

# Φωτισμός 3D - Θέαση

Illumination - Phong Reflection Model



Αριστοστέλειο Πανεπιστήμιο  
Θεσσαλονίκης

Χάρης Φίλης AEM: 9449  
GitHub Repository Link: [🔗](#)

July 9, 2023

# Contents

0.1	Θεωρητική Εισαγωγή . . . . .	2
0.2	Υλοποίηση Φωτισμού - Code . . . . .	5
0.3	Αποτελέσματα και Παραγόμενες Εικόνες - Rendered Images . .	8

# List of Figures

1	Diffuse Reflection . . . . .	3
2	Reflection Vectors . . . . .	3
3	Specular Reflection - Diffuse Reflection Comparison . . . . .	4
4	Specular Reflection Reflection Vectors . . . . .	5
5	Αποτελέσματα Gouraud Shader . . . . .	8
6	Αποτελέσματα Phong Shader . . . . .	9
7	Παράδειγμα Εκτέλεσης Demo . . . . .	9

## 0.1 Θεωρητική Εισαγωγή

Ένα από τα σημαντικότερα μέρη μια μηχανής γραφικών που αποσκοπούν στην ρεαλιστική απεικόνιση συνθετικών δεδομένων είναι το κομμάτι που ασχολείται με τον φωτισμό. Στην γραφική υπολογιστών σήμερα ο φωτισμός επιτελείται μέσω αλγορίθμων διαδικασίας ιχνηλάτησης της πορείας της ακτίνας (*ray tracing*) από μη σημειακές πηγές και στην συνέχεια την εφαρμογή της αλλαγής στο χρώμα των κορυφών των τριγώνων με βάση ενός μοντέλου ανάκλασης (*reflection model*). Στα πλαίσια του μαθήματος βλέπουμε το κομμάτι κυρίως του μοντέλου ανάκλασης ακτίνας που προκύπτει από σημειακή πηγή (και δεν ανακλάται σε ενδιάμεσα αντικείμενα πριν φτάσει στο αντικείμενο προβολής).

Συγκεκριμένα με τον όρο φωτισμό εννοούμε την διαδικασία υπολογισμού της έντασης της φωτεινής ακτινοβολία που λαμβάνει ο θεατής της σκηνής που δημιουργείται από την μηχανή γραφικών. Η φωτεινή αυτή ακτινοβολία μπορεί να προέρχεται από ανάκλαση, διάχυση ή και αυτοματισμό. Έτσι με βάση αυτή την παραδοχή διακρίνονται τρία βασικά μοντέλα φωτισμού.

### Αυτοφωτισμός - Illumination

Σε αυτή την περίπτωση μιλάμε για το φως που παράγουν τα ίδια τα σώματα και ο υπολογισμός του φωτός που φτάνει στον παρατηρητή, περιορίζεται στον υπολογισμό της έντασης της προσλαμβανόμενης φωτεινότητας το οποίο επηρεάζεται από το μήκος κύματος της ακτινοβολίας  $\lambda$  όσο και από έναν εσωτερικό συντελεστή φωτεινότητας  $k_{\lambda}$ . Συνεπώς η προσλαμβανόμενη φωτεινότητα σε κάθε χρώμα υπολογίζεται από το  $I_{r,g,b} = \lambda r, g, b * k_{r,g,b,\lambda}$

### Διάχυτο Φως - Ambient Light

Το διάχυτο φως αφορά το προσλαμβανόμενο το οποίο προέρχεται από σώματα που φωτίζονται από άλλα σώματα. Για παράδειγμα το φως του φεγγαριού την νύχτα έχει μια ομοιόμορφη κατανομή που προσεγγίζει την έννοια του διάχυτους φωτός στο περιβάλλον. Ο υπολογισμός της έντασης αυτού του φωτός και κατ'επέκταση του πόσο αλλάζουν τα χρώματα του φωτιζόμενου αντικειμένου κατά τον φωτισμό δίνεται από τον τύπο:

$$I_{\lambda} = I_{\alpha\lambda} \cdot k_{\alpha\lambda}$$

όπου ο συντελεστής ανάκλασης διάχυτου φωτός  $k_{\alpha\lambda}$  αποτελεί χαρακτηριστικό της ανακλαστική επιφάνειας στην οποία ανήκει το μελετώμενο σημείο και  $I_{\alpha\lambda}$  είναι η ένταση του διάχυτου φωτισμού. Για τιμή του γινομένου 1 το αντικείμενο εμφανίζεται απολύτως λευκό.

### Φως Διάχυτης Ανάκλασης - Diffuse Reflection Light

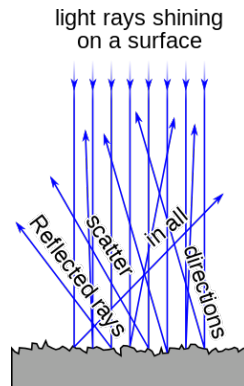


Figure 1: Diffuse Reflection

Με τον όρο διάχυτη ανάκλαση (*diffuse reflection - Lambertian reflection*) ορίζεται το είδος του φωτισμού που προκύπτει από ανάκλαση των φωτεινών ακτίνων προερχόμενων από σημειακή πηγή που διαχέει το φως όπως το φεγγάρι προς όλες τις κατευθύνσεις, σύμφωνα με τον νόμο *Lambert*. Το μοντέλο αυτό προτείνει, πως μια στοιχειώδης περιοχή  $dE$  περί του σημείου στο οποίο πέφτει μια φωτεινή ακτίνα ανακλά προς συγκεκριμένη κατεύθυνση, φωτεινή ισχύ:

$$I_\lambda \propto dE \cdot \cos b$$

όπου,  $b$  η γωνία θέασης του που σχηματίζεται μεταξύ, του κάθετου στην επιφάνεια, μοναδιαίου διανύσματος  $\hat{N}$  και του μοναδιαίου διανύσματος που δείχνει στον παρατηρητή  $\hat{V}$  όπως φαίνεται στην εικόνα παρακάτω:

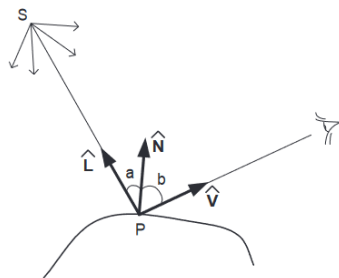


Figure 2: Reflection Vectors

Σύμφωνα με αυτό το μοντέλο η προσλαμβανόμενη φωτεινότητα  $I_\lambda$  εξαρτάται κυρίως από την υφή της επιφάνειας και όχι από την θέση του παρατηρητή (εικονική

κάμερα στα συστήματα γραφικής) καθώς και από το πως πέφτει η ακτίνα του φωτός πάνω στην επιφάνεια.

Συνεπώς ο τρόπος υπολογισμού της έντασης που φτάνει στα μάτια μας αυτού του μοντέλου είναι:

$$I_\lambda = I_{p,\lambda} \cdot k_d \cdot \langle \hat{N} \cdot \hat{V} \rangle$$

όπου  $k_d$  είναι ο συντελεστής διάχυτης ανάκλασης ενώ  $I_{p,\lambda}$  είναι η ένταση της προσπίπτουσας ακτινοβολίας. Σε αυτά μπορεί να προστεθεί και η εξασθένιση της φωτεινής ακτινοβολίας λόγω απόστασης  $f_{att} = \frac{1}{d^2}$  ωστόσο αυτό δεν κρίνεται σημαντικό στις περισσότερες περιπτώσεις και συνήθως παίζει αισθητικό ρόλο.

### Κατοπτρικός Φωτισμός - Specular Lighting

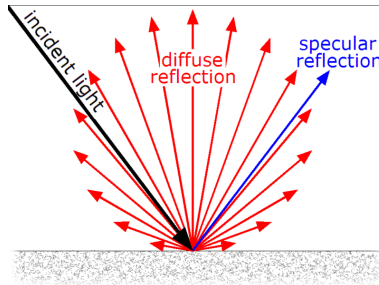


Figure 3: Specular Reflection - Diffuse Reflection Comparison

Το μοντέλο κατοπτρικής ανάκλασης προτείνει πως σημαντικό ρόλο στην μεταβολή του χρώματος που παρατηρεί η εικονική κάμερα παίζει η ανακλώμενη ακτίνα που πλέον όπως φαίνεται και στην παραπάνω εικόνα είναι συγκεντρωμένη και δεν διαχέεται στον χώρο.

Συνεπώς το μοντέλο αυτό αποτυπώνει το φωτισμό σε επιφάνειες που φαίνονται γυαλιστερές και αποτελούν τέλεια κάτοπτρα του φωτός. Έτσι η τιμή της προσλαμβανόμενης ακτινοβολίας δίνεται από τον τύπο:

$$I_\lambda = I_{p\lambda} \cdot k_s \cdot \delta(b - a)$$

Και ο παρατηρητής αντιλαμβάνεται την ακτινοβολία μόνο όταν η γωνία  $b$  ισούται με την γωνία  $a$  (γωνία πρόσπτωσης = γωνία ανακλάσεως). Σε άλλες γωνίες αυτό το μοντέλο μετατρέπεται σε μοντέλο ιδανικού ανακλαστήρα όπου η  $\delta$  του Dirac αντικαθίσταται με τον συνημίτονο της γωνία  $b-a$ .

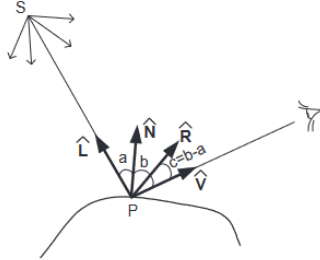


Figure 4: Specular Reflection Reflection Vectors

### Κατοπτρικό μοντέλο ανάκλασης - Phong Reflection Model

Το κατοπτρικό μοντέλο ανάκλασης phong είναι αυτό που ολοκληρώνει το ιδανικό μοντέλο ανακλαστήρα εισάγοντας μια σταθερά στην οποία υψώνεται το συνημίτονο της γωνία  $b-a$ . Συνεπώς:

$$I_{\lambda} = I_{p\lambda} \cdot k_{s\lambda} \cdot (\cos(b-a))^{n_{phong}}$$

Αυτό με χρήση του νόμου της ανάκλασης καθώς και του εσωτερικού γινομένου και με βάση τα διανύσματα κατοπτρικής ανάκλασης στο Σχήμα 4 γράφεται:

$$\begin{aligned} I_{\lambda} &= I_{p\lambda} k_{s\lambda} (\hat{R} \cdot \hat{V})^n \\ &= I_{p\lambda} k_{s\lambda} ((2\hat{N}(\hat{N} \cdot \hat{L}) - \hat{L}) \cdot \hat{V})^n \end{aligned}$$

Η ολοκλήρωση του μοντέλου φωτισμού αντιστοιχεί στο άθροισμα όλων των παραπάνω φωτισμών στο χρώμα του αντικειμένου που προβάλλεται.

## 0.2 Υλοποίηση Φωτισμού - Code

Στο συγκεκριμένο παραδοτέο έγινε χρήση των συναρτήσεων που δημιουργήθηκαν στα προηγούμενα δύο παραδοτέα καθώς και νέων αλλά και παραμετροποιήσεων των ήδη υπαρχουσών. Το μεγαλύτερο μέρος των υλοποιημένων συναρτήσεων είναι εντός του καταλόγου `/src/common_utils` ενώ τα demo που βρίσκονται στη ρίζα(root) τρέχουν χωρίς εξωτερικά ορίσματα και κάνουν κλήση συναρτήσεων από αυτόν τον κατάλογο. Τέλος η συνάρτηση φόρτωσης του numpy array αλλάζει καθώς αλλάζουν οι μεταβλητές που περιέχει το αρχείο.

### Αντικείμενα PhongMaterial, PointLight

Τα συγκεκριμένα αντικείμενα όπως αναγράφεται στην εκφώνηση κρατάνε τις απαραίτητες μεταβλητές και συντελεστές για την υλοποίηση του κατοπτρικού

μοντέλου ανάκλασης. Περισσότερο παίζουν ρόλο δομές δεδομένων για να είναι πιο οργανωμένος ο κώδικας παρά έχουν την πλήρη λειτουργικότητα ενός αντικειμένου όπως ορίζουν αρχές του αντικειμενοστραφή προγραμματισμού. Συγκεκριμένα το αντικείμενο PhongMaterial κρατά του συντελεστές ανάκλασης (ka, kd, ks) για κάθε είδος φωτισμού καθώς και τον συντελεστή  $n\_phong$ , ενώ το PointLight αντικείμενο κρατά τη θέση των φωτεινών πηγών καθώς και την ένταση σε κάθε κανάλι των φωτεινών πηγών. Επιπλέον γίνεται έλεγχος αν οι εντάσεις των φωτεινών πηγών ανήκουν στο διάστημα  $[0,1]$ .

### Υπολογισμός των κανονικών διανυσμάτων των κορυφών των τριγώνων

Υπεύθυνη για τον υπολογισμό των κανονικών διανυσμάτων των κορυφών των τριγώνων είναι η συνάρτηση `calculate_Normals`. Η συνάρτηση παίρνει ως ορίσματα τις 3D κορυφές των τριγώνων και τους δείκτες στα `faces` δηλαδή στο που ξεκινάει νέα τριάδα τριγώνου. Επομένως επιλέγονται οι κορυφές του κάθε τριγώνου και για να υπολογιστεί το κάθετο διάνυσμα στην επιφάνεια υπολογίζεται το εξωτερικό γινόμενο δυο πλευρών του τριγώνου. Στην συνέχεια τα κάθετα διανύσματα κανονικοποιούνται πριν επιστραφούν από την συνάρτηση.

### Μοντέλο Φωτισμού - PhongIlluminationModel

Στο αρχείο `lighting.py` βρίσκεται το αντικείμενο που είναι υπεύθυνο για τον υπολογισμό κάθε είδους φωτισμού. Ως ορίσματα δέχεται τις παραμέτρους από τα αντικείμενα που περιγράφηκαν προηγουμένως, ενώ διαθέτει 4 συναρτήσεις που υλοποιούν τα μοντέλα φωτισμού που περιγράφηκαν στην Θεωρητική Εισαγωγή. Η συνάρτηση `light` υπολογίζει και το διάχυτο και το φως αυτοματισμού και το κατοπτρικό φως και τα αθροίζει επιστρέφοντας τον φωτισμό ενός σημείου με βάση το ιδανικό μοντέλο φωτισμού. Οι συναρτήσεις είναι απλές και έχουν και σχόλια επεξήγησης.

### Διόρθωση Αλγορίθμων Σκίασης και Συνάρτηση Αποτύπωσης Εικόνας

Η συνάρτηση αποτύπωσης/φωτογράφισης χρησιμοποιεί μια παραμετροποιημένη μορφή της `render` συνάρτησης που κάνει `render` τον `rasterized canvas`. Αρχικά χρησιμοποιεί το 2ο παραδοτέο για να μετατρέψει τα 3D σημεία σε σημεία πάνω στην φωτογραφία και επίσης υπολογίζει και τα κάθετα διανύσματα. Στην συνέχεια ανάλογα την επιλογή της `binary` τιμής `shader` επιλέγεται ο κατάλληλος σκιαστής και καλείται η `render_with_Illumination` η οποία τοποθετεί τα τρίγωνα με ανάποδο `depth order` και στην συνέχεια κάθε τρίγωνο το κάνει σκίαση

με μία από τις συναρτήσεις `shadePhong` ή `shadeGouraudRev` η οποίες ζητείται να υλοποιηθούν ξανά από το παραδοτέο.

### Συναρτήσεις Σκίασης

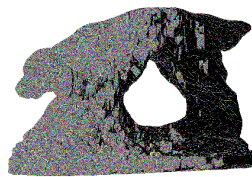
Σε αυτό το παραδοτέο παλιές συναρτήσεις σκίασης παραμετροποιούνται και πλέον υπολογίζουν πέραν από το χρώμα του κάθε σημείου που είναι εντός του τριγώνου και το φωτισμό του τριγώνου και την μεταβολή που προκαλείται στο χρώμα του. Συγκεκριμένα υπάρχει η μεταβλητή `pointLights` που έχει ένα πλήθος φωτεινών πηγών με συγκεκριμένες θέσεις και εντάσεις χρωμάτων καθώς και ειδική μεταβλητή για το υλικό το οποίο φωτίζεται το οποίο συναντάται ως `mat` ή `phongMaterial`.

Αρχικά υπολογίζονται οι ενεργές ακμές και τα σημεία που είναι εντός του τριγώνου με βάση το πρώτο παραδοτέο. Ωστόσο, η διαφορά μεταξύ των δύο shaders είναι ότι ο `phong shader` εφαρμόζει την παρεμβολή που εφαρμόζεται στον αλγόριθμο `scanline` και πάνω στα κάθετα διανύσματα των τριγώνων. Έτσι κάνει μια πιο ομοιόμορφη μετάβαση στο πως ανακλάται το κατοπτρικό και το διάχυτο φως. Αντίθετα αυτή η παρεμβολή δεν γίνεται στο κομμάτι του `gouraud shader` και το χρώμα υπολογίζεται με βάση τα μόνο την παρεμβολή στο χρώμα των κορυφών, το οποίο έχει αναβαθμιστεί με βάση τα κάθετα διανύσματα που δίνονται ως ορίσματα. Ο τρόπος που επιλέγονται τα κάθετα διανύσματα γίνεται με ζευγάρια σημείων του τριγώνου στις ακμές του. Μέσα στους shaders δημιουργούνται αντικείμενα φωτισμού ανάλογα τον αριθμό πηγών και τοποθετούνται σε πίνακα. Για κάθε πηγή υπολογίζεται ο φωτισμός και προστίθεται στο χρώμα της εικόνας ανάλογα τι είδος φωτισμού θέλουμε με βάση την `LightParam`. Συγκεκριμένα μπορούμε να έχουμε κάθε μοντέλο φωτισμού ξεχωριστά ή όλα μαζί αθροιστικά. Ο λόγος που διαφέρουν οι δύο shaders όπως είπαμε είναι στο γεγονός ότι χρησιμοποιούν διαφορετικά τα κανονικά διανύσματα και στον `phong shader` γίνεται μια διαδικασία υπολογισμού του μέσου ας πούμε κανονικού διανύσματος της επιφάνειας του τριγώνου με κανονικά τριγραμμική παρεμβολή η οποία ωστόσο απλοποιείται σε τρεις γραμμικές παρεμβολές με βάση τον αλγόριθμο `scanline`.

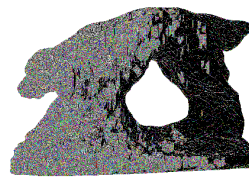


### 0.3 Αποτελέσματα και Παραγόμενες Εικόνες - Rendered Images

Λόγω προβλημάτων στην υλοποίηση του πρώτου παραδοτέου που είχε θέμα με την σκίαση και το ανακάτεμα των χρωμάτων υπάρχει θέμα στην αποτύπωση των χρωμάτων στις εικόνες. Από ότι φαίνεται μετά από ορισμένες τροποποιήσεις το 2 παραδοτέο εκτελεί σωστά την προβολή και τους μετασχηματισμούς ενώ ο φωτισμός μια και δεν είχε σημαντικές αλλαγές στον κώδικα ενώ οι συναρτήσεις φωτισμού ήταν σχετικά απλές θεωρώ πως δεν έχουν θέμα. Ωστόσο αυτό δεν αλλάζει το μη ικανοποιητικό αποτέλεσμα. Παρακάτω παρατίθενται 4 εικόνες για κάθε είδος σκιαστή (shader) καθώς υπάρχει και δεύτερη παράμετρο ελέγχου που επιλέγει μεταξύ μοντέλων φωτισμού LightParam.



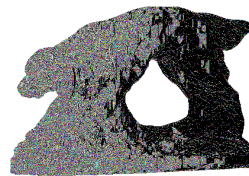
(a) Ambient Lighting



(b) Diffuse Lighting



(c) Specular Lighting



(d) Full Phong Reflection

Figure 5: Αποτελέσματα Gouraud Shader

Όσον αφορά τον χρόνο εκτέλεσης και στα δύο demo, παρατηρείται ότι στο demo\_phong ο χρόνος είναι κατά 1 λεπτό σχεδόν μεγαλύτερος στα αντίστοιχα μοντέλα φωτισμού ενώ μεταξύ ενός μοντέλου φωτισμού και του επόμενου υπ-

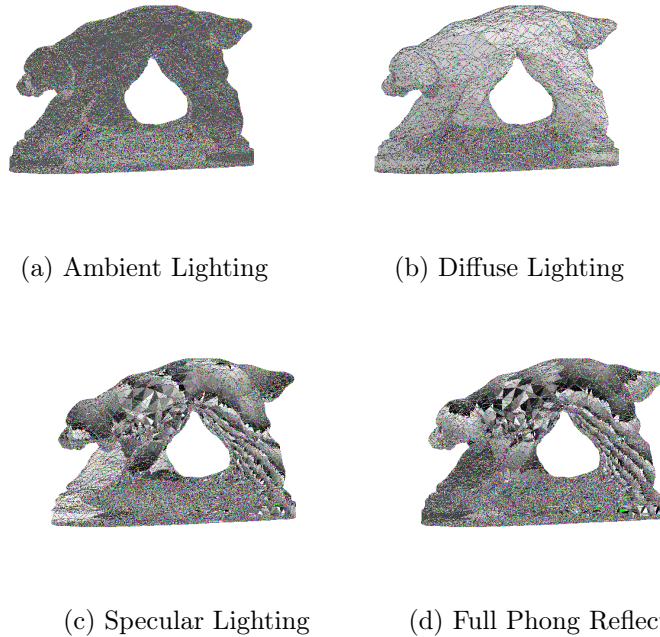


Figure 6: Αποτελέσματα Phong Shader

άρχει καθυστέρηση 9-15 δευτερολέπτων και συνολικά ο πιο αργός αλγόριθμος τρέχει περίπου στα 2 λεπτά. Αυτά τα αποτελέσματα είναι λογικά δεδομένου ότι οι αλγόριθμοι είναι σειριακοί και δεν τρέχουν παράλληλα στην κάρτα γραφικών και επιπλέον από το ένα επίπεδο φωτισμού στο επόμενο έχουμε αύξηση της υπολογιστικής πολυπλοκότητας. Επομένως shade\_gouraud avg ETA:1min , shade\_phong avg ETA:2min. Αυτά τυπώνονται κατά την κλήση των προγραμμάτων.

Ο τρόπος κλήσης το demo είναι χωρίς να πειραχτούν οι αντίστοιχοι φάκελοι να καλέσουμε τον interpreter και το όνομα του script i.e. python.exe ./demo\_phong.py  
Τα demo έχουν ελεγχθεί ότι τρέχουν και παρατίθεται τι τυπώνει το τερματικό κατά την διάρκεια εκτέλεσης:

```
Shading diffuse only lighting finished in 2 minutes 36 seconds
Processing triangles: 100% | 999/9999 [01:00:00:00, 51.7617/s]
Shading Specular only lighting finished in 3 minutes 9 seconds
Processing triangles: 100% | 999/9999 [01:29:00:00, 47.7215/s]
Shading Phong full lighting finished in 3 minutes 30 seconds
```

Figure 7: Παράδειγμα Εκτέλεσης Demo