DETEKCIJA IVICA UNUTAR SLIKE

PARALELNO PROGRAMIRANJE

Katarina Vučić SV29/2020

SADRŽAJ

[ANALIZA PROBLEMA 4](#_Toc106140061)

[KONCEPT RJEŠENJA 4](#_Toc106140062)

[DETEKCIJA PREWITT OPERATOROM 4](#_Toc106140063)

[DETEKCIJA IVICA U OKOLINI PIKSELA 5](#_Toc106140064)

[PROGRAMSKO RJEŠENJE 5](#_Toc106140065)

[DETEKCIJA PREWITT OPERATOROM 5](#_Toc106140066)

[DETEKCIJA IVICA U OKOLINI PIKSELA 6](#_Toc106140067)

[PRIMJERI ULAZNIH I IZLAZNIH SLIKA 7](#_Toc106140068)

[ISPITIVANJE 12](#_Toc106140069)

[TABELARNI PRIKAZ SERIJSKOG IZVŠENJA RAZLIČITIH DIMENZIJA SLIKE 12](#_Toc106140070)

[TABELARNI I GRAFIČKI PRIKAZI PARALELNOG IZVRŠENJA RAZLIČITIH DIMENZIJA PROBLEMA 12](#_Toc106140071)

[ANALIZA REZULTATA 15](#_Toc106140072)

# ANALIZA PROBLEMA

Detekcija ivica uključuje niz matematičkih metoda čiji je cilj identifikacija ivica na digitalnoj slici. Ona predstavlja osnovni alat u obradi slike. Možemo koristiti različite pristupe detekcije ivica, kao što su detekcija prewitt operatorom ili detekcija sa najužom okolinom. Svaki od pristupa daje različite izlaze za iste ulaze (slike).

# KONCEPT RJEŠENJA

Korištena su dva pristupa rješenju problema: detekcija sa Prewitt operatorom i detekcija ivica u okolini piksela.

## DETEKCIJA PREWITT OPERATOROM

Ulazna slika se filtrira operatorima za horizontalne i vertikalne ivice. Operatori ovog pristupa su zapravo kvadratne matrice, najčešće dimenzija 3x3 i rjeđe 5x5 piksela. Trenutni piksel (onaj koji ima potpunu okolinu) ulazne slike se preklapa sa centralnim pikselom operatora. Korespondentni pikseli se množe sa operatorima i nakon toga se sabiraju svi proizvodi, a rezultujuća vrijednost se dobija zbirom apsolutnih vrijednosti horizontalne i vertikalne komponente. Nakon toga se u zavisnosti od definisanog praga piksel boji u bijelu ili crnu boju. Nakon što proces prođe kroz sve piksele, dobija se crna slika sa jasno izraženim bijelim ivicama.

## DETEKCIJA IVICA U OKOLINI PIKSELA

Ovaj pristup zahtijeva postavljanje komponenata na vrijednosti 0 ili 1 odsijecanjem za određeni prag. Nakon toga, rezultat će biti crno-bijela slika bez izdvojenih ivica. Sljedeći korak je pretraga okoline svakog piksela (osim onih koji nemaju potpunu okolinu). Pod okolinom se smatraju susjedne tačke – iznad, ispod, lijevo, desno i obje dijagonale. Zatim se vrši postavljanje operatora P i O na vrijednosti 0 ili 1 u zavisnosti od te okoline. P operator se postavlja po sljedećem kriterijumu: ukoliko u okolini tačke postoji piksel sa vrijednošću 1, P će imati vrijednost 1, u suprotnom 0. Ukoliko u okolini tačke postoji tačka sa vrijednošću 0, vrijednost operatora O će biti 0, u suprotnom će biti 1. Vrijednost rezultujuće tačke se dobija razlikom apsolutnih vrijednosti operatora P i O i vraćanjem u opseg 255 ili 0 (na crnu ili bijelu boju). Nakon što program prođe kroz svaki piksel, dobija se crna slika sa jasno izdraženim bijelim ivicama.

# PROGRAMSKO RJEŠENJE

Korišćena je biblioteka EasyBMP. Zahvaljujući njoj, slike u BMP formatu su pretvorene u matrice, čije su vrijednosti između 0 i 255.

## DETEKCIJA PREWITT OPERATOROM

Serijska detekcija ivica Prewitt operatorom je implementirana u funkciji filter\_serial\_prewitt. Ukratko opisano, ova funkcija vrši prolazak kroz svaki element matrice (ekvivalentno pikselu slike) i u zavisnosti od pozicije tačke (ako nije jedna od tačaka koja nema odgovarajuću okolinu za zadati filter – ivična tačka), vrši se množenje sa horizontalnim i vertikalnim filterima (u kodu filterHor i filterVer). Kako se vrši množenje i prebacivanje piksela na bijelu, tj. crnu boju, je detaljno opisano u poglavlju KONCEPT RJEŠENJA – DETEKCIJA PREWITT OPERATOROM.

Paralelizovana detekcija ivica slike Prewitt operatorom je smještena u funkciji filter\_parallel\_prewitt. Za razliku od serijskog rješenja, u paralelnom rješenju izvornu matricu dijelimo na 4 podmatrice. Podjela matrica na manje matrice se vrši dok se ne dostigne određeni prag dimenzije podmatrice (u kodu je taj prag opisan konstanom CUTOFF). Nakon dostignutog praga, dalja podjela matrica na podmatrice nema smisla, pa će se program izvršavati serijski nad podopsezima, koji se međusobno izvršavaju paralelno. Radi paralelizacije ovog pristupa, korišćena je Threading Building Blocks (TBB) biblioteka.

## DETEKCIJA IVICA U OKOLINI PIKSELA

Serijsko rješenje je smješteno u funkciji filter\_serial\_edge\_detection. Na početku se vrši postavljanje svakog elementa matrice na vrijednost 0 ili 1, u zavisnosti od praga odsijecanja (u kodu je prag predstavljen konstantom THRESHOLD). Nakon toga se vrši prolazak kroz svaki element matrice i postavljaju se vrijednosti promjenljivih P i O. Detaljan opis kriterijuma postavljanja navedenih promjenljivih, kao i računanje izlazne vrijednosti, je dat u poglavlju KONCEPT RJEŠENJA – DETEKCIJA IVICA U OKOLINI PIKSELA.

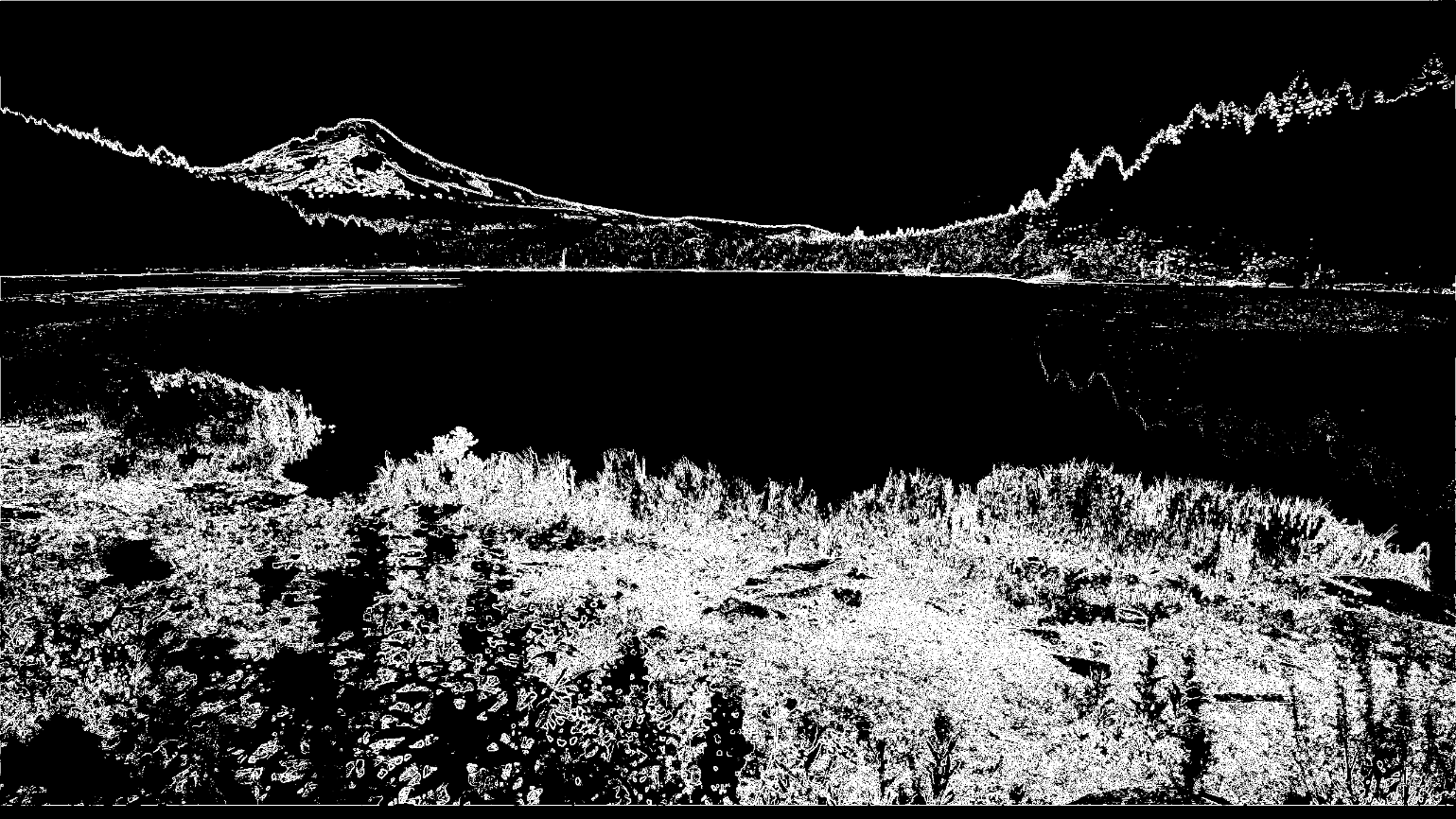
Paralelno rješenje je smješteno u funkcijama filter\_parallel\_edge\_detection i next\_iter\_parallel\_edge\_detection. U funkciji filter\_parallel\_edge\_detection se vrši prebacivanje vrijednost matrice na 0 ili 1, u zavisnosti od praga odsijecanja. Nakon toga se vrši pozivanje funkcije next\_iter\_parallel\_edge\_detection. Navedena funkcija dijeli matricu na 4 podmatrice, dok podmatrice ne dostignu određeni prag dimenzije. Nakon toga, program se izvršava serijski, jer bi dalja paralelizacija izazvala usporenja u izvršenju. Podmatrice se međusobno izvršavaju paralelno. Radi paralelizacije, korišćena je Threading Building Blocks (TBB) biblioteka.

U nastavku se nalaze primjeri slika prije i poslije korišćenja algoritama za detekciju ivica slike.

## PRIMJERI ULAZNIH I IZLAZNIH SLIKA



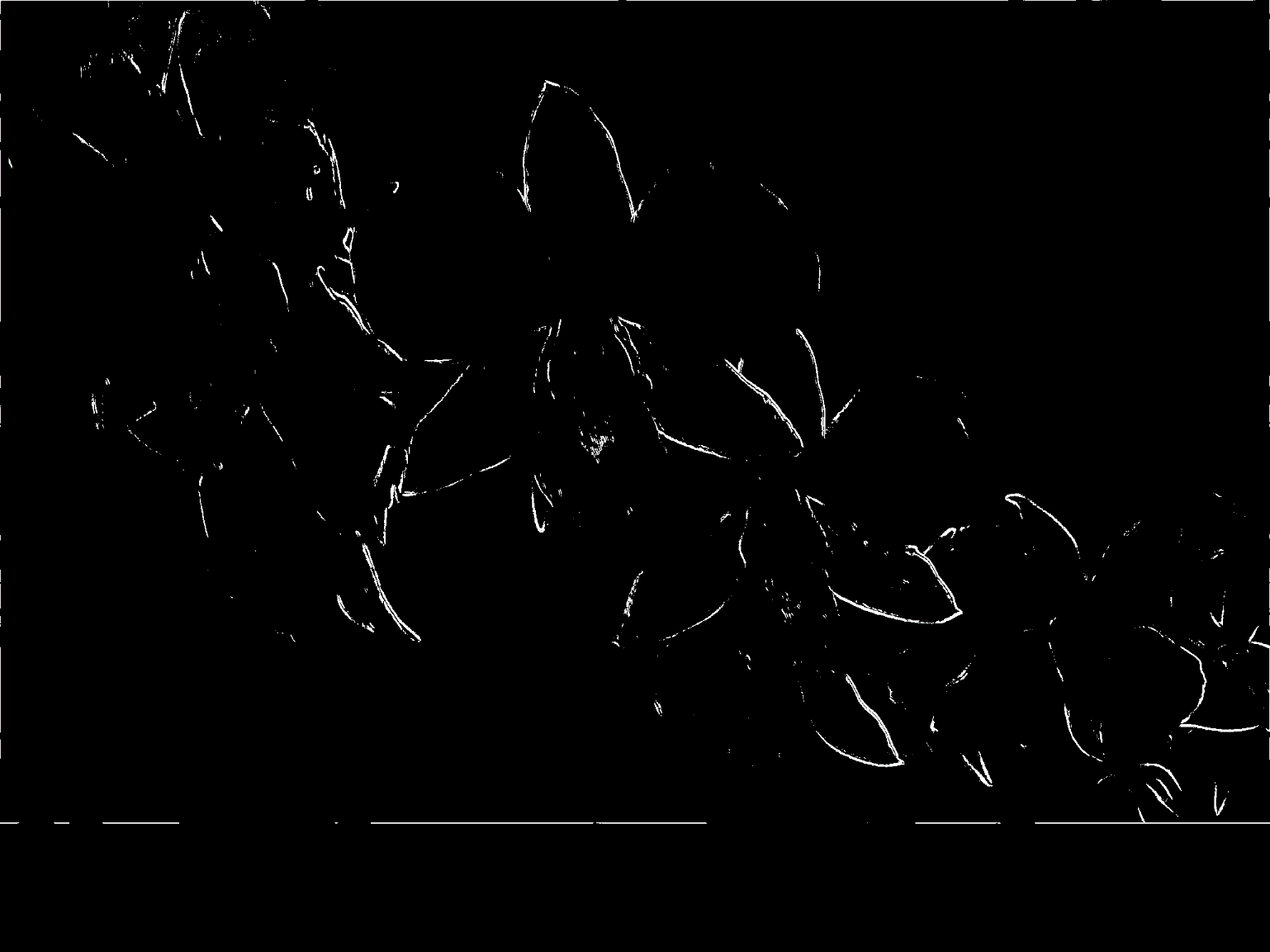
Izvorna slika 1

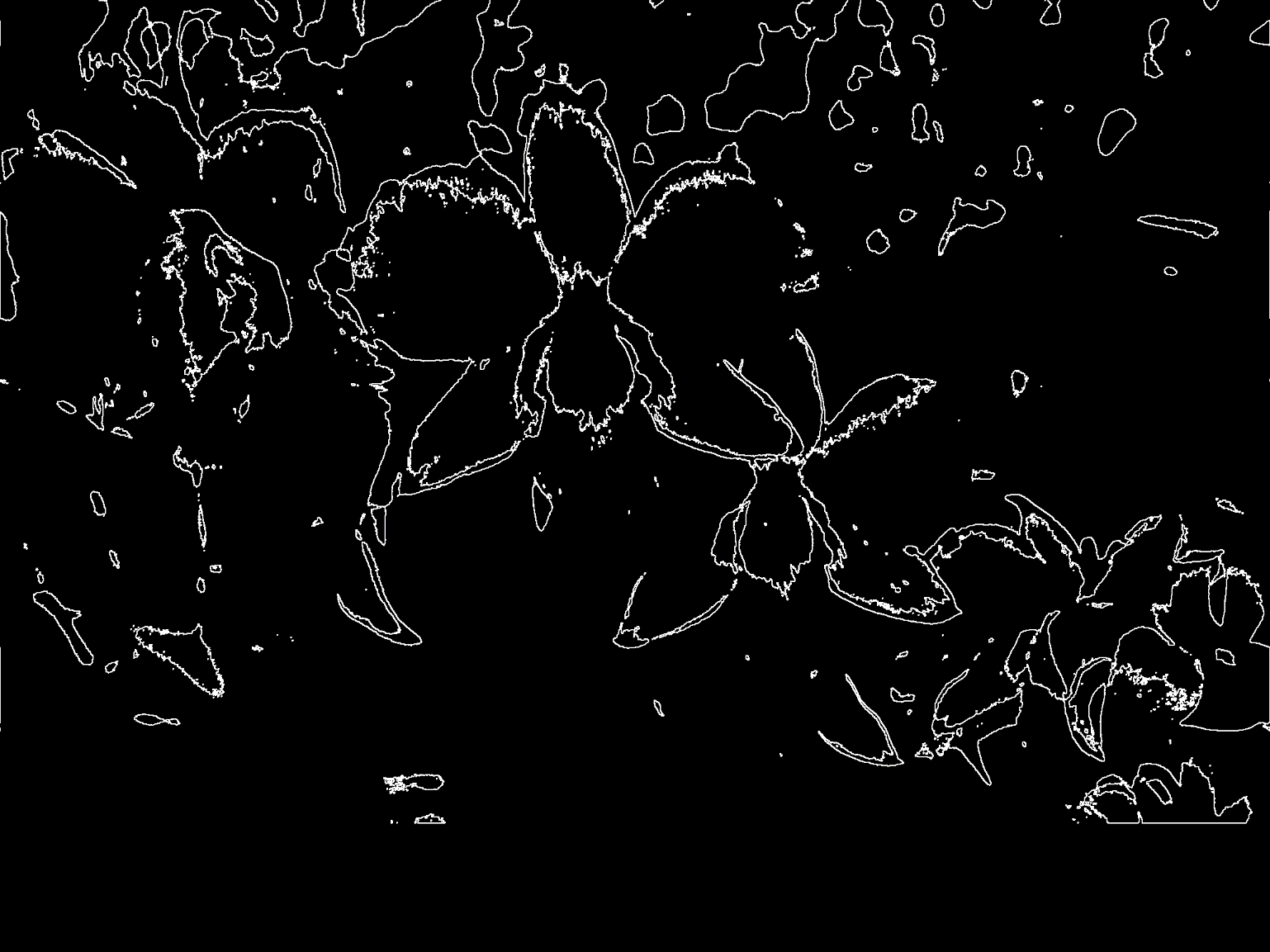


Slika 1 nakon detekcije ivica u okolini piksela

Slika 1 nakon detekcije ivica sa Prewitt operatorom

Izvorna slika 2 – primjer slike sa manje izraženim ivicama

Slika 2 nakon detekcije ivica sa Prewitt operatorom

Slika 2 nakon detekcije ivica u okolini piksela

# ISPITIVANJE

Testiranja su vršena na računaru sa Intel i7-105110U 1.80GHz 2.30GHz procesorom, nad 4 procesorska jezgra i 8 GB RAM memorije, na Windows 10 operativnom sistemu.

## TABELARNI PRIKAZ SERIJSKOG IZVŠENJA RAZLIČITIH DIMENZIJA SLIKE

Ispod se nalazi tabela brzine (ms) serijskog izvršenja različitih veličina slika.

\*udaljenost u tabelama predstavlja parametar udaljenosti u pristupu detekciji ivica u okolini piksela

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Veličina slike | Prewitt 3 x 3 | Prewitt 5 x 5 | Udaljenost 1 | Udaljenost 2 | Udaljenost 3 |
| 3888 x 2592 | 2977.03 | 7127.49 | 2858.31 | 7014.32 | 13539.9 |
| 2560 x 1440 | 1123.48 | 2844.71 | 1068.28 | 2716.09 | 5081.9 |
| 1920 x 1080 | 611.482 | 1421.63 | 575.418 | 1552.54 | 2749.8 |

U ovim testnim slučajevima, najbrži se pokazao algoritam za detekciju piksela u okolini piksela na udaljenosti 1. Bolje performanse od Prewitt-a su nastale usljed manjeg broja operacija nad elementima matrice.

## TABELARNI I GRAFIČKI PRIKAZI PARALELNOG IZVRŠENJA RAZLIČITIH DIMENZIJA PROBLEMA

Ispod se nalazi tabela brzine (ms) paralelnog izvršenja slike veličine 3888 x 2592 sa različitim pragovima odsijecanja.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Prag odsijecanja | Prewitt 3x3 | Prewitt 5x5 | Udaljenost 1 | Udaljenost 2 | Udaljenost 3 |
| 100 | 468.446 | 1325.18 | 557.825 | 1652.64 | 2487.02 |
| 200 | 519.206 | 1474.93 | 550.243 | 1374.4 | 2638.94 |
| 300 | 521.288 | 1329.27 | 548.285 | 1366.21 | 2625.33 |
| 400 | 520.313 | 1468.51 | 575.774 | 1357.02 | 2525.21 |
| 500 | 546.407 | 1403.11 | 569.003 | 1375.87 | 2666.43 |

Postignuto ubrzanje je približno 5 puta bolje u odnosu na serijsko izvršenje. Kao najbolji prag odsijecanja se pokazao prag od 400 piksela.

Ispod se nalazi tabela brzine (ms) paralelnog izvršenja slike veličine 2560 x 1440 sa različitim pragovima odsijecanja.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Prag odsijecanja | Prewitt 3x3 | Prewitt 5x5 | Udaljenost 1 | Udaljenost 2 | Udaljenost 3 |
| 100 | 204.092 | 481.311 | 204.465 | 510.902 | 915.024 |
| 200 | 217.725 | 492.477 | 202.971 | 528.772 | 953.971 |
| 300 | 199.846 | 523.011 | 204.75 | 512.523 | 980.398 |
| 400 | 197.252 | 500.882 | 261.516 | 527.198 | 1018.87 |
| 500 | 207.47 | 543.438 | 207.155 | 512.962 | 937.242 |

Rezultati paralelnog programa su, ponovo, otprilike 5 puta bolji od serijskog izvršenja. Ovoga puta se kao najbolji prag odsijecanja pokazao prag od 300 piksela.

Ispod se nalazi tabela brzine (ms) paralelnog izvršenja slike veličine 1920 x 1080 sa različitim pragovima odsijecanja.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Prag odsijecanja | Prewitt 3x3 | Prewitt 5x5 | Udaljenost 1 | Udaljenost 2 | Udaljenost 3 |
| 100 | 112.673 | 290.744 | 106.841 | 260.332 | 523.83 |
| 200 | 115.898 | 279.954 | 107.976 | 270.129 | 518.512 |
| 300 | 152.369 | 285.637 | 109.328 | 274.703 | 538.32 |
| 400 | 136.034 | 342.719 | 127.306 | 283.28 | 527.201 |
| 500 | 118.147 | 286.928 | 109.741 | 268.946 | 527.403 |

U ovom slučaju, kao najbolja solucija za prag odsijecanja se pokazalo 200 piksela, a postignuto ubrzanje je približno 5 puta.

U svakom od prethodnih primjera se desilo ubrzanje od 5 puta i glavna matrica se dijeli na 16 podmatrica, zbog čega se i prag odsijecanja smanjuje sa smanjenjem veličine slike.

# ANALIZA REZULTATA

U zaključku ćemo primijetiti da smo postigli značajna ubrzanja paralelizacijom problema (ubrzanja od, otprilike, 5 puta). Naravno to ubrzanje bi bilo još značajnije na većim problemima, gdje bi mogli uštediti mnogo više vremena. Za različite dimenzije slike, različiti pragovi odsijecanja će se pronaći kao optimalni.

Takođe, primjećujemo da je najgore performanse pokazao algoritam sa okolinom piksela na udaljenosti od 3 piksela. Ovo ima smisla jer se posmatra veća matrica u odnosu na ostale udaljenosti. Povećanjem udaljenosti, uočene ivice na slikama se podebljavaju, a povećanjem veličine filtera, kod Prewitt operatora, gustina uočenih ivica se smanjuje.

U konačnom zaključku, paralelizacija ovog programa je itekako korisna, posebno za zahtjevnija filtriranja.