

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Simon Isaković

**REZANJE ŠIVOV  
SEAM CARVING**

DIPLOMSKO DELO  
NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Mentor: doc. dr. Patricio Bulić

Ljubljana, 2009



Št. naloge: 00418/2008

Datum: 15.11.2008

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **SIMON ISAKOVIĆ**

Naslov: **REZANJE ŠIVOV**  
**SEAM CARVING**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija

Tematika naloge:

Ocenite primernost metode rezanja šivov za prilagajanja velikosti slik na majhnih zaslonih. Metodo implementirajte v okolju Java Platform, Micro Edition ter preizkusite na emulatorju.

Mentor:

doc. dr. Patricio Bulić



Dekan:

prof. dr. Franc Solina

Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.

Namesto te strani **vstavite** original izdane teme diplomskega dela s podpisom mentorja in dekana ter žigom fakultete, ki ga diplomant dvigne v študentskem referatu, preden odda izdelek v vezavo!



# IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a Simon Isaković,

z vpisno številko 63990215,

sem avtor/-ica diplomskega dela z naslovom:

Rezanje šivov

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom doc. dr. Patricia Bulića
- 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 zbirki "Dela FRI".

V Ljubljani, dne 15.05.2009

Podpis avtorja/-ice:



# Zahvala

Rad bi se zahvalil svojemu mentorju, staršem, predavateljem fakultete ter profesionalnim kolegom, brez katerih to delo ne bi bilo možno.



# Kazalo

<b>Povzetek</b>	<b>1</b>
<b>Abstract</b>	<b>2</b>
<b>1 Uvod</b>	<b>3</b>
<b>2 Predstavitev</b>	<b>5</b>
<b>3 Algoritem</b>	<b>10</b>
3.1 Energetska analiza slike . . . . .	10
3.2 Računanje najboljših možnih šivov . . . . .	11
3.3 Iskanje najboljšega možnega šiva . . . . .	14
3.4 Rezanje slike po šivu . . . . .	15
3.4.1 Vrstni red rezov . . . . .	16
<b>4 Okolje</b>	<b>18</b>
<b>5 Sklepne ugotovitve</b>	<b>20</b>
<b>Seznam slik</b>	<b>23</b>
<b>Literatura</b>	<b>24</b>

# Seznam uporabljenih kratic in simbolov

- EV- energetska vrednost
- MRŠ- metoda rezanja šivov

# Povzetek

Namen tega dela je bil določitev in ocena primernosti metode rezanja šivov prilagajanju velikosti slik majhnim zaslonom. Ker pri teh ne gre samo za omejitve velikosti, marveč tudi različne dimenzije, je bila izbrana metoda rezanja šivov (vir [1]), pri kateri režemo navpične in vodoravne šive povezanih pik, dokler se slika ne prilega željeni velikosti in razmerju. Glavni cilj metode je, da sliko prilagodimo s čim manjšimi vidnimi spremembami v sami sliki tako, da odstranjujemo manj vidne pike, ne da bi ob tem pokvarili okoliških.

Končni rezultati same metode so bili močno odvisni od same slike. Tako pri slikah, kjer je bilo potrebno rezati preveliko število pik, kot tudi pri tistih s preveliko kompleksnostjo vsebine, je bil končni rezultat vidno spremenjen, na slikah z močnim centralnim fokusom in enakomernim ozadjem pa dober.

Metoda je bila preizkušena na emulatorju telefona z zaslonom srednje velikosti ter fizičnem telefonu z manjšim zaslonom in se izkazala primerna za uporabo na napravah srednjih velikosti zaslonov, kot na primer novejših pametnih telefonih, za starejše mobilne telefone z majhnimi zasloni pa je uporaba praktična samo za manjše slike, tako zaradi težav z odstranjevanjem prevelikega števila pik kot zaradi počasnosti.

## Ključne besede:

prilagajanje velikosti slik, rezanje šivov, velikost zaslona, dimenzije zaslona, pametni telefoni

# Abstract

The goal of this work was to gauge the effectiveness of various image resizing methods when used on devices with small displays. Since the problem here is not only small size but also varying display dimensions the method chosen was seam carving, where the image is resized by removing horizontal and vertical seams of connected pixels until the image fits. The aim of the method is to resize the image with minimal visual impact.

The end results greatly depended on the image itself. Both with images where we needed to remove too many pixels and those with visually complex content were visibly altered, whereas images with a strong central focus and subdued backgrounds gave us better results).

The method turned out to be useful for devices with displays of medium sizes, such as smart phones, whereas on older mobile phones with smaller displays the results were only practical for smaller images, both due to the number of pixels removed and because of the (lack of) speed of the device itself.

## Key words:

image resizing, seam carving, display size, display dimensions, smart phones

# Poglavje 1

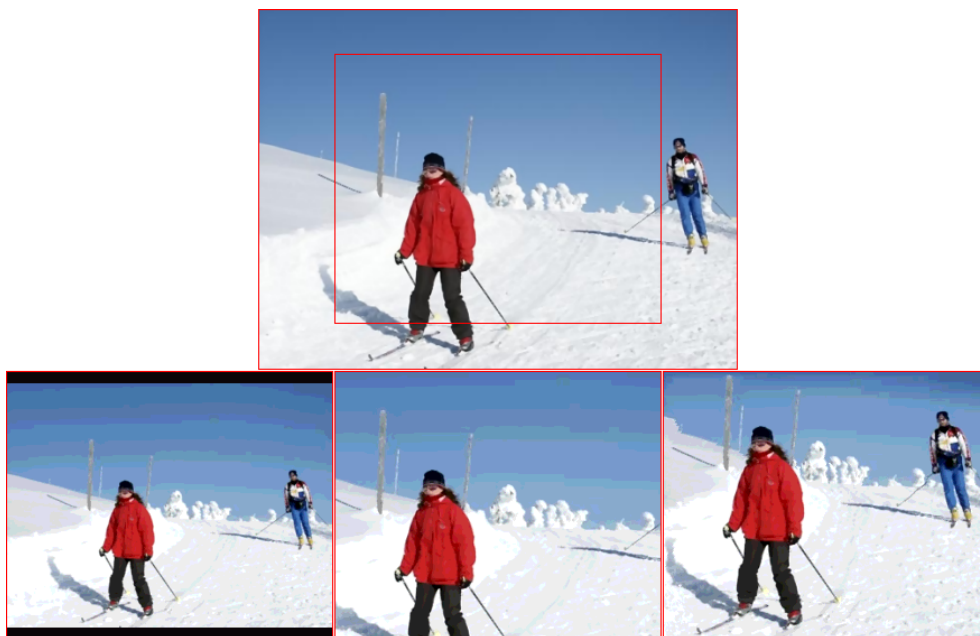
## Uvod

Navkljub tehnološkim napredkom elektronike v zadnjih desetletjih ostajajo določene omejitve povsem fizične narave, ki jih omenjeni napredek ne more odpraviti. Ena od teh je velikost prenosnih naprav, ki morajo ostati dovolj majhne zaradi praktičnih razlogov, njihov zaslon pa mora kljub temu biti sposoben prikazati zadostno količino informacij, da naprava opravlja svojo nalogo. Najbolj vidna kategorija naprav s to omejitvijo so prenosni telefoni, katerih izdelovalci se že dalj časa spogledujejo z različnimi multimedijskimi sposobnostmi, ob strani pa imamo tudi dlančnike ter vedno bolj popularne ultra lahke prenosnike, pri katerih je poglobljena prenosnost in priročnost in ne zmogljivost.

Velikost zaslona je kritičnega pomena pri prikazu slikovnega dela multimedijskih informacij, tako videa kot slik, in tradicionalne metode prilagajanja te vsebine so le delno uporabne - pri metodi zmanjševanja celotno sliko z enakomernim odstranjevanjem in mešanjem pik pomanjšamo na velikost, pri kateri obdržimo razmerje med velikostmi stranic in je celotna vidna na zaslonu (slika 1.1 levo), pri rezanju pa izberemo del slike, ki ga želimo obdržati, in okoli njega odrežemo vse, kar ni vidno na zaslonu (slika 1.1 sredina). Slabost prve metode je neizkoriščenost zaslona, saj pri slikah z razmerji drugačnimi od razmerja zaslona ostajajo deli tega prazni, slabost druge pa izguba velikih kosov slike. Obema je skupen tudi problem, da celotno sliko obravnavata kot enakovredno, zaradi česar lahko pomembne dele preveč pomanjšamo ali celo izgubimo.

Metoda rezanja šivov (MRŠ) se te problematike loti tako, da analizira vsebino same slike in namesto enakomernega odstranjevanja informacij pri zmanjševanju, odstranjuje tisto informacijo, ki jo oceni za manj pomembno (slika 1.1 desno). Sliko najprej analiziramo tako, da vsaki piki določimo vred-

nost glede na to, kako močno se razlikuje od svojih sosedov, nato pa sliko zmanjšujemo z rezanjem šivov (vodoravno ali navpično povezanih sosednjih pik) najnižje vrednosti, torej šivov pik, ki se od svojih sosedov kar najmanj razlikujejo.



Slika 1.1: Primerjava prilagajanja velikosti slike z zmanjševanjem (levo), rezanjem (sredina) in MRŠ (desno) (originalna slika je iz [2]).

# Poglavje 2

## Predstavitev

Osnovna ideja MRŠ je, da pri zmanjševanju slike iz nje odstranjujemo tiste pike, ki so najmanj pomembne. Le-te določimo s primerno analizo, ki za vsako piko oceni, kako vidna je človeškemu očesu (ter s tem, kako moteča bi bila njena odstranitev) - takih analiz je več vrst, od katerih nobena ni optimalna za vse primere. Ker je namen naloge izvedba in ocena primernosti same metode rezanja, smo se zato odločili za eno samo vrsto določanja vrednosti pik. V nadaljevanju bo ta vrednost imenovana kot energetska vrednost (EV) pike.

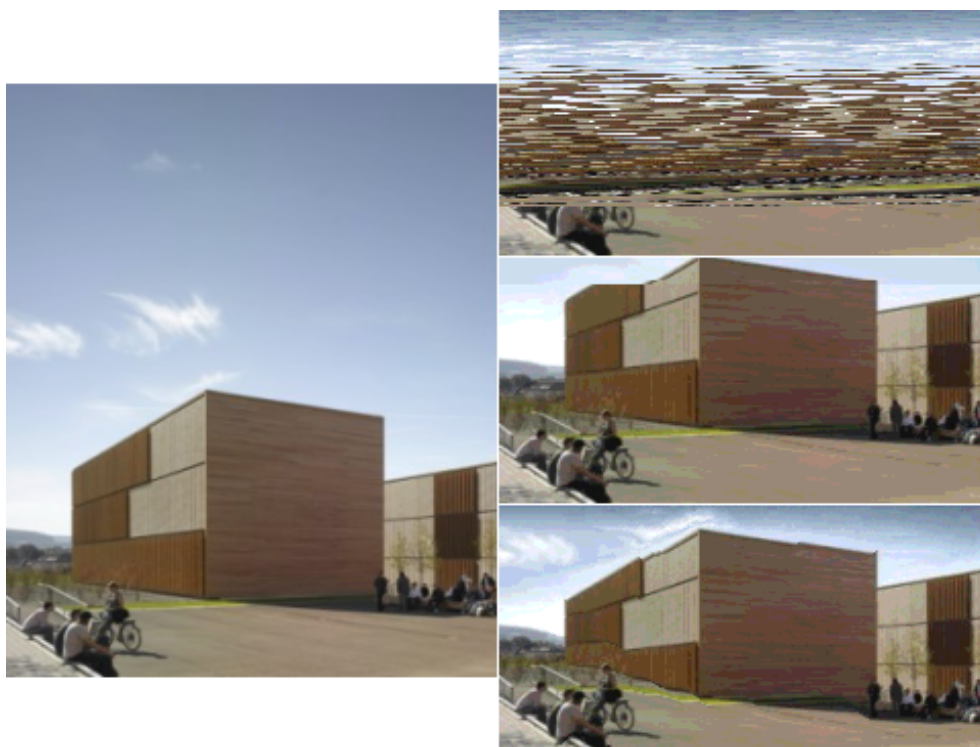
Ko je slika analizirana, lahko pričnemo z odstranjevanjem odvečnih pik. Najbolj očiten način bi bil, da preprosto preračunamo željeno velikost in odstranimo (*celotno – željeno*) število pik z najnižjo EV (slika 2.1 zgoraj). Dobimo energetsko optimalen rezultat, vendar izvedba zmanjševanja na tak način popači pravokotno naravo slike, saj v različnih vrsticah odstranimo različno število pik. Problemu z zamiki se lahko izognemo tako, da odstranimo celotne stolpce/vrstice najnižje energetske vrednosti, vendar na ta način ne dobimo več dobrih energetskih rezultatov, dobimo pa tudi opazne zamike v sliki sami (slika 2.1 sredina).

Pri MRŠ se problema lotimo tako, da odstranjujemo pike na način, ki čim manj vpliva na ostale dele slike, vendar nam še vedno da dober rezultat EV (slika 2.1 spodaj). Namesto izbire pik z najnižjo EV izbiramo vodoravne in navpične šive, katerih EV vsebovanih pik je najnižja.

V nadaljevanju bomo podali nekaj osnovnih definicij pojmov, ki jih bomo uporabljali v nadaljevanju tega dela.

**Definicija 1 (*Piksel.*)** *Piksel*  $(i, j)$  je slikovni element v vrstici  $i$  in stolpcu  $j$ .

**Definicija 2 (*4-sosed.*)** Dve slikovni piki oz. dva piksla sta 4-soseda, če si delita rob. Vsi 4-sosedi piksla  $(i, j)$  so:  $(i-1, j)$ ,  $(i+1, j)$ ,  $(i, j-1)$ , in  $(i, j+1)$ .



Slika 2.1: Primerjava rezanja pik z najnižjo EV (zgoraj), rezanja vrstic z najnižjo vsoto EV (sredina) in rezanja z MRŠ (spodaj) (originalna slika je iz [3]).

**Definicija 3 (8-sosed.)** Dve slikovni piki oz. dva piksla sta 8-soseda, če si delita vsaj en vogal. Vsi 8-sosedi piksla  $(i, j)$  so:  $(i - 1, j - 1)$ ,  $(i - 1, j)$ ,  $(i - 1, j + 1)$ ,  $(i, j - 1)$ ,  $(i, j + 1)$ ,  $(i + 1, j - 1)$ ,  $(i + 1, j)$  ter  $(i + 1, j + 1)$ .

**Definicija 4 (Pot.)** Pot med dvema slikovnima pikama oz. pot med dvema piksloma  $(i_0, j_0)$ ,  $(i_n, j_n)$  je množica piksllov:

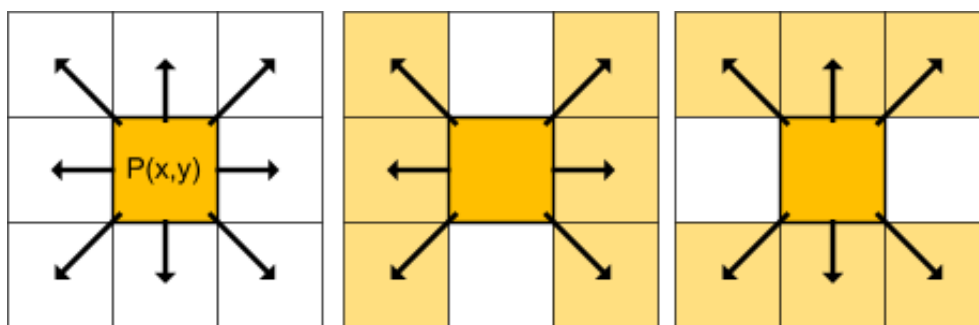
$$\{(i_0, j_0), (i_1, j_1), (i_2, j_2), \dots, (i_k, j_k), (i_{k+1}, j_{k+1}), \dots, (i_{n-1}, j_{n-1}), (i_n, j_n)\}$$

tako, da sta piksla  $(i_k, j_k)$  in  $(i_{k+1}, j_{k+1})$  soseda za vse  $k = 0 \dots n - 1$ .

**Definicija 5 (4-povezana pot.)** 4-povezana pot med dvema slikovnima pikama oz. med dvema piksloma je takšna pot pri kateri je relacija sosesčine 4-sosed.

**Definicija 6 (8-povezana pot.)** 8-povezana pot med dvema slikovnima pikama oz. med dvema piksloma je takšna pot pri kateri je relacija sosesčine 8-sosed.

Šiv bomo definirali kot 8-povezano pot, ki seže od ene stranice slike do nasprotne. Tako bomo definirali vodoravni in navpični šiv, pri čemer bomo v vsakem stolpcu oz. vrstici izbirali le en piksel, tj. upoštevali bomo le 8-sosede, kot je prikazano na sliki 2.2.



Slika 2.2: 8-povezane sosednost pik (levo), sosednost v vodoravnem šivu (sredina) in sosednost v navpičnem šivu (desno).

**Definicija 7 (Vodoravni šiv.)** Vodoravni šiv, ki ga označimo s  $s^h$  je 8-povezana pot od levega roba slike do desnega roba pri čemer v vsakem stolpcu izberemo le en piksel. Predpostavimo, da je v eni vrstici slike  $n$  pikslov. Potem je:

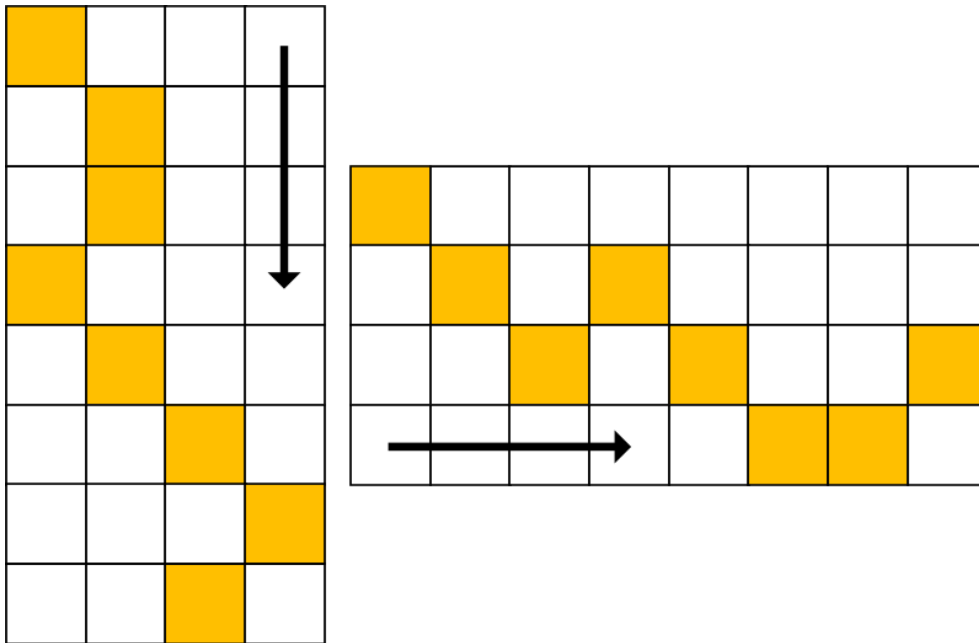
$$s^h = \{(i_k, j_k) : k = 1 \dots n, j = k, |i_k - i_{k-1}| \leq 1\} \quad (2.1)$$

**Definicija 8 (Navpični šiv.)** Navpični šiv, ki ga označimo s  $s^v$  je 8-povezana pot od zgornjega roba slike do spodnjega roba pri čemer v vsaki vrstici izberemo le en piksel. Predpostavimo, da je v enem stolpcu slike  $m$  pikslov. Potem je:

$$s^v = \{(i_k, j_k) : k = 1 \dots m, i = k, |j_k - j_{k-1}| \leq 1\} \quad (2.2)$$

Vodoravni in navpični šiv prikazuje slika 2.3. Sliko prilagajamo željeni velikosti tako, da iz nje režemo vodoravne in navpične šive. EV vrednost takega šiva je vsota EV njegovih pik in šiv z najmanjšo EV je vodoravni ali navpični šiv z najnižjo možno EV.

Na ta način sicer ne odstranimo pik z najnižjo možno energetske vrednostjo, nam pa omenjeni kompromis med odstranjevanjem pik nizke EV in omejevanjem zamikov preostalih pik slike kot rezultat pusti sliko s čim manjšimi vsebinskimi spremembami (slika 2.4).



Slika 2.3: Primer navpičnega (levo) in vodoravnega (desno) šiva.



Slika 2.4: Primer zaporednega rezanja navpičnih šivov, vrstni red od leve proti desni v gornji vrstici, nato spodnji.

# Poglavje 3

## Algoritem

Potek celotnega postopka lahko razdelimo na 4 korake, ki bodo bolj podrobno obravnavani v nadaljevanju:

- energetska analiza slike
- računanje najboljših možnih podšivov
- iskanje najboljšega možnega šiva
- rezanje slike po šivu

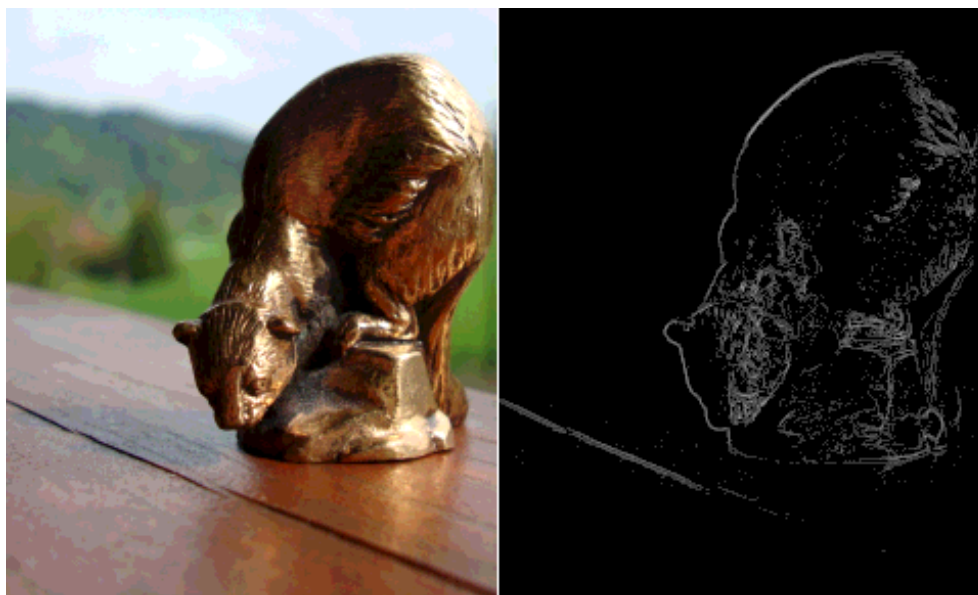
### 3.1 Energetska analiza slike

Za izbiro primernih rezov potrebujemo način, da pikam določimo vrednost, na podlagi katere se odločamo naprej. Osnovna ideja je, da režemo pike, katerih odstranitev bo najmanj opazna, zato je bil uporabljen preprost algoritem iskanja robov, ki piki določi vrednost glede na to, kako močno se razlikuje od pik okoli nje. Razlika dveh pik je pri tem definirana kot vsota absolutnih vrednosti razlik posameznih barvnih komponent (enačba ??). Ker je potrebno to analizo opraviti na vsaki piki, bolj kompleksni algoritmi niso primerni, saj je potrebno v najslabšem primeru po vsakem rezu sliko ponovno analizirati v celoti (kar se v naši izvedbi naloge tudi zgodi), kar naredi rezanje večjega števila šivov nepraktično (slika 3.1).

**Definicija 9** (*Barvne komponente.*) Vsako piko (piksel) slike lahko razdelimo na tri barvne komponente, rdečo ( $R$ ), zeleno ( $G$ ) in modro ( $B$ ), katerih vrednosti v naših slikah segajo od 0 (črna) do 255 (svetlo rdeča/zelena/modra).

**Definicija 10** (*Razlika med piksli.*) Naj bosta  $(x_a, y_a)$  in  $(x_b, y_b)$  dva piksli v sliki in naj bosta  $(R_a, G_a, B_a)$  in  $(R_b, G_b, B_b)$  vrednosti njunih barvnih komponent. Potem definiramo razliko med piksli:

$$(x_a, y_a) - (x_b, y_b) = |R_a - R_b| + |G_a - G_b| + |B_a - B_b|$$



Slika 3.1: Originalna slika (levo) in slika po opravljeni energetski analizi (desno).

Analiza, uporabljena v nalogi, za vsako piko določi kontrast (to je, kako vidna je) tako, da primerja njeno razliko med piko na njeni desni, piko pod njo ter diagonalne pike med njima, te sešteje ter primerno obteži z razdaljo med njimi. To vsoto nato delimo s številom primerjanih sosed, saj ob robovih število teh ni vedno enako (enačba 3.1). Primerjava z ostalimi sosedami je nepotrebna, saj smo razliko med našo piko in njimi upoštevali že pri njihovi analizi.

$$EV_{x,y} = \frac{((x, y) - (x, y + 1)) + ((x, y) - (x + 1, y)) + \frac{((x,y)-(x+1,y+1))}{\sqrt{2}}}{3} \quad (3.1)$$

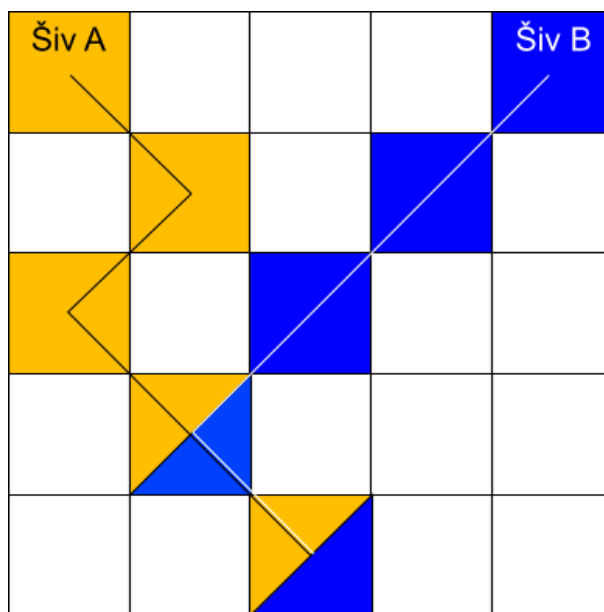
## 3.2 Računanje najboljših možnih šivov

Iskanje najboljšega možnega šiva, torej šiva katerega skupna vsota EV je najnižja možna, se je mogoče lotiti na več načinov. Pri vseh v tem koraku obrav-

navamo celoštevilsko polje, katerega vrednosti vsebujejo EV slike. Ker je razlika med vodoravnimi in navpičnimi šivi le v orientaciji slike, bodo obravnavani vodoravni, za druge pa sliko zavrtimo za 90 stopinj.

Najbolj preprost način je rekurzivna funkcija, ki v vsaki piki kliče sebe na pikah pod sabo in med vrnjenimi vrednostmi vzame najmanjšo, prišteje še svojo in jo vrne. Končni rezultat je, da nam funkcija prve pike vrne vsoto najmanjšega možnega šiva, ki se začne s to piko. To storimo na vseh pikah gornje vrstice in izberemo tisto z najmanjšo vsoto. Možnost izboljšave je, da sprti shranjujemo položaj najboljših pik, ki ga prepisemo, če v sledečem šivu najdemo boljšo izbiro.

Na žalost ta metoda ni primerna za večje slike, še posebej na mobilnih napravah znotraj navideznega okolja, saj je kompleksnost eksponentna ( $O(n^3)$ ) in hitro preraste zmogljivosti naprave. Problem je v dejstvu, da je velika večina primerjav nepotrebna, saj že pri analizi istega šiva vedno znova obravnavamo pike, do katerih smo že prišli po drugačni poti (slika 3.2).



Slika 3.2: Nepotrebna analiza pike, ki smo jo obdelali že v nekem prejšnjem koraku.

Problema se je bilo potrebno lotiti drugače. Iz definicije šiva (7, 2.2) je očitno, da je za vsako piko možen le en najboljši rezultat; lahko ima več podšivov z isto vsoto, vendar ne more istočasno imeti dve različni vsoti, ki sta najmanjši. Če je tako, potem za vsako piko velja, da je najmanjša energetska vrednost

njenih podšivov (tj. šivov, ki bi se začeli v tej piki) vsota energetske vrednosti minimalnega podšiva pod njo ter energetske vrednosti pike same. V nadaljevanju pogledjmo, kako analitično izračunamo energetske vrednosti podšivov posamezne slikovne pike.

**Definicija 11** Naj bo  $A$  2-dimenzionalno polje, dimenzij  $n \times m$ , za  $n, m \in \mathbb{N}$ . Elementi polja  $A$  predstavljajo energetske vrednosti pik in jih izrazim z  $A[x][y] = E_{x,y}$  za vse  $1 \leq x \leq n$  in  $1 \leq y \leq m$ .

**Definicija 12** Naj bo  $B$  2-dimenzionalno polje, dimenzij  $n \times m$ , za  $n, m \in \mathbb{N}$ . Element  $B[x][y]$  predstavlja energetsko vrednost navpičnega šiva, ki se začne v elementu  $B[x][y]$  in se konča v zadnji vrstici polja. To energetsko vrednost v elementu  $B[x][y]$  označimo z  $D_{x,y}$ .

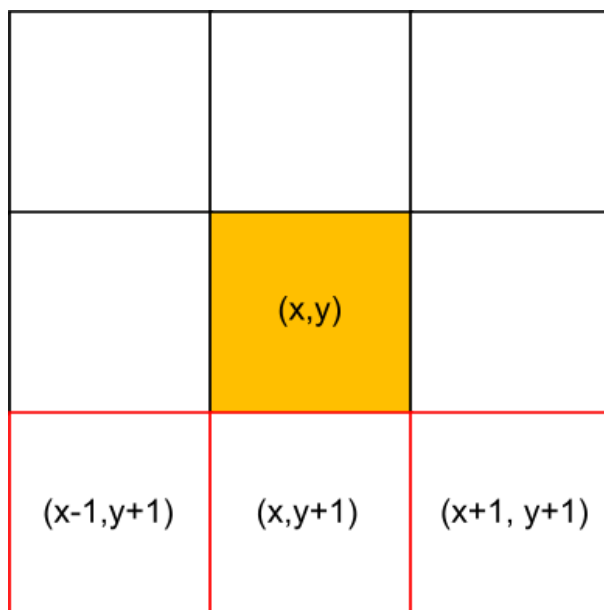
**Definicija 13** Naj bo  $A$  polje definirano v definiciji 11 in  $B$  polje definirano v definiciji 12. Potem elemente polja  $B$  računamo na naslednji način:

$$D_{x,y} = \begin{cases} E_{x,m} & ; y = m \\ E_{1,y} + \min(D_{1,y+1}, D_{2,y+1}) & ; x = 1 \\ E_{x,y} + \min(D_{x-1,y+1}, D_{x,y+1}, D_{x+1,y+1}) & ; 1 < x < n - 1 \\ E_{n,y} + \min(D_{n-1,y+1}, D_{n,y+1}) & ; x = n \end{cases} \quad (3.2)$$

Zaradi teh lastnosti lahko polje preračunamo s pomočjo dinamičnega programiranja, to je računanja optimalnega rezultata s serijo zaporednih korakov, kjer v vsakem uporabimo rešitev prejšnjega koraka za določanje rešitve tega. Končni rezultat je optimalna rešitev problema. V našem primeru vzamemo polje EV in ga preračunamo od spodaj navzgor. Na spodnji vrstici pik vsaka od njih že predstavlja najboljši možni podšiv, saj pod njimi ni drugih vrednosti - najmanjša možna vsota šiva dolžine ene pike je vrednost same pike. Premaknemo se vrstico višje in pri vsaki piki pogledamo tri pod njo, izberemo tisto z najnižjo vrednostjo ter ji prištejemo vrednost pike same (slika 3.4). To ponovimo vrstico za vrstico, dokler ni preračunano celo polje. Zaradi spominskih zahtev sproti prepisujemo kar polje EV, saj ga potrebujemo samo za izračun polja najboljših možnih podšivov, to je polja, kjer vrednost vsake pike predstavlja vsoto EV njenege najboljšega podšiva (slika 3.4, svetlejša barva predstavlja višjo EV).

Sedaj predstavimo postopek s katerim izračunamo vse elemente polja  $B$ , oz. poiščemo vse možne navpične šive:

**Algoritem 1** *Računanje vseh možnih navpičnih šivov.*



Slika 3.3:  $D_{x,y} = E_{x,y} + \min(D_{x-1,y+1}, D_{x,y+1}, D_{x+1,y+1})$ .

**Vhod:** Polje  $A$

**Izhod:** Polje  $B$

**Postopek:**

```

for (y=m; y>=1; y--) { // zacni v zadnji vrstici
  for (x=1; x<=n; x++) {
    B[x][y] = Dxy; // po definiciji 13
  }
}

```

### 3.3 Iskanje najboljšega možnega šiva

Ko imamo izračunano celotno polje vrednosti, določimo najboljši šiv tako, da v gornji vrstici poiščemo piko, katere vrednost je najnižja (saj je to vrednost

2	4	2	2+4	4+3	2+3
4	2	0	4+2	2+2	0+3
2	3	10	2	3	10

Slika 3.4: Energetske vrednosti pik v polju  $A$  (levo) in energetske vrednosti podšivov v polju  $B$  (desno).

najbolj optimalnega šiva te vrstice). Ko jo imamo, se preprosto sprehodimo po šivu navzdol, vedno izibrajoč najmanjšo od spodnjih pik, dokler nimamo celotnega reza (slika 3.5). Postopek podaja naslednji algoritem:

**Algoritem 2** *Izbira šiva z najmanjšo energetsko vrednostjo.*

**Vhod:** Polje  $B$

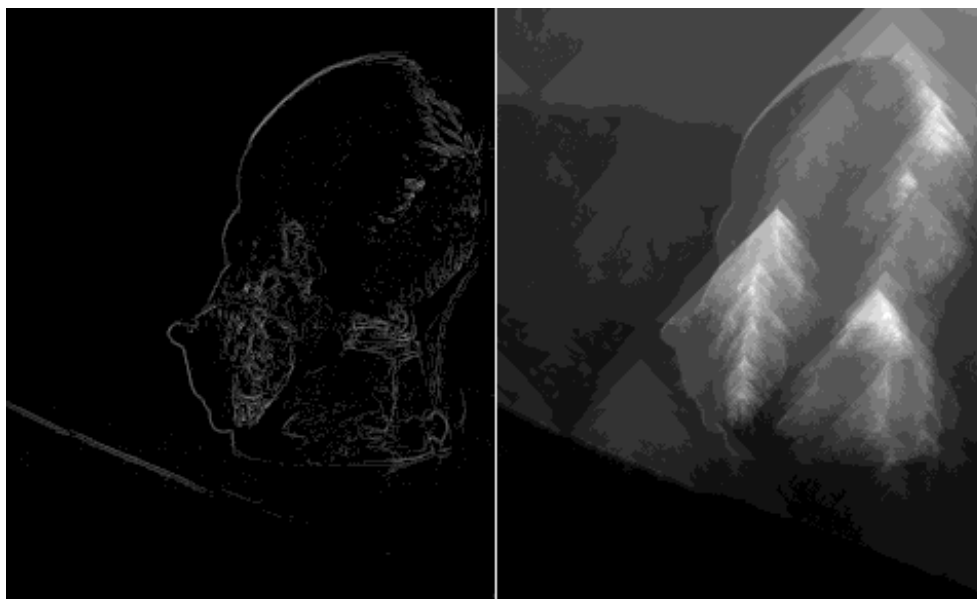
**Izhod:** Vertikalni šiv  $S$ , ki je linearno polje dolžine  $m$ .

**Postopek:**

1. V prvi vrstici polja  $B$  poišči element  $B[x][y]$  z najmanjšo vrednostjo  $D[x][y]$  in ga dodaj v šiv.
2. Ponavljaj za vse vrstice do predzadnje: V šiv dodaj  $B[x-1][y+1]$ , če je  $D_{x-1,y+1} = \min(D_{x-1,y+1}, D_{x,y+1}, D_{x+1,y+1})$ , oz.  $B[x][y+1]$ , če je  $D_{x,y+1} = \min(D_{x-1,y+1}, D_{x,y+1}, D_{x+1,y+1})$ , oz.  $B[x+1][y+1]$ , če je  $D_{x+1,y+1} = \min(D_{x-1,y+1}, D_{x,y+1}, D_{x+1,y+1})$ .

### 3.4 Rezanje slike po šivu

S tako določenim šivom lahko nato režemo originalno sliko. To počnemo tako, da sliko po odsekih ob šivu kopiramo kos ob kos na začetek slike same, saj si



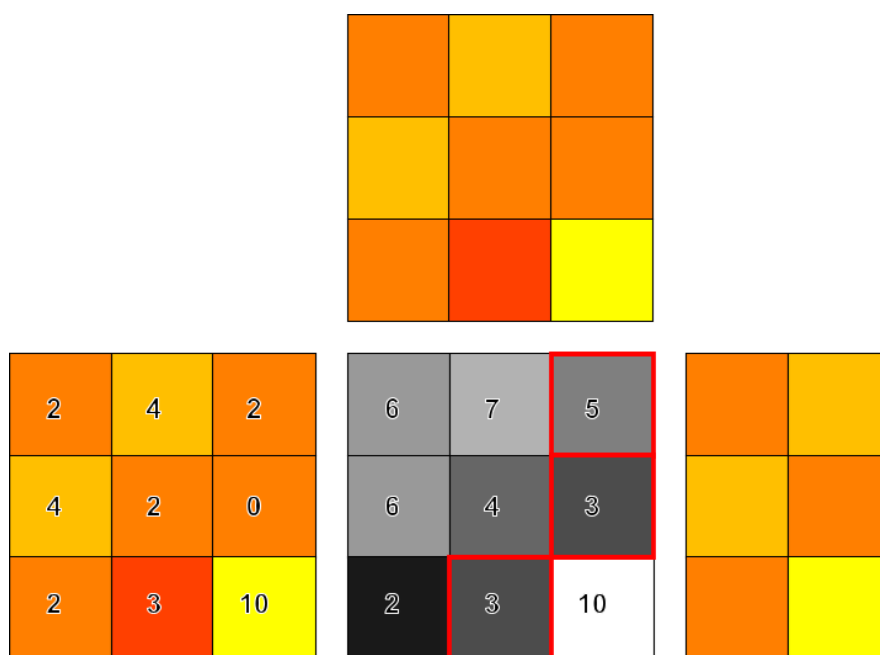
Slika 3.5: Prikaz celotne analize (levo) in njenih preračunanih podšivov.

ne moremo privoščiti vmesne kopije. Ker je celotna slika predstavljena kot eno samo linearno polje, lahko nepotrebna polja na koncu, ki so ostala kot rezultat rezanja, ignoriramo (slika 3.6).

Sedaj postopek rezanja lahko pričnemo znova. Iz razrezane slike generiramo novo polje EV preko starega, iz njega preračunamo polje najboljših možnih podšivov in določimo nov rez.

### 3.4.1 Vrstni red rezov

Kadar režemo sliko tako po vodoravnih kot navpičnih šivih, se lahko vprašamo, kako sliko rezati najprej? Ali je bolje rezati najprej navpični ali vodoravni šiv? Optimalno bi analizirali najboljši šiv pri obeh, delili rezultat z dolžino šiva in tako določili, kdo nam da boljši rezultat, vendar se je izkazalo, da na manj zmogljivejših napravah tu zadanemo hitrostno mejo, ki naredi ta pristop nepraktičen. V sami izpeljavi zato samo režemo najprej po eni, nato pa po drugi dimenziji.



Slika 3.6: Originalna slika (zgoraj), energetska analiza (levo), rezanje šiva iz slike (sredina) in končni rezultat (desno).

# Poglavje 4

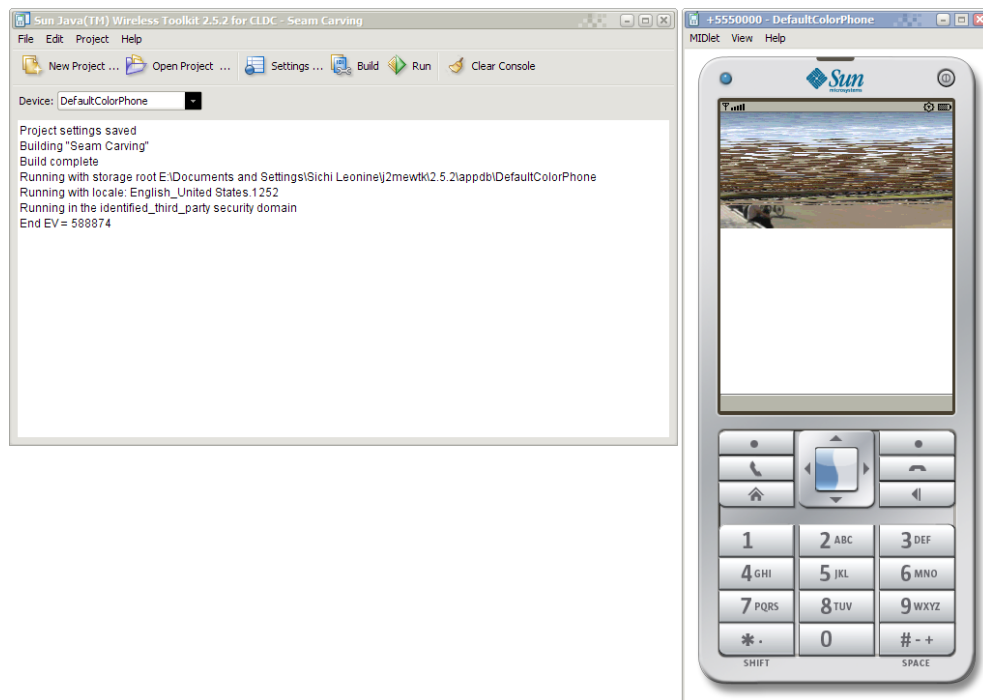
## Okolje

Ker je število različnih naprav z majhnimi zasloni in pripadajočimi operacijskimi sistemi veliko, sem se odločil za izvedbo algoritma v okolju Java Platform, Micro Edition (vir [4]). Dobre strani tega okolja so prenosnost in že obstoječe metode za nalaganje in prikaz slik, med eno glavnih omejitev za nalogo pa količina spomina, ki je na voljo programom tega okolja. Ta se na fizičnih napravah razlikuje, v emulatorju, kjer sem algoritem preizkušal, pa je bila omejena na 160KB. Prav tako je omejevitvev hitrost, saj program teče znotraj Java navideznega okolja, kar izvajanje upočasni.

Kodo samo sem pisal z urejevalnikom Vim (vir [5]), za prevajanje, grajenje paketov ter tek v emulatorju pa sem uporabljal okolje Wireless Toolkit (vir [6], slika 4.1), saj ta omogoča celotno pot razvoja, od prevajanja in razhroščevanja kode do grajenja in podpisovanja paketov, primernih za uporabo na primernih napravah. Za primerjalne teste sem kodo poganjal tudi na lastnem mobilnem telefonu.

Pri praktični izvedbi naloge so vrednosti shranjene v linearnih, celoštevilskih poljih (ang.: integer array), kjer vsako število predstavlja vrednost ene pike. Zaradi spominskih omejitev okolja si nisem mogel privoščiti nepotrebnih kopij različnih analiz slike, zato sem uporabljal le dve - prvo, ki je hranila samo sliko, in drugo, ki je vsebovala rezultate analiz. Sama vrednost pike lahko tako predstavlja barvo, EV, velikost najboljšega podšiva in druge, odvisno od konteksta.

Za primerjavo sem uporabljal okoli 25 slik različnih vsebin (in nekatere izpeljanke teh, za primerjavo uspešnosti ob prisotnosti oziroma odsotnosti motečih elementov) in velikosti tako na emulatorju okolja Wireless Toolkit kot lastnem mobilnem telefonu. Pri večjih slikah je bila MRŠ na mobilnem telefonu nepraktična, saj je količina slikovnih informacij, ki jih je bilo potrebno izrezati,



Slika 4.1: Razvojno okolje Wireless Toolkit (levo) in emulator (desno), ki je del tega.

na njegovem zaslonu velikosti 176x220 odstranila preveč in sliko pokvačila. Dodaten problem je predstavljala tudi počasnost, vsak rez slike dvakrat večje od zaslona je potreboval skoraj sekundo. Veliko bolje se je izkazal emulator, tako zaradi višje hitrosti kot zaradi večjega zaslona (240x290 pik). Rez dvakrat večje slike je vzel okoli pol sekunde, na končnem rezultatu prilagoditve (slika 2.1 spodaj) pa so opazne le manjše napake.

## Poglavje 5

# Sklepne ugotovitve

MRŠ se je izkazala za primerno rešitev problema prilagajanja velikosti slike zaslonom različnih razmerij, vendar ne v vseh primerih. Problemi, ki jo pestijo, so razlika med originalno in željeno velikostjo (odstranitev več kot polovice pik na splošno ni dala dobrih rezultatov), kompleksnost računanja (kljub temu, da ima vsak korak linearno kompleksnost  $O(n)$ , je potrebno za vsak rez izvesti dve analizi celotne slike) ter vsebinska neprimernost vseh slik.

MRŠ je bil najbolj uspešen na slikah, kjer točke zanimanja predstavljajo močan osprednji del na preprostem ozadju. Ker se alogiritem računanja EV o tem odloča glede na kontrast pik v primerjavi z njenimi sosedi, dobi močan osprednji del zaradi kompleksnosti visoko vrednost, enakomerno (oziroma zamegljeno) ozadje pa nizko (slika 5.1 zgoraj).

Za neprimerne so se izkazale slike, katerih vsebina je bila preveč kompleksna, ter slike, pri katerih človeško oko hitro opazi napake (najbolj očiten primer je človeški obraz). Ker metoda, kljub minimiziranju neželenega vpliva na vsebino slike pri rezu, to še vedno spreminja, pri prekompleksnih vsebinah hitro začnemo ustvarjati vidne zamike. Prav tako so bile neprimerne slike z enakomernimi, nekompleksnimi točkami zanimanja in vizualno kompleksnim ozadjem (slika 5.1 spodaj).

Te težave se povečujejo z zmanjševanjem ciljne velikosti, saj pri manjših slikah šiv predstavlja večji vsebinski del slike (in posledično zamik ostalih delov) kot pri večjih, prav tako pa morebitne točke interesa zaradi majhnosti izgubljajo kompleksnost, kar povečuje verjetnost, da jih bo energetska analiza slike ocenila kot manj pomembne. Dodaten problem predstavlja 'šum' v sliki, to je, spremembe v svetlosti in moči naključnih pik, ki jih lahko povzroči kompresija slike, obnašanje vezij v napravi, ki je sliko originalno posnela, in druge. Ker so to lastnosti, na katere je občutljiva energetska analiza, bi bilo idealno



Slika 5.1: Prikaz uspešnosti metode na manjših slikah z močnim centralnim fokusom (zgoraj) in vizualno kompleksnejših (spodaj).

sliko pred to filtrirati (barvo vsake pike delno zliti z barvo njenih sosedov), kar bi odpravilo manjše razlike in izpostavilo ostale, vendar to zaradi hitrostnih omejitev okolja ni bilo možno.

Mnogo od omenjenih problemov se zmanjša ali celo odpravi na večjih zaslonih in hitrejših napravah, zato je uporaba metode bolj praktična na napravah srednje velikosti zaslonov, ki imajo tudi hitrejšo procesorje (na primer pametni telefoni, dlanžniki ter vedno bolj popularni mini prenosniki), vsaj kar se tiče dinamične uporabe. Za manjše naprave je metoda še vedno uporabna, vendar bolj primerna ob vnaprejšnjem prilagajanju (pred prenosom slike na napravo, enkratno računanje na strani naprave ali pa prenos slike z že preračunanimi

šivi), saj bi to omogočilo uporabo boljšega, bolj kompleksnega algoritma analiza energetske vrednosti, ki drugače ne bi bil praktičen.

# Slike

1.1	Primerjava različnih načinov prilagajanja velikosti slike . . . . .	4
2.1	Primerjava različnih načinov reza slike . . . . .	6
2.2	Sosednost pik . . . . .	7
2.3	Vodoravni in navpični šivi . . . . .	8
2.4	Izris šiva . . . . .	9
3.1	Primerjava originalne slike ter slike po analizi . . . . .	11
3.2	Prikaz srečanja šivov . . . . .	12
3.3	Računanje vrednosti podšivov . . . . .	14
3.4	Računanje vrednosti podšivov . . . . .	15
3.5	Analiza in vrednosti šivov . . . . .	16
3.6	Rezanje šiva . . . . .	17
4.1	Razvojno okolje . . . . .	19
5.1	Primerjava uspešne in neuspešne analize . . . . .	21

# Literatura

- [1] (2007) Shai Avidan, Ariel Shamir, “Seam Carving for Content-Aware ImageResizing”.  
Dostopno na:  
<http://portal.acm.org/citation.cfm?id=1276377.1276390>
- [2] Fotografija last Krakonoš Hotel.  
Dostopno na:  
<http://www.hotelkrakonos.cz/en/photogallery/sport-facilities.html>
- [3] Fotografija last Arch Daily.  
Dostopno na:  
<http://www.archdaily.com/page/51/>
- [4] Sun Development Network.  
Dostopno na:  
<http://java.sun.com/javame/index.jsp>
- [5] VIM.  
Dostopno na:  
<http://www.vim.org/>
- [6] Sun Java Wireless Toolkit for CLDC.  
Dostopno na:  
<http://java.sun.com/products/sjwtoolkit/>