

Γραφική με Υπολογιστές 2020

Εργασία #2

Μετασχηματισμοί και προβολές

Στέφανος Παπαδάμ
ΑΕΜ: 8885

12/05/2020

1. Περιγραφή των συναρτήσεων.

1.1 Συνάρτηση rotationMatrix

Στη συγκεκριμένη συνάρτηση υπολογίζεται ο πίνακας μετασχηματισμού R χρησιμοποιώντας τον τύπο του Rodrigues (5.45). Ο πίνακας R έχει μέγεθος 3×3 και για τον σχηματισμό του απαιτούνται μόνο η γωνία περιστροφής θ και το μοναδιαίο διάνυσμα u που είναι παράλληλο προς τον άξονα περιστροφής και τα οποία δίνονται ως ορίσματα στην συνάρτηση. Ο πίνακας είναι ο παρακάτω:

$$\begin{bmatrix} (1 - \cos a)u_x^2 + (\cos a) & (1 - \cos a)u_xu_y - (\sin a)u_z & (1 - \cos a)u_xu_z + (\sin a)u_y \\ (1 - \cos a)u_yu_x + (\sin a)u_z & (1 - \cos a)u_y^2 + (\cos a) & (1 - \cos a)u_yu_z - (\sin a)u_x \\ (1 - \cos a)u_zu_x - (\sin a)u_y & (1 - \cos a)u_zu_y + (\sin a)u_x & (1 - \cos a)u_z^2 + (\cos a) \end{bmatrix}$$

1.2 Συνάρτηση affineTransform

Η συνάρτηση αυτή υλοποιεί σημειακό μετασχηματισμό affine σε έναν πίνακα με στήλες τις συντεταγμένες των σημείων p εφαρμόζοντας την περιστροφή των σημείων p κατά R και έπειτα την μετατόπιση κατά ct . Η συνάρτηση αυτή δέχεται ως ορίσματα τον πίνακα περιστροφής R , το διάνυσμα μετατόπισης ct και τον πίνακα cp με τις συντεταγμένες των σημείων p στα οποία θα εφαρμοσθεί ο μετασχηματισμός. Αρχικά η συνάρτηση μετατρέπει το διάνυσμα μετατόπισης ct σε διάνυσμα στήλη και ελέγχει τον πίνακα με τα τρισδιάστατα διανύσματα cp έτσι ώστε να έχει διάσταση $3 \times n$ δηλαδή οι συντεταγμένες των διανυσμάτων που πρόκειται να μετασχηματιστούν να βρίσκονται σε κάθε στήλη. Έπειτα, εφαρμόζεται ο συνολικός μετασχηματισμός affine χρησιμοποιώντας την πράξη:

- $cq = (R * cp) + ct.$

1.3 Συνάρτηση systemTransform

Η συνάρτηση αυτή υλοποιεί αλλαγή του συστήματος συντεταγμένων από ένα σύστημα με αρχή O και άξονες $(b1', b2', b3')$ σε άλλο σύστημα συντεταγμένων με αρχή $O + v0$ και άξονες

$(b1, b2, b3) = (L(b1'), L(b2'), L(b3'))$. Δέχεται ως ορίσματα τις συντεταγμένες των σημείων cp ως προς το αρχικό σύστημα συντεταγμένων που πρόκειται να αλλάξουν σύστημα συντεταγμένων, τις συντεταγμένες των τριών αξόνων $b1, b2, b3$ και τις συντεταγμένες $c0$ του διανύσματος $v0$ ως προς το αρχικό σύστημα αξόνων. Ο πίνακας μετασχηματισμού L που εφαρμόζεται ισούται με:

$L = [b1, b2, b3]$. Η συνάρτηση και σε αυτή την περίπτωση ελέγχει για τα σωστά μεγέθη των διανυσμάτων και των πινάκων και στη συνέχεια εφαρμόζει την αλλαγή συντεταγμένων. Ο συνολικός τύπος που υπολογίζει τις συντεταγμένες των σημείων στο νέο σύστημα συντεταγμένων είναι ο εξής:

- $dp = L^{-1} * (cp - co)$.

1.4 Συνάρτηση projectCamera

Η συνάρτηση αυτή υπολογίζει τις προοπτικές προβολές των τρισδιάστατων σημείων και υπολογίζει το βάθος του κάθε σημείου πριν την προβολή του. Δέχεται ως ορίσματα την απόσταση w του πετάσματος από το κέντρο, τις συντεταγμένες των αξόνων x_c και y_c της κάμερας και του διανύσματος vc που εκφράζει την απόσταση του κέντρου της κάμερας από το κέντρο O του WCS. Αρχικά, ελέγχονται τα σωστά μεγέθη των πινάκων και των διανυσμάτων και έπειτα υπολογίζεται το διάνυσμα zc με τις συντεταγμένες του άξονα z της κάμερας με χρήση του εξωτερικού γινομένου ανάμεσα στις άλλες δύο συντεταγμένες από τα μοναδιαία διανύσματα της κάμερας. Εφαρμόζεται αλλαγή του συστήματος συντεταγμένων από το WCS στο σύστημα συντεταγμένων της κάμερας CCS με χρήση της συνάρτησης `systemTransform` που υλοποιήθηκε προηγουμένως. Για τον υπολογισμό των διδιάστατων συντεταγμένων χρησιμοποιούνται οι τύποι:

- $xq = (w * xp) / zp$
- $yq = (w * yp) / zp$

Το βάθος κάθε σημείου ισούται με την τρίτη συντεταγμένη στον τρισδιάστατο χώρο οπότε και ισχύει:

- $d = zp$

1.5 Συνάρτηση projectCameraKu

Η συγκεκριμένη συνάρτηση υπολογίζει τις προοπτικές προβολές των τρισδιάστατων σημείων και υπολογίζει το βάθος του κάθε σημείου πριν την προβολή του, χρησιμοποιώντας διαφορετικά ορίσματα για τον καθορισμό της κατεύθυνσης της κάμερας. Δέχεται ως ορίσματα την απόσταση w του πετάσματος από το κέντρο, τις συντεταγμένες cK και cu του σημείου στόχου K και του μοναδιαίου up vector u αντίστοιχα και του διανύσματος vc που εκφράζει την απόσταση του κέντρου της κάμερας από το κέντρο O του WCS. Αρχικά, αφού ελεγχθούν τα σωστά μεγέθη των διανυσμάτων υπολογίζεται το διάνυσμα με τις συντεταγμένες του άξονα z από τον τύπο (6.6). Συγκεκριμένα, υπολογίζεται από το διάνυσμα CK , το οποίο προκύπτει από τη διαφορά του κέντρου της κάμερας και του σημείου K που στοχεύει. Έπειτα διαρείται με το μέτρο της διαφοράς αυτής για να κανονικοποιηθεί. Στη συνέχεια, χρησιμοποιείται ο τύπος (6.7) για τον υπολογισμό της συντεταγμένης y . Ειδικά, από το up vector u αφαιρείται η προβολή του u με το διάνυσμα zc και προκύπτουν οι συντεταγμένες του yc . Η συντεταγμένη x_c προκύπτει ως το εξωτερικό γινόμενο των yc και zc (6.8). Με βάση τις συντεταγμένες που υπολογίστηκαν καλείται η συνάρτηση `projectCamera` που υλοποιήθηκε προηγουμένως για να υπολογιστούν οι προβολές.

1.6 Συνάρτηση rasterize

Η συνάρτηση αυτή απεικονίζει τις συντεταγμένες των σημείων από το σύστημα μίας κάμερας με πέτασμα διάστασης $H \times W$ σε ίντσες, σε ακέραιες θέσεις (pixels) της εικόνας διάστασης $M \times N$ που παράγεται σαν έξοδος από την κάμερα. Δέχεται ως ορίσματα τον πίνακα με τα σημεία P , και τις διαστάσεις H, W, M, N όπως περιγράφηκαν παραπάνω. Αρχικά, κανονικοποιεί τις συντεταγμένες x και y των σημείων έτσι ώστε να ανήκουν στο διάστημα $[0,1]$ και έπειτα τις μετατρέπει σε ακέραιες τιμές πολλαπλασιάζοντας τις με τα M και N αντίστοιχα και έπειτα κάνοντας χρήση της συνάρτησης `floor()` για να τις στρογγυλοποιήσει προς τα κάτω. Η συγκεκριμένη υλοποίηση προήλθε από τη σελίδα [εδώ](#).

1.7 Συνάρτηση photographObject

Η συνάρτηση αυτή υλοποιεί το pipeline της φωτογράφισης και συγκεκριμένα καλεί αρχικά την συνάρτηση `projectCameraKu` για να υπολογιστούν η διδιάστατη προβολή και το βάθος του κάθε σημείου της εικόνας. Οι υπολογισμένες προβολές σε ίντσες δίνονται ως είσοδος στη συνάρτηση `rasterize` για να υπολογίσει τις ακέραιες συντεταγμένες του κάθε σημείου στην εικόνα. Η συνάρτηση `photographObject` επιστρέφει το βάθος και τις συντεταγμένες σε pixels των σημείων της εικόνας.

2. Περιγραφή της λειτουργίας και του τρόπου κλήσης των προγραμμάτων

2.1 Script demo

Αρχικά, στο script `demo.m` φορτώνεται το workspace `hw2.mat`. Έπειτα, καλείται η συνάρτηση `photographObject` η οποία επιστρέφει τις ακέραιες συντεταγμένες και το βάθος των σημείων της εικόνας τα οποία δίνονται ως είσοδος στην `paintObject`. Η συνάρτηση αυτή υλοποιήθηκε στην προηγούμενη εργασία και χρησιμοποιείται για να απεικονίσει το αντικείμενο. Αφού γίνει η απεικόνιση αποθηκεύεται η αρχική μορφή της πάπιας στο αρχείο `0.jpg`. Στη συνέχεια εκτελούνται με τη σειρά τα βήματα 1-3 που περιγράφονται παρακάτω.

2.2 Βήμα 1 – Μετατόπιση κατά t_1

Για την μετατόπιση των σημείων κατά το διάνυσμα t_1 εφαρμόζεται ένας affine μετασχηματισμός σε κάθε στήλη του πίνακα cp , με χρήση της συνάρτησης `affineTransform` στην οποία δίνεται ως είσοδος το διάνυσμα μετατόπισης ct και ο μοναδιαίος πίνακας R (3×3). Έπειτα, γίνεται η διαδικασία της φωτογράφισης όπως περιγράφηκε παραπάνω και αποθηκεύεται το αποτέλεσμα στο αρχείο `1.jpg`.

2.3 Βήμα 2 – Περιστροφή

Για την περιστροφή των σημείων κατά γωνία ϕ περί άξονα που διέρχεται από το O και είναι παράλληλο στο διάνυσμα g , δημιουργείται αρχικά ο πίνακας R χρησιμοποιώντας τη συνάρτηση `rotationMatrix` και στη συνέχεια εφαρμόζεται ένας affine μετασχηματισμός σε κάθε στήλη του πίνακα cp , με χρήση της συνάρτησης `affineTransform` στην οποία δίνεται ως είσοδος το διάνυσμα μετατόπισης $[0,0,0]$ και ο υπολογισμένος πίνακας R . Γίνεται η απεικόνιση του μετασχηματισμένου αντικειμένου και αποθηκεύεται το αποτέλεσμα στο αρχείο `2.jpg`.

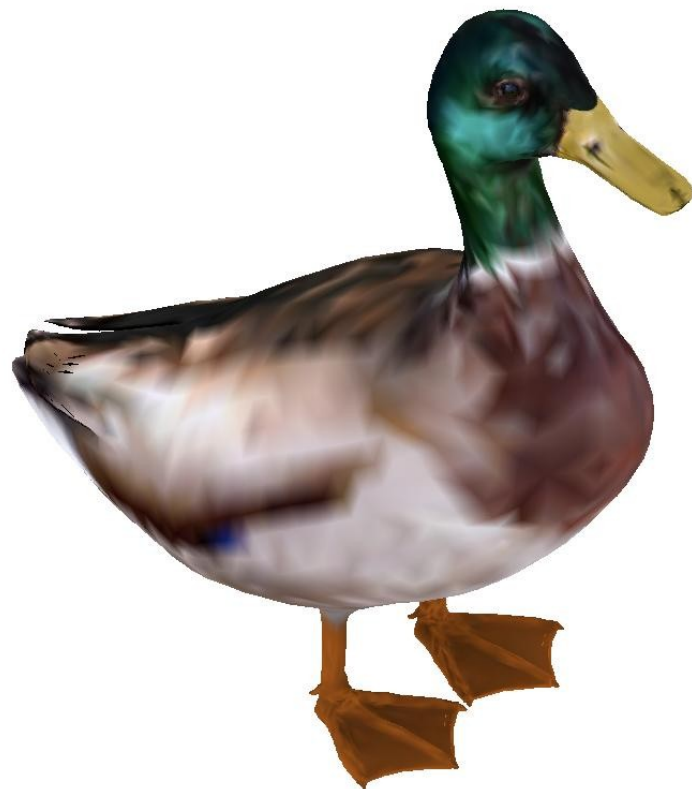
2.4 Βήμα 3 – Μετατόπιση κατά t_2

Εφαρμόζεται η ίδια διαδικασία με το βήμα 1 και αποθηκεύεται το αποτέλεσμα στο αρχείο 3.jpg.

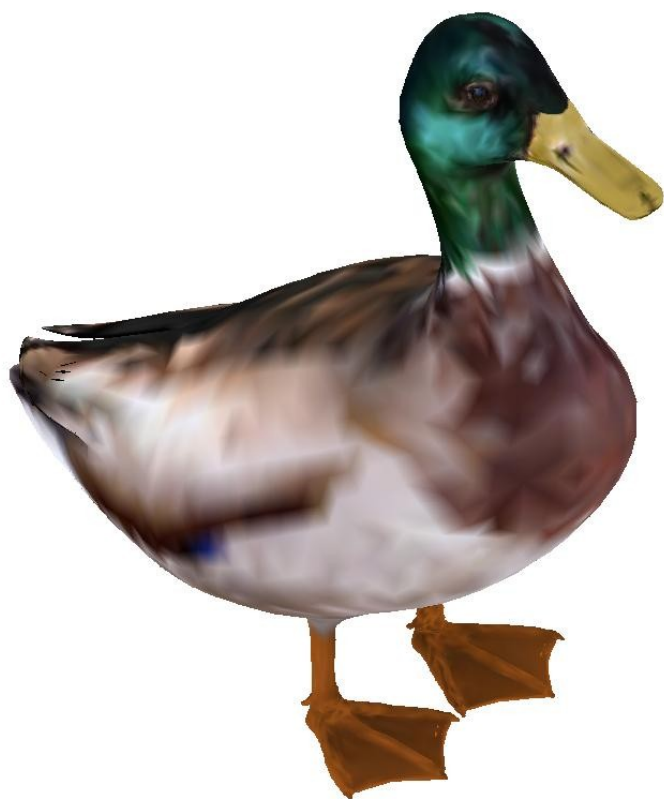
3. Ενδεικτικά αποτελέσματα & σχόλια

Ο χρόνος εκτέλεσης όλης της διαδικασίας είναι περίπου στα 35 δευτερόλεπτα. Για τον χρωματισμό των αντικειμένων χρησιμοποιήθηκαν οι συναρτήσεις χρωματισμού τριγώνων από την πρώτη εργασία. Παρακάτω φαίνονται τα αποτελέσματα από τα βήματα 1 έως 3.

3.1 Βήμα 0 – Αρχική μορφή



3.2 Βήμα 1 – Μετατόπιση κατά t_1



3.3 Βήμα 2 – Περιστροφή



3.4 Βήμα 3 – Μετατόπιση κατά t_2

