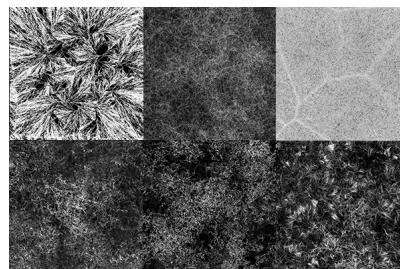


Proceduralno generiranje rasporeda ulica gradova korištenjem slučajnih šetnji

Ivan Vlahov, univ. bacc. ing. comp.
mentorica Željka Mihajlović, prof. dr. sc.
Fakultet Elektrotehnike i Računarstva
Zagreb, Hrvatska
iv51747@fer.hr

Sažetak—U ovom radu prikazane su metode generiranja rasporeda ulica gradova korištenjem djeljivih agregirajućih šetača. Prikazan je utjecaj parametara na konačan izgled rasporeda te su objašnjeni dodaci na osnovni algoritam kako bi se adaptirao za zadani problem. Navedeno je nekoliko vrsta rasporeda ulica koje ovaj algoritam može uspješno i konzistentno generirati.

Gljučne riječi—Proceduralno generiranje, slučajna šetnja, vjerojatnost, geometrija, dizajn videoigara



Slika 1. Primjeri šetnji DAW-ova, Izvor: [1]

I. UVOD

Proceduralno generiranje jako je zastupljena metoda izrade 2D i 3D objekata, tekstura, zvukova i skoro svega što bi moglo poslužiti u izradi videoigara, a u to spadaju i proceduralno generirani gradovi. Kako bi se generirao model grada, za početak je potrebno odrediti njegovu "infrastrukturu", odnosno tlocrt, na koji se različitim metodama mogu dodati objekti poput zgrada, kuća, parkova i sl.

Jedan način za generiranje tlocrta je korištenjem tzv. *Dividing-Aggregating Walkera* (DAW-ova), odnosno djeljivih-agregirajućih šetača opisanih u [1]. U ovom radu bit će ukratko objašnjene te slučajne šetnje, metode kojima je pomoću njih moguće generirati raspored ulica gradova, te neke adaptacije specifične za generiranje rasporeda ulica.

Kao inspiracija za ovaj rad uzet je "City Map" Justina Mathieua [2] te se dio koda za ovaj rad temelji na tom projektu.

II. IMPLEMENTACIJA

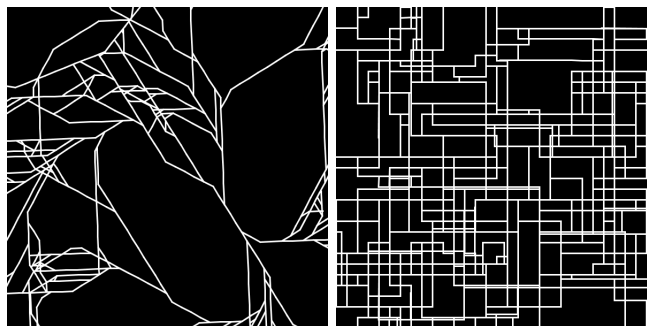
Općenito, slučajna šetnja je matematički proces ponavljanja slučajnih koraka. U slučaju DAW-ova, šetnja se sastoji od više mogućih koraka. Svaki DAW ima svoju poziciju i smjer kretanja. U jednom koraku šetač se može okrenuti za neki kut iz nekog zadanog intervala s nekom zadanom vjerojatnošću. Osim pozicije i smjera kretanja, "pamti" i pozicije koje je već posjetio. Nadalje, u svakom koraku se može podijeliti s nekom zadanom vjerojatnošću. Dijeljenje je proces stvaranja novog DAW-a na istoj poziciji, ali s drugačijim smjerom kretanja. Izračun smjera kretanja također se zadaje nekim parametrom. U [1] navodi se i korak zaustavljanja - u slučaju da šetač naiđe na već posjećenu poziciju, zaustavlja se. U implementaciji iz ovog rada šetač se zaustavlja i kad naiđe na jedan od rubova prostora za iscrtavanje traga.

Ukratko, parametri koji se zadaju DAW-u i koji određuju izgled traga koji ostavi za sobom su:

- Duljina koraka
- Vjerojatnost okretanja
- Kut okretanja
- Vjerojatnost dijeljenja
- Kut dijeljenja

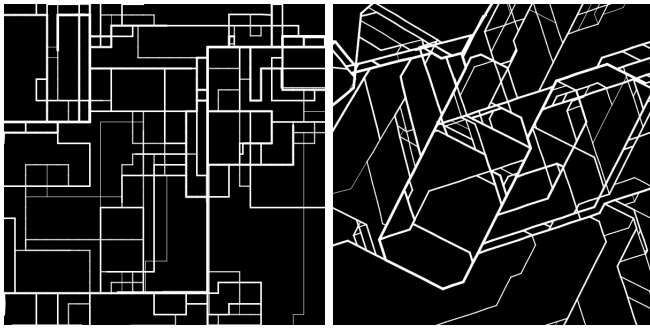
Također, bitan je i početni broj šetača, njihove pozicije i smjerovi.

Program kojim su izrađene sljedeće slike napisan je u Processingu [4] te generira slike u PNG formatu. Sav programski kod objavljen je na GitHub stranici autora [3].



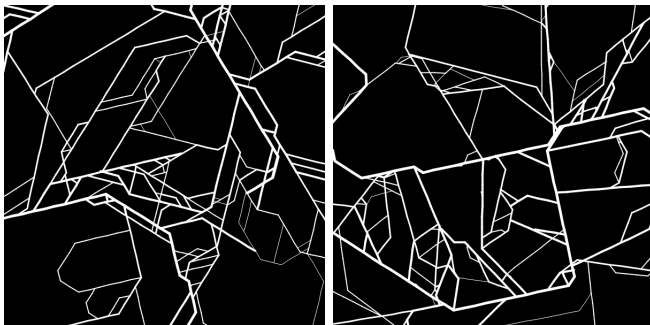
Slika 2. Primjeri s kutovima $\pi/6$ (lijevo) i $\pi/2$ (desno)

Kako bismo razlikovali npr. glavnu cestu od neke male ulice ili puta, možemo dodati parametar stupnja šetača, koji može označavati koliko je neka ulica široka, odnosno koliko širok trag će ostaviti na prostoru za iscrtavanje. Prilikom dijeljenja, šetač može stvoriti samo novog šetača istog ili manjeg stupnja.



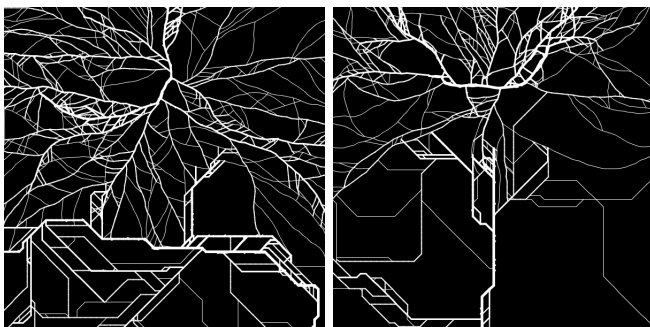
Slika 3. Primjeri s kutovima $\pi/2$ (lijevo) i $\pi/4$ (desno)

Također, skretanja iz jedne ulice u drugu nisu uvijek pod točno određenim kutem, pa bi se i u ovaj model trebala dodati neka slučajnost. Kutu okretanja ili dijeljenja možemo dodati ili oduzeti slučajnu vrijednost uzetu iz normalne distribucije.



Slika 4. Rasporedi ulica s dodanim slojevima slučajnosti

Neki gradovi (npr. Zagreb) sastoje se od više dijelova koji su se razvijali na različite načine, pa i raspored ulica u tim dijelovima nije jednak. Takav izgled može se ostvariti korištenjem više šetača s različitim parametrima.

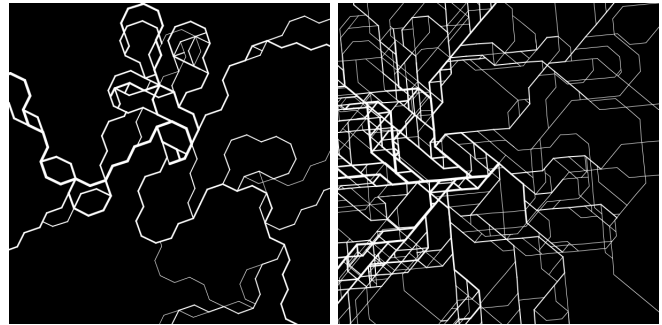


Slika 5. Rasporedi nastali kombiniranjem dvaju šetača s različitim parametrima, gornji šetači stvaraju puno "stablastiju" strukturu nego donji šetači

III. REZULTATI I DISKUSIJA

Iz prikazanih slika može se primijetiti da ova jednostavna metoda, uz jako malo dodatka, može generirati vrlo dobre rasporede ulica. Usprkos tome, nije jednostavno odabrati parametre pomoću kojih će se generirati kvalitetan rezultat.

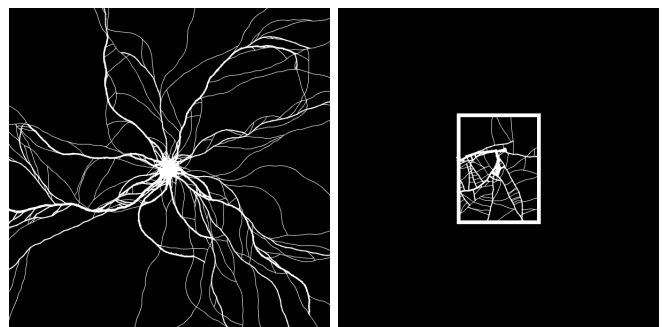
Postavljanjem vjerojatnosti okretanja ili vjerojatnosti dijeljenja na preveliku vrijednost, dovoljan je samo jedan šetač da šetnje vrlo brzo postanu kaotične i/ili ne izgledaju kao rasporedi ulica.



Slika 6. Prevelika vjerojatnost okretanja (lijevo) i prevelika vjerojatnost dijeljenja (desno) s kutom $\pi/4$

Odabirom dobrih vrijednosti kutova, vjerojatnosti i duljine koraka mogu se generirati različite vrste rasporeda ulica. Npr. želimo li generirati raspored sličan onome na otoku Manhattanu ili u Barceloni, odabrat ćemo veće vjerojatnosti dijeljenja i okretanja, a kutove ćemo postaviti na $\pi/2$. Primjer takvog rasporeda može se vidjeti na Slici 2. U slučaju da želimo generirati grad s radijalnim rasporedom, postaviti ćemo više šetača u istu točku s nasumičnim ili jednako razmaknutim početnim smjerovima te većim vjerojatnostima dijeljenja. Želimo li glavne ceste koje idu u jednom pravcu (npr. istok-zapad), a iz kojih se račva više manjih ulica, postaviti ćemo vjerojatnost okretanja na jako malu vrijednost (ili nulu), a vjerojatnost dijeljenja na veliku vrijednost s kutom $\pi/2$.

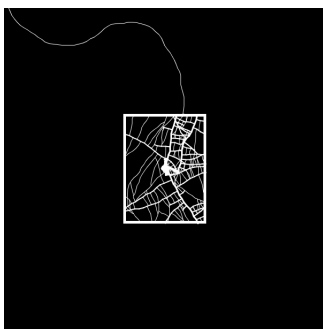
Iako se ulice generiraju proceduralno, možemo i ručno odrediti nekakvu strukturu postavljanjem glavnih ulica ili zauzimanjem nekog prostora za npr. parkove, zelene površine i sl. Unutar takvih struktura bismo mogli generirati jedan tip rasporeda i tako osigurati željene oblike na jednostavan način.



Slika 7. Primjer radijalnog rasporeda (lijevo) te rasporeda ograničenog unaprijed zadanim oblikom (desno)

Jedan od problema pri generiranju rasporeda jest izračun uvjeta zaustavljanja. Zaustavljanje zbog izlaska iz prostora za iscrtavanje je jednostavan zadatak, međutim, zaustavljanje zbog preklapanja s posjećenom koordinatom se pokazalo problematičnim. Spremanje posjećenih koordinata u nekakav

set nije moguće jer bismo morali spremati ogromne količine podataka te ih pretraživati nekoliko stotina ili tisuća puta po sličici što nije praktično. Iz tog razloga je kao uvjet zaustavljanja uzeta provjera boje piksela između trenutne i sljedeće pozicije šetača. Zbog zadnjeg traga, koji se nalazi u diskretnom prostoru za iscrtavanje, može se dogoditi da trag završi malo iza trenutne pozicije, zbog čega se dio puta od trenutne do sljedeće pozicije često treba preskočiti kako se ne bismo nepotrebno zaustavili. Također, zbog ovog problema šetači s malim duljinama koraka se odmah spuste na niži stupanj i relativno brzo se zaustave. Na većim duljinama koraka se može dogoditi da prilikom preskakanja preskočimo i neki put koji je zapravo trebao uzrokovati zaustavljanje. Za ovaj problem tijekom izrade ovog rada nije pronađeno nijedno kvalitetno rješenje osim "pametnog" izbjegavanja problema odabirom ispravnih parametara.



Slika 8. Primjer šetača koji je "preskočio" put koji je trebao uzrokovati zaustavljanje

IV. ZAKLJUČAK I DALJNI RAD

Kao što je navedeno u prošlom poglavlju, ova metoda proizvela je zadovoljavajuće rezultate. Promjenom parametara moguće je generirati različite rezultate, a moguće ih je i kombinirati kako bi se došlo do složenijih rasporeda.



Slika 9. Primjer jednog realističnog rasporeda ulica

Ovaj rad predstavlja tek prvi korak u postupku proceduralnog generiranja gradova. Sljedeći koraci bili bi prebacivanje u vektorski format, dodavanje još proceduralno generiranog sadržaja poput rijeka, mora, šuma, parkova i sl., određivanje položaja i izgleda građevina te generiranje 3D modela iz zadanog tlocrta.

LITERATURA

- [1] Ciphird, "Pattern formation using Dividing-Aggregating Walkers", <https://ciphird.com/2021/03/17/pattern-formation-using-dividing-aggregating-walkers/>
- [2] J. Mathieu, "City Map", <https://openprocessing.org/sketch/1236570>
- [3] I. Vlahov, streets, <https://github.com/vlahovivan/streets>
- [4] Processing, <https://processing.org/>