

Γραφική με Υπολογιστές 2022

Εργασία #2: Μετασχηματισμοί και Προβολές

Ζητούμενα

A. Συνάρτηση μετασχηματισμού τύπου affine

Να υλοποιηθεί τη συνάρτηση

$$c_q = \text{affine_transform}(c_p, \theta, u, t)$$

η οποία μετασχηματίζει ένα σημείο $c_p \in \mathbb{R}^3$ (σε μη ομογενή μορφή), περιστρέφοντάς το κατά γωνία θ και άξονα που διέρχεται από την αρχή του συστήματος συντεταγμένων και είναι παράλληλος προ το \mathbf{u} , και μετατοπίζοντάς το σύμφωνα με το διάνυσμα μετατόπισης \mathbf{t} , όπου:

- $c_p \in \mathbb{R}^3$, είναι ένα 3×1 διάνυσμα με τις συντεταγμένες ενός σημείου p ως προς ένα σύστημα συντεταγμένων.
- θ είναι η γωνία περιστροφής.
- \mathbf{u} είναι ένα 3×1 (κανονικό) διάνυσμα που δηλώνει τον άξονα περιστροφής.
- \mathbf{t} είναι το 3×1 διάνυσμα μετατόπισης.

Η συνάρτηση θα υλοποιεί σημειακό μετασχηματισμό affine. Φροντίστε η συνάρτηση `affine_transform` να λειτουργεί σωστά και στην περίπτωση που τα c_p, c_q είναι πίνακες $3 \times N$ με τις συντεταγμένες σημείων. **Σημείωση:** Η συνάρτηση θα πρέπει να μπορεί να εφαρμόζει μετασχηματισμό είτε μόνο λόγω περιστροφής, ή μόνο λόγω μετατόπισης με κατάλληλα ορίσματα (για παράδειγμα αν $t = \text{None}$ τότε θα πρέπει να εφαρμόζει μόνο μετασχηματισμό περιστροφής, και αντίστοιχα για τη μετατόπιση).

B. Συνάρτηση μετασχηματισμού συστήματος συντεταγμένων

Να υλοποιηθεί η συνάρτηση:

$$d_p = \text{system_transform}(c_p, \theta, u, c_0)$$

όπου:

- $c_p \in \mathbb{R}^3$ είναι η 3×1 στήλη με τις συντεταγμένες ενός σημείου p ως προς ένα σύστημα συντεταγμένων.

- $d_p \in \mathbb{R}^3$ είναι οι συντεταγμένες του ίδιου σημείου ως νέο σύστημα συντεταγμένων με αρχή $o \oplus \mathbf{v}_0$, το οποίο προκύπτει από μετασχηματισμό περιστροφής κατά γωνία θ και άξονα u .
- $c_0 \in \mathbb{R}^3$ είναι η 3×1 στήλη με τις συντεταγμένες του διανύσματος \mathbf{v}_0 ως προς το αρχικό σύστημα αξόνων.

Γ. Συνάρτηση προοπτικής κάμερας

Έστω $c_p \in \mathbb{R}^3$ η 3×1 στήλη με τις συντεταγμένες ενός σημείου ως προς το WCS $\{o, \mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0\}$. Έστω, επίσης, ότι μία προοπτική κάμερα έχει κέντρο $c = o \oplus \mathbf{v}_c$ και μοναδιαία διανύσματα $\{\mathbf{x}_c, \mathbf{y}_c, \mathbf{z}_c\}$. Να υλοποιήσετε τη συνάρτηση:

$$[verts_{2D}, depth] = project_cam(f, c_v, c_x, c_y, c_z, p)$$

όπου τα c_v, c_x, c_y, c_z είναι οι συντεταγμένες των $\mathbf{v}_c, \mathbf{x}_c, \mathbf{y}_c, \mathbf{z}_c$ αντίστοιχα, ως προς το WCS, και f είναι η απόσταση του πετάσματος από το κέντρο (μετρημένη στις μονάδες που χρησιμοποιεί το σύστημα συντεταγμένων της κάμερας).

Η συνάρτηση θα παράγει τις προοπτικές προβολές των τρισδιάστατων σημείων και θα τις επιστρέφει στον πίνακα $verts_{2D}$ διάστασης $2 \times N$. Η συνάρτηση ακόμη θα υπολογίζει το βάθος κάθε σημείου πριν την προβολή του στις 2 διαστάσεις και θα το επιστρέφει στον πίνακα $depth$ διάστασης $1 \times N$. Φροντίστε η $project_camera$ να λειτουργεί σωστά και στην περίπτωση που το p είναι $3 \times N$ πίνακας με τις συντεταγμένες σημείων.

Δ. Συνάρτηση προοπτικής κάμερας

Να υλοποιήσετε τη συνάρτηση:

$$[verts_{2d}, depth] = project_cam_lookat(f, c_{org}, c_{lookat}, c_{up}, verts_{3d})$$

η οποία θα παράγει τις προοπτικές προβολές και το βάθος των σημείων του $verts_{3d}$ όπως και η προηγούμενη αλλά δέχεται ως όρισμα τις συντεταγμένες c_{lookat} και c_{up} (σε μη ομογενή μορφή) του σημείου στόχου K και του μοναδιαίου up vector αντίστοιχα. Το c_{org} περιέχει όπως και πριν τις συντεταγμένες του κέντρου της κάμερας ως προς το WCS.

Ε. Συνάρτηση απεικόνισης

Να υλοποιήσετε τη συνάρτηση:

$$verts_{rast} = rasterize(verts_{2d}, img_h, img_w, cam_h, cam_w)$$

η οποία απεικονίζει τις συντεταγμένες των σημείων από το σύστημα μιας κάμερας με πέτασμα διάστασης $cam_h \times cam_w$ (σε ίντσες), σε ακέραιες θέσεις (pixels) της εικόνας, διάστασης $img_h \times img_w$, που παράγεται σαν έξοδος από την κάμερα κατά την φωτογράφιση.

Σημείωση: Ο άξονας της κάμερας περνά από το κέντρο του ορθογωνίου διάστασης $cam_h \times cam_w$ ενώ η αρίθμηση του $cam_h \times cam_w$ πίνακα της εικόνας ξεκινά από τα κάτω προς τα πάνω και από αριστερά προς δεξιά, και έχει τιμές $[0, \dots, cam_w - 1]$ οριζοντίως και $[0, \dots, cam_h - 1]$ καθέτως.

Z. Συνάρτηση φωτογράφισης

Να υλοποιήσετε τη συνάρτηση:

$$\mathbf{I} = \text{render_object}(\text{verts}_{3d}, \text{faces}, \text{vcolors}, \text{img}_h, \text{img}_w, \text{cam}_h, \text{cam}_w, f, c_{org}, c_{lookat}, c_{up})$$

όπου:

- \mathbf{I} είναι η έγχρωμη εικόνα διάστασης $\text{img}_h \times \text{img}_w \times 3$. Η εικόνα θα περιέχει K χρωματισμένα τρίγωνα.
- verts_{3d} είναι οι τρισδιάστατες συντεταγμένες των σημείων του αντικειμένου διάστασης $L \times 3$.
- faces είναι ο πίνακας που περιέχει δείκτες σε σημεία του πίνακα verts_{3d} που συνθέτουν τις κορυφές των τριγώνων. Ο πίνακας είναι διάστασης $L \times 3$. Η i -οστή γραμμή του πίνακα, δηλώνει τις τρεις κορυφές που σχηματίζουν το τρίγωνο (με αναφορά σε κορυφές του πίνακα verts_{3d} και η αρίθμηση του ξεκινάει από το 0).
- vcolors είναι ο πίνακας με τα χρώματα των κορυφών. Ο πίνακας vcolors είναι διάστασης $L \times 3$. Η i -οστή γραμμή του πίνακα δηλώνει τις χρωματικές συνιστώσες της αντίστοιχης κορυφής.
- img_h και img_w είναι το ύψος και το πλάτος του καμβά αντίστοιχα.
- cam_h και cam_w είναι το ύψος και το πλάτος του πετάσματος της κάμερας (σε ίντσες).
- f είναι η απόσταση του πετάσματος από το κέντρο (μετρημένη στις μονάδες που χρησιμοποιεί το σύστημα συντεταγμένων της κάμερας).
- c_{lookat} είναι οι συντεταγμένες του σημείου στόχου (σε μη ομογενή μορφή).
- c_{up} είναι το μοναδιαίο up vector της κάμερας (σε μη ομογενή μορφή).
- Το c_{org} περιέχει τις συντεταγμένες του κέντρου της κάμερας ως προς το WCS.

η οποία χρησιμοποιεί κατάλληλα τις παραπάνω συναρτήσεις για να υλοποιήσει όλο το pipeline της απεικόνισης ενός αντικειμένου. Επίσης θα πρέπει να χρησιμοποιεί τη συνάρτηση `render` της προηγούμενης εργασίας για να χρωματίσει το αντικείμενο με τη μέθοδο `Gouraud shading`.

Παραδοτέα

1. Οι παραπάνω συναρτήσεις σε μορφή **σχολιασμένου** πηγαίου κώδικα python (= v3.7) με σχόλια γραμμένα στα **αγγλικά** ή **greeklish**, (κοινώς, **μη γράφετε σχόλια με ελληνικούς χαρακτήρες**).
2. script επίδειξης με όνομα `demo.py`. Το script θα πρέπει να καλείται χωρίς εξωτερικά ορίσματα, να διαβάζει το αντικείμενο από το αρχείο `hw2.npy` που σας δίνεται, και να εκτελεί **ένα προκαθορισμένο σύνολο** μετασχηματισμών, ο οποίος περιγράφεται παρακάτω:

Ως είσοδο χρησιμοποιείτε τον πίνακα verts_{3d} , οποίος περιέχει τις τρισδιάστατες συντεταγμένες των K κορυφών των τριγώνων που αποτελούν το αντικείμενο. Δοθέντων των σημείων του πίνακα verts_{3d} , το script σας θα πρέπει να **εκτελεί σειριακά** τα ακόλουθα βήματα:

(α') Τα μετατοπίζει κατά t_1 .

(β') Τα περιστρέφει κατά γωνία ϕ rad περί άξονα παράλληλο προς διάνυσμα \mathbf{u} .

(γ') Τα μετατοπίζει κατά t_2

Κάθε βήμα θα δέχεται ως είσοδο την έξοδο του προηγούμενου. Μετά από κάθε βήμα θα πρέπει να φωτογραφίζετε το αντικείμενο, καλώντας τη συνάρτηση `render_object` με παραμέτρους κάμερας `corg`, `clookat`, `cup` και να το χρωματίζετε καλώντας τη συνάρτηση `render` της πρώτης εργασίας με τη χρήση Gouraud shading.

Συνολικά, θα πρέπει να παράξετε 4 φωτογραφίες του αντικειμένου, μία στην αρχική του θέση, και μία για τα αποτελέσματα των βημάτων (α) - (β). Κάθε φωτογραφία να αποθηκεύεται με με όνομα αρχείου τον αριθμό του βήματος (θεωρώντας ότι η αρχική θέση είναι το βήμα 0) και επέκταση jpg.

Σημείωση 1: Αν δεν είχατε υλοποιήσει τη συνάρτηση `render`, ή αν είχατε κάποιο λάθος στην υλοποίησή της, μπορείτε να χρησιμοποιήσετε τη `render` κάποιου/κάποιας συναδέλφου σας, αρκεί να το δηλώσετε το όνομά του στην αναφορά σας.

Σημείωση 2: Το αρχείο `hw2.py` που σας δίνεται, περιέχει τις παραμέτρους του αντικειμένου (`verts_3d`, `vcolors`, `faces`) καθώς και όλες τις παραμέτρους που θεωρούνται γνωστές (παραμέτρους κάμερας, διανύσματα μετατόπισης, άξονες περιστροφής κτλ.).

3. Αναφορά με:

- Περιγραφή της λειτουργίας και του τρόπου κλήσης των προγραμμάτων.
- Περιγραφή των συναρτήσεων.
- Τα ενδεικτικά αποτελέσματα που παράγονται από το demo.

Παρατηρήσεις

- Μην κάνετε τεχνητές περιστροφές της φωτογραφίας προκειμένου να φαίνεται "ίσιο" το αντικείμενο.
- Οι εργασίες αξιολογούνται με χρήση Python(=v3.7).
- Οι εργασίες είναι **αυστηρά** ατομικές.
- Το background του καμβά είναι λευκό (`rgb = (1.0, 1.0, 1.0)`).
- Υποβάλετε ένα και μόνο αρχείο, τύπου .zip.
- Το όνομα του αρχείου πρέπει να είναι `AEM.zip`, όπου AEM είναι τα τέσσερα ψηφία του Α.Ε.Μ. του φοιτητή της ομάδας.
- Το προς υποβολή αρχείο πρέπει να περιέχει τα αρχεία κώδικα python και το αρχείο `report.pdf` το οποίο θα είναι η αναφορά της εργασίας.
- Η αναφορά πρέπει να είναι ένα αρχείο τύπου PDF, και να έχει όνομα `report.pdf`.
- Όλα τα αρχεία κώδικα πρέπει να είναι αρχεία κειμένου τύπου UTF-8, και να έχουν κατάληξη .py.

- Το αρχείο τύπου zip που θα υποβάλετε δεν πρέπει να περιέχει κανένα φάκελο.
- Για την ονομασία των αρχείων που περιέχονται στο προς υποβολή αρχείο, χρησιμοποιείτε μόνο αγγλικούς χαρακτήρες, και όχι ελληνικούς ή άλλα σύμβολα, πχ “#”, “\$”, “%” κλπ.

Προσοχή: Θα αξιολογηθούν μόνο όσες εργασίες έχουν demos που τρέχουν!