Algoritam popune ivica

Andrija Cvetković 72RN

Neefikasan algoritam koji ima onoliko iteracija koliko ne-horizontalnih linija (ivica).

Bazična verzija ovog algoritma „gleda“ ceo poligon, bira početnu ivicu i kreće se udesno.

Nije bitan redosled odabira narednih ivica ali jeste bitno da se svaka ivica obradi samo jednom.

Uglavnom se ivice obrađuju redosledom: ivica koja sadrži teme trenutne ivice a nije ranije obrađena.

Algoritam, ponašanje kad je ivica izabrana:

* Svi pikseli čije je središte desno od preseka skan linije sa trenutno izabranom ivicom se komplementiraju (ukoliko su obojeni, postaju transparentni (prozirni) i obratno).

Stanje jednog piksela se može promeniti proizvoljan broj puta (onoliko puta koliko se on nalazi desno od svih ivica poligona).

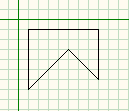
Nakon poslednjeg koraka (obrade svih ivica), svi pikseli unutar poligona su popunjeni (obojeni) a svi van ostaju transparentni.

Modifikacije algoritma:

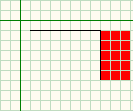
* Iteracija ulevo, bez ikakvog poboljšanja u pogledu optimizacije.
* Dodavanje granice u središte poligona (deljenje poligona na dva dela po vertikali). Poželjno da ta granica prolazi kroz neku od tačaka poligona. Algoritam u tom slučaju smanjuje broj promena stanja istog piksela. Leva i desna strana poligona se nezavisno obrađuju. Leva strana poligona se obrađuje udesno a desna ulevo. Moguće je i obratno ali bi onda u algoritam ulazili i rezultati granične linije (algoritam će i dalje raditi korektno).

Primer bazične verzije algoritma:

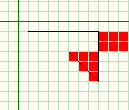
* Poligon nad kojim će algoritam biti izvršen:



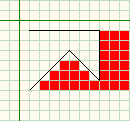
* Prva ivica (horizontalna, od (1,1) do (8,1)) se, po algoritmu, ignoriše. Narednu liniju obrađujemo i svi pikseli sa njene desne strane postaju obojeni.



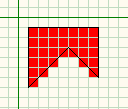
* Obrađivanjem naredne ivice, neki od prethodno obojenih piksela postaju transparentni.



* Analogno, obradom naredne ivice, stanje piksela postaje:

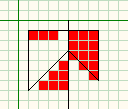


* Nakon poslednjeg koraka poligon ostaje obojen a svi pikseli van transparentni.



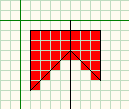
Primer modifikovanog algoritma (sa graničnom linijom po sredini):

* Algoritam u radu:



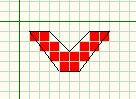
Primenom modifikovanog algoritma, broj koraka ostaje isti (ukoliko se ne koriste niti pa se leva i desna strana u isto vreme obrađuju). Ali broj komplementiranja piksela je znatno manji. Na piksele sa desne strane ne utiču ivice koje su sa leve strane granične ivice i obratno. Rezultat je isti.

* Rezultat modifikovanog algoritma:



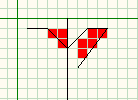
Primer specifičnijih poligona:

* Poligon oblika slova „V“:



U ovom primeru, imperfekcije su prilično vidljive, određeni pikseli unutar poligona nisu obojeni. Razlog tome je to što nije moguće delimično obojiti piksel a algoritam smatra da je piksel unutar poligona ako i samo ako je njegovo središte desno od preseka skan linije i trenutne ivice obrade. Na ovaj način, pikseli koji su „presečeni“ ivicom će za algoritam pripadati poligonu samo ako veći „isečak“ pripada unutrašnjosti poligona (ili, jednostavnije rečeno, središnja tačna piksela pripada unutrašnjosti poligona). Ova pojava je poznata i kao „aliasing“.

* Poligon oblika slova „V“ i modifikovani algoritam (u radu):



* Poligon oblika slova „V“, rotiran za 90 stepeni:

