Γραφική με Υπολογιστές

1η Εργασία Μαθήματος 2022 - 2023

Αλεξόπουλος Δημήτριος ΑΕΜ 10091
aadimitri@ece.auth.gr

Περιεχόμενα

Κατάλογος Σχημάτων

1 Εισαγωγή: Πλήρωση Τριγώνων

Αντικείμενο της παρούσας εργασίας είναι η εφαρμογή ενός αλγορίθμου πλήρωσης πολυγώνων στην ειδική περίπτωση των τριγώνων. Συγκεκριμένα, θα περιγράψουμε την δομή του προγράμματος που απαιτείται και θα εξετάσουμε δύο διαφορετικές εκδοχές απόδοσης χρώματος στο εσωτερικό κάθε τριγώνου.

Η διαδικασία της πλήρωσης τριγώνων, σαν υπο-περίπτωση της διαδικασίας πλήρωσης κλειστών σχημάτων, συνίσταται στην επιλογή των pixels που αντιστοιχούν στο εσωτερικό του τριγώνου, έτσι ώστε να τους αποδοθεί μία συγκεκριμένη (χρωματική) τιμή. Αυτή η διαδικασία πραγματοποιείται σαρώνοντας γραμμή προς γραμμή τον καμβά των pixels. Κάθε σειρά από pixels του εν λόγω καμβά ονομάζεται γραμμή σάρωσης (scan line).

Μια ιδιαίτερα χρήσιμη ιδιαιτερότητα των τριγώνων σε σχέση με τα υπόλοιπα πολύγωνα αποτελεί η κυρτότητά τους. Έτσι, μία οποιαδήποτε γραμμή σάρωσης, είτε δεν έχει κανένα σημείο της (pixel) στο εσωτερικό του κυρτού πολυγώνου, είτε τα σημεία της που βρίσκονται στο εσωτερικό του είναι συνεχόμενα. Η παρατήρηση αυτή, όπως θα δούμε στη συνέχεια, βοηθά στην επιτάχυνση του αλγορίθμου πλήρωσης.

Για τους σκοπούς της εργασίας θα θεωρήσουμε ότι οι κορυφές των τριγώνων είναι σημεία του καμβά, έχουν δηλαδή ακέραιες συντεταγμένες αποκλειστικά εντός του καμβά. Επίσης, θα κάνουμε την παραδοχή ότι σε περίπτωση που κάποια σημεία του περιγράμματος ενός τριγώνου ταυτίζονται με κάποια σημεία του περιγράμματος ενός άλλου τριγώνου, τότε τα συνοριακά σημεία καταχωρούνται στο τρίγωνο που βρίσκεται ψηλότερα ή/και δεξιότερα στον καμβά.

1.1 Αλγόριθμος Πλήρωσης

Ο αλγόριθμος πλήρωσης των τριγώνων θα πρέπει να αποδίδει στο περίγραμμα και το εσωτερικό κάθε τριγώνου τις σωστές χρωματικές τιμές, λαμβάνοντας υπόψην κάθε περίπτωση τριγώνου και κάθε πιθανή γραμμή σάρωσης.

Έστω ότι το τρίγωνό μας ορίζεται από τις κορυφές του $p_k = (x_k, y_k) \in \mathbb{Z}^2$, $k = 0, 1, 2$ και τις πλευρές του e_k, όπου κάθε πλευρά e_k ενώνει τις κορυφές p_k και p_{k+1}. Τότε θα είναι $y_{k,min} = min\{y_k, y_{k+1}\}$ η τεταγμένη του χαμηλότερου άκρου της και αντίστοιχα θα ορίσουμε τα $y_{k,max}$, $x_{k,min}$ και $x_{k,max}$.

Η γραμμή σάρωσης y θα τέμνει μόνο τις πλευρές e_k για τις οποίες ισχύει $y_{k,min} \leq y \leq y_{k,max}$ κι άρα οι γραμμές σάρωσης που μας ενδιαφέρουν είναι μόνο αυτές για τις οποίες ισχύει:

$$y_{min} \equiv min_k\{y_{k,min}\} \leq y \leq max_k\{y_{k,max}\} \equiv y_{max} \tag{1.1}$$

Τα σημεία στα οποία μια γραμμή σάρωσης y τέμνει τις δύο πλευρές του τριγώνου ονομάζονται ενεργά οριακά σημεία της γραμμής σάρωσης y, ενώ οι αντίστοιχες πλευρές ονομάζονται ενεργές πλευρές.

Στην γενικότερη περίπτωση των πολυγώνων θα έπρεπε να εντοπίσουμε ποιά από τα διαστήματα μιας γραμμής σάρωσης βρίσκονται στο εσωτερικό του πολυγώνου (διαστήματα πλήρωσης), ώστε να χρωματίσουμε μόνο αυτά. Ωστόσο, στην περίπτωση των τριγώνων, λόγω της κυρτότητάς τους, υπάρχει μόνο ένα διάστημα πλήρωσης για κάθε μία από τις γραμμές σάρωσης, κι επομένως δεν χρειάζεται να γίνει αυτός ο έλεγχος. Το γεγονός αυτό απλοποιεί σημαντικά τον αλγόριθμο πλήρωσης στην περίπτωση των τριγώνων.

1.1.1 Ειδικές Περιπτώσεις

Στη συνέχεια θα αναφέρουμε κάποιες ειδικές περιπτώσεις τις οποίες θα πρέπει να είναι σε θέση να διαχειριστεί ο αλγόριθμος, ώστε η εικόνα που θα προκύψει από την πλήρωση των τριγώνων να είναι πλήρης και σωστά χρωματισμένη:

- Γραμμές σάρωσης που διέρχονται από οριζόντιες πλευρές:

  Οι οριζόντιες πλευρές των τριγώνων, αν υπάρχουν, θα βρίσκονται ακριβώς επάνω σε μία από τις γραμμές σάρωσης κι επομένως θα έχουν άπειρα σημεία τομής με αυτές. Αυτό προκαλεί πρόβλημα στην διαδικασία χρωματισμού της πλευράς. Για να το αντιμετωπίσουμε εξαιρούμε τις οριζόντιες πλευρές από τη λίστα των ενεργών πλευρών και πληρώνουμε ή όχι τα σημεία τους με βάση τη σύμβαση που περιγράφηκε παραπάνω (δηλαδή βάση του τριγώνου που βρίσκεται ψηλότερα ή/και δεξιότερα στον καμβά).

- Γραμμές σάρωσης που αγγίζουν κορυφές:

  Όταν μια γραμμή σάρωσης τέμνει δία διαδοχικές (μή οριζόντιες) ενεργές πλευρές στην κοινή κορυφή τους το διάστημα πλήρωσης που προκύπτει έχει μηδενικό μήκος. Σε αυτή την περίπτωση χρωματίζουμε απλά το μοναδικό σημείο της κορυφής.

- Γραμμές σάρωσης που διέρχονται από μία κορυφή και μία μη-οριζόντια ακμή:

  Στην περίπτωση αυτή το διάστημα πλήρωσης ορίζεται από το ενεργό σημείο της μη-οριζόντιας ακμής και από την άκρη της άλλης τρέχουσας ενεργής ακμής. Θα πρέπει να φροντίσουμε για την ανανέωση των ενεργών πλευρών, καθώς η μία εκ των δύο έχει φτάσει στο πέρας της.

1.1.2 Αναδρομικός Προσδιορισμός των Ενεργών Πλευρών

Κατασκευάζουμε μια λίστα με τις τρέχουσες ενεργές πλευρές. Η λίστα για την γραμμή σάρωσης $y+1$ προκύπτει από την αντίστοιχη λίστα της γραμμής y αρκεί να:

- Προστεθούν οι πλευρές με $y_{k,min} = y+1$
- Εξαιρεθούν οι πλευρές με $y_{k,max} = y$

1.1.3 Αναδρομικός Προσδιορισμός των Ενεργών Οριακών Σημείων

Μπορούμε να κατασκευάσουμε μία λίστα ενεργών οριακών σημείων ή να φροντίσουμε απλά, κατά τις επαναλήψεις που απαιτούνται για την διάσχιση των διαστημάτων πλήρωσης, να ανανεώνουμε σωστά τα ενεργά οριακά σημεία. Έτσι, τα ενεργά οριακά σημεία της γραμμής σάρωσης $y+1$ προκύπτουν από τα ενεργά οριακά σημεία της γραμμής y αρκέι να:

- Προστεθούν οι χαμηλότερες κορυφές με τεταγμένη $y_{k,min} = y+1$ των νέων πλευρών
- Εξαιρεθούν τα σημεία τομής με τις εξαιρούμενες πλευρές

- Προστεθεί στην τετμημένη όλων των υπόλοιπων σημείων ο όρος $1/m_k$, όπου m_k είναι η κλίση της αντίστοιχης πλευράς. Δηλαδή, $x_{y+1,k} = x_{y,k} + 1/m_k$. Ειδικά, όμως, για τις κατακόρυφες ενεργές πλευρές θα πρέπει να προσέξουμε ότι $x_{y+1,k} = x_{y,k}$

1.2 Ψευδοκώδικας

Έχουμε, λοιπόν, στη διάθεσή μας όλες τις πληροφορίες για να σχεδιάσουμε τον αλγόριθμο πλήρωσης πολυγώνων για την ειδική περίπτωση της πλήρωσης τριγώνων. Τα βήματα παρουσιάζονται παρακάτω:

```
Check for special cases in triangle coordinates

for k = 0:1:2

    Find xkmin, xkmax, ykmin, ykmax of edge k

end

y_min = min_k{y0min, y1min, ...}
y_max = max_k{y0max, y1max, ...}

Find the "active edges" list for the scanline y == y_min
Find the edge point coordinates (x1,y1) & (x2,y2) and their colors
for the scanline y == y_min

for y = y_min:1:y_max

    Update x-coordinates
    Set intersection colors

    for x = min(x1,x2):1:max(x1,x2)

        drawpixel(y,x)

    end

    Update the list of "active edges"

end
```

Στον παραπάνω ψευδοκώδικα η συνάρτηση *drawpixel* αφορά τον χρωματισμό του εκάστοτε διαστήματος πλήρωσης με βάση την μέθοδο flat shading ή gouraud shading. Οι μέθοδοι αυτές θα αναλυθούν παρακάτω.

2 Υλοποίηση

Κατά την υλοποίηση σε python του αλγορίθμου πλήρωσης τριγώνων για την σχεδίαση μιας εικόνας κατασκευάσαμε μία συνάρτηση $render()$, η οποία καλώντας τις βασικές συναρτήσεις μας, $flats()$ ή $gouraud()$, πραγματοποιεί τον χρωματισμό των τριγώνων. Για λόγους ευκολίας και κατανόησης του κώδικα στις συναρτήσεις αυτές καλείται ένας αριθμός από βοηθητικές συναρτήσεις. Στη συνέχεια θα παρουσιάσουμε λεπτομερέστερα την υλοποίηση αυτών των συναρτήσεων, ακολουθώντας τα βήματα της θεωρητικής ανάλυσης που προηγήθηκε στην εισαγωγή.

2.1 helpers

Αρχικά, ορίζουμε μια βοηθητική κλάση για το αντικείμενο της ακμής κι επέιτα έναν αριθμό από βοηθητικές συναρτήσεις. Οι συναρτήσεις αυτές καλούνται στις κύριες συναρτήσεις μας και αποσκοπούν σε μια πιο εύχρηστη και κατανοητή δομή στον κώδικα.

2.1.1 class Edge

Δημιουργήσαμε, αρχικά, το αντικείμενο $Edge$ για την ευκολότερη διαχείριση των χαρακτηριστικών των ακμών. Συγκεκριμένα, τα χαρακτηριστικά που τους δώσαμε είναι τα εξής:

- vertices: οι συντεταγμένες των κορυφών του τριγώνου που αντιστοιχούν στις άκρες της συγκεκριμένης ακμής
- vcolors: οι τιμές (r, g, b) που αντιστοιχούν στις παραπάνω κορυφές
- edge_slope: η κλίση της ακμής
- is_active: εάν η ακμή είναι ενεργή
- y_min, y_max: η ελάχιστη και η μέγιστη αντίστοιχα τεταγμένη της ακμής

2.1.2 interpolate_vectors

Κατασκευάσαμε στη συνέχεια την βοηθητική συνάρτηση $interpolate\_vectors(p1, p2, V1, V2, xy, dim)$ με τα παρακάτω ορίσματα και εξόδους:

- p1, p2: οι 2Δ συντεταγμένες των σημείων στα οποία αντιστοιχούν οι τιμές $V1, V2$
- V1, V2: οι (διανυσματικές) τιμές στις θέσεις p_1 και p_2
- xy: η τετμημένη (x) ή η τεταγμένη (y) του σημείου p ανάλογα με το αν $dim = 1$ ή $dim = 2$ αντίστοιχα
- V: επιστρεφόμενη τιμή μετά την παρεμβολή

Η συνάρτηση αυτή υπολογίζει την διανυσματική τιμή V στην θέση $p = (x, y)$ μέσω γραμμικής παρεμβολής μεταξύ των διανυσμάτων V_1 και V_2 που αντιστοιχούν στις θέσεις $p_1 = (x_1, y_1)$ και $p_2 = (x_2, y_2)$ με την υπόθεση ότι το σημείο p ανήκει στο ευθύγραμμο τμήμα $p_1 p_2$.

Σε περίπτωση που τα δύο σημεία έχουν την ίδια τετμημένη όταν $dim = 1$ ή την ίδια τεταγμένη όταν το $dim = 2$, δηλαδή στην περίπτωση που οι συντεταγμένες που απαιτούνται για την παρεμβολή ταυτίζονται, η συνάρτηση επιστρέφει την τιμή της μιας θέσης (π.χ. V_1).

2.1.3 slope

Η συνάρτηση $slope(a, b)$ έχει τα εξής ορίσματα και εξόδους:

- a, b: τα σημεία, σε συντεταγμένες x, y, που ορίζουν την ευθεία της οποίας την κλίση επιθυμούμε να υπολογίσουμε
- επιστρέφει την ζητούμενη κλίση

Σε περίπτωση που η ευθεία που ορίζουν τα δύο σημεία της εισόδου είναι κατακόρυφη η συνάρτηση επιστρέφει άπειρο.

2.1.4 initialize_variables

Η συνάρτηση $initialize\_variables(edges, active\_edges)$ καλείται από τις συναρτήσεις χρωματισμού των τριγώνων κι έχει τα εξής ορίσματα και εξόδους:

- edges: η λίστα με τις ακμές του τριγώνου
- active_edges: η λίστα με τις ενεργές ακμές του τριγώνου
- x1, y1: οι συντεταγμένες της αρχής του διαστήματος πλήρωσης για την τρέχουσα γραμμή σάρωσης
- x2,y2: οι συντεταγμένες του τέλους του διαστήματος πλήρωσης για την τρέχουσα γραμμή σάρωσης
- color1, color2: τα χρώματα των δύο παραπάνω σημείων

Σκοπός της συνάρτησης αυτής είναι στην περίπτωση που έχουμε οριζόντια ακμή για $y = y\_min$ να αρχικοποιήσει τις συντεταγμένες τις αρχής και του τέλους του διαστήματος πλήρωσης για την τρέχουσα γραμμή σάρωσης, καθώς και το διάνυσμα των αντίστοιχων χρωμάτων τους (r, g, b).

2.1.5 update_active_edges

Η συνάρτηση $update\_active\_edges(edges, active_edges, y)$ χρησιμοποιείται επίσης από τις συναρτήσεις χρωματισμού των τριγώνων σύμφωνα με τον αλγόριθμο που παρουσιάστηκε παραπάνω κι έχει τα εξής ορίσματα και εξόδους:

- edges: η λίστα με τις ακμές του τριγώνου
- active_edges: η λίστα με τις ενεργές ακμές του τριγώνου

- y: η τρέχουσα τεταγμένη κατά την σάρωση

Σκοπός της συνάρτησης αυτής είναι να ανανεώνει την λίστα με τις ενεργές ακμές μετά από κάθε επανάληψη κατά την κατακόρυφη σάρωση του τριγώνου. Για να το πετύχει αυτό ελέγχει εάν έχει έρθει το τέλος μιας ενεργής ακμής κι εάν αυτό ταυτίζεται με την αρχή κάποιας άλλης ακμής.

2.2 flats

Η συνάρτηση $flats(canvas, vertices, vcolors)$ είναι μία από τις δύο βασικές συναρτήσεις μας και αποτελεί την πιο απλή μέθοδο πλήρωσης τριγώνων. Τα ορίσματα και οι έξοδοί της είναι οι εξής:

- canvas: ο $M \times N \times 3$ πίνακας με το *grid* της εικόνας μας για κάθε ένα από τα τρία κανάλια χρώματος (r, g, b)
- vertices: ο 3×2 πίνακας με τις συντεταγμένες σε κάθε γραμμή του μιας κορυφής του τριγώνου
- vcolors: ο 3×3 πίνακας με το χρώμα σε κάθε γραμμή του μιας κορυφής του τριγώνου σε μορφή *RGB* (με τιμές στο διάστημα[0, 1])
- updatedcanvas: ο επιστρεφόμενος $M \times N \times 3$ πίνακας που για όλα τα σημεία του τριγώνου περιέχει τις υπολογισμένες χρωματικές συνιστώσες, καθώς και τα προϋπάρχοντα τρίγωνα της εισόδου *canvas* (επικαλύπτοντας τυχόν κοινά χρωματισμένα σημεία που προϋπήρχαν από την πλήρωση άλλων τριγώνων)

Η συνάρτηση αυτή εκτελεί τον αλγόριθμο που περιγράφηκε με τον ψευδοκώδικα στην θεωρητική ανάλυση που προηγήθηκε. Ερευνά τις περιπτώσεις που όλες οι κορυφές, ή τουλάχιστον δύο από τις κορυφές του τριγώνου, ταυτίζονται και στη συνέχεια εντοπίζει τις αρχικές ενεργές ακμές.

Διαχειρίζεται την πρώτη (χαμηλότερη) γραμμή σάρωσης σαν ξεχωριστή περίπτωση, ενώ στη συνέχεια σαρώνει κατάλληλα κάθε τεταγμένη και γι αυτήν κάθε τετμημένη του καμβά. Αφού γεμίσει το τρέχον διάστημα πλήρωσης, λοιπόν, και για κάθε επανάληψη ανανεώνει, καλώντας την προηγούμενη συνάρτηση, τις ενεργές ακμές, αλλά και τα ενεργά οριακά σημεία.

Προκειμένου να εξασφαλίσει ότι η πλήρωση των περιθωρίων του τριγώνου θα γίνει βάση του τριγώνου που βρίσκεται ψηλότερα ή/και δεξιότερα στον καμβά, δεν χρειάζεται να χρωματιστεί η γραμμή σάρωσης $y = y_{max}$ και η τελευταία τετμημένη της κάθε γραμμής σάρωσης $(x2)$.

Σύμφωνα με τη μέθοδο flat shading όλα τα σημεία του τριγώνου θα έχουν το χρώμα που προκύπτει ως ο μέσος όρος του χρώματος των κορυφών του.

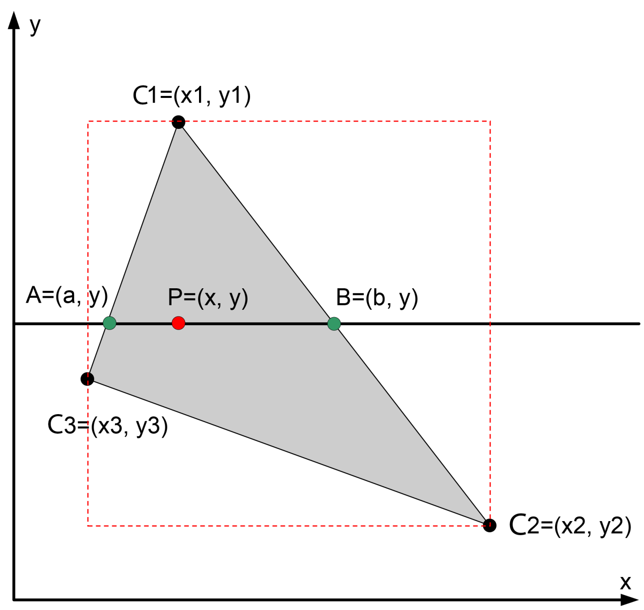
2.3 gourauds

Η συνάρτηση $gourauds(canvas, vertices, vcolors)$ αποτελεί την δεύτεση μέθοδο πλήρωσης τριγώνων που θα μελετήσουμε στην παρούσα εργασία. Τα ορίσματα και οι έξοδοί της είναι ακριβώς τα ίδια με την συνάρτηση $flats()$:

- canvas: ο $M \times N \times 3$ πίνακας με το *grid* της εικόνας μας για κάθε ένα από τα τρία κανάλια χρώματος (r, g, b)
- vertices: ο 3×2 πίνακας με τις συντεταγμένες σε κάθε γραμμή του μιας κορυφής του τριγώνου
- vcolors: ο 3×3 πίνακας με το χρώμα σε κάθε γραμμή του μιας κορυφής του τριγώνου σε μορφή *RGB* (με τιμές στο διάστημα[0, 1])
- updatedcanvas: ο επιστρεφόμενος $M \times N \times 3$ πίνακας που για όλα τα σημεία του τριγώνου περιέχει τις υπολογισμένες χρωματικές συνιστώσες, καθώς και τα προϋπάρχοντα τρίγωνα της εισόδου *canvas* (επικαλύπτοντας τυχόν κοινά χρωματισμένα σημεία που προϋπήρχαν από την πλήρωση άλλων τριγώνων)

Η συνάρτηση αυτή εκτελεί, επίσης, τον αλγόριθμο που περιγράφηκε με τον ψευδοκώδικα στην θεωρητική ανάλυση που προηγήθηκε. Ισχύουν γι΄ αυτήν όλα όσα ισχύουν και για την συνάρτηση $flats()$, ωστόσο αυτή τη φορά δεν έχουν όλα τα σημεία του τριγώνου το χρώμα που προκύπτει ως ο μέσος όρος του χρώματος των κορυφών του.

Τα σημεία του τριγώνου θα έχουν το χρώμα που θα υπολογίζεται με γραμμική παρεμβολή, μέσω της συνάρτησης *interpolate_vectors*(), από το χρώμα των κορυφών του. Συγκεκριμένα, για το χρωματισμό του τριγώνου, με αναφορά στα σημεία του παρακάτω σχήματος, πρώτα θα υπολογίζεται το χρώμα στις θέσεις A και B, με χρήση της συνάρτησης παρεμβολής για τις τιμες χρωμάτων των κορυφών $C1, C3$ και $C1, C2$ αντίστοιχα (δίνοντας τις κατάλληλες συνιστώσες των συντεταγμένων των κορυφών του τριγώνου).



Σχήμα 2.1: *Gouraud triangle shading*

Η πρώτη αυτή φάση υλοποιείται μία φορά για κάθε **scanline** y. Σε δεύτερη φάση, θα πρέπει πάλι με την χρήση της *interpolate_vectors* να γίνει γραμμική παρεμβολή για κάθε σημείο $P = (x, y)$ που ανήκει στην τρέχουσα γραμμή σάρωσης. Έτσι, προκύπτει το χρώμα κάθε σημίου του τριγώνου πάνω στον καμβα.

2.4 render

Η συνάρτηση $render(verts2d, faces, vcolors, depth, shade\_t)$ αξιοποιεί τις παραπάνω συναρτήσεις για τον χρωματισμό του αντικειμένου της εικόνας μας. Έχει τα εξής ορίσματα και εξόδους:

- verts2d: ο $L \times 2$ πίνακας με τις κορυφές των τριγώνων της εικόνας
- faces: ο $K \times 3$ πίνακας που περιέχει τις κορυφές των K τριγώνων. Η i-στη γραμμή του πίνακα δηλώνει τις τρεις κορυφές που σχηματίζουν το τρίγωνο (με αναφορά σε κορυφές του πίνακα $verts2d$ και αρίθμηση που ξεκινά από το 0)
- vcolors: ο $L \times 3$ πίνακας με τα χρώματα των κορυφών. Η i-στη γραμμή του πίνακα δηλώνει τις χρωματικές συνιστώσες της αντίστοιχης κορυφής
- depth: ο $L \times 1$ πίνακας που δηλώνει το βάθος της κάθε κορυφής πριν την προβολή του αντικειμένου στις 2 διαστάσεις
- shate_t: μεταβλητή ελέγχου (τύπου $string$) που καθορίζει τη συνάρτηση χρωματισμού ($Gouraud$ ή $Flat$) που θα χρησιμοποιηθεί, και μπορεί να πάρει τιμές $\{"flat","gouraud"\}$
- M, N: το ύψος και το πλάτος του καμβά αντίστοιχα
- img: η επιστρεφόμενη έγχρωμη εικόνα διάστασης $M \times N \times 3$. Η εικόνα θα περιέχει K χρωματισμένα τρίγωνα τα οποία σχηματίζουν την προβολή ενός $3D$ αντικειμένου στις 2 διαστάσεις

Σκοπός της συνάρτησης αυτής είναι η πλήρωση με τον κατάλληλο τρόπο όλων των τριγώνων της εισόδου. Για τον σκοπό αυτό, στο εσωτερικό της συνάρτησης καλείται η $shade\_triangle()$, η οποία ανάλογα με την τιμή της μεταβλητής $shade\_t$, καλεί την ανάλογη ρουτίνα για το χρωματισμό των εσωτερικών σημείων κάθε τριγώνου.

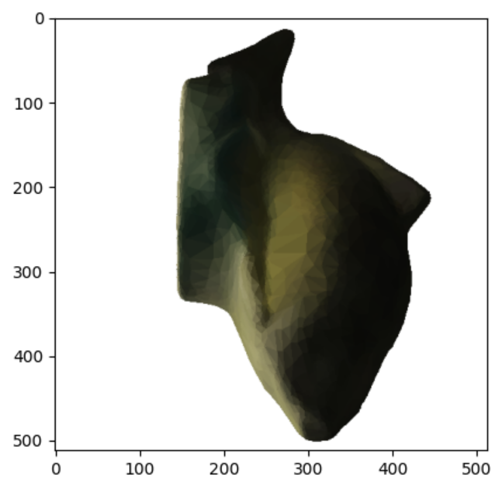
Η σειρά με την οποία χρωματίζονται τα τρίγωνα προκύπτει από τον πίνακα βάθους ($depth$). Ο χρωματισμός ξεκινάει από τα μακρινότερα (αυτά με το μεγαλύτερο βάθος) τρίγωνα και συνεχίζει με τα κοντινότερα. Το βάθος ενός τριγώνου υπολογίζεται ως το κέντρο βάρους τους βάθους των κορυφών του. Θεωρούμε, τέλος, δεδομένο ότι το $background$ του καμβά είναι λευκό ($rgb = (1,1,1)$) κι ότι ο καμβάς έχει διαστάσεις $M = 512$, $N = 512$.

3 Αποτελέσματα

Στη συνέχεια θα παρουσιάσουμε τις εικόνες του αντικειμένου που προκύπτουν με κάθε μία από τις δύο μεθόδους σκίασης. Οι εικόνες εξάγονται από τον κώδικα που υλοποιήθηκε με είσοδο τα δεδομένα που δόθηκαν για τους σκοπούς τις εργασίας. Ακολουθήθηκαν οι συμβάσεις και οι παραδοχές που περιγράφτηκαν παραπάνω.

3.1 **Flat Shading**

Τρέχοντας το *script* επίδειξης της μεθόδου $flatshading$ (*demo_flat.py*) λαμβάνουμε το παρακάτω αποτέλεσμα για τα δεδομένα της εισόδου που δόθηκαν:

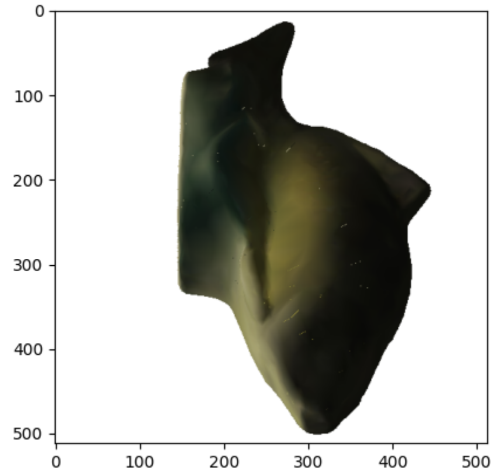


Σχήμα 3.1: *Example: Flat shading*

Σημειώνουμε ότι γεμίζουμε τον καμβά με την μορφή $updatedcanvas[y,x]$, διότι το y ως αριθμός γραμμής αντιστοιχίζεται στην τεταγμένη του καρτεσιανού επιπέδου, ενώ το x ως αριθμός στήλης αντιστοιχίζεται στην τετμημένη.

3.2 **Gouraud Shading**

Τρέχοντας το *script* επίδειξης της μεθόδου $gouraudshading$ (*demo_gouraud.py*) λαμβάνουμε το παρακάτω αποτέλεσμα για τα δεδομένα της εισόδου που δόθηκαν:



Σχήμα 3.2: *Example: Gouraud shading*

Σημειώνουμε και πάλι ότι γεμίζουμε τον καμβά με την μορφή $updatedcanvas[y,x]$, διότι το y ως αριθμός γραμμής αντιστοιχίζεται στην τεταγμένη του καρτεσιανού επιπέδου, ενώ το x ως αριθμός στήλης αντιστοιχίζεται στην τετμημένη.

3.3 Παρατηρήσεις

- Βλέπουμε ότι στην μέθοδο $flatshading$ διακρίνονται οι ακμές των τριγώνων, γεγονός αναμενόμενο, καθώς κάθε τρίγωνο χρωματίζεται εσωτερικά απλά από την μέση τιμή του χρώματος των κορυφών του.

- Στην μέθοδο *gouraudshading* λόγω της γραμμικής παρεμβολής, πρώτα κατακόρυφα κι έπειτα οριζόντια, βλέπουμε ότι το πρόβλημα αυτό έχει εξελιφθεί. Οι μεταβάσεις μεταξύ των τριγώνων είναι πλέον ομαλές δίνοντας μια ικανοποιητική τελική 3Δ αναπαράσταση του αντικειμένου επάνω στο 2Δ επίπεδο. Ωστόσο, παρατηρούμε κάποιες ατέλειες που ενδεχομένως να οφείλονται σε κάποιον λανθασμένο χειρισμό των οριακών περιπτώσεων.

- Όσον αφορά τους χρόνους εκτέλεσης, παρατηρούμε ότι η μέθοδος *gouraud* είναι ελάχιστα πιο χρονοβόρα, αναμενόμενο καθώς καλεί επαναλαμβανόμενα την συνάρτηση παρεμβολής, σε αντίθεση με την μέθοδο $flat$.

Βιβλιογραφία

[1] Σημειώσεις του μαθήματος