

# Γραφική με Υπολογιστές

## Εργασία 3: Θέαση

Φίλιππος Ρωσσίδης AEM:10379

13 Ιουνίου 2024

### 1 Εισαγωγή

Στην εργασία αυτή προσθέτουμε στις συναρτήσεις των προϋγούμενων εργασιών και τη διαδικασία φωτισμού, με συνδιασμό διάχυτου φωτός από το περιβάλλον (ambient light), διάχυτης ανάκλασης (diffuse reflexion) και κατοπρικής ανάκλασης (specular reflexion).

Θα χρησιμοποιηθεί το μοντέλο του Phong.

### 2 Η συνάρτηση *light*

Συνάρτηση που υπολογίζει το φωτισμό (διάνυσμα για rgb) ενός σημείου δεδομένων των συντεταγμένων του, των συντεταγμένων του κάθετου διανύσματος στην επιφάνειά του, του χρώματος του, των συντεταγμένων της τοποθεσίας της κάμερας, των συντελεστών ανάκλασης  $k_a, k_d, k_s$  του εκθέτη phong  $n$  και των τοποθεσιών και εντάσεων των σημειακών φωτεινών πηγών.

Ο τύπος του μοντέλου phong που χρησιμοποιείται είναι ο:

$$I_\lambda = I_{a\lambda}k_{a\lambda} + \sum_{k=1}^N \left[ I_{p\lambda}^k \left( k_{d\lambda} \left( \hat{N} \cdot \hat{L}^k \right) + k_{s\lambda} \left( \hat{R}^k \cdot \hat{V} \right)^n \right) \right] \quad (1)$$

Συγκεκριμένα για να υπολογιστούν οι συντελεστές ανάκλασης για κάθε μήκος κύματος (r,g,b) εφόσον δίνεται ένας γενικός συντελεστής ανάκλασης για κάθε μέθοδο και το χρώμα σε rgb, πολλαπλασιάζεται ο συντελεστής με το διάνυσμα του χρώματος ώστε να προκύψουν οι συντελεστές και καθένα από τα 3 μήκη κύματος, r,g,b. Δηλαδή:

$$\begin{bmatrix} k_{a\{r\}} \\ k_{a\{g\}} \\ k_{a\{b\}} \end{bmatrix} = k_a \cdot \begin{bmatrix} color_r \\ color_g \\ color_b \end{bmatrix}, \text{ ομοίως για } k_d, k_s$$

Στην εξίσωση (1) συμπεριλαμβάνεται ο διάχυτος φωτισμός  $I_{a\lambda}k_{a\lambda}$ , και για κάθε πηγή: ο φωτισμός διάχυτης ανάκλασης  $I_{p\lambda}^k k_{d\lambda} \left( \hat{N} \cdot \hat{L}^k \right)$  και ο φωτισμός κατοπρικής ανάκλασης:  $I_{p\lambda}^k k_{s\lambda} \left( \hat{R}^k \cdot \hat{V} \right)^n$ .

Το διάνυσμα  $\hat{L}$  είναι το μοναδιαίο διάνυσμα που δείχνει από το σημείο προς την φωτεινή πηγή. Υπολογίζεται ως η διαφορά του σημείου από τις συντεταγμένες της κάμερας, και διαιρώντας με το μέτρο του. Το διάνυσμα  $\hat{R}$  είναι το συμμετρικό του  $\hat{L}$  ως προς το  $\hat{N}$  (κάθετο διάνυσμα στην επιφάνεια) και υπολογίζεται ως  $2(\hat{N} \cdot \hat{L})\hat{N} - \hat{L}$ .

Στο κώδικα λοιπόν, υπολογίζονται τα διανύσματα και τα εσωτερικά γινόμενα  $\hat{N} \cdot \hat{L}^k$  και  $\hat{R}^k \cdot \hat{V}$  για κάθε πηγή, το δεύτερο ηψώνεται στην  $n$ , πολλαπλασιάζονται με το χρώμα και τους συντελεστές ανάκλασης, αν κάποιο από αυτά είναι αρνητικό, τότε το θέτουμε ίσο με μηδέν. Προστίθενται όλα στη μεταβλητή *Illumin* η οποία στο τέλος περνάει από clipping στο διάστημα  $[0,1]$  ώστε να έχει έγκυρες τιμές χρώματος και επιστρέφεται.

Αξίζει να σημειωθεί ότι στην παραπάνω υλοποίηση αγνοήθηκε η επίδραση της απόστασης του σημείου από την φωτεινή πηγή.

### 3 Η συνάρτηση *calculate\_normals(verts, faces)*

Για ένα τρίγωνο με κορυφές  $k_1, k_2, k_3$  το κάθετο διάνυσμα υπολογίζεται ως  $\hat{N} = (k_2 - k_1) \times (k_3 - k_1)$ . Αυτό είναι το κάθετο διάνυσμα του κέντρου του τριγώνου, για να βρούμε το κάθετο διάνυσμα για κάθε κορυφή του παίρνουμε το μέσο όρο αυτών των τριγώνων που την έχουν σαν κορυφή τους.

Έτσι, στη συνάρτηση, δημιουργούνται δύο πίνακες *normals\_verts* και *normals\_faces* μεγέθους  $3 \times N_v$  και  $3 \times N_T$ , όπου  $N_v$  το πλήθος των κορυφών και  $N_T$  το πλήθος των τριγώνων. Ο πρώτος πίνακας θα περιέχει τα κάθετα διανύσματα των κορυφών και ο δεύτερος αυτά των τριγώνων.

Για κάθε τρίγωνο στον πίνακα *faces* υπολογίζεται το κάθετο διάνυσμα στο κέντρο του, όπως περιγράφηκε παραπάνω, και έπειτα προστίθεται στις θέσεις του πίνακα *normals\_verts* που αντιστοιχούν στις κορυφές που το απαρτίζουν. Έτσι, αφού ολοκληρωθεί αυτή η διαδικασία για κάθε τρίγωνο, ο πίνακας *normals\_verts* θα περιέχει για κάθε κορυφή το άθροισμα των κάθετων διανυσμάτων των τριγώνων που την χρησιμοποιούν. Τέλος, στον πίνακα αυτόν διαιρούμε κάθε στοιχείο με το μέτρο του, έτσι ώστε τα κάθετα διανύσματα να έχουν μοναδιαίο μέτρο και επιστρέφονται.

### 4 Η συνάρτηση *shade\_gouraud*

Δέχεται σαν ορίσματα συντεταγμένες (σε pixel), κάθετα διανύσματα, χρώματα των τριών κορυφών που απαρτίζουν ένα τρίγωνο, τις 3d συντεταγμένες του κέντρου βάρους του τριγώνου, τη θέση της κάμερας, τους συντελεστές  $k_a, k_d, k_s, n$ , τις θέσεις και εντάσεις των φωτεινών πηγών και μια εικόνα με τυχών προϋπάρχοντα τρίγωνα.

Υπολογίζει το φωτισμό των τριών κορυφών χρησιμοποιώντας σαν θέση τους το κέντρο βάρους του τριγώνου, με τη συνάρτηση *light*, και έπειτα καλείται η συνάρτηση *g\_shading* από την πρώτη εργασία, για να σκιάσει το τρίγωνο με την τεχνική *gouraud shading*.

### 5 Η συνάρτηση *shade\_phong*

Η διαφοροποίηση της με την *shade\_gouraud* έγκειται στο ότι εδώ ο φωτισμός εφαρμόζεται σε κάθε σημείο σκίασης. Συγκεκριμένα χρησιμοποιήθηκε ο κώδικας από τη συνάρτηση *g\_shading* από την πρώτη εργασία, και στο σημείο όπου εφαρμόζεται η σκίαση προστέθηκαν κάποιες γραμμές.

Εφαρμόζεται διπλή γραμμική παρεμβολή για το χρώμα του σημείου (όπως και στην αρχική) αλλά εφαρμόζεται και διπλή γραμμική παρεμβολή για την εύρεση του κάθετου διανύσματος του εκάστοτε σημείου, χρησιμοποιώντας στην παρεμβολή τα κάθετα διανύσματα των τριών κορυφών. Χρησιμοποιώντας το χρώμα και το κάθετο διάνυσμα που βρέθηκε, υπολογίζεται ο φωτισμός με τη συνάρτηση *light* και το αποτέλεσμα τίθεται στη θέση  $[y, x]$  της εικόνας.

### 6 Η συνάρτηση *render\_object*

Είναι η συνάρτηση που θα αποτυπώσει ένα αντικείμενο σε εικόνα. Δέχεται σαν ορίσματα: την μέθοδο σκίασης (*shader* = "gouraud" ή "phong"), τα στοιχεία της κάμερας (*focal, eye, lookat, up*), το χρώμα του υποβάθρου (*bg\_color*), τις διαστάσεις του πλαισίου σε pixel (*M, N*), τις διαστάσεις της κάμερας στις μονάδες του συστήματος συντεταγμένων (*H, W*), τις συντεταγμένες, τα χρώματα και τα τρίγωνα που απαρτίζουν το αντικείμενο (*verts, vert\_colors, faces*), τους συντελεστές ανάκλασης και σταθερά *phong* ( $k_a, k_d, k_s, n$ ) και τις θέσεις και εντάσεις των φωτεινών πηγών  $l_{pos}, l_{int}, l_{amb}$ .

Αρχικά υπολογίζει τα κάθετα διανύσματα κάθε κορυφής με την *calculate\_normals*. Δημιουργεί έναν *numpy* πίνακα  $M \times N \times 3$  με άσσους και τον πολλαπλασιάζει με το *bg\_color*, αυτός είναι ο καμβάς της εικόνας όπου θα αποτυπωθούν τα τρίγωνα. Όπως στην προϋγούμενη εργασία και με τη χρήση των συναρτήσεων που υλοποιήθηκαν για αυτή προβάλλονται τα σημεία στο πέτασμα της κάμερας και όπως στις πρώτες εργασίες, ταξινομούνται τα τρίγωνα (*faces*) με σειρά από το πιο μακρινό στο πιο κοντινό στην κάμερα.

Για κάθε τρίγωνο με την ταξινομημένη σειρά, αν βρίσκεται μπροστά από την κάμερα και μέσα στο πλαίσιο που αυτή ορίζει, υπολογίζεται το κέντρο βάρους του και καλείται η εκάστοτε συνάρτηση (είτε

*shade\_gouraud* είτε *shade\_phong* ) ανάλογα με την μεταβλητή shader. Επιστρέφεται η εικόνα που δημιουργείται.

**Παρατήρηση:** Τα μοναδιαία διανυσματά της κάμερας με τα οποία υπολογίζονται οι συντεταγμένες των κορυφών σε pixel, υπολογίστηκαν στην δεύτερη εργασία έτσι ώστε οι γραμμές να απαριθμούνται από κάτω προς τα πάνω και οι στήλες από αριστερά προς τα δεξιά. Όμως η συνάρτηση *g\_shading* της πρώτης εργασίας γράφτηκε ώστε να θεωρεί ότι τα σημεία που της δίνονται απαριθμούνται από πάνω προς τα κάτω και από αριστερά προς τα δεξιά. Έτσι πριν δοθούν τα σημεία για σκίαση, αντιστρέφονται κατά *y* με την εξίσωση:

$$y_{new} = H - y_{old}$$

όπου *H* το ύψος σε pixel του πλαισίου.

## 7 Το αρχείο *demo.py*

Είναι το αρχείο επίδειξης της λειτουργίας των συναρτήσεων.

Φωτρώνονται τα δεδομένα από το αρχείο *h3.npy*. Καλούνται οι παραπάνω συναρτήσεις για τις παρακάτω περιπτώσεις.

*gouraud shading* για:

- μόνο ambient φωτισμό ( $k_d = k_s = 0$ )
- μόνο diffuse φωτισμο ( $k_a = k_s = 0$ )
- μόνο specular φωτισμό ( $k_a = k_d = 0$ )
- και τα 3 είδη φωτισμού

*phong shading* για

- μόνο ambient φωτισμό ( $k_d = k_s = 0$ )
- μόνο diffuse φωτισμο ( $k_a = k_s = 0$ )
- μόνο specular φωτισμό ( $k_a = k_d = 0$ )
- και τα 3 είδη φωτισμού

*phong shading* για

- μόνο την πρώτη φωτεινή πηγή
- μόνο την δεύτερη φωτεινή πηγή
- μόνο την τρίτη φωτεινή πηγή
- όλες τις φωτεινές πηγές

Δημιουργούνται επομένως 12 εικόνες. Τα αποτελέσματα φαίνονται στα σχήματα 1,2,3.

### 7.1 Σχολιασμός αποτελεσμάτων

- Επειδή η επιλογές του  $k_a = 0.1$  και  $I_a = [0.2, 0.2, 0.2]$  είναι αρκετά μικρές, βλέπουμε ότι στην περίπτωση μόνο διάχυτου φωτισμού από το περιβάλλον η γάτα παραμένει σχεδόν μαύρη.
- Στο διάχυτο φωτισμό η γάτα απορροφά "ομοιόμορφα" ή "επίπεδα" το φως, ως αναμενώταν και στο φωτισμό κατοπρικής ανάκλασης, αντανακλά το φως που πέφτει στις επιφάνειες. Επειδή η σταθερά *phong* που επιλέχθηκε  $n = 1$  δεν είναι μεγάλη, η ανάκλαση δεν είναι έντονη.

- Η διαφορά μεταξύ gouraud και phong σκίασης δεν είναι πολύ μεγάλη, στη συγκεκριμένο παράδειγμα, διότι τα τρίγωνα που απαρτίζουν τη γάτα είναι αρκετά μικρά και έτσι η παρεμβολή στο gouraud είναι αρκετά καλή προσέγγιση του πιο ρεαλιστικού μοντέλου phong. Μεγάλη διαφορά μεταξύ των δύο θα φανεί παρακάτω που χρησιμοποιείται και texture mapping.
- Οι τρεις φωτεινές πηγές έχουν διαφορετικά χρώματα και διαφορετικές τοποθεσίες και έτσι εξηγούνται τα διαφορετικά χρώματα στο τελικό αποτέλεσμα.

## 8 Έξτρα Παραδοτέο

Υλοποιούνται αλλαγές στις παραπάνω συναρτήσεις ώστε το χρώμα του αντικειμένου που σκιάζεται να επιλέγεται από ένα texture map.

Συγκεκριμένα δίνεται ένας πίνακας  $uvs\ 2 \times N_{verts}$  που δίνει για κάθε κορυφή του τριγώνου τις συντεταγμένες της στο texture map, και ο πίνακας  $face\_uv\_indices\ 3 \times N_{faces}$  που περιέχει για κάθε τρίγωνο, τους δείκτες των κορυφών του στον πίνακα uvs.

### 8.1 Η συνάρτηση bilerp

Θα χρησιμοποιηθεί για να βρεθεί το χρώμα ενός σημείου που του αντιστοιχούν συντεταγμένες  $uv$  στην εικόνα texture map με  $uv$  floats. Δεδομένου  $uv$  επιστρέφει το χρώμα:

$$\frac{texture\_map[x][y] + texture\_map[x][y + 1] + texture\_map[x + 1][y] + texture\_map[x + 1][y + 1]}{4}$$

όπου  $[x, y] = [uv]$

### 8.2 Η συνάρτηση shade\_phong\_w\_texture

Είναι υπεύθυνη για τη σκίαση ενός τριγώνου με την τεχνική phong και χρησιμοποιώντας το texture map.

Είναι παρόμοια με την *shade\_phong* που περιγράφηκε προϋγουμένως, μόνο που εκτός από παρεμβολή στα κάθετα διανύσματα, εφαρμόζει και διπλή παρεμβολή στα  $uvs$  των κορυφών του τριγώνου (προστίθενται οι όροι normals και uvs στη κλάση που περιγράφει τις ευθείες ώστε να χρησιμοποιηθεί ο αλγόριθμος για διπλή παρεμβολή της πρώτης εργασίας). Έτσι για κάθε σημείο  $[x, y]$  στο εσωτερικό του τριγώνου υπολογίζει το  $uv$  βρίσκει με τη *bilerp* το χρώμα που του αναλογεί από το texture map, το φωτίζει με τη συνάρτηση *light* με το κάθετο διάνυσμα που υπολογίστηκε με παρεμβολή και το χρωματίζει.

### 8.3 Η συνάρτηση render\_object\_w\_texture

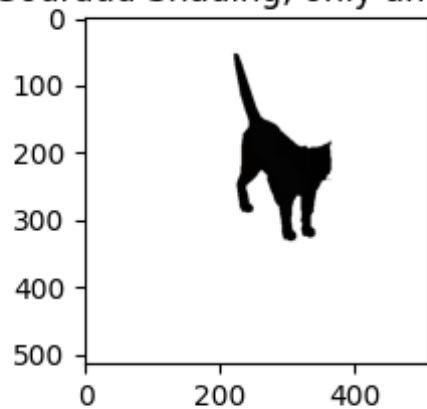
Είναι η συνάρτηση που θα εμφανίσει ένα αντικείμενο χρησιμοποιώντας το texture map. Είναι παρόμοια με την αντίστοιχη *render\_object* με τη μόνη διαφορά ότι ταξινομεί και τον πίνακα *faces\_uv\_sorted* από το πιο μακρινό τρίγωνο στο πιο κοντινό και για κάθε τρίγωνο βρίσκει και τις συντεταγμένες  $uv$  των κορυφών του στην εικόνα texture map.

Αν η μέθοδος σκίασης είναι gouraud υπολογίζει κατευθείαν τα χρώματα από τις τρεις κορυφές μέσω του texture map και σκιάζει με τη χρήση της *g\_shading*. Αποτέλεσμα αυτού είναι ότι η λεπτομέρεια του texture map στο εσωτερικό των τριγώνων χάνεται γιατί εφαρμόζεται παρεμβολή μεταξύ των τριών χρωμάτων των κορυφών.

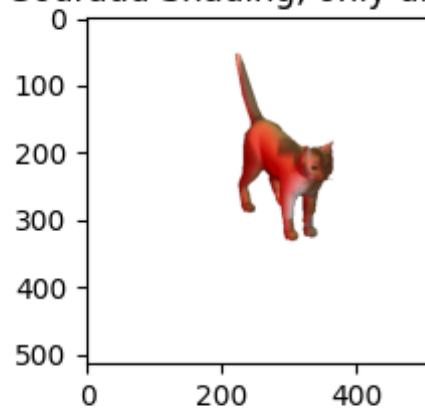
Αν η μέθοδος σκίασης είναι phong καλείται η συνάρτηση *shade\_phong\_w\_texture* και έτσι κάθε σημείο του τριγώνου παίρνει το χρώμα που του αντιστοιχεί από το texture map και φωτίζεται. Έτσι έχει πιο ρεαλιστικά αποτελέσματα.

Τα αποτελέσματα της χρήσης texture map φαίνονται στα σχήματα 4,5.

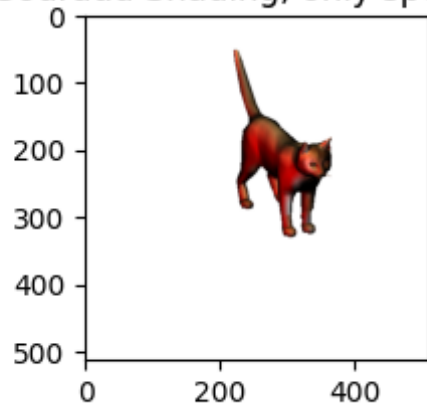
Gouraud Shading, only ambient



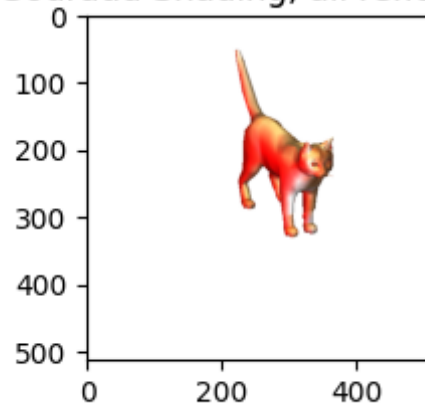
Gouraud Shading, only diffuse



Gouraud Shading, only specular

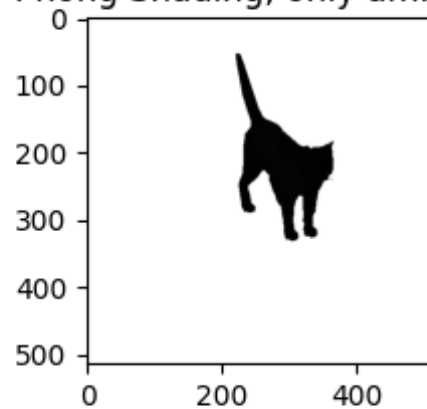


Gouraud Shading, all reflexions

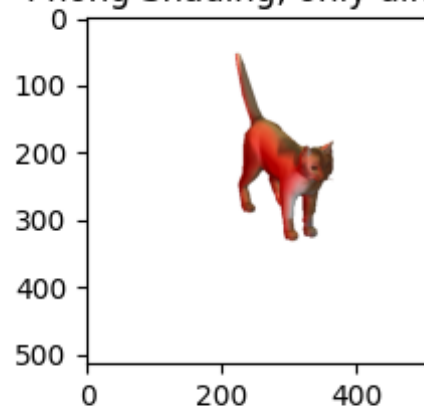


Σχήμα 1: Gouraud shading με διαφορετικούς τρόπου ανάκλασης

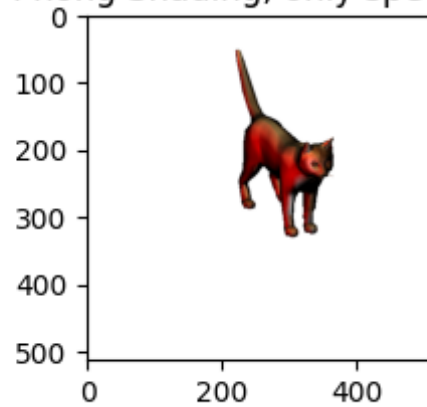
Phong Shading, only ambient



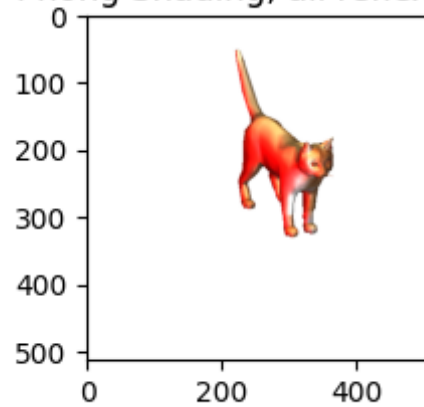
Phong Shading, only diffuse



Phong Shading, only specular

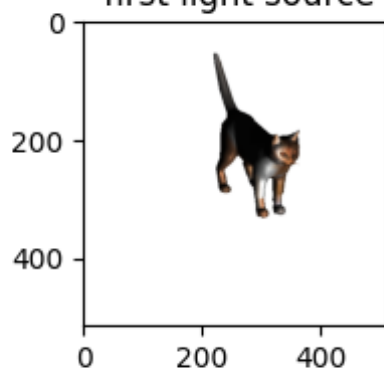


Phong Shading, all reflexions

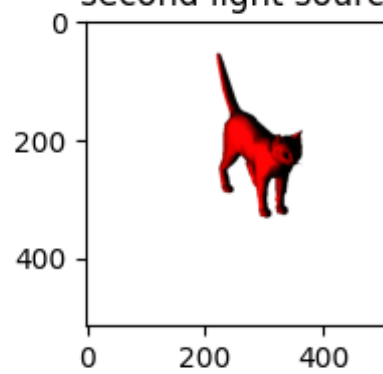


Σχήμα 2: Phong shading με διαφορετικούς τρόπου ανάκλασης

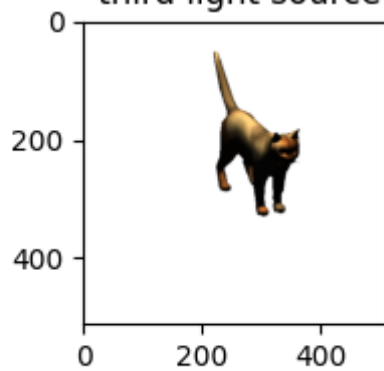
Phong Shading, all reflexions,  
first light source



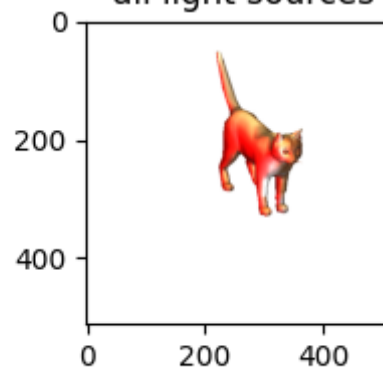
Phong Shading, all reflexions,  
second light source



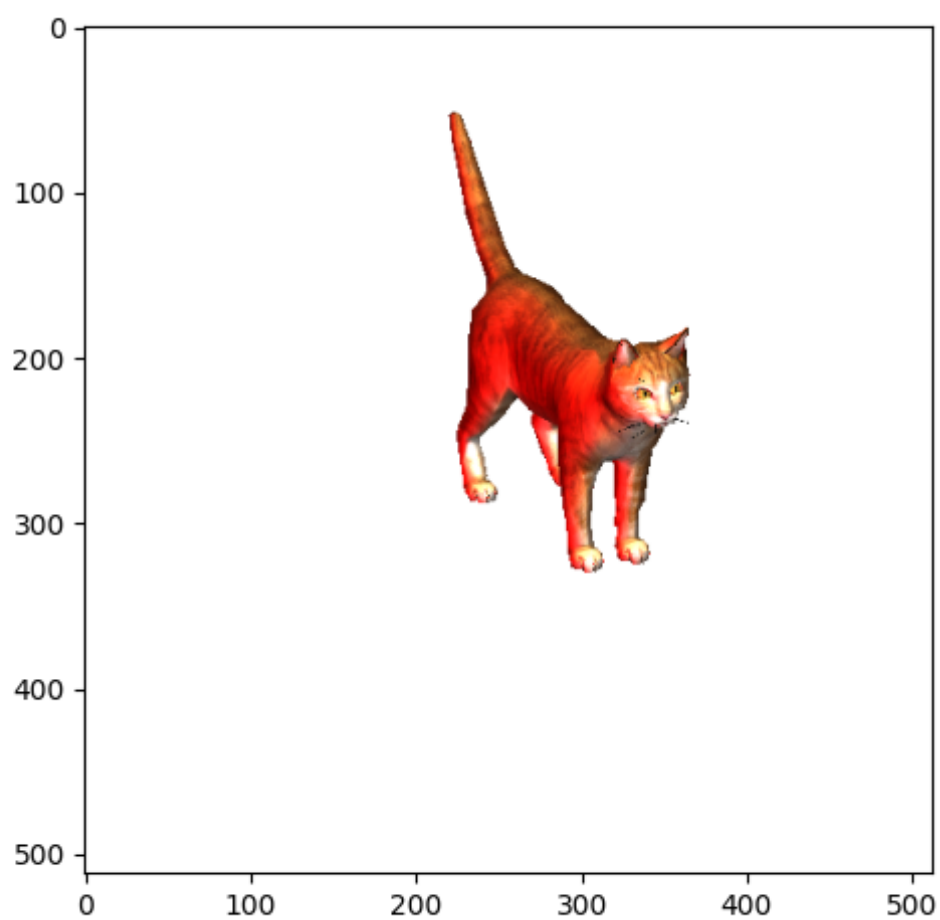
Phong Shading, all reflexions,  
third light source



Phong Shading, all reflexions,  
all light sources

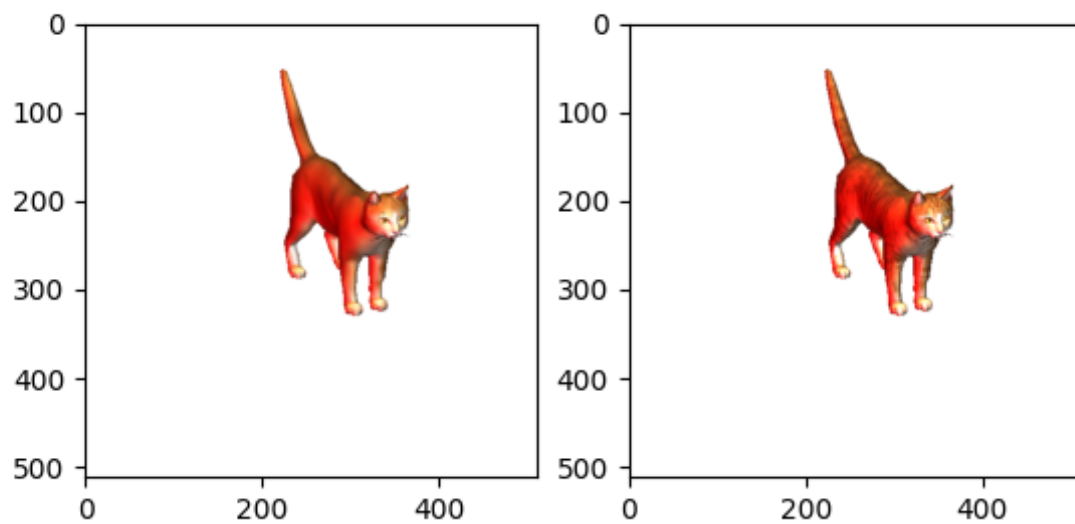


Σχήμα 3: Phong shading με διαφορετικές πηγές φωτός



Σχήμα 4: Phong shading με χρήση texture map





Σχήμα 5: Gouraud vs Phong shading με χρήση texture map