

# Γραφική με Υπολογιστές 2025

## Εργασία #3: Θέαση

Στην εργασία αυτή καλείστε να χρησιμοποιήσετε τους αλγορίθμους που υλοποιήθηκαν στις δύο προηγούμενες εργασίες με κάποιες αλλαγές προκειμένου να δημιουργηθεί ένα πλήρες πλαίσιο δημιουργίας φωτογραφιών μίας εικονικής σκηνής.

### Ζητούμενα

#### A. Φωτισμός και Υλικό επιφάνειας

Να αναπτύξετε την κλάση `MatPhong`, η οποία αναπαριστά ένα υλικό με ιδιότητες φωτισμού Phong.

```
1 class MatPhong:
2     def __init__(self, ka: float, kd: float, ks: float, n: float) -> None:
3         # Initialization ...
```

Η κλάση θα πρέπει να δέχεται ως παραμέτρους κατά την αρχικοποίησή της τους ακόλουθους συντελεστές:

- *ka*: Ο συντελεστής ανάκλασης διάχυτου φωτός από το περιβάλλον.
- *kd*: Ο συντελεστής διάχυτης ανάκλασης.
- *ks*: Ο συντελεστής κατοπτρικής ανάκλασης.
- *n*: Ο συντελεστής Phong.

Να δημιουργηθεί η συνάρτηση:

```
4     pt_l = light(
5         pt: np.ndarray,
6         nrm: np.ndarray,
7         vclr: np.ndarray,
8         cam_pos: np.ndarray,
9         mat: MatPhong,
10        l_pos: Union[np.ndarray, List[np.ndarray]],
11        l_int: Union[np.ndarray, List[np.ndarray]],
12        l_amb: np.ndarray
13    ) -> np.ndarray:
14        # Calculates the resulting illumination of a point
15        # due to its phong material and the camera viewpoint.
```

η οποία υπολογίζει το φωτισμό ενός σημείου  $\mathbf{pt}$ , το οποίο ανήκει σε μία επιφάνεια με υλικό τύπου `MatPhong`, με:

- $\mathbf{pt}$  είναι ένα διάνυσμα-στήλη  $1 \times 3$  με τις συντεταγμένες του σημείου της επιφάνειας.
- $\mathbf{nrm}$  είναι ένα διάνυσμα-στήλη  $1 \times 3$  με τις συντεταγμένες του κανονικού διανύσματος της επιφάνειας στο σημείο  $\mathbf{pt}$  (δηλαδή του κάθετου διανύσματος προς την επιφάνεια). Το διάνυσμα έχει φορά προς το εξωτερικό της επιφάνειας, δηλαδή προς την πλευρά του παρατηρητή.
- $\mathbf{vclr} = [c_r, c_g, c_b]^T$  είναι το  $1 \times 3$  διάνυσμα με τις συνιστώσες του χρώματος του σημείου  $\mathbf{pt}$ . Κάθε συνιστώσα ανήκει στο διάστημα  $[0, 1]$ .
- $\mathbf{cam\_pos}$  είναι ένα διάνυσμα-στήλη  $1 \times 3$  με τις συντεταγμένες του παρατηρητή (δηλαδή της κάμερας).
- $\mathbf{mat}$  είναι ένα αντικείμενο τύπου `MatPhong`.
- $\mathbf{l\_pos}$  είναι η θέση μίας (ή πολλών) σημειακών πηγών φωτισμού ( $N \times 3$ ).
- $\mathbf{l\_int}$  είναι η ένταση μίας (ή πολλών) σημειακών πηγών φωτισμού ( $N \times 3$ ).
- $\mathbf{l\_amb}$  είναι η ένταση διάχυτου φωτός από το περιβάλλον.

Η συνάρτηση υπολογίζει την ένταση της τριχρωματικής ακτινοβολίας  $\mathbf{pt\_l} = [I_r, I_g, I_b]^T$ , που ανακλάται από το σημείο  $\mathbf{pt}$ . Η ένταση συνεισφέρει αθροιστικά στο χρώμα του εικονοστοιχείου (pixel). Η συνάρτηση θα πρέπει να λειτουργεί σωστά τόσο για μία, όσο και για περισσότερες σημειακές πηγές φωτισμού, αθροίζοντας τη συνεισφορά κάθε πηγής.

## B. Shading

### B.1 Υπολογισμός κανονικών διανυσμάτων επιφάνειας

Έστω 3D αντικείμενο που αποτελείται αποκλειστικά από  $N_T$  τρίγωνα. Να υλοποιηθεί η συνάρτηση:

```
16 nrm = calc_normals(pts: np.ndarray, t_pos_idx: np.ndarray) -> np.ndarray:  
17 # Calculate normals per vertex of a triangle mesh.
```

όπου:

- $\mathbf{pts}$  είναι ένας πίνακας  $3 \times N_v$  με τις συντεταγμένες των κορυφών του αντικειμένου.
- $\mathbf{t\_pos\_idx}$  είναι ένας πίνακας  $3 \times N_T$  που περιγράφει τα τρίγωνα. Η  $k$ -οστή στήλη του  $\mathbf{t\_pos\_idx}$  περιέχει τους αύξοντες αριθμούς των κορυφών του  $k$ -οστού τριγώνου του αντικειμένου,  $1 \leq k \leq N_T$ . Η σειρά παράθεσης των κορυφών σηματοδοτεί με τον κανόνα του δεξιόστροφου κοχλία την κατεύθυνση του κανονικού διανύσματος και άρα και προς τα που είναι η εξωτερική πλευρά του αντικειμένου.

Η συνάρτηση υπολογίζει τον  $3 \times N_v$  πίνακα με τις συντεταγμένες των κάθετων διανυσμάτων σε κάθε σημείο (κορυφή) της επιφάνειας που ορίζει το αντικείμενο.

## B2. Συνάρτηση φωτογράφισης

Να υλοποιήσετε τη συνάρτηση:

```
18     img = render_object(  
19         v_pos: np.ndarray,  
20         v_uvs: np.ndarray,  
21         t_pos_idx: np.ndarray,  
22         tex: np.ndarray,  
23         plane_h: int,  
24         plane_w: int,  
25         res_h: int,  
26         res_w: int,  
27         focal: float,  
28         eye: np.ndarray,  
29         up: np.ndarray,  
30         target: np.ndarray,  
31         mat: MatPhong,  
32         l_pos: Union[np.ndarray, List[np.ndarray]],  
33         l_int: Union[np.ndarray, List[np.ndarray]],  
34         l_amb: np.ndarray,  
35         shader: str  
36     ) -> np.ndarray:  
37         """  
38         Render a 3D object with the specified parameters.  
39         """
```

η οποία δημιουργεί την έγχρωμη φωτογραφία *img* ενός 3D αντικειμένου, υπολογίζοντας το χρώμα με βάση τα μοντέλα φωτισμού της ενότητας A. Συγκεκριμένα υλοποιεί την παρακάτω διαδικασία:

1. Υπολογίζει τα κανονικά διανύσματα των κορυφών του αντικειμένου μέσω της συνάρτησης `calculate_normals`.
2. Προβάλλει τις κορυφές των τριγώνων σε ορθογώνιο πέτασμα κάμερας με χρήση της συνάρτησης `perspective_project` της εργασίας #2. Τα τρίγωνα με κορυφή εκτός του πετάσματος δεν θα χρωματίζονται.
3. Καλεί επανειλημμένα την αντίστοιχη συνάρτηση πλήρωσης που έχει επιλεγεί βάσει της τιμής της μεταβλητής `shader` για να χρωματίσει κάθε τρίγωνο του αντικειμένου, ξεκινώντας από εκείνα με το μεγαλύτερο βάθος (όπως και στις προηγούμενες εργασίες).

Τα ορίσματα που δέχεται έχουν ως εξής:

- `v_pos` είναι ένας πίνακας  $3 \times N_v$  με τις συντεταγμένες των κορυφών του αντικειμένου.
- `v_uvs` είναι πίνακας διάστασης  $2 \times N_v$  που σε κάθε γραμμή του περιέχει τις κανονικοποιημένες συντεταγμένες  $(u, v) \in [0, 1]^2$ , που δείχνουν τις θέσεις της εικόνας `texture tex`.
- `t_pos_idx` όπως ορίστηκε για τη συνάρτηση `calculate_normals`.
- `tex` είναι η εικόνα της υφής του αντικειμένου.
- `plane_h, plane_w` περιγράφουν τις φυσικές διαστάσεις του πετάσματος της κάμερας σε μονάδες μήκους ταυτόσημες με αυτές που χρησιμοποιούνται στο σύστημα συντεταγμένων της κάμερας.
- `res_h, res_w` είναι οι διαστάσεις της παραγόμενης εικόνας σε pixel (δηλαδή `res_w × res_h` pixels).

- `focal` είναι η απόσταση του πετάσματος από το κέντρο της κάμερας (στις μονάδες μέτρησης που χρησιμοποιεί το σύστημα συντεταγμένων της κάμερας).
- `eye` το  $3 \times 1$  διάνυσμα με τις συντεταγμένες του κέντρου της κάμερας ως προς το WCS.
- `target` το  $3 \times 1$  διάνυσμα με τις συντεταγμένες του σημείου στόχου της κάμερας.
- `up` το  $3 \times 1$  μοναδιαίο `up` διάνυσμα της κάμερας.
- `mat` ένα αντικείμενο τύπου `MatPhong`.
- `l_pos` είναι η θέση μίας (ή πολλών) σημειακών πηγών φωτισμού ( $N \times 3$ ).
- `l_int` είναι η ένταση μίας (ή πολλών) σημειακών πηγών φωτισμού ( $N \times 3$ ).
- `light_amb` =  $[I_r, I_g, I_b]^T$  είναι το  $3 \times 1$  διάνυσμα με τις συνιστώσες της έντασης της διάχυτης ακτινοβολίας του περιβάλλοντος. Κάθε συνιστώσα ανήκει στο διάστημα  $[0, 1]$ .
- `shader` είναι δυαδική μεταβλητή ελέγχου που χρησιμοποιείται για να επιλέξει τη συνάρτηση (που περιγράφονται στη συνέχεια) που θα χρησιμοποιηθεί για την πλήρωση των τριγώνων. Θεωρείστε ότι `shader` = "gouraud" αντιστοιχεί σε *Gouraud shading*, και `shader` = "phong" σε *Phong shading*.

Για δεδομένο τρίγωνο, να υπολογίζετε τα διανύσματα  $V$  και  $L$  των μοντέλων φωτισμού (με αναφορά στα σχήματα των σημειώσεων για τα μοντέλα φωτισμού <sup>1)</sup> μία φορά χρησιμοποιώντας ως σημείο  $P$  το κέντρο βάρους του τριγώνου (πριν την προβολή του) και να τα θεωρείτε σταθερά για όλα τα σημεία του τριγώνου.

### B3. Gouraud Shading

Να υλοποιηθεί η συνάρτηση:

```

40     updated_img = shade_gouraud(
41         v_pos: np.ndarray,
42         v_nrm: np.ndarray,
43         v_uvs: np.ndarray,
44         tex: np.ndarray,
45         cam_pos: np.ndarray,
46         mat: MatPhong,
47         l_pos: Union[np.ndarray, List[np.ndarray]],
48         l_int: Union[np.ndarray, List[np.ndarray]],
49         l_amb: np.ndarray
50         img: np.ndarray
51     ) -> np.ndarray:
52         """
53         Shade a triangle and update the specified image.
54         """

```

η οποία υπολογίζει χρωματίζει ένα δοθέν τρίγωνο βάσει του πλήρους μοντέλου φωτισμού (χρησιμοποιώντας δηλαδή τις συναρτήσεις της ενότητας 8.3). Η συνάρτηση θα πρέπει να χρησιμοποιεί γραμμική παρεμβολή για την εύρεση των συντεταγμένων  $(u, v)$  βάσει των οποίων θα δειγματοληπτεί από την εικόνα υφής `tex` για την εύρεση του χρώματος στα εσωτερικά σημεία του τριγώνου. Πιο αναλυτικά:

<sup>1)</sup>δηλαδή ως  $L$  και  $V$  εδώ εννοούνται τα διανύσματα από ένα σημείο  $P$  προς την πηγή φωτός και τον παρατηρητή αντίστοιχα, βλέπε σχήμα 8.3 σελίδα 99

- Ο πίνακας  $v\_pos$ , διάστασης  $3 \times 3$  περιέχει τις συντεταγμένες των κορυφών του τριγώνου μετά την προβολή τους στο πέτασμα της κάμερας.
- Ο πίνακας  $v\_nrm$  διάστασης  $3 \times 3$  περιέχει στις στήλες του τα κανονικά διανύσματα των κορυφών του τριγώνου.
- Ο πίνακας  $v\_uvs$ , διάστασης  $3 \times 2$  περιέχει τις  $(u, v)$  συντ/νες των σημείων του δοθέντος τριγώνου.
- $tex$  είναι η εικόνα της υφής του αντικειμένου.
- $cam\_pos$  είναι ένα διάνυσμα-στήλη  $3 \times 1$  με τις συντεταγμένες του παρατηρητή (δηλαδή της κάμερας).
- $mat$  ένα αντικείμενο τύπου `MatPhong`.
- $l\_pos$  είναι η θέση μίας (ή πολλών) σημειακών πηγών φωτισμού ( $N \times 3$ ).
- $l\_int$  είναι η ένταση μίας (ή πολλών) σημειακών πηγών φωτισμού ( $N \times 3$ ).
- $l\_amb = [I_r, I_g, I_b]^T$  είναι το  $1 \times 3$  διάνυσμα με τις συνιστώσες της έντασης της διάχυτης ακτινοβολίας του περιβάλλοντος. Κάθε συνιστώσα ανήκει στο διάστημα  $[0, 1]$ .
- $img$  είναι εικόνα (πίνακας διάστασης  $res\_h \times res\_w \times 3$ ) με τυχόν προϋπάρχοντα τρίγωνα
- $updated\_img$  είναι πίνακας διάστασης  $res\_h \times res\_w \times 3$  που για τα σημεία του τριγώνου θα περιέχει τις αντίστοιχες χρωματικές συνιστώσες ( $R_i, G_i, B_i$ ) καθώς και τα προϋπάρχοντα τρίγωνα της εισόδου  $img$  (επικαλύπτοντας τυχόν κοινά χρωματισμένα σημεία που προϋπήρχαν από άλλα τρίγωνα).

#### B4. Phong Shading

Να υλοποιηθεί η συνάρτηση:

```

55  updated_img = shade_phong(
56      v_pos: np.ndarray,
57      v_nrm: np.ndarray,
58      v_uvs: np.ndarray,
59      tex: np.ndarray,
60      cam_pos: np.ndarray,
61      mat: MatPhong,
62      l_pos: Union[np.ndarray, List[np.ndarray]],
63      l_int: Union[np.ndarray, List[np.ndarray]],
64      l_amb: np.ndarray
65      img: np.ndarray
66  ) -> np.ndarray:
67      """
68      Shade a triangle and update the specified image.
69      """

```

η οποία υπολογίζει το χρώμα των σημείων του τριγώνου πραγματοποιώντας παρεμβολή τόσο στα κανονικά διανύσματα όσο και στις συντ/νες  $(u, v)$ . Συγκεκριμένα:

- Για το δοθέν τρίγωνο, θα υπολογίζει τα κανονικά διανύσματα των αρχικών σημείων (δηλαδή, πριν την προβολή) κατά μήκος των ενεργών πλευρών εκτελώντας γραμμική παρεμβολή στα κανονικά διανύσματα των κορυφών της πλευράς.

- Για κάθε εσωτερικό σημείο, θα υπολογίζει το κανονικό διάνυσμα κατά μήκος του scan line εκτελώντας γραμμική παρεμβολή στα κανονικά διανύσματα που αντιστοιχούν στα ενεργά σημεία της πλευράς.
- Παρόμοια διαδικασία θα πραγματοποιείται και για τις συντ/νες  $(u, v)$  των σημείων.
- Έχοντας υπολογίσει το κανονικό διάνυσμα και το χρώμα για ένα σημείο, το χρώμα του θα προκύπτει χρησιμοποιώντας τη συνάρτηση `light()`.

Τα ορίσματα της συνάρτησης είναι όμοια με της συνάρτησης `shade_gouraud()`.

## Παραδοτέα

1. Οι παραπάνω συναρτήσεις σε μορφή **σχολιασμένου** πηγαίου κώδικα python ( $\leq$  v3.10) με σχόλια γραμμένα στα **αγγλικά** ή **greeklish**. (κοινώς, **μη γράφετε σχόλια με ελληνικούς χαρακτήρες**).
2. Αναφορά με:
  - Περιγραφή της λειτουργίας και του τρόπου κλήσης των προγραμμάτων.
  - Περιγραφή των συναρτήσεων.
  - Τα ενδεικτικά αποτελέσματα που παράγονται από το demo.
3. script επίδειξης με όνομα `demo.py`. Το script θα πρέπει να καλείται χωρίς εξωτερικά ορίσματα, να διαβάζει το αντικείμενο από το αρχείο `hw3.npy` που σας δίνεται, και να παράγει ενδεικτικές φωτογραφίες του αντικειμένου. Συγκεκριμένα, για κάθε επιλογή shader (Gouraud ή Phong) να παράγει και να αποθηκεύει 4 φωτογραφίες:
  - 3 με χρήση μόνο ενός είδους φωτισμού (ambient, diffusion, specular) και μία τέταρτη με συνδυασμό όλων των προηγούμενων. Συνολικά, θα πρέπει να παραχθούν 8 φωτογραφίες, 4 για κάθε shader. Οι τιμές όλων των μεταβλητών που θα χρειαστείτε περιέχονται στο αρχείο `hw3.npy`. Τέλος, θεωρείστε πως οι συντεταγμένες του αντικειμένου (verts) δίνονται ως προς το WCS.
  - Ακόμα, όσον αφορά το φωτισμό, σας δίνονται τρεις (3) σημειακές πηγές φωτισμού με διαφορετική θέση και διαφορετική ένταση. Ζητείται να αποτυπώσετε μία εικόνα για κάθε πηγή φωτισμού, και μία με όλες μαζί.

## Παρατηρήσεις

- Στο `.zip` αρχείο της εργασίας να περιέχεται και το αρχείο `hw3.npy` που σας δίνεται.
- Οι εργασίες αξιολογούνται με χρήση Python v3.10.
- Για ανοίξετε την εικόνα `texture_map`, μπορείτε να χρησιμοποιήσετε οποιαδήποτε βιβλιοθήκη της αρεσκείας σας (ενδεικτικά `opencv-python`, `pillow`, `imageio`).
- Οι εργασίες είναι **αυστηρά** ατομικές.
- Υποβάλετε ένα και μόνο αρχείο, τύπου `.zip`.

- Το όνομα του αρχείου πρέπει να είναι `AEM.zip`, όπου AEM είναι τα τέσσερα ψηφία του Α.Ε.Μ. του φοιτητή της ομάδας.
- Η αναφορά πρέπει να είναι ένα αρχείο τύπου PDF, και να έχει όνομα `report.pdf`.
- Το προς υποβολή αρχείο πρέπει να περιέχει τα αρχεία κώδικα Python και το αρχείο `report.pdf` το οποίο θα είναι η αναφορά της εργασίας.
- Όλα τα αρχεία κώδικα πρέπει να είναι αρχεία κειμένου τύπου UTF-8, και να έχουν κατάληξη `.py`.
- Το αρχείο τύπου zip που θα υποβάλλετε δεν πρέπει να περιέχει κανένα φάκελο.
- Για την ονομασία των αρχείων που περιέχονται στο προς υποβολή αρχείο, χρησιμοποιείτε μόνο αγγλικούς χαρακτήρες, και όχι ελληνικούς ή άλλα σύμβολα, πχ “#”, “\$”, “%” κλπ.

**Προσοχή: Θα αξιολογηθούν μόνο όσες εργασίες έχουν demos που τρέχουν!**