

UNIVERZA V MARIBORU

FAKULTETA ZA ELEKTROTEHNIKO,
RAČUNALNIŠTVO IN INFORMATIKO

Danijel Žlaus

POHITRITEV TRANSFORMACIJE DOMNEVNIH
RAZDALJ

Magistrsko delo

Maribor, avgust 2016

POHITRITEV TRANSFORMACIJE DOMNEVNIH RAZDALJ

Magistrsko delo

Študent: Danijel Žlaus

Študijski program: Računalništvo in informacijske tehnologije (MAG)

Mentor: doc. dr. Domen Mongus, univ. dipl. inž. rač. in inf.



Univerza v Mariboru



Fakulteta za elektrotehniko,
računalništvo in informatiko

Smetanova ulica 17
2000 Maribor, Slovenija

Številka: E5000500

Datum in kraj: 14. 06. 2016, Maribor

Na osnovi 330. člena Statuta Univerze v Mariboru (Ur. l. RS, št. 44/2015)
izdajam

SKLEP O MAGISTRSKEM DELU

1. **Danijelu Žlausu**, študentu študijskega programa 2. stopnje RAČUNALNIŠTVO IN INFORMACIJSKE TEHNOLOGIJE, se dovoljuje izdelati magistrsko delo.

2. Tema magistrskega dela je pretežno s področja Inštituta za računalništvo.

MENTOR: doc. dr. Domen Mongus

3. Naslov magistrskega dela:

POHITRITEV TRANSFORMACIJE DOMNEVNIH RAZDALJ

4. Naslov magistrskega dela v angleškem jeziku:

OPTIMIZATION OF QUASI DISTANCE TRANSFORM

5. Magistrsko delo je potrebno izdelati skladno z »Navodili za izdelavo magistrskega dela«. Skladno s 6. členom *Pravilnika o postopku priprave in zagovora magistrskega dela na študijskih programih 2. stopnje UM* je bilo odobreno podaljšanje roka za oddajo magistrskega dela do 19. 06. 2017. Magistrsko delo študent-ka odda v treh izvodih (dva trdo vezana izvoda in en v spiralo vezan izvod) v pristojni referat ter elektronski izvod v Digitalno knjižnico Univerze v Mariboru.

Pravni pouk: Zoper ta sklep je možna pritožba na senat članice v roku 15 dni.

Obvestiti:

1. kandidata
2. mentorja
3. odložiti v arhiv



Dekan:

red. prof. dr. Borut Žalik

Pohitritev transformacije domnevnih razdalj

Ključne besede: matematična morfologija, transformacija domnevnih razdalj, transformacija razdalj, časovna zahtevnost, optimizacijske metode

UDK: 004.93'11(043.2)

Povzetek:

V magistrskem delu opisujemo pohitritev transformacije domnevnih razdalj, ki je izpeljanka tradicionalnih algoritmov transformacij razdalj. Transformacije razdalj običajno delujejo nad dvodimenzionalnimi binarnimi slikami, kjer vsakemu elementu osredja določijo oddaljenost do najbližjega elementa ozadja. Kadar slika ni binarna, je nad njo potrebno izvesti dodano predprocesiranje, ki vključuje korak binarizacije. Nasprotno pa lahko transformacijo domnevnih razdalj uporabimo neposredno nad sivinskimi, barvnimi in multispektralnimi slikami in se tako izognemo pogoste neželenemu predprocesiranju. Slabost tega pristopa pa je časovna zahtevnost, ki je v naivni implementaciji kar $O(N^{2.5})$. V magistrskem delu predstavimo pohitren algoritem transformacije domnevnih razdalj ter teoretično analizo njegove časovne zahtevnosti. Nad implementiranim algoritmom izvedemo tudi meritve, s čimer potrdimo teoretične časovne zahtevnosti pohitrenega pristopa, ki je enaka $O(N^{1.5})$ v pričakovanem ter $O(N^2)$ v najslabšem primeru.

Optimization of Quasi Distance Transform

Keywords: mathematical morphology, quasi distance transform, distance transform, time complexity, optimisation methods

UDC: 004.93'11(043.2)

Abstract:

This thesis presents an optimisation of Quasi Distance Transform, derived from traditional distance transformation algorithms. Distance transform is typically applied on two-dimensional binary images, where each foreground element is assigned the distance to its closest background element. When the image is not binary, additional preprocessing with binarisation is required. On the other hand, Quasi Distance Transform can be applied directly to grayscale, colour or multi-spectral images, thus avoiding often unwanted preprocessing. However, the weakness of the method is its time complexity, which is in naive implementation equal to $O(N^{2.5})$. In this thesis, we present a new optimised algorithm and the theoretical analysis of its time complexity. We confirm the theoretical time complexity for quasi distance transformation together with measurements, thus proving they are $O(N^{1.5})$ for the expected and $O(N^2)$ for the worst case.

Kazalo

1	Uvod	1
2	Teoretične osnove	3
2.1	Norme	3
2.2	Transformacija razdalj	5
2.3	Popolna urejenost vhodnih podatkov	6
2.4	Matematična morfologija	7
2.4.1	Morfološko krčenje	8
2.4.2	Morfološko širjenje	9
2.4.3	Morfološko odpiranje	10
2.4.4	Morfološko zapiranje	10
2.4.5	Skrajno krčenje	11
2.4.6	Transformacija domnevnih razdalj	13
3	Pohitritev algoritma transformacij domnevnih razdalj	15
3.1	Časovna zahtevnost transformacije domnevnih razdalj	15
3.2	Razčlenitev zaporednih oken	16
3.3	Ponovna uporaba informacij iz prejšnjega položaja	18
3.4	Dodatna razčlenitev zaporednih oken in analiza odvisnosti	19
3.5	Časovna zahtevnost	21
3.5.1	Pričakovana časovna zahtevnost	21
3.5.2	Časovna zahtevnost v najslabšem primeru	23
3.6	Večnitno procesiranje	24

<i>KAZALO</i>	VII
4 Rezultati	26
4.1 Meritve	26
4.2 Potrditev pričakovane časovne zahtevnosti	28
4.3 Potrditev časovne zahtevnosti v najslabšem primeru	29
4.4 Meritve z uporabo večnitnost	30
4.5 Meritve pomnilniške zahtevnost	32
5 Sklep	33

Slike

2.1	Primer poti med dvema točkama p in q (zelena, modra in rdeča) z enako Manhattanovo razdaljo.	4
2.2	Razdalja med p in q po Evklidski normi, ki je enaka dolžini vektorja med točkama p in q	4
2.3	Razdalja med p in q po Chebyshejevi normi, ki je enaka največji absolutni razliki koordinat.	5
2.4	Primeri transformacij razdalj pri danih normah L_1 , L_2 in L_∞	6
2.5	Razrez sivinske slike na serijo binarnih slik.	8
2.6	Primer krčenja sivinske slike z oknoma različnih velikosti.	9
2.7	Primer širjenja sivinske slike z oknoma različnih velikosti.	10
2.8	Primer odpiranja sivinske slike z oknoma različnih velikosti.	10
2.9	Primer zapiranja sivinske slike z oknoma različnih velikosti.	11
2.10	Primer skrajne erozije.	12
2.11	Primer transformacije domnevnih razdalj.	14
3.1	Razčlenitev oken w^1 , w^2 in w^3	17
3.2	Razčlenjeno okno $w^{0 \rightarrow 7}$	17
3.3	Posplošitev premikov iz predhodnega položaja p_o na nov položaj p z uporabo zasukov.	18
3.4	Dodatna razčlenitev okna $w^{0 \rightarrow 4}$	19
3.5	Prekrivanje oken na položajih p in p_o	20
3.6	Ena izmed možnih delitev slike, kadar uporabimo štiri niti.	25
4.1	Izvajalni časi pri različnih velikosti slik.	28

4.2	Izvajalni časi pri različnih velikostih slik, kjer je časovna os logaritmirana.	28
4.3	Povprečen čas za operacijo.	29
4.4	Povprečen čas za operacijo z monokromatsko sliko.	30
4.5	Primerjava hitrosti kadar uporabimo eno, dve, tri ali štiri niti.	31
4.6	Relativna primerjava pohitritve glede na izvajalni čas ene niti.	31

Tabele

4.1	Izpiski izvajalnih časov pri različnih velikosti slik.	27
4.2	Povprečna pohitritev glede na število uporabljenih niti.	31
4.3	Poraba pomnilnika glede na število uporabljenih niti.	32

Poglavje 1

Uvod

Transformacija razdalj predstavlja družino algoritmov [1], ki jih pogosto uporabljamo nad dvodimenzionalnimi binarnimi slikami. Transformacija razdalj pripiše vsakemu elementu ospredja razdaljo do njemu najbližjega elementa ozadja. Metode, ki iz tega izhajajo, pa omočajo na primer ločevanje prekritih objektov, izračun skeleta, načrtovanju poti in ujemanje oblik [1][2][3][4]. Predpogoj za izvedbo transformacije razdalj je enolično definirano ospredje in ozadje. Tradicionalni pristopi se zato izvajajo nad binarnimi podatki, kjer vrednost 1 razumemo kot ospredje in vrednost 0 kot ozadje. Pri obdelavi sivinskih, barvnih ali multispektralnih slik pa takšni algoritmi zahtevajo dodatno predprocesiranje, ki vključuje binarizacijo, kar je pogosto nezaželeno in lahko vodi do napak.

Transformacija domnevnih razdalj (ang. quasi distance transformation - QDT), ki jo je uvedel Beucher[5], se od preostalih transformacij razdalj razlikuje v tem, da jo je mogoče uporabiti neposredno nad večdimenzionalnimi vhodnimi podatki in zato ne zahteva predhodne binarizacije. Operator izhaja iz matematične morfologije, ki podaja algebrajično osnovo za računsko vrednotenje geometrijskih oblik in tekstur. Slabost metode domnevnih razdalj pa je njena časovna zahtevnost, ki je v naivni implementaciji enaka $O(N^{2.5})$, medtem ko klasične transformacije razdalj zahtevajo linearno časovno zahtevnost $O(N)$ [6] [7]. V obeh primerih N predstavlja število elementov vhodne slike.

Predmet tega magistrskega dela je razvoj učinkovitega algoritma transformacije domnevnih razdalj. Teoretično časovno zahtevnost algoritma pri tem tudi formalno dokažemo in jo podpremo z učinkovito implementacijo ter izvedbo meritev. V poglavju 2 predstavimo teoretične osnove za izgradnjo algoritma domnevnih razdalj. Poglavje 3 je namenjeno predsta-

vitvi algoritma ter analizi njegove časovne zahtevnosti. Na koncu poglavja 3 ovrednotimo pomnilniško zahtevnost razvite metode ter uporabo večnitnosti za dodatno pohiritev algoritma. Rezultati meritev so predstavljeni v poglavju 4, med tem ko poglavje 5 podaja sklepe diplomskega dela.

Poglavje 2

Teoretične osnove

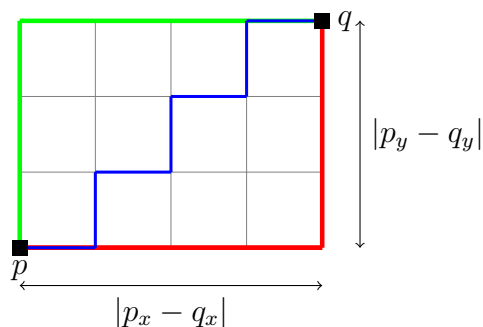
Transformacija domnevnih razdalj je definirana kot operator matematične morfologije. V magistrskem delu obravnavamo dvodimenzionalne slike, ki so sestavljene iz posameznih slikovnih elementov, tako imenovanih pikslov. Slikovni element definiramo kot $p = (x, y) \in E$, pri čemer $E \subset \mathbb{Z}^2$. Naj bo S funkcija, ki preslika položaj elementa v celoštevilsko vrednost pripadajoče intenzitete, $S(p) : E \rightarrow \mathbb{Z}$. Transformacija domnevnih razdalj vsakemu elementu osredja $p^\circ \in E$ določi vrednost razdalje r_p , do njemu najbližjega elementa ozadja $p^\bullet \in E$. Glede na podane notacije v nadaljevanju zato najprej predstavimo definicijo norme in njen vpliv na rezultate transformacij razdalj. Nato podamo definicije lastnosti, katerim mora zadostiti vsaka množica vhodnih podatkov, nad katero je vzpostavljena popolna urejenost. Popolna urejenost množice nam omogoča uporabo operatorjev matematične morfologije[8] in posledično tudi operatorjev na katerih temelji transformacija domnevnih razdalj. Slednje predstavimo na koncu tega poglavja.

2.1 Norme

Predpostavimo slikovna elementa $p, q \in \mathbb{Z}^2$. Razdalja med njima je določena z izračunom norme njune vektorske razlike. Najpogosteje uporabljene norme so L_1 , L_2 in L_∞ . Manhattanova razdalja L_1 (podana z enačbo 2.1) je definirana kot vsota vseh absolutnih koordinat.

Primer Manhattanove razdalje prikazuje slika 2.1.

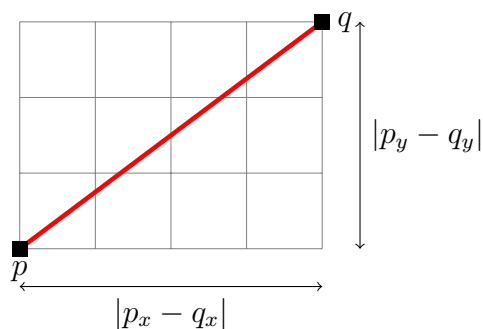
$$L_1(p - q) = \sum_i |p_i - q_i| = |p_x - q_x| + |p_y - q_y| \quad (2.1)$$



Slika 2.1: Primer poti med dvema točkama p in q (zelena, modra in rdeča) z enako Manhattanovo razdaljo.

Evklidska razdalja L_2 (podana z enačbo 2.2) je v dveh dimenzija enaka Pitagorovem izreku. Primer evklidske razdalje je prikazan na sliki 2.2.

$$L_2(p - q) = \sqrt{\sum_i (p_i - q_i)^2} = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \quad (2.2)$$

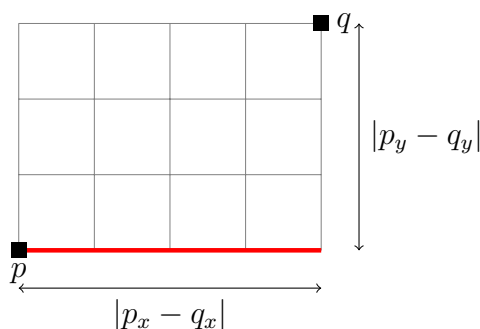


Slika 2.2: Razdalja med p in q po Evklidski normi, ki je enaka dolžini vektorja med točkama p in q .

Chebyshejeva razdalja L_∞ (podana z enačbo 2.3) je definirana z največjo absolutno razliko koordinat, kar je posledica enačbe 2.4, kadar jo ovrednotimo z $\lim_{p \rightarrow \infty}$. Primer Chebyshe-

jeve razdalje je prikazan na sliki 2.3.

$$L_\infty(p - q) = \max(|p_0 - q_0|, |p_1 - q_1|, \dots, |p_n - q_n|) = \max(|p_x - q_x|, |p_y - q_y|) \quad (2.3)$$



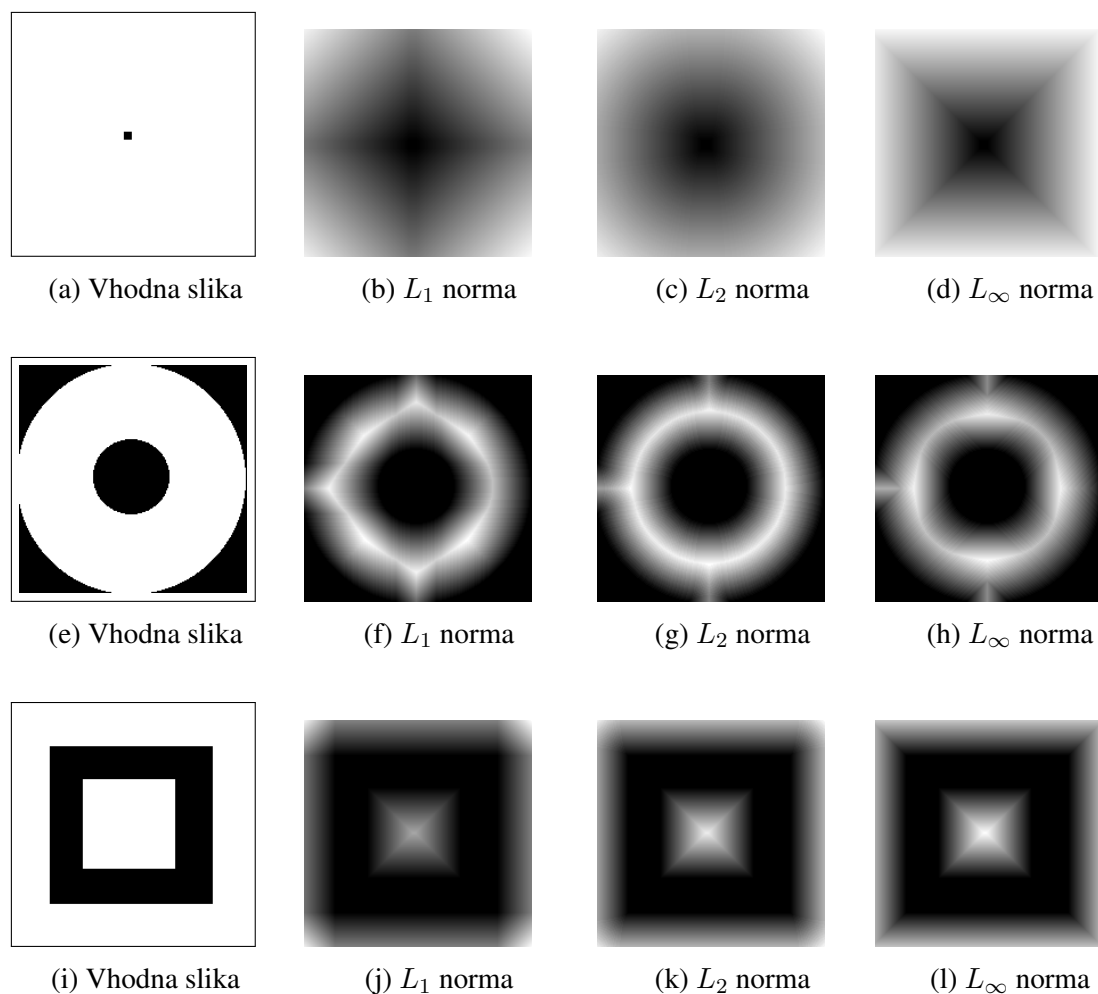
Slika 2.3: Razdalja med p in q po Chebyshejevi normi, ki je enaka največji absolutni razliki koordinat.

V splošnem pa velja, da so L_1 , L_2 in L_∞ posebni primeri posplošene p -norme, pri $p = 1$, $p = 2$ in $p = \infty$.

$$L_p(p - q) = \sqrt[p]{\sum_i |p_i - q_i|^p} \quad (2.4)$$

2.2 Transformacija razdalj

Transformacija razdalj je operator, ki vsakemu elementu ospredja $p^\circ \in E$ dodeli celoštevilsko vrednost do najbližjega elementa ozadja $p^\bullet \in E$. Vrednost transformacije v elementih ozadja je posledično enaka 0, pri čemer je formalna definicija preslikave podana kot $dt(p) : E \rightarrow \mathbb{Z}$. Na sliki 2.4 lahko vidimo primere transformacij razdalj nad različnimi binarnimi slikami z uporabo različnih norm. Vrednosti 0 določajo ozadje slike, medtem ko vrednosti 1 določajo ospredje.

Slika 2.4: Primeri transformacij razdalj pri danih normah L_1 , L_2 in L_∞ .

2.3 Popolna urejenost vhodnih podatkov

Metode matematične morfologije zahtevajo popolno urejenost med obravnavanimi vrednostmi[8].

Da je poljubna množica X popolno urejena, morajo zanjo veljati naslednje lastnosti[9]:

1. Ne-refleksivnost: $a < a$ ne drži za noben $a \in X$
2. Simetričnost: kadar $a < b$ drži, potem izraz $b < a$ ne drži, za vsak $a, b \in X$
3. Transitivnost: kadar drži $a < b$ in $b < c$ mora držati tudi $a < c$, za vsak $a, b, c \in X$
4. Sovisnost: kadar drži $a \leq b$ ali $b \leq a$ za vsak $a, b \in X$

2.4 Matematična morfologija

Danes še vedno pogosto delimo metode matematične morfologije v binarne in sivinske, čeprav je Soille[10] podal splošno definicijo sivinske matematične morfologije, ki velja za vse vrste vrednosti nad katerimi je opredeljena popolna urejenost. Tako je sivinska matematična morfologija popolnoma nadomestila binarno. Pred tem pa je Heijmans[11] že dokazal, da je vse sivinske operatorje matematične morfologije mogoče realizirati z binarnimi morfološkimi operacijami in uporabo pragov. To storimo tako, da sivinsko sliko predstavimo kot serijo binarnih slik, definiranih z bitnimi ravninami, kot je prikazano na sliki 2.5. Ker lahko sivinsko morfologijo izrazimo z binarno morfologijo in obratno, slednja zadostuje vsem lastnostim sivinske morfologije.

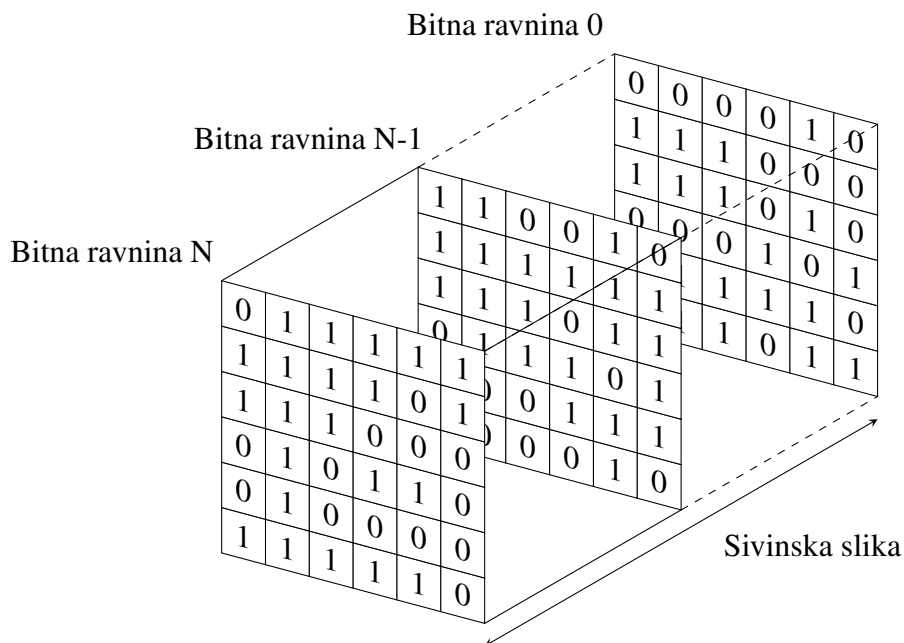
Osnovne metode matematične morfologije definiramo s strukturnim elementom w . Strukturni element je lahko poljubne oblike in deluje kot okno nad sliko S . Naj torej w^i predstavlja kvadratni strukturni element velikosti i (podano z enačbo 2.5), pri čemer je število vsebovanih elementov podano z enačbo 2.6.

$$w^i = (2i + 1) \times (2i + 1), \quad (2.5)$$

$$|w^i| = (2i + 1)^2. \quad (2.6)$$

Uporabo strukturnega elementa nad položajem p predstavimo z enačbo:

$$(S \circ w^i)(p). \quad (2.7)$$



Slika 2.5: Razrez sivinske slike na serijo binarnih slik.

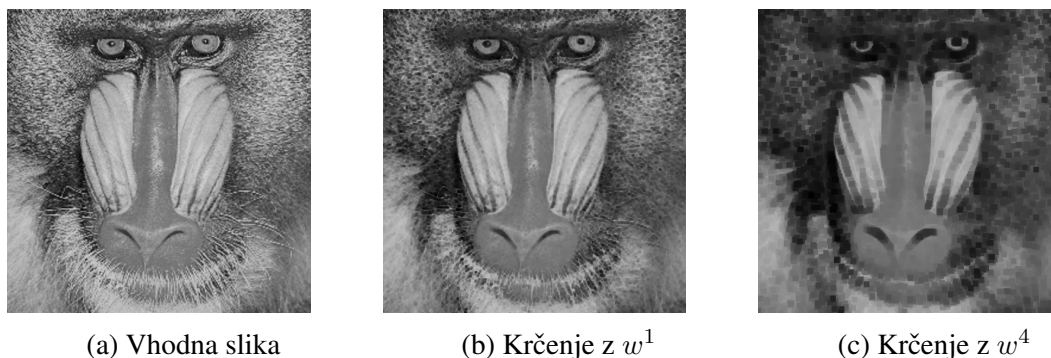
2.4.1 Morfološko krčenje

Morfološko krčenje (podano z enačbo 2.8) lahko razumemo kot iskanje minimalne vrednosti v S znotraj okna w , glede na p .

$$[S \ominus w](p) = \min_{\forall o \in w} \{S(p + o)\} \quad (2.8)$$

Ko opravimo krčenje za vsak položaj $p \in E$, ustvarimo novo sliko S' . Krčenje lahko ponovimo nad sliko S' ter tako definiramo S'' . Za primer vzemimo strukturni element w^1 in sliko S . Če opravimo dvojno krčenje slike S z oknom w^1 , je rezultat enak krčenju slike S z oknom w^2 , kar dokazuje enačba 2.9. Primer krčenja prikazuje slika 2.6.

$$\begin{aligned} [S \ominus w^1] &= S' & (2.9) \\ [S' \ominus w^1] &= S'' \\ [[S \ominus w^1] \ominus w^1] &= S'' \\ [S \ominus [w^1 \ominus w^1]] &= S'' \\ [S \ominus w^2] &= S'' \end{aligned}$$



(a) Vhodna slika

(b) Krčenje z w^1 (c) Krčenje z w^4

Slika 2.6: Primer krčenja sivinske slike z oknom različnih velikosti.

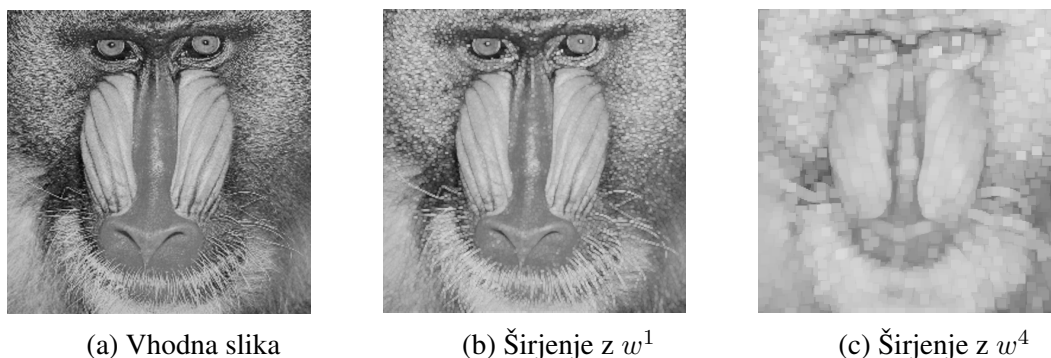
2.4.2 Morfološko širjenje

Morfološko širjenje (podano z enačbo 2.10) ima enake lastnosti kot krčenje, le da v tem primeru namesto minimalne vrednosti znotraj okna w iščemo maksimalno vrednost:

$$[S \oplus w](p) = \max_{\forall o \in w} S(p + o) \quad (2.10)$$

Posledično je širjenje možno izraziti s krčenjem. To izvedemo tako, da vrednosti vhodne slike negiramo. Z drugimi besedami če si sliko predstavljamo kot topografski relief, le tega "obrnemo na glavo". Nad negirano sliko izvedemo krčenje, rezultat pa ponovno negiramo (glej enačbo 2.11). Primer širjena prikazuje slika 2.7.

$$[S \oplus w](p) = -[-S \ominus w](p) \quad (2.11)$$



(a) Vhodna slika

(b) Širjenje z w^1 (c) Širjenje z w^4

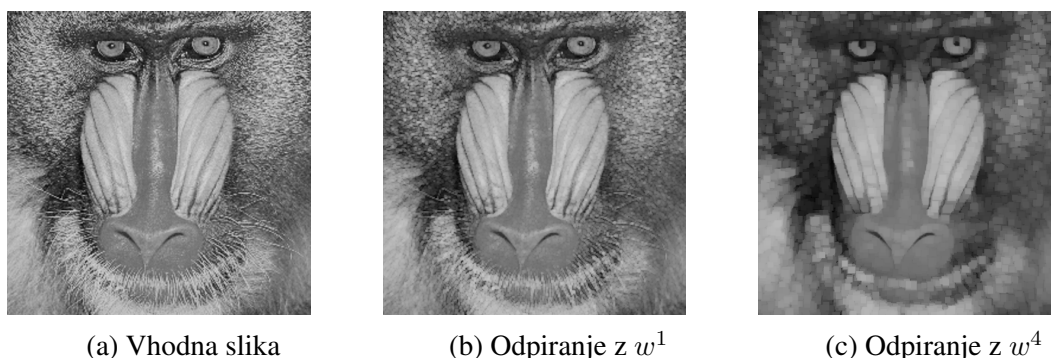
Slika 2.7: Primer širjenja sivinske slike z oknoma različnih velikosti.

2.4.3 Morfološko odpiranje

Morfološko odpiranje (podano z enačbo 2.12) spada med temeljne operacije matematične morfologije in je definirano kot kompozicija krčenja in širjenja. Odpiranje je idempotentno, kadar uporabimo eno samo okno w (glej enačbo 2.13). Primer odpiranja prikazuje slika 2.8.

$$[S \circ w](p) = [[S \ominus w] \oplus w](p) \quad (2.12)$$

$$[[S \circ w] \circ w](p) = [S \circ w] \quad (2.13)$$



(a) Vhodna slika

(b) Odpiranje z w^1 (c) Odpiranje z w^4

Slika 2.8: Primer odpiranja sivinske slike z oknoma različnih velikosti.

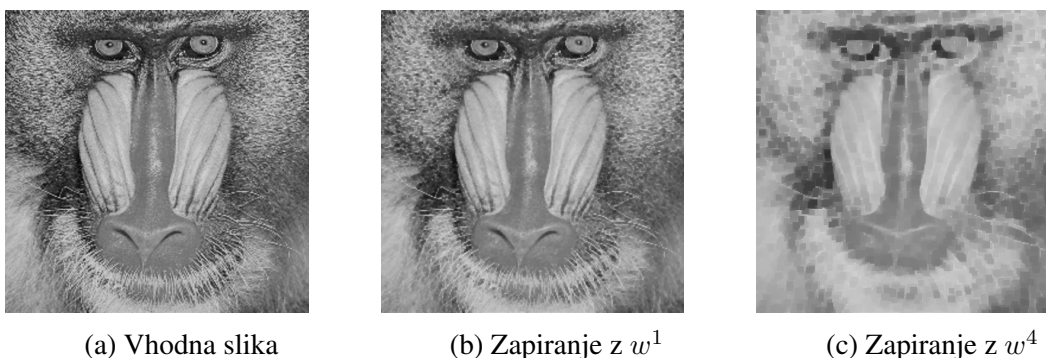
2.4.4 Morfološko zapiranje

Podobno kot morfološko odpiranje, tudi morfološko zapiranje (podano z enačbo 2.14) spada med osnovne metode matematične morfologije in je definirano kot kompozicija širjenja in

krčenja. Tudi ta operacija je idempotentna (glej enačbo 2.15). Primer zapiranja prikazuje slika 2.9.

$$[S \bullet w](p) = [[S \oplus w] \ominus w](p) \quad (2.14)$$

$$[[S \bullet w] \bullet w](p) = [S \bullet w] \quad (2.15)$$



Slika 2.9: Primer zapiranja sivinske slike z oknom različnih velikosti.

2.4.5 Skrajno krčenje

Skrajno krčenje je neposredno izpeljana iz morfološkega krčenja. Metoda opravi serijo krčitev s postopnim povečevanjem strukturnega elementa, od najmanjšega w^0 do največjega w^n , kjer je n največja stranica slike S .

Skrajno krčenje ima dva izhoda. Prvi izhod je transformacijska funkcija $r(f)$, ki prestavlja največjo razliko med vrednostnima dveh zaporednih krčenj. Drugi izhod je asociativna funkcija $s(f)$, ki predstavlja velikost okna pri katerem je prišlo do največje razlike. Zveznosti asociativne funkcije $s(f)$ pri tem ni možno zagotoviti. Bistveno težavo povzroča skrajno krčenje elementov z nizkimi vrednostmi oziroma lokalnih minimumov slike. Lastnost lokalnih minimumov je, da v neposredni okolici ni elementa z manjšo vrednostjo. Ti elementi bodo posledično skrčeni zgolj z velikimi okni, elementi v njihovi neposredni okolici pa z majhnimi okni. Kadar je vrednost lokalnega minimuma blizu globalnega minimuma, to dodatno okrepi potrebo po velikem oknu.

Slika 2.10 prikazuje vhodno sliko ter obe izhodni funkciji skrajnega krčenja. Opazimo lahko, da ima asociativna funkcija $s(f)$ velike vrednosti pretežno v elementih, ki imajo v

vhodni sliki nizke vrednosti. Pri teh elementih lahko v transformacijski funkciji $r(f)$ vidimo, da so tudi maksimalne razlike precej majhne.



(a) Vhodna slika

(b) Transformacijska funkcija
 $r(f)$ (c) Asociativna funkcija
 $s(f)$ (d) Histogramsko normalizirana
 $r(f)$ (e) Histogramsko normalizirana
 $s(f)$

Slika 2.10: Primer skrajne erozije.

2.4.6 Transformacija domnevnih razdalj

Transformacija domnevnih razdalj je neposredno izpeljana iz skrajnega krčenja, pri čemer odpravi nezveznosti v asociativni funkciji $s(f)$. Primere slednjih lahko vidimo na sliki 2.11(d).

Nezveznost v $s(f)$ odpravimo tako, da nad $s(f)$ prisilno vzpostavimo tako imenovano Lipschitzovo kontinuiteta. Ta omejuje maksimalno absolutno razliko med sosednjimi elementi. Za potrebe metode uporabimo 1-Lipschitzovo kontinuiteto, ki omeji največjo absolutno razliko z vrednostjo 1. Asociativno funkcijo $s(f)$ prisilimo v 1-Lipschitzovo kontinuiteto tako, da opravimo štiri kardinalne prehode skozi sliko. V vsakem prehodu popravimo vrednost slikovnega elementa p , če je njegova vrednost za več kot 1 večja od najmanjše vrednosti elementa v neposredni okolici. Posledično definiramo vrednost p_{min} kot:

$$p_{min} = p + \arg \min_{o \in \{-1,0,1\}^2} (s(f)(p + o)) \quad (2.16)$$

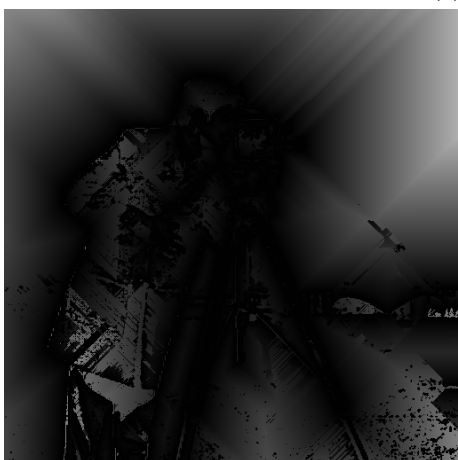
Kadar je razlika večja od 1, slikovnemu elementu p priredimo novo vrednost. Ta je natanko 1 večja od vrednosti $s(f)(p_{min})$, ne glede na prejšnjo vrednost $s(f)(p)$, glej enačbi 2.17 in 2.18. Ko opravimo vse štiri prehode, pridobimo novo funkcijo $s(f)'$, ki ima vzpostavljeno Lipschitzovo kontinuiteto reda 1. Zaradi načina odprave nezveznosti, dobljena funkcija predstavlja Chebyshejevo transformacijo razdalj. Primer transformacije domnevnih razdalj prikazuje slika 2.11.

$$\delta(p) = s(f)(p) - \min_{o \in \{-1,0,1\}^2} s(f)(p + o). \quad (2.17)$$

$$s(f)'(p) = s(f)(p) - \begin{cases} \delta(p) - 1 & \text{če } \delta(p) > 1, \\ 0 & \text{drugače} \end{cases} \quad (2.18)$$



(a) Vhodna slika



(b) Asociativna funkcija
 $s(f)$



(c) Zvezna asociativna funkcija
 $s(f)'$



(d) Histogramsko normalizirana
 $s(f)$



(e) Histogramsko normalizirana
 $s(f)'$

Slika 2.11: Primer transformacije domnevnih razdalj.

Poglavje 3

Pohitritev algoritma transformacij domnevnih razdalj

3.1 Časovna zahtevnost transformacije domnevnih razdalj

Transformacija domnevnih razdalj zahteva izračun asociativne funkcije $s(f)$ skrajnega krčenja. Zato najprej ovrednotimo časovno zahtevnost samega skrajnega krčenja in iz nje izpeljemo časovno zahtevnost transformacije domnevnih razdalj. Za namene izpeljave časovne zahtevnosti predpostavimo, da je slika S kvadratne oblike velikosti $n \times n$. Naj N predstavlja število elementov v sliki S , kjer je $N = |E| = n^2$. Operator skrajnega krčenja bo v centralnem elementu $p_c = (\frac{n}{2}, \frac{n}{2})$ opravil operacijo krčenja natanko n -krat, z okni velikosti w^0, w^1, \dots, w^n . Število obiskanih elementov pri elementu p_c izračunamo z vrsto:

$$|w^0| + |w^1| + \dots + |w^n| = \sum_{i=0}^n |w^i| = \sum_{i=0}^n (2i+1)^2 = \sum_{i=0}^n 4i^2 + 4i + 1 = 4n^3 + 12n^2 + 11n. \quad (3.1)$$

Iz enačbe 3.1 sledi časovna zahtevnost skrajnega krčenja pri elementu p_c :

$$O(4n^3 + 12n^2 + 11n) = O(n^3) = O(N^{1.5}). \quad (3.2)$$

Ker moramo to operacijo ponoviti za vsak $p \in E$, torej N -krat, je časovno zahtevnost skrajnega krčenja za celotno sliko S enaka:

$$O(N * N^{1.5}) = O(N^{2.5}). \quad (3.3)$$

Vzpostavitev Lipschitzove kontinuitete zahteva štiri dodatne prehode skozi celotno sliko, kar pomeni da je časovna zahtevnost transformacije domnevnih razdalj enaka:

$$O(4N + N^{2.5}) = O(N^{2.5}). \quad (3.4)$$

Iz enačbe 3.4 vidimo, da je časovna zahtevnost transformacije domnevnih razdalj enaka časovni zahtevnosti skrajnega krčenja. Vse pohitritve bodo torej opravljene nad metodo skrajnega krčenja, saj le ta določa dejansko časovno zahtevnost transformacije domnevnih razdalj.

3.2 Razčlenitev zaporednih oken

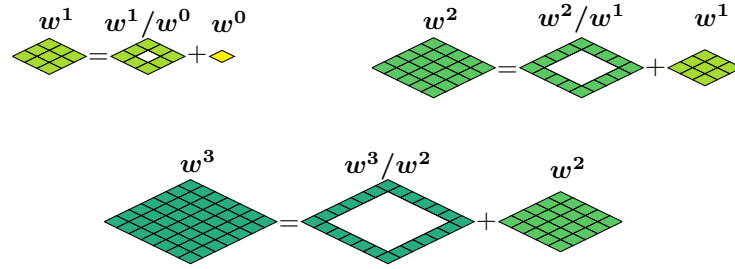
Za izboljšanje časovne zahtevnosti skrajnega krčenja se zanašamo na dekompozicijo okna, kjer okno w^{i+1} izrazimo z uporabo okna w^i , kot je prikazano na sliki 3.1. Če ponovno uporabimo že izračunano vrednost krčitve z oknom w^i , potem je za izračun krčitve z oknom w^{i+1} potrebno upoštevati zgolj vrednosti na obrobju okna, torej w^{i+1}/w^i ter že znano vrednost krčenja z oknom w^i . Iz tega nabora vrednosti poiščemo novo najmanjšo vrednost (glej enačbo 3.5).

$$[S \ominus w^{i+1}](p) = \min \left([S \ominus w^i](p), [S \ominus (w^{i+1}/w^i)](p) \right) \quad (3.5)$$

Naj $w^{i \rightarrow i+1}$ predstavlja vse vrednosti okna w^{i+1} , ki niso vsebovane v oknu w^i . Torej:

$$w^{i \rightarrow i+1} = w^{i+1}/w^i, \quad (3.6)$$

Če okno w^i vsebuje $(2i+1)^2$ elementov, potem oknu w^{i+1} vsebuje $(2(i+1)+1)^2 = (2i+3)^2$

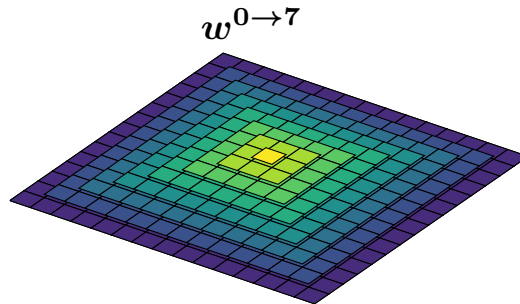


Slika 3.1: Razčlenitev oken w^1 , w^2 in w^3 .

elementov. Glede na enačbo 3.6, je število elementov v obrobju $w^{i \rightarrow i+1}$ enako:

$$|w^{i \rightarrow i+1}| = |w^{i+1}| - |w^i| = (2i + 3)^2 - (2i + 1)^2 = 8i + 8. \quad (3.7)$$

Na takšen način uspešno razčlenimo okna na serijo obrobij $w^{i \rightarrow i+1}$. Celotno skupek obrobij bomo označili z $w^{0 \rightarrow n}$. Primer okna $w^{0 \rightarrow 7}$, ki je sestavljeno iz teh obrobij lahko vidimo na sliki 3.2.



Slika 3.2: Razčlenjeno okno $w^{0 \rightarrow 7}$.

Časovno zahtevnost pri elementu p_c lahko torej izračunamo z sledečo vrsto:

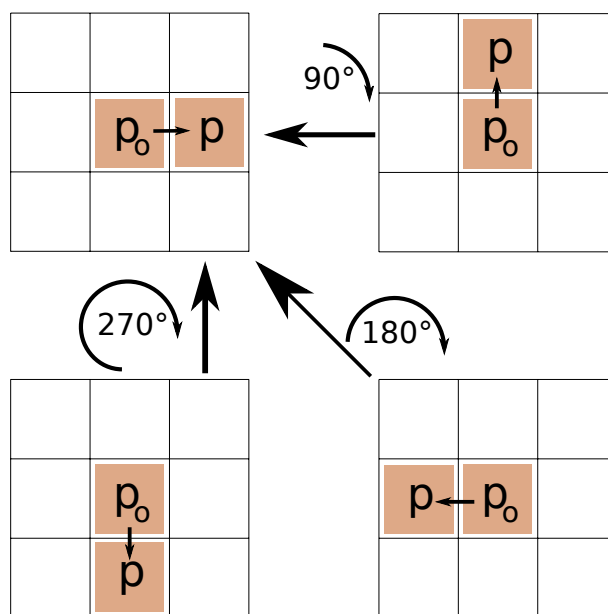
$$|w^{0 \rightarrow n}| = |w^{0 \rightarrow 1}| + |w^{1 \rightarrow 2}| + \dots + |w^{(n-1) \rightarrow n}| = \sum_{i=0}^n |w^{i \rightarrow i+1}| = \sum_{i=0}^n 8i + 8 = 4n^2 + 12n + 8, \quad (3.8)$$

kar vodi v oceno časovne zahtevnosti za posamezen element $O(4n^2 + 12n + 8) = O(4n^2) = O(N)$ in časovno zahtevnost $O(N * N) = O(N^2)$ za celotno sliko S . Da opravimo skrajno krčenje pri elementu p je potrebno obiskati vse elemente slike natanko enkrat. V primerjavi z naivno metodo tako dosežemo občutno pohitritev saj smo v slednjem morali nekatere elemente obiskati večkrat.

S ponovno uporabo informacij krčenja z oknom w^i v oknu w^{i+1} smo znižali časovno zahtevnosti iz $O(N^{2.5})$ na $O(N^2)$.

3.3 Ponovna uporaba informacij iz prejšnjega položaja

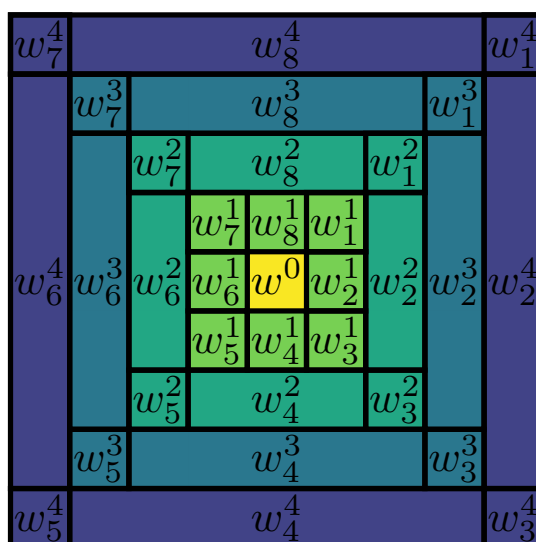
Dodatno pohitritev ni več možno doseči neposredno z znižanjem števila dostop do slike pri $p \in E$, saj smo v podpoglavju 3.2 to število minimizirali. Za dodatno pohitritev metode se zato osredotočimo na zmanjševanje števila dostopov do same slike pri preostalih položajih. To izvedemo tako, da ponovno uporabimo informacije iz predhodno obiskanega elementa p_o . Ker se lahko v naslednji element p pomaknemo v katerokoli izmed štirih kardinalnih smeri, je lahko predhodni element p_o na štirih različnih položajih. Kot prikazuje slika 3.3, je z uporabo zasukov možno prevesti vse premike na en premik.



Slika 3.3: Posplošitev premikov iz predhodnega položaja p_o na nov položaj p z uporabo zasukov.

3.4 Dodatna razčlenitev zaporednih oken in analiza odvisnosti

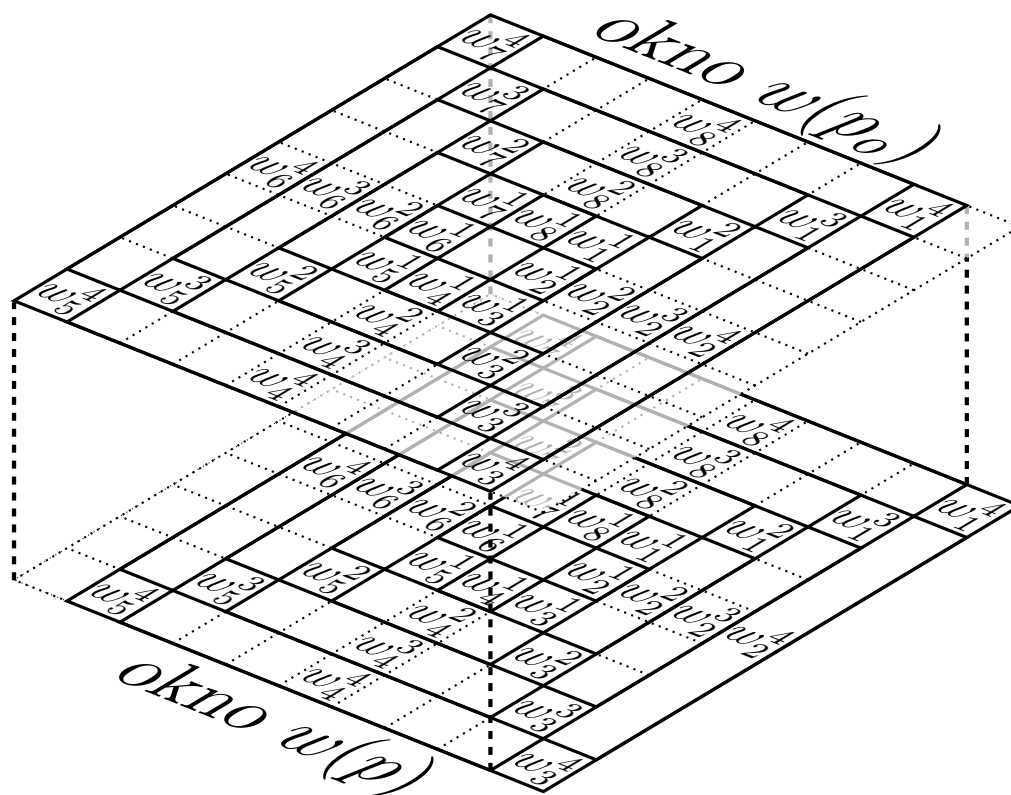
Za opis odvisnosti okna $w^{0 \rightarrow n}$ na položaju p od okna na položaju p_o , bomo okni dodatno razčlenili. Vsako okno bo dodatno razdeljeno na diagonale in na odseke med diagonalnimi elementi. Diagonalni elementi bodo poimenovani $w_{\{1,3,5,7\}}^i$, medtem ko bodo odseki poimenovali $w_{\{2,4,6,8\}}^i$, kot to prikazuje slika 3.4. Opisana metoda uporablja dva takšna okna, kjer je prvo okno na položaju p_o že ustvarjeno, sosednje okno na položaju p , pa še ne. Okni prekrijemo, kot to prikazuje slika 3.5, in opravimo analizo odvisnosti.



Slika 3.4: Dodatna razčlenitev okna $w^{0 \rightarrow 4}$.

Izkaže se, da vsi diagonalni elementi $w_{\{1,3,5,7\}}^i$ služijo samo kot podporni elementi pri izmenjavi vrednosti za odseke $w_{\{2,4,6,8\}}^i$. Teh torej ne moremo izmenjati med oknoma $w(p)$ in $w(p_o)$ in smo njihove vrednosti primorani ponovno ovrednoti. Za preostale odseke pa uporabimo tri strategije za izmenjavo informacij med oknoma:

- Hrbtna izmenjava je strategija izmenjave vrednosti, s katero ovrednotimo vrednosti levih odsekov $w_{\{2,4,6,8\}}^i(p)$. Minimalna vrednost odseka je pridobljena z minimalno vrednostjo diagonal 5,7 in odseka 6 predhodnega okna $w^{i-1}(p_o)$, $w_{\{2,4,6,8\}}^i(p) = \min w_{\{5,6,7\}}^{i-1}(p_o)$. Ta strategija potrebuje konstantni čas in je vedno uspešna.



Slika 3.5: Prekrivanje oken na položajih p in p_o .

- Stranska izmenjava je strategija izmenjave vrednosti za spodnje in zgornje odseke okna, $w_{4/8}^i(p)$. Vrednost določimo z uporabo diagonalne vrednosti $3/1$ in odseka $4/8$, $w_{4/8}^i(p) = \min w_{\{4,3\}/\{8,1\}}^i(p_o)$. Kadar je vrednost $w_{4/8}^i(p)$ enaka diagonalni vrednosti $w_{5/7}^i(p)$, vrednost zavržemo in poiščemo pravi minimum odseka v sliki S . To moramo narediti zato, ker obstaja možnost, da se je minimalna vrednost nahajala na skrajnem robu odseka v $w_{4/8}^i(p_o)$. Posledično se ta vrednost v oknu $w(p)$ nahaja na diagonalni $w_{5/7}^i(p)$.
- Čelna izmenjava je strategija izmenja vrednosti za desne odseke $w_2^i(p)$. Vrednost določimo tako, da prevzamemo vrednost prekrivajočega se odseka iz $w_2^{i+1}(p_o)$, $w_2^i(p) = w_2^{i+1}(p_o)$. Kadar je ta vrednost enaka kateri izmed diagonal $w_{1/3}^i(p)$, potem zaradi istega razloga kot v stranski izmenjavi, tudi tukaj vrednost odseka zavržemo in poiščemo nov minimum odseka v sliki S .

3.5 Časovna zahtevnost

Za ovrednotenje časovne zahtevnosti opisanega postopka je potrebno analizirati časovne zahtevnosti za diagonalne elemente in posamične strategije. Določitev vrednosti diagonalnega elementa je konstantna $O(1)$, saj ta predstavlja samo en element. V razčlenjenem oknu $w^{0 \rightarrow n}$ je v celoti $4n$ diagonalnih elementov, kar določa časovno zahtevnost tega koraka kot:

$$O(4n) = O(n). \quad (3.9)$$

Hrbtne strategije izmenjav elementov je dejansko iskanje novega minimalnega elementa izmed trojice vrednosti $w_{\{5,6,7\}}^{i-1}(p_o)$, kar zahteva dve operaciji. Časovna zahtevnost za vse hrbtne odseke določimo kot:

$$\sum_1^n 2 = 2n, \quad (3.10)$$

$$O(2n) = O(n). \quad (3.11)$$

Stranska in čelna strategija izmenjav nam predstavljata večjo težavo, saj nista deterministični. Posledično imata te dve strategiji pričakovano časovno zahtevnost ter časovno zahtevnost v najslabšem primeru.

3.5.1 Pričakovana časovna zahtevnost

Stranska izmenjava novo vrednost dodeli na sledeči način:

$$w_4^i(p) = \begin{cases} \min(w_{\{4,3\}}^i(p_o)) & \text{če } \min(w_{\{4,3\}}^i(p_o) \neq w_5^i(p), \\ \text{iskanje odseka v } S & \text{drugače} \end{cases} \quad (3.12)$$

$$w_i^s(p) = \begin{cases} \min(w_{\{8,1\}}^i(p_o)) & \text{če } \min(w_{\{8,1\}}^i(p_o) \neq w_7^i(p), \\ \text{iskanje odseka v } S & \text{drugače} \end{cases} \quad (3.13)$$

Medtem ko čelna izmenjava dodeli vrednost na sledeči način:

$$w_2^i(p) = \begin{cases} w_2^{i+1}(p_o) & \text{če } w_2^{i+1}(p_o) \neq w_1^i(p) \text{ in } w_2^{i+1}(p_o) \neq w_3^i(p), \\ \text{iskanje odseka v } S & \text{drugače} \end{cases} \quad (3.14)$$

Minimalno vrednost skrajnega čelnega odseka $w_2^n(p)$ moramo ponovno poiskati v sliki S , saj nima prekrivanja z oknom $w(p_o)$, kar lahko vidimo na sliki 3.5. Naj $\Pr(w_X^i)$ predstavlja verjetnost, da moramo vrednost odseka w_X^i poiskati v sliki in naj $|w_X^i|$ predstavlja število elementov v odseku. Pri stranski izmenjavi je verjetnost, da je minimalna vrednost na skrajni lokaciji obratno sorazmerna z velikostjo odseka:

$$\Pr(w_{4/8}^i) = \frac{1}{|w_{4/8}^i|}. \quad (3.15)$$

Pričakovana časovna zahtevnost za stransko izmenjavo je potem seštevek operacij pri izmenjavi ter dodatno delo, kadar moramo vrednost zavreči in jo ponovno poiskati v sliki:

$$\begin{aligned} 2 * \left(\sum_1^n 1 + \sum_{i=1}^n [|w_{4,8}^i| * \Pr(w_{4,8}^i)] \right) &= 2 * \left(\sum_1^n 1 + \sum_{i=1}^n 1 \right) \\ &= 2n + 2n \\ &= 4n. \end{aligned} \quad (3.16)$$

Pričakovano časovno zahtevnost lahko tako podamo kot:

$$O(4n) = O(n). \quad (3.17)$$

Verjetnost da moram pri čelni izmenjavi zavreči vrednost odseka je obratno sorazmerna z velikostjo. Dodatno moramo upoštevati, da se nam to lahko zgodi na obeh ekstremih odseka.

Posledično:

$$\Pr(w_2^i) = \frac{|w_2^{i+1}| - |w_2^i|}{|w_2^{i+1}|} = \frac{2}{|w_2^{i+1}|}. \quad (3.18)$$

Iz enačbe 3.20 sledi pričakovano število operacij za čelno izmenjavo:

$$\begin{aligned}
 \sum_1^n 1 + \sum_{i=1}^n [|w_2^i| * \text{Pr}(w_2^i)] &= n + \sum_{i=1}^n \frac{2|w_2^i|}{|w_2^{i+1}|} \\
 &= n + \sum_{i=1}^n \frac{4i + 2}{2i + 3} \\
 &\approx n + \sum_{i=1}^n \frac{4i}{2i} \\
 &= n + \sum_{i=1}^n 2 \\
 &= n + 2n \\
 &= 3n.
 \end{aligned} \tag{3.19}$$

Poenostavitev v enačbi 3.19 sledi iz limite izraza $\lim_{i \rightarrow \infty} \frac{4i + 2}{2i + 3} = \frac{4i}{2i}$. Časovna zahtevnost za čelno izmenjavo je torej:

$$O(3n) = O(n). \tag{3.20}$$

Skupna pričakovano časovna zahtevnost za vpostavitvev okna p glede na enačbe 3.9, 3.11, 3.17, 3.20 je sledeča:

$$O(n + n + n + n) = O(4n) = O(n) = O(N^{0.5}), \tag{3.21}$$

kar moramo ponoviti N -krat, kar nam da pričakovano časovno zahtevnost:

$$O(N * N^{0.5}) = O(N^{1.5}). \tag{3.22}$$

3.5.2 Časovna zahtevnost v najslabšem primeru

Najslabša časovna zahtevnost se pojavi, kadar je vhodna slika S monokromatska, saj bo to vedno prožilo ponovno iskanje minimuma v odsekih (glej enačbe 3.12, 3.13 in 3.14). Da ocenimo število operacij v najslabšem primeru vzamemo enačbi 3.16 in 3.19 ter iz njiju odstranimo verjetnost ponovnega iskanja $\text{Pr}(w_X^i)$. Iz tega dobimo za stransko izmenjavo

naslednje število operacij:

$$\begin{aligned}
 2* \left(\sum_1^n 1 + \sum_{i=1}^n |w_{4,8}^i| \right) &= 2* \left(n + \sum_{i=1}^n 2i + 1 \right) \\
 &= 2n + n^2 + 3n \\
 &= n^2 + 5n.
 \end{aligned} \tag{3.23}$$

Ter število operacij za čelno izmenjavo:

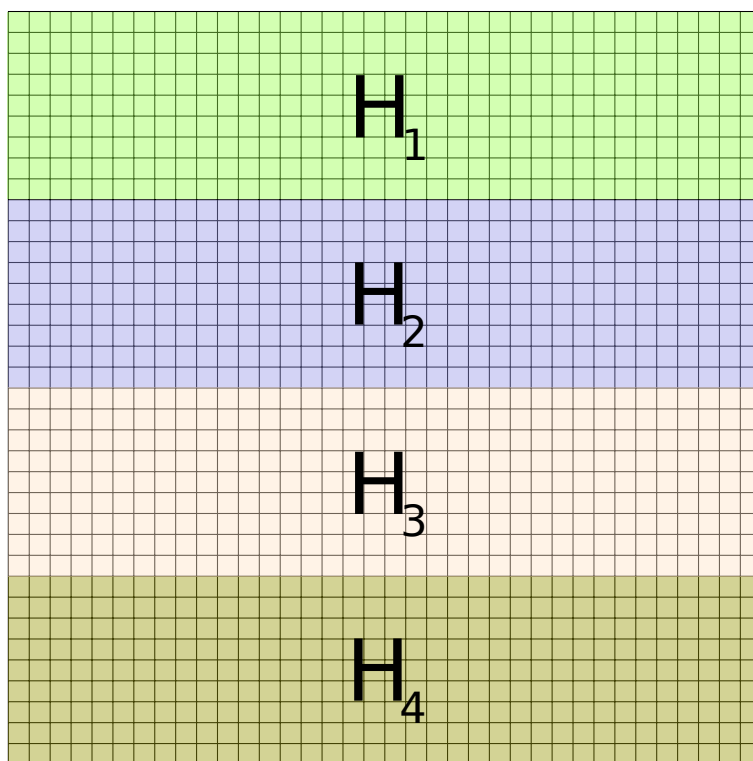
$$\begin{aligned}
 \sum_1^n 1 + \sum_{i=1}^n |w_2^i| &= n + \sum_{i=1}^n 2i + 1 \\
 &= n + \frac{n^2 + 3n}{2} \\
 &= \frac{n^2 + 5n}{2}.
 \end{aligned} \tag{3.24}$$

Iz enačb 3.9, 3.11, 3.23 in 3.24 lahko vidimo, da je največji člen n^2 . Tako dobimo časovno zahtevnost $O(n^2) = O(N)$. Časovna zahtevnost za celotno sliko je v najslabšem primeru potem $O(N * N) = O(N^2)$

3.6 Večnitno procesiranje

Vsi opisani algoritmi transformacije domnevnih razdalj v poglavju 3 spadajo v tako imenovane “sramotno paralelne” (ang. “embarrassingly parallel”) algoritme, kjer za izračun katerekoli vrednosti ne potrebujemo vedeti prejšnjega stanja. Posledično ne utrpimo posledic Amdahlovega zakona, ki nam postavi zgornjo mejo paralelnosti algoritma glede na količino odvisnih stanj. Opisane algoritme bi lahko pognali hkrati na N procesorjih, kjer bi vsakemu procesorju dodelili svoj unikatni položaj. S tem bi se algoritem zaključil v $O(N)$ času. Ker imamo opravka s slikami, je N zelo velik že pri manjših slikah in za to strategijo hitro potrebujemo superračunalnik z več milijon jedri, da bi maksimalno paralelizirali izvajanje algoritma.

Kljub temu da nimamo na razpolago superračunalnika, bomo izkoristili moč moderne več jedrne procesorje, katera ima H niti. Vhodno sliko S preprosto razdelimo na H delov ter vsaki niti dodelimo svoj kos slike, kot to prikazuje slika 3.6. Opisani algoritem v podpoglavju 3.4 je primeren za več nitno procesiranje, vendar moramo nato biti pozorni na dodatno pomnilniško zahtevnost. Vsaka nit potrebuje za hranitev vrednosti prejšnjega okna $w(p_o)$ dodatnega $O(8n) = O(8\sqrt{N}) = O(\sqrt{N})$ pomnilniškega prostora, kar nam pri H nitih da pomnilniško zahtevnost $O(H\sqrt{N})$.



Slika 3.6: Ena izmed možnih delitev slike, kadar uporabimo štiri niti.

Poglavje 4

Rezultati

Testiranje metode smo izvedli na osebni računalniku s specifikacijami:

- procesor Intel 3570k z najvišjo frekvenco delovanja 3800MHz,
- 32GB DDR3 pomnilnika,
- 250GB SSD.

Algoritem smo implementirali v programskem jeziku C++, za prevajalnik pa smo uporabili GNU GCC (verzija 6.1.1). Pri prevajanju izvorne kode smo uporabili sledeče nastavitve:

```
-fPIC -O2 -std=gnu++14 -flto
```

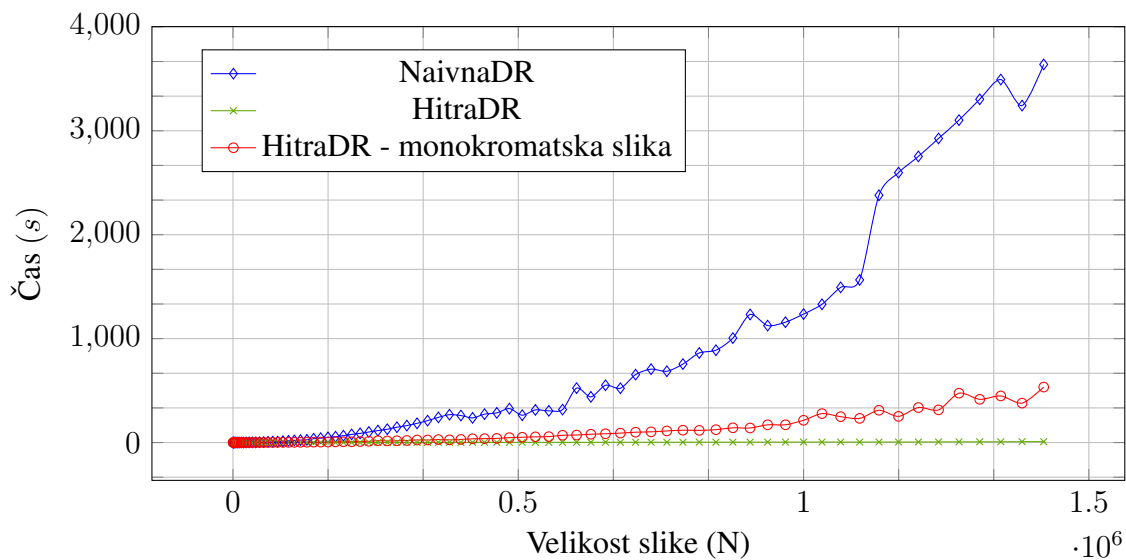
Za namene kontrolirane izvedbe testov časovne zahtevnosti smo ustvarili serijo kvadratnih slik. Stranice slik so bile znotraj območja $n \in [1, 1200]$. Slike so bile ustvarjene s podvzorčenjem dovolj velike izvorne slike. Za testiranje smo uporabili izboljšan naiven algoritem, opisan v podpoglavju 3.2 in ga v kontekstu teh testov imenujmo NaivnaDR ter algoritem iz podpoglavja 3.4 s popolnoma razčlenjenim oknom, ki ga imenujmo HitraDR

4.1 Meritve

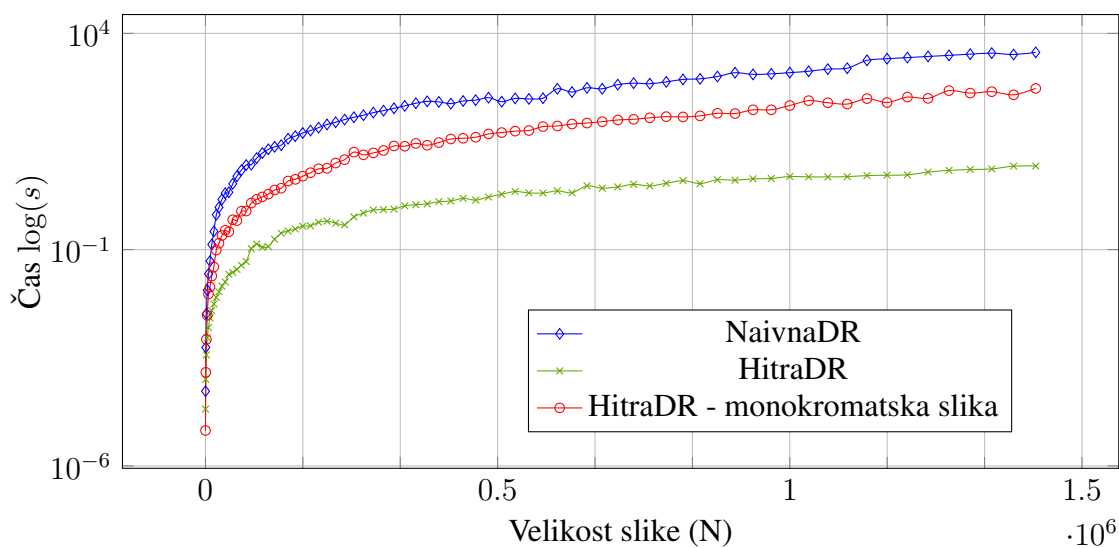
Tabela 4.1 in graf na sliki 4.1 prikazujeta rezultate časovnih meritev predstavljenih algoritmov. V grafu na sliki 4.2 vidimo iste podatke, vendar je graf logaritmiran po časovni osi, s čimer želimo boljše predstaviti razlike med algoritmi.

Velikost slike (<i>N</i>)	NaivnaDR (s)	HitraDR (s)	HitraDR monokromatska slika (s)
10,816	0.13	0.00	0.02
28,224	1.45	0.01	0.21
78,400	9.23	0.11	1.20
87,616	13.05	0.13	1.46
238,144	101.60	0.38	12.06
254,016	115.03	0.57	17.97
341,056	210.76	1.03	24.58
553,536	305.11	2.05	56.80
760,384	686.10	2.97	111.65
906,304	1,232.43	4.02	140.01
1,132,096	2,379.59	5.15	310.60
1,166,400	2,597.37	5.26	251.72
1,201,216	2,753.45	5.35	336.70
1,236,544	2,926.26	6.22	314.50
1,308,736	3,303.38	7.14	416.07
1,420,864	3,638.55	8.57	533.78

Tabela 4.1: Izpiski izvajalnih časov pri različnih velikosti slik.



Slika 4.1: Izvajalni časi pri različnih velikosti slik.

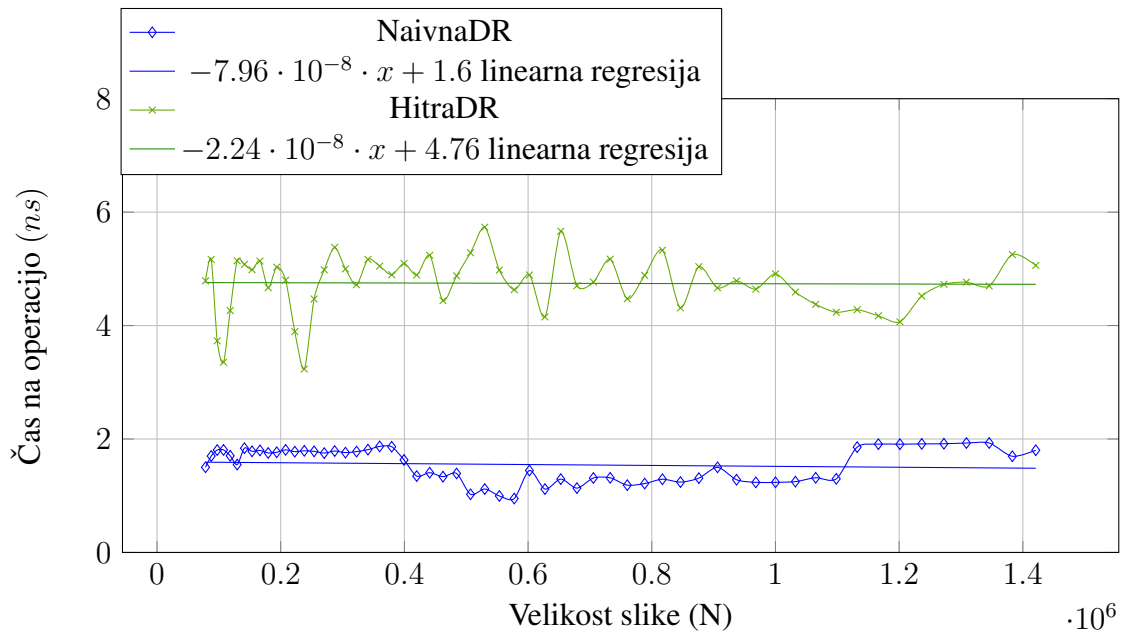


Slika 4.2: Izvajalni časi pri različnih velikostih slik, kjer je časovna os logaritmirana.

4.2 Potrditev pričakovane časovne zahtevnosti

Z namenom potrditve obeh teoretičnih časovnih zahtevnosti smo meritve normalizirali. To naredimo tako, da meritve časov delimo s teoretično časovno zahtevnostjo algoritmov NaivnaDR in HitraDR, ki sta $O(N^2)$ in $O(N^{1.5})$. Kot rezultat dobimo povprečno število nanosekund, ki jih potrebujemo da opravimo operacijo, če je le teh $O(N^2)$ in $O(N^{1.5})$. Nad normali-

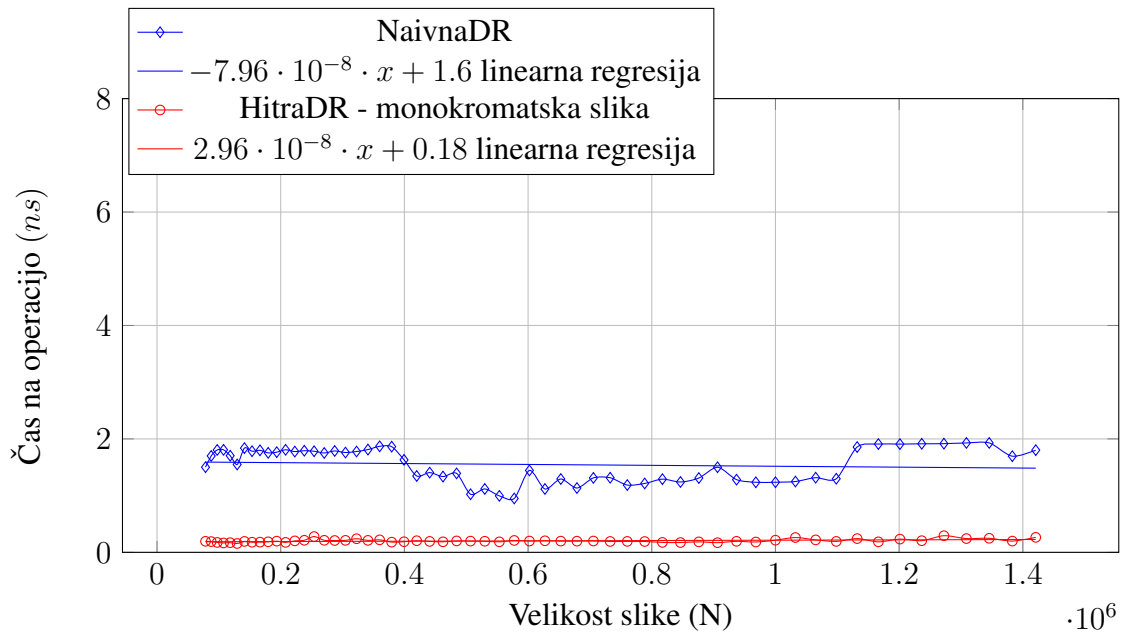
ziranimi podatki izvedemo linearno regresijo, kjer kot izhod dobimo premici, ki sta skoraj paralelni z osjo x. To potrjuje dejstvo, da se povprečni čas za operacijo ne spreminja glede na velikost vhoda, kar posledično dokazuje pravilnost ocenjene teoretične časovne zahtevnosti. Normalizirane podatke ter linearno regresijo prikazuje graf na sliki 4.3.



Slika 4.3: Povprečen čas za operacijo.

4.3 Potrditev časovne zahtevnosti v najslabšem primeru

V podpoglavju 3.5.2 smo določili, da je teoretična časovna zahtevnost v primeru monokromatske slike $O(N^2)$. V ta namen ponovno delimo meritve s teoretično časovno zahtevnostjo ter nad temi podatki izvedemo linearno regresijo. Rezultate lahko vidimo v grafu na sliki 4.4, ki potrjujejo teoretično časovno zahtevnost v najslabšem primeru.



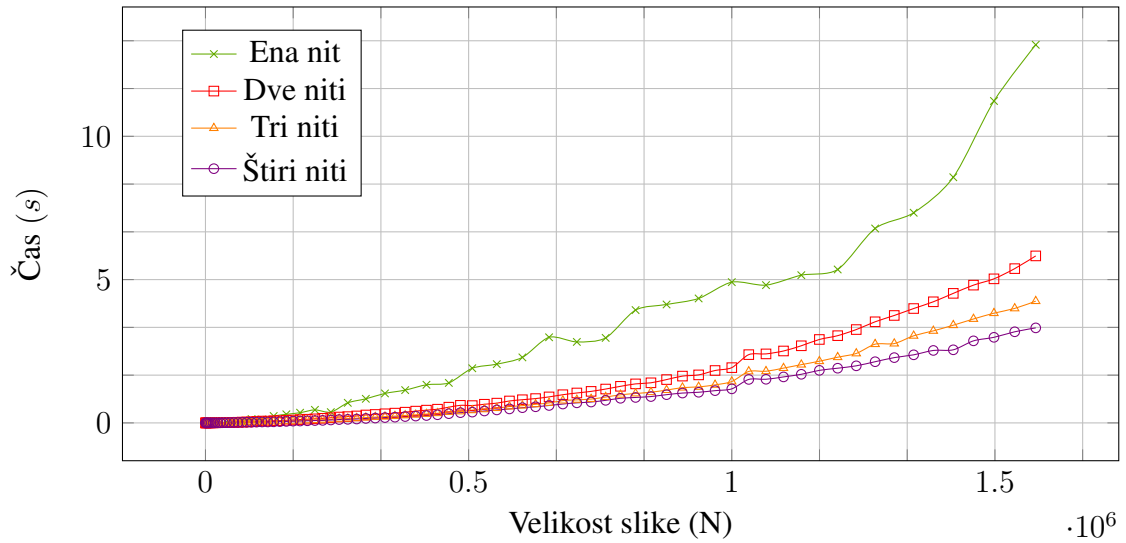
Slika 4.4: Povprečen čas za operacijo z monokromatsko sliko.

4.4 Meritve z uporabo večnitnost

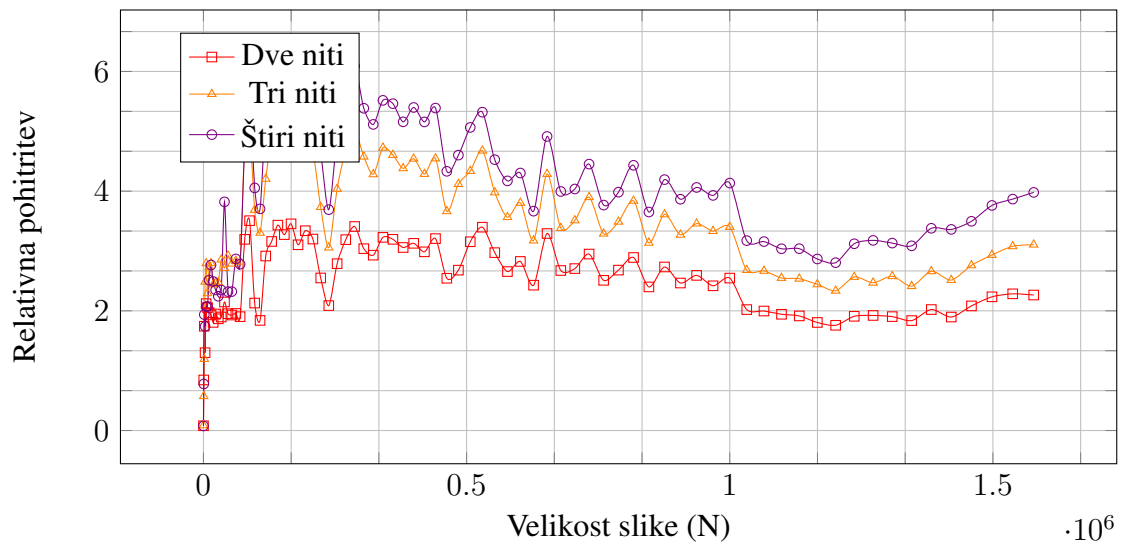
Z meritvami večnitne implementacije želimo preučiti vpliv tovrstnih pohitritev na izvajalni čas. V grafu na sliki 4.5 so prikazane meritve metode HitraDR, med tem ko slika 4.6 prikazuje relativno pohitritev glede na eno nitne meritve. Povprečne pohitritve prikazuje tabela 4.2, pri čemer so vrednosti izračunane iz prikazanih podatkov grafa na sliki 4.6. Iz tabele lahko vidimo, da je pohitritev linearna, glede na število uporabljenih niti. To potrjuje, da je algoritem “sramotno paralelen”.

Število niti	Povprečna pohitritev
1	1
2	2.457
3	3.419
4	3.936

Tabela 4.2: Povprečna pohitritev glede na število uporabljenih niti.



Slika 4.5: Primerjava hitrosti kadar uporabimo eno, dve, tri ali štiri niti.



Slika 4.6: Relativna primerjava pohitritve glede na izvajalni čas ene niti.

Velikost slike (N)	Število uporabljenih niti	Poraba pomnilnika (B)	Razlika (B)
262,144	1	4,602,664	–
–	2	4,603,624	960
–	3	5,155,544	551,920
–	4	5,156,560	1,016
1,048,576	1	24,063,640	–
–	2	26,213,024	2,149,384
–	3	28,362,424	2,149,400
–	4	30,511,864	2,149,440

Tabela 4.3: Poraba pomnilnika glede na število uporabljenih niti.

4.5 Meritve pomnilniške zahtevnost

Z meritev pomnilniške zahtevnosti smo uporabili orodje valgrind (verzija 3.11). Meritve smo opravili glede na različno število niti (glej tabelo 4.3). Pri manjših vhodnih slikah ni bilo možno konsistentno izmeriti pomnilniške porabe, saj je bil izvajalni čas prekratek da bi opravili zanesljive meritve. Ko smo povečali velikost vhodne slike je orodje pravilno zaznalo povečano porabo pomnilnika, kjer lahko vidimo da za vsako dodatno nit porabimo $2N$ zlogov (ang. bytes - B) dodatnega pomnilnika. To je več kot bi pričakovali iz teoretičnih rezultatov, kar kaže na slabo pomnilniško optimizirano implementacijo algoritma. Vseeno pa je po pričakovanjih rast pomnilniške zahtevnosti linearna glede na število uporabljenih niti.

Poglavje 5

Sklep

V magistrskem delu smo razvili pohitren algoritem transformacij domnevnih razdalj. Pohitritev je dosežena z zmanjšanjem števila dostopov do same slike. V razvoju smo ocenili tudi teoretično časovno zahtevnost algoritma. Izkazalo se je, da ima razvita metoda dve časovni zahtevnosti in sicer pričakovano, ki je enaka $O(N^{1.5})$, ter najslabšo, ki je enaka $O(N^2)$. Najslabša časovna zahtevnost nastopi v primeru monokromatskih slik. V ostalih primerih pa postane relevantna pričakovana časovna zahtevnost.

Razvito metodo smo implementirali v programskem jeziku C++ in jo uporabili za izvedbo meritev v potrditev teoretične časovne zahtevnosti. Z rezultati smo potrdili obe teoretični časovni zahtevnosti ter potrdili linearno pohitritev z uporabo večnitnosti. Meritve porabe pomnilnika žal niso potrdile teoretične ocene, kar nakazuje na slabo implementacijo iz vidika pomnilniške učinkovitosti.

Literatura

- [1] G. J. Grevera, “Distance transform algorithms and their implementation and evaluation,” in *Deformable Models*, pp. 33–60, Springer, 2007.
- [2] O. Cuisenaire, “Distance transformations: fast algorithms and applications to medical image processing,” 1999.
- [3] F. Y. Shih and C. C. Pu, “A skeletonization algorithm by maxima tracking on euclidean distance transform,” *Pattern Recognition*, vol. 28, no. 3, pp. 331–341, 1995.
- [4] R. Jarvis, “Distance transform based path planning for robot navigation,” *Recent Trends in Mobile Robots*, vol. 11, pp. 3–31, 1993.
- [5] S. Beucher, “Numerical residues,” *Image and Vision Computing*, vol. 25, pp. 405–415, 2007.
- [6] H. Breu, J. Gil, D. Kirkpatrick, and M. Werman, “Linear time euclidean distance transform algorithms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 5, pp. 529–533, 1995.
- [7] R. Fabbri, L. D. F. Costa, J. C. Torelli, and O. M. Bruno, “2D Euclidean distance transform algorithms: A comparative survey,” *ACM Computing Surveys*, vol. 40, pp. 1–44, 2008.
- [8] C. Ronse, “Why mathematical morphology needs complete lattices,” *Signal processing*, vol. 21, no. 2, pp. 129–154, 1990.
- [9] “Totally ordered set.” <http://mathworld.wolfram.com/TotallyOrderedSet.html>. Dostopno dne: 2016.08.03.

- [10] P. Soille, *Morphological Image Analysis: Principles and Applications*. Springer, 2nd ed., 2004.
- [11] H. J. A. M. Heijmans, "Theoretical aspects of gray-level morphology," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 568–582, 1991.



Univerza v Mariboru



Fakulteta za elektrotehniko,
računalništvo in informatiko

IZJAVA O AVTORSTVU

Spodaj podpisani/-a

Danijel Žlaus

z vpisno številko

E5000500

sem avtor/-ica diplomskega dela z naslovom:

Pohitritev transformacije domnevnih razdalj

(naslov diplomskega dela)

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek)

doc. dr. Domen Mongus

in somentorstvom (naziv, ime in priimek)

- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela.
- soglašam z javno objavo elektronske oblike diplomskega dela v DKUM.

V Mariboru, dne 22.08.2016

Podpis avtorja/-ice:



Univerza v Mariboru



Fakulteta za elektrotehniko,
računalništvo in informatiko

IZJAVA O USTREZNOSTI ZAKLJUČNEGA DELA

Podpisani mentor :

Domen Mongus

(ime in priimek mentorja)

in somentor (eden ali več, če obstajata):

(ime in priimek somentorja)

Izjavljam (-va), da je študent

Ime in priimek: Danijel Žlaus

Vpisna številka: E5000500

Na programu: Računalništvo in informacijske tehnologije (MAG)

izdelal zaključno delo z naslovom:

Pohitritev transformacije domnevnih razdalj

(naslov zaključnega dela v slovenskem in angleškem jeziku)

Optimization of Quasi Distance Transform

v skladu z odobreno temo zaključnega dela, Navodilih o pripravi zaključnih del in mojimi (najinimi oziroma našimi) navodili.

Preveril (-a, -i) in pregledal (-a, -i) sem (sva, smo) poročilo o plagiatstvu.

Datum in kraj:

22.08.2016, Maribor

Podpis mentorja:

Datum in kraj:

Podpis somentorja (če obstaja):

Priloga:

- Poročilo o preverjanju podobnosti z drugimi deli.



Univerza v Mariboru



Fakulteta za elektrotehniko,
računalništvo in informatiko

**IZJAVA O ISTOVETNOSTI TISKANE IN ELEKTRONSKE VERZIJE ZAKLJUČNEGA
DELA IN OBJAVI OSEBNIH PODATKOV DIPLOMANTOV**

Ime in priimek avtorja-ice: Danijel Žlaus

Vpisna številka: E5000500

Študijski program: Računalništvo in informacijske tehnologije (MAG)

Naslov zaključnega dela: Pohitritev transformacije domnevnih razdalj

Mentor: doc. dr. Domen Mongus

Somentor: _____

Podpisani-a Danijel Žlaus izjavljam, da sem za potrebe arhiviranja oddal elektronsko verzijo zaključnega dela v Digitalno knjižnico Univerze v Mariboru. Zaključno delo sem izdelal-a sam-a ob pomoči mentorja. V skladu s 1. odstavkom 21. člena Zakona o avtorskih in sorodnih pravicah dovoljujem, da se zgoraj navedeno zaključno delo objavi na portalu Digitalne knjižnice Univerze v Mariboru.

Tiskana verzija zaključnega dela je istovetna z elektronsko verzijo elektronski verziji, ki sem jo oddal za objavo v Digitalno knjižnico Univerze v Mariboru.

Zaključno delo zaradi zagotavljanja konkurenčne prednosti, varstva industrijske lastnine ali tajnosti podatkov naročnika: _____ ne sme biti javno dostopno do _____ (datum odloga javne objave ne sme biti daljši kot 3 leta od zagovora dela).

Podpisani izjavljam, da dovoljujem objavo osebnih podatkov, vezanih na zaključek študija (ime, priimek, leto in kraj rojstva, datum zaključka študija, naslov zaključnega dela), na spletnih straneh in v publikacijah UM.

Datum in kraj: 22.08.2016 Podpis avtorja-ice: 

Podpis mentorja: _____
(samo v primeru, če delo ne sme biti javno dostopno)

Podpis odgovorne osebe naročnika in žig: _____
(samo v primeru, če delo ne sme biti javno dostopno)