Γραφικά Υπολογιστών

Ιόνιο Πανεπιστήμιο
Τμήμα Πληροφορικής

Στέργιος Παλαμάς, Επίκουρος Καθηγητής

Μάθημα 11:

Μετασχηματισμοί 3D

3D Affine Transformations

Affine Transformations

$$
\underbrace{
\underset{\text{3x3}}{\overset{\textbf{2D}}{\begin{bmatrix} a & c & e \\ b & d & f \\ 0 & 0 & 1 \end{bmatrix}}}
\qquad
\underset{\text{4x4}}{\overset{\textbf{3D}}{\begin{bmatrix} a & d & g & j \\ b & e & h & k \\ c & f & i & l \\ 0 & 0 & 0 & 1 \end{bmatrix}}}
}_{\text{homogeneous coordinates}}
$$

Είχαμε καταφέρει στους 2D μετασχηματισμούς να εκφράσουμε τα translation, rotation και scaling με 3x3 πίνακες.

Κατ’ αναλογία, στις 3 διαστάσεις θα χρειαστούμε πίνακες 4x4

3D Affine Transformations

Scale

2D

$$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3x3

3D

$$\begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4x4

Για το Scaling η μορφή του πίνακα 3D είναι ανάλογη με αυτή του 2D, με την προσθήκη μιας πρόσθετης τρίτης κλιμάκωσης κατά τον άξονα z.

Translation

2D

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

3x3

3D

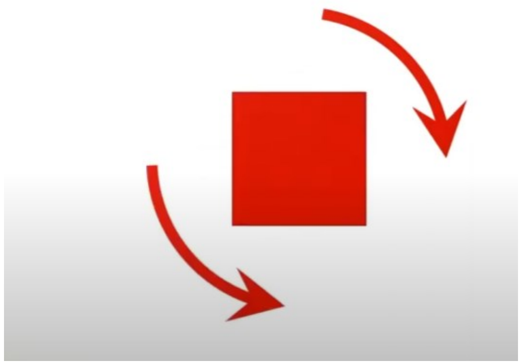
$$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4x4

homogeneous coordinates

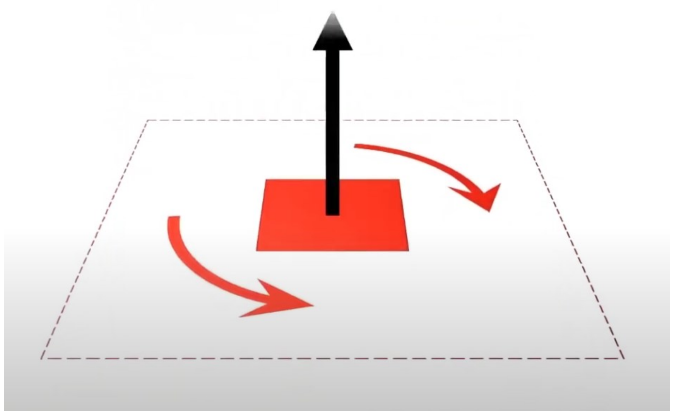
Η μετατόπιση (translation) επίσης εκφράζεται με έναν πίνακα 4x4, ανάλογο με αυτόν του 2D, με την προσθήκη μιας επιπλέον μετατόπισης κατά τον άξονα z.

Rotation

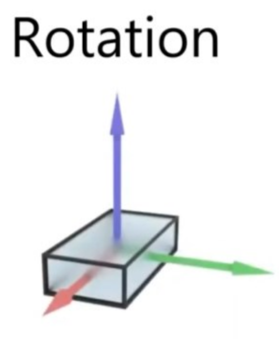


Στην περιστροφή (rotation) όμως τα πράγματα δεν είναι το ίδιο απλά.

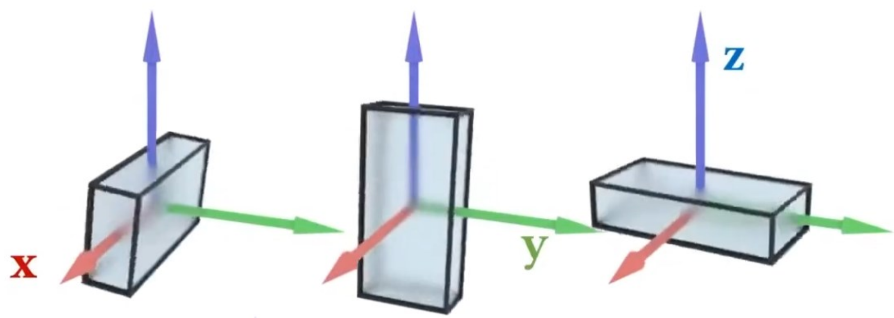
Ενώ στις 2 διαστάσεις (2d) η περιστροφή μπορούσε να γίνει πάνω στην επιφάνεις, μόνο ως προς έναν άξονα (κάθετο) είτε δεξιόστροφα, είτε αριστερόστροφα ….

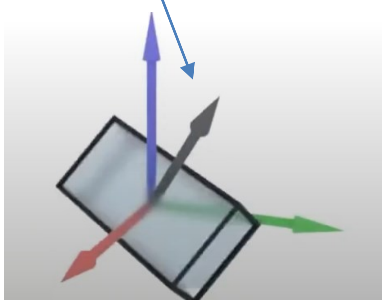


3D Affine Transformations

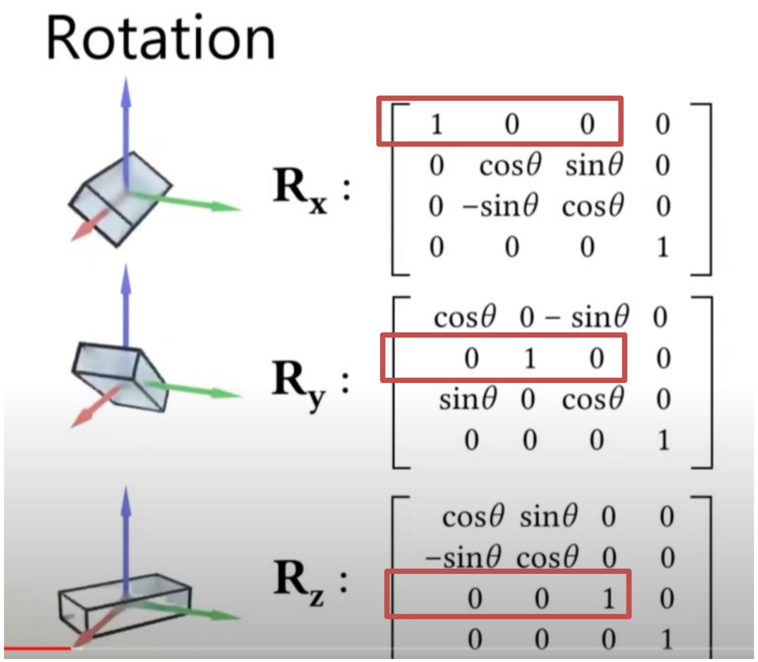


… στις τρείς διαστάσεις, η περιστροφή μπορεί να είναι γύρω από οποιονδήποτε άξονα (x, y, z) αλλά και γενικότερα γύρω από οποιοδήποτε άξονα-διάνυσμα.





3D Affine Transformations



Στον πίνακα περιστροφής γύρω από τον x η πρώτη γραμμή είναι 1 0 0 0 καθώς οι συντεταγμένες x των σημείων δεν αλλάζουν όταν η περιστροφή είναι γύρωα από τον άξονα x.

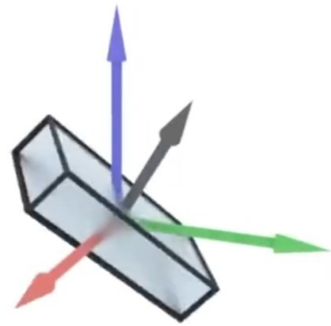
Αντίστοιχα στον πίνακα περιστροφής γύρω από τον y, η γραμμή 2 είναι 0 1 0 0 καθώς οι συντέταγμενες y των σημείψν δεν αλλάζουν στην περιστροφή γύρω από τον y

Και αντίστοιχα στην περιστροφή γύρω από τον z

Οι πίνακες περιστροφής γύρω από τους άξονες x,y,z έχουν ανάλογη μορφή με τους 2D . Τα ημίτονα και συνημίτονα μπαίνουν στις γραμμές των αξόνων που οι συντεταγμένες αλλάζουν κατά την περιστροφή.

Rotation

$$\mathbf{R} = \mathbf{R_z}(\alpha)\,\mathbf{R_y}(\beta)\,\mathbf{R_x}(\gamma)$$



$$= \overset{\text{yaw}}{\begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}} \overset{\text{pitch}}{\begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}} \overset{\text{roll}}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}}$$

$$= \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{bmatrix}$$

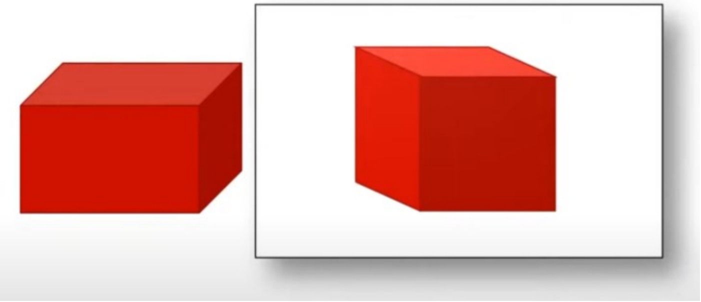
Rotation

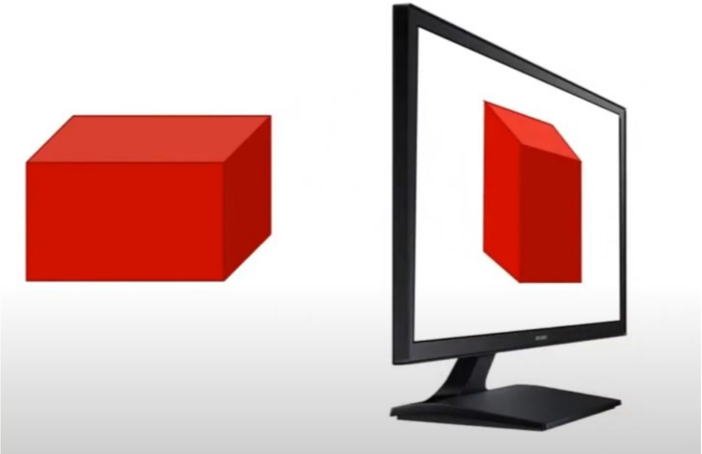
$$\mathbf{R_z}(\alpha)\,\mathbf{R_y}(\beta)\,\mathbf{R_x}(\gamma)$$
$$\neq \mathbf{R_y}(\beta)\,\mathbf{R_z}(\alpha)\,\mathbf{R_x}(\gamma)$$
$$\neq \mathbf{R_x}(\gamma)\,\mathbf{R_z}(\alpha)\,\mathbf{R_y}(\beta)$$
$$\neq \mathbf{R_z}(\alpha)\,\mathbf{R_x}(\gamma)\,\mathbf{R_y}(\beta)$$
$$\neq \mathbf{R_y}(\beta)\,\mathbf{R_x}(\gamma)\,\mathbf{R_z}(\alpha)$$
$$\neq \mathbf{R_x}(\gamma)\,\mathbf{R_y}(\beta)\,\mathbf{R_z}(\alpha)$$

Η Περιστροφή γύρω από οποιονδήποτε άξονα μπορεί να αναλυθεί σε μια:

- περιστροφή κατά **γ** γύρω από τον **x** – $R_x(\gamma)$
- περιστροφή κατά **β** γύρω από τον **y** – $R_y(\beta)$
- περιστροφή κατά **α** γύρω από τον **z** – $R_z(\alpha)$

Οπότε ο συνολικός πίνακας περιστροφής έχει την παραπάνω μορφή. **Προσοχή :** Η σειρά περιστροφών στην παραπάνω σχέση είναι σημαντική. Πρώτα εκτελείται η περιστροφή κατά x, μετά κατά y και τέλος κατά z .

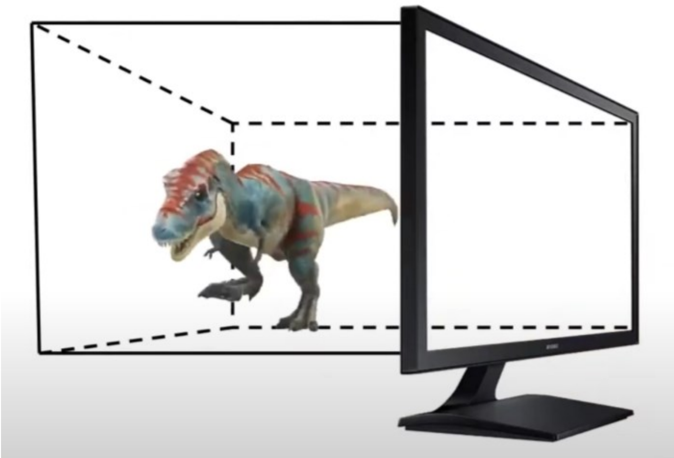


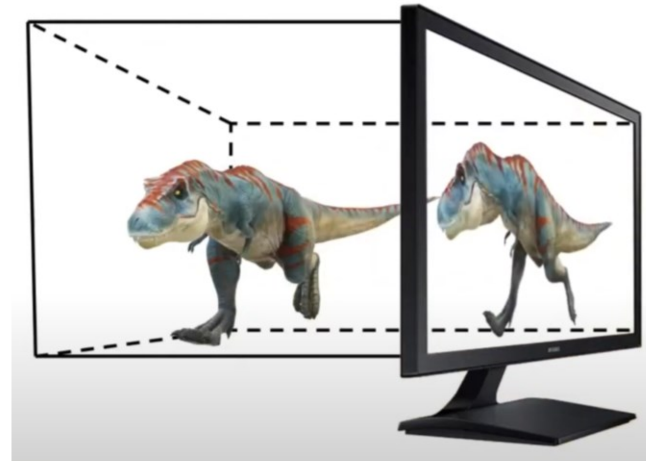


Το ζητούμενο στο rendering μιας 3D σκηνής είναι να απεικονιστεί στη δισδιάστατη οθόνη μας, μια αναπαράσταση της 3D σκηνής από μια συγκεκριμένη οπτική γωνία παρατήρησης (κάμερα).

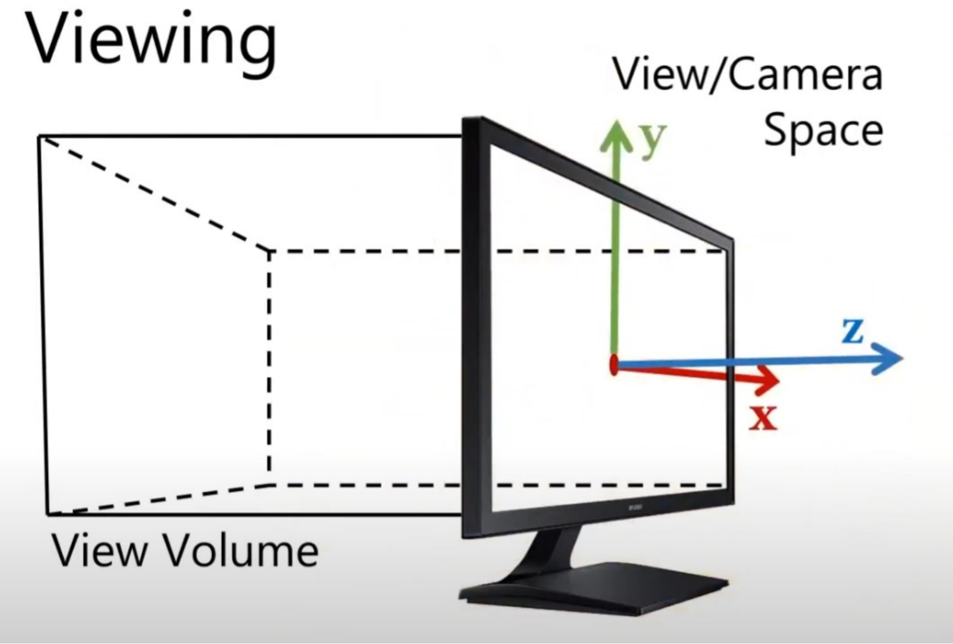
Projection Transformations

Viewing





Είναι σαν πίσω από την οθόνη μας να υπάρχει ένα εικονικό 3D περιβάλλον και με κάποιον τρόπο να πρέπει να πάρω μια απεικόνισή του , **μια προβολή του από συγκεκριμένη οπτική γωνία στην οθόνη**.

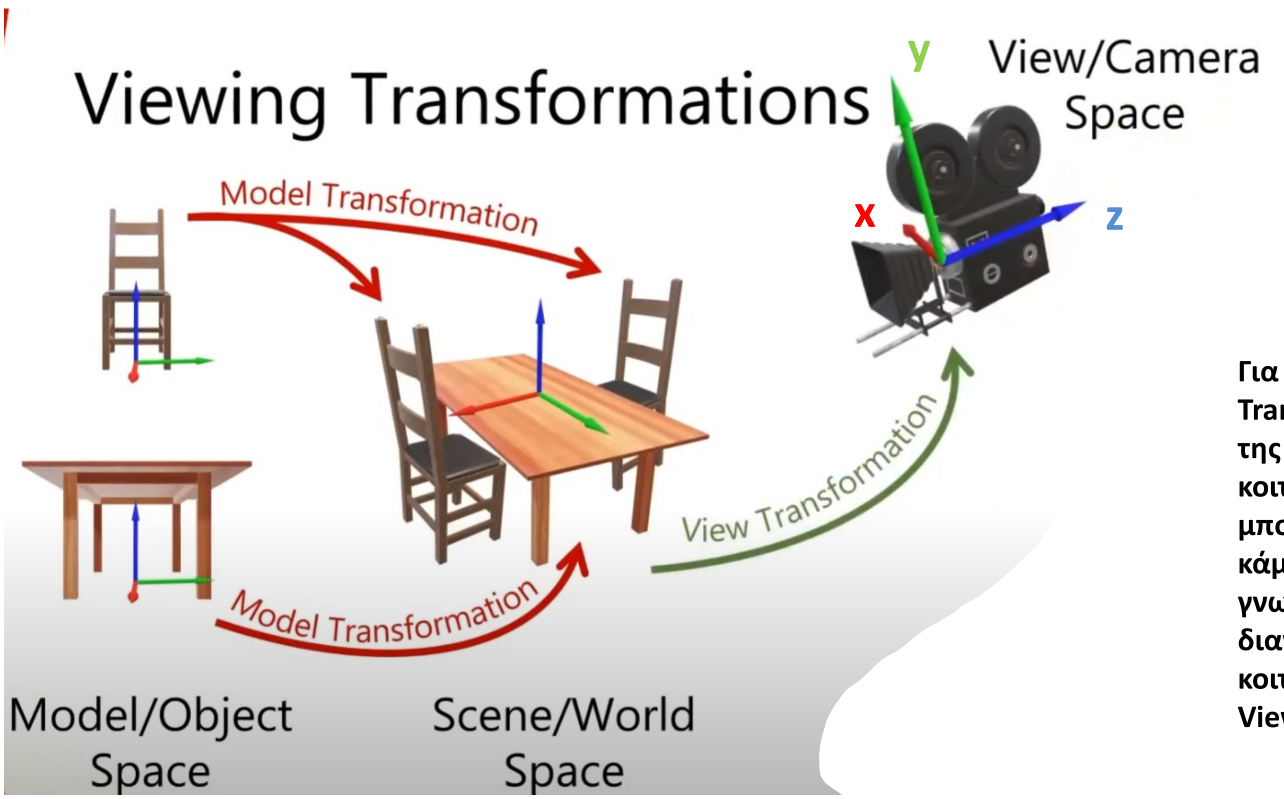


Ορίζουμε λοιπόν ένα σύστημα συντεταγμένων στην οθόνη μας όπως φαίνεται παραπάνω (**View/Camera Space**).

Τη διεύθυνση του x και y την βάζουμε όπως έχουμε συνηθίσει να δουλεύουμε στις 2 διαστάσεις οπότε το z αντιπροσωπεύει το βάθος.

Ο ορατός στην κάμερα (και επομένως στην οθόνη) τρισδιάστατος χώρος ονομάζεται **View Volume**

Projection Transformations



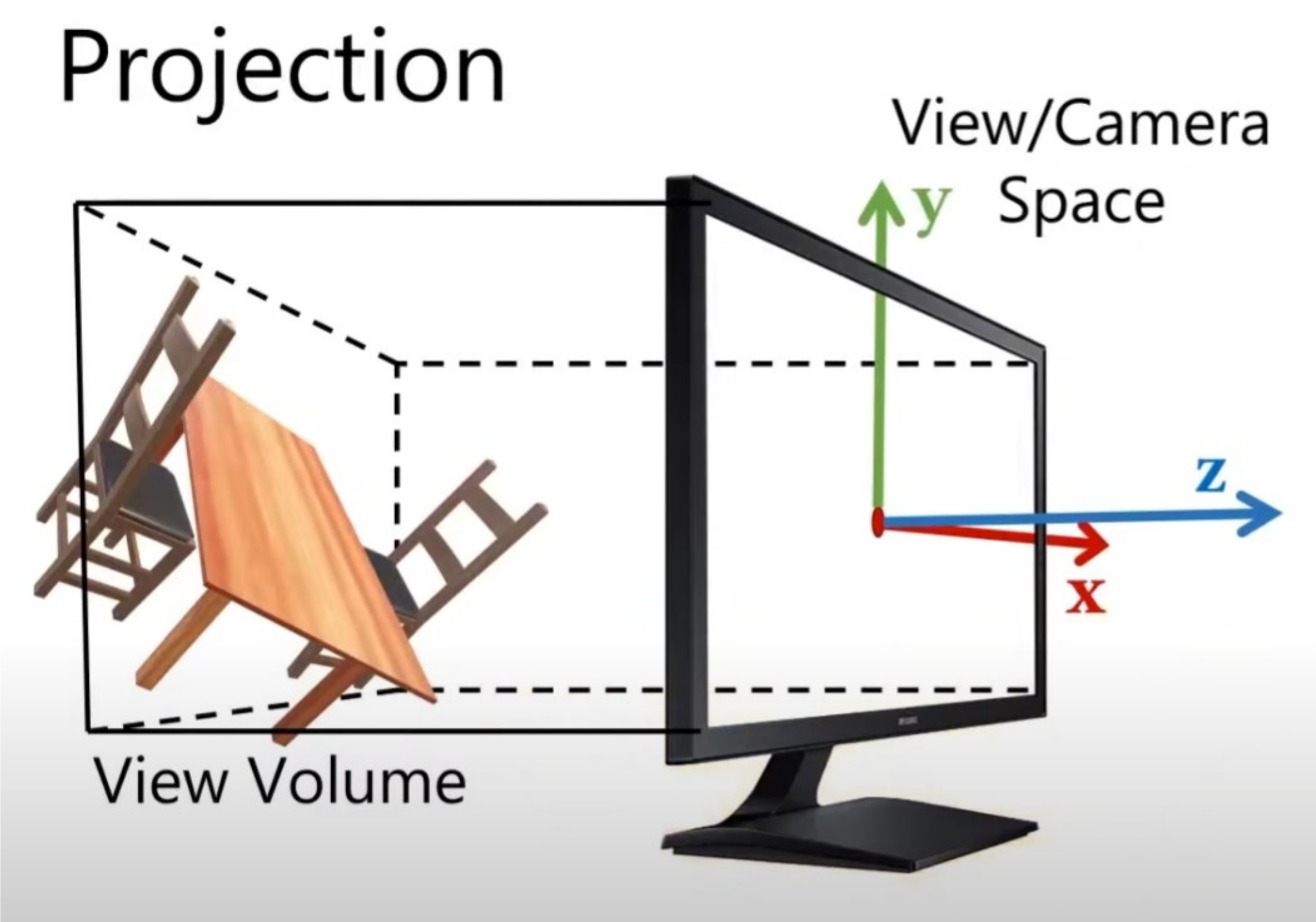
Για να ορίσουμε το «View Transformation" , ορίζουμε τη θέση της κάμερας, την κατεύθυνση που κοιτάει (-z) και το διάνυσμα y . Μετά μπορούμε να βρούμε τη θέση της κάμερας στο World Space και γνωρίζοντας και τα δύο παραπάνω διανύσματα (y και κατεύθυνση που κοιτάει) μπορούμε να φτιάξουμε το View Transformation

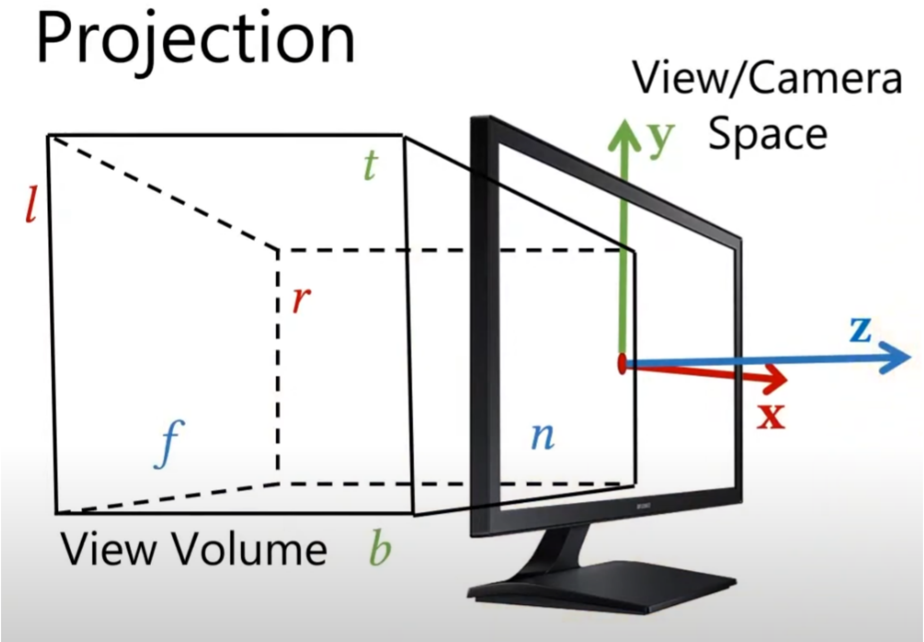
Τα τρισδιάστατα μοντέλα που χρησιμοποιούμε είναι σχεδιασμένα στο δικό τους σύστημα συντεταγμένων (**Model/Object Space**).

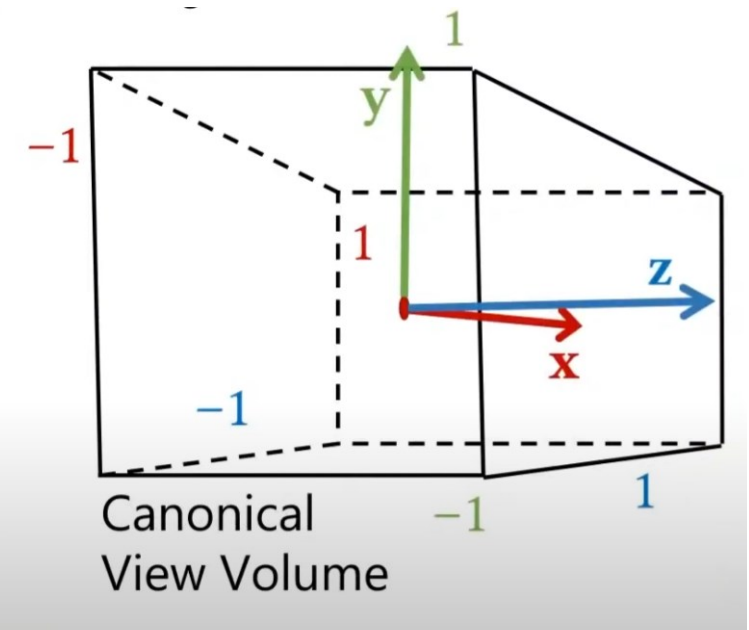
Όταν τα τοποθετούμε στη σκηνή, τα μετακινούμε , τα περιστρέφουμε, τα μεγαλώνουμε ή μικραίνουμε (scale) και πλέον ανήκουν στο **Scene/World Space** . Η Σκηνή έχει το δικό της σύστημα συντεταγμένων.

Τη σκηνή εμείς την παρακολουθούμε/ζωγραφίζουμε μέσω μιας «Κάμερας» η οποία έχει το δικό της σύστημα συντεταγμένων **(View/Camera Space).**

Με τους κατάλληλους μετασχηματισμούς λοιπόν πάμε από το Model/Object Space στο Scene/World Space και μετά στο View/Camera Space για να «ζωγραφίσουμε» τη σκηνή σε οθόνη.


Projection
View/Camera
Space
y
z
x
View Volume





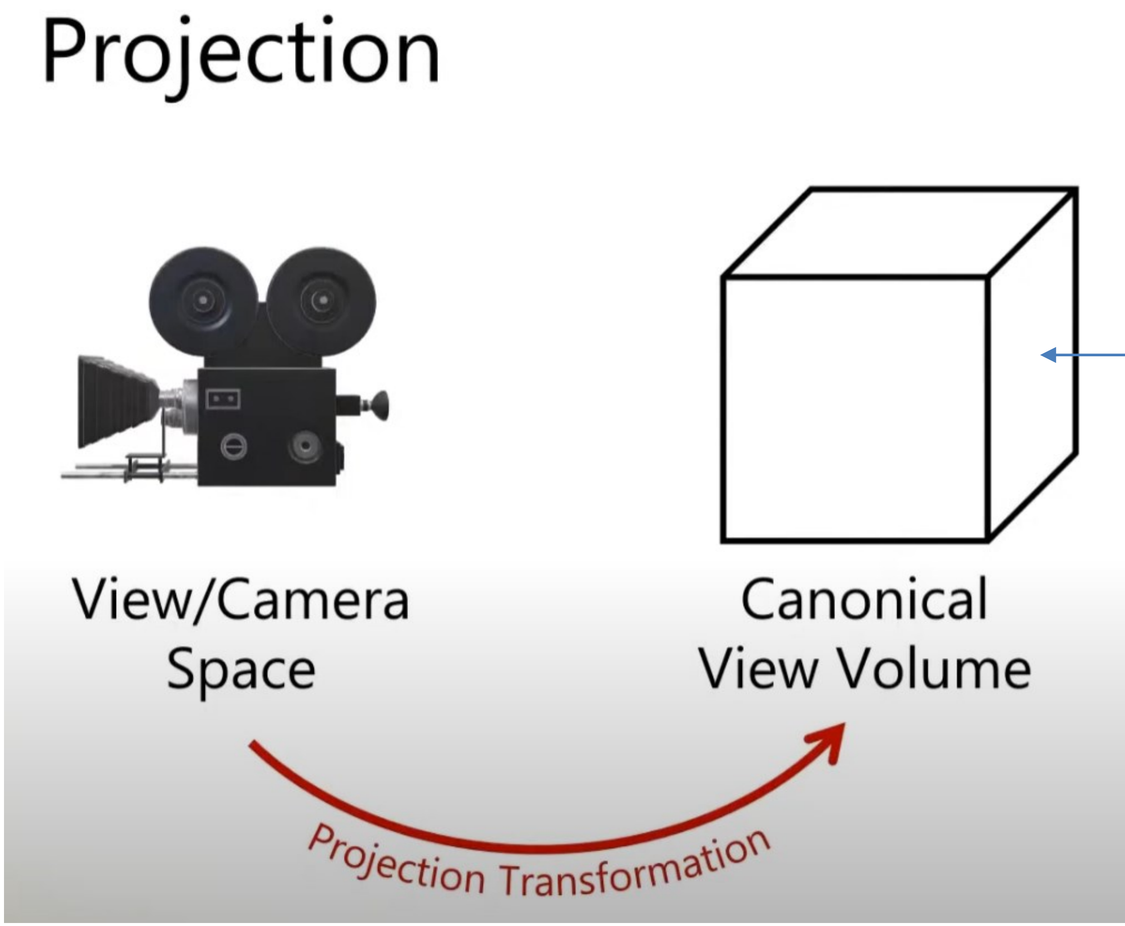
Για να ορίσουμε το View Volume , οριζουμε σε κάθε άξονα μια ελάχιστη και μια μέγιστη τιμή:

Άξονας x : Ελάχιστη **l (left)** → Μέγιστη **r (right)**
Άξονας y : Ελάχιστη **b (bottom)** → Μέγιστη **t (top)**
Άξονας z : Ελάχιστη **f (far)** → Μέγιστη **n (near)**

Αυτό θα είναι το View Volume και ό,τι περιέχεται μέσα σε αυτό στη σκηνή θα ζωγραφίζεται στην οθόνη.

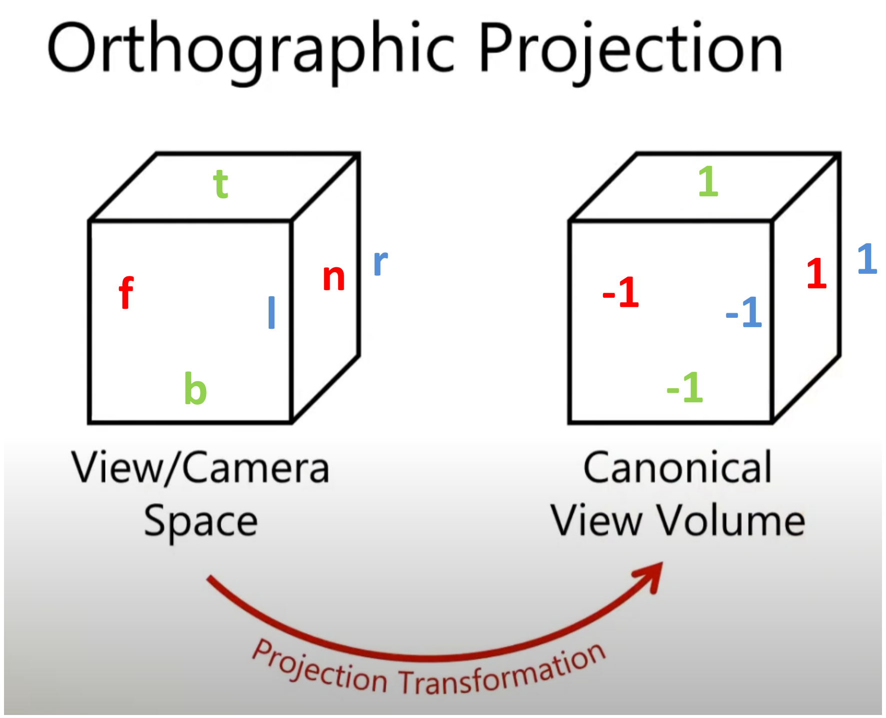
Στο **κανονικοποιημένο (canonical) View Volume στη συνέχεια , θέτουμε -1 στην ελάχιστη και 1 στη μέγιστη συντεταγμένη σε κάθε πλευρά.**

Projection Transformations



Η τελική εικόνα θα σχηματιστεί σε αυτή την πλευρά. Κρατάμε ωστόσο και τα δεδομένα βάθους (z data) για να χρησιμοποιηθούν για άλλες διεργασίες

Projection ονομάζουμε τη μετατροπή του Camera Space στο Canonical View Volume



Στην ορθογραφική προβολή παίρνουμε το Camera Space και το μετασχηματίζουμε στο Canonical View Volume.

Orthographic Projection

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Συντεταγμένες στο Camera Space

Συντεταγμένες στο Canonical View Volume

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ \boxed{0 \quad 0 \quad 0 \quad 1} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Την τελευταία γραμμή τη χρησιμοποιούμε πάντα για τις ομοιογενείς συντεταγμένες ώστε να μπορούμε να εκφράσουμε και το Translation με πίνακες

Projection Transformations

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\downarrow$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} ? & 0 & 0 & ? \\ 0 & ? & 0 & ? \\ 0 & 0 & ? & ? \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Πηγαίνοντας από το Camera Space στο Canonical Space, στην ορθογραφική προβολή , παρατηρούμε ότι οι άξονες x,y,z είναι ευθυγραμμισμένοι στα δύο spaces. **Δε θα χρειαστούμε λοιπόν περιστροφές.**

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \boxed{\frac{2}{r-l}} & 0 & 0 & \frac{-2l}{r-l} - 1 \\ 0 & \boxed{\frac{2}{t-b}} & 0 & \frac{-2b}{t-b} - 1 \\ 0 & 0 & \boxed{\frac{2}{f-n}} & \frac{-2n}{f-n} - 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Scaling στον x άξονα = $\dfrac{\text{Πλάτος του Canonical Volume}}{\text{Πλάτος του View Volume}} = \dfrac{1-(-1)}{r-l} = \dfrac{2}{r-l}$

Scaling στον y άξονα = $\dfrac{\text{Πλάτος του Canonical Volume}}{\text{Πλάτος του View Volume}} = \dfrac{1-(-1)}{t-b} = \dfrac{2}{t-b}$

Scaling στον z άξονα = $\dfrac{\text{Πλάτος του Canonical Volume}}{\text{Πλάτος του View Volume}} = \dfrac{1-(-1)}{f-n} = \dfrac{2}{f-n}$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & \boxed{\frac{-2l}{r-l} - 1} \\ 0 & \frac{2}{t-b} & 0 & \boxed{\frac{-2b}{t-b} - 1} \\ 0 & 0 & \frac{2}{f-n} & \boxed{\frac{-2n}{f-n} - 1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Μετατόπιση στον x άξονα

Μετατόπιση στον z άξονα

Μετατόπιση στον y άξονα

Παρατηρούμε πχ ότι :

- αν x = l (left) τότε x’ = 2l/(r-l) – 2l/(r-l) – 1 = -1
- ενώ αν x=r , τότε x’ = 2r/(r-l) – 2l/(r-l) -1 = 2(r-l)/(r-l) – 1 = 1

Οπότε βλέπουμε ότι το mapping του left → right του View Volume γίνεται σωστά στο -1 → 1 του canonical volume

Projection Transformations

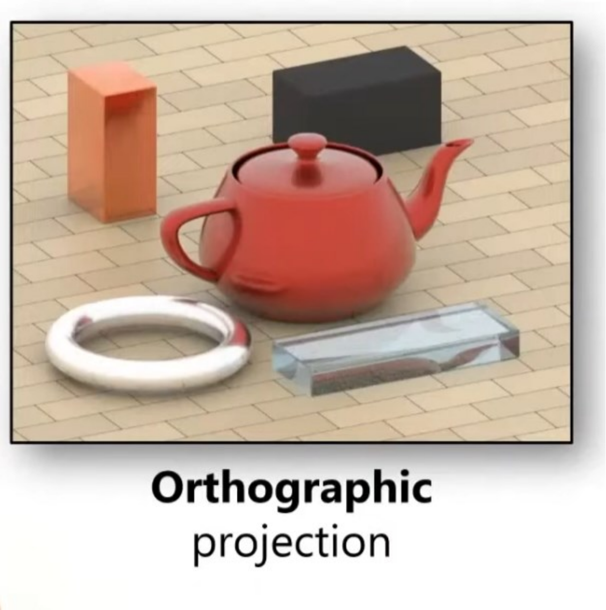
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & \frac{-2l}{r-l} - 1 \\ 0 & \frac{2}{t-b} & 0 & \frac{-2b}{t-b} - 1 \\ 0 & 0 & \frac{2}{f-n} & \frac{-2n}{f-n} - 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

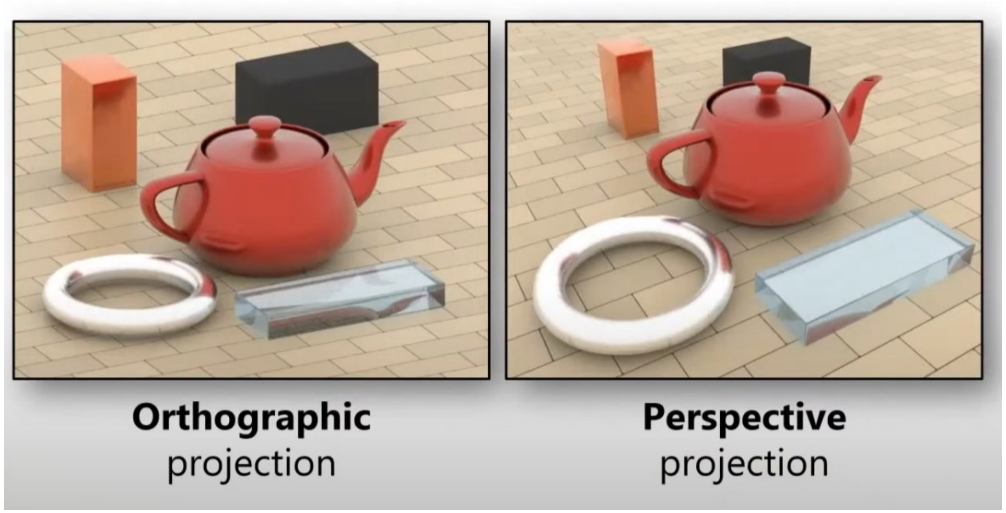
Εναλλακτικά ο πίνακας μετασχηματισμού του orthographic projection μπορεί να γραφεί και έτσι .

πχ
-2l/(r-l) – 1 = -2l/(r-l) – (r-l)/(r-l) =
(-2l – r + l) / (r-l) = (-l-r)/(r-l) = -(l+r)/(r-l)

Projection Transformations



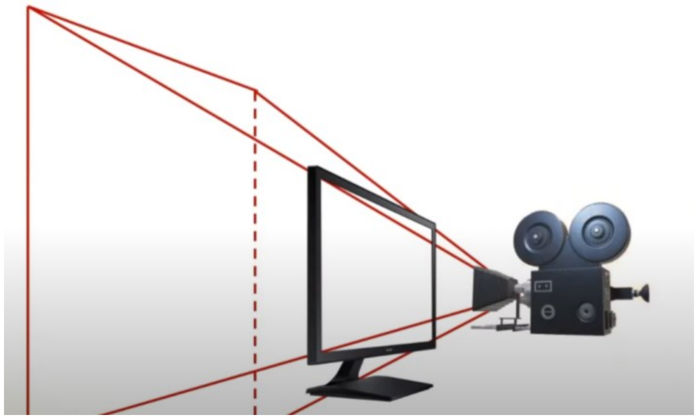
Η ορθογραφική προβολή **δεν περιέχει καθόλου προοπτική , οπότε δεν είναι σύμφωνη με τον τρόπο που βλέπουμε τρισδιάστατα.**
Οι παράλληλες γραμμές παραμένουν παράλληλες.



Η perspective projection παράγει ένα πολύ πιο φυσικό τρόπο απεικόνισης των 3D αντικειμένων, με προοπτική βάθους, παρόμοια με τον τρόπο που βλέπουμε στην καθημερινή μας ζωή.

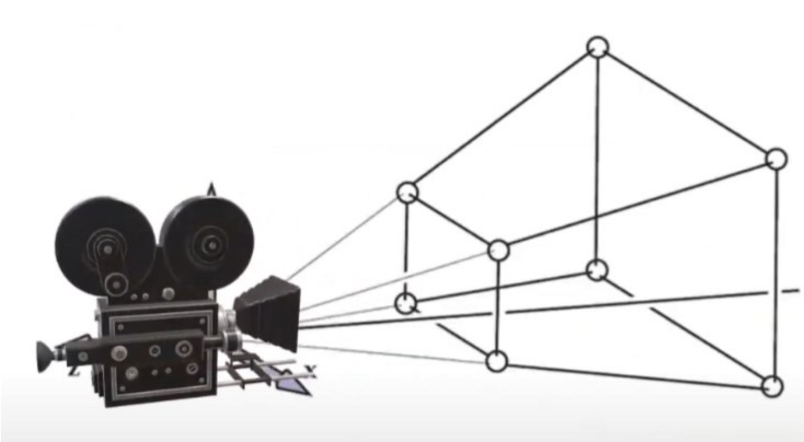
Projection Transformations

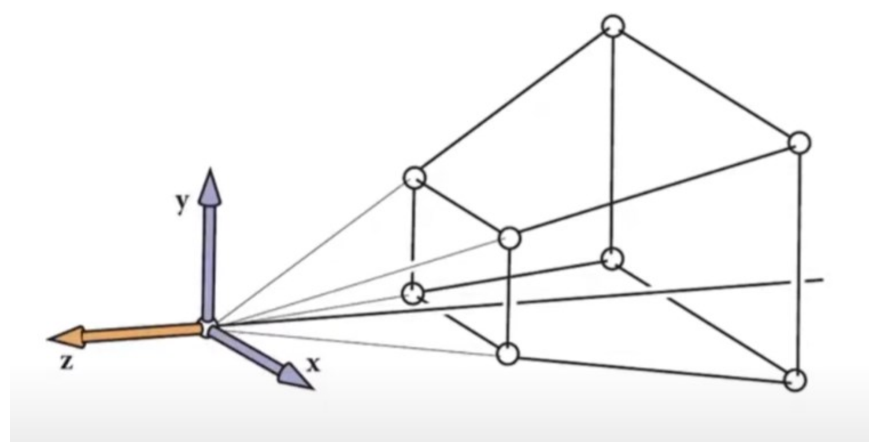
Perspective Projection

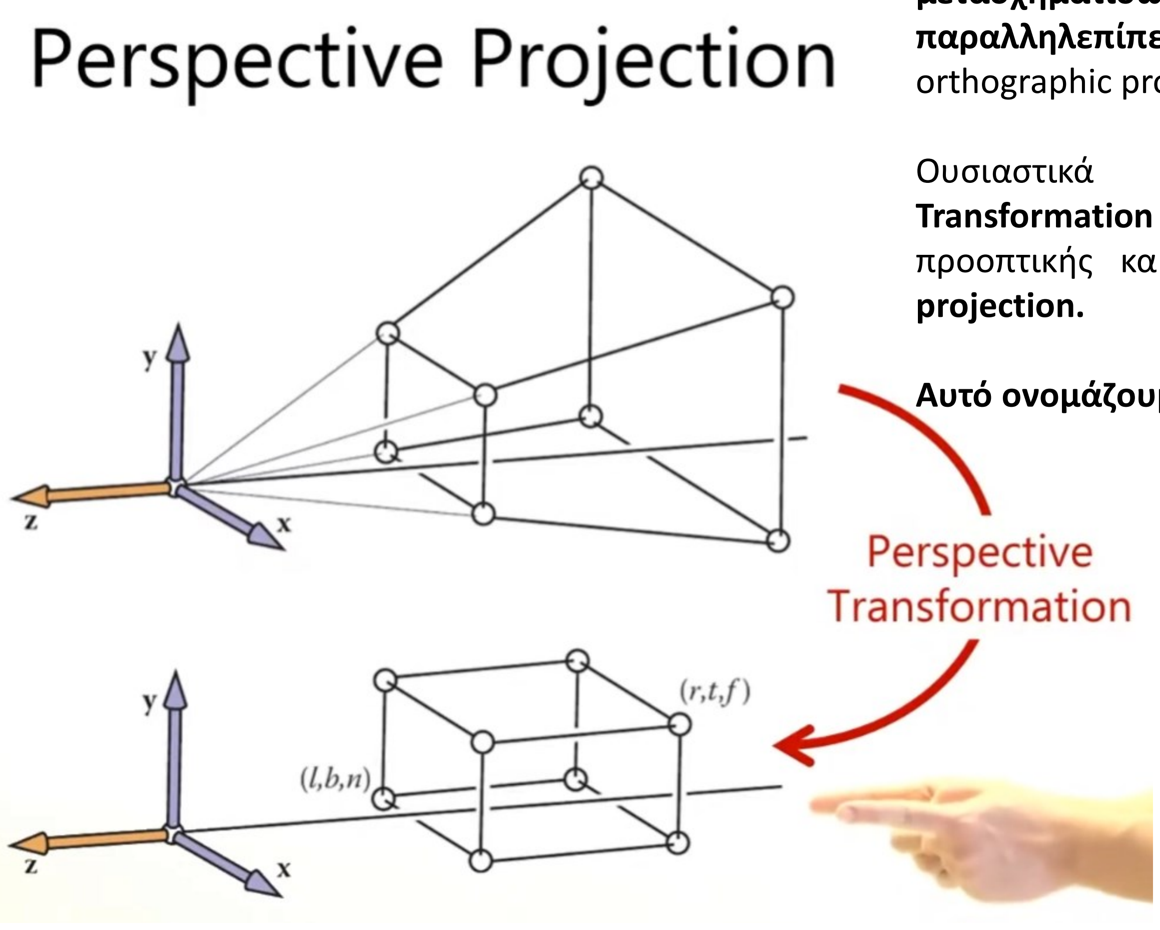


Ο λόγος που βλέπουμε με προοπτική είναι ότι τα μάτια μας έχουν μικρή επιφάνεια (το ίδιο και οι φακοί των καμερών).

Το οπτικό πεδίο μοιάζει με τραπέζιο και όχι με ορθογώνιο παραλληλεπίπεδο που βλέπαμε στο orthographic projection.





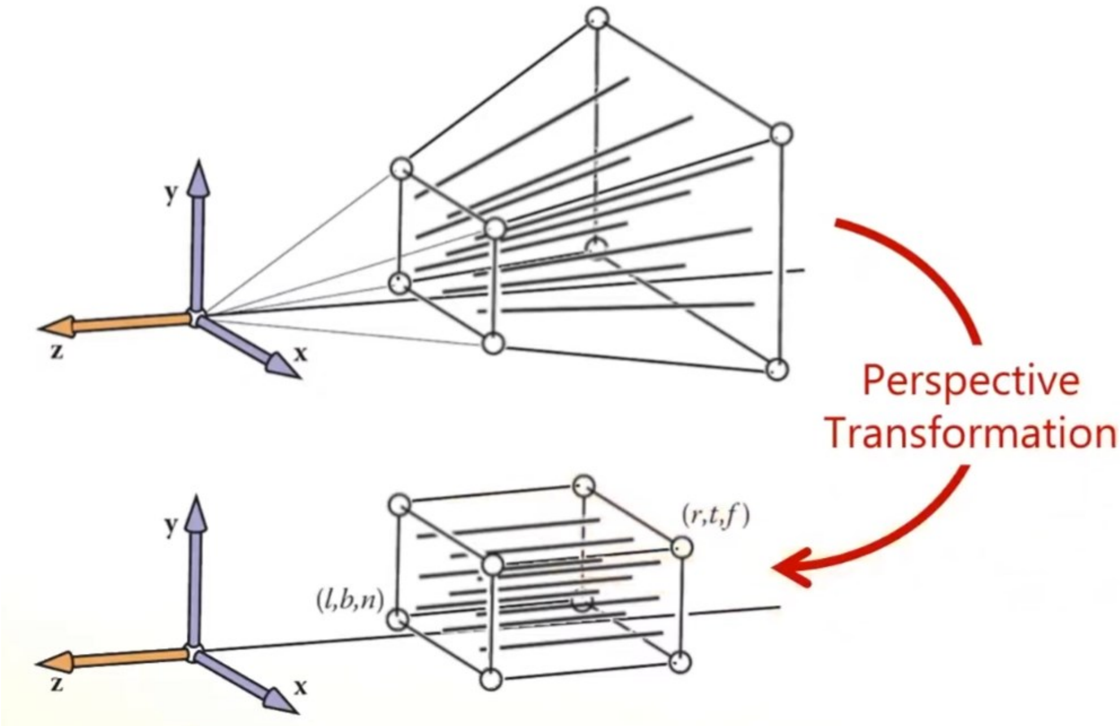


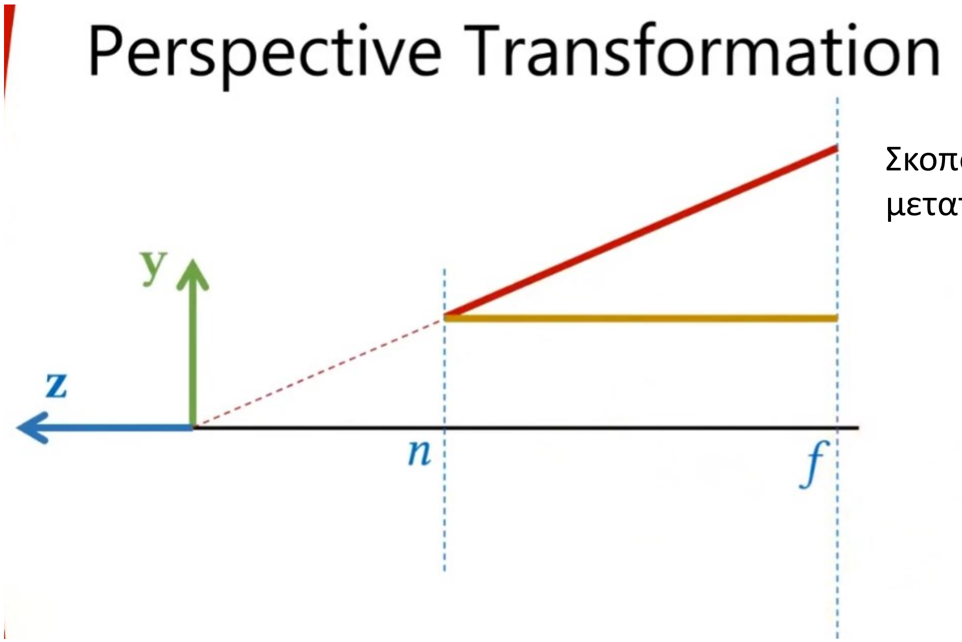
Σκοπός λοιπόν του Perspective Projection είναι να μετασχηματίσω το τραπέζιο σε ένα ορθογώνιο παραλληλεπίπεδο το οποίο μπορώ να χειριστώ όπως στο orthographic projection.

Ουσιαστικά λοιπόν εφαρμόζουμε **Perspective Transformation** για να εισάγουμε μια παραμόρφωση προοπτικής και στη συνέχεια κάνουμε **orthographic projection.**

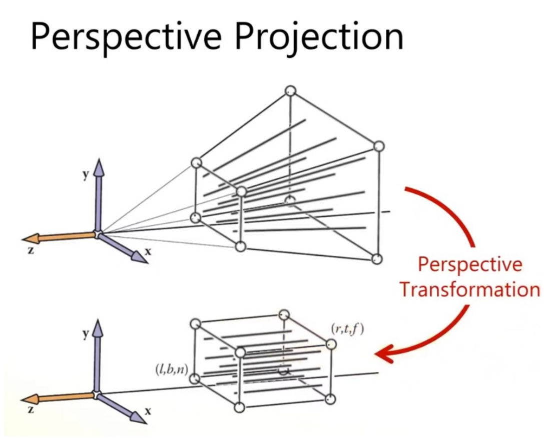
Αυτό ονομάζουμε συνολικά ως «Perspective Projection»

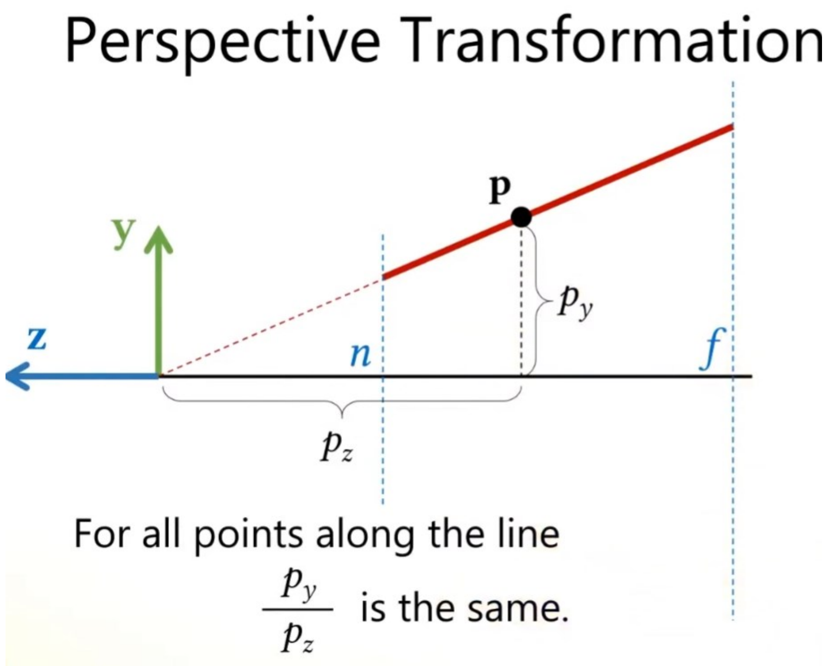
Perspective Projection





Σκοπός μας με το Perspective Transformation είναι να μετατρέψουμε την κεκλιμένη «ακτίνα» σε οριζόντια.

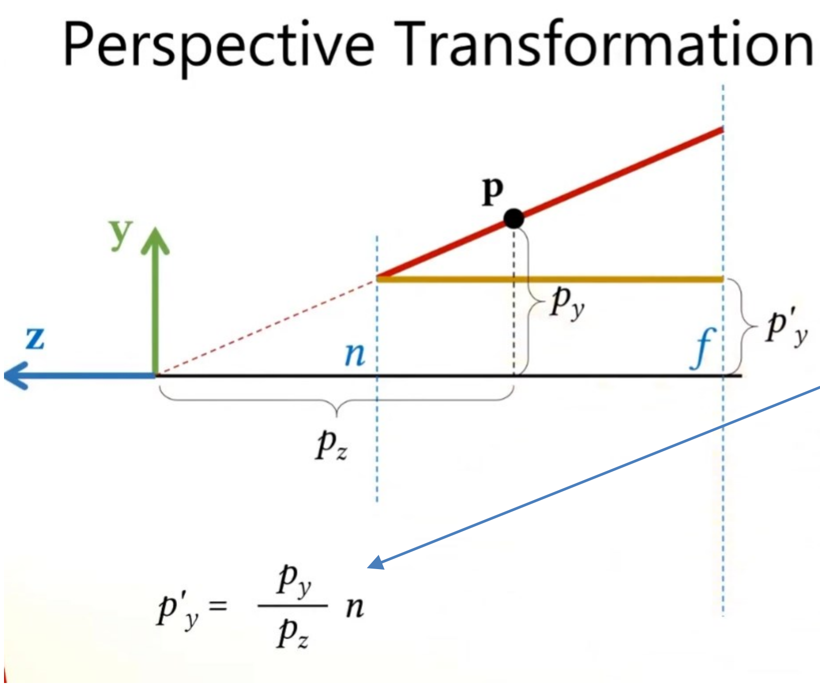




Αν πάρουμε ένα οποιοδήποτε σημείο p κατά μήκος του κεκλιμένου τμήματος, οι συντεταγμένες του Py και Pz θα έχουν σταθερό λόγο. Δηλαδή το Py/Pz παραμένει σταθερό σε όλο το μήκος .

Αυτό συμβαίνει αφού ο λόγος Py/Pz είναι η εφαπτομένη της γωνίας που δεν αλλάζει αφού η γωνία είναι η ίδια.

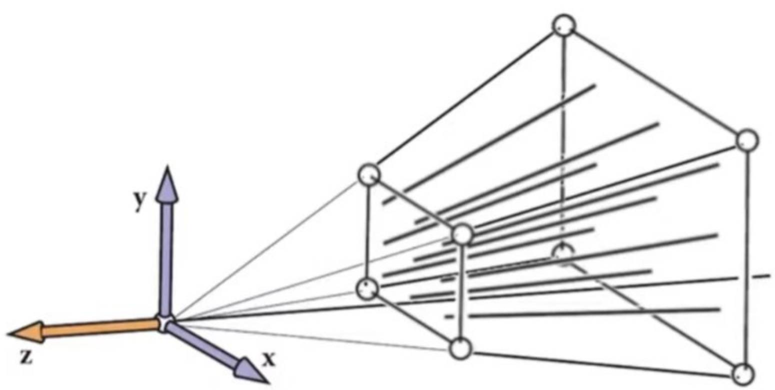
Projection Transformations



Μετασχηματίζοντας το κεκλιμένο τμήμα στο παράλληλο , όλα τα σημεία του παράλληλου τμήματος θα έχουν το ίδιο y (P’y).

Το P’y δίνεται από τον τύπο (Py/Pz) n , αφού :

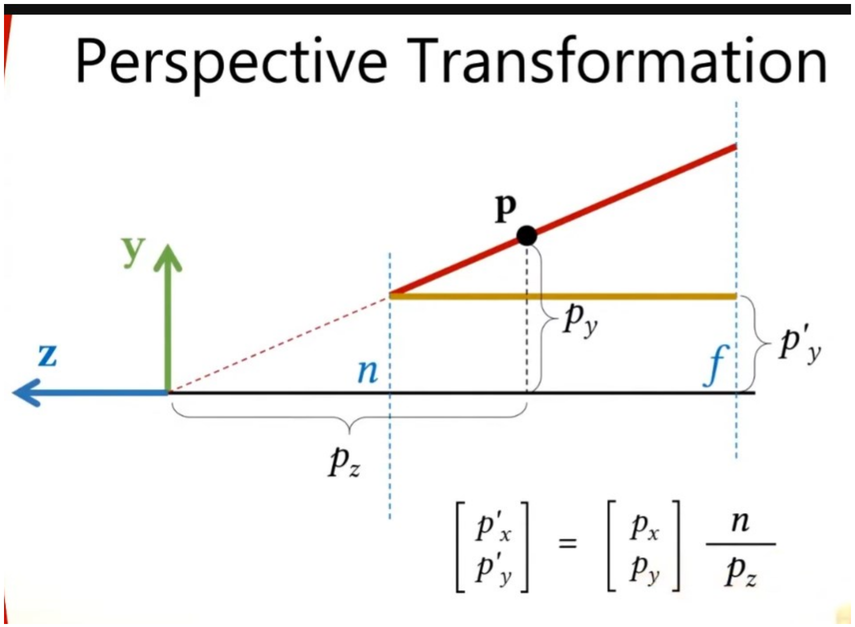
Py/Pz = P’y/n → P’y = n * (Py/Pz)



Αυτό που περιγράψαμε στο επίπεδο YZ ισχύει συμμετρικά και στο επίπεδο XZ καθώς τα κεκλιμένα τμήματα συγκλίνουν προς το κέντρο και ως προς το YZ και ως προς το XZ.

$$p'_y = \frac{p_y}{p_z} n \qquad p'_x = \frac{p_x}{p_z} n$$

Projection Transformations



Τα P’x και P’y λοιπόν μπορούν να υπολογιστούν από τα Px, Py και Pz.

Το ό,τι διαιρούμε τα Px και Py με το Pz δε μας βολεύει στην προσπάθεια να εκφράσουμε τον μετασχηματισμό με τη μορφή ενός transformation array.

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \alpha\, p_x \\ \alpha\, p_y \\ \alpha\, p_z \\ \alpha \end{bmatrix}$$

Επεκτείνουμε λοιπόν το «κόλπο» των Homogenous Coordinates και χρησιμοποιούμε μια τιμή «α» με την οποία πολλαπλασιάζουμε όλες τις συντεταγμένες.

Projection Transformations

$$\begin{bmatrix} p'_x \\ p'_y \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \end{bmatrix} \frac{n}{p_z}$$

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \alpha\, p_x \\ \alpha\, p_y \\ \alpha\, p_z \\ \alpha \end{bmatrix}$$

$$\begin{bmatrix} p'_x \\ p'_y \\ p'_z \\ 1 \end{bmatrix} = \begin{bmatrix} n\, p_x / p_z \\ n\, p_y / p_z \\ ? \\ 1 \end{bmatrix} \equiv \begin{bmatrix} n\, p_x \\ n\, p_y \\ ? \\ p_z \end{bmatrix}$$

Χρησιμοποιούμε λοιπόν για α το Pz

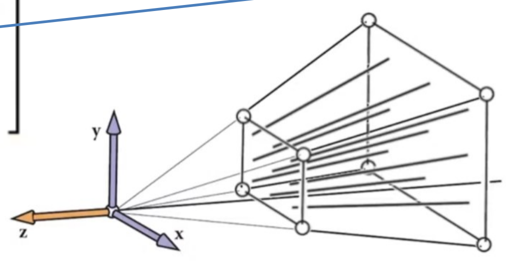
Σημαίνει «αντιστοιχεί» - «είναι ανάλογο»

Το παραπάνω μπορεί να γραφεί με χρήση transformation matrix έτσι:

$$\begin{bmatrix} p'_x \\ p'_y \\ p'_z \\ 1 \end{bmatrix} \equiv \begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & \boxed{? \quad ?} \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$

Όλα αυτά είναι πολλαπλασιασμένα με Pz – Εξού και δε λύνω το πρόβλημα βάζοντας ? ? → 1 0

Δεν ξέρω όμως τι να κάνω ακόμα με το P’z.



Το Pz ιδανικά θα ήθελα να το διατηρήσω ανέπαφο αφού αφενός δεν εμπλέκεται στη δημιουργία της αίσθηση προοπτικής οπότε δεν επηρεάζεται από το perspective transformation.

$$\begin{bmatrix} p'_x \\ p'_y \\ p'_z \\ 1 \end{bmatrix} \equiv \begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & \boxed{n+f \quad -fn} \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$

$p_z = n \Rightarrow$

$$\begin{aligned} p'_z &= ((n+f)\, p_z - fn) / p_z \\ &= (n+f) - fn / p_z \\ &= (n+f) - f \\ &= n \end{aligned}$$

$p_z = f \Rightarrow$

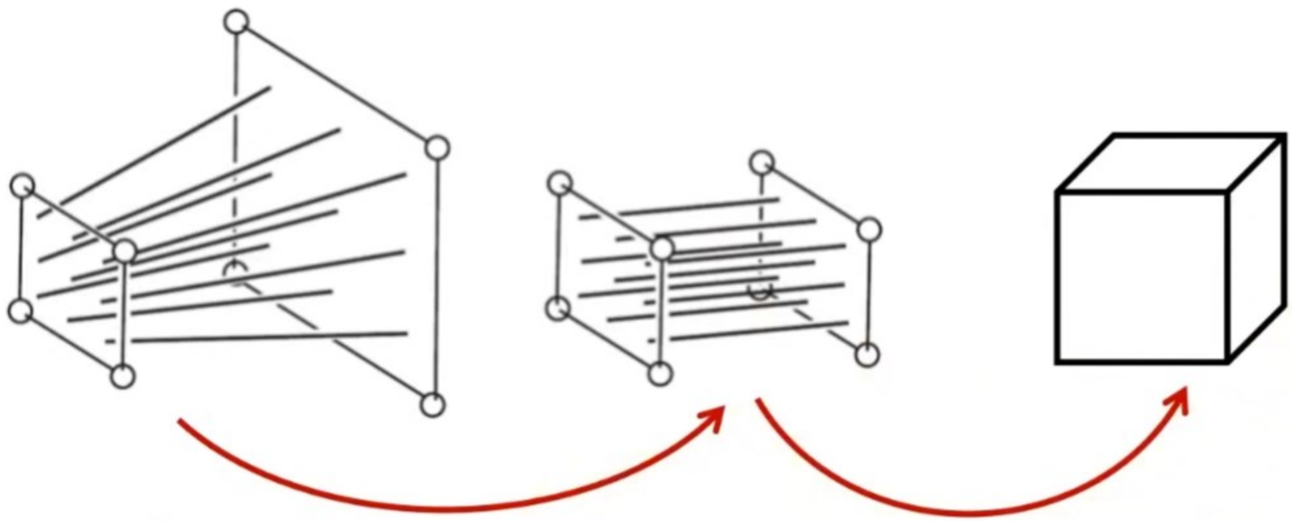
$$\begin{aligned} p'_z &= (n+f) - n \\ &= f \end{aligned}$$

Βάζοντας αυτές τις τιμές, πετυχαίνουμε στα δύο άκρα του άξονα z , στα n και f, να έχουμε: P’z = Pz.

Ενδιάμεσα θα υπάρχει κάποια παραμόρφωση στις τιμές Pz (δηλαδή P’z ≠ Pz) αλλά δε μας πειράζει.

Projection Transformations

Perspective Projection



Perspective transformation

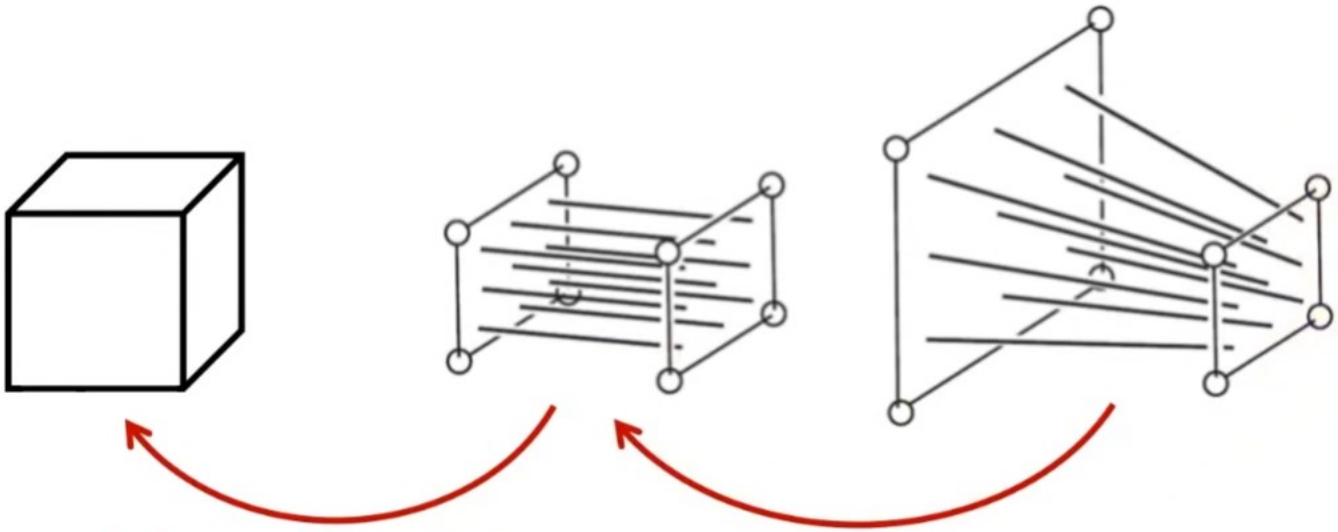
$$\begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -fn \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Orthographic Projection

$$\begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Projection Transformations

Perspective Projection



$$\begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -fn \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Αυτή είναι πιο σωστή απεικόνιση καθώς πρώτα πολλαπλασιάζουμε με τον πίνακα του Perspective transformation και μετά με του orthographic projection

Projection Transformations