



Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Γραφική με Υπολογιστές

Εργασία 2 - Μετασχηματισμοί και προβολές

Λώλος Ιωάννης 10674

26 Μαΐου 2025

Περιεχόμενα

1	Σκοπός της εργασίας	2
2	Υλοποίηση των συναρτήσεων	3
2.1	Η συνάρτηση <code>translate</code>	3
2.2	Η συνάρτηση <code>rotate</code>	3
2.3	Η συνάρτηση <code>compose</code>	3
2.4	Η συνάρτηση <code>convert2affine</code>	4
2.5	Η συνάρτηση <code>world2view</code>	4
2.6	Η συνάρτηση <code>lookat</code>	5
2.7	Η συνάρτηση <code>perspective_project</code>	6
2.8	Η συνάρτηση <code>rasterize</code>	7
2.9	Η συνάρτηση <code>render_object</code>	8
3	Αποτελέσματα	10

Σκοπός της εργασίας

Η παρούσα εργασία έχει ως στόχο την υλοποίηση ενός βασικού pipeline απόδοσης 3D σκηνών. Στα πλαίσια της εργασίας αναπτύχθηκαν συναρτήσεις γεωμετρικών μετασχηματισμών, μετατροπής συντεταγμένων από το WCS στο σύστημα της κάμερας, προσανατολισμού της κάμερας με βάση τη συνάρτηση `lookat`, προβολής με `pinhole` κάμερα, καθώς και `rasterization` για την απεικόνιση των σημείων σε εικόνα.

Η τελική συνάρτηση `render_object` συνδυάζει όλα τα παραπάνω βήματα και παράγει την απεικόνιση ενός αντικειμένου σε μορφή εικόνας.

Για την αξιολόγηση των αλγορίθμων, υλοποιήθηκαν δύο `demos`, τα οποία προσομοιώνουν ένα αυτοκίνητο κινούμενο σε κυκλική τροχιά, με μία κάμερα πάνω του, η οποία είτε παραμένει στραμμένη μπροστά στην κατεύθυνση κίνησης, είτε παρακολουθεί ένα σταθερό σημείο στον χώρο.

Η αναφορά περιγράφει την υλοποίηση των συναρτήσεων, τον τρόπο λειτουργίας τους και τα αποτελέσματα των προσομοιώσεων.

Υλοποίηση των συναρτήσεων

2.1 Η συνάρτηση `translate`

Η `translate` επιστρέφει έναν πίνακα 4×4 που αναπαριστά μία μεταφορά στον τρισδιάστατο χώρο. Το διάνυσμα μεταφοράς $t_{vec} = (t_x, t_y, t_z)$ ενσωματώνεται στον πίνακα μετασχηματισμού ως εξής:

$$T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

Ο έλεγχος ορθότητας διασφαλίζει ότι το διάνυσμα t_{vec} έχει μορφή $(3, 1)$.

2.2 Η συνάρτηση `rotate`

Η `rotate` δημιουργεί έναν πίνακα μετασχηματισμού 4×4 που αναπαριστά περιστροφή γύρω από αυθαίρετο άξονα $u = (u_x, u_y, u_z)$, κατά γωνία θ (σε rad), γύρω από ένα σημείο $c = (c_x, c_y, c_z)$. Η περιστροφή βασίζεται στον τύπο του Rodrigues:

$$R = I \cdot \cos \theta + (1 - \cos \theta)uu^\top + [u]_\times \sin \theta \quad (2.2)$$

όπου $[u]_\times$ είναι ο αντίστροφος πίνακας του u . Ο τελικός affine πίνακας είναι:

$$T = \begin{bmatrix} R & c - R \cdot c & 0 & 1 \end{bmatrix} \quad (2.3)$$

Υπάρχουν έλεγχοι εγκυρότητας για το σχήμα των παραμέτρων και για την περίπτωση μηδενικού διανύσματος άξονα.

2.3 Η συνάρτηση `compose`

Η `compose` εκτελεί σύνθεση δύο affine μετασχηματισμών 4×4 , εφαρμόζοντας πρώτα τον πίνακα A του πρώτου και κατόπιν τον πίνακα B του δεύτερου. Ο

τελικός μετασχηματισμός είναι:

$$T = A \cdot B \quad (2.4)$$

Απαιτείται οι πίνακες εισόδου να έχουν σχήμα $(4, 4)$, αλλιώς η συνάρτηση εγείρει σφάλμα.

2.4 Η συνάρτηση `convert2affine`

Η `convert2affine` δεν αποτελεί μέρος της εργασίας, αλλά αναπτύχθηκε ως βοηθητική συνάρτηση. Συνδυάζει έναν πίνακα περιστροφής $R \in R^{3 \times 3}$ και ένα διάνυσμα μεταφοράς $t \in R^3$ σε έναν affine μετασχηματισμό $T \in R^{4 \times 4}$ της μορφής:

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \quad (2.5)$$

Η συνάρτηση περιλαμβάνει ελέγχους εγκυρότητας για τις διαστάσεις των R και t .

2.5 Η συνάρτηση `world2view`

Η `world2view` μετασχηματίζει σημεία από το παγκόσμιο σύστημα συντεταγμένων στο σύστημα συντεταγμένων της κάμερας (view frame). Για κάθε σημείο $p \in R^3$, εφαρμόζεται ο εξής μετασχηματισμός:

$$p_{cam} = R^\top (p - c_0) \quad (2.6)$$

όπου R είναι ο πίνακας περιστροφής της κάμερας σε σχέση με το παγκόσμιο σύστημα και c_0 το κέντρο της κάμερας σε παγκόσμιες συντεταγμένες. Η υλοποίηση στηρίζεται στη δημιουργία ενός affine μετασχηματισμού T μέσω της συνάρτησης `convert2affine`, με:

$$T = \begin{bmatrix} R^\top & -R^\top c_0 \\ 0 & 1 \end{bmatrix} \quad (2.7)$$

Τα σημεία pts επεκτείνονται σε ομογενείς συντεταγμένες και κατόπιν μετασχηματίζονται με τον πίνακα T . Η τελική έξοδος αγνοεί τη 4η συντεταγμένη και επιστρέφει σημεία στο χώρο της κάμερας.

Απαιτείται:

- $pts \in R^{N \times 3}$: τα N σημεία προς μετασχηματισμό,

- $R \in R^{3 \times 3}$: πίνακας περιστροφής,
- $c_0 \in R^3$: κέντρο της κάμερας.

Η συνάρτηση επιστρέφει πίνακα $N \times 3$ με τα σημεία στο σύστημα της κάμερας.

2.6 Η συνάρτηση `lookat`

Η `lookat` υπολογίζει το σύστημα συντεταγμένων της κάμερας (δηλαδή τον πίνακα περιστροφής R και το διάνυσμα θέσης t) ώστε η κάμερα να «κοιτάζει» από τη θέση `eye` προς το σημείο `target`, με την καθορισμένη κατεύθυνση `up` ως αναφορά για τον κατακόρυφο άξονα.

Τα ορίσματα της συνάρτησης είναι:

- z_c : διάνυσμα θέασης (από `eye` προς `target`),
- x_c : διάνυσμα δεξιάς κατεύθυνσης, κάθετο στο επίπεδο που ορίζεται από τα `up` και z_c ,
- y_c : διάνυσμα κατακόρυφης κατεύθυνσης της κάμερας, ορθογώνιο στο z_c .

Τα διανύσματα υπολογίζονται ως εξής:

$$z_c = \frac{\text{target} - \text{eye}}{|\text{target} - \text{eye}|} \quad y_c = \frac{\text{up} - (\text{up} \cdot z_c)z_c}{|\text{up} - (\text{up} \cdot z_c)z_c|} \quad x_c = y_c \times z_c \quad (2.8)$$

Ο πίνακας περιστροφής $R \in R^{3 \times 3}$ σχηματίζεται με στήλες τα διανύσματα x_c, y_c, z_c :

$$R = [x_c \quad y_c \quad z_c] \quad (2.9)$$

Το διάνυσμα θέσης της κάμερας t είναι απλώς το `eye` αναδιαμορφωμένο ως διάνυσμα γραμμής:

$$t = \text{eye}^\top \in R^{1 \times 3} \quad (2.10)$$

Η συνάρτηση επιστρέφει:

- $R \in R^{3 \times 3}$: τον πίνακα περιστροφής,
- $t \in R^{1 \times 3}$: τη θέση της κάμερας στο WCS.

2.7 Η συνάρτηση `perspective_project`

Η `perspective_project` υλοποιεί την προβολή 3D σημείων από το παγκόσμιο σύστημα συντεταγμένων στο επίπεδο εικόνας μιας κάμερας, βάσει του μοντέλου `pinhole`.

Πιο συγκεκριμένα, η συνάρτηση λαμβάνει ως είσοδο:

- Ένα πίνακα $N \times 3$ με 3D σημεία στο παγκόσμιο σύστημα,
- Την focal distance f της κάμερας,
- Έναν πίνακα περιστροφής $R \in R^{3 \times 3}$,
- Ένα διάνυσμα μεταφοράς $t \in R^3$.

Η διαδικασία περιλαμβάνει τα εξής βήματα:

1. Τα σημεία μετασχηματίζονται από το παγκόσμιο σύστημα στο σύστημα της κάμερας μέσω της συνάρτησης `world2view`:

$$cam_pts = RT(p - c_0) \quad (2.11)$$

2. Από τα σημεία $cam_pts = (X, Y, Z)$ στο σύστημα της κάμερας, εφαρμόζεται προοπτική προβολή:

$$x = f \cdot \frac{X}{Z}, \quad (2.12)$$

$$y = f \cdot \frac{Y}{Z} \quad (2.13)$$

3. Τα τελικά δισδιάστατα σημεία προκύπτουν ως:

$$proj_pts = \begin{bmatrix} x & y \end{bmatrix} \in R^{N \times 2}$$

Η συνάρτηση επίσης επιστρέφει τα βάθη Z των σημείων στο σύστημα της κάμερας, που είναι απαραίτητα για την ανακατασκευή ή έλεγχο εγκυρότητας προβολών.

Έλεγχοι εγκυρότητας:

- Αν κάποιο σημείο έχει $Z = 0$, προκαλείται σφάλμα `ZeroDivisionError`, καθώς οδηγεί σε προβολή στο άπειρο.

Η συνάρτηση επιστρέφει:

- $proj_pts \in R^{N \times 2}$: τις προβαλλόμενες δισδιάστατες συντεταγμένες της εικόνας,
- $Z \in R^N$: τα αντίστοιχα βάθη των σημείων στο σύστημα της κάμερας.

2.8 Η συνάρτηση `rasterize`

Η συνάρτηση `rasterize` μετατρέπει δισδιάστατες συντεταγμένες σημείων στο επίπεδο εικόνας της κάμερας σε διακριτές συντεταγμένες pixel.

Είσοδοι

- `pts_2d` $\in R^{N \times 2}$: Πίνακας με N σημεία (x, y) στο επίπεδο εικόνας σε WCS.
- `plane_w` > 0 : Πλάτος του αισθητήρα.
- `plane_h` > 0 : Ύψος του αισθητήρα.
- `res_w` $\in N$: Οριζόντια ανάλυση (αριθμός pixel κατά πλάτος).
- `res_h` $\in N$: Κάθετη ανάλυση (αριθμός pixel κατά ύψος).

Διαδικασία μετατροπής

Για κάθε σημείο (x, y) :

1. Μετασχηματίζεται σε κανονικοποιημένες συντεταγμένες (u, v) στο διάστημα $[0, 1]$: $u = \frac{x + \frac{w}{2}}{w}, v = \frac{y + \frac{h}{2}}{h}$. Δηλαδή θεωρείται ότι το κέντρο του επιπέδου εικόνας είναι στο $(0, 0)$ και οι τιμές $x \in [-\frac{w}{2}, \frac{w}{2}]$, $y \in [-\frac{h}{2}, \frac{h}{2}]$.

2. Οι κανονικοποιημένες συντεταγμένες μετατρέπονται σε συντεταγμένες pixel:

$$\text{col} = \lfloor u \cdot \text{res}_w \rfloor, \quad \text{row} = \lfloor (1 - v) \cdot \text{res}_h \rfloor$$

Ο όρος $1 - v$ εφαρμόζεται για να αντιστραφεί ο κατακόρυφος άξονας y , καθώς στο σύστημα εικόνας το $(0, 0)$ είναι επάνω αριστερά.

3. Τέλος, γίνεται περιορισμός των συντεταγμένων ώστε να παραμένουν εντός της επιτρεπτής περιοχής:

$$\text{col} \in [0, \text{res}_w - 1], \quad \text{row} \in [0, \text{res}_h - 1]$$

2.9 Η συνάρτηση `render_object`

Η συνάρτηση `render_object` υλοποιεί έναν βασικό pipeline rasterization για την απεικόνιση τρισδιάστατων αντικειμένων, χρησιμοποιώντας το μοντέλο pinhole camera. Η διαδικασία περιλαμβάνει μετασχηματισμό συντεταγμένων, προβολή προοπτικής, μετατροπή σε pixel και rasterization με την ήδη υλοποιημένη συνάρτηση `render_img`.

Είσοδοι

- $v_pos \in R^{N \times 3}$: Συντεταγμένες κορυφών του αντικειμένου στο WCS.
- $v_clr \in [0, 1]^{N \times 3}$: Χρώματα RGB των κορυφών.
- $t_pos_idx \in N^{M \times 3}$: Τριγωνικές επιφάνειες με τη μορφή δεικτών σε κορυφές.
- $plane_w, plane_h \in R_+$: Διαστάσεις του αισθητήρα.
- $res_w, res_h \in N$: Ανάλυση εξόδου σε pixels.
- $focal \in R_+$: Εστιακή απόσταση της κάμερας.
- $eye \in R^3$: Θέση της κάμερας στο WCS.
- $up \in R^3$: Κατεύθυνση “πάνω” της κάμερας.
- $target \in R^3$: Σημείο που κοιτάζει η κάμερα.
- $uvs \in R^{N \times 2}$ (προαιρετικό): Συντεταγμένες ανά κορυφή για texture shading.
- $texImg \in [0, 1]^{H \times W \times 3}$ (προαιρετικό): texture image.

Ροή του αλγορίθμου

1. **Υπολογισμός παραμέτρων κάμερας:** Η συνάρτηση `lookat` υπολογίζει τον πίνακα περιστροφής R και τη θέση t της κάμερας, βάσει του σημείου θέασης και της επιθυμητής διεύθυνσης “up”.
2. **Προβολή κορυφών στο επίπεδο εικόνας:** Η `perspective_project` εφαρμόζει προοπτική προβολή στις κορυφές ώστε να προκύψουν 2D συντεταγμένες (x, y) και βάθος z στο σύστημα της κάμερας.
3. **Rasterization συντεταγμένων:** Η `rasterize` μετατρέπει τις συντεταγμένες του επιπέδου εικόνας σε διακριτές pixel συντεταγμένες (col, row) .

4. Τελικό Rendering:

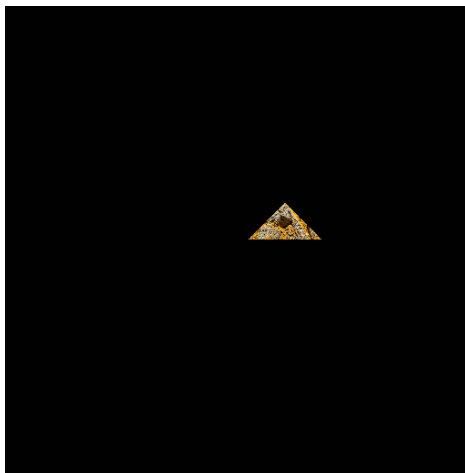
- Αν δεν παρέχονται `uvs` και `texImg`, η `render_img` καλείται με `flat shading`, αποδίδοντας χρώματα από τις κορυφές.
- Διαφορετικά, γίνεται `rasterization` με `texture shading`, με χρήση UV συντεταγμένων και δειγματοληψία από το `texImg`.

Έξοδος

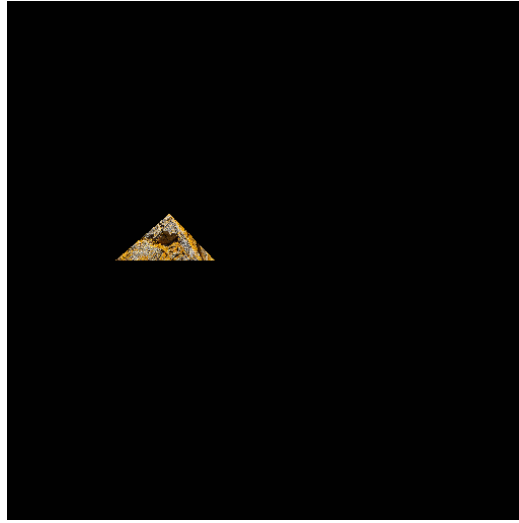
Η τελική εικόνα RGB με τιμές $[0,1]$, έτοιμη για προβολή ή αποθήκευση.

Αποτελέσματα

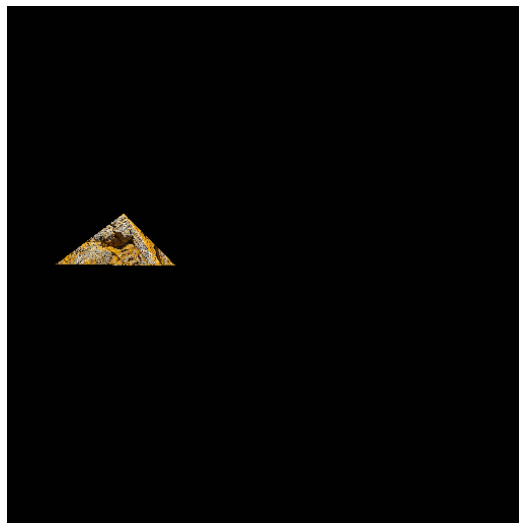
Τα αποτελέσματα της εργασίας είναι δύσκολο να αποδοθούν στην αναφορά, καθώς είναι υπό τη μορφή βίντεο. Παρακάτω παρουσιάζονται μερικά στιγμιότυπα:



Σχήμα 3.1: Frame 003



Σχήμα 3.2: Frame 100



Σχήμα 3.3: Frame 124

Η εικόνα που χρησιμοποιήθηκε ως texture image είναι η παρακάτω:



Σχήμα 3.4: Η texture εικόνα