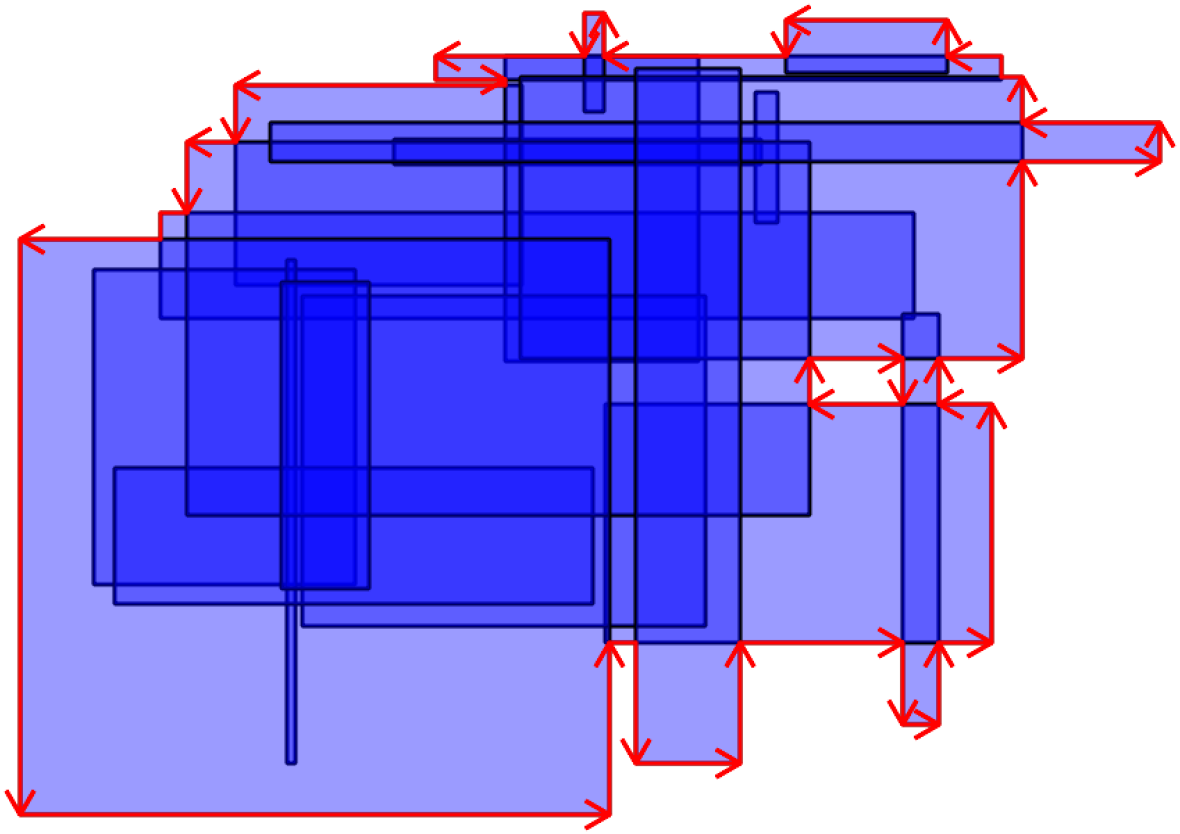
Geometrijski algoritmi @ MATF

Algoritam za određivanje konture skupa pravougaonika

Nikola Vuković



Opis problema

Za dati skup pravougaonika odrediti skup duži koje ograničavaju konturu njihove unije.

Ulaz: *skup od 2*n tačaka u ravni koje čine suprotna temena pravougaonika*

Izlaz: *skup orijentisanih duži koje čine konturu*

Naivno rešenje problema

Oba rešenja se sastoje iz dve faze. Prva faza u oba slučaja prolazi kroz pravougaonike tehikom brišuće prave, sa ciljem da izdvoji vertikalne ivice konture. Druga faza, preuzeta uz dopunu iz

teksta Finding the contour of a union of iso-oriented rectangles (Lipski, Preparata 1980), povezuje vertikalne ivice horizontalnim.

Naivno rešenje održava skup tačaka koje su trenutno unutar skupa J. Obilazimo sve vertikalne ivice pravougaonika redom s leva na desno. Na početku pravougaonika ubacujemo tačke u J, i ako menjaju granice J, ubacujemo tu razliku u skup vertikalnih ivica. Suštinski tražimo I ∩ ^J. Pri obradi desne (izlazne) ivice pravougaonika izbacujemo dve tačke koje čine ivicu iz J i ponovo tražimo I ∩ ^J i ubacujemo u skup vertikalnih ivica. Algoritam je kompleksan jer postoji dosta specijalnih slučajeva koje mora da pokrije. Bez ulaženja u detalje, za svaku vertikalnu stranicu pravougaonika prolazimo kroz sve tačke trenutno u J, što u teoriji znači da je najgori slučaj O(m*p) gde je m broj pravougaonika a p broj ivica u konturi. Ova implementacija je manje optimalna od drugog algoritma u kom je kompleksnost ove faze O(mlogm), ali ipak ne menja najgori slučaj $O(m^2)$ - više u analizi LP algoritma.

Druga faza spaja vertikalne ivice horizontalnim. Proces neformalno opisujemo u nastavku: Za svaku trojku (x,g,d) koja predstavlja vertikalnu ivicu konture, kreira se još jedna trojka (x,d,g), gde je g gornji kraj ivice a d donji ($g > d$). Novonastali skup trojki (x,y_1,y_2) se sortira leksikografski po redu (y_1,x). Za svake dve trojke $a_{2k-1} = (l,y,y_1)$ i $a_{2k} = (d,y,y_2)$ kontura sadrži horizontalnu ivicu (y,l,d). Ivica je orijentisana s leva na desno ako je ivica kojoj odgovara (l,y,y_1) usmerena ka dole i $y < y_1$, ili ako je ivica usmerena ka gore i $y_1 < y$. U suprotnom je orijentisana sa desna na levo. Ova faza je detaljnije opisana u LP 1980.

Lipski-Preparata algoritam

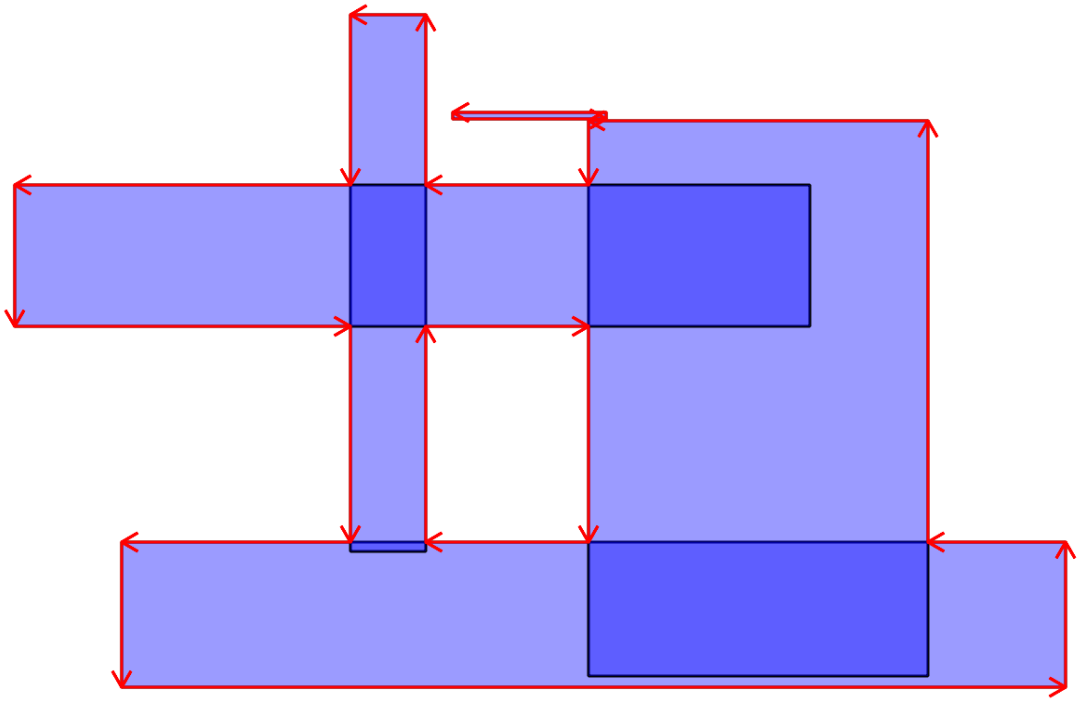
Efikasnija verzija pomenutog algoritma se zasniva na istom principu. Razlika je u procesu pronalaženja I ∩ ^J. Dok naivni algoritam prolazi kroz sve tačke trenutno u J, algoritam opisan u LP 1980 koristi segmentno drvo za efikasno pronalaženje preseka tekuće ivice i J. Svaki čvor u stablu predstavlja jednu hipotetičku stranicu. Tačke se normalizuju tako što se sortiraju po Y koordinati i u čvoru se čuva njihov indeks umesto Y koordinata. Čvor takođe čuva status: prazan ako segment koji pokriva nije sadržan u J; pun ako je segment koji pokriva sadržan u J ivicom koja ga pokriva od početka do kraja; i delimičan ako sam segment nije pokriven jednom stranicom od početka do kraja, ali neki njegovi deca-segmenti jesu. Pri obradi stranica pravougaonika, koristi se stek koji čuva trenutni presek I ∩ ^J. Na kraju obrade stek sadrži traženi presek.

Napomena: u odnosu na originalni pseudokod pomenutog rada, dodat je argument funkciji compl koji označava da li je roditelj čvora koji se obrađuje pun. Bez ove informacije, dodavali bismo stranice koje su već pokrivene sa J. Nisam siguran da li je ovo propust originalnog

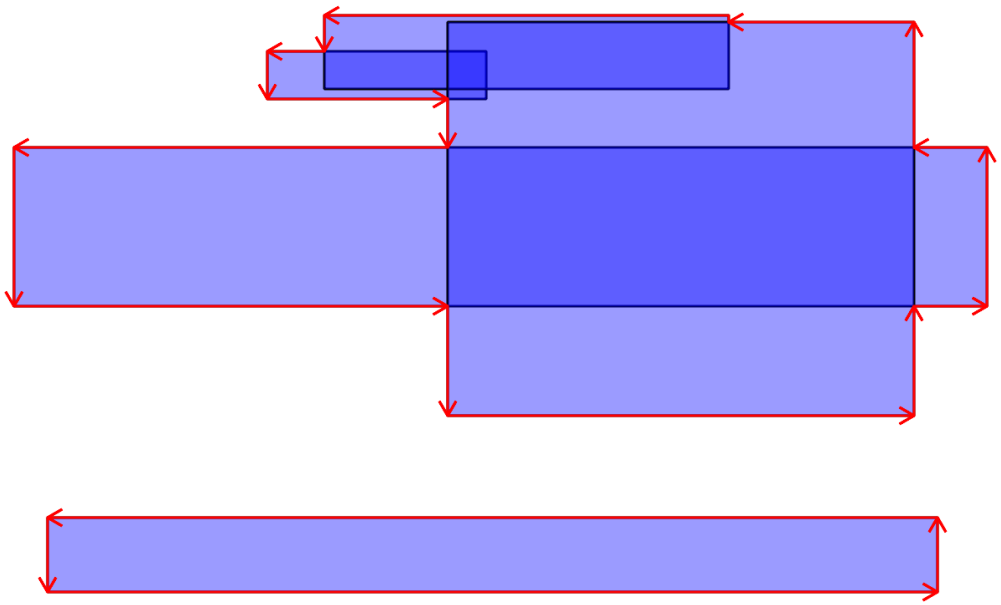
pseudokoda, ili su autori pretpostavljali da će se ovaj detalj rešiti kroz implementaciju, ili sam pak prevideo neki deo opisanog algoritma. U svakom slučaju, suština originalnog algoritma je ispoštovana i implementirana.

Složenost algoritma je opisana i dokazana u pomenutom tekstu kao $O(mlogm + p\ log(2m^2/p))$, gde je m broj pravougaonika a p broj ivica u konturi. Kako je p ograničeno sa m (tačnije za m>4 važi da p pripada $[4, m^2 + 4m]$), ako složenost izražavamo samo kroz veličinu ulaza, može se definisati kao $O(m^2)$ u najgorem slučaju. Tekst dalje dokazuje da postoje slučajevi za koje se algoritam bolje ponaša. Slučaj kada oblik ne sadrži rupe algoritam obrađuje u O(mlogm). Postoje drugi tekstovi koji predlažu poboljšanje vremenske i prostorne efikasnosti u odnosu na ovaj algoritam, ali oni ovde nisu obrađeni.

Vizuelizacija algoritma (opciono)

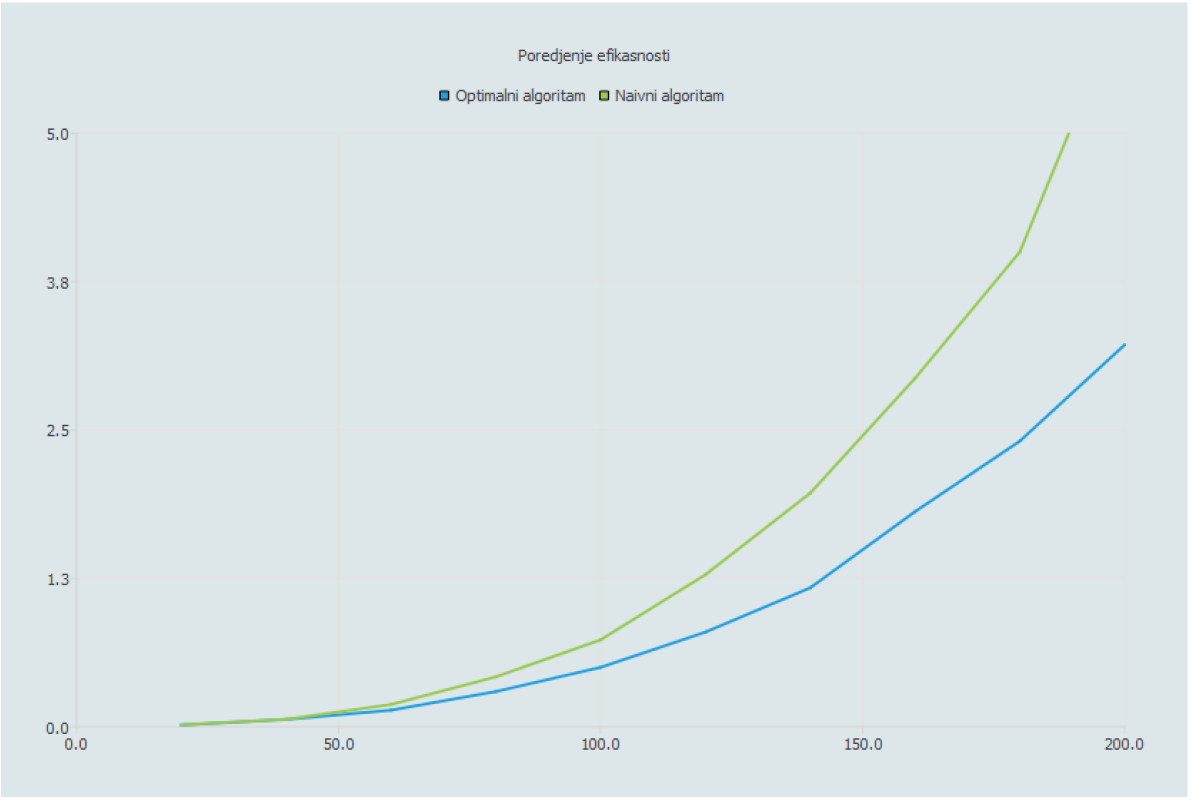


Kontura oblika koji sadrži rupu



Kontura oblika koji se sastoji od odvojenih delova

Poredjenje efikasnosti naivnog i naprednog algoritma



Poređenje vremena izvršavanja od 20 do 200 pravougaonika

Testiranje ispravnosti algoritma

Ovde treba da bude opisano na koji način je izvršeno testiranje algoritma.

Naziv testa	Opis testa	Ulaz (pravouganonici)	Očekivani izlaz oba algoritma (kontura)
jedan_pravougaonik	Jedan pravougaoni	100 100 200 200	100 200 100 100 200 100 200 200 100 100 200 100 200 200 100 200
dva_odvojena_pravougaonika	Dva odvojena pravougaonika	100 100 200 200 300 100 400 200	100 200 100 100 200 100 200 200 300 200 300 100 400 100 400 200 100 100 200 100 300 100 400 100 200 200 100 200 400 200 300 200
dva_pravougaonika_sa_presecima	Dva pravougaonika koji se seku	100 100 200 200 150 150 250 250	100 200 100 100 150 250 150 200 200 100 200 150 250 150 250 250 100 100 200 100 200 150 250 150 150 200 100 200 250 250 150 250
konkavni_oblik	Skup pravougaonika čija unija je konkavna po X i Y osama (ima šupljine)	100 100 400 200 150 150 200 400 300 150 350 300 250 250 400 350	100 200 100 100 150 400 150 200 200 200 200 400 250 350 250 250 300 250 300 200 350 200 350 250 400 100 400 200 400 250 400 350 100 100 400 100 150 200 100 200 300 200 200 200 400 200 350 200 250 250 300 250 350 250 400 250 400 350 250 350 200 400 150 400
oblik_s_rupom	Skup pravougaonika čija unija ima rupu	100 100 400 200 150 150 200 300	100 350 100 250 100 200 100 100

	okruženu drugim pravougaonicima	300 150 350 300 100 250 400 350	150 250 150 200 200 200 200 250 300 250 300 200 350 200 350 250 400 100 400 200 400 250 400 350 100 100 400 100 150 200 100 200 300 200 200 200 400 200 350 200 100 250 150 250 200 250 300 250 350 250 400 250 400 350 100 350
pravougaonici_koji_se_prekrivaju	Tri pravougaonika koji se poklapaju po dužini, ili se prekrivaju potpuno.		100 300 100 200 100 200 100 100 400 100 400 200 400 200 400 300 100 100 400 100 400 300 100 300