



Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Ηλεκτρονικής και Υπολογιστών

ΓΡΑΦΙΚΗ ΜΕ ΥΠΟΛΟΓΙΣΤΕΣ

Εργασία 2^η : Μετασχηματισμοί και Προβολές

Εαρινό εξάμηνο 2023/24

Δεϊρμεντζόγλου Ιωάννης
Τομέας Ηλεκτρονικής και Υπολογιστών
Α.Ε.Μ.: 10015
Email: deirmentz@ece.auth.gr

ΕΙΣΑΓΩΓΗ

Η αναπαράσταση και απεικόνιση τρισδιάστατων αντικειμένων αποτελεί κρίσιμο κομμάτι στον κόσμο των υπολογιστικών γραφικών και της υπολογιστικής όρασης. Στην προηγούμενη εργασία ζητούμενο ήταν η δημιουργία του αλγόριθμου πλήρωσης τριγώνων και σάρωσης γραμμών για την απεικόνιση σε εικόνα ενός δοσμένου συνόλου 2D σημείων με συγκεκριμένο χρώμα σε RGB συνιστώσες το κάθε ένα. Σε αυτή την εργασία σκοπός είναι η προβολή τρισδιάστατων σκηνικών στο πέτασμα της κάμερας και η αποτύπωση της εικόνας του σκηνικού στον καμβά του υπολογιστή, δηλαδή ένα προηγούμενο βήμα πριν δοθούν οι 2D συντεταγμένες προς απεικόνιση σε καμβά υπολογιστή. Η διαδικασία περιλαμβάνει περιστροφή και μετατόπιση των προβαλλόμενων σημείων καθώς και περιστροφή και μετατόπιση της (θέσης της) εικονικής κάμερας. Για την αντιστοίχιση των 3D σημείων σε 2D χρησιμοποιείται η προοπτική προβολή που ουσιαστικά είναι η τέχνη της προβολής μιας τρισδιάστατης εικόνας και της δημιουργίας της αίσθησης του βάθους σε μια επίπεδη επιφάνεια.

A. Κλάση Μετασχηματισμών

Η κλάση Class Transform η οποία υλοποιεί τον affine μετασχηματισμό και περιέχει συναρτήσεις για περιστροφή, μετατόπιση και μετασχηματισμό 3D σημείων αποτελείται από συναρτήσεις οι οποίες περιγράφονται παρακάτω :

- **rotate(self, theta: float, u: np.ndarray)**

Η συνάρτηση αυτή δέχεται ως ορίσματα μία γωνία και ένα μοναδιαίο διάνυσμα και τροποποιεί τον πίνακα mat, ώστε να υλοποιεί περιστροφή κατά γωνία angle (rad) γύρω από τον άξονα u που διέρχεται από το κέντρο των αξόνων. Για να το κάνει αυτό, υπολογίζει τον πίνακα περιστροφής R που ανανεώνει τον πίνακα mat με βάση τον τύπο του Rodrigues :

$$\begin{aligned} \mathbf{R} &\triangleq (1 - \cos a) \begin{bmatrix} u_x^2 & u_x u_y & u_x u_z \\ u_y u_x & u_y^2 & u_y u_z \\ u_z u_x & u_z u_y & u_z^2 \end{bmatrix} + \cos a \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \sin a \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} \\ &= \begin{bmatrix} (1 - \cos a)u_x^2 + (\cos a) & (1 - \cos a)u_x u_y - (\sin a)u_z & (1 - \cos a)u_x u_z + (\sin a)u_y \\ (1 - \cos a)u_y u_x + (\sin a)u_z & (1 - \cos a)u_y^2 + (\cos a) & (1 - \cos a)u_y u_z - (\sin a)u_x \\ (1 - \cos a)u_z u_x - (\sin a)u_y & (1 - \cos a)u_z u_y + (\sin a)u_x & (1 - \cos a)u_z^2 + (\cos a) \end{bmatrix} \end{aligned}$$

- **translate(self, t: np.ndarray)**

Η συνάρτηση αυτή τροποποιεί τον πίνακα mat, ώστε να υλοποιεί μετατόπιση κατά διάνυσμα t. Ουσιαστικά το διάνυσμα μετατόπισης σε ομογενή μορφή τοποθετείται στην 4^η στήλη του πίνακα mat.

- `transform_pts(self, pts: np.ndarray)`

Η συνάρτηση αυτή δέχεται ως ορίσματα τις συντεταγμένες ενός σημείου ή μιας λίστας σημείων 3D και τα επιστρέφει αφού εφαρμόσει σε αυτά περιστροφή και μετατόπιση υπολογίζοντας το γινόμενο τους με τον 4x4 πίνακα `mat`. Να σημειωθεί ότι η συνάρτηση δέχεται σε μη ομογενή μορφή τα σημεία ($N \times 3$ διάστασης ο πίνακας `pts`) τα μετατρέπει αρχικά σε ομογενή μορφή και στην συνέχεια αφού το γινόμενο υπολογιστεί και τα μετασχηματισμένα σημεία `transformed_pts` βρίσκονται σε ομογενή μορφή επαναφέρονται σε μη ομογενή για να γίνουν έξοδος της συνάρτησης.

Γ. Συνάρτηση αλλαγής συστήματος συντεταγμένων

- `world2view(pts: np.ndarray, R: np.ndarray, c0: np.ndarray)`

Η συνάρτηση αυτή έχει ως είσοδο τις συντεταγμένες **pts** ενός ή περισσότερων 3D σημείων ως προς το αρχικό σύστημα συντεταγμένων, τις συντεταγμένες **c0** του διανύσματος μετατόπισης του συστήματος συντεταγμένων επίσης ως προς το αρχικό σύστημα και τον πίνακα περιστροφής **R**. Αρχικά, οι 3D συντεταγμένες μετατρέπονται ως προς το μετατοπισμένο σύστημα και έπειτα εφαρμόζεται η περιστροφή με τον πίνακα **R**. Οι μετασχηματισμένες συντεταγμένες ως προς το σύστημα της κάμερας είναι η έξοδος της συνάρτησης.

Δ. Συνάρτηση προσανατολισμού κάμερας

- `lookat(eye: np.ndarray, up: np.ndarray, target: np.ndarray)`

Η συνάρτηση αυτή έχει ως είσοδο το σημείο του κέντρου της κάμερας, το σημείο που «στοχεύει» η κάμερα, καθώς και τον κατακόρυφο προσανατολισμό της (*up vector*), του οποίου η προβολή στο επίπεδο της κάμερας είναι παράλληλη. Αφού μετασχηματίσει αυτή την έκφραση σε διανύσματα βάσης, με δεδομένο ότι αυτά θα είναι ορθοκανονικά. Μέσα στην *lookat* δημιουργείται ο πίνακας περιστροφής **R** με βάση τα μοναδιαία διανύσματα της κάμερας και υπολογίζεται το διάνυσμα μετατόπισης **t** όπως υποδεικνύεται από την θεωρία (ταυτίζεται με το σημείο *eye* με συντεταγμένες ως προς **WCS**).

E. Συνάρτηση προοπτικής προβολής με pinhole κάμερα

- `perspective_project(pts: np.ndarray, focal: float, R: np.ndarray, t: np.ndarray)`

Η συνάρτηση αυτή δέχεται ως ορίσματα μία λίστα από σημεία στον τρισδιάστατο χώρο ως προς το WCS, την απόσταση f του πετάσματος από το κέντρο της κάμερας, τον πίνακα περιστροφής R και το διάνυσμα μετατόπισης t . Αρχικά, καλείται η συνάρτηση `world2view()` ώστε να γίνει μεταφορά της λίστας με τα σημεία στο νέο σύστημα συντεταγμένων της κάμερας (**CCS**) από το **WCS**. Στην συνέχεια, υπολογίζεται το βάθος των σημείων εισόδου (άξονας του φακού που ορίζεται από το μοναδιαίο διάνυσμα z_c). Έπειτα, υπολογίζονται οι προοπτικές προβολές `x_perspective`, `y_perspective` όπου επιστρέφονται μαζί με το βάθος `depth`.

ΣΤ. Συνάρτηση απεικόνισης

- `rasterize(pts_2d: np.ndarray, plane_w: int, plane_h: int, res_w: int, res_h: int)`

Στη συνάρτηση `rasterize` γίνεται απεικόνιση των σημείων από το πέτασμα της κάμερας σε εικόνα, διαφορετικών διαστάσεων, η οποία είναι το αποτέλεσμα της φωτογράφισης. Έχει ως είσοδο τις διαστάσεις του πετάσματος της κάμερας και τις διαστάσεις του καμβά της εικόνας. Αρχικά, υπολογίζεται ο λόγος της κάθε φυσικής διάστασης προς την αντίστοιχη διάσταση της εικόνας (`scale_w`, `scale_h`), ο οποίος συμβολίζει τις φυσικές διαστάσεις ενός pixel. Γνωρίζοντας ότι το κέντρο της κάμερας θεωρείται το (0,0) τότε μετακινείται το πέτασμα κατά `plane_w/2` προς τα θετικά x και κατά `plane_h/2` κατά τα θετικά y για να «ευθυγραμμιστούν» οι δύο επιφάνειες και να επιτευχθεί η απεικόνιση στις νέες διαστάσεις. Τέλος, αυτές οι διαστάσεις x και y διαιρούνται με τον λόγο που βρέθηκε παραπάνω πριν και στρογγυλοποιείται το αποτέλεσμα.

Ζ. Συνάρτηση φωτογράφισης

- `render_object(v_pos, v_clr, t_pos_idx, plane_h, plane_w, res_h, res_w, focal, eye, up, target)`

Η συνάρτηση υλοποιεί την τελική απεικόνιση του αντικειμένου βάσει των δοσμένων δεδομένων αυτού. Συγκεκριμένα, καλεί την συνάρτηση προοπτικής

κάμερας με στόχο **lookat()** ώστε να υπολογίσει τον πίνακα περιστροφής **R** και το διάνυσμα μετατόπισης **t** και στην συνέχεια η **perspective_project** για τις συντεταγμένες των 2D σημείων πάνω στο πέτασμα της εικονικής κάμερας. Στην συνέχεια καλείται η συνάρτηση απεικόνισης **rasterize()** ώστε να μετατρέψει τα σημεία αυτά σε pixel της εικόνας. και τέλος την συνάρτηση της 1ης εργασίας **render_img()** με χρωματισμό κατά Gouraud για να σχηματίσει τα αντικείμενα και τέλος βγάζει στην έξοδο την εικόνα.

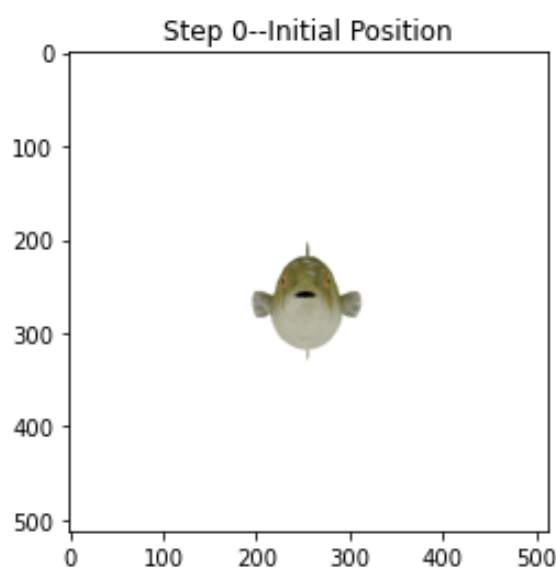
Demo

Το **demo.py** αποτελεί ένα script επίδειξης των συναρτήσεων. Αρχικά, αφού φορτώσει τα δεδομένα, 'φωτογραφίζει' το αντικείμενο στην αρχική του θέση και παράγει μια εικόνα με την συνάρτηση **render_object** και την αποθηκεύει ως **'initial_image.jpg'**.

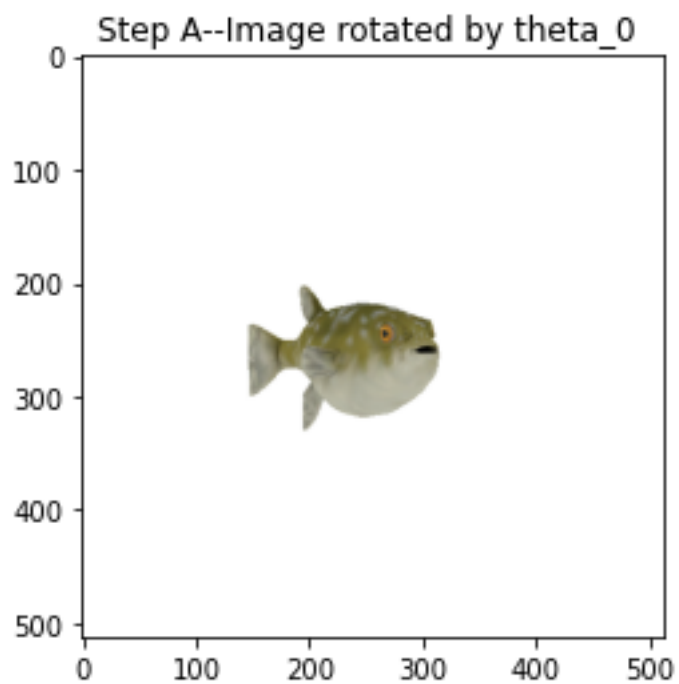
Στη συνέχεια, δημιουργεί το **transformObject (κλάση Transform)** που θα χρειαστεί στη συνέχεια. Μετά, χρησιμοποιείται η συνάρτηση **rotate** για την υλοποίηση του α) οπού εφαρμόζει περιστροφή κατά **theta_0** στο αντικείμενο φωτογράφισης και το φωτογραφίζει ξανά με την συνάρτηση **render_object** και την αποθηκεύει ως **'Step_A_Image.jpg'**. Προηγείται ο μετασχηματισμός των 3D σημείων με την συνάρτηση **transform_pts**. Με παρόμοια διαδικασία, χρησιμοποιείται η **translate** για μετατόπιση κατά **t_1, t_0** για το ψαρί φωτογραφίζοντάς το σε κάθε βήμα, δημιουργώντας άλλες 2 εικόνες, τις **'Step_B_Image.jpg'** και **'Step_C_Image.jpg'**.

Σημειώνεται ότι οι μετασχηματισμοί των σημείων του αντικειμένου γίνονται σύμφωνα με το σύστημα συντεταγμένων με το οποίο περιγράφονται τα σημεία του αντικειμένου και όχι με το σύστημα συντεταγμένων της κάμερας.

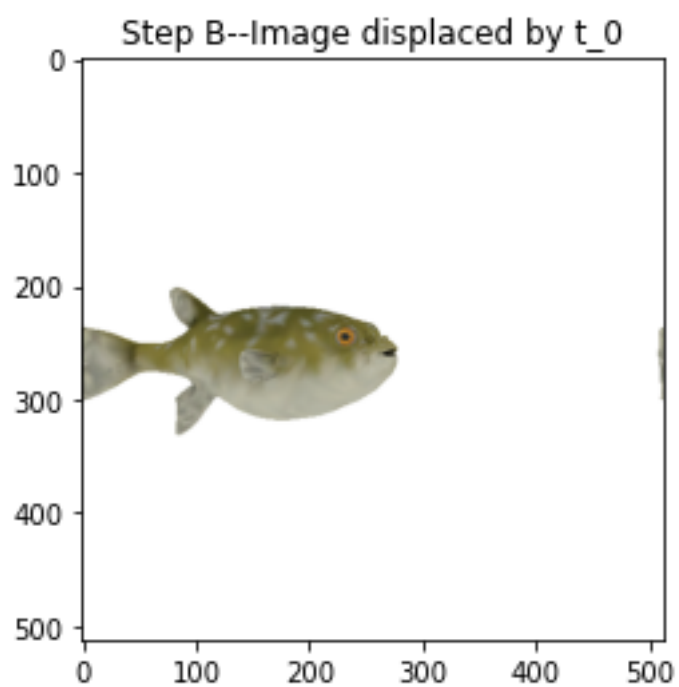
Παρακάτω παρατίθενται τα αποτελέσματα από το demo :



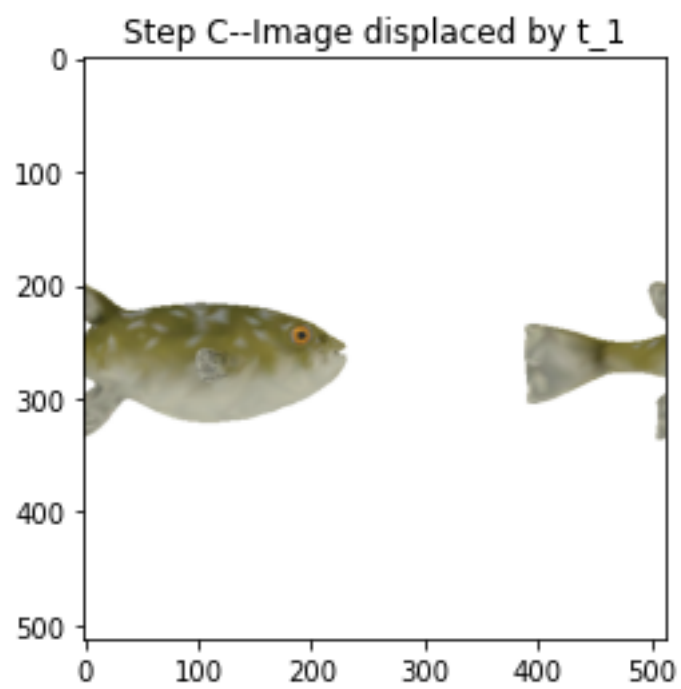
Εικόνα 1 : Αρχική εικόνα



Εικόνα 2 : Στροφή κατά $\theta_0 = 0.5235 \text{ rad}$ ως προς το $\text{rot_axis}_0 = [0,1,0]$



Εικόνα 3 : Μετατόπιση κατά διάνυσμα t_0



Εικόνα 4 : Μετατόπιση κατά διάνυσμα t_1