



Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Ηλεκτρονικής και Υπολογιστών

Εργασία 3 στην Γραφική με Υπολογιστές

Εργασία του
Φώτη Αλεξανδρίδη, ΑΕΜ: 9953
faalexandr@ece.auth.gr

5 Ιουλίου 2023

Περιεχόμενα

1 Overview	2
1.1 Στόχος	2
1.2 Υλοποίηση	2
2 Code	3
2.1 Γενική οργάνωση κώδικα	3
2.2 Το αρχείο <code>illumination.py</code>	3
2.3 Το αρχείο <code>shading.py</code>	4
2.4 Το αρχείο <code>demo.py</code>	5
3 Αποτελέσματα	6

Κεφάλαιο 1

Overview

1.1 Στόχος

Ο στόχος της εργασίας είναι η φωτογράφιση αντικειμένου στον χώρο με την ύπαρξη φωτισμού. Τα δεδομένα μας δίνονται και μας ζητείται η αναπαράστασή τους με την χρήση διαφορετικών μοντέλων φωτισμού και χρωματισμού.

Τα σημεία μας δίνονται από αρχείο δεδομένων που δίνεται με την εκφώνηση της άσκησης [1], με το τελικό σχήμα να είναι ένας σκύλος. Για την φωτογράφιση του αντικειμένου χρησιμοποιήθηκαν οι συναρτήσεις προοπτικής προβολής σε κάμερα της εργασίας 2. Για τον χρωματισμό εκδοχής της τελικής εικόνας, χρησιμοποιείται η μέθοδος χρωματισμού Gouraud που είχε αναπτυχθεί στα πλαίσια της πρώτης εργασίας. Μιας και τόσο η πρώτη όσο και η δεύτερη εργασία υλοποιήθηκε χωρίς κάποια απώλεια, η εισαγωγή των τμημάτων της υλοποίησής τους σε αυτή την εργασία δεν προκάλεσε ουδεμία καθυστέρηση στην ανάπτυξή της.

1.2 Υλοποίηση

Έχουν αναπτυχθεί οι 5 ζητούμενες από την εκφώνηση συναρτήσεις και οι 2 κλάσεις, χωρισμένες σε 2 αρχεία πηγαίου κώδικα python. Επιπρόσθετα έχει αναπτυχθεί ένα τρίτο αρχείο-πρόγραμμα για την επίδειξη της λειτουργικότητας του συστήματος.

Χρησιμοποιούνται επίσης τρία αρχεία από την εργασία 2, το αρχείο `helpers.py`, το αρχείο `fill_triangles.py` και το αρχείο `camera_functions.py`, ακριβώς όπως χρησιμοποιήθηκαν στην δεύτερη εργασία. Μιας και η λειτουργία τους έχει αναλυθεί εκτενώς στην αναφορά της δεύτερης εργασίας, δεν θα γίνει η επεξήγησή τους ξανά στο κομμάτι επεξήγησης του κώδικα.

Κεφάλαιο 2

Code

2.1 Γενική οργάνωση κώδικα

Όπως αναφέρθηκε, οι συναρτήσεις που μας ζητούνται έχουν υλοποιηθεί σε 2 αρχεία, τα `illumination.py` και `shading.py`. Για την επίδειξη της λειτουργικότητας, ένα τρίτο αρχείο, το `demo.py` διαβάζει δεδομένα από το παρεχόμενο `h3.npy` και χρησιμοποιεί τον κώδικα που αναπτύχθηκε για να καταγράψει οχτώ έγχρωμες RGB εικόνα από την προοπτική προβολή των σημείων του χώρου από ένα συγκεκριμένο σημείο προς ένα άλλο, με τα διαφορετικά μοντέλα φωτισμού και χρωματισμού που ζητούνται.

Ακόμη, από την δεύτερη εργασία χρησιμοποιούνται οι συναρτήσεις: `camera_looking_at` και `rasterize` του αρχείου `camera_functions.py`, για την φωτογράφιση των σημείων από τον τρισδιάστατο χώρο σε μια ψηφιακή εικόνα. Η λειτουργία τους έχει εξηγηθεί ενδελεχώς στην αναφορά της δεύτερης εργασίας και έτσι εδώ δεν θα αναλυθεί καθόλου.

2.2 Το αρχείο `illumination.py`

Το αρχείο αυτό περιέχει τις δύο ζητούμενες κλάσεις υλικού και σημειακής φωτεινής πηγής που ζητούνται, καθώς και την συνάρτηση φωτισμού.

Αρχικά, υλοποιούμε τις κλάσεις `PhongMaterial` και `PointLight`, όπως αυτές μας ζητούνται από την εκφώνηση [1]. Πρόκειται για υπεραπλουστευμένες κλάσεις που απλά διατηρούν κάποιες μεταβλητές οργανωμένες, χωρίς να προσφέρουν κάποια λειτουργικότητα.

Ακόμη, υλοποιούμε την συνάρτηση `light`:

```
def light(point, normal, vcolor, cam_pos, mat,
lights, light_amb):
```

η οποία είναι υπεύθυνη για τον υπολογισμό της τελικής τιμής φωτεινής ακτινοβολίας στο σημείο `point` που έχει κανονικό διάνυσμα `normal`, χρώμα `vcolor`, η θέση του παρατηρητή είναι `cam_pos`, το υλικό είναι `mat`, οι φωτεινές πηγές δίνονται στον πίνακα `lights` και η ένταση του ambient φωτός στην μεταβλητή `light_amb`.

Η συνάρτηση πρώτα υπολογίζει για κάθε φωτεινή πηγή τον φωτισμό λόγω diffusion και λόγω μοντέλου Phong, πολλαπλασιάζει την ακτινοβολία με τις τιμές του χρώματος του σημείου και τις αθροίζει. Τέλος, πραγματοποιεί την ίδια διαδικασία για το ambient φως, και επιστρέφει την τελική τριχρωματική ένταση φωτός που βλέπει ο παρατηρητής. Τα μοντέλα φωτισμού περιγράφονται μαθηματικά εκτενώς στις σημειώσεις [2].

2.3 Το αρχείο `shading.py`

Το αρχείο αυτό περιέχει τις τέσσερις υπόλοιπες ζητούμενες συναρτήσεις σχετικά με την εμφάνιση φωτογραφίας ενσωματώνοντας μοντέλα φωτισμού.

Αρχικά έχουμε την συνάρτηση `calculate_normals`:

```
def calculate_normals(verts, faces):
```

η οποία λαμβάνει το σύνολο των τρισδιάστατων σημείων στον χώρο μέσω της μεταβλητής `verts`, καθώς και τις τριάδες σημείων που αποτελούν τρίγωνα, και υπολογίζει το κανονικό διάνυσμα σε κάθε σημείο.

Για να το κάνει αυτό, πρώτα υπολογίζει το κανονικό διάνυσμα για κάθε τρίγωνο. Μιας και ένα σημείο μπορεί να ανήκει σε παραπάνω από ένα τρίγωνο, για εκείνα τα σημεία υπολογίζεται το διανυσματικό άθροισμα, και τέλος το τελικό διάνυσμα κανονικοποιείται ώστε να έχει μοναδιαίο μέτρο.

Έπειτα, υλοποιήθηκε η συνάρτηση `render_object`:

```
def render_object(shader, focal, eye, lookat,
up, bg_color, M, N, H, W, verts,
vert_colors, faces, mat,
lights, light_amb):
```

η οποία είναι αντίστοιχη της ομώνυμης συνάρτησης της δεύτερης εργασίας. Πρακτικά λαμβάνει όλες τις παραμέτρους που χρειάζονται για να δημιουργηθεί η έγχρωμη εικόνα με την φωτογράφιση των σημείων στον χώρο, και η μόνη διαφορά από την προηγούμενη εργασία είναι η ενσωμάτωση μοντέλων φωτισμού, με την χρήση των δύο ακόλουθων συναρτήσεων, ανάλογα με την μεταβλητή ελέγχου `shader`.

Ακόμη, υλοποιήθηκε η συνάρτηση `shade_gouraud`:

```
def shade_gouraud(vertsp, vertsn, vertsc, bcoords, cam_pos,
mat, lights, light_amb, X):
```

η οποία λαμβάνει τις δισδιάστατες συντεταγμένες των σημείων ενός τριγώνου στην εικόνα στην μεταβλητή `vertsp`, και οι υπόλοιπες παράμετροι αφορούν τον χρωματισμό με βάση το φωτεινό μοντέλο. Ακόμη, `X` είναι η ως τώρα εικόνα.

Για να το πετύχει αυτό, η συνάρτηση υπολογίζει με βάση το μοντέλο φωτισμού (τη συνάρτηση `light`) το χρώμα καθεμιάς κορυφής, και έπειτα χρησιμοποιεί την συνάρτηση `gourauds` για τον χρωματισμό του τριγώνου στην εικόνα.

Τέλος, υλοποιήθηκε η συνάρτηση `shade_phong`:

```
def shade_phong(vertsp, vertsn, vertsc, bcoords, cam_pos,
mat, lights, light_amb, X):
```

με ίδια ορίσματα με την `shade_gouraud`. Η συνάρτηση αυτή αποτελεί μια επέκταση της συνάρτησης χρωματισμού τριγώνου `gourauds` με την έννοια ότι αντί να βρίσκουμε μόνο το χρώμα κάθε σημείου με διπλή χρήση της συνάρτησης `interpolate_vectors` βρίσκουμε και το κανονικό διάνυσμα, και καλούμε το μοντέλο φωτισμού για κάθε σημείο, βρίσκοντας έτσι το ορθό χρώμα για κάθε εικονοστοιχείο στην οθόνη.

2.4 Το αρχείο `demo.py`

Το πρόγραμμα αυτό έχει ως σκοπό την επίδειξη των λειτουργιών που αναπτύχθηκαν στις υπόλοιπες συναρτήσεις με μια συγκεκριμένη διαδικασία που μας ζητείται από την εκφώνηση.

Αρχικά, διαβάζουμε όλα τα δεδομένα από το παρεχόμενο αρχείο `h3.npy`. Έπειτα, κατασκευάζουμε 4 αντικείμενα τύπου `PhongMaterial`: ένα με όλες τις σταθερές με τις πραγματικές τιμές τους και τα άλλα τρία όπου το καθένα έχει μόνο μία σταθερά διαδοχικά μη μηδενική (k_a , k_d και k_s). Έτσι αργότερα, μπορούμε να φωτογραφίσουμε εύκολα εικόνα μόνο με το εκάστοτε είδος φωτισμού χωρίς να κάνουμε αλλαγές κώδικα.

Έπειτα, κατασκευάζουμε τα αντικείμενα τύπου `PointLight` από τις παραμέτρους που δίνονται.

Τέλος, καλούμε την συνάρτηση `render_object` 8 φορές, πραγματοποιώντας όλους τους δυνατούς συνδυασμούς μεταξύ των δύο μεθόδων χρωματισμού "`gouraud`" και "`phong`", αποθηκεύοντας τις παραγόμενες εικόνες. Οι εικόνες περιέχουν το όνομα της μεθόδου χρωματισμού και έναν αριθμό από το 0 έως το 3 (0: όλες οι μέθοδοι φωτισμού, 1: μόνο `ambient` φωτισμός, 2: μόνο `diffused` φωτισμός, 3: μόνο κατοπτρικά ανακλόμενος φωτισμός).

Για την αποθήκευση των εικόνων χρησιμοποιείται η βιβλιοθήκη `matplotlib` χάριν απλότητας (θα μπορούσε να υλοποιηθεί η ίδια λογική αποθήκευσης όπως στις προηγούμενες 2 εργασίες χωρίς προβλήματα).

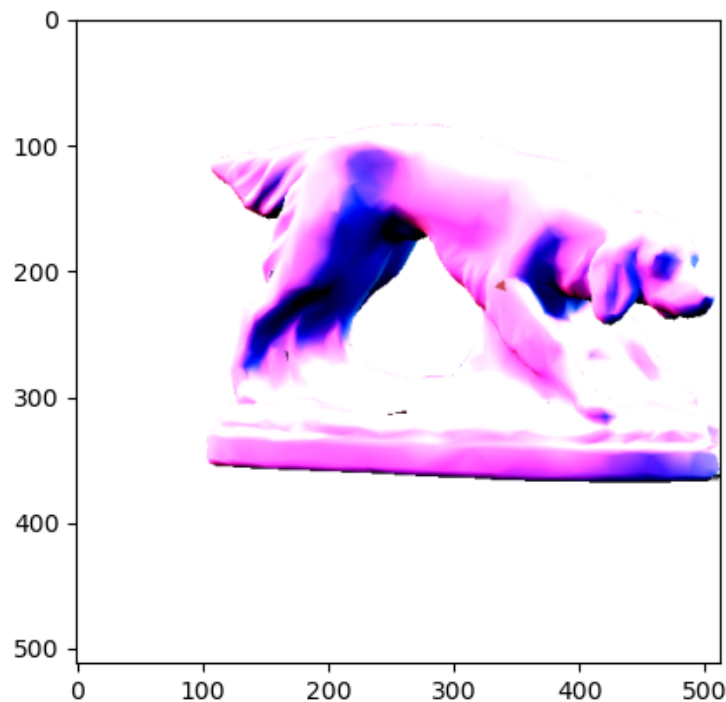
Κεφάλαιο 3

Αποτελέσματα

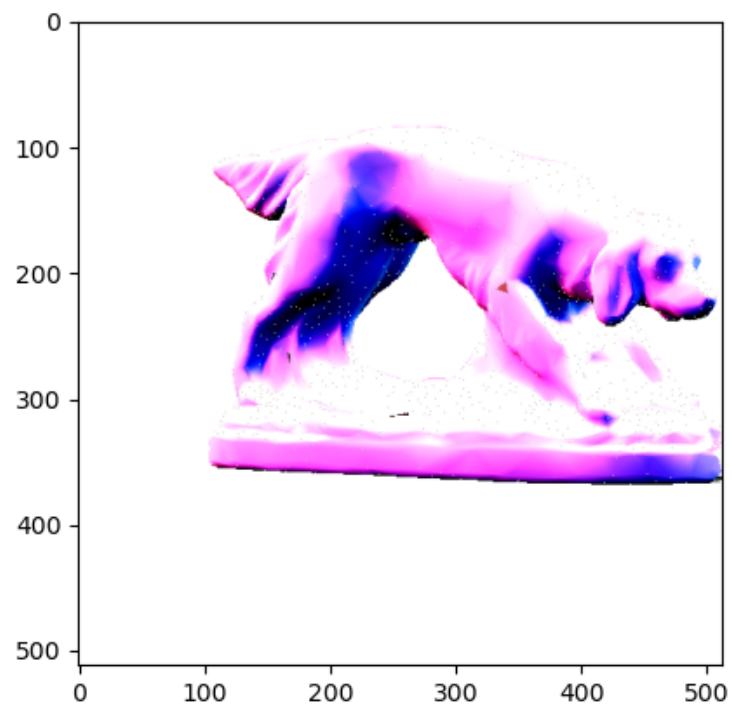
Οι παράμετροι, καθώς και τα διάφορα σημεία μας δίνονται στο αρχείο δεδομένων `h3.npy`.

Για να καταγράψουμε τις εικόνες με καθέναν από τους διαφορετικούς φωτισμούς που ζητούνται, κάναμε χρήση του παρακάτω τεχνάσματος: Δημιουργήσαμε 4 αντικείμενα τύπου `PhongMaterial`, ένα με όλες τις τιμές των δεικτών, έναν με μηδενικές παραμέτρους k_a, k_s , ένα με μηδενικές παραμέτρους k_a, k_s και ένα με μηδενικές παραμέτρους k_a, k_d . Έτσι καταφέραμε να απομονώσουμε το καθένα από τα τρία είδη φωτισμού, εξαλείφοντας τα άλλα δύο σε κάθε περίπτωση.

Οι εικόνες με τον συνολικό φωτισμό φαίνονται παρακάτω:

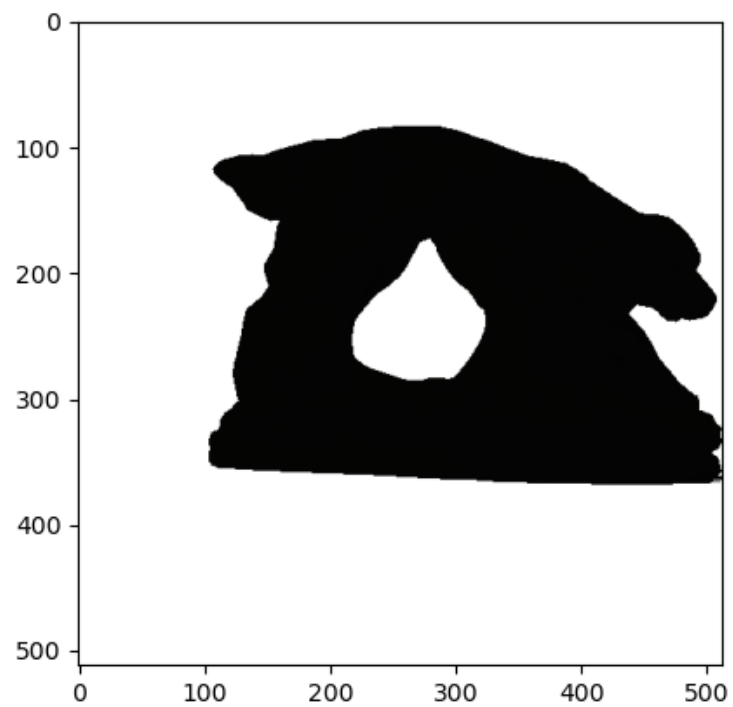


Σχήμα 3.1: Gouraud Shading

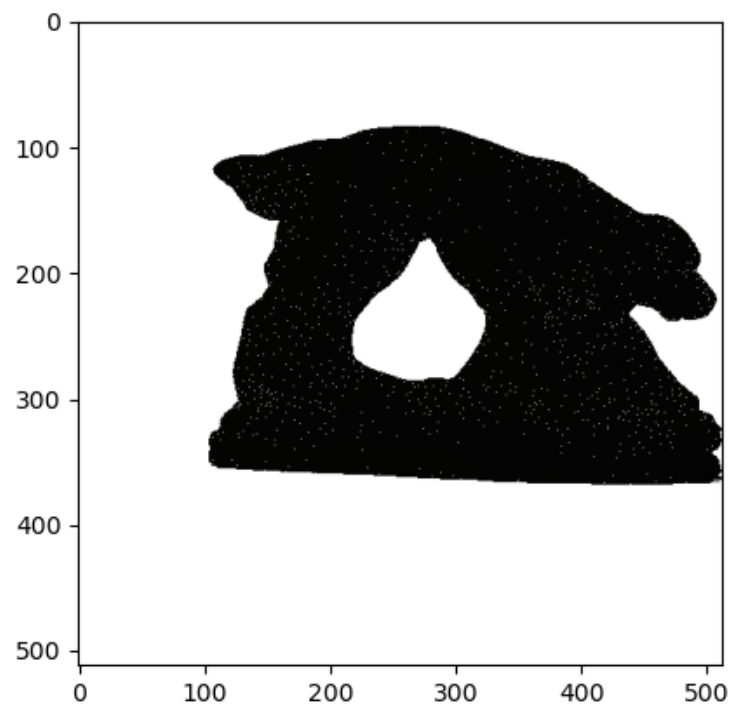


Σχήμα 3.2: Phong Shading

Οι εικόνες μονάχα με ambient φωτισμό:

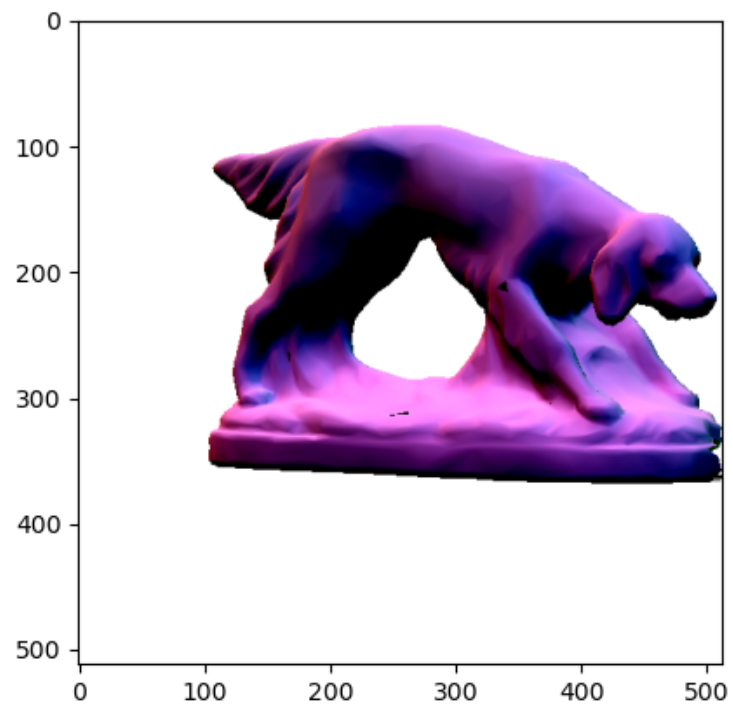


Σχήμα 3.3: Gouraud Shading

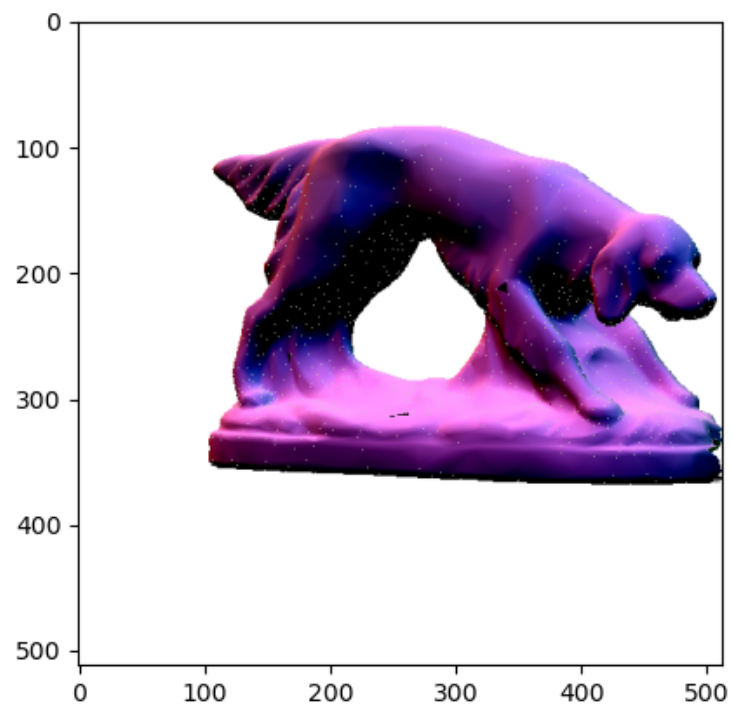


Σχήμα 3.4: Phong Shading

Οι εικόνες μονάχα με διάχυτο φωτισμό:

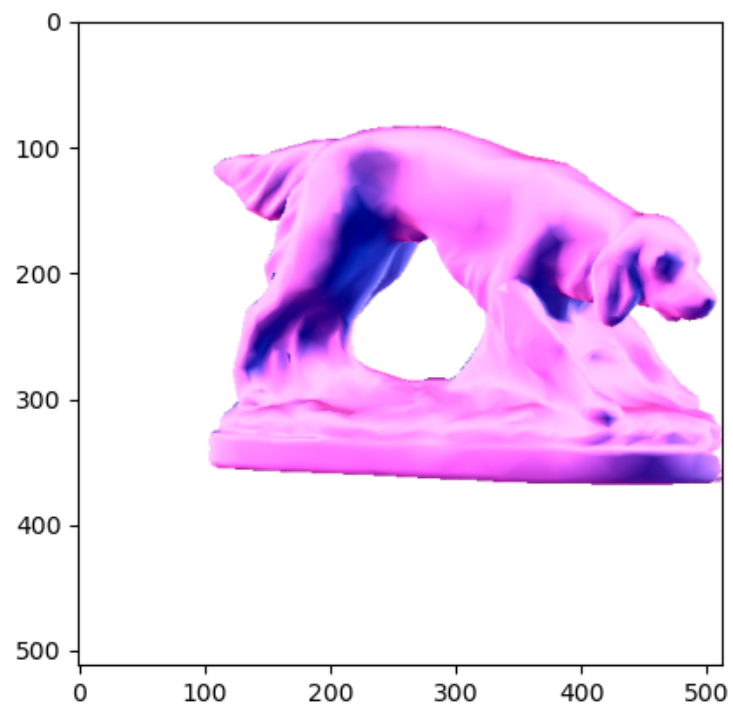


Σχήμα 3.5: Gouraud Shading

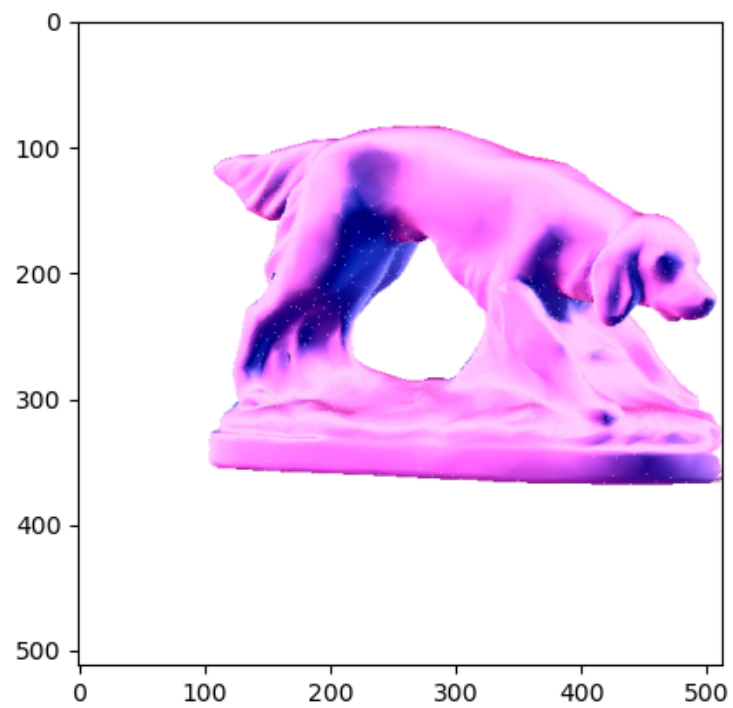


Σχήμα 3.6: Phong Shading

Οι εικόνες μονάχα με κατοπτρικό φωτισμό:



Σχήμα 3.7: Gouraud Shading



Σχήμα 3.8: Phong Shading

Bibliography

- [1] A. Delopoulos, *Computer Graphics assignment 3: vision*, 2023.
- [2] —, *Computer Graphics Notes*, 2009.