



Γραφική με Υπολογιστές 2024

Εργασία #1: Πλήρωση Τριγώνων

Σούρλη Ελένη

Πίνακας περιεχομένων

Συνάρτηση Γραμμικής Παρεμβολής	3
Συναρτήσεις Πλήρωσης Τριγώνων.....	4
Αλγόριθμος Bresenham.....	4
Flat shading	5
Gouraud shading.....	5
Συνάρτηση χρωματισμού αντικειμένου.....	6
Αποτέλεσμα χρωματισμένου αντικειμένου με Flat	7
Αποτέλεσμα χρωματισμένου αντικειμένου με Gouraud	8

Συνάρτηση Γραμμικής Παρεμβολής

Η συνάρτηση δέχεται σαν ορίσματα δύο διανύσματα(διανυσματικές τιμές και θέσεις) και την κατεύθυνση του σημείου στο οποίο εφαρμόζεται η γραμμική παρεμβολή. Επιστρέφει την τιμή που προκύπτει από τη μέθοδο αυτή κατά την κατεύθυνση dim, η οποία δηλώνει ανάλογα με την τιμή της, την τετμημένη x ή τεταγμένη y.

Δίνεται ο ψευδοκώδικας που χρησιμοποιήθηκε:

```
1  %vector interpolation
2  if horizontal interpolation
3      if p1(0)==p2(0)
4          V=(V1+V2)/2
5      else
6          V= V1+(coord-p1(0))*(V2-V1)/(p2(0)-p1(0))
7      end
8  elseif vertical interpolation
9      if p1(1)==p2(1)
10         V=(V1+V2)/2
11     else
12         V= V1+(coord-p1(1))*(V2-V1)/(p2(1)-p1(1))
13     end
14
```

Συναρτήσεις Πλήρωσης Τριγώνων

Δίνεται ο τροποποιημένος ψευδοκώδικας που χρησιμοποιήθηκε:

```
1  %shading triangles
2  find triangle_sides_points(Bresenham algorithm)
3  sort triangle_sides_points
4  for points=(x,y) in triangle_sides_points
5      drawpixel(x,y)
6  end
7  find ymin,ymax
8  for y=ymin:1:ymax
9      find active_points(for every side)
10     find min_col_active_points
11     find max_col_active_points
12     for x=min_col_active_points:1:max_col_active_points
13         (x,y) drawpixel(x,y)
14     end
15 end
16
```

Αλγόριθμος Bresenham

Γίνεται χρήση του αλγορίθμου Bresenham προκειμένου να καθοριστούν οι πλευρές των τριγώνων και τα σημεία τους. Πραγματοποιείται η δημιουργία ενός συνόλου σημείων που αποτελούν τη γραμμή που συνδέει δύο δοσμένα σημεία στο επίπεδο(κορυφές).

Δίνεται ο ψευδοκώδικας που χρησιμοποιήθηκε:

```
1  %Bresenham algorithm
2  DeltaX=x2-x1
3  DeltaY=y2-y1
4  x=x1
5  y=y1
6  m=DeltaY-DeltaX
7  if m>0
8      DeltaX,DeltaY=DeltaY,DeltaX
9      x,y=y,x
10     x1,y1=y1,x1
11     x2,y2=y2,x2
12     f=2*DeltaY-DeltaX
13     drawpixel(y,x)
14 end
15 else
16     f=2*DeltaY-DeltaX
17     drawpixel(x,y)
18 end
19 for i=0:1:DeltaX-1
20     if f>0
21         y=y+1(if y<y2 else y=y-1)
22         f=f+2*DeltaY-2*DeltaX
23     end
24     else
25         f=f+2*DeltaY
26         x=x+1(if x<x2 else x=x-1)
27         if m>0
28             drawpixel(y,x)
29         else
30             drawpixel(x,y)
31     end
32 end
```

Flat shading

Η συνάρτηση θα λάβει μια εικόνα (img), μια λίστα από δισδιάστατες κορυφές (vertices) και μια λίστα από χρώματα κορυφών (vcolors). Στη συνέχεια, θα ζωγραφιστεί ένα τρίγωνο πάνω στην εικόνα χρησιμοποιώντας μια επίπεδη γέμιση για δοσμένα σημεία και χρώματα. Η συνάρτηση θα επιστρέψει την τροποποιημένη εικόνα. Ο χρωματισμός του τριγώνου γίνεται ανάλογα με τα χρώματα των κορυφών του τριγώνου με ίδιο χρώμα για όλα τα σημεία του. Το κάθε χρώμα προκύπτει από τον διανυσματικό μέσο όρο των χρωμάτων των τριών κορυφών.

Gouraud shading

Η συνάρτηση θα λάβει μια εικόνα (img), μια λίστα από δισδιάστατες κορυφές (vertices) και μια λίστα από χρώματα κορυφών (vcolors). Στη συνέχεια, θα ζωγραφιστεί ένα τρίγωνο πάνω στην εικόνα χρησιμοποιώντας γέμιση για δοσμένα σημεία και χρώματα. Ο χρωματισμός κάθε σημείου του τριγώνου γίνεται με χρώμα που προκύπτει από γραμμική παρεμβολή των χρωμάτων των κορυφών του. Αρχικά, χρωματίζονται τα ενεργά σημεία για την τρέχουσα σάρωση και έπειτα τα υπόλοιπα σημεία της γραμμής σάρωσης που βρίσκονται εντός του τριγώνου.

Συνάρτηση χρωματισμού αντικειμένου

Δίνεται ο ψευδοκώδικας που χρησιμοποιήθηκε:

```
1  %render images
2  Create image with side MxN,M=512,N=512
3  for i=1:1:K
4      triangle_depth=mean(depth(top of triangles))
5  end
6  sort triangles by depth
7  if shading=='f'
8      for k=1:1:K
9          Create image with f_shading
10     end
11 else if shading=='g'
12     for k=1:1:K
13         Create image with f_shading
14     end
15
```

Η συνάρτηση θα λάβει έναν πίνακα με τις κορυφές K των χρωματισμένων τριγώνων (faces), έναν πίνακα με τις κορυφές των τριγώνων που εμπεριέχονται στην εικόνα (vertices), έναν πίνακα με τα χρώματα των κορυφών (vcolors) και ένα πίνακα με το βάθος της κάθε κορυφής (depth). Μέσω του ορίσματος shading γίνεται επιλογή ανάμεσα στις συναρτήσεις χρωματισμού Flat και Gouraud με το 'f','g' αντίστοιχα. Ο χρωματισμός ξεκινά από τα τρίγωνα με το μεγαλύτερο βάθος και συνεχίζει σε αυτά με το μικρότερο. Ο καμβάς που χρησιμοποιείται είναι $M = 512$, $N = 512$ σε λευκό χρώμα.

Αποτέλεσμα χρωματισμένου αντικειμένου με Flat

Μέσω του αρχείου `demo_f.py` φορτώνονται τα δεδομένα εισόδου (πίνακες `vertices`, `ncolors`, `faces`, `depth` της `render_img`) που δίνονται από το αρχείο `hw1.npy`. Γίνεται κλήση της συνάρτησης χρωματισμού `render_img`, η οποία πραγματοποιεί χρωματισμό του αντικειμένου με απόδοση χρώματος Flat. Η εικόνα αποθηκεύεται μέσω της συνάρτησης `imsave` με το όνομα `'shade_flat.png'`



Αποτέλεσμα χρωματισμένου αντικειμένου με Gouraud

Μέσω του αρχείου `demo_g.py` φορτώνονται τα δεδομένα εισόδου (πίνακες `vertices`, `ncolors`, `faces`, `depth` της `render_img`) που δίνονται από το αρχείο `hw1.npy`. Γίνεται κλήση της συνάρτησης χρωματισμού `render_img`, η οποία πραγματοποιεί χρωματισμό του αντικειμένου με απόδοση χρώματος Flat. Η εικόνα αποθηκεύεται μέσω της συνάρτησης `imsave` με το όνομα `'shade_gouraud.png'`.

