

Γραφική με Υπολογιστές 2020

Εργασία #1 : Πλήρωση τριγώνων

Παπαδάμ Στέφανος
ΑΕΜ: 8885

10/3/2020

1. Περιγραφή της λειτουργίας και του τρόπου κλήσης των προγραμμάτων

1.1 Συναρτήσεις `demoFlat` & `demoGouraud`

Αρχικά, οι δύο συναρτήσεις `demo` αφού καθαρίσουν το `workspace` και κλείσουν τυχόν ανοικτά παράθυρα φορτώνουν τα δεδομένα από το αρχείο `duck_hw1.mat`, δίνουν στη μεταβλητή `painter` την τιμή `Flat` ή `Gouraud` και στη συνέχεια καλούν την συνάρτηση `paintObject` με το κατάλληλο όρισμα της `painter`, για την διαδικασία του χρωματισμού των εικόνων. Μετά την εκτέλεση της συνάρτησης, επιστρέφεται και αποθηκεύεται το αποτέλεσμα της χρωματισμένης εικόνας στη μεταβλητή `image` και εμφανίζεται το αποτέλεσμα με την χρήση της συνάρτησης `imshow`. Στο τέλος των συναρτήσεων `demo` αποθηκεύεται η εικόνα στο `path` που εκτελείται η συνάρτηση με χρήση της συνάρτησης `imwrite`.

1.2 Συνάρτηση `paintObject`

Στην συνάρτηση `paintObject` αρχικοποιούνται κάποιες μεταβλητές που θα χρησιμοποιηθούν στη συνέχεια όπου συμπεριλαμβάνεται και ο πίνακας X που περιέχει την εικόνα, μεγέθους $M \times N$ (1200x1200) και αρχική τιμή 1 σε κάθε κελί του έτσι ώστε το background να χρωματιστεί άσπρο. Στην συνέχεια, δημιουργείται ο πίνακας d , ο οποίος περιέχει για κάθε ένα από τα K τρίγωνα, το μέσο βάθος των κορυφών του. Επίσης, δημιουργείται ο πίνακας `matrix` όπου αποθηκεύεται στην πρώτη στήλη του ο πίνακας d με το μέσο βάθος του κάθε τριγώνου και στις στήλες 2-4 διατηρούνται οι κορυφές του κάθε τριγώνου μέσω του πίνακα F . Έτσι στη συνέχεια με τη συνάρτηση `sortrows` ταξινομείται ο πίνακας `matrix` με βάση την πρώτη στήλη που περιέχει τα βάθη και διατηρείται με αυτόν τον τρόπο η αντιστοιχία με τις κορυφές.

Στη συνέχεια, για κάθε ένα από τα K τρίγωνα, με την σειρά που έχει οριστεί από την ταξινόμηση, καλούμε την συνάρτηση χρωματισμού τριγώνων, `triPaintFlat` ή

triPaintGouraud ανάλογα με την τιμή που έχει το όρισμα *painter*. Ως ορίσματα στην συνάρτηση αυτή, δίνονται η εικόνα *X*, οι συντεταγμένες των τριών κορυφών του τριγώνου και το χρώμα στις κορυφές αυτές.

1.3 Συναρτήσεις *triPaintFlat* & *triPaintGouraud*

Στις συναρτήσεις αυτές εκτελείται η διαδικασία χρωματισμού των τριγώνων. Συγκεκριμένα, κάθε μία εκτέλεση της, προσθέτει και ένα χρωματισμένο τρίγωνο στην συνολική εικόνα. Στην περίπτωση της *triPaintFlat*, απλά χρησιμοποιείται ο μέσος όρος των χρωμάτων των κορυφών, ενώ στην *triPaintGouraud* υπολογίζεται το χρώμα των ακμών από τη γραμμική παρεμβολή των χρωμάτων των κορυφών της αντίστοιχης πλευράς ενώ για τα εσωτερικά σημεία του τριγώνου καλείται η *colorInterp* η οποία εκτελεί γραμμική παρεμβολή μεταξύ των ενεργών σημείων για τον υπολογισμό του χρώματος κάθε pixel του εσωτερικού του τριγώνου.

2. Περιγραφή της διαδικασίας χρωματισμού τριγώνων

2.1 Συνάρτηση *triPaintFlat*

2.1.1 Περιγραφή μεταβλητών

Στην συνάρτηση αυτή δίνονται ως ορίσματα οι συντεταγμένες των τριών κορυφών (πίνακας *V* διάστασης 3×2), το χρώμα των κορυφών αυτών (πίνακας *C* διάστασης 3×3), καθώς και η υπόλοιπη εικόνα (πίνακας *X* διάστασης $1200 \times 1200 \times 3$). Στη συγκεκριμένη συνάρτηση αρχικά αρχικοποιούνται σε τρεις μεταβλητές *r,g,b* οι μέσοι όροι των χρωμάτων των κορυφών για να χρησιμοποιηθούν στη συνέχεια για το βάψιμο του τριγώνου. Έπειτα, υπολογίζονται τα *xmin*, *xmax*, *ymin*, *ymax* για κάθε πλευρά του τριγώνου και μέσω των πινάκων *ymin* και *ymax* υπολογίζουμε τη μέγιστη (*y_maximum*) και ελάχιστη τιμή (*y_minimum*) που θα λάβει το *y* για να σαρώσει όλο το τρίγωνο. Στις μεταβλητές *countYmax* και *countYmin* μετρούνται οι κορυφές που έχουν ίση *ymax* ή *ymin* τιμή αντίστοιχα έτσι ώστε να εξεταστεί η περίπτωση που υπάρχουν οριζόντιες πλευρές στο πάνω ή κάτω μέρος του τριγώνου. Στην δομή (struct) *array* αποθηκεύονται κάποιες βασικές πληροφορίες για κάθε πλευρά του τριγώνου οι οποίες θα χρειαστούν αργότερα. Συγκεκριμένα, περιλαμβάνει τα εξής:

- *X1* : Η τετμημένη της πρώτης κορυφής.
- *Y1* : Η τεταγμένη της πρώτης κορυφής.
- *X2* : Η τετμημένη της δεύτερης κορυφής.
- *Y2* : Η τεταγμένη της δεύτερης κορυφής.
- *M* : Η κλίση της πλευράς.

Η δομή *activePoints* διατηρεί τις πληροφορίες για τα δύο εκάστοτε ενεργά σημεία και συγκεκριμένα τις συντεταγμένες (*x,y*) του κάθε σημείου και τις κλίσεις (*m1,m2*) των πλευρών στις οποίες ανήκουν τα ενεργά σημεία. Ο πίνακας *activeEdges* διατηρεί δύο νούμερα τα οποία ανταποκρίνονται στις δύο ενεργές πλευρές. Έπειτα, ενημερώνονται τα στοιχεία των *activePoints* και *activeEdges* σχετικά με την τιμή του *y_minimum*.

Τέλος οι μεταβλητές *countX*, *countY*, *countM* εντοπίζουν τις περιπτώσεις που οι τρεις κορυφές του τριγώνου ανήκουν στην ίδια ευθεία.

2.1.2 Τρόπος λειτουργίας αλγορίθμου σάρωσης.

Ο αλγόριθμος σάρωσης ξεκινάει με τον έλεγχο αν οι τρεις κορυφές βρίσκονται στην ίδια ευθεία ή όχι. Στη γενική περίπτωση που οι τρεις κορυφές δεν σχηματίζουν ευθεία εκτελείται μια δομή επανάληψης με το *y* να παίρνει τιμές από το *y_minimum* έως το *y_maximum*, όπου και πραγματοποιείται ο χρωματισμός του τριγώνου με τα εξής βήματα:

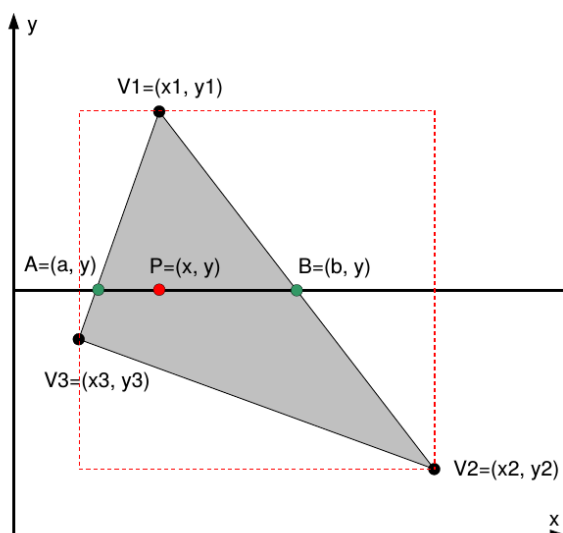
- Διατάσσεται η δομή *activePoints* με βάση την τιμή του *x* κατά αύξουσα σειρά.
- Σαρώνεται κάθε γραμμή δίνοντας τιμές στο *x* από το 1 μέχρι το 1200 το οποίο είναι και η οριζόντια διάσταση του καμβά.
- Ελέγχεται αν το *x* έχει φτάσει στο πρώτο ενεργό σημείο οπότε και αυξάνεται η τιμή του *cross_count* κατά 1 και τίθεται η σημαία *set* ίση με 1 για να γνωρίζει ο αλγόριθμος ότι το πρώτο ενεργό σημείο ξεπεράστηκε.
- Σε περίπτωση που το *cross_count* έχει περιττή τιμή σημαίνει ότι το εκάστοτε σημείο βρίσκεται ανάμεσα στα ενεργά σημεία οπότε και χρωματίζεται το αντίστοιχο *pixel* με το χρώμα που υπολογίστηκε στην αρχή δίνοντας τις κατάλληλες τιμές στα στοιχεία *X(y,x,1)* , *X(y,x,2)* , *X(y,x,3)*.
- Ελέγχεται αν το *x* έχει ξεπεράσει το δεύτερο ενεργό σημείο έτσι ώστε να βγει από την επανάληψη.
- Όταν τελειώσει η *for loop* του *x* ελέγχεται αν η επόμενη γραμμή πρόκειται να φτάσει σε σημείο όπου θα γίνει αλλαγή στις ενεργές πλευρές.
- Αν δεν ισχύει η προηγούμενη συνθήκη τότε το πρόγραμμα μεταφέρεται στο *else* και ανανεώνει τη δομή *activePoints* σύμφωνα με τον τύπο $x+1/m$ για τις τετμημένες και $y+1$ για τις τεταγμένες.
- Σε περίπτωση που η συνθήκη ισχύει και δεν υπάρχουν οριζόντιες πλευρές ($countYmax \sim= 3 \ \&\& \ countYmin \sim= 3$) σημαίνει ότι θα πρέπει να γίνει αλλαγή στις ενεργές πλευρές στον πίνακα *activeEdges* και στα ενεργά σημεία *activePoints*. Εντοπίζεται στον πίνακα *array* τα σημεία τα οποία αντιστοιχούν στη μέγιστη τιμή του πίνακα *ymin* που είναι στην ουσία και η τιμή του *y* για την πλευρά αλλαγής και αποθηκεύονται στον πίνακα *index*. Οι τιμές του πίνακα *index* είναι στην ουσία οι νέες ενεργές πλευρές. Έπειτα σχηματίζεται ένας πίνακας με τις τιμές του *activeEdges* και του *index* και μέσω της συνάρτησης *histcounts* εντοπίζεται ποιο στοιχείο έχει συχνότητα 2 το οποίο θα αντιστοιχεί και στην πλευρά που πρέπει να αφαιρεθεί. Έπειτα στον πίνακα *newActiveEdges* αποθηκεύονται οι δείκτες των νέων ενεργών πλευρών. Στο τέλος ενημερώνεται η δομή *activePoints* με βάση τις νέες υπολογισμένες τιμές που προέκυψαν και στον πίνακα *activeEdges* αποθηκεύονται οι νέες ενεργές πλευρές.
- Η διαδικασία επαναλαμβάνεται για κάθε τριάδα κορυφών.

Η παραπάνω διαδικασία συμβαίνει στη γενική περίπτωση όπου δεν υπάρχουν τριάδες κορυφών που ανήκουν στην ίδια ευθεία. Σε περίπτωση που οι τρεις κορυφές σχηματίζουν μία ευθεία τότε ο έλεγχος του προγράμματος μεταφέρεται στο *else* της αρχικής συνθήκης όπου ελέγχεται με τη σειρά αν τα σημεία βρίσκονται σε ευθεία παράλληλη στον άξονα x , στον άξονα y ή έχουν την ίδια κλίση. Στην περίπτωση που τα σημεία βρίσκονται σε ευθεία παράλληλη στον άξονα y τότε σαρώνονται αυτά τα σημεία από την ελάχιστη μέχρι τη μέγιστη τιμή του y και χρωματίζονται. Αντίστοιχη λειτουργία συμβαίνει όταν τα σημεία βρίσκονται σε ευθεία παράλληλη στον άξονα x όπου χρωματίζονται τα σημεία της ευθείας από την ελάχιστη μέχρι τη μέγιστη τιμή του x κατά μήκος του ευθύγραμμου τμήματος. Η τελευταία περίπτωση είναι αυτή όπου τα σημεία σχηματίζουν ευθεία με μία συγκεκριμένη μη μηδενική κλίση. Στη συγκεκριμένη περίπτωση σαρώνεται η ευθεία από την ελάχιστη μέχρι τη μέγιστη τιμή του y και η τιμή του x αυξάνεται κατά $1/m$ στρογγυλοποιώντας το αποτέλεσμα για να δημιουργηθεί ακέραιος αριθμός και να χρωματιστεί το αντίστοιχο pixel. Μετά την ολοκλήρωση ολόκληρης της παραπάνω διαδικασίας επιστρέφεται η ανανεωμένη εικόνα X .

2.2 Συνάρτηση triPaintGouraud

Η βασική ιδέα της συνάρτησης αυτής ταυτίζεται με την *triPaintFlat*. Συγκεκριμένα οι μεταβλητές που χρησιμοποιούνται καθώς και ο τρόπος εφαρμογής του αλγορίθμου σάρωσης ταυτίζονται με αυτούς της *triPaintFlat* όπως περιγράφονται στις παραπάνω ενότητες. Ωστόσο υπάρχουν βασικές διαφορές στον τρόπο υπολογισμού του χρώματος. Η συνάρτηση αυτή διαφοροποιείται εντελώς από την *triPaintFlat* στο σημείο υπολογισμού του χρώματος για τον χρωματισμό των pixel.

Ο τρόπος υπολογισμού του χρώματος χωρίζεται σε δύο κατηγορίες ανάλογα με το σημείο που πρόκειται να χρωματιστεί. Συγκεκριμένα, αρχικά γίνεται για κάθε γραμμή σάρωσης y μία σάρωση του καμβά κατά x έτσι ώστε να υπολογιστεί το χρώμα των πλευρών του τριγώνου. Με αναφορά στην Εικόνα 1 χρησιμοποιείται γραμμική παρεμβολή ανάμεσα στα σημεία $V1$ και $V3$ για τον υπολογισμό του χρώματος στο σημείο A και ανάμεσα στα σημεία $V1$ και $V2$ για τον υπολογισμό του χρώματος στο B . Συγκεκριμένα, μέσα στη πρώτη *for loop* του x υπολογίζονται τα χρώματα των άκρων A και B ως εξής:



Εικόνα 1: Τρόπος προσδιορισμού χρώματος μέσω γραμμικής παρεμβολής.

- Μόλις προσεγγιστεί το πρώτο ενεργό σημείο εντοπίζεται στη δομή *array* για ποιο σημείο πρόκειται και αποθηκεύεται η θέση του στη μεταβλητή *found*.
- Εφόσον είναι γνωστή η θέση του πρώτου ενεργού σημείου στην δομή *array* άρα και η πλευρά στην οποία ανήκει, αποθηκεύεται από τον πίνακα C σε μεταβλητές *r1,r2,g1,g2,b1,b2* τα χρώματα των δύο κορυφών αντίστοιχα.
- Έπειτα με βάση τις τιμές χρωμάτων που αποδόθηκαν παραπάνω στις κορυφές υπολογίζεται με γραμμική παρεμβολή το χρώμα του πρώτου ενεργού σημείου και αποθηκεύεται στον πίνακα A (3x1). Λαμβάνεται μία γενική περίπτωση όπου η πλευρά στην οποία ανήκει το σημείο έχει μία κλίση ως προς τον άξονα x και μία για την περίπτωση που η πλευρά είναι κάθετη στον άξονα x. Το χρώμα που υπολογίστηκε αποδίδεται στο κατάλληλο σημείο.
- Μόλις προσεγγιστεί και το δεύτερο ενεργό σημείο ακολουθείται η ίδια διαδικασία για να υπολογιστεί το χρώμα του δεύτερου ενεργού σημείου το οποίο αποθηκεύεται στον πίνακα B(3x1) και αποδίδεται στο pixel του δεύτερου ενεργού σημείου.

Στη συνέχεια, για τη γραμμή σάρωσης y ξεκινάει να σαρώνεται ξανά ο κανβάς κατά x για να χρωματίσει τα τρίγωνα στο εσωτερικό του εφόσον είναι γνωστά τα χρώματα των δύο ενεργών σημείων. Συγκεκριμένα γίνονται τα εξής βήματα:

- Μόλις προσεγγιστεί το πρώτο ενεργό σημείο ή ξεπεραστεί το δεύτερο αυξάνεται η μεταβλητή *cross_count* κατά 1 για να είναι γνωστό αν έχει γίνει περιττό ή άρτιο πλήθος διασχίσεων.
- Αν βρίσκεται σε περιττό πλήθος τότε καλείται η συνάρτηση *colorInterp* με ορίσματα τις τετμημένες a και b των δύο ενεργών σημείων, την τετμημένη x του σημείου που πρόκειται να χρωματιστεί και τα χρώματα A και B των δύο ενεργών σημείων που υπολογίστηκαν παραπάνω.
- Μέσα στη συνάρτηση *colorInterp* εφαρμόζεται γραμμική παρεμβολή μεταξύ των σημείων και επιστρέφεται το υπολογισμένο χρώμα μέσω του πίνακα *color* (3x1).
- Ο έλεγχος επιστρέφεται στην συνάρτηση *triPaintGouraud* και αποδίδονται οι χρωματικές συνιστώσες στο σημείο x.

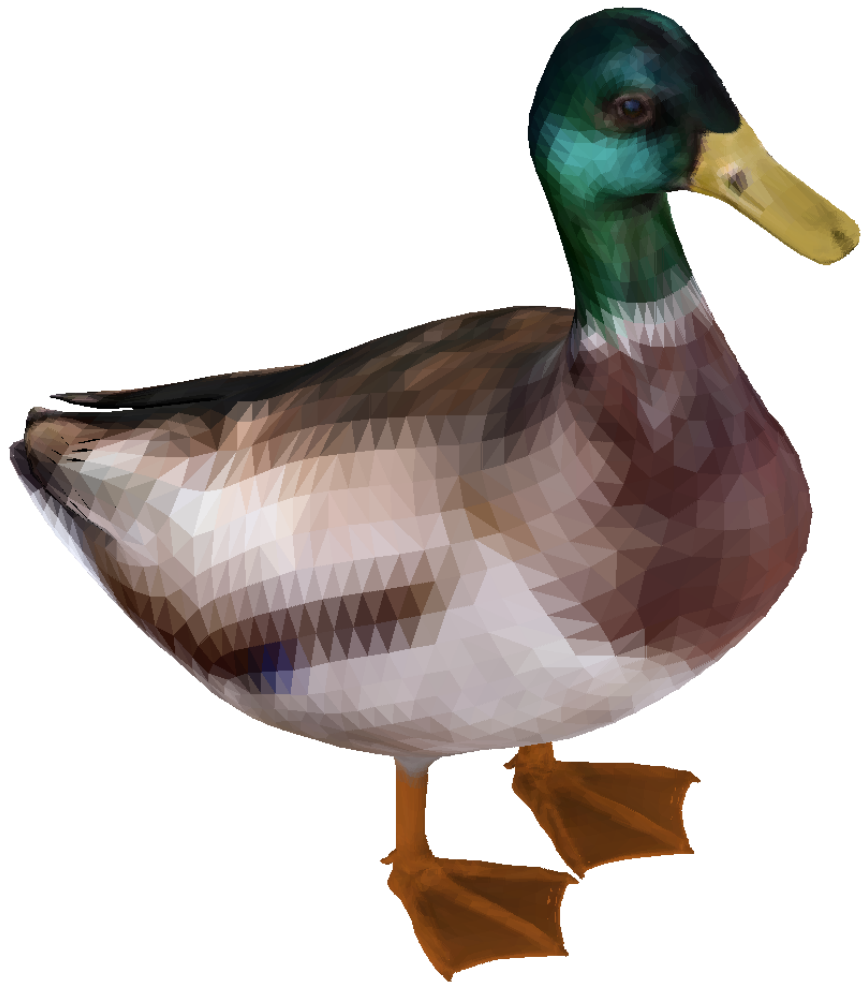
Ολόκληρη η παραπάνω διαδικασία για τα εσωτερικά σημεία και τις πλευρές επαναλαμβάνεται για κάθε γραμμή σάρωσης y.

3. Ενδεικτικά αποτελέσματα & σχόλια

3.1 demoFlat

Το αποτέλεσμα που προκύπτει από την εκτέλεση του *demoFlat* είναι ικανοποιητικό καθώς διακρίνονται όλα τα τρίγωνα να έχουν χρωματιστεί με συγκεκριμένο χρώμα και να μην υπάρχει επικάλυψη κανενός τριγώνου με άλλο. Επίσης ο χρόνος εκτέλεσης αυτής της μεθόδου είναι αρκετά ικανοποιητικός καθώς υπολογίζεται περίπου στα

3.7 - 4 δευτερόλεπτα. Το αποτέλεσμα της *demoFlat* φαίνεται στην Εικόνα 2:



Εικόνα 2: Αποτέλεσμα μετά την εκτέλεση της Flat μορφής.

3.2 demoGouraud

Το αποτέλεσμα με την εκτέλεση της *demoGouraud* φαίνεται να είναι πολύ καλύτερο οπτικά καθώς φαίνεται να σχηματίζεται μια πιο ομαλή απόχρωση στην εικόνα μετά την εφαρμογή της γραμμικής παρεμβολής χωρίς να γίνεται διαχωρισμός στα χρώματα των τριγώνων αλλά μεταβαίνοντας πιο ομαλά από το ένα τρίγωνο στο άλλο. Ο χρόνος εκτέλεσης και σε αυτή τη μέθοδο είναι αρκετά ικανοποιητικός καθώς υπολογίζεται περίπου στα 7.7 - 8 δευτερόλεπτα. Το αποτέλεσμα της *demoGouraud* φαίνεται στην Εικόνα 3:



Εικόνα 3: Αποτέλεσμα μετά την εκτέλεση της Gouraud μορφής.