

PROJEKTA NALOGA

VPSA

2021/2022

Marcel Ujčíč

Serijski algoritem

Implementacija

Serijski arlogiretm sem implementiraj v jeziku C. Pri implementaciji sem si pomagal s knjižnico Freeimage. Freeimage je odprto kodna knjižnica, ki je namenjena za obdelavo slik v formatih PNG, BMGP, JPEG itd..

Algoritem glavne zanke

Algoritem glavne zanke je sestavljen iz naslednjih korakov:

1. Branje slike v rezerviran pomnilnik
2. Incializacija centroidov
3. Dokler niso vse interacije obdelane, pojdi skozi celotno velikost slike in poišči najmanjšo razdaljo za vsako slikovno točko.
4. Potem, ko imajo vse slikovne točke razdalje, izračunaj povprečje za vsak centroid.
5. Ko so vse interacije opravljene, zapiši nove indekse centroidov v sliko.

Algoritem OpenMP

Za implementacijo algoritma sem uporabil knjižnico OpenMP. Kot osnovo na kateri sem implementiral sem uporabil sekvenčni algoritem.

Na začetku programa sem dodal preverjanje podanih ar gumentov, ki so število niti in število gruč. Po tem sem dodal klic, ki vzame argument in nastavi število niti, če v argumentu ni število niti je prevzeta vrednost 16 niti.

V glavnem porgrami, sem dodal klic `#pragma omp prallerl for schedule (dynamic,1)`, ki paralizira for zanko. To sem ponovil dvakrat v glavni zanki, pri iskanju najmanjše evklidove razdalje in pri računanju povprečja za vsako gručo.

Algoritem OpenCL

Pri implementaciji, sem uporabil knjižnico OpenCL, ki je odprt standard za razvoj programske opreme, ki teče na raznoliki strojni opremi. V tem primeru so to grafične procesne enote.

Za osnovo sem uporabil sekvenčni algoritem, ki sem ga razdelil na dva dela.

Koraki algoritma

1. V prvem koraku se inicializirajo okolje in centriodi ter izbere napravo (GPU) na kateri se bo izvajal algoritem. Nato se na izbrano napravo pripravi kontekst in ukazna vrsta. Iz posebne datoteke kernel.cl se naloži programska koda, ki bo tekla na napravi in se zanjo ustrezno prevede.
2. V drugem koraku se na GPU prenese slika, katero želimo zmanjšati. Hkrati se rezervira prostor tudi za vmesne rezultate in se prenese na GPU tudi polje s centriodi.
3. Na to se v glavni zanki, N krat zažene najprej kernel za izračun razdalij na to še kernel za izračun povprečja.
4. Kernel za računanje razdalj se izvaja bolj ali manj enako kakor sekvenčna verzija, le da je tu glavna zanka zamenjana s globalnimi niti.
5. Kernel za računanje povprečja je tudi zgrajen na sekvenčni verziji, in je glavna for zanka zamenjana s nitmi.
6. Ko se izvedejo vse interacije se iz GPE prenesejo centriodi in se še določijo oz. Zapišejo nove barve v izhodno sliko.
7. Počisti vse rezervirana polja, program, čakalna vrsta.

Rezultati

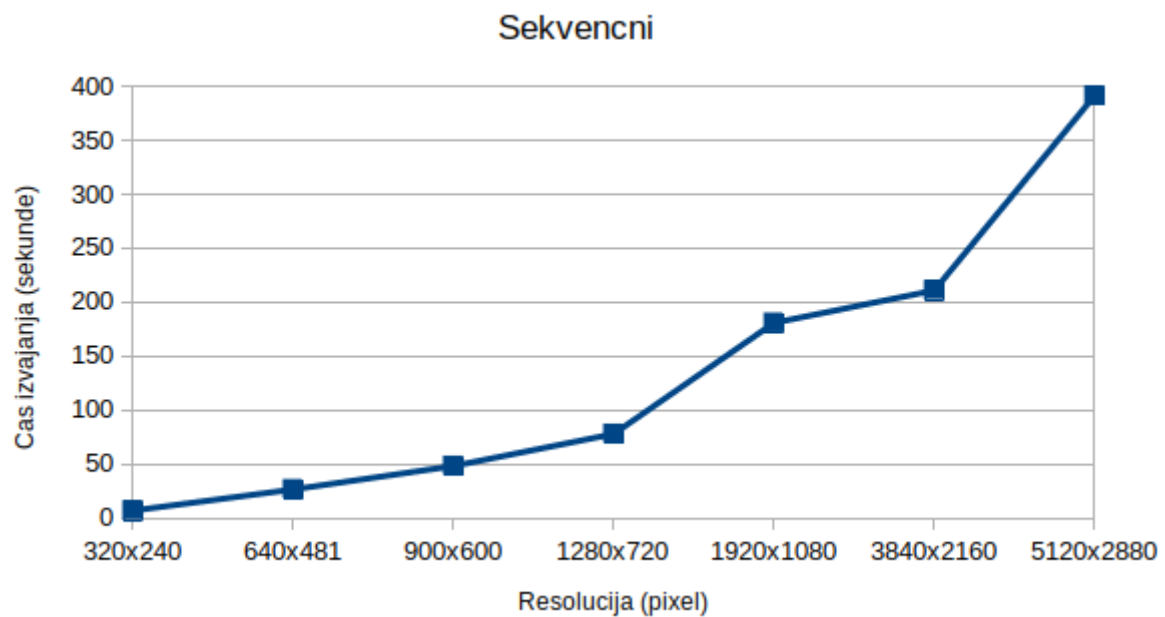
Vse tri algoritme sem pognal na slikah različnih velikosti od 320x240 do 5120x2880 slikovnih točk. Poleg spreminjanja same resolucije slik sem spreminjal tudi število barv od 8 do 128. Ter pri algoritmu openMP spreminjal število niti.

Sekvenčna verzija

Pričakovano je sekvenčna verzija najpočasnejša izmed vseh treh.

Prikaz časovne tahtevnosti sekvenčnega algoritma na 64barvah, 100 iteracij.

Velikost	Čas izvajanja (s)
320x240	7.1
640x481	26.7
900x600	48.4
1280x720	78.03
1920x1080	180.9
3840x2160	211.02
5120x2880	391.28



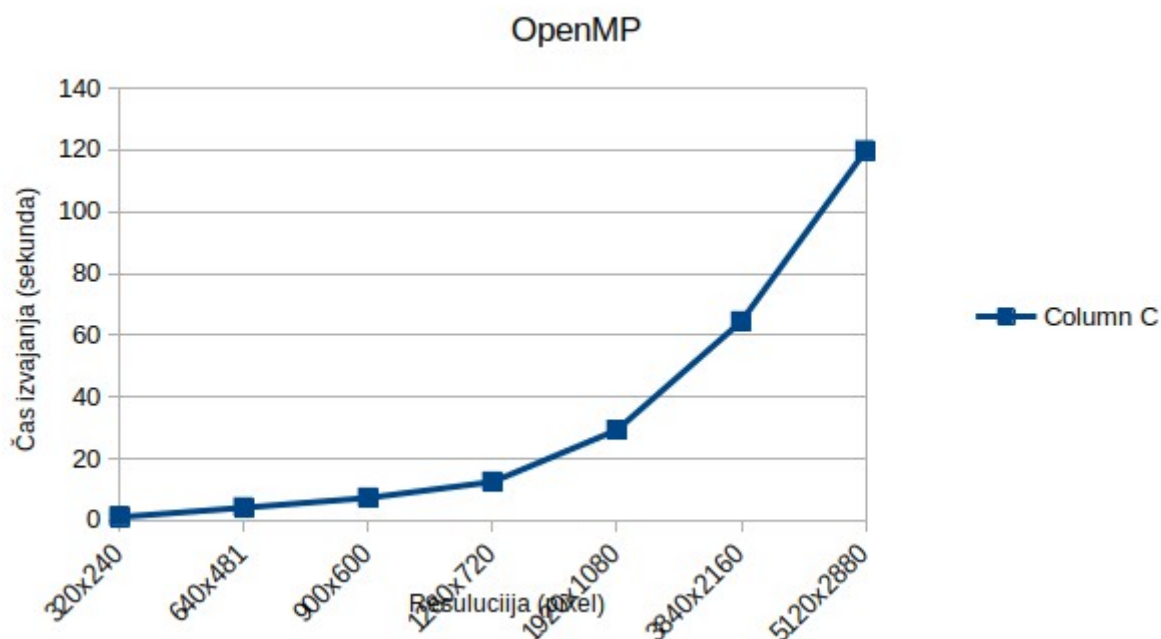
Slika 1: Prikaz časov izvajanja pri sekvenčnem algoritmu za različne resolucije slik

Algoritem OpenMP

Algoritem OpenMP je bil v povprečju za 5.6x hitrejši od sekvenčne verzije.

Prikaz tabele za algoritem openMP na 64 barvah, 100 interacij in 16niti.

Velikost	Čas izvajanja (s)
320x240	1.03
640x481	4.07
900x600	7.25
1280x720	12.46
1920x1080	29.25
3840x2160	64.42
5120x2880	119.76



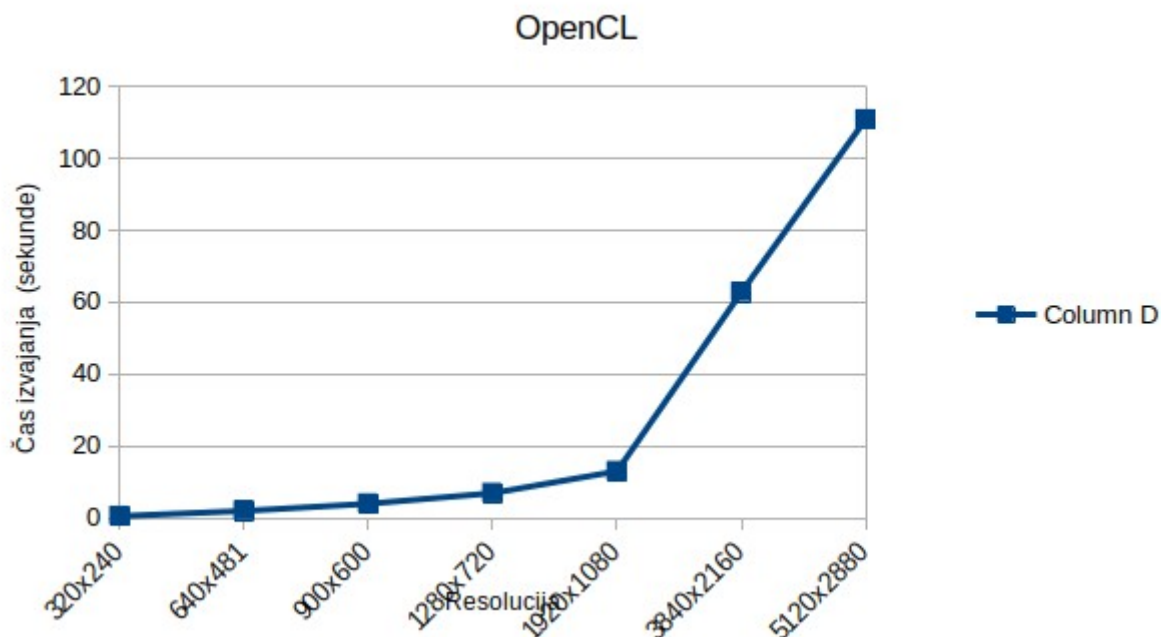
Slika 2: Prikaz časov izvajanja za openMP glede na različne resolucije slike

Algoritem OpenCL

Zadnji algoritem je bil pričakovano najhitrejši izmed vseh. Saj je glede na sekvenčni algoritem kar 9.8x v povprečju hitrejši.

Prikaz tabele in grafikona s časi za algoritem openCL 64 barv, 100 iteracij

Velikost	Čas izvajanja (s)
320x240	0.6
640x481	2.04
900x600	4.03
1280x720	6.98
1920x1080	13.07
3840x2160	62.76
5120x2880	110.77



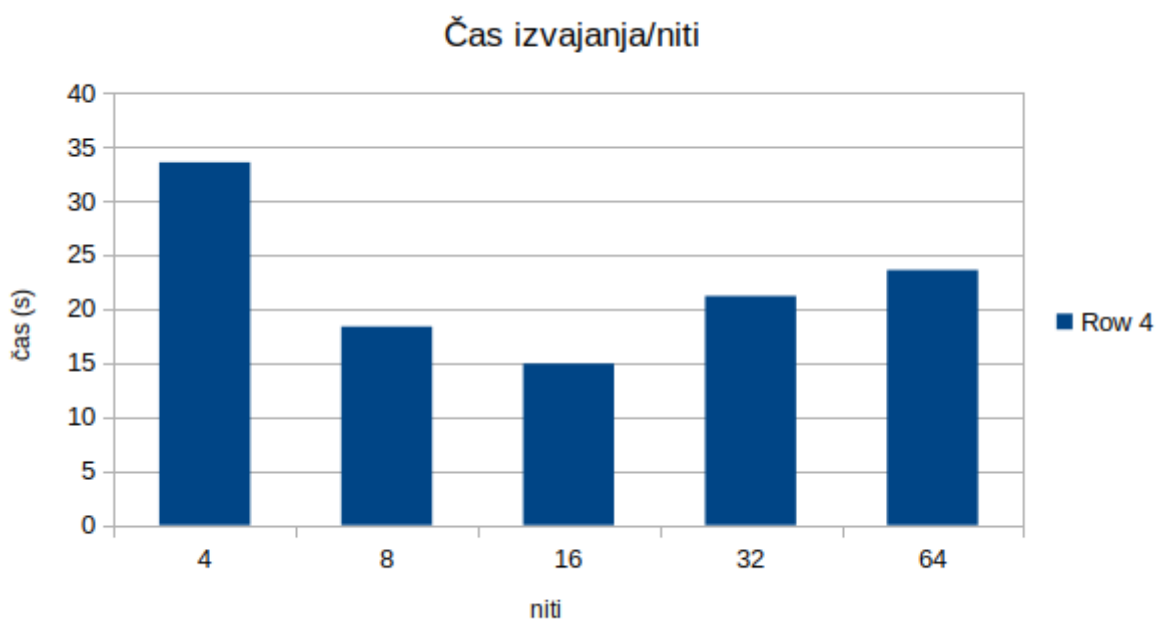
Slika 3: prikaz časa izvajanja za algoritem openCL za različne resolucije

Primerjave

Časovna zahtevnost algoritma openMP na sliki 1920x1080 in na nitih od 4 do 64.

Prikaz časa izvajanja (sekunde), glede na niti.

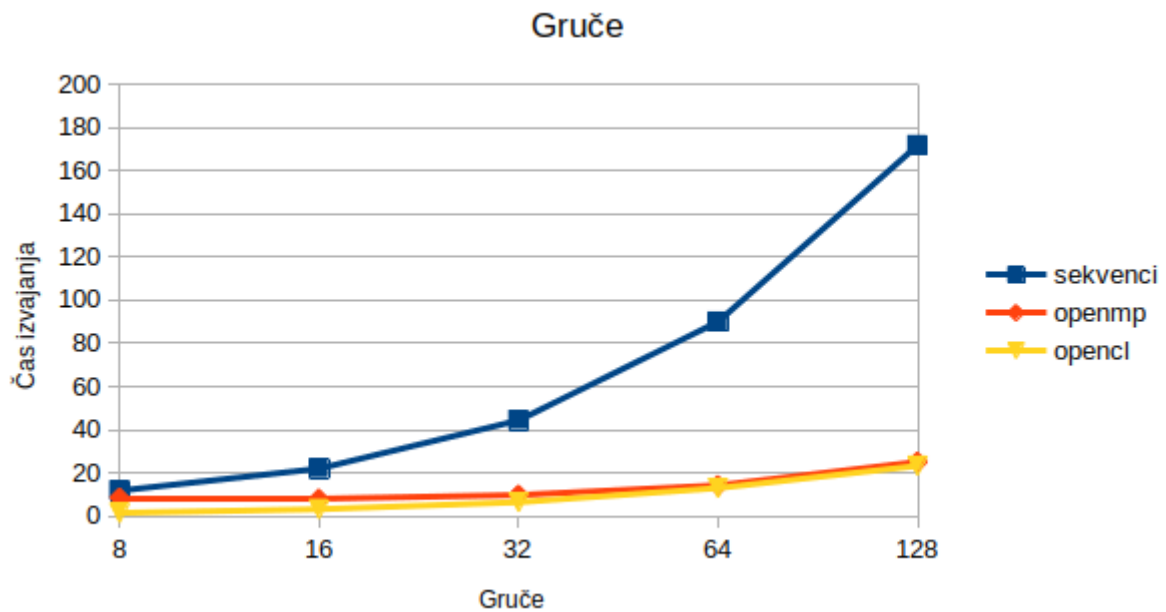
Velikost/niti	4	8	16	32	64
1920x1080	33.54	18.36	14.94	21.19	23.59



Slika 4: Čas izvajanja openMP slika 1920x1080, 64 barv, različno niti

Časovna primerjava (sekunde) vseh treh algoritmov na sliki 1920x1080, število niti 16 openMP, različno število barv od 8 do 128.

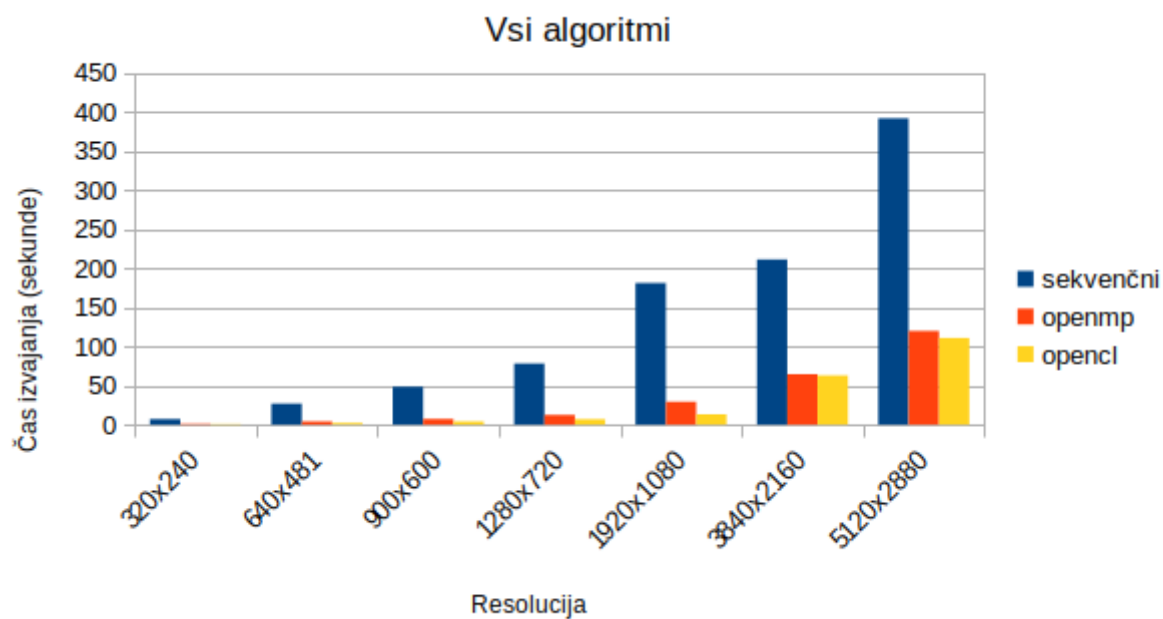
Algoritm/barve	8	16	32	64	128
Sekvenčni	11.89	22.06	44.39	90.05	171.74
OpenMP	8.54	7.98	9.84	14.31	25.17
OpenCL	1.61	3.26	6.53	13.07	23.34



Slika 5: primerjava vseh treh algoritmov glede na število barv

Časi izvajanja vseh treh algoritmov po različnih resolucijah, vsi 64barv in 100 interacij.

Velikost	Sek (s)	openMP (s)	openCL (s)
320x240	7.1	1.03	0.601855
640x481	26.7	4.07	2.036505
900x600	48.4	7.25	4.032657
1280x720	78.03	12.46	6.978918
1920x1080	180.9	29.25	13.071947
3840x2160	211.02	64.42	62.760445
5120x2880	391.28	119.76	110.769983



Slika 6: Časovna zahtevnost vseh treh algoritmov

Priloge

Primerjava slike original, ter po poganjanju skozi sekvenčni, OpenMP in OpenCL algoritme.

Velikost 1280x720, 64 barv po kompresiji.



Slika 7: Original 1280x720



Slika 8: Po kompresiji 64barv



Slika 9: Po kompresiji 8barv