Hranica a obrys oblasti

Venujme sa znovu diskrétnym množinám v bitovej mape, ktoré sme nazývali oblasti. Kvôli zmenšeniu pamäťovej náročnosti na zápis oblasti, má zmysel oblasť popísať čo najmenším počtom jej bodov. Z tohoto dôvodu sa budeme zaoberať algoritmom na určenie hranice oblasti a algoritmom na určenie obrysu - orientovanej hranice oblasti.

Pripomeňme, ako je definovaná hranica množiny v euklidovskej, spojitej rovine: sú to tie body množiny, ktoré majú v ľubovoľnom svojom okolí bod z množiny, aj bod z doplnku množiny.

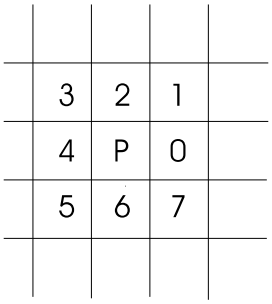
V diskrétnej rovine definujme hranicu obdobne:

Hranica oblasti je množina tých bodov oblasti, ktoré majú suseda z doplnku oblasti.

Obrys oblasti je postupnosť hraničných bodov oblasti, orientovaná hranica.

Pri definícii, teda aj pri vytváraní hranice resp. obrysu môžeme uvažovať 4-susednosť alebo 8-susednosť.

Pripomenieme označenie susedov bodu P :



Obrázok 1: Susednosť v bitovej mape, resp. v rastri

i-teho suseda bodu P so súradnicami $[p, q]$ budeme označovať $neighb(P, i)$ resp., $neighb(p, q, i)$.

Jednoduchý algoritmus hranice

Majme danú oblasť, množinu bodov v bitovej mape. Nech je bitová mapa reprezentovaná maticou $H[i, j], i = 1, \ldots, x_{max}$; $j = 1, \ldots, y_{max}$. Ak $H[i, j] = 1$, bod patrí oblasti, ak $H[i, j] = 0$ bod patrí doplnku oblasti. Pri určovaní hranice uvažujme 4-susednosť.

Na výstupe nasledovného algoritmu je hranica oblasti označená v bitovej mape hodnotami 2.

```
procedure Boundary
    begin
        for i=1 to x_max
```

```
            for j=1 to y_max
                if H(i,j)=1 then
                begin
                    if H(neighb(i,j,0))=0 then H[i,j]:=2 else
                    if H(neighb(i,j,2))=0 then H[i,j]:=2 else
                    if H(neighb(i,j,4))=0 then H[i,j]:=2 else
                    if H(neighb(i,j,6))=0 then H[i,j]:=2;
                end;
    end;
```

Jednoduchý algoritmus obrysu

Majme danú oblasť rovnakým spôsobom ako v predošlom prípade. Pri vytváraní obrysu budeme teraz uvažovať 8-súvislosť.

Princíp algoritmu spočíva v tom, že si budeme značiť (napr. odkladať do zásobníka) body, ktoré identifikujeme ako hraničné. Ak sa nám vždy podarí nájsť najbližší hraničný bod k tomu, ktorý už máme, pričom budeme postupovať vždy v rovnakom smere, v zásobníku bude obrys oblasti vtedy, keď sa vrátime do východzieho bodu.

Nájdime prvý obrysový bod. Ak to urobíme tak, že prezrieme po riadkoch všetky body bitovej mapy od ľavého vrchného bodu po prvý bod z oblasti (nech je to P_0), máme istotu, že sused číslo 4 bodu P_0 je z doplnku množiny. Ak chceme vytvoriť obrys oblasti orientovaný v smere proti chodu hodinových ručičiek, prehľadajme susedov bodu P_0 od suseda číslo 5.

Ak nájdeného suseda - ďalší obrysový bod označíme P_1 a vieme, že sme ho našli ako j-teho suseda bodu P_0, pri hľadaní bodu P_2 môžme prehľadávať susedov bodu P_1 od toho suseda, ktorý zodpovedá $j-1$ susedovi P_0.

Transformáciu medzi číslami susedov jednotlivých obrysových bodov je možné reprezentovať tabuľkou.

Uveďme algoritmus v pseudokóde:

1. nájdi prvý bod obrysu P_0
2. $P := P_0; j := 5;$
3. začiatok cyklu
   - kým $H(neighb(P, j)) \neq 1$ opakuj $j := j + 1$
   - $P := neighb(P, j); j := trans(j)$
   - odlož P
4. ak $P \neq P_0$ choď na bod 3 (koniec cyklu)

Algoritmus používa túto transformačnú tabuľku:

j	0	1	2	3	4	5	6	7
trans	7	7	1	1	3	3	5	5

Je nutné poznamenať, že uvedený algoritmus nájde vonkajší obrys oblasti, pre hľadanie vnútorného obrysu je ho však možné jednoducho modifikovať.

Použitá literatúra

1. Ružický, E. - Ferko, A. 1995. Počítačová grafika a spracovanie obrazu. 1995. ISBN 80-967180-2-9. Bratislava: SAPIENTIA 1995.

2. Žára, J. a kol. 1998. Moderní počítačová grafika. ISBN 80-7226-049-9. Computer Press 1998

3. Juraj Štugel, www.pg.miesto.sk