Ανάλογα με τη μορφή των πολυγώνων της εφαρμογής επιλέγεται ο αντίστοιχος αλγόριθμος των 4 ή 8 γειτονικών σημείων.

Αλγόριθμος 2. Γέμισμα με βάση το υπάρχον χρώμα (interior fill)

Η μόνη διαφορά από τον προηγούμενο αλγόριθμο είναι ότι αντί για το χρώμα της περιμέτρου δίνεται στην είσοδο το υπάρχον εσωτερικό χρώμα του πολυγώνου και επομένως η αναδρομή σταματάει όταν φτάσουμε σε ένα pixel που δεν έχει το χρώμα αυτό. Στο πρόγραμμα η παράμετρος bound_color μπορεί να μετονομαστεί σε interior_color και η μόνη αλλαγή που χρειάζεται να γίνει στον κώδικα είναι η μετατροπή της συνθήκης σε if (current_color = =interior_color).

Οι παραπάνω αλγόριθμοι έχουν το μειονέκτημα ότι για κάθε pixel που χρωματίζεται γίνονται 4 ή 8 αναδρομικές κλήσεις με συνέπεια να έχουν χαμηλή ταχύτητα και εύκολα να τελειώνει ο χώρος στοίβας του υπολογιστή.

Σάρωση πολυγώνου (scan conversion)

Οι αλγόριθμοι σάρωσης στηρίζονται στην εύρεση των σημείων τομής της περιμέτρου του πολυγώνου με τις οριζόντιες γραμμές σάρωσης της οθόνης. Στην συνέχεια θα περιγραφούν αλγόριθμοι σάρωσης, που αφορούν το χρωματισμό ενός ή περισσοτέρων (επικαλυπτόμενων ή μη) πολυγώνων. Η διαπίστωση του εάν ένα σημείο (x, y) είναι εντός ενός πολυγώνου γίνεται ως εξής:

Μετράμε το πλήθος των σημείων τομής της γραμμής σάρωσης της οθόνης, από τη θέση (x, y) μέχρι τη θέση $(-\infty, y)$, με την περίμετρο του πολυγώνου και εφόσον αυτό είναι περιττός αριθμός τότε το σημείο (x, y) είναι εντός, διαφορετικά, αν είναι άρτιος, το σημείο είναι εκτός του πολυγώνου.

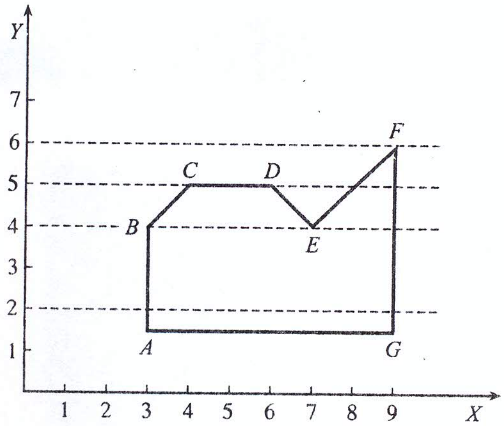
(YX)–Αλγόριθμος

Βήμα 1: Για κάθε πλευρά του πολυγώνου βρίσκουμε όλα τα σημεία τομής της με τις γραμμές σάρωσης. Έτσι κατασκευάζεται μια λίστα (x, y) σημείων τομής.

Βήμα 2: Ταξινόμηση της προηγούμενης λίστας με κύριο κλειδί το y και δευτερεύον το x.

Βήμα 3: Απομάκρυνση κατά ζεύγη των στοιχείων από τη λίστα και χρωματισμός των pixels του ευθύγραμμου τμήματος που προσδιορίζεται από κάθε ζεύγος τομών με το χρώμα του πολυγώνου π.χ. για τα γειτονικά σημεία τομής (x_1, y_1) και (x_2, y_2) θα ισχύει $y_1 = y_2$ και $x_1 \leq x_2$ και ο χρωματισμός γίνεται για τα pixels (x, y), όπου $y = y_1$ και $x_1 \leq x \leq x_2$.

Ένα πρόβλημα που παρουσιάζεται κατά τον προσδιορισμό των σημείων τομής φαίνεται στο σχήμα 2.10.



Σχήμα 2.10: Πρόβλημα σημείων τομής (*YX*).

Οι κορυφές *B, C, D, E* και *F* βρίσκονται πάνω σε γραμμές σάρωσης. Το ερώτημα εδώ είναι πόσα σημεία τομής θα θεωρούμε για κάθε τέτοια κορυφή του πολυγώνου. Αυτό αντιμετωπίζεται με τις 2 παρακάτω απλές παραδοχές.

(1) Κάθε μη οριζόντια πλευρά θεωρείται ανοιχτή στο άκρο με τη μεγαλύτερη y– τιμή και κλειστή στο άκρο με τη μικρότερη y– τιμή. Δηλαδή η τομή μιας γραμμής σάρωσης με το κάτω άκρο μιας πλευράς φυλάσσεται ενώ η τομή γραμμής σάρωσης με το άνω άκρο δεν φυλάσσεται.

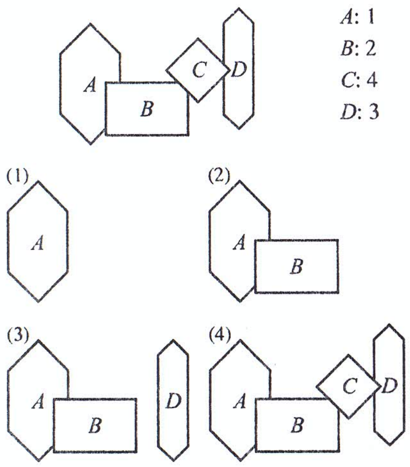
(2) Οι οριζόντιες πλευρές αγνοούνται.

Έτσι στην κορυφή *B* φυλάσσεται μία τομή (της γραμμής σάρωσης 4 με την πλευρά *BC*),στην κορυφή *E* δύο τομές (της γραμμής σάρωσης 4 με τις πλευρές *DE* και *EF*) και στην κορυφή *F* καμιά τομή (αφού το *F* είναι το άνω άκρο των 2 πλευρών που καταλήγουν στην *F*). Η πλευρά *CD* αγνοείται αφού είναι οριζόντια.

Ο (*YX*)-αλγόριθμος μπορεί να χρησιμοποιηθεί και όταν πρόκειται να σχεδιαστούν περισσότερα του ενός πολύγωνα στην οθόνη, τα οποία πιθανώς επικαλύπτονται. Στην προκειμένη περίπτωση θεωρούμε ότι τα πολύγωνα έχουν προτεραιότητα σχεδίασης, δηλ. για παράδειγμα το πολύγωνο με προτεραιότητα 1 προηγείται στην σχεδίαση από εκείνο με προτεραιότητα 2 (σχήμα 2.11). Η μέθοδος αυτή είναι γνωστή με το όνομα *αλγόριθμος του ζωγράφου* επειδή αναπαριστά τον τρόπο εργασίας ενός ζωγράφου.

Ζωγράφο

Η πραγματοποίηση αυτού του αλγορίθμου γίνεται ως εξής: Στην αρχή τίθεται η οθόνη στο χρώμα του φόντου (background), στη συνέχεια γίνεται σάρωση στα πολύγωνα αρχίζοντας με το πολύγωνο της μικρότερης προτεραιό-



Σχήμα 2.11: P–(YX) αλγόριθμος.

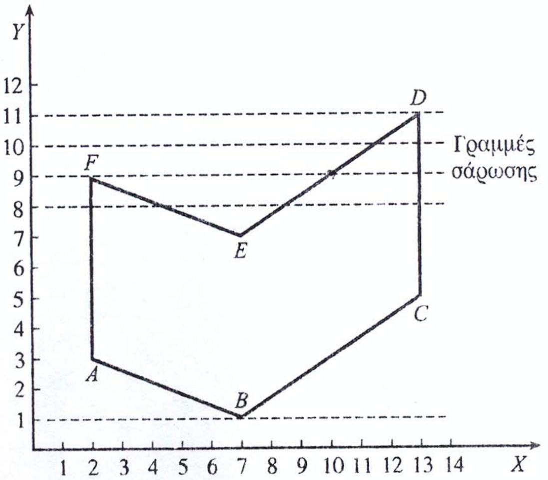
τητας. Επομένως πολύγωνα υψηλότερης προτεραιότητας θα καλύπτουν πολύγωνα χαμηλότερης προτεραιότητας εκεί όπου αυτά επικαλύπτονται. Ο αλγόριθμος του ζωγράφου είναι ένας P-(YX) αλγόριθμος γιατί απαιτούνται 3 ταξινομήσεις: πρώτο κλειδί η προτεραιότητα των πολυγώνων, δεύτερο κλειδί η y-συντεταγμένη και τρίτο κλειδί η x-συντεταγμένη των σημείων τομής.

Y–X–αλγόριθμος

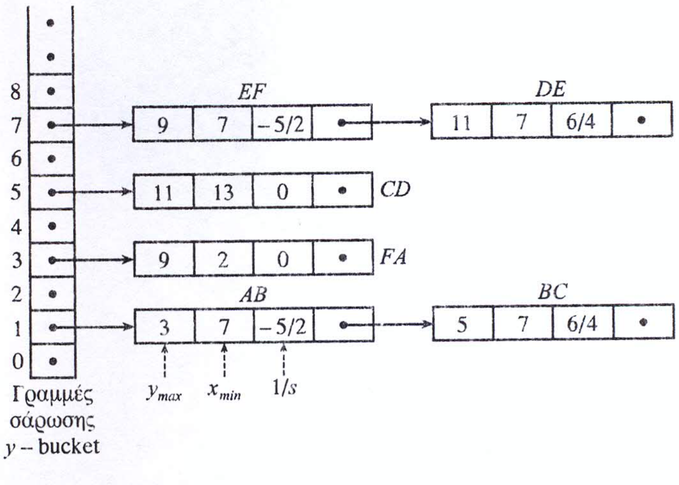
Ο αλγόριθμος αυτός βελτιώνει τον (YX)-αλγόριθμο για τρεις λόγους: (1) τα σημεία τομής γραμμής σάρωσης με πλευρά πολυγώνου υπολογίζονται γρήγορα, για διαδοχικές γραμμές σάρωσης με βάση την κλίση της πλευράς, (2) το κόστος της ταξινόμησης μειώνεται εκμεταλευόμενοι τις ιδιότητες της συνάφειας του πολυγώνου, (3) λαμβάνονται υπόψη μόνο οι τομές με γραμμές σάρωσης που βρίσκονται μεταξύ της ελάχιστης και μέγιστης y-συντεταγμένης του πολυγώνου. Τα βασικά βήματα του αλγορίθμου είναι τα εξής:

Βήμα 1: Ταξινόμηση των πλευρών του πολυγώνου σύμφωνα με τις ελάχιστες y-τιμές τους. Η δομή που χρησιμοποιείται είναι ένα bucket list στο οποίο η κάθε λίστα αντιστοιχεί σε μια γραμμή σάρωσης. Η δομή αυτή ονομάζεται πίνακας πλευρών. Τα στοιχεία της κάθε λίστας είναι οι απαραίτητες πληροφορίες για τις πλευρές που ξεκινούν (ελάχιστο y) από εκείνη την γραμμή σάρωσης (σχήμα 2.12 και σχήμα 2.13).

Βήμα 2: Για κάθε γραμμή σάρωσης y_i από την ελάχιστη ως τη μέγιστη y-τιμή του πολυγώνου επαναλαμβάνεται η εξής διαδικασία.



Σχήμα 2.12: $Y-X$ αλγόριθμος.

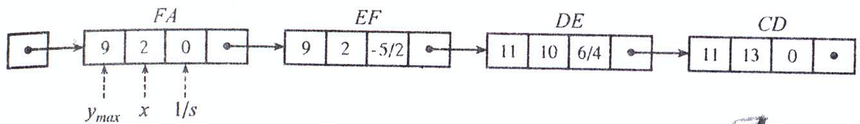


Σχήμα 2.13: Πίνακας πλευρών $Y-X$.

Βήμα 2.1: Ενημέρωσε τη λίστα ενεργών πλευρών (*ΛΕΠ*) (σχήμα 2.14) από το y-bucket του πίνακα πλευρών, που αντιστοιχεί στην παρούσα γραμμή σάρωσης, τοποθετώντας τα νεοεισερχόμενα στη *ΛΕΠ* στοιχεία στη σωστή τους θέση ως προς την x-συντεταγμένη. Ταξινόμησε την *ΛΕΠ* αν χρειάζεται.

Βήμα 2.2: Χρωμάτισε τα τμήματα της γραμμής σάρωσης που είναι εσωτερικά του πολυγώνου λαμβάνοντας ζεύγη σημείων τομής από τη *ΛΕΠ*.

$\Delta m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{9-3}{2-2} = \infty \quad \frac{1}{\infty} = 0$



(α) για γραμμή σάρωσης 9

$\frac{11-7}{13-7} = \frac{4}{6} = s \quad \frac{1}{s} = \frac{6}{4}$

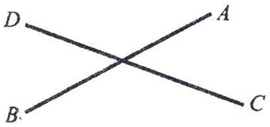
(β) για γραμμή σάρωσης 10

Σχήμα 2.14: Λίστα ενεργών πλευρών $Y-X$.

Βήμα 2.3: Αφαίρεσε από τη *ΛΕΠ* στοιχεία, των οποίων η τιμή y_{max} είναι ίση με τη γραμμή σάρωσης.

Βήμα 2.4: Υπολόγισε την τιμή x των στοιχείων της *ΛΕΠ* για την επόμενη γραμμή σάρωσης ($x = x + 1/s$). Το $1/s$ είναι πληροφορία που βρίσκεται στα στοιχεία της *ΛΕΠ*.

Ο $Y-X$-αλγόριθμος μπορεί εύκολα να μετατραπεί σε έναν $Y-(PX)$-αλγόριθμο, έτσι ώστε να γίνεται σάρωση με σκοπό το χρωματισμό ενός συνόλου πολυγώνων, λαμβάνοντας υπόψη σε κάθε γραμμή σάρωσης τις προτεραιότητες. Η *ΛΕΠ* θα περιέχει εγγραφές για όλα τα πολύγωνα που τέμνουν την παρούσα γραμμή σάρωσης, οι οποίες ταξινομούνται πρώτα με βάση την προτεραιότητα και στη συνέχεια με βάση την x-τιμή. Κατά κανόνα λόγω της ιδιότητας της συνάφειας (scan line coherence) σπάνια θα χρειαστεί επαναταξινόμηση της *ΛΕΠ* από γραμμή σάρωσης σε γραμμή σάρωσης (βλ. Βήμα 2.1) καθώς καταστάσεις της ακόλουθης μορφής είναι σπάνιες



Αλγόριθμος Κρίσιμων Σημείων

Ο αλγόριθμος αυτός είναι τύπου $Y-X$ και στηρίζεται στη δημιουργία και ενημέρωση της *ΛΕΠ* χρησιμοποιώντας την έννοια των κρίσιμων σημείων αποφεύγοντας τελείως την κατασκευή του πίνακα πλευρών. Κρίσιμο σημείο θεωρείται κάθε κορυφή πολυγώνου της οποίας η y-συντεταγμένη είναι μικρότερη από τις y-συντεταγμένες των εκατέρωθεν γειτονικών της κορυφών (τοπικό ελάχιστο). Αποτέλεσμα είναι η βελτίωση της πολυπλοκότητας σε σχέση με τον προηγούμενο $Y-X$ αλγόριθμο. Τα βασικά βήματα του αλγορίθμου είναι τα εξής (βλ. σχήμα 2.15):