

Algoritmi za konstrukciju Voronoj dijagrama i
Deloneove triangulacije
Master teza

Elena Đurović 6/06

19. 10. 2008.

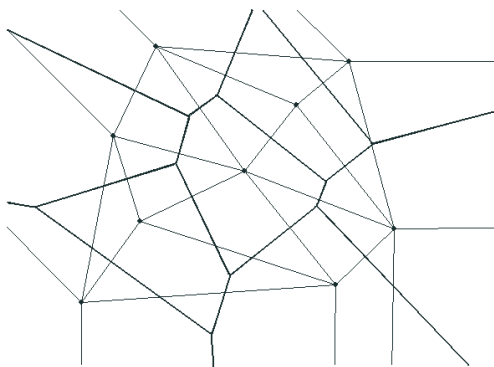
Sadržaj

1	Uvod	5
2	Voronoj dijagram	7
2.1	Definicija i osnovne karakteristike	8
2.2	Konstrukcija Voronoj dijagrama	11
2.2.1	Inkrementalni algoritam	11
2.2.2	Algoritam zasnovan na razlaganju	13
2.2.3	Algoritam pokretne linije	16
2.2.4	Implementacija algoritma pokretne linije	24
3	Deloneova triangulacija	27
3.1	Triangulacije planarnog skupa tačaka	29
3.2	Deloneova triangulacija	32
3.3	Konstrukcija Deloneove triangulacije	36
3.3.1	Algoritam zasnovan na razlaganju	36
3.3.2	Inkrementalni algoritam	41
3.3.3	Implementacija inkrementalnog algoritma	49
4	Zaključak	51

U radu se razmatraju dva geometrijska problema u ravni: konstrukcija Voronoi dijagrama (takozvani *poštanski problem*), odnosno Deloneove triangulacije datog skupa tačaka u ravni.

Za dati skup tačaka u ravni, koje ćemo u daljem tekstu nazvati *centrima*, konstrukcija Voronoi dijagrama podrazumeva podelu ravni na ćelije, tako da je svakoj tački unutar jedne ćelije najbliži onaj centar koji se nalazi u toj ćeliji. Naime, svaka ćelija definisana je jednim centrom, pa se rešenje ovog problema može primeniti, na primer, na poštanski problem: ako se znaju položaji pošta u nekoj oblasti, odrediti za svakog korisnika koja pošta mu je najbliža.

Deloneov graf nad datim skupom tačaka u ravni je dualan graf Voronoi dijagramu. Drugim rečima, čvorovi ovog grafa su centri, a grane povezuju čvorove, koji odgovaraju centrima susednih Voronoi ćelija. Na slici 1.1 su prikazani Voronoi dijagram (podebljane linije) i Deloneova triangulacija (obične linije) skupa od 9 centara.



Slika 1.1: Voronoi dijagram i Deloneova triangulacija

Voronj dijagram

Pretpostavimo da menadžer lanca supermarketa planira da otvori nove prodavnice na određenoj teritoriji. Za procenu da li će novi objekti biti profitabilni potrebno je odrediti koliko će kupaca privući pojedine prodavnice. Zbog toga menadžer pravi model ponašanja potencijalnih kupaca: kako kupci odlučuju gde će kupovati namirnice? Posmatrajući ovaj problem nešto apstraktnije, može se razmatrati skup objekata na zadatim lokacijama (nazovimo ih *centri*), koji nude usluge. Za svaki centar želimo da znamo gde žive ljudi - korisnici odgovarajućeg objekta.

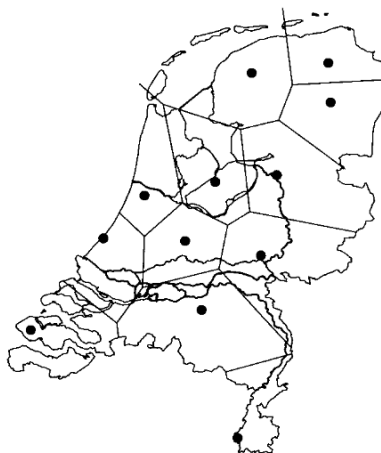
U računarskoj geometriji, ovaj problem je poznat kao *poštanski problem*: sa koje pošte korisnici šalju svoja pisma? Da bismo dalje analizirali ovaj problem polazimo od sledećih pretpostavki:

- cena usluge je ista u svakom centru;
- cena usluge se uvećeva se za cenu transporta do centra;
- cena transporta do centra proporcionalna je Euklidskom rastojanju do centra;

Ove idealizovane pretpostavke omogućuju grubu procenu tržišnih oblasti pojedinih centara.

Geometrijski se ovakav model može predstaviti na sledeći način. Pretpostavke, koje smo izneli, indukuju podelu razmatrane oblasti u manje regione - tržišne oblasti posmatranih centara, tako da ljudi koji žive u istom regionu, koriste usluge istog centra. Pretpostavlja se da ljudi koriste usluge u najbližem centru. Prema tome tržišna oblast datog centra pokriva sve tačke od kojih je taj centar bliži nego bilo koji drugi. U primeru sa slike 2.1 centri su glavni gradovi 12 provincija Holandije.

Model kojim se svakoj tački dodeljuje najbliži centar zove se *Voronj model dodeljivanja*. Podela indukovana ovim modelom zove se *Voronj dijagram* [1] skupa centara. Iz Voronj dijagrama mogu se dobiti razne informacije o tržišnim



Slika 2.1: Tržišne oblasti Holandije, pri čemu su centri glavni gradovi provincija

oblastima, na primer o njihovoj konkurentnosti u graničnim oblastima. Međutim, ovi dijagrami se ne koriste samo u tržišnoj ekonomiji, već i u fizici, astronomiji, robotici, i mnogim drugim oblastima.

2.1 Definicija i osnovne karakteristike

Označimo Euklidsko rastojanje između tačaka p i q , odnosno dužinu duži \overline{pq} , sa $dist(p, q)$. U dvodimenzionj ravni važi:

$$dist(p, q) := \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}.$$

Definicija 1 Neka je $P := \{p_1, p_2, \dots, p_n\}$ skup n različitih centara u ravni. Definišimo Voronj dijagram $Vor(P)$ skupa P kao podebu ravni na n ćelija $\mathcal{V}(p_i)$, od kojih svaka odgovara jenom centru $p_i, i = 1, 2, \dots, n$. Pri tome za svaku ćeliju važi $\mathcal{V}(p_i) = \{q \mid dist(p_i, q) < dist(p_j, q), \text{ za svako } j \neq i\}$.

Ponekad ćemo koristiti 'Vor(P)' ili 'Voronj dijagram' samo da ukažemo na ivice ili temena podele ravni. Tako, kada kažemo da je Voronj dijagram povezan¹, želimo reći da skup ivica i temena formira povezan skup. U odnosu na terminologiju uvodnog dela, $\mathcal{V}(p_i)$ je tržišna oblast centra p_i .

Razmotrimo detaljnije Voronj dijagram. Za dve tačke u ravni p i q , definišimo simetralu para (p, q) kao simetralu duži \overline{pq} . Simetrala deli ravan na dve poluravni. Otvorenu poluravan koja sardži p obeležimo sa $h(p, q)$, a otvorenu poluravan koja sadrži q obeležimo sa $h(q, p)$. Primitimo sada da važi $r \in h(p, q)$ ako

¹Za dve tačke kažemo da su povezane, ako postoji poligonalna linija, koja ih spaja.

i samo ako $\text{dist}(r, p) < \text{dist}(r, q)$. Iz ovoga sledi (slika 2.2 a)):

$$\mathcal{V}(p_i) = \bigcap_{1 \leq j \leq n, j \neq i} h(p_i, p_j).$$

Pošto je $\mathcal{V}(p_i)$ presek $n - 1$ poluravni, ćelija $\mathcal{V}(p_i)$ je otvorena konveksna oblast (moguće neograničena), ograničena najviše sa $n - 1$ temenom i $n - 1$ ivicom.

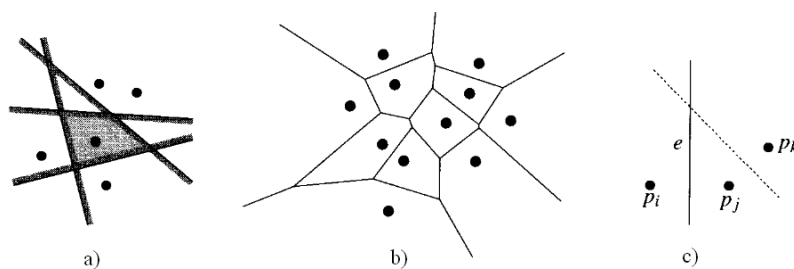
Kako izgleda kompletan Voronoi dijagram? Upravo smo rekli da je svaka ćelija presek nekoliko poluravni. Tako je i ceo Voronoi dijagram podela ravni na ćelije čije ivice su duži ili poluprave. U specijalnom slučaju, kada su centri kolinearni, ivice su paralelne prave:

Teorema 1 *Neka je $n \geq 3$ i neka je P skup n centara u ravni. Ako su svi centri kolinearni, $\text{Vor}(P)$ čini n paralelnih linija. Ako pak centri nisu kolinearni, $\text{Vor}(P)$ je povezan skup ivica koje su ili duži ili poluprave.*

Dokaz: Prvi deo teoreme se lako dokazuje. Pretpostavimo zato da nisu svi centri kolinearni.

Dokažimo najpre da su ivice duži ili poluprave. Već znamo da su ivice linijski segmenti. Naime, ivice su delovi simetrala duži između centara. Pretpostavimo sada suprotno, da postoji ivica e , koja je prava, a ne linijski segment (slika 2.2 c)). Neka je e simetrala ćelija duži $\overline{p_i p_j}$ i neka je $p_k \in P$ centar, koji nije kolinearan ni sa jednim od centara p_i i p_j . Tada ni simetrala duži $\overline{p_j p_k}$ nije paralelna sa e , a to znači da seče e . Ali deo prave e koji leži unutar poluravni $h(p_k, p_j)$ ne može biti granica ćelije $\mathcal{V}(p_j)$, zato što je bliži centru p_k nego centru p_j . Kontradikcija!

Dokažimo još da je skup $\text{Vor}(P)$ povezan. Ako ovo ne bi važno postojala bi Voronoi ćelija $\mathcal{V}(p_i)$ koja bi delila ravan na dva dela. S obzirom da su Voronoi ćelije konveksne, $\mathcal{V}(p_i)$ bi bio pojas ograničen dvema paralelnim pravama. Međutim upravo smo dokazali da ivice ćelija nisu prave, pa ponovo dolazimo do kontradikcije! \square



Slika 2.2: linijski segment

Razmotrimo sada kompleksnost Voronoi dijagrama, odnosno ukupan broj temena i ivica.

Teorema 2 *Za $n \geq 3$, broj temena u Voronj dijagramu skupa od n centara u ravni je najviše $2n - 5$, a broj ivica je najviše $3n - 6$.*

Dokaz: Ako su svi centri kolinearni teorema se lako dokazuje na osnovu prethodne teoreme. Pretpostavimo zato da centri nisu kolinearni. Na osnovu Ojlerove teoreme za savki povezani planarni graf² sa m_v čvorova, m_e ivica i m_f površi važi sledeća jednakost:

$$m_v - m_e + m_f = 2.$$

Ovu toremu ne možemo primeniti direktno na $Vor(P)$. Naime, neke ivice su mu poluprave, te zato ovaj graf nije odgovarajući graf. Ali ako dodamo jedno teme v_∞ u beskonačnosti i sve polurave povežemo sa tim temenom, dobili smo graf na koji možemo da primenimo Ojlerovu toremu. To znači da za n_v temena, n_e ivica i n centara Voronj dijagrama $Vor(P)$ važi:

$$(n_v + 1) - n_e + n = 2. \quad (2.1)$$

Svaka ivica ima dva temena. Ako sumiramo stepene svih temena, dobićemo dupli broj ivica. Međutim, svako teme uključujući i v_∞ ima stepen najmanje tri. Zato važi:

$$2n_e \geq 3(n_v + 1). \quad (2.2)$$

Sada se iz jednakosti (2.1) i nejednakosti (2.2) lako dokazuje teorema. \square

Znamo da su ivice Voronj dijagrama delovi simetrala duži koje povezuju centre, a temena presečne tačke simetrala. Međutim, ne određuju sve simetrane delove ivica, niti su sve presečne tačke simetrala temena u $Vor(P)$. Uvedimo zato sledeću definiciju.

Definicija 2 *Najveća prazna kružnica za tačku q u P , obeležimo je sa $C_P(q)$, je najveća kružnica sa centrom u q , koja ne sadrži ni jednu tačku iz P unutar kruga.*

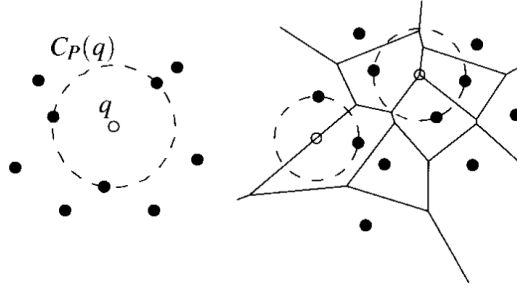
Sledeća teorema karakteriše temena i ivica Voronj dijagrama, kao što je prikazano slikom 2.3.

Teorema 3 *Za Voronj dijagram $Vor(P)$ skupa centara P , važi sledeće:*

- *tačka q je teme Voronj dijagrama ako i samo ako najveća prazna kružnica $C_P(q)$ sadrži tri ili više centara (na kružnici, a ne unutar kruga).*
- *simetrana duži $\overline{p_i p_j}$ određuje ivicu u $Vor(P)$ ako i samo ako postoji tačka q , na simetrali, takva da kružnica $C_P(q)$ sadrži p_i i p_j , ali ne sadrži nijedan drugi centar.*

Dokaz: Dokažimo redom:

²Za graf kažemo da je planaran, ako ne postoje dve ivice koje se međusobno seku.

Slika 2.3: Najveća prazna kružnica za tačku q u P

- Pretpostavimo da postoji tačka q takva da $C_P(q)$ sadrži tri ili više centara. Neka su p_i, p_j i p_k ta tri centra. Kako unutar $C_P(q)$ ne postoji nijedan drugi centar, q mora biti na granici svake od ćelija $\mathcal{V}(p_i), \mathcal{V}(p_j)$ i $\mathcal{V}(p_k)$ i q mora biti teme u $Vor(P)$.
S druge strane, svako teme q iz $Vor(P)$ je krajnja tačka za najmanje tri ivice, a samimi tim to teme je teme i za najmanje tri ćelije $\mathcal{V}(p_i), \mathcal{V}(p_j)$ i $\mathcal{V}(p_k)$. Teme q mora biti jednako udaljeno od p_i, p_j i p_k i ne može da postoji nijedan centar bliži q , jer u suprotnom ćelije $\mathcal{V}(p_i), \mathcal{V}(p_j)$ i $\mathcal{V}(p_k)$ ne bi delile zajedničko teme q . To znači da unutrašnjost kruga na čijoj ivici su p_i, p_j i p_k ne sadrži nijedan drugi centar.
- Pretpostavimo da postoji tačka q . Kako $C_P(q)$ ne sadrži nijedan centar u svojoj unutrašnjosti, a p_i i p_j su na kružnici, to važi sledeće $dist(q, p_i) = dist(q, p_j) \leq dist(q, p_k)$ za svako $1 \leq k \leq n$. Iz ovog sledi da q leži ili na ivici ili na temenu $Vor(P)$. Na osnovu prvog dela teoreme q ne može biti teme u $Vor(P)$. Znači q leži na ivici, koja je određena simetralom duži $\overline{p_i p_j}$.
Obrnuto, neka simetrala duži $\overline{p_i p_j}$ definiše ivicu u $Vor(P)$. Najveća prazna kružnica sa centrom u bilo kojoj tački ove ivice sadrži centre p_i i p_j i nijedan drugi centar. \square

2.2 Konstrukcija Voronoi dijagrama

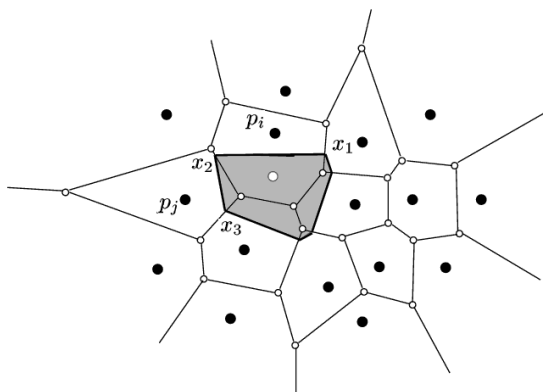
U ovom odeljku biće izloženi algoritmi za konstrukciju Voronoi dijagrama. Prvi od njih je inkrementalni algoritam. Potom se analizira strategija razlaganja, te algoritam pokretne linije.

2.2.1 Inkrementalni algoritam

Inkrementalni algoritam [4] proučavali su najpre Grin i Sibson (*eng.* Green & Sibson). Voronoi dijagram $Vor(P)$, gde je $P := \{p_1, p_2, \dots, p_n\}$ bi se konstruisao od $Vor(P \setminus \{p\})$ ubacivanjem tačke p .

Inkrementalni algoritam počinje od jednostavnog Voronj dijagrama sa dva ili tri centra, a potom se dijagram modifikuje dodavanjem jednog po jednog novog centra. Za $m = 1, 2, \dots, n$, obeležimo sa Vor_m Voronj dijagram od m centara p_1, p_2, \dots, p_m . Potom algoritma simulira prelazak sa Vor_{m-1} na Vor_m za svako m .

Osnovna ideja inkrementalnog algoritma je sledeća. Pretpostavimo da već imamo izgrađen Voronj dijagram Vor_{m-1} , kao što je prikazano tanjim linijama na slici 2.4, a potom dodajemo novi centar p_m . Najpre pronalazimo centar, neka je to p_i , čija ćelija sadrži novi centar p_m . Nađimo simetralu između centara p_i i p_m i obeležimo je sa $B(p_i, p_m)$. Ova simetrala seče granične ivice $\mathcal{V}(p_i)$ ćelije u dvema tačkama x_1 i x_2 . Presečne tačke označimo tako da x_1x_2 grade usmerenu duž, takvu da se centar p_m nalazi levo od nje. Ovaj vektor deli ćeliju na dva dela. Levi deo pripadaće novoj ćeliji $\mathcal{V}(p_m)$, a ovim smo dobili i novu ivicu ćelije $\mathcal{V}(p_i)$.



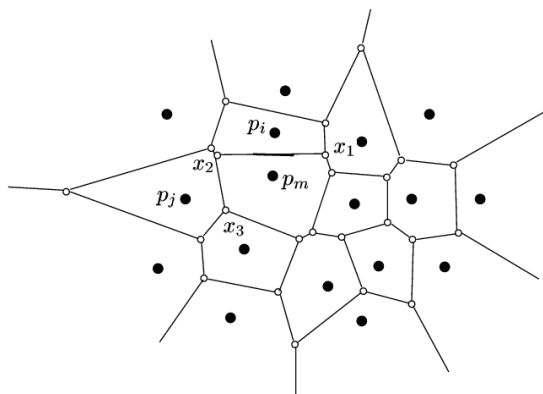
Slika 2.4: Inkrementalni algoritma za konstrukciju Voronj dijagrama

Počevši od ove ivice proširimo granicu nove ćelije na sledeći način. Kako $B(p_i, p_m)$ seče ivicu ćelije $\mathcal{V}(p_i)$ u x_2 , simetrala tu seče i ćeliju susedenu ovoj ćeliji i neka je to ćelija $\mathcal{V}(p_j)$. Zatim nalazimo simetralu $B(p_m, p_j)$ između centra p_j i p_m , a time i novu presečnu tačku x_3 , koja seče ivicu ćelije $\mathcal{V}(p_j)$. Slično, nalazimo segmente povezanih simetrala, susednih centara, sve dok ne dodemo do početne tačke x_1 . Označimo ovu sekvencu sa $(x_1x_2, x_2x_3, \dots, x_{m-1}x_m, x_mx_1)$. Ona formira lanac obrnut od kazaljke na satu, koji zapravo gradi ivice nove Voronj ćelije centra p_m . Ostaje još da se iz nove ćelije obrišu ostaci Vor_{m-1} Voronj dijagrama. Dobijeni Vor_m prikazan je slikom 2.5.

Jasno je da sekvencu $(x_1x_2, x_2x_3, \dots, x_{m-1}x_m, x_mx_1)$ gradi povezanu poligonalnu liniju, pa je samim tim algoritam korektan.

Kako je $n = |P|$, ćelija centra $p_i \in P$ može da ima do $n - 1$ ivica. To znači da je vreme potrebno za izvršavanje ovog algoritma $\mathcal{O}(n^2)$. Stoga važi sledeća lema.

Lema 1 *Vremenska složenost algoritma je $\mathcal{O}(n^2)$, a prostorna $\mathcal{O}(n)$.*



Slika 2.5: Nakon što je unesen novi centar u Voronoj dijagram

2.2.2 Algoritam zasnovan na razlaganju

Šamos i Hoey (*eng.* Shamos & Hoey) predstavili su prvi $\mathcal{O}(n \log n)$ deterministički algoritam za računanje Voronoj dijagrama u ravni. Ovaj algoritam koristi strategiju *razlaganja* ili kako se još zove strategiju *zavadi pa vladaj* (*eng.* divide & conquer) [3], [2], koja je jedna od osnovnih paradigmi za dizajniranje efikasnih algoritama. Ovom strategijom problem se deli na nekoliko jednostavnijih podproblema, koji su približno jednake veličine, a rešenje originalnog problem se dobija *povezivanjem* rešenja podproblema. Ovim pristupom Šamos i Hoey su skup centara P vertikalnom linijom podelili na dva podskupa P_L i P_R približno iste veličine. Zatim se $Vor(P_L)$ skupa P_L i $Vor(P_R)$ skupa P_R grade rekursivno.

Algoritam VoronojDijagram (P)

Ulaz. Za $n > 3$ skup $P := \{p_1, p_2, \dots, p_n\}$ skup centara u ravni.

Izlaz. Voronoj dijagram $Vor(P)$.

1. Sortirati tačke skupa P rastuće u donosu na x -koordinatu.
2. Neka je t ceo deo od $n/2$. Podeliti P na $P_L := \{p_1, p_2, \dots, p_t\}$ i $P_R := \{p_{t+1}, p_{t+2}, \dots, p_n\}$.
3. Konstruisati $Vor(P_L)$ i $Vor(P_R)$ rekursivno.
4. Povezati $Vor(P_L)$ i $Vor(P_R)$ u $Vor(P)$ algoritmom **PovežiVoronoj**.
5. return $Vor(P)$

Definicija 3 Označimo sa $B(P_L, P_R)$ skup ivica iz $Vor(P)$, koje razdvajaju centre iz P_L i ćelije iz $Vor(P_R)$.

Korak *povezivanja* uključuje računanje skupa ivica $B(P_L, P_R)$. Ideja se zasniva na sledećoj lemi.

Lema 2 *Ivice u $B(P_L, P_R)$ formiraju y -monotoni poligonijalni lanac. U $Vor(P)$, ćelije svih centara u P_L su levo od $B(P_L, P_R)$, dok su ćelije svih centara u P_R desno.*

Dokaz: Neka je b proizvoljna ivica iz $B(P_L, P_R)$ i neka su $l \in P_L$ i $r \in P_R$ centri ćelija koje odgovaraju ivici b . Kako l ima manju x -koordinatu od r , b ne može biti horizontalna i ćelija kojoj odgovara l mora biti sa leve strane. \square

Definicija 4 *Neka su L i R dva skupa tačaka u ravni, takvi da sve tačke u L imaju manju x -koordinatu od tačaka u R . Potpornom linijom skupova L i R , nazivamo duž, koja je deo tangente skupova L i R . Sve tačke skupova nalaze se sa jedne strane tangente ili joj pripadaju. Potporna linija je duž, čije jedno teme je tačka iz L , a drugo iz R . Ukoliko tangenta sadrži više temena skupa L , bira se ono sa najvećom x -koordinatom. Ukoliko tangenta sadrži više temena skupa R , bira se ono sa najmanjom x -koordinatom.*

Da bise od $Vor(P_L)$ i $Vor(P_R)$ konstruisao $Vor(P)$, najpre se određuje početna ivica poligonijalnog lanca. Ta ivica je poluprava, odnosno jedno njeno teme je u beskonačnosti. A određena je kao simetrala donje potporne linije (slika 2.6).

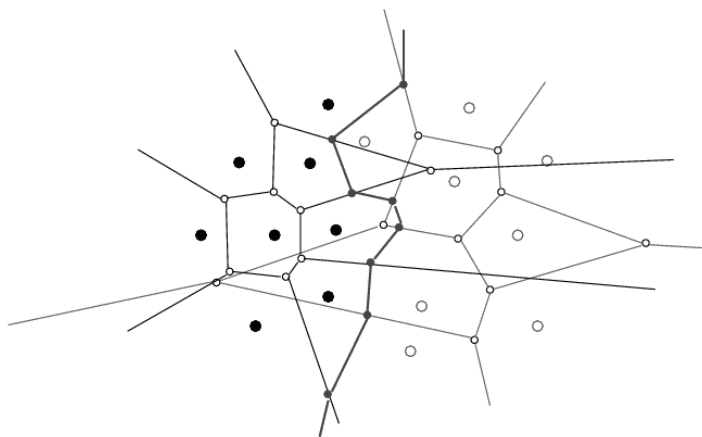
Algoritam PovežiVoronj (P)

Ulaz. Voronj dijagrami $Vor(P_L)$ i $Vor(P_R)$ skupa P_L i P_R .

Izlaz. Voronj dijagram skupa $P = P_L \cup P_R$.

1. Naći donju potpornu liniju algoritmom **DonjaPotpornaLinija**
2. $w_0 \leftarrow$ tačka u negativnoj beskonačnosti duž uspravne simetrale donje potporne linije čija temena su p_L iz P_L i p_R iz P_R .
 $i \leftarrow 0$
3. **while** potporna linija $\overline{p_L p_R}$ nije gornja potporna linija
 - (a) $i \leftarrow i + 1$
 - (b) Presečnu tačku $B(P_L, P_R)$ i ivice $Vor(P_L)$ označiti sa a_L .
 - (c) Presečnu tačku $B(P_L, P_R)$ i ivice $Vor(P_R)$ označiti sa a_R .
 - (d) **if** ($a_L.y < a_R.y$)
then
 $w_i \leftarrow a_L$
 $p_L \leftarrow$ centar sa druge strane Voronj ivice, koja sadrži a_L .
else
 $w_i \leftarrow a_R$
 $p_R \leftarrow$ centar sa druge strane Voronj ivice, koja sadrži a_R .
4. $m \leftarrow i$
 $w_{m+1} \leftarrow$ tačka u pozitivnoj beskonačnosti duž uspravne simetrale duži $\overline{p_L p_R}$.

5. Dodati poligonalnu liniju $(w_0w_1, w_1w_2, \dots, w_mw_{m+1})$ i obrisati iz $Vor(P_L)$ delove desno od poligonalne linije, a iz $Vor(P_R)$ delove levo od poligonalne linije.
6. **return** Voronoj dijagram



Slika 2.6: Poslednja iteracija u građenju linije razdvajanja levog i desnog Voronoj dijagrama, pre nego se obrišu njihovi nepotrebni delovi

Izložimo sad algoritam za konstrukciju donje potporne linije.

Algoritam DonjaPotpornaLinija (P)

Ulaz. Dva skupa tačaka P_L i P_R , takva da sve tačke iz P_L imaju x -koordinatu manju od tačaka iz P_R .

Izlaz. Duž, čija su temena $u \in P_L$ i $v \in P_R$ i koja je donja potporna linija poligona P_L i P_R

1. Naći tačaku u iz P_L sa najvećom x koordinatom i v iz P_R sa najmanjom x koordinatom.
2. Ponavljati sledeće korake naizmenično
 - (a) **while** teme $next[u]$ niže od \overline{uv}
 $u \leftarrow next[u]$
 - (b) **while** teme $next[v]$ niže od \overline{uv}
 $v \leftarrow next[v]$
3. **return** \overline{uv} .

Važni delovi algoritma su nalaženje linije podele i spajanje $Vor(P_L)$ i $Vor(P_R)$ u $Vor(P)$.

Naime, računanje horizontalne ili vertikalne linije podele je veoma jednostavno, ako su centri sortirani po x i y koordinati. Bilo koji optimalan algoritam, na primer *heap sort* ili *merge sort* izvršavaju se u $\mathcal{O}(n \log n)$ vremenu.

Koraci 1 i 5 algoritma **PovežiVornoj** se izvršavaju za $\mathcal{O}(n)$, koraci 2 i 4 se izvršavaju u konstantnom vremenu, a korak 3 se ponavlja najviše $\mathcal{O}(n)$ puta, pa je ukupno vreme potrebno za izvršavanje algoritma **PovežiVornoj** $\mathcal{O}(n)$. Ovim dolazimo do sledeće leme.

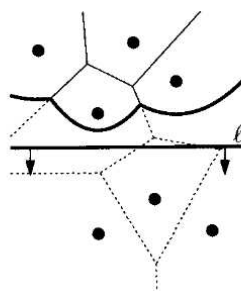
Lema 3 *Vremenska složenost algoritma je $\mathcal{O}(n \log n)$.*

2.2.3 Algoritam pokretne linije

Fortunov algoritam ili algoritam pokretne linije [1] ime je dobio po svom autoru, a Voronoj dijagram konstruiše za vreme $\mathcal{O}(n \log n)$.

Strategija ovog algoritma zasniva se na pomeranju *horizontalne pokretne linije* od vrha prema dnu ravni. Dok se linija pomera ka dnu ravni, pamte se informacije o preseccima sa pokretnom linijom, potrebne za izračunavanje Voronoj dijagrama.

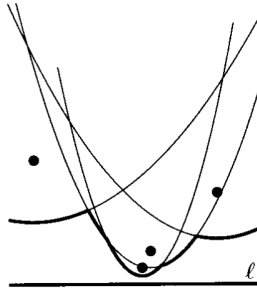
Primenimo ovu strategiju na računanje Voronoj dijagrama skupa tačaka, odnosno centara $P := \{p_1, p_2, \dots, p_n\}$ u ravni. Pomerajmo, shodno opisanoj strategiji, pravu l od vrha ka dnu. Ova strategija, podrazumeva i čuvanje informacije o preseccima sa pokretnom linijom u Voronoj dijagramu, kao što smo prethodno rekli. To, ipak, nije tako jednostavno, zato što $\text{Vor}(P)$ iznad l ne zavisi samo od centara koji leže iznad l , već i onih koji leže ispod l . Drugim rečima, kada pokretna linija dostigne najviše teme ćelije $\mathcal{V}(p_i)$, još nije dostigla centar p_i (slika 2.7). S obzirom da nemamo sve informacije potrebne za računanje temena, moramo malo da izmenimo strategiju. Umesto čuvanja informacija o preseccima sa pokretnom linijom, čuvaćemo informacije o delovima Voronoj dijagrama koji su iznad l i koji ne mogu biti promenjeni centrima ispod l .



Slika 2.7: Pomeranje pokretne linije

Obeležimo zatvorenu poluravan iznad l sa l^+ . Koji deo Voronoj dijagrama iznad l ne može biti promenjen? Odnosno, za koju tačku $q \in l^+$ sigurno znamo koji joj je najbliži centar? Rastojanje tačke q i bilo kog centra ispod l je veće nego rastojanje q i l . Tako, najbliži centar tački q ne može da leži ispod l ako

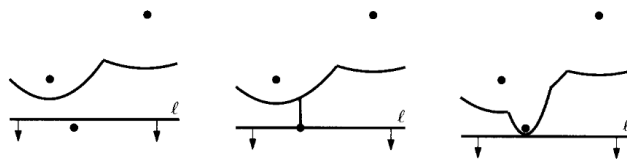
već postoji centar $p_i \in l^+$, koji je na rastojanju ne većem od rastojanja q i prave l . Položaji tačaka koje su na rastojanju bližem nekom centru iznad l nego rastojanju do prave l ograničeni su parabolama, kao na slici 2.8. Graničnu liniju koju grade ove parabole zovemo *obalska linija*. Svaki centar p_i iznad pokretne linije gradi parabolu β_i . Obalska linija, tako, za svaku x -koordinatu, prolazi kroz najnižu tačku svake parabole. Zato je obalska linija x -monotona funkcija, odnosno svaka vertikalna linija je preseca u tačno jednoj tački.



Slika 2.8: Parabole iznad pokretne linije

Primitimo da presečne tačke parabola iscrtavaju ivice Vorinjoj dijagrama, dok se pokretna linija pomera ka dnu. Umesto pamćenja ovih preseka mi pamtimmo obalsku liniju, dok pomeramo l .

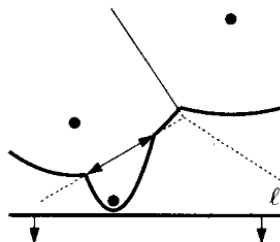
Struktura obalske linije se stalno menja, pomeranjem prave l . Naime, pojavljuju se nove parabole, a već postojeće se smanjuju te potom i nestaju. Razmotrimo najpre, slučaj kad se pojavljuje nova parabola. Kada pokretna linija dosegne novi centar, nova parabola je najpre degenerisana parabola, odnosno linijski segment koji povezuje novi centar i obalsku liniju. Kako se pokretna linija pomera ka dole, tako nova parabola postaje sve šira i šira. Deo nove parabole, postaje deo stare obalske linije. Momenat kada se ubacuje novi centar zovemo *događaj centra*. Slika 2.9 prikazuje proces.



Slika 2.9: Novi luk se pojavljuje u obalskoj liniji pojavom novog centra

Šta se dešava sa Voronj dijagramom za vreme događaja centra? S obzirom da presečne tačke na obalskoj liniji iscrtavaju ivice Voronj dijagrama, pojavljuju se dve nove presečne tačke. Nove presečne tačke se najpre poklapaju, a potom se pomeraju u suprotnim pravcima ispisujući ivicu. Inicijalno, ova ivica nije povezana sa ostalim ivicama dijagrama, ali pomeranjem prave l ona postaje

sve veća i veća, te se potom povezuje sa drugom ivicom i postaje povezana sa ostatkom dijagrama, kao na slici 2.10.



Slika 2.10: Nova parabola se veže za ostatak dijagrama

Da li je moguće da se parabola pojavi u obalskoj liniji i na jedan drugi način?

Lema 4 *Jedini način da se novi luk pojavi u obalskoj liniji je događaj centra.*

Dokaz: Pretpostavimo suprotno, da već postojeća parabola β_j , određena centrom p_j , koja seče obalsku liniju. Novi luk se može desiti na jedan od dva načina.

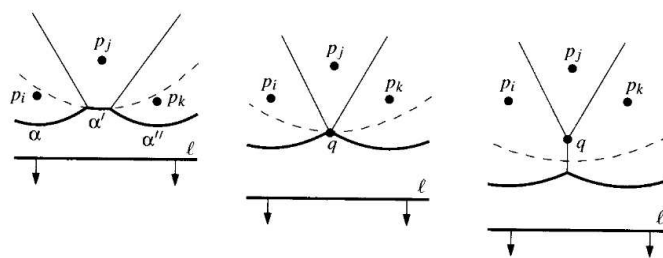
Prvi je da ova parabola seče obalsku liniju u jednoj, a drugi u dvema tačkama. Dokaz se svodi na utvrđivanje kontradikcije u oba slučaja. \square

Posledica ove leme je da se obalska linija u svakom trenutku sastoji od najviše $2n - 1$ lukova parabole.

Proces pri kretanju pokretne linije je smanjivanje paraboličkog luka te njegovo postepeno nestajanje, što je prikazano na slici 2.11. Neka je α' luk parabole, koja nestaje, a α i α'' dva njoj susedna luka. Tri luka α , α' i α'' su određena trima centrima p_i , p_j i p_k . U momentu nestanka parabole α' sve tri parabole prolaze kroz istu tačku q . Ova tačka je na jednakom rastojanju od prave l i triju centara. Zbog toga, postoji kružnica sa centrom u q , koja sadrži p_i , p_j i p_k , a njena najniža tačka leži na pravoj l . Neka druga centar-tačka Voronj dijagrama unutar kruga ne može da postoji, jer bi takva tačka bila na manjem rastojanju do centra kruga q , nego što je rastojanje tačkaka q i l , što je u kontradikciji sa činjenicom da se q nalazi na obalskoj liniji. Iz ovoga dobijamo da q postaje novo teme Voronj dijagrama. Tako, nestankom luka iz obalske linije dve ivice se spajaju i nastaje novo teme Voronj dijagrama. Događaj kada pokretna linija sadrži najnižu tačku prethodno opisane kružnice nazivamo *događaj kruga*. Važi i sledeća lema:

Lema 5 *Jedini način, dakle, da postojeći luk parabole nestane iz obalske liniji je događaj kruga.*

Nakon ovakvog proučavanja promena obalske linije potrebno je još naći odgovarajuće strukture podataka za čuvanje neophodnih informacija za vreme



Slika 2.11: Novi luk nestaje iz obalske linije

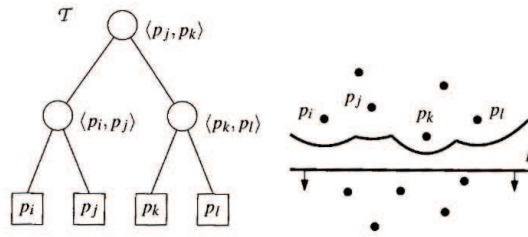
pomeranja pokretne linije. Kako je naš cilj računanje Voronoj dijagrama, potrebna nam je struktura koja čuva već određene delove dijagrama. Takođe, potrebne su nam dve standardne strukture podataka za algoritam pokretne linije. To su niz koji čuva događaje i struktura koja predstavlja status pokretne linije. Ova poslednja struktura koristiće se za predstavljanje obalske linije.

- Voronoj dijagram koji gradimo čuvaćemo u dvostrukopovezanoj listi ivica. Kako bismo poluprave ivice prikazali prikladno ograničićemo ravan razmatranja pravougaonim ramom.
- Obalsku liniju predstavimo uravnoteženim binarnim drvetom pretraživanja \mathcal{T} , kao na slici 2.12. Listovi odgovaraju lukovima obalske linije, koja je monotona na sledeći način: najlevli list predstavlja najlevli luk, sledeći luk predstavlja, drugi najlevli itd. Svaki list, takođe čuva i centar kojim je luk parabole definisan. Unutrašnji čvorovi drveta \mathcal{T} predstavljaju presečne tačke obalske linije. Te presečne tačke su određene susednim parom centara (p_i, p_j) , gde p_i definiše parabolu levo od presečne tačke, a p_j definiše parabolu desno od presečne tačke. Primitimo da se same parabole ne čuvaju.

U drvetu \mathcal{T} čuvamo i pokazivače na druge dve strukture za vreme pomeranja pokretne linije. Naime svaki list u \mathcal{T} čuva i pokazivač na čvor u nizu događaja, odnosno, događaj kruga kojim će parabola da nestane. Ovaj pokazivač je *null* ako događaj kruga ne postoji ili još nije primećen. Konačno i svaki unutrašnji čvor ima pokazivač na dvostruko povezanu listu ivica Voronoj dijagrama. Preciznije, pokazivač na jednu ivicu iscertanu početnom tačkom koju taj čvor predstavlja.

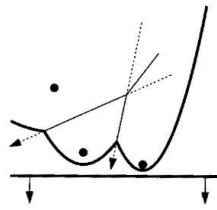
- Niz događaja Q predstavlja niz sa prioritonom, gde je prioritet događaja određen njegovom y -koordinatom. Ovde se čuvaju već poznati događaji. Centar događaja je sačuvan samim centrom. Za događaj krug čuvamo najnižu tačku kruga sa pokazivačem na list u \mathcal{T} koji predstavlja parabolu, koja će nestati ovim događajem.

Sve centre znamo unapred, ali događaje kruga ne. Razmotrimo sada detektovanje događaja kruga.

Slika 2.12: Binarno drvo pretraživanja \mathcal{T}

Pomeranjem pokretne linije, obalska linija menja svoju strukturu. Nove parabole nastaju, a stare nestaju. Može se desiti da nastanu nove tri parabole u nizu. Algoritam mora moći detektovati da li će svake tri parabole definisati mogući događaj kruga. Potencijalni događaji se čuvaju u nizu događaja Q . Postoje dve dodatane mogućnosti. Prva, presečne tačke triju susednih parabola ne konvergiraju. To znači da su pravci kojima se pomeraju takvi da se nikad neće sresti. Druga, čak i ako presečne tačke triju parabola konvergiraju događaj kruga se neće desiti, jer trojka parabola nestaje (možda zbog pojave novog centra na obalskoj liniji) pre događaja. U ovom slučaju događaj zovemo *lažni alarm uzbune*.

Za svaki novi događaj algoritam proverava sve nove trojke susednih lukova. Slikom 2.13 prikazana je trojka lukova. Na primer, za događaj centra dobijamo tri nove trojke lukova, jednu gde je novi luk levi luk trojke, jednu gde je srednji i jednu gde je desni. Kada nova trojka ima konvergirajuće presečne tačke događaj se ubacuje u niz događaja Q . Specijalan slučaj događaja centra je kada je novi luk u sredini, ali presečne tačke neće nikada konvergirati, zato što i levi i desni luk trojke pripadaju jednoj paraboli, pa presečne tačke mogu samo da divergiraju. Prilikom nestajanja trojke parabola, proverava se da li postoji odgovarajući događaj u Q . Ako postoji, onda je to lažni alarm uzbune, pa se događaj briše iz Q . Ovo jednostavno može biti implementirano koristeći pokazivače iz listova u \mathcal{T} ka odgovarajućim događajima kruga u Q .



Slika 2.13: Trojka lukova

Lema 6 *Svako teme u Voronoi dijagramu detektuje se događajem kruga.*

Dokaz: Neka je q razmatrano teme dijagrama, a p_i , p_j i p_k centri kroz koje prolazi kružnica $C(p_i, p_j, p_k)$. Takva kružnica i takva tri centra postoje na osnovu teoreme 3. Zbog jednostavnosti dokaza posmatramo slučaj kada na kružnici ne leže drugi centri i tačka kružnice sa najmanjom x -koordinatom nije nijedan od centara. Pretpostavimo još da ako krenemo od tačke sa najmanjom x -koordinatom po kružnici u pravcu obrnutom od kazaljke na satu centri su raspoređeni sledećim redom p_i, p_j, p_k .

Treba da pokažemo da upravo pre nego pokretna linija dostigne tačku sa najmanjom x -koordinatom kružnice $C(p_i, p_j, p_k)$, postoje tri susedna luka $\alpha, \alpha', \alpha''$ na obalskoj liniji definisana centrima p_i, p_j, p_k . Samo u tom slučaju desiće se događaj kruga. Neka pokretna linija još nije dosegla tačku sa najmanjom x -koordinatom kružnice, ali je na jako malom rastojanju od te tačke. Kako kružnica $C(p_i, p_j, p_k)$, ne sadrži nijedan drugi centar u svojoj unutrašnjosti, to u razmatranom mometu postoji kružnica kroz tačke p_i i p_j , koja dodiruje pokretnu liniju i ne sadrži nijedan centar u svojoj unutrašnjosti. Iz ovog zaključujemo da postoje susedni lukovi na obalskoj liniji definisani centrima p_i i p_j . Na sličan način dokazujemo postojanje susednih lukova i za centre p_j i p_k . Dva puta spominjemo luk definisan centrom p_j . Jasno je da to nisu dva luka veće samo jedan, pa iz ovoga vidimo da postoje tri susedna luka definisana centrim p_i, p_j i p_k . Zbog toga imamo odogovarajući događaj kruga u Q i novo teme Voronoj dijagrama je detektovano. \square

Opišimo sada algoritam detaljnije. Primitimo dodatno da nakon što su svi događaji obrađeni i niz događaja Q ostane prazan, obalska linija i dalje postoji. Presečne tačke su i dalje prisutne i odgovaraju polupravima, ivicama Voronoj dijagrama. Dvostruko povezana lista ivica ne može da predstavi ovakve ivice dijagrama. Zato ograničavamo oblast iscrtavanja Voronoj dijagrama, dodajući granični okvir u scenu. Taj okvir je pravougaonik, koji se bira tako da se svi centri i sve ivice dijagrama nalaze u njemu i na taj način ograničavamo poluivice, odnosno one ivice koje su zapravo poluprave. Na ovaj način se stiče utisak da se dijagram iscrtavao na nekoj ograničenoj površi, kao što je na primer list papira, ili prozor aplikacije. Kompletan algoritam izgleda ovako:

Algoritam Voronojdijagram (P)

Ulaz. Skup $P := \{p_1, p_2, \dots, p_n\}$ centara u ravni.

Izlaz. Voronoj dijagram $\text{Vor}(P)$ unutar pravougaonog rama, dat u vidu dvostruko povezane liste ivica \mathcal{D} .

1. Inicijalizovati niz događaja Q svim događajima centra. *Status* struktura T je inicijalno prazna kao i dvostruko povezana lista ivica \mathcal{D} .
2. **while** Q nije prazan
3. **do** izbaciti element sa najvećom y -koordinatom **if** Q
4. **if** događaj 'događaj centra', centra p_i
5. **then** **ObradiDogađajCentra**(p_i)

6. **else ObradiDogađajKrug** (γ) . γ je list u \mathcal{T} - luk koji nestaje
7. Unutrašnji čvorovi koji još postoje u \mathcal{T} odgovaraju polupravim ivicama Voronj dijagrama. Uvesti pravougaoni ram u scenu, takav da sadrži sve čvorove Voronj dijagrama, a poluprave ivice se iscrtavaju do granica rama.
8. Dodati poluivice u dvostruko povezanu listu ivica .

Procedure za obradu događaja:

ObradiDogađajCentra(p_i)

1. Ako je \mathcal{T} prazan, ubaciti p_i u njega. Inače nastaviti sa koracima 2 – 5.
2. Tražiti luk α u \mathcal{T} vertikalno iznad p_i . Ako list koji predstavlja α ima pokazivač na događaj kruga u Q , onda je ovo lažni alarm uzbune i mora biti izbrisan iz Q .
3. Zameniti list u \mathcal{T} koji predstavlja α , poddrvetom koje sadrži tri lista. Srednji list čuva novi centar p_i , a druga dva lista čuvaju centar p_j , koji je bio originalno čuvan sa lukom α . Parovi (p_i, p_j) i (p_j, p_i) predstavljaju nove presečne tačke u dva nova unutrašnja čvora. Izvesti rebalansiranje stabla \mathcal{T} , ako je potrebno.
4. Kreirati nove slogove za poluivice u strukturi Voronj dijagrama za ivice $\mathcal{V}(p_i)$ i $\mathcal{V}(p_j)$, koje će biti iscrtane sa dve nove presečne tačke.
5. Proveriti trojke susednih lukova kada je novi luk za p_i levi luk, da bi se utvrdilo da li presečne tačke konvergiraju. Ako konvergiraju dodati novi događaj centra u Q i novi pokazivač između čvora u \mathcal{T} i čvora u Q . Isto je potrebno uraditi i za trojku u kojoj je novi luk desni luk.

ObradiDogađajKrug(γ)

1. Obrisati list γ , koji predstavlja luk α koji nestaje iz \mathcal{T} . Ažurirati podatke koji predstavljaju presečne tačke u unutrašnjim čvorovima. Obaviti rebalansiranje stabla \mathcal{T} , ako je potrebno. Izbrisati sve događaje kruga iz Q vezane za α . Oni mogu biti pronađeni koristeći pokazivače prethodnika i sledbenika lista γ u \mathcal{T} . (Događaj kuga gde je α srednji luk je upravo obrađen i već je izbrisan iz Q .)
2. Dodati centar kruga, koji izvodi događaj, kao čvor slog u dvostruko povezanu listu ivica \mathcal{D} . Kreirati, takođe dve poluivice, koje odgovaraju novoj presečnoj tački obalske linije. Potrebno je postaviti i pokazivač, među njima. Dodati tri nova sloga u slogove poluivica, koje završavaju u čvoru.
3. Proveriti novu trojku susednih lukova, koja za srednji luk α ima nekadašnjeg levog suseda, da li presečne tačke konvergiraju. U tom slučaju ubaciti odgovarajući događaj kruga u Q i postavi pokazivač između novog događaja kruga u Q i odgovarajućeg lista u \mathcal{T} . Potrebno je isto uraditi i za trojku u kojoj je bivši desni sused sada desni luk.

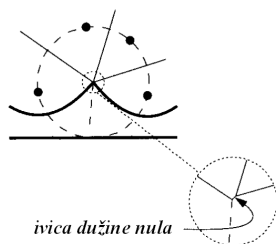
Lema 7 *Vremenska složenost algoritma je $\mathcal{O}(n \log n)$, a prostorna $\mathcal{O}(n)$.*

Dokaz: Primitivne operacije na drvetu \mathcal{T} i nizu događaja Q , kao što su ubacivanje i brisanje elemenata imaju vremensku složenost $\mathcal{O}(n \log n)$. Za primitivne operacije na dvostruko povezanoj listi ivica potrebno je konstantno vreme. Da bismo obradili događaj, obavljamo konstantan broj takvih primitivnih operacija, pa je vremenska složenost događaja $\mathcal{O}(\log n)$. Da bismo prebrojali događaje kruga, napomenimo da svaki takav događaj definiše teme dijagrama. Lažni alarmi uzbune se brišu uz Q , pre procesiranja. Oni se kreiraju i brišu za vreme procesiranja drugih 'pravih' događaja i vreme koje se potroši na njihovu obradu je već sumiramo kroz obradu 'pravih' događaja. Kako je broj događaja kruga koje algoritam obradi $2n - 5$, to je tvrdnja leme dokazano. \square

Na kraju, nekoliko napomena o graničnim slučajevima.

Algoritam događaje obrađuje ka dole, tako da postoji degenerisani slučaj kada dva ili više događaja leže na horizontalnoj liniji. To se može na primer desiti kada dva centra imaju istu y -koordinatu. Ovi događaju mogu biti obrađeni u bilo kom redosledu ako su njihove x -koordinate različite. Međutim ako se ovako nešto desi na samom početku algoritma, tj. ako drugi događaj centra ima istu y -koordinatu kao i prvi potreban je specijalni kod, zato što još ne postoji luk iznad drugog centra.

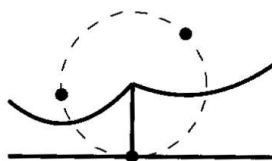
Pretpostavimo da postoje istovremeni događaji. Na primer nekoliko istovremenih događaja kruga, gde postoje četiri ili više centra na kružnici. Centar ovog kruga je čvor Voronoi dijagrama. Stepen ovog čvora je najmanje četiri. Trebalo bi da pišemo specijalan kod da bismo obradili ove slučajeve, ali nema potrebe za tim. Šta će se desiti ako pustimo algoritmu da obradi ovaj slučaj na uobičajen način? Umesto da proizvede čvor stepena četiri on će proizvesti dva čvora stepena tri na istoj lokaciji, sa ivicom dužine nula između njih (slika 2.14). Ako je potrebno, ova ivica može da se izbaciti u postprocesirajućem koraku.



Slika 2.14: Granični slučaj, ivica dužine nula

Sledeći degenerativni slučaj može da se desi pri obradi događaja. Ovo se dešava kada je centar p_i lociran upravo ispod presečne tačke dvaju lukova i to na obalskoj liniji (slika 2.15). U ovom slučaju algoritam deli jedan od lukova i ubacuje luk centra p_i između ta dva dela. Međutim jedan od dva nova dela je

veličine nula i on je sada srednji luk trojke koja definiše događaj kruga. Najniža tačka kružnice se poklapa sa p_i . Algoritam ubacuje ovaj događaj kruga u niz događaja Q . Kada se ovaj događaj kruga obradi, čvor Voronoj dijagrama je ispravno kreiran a luk veličine nula se briše. Drugi degenerativni slučaj se dešava kada su tri susedna luka obalske linije definisana sa tri kolinearna centra. Tada ovi centri ne definišu ni krugove ni događaje kruga.



Slika 2.15: Granični slučaj, centar lociran upravo ispod presečne tačke dvaju lukova na obalskoj liniji

Na kraju možemo da zaključimo da ovaj algoritam ispravno obrađuje degenerisane slučajeve i da i u degenerisanim slučajevima važi lema 7, pa zato važi sledeća teorema.

Teorema 4 *Voronoj dijagram skupa od n centara u ravni može biti konstruisan algoritmom pokretne linije u vremenu $\mathcal{O}(n \log n)$, sa iskorišćenošću memorije $\mathcal{O}(n)$.*

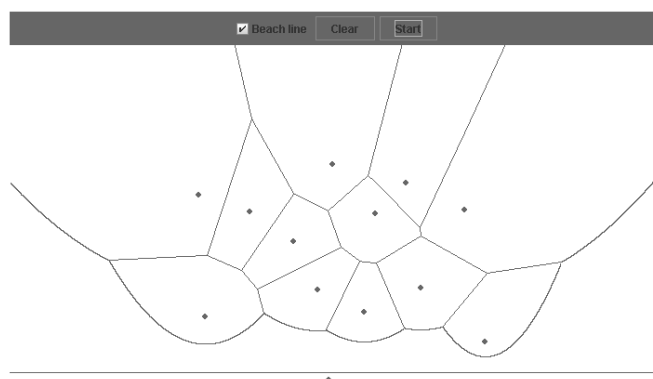
2.2.4 Implementacija algoritma pokretne linije

Prilog ovom radu je Voronoj dijagram implementiran algoritmom pokretne linije kao aplet u Javi 1.4.

Slika 2.16 ilustruje animaciju implementiranog algoritma. Najpre se očekuje unos centara. Potom su na raspolagnju dva dugmeta, jedno kojim se pokreće iscrtavanje dijagrama i drugo kojim se započinje nova iteracija unosa centara. U meniju se, takođe, nalazi opcija prikazivanja *obalske linije*.

U implemetiranoj strukturi nalazi se fajl *applet.html*, kojim se pokreće kompletan projekat. Struktura projekta se zaniva na jednom direktorijumu u kom se nalaze Java klase i navedenom html fajlu izvan njega.

Struktura podataka je sledeća. Tačka se pamti kroz svoje dve koordinate x i y , kao i događaj *event*. Događaj kruga pored koordinata tačke u kojoj će se desiti, pamti i poluprečnik kruga u momentu dodavanja treće parabole, te parabolu za koju je vezan. Svi očekivani događaji su vezani u dvostruko povezanu listu događaja. Parabola je određena centrom i trenutnim položajem direktrise. Parabole su povezane u stablo parabola. Osnovna Voronoj kalasa zasnovana je na dva vektora, vektor centara i vektor linija koje se iscrtavaju. Odavde vidimo da je prostorna složenost algoritma n^2 .



Slika 2.16: Implementacija inkrementalnog algoritma Deloneove triangulacije

Na osnovu centra i pozicije direktrise određuje se jednačina parabole u formi:

$$a_0x^2 + a_1x + a_2 = y$$

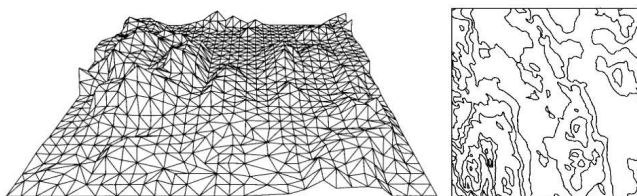
Ubacivanjem svake nove parabole drvo prabola i lista događaja se menjaju. Samo prikazivanje scene se odvija na sledeći način. Prikazju se centri, te linije iz vektora, a potom se crtaju nedovršene linije. Naime za svaku parabolu, poznata je tačka od koje se crta nedovršena linija sve dok se pomeranjem obalske linije ne prikaže cela linija, kada se ona doda u vektor celih linija.

Kao što je već pri detaljnom opisu ovog algoritma obrazloženo vremenska složenost ovog algoritma je $n \log(n)$.

Deloneova triangulacija

U prethodnom poglavlju, razmatrali smo primer modelovanja površine zemlje, zanemarujući treću dimenziju. Naime, kao primer za Voronoi dijagram navodili smo zemlju kao što je Holandija. Međutim, ukoliko je reč u zemlji kao što je Švajcarska, ne možemo da zanemarimo treću dimenziju.

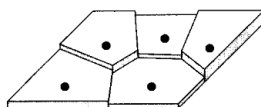
Modelujmo, zato, deo površine zemlje kao *teren*. Teren je 2-dimezionalna površina u 3-dimezionalnom prostoru sa sledećom karakteristikom: svaka vertikalna linija preseca teren najviše u jednoj tački. Drugim rečima, to je graf funkcije $f : A \subset \mathbb{R}^2 \rightarrow \mathbb{R}$, koja dodeljuje visinu $f(p)$ svakoj tački iz domena A . Teren može biti vizualizovan bilo perspektivnim crtanjem ili izohipsama, jednake visine, kao na topografskom modelu (slika 3.1).



Slika 3.1: Predstavljanje terena perspektivno ili izohipsama

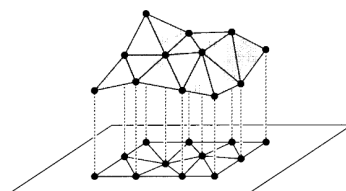
Visinu neke tačke na zemlji znamo jedino ako je izmerimo. To znači, da kada razmatramo neki teren, onda vrednosti funkcije f znamo samo na nekom konačnom skupu $P \subset A$, koji je skup tačaka uzorka. Iz visina tačaka uzorka potrebno je da na neki način aproksimiramo visinu drugih tačaka domena. Najjednostavniji pristup bi svakoj tački $p \in A$ dolelio visinu najbliže tačke uzorka. Ovim bi smo dobili diskretni teren (slika 3.2), koji ne izgleda prirodno.

Pristupimo zato aproksimaciji terena na sledeći način. Najpre odredimo *triangulaciju* skupa P : napravimo podelu ravni na trouglove, čija su temena tačke iz skupa P . Zatim, pridružimo svakom temenu odgovarajuću visinu. Ovim



Slika 3.2: Diskretni teren

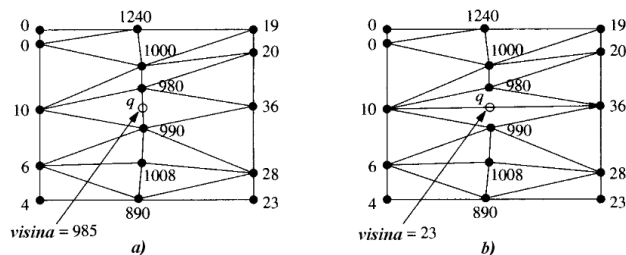
dobijamo trouglove u 3-dimenzionalnom prostoru, što je prikazano na slici 3.3. Ovako dobijeni *poliedralni teren*, je graf neprekidne funkcije, predstavljene linijskim segmentima. Poliedralni teren može se uzeti kao aproksimacija originalnog terena.



Slika 3.3: Proces dobijanja poliedralnog terena skupa tačaka iz uzorka

Ostaje pitanje: Kako da obavimo triangulaciju skupa tačaka iz uzorka? Generalno, to može da se uradi na mnogo načina. Ali koja triangulacija nam za aproksimaciju terena najviše odgovara? Ispravan odgovor na ovo pitanje ne postoji. Naime, mi ne znamo kako izgleda terena. Jedino što znamo jeste visina u tačkama iz uzorka. Kako drugu informaciju nemamo onda bilo koja triangulacija izgleda jednako dobra. Međutim, neke triangulacije izgledaju prirodnije od drugih. Primer sa slike 3.4 pokazuje dve triangulacije istog skupa tačaka. Na osnovu visina koje su zadate za tačke uzorka, zaključujemo da je razmatrani teren, planinski greben. Triangulacije (a) pokazuje upravo takvu situaciju. U triangulaciji (b), međutim, možemo da uočimo "pomerenu" ivicu, koja ocrtava strmu udolinu planinskog grebena. Intuitivno zaključujemo da ova triangulacija nije dobra. Možemo li zato reći da je triangulacija (a) bolja od triangulacije (b)?

Problem kod triangulacije (b) je u tome što je visina tačke q određena dvema tačkama koje su relativno udaljene. Ovo se dešava zato što tačka q leži na sredini ivice dva duga i oštra trougla. Problem je to što ovi trouglovi imaju dva jako oštra ugla, pa za njih kažemo da su "tanki". Čini se da je triangulacija koja ima male uglove loša. Zbog toga ćemo porediti triangulacije, poredići uglove. Kako postoji konačan broj triangulacija datog skupa P , to mora postojati i optimalna triangulacija, sa najvećim minimalnim uglom. To će biti triangulacija koju tražimo.

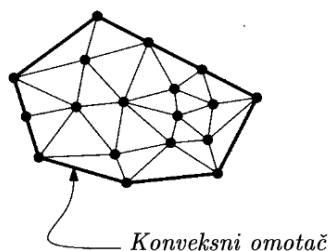


Slika 3.4: Pomeranje jedne ivice može da napravi veliki problem

3.1 Triangulacije planarnog skupa tačaka

Neka je $P := \{p_1, p_2, \dots, p_n\}$ skup tačaka u ravni. Da bismo formalno definisali triangulaciju skupa P , definišimo najpre *maksimalnu planarnu podelu* kao skup ivica S , čija su temena tačke iz P , tako da bilo koja ivica koje nije u S preseca neku od postojećih ivica. Triangulacija skupa P je definisana kao maksimalna planarana podela.

Iz ove definicije vidimo da triangulacija sigurno postoji. Svaki element podele je trougao, osim neograničenih delova. Međutim, šta je sa neograničenim delovima? Segmenti koji vezuju susedne tačke, na granici oblasti u kojoj su tačke iz P su ivice i to u bilo kojoj triangulaciji \mathcal{T} . Unija ovih graničnih ivica triangulacije \mathcal{T} formira konveksni omotač skupa tačaka P . Zbog toga su "neograničene" oblasti uvek ograničene konveksnim omotačem, kao na slici 3.5.

Slika 3.5: Konveksni omotač skupa tačaka iz P

Broj trouglova je isti u svakoj triangulaciji, kao i broj ivica. Tačan broj zavisi naravno, od broja tačaka u P , koje ulaze u konveksni omotač. Broj tačaka na granici konveksnog omotača nije obavezno jednak broju temena konveksnog omotača. Pri tome važi sledeća torema.

Teorema 5 Neka je P skup n tačaka u ravni, koje nisu sve kolinearne. I neka je k broj tačaka iz P , koje leže na konveksnom omotaču skupa P . Tada bilo koja triangulacija skupa P broji $2n - 2 - k$ trouglova i $3n - 3 - k$ ivica.

Dokaz: Na osnovu Ojlerove formule za n -čvorova, n_e -ivica i n_f trougaonih površi povezanog planarnog grafa važi:

$$n - n_e + n_f = 2$$

Koristeći ovo tvrđenje dokazujemo teoremu. \square

Neka je \mathcal{T} triangulacija skupa P od m trouglova. Sortirajmo $3m$ uglova triangulacije \mathcal{T} rastuće. Za rezultujuću sekvencu $\alpha_1, \alpha_2, \dots, \alpha_{3m}$ važi $\alpha_i \leq \alpha_j$ za svako $i < j$. Nazovimo $A(\mathcal{T}) := (\alpha_1, \alpha_2, \dots, \alpha_{3m})$ vektorom uglova triangulacije \mathcal{T} . Neka je \mathcal{T}' druga triangulacija skupa tačaka P i neka je $A(\mathcal{T}') := (\alpha'_1, \alpha'_2, \dots, \alpha'_{3m})$ njen vektor uglova. Kažemo da je vektor uglova triangulacije \mathcal{T} veći od vektora uglova triangulacije \mathcal{T}' ako je $A(\mathcal{T})$ leksikografski veći od $A(\mathcal{T}')$, odnosno ako postoji indeks i , gde je $1 \leq i \leq 3m$, takav da važi

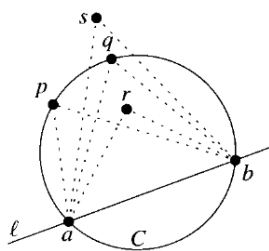
$$\alpha_j = \alpha'_j \text{ za svako } j < i, \quad \alpha_i > \alpha'_i$$

Obeležimo ovo kao $A(\mathcal{T}) > A(\mathcal{T}')$. Za triangulaciju \mathcal{T} kažemo da je *ugao-optimalna* ako važi $A(\mathcal{T}) \geq A(\mathcal{T}')$ za sve druge triangulacije \mathcal{T}' skupa P . Ugao-optimalne triangulacije su za nas interesantne, jer su dobre za izgradnju poliedralnog terena nad skupom tačaka uzorka.

Razmotrimo sada optimalne triangulacije. Podsetimo se, zato, Talesove teoreme, ali najpre obeležimo manji konveksni ugao definisan trima tačkama p, q, r sa $\angle pqr$.

Teorema 6 Neka je C kružnica, a l linija koja preseca C u dvema tačkama a i b i neka su p, q, r i s tačke koje leže sa iste strane prave l . Pretpostavimo da p i q leže na C , r unutar C , a s izvan C . Tada važi (slika 3.6):

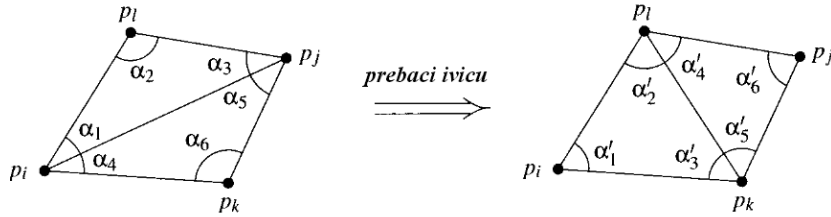
$$\angle arb > \angle apb = \angle aqb > \angle asb$$



Slika 3.6: Talesova teorema

Posmatrajmo ivicu $e = \overline{p_i p_j}$ triangulacije \mathcal{T} skupa P . Ako e nije ivica konveksnog omotača, onda je ona ivica dva trougla $p_i p_j p_k$ i $p_i p_j p_l$. Ako ova dva

trougla formiraju konveksan četvorougao, možemo da dobijemo i novu triangulaciju uklanjajući ivicu $\overline{p_i p_j}$ iz \mathcal{T} i ubacujući ivicu $\overline{p_k p_l}$ umesto nje. Ovu operaciju zovemo *prebacivanje ivice* (eng. *edge flip*). Jedina razlika je u vektoru uglova triangulacija \mathcal{T} i \mathcal{T}' gde su šest uglova $\alpha_1, \dots, \alpha_6$ iz $A(\mathcal{T})$, zamenjeni sa $\alpha'_1, \dots, \alpha'_6$ u $A(\mathcal{T}')$. Slika 3.7 ilustruje prebacivanje ivice. Ivicu $e = \overline{p_i p_j}$, sa ove slike, zovemo *ilegalnom ivicom*, ako važi $\min_{1 \leq i \leq 6} \alpha_i < \min_{1 \leq i \leq 6} \alpha'_i$.



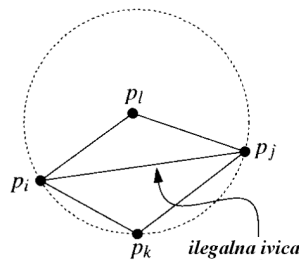
Slika 3.7: Prebacivanje ivice

Drugim rečima ivica je ilegalna ako možemo povećati najmanji ugao prebacujući ivicu. Sledeće tvrđenje direktno sledi iz definicije ilegalne ivice.

Tvrđenje 1 Neka je \mathcal{T} triangulacija sa ilegalnom ivicom e . Neka je \mathcal{T}' triangulacija dobijena iz \mathcal{T} prebacivanjem ivice e . Tada važi $A(\mathcal{T}') > A(\mathcal{T})$.

Izpostavlja se da nije neophodno računati uglove $\alpha_1, \dots, \alpha_6, \alpha'_1, \dots, \alpha'_6$ da bi se proverilo da li je ivica legalna. Umesto toga može se koristiti jednostavan kriterijum izložen sledećom lemom. Tačnost ovog kriterijuma sledi iz Talesove teoreme.

Lema 8 Neka je $\overline{p_i p_j}$ zajednička ivica trouglova $p_i p_j p_k$ i $p_i p_j p_l$ i neka je C kružnica kroz tačke p_i, p_j i p_k . Ivica $\overline{p_i p_j}$ je ilegalna akko tačka p_l leži u unutrašnjosti kruga C , kao na slici 3.8. Takođe važi i da ukoliko tačke p_i, p_j, p_k i p_l formiraju konveksni četvorougao, onda je jedna od ivica $\overline{p_i p_j}$ i $\overline{p_k p_l}$ ilegalna ivica.



Slika 3.8: Ilegalna ivica konveksnog četvorougla

Definišimo *legalnu triangulaciju* kao triangulaciju, koja ne sadrži nijednu ilegalnu ivicu. Iz prethodno razmatranog, zaključujemo da su jedino ugao-otimalne triangulacije legalne. Računanje legalne triangulacije je jednostavno, kad nam je data početna triangulacija, prebacivanjem ilegalnih ivica sve dok sve ivice ne postanu legalne.

Algoritam LegalTriangulation (P)

Ulaz. Neka triangulacija \mathcal{T} skupa P .

Izlaz. Legalna triangulacija skupa P .

1. **while** \mathcal{T} sadrži ilegalnu ivicu $\overline{p_i p_j}$
2. **do** prebaci ivicu $\overline{p_i p_j}$
3. Neka su $p_i p_j p_k$ i $p_i p_j p_l$ trouglovi sa ivicom $\overline{p_i p_j}$
4. Izbaci ivicu $\overline{p_i p_j}$ iz \mathcal{T} , a ubaci $\overline{p_k p_l}$
5. **return** \mathcal{T}

Kako se ovaj algoritam zaustavlja? Na osnovu tvrdjenja 1 vektor ugla triangulacije \mathcal{T} se povećava svakom iteracijom petlje. S obzirom da postoji samo konačan broj triangulacija skupa P , ovim je dokazana konačnost algoritma. Kad se algoritam zaustavi, dobijamo kao izlaznu vrednost legalnu triangulaciju. Iako se algoritam sigurno zaustavlja, on je jako spor.

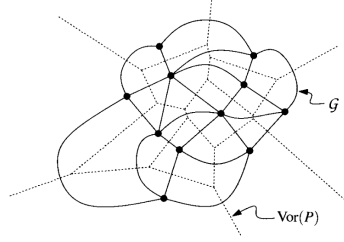
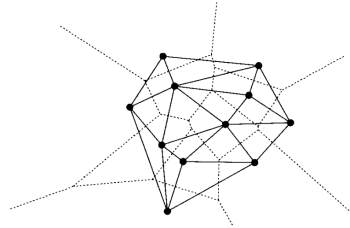
3.2 Deloneova triangulacija

Neka je P skup od n tačaka ili *centara* u ravni. Podsetimo se da je Voronoj dijagram skupa P podela ravni na n regiona, svaki za po jedan centar iz P , takvih da region centra $p \in P$ sadrži sve tačke u ravni kojima je p najbliži centar. Voronoj dijagram skupa P obeležavali smo sa $Vor(P)$. Region centra p nazivali smo Voronoj ćelijom i obeležavali sa $\mathcal{V}(p)$.

U ovom delu proučavaćemo dualni graf Voronoj dijagrama. Ovaj graf \mathcal{G} ima čvorove u svakoj ćelije Voronoj dijagrama, odnosno svakom centru Voroni dijagrama. Lukovi ovog grafa spajaju čvorove koji odgovaraju onim ćelijama koje dele zajedničku ivicu u Voronoj dijagramu, kao na slici 3.9. Postoji 1-na-1 preslikavanje između ograničenih oblasti u \mathcal{G} i temena u $Vor(P)$.

Pogledajmo sad graf \mathcal{G} izgrađen pravim linijama, gde je čvor koji odgovara Voronoj ćeliji $\mathcal{V}(p)$ p , a luk koji spaja čvorove $\mathcal{V}(p)$ i $\mathcal{V}(q)$ linijski segment \overline{pq} . Primer je na slici 3.10. Ovakav graf nazivamo *Deloneov graf* skupa P i obeležavamo sa $\mathcal{DG}(P)$.

Graf je dobio ime po matematičaru Borisu Nikolaevichu Deloneu. Kako je njegov rad objavljen na francuskom jeziku, ime je prevedeno na francuski i kao takvo i danas se koristi (Delaunay). Ispostavilo se da ovaj graf ima mnogo iznenađujućih osobina. Prva je da je on uvek planarni graf, odnosno ne postoje dve ivice koje se međusobno seku.

Slika 3.9: Dualni graf za $Vor(P)$ Slika 3.10: Deloneov graf $\mathcal{DG}(P)$

Teorema 7 *Deloneov graf tačka u ravi je planarni graf.*

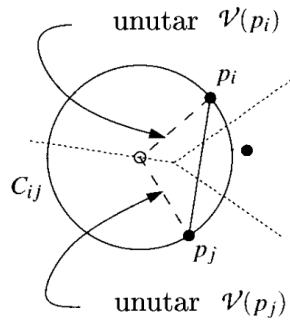
Dokaz: Da bismo ovo dokazali potrebna nam je osobina ivica Voronoi dijagrama iz teoreme 3. Ponovimo zato tu osobinu. Ovaj put ćemo je reći treminima Deloneovog grafa.

Ivica $\overline{p_i p_j}$ je ivica Deloneovog grafa $\mathcal{DG}(P)$ ako i samo ako postoji kružnica C_{ij} kojoj pripadaju p_i i p_j i nijedan drugi centar iz P ne leži unutar kržnice, a centar ove kružnice je na ivici Voronoi dijagrama definisanoj sa $\mathcal{V}(p_i)$ i $\mathcal{V}(p_j)$ (slika 3.11).

Definišimo t_{ij} trougao, čija su temena p_i, p_j i centar kružnice C_{ij} . Primetimo da se ivica trougla koja povezuje p_i i centar C_{ij} nalazi unutar $\mathcal{V}(p_i)$, a analogno važi i za p_j (slika 3.11). Neka je $\overline{p_k p_l}$ druga ivica u $\mathcal{DG}(P)$. Definišimo za nju kružnicu C_{kl} i trougao t_{kl} slično kao i za $\overline{p_i p_j}$.

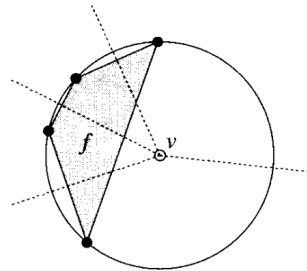
Pretpostavimo suprotno, da se ivice $\overline{p_i p_j}$ i $\overline{p_k p_l}$ seku. I p_k i p_l moraju da leže izvan C_{ij} , a to znači i izvan t_{ij} . S obzirom na pretpostavku, to znači da $\overline{p_k p_l}$ mora da seče neku od ivica trougla t_{ij} , čije teme je centar kružnice C_{ij} . Slično tome i $\overline{p_i p_j}$ mora da seče neku od ivica trougla t_{kl} , čije teme je centar kružnice C_{kl} . Iz ovoga dobijamo da jedna od ivica trougla t_{ij} , čije teme je centar kružnice C_{ij} mora da seče jednu od ivica trougla t_{kl} , čije teme je centar kružnice C_{kl} . Ali, ovo je kontradiktorno činjenici da se ove ivice nalaze u disjunktinim Voronoi ćelijama. \square

Deloneov graf skupa P nastao je iz dualnog grafa Voronoi dijagramu. Ranije smo već primetili da svakoj ograničenoj oblasti u dijagramu odgovara jedan čvor u $Vor(P)$. Granične ivice ograničenih oblasti odgovaraju Voronoi ivi-



Slika 3.11: Ilica Deloneovog grafa

cama, odgovarajućih Voronj ćelija. Ako je čvor v u $Vor(P)$ čvor Voronj ćelija sa centrima p_1, p_2, \dots, p_k , onda odgovarajuća oblast f u $\mathcal{DG}(P)$ ima temena p_1, p_2, \dots, p_k . Na osnovu teoreme 3, tačke p_1, p_2, \dots, p_k leže na kružnici oko v , gradeći konveksan k -tougao, kao na slici 3.12.



Slika 3.12: Čvor Voronj ćelije

Definicija 5 Za skup tačaka kažemo da je u opštem položaju ako ne postoji nijedna četvorka tačaka na kružnici.

Ako su tačke u P birane slučajnim izborom, verovatnoća da četiri tačke leže na kružnici je veoma mala. Neka je P u opštem položaju. Onda svi čvorovi Voronj dijagrama imaju stepen tri iz čega sledi da su sve ograničene oblasti u $\mathcal{DG}(P)$ trouglovi. Ovo je razlog zašto se $\mathcal{DG}(P)$ često naziva *Deloneovom triangulacijom* skupa P . Mi ćemo biti malo precizniji, pa ćemo $\mathcal{DG}(P)$ zvati *Deloneov grafom* skup P . A *Deloneovu triangulaciju* definisaćemo kao triangulaciju koja se dobija dodavanjem ivica u Deloneov graf. Kako su sve ograničene oblasti u $\mathcal{DG}(P)$ konveksne, dobijanje takve triangulacije je jednostavno. Iz ovoga dobijamo i da je Deloneova triangulacija [1] jedinstvena ako i samo ako je $\mathcal{DG}(P)$ triangulacija, što znači da je P u opštem položaju.

Ponovimo sada teoremu 3 o Voronj dijagramu u terminima Deloneove triangulacije.

Teorema 8 *Neka je P skup tačaka u ravni.*

- *Tri tačke $p_i, p_j, p_k \in P$ su čvorovi iste oblasti u Deloneovom grafu skupa P ako i samo ako kružnica kroz tačke p_i, p_j i p_k ne sadrži druge tačke iz P unutar kruga.*
- *Dve tačke $p_i, p_j \in P$ grade ivicu Deloneovog grafa skupa P ako i samo ako postoji kružnica koja sadrži p_i i p_j ali ne sadrži nijednu drugu tačku iz P unutar kruga.*

Iz ove teoreme dobijamo da važi i sledeća teorama.

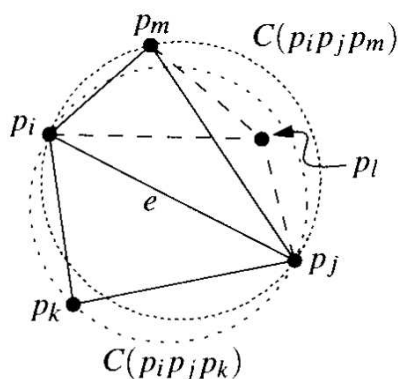
Teorema 9 *Neka je P skup tačaka u ravni, a T triangulacija skupa P . T je Deloneova triangulacija skupa P ako i samo ako opisani krug bilo kog trougla u T ne sadrži tačke iz P unutar kruga.*

Već smo rekli da je triangulacija dobra za interpolaciju reljefa ako je njen vektor uglova što je moguće veći. Posmatrajmo sada vektor uglova Deloneove triangulacije. Počnimo od legalne triangulacije.

Teorema 10 *Neka je P skup tačaka u ravni. Triangulacija T skupa P je legalna ako i samo ako je T Deloneova triangulacija skupa P .*

Dokaz: Iz definicije sledi da je Deloneova triangulacija legalna.

Dokažimo da je legalna triangulacija Deloneova kontradikcijom. Pretpostavimo zato da je T legalna triangulacija skupa P koja nije Deloneova. Na osnovu teoreme 8 sledi da postoji trougao $p_i p_j p_k$ čiji opisani krug $C(p_i, p_j, p_k)$ sadrži tačku $p_l \in P$ u svojoj unutrašnjosti (slika 3.13). Neka je $e := \overline{p_i p_j}$, ivica trougla $p_i p_j p_l$, takva da trougao $p_i p_j p_l$ ne preseca $p_i p_j p_k$. Od svih parova $(p_i p_j p_k, p_l)$ u



Slika 3.13: Triangulacija je legalna akko je Deloneva triangulacija

T , odaberimo onaj sa maksimalnim uglom $\angle p_i p_l p_j$. Posmatrajmo sada trougao

$p_i p_j p_m$, susedni trouglu $p_i p_j p_k$, a koji dele zajedničku ivicu e . Kako je \mathcal{T} legalna triangulacija to je i e legalna ivica. Na osnovu leme 8 znači da tačka p_m ne leži u unutrašnjosti kružnice $C(p_i p_j p_k)$. Kružnica $C(p_i p_j p_m)$ sadrži deo kružnice $C(p_i p_j p_k)$, koji određuje duž e . Iz ovoga sledi da $p_l \in C(p_i p_j p_m)$. Pretpostavimo da je $\overline{p_j p_m}$ ivica trougla $p_i p_j p_m$, takva da $p_j p_m p_l$ ne preseca $p_i p_j p_m$. Sada je $\angle p_j p_l p_m > \angle p_i p_l p_j$ što je suprotno definicije para $(p_i p_j p_k, p_l)$. \square

S obzirom da bilo koja ugao-optimalna triangulacija mora biti legalna, iz teoreme 10 sledi da je ugao-optimalna triangulacija skupa P Deloneova triangulacija skupa P . Kada je P u opštem položaju postoji samo jedna legalna triangulacija, koja je i jedina ugao-optimalna triangulacija, a možemo reći i jedina Deloneova triangulacija koja se podudara sa Delonevim grafom. Kada P nije u opštem položaju, onda je bilo koja triangulacija Deloneovog grafa legalna. Ali ne moraju sve Deloneove triangulacije biti ugao-optimalne. Međutim njihovi vektori uglova se ne razlikuju puno. Koristeći Telsovu teoremu može se dokazati da je minimalan ugao bilo koje triangulacije skupa tačaka, koje leže na istoj kružnici uvek isti. To znači da minimalan ugao ne zavisi od triangulacije. Odnosno, bilo koja triangulacija, koja Deloneov graf prevodi u Deloneovu triangulaciju ima isti minimalni ugao. Pa tako važi i sledeća teorema.

Teorema 11 *Neka je P skup tačaka u ravni. Bilo koja ugao-optimalna triangulacija skupa P je Deloneova triangulacija skupa P . Šta više, bilo koja Deloneova triangulacija skupa P ima veći minimalni ugao od bilo koje druge triangulacije skupa P .*

3.3 Konstrukcija Deloneove triangulacije

Već znamo kako se računa Voronj dijagram i da iz njega možemo veoma lako da dobijemo Deloneovu triangulaciju. U ovom odeljku, iznećemo drugačiji pristup konstrukciji Deloneove triangulacije. Predstavićemo dva algoritma: algoritam zasnovan na razlaganju i inkrementalni algoritam.

3.3.1 Algoritam zasnovan na razlaganju

Ovaj algoritam, brzine izvršavanja $\mathcal{O}(n \log n)$, predstavili su Guibas i Stolfi (eng. Guibas & Stolfi) 1985. godine [5]. Već znamo da se strategija *razlaganja* zasniva na podeli problema na manje podprobleme, koji su približno jednake veličine, a rešenje originalnog problem se dalje dobija *povezivanjem* rešenja podproblema.

Struktura algoritma data je sledećim pseudokodom:

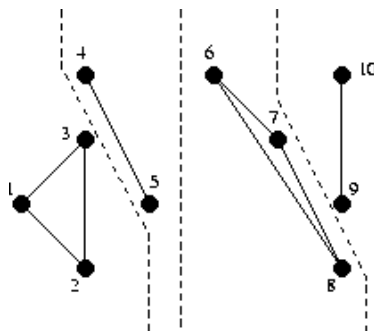
Algoritam DeloneTriangulacija (P)

Ulaz. Za $n > 3$ skup $P := \{p_1, p_2, \dots, p_n\}$ tačaka u ravni.

Izlaz. Deloneova triangulacija skupa P .

1. Sortirati tačke rastuće u odnosu na x -koordinatu, a ako imaju istu x -koordinatu redosled im se određuje na osnovu y -koordinate.
2. Neka je t ceo deo od $n/2$. Podeliti P na $P_L := \{p_1, p_2, \dots, p_t\}$ i $P_R := \{p_{t+1}, p_{t+2}, \dots, p_n\}$.
3. Konstruisati Deloneovu triangulaciju skupa (P_L) i Deloneovu triangulaciju skupa P_R rekurzivno.
4. Povezati Deloneovu triangulaciju skupa P_L i Deloneovu triangulaciju skupa P_R u Deloneovu triangulaciju skupa P algoritmom **PovežiDelone**.
5. return Deloneova triangulacija skupa P

Nakon što su tačke sortirane, ulazjenje u rekurziju deljenjem skupova tačaka, se zaustavlja kada skupovi ne sadrže više od tri tačke (slika 3.14). To znači da najmanji skupovi mogu biti ili linijski segmenti između dve tačke ili trouglovi određeni trima tačkama.



Slika 3.14: Podela tačaka rekurzivnim algoritmom

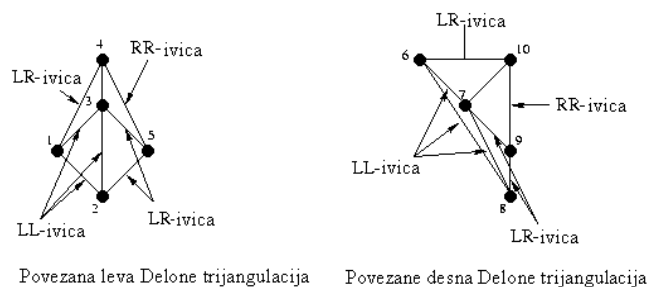
Pre nego izložimo detaljnije povezivanje Deloneove triangulacije, uvedimo novu terminologiju. LL -ivicama nazivamo ivice iz već postojeće leve triangulacije. Obe krajnje tačke (temena) tih ivica su u levom skupu tačaka. RR -ivicama nazivamo ivice iz već postojeće desne triangulacije. Obe krajnje tačke tih ivica su u desnom skupu tačaka. Prilikom *povezivanja* leve i desne triangulacije nastaju nove ivice, koje zovemo LR -ivice. Njihova jedna krajnja tačka pripada levom, a druga desnom skupu tačaka. Videćemo da prilikom procesa povezivanja LL i RR ivice mogu biti izbrisane, ali nove ne mogu biti kreirane.

Prvo *povezivanje* je veoma jednostavno, bez brisanja ivica, kao na slici 3.15.

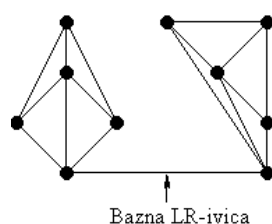
Ostalo nam je još jedno povezivanje. Sve ivice sa leve strane su LL -ivice, a sve ivice sa desne strane su RR -ivice.

Prvi korak pri povezivanju dve polovine je ubacivanje bazne LR -ivice (slika 3.16). To je najniža LR -ivica, koja ne preseca nijednu LL ni RR ivicu.

Sada je potrebno dodati sledeću LR ivicu, iznad bazne LR -ivice. Jedna krajnja tačka te ivice biće desna ili leva krajnja tačka bazne LR -ivice, a druga



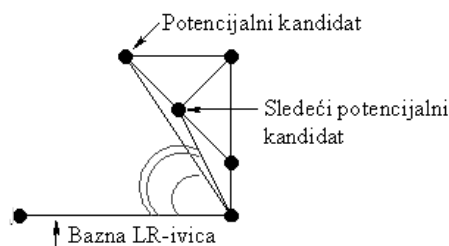
Slika 3.15: Jednostavno povezivanje malog skupa tačaka



Slika 3.16: Bazna ivica između leve i desne triangulacije

će biti iz levog ili desnog podskupa tačaka. Ovaj izbor ćemo da suzimo na jednog kandidata iz levog podskupa i jednog kandidata iz desnog podskupa.

Počnimo sa desne strane. Prvi potencijalni kandidat je tačka povezana sa desnom krajnjom tačkom bazne LR -ivice preko RR -ivice, koja određuje najmanji ugao u pravcu obrnutom od kazaljke na satu, počevši od bazne LR -ivice. Slično, sledećeg kandidata biramo po sledećem najmanjem uglu, kao što je prikazano na slici 3.17.

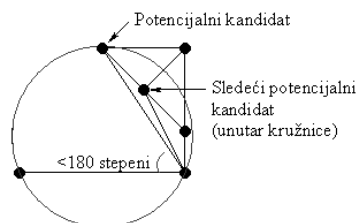


Slika 3.17: Izbor potencijalnog kandidata i sledećih potencijalnih kandidata

Za potencijalne kandidate se proveravaju sledeća dva kriterijuma:

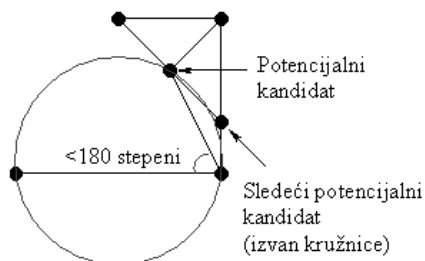
1. Ugao u pravcu obrnutom od kazaljke na satu, počevši od bazne LR -ivice mora biti manji od 180 stepeni.

2. Kružnica kroz krajeve tačke bazne LR -ivice i potencijanog kandidata, ne sme sadržati sledećeg potencijalnog kandidata unutar (slika 3.18).



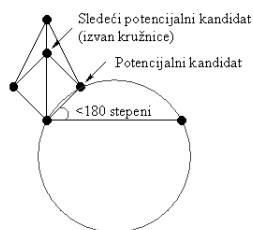
Slika 3.18: Potencijalni kandidat zadovoljava prvi, ali ne i drugi uslov

Ako su oba kriterijuma ispunjena, kandidat postaje konačni desni kandidat. Ako prvi uslov nije ispunjen, kandidat sa desne strane nije izabran. Ako je prvi kriterijum ispunjen, a drugi nije, onda se ivica, koja vezuje potencijanog kandidata i desnu krajnju tačku bazne LR -ivice briše (slika 3.19). Proces se onda nastavlja sa sledećim potencijalnim kandidatom, sve dok se ne izabere desni kandidat ili dok se ne dođe do zaključka da kandidat ne može biti izabran.



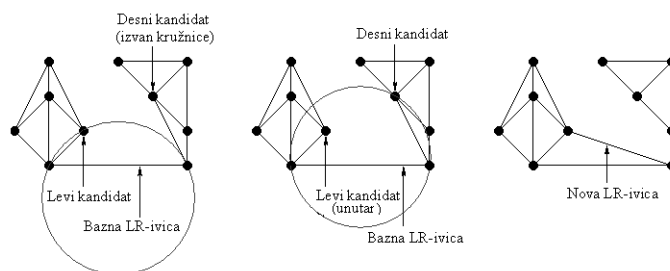
Slika 3.19: Nakon brisanja nepotrebne RR -ivice desni kandidat je pronađen

Proces se potom ponavlja pri pronalaženju levog kandidata (slika 3.20).



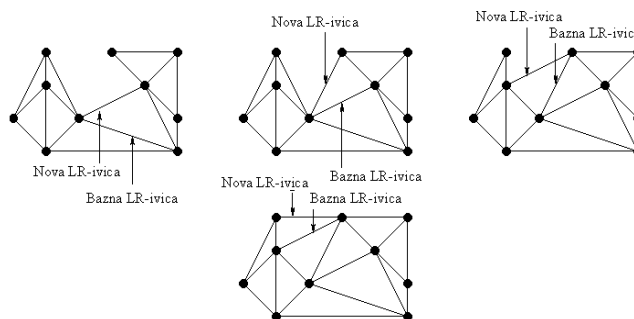
Slika 3.20: Sa leve strane, potencijalni kandidat zadovoljava oba kriterijuma

Kada ni levi ni desni kandidat ne mogu biti nađeni proces povezivanja je završen. Ako je pronađen samo jedan kandidat, njim je određena LR -ivica koju tražimo. U slučaju da su pronađena oba kandidata, nova LR -ivica se određuje sledećim testom: ako se desni kandidat ne nalazi unutar kružnice definisane dvema krajnjim tačkama bazne LR -ivice i levim kandidatom, onda levi kandidat određuje LR -ivicu i obrnuto (slika 3.21). S obzirom da Deloneova triangulacija mora postojati barem jedan od kandidata će ispuniti ovaj zahtev.



Slika 3.21: U primeru, samo levi kandidat ispunjava uslov i određuje novu ivicu

Nakon što je pronađena nova LR -ivica proces se ponavlja sa novom LR -ivicom, kao baznom LR -ivicom. Proces se zato nastavlja, kao što je prikazano slikom 3.22



Slika 3.22: LR -ivice se dodaju i gradi se nova Deloneova triangulacija

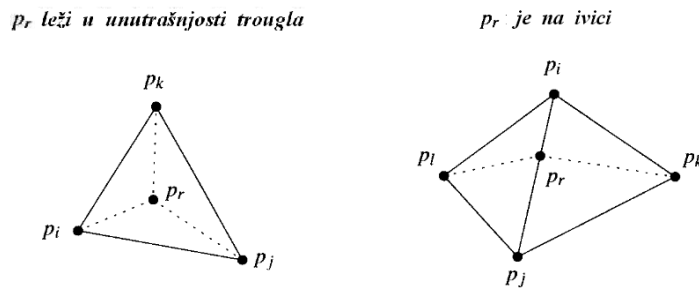
Računanje horizontalne ili vertikalne linije podele je veoma jednostavno, ako su centri sortirani po x i y koordinati. Bilo koji optimalan algoritam, na primer *heap sort* ili *merge sort* izvršavaju se u $\mathcal{O}(n \log n)$ vremenu. Koraci povezivanja sa izvršava za $\mathcal{O}(n)$. Ovim dolazimo do sledeće leme.

Lema 9 *Vremenska složenost algoritma je $\mathcal{O}(n \log n)$.*

3.3.2 Inkrementalni algoritam

Konstrukcija Deloneove triangulacije skupa P inkrementalnim algoritmom [1] započinje formiranjem velikog trougla $p_{-1}p_{-2}p_{-3}$ koji sadrži sve tačke skupa P . Ovo znači da Deloneovu triangulaciju računamo za skup $P \cup \Omega$, gde je $\Omega := \{p_{-1}, p_{-2}, p_{-3}\}$. Naknadno se odbacuju tačke skupa Ω , kao i suvišne ivice. Tačke p_{-1}, p_{-2}, p_{-3} potrebno je odabrati dovoljno daleko, tako da ne promene nijedan trougao Deloneove triangulacije skupa P . Odnosno moramo biti sigurni da ne leže ni u jednom krugu određenom trima tačkama iz P . Detaljno ćemo razmotriti ovaj problem nakon samog algoritma.

Algoritam je slučajni inkrementalni algoritam. Tačke se dodaju slučajnim izborom, a algoritam održava Deloneovu triangulaciju trenutnog skupa tačaka. Prilikom dodavanja nove tačke p_r , najpre pronalazimo trougao trenutne triangulacije, koji sadrži tačku p_r , te potom dodajemo ivice od p_r ka temenima tog trougla. Ako se desi da se p_r nalazi na ivici e , već postojećeg trougla, dodaju se ivice od p_r ka suprotnim temenima trouglova kojima je ivica e zajednička ivica. Slika 3.23 ilustruje oba slučaja.



Slika 3.23: Dva slučaja dodavanja novog temena p_r

Ovim postupkom dobijamo triangulaciju, ali ovo ne mora biti Deloneova triangulacija. Razlog tome je što dodavanje tačke p_r , može da promeni neku od postojećih ivica u ilegalnu ivicu. Da bismo ovo izbegli, pozivamo funkciju `LegalizujIvicu`, za svaku potencijalno ilegalnu ivicu. Ova procedura zamenjuje ilegalne ivice legalnim koristeći proceduru prebacivanja ivice. Najpre izložimo detaljan opis glavnog algoritma.

Algoritam DeloneTriangulacija (P)

Ulaz. Skup P od n tačaka u ravni.

Izlaz. Deloneova triangulacija skupa P .

1. Neka su p_{-1}, p_{-2} i p_{-3} takve tri tačke da su sve tačke iz P sadržane u trouglu $p_{-1}p_{-2}p_{-3}$.
2. Početna triangulacija \mathcal{T} sastoji se od jednog trougla $p_{-1}p_{-2}p_{-3}$.
3. **for** $r \leftarrow 1$ to n

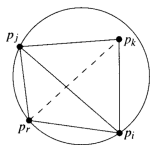
4. **do** (* Ubaci p_r u \mathcal{T} : *)
5. Naći trougao $p_i p_j p_k \in \mathcal{T}$, koji sadrži p_r .
6. **if** p_r leži u unutrašnjosti trougla $p_i p_j p_k$
7. **then** Dodati ivice iz p_r ka temenima trougla $p_i p_j p_k$.
8. LegalizujIvicu($p_r, \overline{p_i p_j}, \mathcal{T}$)
9. LegalizujIvicu($p_r, \overline{p_j p_k}, \mathcal{T}$)
10. LegalizujIvicu($p_r, \overline{p_k p_i}, \mathcal{T}$)
11. **else** (* p_r leži na ivici trougla $p_i p_k p_j$, recimo na ivici $\overline{p_i p_j}$ *)
 Dodaj ivicu iz p_r ka temenu p_k i prema trećem temenu p_i drugog
 trougla koji deli ivicu $\overline{p_i p_j}$ sa trouglom $p_i p_k p_j$.
12. LegalizujIvicu($p_r, \overline{p_i p_i}, \mathcal{T}$)
13. LegalizujIvicu($p_r, \overline{p_i p_j}, \mathcal{T}$)
14. LegalizujIvicu($p_r, \overline{p_j p_k}, \mathcal{T}$)
15. LegalizujIvicu($p_r, \overline{p_k p_i}, \mathcal{T}$)
16. Izbaciti temena p_{-1}, p_{-2} i p_{-3} i njihove ivice iz triangulacije \mathcal{T} .
17. **return** \mathcal{T}

Razmotrimo sada detaljnije pretvaranje triangulacije u Deloneovu triangulaciju, što je obavljeno koracima posle 8, odnosno posle 12. Ne osnovu teoreme 10 važi da je triangulacija Deloneova triangulacija ako su sve njene ivice legalne. Algoritmom DeloneTriangulacija, prebacujemo ilegalne ivice sve dok triangulacija ne postane ponovo legalna. Pitanje je koja ivica može da postane ilegalna (slika 3.24), ubacivanjem nove tačke p_r . Ivica $p_i p_j$ koja je bila legalna, može da postane ilegalna ako je jedan od njoj susednih trouglova promenjen. Znači samo ivice novih trouglova treba da budu proverene. Ovo je urađeno pomoćnom funkcijom LegalizujIvicu, koja testira i ako je potrebno prebacuje ivicu. Ako LegalizujIvicu prebaci ivicu, druge ivice takođe mogu postati ilegalne. Zbog toga se ova funkcija poziva rekurzivno za sve potencijalno ilegalne ivice.

LegalizujIvicu ($p_r, \overline{p_i p_j}, \mathcal{T}$)

1. (* Tačka p_r je tačka koja se ubacuje, a $\overline{p_i p_j}$ ivica triangulacije \mathcal{T} , koja bi možda mogla biti prebačena. *)
2. **if** je ivica $\overline{p_i p_j}$ ilegalna
3. **then** Neka je $p_i p_j p_k$ trougao susedan trouglu $p_i p_j p_r$ duž ivice $\overline{p_i p_j}$.
4. (* Prebaci ivicu $\overline{p_i p_j}$ *) Zameni $\overline{p_i p_j}$ sa $\overline{p_r p_k}$.

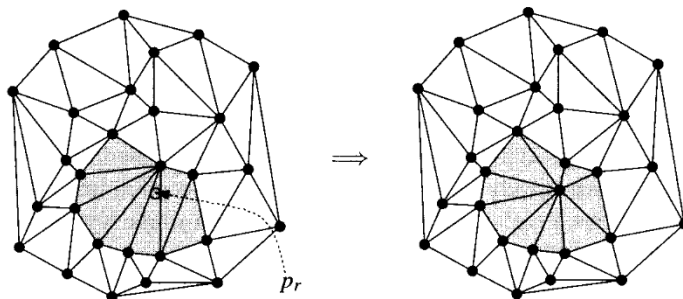
5. LegalizujIvicu $(p_r, \overline{p_i p_k}, \mathcal{T})$
6. LegalizujIvicu $(p_r, \overline{p_k p_j}, \mathcal{T})$



Slika 3.24: Primer ilegalne ivice

Provera u liniji 2, da li je ivica ilegalna može da se obavi primenom leme 8. Dodatna komplikacija su suvišne tačke p_{-1}, p_{-2} i p_{-3} . Ali pre nego se oslobodimo ovih tačaka dokažimo da je algoritam korektan.

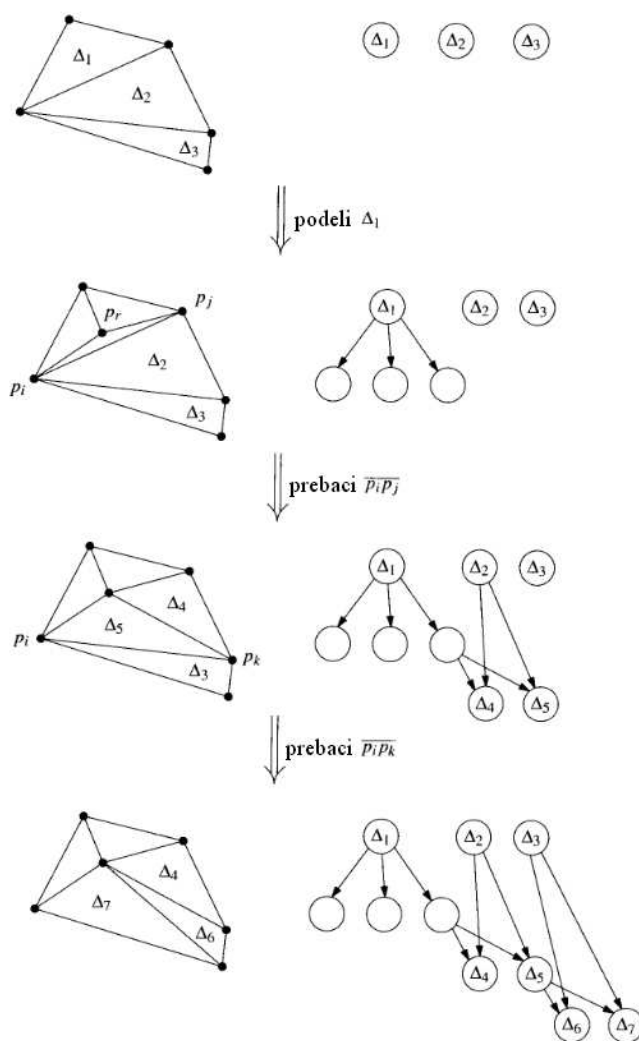
Da bismo dokazali korektnost algoritma, treba da pokažemo da nakon poziva funkcije LegalizujIvicu nije ostala nijedna ilegalna ivica. Jasno da svaka nova ivica nastala ubacivanjem tačke p_r ima za teme tačku p_r . Slika 3.25 to i pokazuje. Ranije smo zaključili da ivica može postati ilegalna samo ako je neki od susednih trouglova izmenjen. Kako algoritam testira sve ivice koje mogu postati ilegalne, dolazimo do zaključka da je algoritam korektan. Do beskonačne petlje, takođe ne može doći, zato što svako prebacivanje ivice povećava ugao-vektora razmatrane triangulacije.

Slika 3.25: Nove ivice su susedne tački p_r , odnosno jedno njihovo teme je u p_r

Ostaje još da opišemo dva važna koraka. Kako da se pronađe trougao koji sadrži tačku p_r i kako da se ispravno tretiraju tačke p_{-1}, p_{-2} i p_{-3} u testu u liniji 2 funkcije LegalizujIvicu.

Da bismo pronašli trougao koji sadrži p_r , paralelno sa građenjem Deloneove triangulacije gradićemo i drvoidnu strukturu pozicija tačaka \mathcal{D} . \mathcal{D} je usmereni aciklični graf. Listovi odgovaraju trouglovima trenutne triangulacije \mathcal{T} . Unutrašnji čvorovi odgovaraju nekim od trouglova iz prethodne faze i već su uništeni.

Kako se gradi struktura \mathcal{D} ? Njeno inicijalno stanje je jedan list koji odgovara trouglu $p_{-1}p_{-2}p_{-3}$. Pretpostavimo sad da delimo trougao $p_i p_j p_k$ tekuće triangulacije na tri (dva) nova trougla. U \mathcal{D} će biti dodata tri (dva) nova lista, a list $p_i p_j p_k$ će postati novi unutrašnji čvor sa pokazivačima na nove listove. Slično i kada zamenjujemo dva trougla $p_k p_i p_j$ i $p_i p_j p_l$ trouglima $p_k p_i p_l$ i $p_k p_l p_j$, prebacivanjem ivice, kreiramo listove za dva nova trougla, a čvorovi $p_k p_i p_j$ i $p_i p_j p_l$ dobijaju pokazivače na nove listove. Slika 3.26 je primer promene strukture \mathcal{D} pri dodavanju nove tačke p_r .



Slika 3.26: Efekat ubacivanja tačke p_r u trougao Δ_1 na strukturu \mathcal{D}

Koristeći \mathcal{D} možemo da lociramo novu tačku p_r , koja se dodaje trenutnoj triangulaciji. Počnimo od korena drveta \mathcal{D} , koji odgovara početnom trouglu $p_{-1}p_{-2}p_{-3}$. Proverimo u kojem od tri deteta leži p_r . I tako nastavljamo dok ne dođemo do lista stable \mathcal{D} . Tom listu dogovara trougao u kojem se nalazi p_r . Kako je stepen svakog čvora najviše tri, potrebno nam je linearno vreme da bismo našli list koji odgovara novoj tački.

Preostalo nam je još samo da nađemo način za lociranje tačaka p_{-1}, p_{-2} i p_{-3} i da implementiramo proveru da li je ivica legalna. Tačke p_{-1}, p_{-2} i p_{-3} treba da se lociraju dovoljno daleko, tako da ne utiču na Deloneovu triangulaciju skupa P . S druge strane ne želimo ni da uvedemo jako velike koordinate potrebne za ovo. Zato biramo tačke tako da trougao $p_{-1}p_{-2}p_{-3}$ sadrži P , a potom modifikujemo proveru legalne ivice tako da ona funkcioniše, ma kako daleko mi izabrali tačke p_{-1}, p_{-2} i p_{-3} .

Preciznije, temena početnog trougla biramo na sledeći način, $p_{-1} := (3M, 0)$, $p_{-2} := (0, 3M)$ i $p_{-3} := (-3M, -3M)$, gde je M maksimalna apsolutna vrednost bilo koje koordinate tačaka u P . Ovim smo osigurali da je P sadržan u trouglu $p_{-1}p_{-2}p_{-3}$. Međutim, kada proveravamo da li je ivica ilegalna mi ne koristimo ove koordinate. Umesto toga uzimamo da p_{-1} leži izvan bilo kog kruga određenog trima tačkama iz P , da p_{-2} leži izvan bilo kog kruga definisanog trima tačkama iz $P \cup \{p_{-1}\}$ i da p_{-3} leži izvan bilo kog kruga definisanog trima tačkama iz $P \cup \{p_{-1}, p_{-2}\}$. Proveru implementiramo na sledeći način. Neka je $\overline{p_i p_j}$ ivica koju testiramo, a p_k i p_l temena trouglova koji dele ivicu $\overline{p_i p_j}$. Postoji nekoliko mogućnosti:

- i i j su negativni: U ovom slučaju ivicu $\overline{p_i p_j}$ smatramo legalnom, jer želimo da sačuvamo ivice velikog trougla.
- tačno jedan od indeksa i, j, k, l je negativan: Ne želimo da specijalna tačka uništi neku od Deloneovih ivica među tačkama iz P . Pa, ako je i ili j negativan, onda je $\overline{p_i p_j}$ ilegalna ivica i zamenićemo je sa $\overline{p_k p_l}$, a u suprotnom ivica $\overline{p_i p_j}$ je legalna.
- tačno dva od indeksa i, j, k, l su negativni: Jedan od i, j je negativan i jedan od k, l mora biti negativan. Ne možemo da razmatramo da su i i k i l negativni ili slučaj da su i i j negativni, jer smo to razmatrali u prvom razmotrenom slučaju. Zato, ako je negativni indeks iz para i, j manji nego negativni indeks iz para k, l onda ivicu $\overline{p_i p_j}$ proglašavamo legalnom, a u suprotnom ilegalnom.
- tačno tri od indeksa i, j, k, l su negativni: Ovo se ne može desiti. I i i j ne mogu oba biti negativni jer je to prvi slučaja koji smo razmatrali. Slučaj da su i i k i l negativni je takođe isključen, jer jedna od tačaka p_k i p_l mora biti tačka p_r koja je upravo ubačena.

Posmatrajmo strukturne promene nastale algoritmom. Odnosno broj trouglova kreiranih i obrisanih kroz algoritamske petlje. Uvedimo, najpre notaciju koju ćemo koristiti: $P := \{p_1, \dots, p_r\}$ i $\mathcal{DG}_r := \mathcal{DG}\{\Omega \cup P_r\}$.

Lema 10 *Broj trouglova kreiranih algoritmom Deloneove triangulacije je najviše $9n + 1$.*

Dokaz: Prvo se kreira jedan trougao čija temena su data skupom Ω . U r -toj iteraciji algoritma, kada ubacujemo tačku p_r , delimo jedan ili dva trougla i kreiramo tri ili četiri nova trougla. Ovo deljenje kreira isti broj ivica u \mathcal{DG}_r , redom $\overline{p_r p_i}, \overline{p_r p_j}, \overline{p_r p_k}$ (a možda i $\overline{p_r p_l}$). Za svaku novu ivicu, potom pozivamo algoritam legalizacije, kojim kreiramo dalje nove trouglove. Ovim se kreiraju i nove ivice, čije je teme tačka p_r . Ako nakon ubacivanja tačke p_r imamo k novih ivica, čije je teme u tački p_r , onda smo kreirali najviše $2(k-3)+3 = 2k-3$ novih trouglova. Broj k je stepen tačke p_k u \mathcal{DG} i obeležavamo ga sa $\text{deg}(p_r, \mathcal{DG}_r)$.

Šta je onda očekivani stepen tačke p_r u svim mogućim permutacijama skupa P ? Ovim želimo da ograničimo očekivani stepen tačke p_r , ili bilo kog slučajnog elementa skupa P . Na osnovu teoreme 2 Deloneov graf \mathcal{DG}_r ima najviše $3(r+3) - 6$ ivica. Tri od njih su ivice trougla $p_{-1}p_{-2}p_{-3}$, pa je zato ukupni stepen temena u P_r manji od $2[3(r+3) - 9] = 6r$. Ovo znači da je očekivani stepen proizvoljne tačke u P_r najviše 6. Iz ovoga možemo da ograničimo broj trouglova kreiranih u r -tom koraku na sledeći način:

$$\begin{aligned} E[\text{broj trouglova kreiranih u } r - \text{tom koraku}] &\leq E[2 \text{deg}(p_r, \mathcal{DG}_r) - 3] \\ &= 2E[\text{deg}(p_r, \mathcal{DG}_r)] - 3 \leq 2 * 6 - 3 = 9 \end{aligned}$$

Ukupan broj kreiranih trouglova je jedan trougao $p_{-1}p_{-2}p_{-3}$, koji je postojao na samom početku kome se dodaje broj kreiranih trouglova pri svakom ubacivanju nove tačke. Ovim dobijamo da je očekivani broj trouglova ograničen sa $1 + 9n$. \square

Sada dolazimo do važnog zaključka.

Teorema 12 *Deloneova triangulacija skupa P od n tačaka u ravni može biti izračunata u vremenu $\mathcal{O}(n \log n)$ i koristeći $\mathcal{O}(n)$ memorije.*

Dokaz: Korektnost algoritma je već dokazana. Što se tiče memoriskih zahteva, struktura za pretraživanje \mathcal{D} može biti više nego linearna. Međutim, svaki čvor u \mathcal{D} odgovara jednom trouglu, a na osnovu prethodne leme očekivani broj čvorova je $\mathcal{O}(n)$.

Da bismo ograničili vreme koje algoritam zahteva, ignorišimo najpre vreme potrebno za lociranje trougla. Sada je vreme proporcionalno broju kreiranih trouglova. Na osnovu prethodne lema, odbacujući vreme potrebno za lociranje trougla, zaključujemo da je vreme potrebno za algoritam $\mathcal{O}(n)$.

Ostaje još da odredimo vreme potrebno za lociranje trougla, kome nova tačka pripada. Vreme potrebno za lociranje tačke p_r u trenutnoj triangulaciji je linearno, shodno broju čvorova koje algoritam poseti u \mathcal{D} . Bilo koji posećeni čvor odgovara trouglu koji je kreiran u nekoj prethodnoj fazi, odnosno izbačen je i sadrži p_r .

Trugao $p_i p_j p_k$ može biti izbačen iz triangulacije zbog jednog od dva razloga:

- Nova tačka p_l je ubačena unutar (ili na ivicu) trougla $p_i p_j p_k$ i trougao se deli na tri ili dva nova trougla.
- Prebacivanjem vice truglovi $p_i p_j p_k$ i $p_i p_j p_l$ su zamenjeni truglovima $p_k p_i p_l$ i $p_k p_j p_l$.

U prvom slučaju trougao $p_i p_j p_k$ je bio Deloneov trougao pre nego smo ubacili tačku p_l . U drugom slučaju ili je $p_i p_j p_k$ bio Deloneov trougao, a potom je p_l ubačena ili je $p_i p_j p_l$ bio trougao, a potom je p_k ubačen. Ako je $p_i p_j p_l$ bio Deloneov trougao, onda činjenica da je ivica $\overline{p_i p_j}$ prebacivana, znači da i p_k i p_r leže unutar opisanog kruga trougla $p_i p_j p_l$.

Opisani krug svakog trougla Δ , koji posetimo locirajući novu tačku p_r , sadrži unutar sebe podskup skupa P . Obeležimo ga sa $K(\Delta)$. Pri tome naravno važi $p_r \in K(\Delta)$. Trougao Δ može biti posećen najviše jednom za bilo koju tačku iz $K(\Delta)$. Zato je ukupno vreme za lociranje tačke:

$$\mathcal{O}(n + \sum_{\Delta} (\text{card}(K(\Delta)))) \quad (3.1)$$

gde je sumirano po svim Deloneovim trouglovima Δ kreiranim algoritmom. Sledećom lemom dokazaćemo da je ova suma upravo $\mathcal{O}(n \log n)$. \square

Ostaje da se odredi granica broja elemenata u $K(\Delta)$. Ako je Δ trougao Deloneove triangulacije skupa $\Omega \cup P_r$, koliko može biti $\text{card}(K(\Delta))$? Za $r = 1$ kardinalnost skupa je tačno n , a za $r = n$ je nula. A šta je sa vrednostima između graničnih slučajeva? Broj tačaka koje leže unutar opisanog kruga trougla $\Delta \in \mathcal{DG}_r$ je oko $\mathcal{O}(n/r)$. Međutim, ovo ne važi za sve trouglove u \mathcal{DG}_r . Suma (3.1) ponaša se kao da važi za sve. Dokažimo zato da je ovo tvrđenje tačno.

Lema 11 *Ako je P skup tačaka u opštem položaju, onda važi*

$$\sum_{\Delta} (\text{card}(K(\Delta))) = \mathcal{O}(n \log n).$$

gde se sumira po svim Deloneovim trouglovima Δ kreiranim algoritmom.

Dokaz: S obzirom da je P u opštem položaju, svaki podskup P_r je takođe u opštem položaju. Iz ovoga sledi da je triangulacija nakon dodavanja tačke p_r jedinstvena triangulacija nad $\mathcal{DG}(\Omega \cup P_r)$. Označimo skup trouglova iz $\mathcal{DG}(\Omega \cup P_r)$ sa \mathcal{T}_r . Po definiciji, skup Deloneovih trouglova kreiranih u r fazi je $\mathcal{T}_r \setminus \mathcal{T}_{r-1}$. Predefinišimo sumu koju želimo da ograničimo sa

$$\sum_{r=1}^n \left(\sum_{\Delta \in \mathcal{T}_r \setminus \mathcal{T}_{r-1}} \text{card}(K(\Delta)) \right).$$

Za tačku q , obeležimo sa $k(P_r, q)$ broj trouglova $\Delta \in \mathcal{T}_r$, takvih da $q \in K(\Delta)$, a sa $k(P_r, q, p_r)$ broj trouglova $\Delta \in \mathcal{T}_r$ za koje važi ne samo da $q \in K(\Delta)$, već i da

je p_r jedno od temena trougla Δ . Podsetimo se da je p_r temene svih Deloneovih trouglova kreiranih u r fazi, pa zato važi

$$\sum_{\Delta \in \mathcal{T}_r \setminus \mathcal{T}_{r-1}} \text{card}(K(\Delta)) = \sum_{q \in P \setminus P_r} k(P_r, q, p_r). \quad (3.2)$$

Kako je tačka p_r bilo koja r -ta po redu izabrana tačka iz skupa P , to $k(P_r, q, p_r)$ zavisi samo od izbora tačke p_r . S druge strane trougao $\Delta \in \mathcal{T}_r$ ima za teme tačku p_r sa verovatnoćom $3/r$, pa važi

$$E[k(P_r, q, p_r)] \leq \frac{3k(P_r, q)}{r}.$$

Ako ovo dalje sumiramo po $q \in P \setminus P_r$ i koristimo jednakost (3.2), dobijamo da važi

$$E\left[\sum_{\Delta \in \mathcal{T}_r \setminus \mathcal{T}_{r-1}} \text{card}(K(\Delta))\right] \leq \frac{3}{r} \sum_{q \in P \setminus P_r} k(P_r, q). \quad (3.3)$$

Svaki $q \in P \setminus P_r$ ponaša se isto kao i p_{r+1} , pa tako dobijamo

$$E[k(P_r, p_{r+1})] = \frac{1}{n-r} \sum_{q \in P \setminus P_r} k(P_r, q).$$

Spajajući ovo dalje sa jednakošću (3.3) dobijamo

$$E\left[\sum_{\Delta \in \mathcal{T}_r \setminus \mathcal{T}_{r-1}} \text{card}(K(\Delta))\right] \leq 3\left(\frac{n-r}{r}\right)E[k(P_r, p_{r+1})].$$

Šta je $k(P_r, p_{r+1})$? To je broj trouglova Δ u \mathcal{T}_r za koje važi da $p_{r+1} \in K(\Delta)$. Na osnovu prvog kriterijuma teoreme (8), ovi trouglovi su upravo trouglovi iz \mathcal{T}_r , koji će nestati ubacivanjem tačke p_{r+1} . Preformulišimo zato prethodnu nejednakost

$$E\left[\sum_{\Delta \in \mathcal{T}_r \setminus \mathcal{T}_{r-1}} \text{card}(K(\Delta))\right] \leq 3\left(\frac{n-r}{r}\right)E[\text{card}(\mathcal{T}_r \setminus \mathcal{T}_{r+1})].$$

Torema (5) kaže da je broj trouglova u \mathcal{T}_m tačno $2(m+3) - 2 - 3 = 2m+1$. Zato je broj trouglova uništenih ubacivanjem tačke p_{r+1} za dva manji od broja trouglova kreiranih ubacivanjem tačke p_{r+1} tako da prethodnu nejednakost možemo ponovo da preformulišemo

$$E\left[\sum_{\Delta \in \mathcal{T}_r \setminus \mathcal{T}_{r-1}} \text{card}(K(\Delta))\right] \leq 3\left(\frac{n-r}{r}\right)(E[\text{card}(\mathcal{T}_{r+1} \setminus \mathcal{T}_r)] - 2).$$

Do sada smo P_r smatrali fiksним skupom. Ali sad možemo uzeti prosečne

vrednosti nad svim skupovima $P_r \subset P$ i to sa obe strane prethodne nejednakosti. Ali prethodna nejednakost važi za sve permutacije skupa P .

Znamo i da je broj trouglova kreiranih ubacivanjem tačke p_{r+1} jednak broju ivica čije je teme tačka p_{r+1} u \mathcal{T}_{r+1} kao i da je očekivani broj ivica najviše 6. Tako zaključujemo da važi

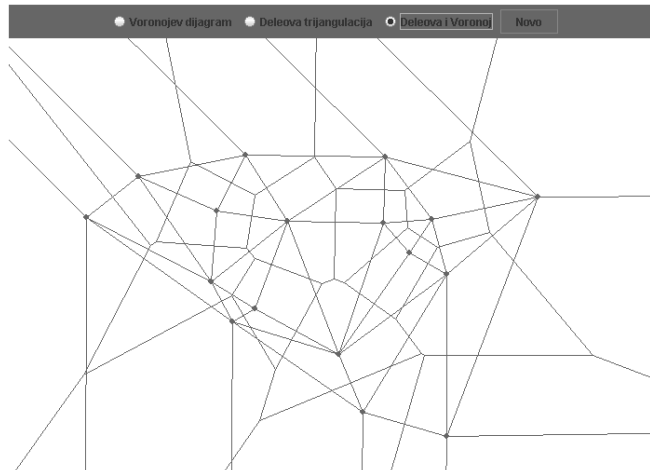
$$E\left[\sum_{\Delta \in \mathcal{T}_r \setminus \mathcal{T}_{r-1}} \text{card}(K(\Delta))\right] \leq 12\left(\frac{n-r}{r}\right).$$

Sumiranjem po r dokazujemo lemu. \square

3.3.3 Implementacija inkrementalnog algoritma

Prilog ovom radu je inkrementalni algoritam Deloneove triangulacije implementiran kao aplet u Javi 1.4.

Slika 3.27 ilustruje animaciju implementiranog algoritma. Sa unosom svakog novog centra, odnosno klikom mišem na panel, prikazuje se nova scena. Ponuđene su tri opcije. A one su, Voronoj dijagrama, Deloneove triangulacije ili iscrtavanje oba grafa istovremeno. Među ponuđenim akcijama nalazi se i dugme za poništavanje unesenih centara, čime se nudi unos novih centara te ponovno prikazivanje grafova sa novoizabranim centrima.



Slika 3.27: Implementacija inkrementalnog algoritma Deloneove triangulacije

U implemetiranoj strukturi nalazi se fajl *applet.html*, kojim se pokreće kompletni projekat. Struktura projekta se zaniva na jednom direktorijumu u kom se nalaze Java klase i navedenom html fajlu izvan njega.

Za realizaciju ovog algoritma potrebno je najpre izneti detalje organizacije struktura podataka. Centar je klasa sa nizom od dva elementa, koja predstavlja x i y koordinatu. Trougao se pamti kao lista tri tačke, čiji redosled nije

bitan. Sama triangulacija pamti se kao hash mapa. Tu je ključ trougao, a element je skup njemu susednih trouglova. To će reći da je prostorna složenost algoritma $\mathcal{O}(n^2)$.

Algoritam najpre nalazi postojeći trougao kome pripada novounesena tačka. Ovde se koristi mešoviti proizvod, odnosno osobina mešovitog proizvoda da menja znak, ukoliko se promeni redosled tačaka za koje se računa. Neka je razmatrani trougao $\triangle ABC$. Onda je ovo glavna matrica za koju računamo determinantu:

$$\begin{bmatrix} 1 & 1 & 1 \\ a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \end{bmatrix}$$

Neka je nova tačka D . Ona se nalazi unutar trougla $\triangle ABC$ ako i samo ako se nalazi sa iste strane duži BC kao i tačka A , sa iste strane duži AB kao i tačka C i sa iste strane duži CA kao i tačka B . Da bismo ovo proverili računamo, determinantu matrice koje se od prethodno navedene razlikuju u tome, što ćemo a_0 i a_1 zameniti koordinatama nove tačke i tako proveriti da li se nova tačka nalazi sa iste strane duži BC kao i tačka A , te analogno za ostale dve duži. Ako je u sva tri slučaja znak vrednosti determinate isti kao i znak glavne determinante, onda se tačka nalazi unutar trougla. Ako je vrednost neke od poslednje tri determinante nula onda se nova tačka nalazi na ivici razmatranog trougla. Ovom implemetacijom taj slučaj tretira se isto kao da D je unutar trougla.

Sledeći korak je nalaženje onih trouglova na koje utiče novi centar te zamena tih trouglova novim trouglovima. To su svi oni trouglovi u čijem se opisanom krugu nalazi novi centar. Za njih algoritam, nalazi omotač. Omotač je zapravo lista duži koje formiraju mnogougao. Novi trouglovi su te duži kojima se kao treće teme dodeljuje novi centar, kao što je prikazano na slici 3.25. Prilikom zamene starih trouglova novima, potrebno je ažurirati i liste susednih trouglova. Vremenska složenost ovog algoritma je tako $\mathcal{O}(n \log n)$, a u najgorem slučaju n^2 .

Implementacija Voronj dijagrama je specifična. Pri iscrtavanju ovog dijagrama korišćeno je sledeće tvrđenje.

Tvrđenje 2 *Duži koje spajaju centre opisanih krugova susednih trouglova Deloneove triangulacije grade Voronj dijagram.*

Dokaz: Ova teorema je posledica teorema [3] i [9]. \square

4

Zaključak

U ovom radu su prikazani inkrementalni algoritam, algoritam zasnovan na razlaganju i algoritam pokretne linije za Voronoj dijagram, te inkrementalni algoritam i algoritam zasnovan na razlaganju za konstrukciju Deloneove triangulacije nad zadatim skupom tačaka u ravni. Kako su ova dva grafa dualna međusobno, iz konstruisanog jednog od njih može se konstruisati drugi.

Dalji rad na ovu temu mogao bi se baviti algoritmima za konstrukciju Voronoj dijagrama i Deloneove triangulacija za skup tačaka u prostoru, odnosno za skup tačaka u više dimenzionom prostoru.

Bibliografija

- [1] M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf: Computational Geometry Algorithms and Applications, Springer-Verlag Berlin Heidelberg 1997, 2000, poglavlja 7, 9
- [2] F. Aurenhammer, R. Klein: Voronoi Diagrams, Technical Report 198, FernUniversität Hagen, Department of Computer Science, Germany, 1996.
- [3] R. Bin Muhammad: Voronoi Diagram using Divide-and-Conquer Paradigm; <http://www.personal.kent.edu/~rmuhamma/Compgeometry/MyCG/Voronoi/DivConqVor/divConqVor.htm>
- [4] R. Bin Muhammad: Voronoi / Delaunay / Convex Hull Incremental Method - Applet; <http://www.personal.kent.edu/~rmuhamma/Compgeometry/MyCG/Voronoi/Incremental2/incremental2.htm>
- [5] S. Peterson: Computing Constrained Delaunay Triangulations in the Plane <http://www.geom.uiuc.edu/~samuelp/del-project.html>