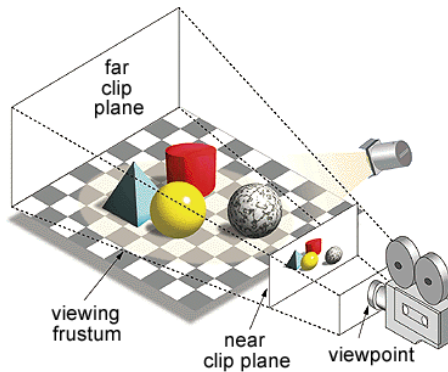




ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ



8^ο Εξάμηνο : «Γραφική με Υπολογιστές»

Ακαδ. έτος 2021-22 | Ημ.Παράδοσης 11/7/2022

3^η Εργασία : «Θέαση»

του Σταύρου Βασίλειου Μπουλιόπουλου 9671

Διδάσκων θεωρίας : Αναστάσιος Ντελόπουλος

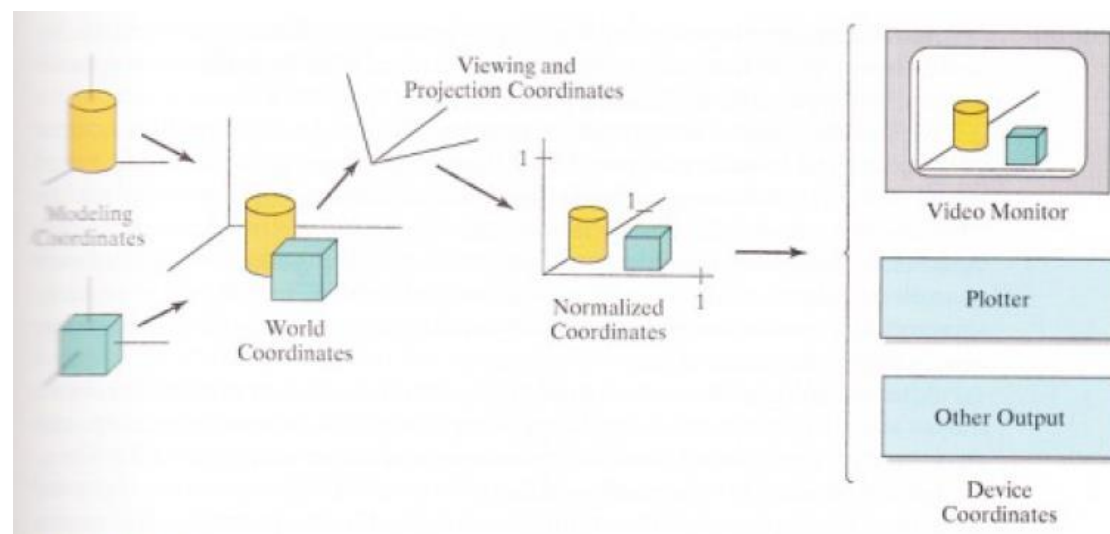
Επιβλέπων εργασίας : Αντώνιος Καρακώπτας

Περιεχόμενα

1.Εισαγωγή και ζητούμενα.....	2
2. Εκτέλεση των αρχείων κώδικα	2
3. Επεξήγηση των προβλημάτων που αντιμετωπίσα,παραδοχές και η δομή του κώδικα.....	3
4. Αποτελέσματα της θέασης και του φωτισμού	6

1.Εισαγωγή και ζητούμενα

Στην προηγούμενη εργασία δημιουργήσαμε το στάδιο της προβολής τρισδιάστατων σκηνικών στο πέτασμα της κάμερας και ύστερα στον καμβά του υπολογιστή. Σε αυτή την εργασία σκοπός είναι να συμπεριλάβουμε αλγορίθμους που σχετίζονται με την θέαση του αντικειμένου ως προς μια πηγή φωτός για την προβολή του αντικειμένου σε ένα πλήρες εικονικό σκηνικό.



Εικόνα 1: 3D modeling coordinates to 2D image coordinates

Ζητήθηκαν συναρτήσεις με τις οποίες υπολογίζουμε τα 3 βασικά μοντέλα φωτισμού και κάνουμε σκίαση τριγώνων βάσει Gouraud(Vertex Shader) και Phong(Fragment Shader).

2. Εκτέλεση των αρχείων κώδικα

Η απεικόνιση των αποτελεσμάτων(4 πειράματα για κάθε shader) και η αποθήκευση αυτών γίνεται χρησιμοποιώντας για γλώσσα προγραμματισμού Python(version=3.7) στο αρχείο εκτέλεσης **demo.py**, το οποίο περιέχει τις απαιτούμενες

βιβλιοθήκες και τις ζητούμενες συναρτήσεις(line6 έως line343).

3. Επεξήγηση των προβλημάτων που αντιμετώπισα, παραδοχές και η δομή του κώδικα

Τα προβλήματα που ήρθα αντιμέτωπος ήταν η προσεκτική ανάλυση βήμα-βήμα των πράξεων μεταξύ στοιχείων πινάκων που κάνω ,ώστε να αποδώσω την σωστή πολυδιάστατη έξοδο που απαιτεί η κάθε συνάρτηση.

Οι **παραδοχές** που έκανα ήταν οι εξής:

- μικροαλλαγές συνάρτησης **render_object()** προηγούμενης εργασίας για να καλύπτει τα ερωτήματα του B2 ως προς τις εισόδους και τις εσωτερικές ελλοχευμένες κλήσεις των shaders και επίσης πρόσθεσα ως όρισμα **lighter** τον τύπο μοντέλου φωτισμού που θέλουμε να συμπεριλάβουμε κατά τον υπολογισμό της έντασης του χρώματος

- πρόσθεσα στα ζητούμενα ερωτήματα τις βοηθητικές συναρτήσεις:

- **color_lighter()** για τον υπολογισμό του φωτισμού ανάλογα του ορίσματος εισόδου **lighter mode**

- **interpolate_normal()** για τον υπολογισμό του κανονικού διανύσματος ενός σημείου με κυρτή γραμμική παρεμβολή μεταξύ δύο σημείων

- uncomment line65(αν επιθυμήσουμε τεχνητή περιστροφή για να φαίνεται ίσιο το αντικείμενο με τον σκύλο)

Η **ροή του κώδικα** στο αρχείο είναι η εξής:

- import packages

- ορισμός υπολογιστικών συναρτήσεων με αναλυτικά σχόλια

- άντληση και επεξεργασία δεδομένων από **h3.npy**

- κάλεσμα της **render_object()** για την προβολή του αντικειμένου και την δημιουργία 2D εικόνας ,όπου πρώτο όρισμα εισόδου είναι ο τύπος του shader ,ενώ τελευταίο ο τύπος φωτισμού που θέλουμε να ληφθεί υπόψιν
- απεικόνιση και αποθήκευση εικόνας

Η λειτουργία των συναρτήσεων είναι η εξής :

Θεωρώ k_a, k_d, k_s συντελεστές ανάκλασης και n βαθμός διασποράς Phong των ανακλώμενων ακτίνων.

-η **ambient_light()** υπολογίζει το διάχυτο φως από το περιβάλλον επιστρέφοντας την τριχρωματική ένταση ακτινοβολίας με τον τύπο: $I_{ambient} = I_a k_a$.

-η **diffuse_light()** δέχεται στην είσοδο την θέση της φωτεινής πηγής, το σημείο ενδιαφέροντος P ,το χρώμα αυτού c και το κανονικό διάνυσμα N της επιφάνειας του P και υπολογίζει την διάχυτη ανάκλαση με τον τύπο: $I_{diffuse} = c * I_p k_d (N \cdot L)$,όπου L το κανονικό διάνυσμα με φορά από το σημείο P προς την πηγή φωτός και θεωρώντας ότι $f_{att} = 1$,δηλαδή δεν υπάρχει εξασθένηση έντασης λόγω απόστασης.

-η **specular_light()** υπολογίζει την κατοπτρική ανάκλαση γενικευμένου μοντέλου μη ιδανικού ανακλαστήρα με τον τύπο: $I_{specular} = c * I_p k_s \cos(b - a)^n$ με $\cos(b - a) = R \cdot V$. V είναι το κανονικό διάνυσμα με φορά από το σημείο P προς την θέση της κάμερας-θεατή και R είναι το κανονικό διάνυσμα που περιγράφει την ευθεία της ιδανικής ανάκλασης που υπολογίζεται από τον τύπο:

$$R = 2N(N \cdot L) - L$$

Ο τελικός υπολογισμός της αποδιδόμενης έντασης χρώματος υπολογίζεται από τον τύπο $I = I_{ambient} + I_{diffuse} + I_{specular}$

και διαχειρίζεται βάσει του ορίσματος `lighter` στην `color_lighter()` .

-η `calculate_normals()` υπολογίζει τα κανονικά διανύσματα των κορυφών όλων των τριγώνων, τα οποία υπολογίζονται σύμφωνα με τον κανόνα του δεξιόστροφου κοχλία από τις κορυφές του κάθε τριγώνου. Συγκεκριμένα, οι ονομαζόμενες κορυφές A, B, C δημιουργούν τα πλευρικά κανονικοποιημένα διανύσματα AB, BC και από το εξωτερικό γινόμενο αυτών $AB \times BC$ υπολογίζεται το κανονικό διάνυσμα του κάθε τριγώνου. Κάθε τέτοιο διάνυσμα αθροίζεται στα άλλα του ίδιου τριγώνου $N(TriangleIndex) = \sum_k^{face_indices} AB \times BC_k$, ώστε μετά από την άθροιση να βρούμε το επιστρεφόμενο ζητούμενο κανονικοποιώντας το διάνυσμα $N(TriangleIndex)$ για κάθε τρίγωνο.

-η `render_object()` διαφέρει ελάχιστα από τις προηγούμενες εργασίες και δημιουργεί την εικόνα που επιστρέφεται. Αρχικά, υπολογίζει τα κανονικά διανύσματα της επιφάνειας με την προηγούμενη συνάρτηση και στην συνέχεια καλεί τις `project_cam_ku()` και `rasterize()` για την προβολή των σημείων στο πέτασμα της κάμερας και την μετατροπή αυτών σε θέσεις pixel στον καμβά του υπολογιστή. Τέλος, υπολογίζει το νέο βάθος κάθε τριγώνου και καλεί βάσει της σειράς βάθους τις συναρτήσεις πλήρωσης τριγώνων ανάλογα το *shader mode* που επιλέχθηκε από το όρισμα εισόδου.

-η `shade_gouraud()` διαφέρει ελάχιστα από τις προηγούμενες εργασίες και πληρώνει το τρίγωνο. Πλέον λαμβάνεται υπόψιν ο υπολογισμός του χρώματος βάσει των μοντέλων φωτισμού. Συγκεκριμένα, για κάθε κορυφή του τριγώνου υπολογίζεται η ένταση χρώματος βάσει των προαναφερθέντων μοντέλων φωτισμού. Τέλος, ακολουθείται ο γνωστός αλγόριθμος

χρωματισμού του Gouraud βάσει των νέων τριών χρωμάτων των κορυφών.

-η **shade_phong()** κρατάει βασικά στοιχεία της πλήρωσης τριγώνων με Gouraud αλλά στην προκειμένη περίπτωση ο αλγόριθμος του Phong αντί για μόνο παρεμβολή μεταξύ των σημείων για τον υπολογισμό του χρώματος, πρέπει να γίνει παρεμβολή για τον υπολογισμό του φωτισμού. Συγκεκριμένα, για κάθε ενδιαμέσο σημείο γίνεται γραμμική παρεμβολή για να βρεθεί το κανονικό του διάνυσμα με την βοήθεια της **interpolate_normal()** και σε επόμενο στάδιο υπολογίζουμε την ένταση φωτεινότητας χρώματος βάσει των προαναφερθέντων μοντέλων φωτισμού.

4. Αποτελέσματα της Θέασης και του φωτισμού

Τα αποτελέσματα κρίνονται ικανοποιητικά ως προς την λογική αν και είναι σκοτεινά, γιατί δεν έκανα κάποια τεχνητή περιστροφή ή αλλαγή στον προσανατολισμό του σκύλου. Επίσης, δόθηκαν τα περισσότερα δεδομένα στο **npv** αρχείο, αλλά κράτησα από την προηγούμενη εργασία για την κάμερα **focal length = 70**.

Για κάθε πείραμα επιλέγεται ο τύπος φωτισμού που επιθυμούμε βάσει του ορίσματος:

lighter={combined, ambient, diffusion, specular}

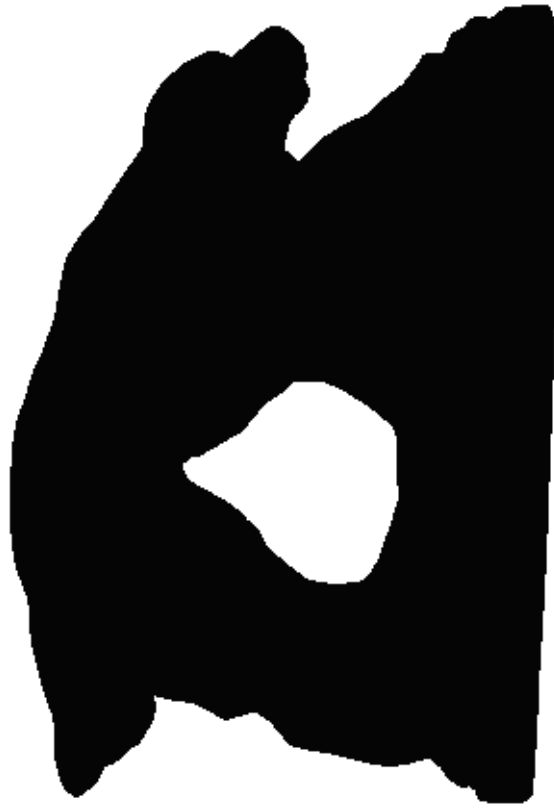
Θέαση και φωτισμός του αντικειμένου

Μέσος χρόνος για κάθε εικόνα-πείραμα:

- Gouraud shading~5με10sec
- Phong shading~15με26sec



Εικόνα 2: Shader: Gouraud, Lighter: Combined g1.jpg



Εικόνα 3: Shader: Gouraud, Lighter: Ambient g2.jpg



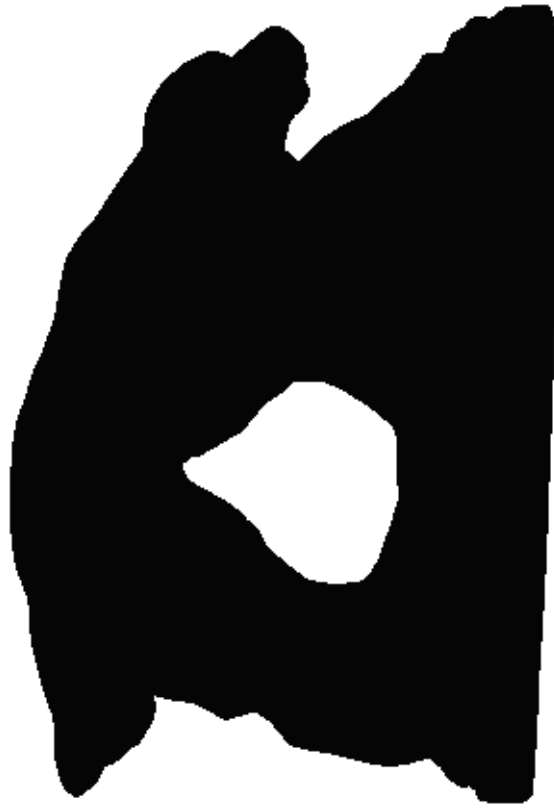
Εικόνα 4: Shader: Gouraud, Lighter: Diffusion g3.jpg



Εικόνα 5: Shader: Gouraud, Lighter: Specular g4.jpg



Εικόνα 6: Shader: Phong, Lighter: Combined ph1.jpg



Εικόνα 7: Shader: Phong, Lighter: Ambient ph2.jpg



Εικόνα 8: Shader: Phong, Lighter: Diffusion ph3.jpg



Εικόνα 9: Shader: Phong, Lighter: Specular ph4.jpg