INFORMATICA GRAFICA – SSD ING-INF/05
Sistemi di elaborazione delle informazioni
a.a. 2007/2008

CAP 5. Pipeline grafica

Introduzione

- Pipeline grafica:= **sequenza di trasformazioni** che i dati grafici devono *attraversare* per il *rendering*
- Foley, van Dam, Feiner, Hughes. *Computer Graphics, Principles and Practise.* Addison-Wesley
- Standard grafico **PHIGS** per il 3D
- Obiettivo: scelta dei ***modelli di vista*** (insieme di parametri) per generare *immagini realistiche* o *disegni tecnici*

L’operazione di proiezione

- **Proiezione** : e’ una trasformazione geometrica con il dominio in uno spazio di dimensione *n* ed il codominio in uno spazio di dimensione *n-1* (o minore):

$$f : \mathrm{R}^n \rightarrow \mathrm{R}^m \quad m < n$$

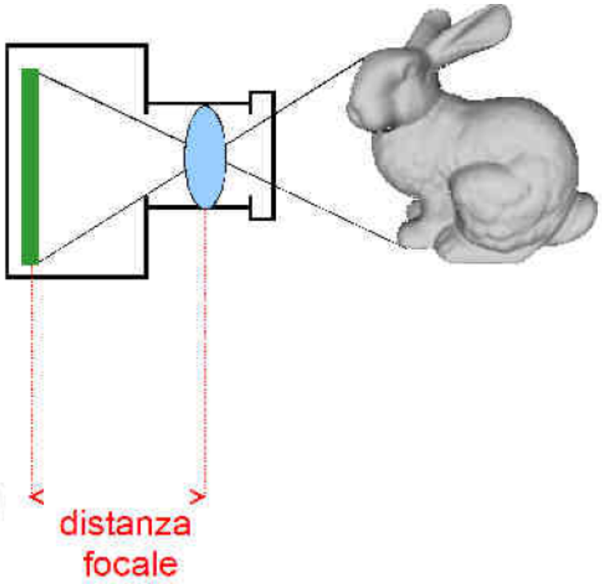
- In Computer Graphics le trasformazioni di proiezione utilizzate sono quelle dallo spazio 3D (il mondo dell’applicazione) al 2D (la superficie del dispositivo di output)

Il mondo in 3D della visione umana

- Il processo di formazione dell'immagine in un sistema ottico
- I raggi di luce riflessi dalla scena raggiungono l'**occhio** dell'osservatore e sono intercettati dalla retina
- La **retina** e' una piccola porzione di superficie sferica: approssimazione con il piano tangente
- L'immagine percepita dai recettori luminosi e' trasmessa

Il mondo 3D nei calcolatori

- Modello concettuale fine anni 70, **ACM Siggraph**
- La proiezione dei punti e’ derivata **algebricamente**
- Viene applicata una trasformazione lineare di **rango due**
- Viene **mappato** uno spazio 3D in uno spazio 2D
- Si usano **coordinate omogenee** trasformazione :=matrice * vettore

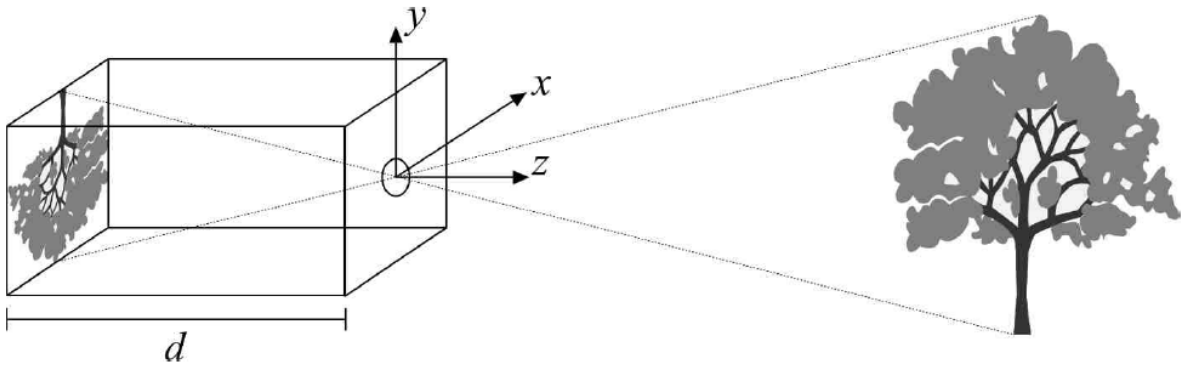


La macchina fotografica virtuale

- Se il rendering consiste nel creare una vista allora:
  - La definizione del modello nel mondo dell'applicazione è **indipendente** dalla posizione di osservazione della scena (gli oggetti nel mondo reale sono indipendenti dalle fotografie scattate loro);
  - In un sistema grafico le funzioni destinate alla **modellazione** ed al **posizionamento** della macchina fotografica sono **distinte e separate**.

Un tipo semplice di proiezione: la macchina fotografica virtuale

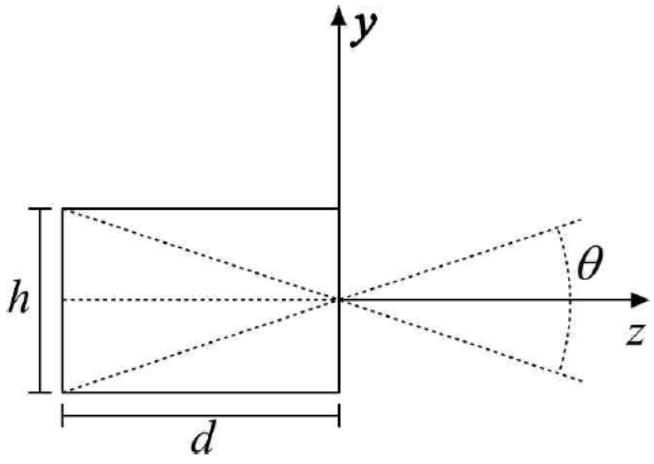
- La metafora utilizzata per descrivere le relazioni scena/osservatore è quella della macchina fotografica virtuale (***synthetic camera***).



La macchina fotografica virtuale

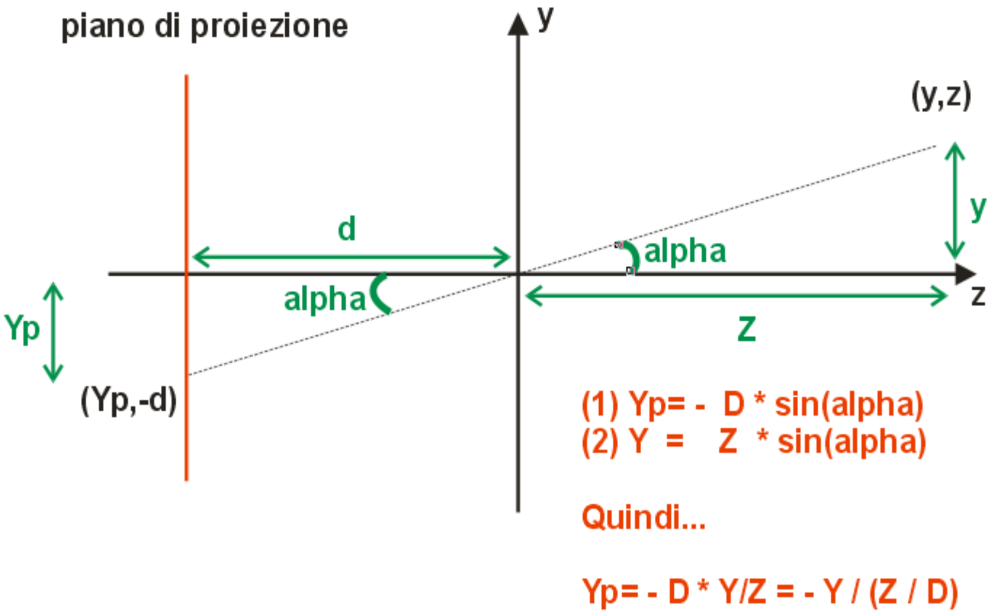
- La macchina fotografica virtuale è costituita da un parallelepipedo in cui la faccia anteriore presenta un foro di dimensioni infinitesime (**pinhole camera**) e sulla faccia posteriore si formano le immagini;

- Immagini nitide, nessun problema di luminosità
- L'**angolo di vista** può essere modificato variando il rapporto tra la distanza focale (*d*) e la dimensione del piano immagine (h).



La macchina fotografica virtuale

- Il generico punto **P=(x,y,z)** della scena ha sul piano immagine coordinate **Pp=(x_p,y_p,-d)**.

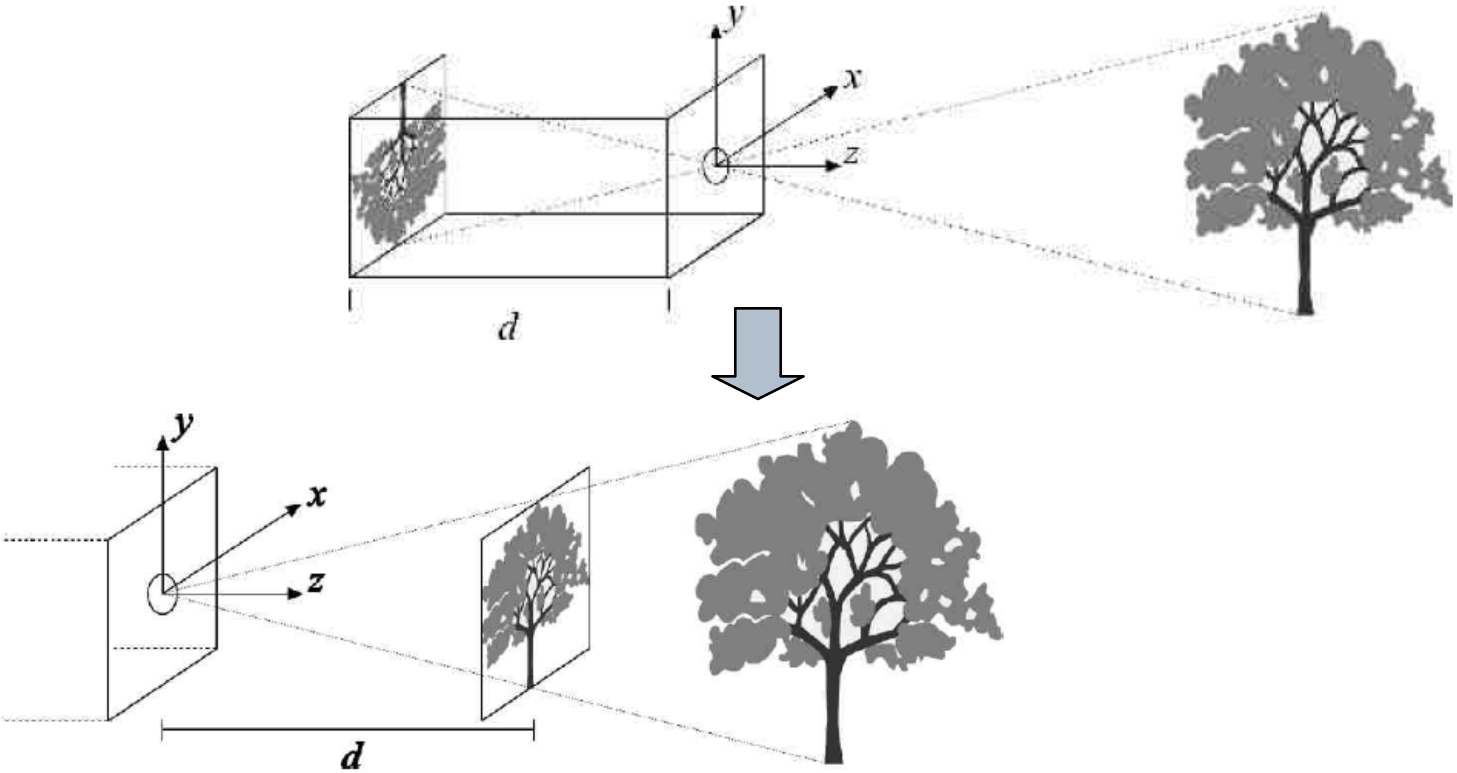


$$x_p = -\frac{x}{z/d} \quad y_p = -\frac{y}{z/d}$$

IMPORTANTE: La trasformazione non è lineare, non è affine, non è reversibile! Provare ad invertire il sistema, la componente Z non puo' essere determinata.

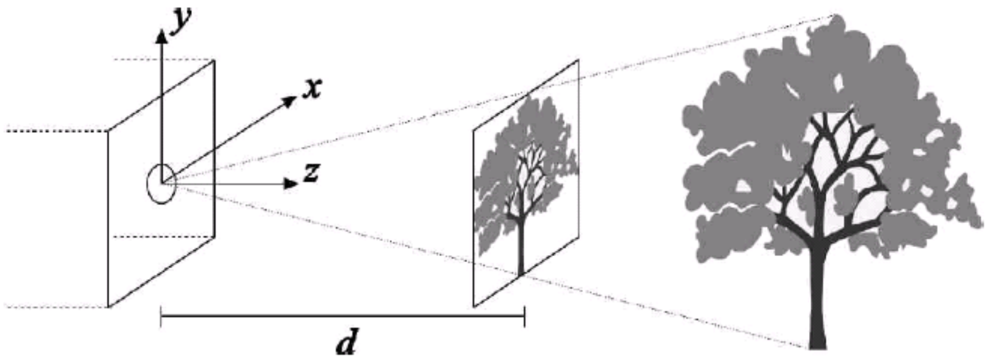
La macchina fotografica virtuale

- Per convenzione si assume l'esistenza di un **piano immagine** tra la scena ed il centro di proiezione



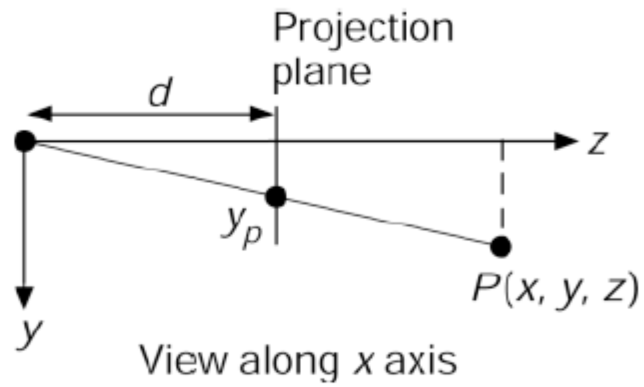
La macchina fotografica virtuale

- In questo caso:



$$\frac{z}{d} = \frac{y}{y_p} \qquad y_p = \frac{y}{z/d}$$

Rispetto a prima cambia solo il segno!



La macchina fotografica virtuale

- La matrice di trasformazione prospettica in coordinate **omogenee**:

$$M_{\text{per}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix}$$

La macchina fotografica virtuale

- La matrice di trasformazione prospettica in coordinate **omogenee**:

$$\begin{bmatrix} x \\ y \\ z \\ z/d \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} \dfrac{x}{z/d} \\ \dfrac{y}{z/d} \\ d \end{bmatrix}$$

Determinante nullo … *tutti i punti avranno* ***3a coordinata =d*** !

Modello di vista

- **Le proiezioni nel caso generale sono molto piu' complesse**

- In tre dimensioni si devono specificare
  1. un **view volume** (volume di vista), cioè la parte di spazio che contiene gli oggetti che andranno visualizzati
  2. una **proiezione** su di un certo piano
  3. una **window** su questo piano
  4. e la **viewport** sul dispositivo di uscita

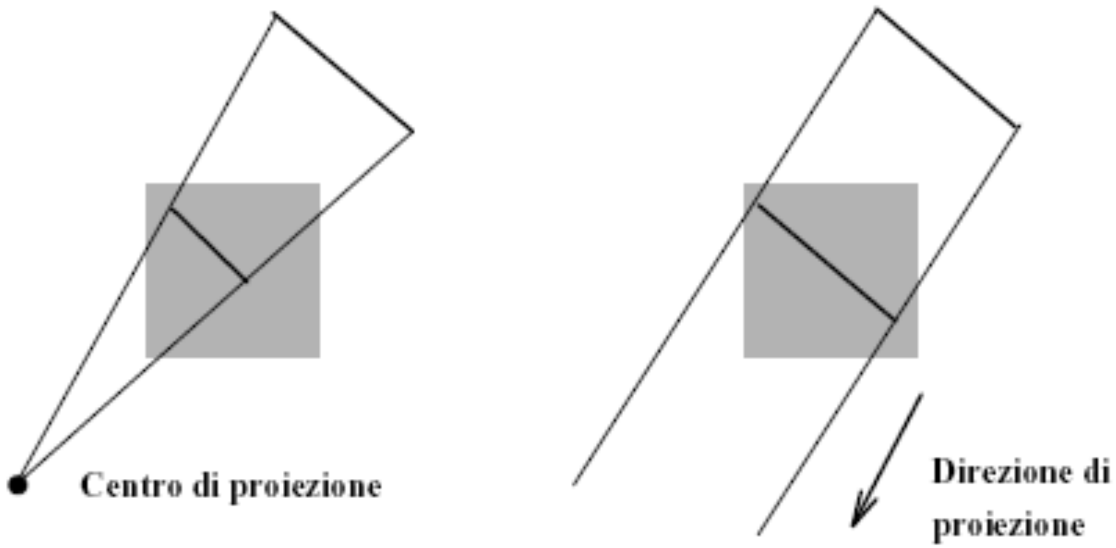
- ***4 parametri di vista*** definiscono completamente la scena

- Sono specificati in coordinate mondo **WC**

- Un **modello di vista** e' l'insieme di valori attuali dei parametri di vista

Le Proiezioni

- La proiezione di un punto 3D è definita come l'intersezione di una linea retta, detta **raggio proiettore**, che emana dal **centro di proiezione** (un punto che può essere all'infinito) e passa dal punto da proiettare, con il **piano di proiezione**.



Proiezione prospettica e parallela di un segmento.

Le Proiezioni

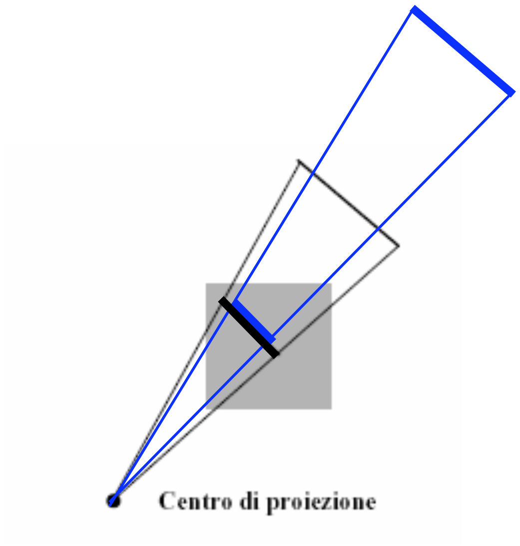
- La classe di proiezioni che tratteremo e’ quella delle **proiezioni planari geometriche**:
  - la superficie di proiezione è **piana** ed i raggi proiettori sono **linee rette**.
- Vi sono proiezioni che non appartengono a questa classe - molte proiezioni cartografiche sono **non planari** e/o **non geometriche**
  - per esse non vale che la proiezione di un segmento è ancora un segmento.

Le Proiezioni

- Le proiezioni planari geometriche si dividono in due classi principali

  Proiezioni Prospettiche

  - la distanza fra centro di proiezione e piano di proiezione è finita
  - il **centro di proiezione** (**COP**) è un punto in coordinate omogenee (x, y, z, 1)
  - creano una vista della scena *realistica*
  - oggetti che si allontanano dal viewer appaiono sempre piu' piccoli

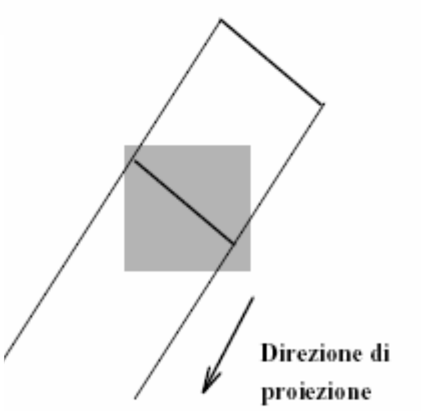


Le Proiezioni

- Le proiezioni planari geometriche si dividono in due classi principali:

  ### Proiezioni Parallele

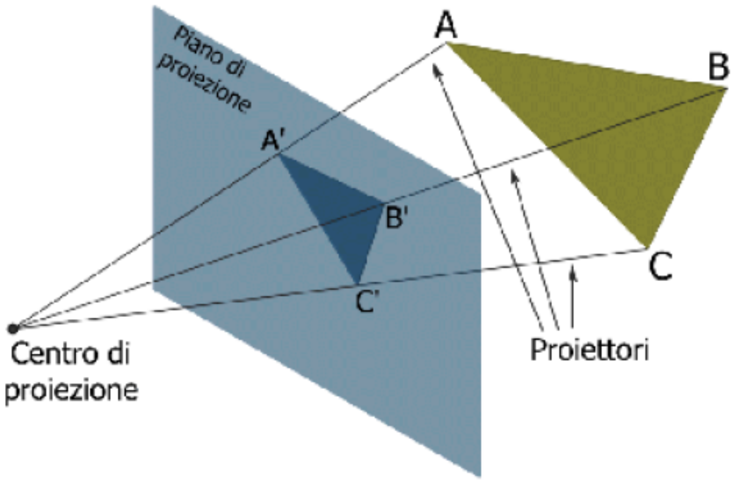
  - la distanza fra centro di proiezione e piano di proiezione è infinita
  - **DOP** (**Direzione di proiezione**) è un vettore ==>
    la differenza fra due punti (x, y, z, 1) - (x0, y0, z0, 1) = (a, b, c, 0)
  - gli oggeti **non** cambiano dimensione sullo schermo allontandoli o avvicinandoli al piano di proiezione.



Proiezioni Prospettiche - Caratteristiche

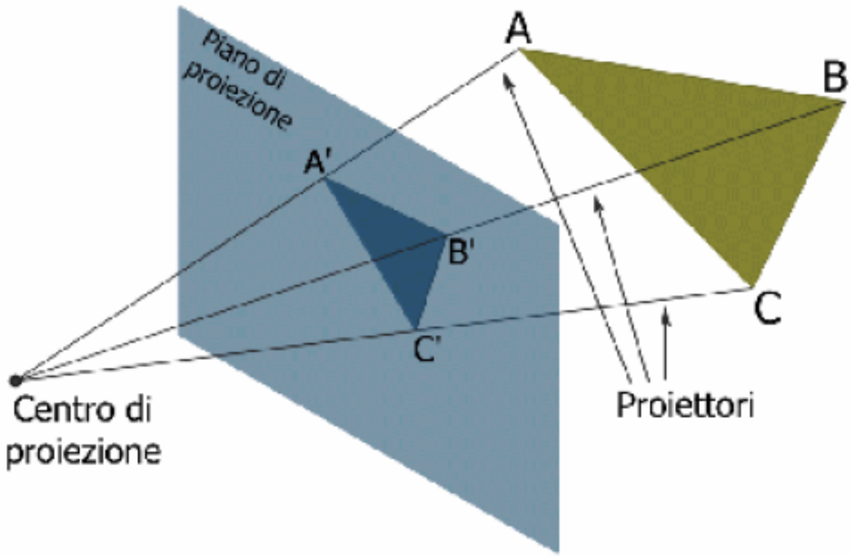
- Questo tipo di proiezioni, il più diffuso, corrisponde a quanto accade in una macchina fotografica/occhio umano
- E' più realistica della parallela in quanto riproduce la **visione reale**
- Gli oggetti appaiono piu **piccoli** al crescere della distanza dal piano di proiezione
- la proiezione è definita per mezzo di un insieme di rette di proiezione

  (i **proiettori**) aventi origine comune in un centro di proiezione, passanti per tutti i punti dell'oggetto da proiettare ed intersecanti un piano di proiezione.



Proiezioni Prospettiche - Caratteristiche

- le linee parallele che **non sono parallele al piano di proiezione** danno origine a linee proiettate che convergono verso un punto al finito nel piano di proiezione, detto **punto di fuga**

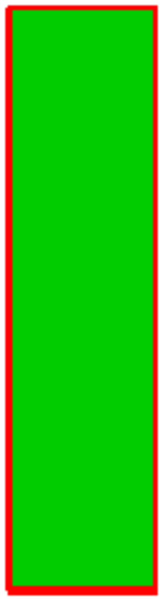


- gli **angoli** fra le linee non sono mantenute dalla proiezione, tranne che gli angoli formati da due linee entrambe parallele al piano di proiezione

- le **lunghezze**, non sono misurabili direttamente

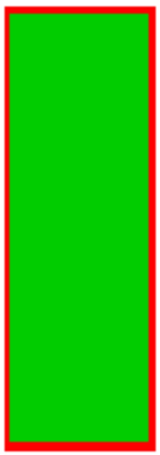
Proiezioni prospettiche. Caratteristiche

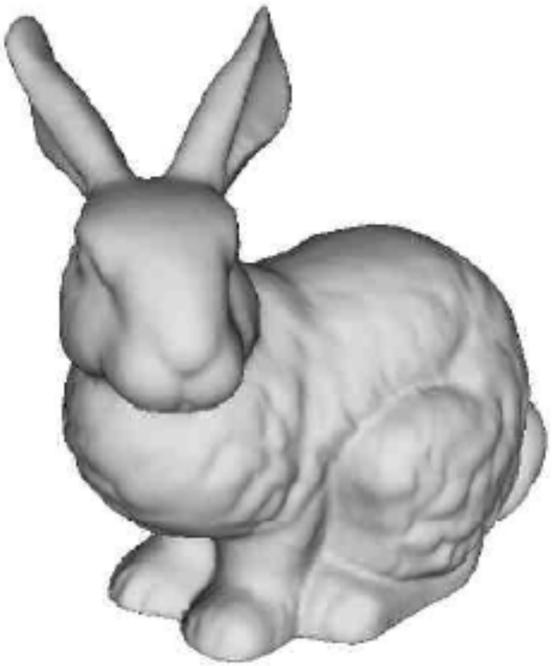
- Al variare della distanza focale (d)



Proiezioni prospettiche. Caratteristiche

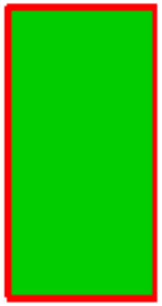
- Al variare della distanza focale (d)

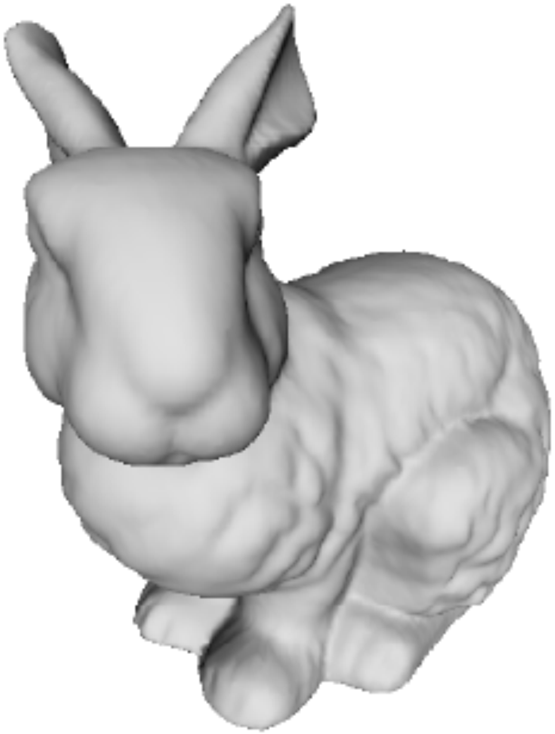




Proiezioni prospettiche. Caratteristiche

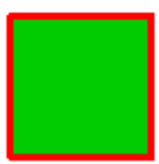
- Al variare della distanza focale (d)

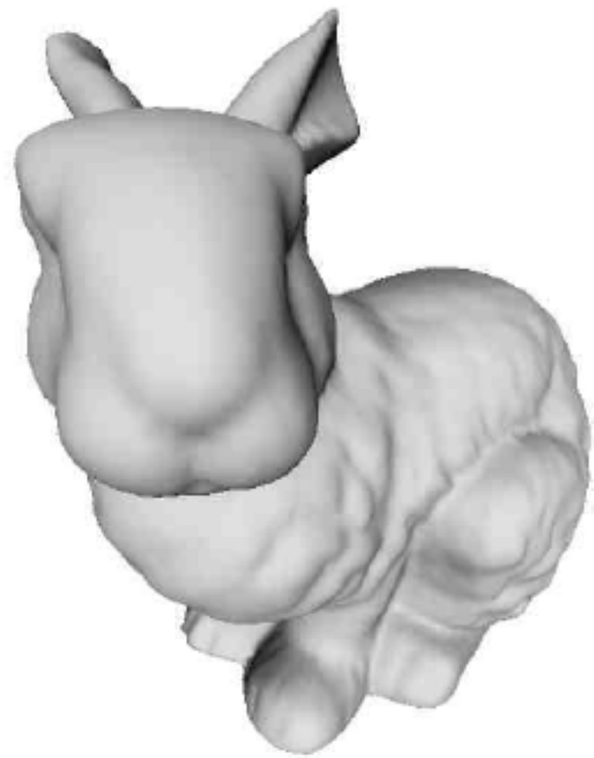




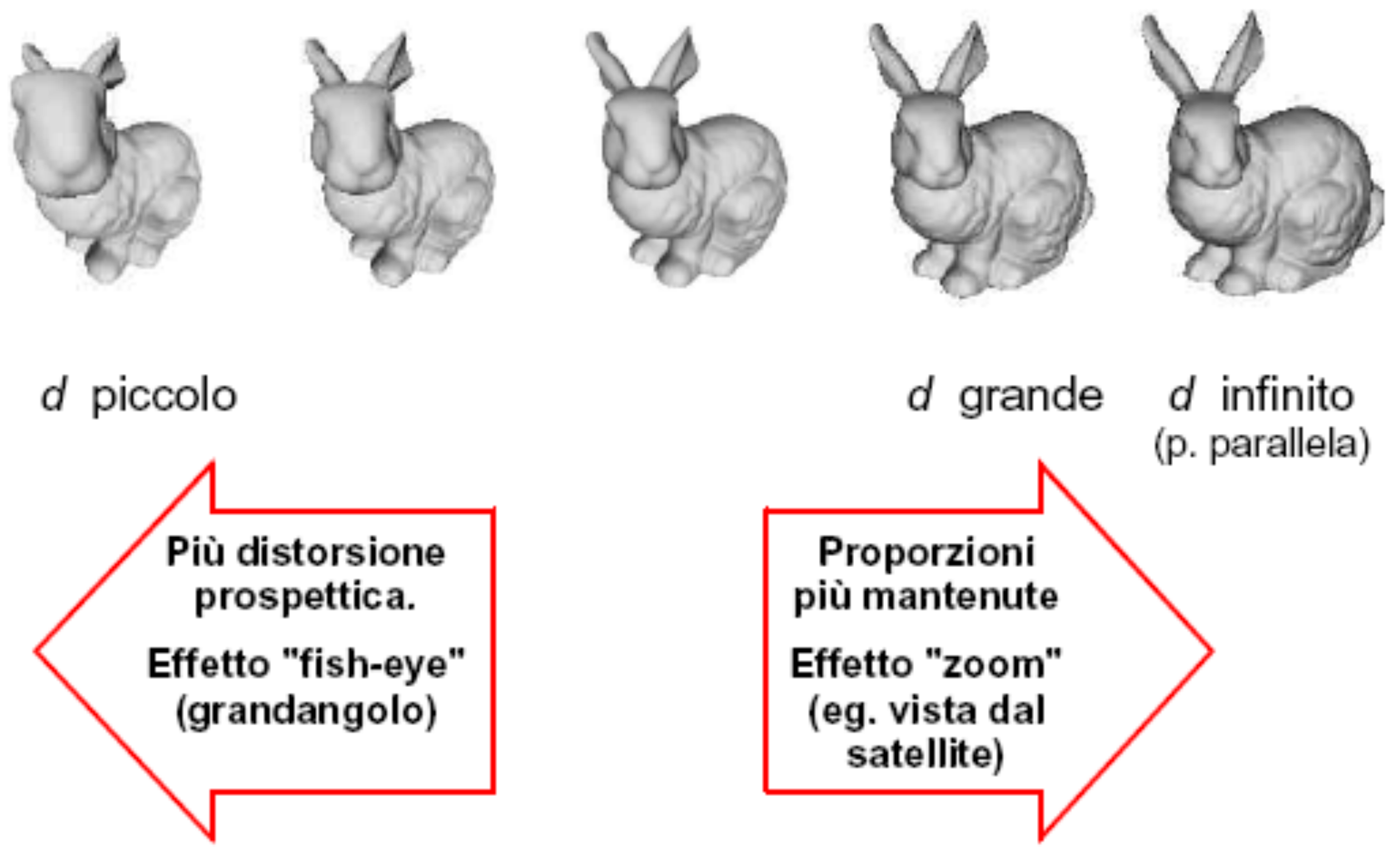
Proiezioni prospettiche. Caratteristiche

- Al variare della distanza focale (d)





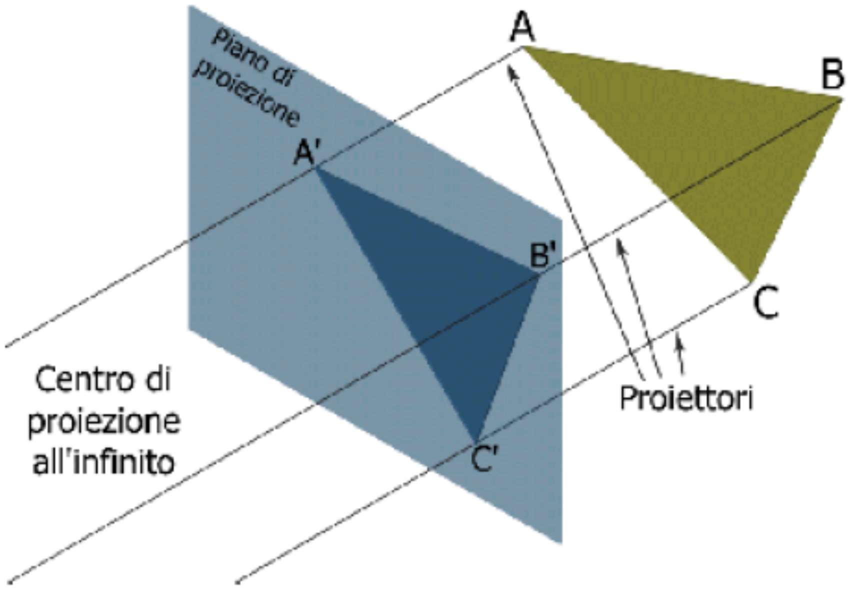
Proiezioni Prospettiche - Caratteristiche



In OpenGL il FOV (Field of View - radianti**) e' legato alla distanza focale: all'aumentare del valore diminuisce la distanza focale.**

Proiezioni Parallele - Caratteristiche

- Distanza infinita tra il centro ed il piano di proiezione
- Sono molto utili quando si voglia far sì che **linee parallele** nel modello tridimensionale rimangano tali nella proiezione
- Usate nella progettazione, essendo **distanze ed angoli** di linee parallele al piano di proiezione sono misurabili direttamente dal disegno - la proiezione non le altera!



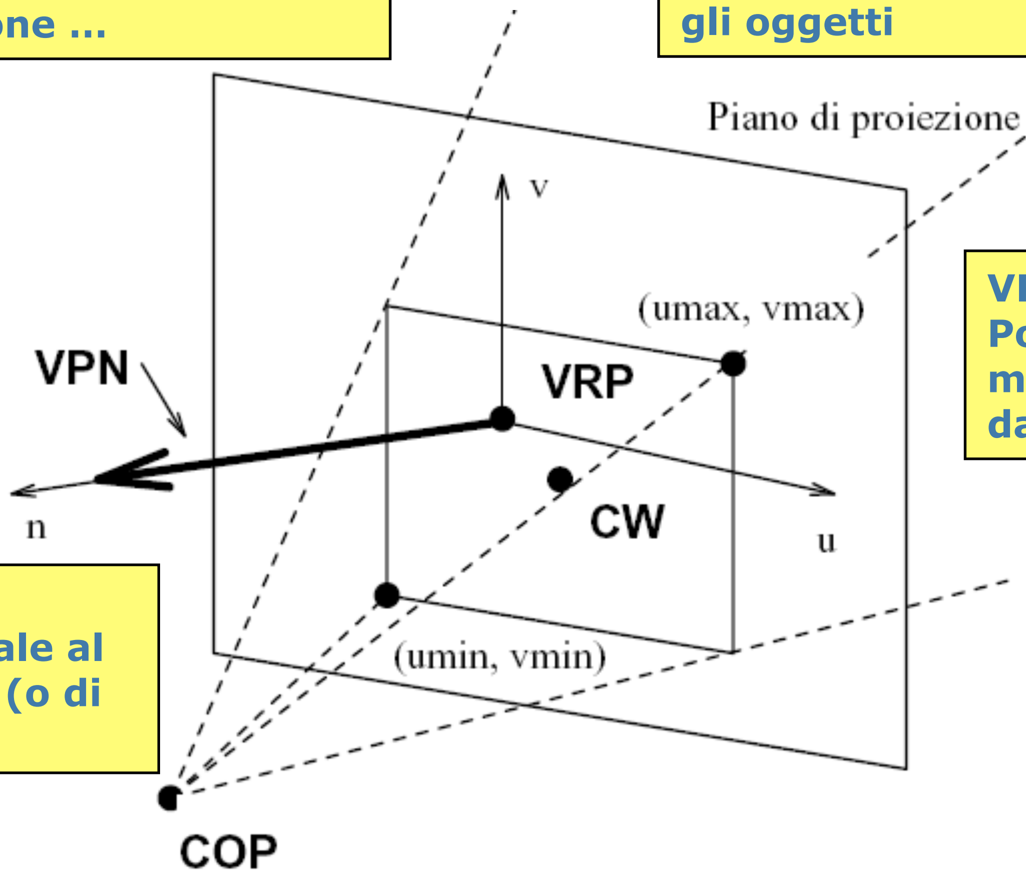
Matematica delle Proiezioni

- *Definisco il tipo di proiezione:*
  - Per le proiezioni prospettiche
    - **COP** (*Center of Projection):* e’ il punto comune delle rette proiettanti la scena
  - Per le proiezioni parallele
    - **DOP** (*Direction of Projection): individua la direzione delle rette proiettanti*
- *Definisco il piano di proiezione (o* piano di vista*)*
  - **VRP** (*View Reference Point*): e’ il punto di riferimento della vista. Il punto mirato dall’osservatore (**in OpenGL il target**) - *si trova sul piano di proiezione!*
  - **VPN** (*View Plane Normal)*: e’ la normale al piano di vista
- *Definisco la “rotazione” della camera, come se ruotassi la camera di un certo angolo (ad esempio, voglio fotografare la scena capovolta!)*
  - **VUP** = *View Up Vector* (e’ il vettore di direzione dell’immagine)

Esempio: Proiezioni **Prospettiche**

V:= Per adesso consideralo come il View Up Vector (VUP) … definisce la rotazione …

View Plane (o Projection Plane) … dove sono proiettati gli oggetti



VRP (View Reference Point) := Punto mirato dall’osservatore

VPN (View Plane Normal) … la normale al piano di proiezione (o di vista)

COP (Center Of Projection) … da dove partono i raggi

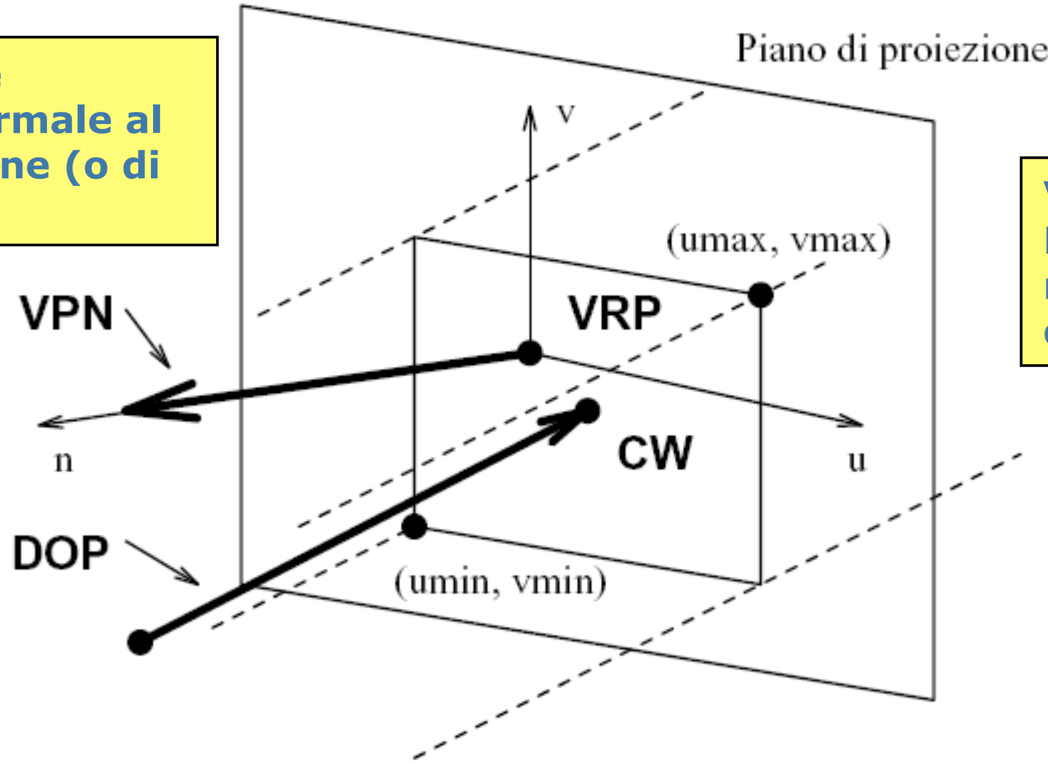
Esempio: Proiezioni **Parallele**

V:= Per adesso consideralo come il View Up Vector (VUP) … definisce la rotazione …

View Plane (o Projection Plane) … dove sono proiettati gli oggetti

VPN (View Plane Normal) … la normale al piano di proiezione (o di vista)

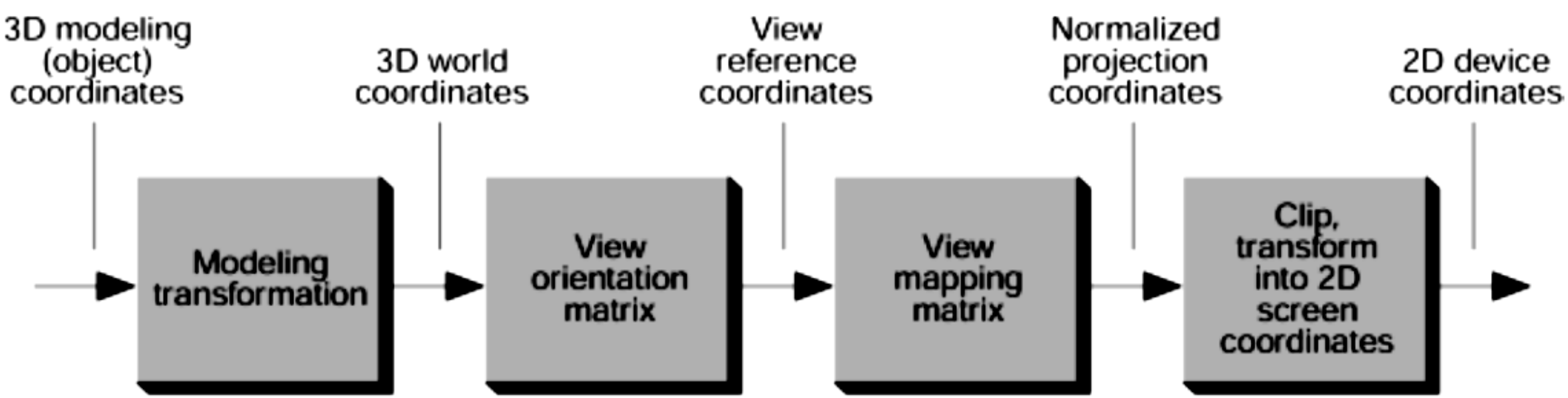
VRP (View Reference Point) := Punto mirato dall'osservatore



DOP (Direction Of Projection) …
"come" (direzione) si proiettano gli oggetti geometrici

Sistemi di Coordinate

- I sistemi di coordinate utilizzati nella Pipeline 3D in **PHIGS**:
  - Coordinate di modellazione **MC**
  - Coordinate mondo **WC**
  - Coordinate di vista **VRC**
  - Coordinate di proiezione normalizzate **NPC**
  - Coordinate di dispositivo **DC**

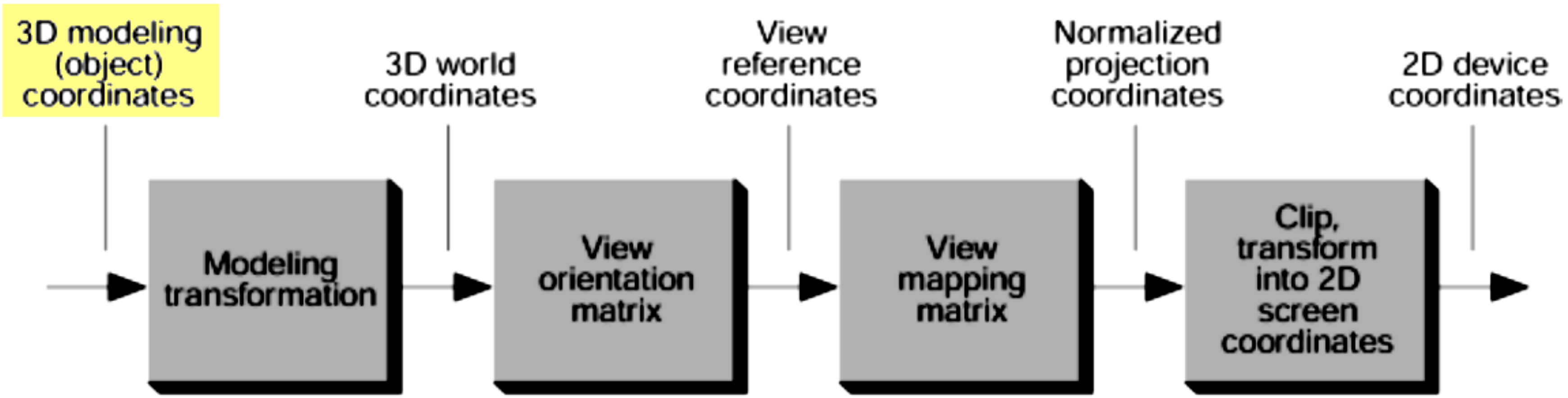


Sistemi di Coordinate: Coordinate di Modellazione (MC)

- Sono coordinate **locali** ad ogni struttura

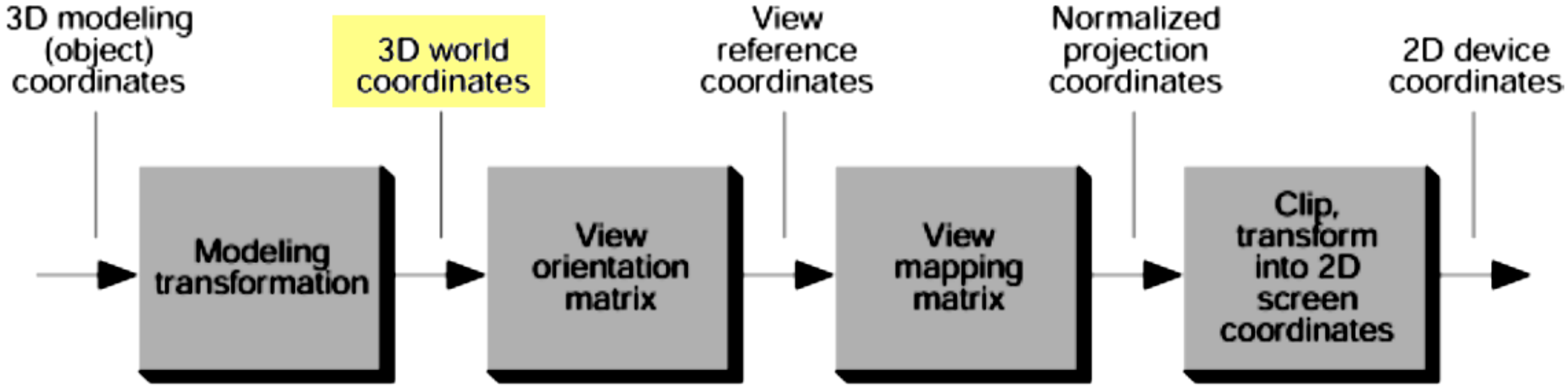
- Sono molto utili per la modellollazione quando si utilizza un **modello gerarchico**

- La visita **depth first search** (**DFS**) linearizzera' la gerarchia di strutture



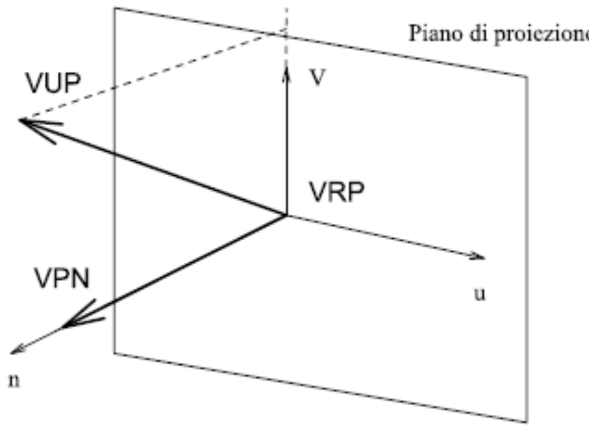
Sistemi di Coordinate: Coordinate Mondo (WC)

- Sono le coordinate **globali** della struttura
- Spesso coincidono con le coordinate locali della **radice** della rete gerarchica (la STRUCT!)
- Riferimento comune per ogni primitiva grafica della scena
- Utilizzate per definire **la posizione e orientamento camera fotografica** in un modello di vista

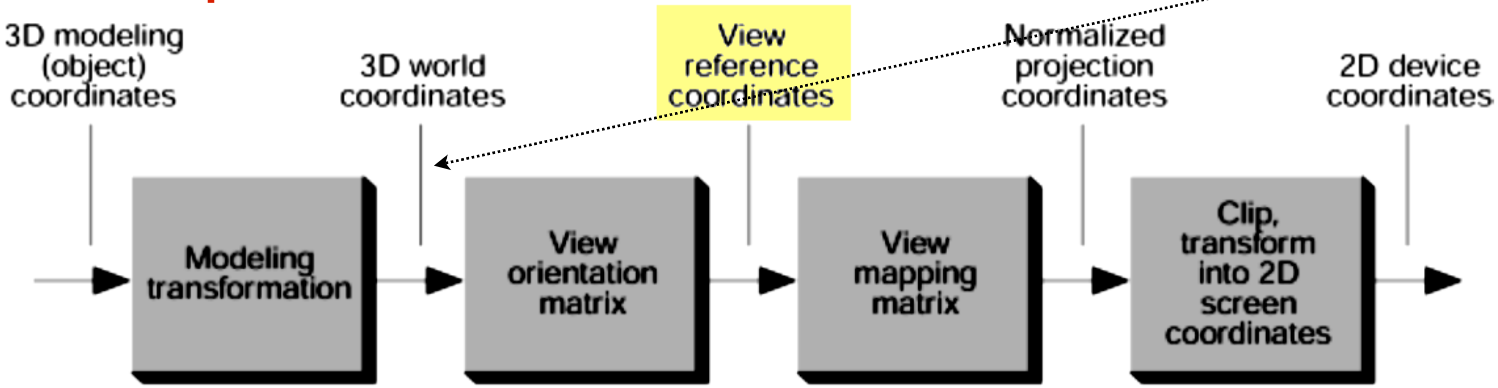


Sistemi di coordinate: Viewing Reference Coordinate (VRC)

- Origine coincidente con il *View Reference Point* (**VRP**)
- l'asse **n** (l'unico che non giace sul piano di proiezione) parallelo a *View Plane Normal* (**VPN**)
- Resta un grado di liberta, ovvero l'angolo che specifica l'orientamento dei restanti due assi:
  - *View Up Vector* (**VUP**) la cui proiezione sul piano di proiezione cade sulla parte positiva dell'asse v.
- Definisco i limiti della *finestra* 2d sul view plane in VRC, insieme alle distanze dei piani *front e back* paralleli al *view plane*
- **VRC serve per determinare il volume di vista!**



Il punto VRP ed i due vettori VPN e VUP sono specificati utilizzando il sistema di coordinate "mondo" (WC)



Approfondimento sul **volume di vista**

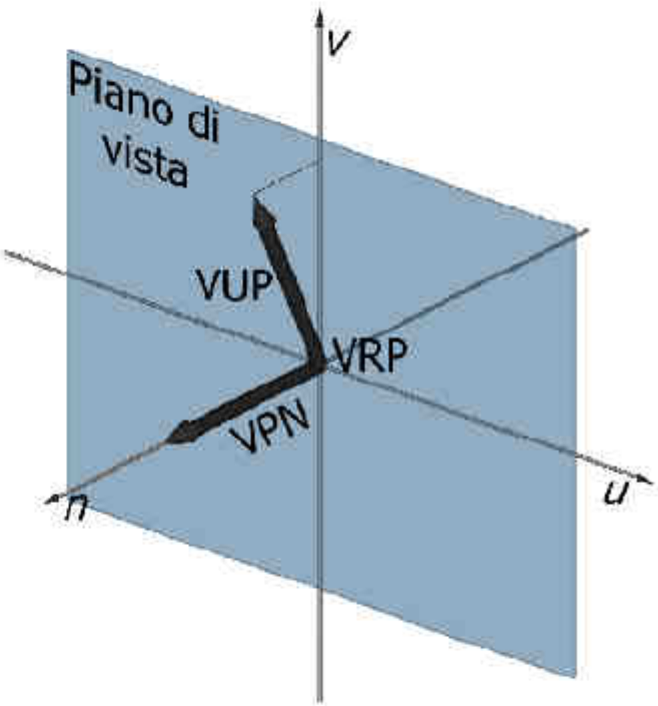
- Nella maggior parte delle applicazioni il volume di vista si definisce come **volume finito** al fine di:

  - scartare oggetti **troppo lontani** che sarebbero comunque quasi invisibili (con il solo l’effetto di rallentare inutilmente il rendering della scena);

  - evitare che oggetti **troppo vicini** al centro di proiezione invadano l’immagine nel caso in cui il punto di vista sia interno alla scena (es in OpenGL il front plane…)

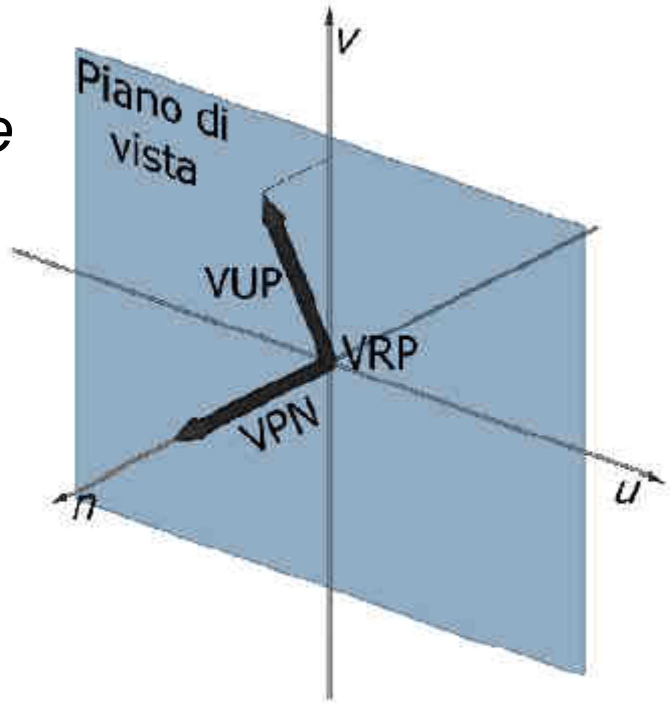
Il volume di vista

- Il piano di proiezione o di vista (*view plane*) è definito tramite un punto sul piano detto *view reference point* (VRP) ed un vettore normale al piano detto *view-plane normal* (VPN).



Il volume di vista

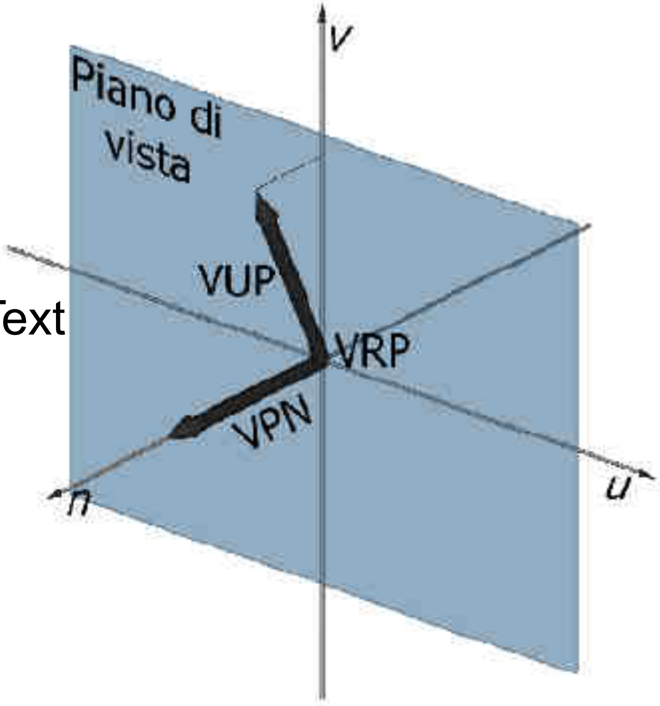
- Sul piano è definito un sistema di coordinate (u, v, n), indicato come sistema *viewing reference coordinate* (VRC), con origine in VRP.



Il volume di vista

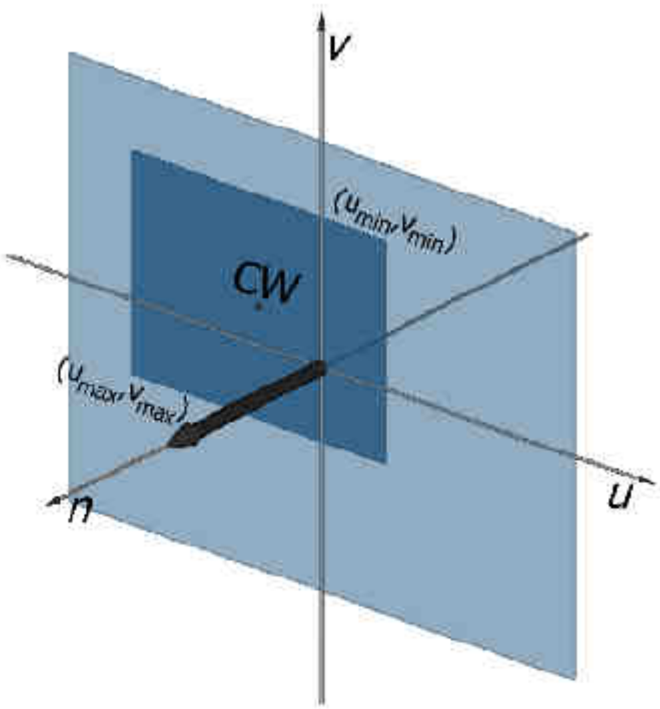
- Uno degli assi di VRC è definito dal VPN (l'asse *n*), un secondo asse (l'asse *v*) è definito dalla proiezione sul piano di un vettore detto *view up vector* (VUP) ed il terzo asse, *u*, è definito in maniera tale che i tre assi (*u*, *v*, *n*) formino un sistema di coordinate destrorso.

Text



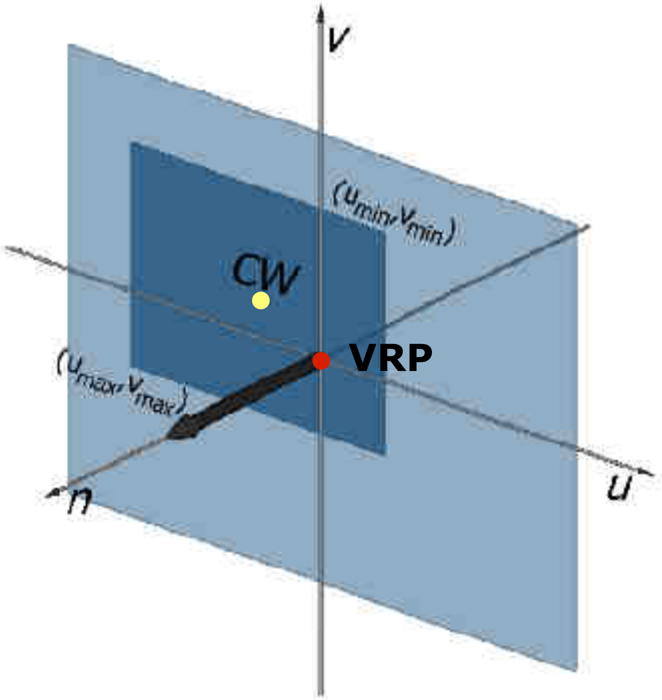
Il volume di vista

- A questo punto è possibile definire la window nel sistema VRC tramite le sue coordinate u_{min}, u_{max}, v_{min} e v_{max}



Il volume di vista

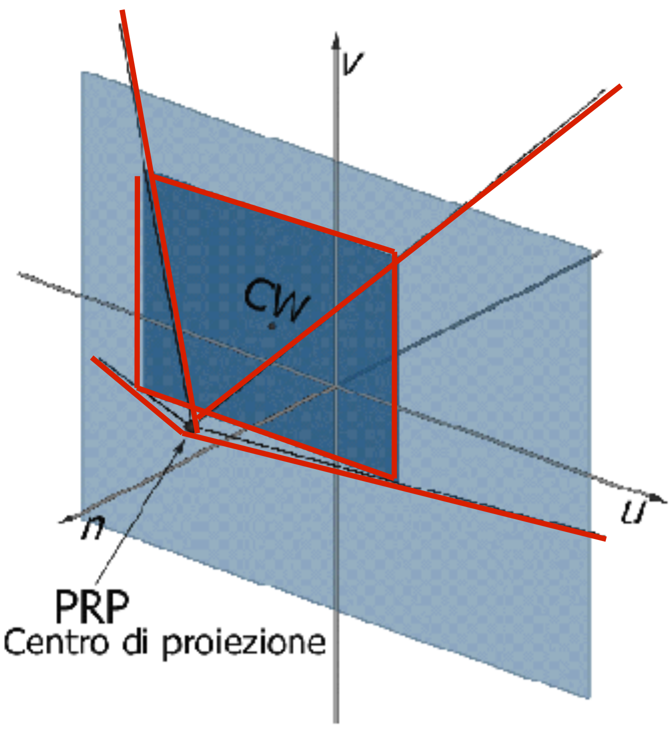
- La window non deve essere necessariamente simmetrica rispetto al VRP;
- Gli spigoli sono allineati con gli assi ed il centro (CW) è definito implicitamente dagli altri parametri.



Il volume di vista: caso prospettico

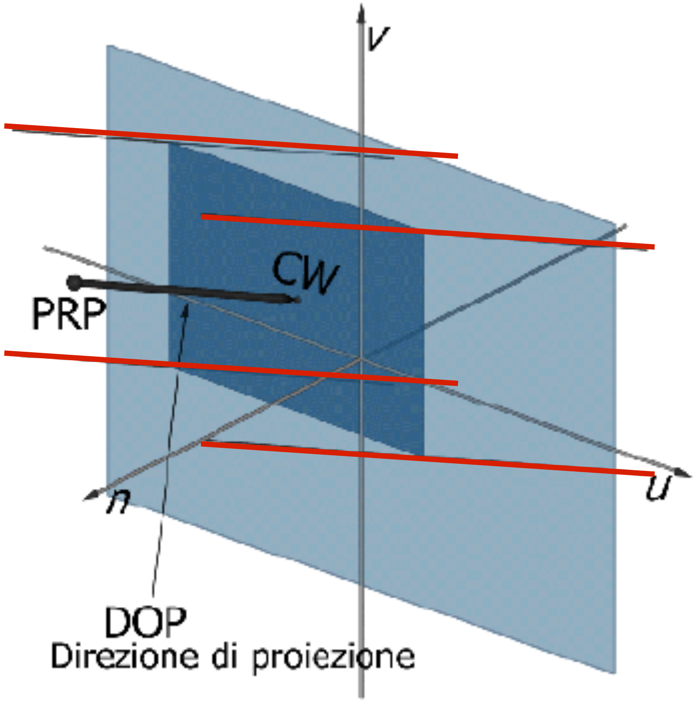
E' il COP trasformato in VRC

- Il centro di proiezione (prospettica) è definito dalla posizione del *projection reference point* (PRP);
- Il PRP viene definito in coordinate VRC anziché in WC (ne segue che le posizioni relative di PRP e VRP non variano al variare di VUP o VRP).



Il volume di vista: caso **parallelo**

- Anche la direzione di proiezione DOP (nella proiezione parallela) è definita dalla posizione del *projection reference point* (PRP)
- DOP è definita dal vettore PRP-CW

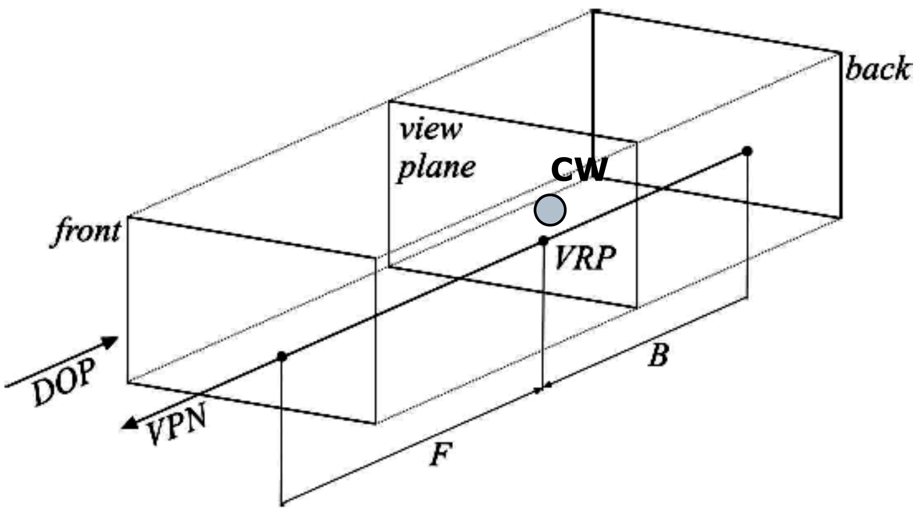


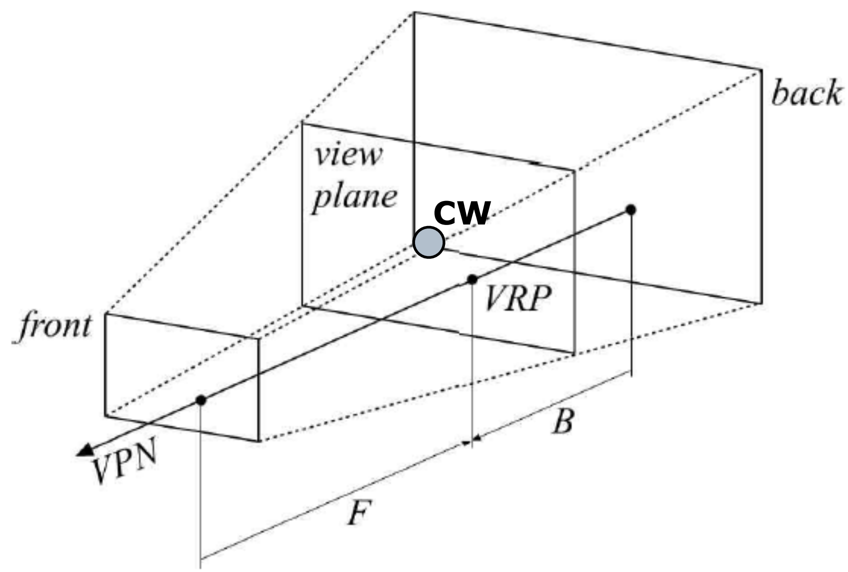
Il volume di vista

- Il volume di vista finito presenta sei facce:

  un **parallelepipedo** nel caso di proiezioni parallele;

  un **tronco di piramide** nel caso di proiezioni prospettiche.

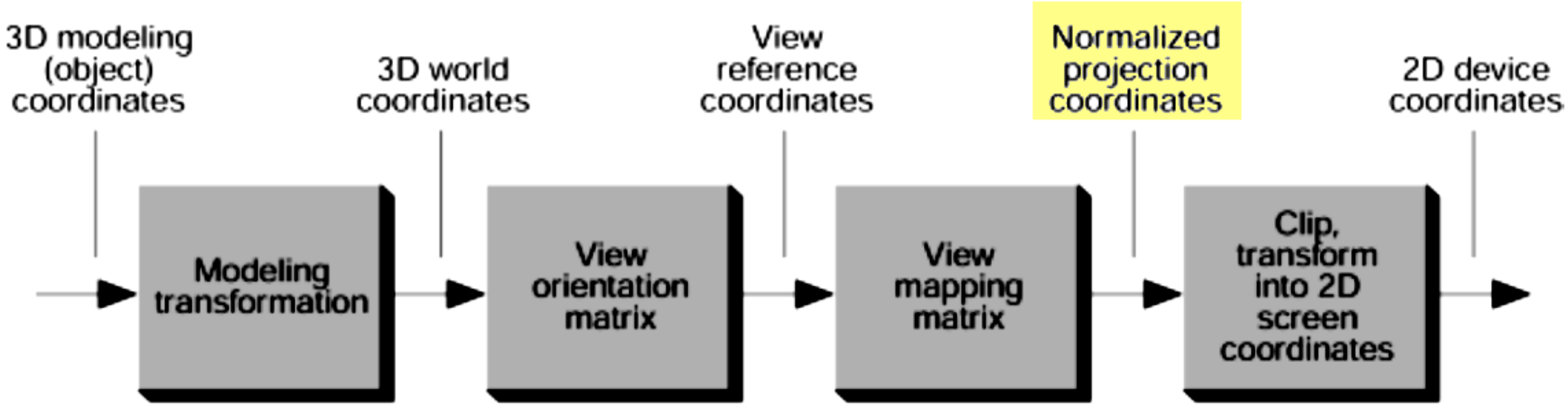




Nota, i volume di vista posso non essere retti.
Tramite la PIPELINE 3D li renderemo retti!

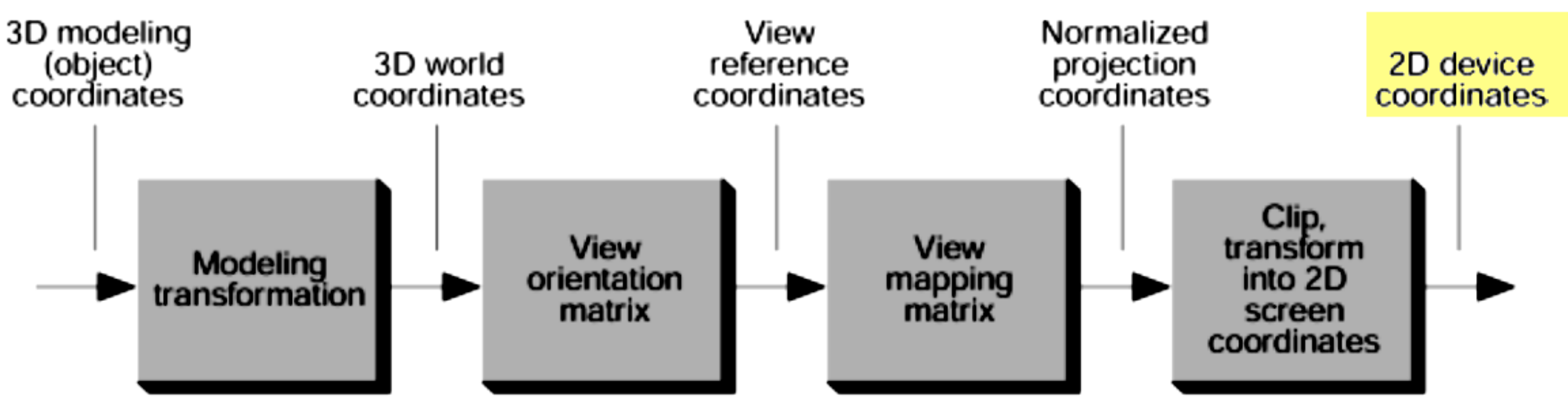
Sistemi di coordinate: Coordinate di proiezione Normalizzate (**NPC**)

- Utilizzate per produrre concretamente la **proiezione**
- **Facilitano il clipping** delle primitive che sono fuori dalla regione di interesse
- La terza coordinata del sistema NPC e’ la ***profondità prospettica***. Usata per calcolare l’occlusione relativa tra parti della scena
- La proiezione effettiva e’ ottenuta molto semplicemente **eliminando questa coordinata** (sia caso prospettico che parallelo)



Sistemi di coordinate: coordinate di dispositivo (DC)

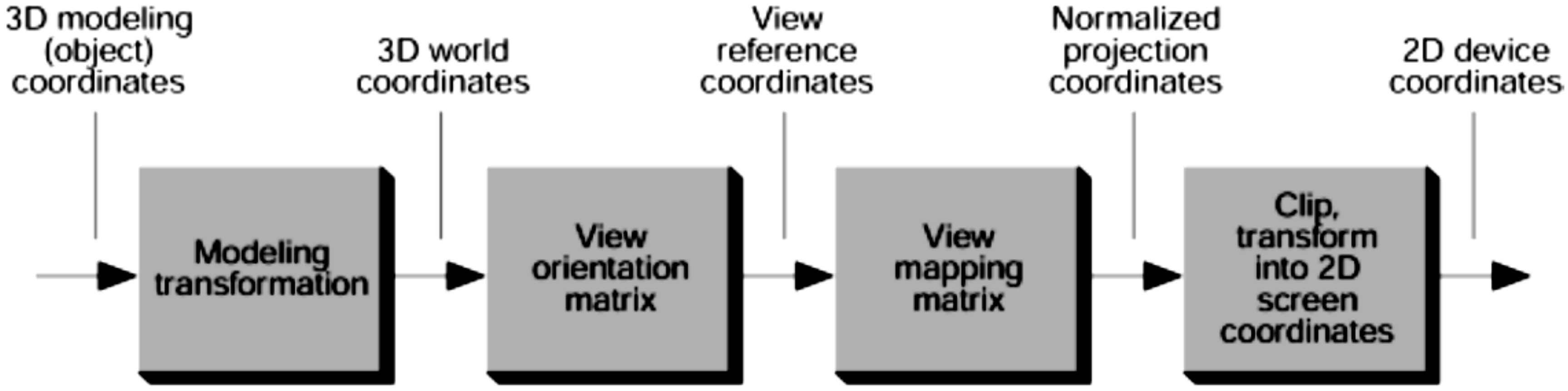
- Sono coordinate **discrete** 3D dipendenti dal **dispositivo**
- Sono pienamente 3D in **PHIGS** dove i dispositivi grafici sono considerati 3D
- Piu’ spesso un array 2D di reali (**z-buffer+frame buffer**) che memorizza il colore per ogni pixel + profondità per la rimozione parti nascoste



Come passare da un sistema di riferimento ad un altro

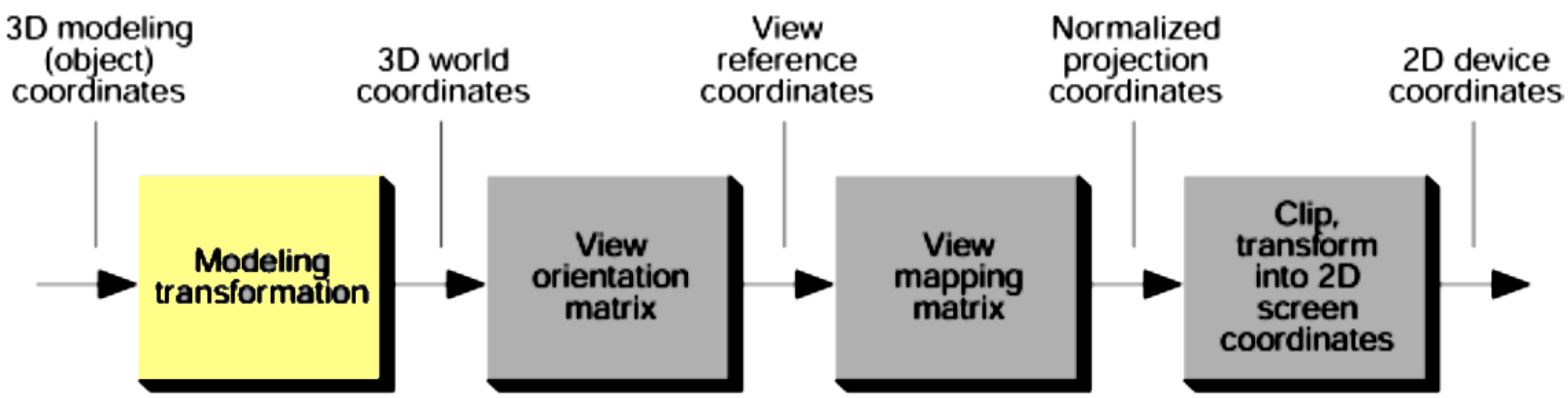
Come **opera** la Pipeline 3D:

- Visita della rete di strutture **MC->WC**
- Orientamento di vista (*view orientation*) **WC->VRC**
- Trasformazione di vista (*view mapping*) **VRC->NPC**
- Trasformazione di dispositivo **NPC -> DC**



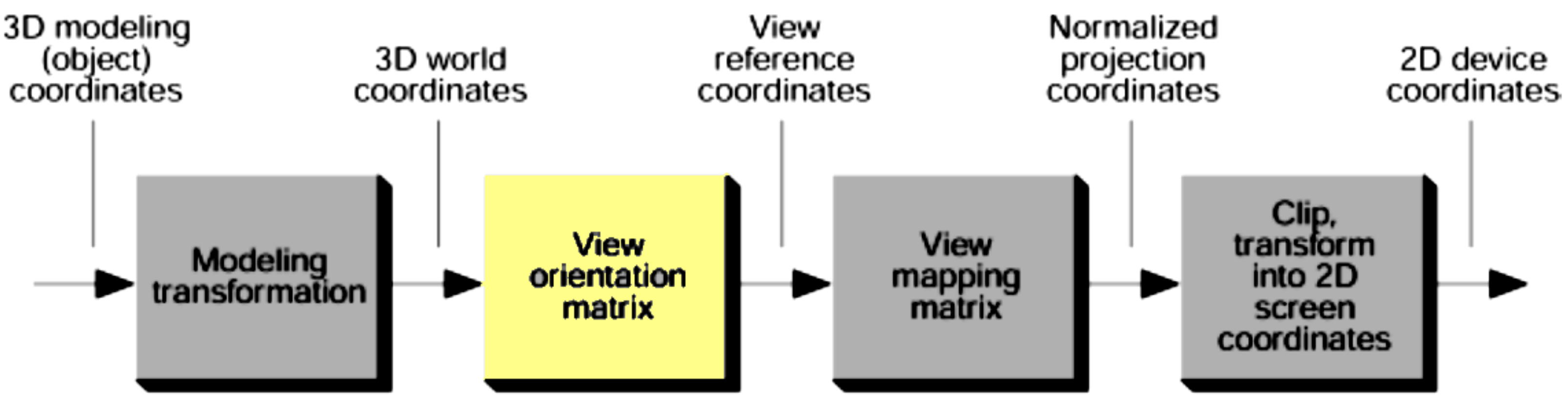
1. Visita della rete di strutture

- Molto semplice: trasforma le coordinate di modellazione **locali** alle sottostrutture alle coordinate mondo **globali**
- Si puo’ fare il **clipping** delle primitive esterne al volume di vista
- Equivalente ad una **visita in profondità**:
  - moltiplica ciascuna primitiva per la matrice di trasformazione corrente (CTM)



2. Orientamento di vista (**view orientation**)

- Obiettivo:
  1. portare il punto mirato dall'osservatore (VRP) nell'origine
  2. Cambiamento assi:
     1. **VPN** vada nell'asse *z*
     2. **v**:= la proiezione di **VUP** sul piano di vista vada nell'asse *y*
     3. il prodotto vettore di **VPN** X **v** vada in asse *x*



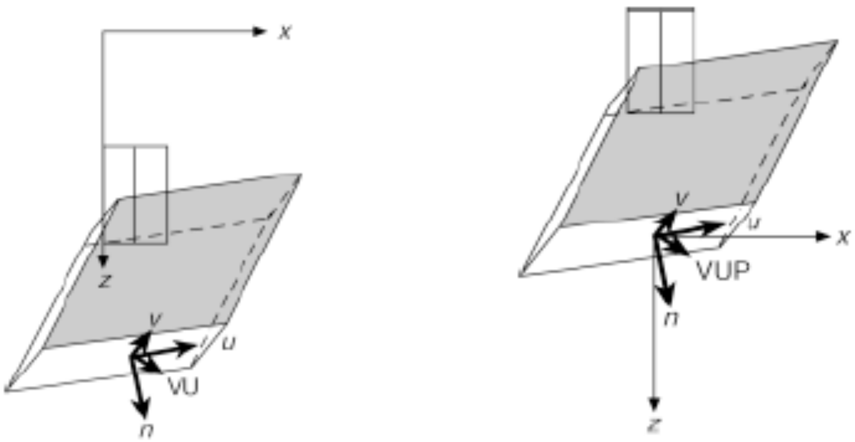
Da fare ...

2. Orientamento di vista (**view orientation**)

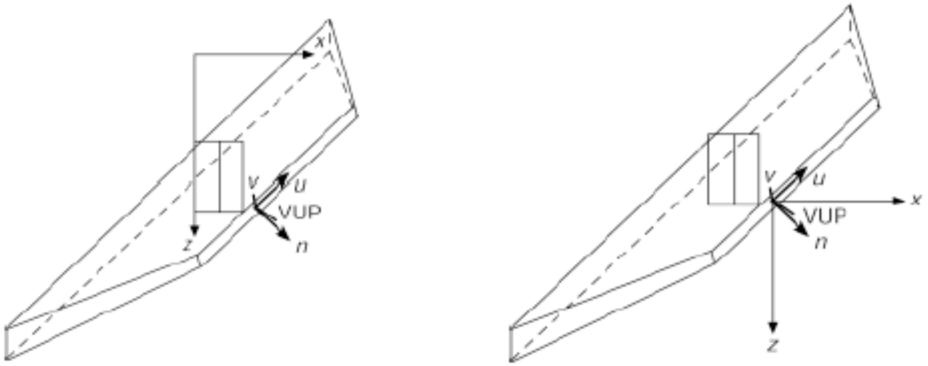
- Traslazione in modo che *"il punto mirato dall'osservatore (VRP) vada nell'origine"*

$$T = \begin{bmatrix} 1 & 0 & 0 & -vrp_x \\ 0 & 1 & 0 & -vrp_y \\ 0 & 0 & 1 & -vrp_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matrice T(-VRP)
Stessa caso prospettico e parallelo



Es. Caso parallelo



Es. Caso prospettico

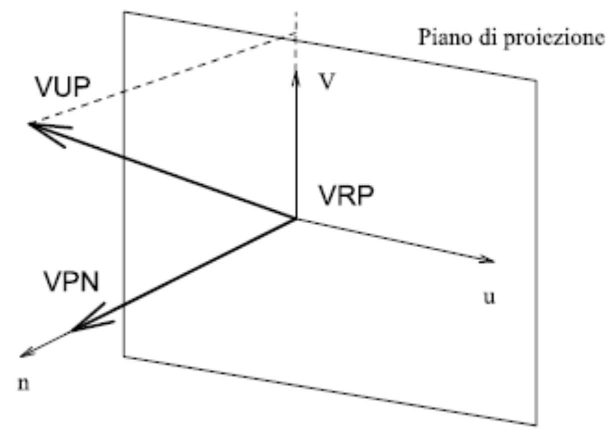
2. Orientamento di vista (**view orientation**)

- **Obiettivo:** ruotare il sistema in modo che..

  VPN vada in z

  V vada in y

  VPN X **V** vada in x



$$\begin{pmatrix} R_{ux} & R_{vx} & R_{nx} & 0 \\ R_{uy} & R_{vy} & R_{ny} & 0 \\ R_{uz} & R_{vz} & R_{nz} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Matrice R funzione di VPN e VUP
Non cambia fra prospettica e parallela

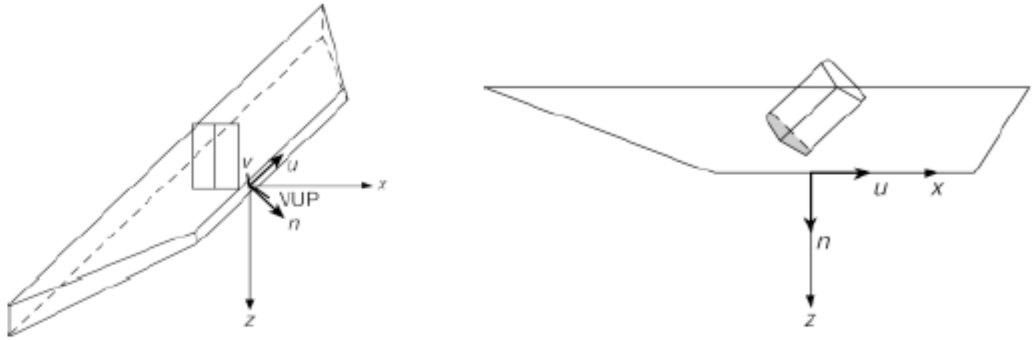
Come si "legge":

Il versore x (1,0,0,0) va in versore Ru = VUP x VPN … *normalizzare!*

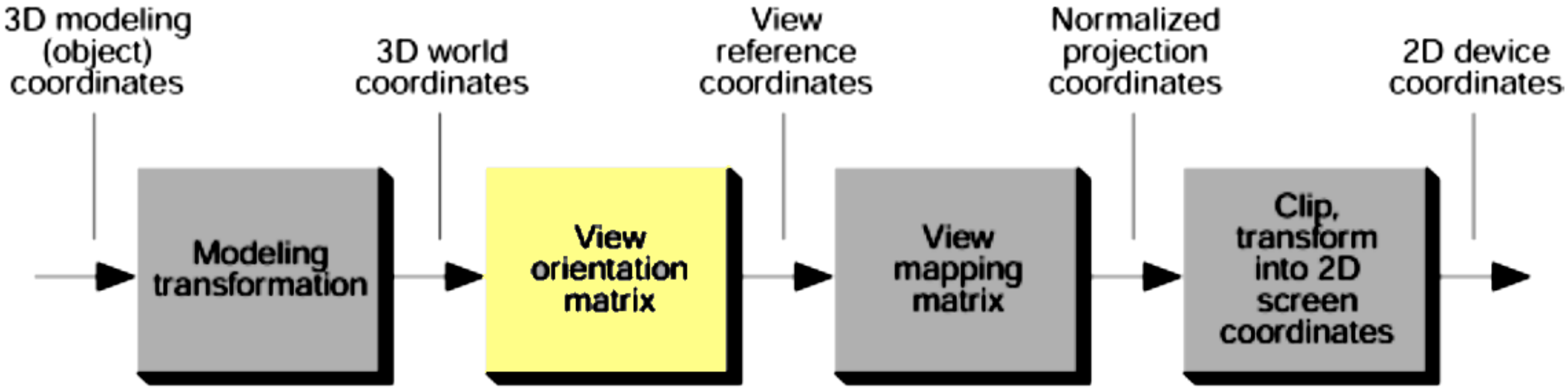
Il versore y (0,1,0,0) va in versore Rv = VPN x Ru … normalizzare*!*

Il versore z (0,0,1,0) va in versore Rn = VPN … normalizz*are!*

2. Orientamento di vista (**view orientation**)



Es. Caso prospettico



COMPLETATO!

3. Trasformazione di vista (**view mapping**)

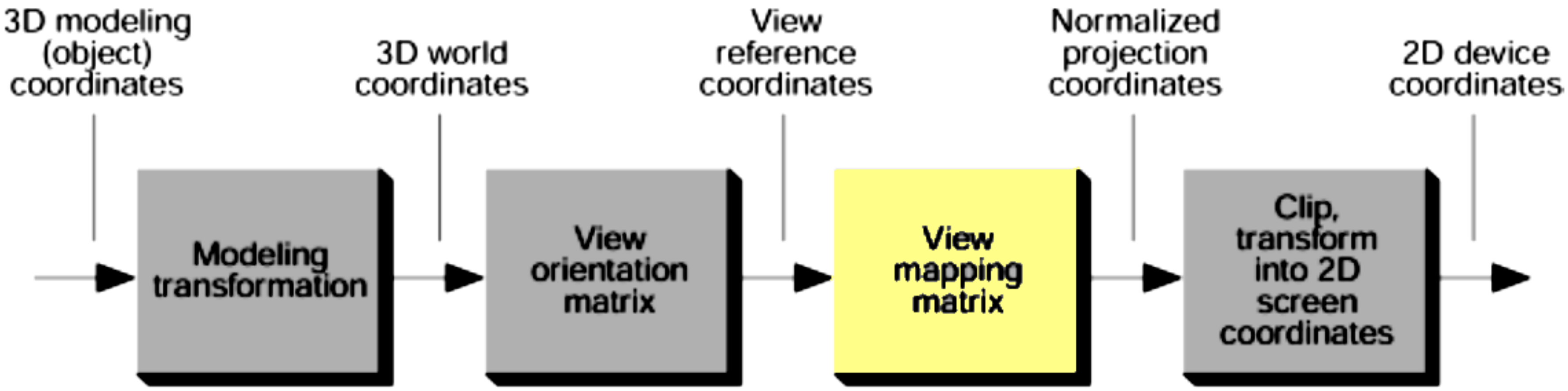
Obiettivo:

Trasformare il *volume di vista* VRC (ottenuto nel punto 1) nel **volume canonico**

PERCHE?

clipping delle primitive della scena per rimuovere le parti esterne al volume stesso. Risulta conveniente mappare i volumi di vista in volumi canonici che semplifichino la fase di clipping.

Il **volume canonico** si ottiene in modo **diverso** per il caso prospettico e caso parallelo…



Da fare …

3. Trasformazione di vista (view mapping) caso parallelo

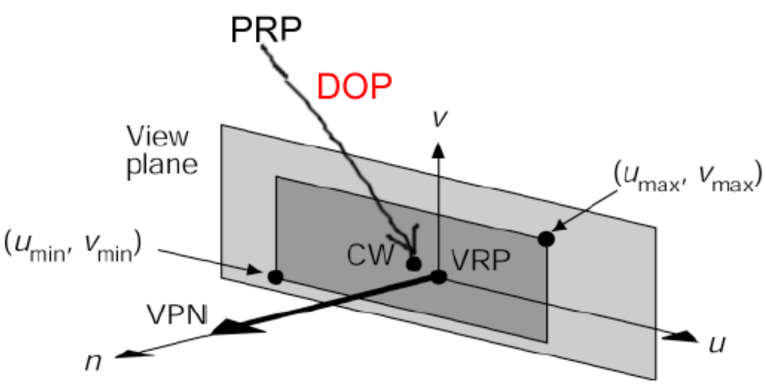
Obiettivo:

voglio trasformare il volume di vista (in generale non retto) nel volume canonico parallelo

[-1,1][-1,1]*[-1,0]*

Fase 3.par.1 Porto la Direction of Projection (DOP) a coincidere con l'asse z.

$$PRP = \begin{bmatrix} prp_u \\ prp_v \\ prp_n \\ 1 \end{bmatrix}, CW = \begin{bmatrix} \frac{u_{max}+u_{min}}{2} \\ \frac{v_{max}+v_{min}}{2} \\ 0 \\ 1 \end{bmatrix}, DOP = \begin{bmatrix} \frac{u_{max}+u_{min}}{2} - prp_u \\ \frac{v_{max}+v_{min}}{2} - prp_v \\ -prp_n \\ 1 \end{bmatrix}, SH_{par} = \begin{bmatrix} 1 & 0 & sh_x & 0 \\ 0 & 1 & sh_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Il versore u rimane uguale

Il versore v rimane uguale

Il versore n viene trasformato in modo che DOP diventi parallelo a versore n

3. Trasformazione di vista (view mapping) caso parallelo

$$SH_{par} = SH_{par}(shx_{par}, shy_{par}) = \begin{bmatrix} 1 & 0 & shx_{par} & 0 \\ 0 & 1 & shy_{par} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$DOP' = [0 \;\; 0 \;\; dop_z \;\; 0]^T = SH_{par} \cdot DOP.$$

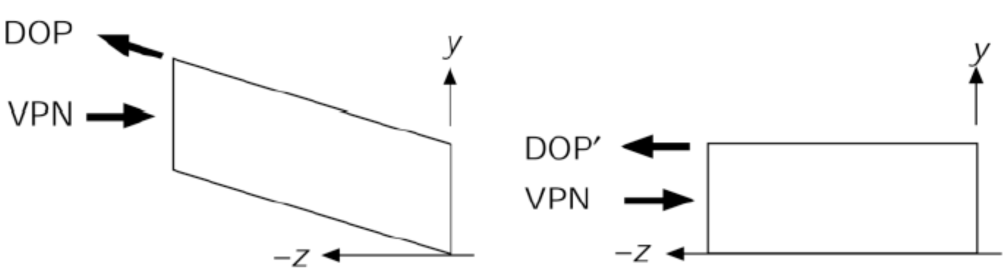
$DOP'_{x'}$ = 0 = DOP_x+shx_{par}*DOP_z

DOP'_y = 0 = DOP_y+shy_{par}*DOP_z

DOP'_z = DOP_z

$$shx_{par} = -\frac{dop_x}{dop_z}, \quad shy_{par} = -\frac{dop_y}{dop_z}.$$

Se dop_x=dop_y=0 allora Sh_{par}=Matrice identità
Infatti DOP e' parallelo a VPN e non c'è bisogno



$$\begin{bmatrix} 1 & 0 & \dfrac{1/2(u_{\max}+u_{\min})-prp_u}{prp_n} & 0 \\ 0 & 1 & \dfrac{1/2(v_{\max}+v_{\min})-prp_v}{prp_n} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. Trasformazione di vista (view mapping) caso parallelo

Obiettivo

Fase 3.par.2 Riscalare il volume (retto) dalle dimensioni attuali

$$[u_{min}, u_{max}]*[v_{min}, v_{max}]*[B,F]$$

nel volume retto $[-1,+1]*[-1,+1]*[-1,0]$

che e' finalmente il **volume canonico parallelo**!

3.par.2.1 Tralare il centro del volume in origine

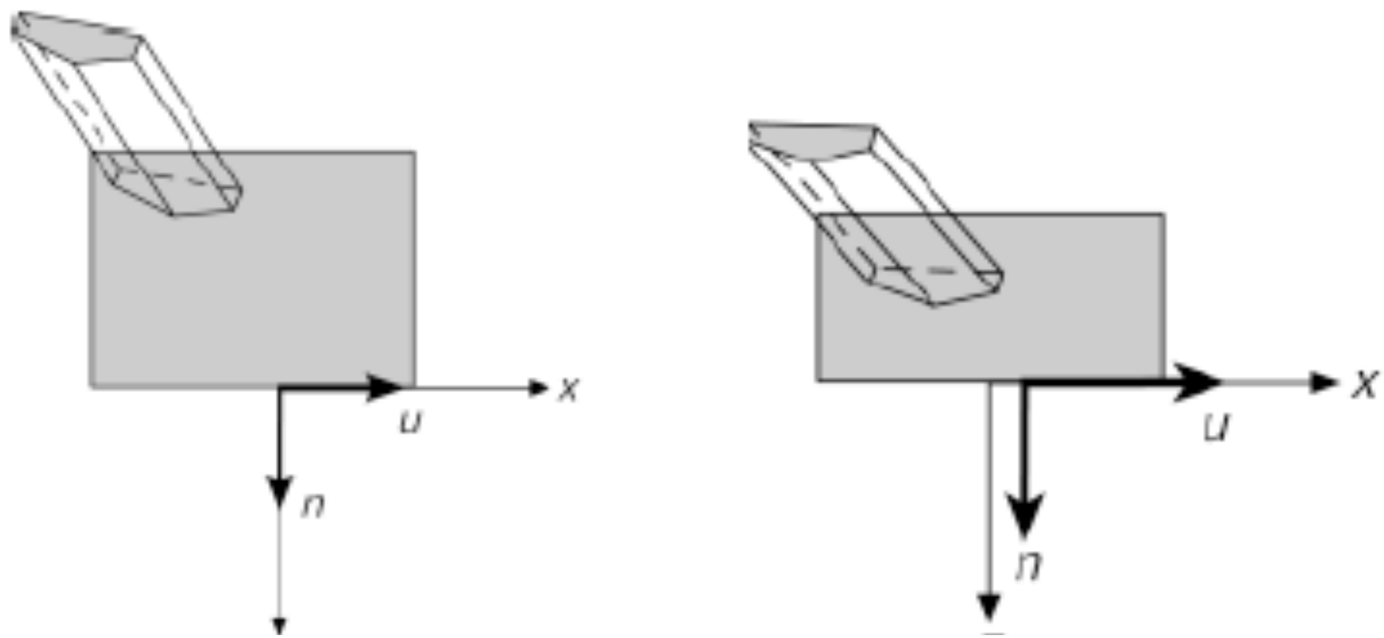
$$T_{par} = \begin{bmatrix} 1 & 0 & 0 & -(u_{max}+u_{min})/2 \\ 0 & 1 & 0 & -(v_{max}+v_{min})/2 \\ 0 & 0 & 1 & -F \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

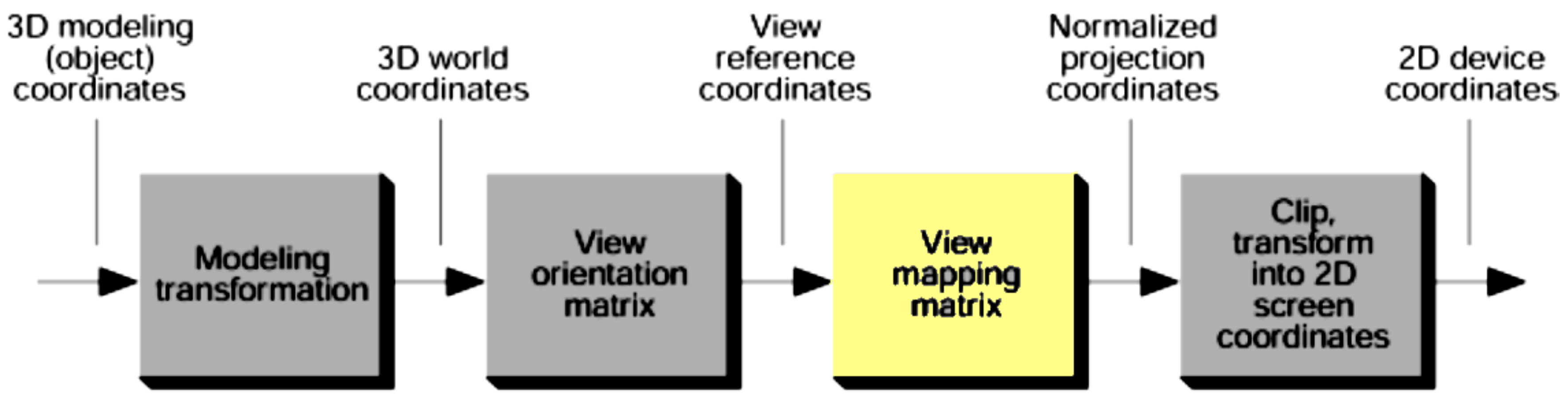
3.par.2.2 Scalare in modo che le dimensioni siano [2,2,1]

$$S_{par} = \begin{bmatrix} 2/(u_{max}-u_{min}) & 0 & 0 & 0 \\ 0 & 2/(v_{max}-v_{min}) & 0 & 0 \\ 0 & 0 & 1/(F-B) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Il Front Plane (F) va in 0, il Back plane (B) va in -1

3. Trasformazione di vista (**view mapping**) **caso parallelo**





Fatto! (per il Caso parallelo)

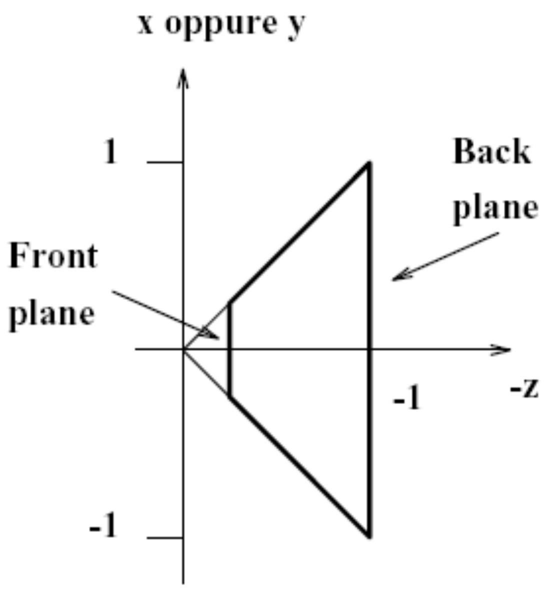
Obiettivo

trasformazione del volume di vista nel **volume canonico prospettico** definito da:

$x = z \quad ; x = -z$

$y = z \quad ; y = -z$

$z = -z_{min} \; ; z = -1$

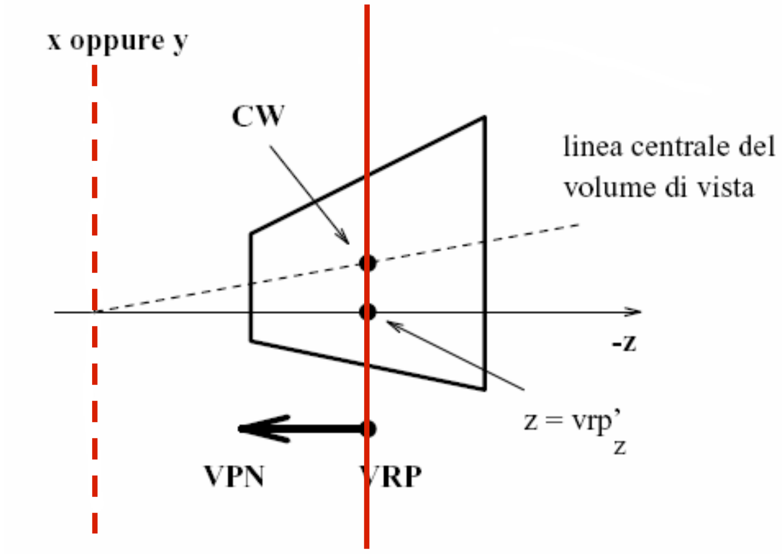


3. Trasformazione di vista (view mapping) caso prospettico

Fasi di trasformazione:

- **3.prosp.1** Traslazione in modo che il centro di proiezione (COP), trasformato in PRP, sia nell'origine

  T (-PRP)

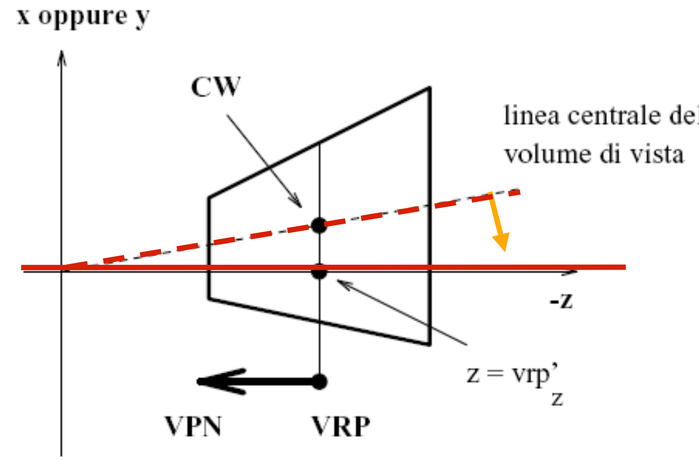


- **3.prosp.2** Deformazione di taglio (*shear*) in modo tale che la linea centrale del volume di vista sia parallela all'asse z.

  Stessa matrice Hz di scorrimento del caso parallelo. Trasforma il DOP:=CW-PRP in z

  VRP e' trasformato in ....

$$\mathrm{VRP}' = (\boldsymbol{H}_z \circ \boldsymbol{T}(-\mathrm{PRP}))(0,0,0,1)^T$$



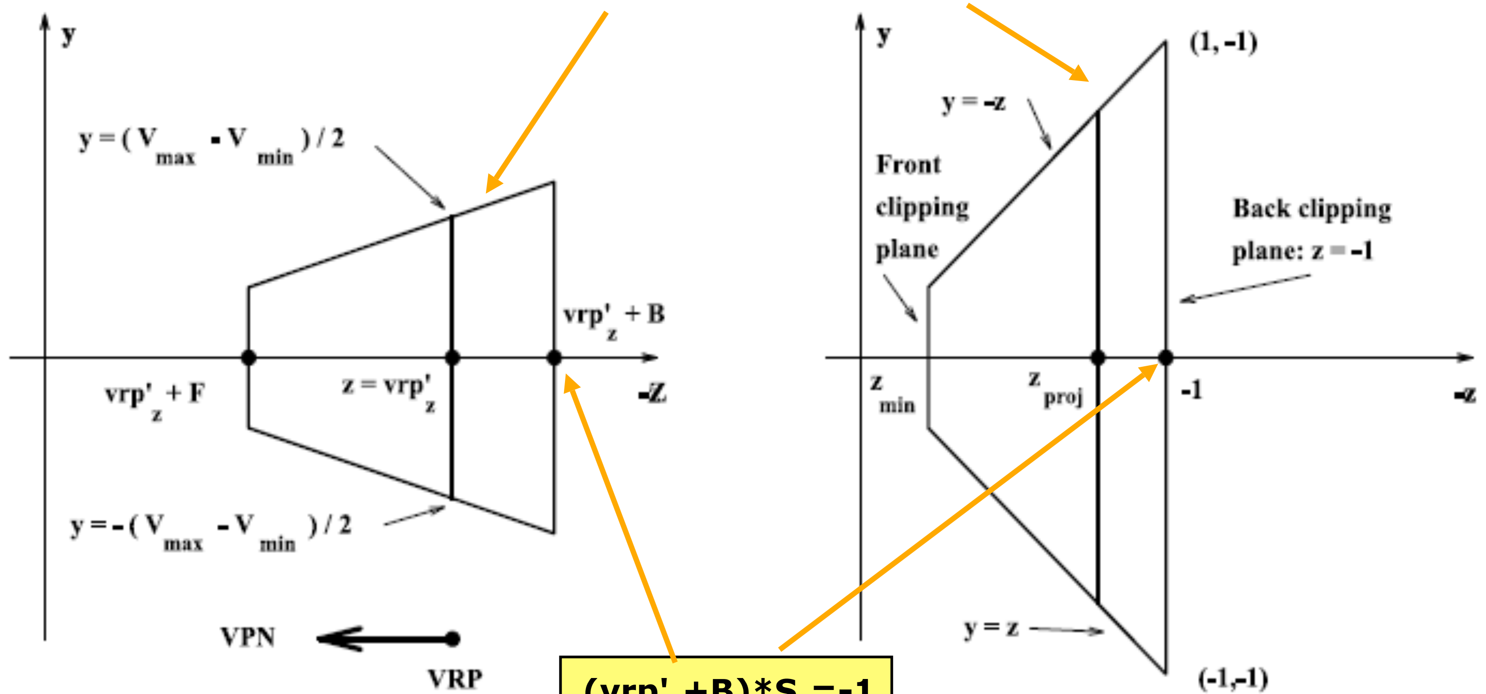
3. Trasformazione di vista (**view mapping**) **caso prospettico**

3.prosp.3

Si scalano (diversamente) **x** ed **y** in modo da ottenere una inclinazione di **45 gradi** per i quattro piani laterali della piramide di vista. E scalo **uniformemente** lo spazio tridimensionale a muovere il piano **z** = B (il piano back) nel piano z = −1

$(v_{max}-v_{min})/2 * S_y = -vrp_z$` quindi $S_y = -2 * vrp_z / (v_{max}-v_{min})$

per x e' analoga!



$(vrp'_z+B)*S_z=-1$

$S_z=-1/(vrp'_z+B)$

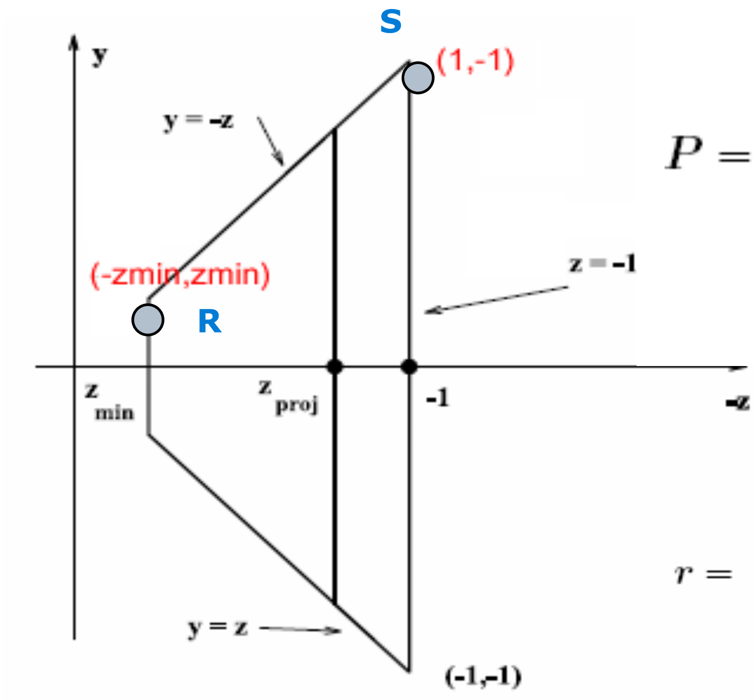
Lo scaling trovato su z viene eseguito sulle 3 coordinate!!

$$S\left(\frac{2\cdot vrp'_z}{(u_{max}-u_{min})(vrp'_z+B)},\ \frac{2\cdot vrp'_z}{(v_{max}-v_{min})(vrp'_z+B)},\ \frac{-1}{vrp'_z+B}\right)$$

3. Trasformazione di vista (**view mapping**) caso prospettico

3.prosp.4

Devo trasformare la piramide retta in volume canonico [-1,1]*[-1,1]*[-1,0]



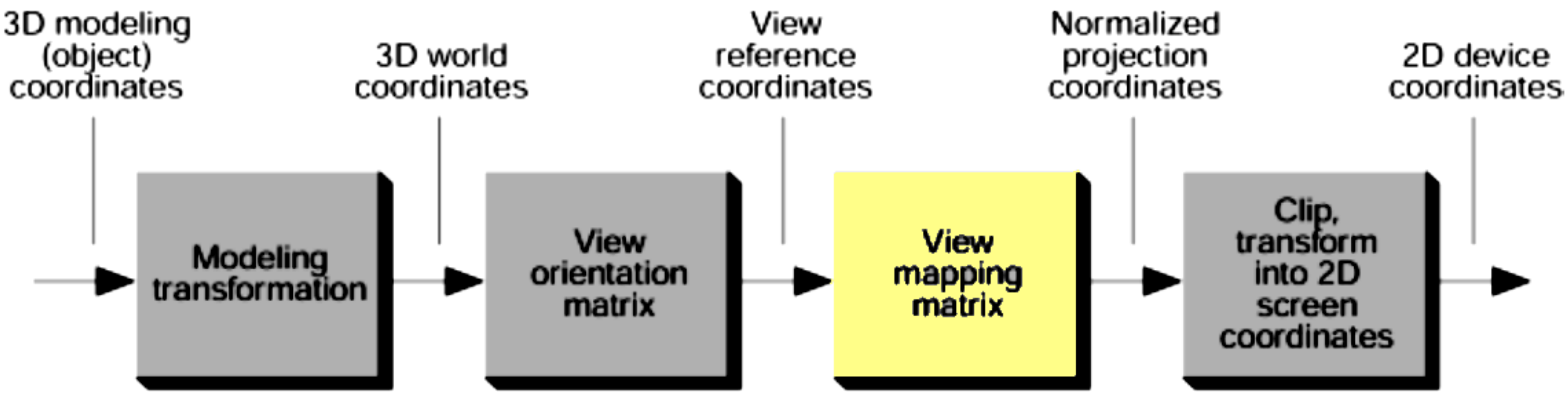
$$P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{1+z_{min}} & \frac{-z_{min}}{1+z_{min}} \\ 0 & 0 & -1 & 0 \end{pmatrix}, \quad z_{min} \neq -1$$

dimostrate che funziona applicando P ai punti R e S!

$$r = \begin{pmatrix} -z_{min} \\ -z_{min} \\ z_{min} \\ 1 \end{pmatrix}^T \longrightarrow P(r) = \begin{pmatrix} -z_{min} \\ -z_{min} \\ 0 \\ -z_{min} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

$$s = \begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \end{pmatrix}^T \longrightarrow P(s) = \begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \end{pmatrix}$$

3. Trasformazione di vista (**view mapping**) **caso prospettico**



Fatto! (per il Caso prospettico)

4. Trasformazione di dispositivo

Obiettivo

Si riduce ad un problema di mapping dal box:

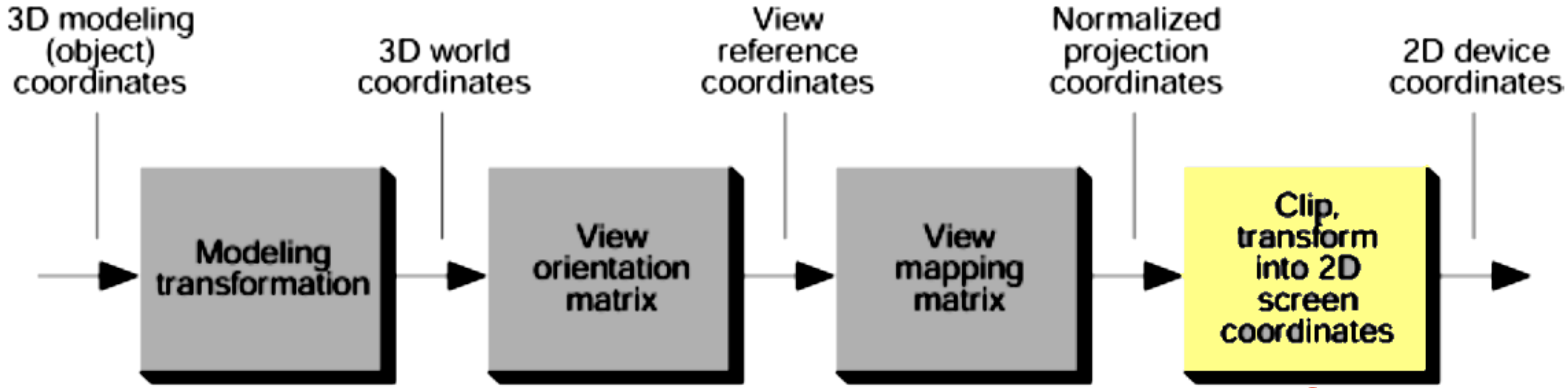
$$W = [w_1, w_4] \times [w_2, w_5] \times [w_3, w_6]$$

al box:

$$V = [v_1, v_4] \times [v_2, v_5] \times [v_3, v_6]$$

Soluzione:

$$\boldsymbol{T}_D = \boldsymbol{T}(v_1, v_2, v_3) \circ \boldsymbol{S}\left(\frac{v_4 - v_1}{w_4 - w_1}, \frac{v_5 - v_2}{w_5 - w_2}, \frac{v_6 - v_3}{w_6 - w_3}\right) \circ \boldsymbol{T}(-w_1, -w_2, -w_3)$$



Da fare... Fatto!

Classificazione delle proiezioni

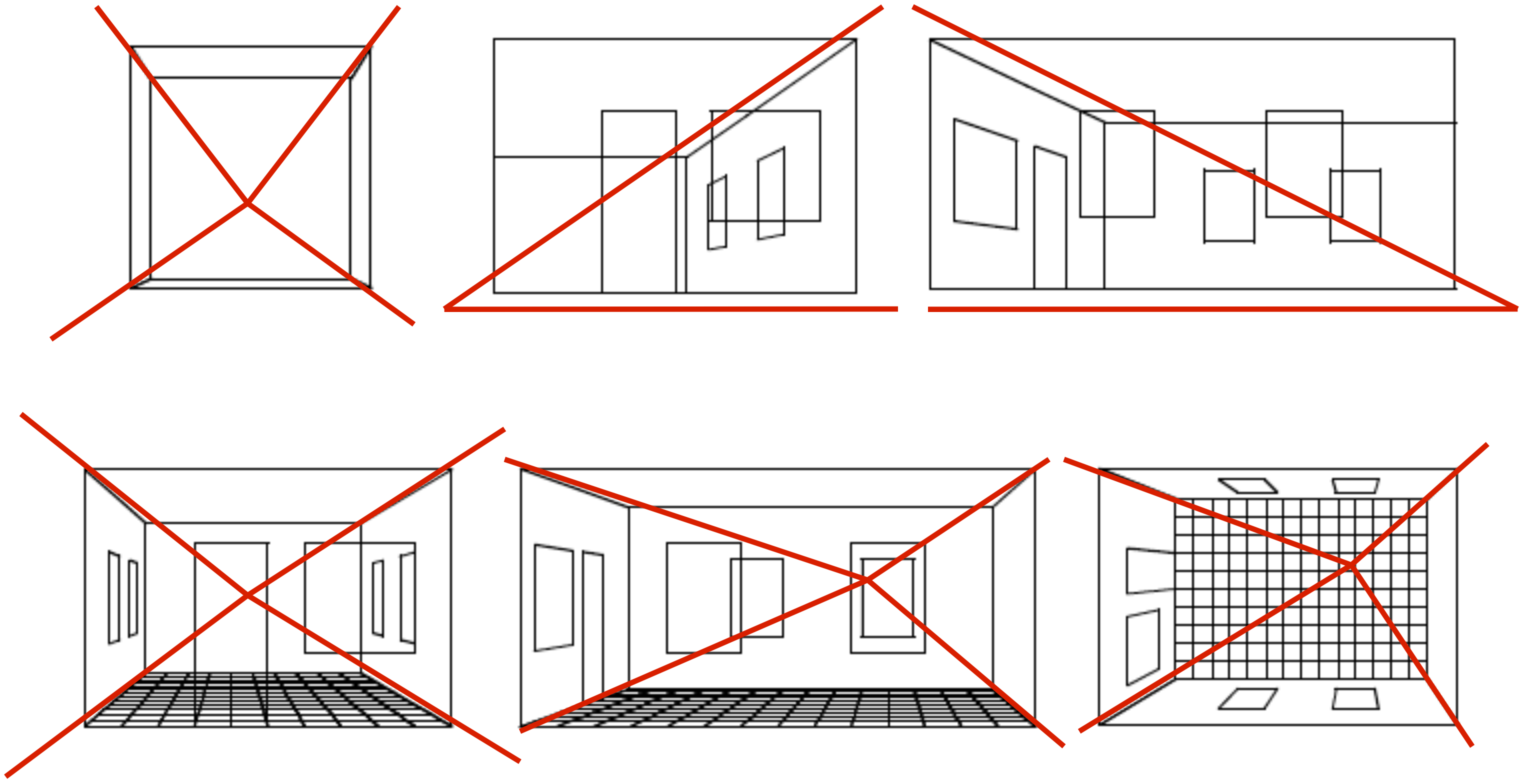
Prospettiche	Centrali (1-punto)	
	Accidentali (2-punti)	
	Oblique (3-punti)	
Parallele	Ortografiche	Semplici
		Multiple
	Assonometriche	Ortogonali
		Oblique

Proiezioni prospettiche

- Se il piano di vista e':
  - parallelo ad un piano coordinato la prospettiva si dice ad **un punto** (1 punto di fuga) o **centrale**.
  - parallelo ad un asse coordinato la prospettiva si dice a **due punti** (2 punti di fuga) o **accidentale**.
  - altrimenti la prospettiva si dice a **tre punt**i (3 punti di fuga) o **obliqua** (oppure ***fotografica***)

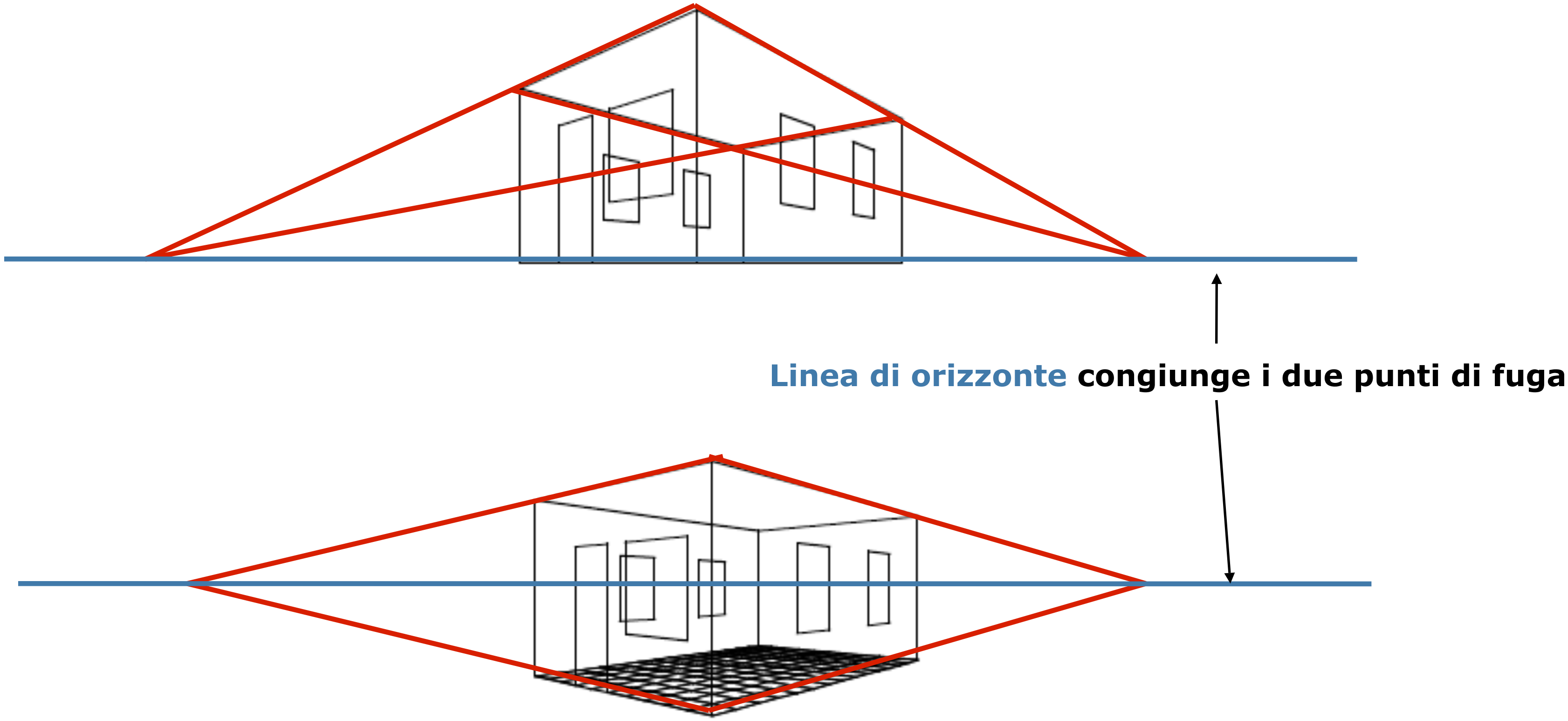
Proiezioni prospettiche

- Esempi di prospettive **centrali** - 1 punto di fuga



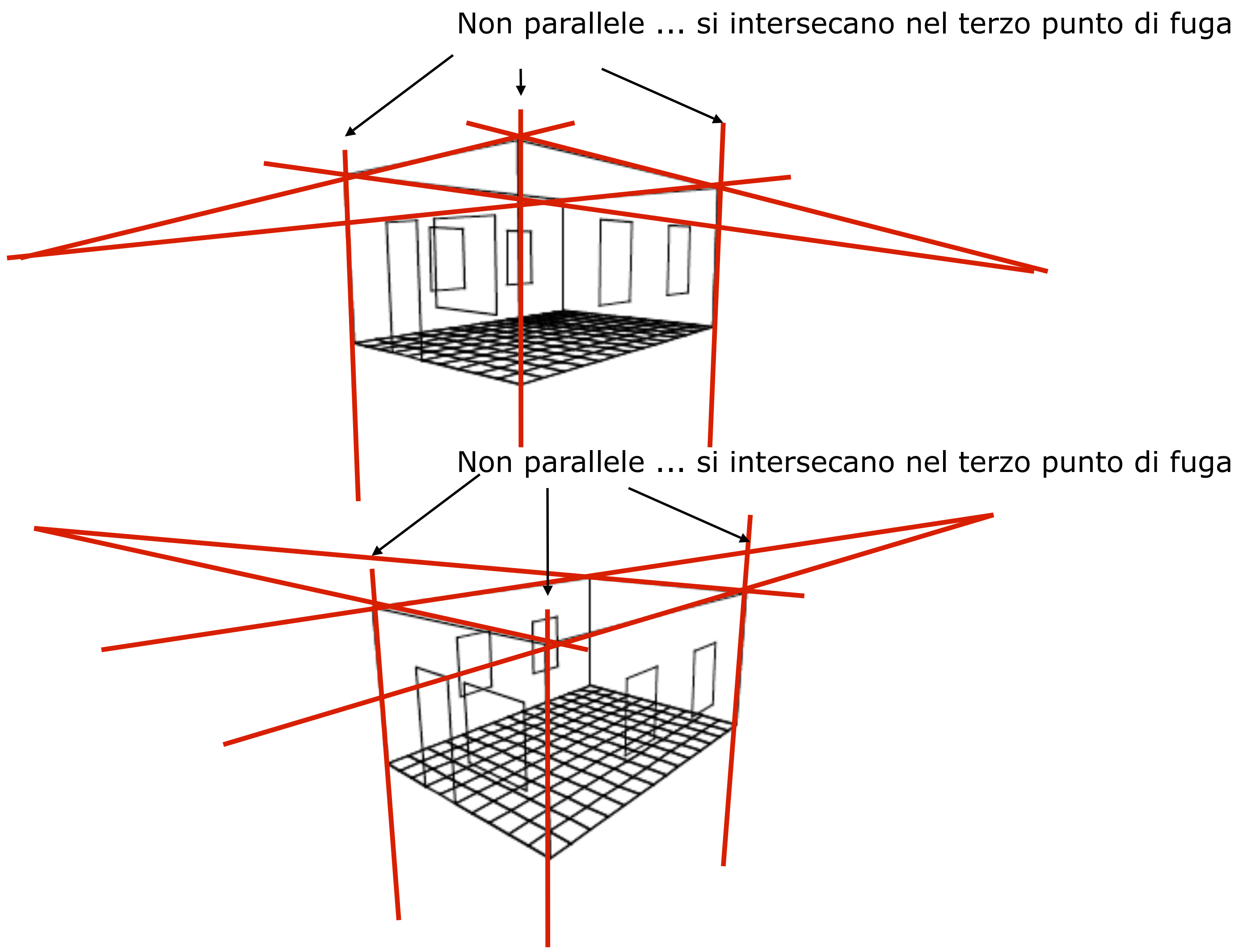
Proiezioni prospettiche

- Esempi di prospettive **accidentali** - **2** punti di fuga



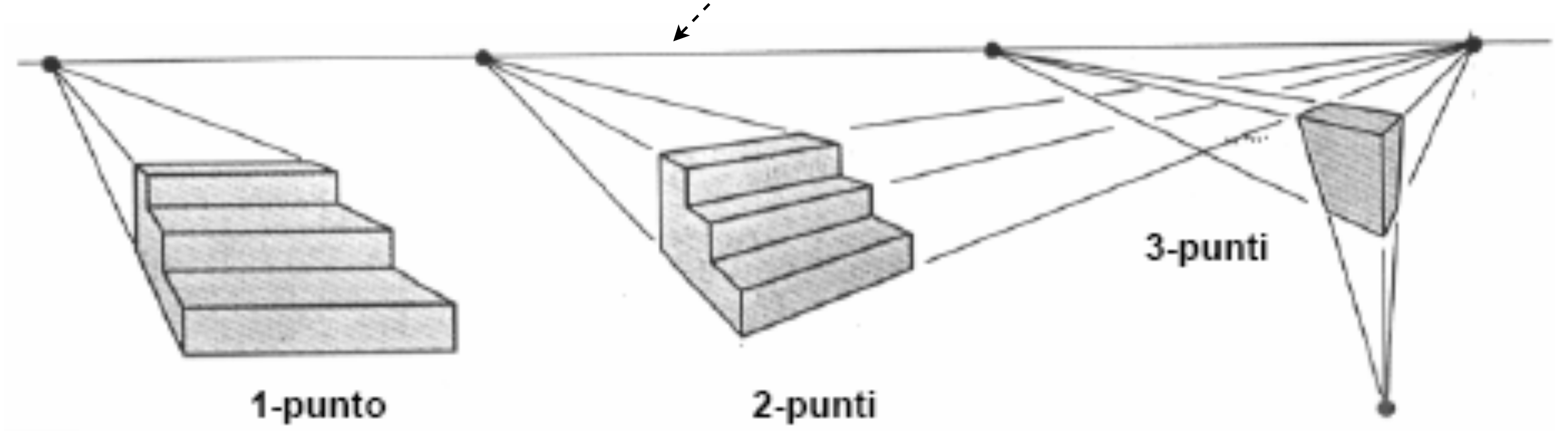
Proiezioni prospettiche

- Esempi di prospettive **oblique** - **3** punti di fuga



Proiezioni prospettiche. Riassunto

Linea dell’orizzonte

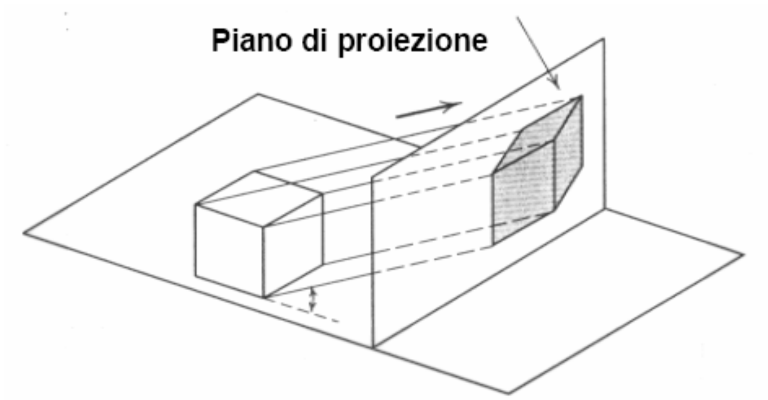


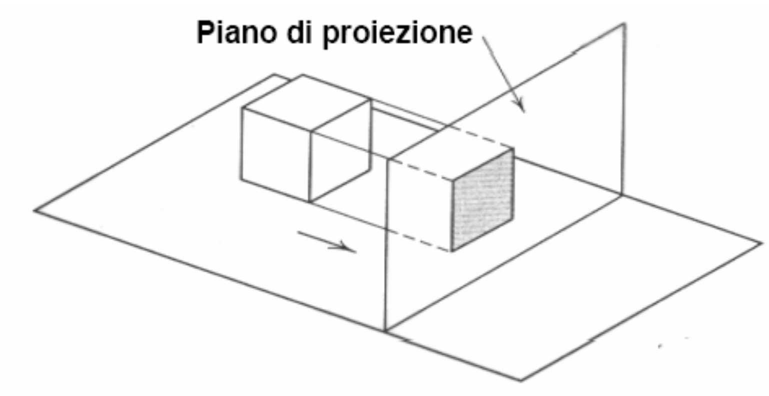
Centrale Accidentale Obliqua

Proiezioni - Nomenclatura

Prospettiche	Centrali	(1-punto)
	Accidentali	(2-punti)
	Oblique	(3-punti)
Parallele	Ortografiche	Semplici
		Multiple
	Assonometriche	Ortogonali
		Oblique

Proiezioni parallele





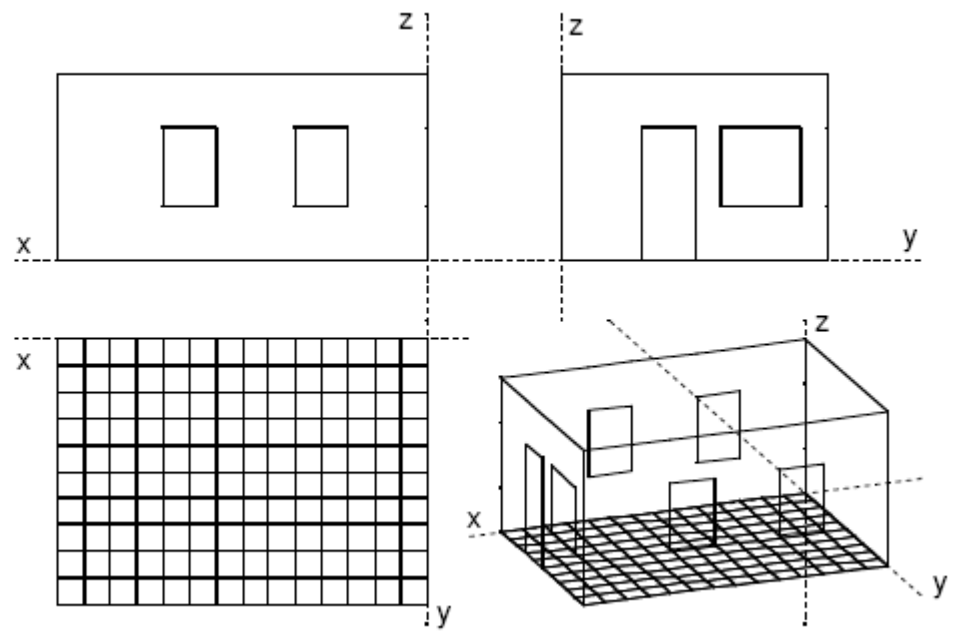
Proiezioni Parallele Ortografiche

- **Ortografiche**: quando il **view plane** e' un **piano coordinato** x=0, y=0 o z=0, e la DOP e' parallela all'asse coordinato che gli e' ortogonale, rispettivamente x(1,0,0), y(0,1,0) o z(0,0,1).

- La proiezione non dipende da VRP per il parallelismo delle rette proiettanti - metto sempre **VRP** in **(0,0,0)** ... quindi si trova su DOP!

- Le **distanze misurate** sulla proiezione coincidono con le distanze misurate sul modello 3D.

Proiezione di Monge
proiezione ortografica su 2, spesso 3, piani coordinati

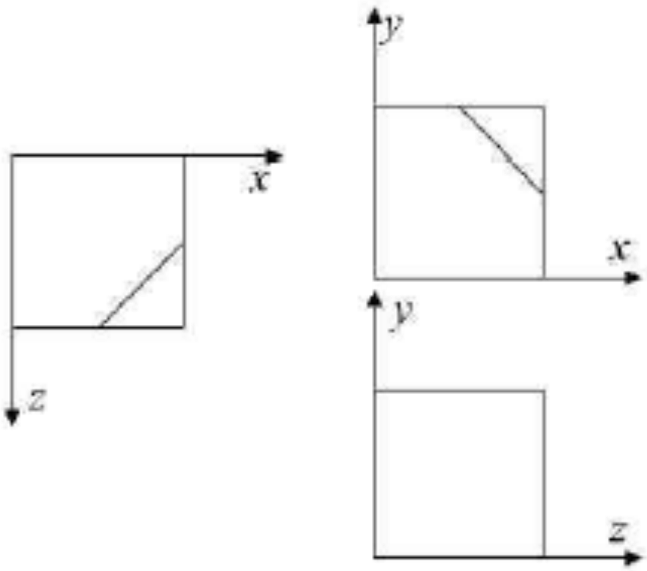


Proiezioni ortografiche di Monge, e ulteriore assonometria

Proiezioni parallele ortografiche

Piante / Alzati: il piano di proiezione è perpendicolare ad uno degli assi cartesiani

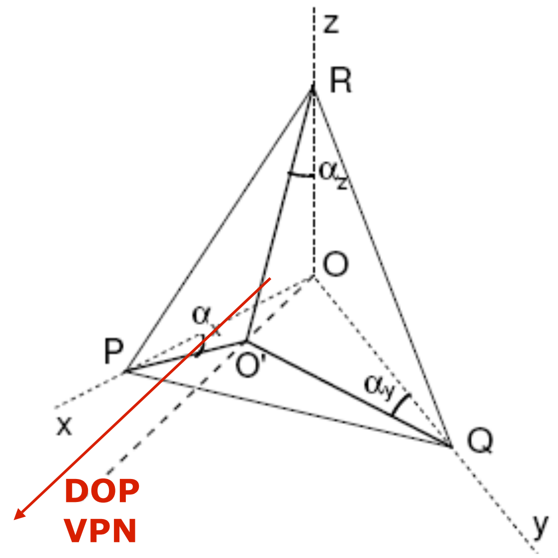
Le distanze misurate sulla proiezione coincidono con le distanze misurate sul modello 3D.



Proiezioni parallele assonometriche

- Tutte le restanti proiezioni parallele (*non ortografiche*) si dicono **assonometriche**
  - se il DOP e’ parallelo a VPN si dice assonometrica **ortogonale** - *DOP ortogonale al piano di proiezione!*
  - altrimenti si dice assonometrica **obliqua**
- Le distanze misurate sulla proiezione sono diverse dalle distanze reali.

Proiezioni parallele assonometriche ortogonali



$$\mathrm{DOP} = \mathrm{VPN} = \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix}$$

$$\begin{aligned} n_x &= \cos(\frac{\pi}{2} - \alpha_x) = \sin \alpha_x \\ n_y &= \cos(\frac{\pi}{2} - \alpha_y) = \sin \alpha_y \\ n_z &= \cos(\frac{\pi}{2} - \alpha_z) = \sin \alpha_z \end{aligned}$$

•Se tre angoli uguali ad alpha, la proiezione si dice **isometrica** - sono rispettate le proporzioni rispetto al modello 3D, cioe' *sono in scala*

||DOP|| = SQRT(3*sin(alpha)^2)= 1 -> alpha=arcsin(sqrt(1/3)) **~ 35 gradi**

•Se due angoli uguali la proiezione si dice **dimetrica** - nella proiezione si hanno due fattori di scala distinti rispetto al modello 3D

Ne esistono in numero infinito - quella piu' comune ha i due angoli di ~ **19 e 61 gradi**

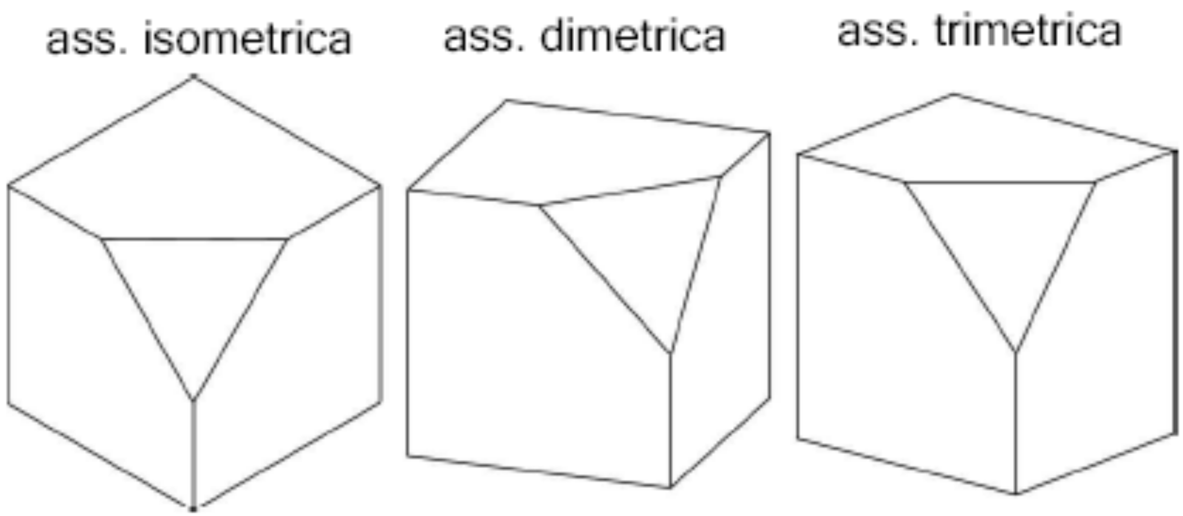
•Se gli angoli sono diversi la proiezione si dice **trimetrica**- tre fattori di scala distinti rispetto al modello 3D

Ne esistono in numero infinito: quella piu' comune ha angoli di ~ **27, 60 e 9 gradi**

Proiezioni parallele assonometriche ortogonali, esempi

Assonometriche: il piano di proiezione non è perpendicolare ad alcuno degli assi cartesiani

Le distanze misurate sulla proiezione sono diverse dalle distanze reali.



Proiezioni parallele assonometriche ortogonali, esempi

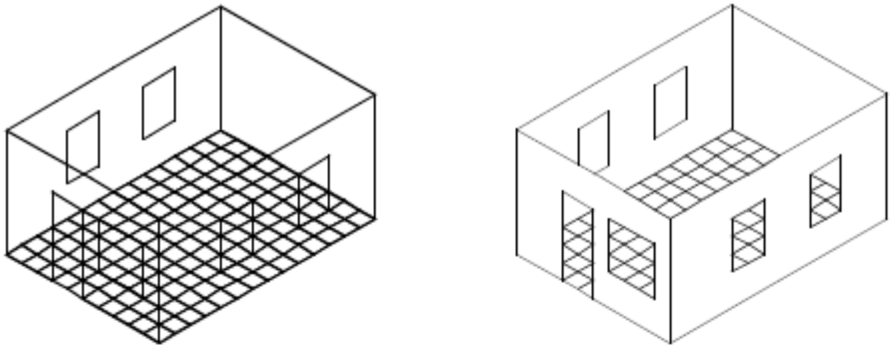


Figura 5.21 Proiezione ortogonale isometrica

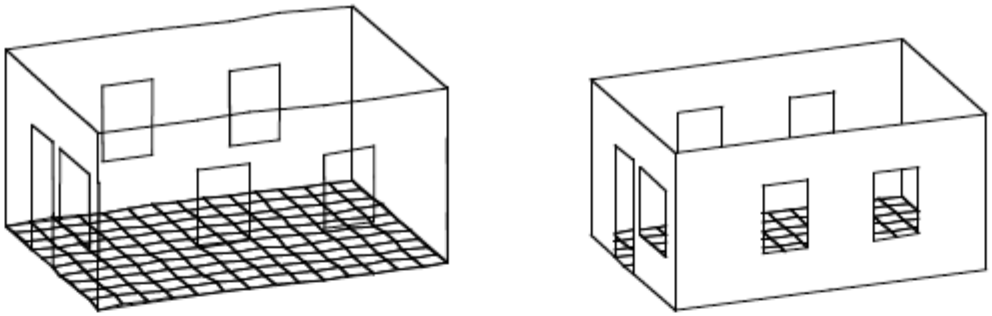


Figura 5.22 Proiezione ortogonale dimetrica

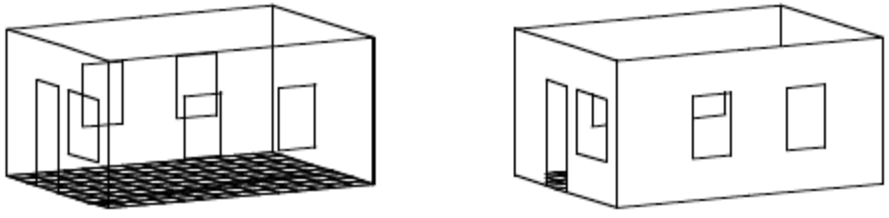
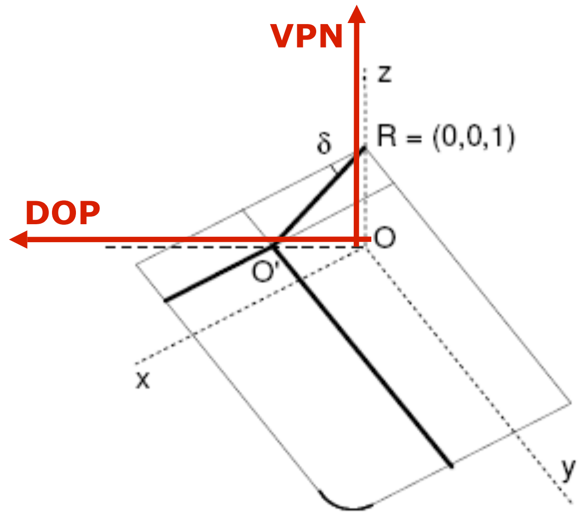


Figura 5.23 Proiezione ortogonale trimetrica

Proiezioni parallele assonometriche oblique

- Le proiezioni assonometriche **oblique** hanno il DOP **non parallelo** con il VPN!

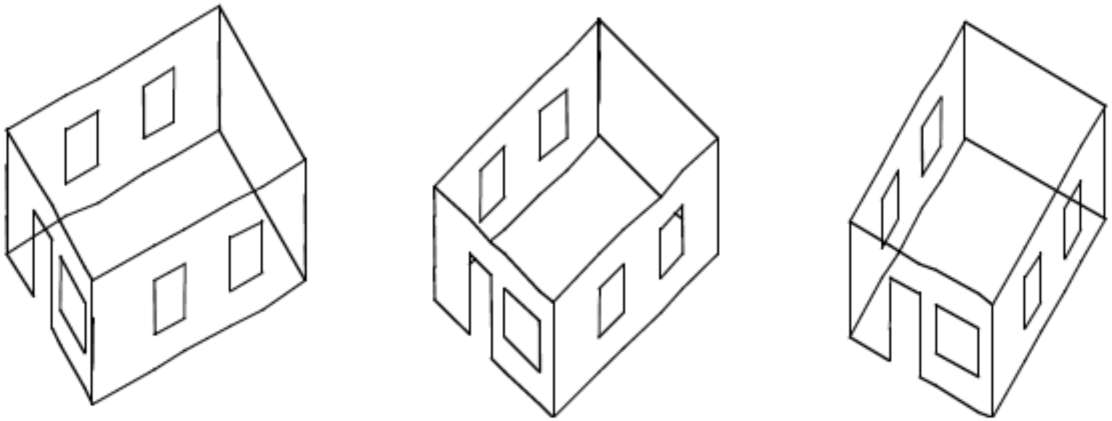
- Tra le infinite assonometriche oblique le **cavaliere** sono quelle con il view plane parallelo a un piano coordinato.



$$\mathrm{DOP} = \begin{pmatrix} |O'R|\cos\delta \\ |O'R|\sin\delta \\ 1 \end{pmatrix},$$

- Tra le cavaliere:
  - **Militari** o cavaliere **isometriche** - quando ||O’R|| = 1.0
  - **Cabinet** o cavaliere **dimetriche** - quando ||O’R|| = 0.5

Proiezioni parallele assonometriche oblique, esempi



Esempi di proiezioni militari (==isometriche!)

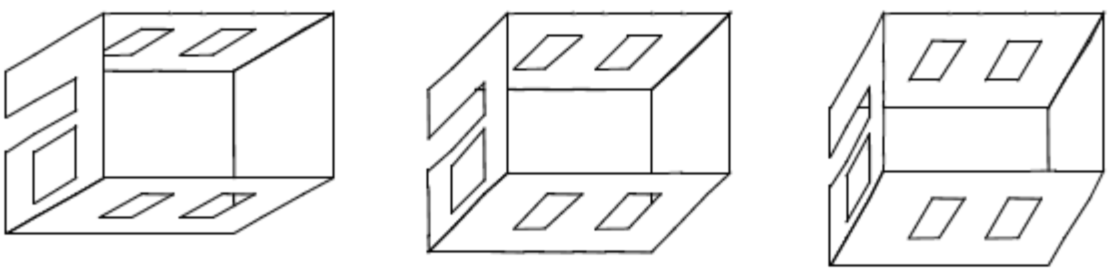


Figura 5.26 Proiezioni militari standard con VUV $= -e_2$

Proiezioni parallele assonometriche oblique, esempi

Esempi di proiezioni militari (==isometriche!)

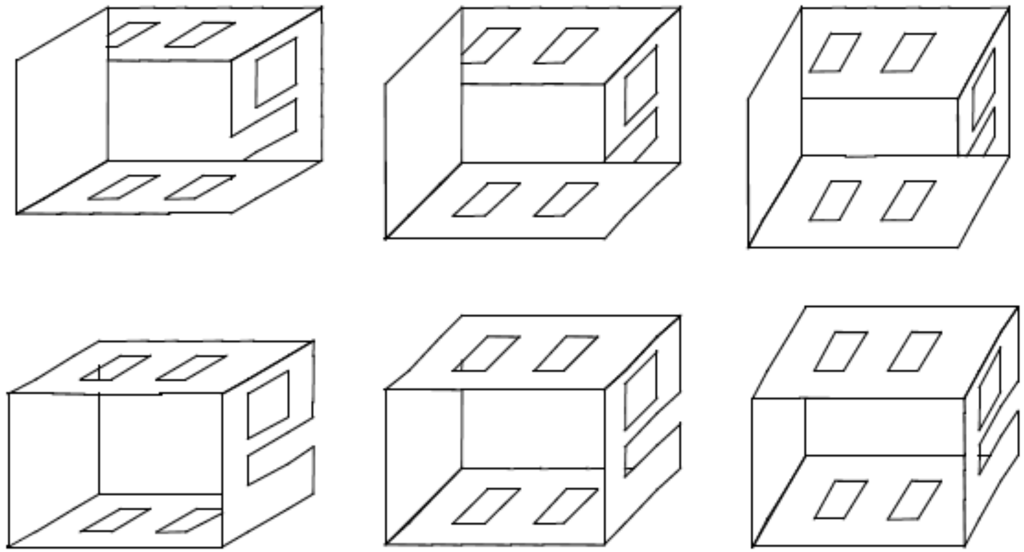


Figura 5.27 Proiezioni militari standard con VUV $= e_2$ e DOP nel terzo e primo ottante, rispettivamente

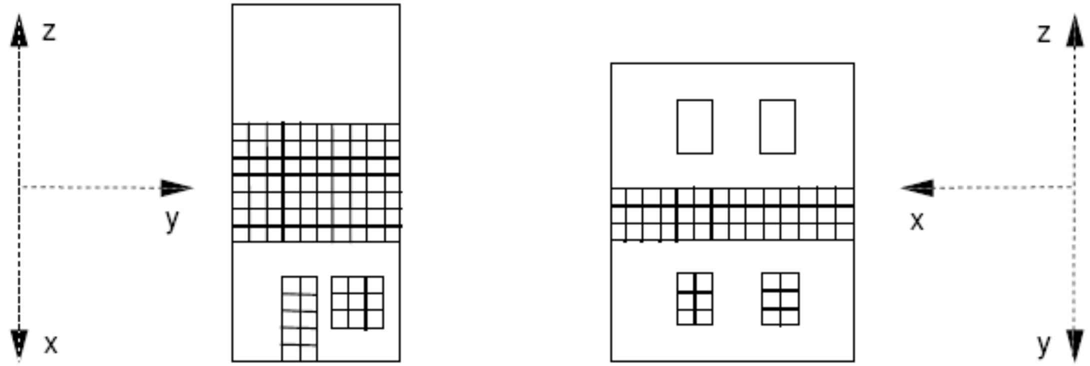


Figura 5.28 (a) Proiezione militare *frontale*, con DOP in xz (b) proiezione militare *laterale* con DOP in yz

Proiezioni parallele assonometriche oblique, esempi

Esempi di proiezioni cabinet (==dimetriche!)

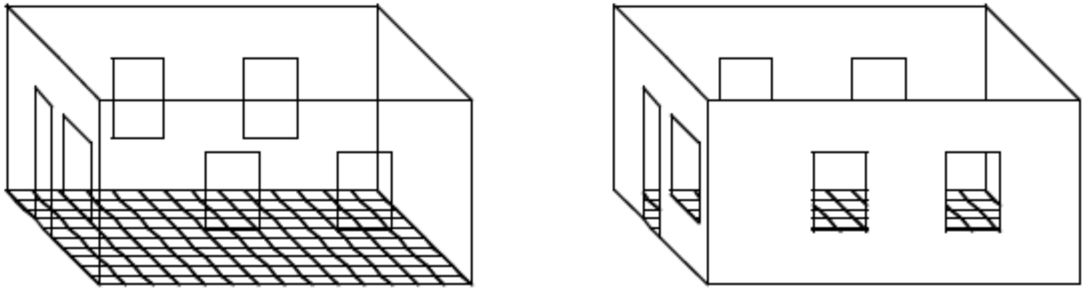


Figura 5.29 Proiezione cabinet standard con piano di vista verticale

Importanza della trasformazione prospettica

Aiuta a rimuovere le superfici nascoste di una scena! Perche' ...

Il test di visibilità su coppie di punti e'

1. dati due punti **p** e **q**, scoprire se sono allineati con l'osservatore **o**;

2. dati tre punti allineati **p**, **q** e **o**, determinare se **p** oppure **q** sia più vicino a **o**.

- Soluzione geometrica ... **senza trasformazione prospettica**

  - ✓ Condizione di **allineamento**

    $(\mathbf{p} - \mathbf{o}) \times (\mathbf{q} - \mathbf{o}) = \mathbf{0}$ ... *tre determinati 2 × 2 devono essere nulli*

  - ✓ Condizione di **vicinanza**

    $|\mathbf{p} - \mathbf{o}| < |\mathbf{q} - \mathbf{o}|$ ... *6 prodotti, 6 differenze e quattro somme*

Importanza della trasformazione prospettica

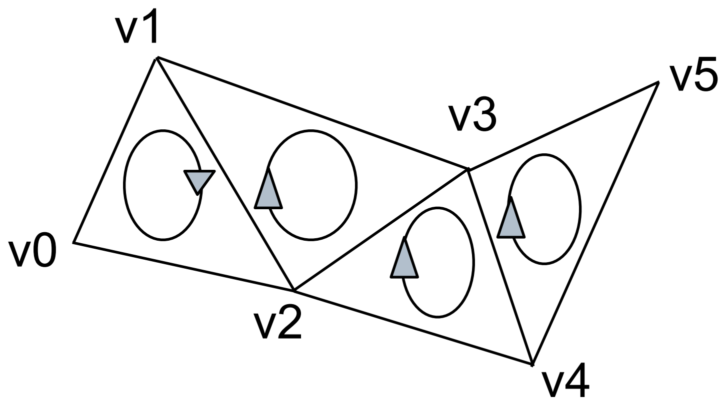
Nel sistema **NPC** le coordinate dei punti invece possono essere interpretate come equivalenti alla loro profondità prospettica

- Il **test di visibilità** in coordinate NPC ...
  - ✓ si riduce algebricamente alla seguente semplice formulazione:
    1. $x_p = x_q$ e $y_p = y_q$ ... *due confronti numerici!*
    2. $z_p < z_q$ ... *un confronto numerico!*
- Inoltre per la rimozione delle superfici nascoste della scena la procedura e' la stessa per le proiezioni **prospettiche** e **parallele**

Reminder

OpenGL - GL_TRIANGLE_STRIP

- GL_TRIANGLE_STRIP disegna una serie di triangoli usando i vertici v0->v1->v2, quindi v2->v1->v3, quindi v2->v3->v4, e cosi’ via. L’ordine scelto deriva dalla necessita’ di assicurare che tutti i triangoli siano disegnati con un orientamento tale che la Triangle Strip possa formare correttamente una parte di una superfice.



- Quindi se 0, 1, 2, 3, 4, 5, 6, 7, ... sono gli indici dei vertici, i triangoli della strip sono:
    - 0, 1, 2 ... in ordine!
    - 2, 1, 3 ... i primi due sono in ordine inverso!
    - 2, 3, 4 ... in ordine!
    - 4, 3, 5 ... i primi due sono in ordine inverso!
    - 4, 5, 6 ...
    - 6, 5, 7 ...