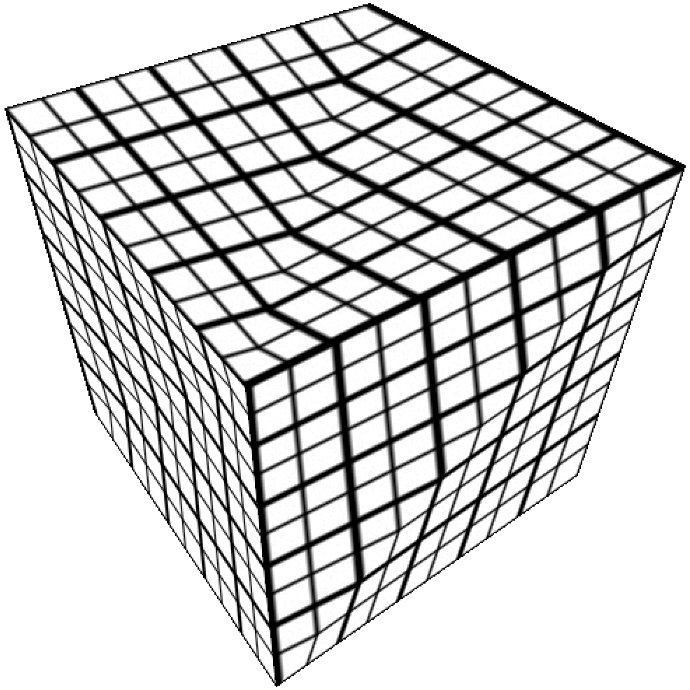
Proiezione Prospettica con Profondità e ZBuffer



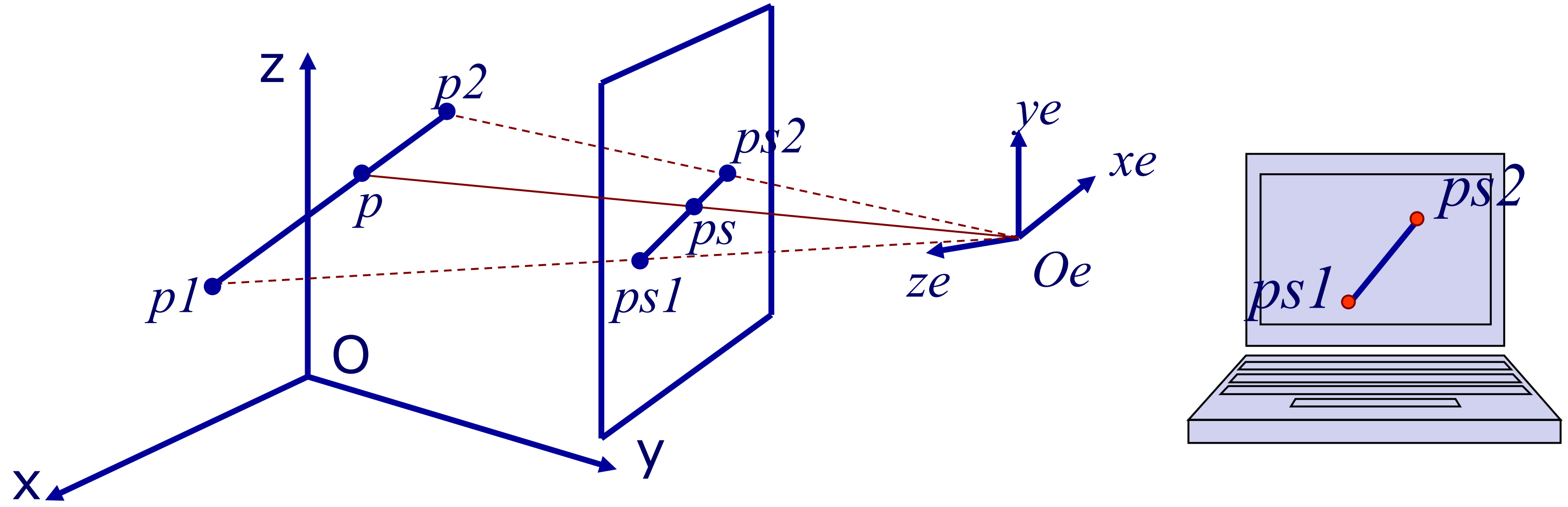
Algoritmo Z-buffer

Riprendiamo l’algoritmo Z-Buffer (o Depth buffer) per la rimozione di parti nascoste, …

```
void zbuffer() {

for (y = 0; y < Vymax; y++)
  for(x = 0; x < Vxmax; x++) {
      FrameBuffer(x,y) = BACK_Col;
      ZBuffer(x,y,INF_Val);
 }

 for (every triangle)
   for (all pixel (px,py) of the projected triangle) {

      pZ = computeZ(px,py);
      if (pZ < ZBuffer(px,py))  //pixel is visible
      {
         FrameBuffer(px,py) = computeColor(px,py);
         ZBuffer(px,py) = pZ;
      }
   }
}
```

Profondità di un Pixel

… vogliamo affrontare il problema di come calcolare la profondità di un pixel.



Problema: siano *ps1* e *ps2* i proiettati di *p1* e *p2*; per tutti i punti/pixel *ps* del segmento di estremi *ps1 e ps2* si vuole determinare la sua profondità, cioè la coordinata *z* di *p* nel sistema *xe ye ze Oe* dell'osservatore di cui *ps* è la proiezione.

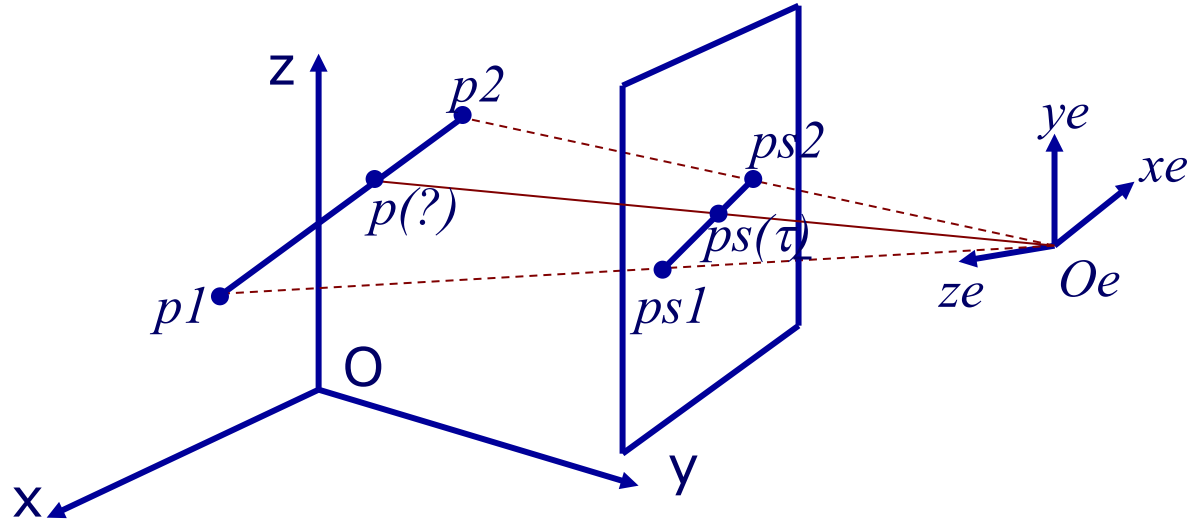
Profondità di un Pixel

Idea: scriviamo il segmento di estremi $ps1$ e $ps2$ in forma baricentrica/parametrica:

$$ps(\tau) = (1-\tau)\, ps1 + \tau\, ps2 \quad \text{con } \tau \in [0, 1].$$

Per ogni punto ps determiniamo la sua coordinata τ e gli associamo come profondità (coordinata z) quella del punto p del segmento $p1$- $p2$, definito dalla stessa coord. baricentrica τ

$$p(\tau) = (1-\tau)\, p1 + \tau\, p2$$

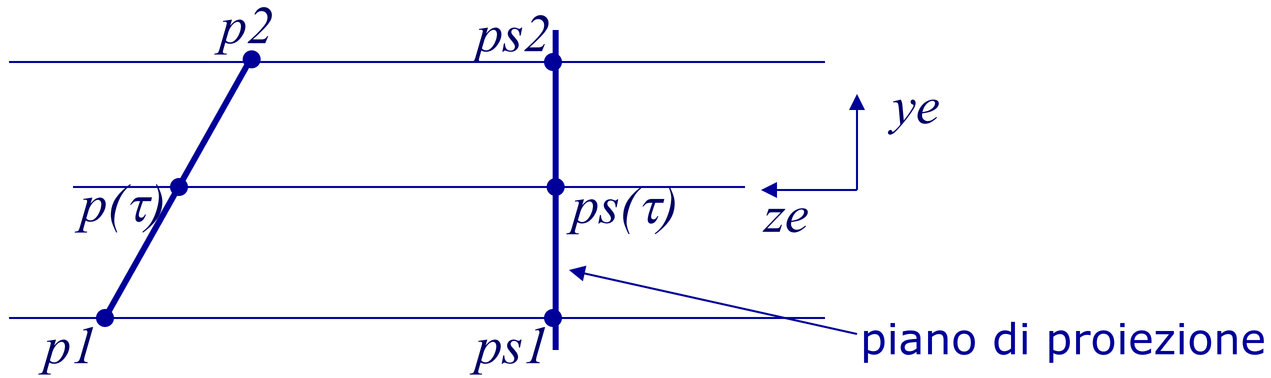


Questo modo di procedere è CORRETTO?

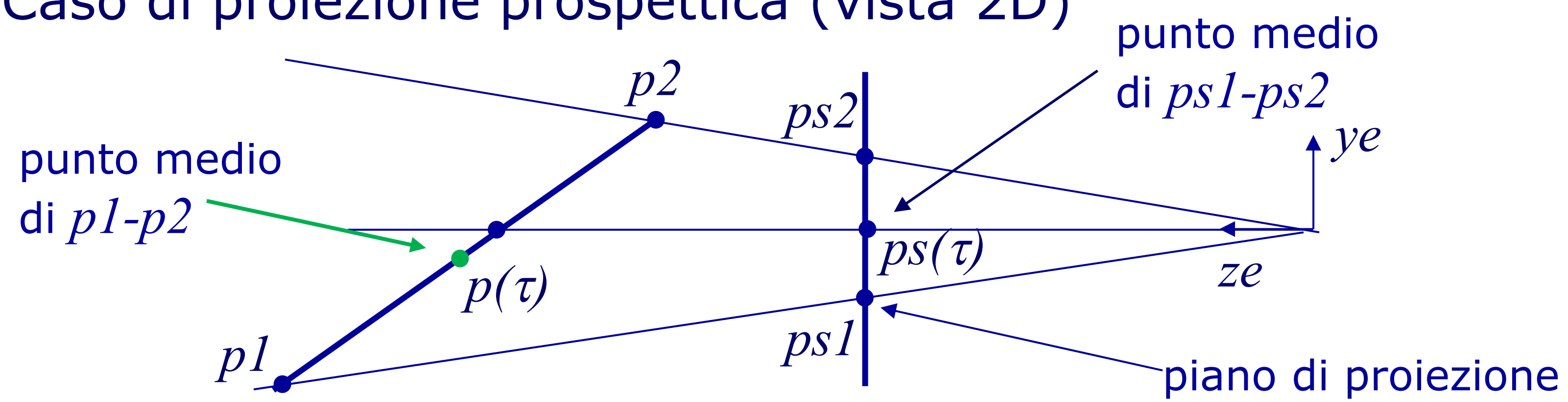
Profondità di un Pixel

E' CORRETTO solo se si sta effettuando una proiezione parallela, ma NON sarà CORRETTO nel caso di proiezione prospettica.

Caso di proiezione parallela (vista 2D)

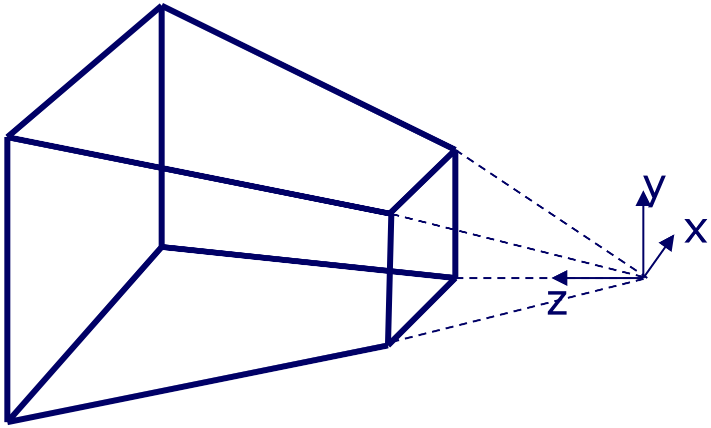


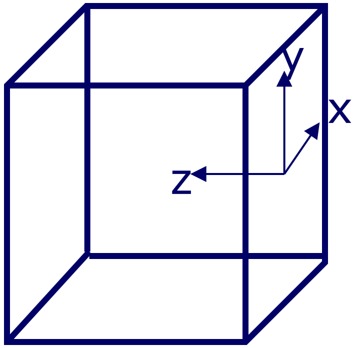
Caso di proiezione prospettica (vista 2D)



Prospettiva con Profondità

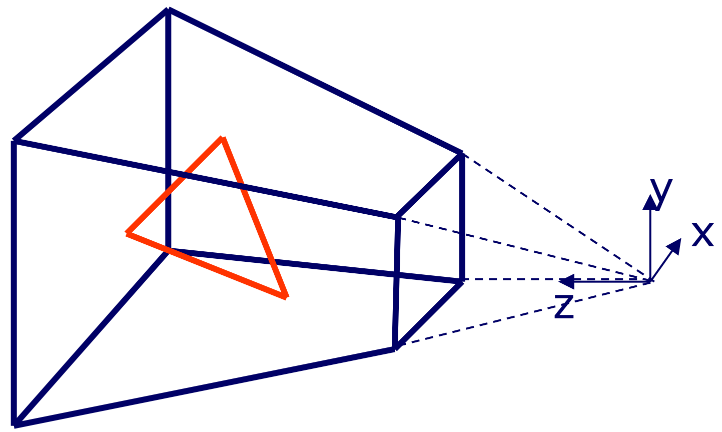
L'obiettivo è quello di definire una trasformazione prospettica dal "sistema 3D dell'osservatore" al "sistema 3D del piano di proiezione" con lo scopo di deformare lo spazio 3D di un tronco di piramide in uno spazio 3D di un cubo;
questo permetterà di interpretare la proiezione prospettica come una proiezione parallela e poter applicare un calcolo semplice (mediante coordinate baricentriche) per la profondità dei pixel.

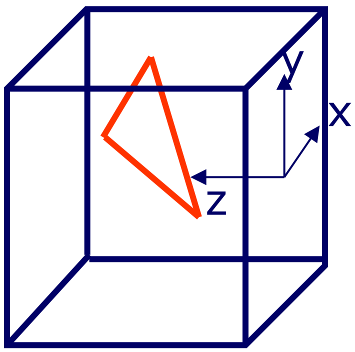




Prospettiva con Profondità

La trasformazione che si cerca deve essere tale da mantenere l'ordine di profondità dei punti trasformati, nella stessa relazione dei punti originali ed ancora deve essere tale da trasformare rette in rette e piani in piani in quanto una geometria poligonale piana sottoposta a questa trasformazione dovrà rimanere poligonale piana.





Prospettiva con Profondità

Teorema. 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" (xe,ye,ze,Oe) allo "spazio del piano di proiezione" (xw,yw, zw,Ow) con la proprietà di trasformare rette in rette e piani in piani, per $\alpha, \beta \neq 0$.

Prospettiva con Profondità

Vediamo per rette: se un segmento viene trasformato in un segmento dovrà esistere una relazione tra i parametri delle due forme parametriche.

Sia $p(t) = (1\text{-}t)\, p1 + t\, p2 \quad t \in [0, 1]$

un segmento nello spazio dell'osservatore con $p_i = [x_i, y_i, z_i]$ $i=1,2$.

Siano $ps_i = [xs_i, ys_i, zs_i]$ $i=1,2$ i punti ottenuti applicando la trasformazione, cioè $ps_i = [x_i / z_i, y_i / z_i, \alpha+\beta / z_i]$ ed il segmento relativo sia $ps(\tau) = (1\text{-}\tau)\, ps1 + \tau\, ps2 \quad \tau \in [0, 1]$.

Che relazione c'è tra t e τ ? La relazione si ricaverà imponendo che $p(t)$ venga trasformato in $ps(\tau)$; guardiamo l'ascissa del punto trasformato e a cosa corrisponde:

$$\frac{(1-t)x_1 + tx_2}{(1-t)z_1 + tz_2} = (1-\tau)\frac{x_1}{z_1} + \tau \frac{x_2}{z_2}$$

Prospettiva con Profondità

i passaggi:

$$\frac{(1-t)x_1 + tx_2}{(1-t)z_1 + tz_2} = (1-\tau)\frac{x_1}{z_1} + \tau\frac{x_2}{z_2}$$

$$\frac{(1-t)x_1 + tx_2}{(1-t)z_1 + tz_2} = \frac{x_1}{z_1} + \tau\left(\frac{x_2}{z_2} - \frac{x_1}{z_1}\right)$$

ricaviamo τ: $\tau = \left(\frac{(1-t)x_1 + tx_2}{(1-t)z_1 + tz_2} - \frac{x_1}{z_1}\right)\frac{1}{\left(\frac{x_2}{z_2} - \frac{x_1}{z_1}\right)}$

e facendo un po' di conti:

$$\boxed{\tau = \frac{tz_2}{(1-t)z_1 + tz_2}} \qquad (2)$$

Prospettiva con Profondità

Dalla relazione (2), si può ricavare banalmente la (3):

(2) $$\boxed{\tau = \frac{t \cdot z_2}{(1-t)\cdot z_1 + t \cdot z_2}}$$

$$(1-\tau) = \frac{(1-t)\cdot z_1}{(1-t)\cdot z_1 + t \cdot z_2}$$ (3)

e ricavando t, si ottengono le relazioni inverse (4) e (5):

(4) $$t = \frac{\tau \cdot z_1}{\tau \cdot z_1 + (1-\tau)\cdot z_2}$$

$$(1-t) = \frac{(1-\tau)\cdot z_2}{\tau \cdot z_1 + (1-\tau)\cdot z_2}$$ (5)

Prospettiva con Profondità

Verifichiamo che la proiezione (1) trasforma rette in rette. Si considera la proiezione di un punto *p(t)* della retta per *p1 p2* e si verifica che stia sulla retta per *ps1 ps2*.

$$ps = \left[\frac{(1-t)\cdot x_1 + tx_2}{(1-t)\cdot z_1 + t\cdot z_2}, \frac{(1-t)\cdot y_1 + ty_2}{(1-t)\cdot z_1 + t\cdot z_2}, \alpha + \frac{\beta}{(1-t)\cdot z_1 + t\cdot z_2} \right]$$

Consideriamo la prima coordinata di *ps* e moltiplichiamo e dividiamo per z_1 e z_2:

$$\frac{(1-t)\cdot x_1}{(1-t)\cdot z_1 + t\cdot z_2}\frac{z_1}{z_1} + \frac{tx_2}{(1-t)\cdot z_1 + t\cdot z_2}\frac{z_2}{z_2}$$

(2) $$\tau = \frac{t\cdot z_2}{(1-t)\cdot z_1 + t\cdot z_2}$$

$$(1-\tau) = \frac{(1-t)\cdot z_1}{(1-t)\cdot z_1 + t\cdot z_2}$$ (3)

Prospettiva con Profondità

Utilizzando le relazioni (3) e (2) si ha:

$$(1-\tau)\frac{x_1}{z_1}+\tau\frac{x_2}{z_2}$$

Analogamente per la seconda coordinata di ps.

Consideriamo ora la terza coordinata di ps; dalle (2) e (3), dividendo per z_2 la prima e z_1 la seconda e sommando si ottiene:

$$\frac{\tau}{z_2}+\frac{(1-\tau)}{z_1}=\frac{1}{(1-t)\cdot z_1+t\cdot z_2}$$

vediamo come sostituirla nella terza coordinata di ps:

Prospettiva con Profondità

$$\alpha + \frac{\beta}{(1-t)\cdot z_1 + t\cdot z_2} = \alpha(1-\tau) + \alpha\tau + \beta\left(\frac{1-\tau}{z_1} + \frac{\tau}{z_2}\right) =$$

$$= (1-\tau)\left(\alpha + \frac{\beta}{z_1}\right) + \tau\left(\alpha + \frac{\beta}{z_2}\right)$$

e quindi le coordinate di ps si possono riscrivere:

$$ps = \left[(1-\tau)\frac{x_1}{z_1} + \tau\frac{x_2}{z_2}, (1-\tau)\frac{y_1}{z_1} + \tau\frac{y_2}{z_2}, (1-\tau)\left(\alpha + \frac{\beta}{z_1}\right) + \tau\left(\alpha + \frac{\beta}{z_2}\right)\right]$$

questa è la verifica che si tratta di un punto sul segmento $ps1\ ps2$ relativo al parametro τ corrispondente di t e quindi che rette vengono trasformate in rette.

Prospettiva con Profondità

In conclusione le relazioni (2) e (3) forniscono i parametri τ e $(1-\tau)$ che se combinati con le coordinate dei vertici proiettati permettono di ottenere le coordinate proiettate dei punti originali, in particolare per la componente z:

$$\boxed{zs} = (1-\tau)(\alpha+\beta/z_1)+\tau(\alpha+\beta/z_2)$$

$$= \frac{(1-t)z_1}{(1-t)z_1+tz_2}(\alpha+\beta/z_1)+\frac{tz_2}{(1-t)z_1+tz_2}(\alpha+\beta/z_2)$$

$$= \alpha+\frac{\beta}{(1-t)z_1+tz_2}$$

$$= \boxed{\alpha+\beta/z}$$

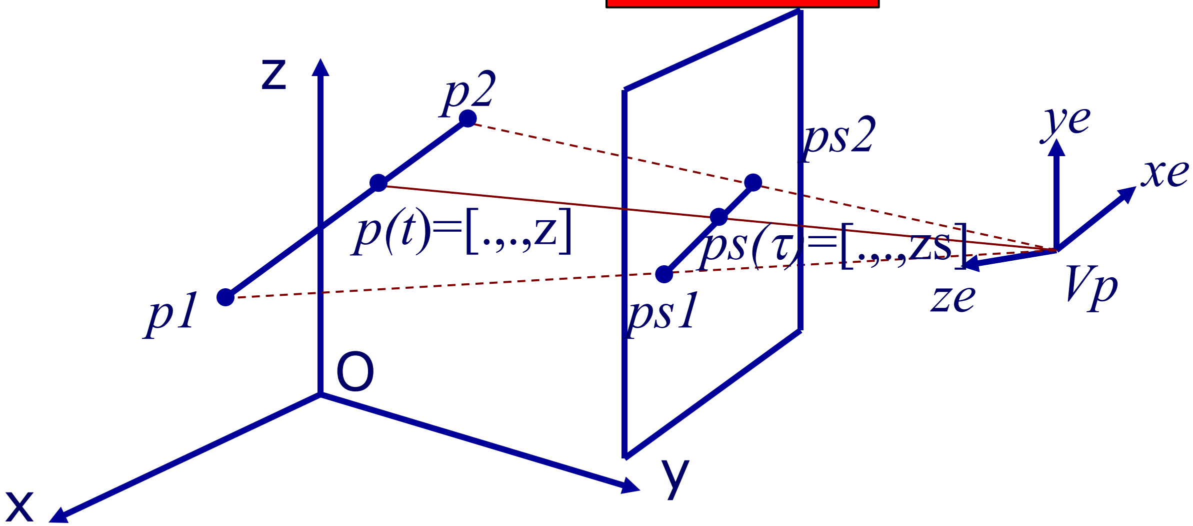
Prospettiva con Profondità

Da quest'ultima, per avere il valore originale z possiamo utilizzare la formula:

$$\boxed{z = \frac{\beta}{zs - \alpha}}$$

dove:

$$zs = (1-\tau)\cdot zs_1 + \tau \cdot zs_2$$



oppure, da τ:

$$t = \frac{\tau \cdot z_1}{\tau \cdot z_1 + (1-\tau)\cdot z_2}$$

Si noti che possiamo usare le coordinate proiettate xs e ys per il disegno e zs per la profondità, ma per il calcolo corretto di colore e texture è essenziale recuperare le coordinate 3D e quindi utilizzare le relazioni viste tra i parametri t e τ o la formula sopra ricavata per la z originale.

Prospettiva con Profondità

Vediamo per piani: se un triangolo viene trasformato in un triangolo dovrà esistere una relazione tra i parametri delle due forme parametriche.

Sia $p(u,v,s) = u\,p1 + v\,p2 + s\,p3 \quad u,v,s \in [0,1] \;\; u+v+s=1$

un triangolo nello spazio dell'osservatore con $p_i = [x_i, y_i, z_i]$ $i=1,2,3$.

Siano $ps_i = [xs_i, ys_i, zs_i]$ $i=1,2,3$ *i punti* ottenuti applicando la trasformazione, cioè $ps_i = [x_i/z_i, y_i/z_i, \alpha+\beta/z_i]$ ed il triangolo relativo sia

$$ps(u',v',s') = u'ps1 + v'ps2 + s'ps3 \;\; u',v',s' \in [0,1], \quad u'+v'+s'=1.$$

Che relazione c'è tra u, v, s e u', v', s'? La relazione si ricava imponendo che $p(u, v, s)$ venga trasformato in $ps(u', v', s')$; guardiamo l'ascissa del punto trasformato e a cosa corrisponde:

$$\frac{ux_1 + vx_2 + sx_3}{uz_1 + vz_2 + sz_3} = u'\frac{x_1}{z_1} + v'\frac{x_2}{z_2} + s'\frac{x_3}{z_3}$$

Prospettiva con Profondità

Si ricavano le relazioni dirette:

(6) $$u' = \frac{u \cdot z_1}{u \cdot z_1 + v \cdot z_2 + s \cdot z_3}$$

$$v' = \frac{v \cdot z_2}{u \cdot z_1 + v \cdot z_2 + s \cdot z_3}$$ (7)

$$s' = \frac{s \cdot z_3}{u \cdot z_1 + v \cdot z_2 + s \cdot z_3}$$ (8)

Dalle (6), (7) e (8) si ricava:

$$\frac{u'}{z_1} + \frac{v'}{z_2} + \frac{s'}{z_3} = \frac{1}{u \cdot z_1 + v \cdot z_2 + s \cdot z_3}$$ (9)

e quindi le relazioni inverse delle (6), (7) e (8) …

Prospettiva con Profondità

… infatti ponendo:

$$fz = \frac{u'}{z_1} + \frac{v'}{z_2} + \frac{s'}{z_3} = \frac{1}{u \cdot z_1 + v \cdot z_2 + s \cdot z_3} \qquad (9')$$

dalle (6), (7) e (8) si hanno le:

(6′) $u = \dfrac{u'}{z_1 \cdot fz}$ (7′) $v = \dfrac{v'}{z_2 \cdot fz}$ (8′) $s = \dfrac{s'}{z_3 \cdot fz}$

Prospettiva con Profondità

Verifichiamo che la proiezione (1) trasforma piani in piani. Si considera la proiezione di un punto *p(u,v,s)* del piano per *p1p2p3* e si verifica che stia sul piano per *ps1ps2ps3*.

$$ps = \left[\frac{u \cdot x_1 + v \cdot x_2 + s \cdot x_3}{u \cdot z_1 + v \cdot z_2 + s \cdot z_3}, \frac{u \cdot y_1 + v \cdot y_2 + s \cdot y_3}{u \cdot z_1 + v \cdot z_2 + s \cdot z_3}, \alpha + \frac{\beta}{u \cdot z_1 + v \cdot z_2 + s \cdot z_3} \right]$$

dalle (6), (7) e (8) e dalla (9), si ha:

$$ps = \left[u'\frac{x_1}{z_1} + v'\frac{x_2}{z_2} + s'\frac{x_3}{z_3}, u'\frac{y_1}{z_1} + v'\frac{y_2}{z_2} + s'\frac{y_3}{z_3}, u'\left(\alpha + \frac{\beta}{z_1}\right) + v'\left(\alpha + \frac{\beta}{z_2}\right) + s'\left(\alpha + \frac{\beta}{z_3}\right) \right]$$

Prospettiva con Profondità

Dalle relazioni (6), (7) e (8):

$$u' = \frac{u \cdot z_1}{u \cdot z_1 + v \cdot z_2 + s \cdot z_3} \qquad v' = \frac{v \cdot z_2}{u \cdot z_1 + v \cdot z_2 + s \cdot z_3} \qquad s' = \frac{s \cdot z_3}{u \cdot z_1 + v \cdot z_2 + s \cdot z_3}$$

combinare con *u', v', s'* le coordinate proiettate, permette di avere i valori originali proiettati, in particolare vediamo la componente z:

$$\boxed{zs} = u'(\alpha + \beta / z_1) + v'(\alpha + \beta / z_2) + s'(\alpha + \beta / z_3)$$

$$= \frac{uz_1}{uz_1 + vz_2 + sz_3}(\alpha + \beta / z_1) + \frac{vz_2}{uz_1 + vz_2 + sz_3}(\alpha + \beta / z_2) + \frac{sz_3}{uz_1 + vz_2 + sz_3}(\alpha + \beta / z_3)$$

$$= \alpha + \frac{\beta}{uz_1 + vz_2 + sz_3}$$

$$= \boxed{\alpha + \beta / z}$$

Prospettiva con Profondità

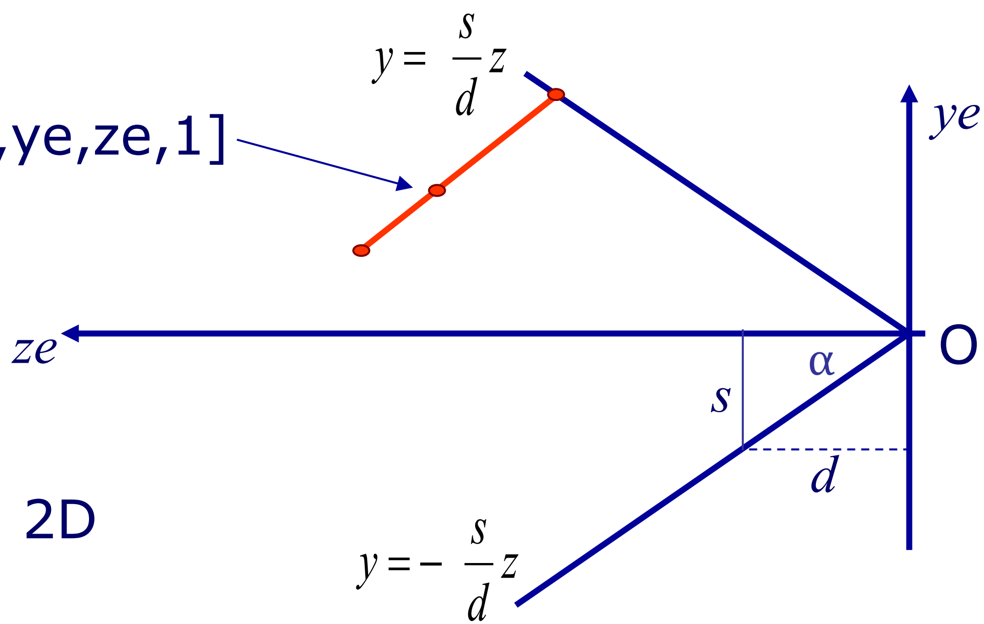
Da quest'ultima, per avere il valore originale z possiamo utilizzare la formula:

$$\boxed{z = \frac{\beta}{zs - \alpha}} \qquad (10)$$

Si noti che possiamo usare le coordinate proiettate xs e ys per il disegno e zs per la profondità, ma per il calcolo corretto di colore e texture è essenziale recuperare le coordinate 3D e quindi utilizzare le relazioni viste tra i parametri o la formula sopra ricavata.

Prospettiva con Profondità

Cerchiamo di capire di più sulla trasformazione (1):



Vista 2D

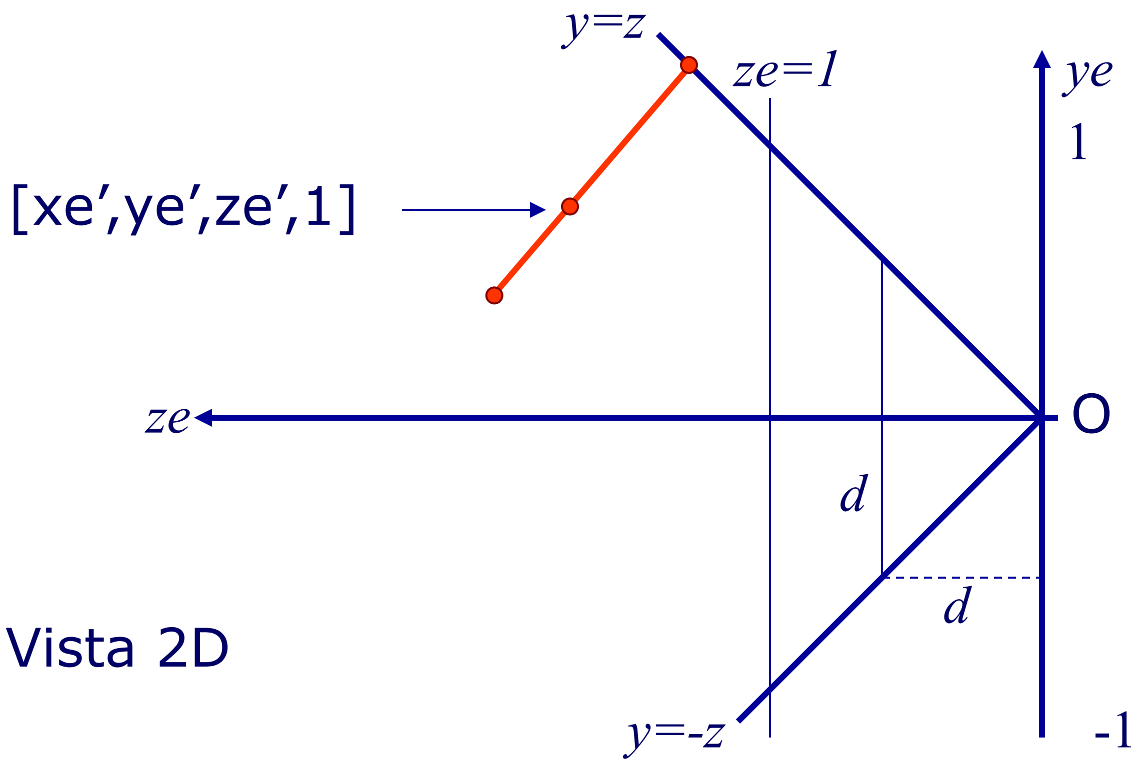
prima di applicarla effettuiamo la seguente trasformazione di scala:

$[xe',ye',ze',1]^T = S\,[xe,ye,ze,1]^T$ con $S = \begin{pmatrix} d/s & 0 & 0 & 0 \\ 0 & d/s & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

ed $s = d \tan(\alpha)$

Prospettiva con Profondità

Otterremo così una piramide di vista di 90 gradi, od anche con le facce laterali che sono i piani bisettori, e senza che vengano alterate le coordinate z dei punti



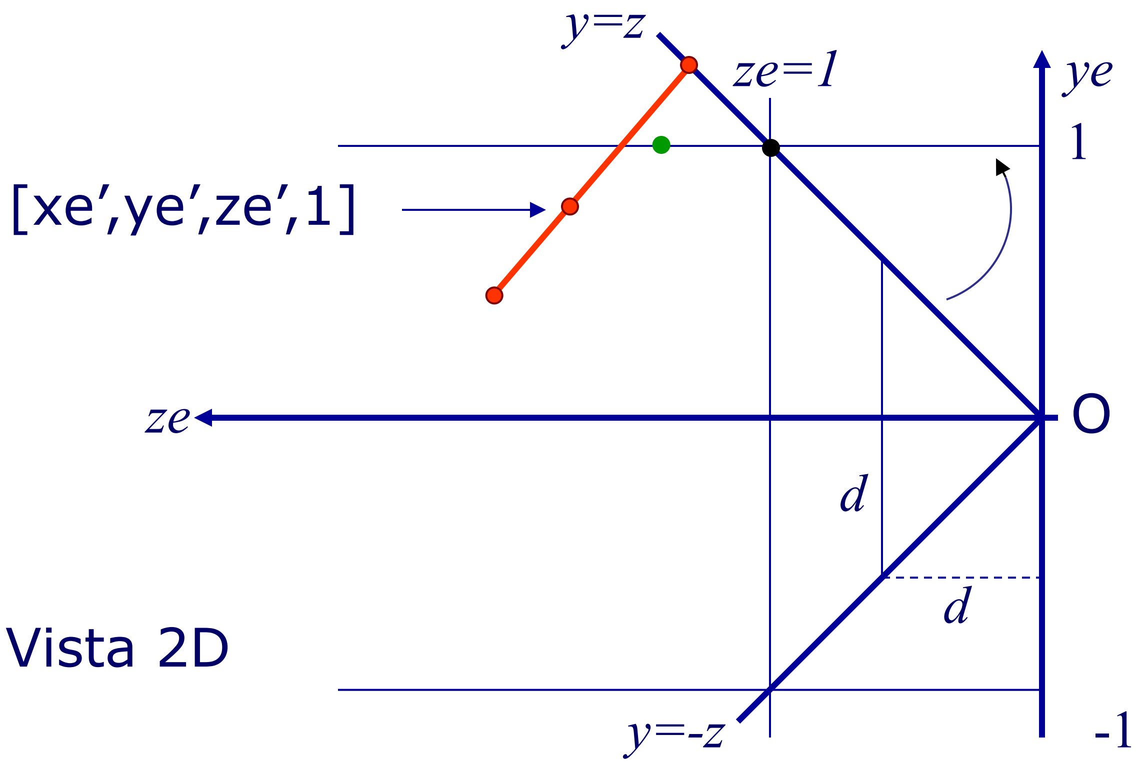
Vista 2D

Prospettiva con Profondità

Applichiamo ora la trasformazione (1):

i punti della forma $\left[\bullet, \bar{z}, \bar{z}\right]$ vengono traformati in $\left[\bullet, 1, \alpha + \beta / \bar{z}\right]$

La retta $y=z$ viene trasformata nella retta $y=1$;

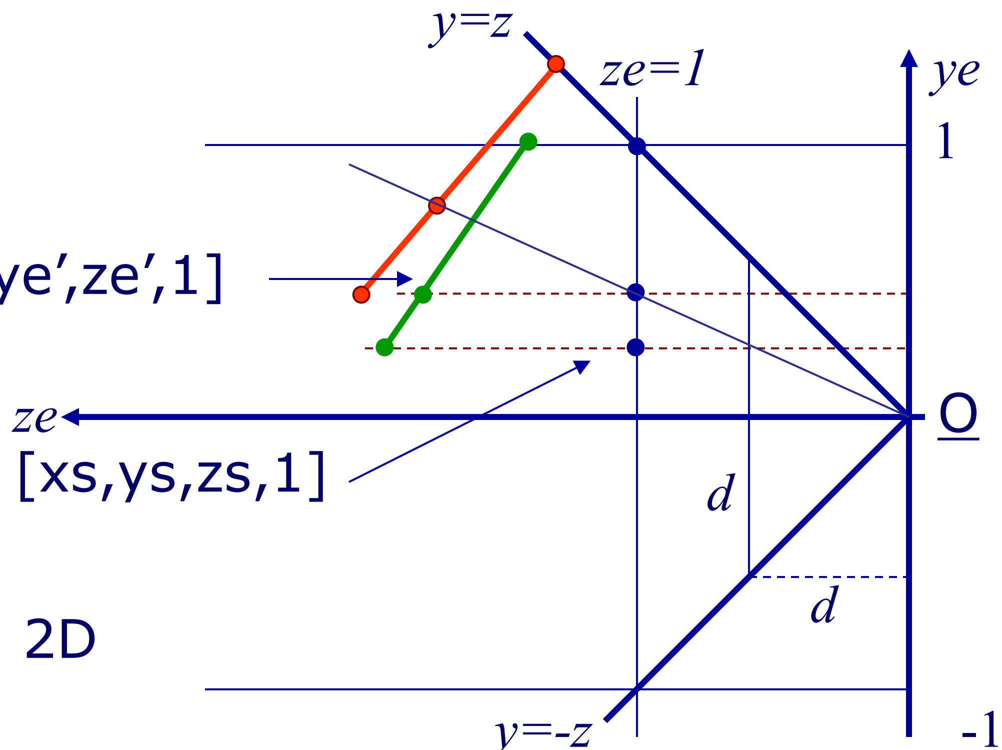


Vista 2D

Prospettiva con Profondità

tutte le rette uscenti dall'origine $y=mz$ saranno trasformate nelle rette $y=m$.
In pratica ruotano sul punto di interzezione di $y=mz$ con $ze=1$.

Ancora, punti allineati, una volta trasformati, saranno allineati.



Vista 2D

Prospettiva con Profondità

E' possibile scegliere α e β in modo da ottenere un intervallo conveniente per i valori zs trasformati.

Scegliendo $\beta < 0$ si conserva la nozione intuitiva di profondità, cioè se un punto ha un valore ze maggiore di un altro, il suo trasformato zs resterà maggiore.

Siano dati due punti tali che $ze1 < ze2$, e verifichiamo che $zs1 < zs2$ con $zs_i = \alpha + \beta / ze_i$ e $\beta<0$.

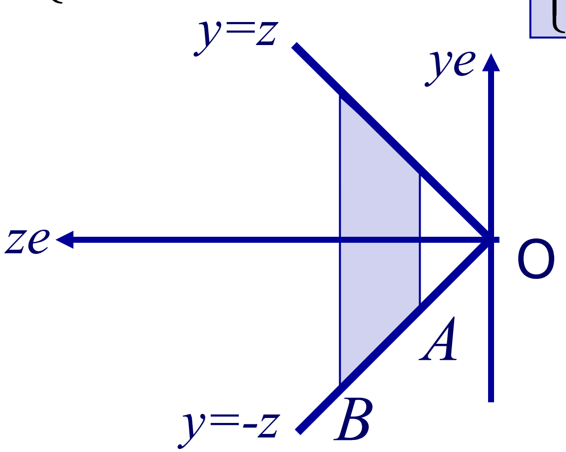
Sarà: $$\frac{1}{ze1} > \frac{1}{ze2}; \quad \text{e se} \quad \beta < 0 \quad \frac{\beta}{ze1} < \frac{\beta}{ze2} \quad c.v.$$

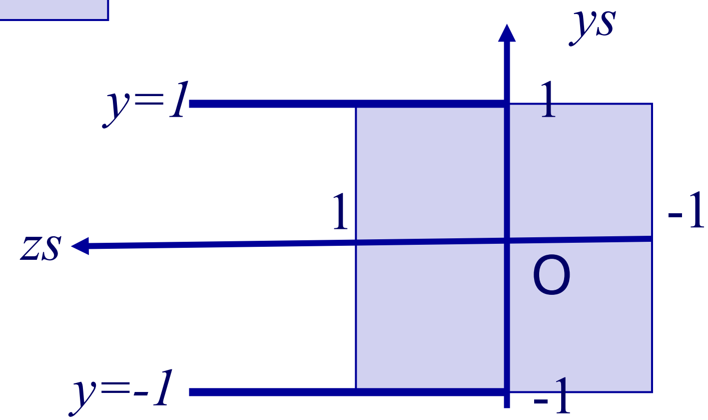
Prospettiva con Profondità

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

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

$$\begin{cases} \beta = -A(1+\alpha) \\ \alpha = \dfrac{A+B}{B-A} \end{cases} \qquad \boxed{\begin{cases} \beta = -\dfrac{2AB}{B-A} \\ \alpha = \dfrac{A+B}{B-A} \end{cases}} \qquad (11)$$





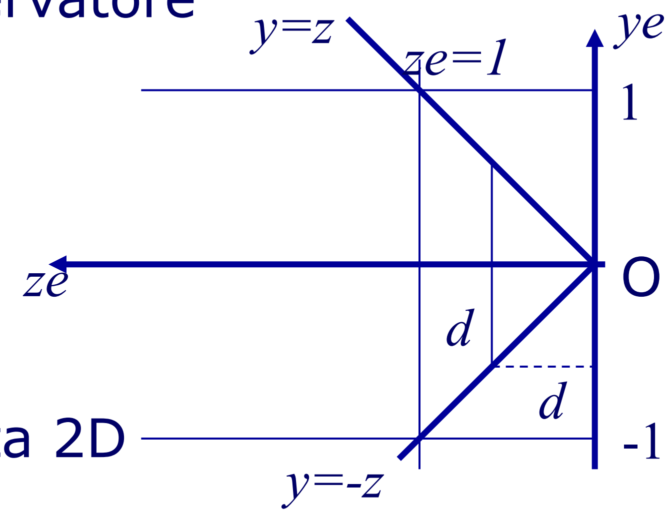
Pipeline di Vista

1. Trasformazione del sistema di riferimento da sistema oggetto a sistema osservatore

2. Definito il piano di proiezione a distanza d dall'osservatore e l'apertura angolare α, così ché $s=d\ tan(\alpha)$ si applichi la scala S che porta la piramide di vista ad avere come facce laterali i piani bisettori nel sistema dell'osservatore

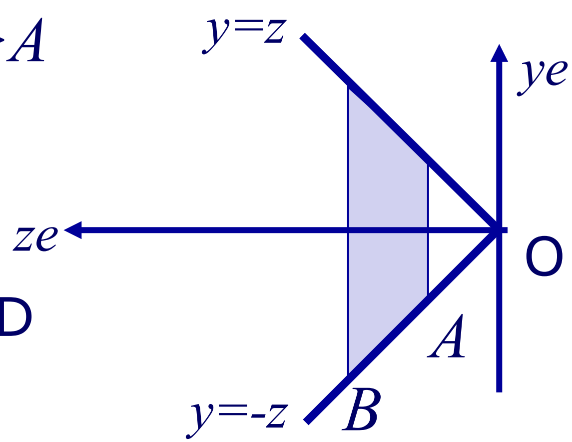
$$S = \begin{pmatrix} d/s & 0 & 0 & 0 \\ 0 & d/s & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Vista 2D



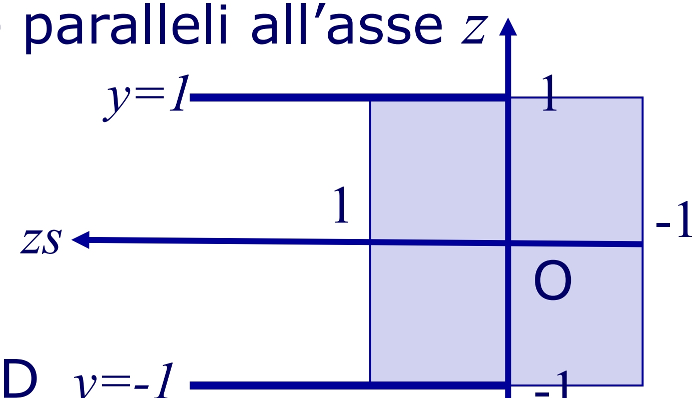
Pipeline di Vista

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



Vista 2D

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



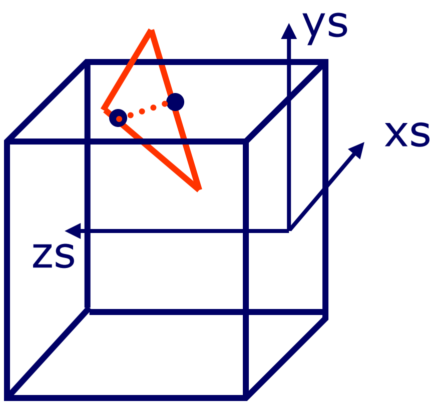
$$\begin{cases} xs = xe / ze \\ ys = ye / ze \\ zs = \alpha + \beta / ze \end{cases} \qquad \begin{cases} \beta = -\dfrac{2AB}{B-A} \\ \alpha = \dfrac{A+B}{B-A} \end{cases}$$

Vista 2D

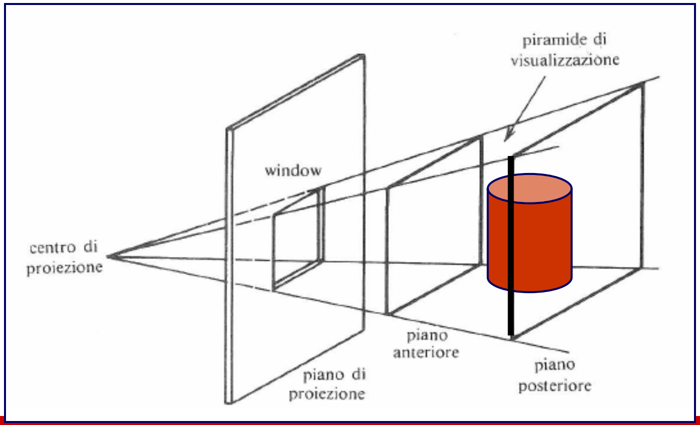
Pipeline di Vista

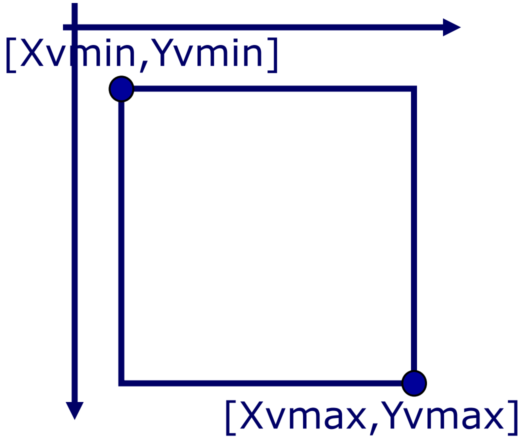
5.Clipping 3D sulle pareti laterali del cubo

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



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





Pipeline di Vista

Trasformazione Sistema di Riferimento
(x,y,z,O) → (xe,ye,ze,Vp)

↓

Scala della Piramide di Vista

↓

Clipping 3D rispetto al front e back plane

↓

Prospettiva con profondità
(xe,ye,ze,Vp) → (Xw,Yw,Zw,Ow)

↓

Clipping 3D rispetto alle pareti laterali del cubo

↓

Trasformazione Window-Viewport
(Xw,Yw,Ow) → (Xv,Yv,Ov)

Rasterizzazione di Triangoli per Z-buffer

Nel caso di triangoli ottenuti per proiezione con correzione prospettica per ogni vertice del triangolo 3D, avremo

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

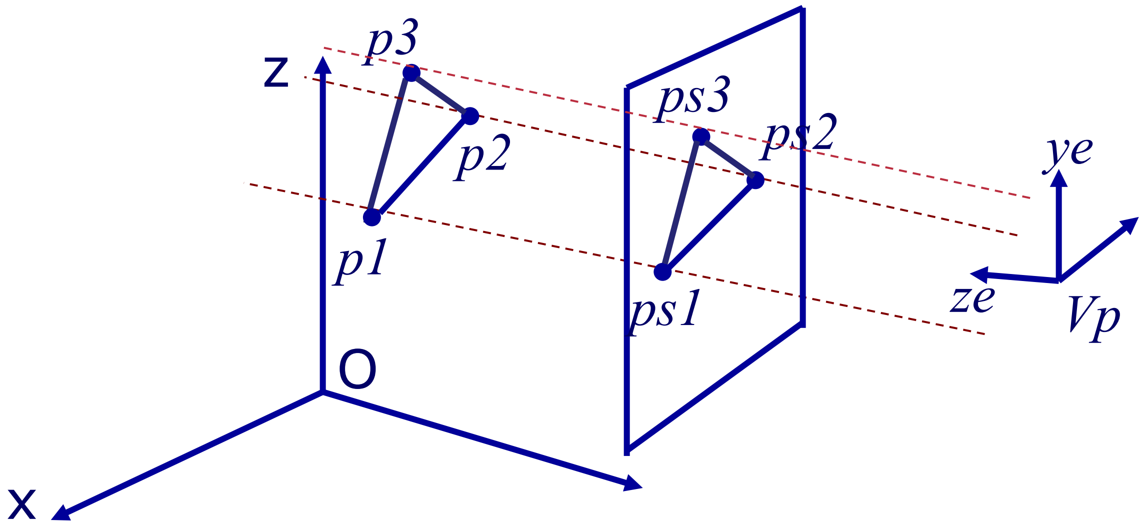
con alpha e beta valori per correzione prospettica.
Quindi si applica la rasterizzazione al “triangolo 2D”.

Dalle coordinate xs, ys, zs ottenute per proiezione con correzione prospettica dei vertici, l’algoritmo di rasterizzazione genererà contemporaneamente le coordinate dei pixel del triangolo 2D, e una coordinata z che rappresenta la profondità del punto 3D nello Spazio del Piano di Proiezione a cui il pixel corrisponde, infatti …

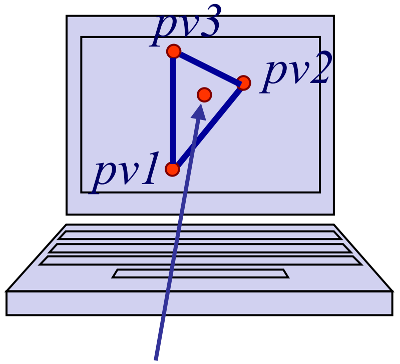
Rasterizzazione di Triangoli per Z-buffer

… determinate le coordinate [xs, ys, zs] proiettate dei vertici di un triangolo ed effettuato il mapping window-viewport, l'algoritmo di rasterizzazione, per ogni pixel interno al triangolo potrà determinarne la profondità. Siano [u', v', s'] le sue coordinate baricentriche, allora la sua profondità z sarà determinata come

$$z = u' * zs1 + v' * zs2 + s' * zs3$$



Spazio dell'Osservatore e proiezione parallela



Coord. Baricentriche
[u, v, s]

Rasterizzazione di Triangoli per Z-buffer

Se si vuole utilizzare l'algoritmo di rasterizzazione basato su scan-conversion, questi dovrà essere modificato per gestire anche le coordinate z e le strutture ET e AET dovranno essere ampliate:

ET

Ymax	Xmin	1/m	Zmin	1/mz	

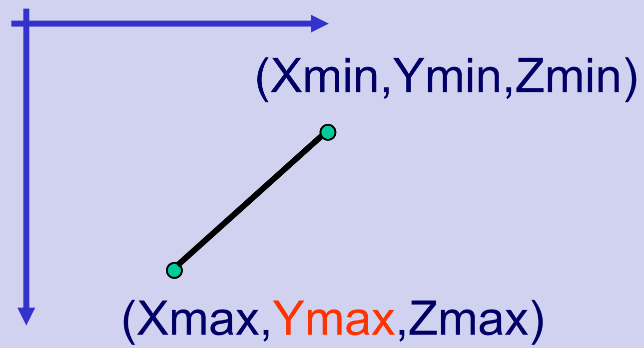
$$\frac{1}{m} = \frac{Xmax-Xmin}{Ymax-Ymin} \qquad \frac{1}{mz} = \frac{Zmax-Zmin}{Ymax-Ymin}$$

Ymax	X	1/m	Z	1/mz	

AET

```
Y++;
X + = 1/m;
Z + = 1/mz;
```

Nota: con Xmax si intende la coord. x associata al punto di maggior ordinata



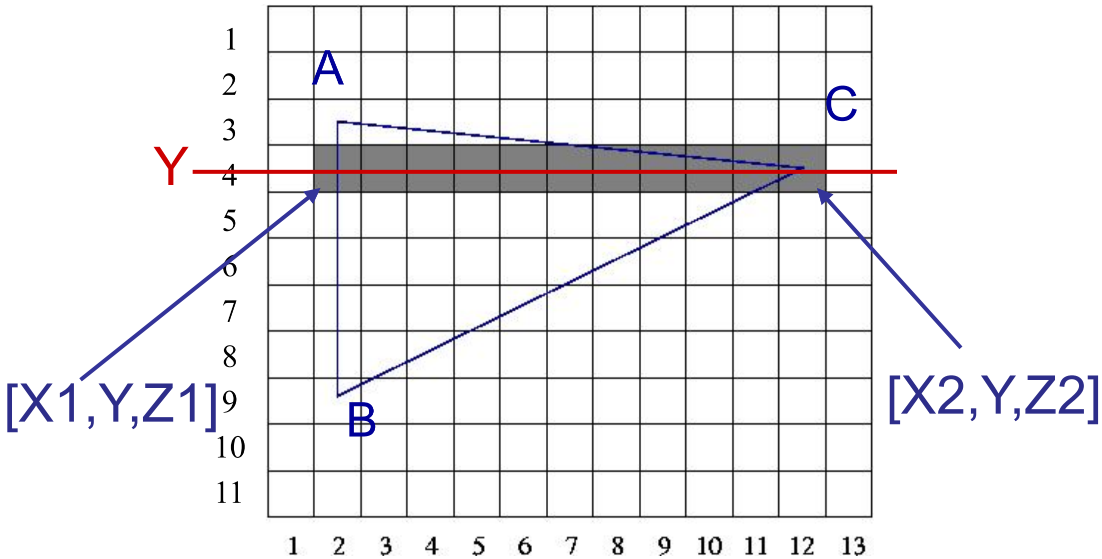
Rasterizzazione di Triangoli per Z-buffer

Sulla scanline Y, determinata la coppia di ascisse X1 e X2 fra cui accendere i pixel, avremo anche le coordinate Z1 e Z2 che rappresentano le profondità dei pixel (X1,Y) e (X2,Y).

Per ogni pixel (X,Y) con X=X1,…,X2, la relativa profondità Z potrà essere calcolata come:

```
n=X2-X1;
dZ=(Z2-Z1)/n;
X=X1;
Z=Z1;
for i=1,..,n
   X ++;
   Z += dZ;
```

Algoritmo di Linea incrementale



Rasterizzazione di Triangoli per Z-buffer e gestione colori

Per ottenere le componenti colore corrette, una volta determinate le componenti interpolate si possono calcolare direttamente le relative coordinate baricentriche [u', v', s'] sul triangolo schermo e da queste, utilizzando le relazioni inverse delle (6), (7) e (8) che danno [u , v , s], calcolare le informazioni colore originali come:

```
Re = u * Re1 + v * Re2 + s * Re3 ;
Ge = u * Ge1 + v * Ge2 + s * Ge3;
Be = u * Be1 + v * Be2 + s * Be3;
```

Coord. baricentriche

dove

```
u = u' / (ze1 * fz) ;
v = v' / (ze2 * fz) ;
s = s' / (ze3 * fz) ;
```

vedi formule (6'), (7') e (8')

con

```
fz = u'/ze1 + v'/ze2 + s'/ze3;
```

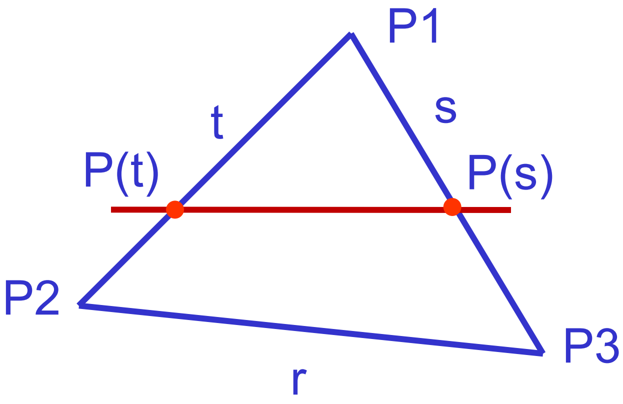
vedi formula (9').

Rasterizzazione di Triangoli per Z-buffer

Analogamente per ottenere le componenti colore corrette, una volta determinate le componenti interpolate si possono calcolare incrementalmente le relative coordinate parametriche [r', s', t'] sul triangolo schermo e da queste, utilizzando le relazioni inverse (2) e (3) che danno [r , s , t], calcolare le informazioni colore originali come:

Re = [1-(1-r)*t-r*s]*Re1 + [(1-r)*t]*Re2 + [r*s]*Re3;
Ge = [1-(1-r)*t-r*s]*Ge1 + [(1-r)*t]*Ge2 + [r*s]*Ge3;
Be = [1-(1-r)*t-r*s]*Be1 + [(1-r)*t]*Be2 + [r*s]*Be3;

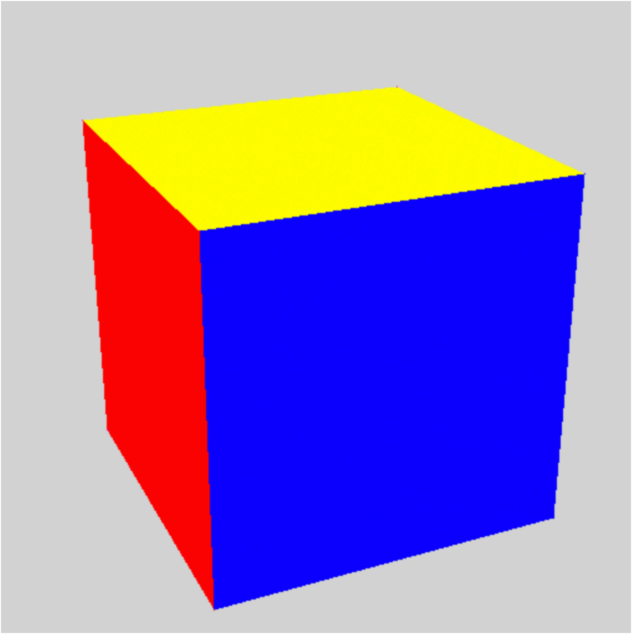
Coord. parametriche



P(t) = (1-t) P1 + t P2
P(s) = (1-s) P1 + s P3

P(r,s,t)=(1-r) P(t) + r P(s)=
=(1-r) [(1-t) P1 + t P2] + r [(1-s) P1 + s P3]

Esempio



Vedi:
HTML5_2d_3/persp_cube_zbuffer_color.html e .js

Esempio

Vedi:
HTML5_2d_3/persp_cube_zbuffer_texture_uv.html e .js

Esercizio 3

Si propongono i seguenti esercizi da realizzare modificando il codice persp_cube_zbuffer_texture_uv.js nell'archivio messo a disposizione:

1. modificare l'associazione vertici delle facce del cubo e coordinate u,v texture/immagine
2. utilizzare come texture una immagine personale e la si applichi alle facce del cubo
3. texturare il cubo/mesh dando le coordinate u,v texture/immagine relative ai vertici delle facce per realizzare uno sky-box (si usi l'immagine sky-box.png); successivamente si visualizzi la scena muovendo la camera dentro il cubo (attenzione non abbiamo il clipping)
4. modificare il codice per sperimentare la proiezione prospettica senza correzione (vedi immagine iniziale di queste slide)

Lo si chiami persp_zbuffer_texture.js

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Giulio Casciola
Dip. di Matematica
giulio.casciola at unibo.it