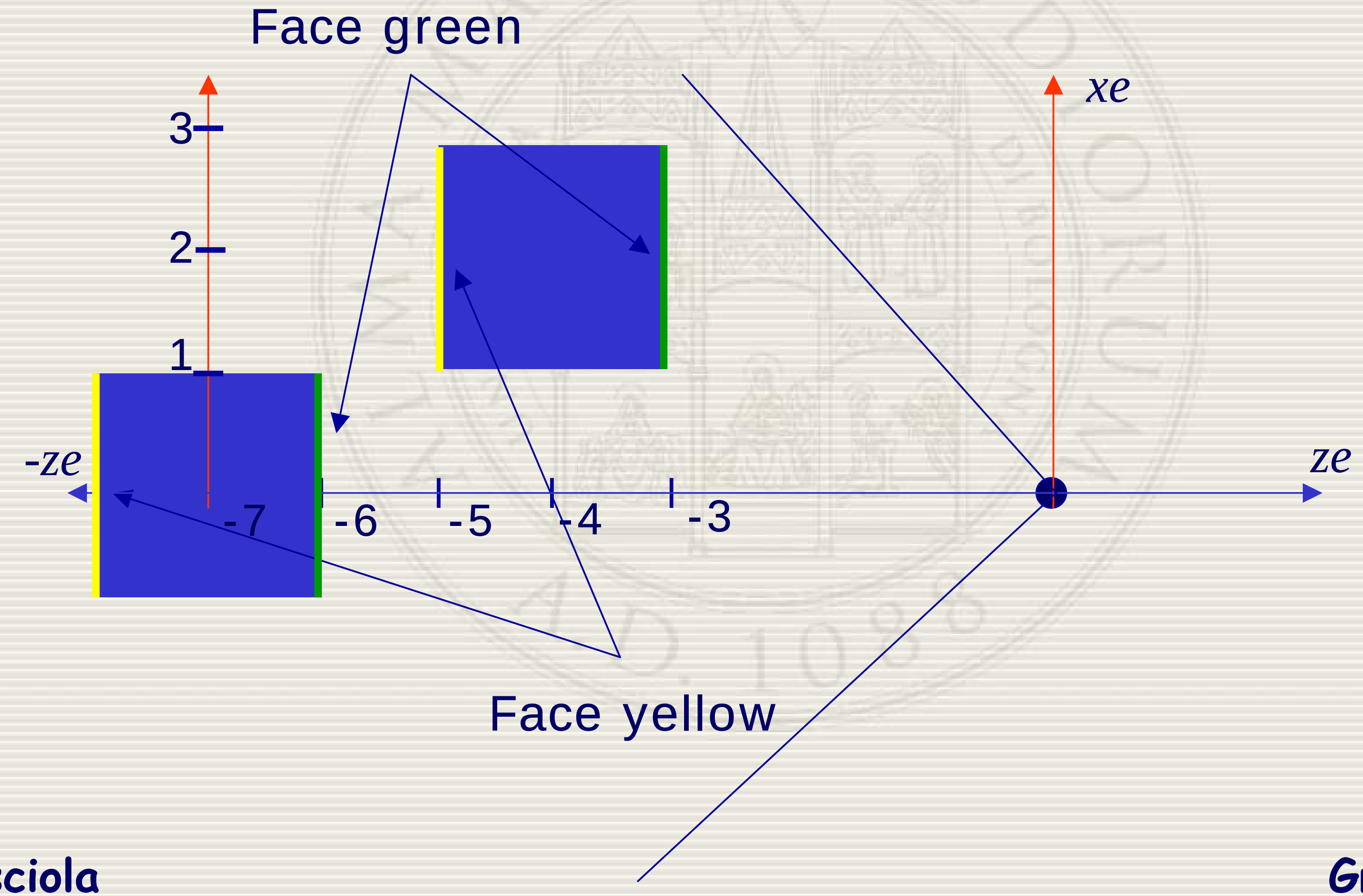
glutsimple_mult.c

La geometria: due cubi posizionati nello spazio come mostrato in figura; quello centrato nell'origine lo vogliamo ruotare, l'altro lo teniamo fisso:



glutsimple_mult.c

Posizioniamo la camera in [7,0,0,1]; questo equivale ad applicare una traslazione di [-7,0,0,0] a tutta la geometria.



glutsimple_mult.c

Effettuiamo delle rotazioni attorno agli assi coordinati (ma solo sul cubo centrato nell'origine).

$$M = \begin{pmatrix} m_0 & m_1 & m_2 & 0 \\ m_4 & m_5 & m_6 & 0 \\ m_8 & m_9 & m_{10} & 0 \\ m_{12} & m_{13} & m_{14} & 1 \end{pmatrix} \qquad R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c & s & 0 \\ 0 & -s & c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

ModelView Matrix — Rotazione intorno all'asse x

$$M \cdot R_x(\theta) = \begin{pmatrix} m_0 & m_1 c - m_2 s & m_1 s + m_2 c & 0 \\ m_4 & m_5 c - m_6 s & m_5 s + m_6 c & 0 \\ m_8 & m_9 c - m_{10} s & m_9 s + m_{10} c & 0 \\ m_{12} & m_{13} & m_{14} & 1 \end{pmatrix}$$

glutsimple_mult.c

Anche se trasparente all'utente, le matrici conservate nello stack modelview potrebbero, se necessario, essere trasposte. Ricordiamo che ci sono due convenzioni:

$$[x', y', z', 1] = [x, y, z, 1]\, M \qquad \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = M^T \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Nello stack vengono memorizzate le matrici M, cioè con l'ultima colonna $[0,0,0,1]^T$;

Questo lascia pensare che una nuova matrice, quando viene aggiunta alla stack, sia post moltiplicata ($M \cdot R_x(\theta)$), invece verrà premoltiplicata, poiché la prima matrice definita sarà l'ultima applicata.

$$M := R_x(\theta) \cdot M$$

Prospettiva con Profondità

La “proiezione”

$$\begin{cases} xs = xe/ze \\ ys = ye/ze \\ zs = \alpha + \beta/ze \end{cases} \qquad (1)$$

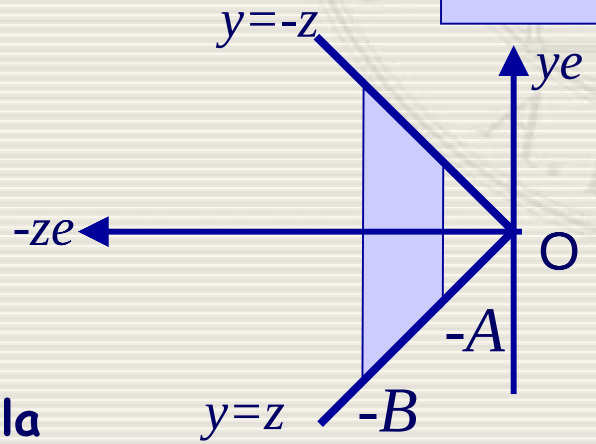
comporta una trasformazione dallo “spazio dell’osservatore” allo “spazio del piano di proiezione” con la proprietà di trasformare rette in rette e piani in piani.

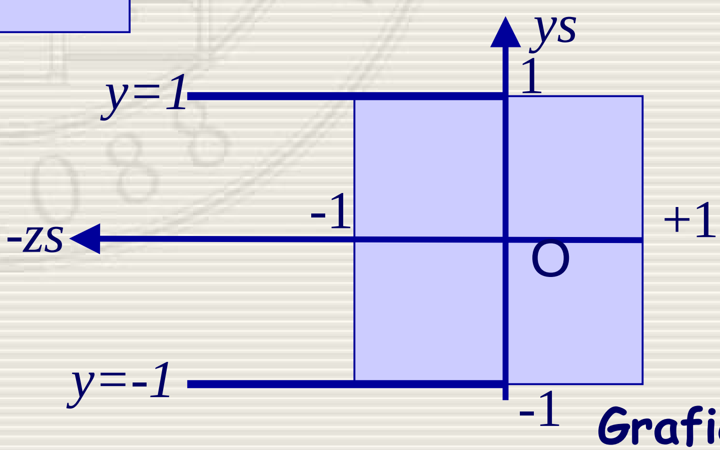
Prospettiva con Profondità

Nello scegliere i valori α e β si può considerare l'intervallo in cui è compreso ze, sia per esempio $[-A,-B]$ con $B > A > 0$ (caso OpenGL), e si voglia ottenere $zs \in [+1,-1]$; allora sarà:

$$\begin{cases} +1 = \alpha - \beta / A \\ -1 = \alpha - \beta / B \end{cases} \qquad \begin{cases} A = A\alpha - \beta \\ B\alpha - \beta = -B \end{cases} \qquad \begin{cases} \beta = A\alpha - A \\ B\alpha - A\alpha + A = -B \end{cases}$$

$$\begin{cases} \beta = A\alpha - A \\ \alpha = -\dfrac{B+A}{B-A} \end{cases} \qquad \boxed{\begin{cases} \beta = -\dfrac{2A \cdot B}{B-A} \\ \alpha = -\dfrac{B+A}{B-A} \end{cases}}$$





gluPerspective(2θ, 1 , A, B);

$[xw,yw,zw,w] = [xe,ye,ze,1]$ A con $A = \begin{pmatrix} d/s & 0 & 0 & 0 \\ 0 & d/s & 0 & 0 \\ 0 & 0 & -\alpha & -1 \\ 0 & 0 & -\beta & 0 \end{pmatrix}$

dove $s = d \tan(\theta)$; da cui

$[xw,yw,zw,w] = [xe\, d/s ,ye\, d/s , -\alpha ze - \beta, -ze]$

Quindi applicando la (1)

$$xs = \frac{xw}{w} = \frac{xe}{-ze}\frac{d}{s}$$

$$ys = \frac{yw}{w} = \frac{ye}{-ze}\frac{d}{s}$$

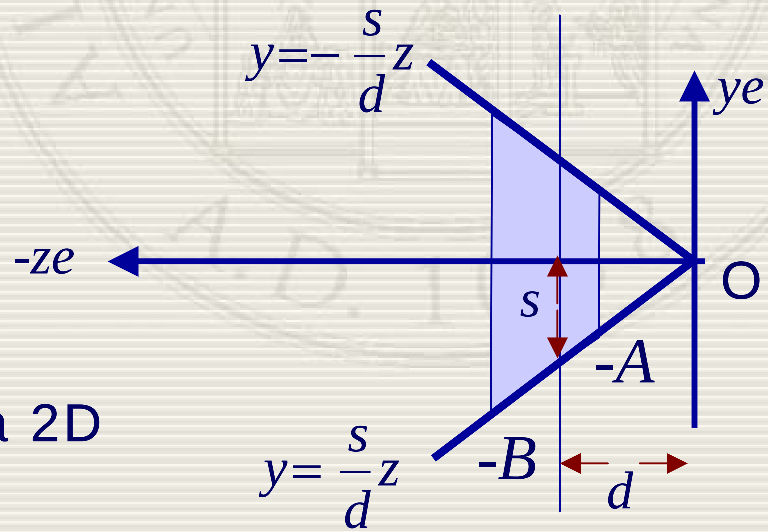
$$zs = \frac{zw}{w} = \frac{-\alpha ze - \beta}{-ze} = \alpha + \frac{\beta}{ze}$$

Pipeline di Vista

Ripercorriamo la Pipeline di Vista con i parametri definiti in OpenGL

1.Trasformazione del sistema di riferimento da sistema oggetto al sistema osservatore (in OpenGL la camera è per default nell'origine del sistema oggetto e la direzione di vista è -z)

2.Clipping 3D solo rispetto al front e back plane, ripettivamente per $z=-A$ e $z=-B$ con $B>A>0$

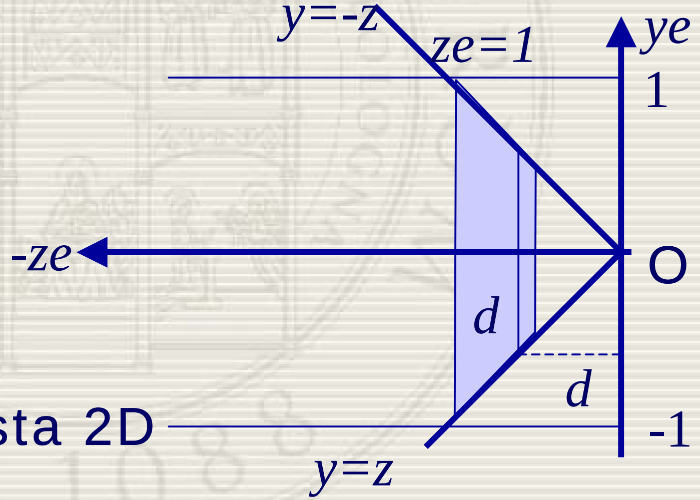


Vista 2D

Pipeline di Vista

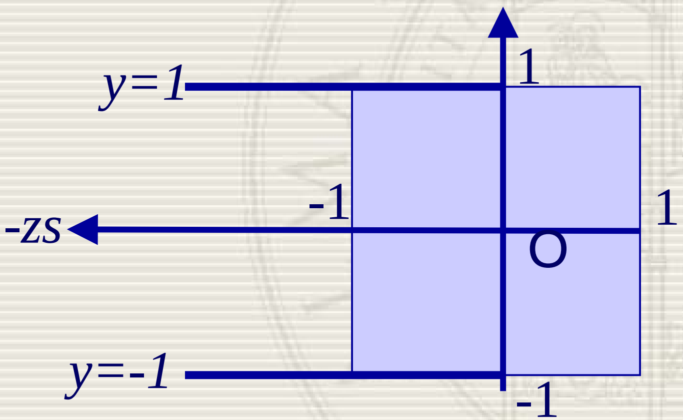
3.Definito il piano di proiezione a distanza d dall'osservatore e l'apertura angolare θ, così ché $s=d\,tan(\theta);$ si definisca la scala S che porta la piramide di vista ad avere le facce laterali ad essere i piani bisettori nel sistema dell'osservatore

$$S=\begin{pmatrix} d/s & 0 & 0 & 0 \\ 0 & d/s & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



Pipeline di Vista

4.Trasformazione dallo spazio dell'osservatore allo spazio del piano di proiezione che porta tutte le z in [+1,-1] e i piani del tronco di piramide ad essere paralleli all'asse z



Vista 2D

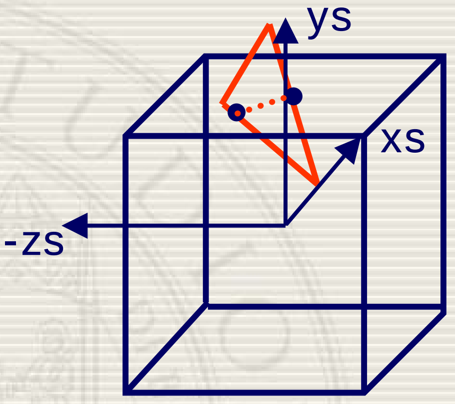
$$\begin{cases} xs = \dfrac{xe}{-ze} \\ ys = \dfrac{ye}{-ze} \\ zs = \alpha + \dfrac{\beta}{ze} \end{cases}$$

$$\begin{cases} \beta = -\dfrac{2A \cdot B}{B - A} \\ \alpha = -\dfrac{B + A}{B - A} \end{cases}$$

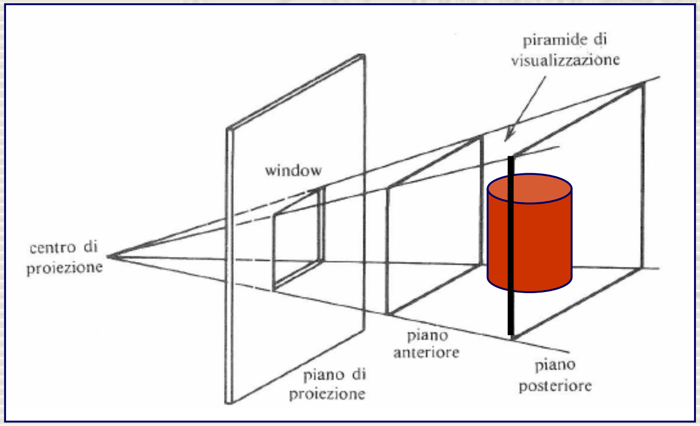
Pipeline di Vista

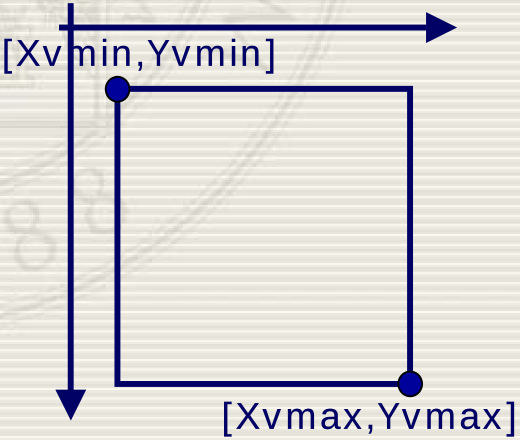
5.Clipping 3D sulle pareti laterali del cubo

$$-1 <= xs <= 1$$
$$-1 <= ys <= 1$$



6.trasformazione window-viewport, $[xs,ys] \longrightarrow [xv,yv]$;
la window ora è [-1,1]x[-1,1]





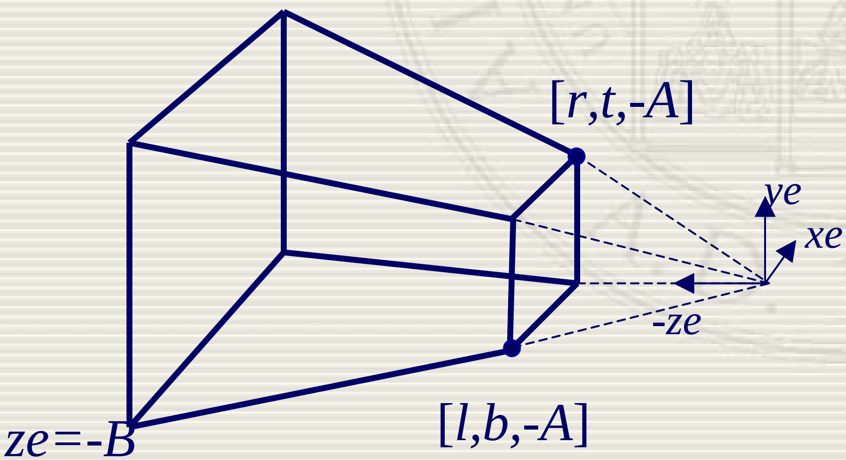
glFrustum(l, r, b, t, A, B);

$[xw,yw,zw,w] = [xe,ye,ze,1]\,\mathrm{A}$ con $A=\begin{pmatrix} \frac{2A}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2A}{t-b} & 0 & 0 \\ \frac{r+l}{r-l} & \frac{t+b}{t-b} & -\alpha & -1 \\ 0 & 0 & -\beta & 0 \end{pmatrix}$

Da cui

$$[xw, yw, zw, w] = [xe\frac{2A}{r-l} + ze\frac{r+l}{r-l}, ye\frac{2A}{t-b} + ze\frac{t+b}{t-b}, -\alpha ze - \beta, -ze]$$

Quindi applicando la (1)



$$xs = \frac{xw}{w} = \frac{xe}{-ze}\frac{2A}{r-l} - \frac{r+l}{r-l}$$

$$ys = \frac{yw}{w} = \frac{ye}{-ze}\frac{2A}{t-b} - \frac{t+b}{t-b}$$

$$zs = \frac{zw}{w} = \frac{-\alpha ze - \beta}{-ze} = \alpha + \frac{\beta}{ze}$$