Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ

Γραφική με Υπολογιστές

Άνοιξη 2021 - 8ο Εξάμηνο

-Εργασία 2-

Από 3Δ συντεταγμένες σε προβολή στην οθόνη Η/Υ

Ονοματεπώνυμο: Τσούσης Παναγιώτης

ΑΕΜ: 9590

Email: pitsousis@ece.auth.gr

Περιεχόμενα

Εισαγωγή

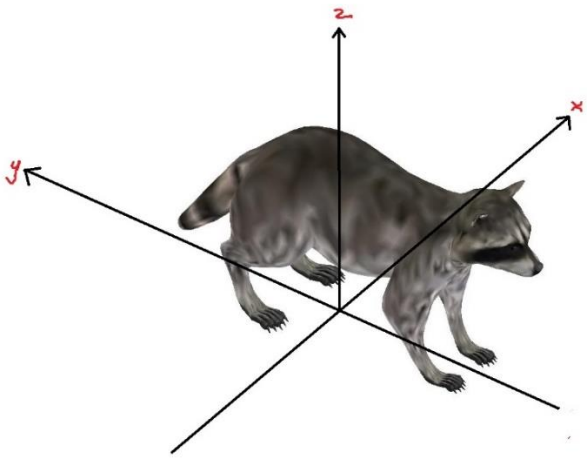
Η εργασία αυτή έχει ως θέμα τη μετατροπή συντεταγμένων από τον 3Δ κόσμο σε pixels στην οθόνη. Η διαδικασία αυτή περιλαμβάνει περιστροφή και μετατόπιση των προβαλλόμενων σημείων καθώς και περιστροφή και μετατόπιση της (θέσης της) εικονικής κάμερας. Αποτελεί επέκταση της πρώτης εργασίας, κομμάτι της οποίας χρησιμοποιούμε για να απεικονίσουμε το αντικείμενο που προκύπτει από τα τελικά pixels που παράγονται από τη δεύτερη εργασία. Για την υλοποίησή της, δημιουργήθηκε ένα πρόγραμμα σε Matlab έκδοσης R2018a. Όλα τα αρχεία που απαιτούνται για να τρέξει το πρόγραμμα επίδειξης demo.m, συμπεριλαμβανομένων των αρχείων από την πρώτη εργασία, έχουν αποσταλεί στο αρχείο με τον πηγαίο κώδικα. Ο κώδικας είναι σχολιασμένος.

Παρατηρήσεις

Πριν εξηγηθεί ο πηγαίος κώδικας και δοθούν τα αποτελέσματα του demo όπως ζητήθηκε, θα ήταν χρήσιμο να γίνουν μερικές παρατηρήσεις σχετικά με την εργασία.

Αρχικό σύστημα συντεταγμένων

Θεωρήθηκε χρήσιμο, αφού είχε υλοποιηθεί ο κώδικας, να βρεθεί προσεγγιστικά το αρχικό σύστημα συντεταγμένων, σύμφωνα με το οποίο περιγράφονται τα σημεία αρχικά. Αυτό έγινε, για να μπορεί να αξιολογηθεί η ορθή λειτουργία του κώδικα ευκολότερα. Η μέθοδος με την οποία έγινε αυτό είναι μέσω περιστροφής του αντικειμένου γύρω από τους άξονες x,y και z. Έτσι, όχι μόνο γνωρίζουμε τη θέση των αξόνων, αλλά και τη φορά τους, καθώς η περιστροφή με τον πίνακα Rodrigues είναι δεξιόστροφη, αν κοιτάξουμε το αντικείμενο από το 'πίσω' μέρος του άξονα κατά μπροστά. Διαπιστώθηκε ότι η θέση του αρχικού συστήματος, βλέποντάς από την εικονική κάμερα που δόθηκε στην εργασία, είναι προσεγγιστικά όπως φαίνεται στην εικόνα 1.



Εικόνα 1: Αρχικό σύστημα συντεταγμένων

Περιττή συνάρτηση system_transform(cp,T,c0)

Κατά την εκπόνηση της εργασίας, έγινε αντιληπτό ότι μία από τις ζητούμενες συναρτήσεις, η system_transform(cp,T,c0), είναι περιττή. Η συνάρτηση αυτή θα πρέπει να υλοποιεί *μόνο* περιστροφή του συστήματος συντεταγμένων σύμφωνα με το οποίο περιγράφονται τα σημεία και επιστρέφει τις νέες συντεταγμένες των σημείων. Ως ορίσματα παίρνει τα σημεία cp που περιγράφονται σύμφωνα με το παλιό σύστημα συντεταγμένων, τον πίνακα μετασχηματισμού T που θα υλοποιεί το μετασχηματισμό και το διάνυσμα c0 γύρω από τον οποίο περιστρέφεται το σύστημα συντεταγμένων. Επιστρέφει τις συντεταγμένες των σημείων cp στο καινούριο σύστημα συντεταγμένων.

Όμως, σύμφωνα με την παραπάνω περιγραφή, η συνάρτηση έχει μερικά προβλήματα. Αρχικά, δεν έχει ως όρισμα τη γωνία κατά την οποία περιστρέφουμε το σύστημα. Άρα δε γίνεται να λειτουργήσει σωστά η συνάρτηση. Όμως, μπορούμε να ενσωματώσουμε τη γωνία στον πίνακα T σύμφωνα με τον τύπο του Rodrigues. Αν, όμως, εφαρμόσουμε αυτό, τότε το όρισμα c0 είναι περιττό, αφού στον πίνακα T ενσωματώνεται και ο άξονας περιστροφής. Ας υποθέσουμε ότι βγάζουμε, λοιπόν, το c0 και βάζουμε τον αρχικοποιημένο πίνακα μετασχηματισμού T. Το μόνο που μένει να κάνουμε, πλέον, είναι να εφαρμόσουμε ένα μετασχηματισμό affine για κάθε σημείο. Αλλά, όπως θα δούμε παρακάτω, υπάρχει ήδη η συνάρτηση affine_transform που το κάνει αυτό. Συνεπώς, η συνάρτηση system_transform είναι περιττή. Επίσης, στα πλαίσια της εργασίας, δε χρειαζόμαστε μόνο περιστροφή του συστήματος αξόνων, αλλά και μετατόπιση. Αυτό, βέβαια, ενσωματώνεται στον πίνακα T πλέον.

Επειδή έχει ζητηθεί από την εκφώνηση, έχει υλοποιηθεί, αλλά δεν παρουσιάζει καμία διαφορά από τη συνάρτηση affine_transform.

Περιγραφή Κώδικα

Ακολουθεί η περιγραφή της κλάσης και των συναρτήσεων που δημιουργήθηκαν.

Κλάση transformation_matrix

Η κλάση transformation_matrix έχει ως property ένα πίνακα T μεγέθους 4x4. Αρχικοποιείται ως ένας μοναδιαίος πίνακας.

Τα functions της κλάσης είναι το rotate(angle, u) και το translate(t). Τα functions αυτά τροποποιούν τον πίνακα T, ώστε να είναι ένας ομογενής πίνακας μετασχηματισμού που υλοποιεί ένα σημειακό μετασχηματισμό affine.

Το function rotate(angle, u) τροποποιεί τον πίνακα T, 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, εκμεταλλεύεται τον τύπο του Rodrigues.

Το function translate(t) τροποποιεί τον πίνακα T, ώστε να υλοποιεί μετατόπιση κατά διάνυσμα t.

Συνάρτηση cq = affine_transform(cp, transform)

Η συνάρτηση αυτή υλοποιεί σημειακό μετασχηματισμό affine. Ως ορίσματα, έχει τα σημεία εισόδου cp πάνω στα οποία εφαρμόζεται ο μετασχηματισμός και ένα αντικείμενο transformation_matrix transform το οποίο περιέχει ένα αρχικοποιημένο πίνακα μετασχηματισμού. Η έξοδος cq είναι τα μετασχηματισμένα σημεία.

Στο εσωτερικό της συνάρτησης γίνεται απλά ο πολλαπλασιασμός κάθε σημείου με τον πίνακα T ώστε να παραχθεί το αποτέλεσμα. Προφανώς, τα σημεία περιγράφονται σε ομογενείς συντεταγμένες.

Συνάρτηση dq = system_transform(cp, transform)

Σημείωση: Η συνάρτηση αυτή εκτελεί την ίδια διαδικασία με την affine_transform, αλλά σαν όρισμα συνήθως έχει πιο συγκεκριμένο πίνακα μετασχηματισμού.

Η συνάρτηση αυτή υλοποιεί σημειακό μετασχηματισμό affine. Ως ορίσματα, έχει τα σημεία εισόδου cp πάνω στα οποία εφαρμόζεται ο μετασχηματισμός και ένα αντικείμενο transformation_matrix transform το οποίο περιέχει ένα αρχικοποιημένο πίνακα μετασχηματισμού. Η έξοδος cq είναι τα μετασχηματισμένα σημεία.

Στο εσωτερικό της συνάρτησης γίνεται απλά ο πολλαπλασιασμός κάθε σημείου με τον πίνακα T ώστε να παραχθεί το αποτέλεσμα. Προφανώς, τα σημεία περιγράφονται σε ομογενείς συντεταγμένες.

Συνάρτηση [P,D] = project_cam(w,cv,cx,cy,cz,p)

Η συνάρτηση αυτή δέχεται ως είσοδο κάποια σημεία στον 3Δ χώρο τα οποία περιγράφονται σε ένα σύστημα συντεταγμένων και ένα νέο σύστημα συντεταγμένων που ορίζει μια κάμερα και παράγει στην έξοδο τις 2Δ συντεταγμένες της προβολής των σημείων πάνω στο πέτασμα της κάμερας και το βάθος τους.

Έχει ως ορίσματα:

- w : Η απόσταση του εικονικού πετάσματος από το φακό
- cv : Το κέντρο του νέου συστήματος συντεταγμένων
- cx : Ο άξονας x της εικονικής κάμερας
- cy : Ο άξονας y της εικονικής κάμερας
- cz : Ο άξονας z της εικονικής κάμερας
- p : Τα 3Δ σημεία εισόδου

και παράγει ως έξοδο:

- P: ο πίνακας με τις 2Δ συντεταγμένες των σημείων πάνω στο πέτασμα

- D: Το βάθος των σημείων

Για να λειτουργήσει, η συνάρτηση αυτή εκμεταλλεύεται τους μαθηματικούς τύπους της προοπτικής προβολής και για της αλλαγής συστήματος συντεταγμένων.

Συνάρτηση [P,D] = project_cam_ku(w,c_center,c_look,c_up,p)

Η συνάρτηση αυτή δέχεται ως είσοδο κάποια σημεία στον 3Δ χώρο τα οποία περιγράφονται σε ένα σύστημα συντεταγμένων και ένα νέο σύστημα συντεταγμένων που ορίζει μια κάμερα και παράγει στην έξοδο τις 2Δ συντεταγμένες της προβολής των σημείων πάνω στο πέτασμα της κάμερας και το βάθος τους. Σε σύγκριση με την project_cam, απλά διαφέρει στα ορίσματα, τα οποία στην project_cam_ku είναι πιο φυσικά για έναν άνθρωπο.

Έχει ως ορίσματα:

- w : Η απόσταση του εικονικού πετάσματος από το φακό
- c_center : Το κέντρο του νέου συστήματος συντεταγμένων
- c_look : Το σημείο στο οποίο ‘κοιτάει’ η κάμερα
- c_up: Ο διάνυσμα που δείχνει την ‘κατά πάνω’ διεύθυνση της κάμερας
- p : Τα 3Δ σημεία εισόδου

και παράγει ως έξοδο:

- P: ο πίνακας με τις 2Δ συντεταγμένες των σημείων πάνω στο πέτασμα
- D: Το βάθος των σημείων

Για να λειτουργήσει, η συνάρτηση αυτή εκμεταλλεύεται τους μαθηματικούς τύπους για τη μετατροπή των c_look και c_up για να υπολογίσει τους άξονες του νέου συστήματος συντεταγμένων. Στη συνέχεια, απλά καλεί την project_cam.

Συνάρτηση P_rast = rasterize(P,M,N,H,W)

Η συνάρτηση αυτή είναι υπεύθυνη για τη ‘μετάφραση’ των σημείων ενός πετάσματος κάμερας σε συντεταγμένες pixel μιας εικόνας. Ως ορίσματα, έχει:

- P: τα σημεία του πετάσματος προς μετάφραση
- M: το πλάτος της εικόνας (σε pixels)
- N: το ύψος της εικόνας (σε pixels)
- H: το ύψος του πετάσματος (σε κλίμακα της κάμερας)
- W: το πλάτος του πετάσματος (σε κλίμακα της κάμερας)

και παράγει ως έξοδο:

- P_rast: τις ακέραιες 2Δ συντεταγμένες του pixel της εικόνας.

Για να υλοποιηθεί η παραπάνω διαδικασία, αρκούν μερικές απλές πράξεις, όπως το πόσα pixel αντιστοιχούν σε μία μονάδα μήκους της κάμερας καθώς και στρογγυλοποίηση στον κοντινότερο ακέραιο για να έχουμε ακέραιες τιμές.

Συνάρτηση img = render_object(p, F,C,M,N,H,W,w, cv, clookat, cup)

Η συνάρτηση αυτή είναι υπεύθυνη για τη δημιουργία μιας εικόνας από 3Δ συντεταγμένες κορυφών τριγώνων ενός αντικειμένου. Ως ορίσματα, έχει:

- p : οι 3Δ συντεταγμένες όλων των σημείων που αποτελούν κορυφές τριγώνων του αντικειμένου προς απεικόνιση.
- F: πίνακας που περιέχει τις κορυφές κάθε τριγώνου σε κάθε γραμμή του.
- M: το πλάτος της εικόνας (σε pixels)
- N: το ύψος της εικόνας (σε pixels)
- H: το ύψος του πετάσματος (σε κλίμακα της κάμερας)
- W: το πλάτος του πετάσματος (σε κλίμακα της κάμερας)
- w : Η απόσταση του εικονικού πετάσματος από το φακό

- cv : Το κέντρο του νέου συστήματος συντεταγμένων
- clookat : Το σημείο στο οποίο ‘κοιτάει’ η κάμερα
- cup: Ο διάνυσμα που δείχνει την ‘κατά πάνω’ διεύθυνση της κάμερας

και παράγει ως έξοδο:

- img: πίνακας MxNx3 που αναπαριστά μια έγχρωμη εικόνα.

Η συνάρτηση αυτή αρχικά καλεί την project_cam_ku ώστε να υπολογίσει τις συντεταγμένες των σημείων πάνω στο πέτασμα της εικονικής κάμερας. Στη συνέχεια, καλεί τη συνάρτηση rasterize ώστε να μετατρέψει τα σημεία αυτά σε pixel της εικόνας. Ίσως χρειαστεί να επεκτείνει την εικόνα, με σκοπό να σχηματίσει τα αντικείμενα της εικόνας σωστά. Καλεί τη συνάρτηση render της εργασίας 1 για να σχηματίσει τα αντικείμενα και τέλος βγάζει στην έξοδο την εικόνα. Αν η εικόνα που έγινε render είχε επεκταθεί τότε κάνει και την ανάλογη περικοπή.

Πρόγραμμα demo.m

Το πρόγραμμα demo.m 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 **‘0.jpg’**. Στη συνέχεια, δημιουργεί 3 μετασχηματισμούς **trans1, trans2, trans3** (κλάσεις **transformation_matrix**) που θα χρειαστούμε στη συνέχεια. Μετά, εφαρμόζει μια **μετατόπιση t1** μέσω του **trans1** με τις συναρτήσεις **trans1.translate** και **affine_transform** στο αντικείμενο φωτογράφισης και το φωτογραφίζει ξανά με την συνάρτηση **render_object** και την αποθηκεύει ως **‘1.jpg’**. Με όμοια διαδικασία, περιστρέφει και μετατοπίζει το αντικείμενο φωτογραφίζοντάς το σε κάθε βήμα, δημιουργώντας άλλες 2 εικόνες, τις **‘2.jpg’** και **‘3.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. Επίσης, η δόμηση του demo έγινε με βάση το demo_sample που δόθηκε.

Αποτελέσματα Demo

Παρακάτω δίνονται οι εικόνες που παράγονται από το πρόγραμμα demo.

Αρχική θέση – Εικόνα '0.jpg'

Εικόνα 2: Αρχική θέση του αντικειμένου

Μετατόπιση κατά t1 – Εικόνα ‘1.jpg’

Εικόνα 3: Μετατόπιση του αντικειμένου κατά t1

Περιστροφή κατά theta γύρω από g – Εικόνα ‘2.jpg’

Εικόνα 4: Περιστροφή του αντικειμένου κατά γωνία theta γύρω από άξονα g

Μετατόπιση κατά t2 – Εικόνα ‘3.jpg’

Εικόνα 5: Μετατόπιση του αντικειμένου κατά t2

Με βάση το αρχικό σύστημα συντεταγμένων που δόθηκε στις παρατηρήσεις στην αρχή, καθώς και τις τιμές των μετασχηματισμών, το πρόγραμμα φαίνεται να λειτουργεί όπως θα έπρεπε για κάθε μετασχηματισμό.