

Γραφική με Υπολογιστές - Μετασχηματισμοί και Προβολές

Γκούντρας Ιωάννης

AEM:10332

8 Μαΐου, 2024

Abstract

Σε αυτή την εργασία πλέον μεταφερόμαστε στις 3 διαστάσεις και φωτογραφίζουμε μέσω μίας εικονικής κάμερας ένα τρισδιάστατο αντικείμενο. Για τον χρωματισμό του και την δημιουργία της φωτογραφίας χρησιμοποιούνται συναρτήσεις από το προηγούμενο παραδοτέο.

1 Κλάση Μετασχηματισμών

Στην κλάση `Transform` υλοποιούνται οι συναρτήσεις που επιτρέπουν affine μετασχηματισμούς περιστροφής και μετατόπισης. Επιλέχθηκαν να χρησιμοποιηθούν ομογενείς συντεταγμένες.

Συγκεκριμένα για την περιστροφή υλοποιήθηκε ο τύπος του Rodrigues:

$$\mathbf{R}_h \triangleq \left[\begin{array}{c|c} \mathbf{R} & \mathbf{0}_{3 \times 1} \\ \hline \mathbf{0}_{1 \times 3} & 1 \end{array} \right]$$

Αφού βρεθεί ο \mathbf{R}_h , υπολογίζεται και ο ανάστροφος του (που ισοδυναμεί με τον αντίστροφο καθώς ο πίνακας είναι ορθοκανονικός). Τελικά χρησιμοποιείται ο ανάστροφος γιατί ζητείται δεξιόστροφη (ωρολογιακή) περιστροφή, ενώ ο τύπος του Rodrigues είναι για αριστερόστροφη (αντιωρολογιακή).

Για τον μετασχηματισμό των σημείων στην `transform_pts`, οι συντεταγμένες κάθε σημείου μετασχηματίζονται πρώτα σε ομογενείς, υπολογίζεται το εσωτερικό γινόμενο του `self.mat` με το σημείο

$$\text{self.mat} \cdot \text{point} \tag{1}$$

και τέλος επαναφέρονται σε μη ομογενείς και προστίθενται σε ένα νέο πίνακα ανανεωμένων σημείων, ο οποίος και επιστρέφεται.

2 Συνάρτηση αλλαγής συστήματος συντεταγμένων

Η συνάρτηση `world2view` για κάθε ένα από τα σημεία εισόδου, υπολογίζει το ανανεωμένο σημείο ως εξής:

$$\text{new_point} = R^T \cdot (\text{point} - c_0) \quad (2)$$

Το κάθε ανανεωμένο σημείο γίνεται `append` σε έναν καινούριο πίνακα ο οποίος και επιστρέφεται.

3 Συνάρτηση προσανατολισμού κάμερας

Η συνάρτηση `lookat` παίρνοντας σαν είσοδο τις παραμέτρους της κάμερας:

- το σημείο του κέντρου της κάμερας
- το μοναδιαίο `up` διάνυσμα της κάμερας
- το σημείο στόχος

παράγει τον πίνακα περιστροφής R που χρειάζεται για να μετασχηματιστούν τα σημεία από το WCS στο σύστημα συντεταγμένων της κάμερας και το διάνυσμα μετατόπισης t . Για να γίνει αυτό:

- Υπολογίζεται το \hat{z}_c

$$\hat{z}_c = \frac{\text{target} - \text{eye}}{|\text{target} - \text{eye}|} \quad (3)$$

- Υπολογίζεται το βοηθητικό a

$$a = \text{up} - \langle \text{up}, \hat{z}_c \rangle \hat{z}_c \quad (4)$$

- Υπολογίζεται το \hat{y}_c

$$\hat{y}_c = \frac{a}{|a|} \quad (5)$$

- Τέλος υπολογίζεται το \hat{x}_c

$$\hat{x}_c = \hat{y}_c \times \hat{z}_c \quad (6)$$

Από αυτά παράγεται ο πίνακας R ενώ το διάνυσμα μετατόπισης t ισούται με το `eye`.

4 Συνάρτηση προοπτικής προβολής με `pinhole` κάμερα

Η συνάρτηση `perspective_project` είναι υπεύθυνη για την προβολή των 3D σημείων στις 2 διαστάσεις της κάμερας.

Αρχικά τα σημεία της εισόδου μετασχηματίζονται στο σύστημα συντεταγμένων της κάμερας μέσω της `world2view`. Στη συνέχεια οι τιμές z των σημείων αποθηκεύονται σε ένα `array` γιατί αντιπροσωπεύουν το `depth` των 2D σημείων που θα προκύψουν. Για κάθε μετασχηματισμένο σημείο υπολογίζεται το νέο x και y :

$$\begin{bmatrix} x_{\text{proj}} \\ y_{\text{proj}} \end{bmatrix} = \frac{\text{focal}}{z} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \quad (7)$$

τα οποία γίνονται append σε ένα νέο array. Τέλος επιστρέφεται αυτό το array και τα depth των σημείων.

5 Συνάρτηση απεικόνισης

Η συνάρτηση `rasterise` μετασχηματίζει τα σημεία του αισθητήρα της κάμερας σε pixel στο επίπεδο της φωτογραφίας. Αρχικά υπολογίζεται το scale στους άξονες x και y :

$$\text{scale}_x = \frac{\text{res}_w}{\text{plane}_w} \quad (8)$$

$$\text{scale}_y = \frac{\text{res}_h}{\text{plane}_h} \quad (9)$$

Στη συνέχεια για κάθε σημείο του αισθητήρα προστίθεται στο x το $\frac{\text{plane}_w}{2}$ και στο y το $\frac{\text{plane}_h}{2}$ ώστε να μετακινηθούν οι άξονες, και το αποτέλεσμα πολλαπλασιάζεται με scale_x και scale_y αντίστοιχα. Τα νέα σημεία στρογγυλοποιούνται σε ακέραιο και γίνονται append σε ένα array το οποίο και επιστρέφεται.

Μπορούμε να επιβεβαιώσουμε την ορθότητα της διαδικασίας με το παράδειγμα του σχήματος 1, όπου το αρχικό σημείο είχε συντεταγμένες $(-\text{plane}_w/2, 0)$ ενώ το τελικό $(0, \text{res}_h/2)$. Η συνάρτησή μας οδηγεί από το πρώτο στο δεύτερο.

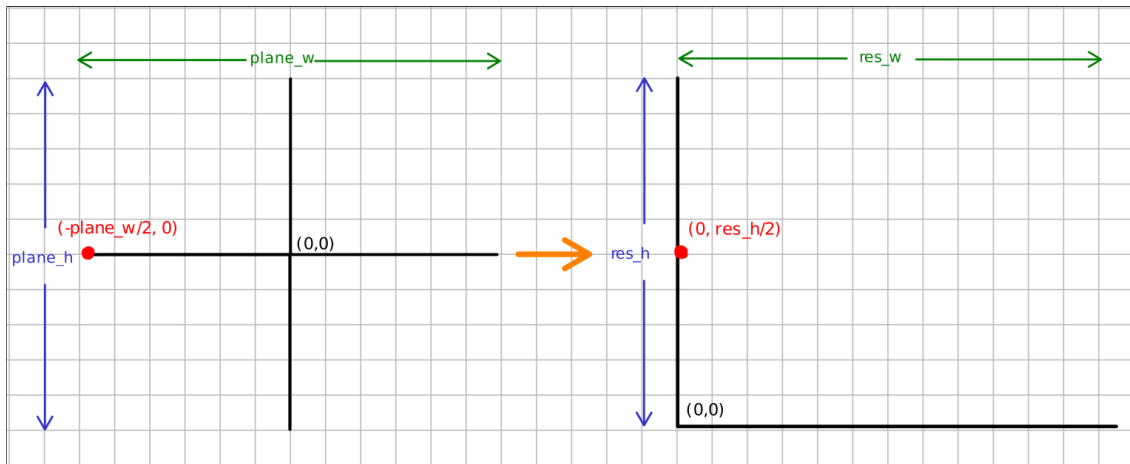


Figure 1: Παράδειγμα μετασχηματισμού σημείου από τον αισθητήρα της κάμερας σε pixel

6 Συνάρτηση φωτογράφισης

Η `render_object` είναι αυτή που παράγει την φωτογραφία. Αρχικά υπολογίζει τον R_c και τον t_c μέσω της `lookat`. Έπειτα μετασχηματίζει τα σημεία της εισόδου της στο σύστημα της κάμερας και αποθηκεύει το depth τους. Στη συνέχεια καλεί την `rasterise` ώστε να παραχθούν τα pixels και τέλος γίνεται `render` η φωτογραφία χρησιμοποιώντας τη συνάρτηση `render_img` του προηγούμενου παραδοτέου με παράμετρο χρωματισμού "g", για Gouraud shading. Η συνάρτηση επιστρέφει την φωτογραφία σαν πίνακα.

7 Demo

Στην demo αρχικά φορτώνεται το αρχείο `hw2.npy` που περιέχει όλα τα δεδομένα της εργασίας. Καλείται η `render_object` και φωτογραφίζεται το αντικείμενο στην αρχική του θέση. Στη συνέχεια αρχικοποιείται ένα αντικείμενο της κλάσης `Transform` και χρησιμοποιείται για να περιστρέψει τα σημεία `v_pos` κατά `theta rad` περί άξονα παράλληλο προς διάνυσμα `rot_axis`. Το αντικείμενο φωτογραφίζεται ξανά. Ακολουθούν δύο μετατοπίσεις, μετά από τις οποίες το αντικείμενο φωτογραφίζεται αντίστοιχα. Η μετατόπιση στο τελευταίο βήμα γίνεται κατά τον z άξονα αφού το διάνυσμα μετατόπισης είναι το $[0 \ 0 \ -1]$. Αυτό έχει σαν αποτέλεσμα το αντικείμενο να πλησιάσει την κάμερα, και να φανεί μεγαλύτερο στην φωτογραφία.

Για κάθε μετασχηματισμό χρησιμοποιείται το ίδιο αντικείμενο της κλάσης `Transform`, πράγμα που σημαίνει ότι σε κάθε βήμα ο νέος μετασχηματισμός προστίθεται στους προηγούμενους. Για αυτό τον λόγο κάθε φορά καλείται η `transform_pts` του αντικειμένου με όρισμα τα αρχικά σημεία `v_pos`, αλλιώς οι μετασχηματισμοί θα εφαρμόζονταν παραπάνω από μία φορές. Επιπλέον, σε κάθε βήμα υπολογίζεται και γίνεται `print` και ο χρόνος που χρειάστηκε για να πραγματοποιηθεί. Τέλος όλες οι φωτογραφίες αποθηκεύονται ως αρχεία `.jpg` και εμφανίζονται μέσω της `imshow` της `opencv`.

Αξίζει να σημειωθεί ότι η `OpenCV` χρησιμοποιεί διαφορετικό σύστημα συντεταγμένων από αυτό για το οποίο υλοποιείται η εργασία. Στην `OpenCV` η αρχή των αξόνων βρίσκεται **επάνω** αριστερά και ενώ τα x μεγαλώνουν από αριστερά προς τα δεξιά, τα y μεγαλώνουν από επάνω προς τα κάτω. Στις συναρτήσεις της εργασίας η αρχή των αξόνων θεωρείται **κάτω** αριστερά, με τα y να μεγαλώνουν από κάτω προς τα επάνω. Αυτή η διαφοροποίηση μπορεί να δημιουργήσει καθρεπτισμούς στον άξονα y της τελικής φωτογραφίας του αντικειμένου.

Οι φωτογραφίες που παράγονται από την demo φαίνονται παρακάτω.



Figure 2: Αρχική θέση αντικειμένου



Figure 3: Φωτογράφιση αντικειμένου μετά την περιστροφή στο βήμα α

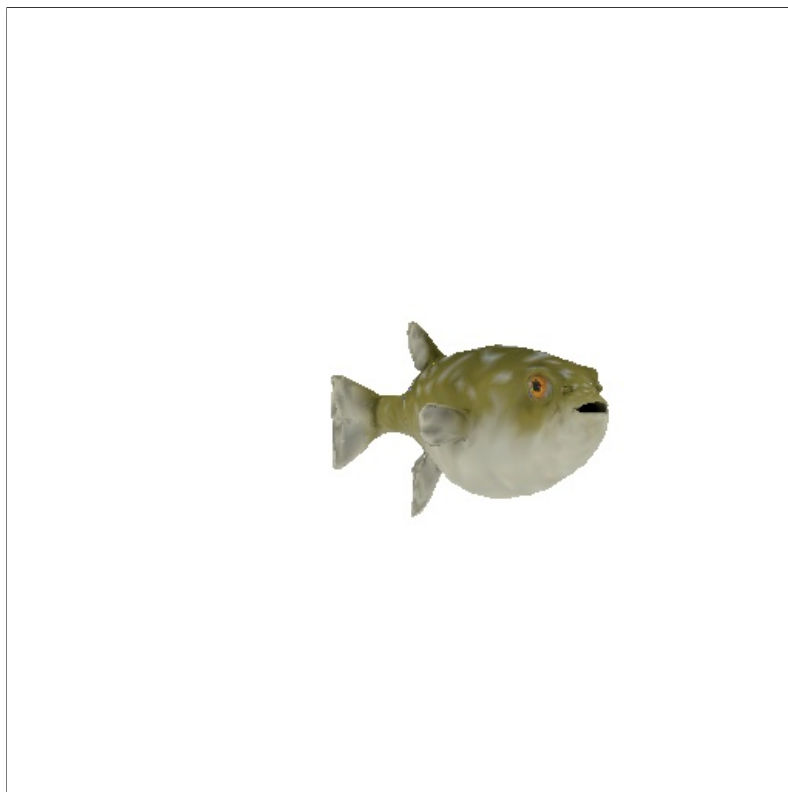


Figure 4: Φωτογράφιση αντικειμένου μετά την μετατόπιση στο βήμα β



Figure 5: Φωτογράφιση αντικειμένου μετά την μετατόπιση στο βήμα γ