Γραφική με Η/Υ
Εργασια 1
Πλήρωση Τριγώνων

Μπεκιάρης Θεοφάνης AEM:8200

Περιγραφή της λειτουργίας και του τρόπου κλήσης των προγραμμάτων.
Τα προγράμματα που περιέχονται στην εργασίας είναι όπως ακριβώς ζητούνται στην εκφώνηση της.Στην εργασία περιέχονται οι συναρτήσεις Tripaint,findColor,findColor2,Painter καθώς και ένα script με τα δεδομένα σημείων,τριγώνων και χρωμάτων που αναφέρονται στην εκφώνηση για την δημιουργία της εικόνας MxN.Τελός για να εκτελεστούν τα προγράμματα και να παραχθούν οι εικόνες με την χρήση των αντίστοιχων αλγορίθμων αρκεί η εκτέλεση των scripts με ονόματα demo1A, demo1B, demo1C.

Περιγραφή της διαδικασίας χρωματισμού των τριγώνων .
Τα τρία προγράμματα που βρίσκονται στις αντίστοιχες Tripaint διαφέρουν κατα κύριο λόγο στον τρόπο με τον οποίο υπολογίζουν τα ενεργά σημεία των τριγώνων προς πλήρωση-χρωματισμό.

Το πρόγραμα της TripaintA υπολογίζει τα ενεργά σημεία επιλύοντας απευθείας τις εξισώσεις των πλευρών με βαση τις εξισώσεις

$$y=1/m*x+b \text{ και } y=y\_scanline \text{ οπου } d*m=y1*m-x1$$

και (x1,y1) σημεία της πλευράς.

Το πρόγραμα της TripaintB υπολογίζει αναδρομικά τα ενεργά σημεία σύμφωνα με τις εξισώσεις των σημειώσεων του μαθήματος και πιο συγκεκριμένα της σχέσης

$$d_{i+1}=d_i+m$$

οπου di τα ενεργά σημεία και m απο την σχέση $y=1/m*x+b$ των εξισώσεων των πλευρών.

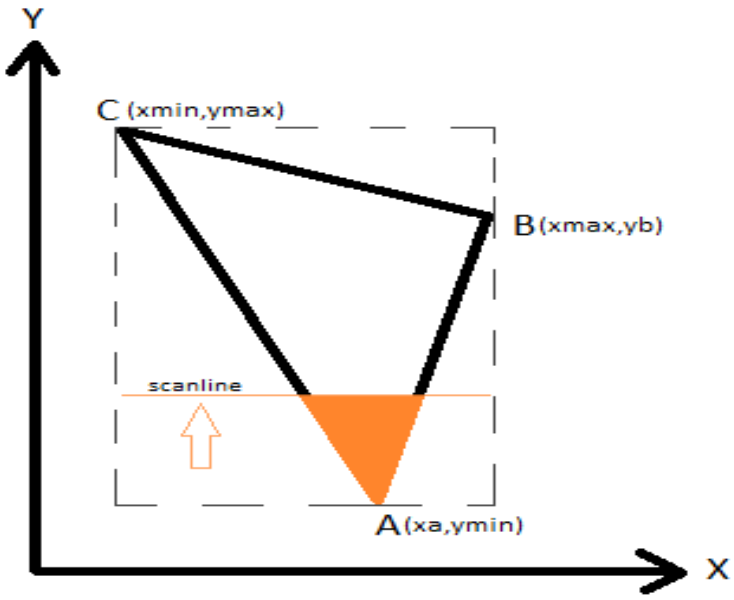
Το πρόγραμα της TripaintC υπολογίζει τα ενεργά σημεία χρησιμοποιόντας την λογική που περιγράφεται στην εκφώνηση της εργασίας.Μετασχηματίζουμε τα ενεργά σημεία μέσω της σχέσης $x'=x*\Delta y$ και υπολογίζουμε τα επόμενα αναδρομικά απο τον τύπο $x'_{y+1} = x'_y + \Delta x$. Τέλος σαρώνουμε την κάθε scanline γραμμή και εντοπίζουμε το εσωτερικό του τριγώνου το οποίο χρωματίζεται με την βοήθεια των σχέσεων.

$$x' \geq x'_y \text{ , εάν } \Delta y > 0$$
$$x' \leq x'_y \text{ , εάν } \Delta y < 0$$

Αναλυτικότερα για τους αλγόριθμους

Και τα 3 προγράμματα ακολουθούν την λογική του αλγορίθμου που βρίσκεται στις σημειώσεις,δηλαδή για κάθε scanline υπολογίζουμε τα ενεργά σημεία(ή τα μετασχματισμένα για την περίπτωση της TripaintC),σαρώνουμε την scanline και τα σημεία ανάμεσα στα ενεργά σημεία τα χρωματίζουμε.Σε σχέση ομως με τον αλγόριθμο των σημειώσεων που αντιστοιχεί σε πολύγωνα εφόσον γνωρίζουμε οτι εχουμε τρίγωνο υπάρχουν μερικές διαφορές ως προς τα όρια της εικόνας στα οποία θα τρέχει ο αλγόριθμος.Συγκερκιμένα αντί να σαρώνουμε όλη την εικόνα θα κινούμαστε

μόνο μέσα στα όρια των xmin,xmax,ymin,ymax των σημείων του τριγώνου και επιπλέον αντί να κάνουμε συνεχώς έλεγχο για τις ενεργές πλευρές επειδή έχουμε μόλις 3 πλευρές μπορούμε να υπόλογίσουμε εύκολα οτι η ανανέωση θα γίνει στο σημείο με ενδίαμεσο υψος y(αφου το scanline σαρώνει κατα y) σε σχέση με τα άλλα 2 σημεία του τριγώνου.Για την καλύτερη εξήγηση του αλγορίθμου θα χρησιμοποιήσουμε ως αναφορά το παρακάτω σχήμα.



Με γενική περιγραφή ο αλγόριθμος των Tripaint έχει ως εξής:
1)Αρχικά βρίσκουμε το σημείο που του αντιστοιχεί το ymin οπου στο σχήμα ειναι το σημείο A
2)Φτιάχνουμε τον πίνακα ενεργών ακμών που θα είναι ο

EnergesAkmes=[AC AB]=|xa ya xa ya|
|xc yc xb yb|

και των πίνακα ενεργών χρωμάτων XrwmataShm=|Ca Ca;Cc Cb|$_{2x2x3}$.
3)Βρίσκουμε το σημείο που θα γίνει η ανανέωση των ακμών οπου στο σχήμα ειναι το B και αποθηκέυουμε την τιμή της τεταγμένης yb στην μεταβλήτή a την οποία θα χρησιμοποιήσουμε αργότερα για έλεγχο των ιδικών περιπτώσεων για οριζόντιες πλευρές.
4)Ορίζουμε τον πίνακα ενεργών σημείων EnergShmeia και το αρχικοποιούμε με την τετμημένη του χαμηλότερου σημειου A.Στην περίπτωση όπου ξεκινάμε με οριζόντια πλευρά (στον κώδικα θα πρέπει a=yimn οπου a η τεταγμένη του σημείου B)αντικαθιστούμε τον πίνακα ενεργών ακμών κατάλληλα,για παράδειγμα αν η πλευρά AB είναι οριζόντια τότε ο πίνακα γίνεται EnergesAkmes=[AC BC].
5)Σαρώνουμε κάθε scanline γραμμή και χρωματίζουμε μέχρι το ύψος του σημείο B.
6)Ελέγχουμε αν έχουμε οριζόντια γραμμή για ymax,δηλαδή αν για παράδειγμα στο σχήμα έχουμε τα C και B στο ίδιο ύψος (στον κλώδικα ελέγχουμε άν a=ymax οπου a η τεταγμένη yb του B)και άν αυτό ισχύει σταματάμε αλλιώς συνεχίζουμε το χρωματισμό μέχρι το σημείο C και ο αλγόριθμος τελειώνει.

Παραδοχές

-Θεωρούμε ότι οι δεκαδικές τιμές των θέσεων των pixel του τριγώνου που είναι προς χρωματισμό στρογυλλοποιούνται πάντα προς τα κάτω με την συνάρτηση floor.Για τα όρια των πλευρών των τριγώνων αυτό σημαίνει ότι για 2 γειτονικά τρίγωνα το τρίγωνο που βρίσκονται πρός μεγαλύτερες τιμές θα καταλαμβάνει την κοινή πλευρά.
-Στα δεδομένα που δίνοται στην εκφώνηση θεωρώ οτι ο πίνακας T των σημείων είναι της μορφής [x1 y1;x2 y2; ….] και οι διαστάσεις τις είκονας είναι για x N=400 και για y M=300.

Γραμμική παρεμβολή για τον υπολογισμό των χρωμάτων

Όπως αναφέρεται και στην εκφώνιση της εργασίας θα πρέπει να υπολογίζουμε κάθε φορά για κάθε scanline τις χρωματικές συνιστώσες CA και CB με γραμμική παραμβολή και η σχέση της είναι

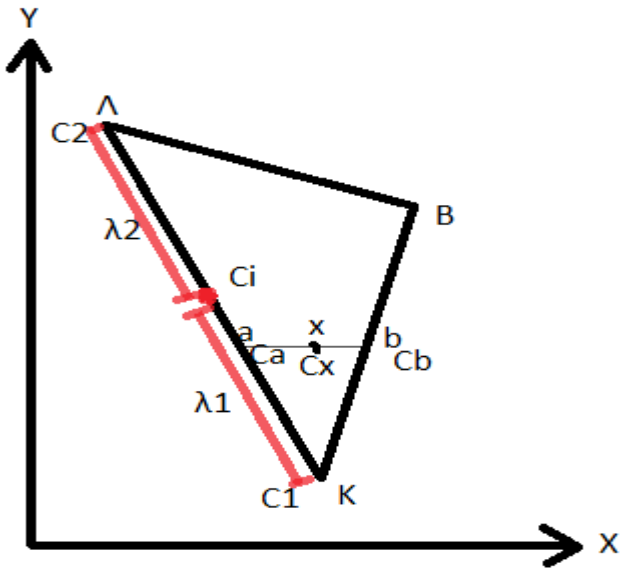
$$Ci=(\lambda1*C2+\lambda2*C1)/(\lambda1+\lambda2)$$

οπου αν για παράδειγμα έχουμε την πλευρά ΚΛ και C1 η χρωματική συνιστώσα της Κ κορυφής και C2 της Λ τότε λ1 η απόσταση του τρέχον σημείου i με συνιστώσα Ci απο το Κ και λ2 η αποστασή του απο το Λ. Οι συναρτήσεις findColor υπολογίζουν τα χρώματα απο τα αντίστοιχα CA και CB με την βοήθεια του αναδρομικού τύπου που προκλυπτεί απο την αντικάτάσταση των λ1 και λ2 στην παραπάνω σχέση με τις παραστασεις απόστασης που περιέχουν τα x,y και χρησιμοποιόντας την σχέση X(i+1)=X(i)+1 μεταξύ διαδοχικών σημείων πάνω στην ίδια γραμμή scanline.Ο αναδρομικός τύπος που προκύπτει είναι

$$Ci_{+1}=Ci+(CB-CA)/(a+b)$$

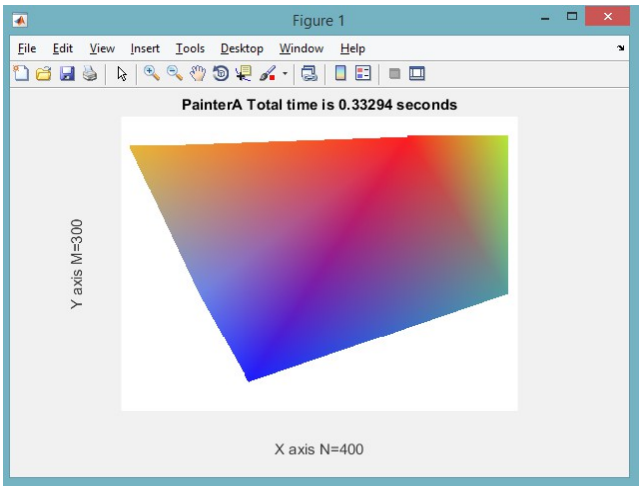
οπου a,xb οι τετμημένες των ενεργών σημείων με CA και CB αντίστοιχα.Αρα για αρχική τιμή ίση με CA και για σημείο της scanline που βρίσκεται στην θέση x σε απόσταση x-a απο το σημείο a αυτό σημαίνει ότι πρέπει να πρόσθέσουμε x-a φορές των αναδρομικό όρο (CB-CA)/(a+b) στην αρχική τιμή CA.Τελικά βλέπουμε ότι για ένα σημείο x η χρωματική του συνιστώσα είναι.

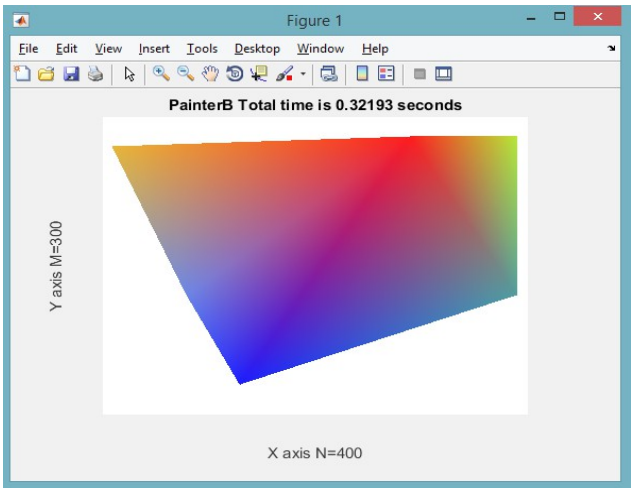
$$Cx=CA+(x-a)*(CB-CA)/(a+b)$$

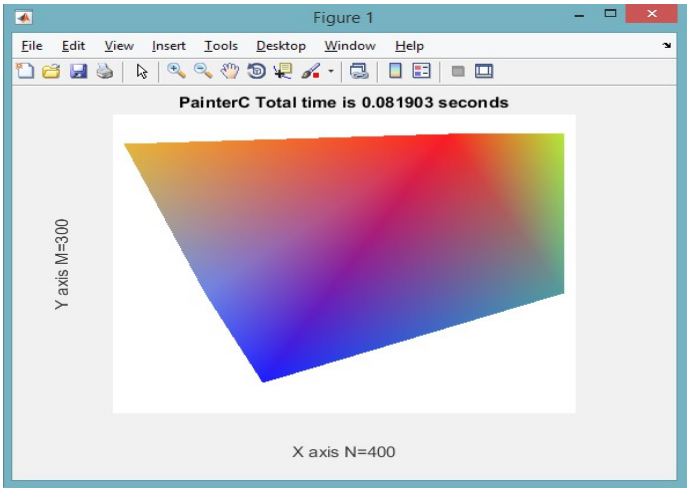


Αποτελέσματα και σχόλια

Μετά απο την εκτέλεση των 3 αλγορίθμων με τα δεδομένα που δίνονται στην εκφώνηση η εικόνα που προκύπτει και για τις 3 περιπτώσεις χωρίς κάποια διαφορά μεταξύ των 3 υλοποιήσεων είναι η εξής.







Όσον αφορά την ποιότητα των αποτελεσμάτων οι 3 αλγόριθμοι δεν φαίνεται να έχουν καμία διαφορά στην παραγώμενη εικόνα.Αν και ο αλγόριθμος της TripaintB μπορεί να εισάγει σφάλματα στον υπολογισμό των ενεργών σημείων λόγο στρογυλλοποιήσεων και αδυναμία για πλήρη αναπαράσταση ενός δεκαδικού αριθμού,το σφαλμά αυτό είναι πολύ μικρό για να φανεί σε εικόνα στις διαστάσεις της οθόνης ενός υπολογιστή.Για παράδειγμα ένα σφάλμα έκτου δεκαδικού ψηφίου 0,000001 θα πρέπει να πολλαπλασιαστεί με το ενα εκατομμύριο 1000000*0.00000=1 για να προκύψει απόκλιση 1 μονάδας,δηλάδή σε μία οθόνη πλάτους ενός εκατομμυρίου pixel (εννοούμε ότι η μία διασταση θα έχει ενα εκατομμύριο pixel) θα προκύψει σφάλμα σε ένα pixel.Βέβαια εάν η ακρίβεια των δεκαδικών ψηφίων είναι πολύ μικρότερη και το μέγεθος της οθόνης μεγάλο τότε το σφάλμα μπορεί να είναι συμαντικό.Οι χρόνοι εκτέλεσης των 2 πρώτων προγραμμάτων για τα δεδομενά και το μέγεθος εικόνας που ζητούνται δεν έχουν κάποια συμαντική διαφορά αφού τα προγράμματα διαφέρουν μόνος ως προς τον τρόπο υπολογισμού των ενεργών σημείων και το υπόλοιπο κομμάτι τους είναι ίδιο.Ο πρώτος αλγόριθμος μπορεί να εκτελεί περισσότερες πράξεις για τον υπολογισμό των ενεργών σημείων αλλά η διαφορά στους χρόνους είναι μικρή για να φανεί σε τόσο μικρό πλήθος επαναλήψεων .Ο τρίτος αλγόριθμος διαφέρει απο τους 2 προηγούμενους σε αρκετά σημεία και για αυτο ο χρόνος εκτέλεσης του διαφέρει και αυτός επίσης.Φαίνεται οτι η εκτέλεση του είναι πιο γρήγορη για τα ζητούμενα δεδομένα και ο πιθανός λόγος είναι ότι διαχειρίζεται μόνο ακέραιους αριθμούς οπου η διαχείρηση τους(πράξεις προσθεσης,πολλαπλασιασμου κλπ) απο τον υπολογιστή είναι πολύ πιο απλή και γρήγορη σε σχέση με τους δεκαδικούς αριθμούς που χρησιμοποιούν τα προηγούμενα προγράμματα.

Αλλάζοντας τα δεδομένα και το μέγεθος της εικόνας σε M=3000 και N=4000 και εκτελώντας ξανά τα προγράμματα βλέπουμε οτι ο χρόνος εκτέλεσης της TripaintB ειναι πιό μικρός σε σχέση με την TripaintA για μεγαλύτερο πλήθος δεδομένων και ο χρόνος της TripaintC αυξάνει με μεγάλη διαφορά σε σχέση με τις άλλες 2.Αυτο συμβαίνει επειδή στην υλοποίηση της TripaintC η σάρωση της κάθε γραμμής γίνεται απο το xmin εως και το xmax ενώ στις άλλες 2 υλοποιήσεις δεν σκανάρουμε όλα τα x της κάθε γραμμής αλλά πάμε απευθείας και χρωματίζουμε ανάμεσα στα ενεργά σημεία γλυτώνοντας σημαντικό χρόνο από περίττους ελέγχους και υπολογισμούς.