Skelet oblasti

V niektorých aplikáciach počítačovej grafiky je nutné vytvoriť pre danú oblasť čo najmenšiu množinu bodov, ktorá úplne charakterizuje tvar danej množiny.

Z aplikácií to môže byť napríklad algoritmus rozoznávania písma z množín získaných skenerom, ktoré sú spravidla zaťažené chybou a majú jemné odchýlky pre to isté písmeno. Preto sa každá zosnímaná množina nahradí čiarou, ktorá charakterizuje jej tvar. U tejto čiary sa skúmajú charakteristické prvky pre jednotlivé písmená.

Budeme teraz hovoriť o skelete (kostre) oblasti, teda o množine bodov oblasti, ktoré majú také vlastnosti, že je možné ich na uvedené účely využiť.

V euklidovskej rovine je možné skelet množiny presne definovať. Za skelet množiny v spojitej rovine sa považuje zjednotenie stredov kružníc vpísaných do oblasti, ktoré sa hranice oblasti dotýkajú v aspoň dvoch bodoch. Takýto skelet je množina dimenzie nanajvíš jedna a pôvodnú množinu je zo skeletu možné rekonštruovať (polomery jednotlivých kružníc si musíme uložiť).

Analogická definícia v diskrétnej bitovej mape pochopiteľne nefunguje, skelet oblasti musíme popísať inak. Exaktná definícia skeletu oblasti v bitovej mape neexistuje, presne vieme uviesť len požiadavky, ktoré ma skelet spĺňať. S trochou nadsádzky je možné povedať, že skelet oblasti je to, čo vyrobí skeletovací algoritmus.

Základné požiadavky na skelet

Skelet oblasti by malo tvoriť *čo najmenej bodov*, je vhodné aby bol tvorený čiarami hrúbky jedna.

Skelet oblasti musí čo najpresnejšie *vystihovať tvar oblasti.* Túto požiadavku možno dosiahnuť tak, že budú splnené nasledovné dva body:

- Skelet oblasti musí byť s oblasťou *topologicky ekvivalentný*, t.j. musí mať rovnaký počet komponentov súvislosti ako oblasť a doplnok skeletu v bitovej mape, musí mať zas rovnaký počet komponentov súvislosti ako doplnok pôvodnej množiny.

- *Koncové body línií* pôvodnej oblasti musia byť v skelete zachované.

Skelet oblasti by mal zodpovedať čo najpresnejšie *strednej línii* pôvodnej oblasti, aby bolo možné túto z neho aspoň približne rekonštruovať.

Je jasné, že presná, exaktná je len podmienka o topologickej ekvivalencii. Nepresnosť ostatných podmienok je príčinou existencie rôznych skeletovacích algoritmov, ktoré pre jednu oblasť vytvoria často rôzne skelety.

Jednoduchý algoritmus skeletu

Tento algoritmus funguje na princípe vyhľadávania tých bodov obrysu množiny, ktoré je možné označiť za skeletové na základe splnenia jednej z troch jednoduchých podmienok. Body obrysu, ktoré neboli označené ako skeletové sa z oblasti vymažú a algoritmus určí znova obrys takejto zmenšenej množiny, v ňom skeletové body, ostatné body obrysu vymaže atď., až dovtedy, pokiaľ sú všetky body nájdeného obrysu označené za skeletové.

Nevýhodou popísaného algoritmu je to, že výsledkom nemusí byť vo všeobecnosti útvar zložený z čiar hrúbky jedna.

Uveďme niekoľko definícií, ktoré využijeme pri formulácii podmienok skeletového bodu.

vnútro množiny sú body množiny, ktoré nie sú obrysovými bodmi

viacnásobný obrysový bod je bod množiny, ktorý sa v obrysovej postupnosti vyskytuje viac ako raz

obrysový susedia sú body množiny, ktoré sa v obrysovej postupnosti nachádzajú hneď za sebou

A teraz k tomu podstatnému v algoritme: Bod obrysu množiny nazveme skeletovým, ak spĺňa aspoň jednu z nasledovných vlastností:

1. bod nemá susedov z vnútra množiny
2. bod je viacnásobný obrysový bod
3. bod má priameho suseda z obrysu množiny, ale tento nie je jeho obrysový sused

Poznamenajme, že preskúmanie jednotlivých podmienok je možné pri prechode už vytvoreného obrysu množiny. Z hľadiska implementácie je výhodné jednotlivé podmienky popísať maticami - maskami, ktoré popisujú skúmany bod a jeho okolie.

Toto sú základné matice, ktoré spolu so svojimi otočeniami o 90 stupňov popisujú podmienky 1. - 3. overované pri prechode po už vytvorenom obryse.

$1'$	$1'$	$1'$	A	A	A	A	A	A	A	A	C
$1'$	P	$1'$	0	P	0	A	P	0	0	P	2
$1'$	$1'$	$1'$	B	B	B	A	0	B	B	B	C

P - skúmaný bod

0 - bod z doplnku množiny

1 - bod z vnútra množiny

2 - obrysový bod

$1'$ - bod, ktorý nie je z vnútra množiny (obrysový bod, alebo bod z doplnku)

A - ľubovoľný bod, aspoň jeden z takto označených bodov patrí množine

B - ľubovoľný bod, aspoň jeden z takto označených bodov patrí množine

C - ľubovoľný bod, aspoň jeden z takto označených bodov patrí množine

$*$ - ľubovoľný bod

Algoritmus by sa stal efektívnejším ak by bolo možné overiť jednotlivé podmienky už pri vytváraní obrysu a to pomocou lokálnej charakteristiky. Je zrejmé, že matice - masky v takomto prípade budú väčšie ako v predošlom (pravdepodobne budú postačovať matice $5 * 5$) a bude ich aj viac typov. (Autorovi nie je známy kompletný zoznam matíc, ktoré by bezo zvyšku vystihovali všetky prípady, v ktorých je niektorá z podmienok splnená.)

Úloha

Pokúste sa nájsť čo najviac masiek, ktoré by bolo možné použiť na overenie podmienok 1. - 3., už pri vytváraní obrysu.

Zhang - Suen algoritmus skeletu

Predošlý algoritmus vyhľadával medzi obrysovými bodmi body skeletové. Algoritmus Zhang - Suen vyhľadá také body množiny, ktoré nebudú skeletové a vymaže ich z množiny. Nižšie uvedieme podmienky, ktoré ak budú všetky súčasne splnené pre niektorý bod, tento bude možné vymazať.

Algoritmus bude pracovať v dvoch krokoch, aby bola dosiahnutá požiadavka na stredovú líniu skeletu.

V prvom kroku sa bod množiny vymaže ak sú súčasne splnené podmienky:

$V_1 : 2 \leq H(neighb(P,0)) + H(neighb(P,1)) + \ldots + H(neighb(P,7)) \leq 6$

$V_2 : num_{01}(P) = 1$

$V_3 :$ H(neighb(P,0)).H(neighb(P,2)).H(neighb(P,4))=0

$V_4 :$ H(neighb(P,2)).H(neighb(P,4)).H(neighb(P,6))=0

Podmienka V_1 zaručuje, že nebude vylúčený koncový bod oblasti. ($H(Q)$ má hodnotu 1 ak je bod Q z množiny, 0 ak je z doplnku množiny a $neighb(P,i)$ označuje i-teho suseda bodu P)

$num_{01}(P)$ označuje počet $0-1$ prechodov v okolí bodu P, podmienka V_2 teda zaručuje, že nebude pretrhnutá tenká línia bodov množiny.

Podmienky V_3 a V_4 hovoria o tom, že v prvom kroku sa vylučujú body z hornej alebo ľavej časti hranice, prípadne z pravého dolného rohu. V druhom kroku algoritmu budú tieto podmienky zmenené na symetrické. Tieto podmienky je možné charakterizovať maticami - maskami.

V_3 a zároveň V_4 sú splnené ak okolie bodu má tvar:

$*$	0	$*$
$*$	P	$*$
$*$	$*$	$*$

$*$	$*$	$*$
0	P	$*$
$*$	$*$	$*$

$*$	$*$	$*$
$*$	P	0
$*$	0	$*$

V druhom kroku sa bod množiny vymaže ak sú súčasne splnené podmienky V_1, V_2, V_3', V_4'.

V_3' : H(neighb(P,0)).H(neighb(P,2)).H(neighb(P,6))=0

V_4' : H(neighb(P,0)).H(neighb(P,4)).H(neighb(P,6))=0

V_3' a zároveň V_4' sú splnené ak okolie bodu má tvar:

$*$	$*$	$*$
$*$	P	0
$*$	$*$	$*$

$*$	$*$	$*$
$*$	P	$*$
$*$	0	$*$

$*$	0	$*$
0	P	$*$
$*$	$*$	$*$

Algoritmus

```
Procedure Zhang_Suen
begin
    remain := TRUE;
    while remain do
        begin
            remain := FALSE;
            for P do
                if h[P]=1 and
                    2<=H(neighb(P,0))+...+H(neighb(P,7))<=6 and
                    H(neighb(P,0)).H(neighb(P,2)).H(neighb(P,4))=0 and
                    H(neighb(P,2)).H(neighb(P,4)).H(neighb(P,6))=0
                then put_in_stack(P);
            for P in stack do
                begin
                    h[P] := 0;
                    remain := TRUE;
                end
            for P do
                if h[P]=1 and
                    2<=H(neighb(P,0))+...+H(neighb(P,7))<=6 and
                    H(neighb(P,0)).H(neighb(P,2)).H(neighb(P,6))=0 and
                    H(neighb(P,0)).H(neighb(P,4)).H(neighb(P,6))=0
                    then put_in_stack(P);
            for P in stack do
                begin
                    h[P] := 0;
                    remain := TRUE;
                end
        end
end.
```

Použitá literatúra

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