Τμήμα Μηχανικών Η/Υ &Πληροφορικής

Γραφικά Υπολογιστών και Συστήματα Αλληλεπίδρασης

Ακαδημαϊκό Έτος 2017-2018

Πανεπιστήμιο Ιωαννίνων

**ΑΣΚΗΣΕΙΣ διαφανειών 1 και 3**

**Θέμα 1ο Υλικό:** Ποια είναι η κύρια διαφορά μεταξύ μιας οθόνης πλεγματικής σάρωσης και μιας οθόνης διανυσματικής σάρωσης, σε ποιες σκηνές δεν μπορεί να ανταποκριθεί η οθόνη διανυσματικής σάρωσης;

**Θέμα 2ο Υλικό:** Θέλετε να απεικονίσετε γραφικά σε ανάλυση 1024 x 768 (1024 στήλες και 768 γραμμές) με βάθος χρωμάτων 32 bit και με ρυθμό ανανέωσης 100Ηz. Αν οι χρόνοι οριζόντιας και κάθετης επαναφοράς είναι αντίστοιχα 2μsec και 1500μsec βρείτε το ελάχιστο εύρος της οθόνης που μπορεί να υποστηρίξει το παραπάνω σχήμα σε (α) διαπλεγμένη και (β) μη διαπλεγμένη λειτουργία. Επίσης (γ) ποιο είναι ένα σχήμα υλικού που μπορεί να υποστηρίξει την μη διαπλεγμένη λειτουργία αφήνοντας την μνήμη video για περισσότερο από 50% του χρόνου ελεύθερη για εγγραφή από την κεντρική μονάδα αν υποθέσουμε ότι έχουμε μνήμη video με χρόνο προσπέλασης 100 nsec;

**Θέμα 3ο Μετατροπή σε εικονοστοιχεία:** Σε ποιες περιπτώσεις λειτουργεί σωστά ο παρακάτω αλγόριθμος μετατροπής ενός ευθυγράμμου τμήματος σε εικονοστοιχεία;

**line(x1, y1, x2, y2, color)**

**int x1, y1, x2, y2, color;**

**{ int d, x, y, dx, dy;**

**dx= x2-x1; dy= y2-y1; d= 2\*dy-dx;**

**incrE= 2dy; incrNE= 2\*(dy-dx);**

**for (x=x1,y=y1; x<=x2; x++) {**

**setpixel(x, y, color);**

**if (d<=0) d+= incrE;**

**else d+= incrNE, y++;**

**}**

**}**

**Θέμα 4ο Μετατροπή σε εικονοστοιχεία:** Τροποποιήστε τον παραπάνω αλγόριθμο ώστε να λειτουργεί σωστά σε όλες τις περιπτώσεις.

**Θέμα 5ο Μετατροπή σε εικονοστοιχεία:** Αποδείξτε ότι οι αλγόριθμοι 1 και 2 παρακάτω παράγουν το ίδιο αποτέλεσμα. Ποιος από τους δύο είναι πιο αποδοτικός και γιατί; Πόσες φορές περιμένετε να είναι κατά μέσο όρο πιο αποδοτικός; Επιβεβαιώστε υλοποιώντας τους σε γλώσσα C όπου η ρουτίνα setpixel ορίζεται ως εξής:

int screen[768][1024];

setpixel(x, y, color)

int x, y, color;

{ screen[x][y]=color;

}

Αλγόριθμος 1:

**circle(int color) {**

**int x, y, R; double d;**

**x= 0; y= R; d= 5.0/4.0-R;**

**symmetric\_points(x, y, color);**

**while (y>=x) {**

**if (d<=0) d+= 2.0\*x + 3.0, x++;**

**else d+= 2.0\*(x-y) + 5.0, x++, y--;**

**symmetric\_points(x, y, color);**

**}**

**}**

Αλγόριθμος 2:

**circle(int color) {**

**int x, y, h, R;**

**x= 0; y= R; h= 1-R;**

**symmetric\_points(x, y, color);**

**while (y>=x) {**

**if (h<0) h+= 2\*x + 3, x++;**

**else h+= 2\*(x-y) + 5, x++, y--;**

**symmetric\_points(x, y, color);**

**}**

**}**

**Θέμα 6ο Μετατροπή σε εικονοστοιχεία:** Δώστε τον τελικό αλγόριθμο μετατροπής μιας έλλειψης σε εικονοστοιχεία με ακέραια αριθμητική και σταθερές διαφορές 2ου βαθμού.