

Γραφική με Υπολογιστές

Εργασία 3

Βασίλειος Αραϊλόπουλος
varailop@ece.auth.gr
AEM: 9475

Ιούλιος 2021

A. Φωτισμός και Υλικό επιφάνειας

Αρχικά, δημιουργήθηκαν οι δύο ζητούμενες κλάσεις *PhongMaterial* και *PointLight*. Η πρώτη περιλαμβάνει τους συντελεστές φωτισμού k_a, k_d, k_s και την σταθερά Phong για την κατοπτρική ανάκλαση, ενώ η δεύτερη περιλαμβάνει την θέση της φωτεινής πηγής και την ένταση που ακτινοβολεί για κάθε χρωματική συνιστώσα. Για τις υπόλοιπες συναρτήσεις εφαρμόστηκαν οι τύποι που περιγράφονται στις σημειώσεις. Αναλυτικότερα, η συνάρτηση που υλοποιεί το διάχυτο φως του περιβάλλοντος επιστρέφει την ένταση της τριχρωματικής ακτινοβολίας σύμφωνα με τον τύπο:

$$I = I_a k_a \quad (1)$$

Στην συνάρτηση που υλοποιεί την διάχυτη ανάκλαση δίνονται το σημείο της φωτεινής πηγής, το ζητούμενο σημείο P και το μοναδιαίο διάνυσμα \hat{N} που είναι κάθετο στην επιφάνεια του σημείου P. Η σχέση που εφαρμόζεται για τον υπολογισμό της έντασης είναι:

$$I = I_p k_d (\hat{N} \cdot \hat{L}) \quad (2)$$

Το \hat{L} είναι το μοναδιαίο διάνυσμα που είναι παράλληλο στην προσπίπτουσα φωτεινή ακτίνα. Αυτό βρίσκεται από την κανονικοποίηση του διανύσματος με αρχή το σημείο P και πέρας το σημείο της φωτεινής πηγής. Με τον παραπάνω τύπο δεν λαμβάνεται υπόψιν η απόσταση που έχει το σημείο από την πηγή, δηλαδή δεν υπάρχει εξασθένιση.

Η τελευταία συνάρτηση υλοποιεί το μοντέλο της κατοπτρικής ανάκλασης από μη ιδανικό ανακλαστήρα στην οποία εφαρμόζεται ο εξής τύπος:

$$I = I_p k_s \cos(b - a)^n \quad (3)$$

Από τις σημειώσεις για τον υπολογισμό του όρου $\cos(b - a)$ χρησιμοποιείται το εσωτερικό γινόμενο $\hat{R} \cdot \hat{V}$ με \hat{R} να είναι το μοναδιαίο διάνυσμα που προκύπτει από τον τύπο:

$$\hat{R} = 2\hat{N}(\hat{N} \cdot \hat{L}) - \hat{L} \quad (4)$$

Σημείωση: Στις συναρτήσεις που υπολογίζουν τον φωτισμό ζητείται στην εκφώνηση η ένταση να συνεισφέρει αθροιστικά στο χρώμα του σημείου. Ωστόσο, αποφάσισα σε καμία από τις τρεις συναρτήσεις να χρησιμοποιήσω την μεταβλητή `color`, δηλαδή να μην την προσθέτω στο τελικό αποτέλεσμα. Επέλεξα να κάνω αυτήν την αλλαγή γιατί όταν υπολογίζεται το χρώμα στις παρακάτω συναρτήσεις `gouraud_shading` και `phong_shading` πρέπει να προστεθούν και τα τρία είδη φωτισμού. Αν υπήρχε και το `color` σε κάθε μια από αυτές τις τρεις συναρτήσεις αθροίζοντας τα τρία είδη φωτισμού θα περιλαμβανόταν το χρώμα 3 φορές. Έτσι, για να αποφύγω αυτό το πρόβλημα έβγαλα τον όρο `color` που προστίθονταν στον φωτισμό και προσθέτω το `color` μια φορά μόνο όταν πάω να υπολογίσω στις συναρτήσεις του shading το χρώμα ενός pixel.

B. Shading

B1. Υπολογισμός κανονικών διανυσμάτων

Σε αυτήν την συνάρτηση υπολογίζονται τα κανονικά διανύσματα των κορυφών των τριγώνων. Αρχικά, βρίσκονται για όλα τα τρίγωνα τα διανύσματα αυτά με τον κανόνα του δεξιόστροφου κοχλία από τις κορυφές του κάθε τριγώνου. Αναλυτικότερα, ονομάζονται οι κορυφές A, B και C , δημιουργούνται τα διανύσματα \vec{AB} και \vec{BC} και από το εξωτερικό γινόμενο $\vec{AB} \times \vec{BC}$ βρίσκεται το κάθετο μοναδιαίο διάνυσμα του κάθε τριγώνου. Έπειτα πρέπει να βρεθεί το μοναδιαίο διάνυσμα που αντιστοιχεί στην κάθε κορυφή. Για να γίνει κάτι τέτοιο πρέπει να βρεθούν όλα τα τρίγωνα στα οποία ανήκει μία κορυφή, να προστεθούν όλα τα κανονικά διανύσματα αυτών των τριγώνων και το άθροισμα αυτό να κανονικοποιηθεί. Η παραπάνω διαδικασία γίνεται με μία *for loop* που τρέχει κάθε κορυφή και με την συνάρτηση *find* βρίσκονται όλα τα τρίγωνα που περιλαμβάνουν την κορυφή. Έτσι, επιστρέφεται ο ζητούμενος πίνακας $3 \times N_v$ με N_v να είναι ο αριθμός όλων των σημείων της επιφάνειας.

B2. Συνάρτηση φωτογράφισης

Συνάρτηση παρόμοια με αυτήν που χρησιμοποιήθηκε στην εργασία 2. Ξεκινάει βρίσκοντας τα κανονικά διανύσματα της επιφάνειας καλώντας την συνάρτηση του B1. Με τις συναρτήσεις *project_cam_ku* και *rasterize* της εργασίας 2 βρίσκονται οι προβολές των σημείων και μετατρέπονται οι μονάδες μήκους σε θέσεις pixel. Και τέλος ακολουθώντας την ίδια διαδικασία με τις προηγούμενες εργασίες υπολογίζεται το βάθος του κάθε τριγώνου και γίνεται η πλήρωση τους με την κατάλληλη συνάρτηση ανάλογα με την τιμή του shader.

B3. Gouraud Shading

Συνάρτηση όμοια με την *paint_triangle_gouraud* των εργασιών 1 και 2. Η μόνη διαφορά είναι ότι στην αρχή της συνάρτησης πρέπει να υπολογιστεί το χρώμα της κάθε κορυφής βασιζόμενοι στον φωτισμό. Για αυτόν τον σκοπό χρησιμοποιούνται οι κλάσεις και οι συναρτήσεις του ερωτήματος A. Υπολογίζονται, δηλαδή, για όλες τις κορυφές ενός

συγκεκριμένου τριγώνου, η ένταση της ακτινοβολίας ανάλογα με τις τιμές της κλάσης *Phong_Material* για τις τρεις περιπτώσεις φωτισμού, οι οποίες προσθέτονται στο χρώμα του σημείου. Χρησιμοποιώντας αυτά τα χρώματα γίνεται η πλήρωση τριγώνων με τον χρωματισμό gouraud.

B4. Phong Shading

Η διαφορά της συνάρτησης με την *shade_gouraud* είναι ότι όπως γίνεται και η παρεμβολή μεταξύ των σημείων για τον υπολογισμό του χρώματος, πρέπει να γίνει και με τον υπολογισμό του φωτισμού. Για κάθε ενδιαμέσο σημείο γίνεται γραμμική παρεμβολή για να βρεθεί το κανονικό του διάνυσμα και με την χρήση των συναρτήσεων του μέρους A βρίσκεται σε κάθε σημείο του τριγώνου η ένταση της φωτεινής ακτινοβολίας και συνεπώς η νέα τριάδα χρωμάτων.

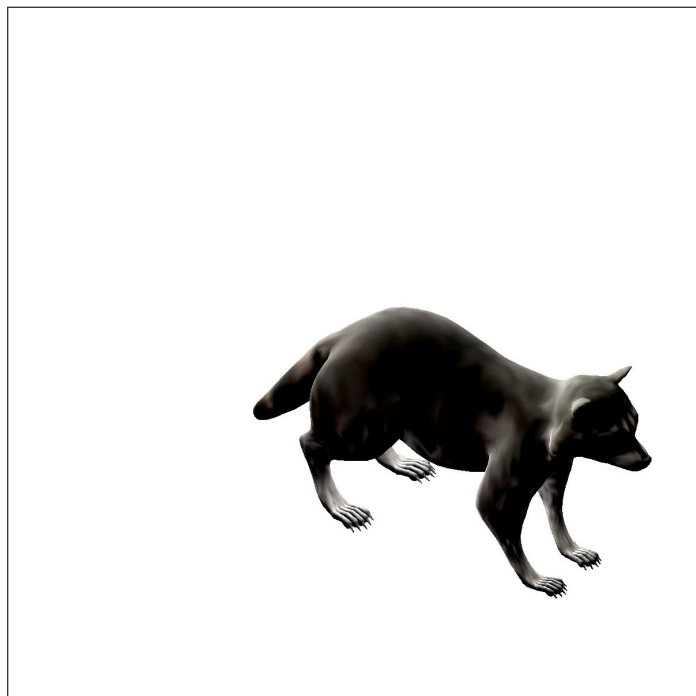
Σημείωση: Μετά από συζήτηση με συναδέλφους κατάλαβα ότι με μία μικρή αλλαγή του κώδικα των συναρτήσεων που περιγράφηκαν στο B3 και B4 ο χρόνος εκτέλεσης του demo μειώθηκε από τα 5 λεπτά στα 5 δευτερόλεπτα! Αναλυτικότερα, αυτή η αλλαγή έγινε στην θέση που βρισκόταν η ανάθεση της εικόνας στην έξοδο της συνάρτησης. Αρχικά, ανέθετα την εικόνα X, που ερχόταν ως είσοδος στην συνάρτηση, στην μεταβλητή Y που είναι η έξοδος της συνάρτησης και "έβαφα" το τρίγωνο πάνω στην Y. Τώρα "βάφω" το τρίγωνο στην μεταβλητή X της εισόδου και αναθέτω την τιμή του Y στην τελευταία γραμμή της συνάρτησης.

Demo και αποτελέσματα

Για την δημιουργία του αρχείου χρησιμοποιήθηκε ο κορμός από το demo που δόθηκε. Οι γραμμές κώδικα που προστέθηκαν είναι για την αρχικοποίηση των κλάσεων *PhongMaterial* και *PointLight* και για την εκτέλεση της συνάρτησης *render*. Πριν από κάθε μία από τις οχτώ περιπτώσεις, πρέπει να αλλάξουν οι μεταβλητές k_a , k_d και k_s που αποθηκεύονται στο *PhongMaterial*, όπως ζητείται από την εκφώνηση. Παρακάτω φαίνονται αυτές οι οχτώ εικόνες που προκύπτουν από την εκτέλεση του demo. Όπως, είχε αναφερθεί και στην εργασία δύο αποφάσισα να κάνω τις κατάλληλες αλλαγές ώστε το ραχούν να έχει σωστό προσανατολισμό και να φαίνεται καλύτερο.



gouraud_ambient.jpg



gouraud_diffusion.jpg



gouraud_specular.jpg



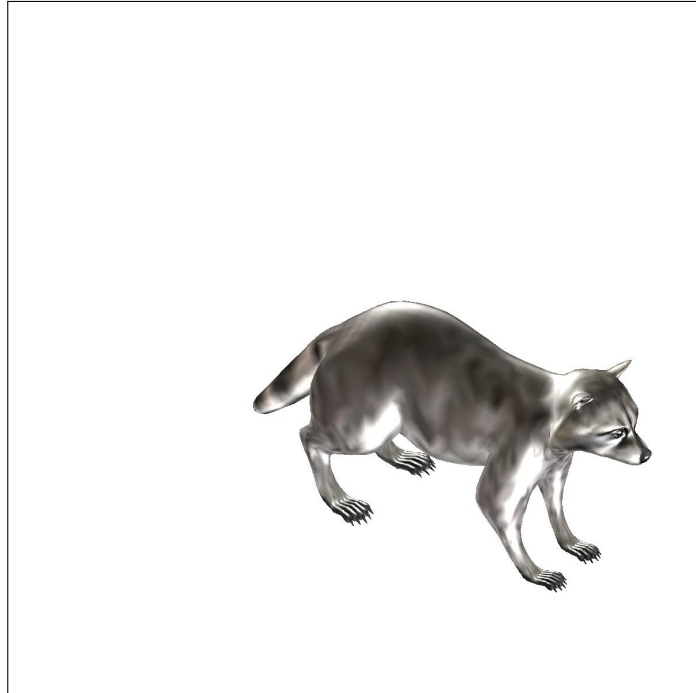
gouraud_all.jpg



phong_ambient.jpg



phong_diffusion.jpg



phong_specular.jpg



phong_all.jpg

Αρχικά, το ραχούν με κάθε φωτισμό φαίνεται να είναι πολύ φωτεινό και στο τελικό αποτέλεσμα χάνονται κάποια χαρακτηριστικά του, όπως η μαύρη λωρίδα στα μάτια του. Συγκρίνοντας, τώρα, με μια γρήγορη ματιά δεν φαίνονται σημαντικές διαφορές ανάμεσα στις δύο μεθόδους shading. Παρατηρώντας, όμως, προσεκτικά τις δύο εικόνες *gouraud_all.png* και *phong_all.png* βλέπουμε ότι στις λεπτομέρειες κερδίζει η δεύτερη. Αυτό γιατί στην πρώτη εικόνα σε ένα σημείο του ραχούν φαίνονται λίγο έντονα τα τρίγωνα, ενώ στην δεύτερη εικόνα φαίνονται πιο ομαλές οι μεταβάσεις και αυτή η ατέλεια σχεδόν χάνεται.



(a) Λεπτομέρεια gouraud



(b) Λεπτομέρεια phong

Παρακάτω φαίνεται ένας ενδεικτικός χρόνος εκτέλεσης για τις 8 εικόνες. Είναι φανερό ότι για την εκτέλεση του shader gouraud χρειάζεται λιγότερος χρόνος σε σχέση με τον phong. Αυτό συμβαίνει μάλλον εξαιτίας του υπολογισμού με γραμμική παρεμβολή των κάθετων διανυσμάτων σε κάθε σημείο του τριγώνου.

```
Elapsed time is 5.144924 seconds.  
Elapsed time is 5.433523 seconds.  
Elapsed time is 5.082061 seconds.  
Elapsed time is 5.224670 seconds.  
Elapsed time is 12.449093 seconds.  
Elapsed time is 12.635393 seconds.  
Elapsed time is 12.383145 seconds.  
Elapsed time is 12.204595 seconds.
```

Ενδεικτικός χρόνος εκτέλεσης.

Δοκίμασα επιπλέον να μην προσθέτω στις συναρτήσεις του μέρους A την μεταβλητή color αλλά να την πολλαπλασιάζω ένα προς ένα και βγήκε το όχι και τόσο καλό αποτέλεσμα:



(a) Ambient



(b) Diffusion



(c) Specular



(d) All