

## Αναφορά στο 1<sup>ο</sup> παραδοτέο Γραφική με υπολογιστές

### Θέμα εργασίας: Πλήρωση τριγώνων

Τα ζητούμενα της εργασίας ήταν

1. Μια συνάρτηση γραμμικής παρεμβολής
2. Μια συνάρτηση για τη διαδικασία χρωματισμού του κάθε τριγώνου flat
3. Μια συνάρτηση για τη διαδικασία χρωματισμού του κάθε τριγώνου gouraud
4. Μια συνάρτηση για τη διαδικασία χρωματισμού ολόκληρου του αντικειμένου
5. Ένα script με όνομα demo\_flat
6. Ένα script με όνομα demo\_gouraud

### Συνάρτηση γραμμικής παρεμβολής

Είναι μια σύντομη συνάρτηση που δέχεται ως είσοδο

- 2 κορυφές  $p1, p2$  με συντεταγμένες  $(x, y)$ ,
- το σημείο παρεμβολής  $a$  με συντεταγμένες  $(x, y)$
- 2 μεταβλητές  $V1, V2$  που δείχνουν το χρώμα των κορυφών  $p1$  και  $p2$  αντίστοιχα με  $(R, G, B)$
- Μια μεταβλητή  $dim$  που δηλώνει με
  - $dim = 1$  η γραμμική παρεμβολή να γίνει οριζόντια (δηλ. Στο  $x$  του σημείου  $a$ ) και
  - $dim = 2$  να γίνει κάθετα (δηλ. Στο  $y$  του σημείου  $a$ )

και έχει ως έξοδο μια μεταβλητή  $value$ , δηλαδή την τιμή που προκύπτει από τη γραμμική παρεμβολή των  $V1$  και  $V2$ .

#### Περιγραφή διαδικασίας γραμμικής παρεμβολής:

Για την οριζόντια παρεμβολή αρχικά ελέγχουμε αν οι τετμημένες των δυο σημείων είναι ίσες. Για να αποφύγουμε διαίρεση με το μηδέν βάζω την τιμή 0.5 στο λάμδα, ώστε να πάρει τιμή τον μέσο όρο των δυο χρωμάτων. Στην άλλη περίπτωση γίνεται κανονικά η διαίρεση και βγάζουμε μια τιμή λάμδα.

Για την κάθετη παρεμβολή έχουμε το ίδιο κλάσμα αλλά για τις τεταγμένες των σημείων.

Τέλος στην τιμή  $value$  βγαίνει η τελική απόχρωση του σημείου που μελετάμε.

#### Ψευδοκώδικας:

```
if dim == 1
    if x1 == x2
        lamda = 0.5
    else
        lamda = (x2 - x) / (x2 - x1)
elseif dim == 2
    lamda = (y2 - y) / (y2 - y1)
value = lamda * V1 + (1 - lamda) * V2
```

## Συνάρτηση για τη διαδικασία χρωματισμού του κάθε τριγώνου flat

Είναι μια συνάρτηση που δέχεται ως είσοδο

- μια εικόνα `img` ( $M \times N \times 3$ ),
- έναν πίνακα `vertices_2d`: ( $x_1, y_1 ; x_2, y_2 ; x_3, y_3$ ) είναι οι συντεταγμένες των κορυφών του τριγώνου,
- έναν πίνακα `vertex_colors`: ( $R_1, G_1, B_1 ; R_2, G_2, B_2 ; R_3, G_3, B_3$ ) είναι τα χρώματα των κορυφών του τριγώνου

και έχει ως έξοδο έναν πίνακα  $Y$  ( $M \times N \times 3$ ), που περιέχει για όλα τα σημεία τα χρώματα ( $R, G, B$ )

### Περιγραφή διαδικασίας χρωματισμού:

Η εκδοχή `flat` χρωματίζει το κάθε τρίγωνο με το χρώμα που προκύπτει από τον μέσο όρο των χρωμάτων των τριών κορυφών του.

Οπότε, ξεκινάμε με τον υπολογισμό του μέσου όρου των χρωμάτων.

Στη συνέχεια, ταξινομούμε τις τρεις κορυφές βάσει του  $y$  τους. Από την χαμηλότερη προς την υψηλότερη.

Έπειτα, υπολογίζουμε τα ενεργά σημεία κάθε πλευράς του τριγώνου. Οι τρεις πλευρές είναι:

- από την χαμηλότερη κορυφή μέχρι την μεσαία, (1-2)
- από την μεσαία κορυφή μέχρι την υψηλότερη (2-3) και
- από την χαμηλότερη κορυφή μέχρι την υψηλότερη (1-3).

Για κάθε πλευρά, αποθηκεύουμε σε έναν πίνακα `x_values`, τις τιμές των  $x$ .

Στην τιμή του  $x$  προστίθεται σε κάθε επανάληψη η ανάστροφη τιμή της κλίσης της ευθείας που σχηματίζεται από τα δυο σημεία της πλευράς δηλαδή τις δυο κορυφές του τριγώνου.

Τελειώνοντας την διαδικασία αυτή, έχουμε τους πίνακες  $x_{12}$ ,  $x_{23}$ ,  $x_{13}$ . Έχουμε δηλαδή όλα τα σημεία του περιβλήματος του τριγώνου.

Έπειτα, προσθέτοντας την  $x_{23}$  στην  $x_{13}$ . Αποκτάμε τον πίνακα  $x_{123}$ .

Έχουμε χωρίσει λοιπόν το τρίγωνο σε 2 κομμάτια. Στην μεγάλη πλευρά, που πάει από το χαμηλότερο σημείο στο υψηλότερο ( με τιμές  $x_{13}$ ) και στις δυο μικρές πλευρές, που η μια από το χαμηλότερο σημείο πάει στο μεσαίο και η άλλη που από το μεσαίο σημείο πάει στο υψηλό (με τιμές  $x_{123}$ ).

Στη συνέχεια επιλέγουμε ένα στοιχείο από τον πίνακα  $x_{13}$  και το συγκρίνουμε με το αντίστοιχο στοιχείο του πίνακα  $x_{123}$ . Ο πίνακας με το στοιχείο με τη μικρότερη τιμή θα γίνει η αριστερή πλευρά του τριγώνου και ο άλλος θα γίνει η δεξιά.

Έτσι, οι δυο πίνακες γίνονται `x_left` και `x_right`.

Τέλος, χρωματίζουμε το τρίγωνο.

Από το χαμηλότερο σημείο του τριγώνου και από αριστερά προς τα δεξιά χρωματίζουμε τον καμβά με το χρώμα που υπολογίσαμε, με οριζόντιες στρώσεις δηλαδή, μέχρι να φτάσουμε στο υψηλότερο σημείο του τριγώνου.

**Ψευδοκώδικας:**

```
mesos oros xromaton r,g,b
color = mean(vertex_colors)

sorting triangle points to y1 <= y2 <= y3
if y2 < y1 {swap (P2, P1)}
if y3 < y1 {swap (P3, P1)}
if y3 < y2 {swap (P3, P2)}
if y1 == y2
    x12_values = x1
else
    m = (x2 - x1) / (y2 - y1)
    x = x1
    i=1
    for y = y1 to y2
        x12_values(i) = x
        x = x + m
        i = i+1
x12 = round(x12_values)

if y2 == y3
    x23_values = x2
else
    m = (x3 - x2) / (y3 - y2)
    x = x2
    i = 1
    for y = y2 to y3
        x23_values(i) = x
        x = x + m
        i = i+1
x23 = round(x23_values)

if y1 == y3
    x13_values = x1
else
    m = (x3 - x1) / (y3 - y1)
    x = x1;
    i = 1;
    for y = y1 to y3
        x13_values(i) = x
        x = x + m
        i = i+1
x13 = round(x13_values)

afairo to teleutaio stoixeio apo tin x12 giati einai idio
me to proto stoixeio tou x23
x123 = [x12 x23]

mid = stoixeio tou x13
if x13(mid) < x123(mid)
    x_left = x13
    x_right = x123
else
    x_left = x123
    x_right = x13
```

```
for y = y1 to y3
  for x = x_left to x_right
    Y(y,x,:) = color
```

## Συνάρτηση για τη διαδικασία χρωματισμού του κάθε τριγώνου gouraud

Είναι μια συνάρτηση που δέχεται ως είσοδο

- μια εικόνα img (MxNx3),
- έναν πίνακα vertices\_2d: (x1, y1 ; x2, y2 ; x3, y3) είναι οι συντεταγμένες των σημείων του τριγώνου,
- έναν πίνακα vertex\_colors: (R1, G1, B1 ; R2, G2, B2 ; R3, G3, B3) είναι τα χρώματα των κορυφών του τριγώνου

και έχει ως έξοδο έναν πίνακα Y (MxNx3), που περιέχει για όλα τα σημεία τα χρώματα (R, G, B)

### Περιγραφή διαδικασίας χρωματισμού:

Η εκδοχή gouraud υπολογίζει το χρώμα των σημείων του τριγώνου με γραμμική παρεμβολή από το χρώμα των κορυφών του.

Αρχικά, ταξινομούμε τις τρεις κορυφές και τα χρώματά τους βάσει του y τους. Από την χαμηλότερη προς την υψηλότερη.

Έπειτα, υπολογίζουμε τα ενεργά σημεία κάθε πλευράς του τριγώνου. Οι τρεις πλευρές είναι:

- από την χαμηλότερη κορυφή μέχρι την μεσαία, (1-2)
- από την μεσαία κορυφή μέχρι την υψηλότερη (2-3) και
- από την χαμηλότερη κορυφή μέχρι την υψηλότερη (1-3).

Για κάθε πλευρά, αποθηκεύουμε σε έναν πίνακα x\_values, τις τιμές των x και σε έναν πίνακα color τα αντίστοιχα χρώματα.

Στην τιμή του x προστίθεται σε κάθε επανάληψη η ανάστροφη τιμή της κλίσης της ευθείας που σχηματίζεται από τα δυο σημεία της πλευράς δηλαδή τις δυο κορυφές του τριγώνου.

Το χρώμα σε κάθε σημείο υπολογίζεται καλώντας τη συνάρτηση γραμμικής παρεμβολής.

Τελειώνοντας την διαδικασία αυτή, έχουμε τους πίνακες x12, x23, x13 , color12, color23, color13. Έχουμε δηλαδή όλα τα σημεία του περιβλήματος του τριγώνου και τα χρώματά τους.

Έπειτα, προσθέτοντας την x23 στην x13 και την color12 στην color23, αποκτάμε τους πίνακες x123 και color123.

Έχουμε χωρίσει λοιπόν το τρίγωνο σε 2 κομμάτια. Στην μεγάλη πλευρά, που πάει από το χαμηλότερο σημείο στο υψηλότερο ( με τιμές x13 και color13) και στις δυο μικρές πλευρές, που η μια από το χαμηλότερο σημείο πάει στο μεσαίο και η άλλη που από το μεσαίο σημείο πάει στο υψηλό (με τιμές x123 και color123).

Στη συνέχεια επιλέγουμε ένα στοιχείο από τον πίνακα x13 και το συγκρίνουμε με το αντίστοιχο στοιχείο του πίνακα x123. Ο πίνακας με το στοιχείο με τη μικρότερη τιμή θα γίνει η αριστερή πλευρά του τριγώνου και ο άλλος θα γίνει η δεξιά.

Έτσι, οι δυο πίνακες γίνονται x\_left και x\_right.

Αντίστοιχα, οι πίνακες με τα χρώματα γίνονται color\_left και color\_right.

Τέλος, χρωματίζουμε το τρίγωνο.

Από το χαμηλότερο σημείο του τριγώνου και από αριστερά προς τα δεξιά χρωματίζουμε τον καμβά με γραμμική παρεμβολή, με οριζόντιες στρώσεις δηλαδή, μέχρι να φτάσουμε στο υψηλότερο σημείο του τριγώνου.

Σημείωση: Για τα σημεία των πλευρών, η γραμμική παρεμβολή γίνεται κάθετα, ενώ για το εσωτερικό του τριγώνου γίνεται οριζόντια.

### Ψευδοκώδικας:

```
sorting triangle points to  $y_1 \leq y_2 \leq y_3$ 
and its colors

if  $y_2 < y_1$ 
    swap (P2, P1)
    swap (V2, V1)
if  $y_3 < y_1$ 
    swap (P3, P1)
    swap (V3, V1)
if  $y_3 < y_2$ 
    swap (P3, P2)
    swap (V3, V2)

if  $y_1 == y_2$ 
    x_values = x1
    color12(1,1) = R1
    color12(1,2) = G1
    color12(1,3) = B1
else
    m = (x2 - x1) / (y2 - y1)
    x = x1
    i=1
    for y = y1 to y2
        x_values(i) = x
        x = x + m
        color12(y) = vector_interp(p1, p2, a, V1, V2, 2)
        i = i+1
x12 = round(x_values)

if  $y_2 == y_3$ 
    x_values = x2
    color23(1,1) = R2
    color23(1,2) = G2
    color23(1,3) = B2
else
    m = (x3 - x2) / (y3 - y2)
    x = x2
    i = 1
    for y = y2 to y3
        x_values(i) = x
        x = x + m
        color23(y) = vector_interp(p2, p3, a, V2, V3, 2)
        i = i+1
x23 = round(x_values)
```

```
if y1 == y3
    x_values = x1
    color13(1,1) = R1
    color13(1,2) = G1
    color13(1,3) = B1
else
    m = (x3 - x1) / (y3 - y1)
    x = x1
    i = 1
    for y = y1 to y3
        x_values(i) = x
        x = x + m
        color13(y) = vector_interp(p1, p3, a, V1, V3, 2)
        i = i+1
x13 = round(x_values)

afairo to teleutaio stoixeio apo tin x12 giati einai idio
me to proto stoixeio tou x23
x123 = [x12 x23]

afairo to teleutaio stoixeio apo tin color12 giati einai idio
me to proto stoixeio tou color23
color123 = [color12; color23]

mid = stoixeio tou x13
if x13(mid) < x123(mid)
    x_left = x13
    color_left = color13
    x_right = x123
    color_right = color123
else
    x_left = x123
    color_left = color123
    x_right = x13
    color_right = color13

i = 1
for y = y1 to y3
    for x = x_left to x_right
        Y(y,x,:) = vector_interp(x_left(i),x_right(i),a, color_left(i),
            color_right(i), 1)
    i = i + 1

sto telos ksana kano paint tis korufes gt kapoies emenan aspres
Y(y1, x1, :) = V1;
Y(y2, x2, :) = V2;
Y(y3, x3, :) = V3;
```

## Συνάρτηση χρωματισμού αντικειμένου

Είναι μια συνάρτηση που δέχεται ως είσοδο

- έναν πίνακα `vertices_2d`: ( $L \times 2$ ) που περιέχει τις συντεταγμένες  $L$  σημείων,
- έναν πίνακα `faces`: ( $K \times 3$ ) που περιέχει τις τρεις κορυφές  $K$  τριγώνων,
- έναν πίνακα `vertex_colors`: ( $L \times 3$ ) που περιέχει τα χρώματα των  $L$  κορυφών,
- έναν πίνακα `depth`: ( $L \times 1$ ) που περιέχει το βάθος της κάθε κορυφής,
- μια string μεταβλητή `renderer` που καθορίζει τη συνάρτηση χρωματισμού που θα χρησιμοποιηθεί (`flat`, `gouraud`)

και έχει ως έξοδο μια εικόνα `img` ( $M \times N \times 3$ ), που περιέχει όλα τα χρωματισμένα τρίγωνα.

### Περιγραφή διαδικασίας χρωματισμού αντικειμένου:

Αρχικά, βρίσκουμε πόσα τρίγωνα υπάρχουν. Αυτό δηλώνεται από τον αριθμό των γραμμών του πίνακα `faces`.

Έπειτα, υπολογίζουμε το βάθος κάθε τριγώνου, παίρνοντας τον μέσο όρο από τα βάθη των τριών κορυφών κάθε φορά και τα αποθηκεύω σε έναν πίνακα `depths`.

Στη συνέχεια, ταξινομούμε τον πίνακα `depths` κατά φθίνουσα σειρά και αποθηκεύουμε το αποτέλεσμα σε νέο πίνακα `sorted_depths`.

Επαναλαμβάνουμε τη διαδικασία ταξινόμησης για τον πίνακα `faces` χρησιμοποιώντας τα `sorted_depths` σαν όρισμά του και αποθηκεύουμε το αποτέλεσμα σε νέο πίνακα `sorted_faces`.

Έπειτα, για κάθε τρίγωνο ξεχωριστά από το πρώτο μέχρι το τελευταίο ακολουθούμε δυο βήματα:

1. Με τη χρήση του `sorted_faces` παίρνουμε τις τρεις κορυφές του τριγώνου που μελετάμε και τα αντίστοιχα χρώματα.
2. Ανάλογα με τη λέξη στη μεταβλητή `renderer`, καλούμε μία από τις δυο συναρτήσεις χρωματισμού τριγώνου.

### Ψευδοκώδικας:

```
arithmos_trigonon
trinum = size(faces, 1)

calculate_depths_mesos_oros_ton_3_korufon
depths = trinum;
for i = 1:trinum
    depths(i) = (depth(faces(i,1)) + depth(faces(i,2)) + depth(faces(i,3))) / 3;

sort_depths_fthinousa_seira
sorted_depths] = sort(depths, 'descend')
```

```
sort trigona fthinousa seira
sorted_faces = faces(sorted_depths, :)

Paint each triangle.
for i = 1 to trinum
    sorted vertices trigonon
    vertices = 3 vertices of sorted_faces(i)

    colors trigonon
    colors = 3 colors of sorted_faces(i)

    Paint the triangle with the correct method.
    if renderer == 'Flat'
        Img = paint_triangle_flat(Img, vertices, colors)
    elseif renderer == 'Gouraud'
        Img = paint_triangle_gouraud(Img, vertices, colors)
```

## Script με όνομα demo\_flat/ demo\_gouraud

Τα scripts αυτά:

- φορτώνουν τα δεδομένα εισόδου από το αρχείο racoon\_hw1.mat που μας δόθηκε,
- πραγματοποιούν τον χρωματισμό των τριγώνων του αντικειμένου χρησιμοποιώντας τη συνάρτηση render (με την επιλογή του αντίστοιχου renderer, Gouraud ή Flat)
- και αποθηκεύουν το τελικό αποτέλεσμα με χρήση της συνάρτησης imwrite του Matlab



**Το αποτέλεσμα της demo\_flat:**

Παρατηρούμε ότι το χρώμα στα τρίγωνα αλλάζει απότομα. Αυτό συμβαίνει επειδή χρωματίζουμε ολόκληρο το τρίγωνο με το ίδιο χρώμα κάθε φορά.

**Το αποτέλεσμα της demo\_gouraud:**

Εδώ με τη χρήση της γραμμικής παρεμβολής κάθε τρίγωνο έχει μια πιο ομαλή μετάβαση από χρώμα σε χρώμα. Έτσι και το τελικό αποτέλεσμα φαίνεται πιο φυσιολογικό.