

Εργασία 3: Θέαση

Πλευρίδη Βασιλική Βαρβάρα (AEM:10454)

Ιούνιος 2024

1 Εισαγωγή

Στόχος της παρούσας εργασίας είναι η δημιουργία ενός πλήρους πλαισίου δημιουργίας φωτογραφιών μιας εικονικής σκηνής υλοποιώντας συναρτήσεις για τον υπολογισμό της έντασης της φωτεινής ακτινοβολίας, προερχόμενη από διάχυτο φως στο περιβάλλον (ambient light), διάχυτη ανάκλαση (diffuse reflection) και κατοπτρική ανάκλαση (specular reflection), αλλά και την προσαρμογή μερικών συναρτήσεων από τα προηγούμενα παραδοτέα.

2 Συναρτήσεις και σχολιασμός υλοποίησης

Παρακάτω σχολιάζονται αναλυτικά οι ζητούμενες συναρτήσεις καθώς και πως αυτές υλοποιήθηκαν σε γλώσσα προγραμματισμού python.

2.1 Συνάρτηση light

Η συγκεκριμένη συνάρτηση υλοποιήθηκε στο ξεχωριστό αρχείο light.py. Στόχος της είναι ο υπολογισμός του φωτισμού σε ένα συγκεκριμένο σημείο το οποίο ανήκει σε μία επιφάνεια με υλικό τύπου- Phong λόγω του διάχυτου φωτισμού από το περιβάλλον, διάχυτης ανάκλασης, και κατοπτρικής ανάκλασης. Οι εξισώσεις που χρησιμοποιούνται για την διαμόρφωση του τελικού μοντέλου φωτισμού είναι οι εξής:

- Συνιστώσα φωτισμού από το περιβάλλον:

$$I_{\text{ambient}} = k_a \cdot I_{\text{ambient_light}}$$

όπου k_a είναι ο συντελεστής διάχυτου φωτός από το περιβάλλον

- Συνιστώσα διάχυτης ανάκλασης:

$$I_{\text{diffuse}} = k_d \cdot (\mathbf{N} \cdot \mathbf{L}) \cdot I_{\text{light}}$$

όπου k_d είναι ο συντελεστής διάχυτης ανάκλασης, \mathbf{N} είναι το κανονικό διάνυσμα της επιφάνειας, \mathbf{L} είναι το διάνυσμα κατεύθυνσης του φωτός, και I_{light} είναι η ένταση της πηγής φωτός.

- Συνιστώσα κατοπτρικής ανάκλασης:

$$I_{\text{specular}} = k_s \cdot (\mathbf{R} \cdot \mathbf{V})^n \cdot I_{\text{light}}$$

όπου k_s είναι ο συντελεστής κατοπτρικής ανάκλασης, \mathbf{R} είναι το διάνυσμα ανάκλασης, \mathbf{V} είναι το διάνυσμα προβολής, και n είναι η σταθερά Phong.

- Διάνυσμα ανάκλασης \mathbf{R} :

$$\mathbf{R} = 2(\mathbf{N} \cdot \mathbf{L})\mathbf{N} - \mathbf{L}$$

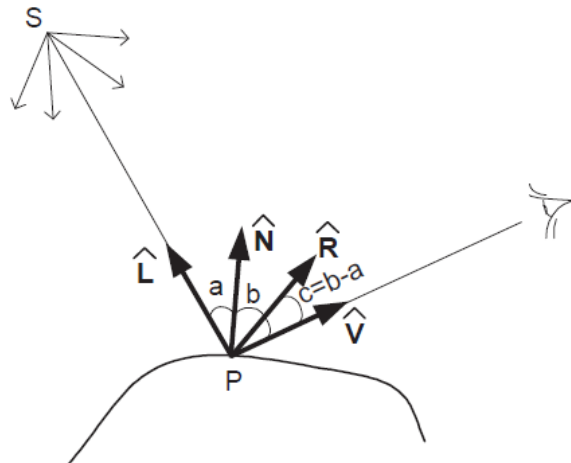
- Η τελική ένταση I στο σημείο στην επιφάνεια:

$$I = I_{\text{ambient}} + \sum_{i=1}^N (I_{\text{diffuse}_i} + I_{\text{specular}_i})$$

Όταν έχουμε πολλές πηγές φωτός, η συνολική φωτεινότητα στο σημείο, όπως φαίνεται και από τον παραπάνω τύπο, υπολογίζεται αθροίζοντας τις συνεισφορές από κάθε πηγή φωτός. Για κάθε πηγή φωτός i , προσθέτουμε την διάχυτη συνιστώσα I_{diffuse_i} και την κατοπτρική συνιστώσα I_{specular_i} , ώστε να λάβουμε υπόψη την συμβολή κάθε πηγής στην συνολική φωτεινότητα.

Σημαντικό είναι να σχολιαστεί ότι στην περίπτωση που η τιμή οποιασδήποτε συνιστώσας φωτισμού προκύψει αρνητική, όπως είναι λογικό, εξισώνεται με το 0 καθώς η αρνητική φωτεινότητα δεν έχει φυσική σημασία. Επίσης, όλα τα παραπάνω διανύσματα είναι μοναδιαία. Τα \mathbf{V} , \mathbf{R} και \mathbf{L} κανονικοποιούνται εντός της συνάρτησης, διαιρώντας τα με την νόρμα τους, ενώ το κανονικό διάνυσμα \mathbf{N} είναι ήδη σε αυτή την μορφή.

Παρακάτω, για καλύτερη κατανόηση των κατευθύνσεων των διανυσμάτων, παρουσιάζονται σχηματικά τα διανύσματα καθώς και η πηγή φωτός(S), η κάμερα και το σημείο υπολογισμού της φωτεινότητας(P):



Τέλος, ο υπολογισμός του επιστρεφόμενου τελικού χρώματος του σημείου προκύπτει από τον υπολογισμό της τελικής έντασης με το διάνυσμα χρώματος αυτού

2.2 Συνάρτηση calculate_normals

Η συνάρτηση αυτή υπολογίζει τα κανονικά διανύσματα των κορυφών ενός τρισδιάστατου μοντέλου με βάση τις συντεταγμένες των κορυφών και τις όψεις του μοντέλου. Για να το πετύχει λοιπόν αυτό, για κάθε όψη του μοντέλου (τρίγωνο του πίνακα faces) υπολογίζει τις συντεταγμένες των κορυφών του τριγώνου από τον πίνακα verts και στην συνέχεια τις ακμές αυτού. Το κανονικό διάνυσμα προκύπτει πολύ εύκολα χρησιμοποιώντας το εξωτερικό γινόμενο των ακμών. Τα αποτελέσματα προστίθεται στο συνεχώς ανανεώσιμο άθροισμα υπολογισμού του κανονικού διανύσματος κάθε κορυφής του τριγώνου. Τέλος, πριν την επιστροφή τους, για κάθε κορυφή, το κανονικό διάνυσμα κανονικοποιείται, ώστε να έχει μήκος 1. Αυτό γίνεται διαιρώντας το με τη νόρμα του, εφόσον η νόρμα δεν είναι μηδέν.

2.3 Συνάρτηση render_object

Η συνάρτηση αυτή έχει ως σκοπό την απόδοση ενός τρισδιάστατου αντικειμένου, λαμβάνοντας υπόψη διάφορες παραμέτρους φωτισμού και σκίασης. Τα βήματα που ακολουθεί είναι τα παρακάτω:

1. Υπολογίζει τα κανονικά διανύσματα των κορυφών του αντικειμένου μέσω της συνάρτησης calculate_normals.
2. Ρυθμίζει την κάμερα κατάλληλα χρησιμοποιώντας την συνάρτηση lookat για να δημιουργήσει τον πίνακα περιστροφής και το διάνυσμα μετάθεσης της κάμερας τα οποία είναι απαραίτητα ορίσματα της συνάρτησης prespective_project, η οποία προβάλλει τις κορυφές των τριγώνων σε δύο διαστάσεις. Τέλος, Τα προβολόμενα σημεία μετατρέπονται σε pixels με τη συνάρτηση rasterize. Να σχολιαστεί ότι αυτές οι 3 παραπάνω συναρτήσεις έχουν υλοποιηθεί στο προηγούμενο παραδοτέο της εργασίας Έργασία 2: Μετασχηματισμοί και Προβολές και εμπεριέχονται στο αρχείο functions_part2.py.

3. Δημιουργεί μια εικόνα με το δοσμένο χρώμα φόντου
4. Υπολογίζεται το μέσο βάθος κάθε όψης και οι όψεις ταξινομούνται με φθίνουσα σειρά βάθους για σωστή απόδοση, όπως γίνεται και στα προηγούμενα παραδοτέα εργασιών.
5. Ο χρωματισμός των τριγώνων γίνεται μέσω ενός for loop, στο οποίο βρίσκονται και αποθηκεύονται σε πίνακες οι γωνίες, τα χρώματα τους αλλά και τα κανονικά διανύσματα αυτών του τριγώνου που επρόκειτο να χρωματιστεί. Σημαντικό είναι να σχολιαστεί ότι αποθηκεύεται και οι συντεταγμένες των κορυφών πριν μετατραπούν σε συντεταγμένες pixel, οι οποίες θα χρησιμοποιηθούν στην συνέχεια για τον ακριβέστερο υπολογισμό του κέντρου βάρους του τριγώνου. Τέλος, ανάλογα με τον επιλεγμένο τύπο σκίασης (**gouraud**, **phong**, ή **phong** με επιλογή χρώματος με βάση **tex_ture_map**), καλείται η αντίστοιχη συνάρτηση σκίασης για την απόδοση του τριγώνου στην εικόνα. Οι συναρτήσεις που καλούνται για την επίτευξη αυτού θα αναλυθούν παρακάτω. Η επιλογή χρώματος με βάση **tex_ture_map**, δηλαδή με **texture_map**, αφορά το έξτρα παραδοτέο το οποίο αναλύεται στην επόμενη ενότητα.
6. Η συνάρτηση επιστρέφει την τελική εικόνα που περιέχει το αποδομένο αντικείμενο.

2.4 Συνάρτηση shade_gouraud

Η συνάρτηση αυτή υπολογίζει το χρώμα στις κορυφές του δοθέντος τριγώνου με βάση το πλήρες μοντέλο φωτισμού και συγκεκριμένα με την βοήθεια της συνάρτησης **light** που αναλύθηκε παραπάνω και στη συνέχεια χρησιμοποιεί γραμμική παρεμβολή χρώματος για την εύρεση του χρώματος στα εσωτερικά σημεία του τριγώνου, με την βοήθεια της συνάρτησης **g_shading** από το πρώτο παραδοτέο εργασίας. Για την υλοποίηση αυτής της εφαρμογής σκίασης, δεν χρειάστηκε κάποια περαιτέρω αλλαγή στην συνάρτηση **g_shading**.

2.5 Συνάρτηση shade_phong

Η συνάρτηση αυτή υπολογίζει το χρώμα των σημείων του τριγώνου πραγματοποιώντας παρεμβολή τόσο στα κανονικά διανύσματα όσο και στα χρώματα των κορυφών. Για την υλοποίηση της σκίασης Phong, θα μπορούσε να δημιουργηθεί μια ξεχωριστή συνάρτηση για την επεξεργασία της. Ωστόσο, δεδομένου ότι η πλειοψηφία της διαδικασίας σκίασης παραμένει κοινή μεταξύ των τεχνικών, επιλέχθηκε να επεκταθεί η υπάρχουσα συνάρτηση **g_shading** για να καλύπτει και τη σκίαση Phong. Αυτή η προσέγγιση μειώνει την επανάληψη του κώδικα και διευκολύνει τη συντήρηση. Άρα, απλά εσωτερικά της **shade_phong** καλείται απλώς η **g_shading**, με τα κατάλληλα ορίσματα. Οι αλλαγές που πραγματοποιήθηκαν, πέρα από την προσθήκη κατάλληλων ορισμάτων εισόδου, βρίσκονται εσωτερικά του for loop χρωματισμού του κάθε pixel του τριγώνου. Επιλέγοντας τον τύπο της σκίασης μέσω μίας συνθήκης if, ακολουθούνται τα παρακάτω βήματα:

- Υπολογίζεται το κανονικό διάνυσμα του pixel με τη συνάρτηση **pixel_color**. Η συνάρτηση **pixel_color** είχε δημιουργηθεί αρχικά για τον υπολογισμό χρώματος ενός pixel με χρήση γραμμικής παρεμβολής. Παρόλα αυτά, χωρίς καμία απολύτως αλλαγή βοηθάει και στον υπολογισμό των κανονικών διανυσμάτων.
- Υπολογίζεται το χρώμα του pixel με την ίδια συνάρτηση.
- Καλείται η συνάρτηση **light** για να υπολογιστεί το τελικό χρώμα του pixel με βάση το κανονικό διάνυσμα, το αρχικό χρώμα και τις παραμέτρους φωτισμού.
- Το αποτέλεσμα αποθηκεύεται στην εικόνα **img**.

Listing 1: Επιλογή σκίασης Phong εσωτερικά της **g_shading**

```

1 if option=="phong":
2     normal=pixel_color(y, allEdges, active_edges, vertices, vertsn, x)
3     color = pixel_color(y, allEdges, active_edges, vertices, vcolors, x)
4     final_color = light(bcoords, normal, color, campos, ka, kd, ks, n, lpos, lint,
5         lamb)
6     img[y][x] = final_color

```

3 Έξτρα παραδοτέο - Επιλογή χρώματος με χάρτη αφής

Στην παρούσα εργασία, υλοποιήθηκε και ο χρωματισμός αντικειμένου με βάση χάρτη αφής (texture map). Για την επίτευξη αυτού, υλοποιήθηκε η συνάρτηση bilerp, η οποία επιστρέφει το χρώμα που αντιστοιχεί στις δοθείσες 2D συντ/νες uv μέσω διγραμμικής παρεμβολής στην εικόνα εισόδου texture_map. Επίσης, ενσωματώθηκε ως επιλογή τύπου σκίασης στις συναρτήσεις render_object και στην g_shading.

3.1 Συνάρτηση bilerp

Στην συνάρτηση αυτή υπολογίζονται οι συντεταγμένες pixel που αντιστοιχούν στις συντεταγμένες UV, λαμβάνοντας υπόψη τις διαστάσεις του χάρτη υφής. Στην συνέχεια, υπολογίζονται οι συντεταγμένες των γειτονικών pixels και έτσι το αντίστοιχοις τιμές τους στο texture_map. Το τελικό χρώμα προκύπτει από τον μέσο όρο των 4 τιμών που προκύπτουν.

3.2 Επέκταση και προσαρμογή των συναρτήσεων render_object και g_shading

3.2.1 Συνάρτηση render_object

Όπως αναφέρθηκε και στην ανάλυση της συνάρτησης αυτής στην ενότητα 2.3, η επιλογή τύπου σκίασης με phong με επιλογή χρώματος με βάση texture_map ενσωματώνονται ως επιλογή όπως και οι άλλες δύο. Βέβαια για αυτή την περίπτωση να σχολιαστεί ότι γίνεται μια φθηνούσα ταξινόμηση των συντεταγμένων uv με βάση το βάθος των τριγώνων, όπως έγινε και για τα τρίγωνα. Επίσης, το texture_map, το οποίο είχε εισέλθει ως εικόνα στην συνάρτηση, μετατρέπεται σε πίνακας και διαιρείται με το 255 έτσι ώστε να μετατραπούν οι τιμές από το εύρος [0,255] στο εύρος [0,1], το οποίο χρησιμοποιείται στην βιβλιοθήκη matplotlib.pyplot με την οποία θα απεικονισθούν αργότερα τα αποτελέσματα.

3.2.2 Συνάρτηση g_shading

Όπως και στην περίπτωση τη σκίασης phong, η γενική συνάρτηση g_shading προσαρμόστηκε έτσι ώστε να καλύπτει και αυτή την περίπτωση. Πάλι οι αλλαγές περιορίζονται στο εσωτερικού του for loop χρωματισμού pixel. Συγκεκριμένα, αναφορικά με την περίπτωση phong με επιλογή χρώματος με βάση texture_map ακολουθούνται τα εξής βήματα:

- Υπολογίζεται το κανονικό διάνυσμα του συγκεκριμένου pixel με την βοήθεια της pixel_color.
- Υπολογίζονται οι συντεταγμένες UV, χρησιμοποιώντας την ίδια συνάρτηση.
- Υπολογίζεται το χρώμα pixel που αντιστοιχεί στο texture_map του με την βοήθεια της bilerp, δηλαδή με διγραμμική παρεμβολή
- Το παραπάνω χρώμα χρησιμοποιείται ως όρισμα στην συνάρτηση light για την εύρεση του τελικού χρώματος.
- Το αποτέλεσμα αποθηκεύεται στην εικόνα img.

Listing 2: Επιλογή σκίασης phong με επιλογή χρώματος με βάση texture_map εσωτερικά της g_shading

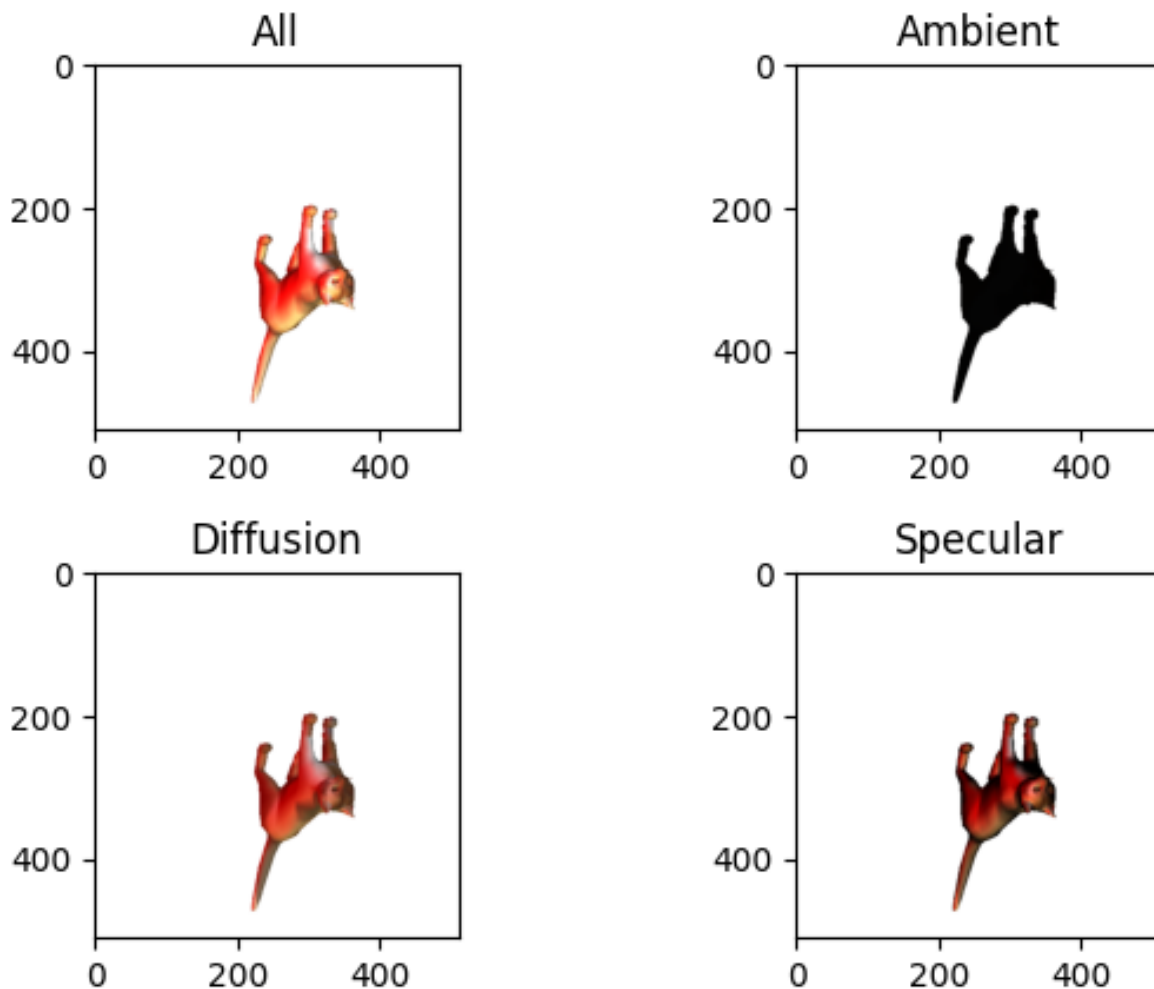
```
1 elif option=="texture_map":
2     normal=pixel_color(y, allEdges, active_edges, vertices, vertsn, x)
3     uv=pixel_color(y, allEdges, active_edges, vertices, uv_triangle, x)
4     color = bilerp(uv, texture_map)
5     final_color = light(bcoords, normal, color, campos, ka, kd, ks, n, lpos, lint,
6                        lamb)
7     img[y][x] = final_color
```

4 Scripts και αποτελέσματα

Για την επαλήθευση της σωστής λειτουργίας των κωδίκων, δημιουργήθηκε το αρχείο `demo.py`, στο οποίο αφού γίνει η φόρτωση των απαιτούμενων δεδομένων από το αρχείο `h3.npy`, καλεί την συνάρτηση `render_object` για να παραχτούν οι φωτογραφίες του αντικειμένου όπου ζητούνται και τυπώνονται στην συνέχεια ενσωματωμένες σε subplots:

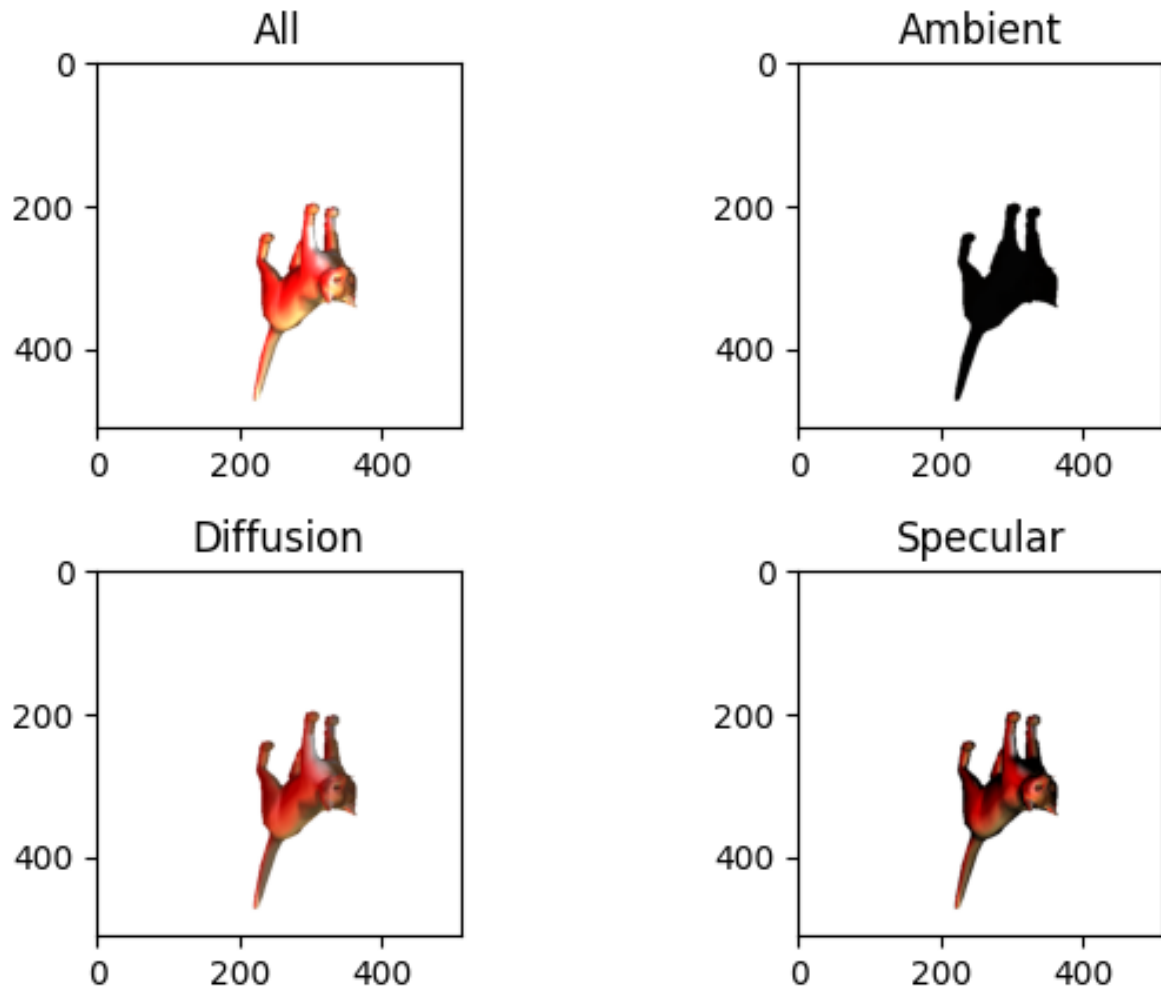
- Για τύπο σκίασης Gouraud με χρήση φωτισμού ambient, diffusion, specular και όλων μαζί αντίστοιχα:

Gouraud Shading Techniques



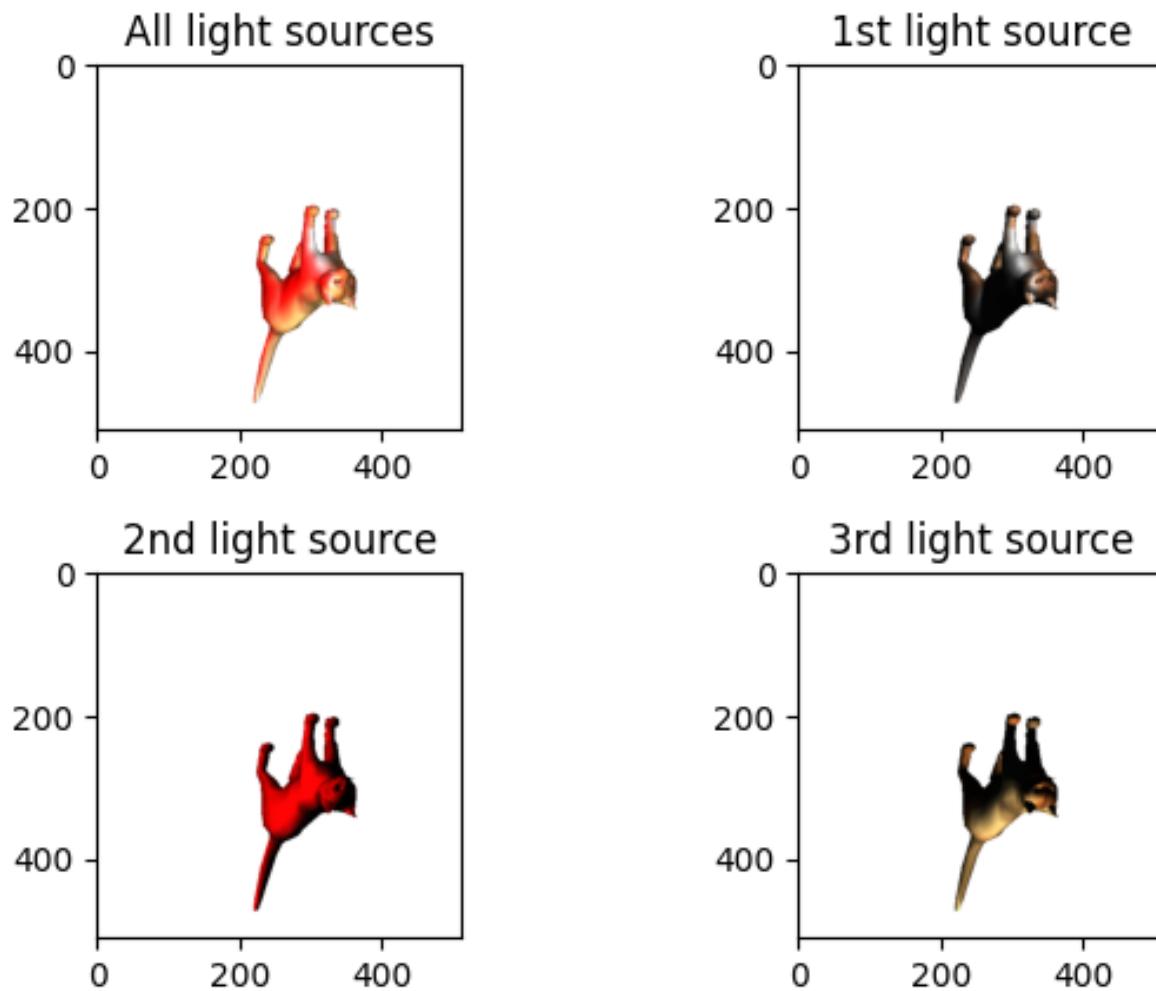
- Για τύπο σκίασης Phong με χρήση φωτισμού ambient, diffusion, specular και όλων μαζί αντίστοιχα:

Phong Shading Techniques



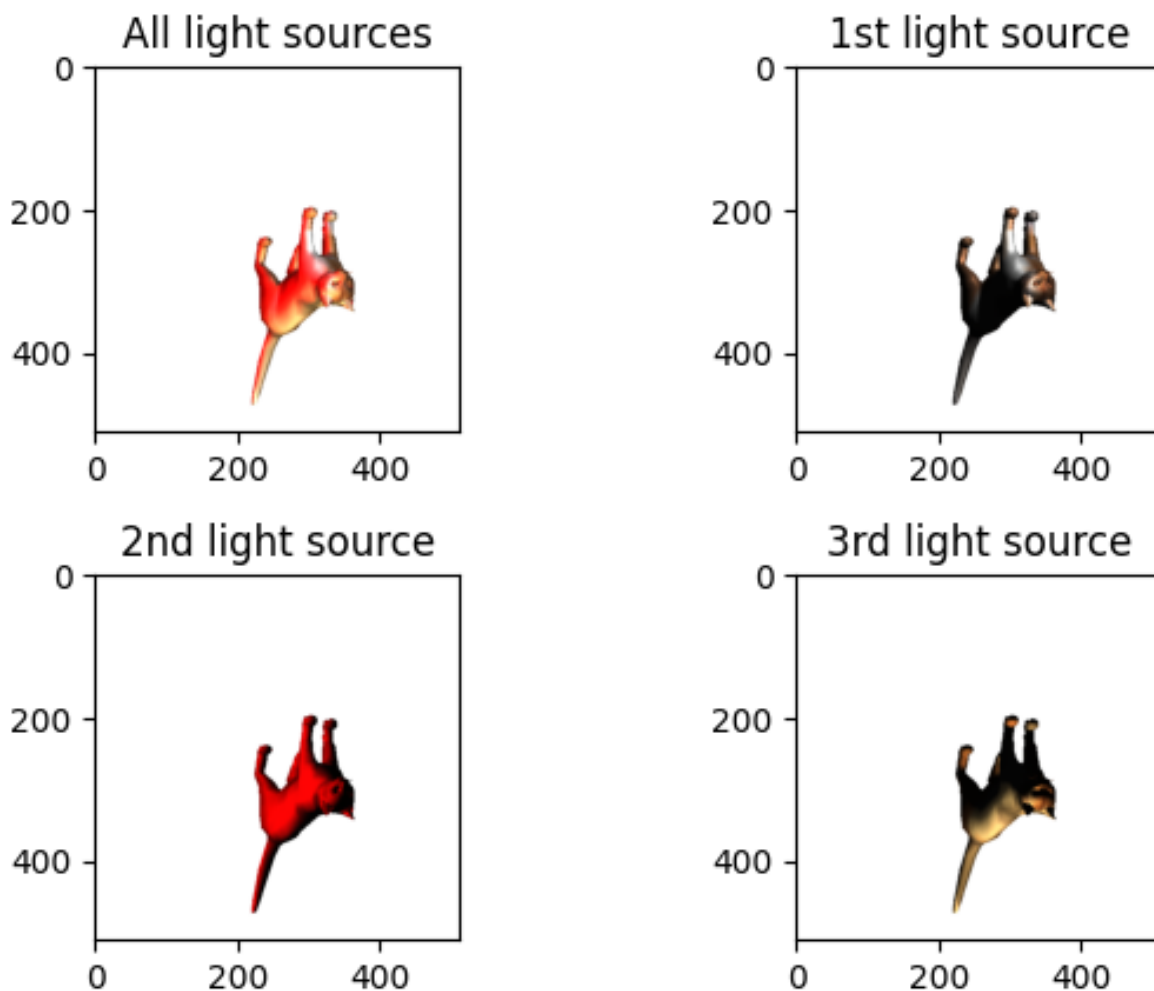
- Για τύπο σκίασης Gouraud με την 1η, 2η, 3η πηγή φωτισμού και όλες μαζί αντίστοιχα:

Gouraud All Shading Techniques with different light sources



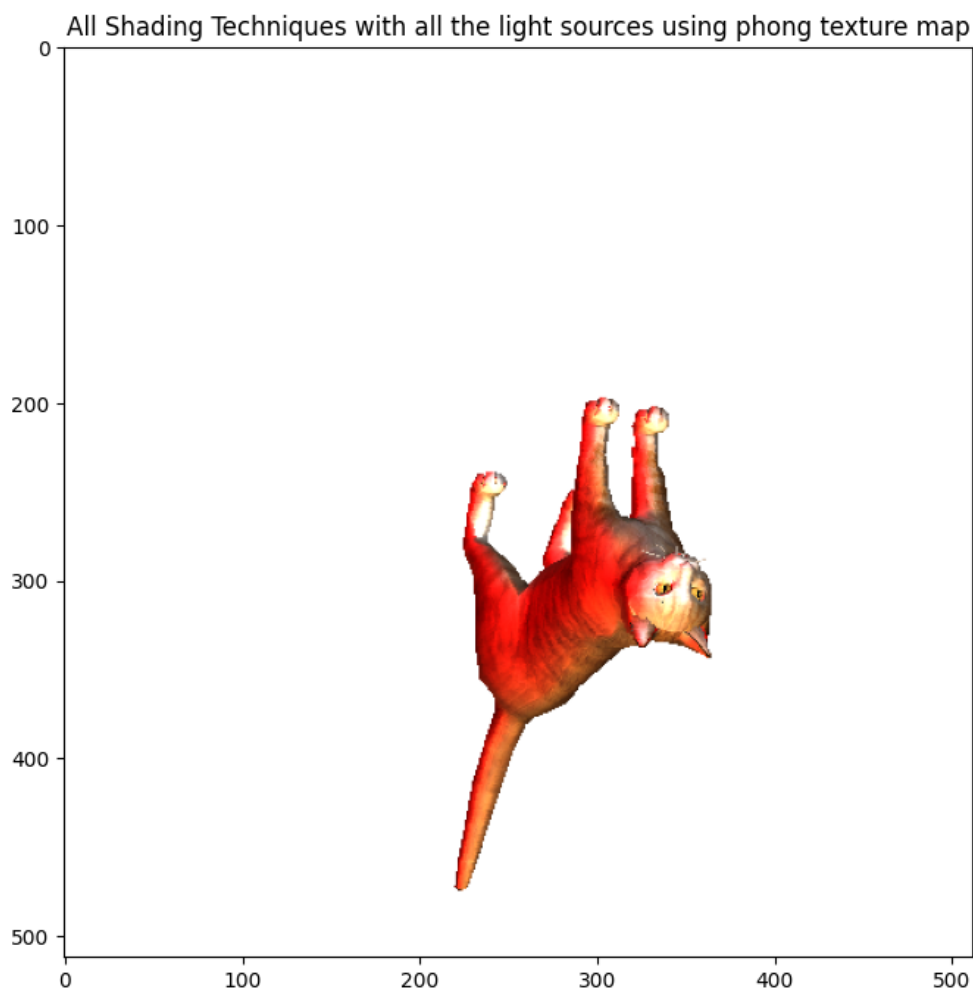
- Για τύπο σκίασης Phong με την 1η, 2η, 3η πηγή φωτισμού και όλες μαζί αντίστοιχα:

Phong All Shading Techniques with different light sources



Όπως είναι φανερό και από τις παραπάνω εικόνες, η διαφορά του τύπου σκίασης Gouraud και Phong είναι αναισιώστη. Αυτό είναι λογικό καθώς τα τρίγωνα του αντικειμένου είναι τόσο μικρά που η διαφορά υπολογισμού του χρώματος του κάθε pixel δεν οδηγεί σε αρκετά διαφορετικό αποτέλεσμα. Παρόλα αυτά η σκίαση τύπου phong σίγουρα οδηγεί σε πιο ρεαλιστικά αποτελέσματα καθώς ο φωτισμός υπολογίζεται σε κάθε πιξελ. Παρόλα αυτά, σίγουρα είναι πιο αργή μέθοδο για τον ίδιο λόγο ακριβώς. Επίσης, όταν το αντικείμενο φωτίζεται από τις 3 διαφορετικές πηγές φωτός, οι διαφορές είναι αρκετά έντονες. Από τα χρώμα του αντικείμενου γίνεται κατανοητό ότι η πρώτη πηγή φωτός είναι άσπρη, η δεύτερη κόκκινη και η τρίτη κίτρινη.

Σχετικά με την περίπτωση τύπου σκίασης phong με επιλογή χρώματος με βάση texture_map, , παρουσιάζεται παρακάτω ένα παράδειγμα αυτής με χρήση όλων των φωτισμών και φωτεινών πηγών. Κάθε άλλη υπο περίπτωση μπορεί να δοκιμαστεί όπως έγινε και παραπάνω, αλλάζοντας τα ορίσματα της render_object:



Όπως είναι φανερό, ο συνδυασμός της διγραμμικής παρεμβολής για την εξαγωγή χρώματος από τον χάρτη υφής και της σκίασης τύπου Phong οδηγεί σε πολύ ρεαλιστικά αποτελέσματα, κάνοντας το αντικείμενο να φαίνεται πιο φυσικό και λεπτομερές.