Fakulteta za Matematiko in Fiziko v Ljubljani

Vzorčenje točk v ravnini

Opis algoritma pri predmetu Računalništvo 1

Jakob Valič

Ljubljana, september 2018

Vsebina

Uvod

Zahvala

Zahvaljujem se Marini Kovač, s katero sva skupaj pripravljala gradivo za predstavitev algoritmov za vzorčenje točk v ravnini in sodelavcu Tadeju, ki mi je pomagal pri obdelavi barvnih slik.

Problem pomanjšave slike

Motivacijski problem je manjšanje slike. Slika je sestavljena iz pikslov. »Piksel predstavlja najmanjšo naslovljivo točko, ki se jo lahko prebere ali nariše.« (vir 1) Piksel nosi informacije o barvni shemi točke, ki jo določa. Barvne slike imajo piksle predstavljene s seznamom [*rdeča, zelena, modra*], kjer je vsaka od naštetih barv prestavljena s številom od 0 do 255. Piksel [0, 0, 255] tako predstavlja modro točko, [0, 0, 0] črno in [255, 255, 255] belo točko.

Imamo torej sliko, sestavljeno iz pikslov. To sliko želimo pomanjšati. Drugače povedano želimo dobiti sliko z manj piksli. Več pikslov bomo združili v en piksel, odvisno od velikost povečave. Lahko bi za pomanjšanje slike uporabili vse piksle, ki se nahajajo v originalni sliki. Tako dobimo najbolj natančno pomanjšavo slike. Vendar želimo problem pomanjšave povezati z vzorčenjem. V ta name bomo vzeli samo določen delež prvotnih pikslov. Za namen te seminarske naloge se bomo piksle poimenovali kar 'točke'.

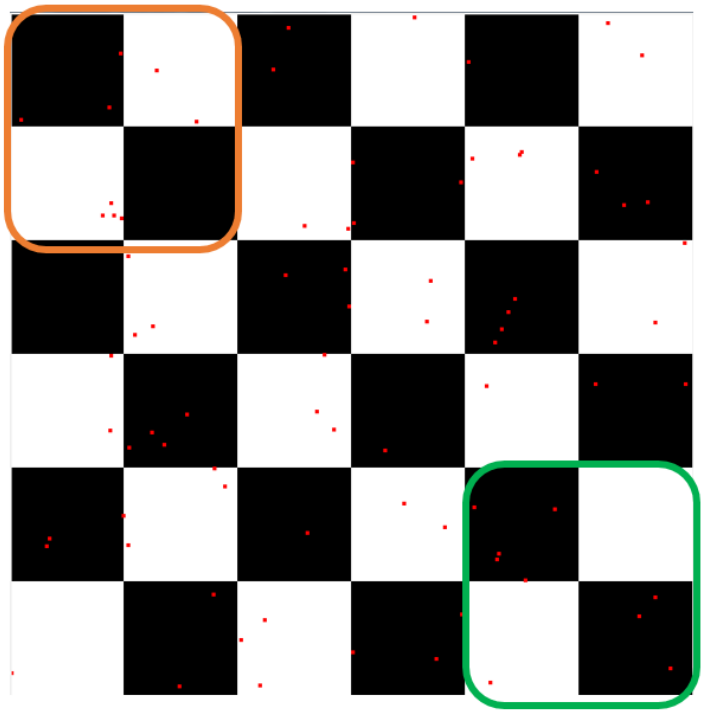
Vzorčenje

Vzorčenje je izbira določenega števila primerkov iz nabora. Rezultat vzorčenja je vzorec. Pri vzorčenju imamo vedno podano velikost vzorca in način, na katerega ta vzorec dobimo. Sedaj si bomo pogledali nekaj načinov vzorčenja: naključno, mrežno in kombinacijo obeh. Ponazorili jih bomo s preprostimi primeri vzorčenja na šahovnici.

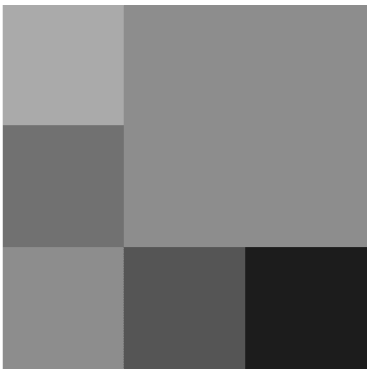
Naključno vzorčenje

Pri naključnem vzorčenju naključno izberemo določeno število točk iz nabora. Zaradi naključnega izbora se nam lahko pojavi problem, da so nekje delilne točke postavljene preveč na gosto, drugje pa preveč na redko. Poglejmo si primer šahovnice dimenzije 6x6. Želimo jo nadomestiti s šahovnico dimenzije 3x3. To dosežemo tako, da naključno izberemo 80 točk. Nato na področju, ki ga določajo kvadrati velikost 2x2 pogledamo vse izbrane točke in določimo njihov povprečen odtenek. Poglejmo si oranžno označeno polje. V njem je 9 izbranih točk, od tega 6 iz belega področja in 3 iz črnega področja. Posledično bo to polje obarvano svetleje od pričakovanega povprečja. Obratno velja za zeleno označeno območje, kjer se prav

tako nahaja 9 izbranih točk, od katerih je samo ena na belem področju, vse ostale so na črnem. Na levi strani imamo pomanjšano šahovnico, ki potrjuje naša opažanja.



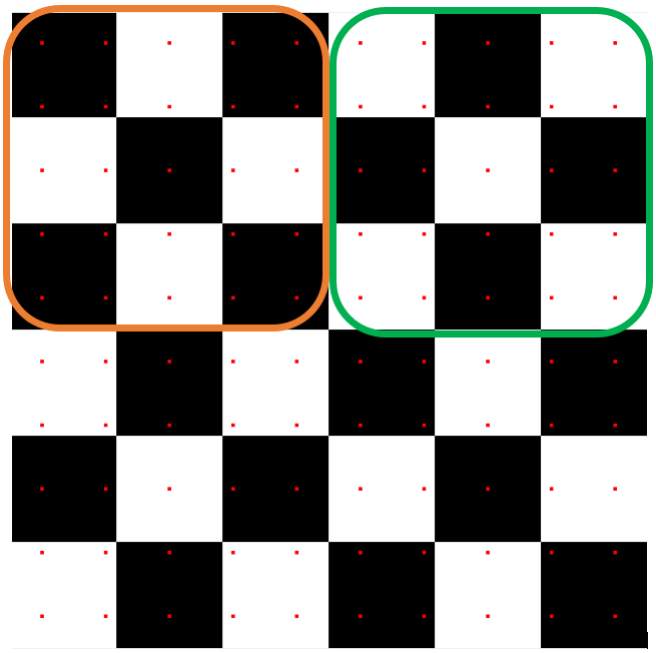
Slika 1: Naključno vzorčenje



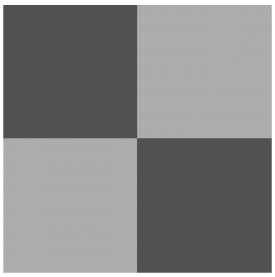
Slika 2: Naključno vzorčenje - pomanjšana

Mrežno vzorčenje

Ideja mrežnega vzorčenja je, da vzamemo točke, ki so enako oddaljene od sosednjih točk. Izbrane točke na koncu izgledajo kot vozlišča mreže. Velikost mreže oziroma razmak med sosednjimi točkami je določen z velikostjo slike in velikostjo vzorca. V sledečem primeru imamo 100 vzorčnih točk. Osnovna šahovnica je dimenzije 6x6. Želimo jo pomanjšati trikrat, torej na šahovnico dimenzije 2x2. Oglejmo si oranžen kvadrat. V njem nastopa 25 vzorčnih točk, 17 od teh je na črnih poljih in 8 na belih. Nadomestili ga bomo s poljem, ki je temnejše od povprečja. V zelenem kvadratu je situacija ravno obratna. S pomočjo mrežnega vzorčenja na šahovnici bomo vedno dobili simetričen vzorec. Dobra lastnost vzorčenja je, da je ponovljivo. Zmeraj bomo dobili enak vzorec.

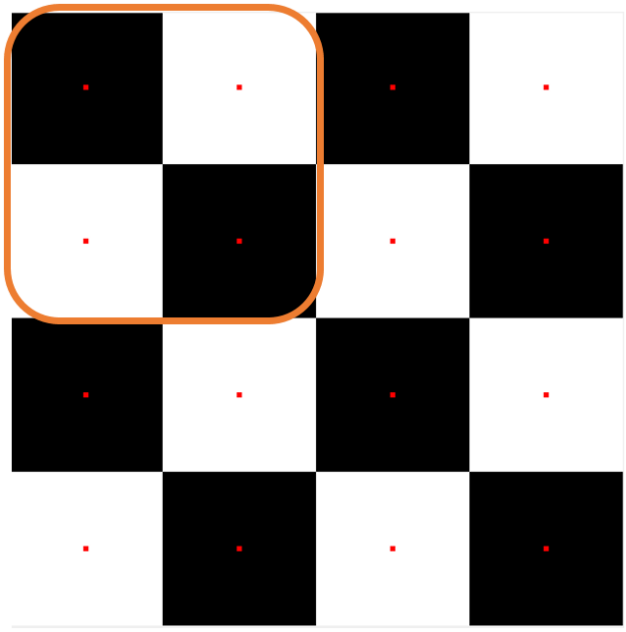


Slika 3: Mrežno vzorčenje



Slika 4: Mrežno vzorčenje - pomanjšana

Problem nastopi v situaciji, ko je v določenem polju mrežnega vzorčenja enako število točk, ki ležijo na belih in črnih poljih. V tem primeru dobimo siv vzorec, s katerega struktura šahovnice sploh več ni vidna. To se zgodi v primeru šahovnice velikost 4x4, ko imamo 16 delilnih točk in jo želimo pomanjšati na velikost 2x2. V vsakem vzorčnem polju sta po 2 točki, ki sta na belemu polju in 2, ki sta na črnem polju. Rezultat je siva slika.



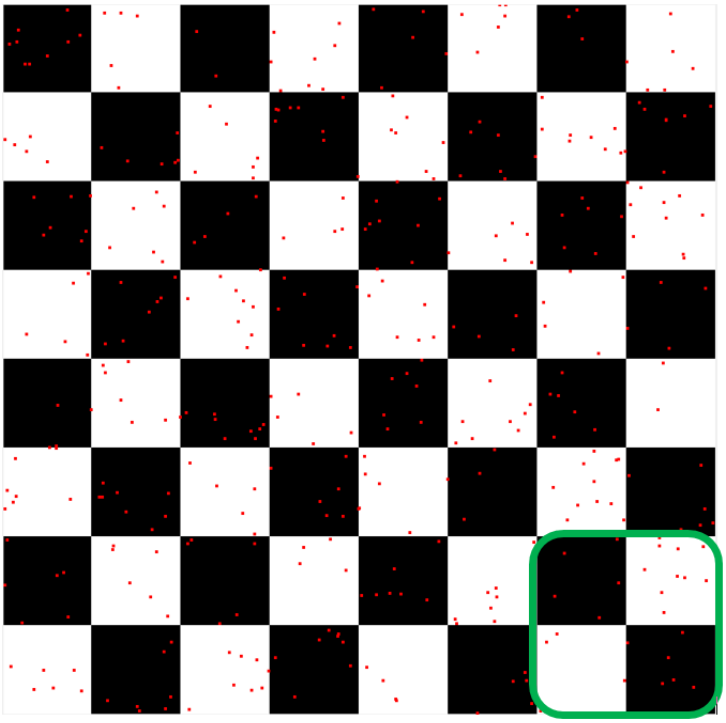
Slika 5: Mrežno vzorčenje - enaka razporeditev

Slika 6: Mrežno vzorčenje - sivina

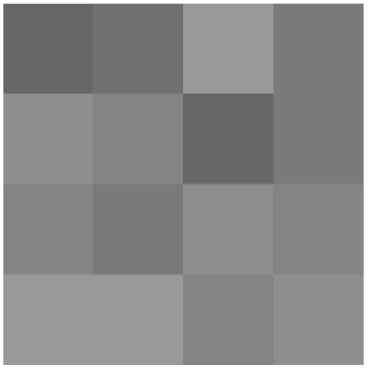
Kombinacija mrežnega in naključnega vzorčenja

Pri tem vzorčenju z mrežo razdelimo sliko. Glede na velikost vzorca določimo, koliko delilnih točk bo v vsakem polju mreže. V našem primeru imamo šahovnico velikosti 8x8. Želimo jo pomanjšati na šahovnico velikost 4x4. Temu primerno bomo mrežo določili tako, da bodo polja

mreže zavzemala področja šahovnice velikosti 2x2. Takih polj mreže je 8. Na voljo imamo 400 vzorčnih točk. Izračunamo, da vsakemu polju mreže pripada 25 vzorčnih točk. Znotraj tega področja jih razporedimo naključno. Na sliki je označeno eno polje mreže. Dobimo dokaj enakomerno porazdeljeno pomanjšano sliko. Vseeno je moč zaslediti obliko šahovnice.



Slika 7: Kombinacija mrežnega in naključnega



Slika 8: Slika kombinacije

V angleščini se to vzorčenje imenuje *jittered*.

Metoda je dobra, saj nam zagotavlja, da bodo glede na polja mreže točke razporejene enakomerno. To pa ne velja za točke znotraj posameznega polja. Zato si bomo ogledali še dva boljša algoritma, in sicer Bridsonov in Mitchellov algoritem.

Bridsonov algoritem

Ta algoritem je razvil Robert Bridson in ga leta 2006 objavil v članku z naslovom *Fast Poisson Disk Sampling in Arbitrary Dimensions*. (glej vir 2) Algoritem temelji na Poissonovi porazdelitvi. Deluje tudi v več dimenzijah, vendar se bomo osredotočili le na vzorčenje v ravnini.

Opis algoritma

Vhodni argumenti:

- *n* - dimenzija prostora, v katerem bomo vzorčenje izvajali (v našem primeru 2) ,
- *r* - minimalna razdaljo med dvema točkama,
- *k* - število kandidatov za naslednji vzorec

Potek algoritma:

- Korak 0: Določimo pomožno mrežo enakih dimenzij kot prostor vzorčenja s polji velikosti $r/\sqrt{n}$, kjer je *n* dimenzija prostora. Tako določena polja bodo vsebovala največ eno vzorčno točko. Z 1 bom označili prisotnost vzorčne točke, z 0 pa da vzorčne točke ni. Pomožna mreža nam olajša iskanje razdalje med točkami.
- Korak 1: Naključno izberemo začetno točko in ga vstavimo v pomožno mrežo. Ustvarimo seznam aktivnih točk in vanj vstavimo začetno točko.
- Korak 2: Iz seznama aktivnih točk naključno izberemo točko, recimo ji x_i. V radiju [*r*, 2*r*] generiramo do *k* kandidatov za naslednjo aktivno točko. Prvega kandidata, ki ustreza pogoju, da je vsaj za *r* oddaljen od vseh ostalih že izbranih (aktivnih in neaktivnih) točk, dodamo v seznam aktivnih točk in ponovimo korak 2. Če je vseh *k* kandidatov preblizu že izbranih točk, točko x_i odstranimo iz seznama aktivnih točk. Ponavljamo korak 2, dokler ni seznam aktivnih točk prazen.

Izhodni podatki:

- Pomožna mreža z vzorci.

Algoritem ima linearno časovno zahtevnost.

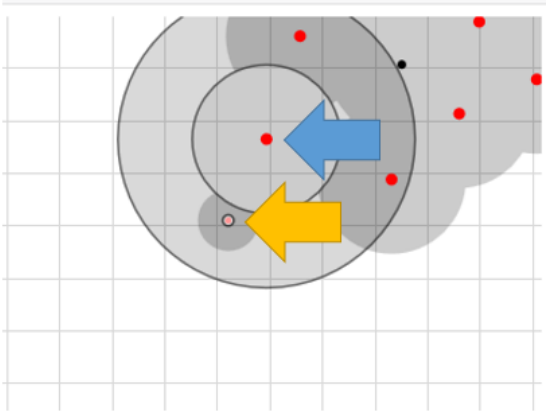
Predstavitev algoritma na primerih:

Vizualizacija algoritma se nahaja na spletni strani Mika Bostocks-a. (glej vir 3) Aktivne točke so označene z rdečo barvo, neaktivne pa s črno. S svetlo sivo barvo je označen krog s polmerom *r*, ki označuje območje okrog vsake točke. Ker morajo biti točke vsaj za *r* narazen, v tem območju ne sme biti druge točke. S temno sivo barvo je označen kolobar na območju [*r*, 2*r*], iz katerega izbiramo nove kandidate za aktivne točke. Z belo so označeni kandidati za nove aktivne točke. Poglejmo si dve ključni situaciji, do katerih pride tekom izvajanja algoritma.

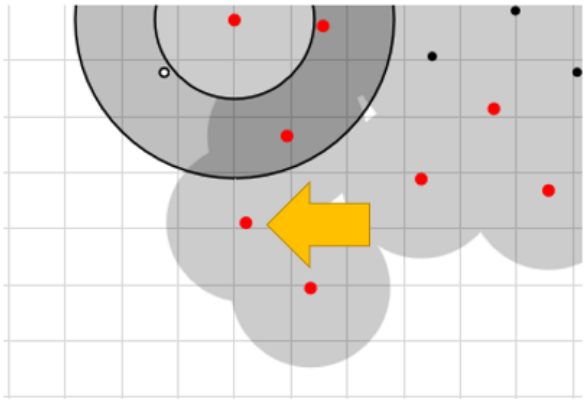
Dodajanje nove aktivne točke

Novo aktivno točko dodamo tedaj, ko je prvi kandidat za novo točko dovolj oddaljen od ostalih aktivnih in neaktivnih točk. Grafično to vidimo tako, da kandidata postavimo na belo polje. Na levi sliki vidimo z modro puščico označeno aktivno točko, okrog katere izbiramo kandidate. Z rumeno puščico je označen kandidat, ki je dovolj oddaljen od vseh ostalih točk. Zato ga dodamo v seznam aktivnih točk in nadaljujemo s korakom 2. Pri tem z modro puščico označene

točke ne odstranimo iz seznama aktivnih točk, saj je v njeni okolici morda še prostor za nove aktivne točke.



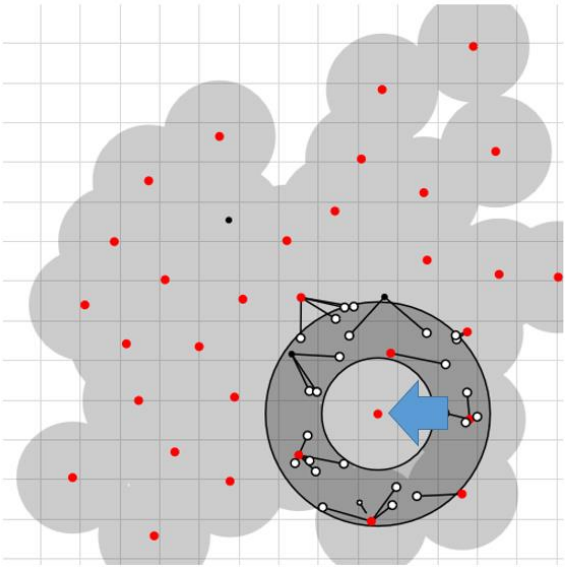
Slika 9: Bridson - dodajanje točke



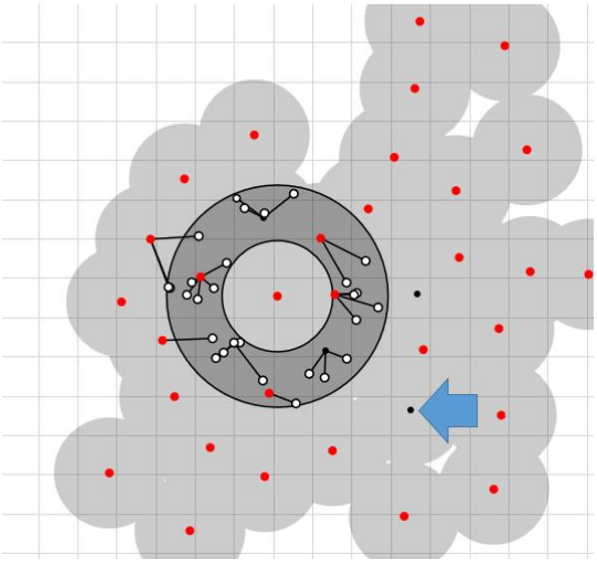
Slika 10: Bridson - dodana točka

Odstranitev točke iz seznama aktivnih točk

Na levi sliki imamo z modro puščico označeno naključno izbrano točko iz seznama rdeče obarvanih aktivnih točk. Okrog nje poskušamo v temnejšem kolobarju najti novo aktivno točko. Kandidati za nove točke so belo pobarvani. Problem je, da ležijo preblizu že obstoječih aktivnih in neaktivnih točk. Z eno izmed točk, ki ji ležijo preblizu, so povezani z daljico. Ker se noben kandidat ne nahaja na prostem območju, točka postane neaktivna (črno obarvana), kar prikazuje slika na desni. Algoritem je medtem že izbral naslednjo aktivno točko, za katero lahko prav tako napovemo, da jo bo odstranil iz seznama aktivnih točk.



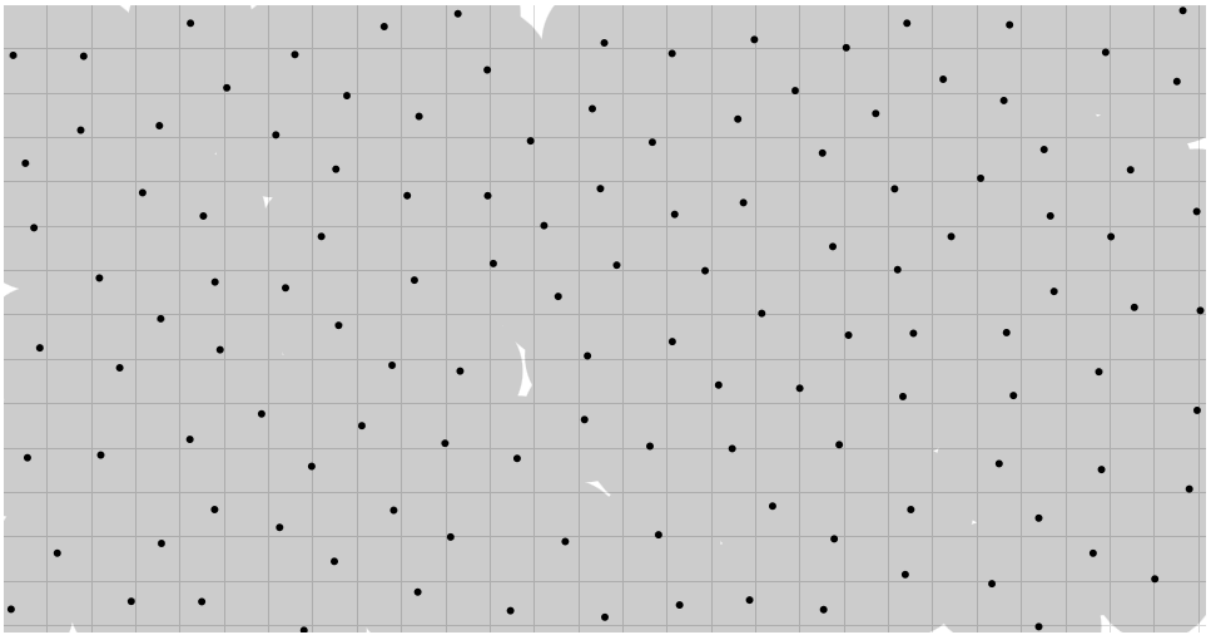
Slika 11: Bridson - ne najdemo novih kandidatov



Slika 12: Bridson - točka postane neaktivna

Konec algoritma

Algoritem se konča, ko ni več aktivnih točk. Vidimo, da ostaja še nekaj manjših področij, obarvanih belo, kjer bi lahko dodali še kakšno točko. Vendar točke tja algoritem ni vstavil, saj je kandidate razporejal naključno.



Slika 13: Konec Bridsonovega algoritma

Mitchellov algoritem

Mitchellov algoritem je objavljen na spletni strani Mika Bostocksa. (glej vir 6) Tudi ta algoritem je različica Poissonovega vzorčenja.

Vhodni podatki:

- *k* – število kandidatov
- *N* – velikost vzorca

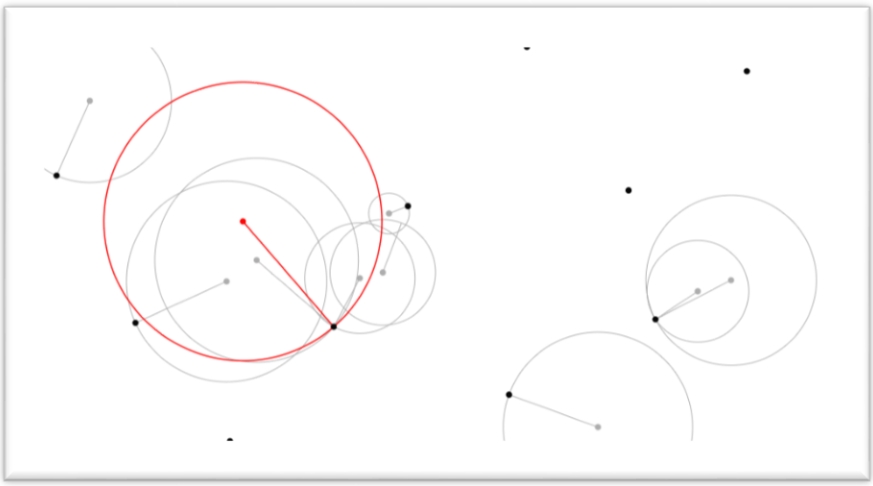
Potek:

- Korak 1: naključno izberemo začetno točko. Ustvarimo seznam izbranih točk in vanj vstavimo začetno točko.
- Korak 2: naključno izberemo *k* kandidatov. Za vsakega kandidata izračunamo oddaljenost do najbližje že postavljene točke, označimo jo z *d*. V seznam izbranih točk vstavimo kandidata z najmanjšo razdaljo *d*. Ponavljamo korak 2, dokler v seznamu izbranih točk ni N točk.

Časovna zahtevnost algoritma je kvadratična zaradi vsakokratnega iskanja najbližje že postavljene točke.

Prikaz delovanja algoritma

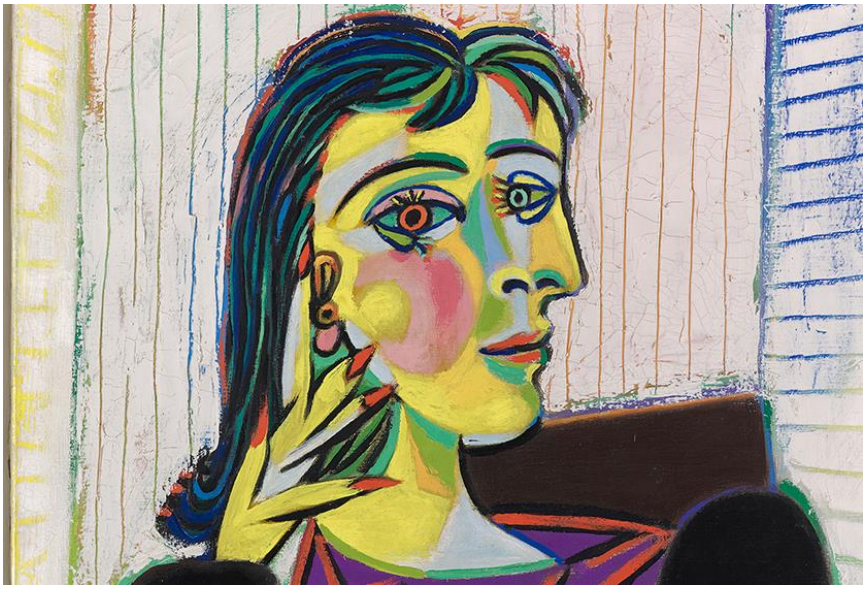
S črno piko so označene točke v seznamu izbranih točk. S sivo so označeni kandidati. Na spodnji sliki vidimo z rdečo krožnico in daljico označenega kandidata, ki je izmed vseh kandidatov najbolj oddaljen od njemu najbližje točke iz seznama izbranih točk, ima najmanjšo razdaljo *d*.



Slika 14: Mitchelle, najbolj oddaljen kandidat

Primer obdelave barvnih slik

Vzemimo na primer sliko Picassa. Original slika izgleda takole:

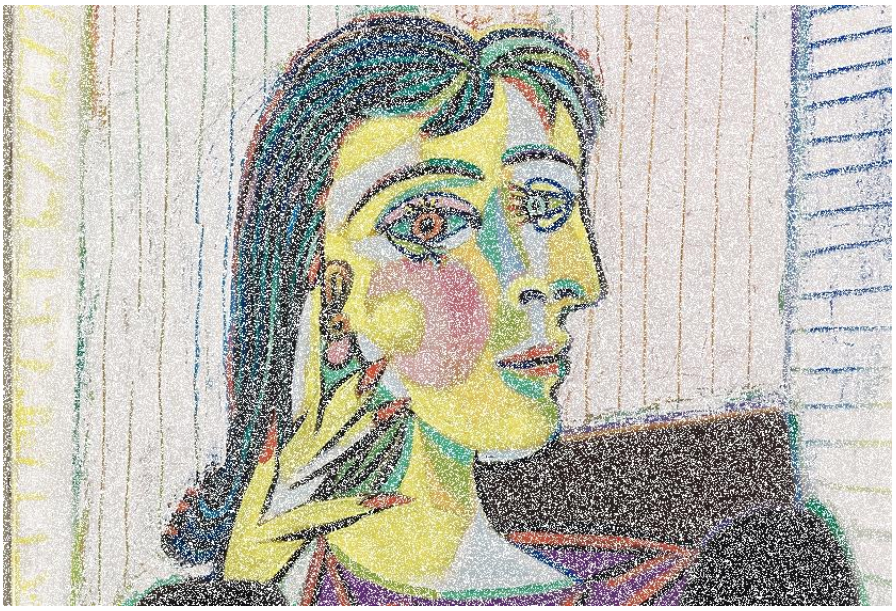


Slika 15: Picasso - original

Sedaj želimo sliko pomanjšati 8-krat. Na njej poženemo algoritem za naključno izbiro vzorčnih točk. Izvedli bomo poskus s 70% vseh točk in nato še s 5% vseh točk. Vse ostale točke bomo obarvali belo.

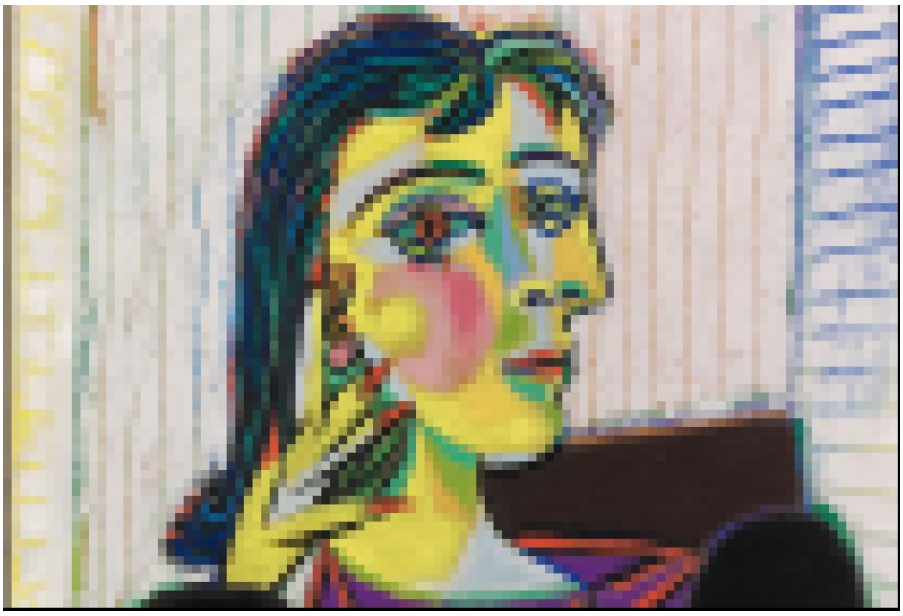
70% točk

Iz izbranih vzorčnih točk je jasno razvidna originalna slika.



Slika 16: Picasso 70%, vzorčne točke

Sedaj poženemo algoritem za združevanje območij. Združevali bomo območja velikosti 8x8. Prikažemo enako veliko sliko, ki je sedaj granulirana in zmanjšano sliko.

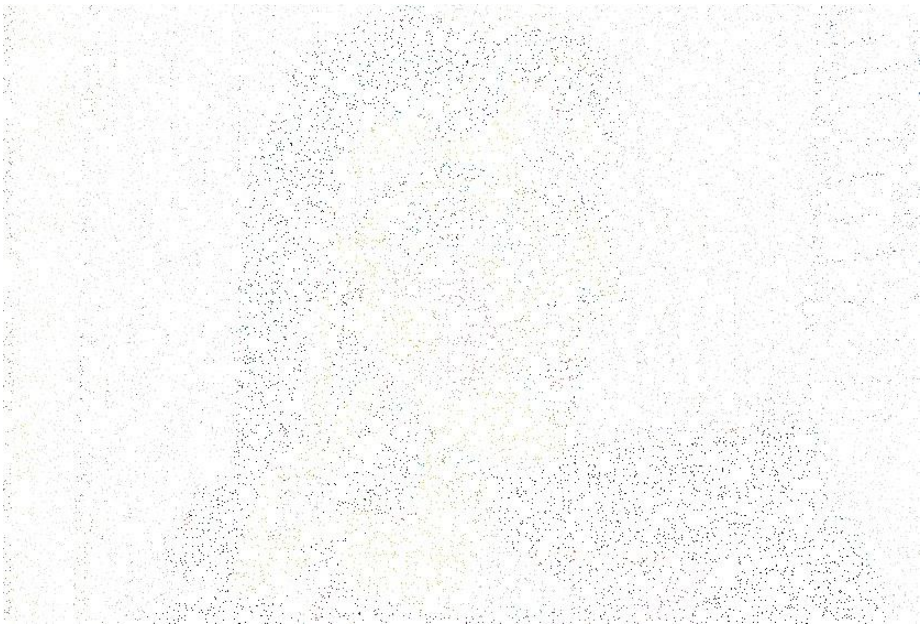


Slika 17: Picasso 70%, enako velika

Slika 18: Picasso 70%, pomanjšana 8-krat

5% točk

Sedaj poskus ponovimo s tem, da izberemo le 5% vseh točk na originalni sliki. Iz vzorčnih točk komaj da razločimo obris prvotne slike.



Slika 19: Picasso 5%, vzorčne točke

Ko točke združimo po območjih velikosti 8x8, jasneje vidimo podobnost z originalom. Moti nas to, da se v sliki pojavljajo beli kvadratki. To so področja, na katerih ni bil izbran niti en piksel, zato so ostala bela.

Slika 20: Picasso 5%, enako velika

Slika 21: Picasso 5%, pomanjšana 8-krat

Primerjava časovne zahtevnosti

Ni bistvene razlike v časovni zahtevnosti za izbiro 5% ali 70% vseh točk. To je bilo tudi pričakovati.

```
Čas za izbiro naključnih 70% pikslov: 1.4s
Čas za izračun pomanjšane slike: 11.3 s
Čas za izbiro naključnih 5% pikslov: 1.9s
Čas za izračun pomanjšane slike: 9.0 s
```

Slika 22: Časovna zahtevnost

Možne izboljšave

- Odstranitev belih področij na barvni sliki.
- Analiza barvnih slik tudi z grid in jittered vzorčenjem in tudi z Bridsonovim algoritmom.

Viri

1. https://sl.wikipedia.org/wiki/Piksel
2. Članek Roberta Bridsona: Fast Poisson Disk Sampling in Arbitrary Dimensions, 2006 https://www.cct.lsu.edu/~fharhad/ganbatte/siggraph2007/CD2/content/sketches/0250.pdf
3. Vizualizacija Bridsonovega algoritma: https://bl.ocks.org/mbostock/dbb02448b0f93e4c82c3
4. Picassova slika: http://www.qm.org.qa/sites/default/files/picasso-giacometti-exh-qm-web-01.jpg
5. Različne vrste vzorčenja: https://www.everything2.com/index.pl?node_id=1028947
6. Mitchellov algoritem: https://bl.ocks.org/mbostock/d7bf3bd67d00ed79695b