

Γραφική με Υπολογιστές

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης -
Τμήμα Ηλεκτρολόγων Μηχανικών Μηχανικών
Υπολογιστών

3η Εργασία Εξαμήνου: Θέαση

Αντωνίου Αντώνιος - aantonii@ece.auth.gr - 9482

Σκοπός της εργασίας

Στο τρίτο και τελευταίο μέρος των εργασιών του μαθήματος, αναλαμβάνουμε τη σκίαση και τη θέαση ενός τρισδιάστατου αντικειμένου χρησιμοποιώντας δύο διαφορετικούς αλγορίθμους, με τρία διαφορετικά είδη φωτισμού.

Τα αρχικά δεδομένα που δίνονται είναι οι συντελεστές ανάκλασης διάχυτου φωτός, διάχυτης και κατοπτρικής ανάκλασης, καθώς και η διάχυτη ακτινοβολία του χώρου, μαζί με την ακτινοβολία και την θέση όποιων άλλων φωτεινών πηγών. Αυτά τα δεδομένα βρίσκονται σε συνδυασμό με τις θέσεις (στο WCS) των κορυφών των τριγώνων από τα οποία απαρτίζεται το αντικείμενο και, φυσικά, τις παραμέτρους της κάμερας (όπως στη 2η εργασία).

Παρακάτω αναλύεται η ανάπτυξη των συναρτήσεων που ζητούνται από την εκφώνηση της εργασίας, καθώς και οι βασικότερες εκ των επιπλέον συναρτήσεων που χρειάστηκαν για την ολοκλήρωση του στόχου.

ambient_light

Δεδομένης διάχυτης ακτινοβολίας I_a και συντελεστή ανάκλασης διάχυτου φωτός k_a , επιστρέφει το φωτισμό ενός (άρα και όλων των σημείων ενός στερεού με σταθερό k_a , σαν αυτό που πραγματευόμαστε).

$$I_a = k_a \cdot I_a$$

diffuse_light

Για σημείο \mathbf{P} κάθε μία από τις θέσεις και τις εντάσεις του φωτός (entries στις `light_positions` και `light_intensities`), υπολογίζουμε τη διάχυτη ανάκλαση, βάσει της θεωρίας:

Έστω \mathcal{S} η θέση και \mathcal{J} ο 3x1 πίνακας της έντασης του φωτός

- $\bar{L} = \mathcal{S} - P$
- $\hat{L} = \frac{\bar{L}}{||\bar{L}||}$
- $\cos(\alpha) = (\hat{N}, \hat{L})$

Όπου α είναι η γωνία μεταξύ του μοναδιαίου \hat{L} και του μοναδιαίου \hat{N} , που είναι γνωστό και παριστά το **normal vector** της επιφάνειας της οποίας το P είναι μέρος.

Τώρα, αν, εν τέλει, βρεθεί πως $\cos(\alpha) > 0$, υπολογίζουμε:

$$I = k_d \cdot \cos(\alpha) \cdot \mathcal{J}$$

Στό τέλος, όλοι αυτοί οι πίνακες I για κάθε πηγή φωτός προστίθενται και το αποτέλεσμα πολλαπλασιάζεται με το **color** που δίνεται στη συνάρτηση για το σημείο P .

specular_light

Η συνάρτηση υπολογισμού φωτισμού από κατοπτρική ανάκλαση, δέχεται τα ίδια ορίσματα με την `diffuse_light`, με την προσθήκη της θέσης του παρατηρητή (δηλαδή της κάμερας, έστω \mathcal{C}) και του **συντελεστή Phong** n .

Ξανά, για κάθε entry στη λίστα των θέσεων και των εντάσεων των φωτεινών πηγών, κι αφού έχουμε εξάγει τα \hat{L} και $\cos(\alpha)$, ακριβώς όπως παραπάνω, έχουμε:

- $$\overline{V} = \mathcal{C} - P$$
- $$\hat{V} = \frac{\overline{V}}{\|\overline{V}\|}$$
- $$\cos(\beta - \alpha) = (2 \cdot \hat{N} \cdot \cos(\alpha) - \hat{L}) \cdot \hat{V}$$
- $$I = k_s \cdot \cos(\beta - \alpha)^n \cdot \mathcal{I}$$

Όπως και πριν, το άθροισμα των παραγόμενων I πολλαπλασιάζεται με το δοθέν **color** και επιστρέφεται.

calculate_normals

Με δεδομένες τις WCS συντεταγμένες των κορυφών των τριγώνων και του πίνακα **face_indices**, που περιγράφει από ποιες κορυφές αποτελείται το καθένα, θέλουμε να υπολογίσουμε τα **normal vectors** σε κάθε κορυφή.

Γι' αυτό το σκοπό, για κάθε τρίγωνο, βρίσκουμε το normal vector του και το προσθέτουμε στα αντίστοιχα index των κορυφών του στον πίνακα *normals*, όπου αποθηκεύεται το αποτέλεσμα που θα επιστραφεί. Αν A, B, C είναι οι κορυφές ενός τυχαίου τριγώνου, τότε για το normal vector του:

- $$\overline{V}_1 = B - A$$
- $$\overline{V}_2 = C - A$$
- $$\overline{N} = \overline{V}_1 \times \overline{V}_2$$

Στο τέλος όλων των επαναλήψεων (όταν δηλαδή έχουμε υπολογίσει το κάθετο διάνυσμα κάθε τριγώνου), έχουμε τη συνεισφορά του κάθε τριγώνου σε όσους κορυφές το αποτελούν. Επιστρέφουμε τα κανονικοποιημένα διανύσματα εντός του *normals*.

render_object

Είναι η κεντρική συνάρτηση του κώδικα. Μέσα σε αυτήν αρχικοποιείται η εικόνα ως πίνακας **numpy array (MxNx3)**, υπολογίζονται τα **κάθετα διανύσματα**, γίνεται **προβολή** στον κόσμο της κάμερας (άρα και **υπολογισμός των depths**) και **rasterize** στις τρισδιάστατες κορυφές και, τέλος κρατάμε μόνο τα τρίγωνα τα οποία έχουν **όλες τους τις κορυφές εντός των ορίων της εικόνας**.

Για αυτά τα τρίγωνα, βρίσκουμε τα **βαρύκεντρα**, των οποίων οι συντεταγμένες στο WCS είναι απλά ο μέσος όρος των ομόλογων συντεταγμένων των κορυφών.

Τώρα, για κάθε ένα τρίγωνο ξεχωριστά, με σειρά που καθορίζεται από τα βάθη τους, με τα πιο απομακρυσμένα να περνάνε πρώτα από επεξεργασία, χρωματίζουμε και σκιάζουμε την περιοχή που καταλαμβάνει. Το πώς, καθορίζεται από τη μετάβλητή `shader`, που παίρνει τις τιμές *"gouraud"* ή *"phong"*, ούτως ώστε να κληθεί η αντίστοιχη συνάρτηση.

Extra: total_lighting

Χρησιμοποιείται, ούτως ώστε να υπολογίσει το συνολικό φωτισμό πάνω σε ένα σημείο του στερεού. Καλεί τις `ambient_light()`, `diffuse_light()` και `specular_light()` και προσθέτει τα παραγόμενα αποτελέσματα.

shade_gouraud

Βρίσκει το χρώμα των τριών κορυφών του τριγώνου. Ύστερα, το μόνο που μένει είναι να καλέσουμε την `shade_triangle()` της πρώτης εργασίας με `shader = "gouraud"`. Το τρισδιάστατο σημείο που δίνετα στην `total_lighting()` είναι το βαρύκεντρο του τριγώνου.

shade_phong

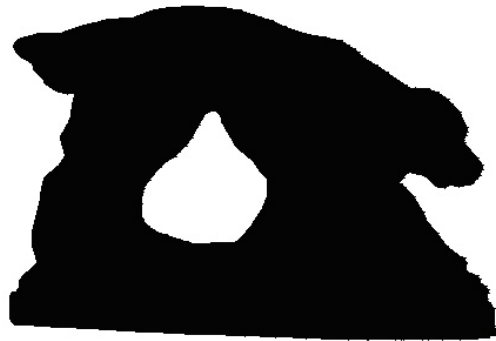
Τροποποιούμε το σώμα της συνάρτησης `shade_triangle()` για `shader = "gouraud"`: εκτός από το interpolation του χρώματος των κορυφών (και ύστερα των εσωτερικών του σημείων), κάνουμε interpolation και των normals.

Αποτελέσματα

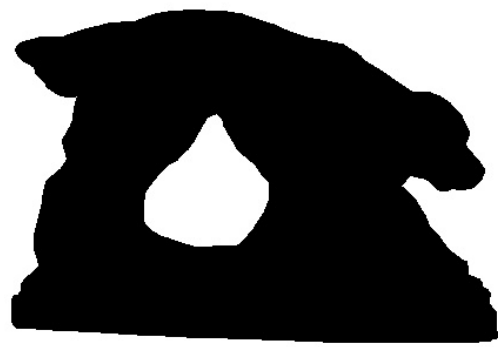
Παρακάτω παραθέτονται τα αποτελέσματα για `shader = "gouraud"` και έπειτα για `shader = "phong"` σε 4 περιπτώσεις:

- Όλα τα είδη του φωτισμού
- Διάχυτο φως
- Διάχυτη ανάκλαση
- Κατοπτρική ανάλυση

Gouraud



Phong



Σας ευχαριστώ για το χρόνο σας

Αντωνίου Αντώνιος - 9482 - aantonii@ece.auth.gr