

PRESEK

List za mlade matematike, fizike, astronome in računalnikarje

ISSN 0351-6652

Letnik 23 (1995/1996)

Številka 1

Strani 34-38

Matija Lokar:

O PRESLIKAVAH RAVNINE, PRAPROTI IN STISKANJU PODATKOV

Ključne besede: matematika, računalništvo, preslikave ravnine, stiskanje podatkov.

Elektronska verzija: <http://www.presek.si/23/1252-Lokar.pdf>

© 1995 Društvo matematikov, fizikov in astronomov Slovenije

© 2010 DMFA - založništvo

Vse pravice pridržane. Razmnoževanje ali reproduciranje celote ali posameznih delov brez poprejšnjega dovoljenja založnika ni dovoljeno.

O PRESLIKAVAH RAVNINE, PRAPROTI IN STISKANJU PODATKOV

Kaj imajo skupnega preslikava ravnine, praproti, kot jo vidite na sliki 1, in stiskanje podatkov? Na prvi pogled nič. Kot bomo videli s pomočjo članka iz novembrske številke revije PC Magazine, pa zveza med njimi obstaja. Tako smo praproti narisali s pomočjo štirih preprostih preslikav ravnine. Če sliko dobro pogledate, lahko opazite, da je vsak list kopija celotne praproti. To lastnost pa uporablja tudi postopek, ki ga uspešno uporabljajo za stiskanje (kompresijo) podatkov.



Slika 1.

Točkam, ki so nekaj časa mirno ždele na koordinatni ravnini, je postalo dolgčas. Zato so se odločile, da si malo pretegnejo noge. Ker tako odlična združba ne more početi ničesar, ne da bi to dobro pretehtala, so to pretegovanje nog opravile po načrtu. Tako se je vsaka točka T s svojega starega mesta, podanega kot par koordinat (x, y) , odpravila na novo mesto T' , katerega položaj je bil dan z (x', y') . Pri tem je veljalo

$$\begin{aligned}x' &= ax + by + e \\y' &= cx + dy + f.\end{aligned}$$

Da bi bilo potovanje zanimivejše, so se vrednosti koeficientov a , b , c , d , e in f od časa do časa spremenile.

Tako se je na pot odpravila tudi točka $(0, 0)$. Ker točke hodijo zelo hitro, je dokaj hitro opravila 10000 korakov. Kot zelo sistematična oseba si je skrbno beležila vsa mesta, ki jih je obiskala, pa tudi koeficiente, ki jih je pri vsakem premiku uporabila za izračun svojega novega položaja.

Opazila je, da so se izmenjevali le štirje nabori koeficientov, vendar so nekateri nastopali pogosto, drugi pa redkeje. Zložila jih je v tabelo:

a	b	c	d	e	f	koliko
0.00	0.00	0.00	0.16	0.00	0.00	100
0.85	0.04	-0.04	0.85	0.00	1.60	8500
0.20	-0.26	0.23	0.22	0.00	1.60	700
-0.15	0.28	0.26	0.24	0.00	0.44	700

Nad sliko obiskanih točk (slika 2) pa se je še sama začudila: “Saj to je čisto prava praproť.”



Slika 2.

Preslikavam, kot so jih opravile naše točke, učeno rečemo afine preslikave. Morda se bomo z njimi natančneje spoznali kdaj drugič. Ker pa smo tokrat v računalniškem delu Preseka, nas predvsem zanima, kako bi tudi mi narisali takole praproť. Zadeva je preprosta. Kot vidimo, moramo uporabiti štiri afine preslikave. Vsaka preslikava ima prirejeno še pogostost uporabe, ki določa njeno pomembnost. V našem primeru je najpomembnejša druga preslikava, ki nastopa v 85% primerov.

Začnemo s točko $(0, 0)$ in s podanimi verjetnostmi izberemo eno od afinih preslikav. Izračunamo novo točko in to ponavljamo. Sprva nesmiselni razpored točk se nam bo ob dovolj ponovitvah izoblikoval v sliko praproťi. Slike zaradi naključne izbire preslikav sicer niso vedno povsem

enake, vendar pa jih pri dovolj velikem številu točk s prostim očesom skoraj ne moremo ločiti. Čeprav ne dvomim, da bi znali program, ki nariše sliko, napisati tudi sami, ga vseeno zapišimo še tukaj.

```

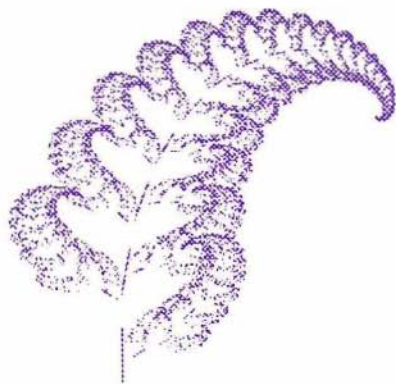
SCREEN 12
WINDOW (-4, 0)-(6, 10)
RANDOMIZE TIMER
X = 0 : Y = 0
WHILE INKEY$ = ""
  R = RND
  IF R < .01 THEN
    A = 0: B = 0: C = 0: D = .16: E = 0: F = 0
  ELSEIF R < .86 THEN
    A = .85: B = .04: C = -.04: D = .85: E = 0: F = 1.6
  ELSEIF R < .93 THEN
    A = .2: B = -.26: C = .23: D = .22: E = 0: F = 1.6
  ELSE
    A = -.15: B = .28: C = .26: D = .24: E = 0: F = .44
  END IF
  NOVIX = (A * X) + (B * Y) + E
  NOVIY = (C * X) + (D * Y) + F
  X = NOVIX
  Y = NOVIY
  PSET (X, Y), 2
WEND

```

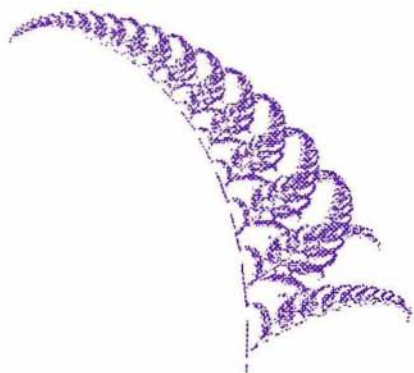
Program je zapisan v QBASICU, programskem jeziku, ki ga lahko uporabljajo vsi, ki imajo na računalnikih operacijski sistem MS-DOS, pa čeravno za ta programski jezik verjetno še vedo ne. Pokličemo ga z ukazom QBASIC in vtipkamo gornje ukaze. S stavkom SCREEN 12 povemo, da uporabljamo grafični način VGA. Točke, ki jih določajo naše štiri afine preslikave, ležijo v kvadratu $(-4, 6) \times (0, 10)$. Zato s ukazom WINDOW poskrbimo, da bomo sliko dobili raztegnjeno na cel zaslon. No, včasih nam kakšna točka pobegne z zaslona, vendar takih enostavno ne upoštevamo. Glavna zanka, v kateri generiramo točke, se ponavlja, dokler ne pritisnemo poljubne tipke. V njej glede na vrednost naključnega števila izberemo ustrezno preslikavo in s stavkom PSET (X, Y), 2 narišemo zeleno (to barvo določa število 2) točko.

Sedaj, ko je program napisan, se lahko malo poigramo s koeficienti v tabeli. Z nekaj poskušanja in kombiniranja lahko ustvarimo prav zanimive slike. Nekaj jih vidimo na slikah 3, 4 in 5. S poskušanjem tudi ugotovimo, da če izpustimo tretjo preslikavo, izgubimo levo polovico veje, četrta preslikava pa je zadolžena za desno stran. Koeficienta b in c skrbita za upognjenost praproti, koeficient f pa vpliva na njeno "potlačенost". Razlogov, zakaj s kombinacijo teh štirih preslikav dobimo sliko praproti,

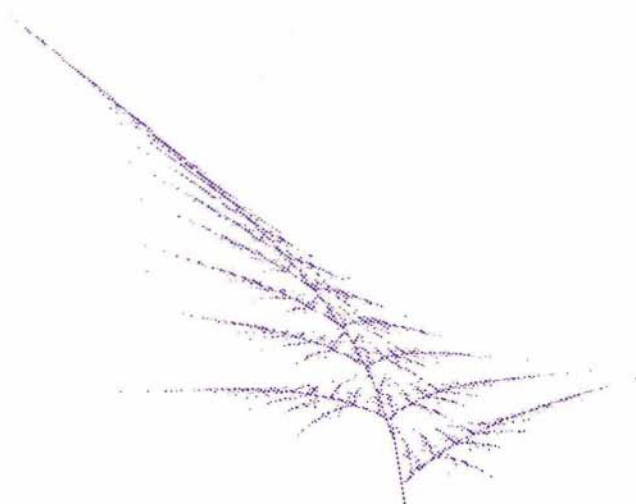
ni lahko pojasniti. Osnovni ključ je samopodobnost slike – veje, ki sestavljajo glavno vejo praproti, so same spet “praproti v malem”. Afine preslikave nam omogočajo, da del ravnine povečamo ali pomanjšamo, ga zasukamo in prestavimo v poljubni smeri. Če torej vzamemo majhen kvadratik iz slike praproti, ga lahko s primernim zaporedjem afinih preslikav razmnožimo po vsej ravnini tako, da nam sestavi celotno sliko.



Slika 3.



Slika 4.



Slika 5.

In kje je tukaj stiskanje podatkov? Da bi v računalnik shranili celotno sliko 2, nam ni potrebno shraniti podatkov, katera izmed $640 \cdot 480 = 307200$ točk zaslona je pobarvana in katera ne, ampak le tabelo koeficientov štirih afinih preslikav. V prvem primeru bi potrebovali približno $307200/8 \doteq 38000$ zlogov pomnilnika. Ker za hranjenje realnega števila ponavadi potrebujemo štiri zloge, za hranjenje tabele koeficientov potrebujemo le $4 \cdot 7 \cdot 4 = 112$ zlogov. To pa je kar 340-kratni prihranek. Si predstavljate, da bi lahko vse podatke v računalniku hranili tako učinkovito. Na eno samo disketo bi šlo več podatkov kot na disk velikosti 400 MB. Konec problemov z vedno premajhnimi diski! In kako to, da nam na diskih naših računalnikov še vedno primanjkuje prostora? Zarota prodajalcev diskov? Žal ni tako. Postopek stiskanja je namreč zelo počasen in računsko zahteven. Prvotne podatke v nestisnjeni obliki lahko dobimo precej hitro – naš program nam je iz stisnjene zapisa slike praproti (tabele koeficientov in verjetnosti izbir posameznih afinih preslikav) kar hitro prikazal njeno podobo na zaslon. Določiti, kako podatke stisniti, pa gre veliko počasneje. Izredno težavno je namreč določiti vse potrebne preslikave, njihovo število, koeficiente in verjetnosti. Drugi razlog je, da običajno podatki niso dovolj "samopodobni". Zato jih ni mogoče učinkovito opisati z nekaj afinimi transformacijami. Bolj uspešno je tako opisovanje pri slikah. Tam namreč "samopodobnost" delov slike lažje dosežemo, če malo "pogoljufamo". Pri sliki namreč (običajno) ni nič hudega, če kako točko malo "popravimo", saj tega praktično ne bomo opazili. Res pa je, da slika, ki jo bomo dobili iz opisa z afinimi transformacijami, ne bo povsem enaka originalu. To si pri sliki lahko dovolimo, pri drugačnih podatkih pa seveda ne. Zato tam uporabljamo drugačne metode stiskanja, ki pa so veliko manj učinkovite. Ponavadi prihranimo največ pol prostora – dosežemo torej faktor 2.

Kot smo že omenili, je postopek opisovanja slike z afinimi preslikavami zelo zapleten in zahteva veliko časa. Dva znanstvenika sta razvila postopek, s katerim lahko to naredimo pri poljubni sliki in sprejemljivem času. Tako je kar nekaj slik, predvsem v multimedijskih enciklopedijah (npr. v Microsoftovi Encarti), zapisanih na ta način. Vsi podatki o postopku niso znani, saj je sam postopek napredaj v obliki programa. O njem vemo to, da slika razdeli na posamezna nepravilna območja in potem določi ustrezne affine preslikave. Te niso vedno le štiri. Lahko jih je tudi več deset ali pa tudi več kot sto. Na ta način dosežejo, da za zapis posamezne slike porabijo v povprečju okoli stokrat manj prostora.