

TRIANGULIRAJMO MNOGOKOTNIK

Marko Lamot*

dr. Borut Žalik**

KLJUČNE BESEDE:

*mnogokotnik,
triangulacija
mnogokotnikov,
računalniška
geometrija, algoritmi*

Izvleček

V članku predstavljamo tehnike delitve mnogokotnikov v trikotnike oz. triangulacijo mnogokotnikov. Namen delitve mnogokotnikov je v poenostavitvi obdelovanja mnogokotnikov, saj so lahko le-ti v geodetskih aplikacijah zelo kompleksni (vsebujejo veliko število konkavnih oglišč, imajo ugnezdene luknje). Vsak mnogokotnik je mogoče triangulirati z vstavljanjem diagonal, kar je razvidno iz dokaza o triangulaciji mnogokotnika. Obstaja veliko postopkov, ki uporabljajo to dejstvo, vendar pa je mogoče triangulirati mnogokotnike tudi s popolnoma drugimi pristopi. Algoritme delitve mnogokotnikov lahko delimo na tri skupine: algoritme, ki temeljijo na vstavljanju diagonale, algoritme, ki temeljijo na Delaunayevi triangulaciji in algoritme, ki uporabljajo za delitev Steinerjeve točke.

KEYWORDS:

*polygon, polygon
triangulation,
computational
geometry, algorithms*

Abstract

This paper considers different approaches how to divide polygons into triangles what is known as a polygon triangulation. Polygons can be very complex in geodesic applications (they could have a lot of concave vertices, they could contain holes) therefore there is often a need to decompose them into simpler components. Every polygon can be triangulated by inserting diagonals what is shown in the proof of existence of polygon triangulation. There are a lot of polygon triangulation techniques which use that fact. However, polygons can be triangulated by some other approaches, too. The algorithms performing polygon triangulation can be classified into three major groups: algorithms, which are based on diagonal insertion, algorithms, which are based on Delaunay triangulation, and algorithms using Steiner's points.

1. UVOD

Mnogokotniki so brez dvoma najpogosteje uporabljeni geometrijski objekti v najrazličnejših inženirskih aplikacijah. V 3D prikazih jih lahko uporabimo za opis lica geometrijskih objektov, predstavljenih z ovojnico (Žalik, 1999), v 2D prikazih pa mnogokrat predstavljajo že končno obliko geometrijskega objekta, ki ga predstavljamo. Paleta teh aplikacij je izjemno široka: od načrtovanja razporeditve opreme v stanovanju, pa do določitve očrtij črk v vektorski predstavitvi. Področje, kjer srečamo izredno zapletene oblike mnogokotnikov, so brez dvoma geografski informacijski sistemi - GIS. V GIS niso redki mnogokotniki, ki so določeni z nekaj 10.000 oglišči in vsebujejo

veliko število (ugnezdenih) lukenj. Prav taki mnogokotniki mnogokrat predstavljajo veliko težavo pri njihovi obdelavi. Zato je ugodno, če jih znamo razdeliti v enostavnejše geometrijske oblike. Mnogokotnike lahko npr. delimo na trikotnike, trapezoide ali celo na mnogokotnike v obliki zvezde. Najbolj raziskani so algoritmi, ki delijo mnogokotnike v trikotnike, temu procesu pa pravimo triangulacija mnogokotnika. Če oglišča mnogokotnika vsebujejo tudi informacijo o nadmorski višini, lahko trianguliranemu mnogokotniku mnogo natančneje določimo ploščino, določimo lahko njegovo najbolj razgledno točko. Nekateri algoritmi Boolovih operacij nad mnogokotniki (preseki ali unija mnogokotnika) konkavne mnogokotnike najprej razdelijo na trikotnike, nad katerimi je omenjene operacije zelo enostavno opraviti, nato pa Boolove operacije z lepljenjem delnih rezultatov tvorijo izhodni mnogokotnik. Algoritme triangulacije lahko uporabimo tudi v aplikacijah interpolacije izolinij, rekonstrukcije 3D objektov in predstavitve ploskev. Algoritmi triangulacije mnogokotnikov so zelo dobro raziskani. Delimo jih lahko v tri skupine, ki se med seboj ločijo po hitrosti in kakovosti tvorjenih trikotnikov.

V tem članku bomo v petih poglavjih predstavili povzetek triangulacijskih tehnik in omenili algoritme, ki so najznačilnejši predstavniki. Drugo poglavje zelo na kratko uvaja osnovna dejstva in terminologijo. V tretjem poglavju predstavljamo pregled postopkov triangulacije mnogokotnikov na osnovi vstavljanja diagonal, četrto poglavje opisuje algoritme omejene Delaunayeve triangulacije, peto poglavje pa podaja algoritme, ki temeljijo na Steinerjevih točkah. V zadnjem poglavju na kratko povzamemo delo.

2. OSNOVNA TERMINOLOGIJA

Vsak enostavni mnogokotnik (stranice se med sabo sekajo le v ogliščih) z n oglišči je možno triangulirati. Ključ do dokaza triangulacije je dejstvo, da ima vsak mnogokotnik diagonalo, ki obstaja, če ima mnogokotnik vsaj eno konveksno oglišče. Velja trditev (O'Rourke, 1994):

- vsak mnogokotnik ima vsaj eno konveksno oglišče,
- vsak mnogokotnik z $n \geq 4$ oglišči ima diagonalo,
- vsak mnogokotnik z n oglišči je mogoče razdeliti na trikotnike z dodajanjem diagonal (nobene ali več diagonal).

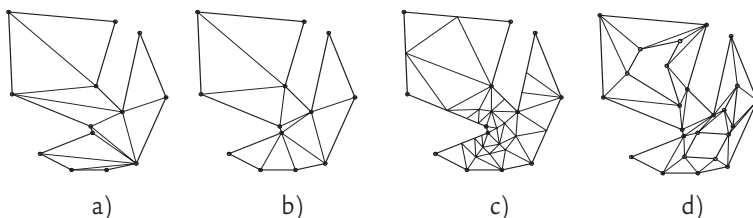
Obstaja mnogo načinov trianguliranja mnogokotnikov. Vse možnosti imajo skupno to, da je število diagonal po končani triangulaciji $n - 3$, število generiranih trikotnikov pa $n - 2$. Dokaze in detajle najdemo na primer v O'Rourkovem delu iz leta 1994.

Iz dokaza o triangulaciji je razviden algoritem, ki bo trianguliral mnogokotnik: razdeli mnogokotnik na dva dela z vstavljanjem diagonale in tako dobljena mnogokotnika rekurzivno deli naprej. Vendar je takšen neposreden pristop preveč neučinkovit - potrebuje $O(n^4)$ časa (O'Rourke,

1994), zato je mnogo avtorjev predlagalo bistveno hitreje algoritme. Avtorji predlagajo različne tehnike in pristope. Nekateri triangulirajo mnogokotnik neposredno, drugi pa ga najprej delijo v enostavnejše mnogokotnike (monotone, konveksne,...), ki jih je mogoče potem triangulirati v linearnem času.

Za nekatere aplikacije je bistveno, da je minimalni notranji kot trikotnika izračunane triangulacije maksimiziran, s čimer zagotovimo "lepe" (čim bolj enakostranične) trikotnike.

Slika 1: Različni načini triangulacije mnogokotnika



Na sliki 1a lahko vidimo triangulacijo mnogokotnika slabe kakovosti, ker je razdeljen na veliko ozkih mnogokotnikov. Algoritmi, ki temeljijo na Delaunayevi triangulaciji točk, zagotavljajo boljše kakovost (slika 1b). Delaunayeva triangulacija množice točk v ravnini je namreč maksimalno povezan ravninski graf, katerega oglišča so dane točke in katerega robovi se med seboj ne sekajo, razen v krajšičih. Poleg tega velja, da očrtan krog kateregakoli trikotnika ne sme vsebovati katerekoli druge točke (t.i. lastnost praznega kroga) (O'Rourke, 1994) in ravno ta lastnost zagotavlja omenjeno boljše kakovost. Kakovost bistveno izboljšamo z uporabo tako imenovanih Steinerjevih točk (slika 1c). Steinerjeve točke so točke, ki jih dodamo med triangulacijo na robove mnogokotnika oz. v notranjost mnogokotnika. Točke v končni triangulaciji predstavljajo oglišča trikotnikov. Seveda lahko z uporabo teh točk bistveno povečamo število trikotnikov na izhodu.

Mnogokotniki lahko vsebujejo tudi luknje (slika 1d), kar je bistvenega pomena za nekatere aplikacije. Nekateri algoritmi ne morejo triangulirati takšne vrste mnogokotnikov, medtem ko je za druge to nepomembno.

V splošnem lahko delimo triangulacijske algoritme mnogokotnikov na tri skupine: algoritme na osnovi vstavljanja diagonale, algoritme, ki temeljijo na Delaunayevi triangulaciji in algoritme, ki uporabljajo za triangulacijo Steinerjeve točke.

V nadaljevanju bomo predstavili splošne lastnosti algoritmov iz omenjenih treh skupin. Atributi, zanimivi za primerjavo, so (Shewchuk, 1999):

- kakovost triangulacije,
- število mnogokotnikov na izhodu,
- možnost triangulacije mnogokotnika z luknjami.

Slika 1 prikazuje, kako različno je možno triangulirati enak mnogokotnik z različnimi algoritmi:

- triangulacija na sliki 1a je generirana s Seidlovim algoritmom (Seidel, 1991), torej z vstavljanjem diagonal;
- slika 1b prikazuje omejeno Delaunajevo triangulacijo (de Floriani, 1992), ki temelji na Delaunayevi triangulaciji;
- slika 1c prikazuje izboljššan Rupertov Delaunayev algoritem (Ruppert, 1994), ki temelji na uporabi Steinerjevih točk.

Očitno je, da algoritmi, ki temeljijo na uporabi Steinerjevih točk, zagotavljajo največjo kakovost, saj imajo edini vgrajen mehanizem za zagotavljanje le-te, vendar pa zato proizvedejo tudi večje število trikotnikov na izhodu. Algoritmi, ki temeljijo na Delaunayevi triangulaciji točk, zagotavljajo največjo možno kvaliteto triangulacije na ogliščih mnogokotnika (slika 1b). Med izgradnjo triangulacije namreč upoštevajo t. i. lastnost praznega kroga. Algoritmi, ki temeljijo na vstavljanju diagonal, nimajo nobenega mehanizma za zagotavljanje kakovosti, zato so lahko triangulacije med seboj zelo različne. Algoritmi, ki temeljijo na Delaunayevi triangulaciji, in tisti, ki temeljijo na vstavljanju diagonal, vedno ustvarijo $n - 2$ trikotnikov.

Algoritmi, ki temeljijo na Delaunayevi triangulaciji točk, praviloma triangulirajo tudi mnogokotnike z luknjami. V bistvu triangulirajo tudi luknje, iz katerih po koncu triangulacije odstranimo trikotnike. Večina algoritmov, ki temeljijo na vstavljanju diagonal, ne more triangulirati mnogokotnikov z luknjami. Edina izjema je Seidelov algoritem. V osnovni izvedbi tudi ta algoritem ni bil namenjen triangulaciji mnogokotnikov z luknjami, vendar se je izkazalo, da razmeroma enostavna izboljšava učinkovito reši ta problem.

3. ALGORITMI TRIANGULACIJE Z VSTAVLJANJEM DIAGONALE

Matematike so zanimali predvsem konstruktivni dokazi (algoritmi) obstoja triangulacije za enostavne mnogokotnike. Zgodovina algoritmov za triangulacijo se začne leta 1911. Tega leta je Lennes objavil rekurzivni »algoritem« z vstavljanjem diagonal med pare oglišč mnogokotnika P . Algoritem je deloval v času $O(n^2)$. Od takrat so te vrste algoritmov prirejali in jih objavljali v velikem številu člankov (Lennes, 1911).

Meisters je tako leta 1975 predlagal malo spremenjen induktiven dokaz za triangulacijo. Predlagal je metodo, ki temelji na iskanju uhljev in rezanju teh uhljev. Oglišče v_i mnogokotnika P je glavno, če nobeno drugo oglišče ne leži v notranjosti trikotnika v_{i-1}, v_i, v_{i+1} oz. na diagonalni $v_{i-1}v_{i+1}$. Glavno oglišče v_i imenujemo uhljev, če diagonala $v_{i-1}v_{i+1}$ leži povsem v notranjosti mnogokotnika P . Pravimo, da sta uhlja v_i in v_j neprekrivajoča, če velja: $\text{notranjost}[v_{i-1}, v_i, v_{i+1}] \cap \text{notranjost}[v_{j-1}, v_j, v_{j+1}] = \emptyset$. Kot je pokazal Meisters, ima razen trikotnika vsak enostaven mnogokotnik P najmanj dva neprekrivajoča uhlja (Meisters, 1975). Neposredna implementacija tega nas

vodi do že omenjene časovne zahtevnosti $O(n^2)$. Vendar je bilo odkrito (leta 1990), da je mogoče s tehniko »obreži in išči« (»prune-and-search«) najti uhelj v linearnem času, kar je zmanjšalo celotno časovno zahtevnost na $O(n^2)$ (Elgindy, 1993).

Prvi algoritem, ki je rušil mejo $O(n^2)$, je bil algoritem, ki so ga predlagali Garey, Johnson, Preparata & Tarjan leta 1978. Njihov algoritem se je izvajal v času $O(n \log n)$. Njihov pristop je zajemal dva koraka: v prvem koraku so delili mnogokotnik v monotone mnogokotnike ($O(n \log n)$), v drugem koraku pa so te monotone mnogokotnike triangulirali v linearnem času ($O(n)$) (Garey, 1978). Drugo metodo z enako časovno zahtevnostjo je predlagal Fournier leta 1984. Popolnoma drug pristop, ki je temeljil na postopku »deli in vladaj«, je predstavil Chazell, prav tako z zgornjo mejo $O(n \log n)$. Po začetnih neuspehih sta leta 1986 Tarjan in Van Wyk uspela predstaviti algoritem s časovno zahtevnostjo $O(n \log \log n)$, ki pa je vseboval zelo zapletene in svojevrsne podatkovne strukture (Tarjan, 1988). Enostavnejši algoritem z enako časovno zahtevnostjo je kasneje predstavil tudi Kirkpatrick (Kirkpatrick, 1990).

Naslednja pospešitev je bil doseg časovne zahtevnosti $O(n \log^2 n)$. (Zapis $\log^2 n$ predstavlja, kolikokrat mora biti log iteriran, da vrednost n zmanjšamo na 1 ali manj.) Algoritmi tega razreda niso bili samo hitrejši, ampak tudi preprostejši. Prvi takšen algoritem je bil naključni algoritem Clarksona. Naključni algoritmi (včasih se imenujejo tudi »Las Vegas« algoritmi) vsebujejo naključne spremenljivke za pospešitev. Algoritem z enako časovno zahtevnostjo je predstavil tudi Kirkpatrick, ta je temeljil na mnogokotnikih z ustrežno omejenimi celoštevilčnimi koordinatami (Toussaint, 1991). Še en naključni algoritem je opisal Seidel (Seidel, 1991), in sicer naključni inkrementalni algoritem, prav tako s časovno zahtevnostjo $O(n \log^2 n)$. Tako je njegova zahtevnost skoraj linearna za veliko večino enostavnih mnogokotnikov. Ta algoritem najprej inkrementalno izračuna trapezoidno delitev mnogokotnika - najprej se izračuna naključna permutacija robov, nato pa se zaporedoma vstavljajo daljice. Seidel je pokazal, da je mogoče to delitev opraviti v $O(n \log^2 n)$ času. Iz te delitve se nato v linearnem času izračunajo monotone mnogokotniške verige, ki jih je mogoče triangulirati v linearnem času.

Hertel in Mehlhorn sta tako opisala algoritem, ki temelji na algoritmu pregledovanja ravnine in teče tem hitreje, čim manj konkavnih oglišč ima. Časovna zahtevnost algoritma je $O(n + r \log r)$, kjer je r število konkavnih oglišč v mnogokotniku P . Prvi korak v tem prilagojenem algoritmu je pridobitev informacije o obliki mnogokotnika. Na žalost pa r ni relevantno merilo oblikovne kompleksnosti mnogokotnika (Hertel, 1983). Chazelle in Incerpi sta prav tako predstavila algoritem, katerega časovna odvisnost je odvisna od oblike. Vendar, za razliko od prej omenjenega algoritma, časovna zahtevnost bolj zvesto sledi oblikovni kompleksnosti mnogokotnika. Opisala sta algoritem, ki teče v času $O(n \log s)$, kjer je $s < n$. Število s meri vijugavost (»sinuosity«) mnogokotnika oz. kolikokrat se meja mnogokotnika menjava

med zaključenima spiralama nasprotno orientacije. Za razliko od r ima s to prednost, da je v praktičnih primerih zelo majhen oz. celo konstanten (Chazelle, 1984).

Toussaint je predlagal algoritem, ki ima časovno zahtevnost $O(n(1+t_0))$; $t_0 < n$ (Toussaint, 1991). Vrednost t_0 meri kompleksnost triangulacije, ki jo poda sam algoritem. Bolj natančno je t_0 število trikotnikov v triangulaciji (izhod algoritma), ki si ne deli nobenega robu z vhodnim mnogokotnikom in je tako povezan s samo obliko mnogokotnika. Čeprav je časovna zahtevnost v najslabšem primeru $O(n^2)$, teče za kar nekaj razredov mnogokotnikov v linearnem času. Praktične prednosti algoritma so te, da je izredno preprost, ne zahteva urejevanja oz. uravnotežene drevesne strukture.

Omenimo še algoritem, ki temelji na Grahamovem pregledu (Kong, 1990). Grahamov pregled je osnovna tehnika vračanja v računalniški geometriji, prvotno uporabljena za izračun konveksne lupine množice točk v ravnini. V Kongovem besedilu je prikazana uporaba Grahamovega pregleda za triangulacijo enostavnih mnogokotnikov v $O(kn)$, kjer je $k - 1$ število konkavnih oglišč v P . Čeprav je v najslabšem primeru zahtevnost algoritma $O(n^2)$ med najenostavnejšimi za implementacijo (Kong, 1990).

Leta 1990 je Bernard Chazelle končno pokazal, da je enostaven mnogokotnik mogoče triangulirati v $O(n)$ času. To odkritje je bilo pomemben teoretični dosežek. Osnova je deterministični algoritem, ki temelji na t.i. vidni mapi («visibility map»), ki je posplošitev trapezoidne delitve (skozi vsako oglišče mnogokotnika se narišejo vodoravne tetive v obe smeri). Kot razlaga Chazelle, njegov algoritem posnema urejanje z zlivanjem («merge sort»), splošno tehniko pri urejanju z »deli in vladaj«. Algoritem pa ima veliko slabost v tem, da je v celotnem postopku ogromno detajlov in ga je zato zelo težko skonstruirati (Chazelle, 1991). Dolgoročni problem triangulacije (namreč triangulacije enostavnega mnogokotnika v linearnem času) je tako teoretično dokončno rešen, vendar je še vedno odprto vprašanje enostavnega, hitrega in praktičnega algoritma za triangulacijo na osnovi omenjenega postopka.

Tabela 1 prikazuje pregled vseh omenjenih algoritmov ter njihove časovne zahtevnosti. Izhajali smo iz O'Rourkeve tabele (O'Rourke, 1994), vendar smo jo dopolnili z dodatnimi algoritmi.

Tabela 1: Razvoj triangulacijskih algoritmov z vstavljanjem diagonale

Čas. zahtevnost	Avtor	Leto
$O(n^2)$	Lennes	1911
$O(n^2)$	Meisters	1975
$O(n^2)$	ElGindy, Everett, Toussaint	1990
$O(n \log n)$	Garey, Johnson, Preparata, Tarjan	1978
$O(n \log n)$	Chazelle	1982
$O(n + r \log r)$	Hertel & Mehlhorn	1983
$O(n \log s)$	Chazelle & Incerpi	1983
$O(n(1 + t_0))$	Toussaint	1988
$O(kn)$	Kong, Everett, Toussaint	1990
$O(n \log \log n)$	Tarjan, Van Wyk	1987
$O(n \log \log n)$	Kirkpatrick	1990
$O(n \log^2 n)$	Clarkson, Tarjan, Van Wyk	1989
$O(n \log^2 n)$	Kirkpatrick, Klawe, Tarjan	1990
$O(n \log^2 n)$	Seidel	1990
$O(n)$	Chazelle	1990

4. ALGORITMI TRIANGULACIJE Z UPORABO DELAUNAYEVE TRIANGULACIJE TOČK

Problem izračuna Delaunayeve triangulacije množice točk v ravnini je dobro znan problem v računalniški geometriji. Delaunayevo triangulacijo množice točk so kasneje razširili tako, da so dopuščali, da vhodna množica vsebuje tudi daljice (v našem primeru so to daljice stranice mnogokotnika). V računalniški geometriji je ta problem znan kot problem omejene (»constrained«) triangulacije, rezultat te triangulacije pa imenujemo omejena Delaunayeva triangulacija (CDT).

Algoritme za CDT lahko razdelimo v dve skupini: algoritme, kjer lahko vhodna množica vsebuje samo daljice, ki tvorijo enostaven mnogokotnik, in algoritme za splošne vhode oz. splošne ravninske grafe (daljice + točke). Razlika med obema skupinama je v tem, da prvi že med izvajanjem upoštevajo, da triangulirajo enostaven mnogokotnik in zato s pridom izkoriščajo to informacijo, medtem ko drugi ne upoštevajo omenjene informacije. V nadaljevanju podajamo glavne predstavnike prve skupine algoritmov.

Lewis in Robinson opisujeta algoritem s časovno zahtevnostjo $O(n^2)$, ki temelji na pristopu »deli in vladaj«. Mejo mnogokotnika rekurzivno delita v mnogokotnike skoraj enake velikosti, ki jih posebej triangulirata. Tako nastalo triangulacijo potem »optimizirata«, da dobita Delaunayevo triangulacijo (Shewchuk, 1999).

Drugi takšen rekurzivni algoritem, ki pa temelji na vidnosti točk, je podala de Floriani. Algoritem izračuna vidni graf oglišč mnogokotnika P v $O(n^2)$ in standardni Voronoijev diagram Q v času $O(n \log n)$. Delaunayeva triangulacija

se zgradi z združitvijo vsake točke p mnogokotnika p z vsemi točkami, ki so vidne tako iz P kot tudi iz Voronoijevih sosedov točke p . Skupna časovna zahtevnost algoritma je $O(n^2)$ (Shewchuk, 1999).

Algoritem časovne zahtevnosti $O(n \log n)$ sta opisala Lee in Lin. Temelji na Chazellovem teoremu odstranjevanja mnogokotnika. Chazelle je pokazal, da lahko v enostavnem mnogokotniku P z n oglišči najdemo dve oglišči A in B v linearnem času tako, da daljica AB cela leži v notranjosti mnogokotnika P . Prav tako je dokazal, da mnogokotnika, ki sta nastala z delitvijo mnogokotnika P z daljico AB , vsebujeta najmanj $n/3$ oglišč. Algoritem Leea in Lina razdeli dani mnogokotnik P v dva podmnogokotnika P_l in P_r , in nato rekurzivno izračuna omejeno Delaunayovo triangulacijo T_l in T_r teh dveh podmnogokotnikov. Celotno triangulacijo T mnogokotnika P dobimo z zlitvijo T_l in T_r (Shewchuk, 1999).

Mnogo več je algoritmov iz druge skupine, kjer se ne upošteva informacije o mnogokotniku med triangulacijo. Ti algoritmi niso tako zanimivi za triangulacijo mnogokotnikov, zato omenimo samo najosnovnejše: Lee in Lin predlagata podoben algoritem za CDT, kot je opisani algoritem za enostavne mnogokotnike in teče v času $O(n^2)$ (Shewchuk, 1999). Chew je predlagal algoritem, ki temelji na pristopu »deli in vladaj« in teče v času $O(n \log n)$ (Chew, 1987). Algoritem, ki temelji na predhodni obdelavi daljic in pretvorbi problema CDT v osnovno Delaunayovo triangulacijo, je podal Boissonnat (Boissonnat, 1988). Od vseh omenjenih algoritmov se razlikuje algoritem, ki ga predlagata Floriani in Puppo (Floriani, 1992) in za razliko od ostalih rešuje problem CDT z vzdrževanjem začetne CDT z zaporednim vstavljanjem točk n in daljic l .

V tabeli 2 so prikazani vsi omenjeni algoritmi, ki izvirajo iz osnovne Delaunayeve triangulacije točk. Vhod določa, ali gre za triangulacijo mnogokotnika ali pa algoritem lahko triangulira tudi splošen ravninski graf.

Časovna zahtevnost	Avtor	Leto	Vhod
$O(n^2)$	Lewis, Robinson	1979	Enostaven mnogokotnik
$O(n \log n)$	Florianis	1985	Enostaven mnogokotnik
$O(n \log n)$	Lee, Lin	1980	Enostaven mnogokotnik
$O(n^2)$	Lee, Lin	1980	Splošen
$O(n \log n)$	Chew	1987	Splošen
$O(n \log n)$	Boissonnat	1988	Splošen
$O(l n^2)$	Floriani, Puppo	1992	Splošen

Tabela 2: Algoritmi omejene Delaunayeve triangulacije za enostavne mnogokotnike

5. ALGORITMI TRIANGULACIJE Z UPORABO STEINERJEVIH TOČK

Posebno poglavje predstavljajo tudi algoritmi, ki poleg same triangulacije poljubnih ravninskih grafov zagotavljajo tudi ustrezno kakovost le-te. Se pravi, da dopuščajo nastavev najmanjšega možnega notranjega kota v

izhodni triangulaciji. To lahko dosežemo samo, če dovolimo uporabo Steinerjevih točk. V tem primeru je seveda število izhodnih trikotnikov vnaprej neznano. Zato pomen kakovosti dobi v tem primeru še dodaten parameter, in sicer je poleg oblike (čim večji notranji kot oz. čim bolj enakokranični trikotniki) trikotnikov potrebno zagotoviti čim bolj optimalno število trikotnikov v triangulaciji. Rezultat triangulacije pogosto imenujemo tudi mreža.

Ena izmed takšnih tehnik kvalitetne triangulacije točk in daljic je tudi t. i. izboljšana Delaunayeva triangulacija (»Delaunay refinement«). Tako se imenuje, ker ohranja osnovno triangulacijo ob tem, da se dodajo nekateri kriteriji za zaporedno določanje in dodajanje novih točk (Chew, 1989). Chew je predstavil takšen algoritem, ki triangulira dani mnogokotnik v mrežo, v kateri so vsi koti med 30° in 120° . Algoritem poda enolično (»uniform«) mrežo, kar pomeni, da so vsi trikotniki v grobem enake velikosti. Ta mreža je optimalna glede na velikost vseh enoličnih mrež. Vendar pa obstajajo primeri, ko mreža vsebuje veliko več trikotnikov kot je potrebno.

Drugi takšen algoritem je podal Ruppert (Ruppert, 1994). Začetno triangulacijo predstavlja kar osnovna Delaunayeva triangulacija točk V (začetna množica vsebuje vsa krajišča daljic in morebitne samostojne točke). V nadaljevanju se nato dodajajo točke k začetni triangulaciji iz dveh razlogov: da izboljšamo obliko trikotnika in da zagotovimo, da so vse daljice z vhoda prisotne tudi v Delaunayevi triangulaciji. Točke, ki jih dodajamo, so središča daljic in središča očrtanih krogov trikotnikov. Algoritem triangulira splošen ravninski graf tako, da imajo vsi trikotniki notranje kote med α in $\pi - 2\alpha$. Parameter α lahko izberemo med 0° in 20° . Velikost trikotnikov je različna. Število trikotnikov na izhodu je optimalno v okviru konstantnega faktorja, ki je odvisen od izbranega kota α .

Obstajajo še drugi algoritmi, ki pa so bolj zapleteni in ne temeljijo na Delaunayevi triangulaciji točk. Baker (Baker, 1988) je podal algoritem, ki triangulira notranjost enostavnega mnogokotnika s trikotniki, katerih notranji koti so med 13° in 90° . Vendar je lahko število tako tvorjenih trikotnikov zelo veliko. Bern, Eppstein in Gilbert (Bern, 1992) so izboljšali to lastnost tako, da so določili novo zgornjo mejo za število tvorjenih mnogokotnikov. Njihova triangulacija teče v $O(n \log n + k)$ času, kjer je n število stranic in k število trikotnikov. Bern, Dobkin in Eppstein (Dobkin, 1995) so pokazali, kako je mogoče triangulirati mnogokotnik z zagotovljeno kakovostjo. Zagotavljajo, da z uporabo $O(n \log n)$ trikotnikov lahko triangulirajo enostaven mnogokotnik s temi trikotniki tako, da največji kot ne bo večji od 150° . Takšno triangulacijo je možno doseči v času $O(n \log n)$.

Časovna zahtevnost	Avtor	Leto	Vhod
$O(n \log n + k)$	Bern, Eppstein	1991	Splošen
$O(n \log n)$	Bern, Dobkin	1995	Enostaven mnogokotnik
$O(n \log n)$	Bern, Mitchell	1995	Enostaven mnogokotnik
$O(M^2)$	Chew	1989	Enostaven mnogokotnik
$O(M^2)$	Ruppert	1994	Splošen

Tabela 3: Algoritmi triangulacije mnogokotnika z uporabo Steinerjevih točk

Še en algoritem so predstavili Bern, Mitchell in Ruppert (Bern, 1995). Njihov algoritem triangulira mnogokotnik z n oglišči (tudi mnogokotnik z luknjami), tako da noben trikotnik v končni triangulaciji ne meri več kot $\pi/2$. Število trikotnikov v takšni triangulaciji je samo $O(n)$. Algoritem teče v $O(n \log^3 n)$ času. Osnovna nova tehnika, uporabljena v algoritmu, je t.i. rekurzivno deljenje diskov. Tabela 3 (parameter M pomeni število trikotnikov na izhodu triangulacije) prikazuje algoritme, opisane v tem poglavju.

6. ZAKLJUČEK

Triangulacija enostavnih mnogokotnikov je pogosta naloga v računalniški geometriji in njenih aplikacijah: v geografskih informacijskih sistemih in generaciji mrež za metodo končnih elementov. Algoritme lahko razvrstimo v tri glavne skupine: algoritme z vstavljanjem diagonale, algoritme omejene Delaunayeve triangulacije in algoritme, ki temeljijo na Steinerjevih točkah. Iz vsake skupine algoritmov smo našli najznačilnejše algoritme. Opisali smo najznačilnejše karakteristike vsake skupine algoritmov: število trikotnikov na izhodu, kvaliteto triangulacije in možnost triangulacije trikotnikov z luknjami.

Razvoj algoritmov triangulacije je še vedno zanimivo področje raziskovanja. Velik izziv predstavlja Chazellov algoritem, ki temelji na vstavljanju diagonal. Čeprav je bil teoretično izdelan že leta 1991, vse do zdaj še ni na voljo enostavne in učinkovite implementacije. Nove algoritme lahko pričakujemo na področju omejene Delaunayeve triangulacije in triangulacije z uporabo Steinerjevih točk. Tudi sami razvijamo algoritem, ki bo učinkovito trianguliral mnogokotnike z velikim številom oglišč in visoko stopnjo vijugavosti.

Literatura

- Baker, B., Grosse, E., Afferty, C.,** *Nonobtuse triangulation of polygons.* *Discrete and Comp. Geometry*, 1988, št. 3, str. 147-168
- Bern, M., Eppstein, D., Gilbert, J. R.,** *Provably good mesh generation.* *Presented at the Proceedings of the 31st Annual Symposium on Foundation of Computer Science*, 1990, str. 231-241.
- Bern, M., Mitchell, S., Ruppert, J.,** *Linear-Size Nonobtuse Triangulation of Polygons.* *Discrete Comput. Geometry*, 1995, št.14, str. 411-428
- Boissonnat, J. D.,** *Shape reconstruction from planar cross sections.* *Comput. Vision Graphics Image Process.*, 1988, št. 44, str. 1-29
- Chazelle, B., Incerpi, J.,** *Triangulation and shape complexity.* *ACM Transactions on Graphics*, 1984, št. 3, str. 135-152
- Chazelle, B.,** *Triangulating a simple polygon in linear time.* *Discrete Computational Geometry*, 1991, št. 6, str. 485-524

- Chew, L. P.**, Constrained Delaunay triangulation. Presented at the Proceedings, Third ACM Symposium on Computational Geometry, 1987, Waterloo, str. 216-222.
- Chew, L. P.**, Guaranteed - quality triangular meshes. Technical report, No. TR-89-983, Cornell University, 1989.
- De Floriani, L., Puppo, E.**, An On-Line Algorithm for Constrained Delaunay Triangulation, *Graphical Models and Image Processing*, 1992, št. 3, str. 290-300
- Dobkin, D., Bern, M., Eppstein, D.**, Triangulating polygons without large Angles. *Computational Geometry & Applications*, 1995, št. 5, str.171-192
- Elgindy, H., Everett, H., Toussaint, G. T.**, Slicing an ear in linear time. *Pattern Recognition Letters*, 1993, št. 14, str. 719-722
- Garey, M.R., Johnson, D.S., Preparata, F. P., Tarjan, R. E.**, Triangulating a simple polygon, *Inform. Process.*, 1978, št. 7, str. 175-180
- Hertel, S., Mehlhorn, K.**, Fast triangulation of simple polygons. Presented at Proceedings 4th Internat. Conf. Theory, 1983, str. 207-215
- Kirkpatrick, D. G., Klawe, M. M., Tarjan, R. E.**, $O(n \log \log n)$ polygon triangulation with simple data structures. Presented at ACM Symposium on Computational Geometry, Berkeley, California, 1990, št. 6, str. 34-43
- Kong, X., Everett, H., Toussaint, G. T.**, The graham scan triangulates simple polygons. *Pattern Recognition Letters*, 1990, št. 11, str. 713-716
- Lenne, N. J.**, Theorems on the simple finite polygon and polyhedron. *American Journal of Mathematics*, 1911, št. 33, str. 37-62
- Meisters, G. H.**, Polygons have ears. *American Mathematical Monthly*, 1975, št. 82, str. 648-651
- O'Rourke, J.**, *Computational Geometry in C*. Cambridge University Press, 1994
- Ruppert, J.**, A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation. NASA Arnes Research Center, Submission to Journal of Algorithms, 1994, <http://jit.arc.nasa.gov/nas/abs.html>
- Seidel, R.**, A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons. *Computational Geometry: Theory and Applications*, 1991, št. 1, str. 51-64
- Shewchuk, J. R.**, Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. Carnegie Mellon University, 1999, <ftp://cs.cmu.edu>
- Tarjan, R. E., Van Wyk, C. J.**, An $O(n \log \log n)$ - time algorithm for a simple polygon triangulation and its evaluation. IEICE Technical report, št. PRU89-41, September 1989
- Toussaint, G. T.**, Efficient triangulation of simple polygons. *The Visual Computer*, 1991, št. 7, str. 280-295
- Žalik, B.**, Geometrijsko modeliranje. Fakulteta za elektrotehniko, računalništvo in informatiko Maribor, 1999

Recenzija: dr. Radoš Šumrada
mag. Dalibor Radovan

Prispelo v objavo: 2000-02-21