

Unaprijeđeni algoritmi za detekciju neispravnosti sirovih keramičkih pločica u stvarnom vremenu

Matić, Tomislav

Doctoral thesis / Disertacija

2014

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering / Sveučilište Josipa Jurja Strossmayera u Osijeku, Elektrotehnički fakultet**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:200:089787>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-02-20**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



SVEUČILIŠTE J. J. STROSSMAYERA
ELEKTROTEHNIČKI FAKULTET OSIJEK

Tomislav Matić

**Unaprijeđeni algoritmi za detekciju neispravnosti sirovih
keramičkih pločica u stvarnom vremenu**

Doktorska disertacija

Osijek, 2014.

Doktorska disertacija je izrađena na Elektrotehničkom fakultetu Osijek, Sveučilišta J. J. Strossmayera u Osijeku.

Mentor: Prof.dr.sc. Željko Hocenski, redoviti profesor

Disertacija ima 160 stranica

Disertacija broj: 43

Povjerenstvo za ocjenu doktorske disertacije

1. Prof.dr.sc. Željko Hocenski, redoviti profesor, Elektrotehnički fakultet Sveučilišta J. J. Strossmayera u Osijeku
2. Prof.dr.sc. Goran Martinović, Elektrotehnički fakultet Sveučilišta J. J. Strossmayera u Osijeku
3. Prof.dr.sc. Vlado Sruk, redoviti profesor, Fakultet elektronike i računarstva Sveučilišta u Zagrebu

Povjerenstvo za obranu doktorske disertacije

1. Prof.dr.sc. Željko Hocenski, redoviti profesor, Elektrotehnički fakultet Sveučilišta J. J. Strossmayera u Osijeku
2. Prof.dr.sc. Goran Martinović, Elektrotehnički fakultet Sveučilišta J. J. Strossmayera u Osijeku
3. Prof.dr.sc. Vlado Sruk, redoviti profesor, Fakultet elektronike i računarstva Sveučilišta u Zagrebu
4. Doc.dr.sc. Tomislav Keser, Elektrotehnički fakultet Sveučilišta J. J. Strossmayera u Osijeku
5. Doc.dr.sc. Alfonzo Baumgartner, Elektrotehnički fakultet Sveučilišta J. J. Strossmayera u Osijeku

Datum obrane disertacije: 25.09.2014.

Zahvale

Zahvaljujem se mentoru na poslijediplomskom doktorskom studiju prof.dr.sc. Željku Hocenskom na velikoj suradnji i podršci tijekom izrade doktorske disertacije bez koje ova disertacija ne bi bila završena.

Zahvaljujem se svojim roditeljima i ostaloj obitelji na potpori koju su mi pružali tijekom mojeg akademskog obrazovanja i koji nikada nisu izgubili nadu u moj završetak dokorskog studija.

Posebno se zahvaljujem svojoj dugogodišnjoj djevojci, Petri Krivak, koja me s puno razumijevanja podržavala u izradi doktorske disertacije.

Također se zahvaljujem svojim radnim kolegama (cimerima) Ivanu Aleksiju i Ivanu Vidoviću koji su nesebično surađivali u znanstvenim istraživanjima potrebnim za završetak ove disertacije.

Sadržaj

| | |
|--|------------|
| <i>Popis oznaka</i> | <i>I</i> |
| <i>Popis slika</i> | <i>V</i> |
| <i>Popis tablica</i> | <i>X</i> |
| <i>Popis algoritama</i> | <i>XII</i> |
| 1. Uvod | 1 |
| 2. Proizvodnja i kontrola kvalitete keramičkih pločica | 4 |
| 2.1. Keramika – povijest, definicija, svojstva i podjela | 4 |
| 2.2. Opis procesa proizvodnje keramičkih pločica | 6 |
| 2.3. Automatiziranost procesa proizvodnje keramičkih pločica..... | 9 |
| 2.4. Kontrola kvalitete keramičkih pločica | 10 |
| 3. Vizualna inspekcija neispravnosti na sirovim keramičkim pločicama | 13 |
| 3.1. Neispravnosti na sirovim keramičkim pločicama | 13 |
| 3.2. Prototip sustava za detekciju neispravnosti na sirovim keramičkim pločicama | 15 |
| 3.3. Postupak obrade sive slike sirove keramičke pločice | 16 |
| 3.4. Predobrada sive slike sirove keramičke pločice | 18 |
| 4. Nova metoda za izdvajanje sirove keramičke pločice od pozadine | 22 |
| 4.1. Izdvajanje objekata na sivoj slici (segmentacija slike) | 22 |
| 4.2. Segmentacija sive slike binarizacijom..... | 23 |
| 4.2.1. Podjela metoda za binarizaciju sive slike | 24 |
| 4.3. Binarizacija slike sirove keramičke pločice | 27 |
| 4.4. Otsu metoda za binarizaciju sive slike | 29 |
| 4.5. Metoda minimalne pogreške | 31 |
| 4.6. Metoda minimalne razlike entropija | 33 |
| 4.7. Prilagodljiva binarizacija zasnovana na detekciji promjene amplitude u 1D signalu | 36 |
| 4.7.1. Detekcija promjene amplitude 1D signala | 37 |
| 4.7.2. Binarizacija sive slike sirove pločice SCD metodom | 42 |

| | |
|--|------------|
| 4.8. Paralelni GPU algoritam PBSCD metode | 50 |
| 4.8.1. Podjela paralelnih platformi | 50 |
| 4.8.2. NVIDIA CUDA paralelna platforma | 51 |
| 4.8.3. Sklopovski model NVIDIA CUDA paralelne platforme | 52 |
| 4.8.4. Programski model NVIDIA CUDA paralelne platforme | 55 |
| 4.8.5. CUDA PBSCD algoritam | 58 |
| 4.9. Eksperimentalni rezultati | 61 |
| 4.9.1. Pogreška binarizacije | 62 |
| 4.9.2. Rad u stvarnom vremenu | 67 |
| 4.9.3. Zaključak | 71 |
| 5. Unaprijeđeni algoritam za detekciju neispravnosti na rubovima i uglovima sirove pločice | 72 |
| 5.1. Postojeće metode za detekciju neispravnosti na rubovima i uglovima sirove keramičke pločice | 72 |
| 5.2. Detekcija ruba sirove keramičke pločice | 74 |
| 5.3. Praćenje konture i određivanje smjera konture sirove keramičke pločice | 77 |
| 5.3.1. Praćenje konture i određivanje smjera konture | 78 |
| 5.3.2. Alternativna kontura i smjer konture | 80 |
| 5.3.3. Sekvencijalni algoritam praćenja konture i određivanja smjera konture | 82 |
| 5.3.4. Primjeri praćenja smjera konture | 83 |
| 5.4. Filtriranje smjera konture | 88 |
| 5.4.1. Osrednjavanje s pomičnim prosjekom | 88 |
| 5.4.2. Ublažavanje s Gausovim filterom | 91 |
| 5.5. Detekcija i lokalizacija neispravnosti na uglovima i rubovima sirove keramičke pločice | 94 |
| 5.5.1. Određivanje razlike smjera konture | 94 |
| 5.5.2. Detekcija neispravnosti na rubovima sirove keramičke pločice | 95 |
| 5.5.3. Lokalizacija uglova i rubnih neispravnosti na sirovim keramičkim pločicama | 96 |
| 5.5.4. Detekcija i lokalizacija neispravnosti na uglovima sirove keramičke pločice | 97 |
| 5.5.5. Primjeri detekcije i lokalizacije neispravnosti na rubovima i uglovima sirove keramičke pločice | 99 |
| 5.5.6. Sekvencijalni algoritam metode za detekciju i lokalizaciju neispravnosti na rubovima i uglovima sirove keramičke pločice | 101 |
| 5.6. Paralelni algoritam metode za detekciju i lokalizaciju neispravnosti na rubovima i uglovima sirove keramičke pločice | 101 |
| 5.7. Eksperimentalni rezultati | 103 |

| | |
|--|------------|
| 5.7.1. Mogućnost detekcije neispravnosti na rubovima i uglovima sirove keramičke pločice | 103 |
| 5.7.2. Rad u stvarnom vremenu..... | 108 |
| 5.7.3. Zaključak..... | 111 |
| 6. Nova metoda za ujednačavanja slike površine sirove keramičke pločice | 113 |
| 6.1. Postojeće metode za ujednačavanje slike površine keramičke pločice | 113 |
| 6.2. Uklanjanje rubova na sivoj slici sirove keramičke pločice..... | 114 |
| 6.3. Uklanjanje pozadine od površine sirove keramičke pločice..... | 116 |
| 6.3.1. Određivanje minimalnih i maksimalnih koordinata redova i stupaca vektora konture | 116 |
| 6.3.2. Uklanjanje pozadine | 118 |
| 6.4. Uklanjanje utjecaja rotacije i translacije na sivoj slici sirove keramičke pločice..... | 120 |
| 6.4.1. Uklanjanje translacije sirove pločice | 120 |
| 6.4.2. Uklanjanje rotacije sirove pločice..... | 122 |
| 6.5. Paralelni algoritam metode za ujednačavanja slike površine sirove keramičke pločice | 124 |
| 6.6. Eksperimentalni rezultati | 126 |
| 6.6.1. Pogreška metode..... | 126 |
| 6.6.2. Rad u stvarnom vremenu..... | 131 |
| 6.6.3. Zaključak..... | 134 |
| 7. Detekcija neispravnosti na površini sirovih keramičkih pločica | 136 |
| 7.1. Postojeće metode za detekciju neispravnosti na površini sirovih keramičkih pločica | 136 |
| 7.2. MALD metoda za detekciju točkastih neispravnosti..... | 138 |
| 7.3. Metoda Haralick-ovih značajki s umjetnom neuronskom mrežom | 140 |
| 8. Zaključak | 143 |
| 9. Literatura..... | 147 |
| 10. Prilozi..... | 156 |
| 10.1. Značajke Basler A102fc kamere..... | 156 |
| 10.2. CUDA GPU značajke u ovisnosti o CC broju..... | 157 |
| Sažetak..... | 158 |
| Abstract..... | 159 |
| Životopis..... | 160 |

Popis oznaka

| | |
|---|--|
| $I(x_d, y_d)$ | Siva slika s geometrijskom distorzijom. |
| $I(x, y)$ | Siva slika. |
| (x, y) | Koordinate piksela na slici. |
| x_r | x koordinata sa uklonjenom radijalnom distorzijom. |
| y_r | y koordinata sa uklonjenom radijalnom distorzijom. |
| x_t | x koordinata sa uklonjenom tangencijalnom distorzijom. |
| y_t | y koordinata sa uklonjenom tangencijalnom distorzijom. |
| x_{rt} | Ukupna distorzija za x koordinatu. |
| y_{rt} | Ukupna distorzija za y koordinatu. |
| f_x | Žarišna duljina za x os. |
| f_y | Žarišna duljina za y os. |
| (c_x, c_y) | Koordinate centra slike. |
| K | Visina slike u pikselim. |
| M | Širina slike u pikselima. |
| γ | Kut rotacije sirove keramičke pločice. |
| $I_B(x, y)$ | Binarna slika. |
| T | Razina praga binarizacije. |
| n_{sub} | Broj diobenih dijelova sive slike. |
| $h(i)$ | Histogram sive slike. |
| $h_n(i)$ | Normalizirani histogram sive slike. |
| (x', y') | Alternativne koordinate piksela na slici. |
| $\varepsilon_1, \varepsilon_2, \varepsilon_3$ | Konstante za karakterizaciju bimodalnog histograma. |
| B | Broj razina intenziteta piksela. |
| C_0 | Svi pikseli pozadine na sivoj slici. |
| C_1 | Svi pikseli objekta na sivoj slici. |
| P_0 | Vjerovatnost pojave piksela klase C_0 . |
| P_1 | Vjerovatnost pojave piksela klase C_1 . |
| μ_0 | Srednja vrijednost piksela klase C_0 . |
| μ_1 | Srednja vrijednost piksela klase C_1 . |
| σ_0 | Standardna devijacija piksela klase C_0 . |
| σ_1 | Standardna devijacija piksela klase C_1 . |
| μ | Srednja vrijednost piksela sive slike. |
| J_T | Kriterijska funkcija za određivanje razine praga binarizacije Otsu metodom i metodom minimalne pogreške. |
| σ_B^2 | Međuklasna varijanca. |
| $f(i), g(i)$ | Distribucije vjerojatnosti. |
| $H_{f,g}$ | Razlika entropija. |
| $I_\mu(x, y)$ | Binarizirana μ -slika. |
| H_{CE} | Kriterijska funkcija za određivanje razine praga binarizacije metodom minimalne razlike entropija. |
| s | 1D signal. |

| | |
|---|---|
| n_i | Skup indeksa aproksimiranog signala. |
| W | Broj indeksa skupa (broj promjena u signalu). |
| A_n | Aproksimirane vrijednosti amplituda. |
| $\tilde{\mathbf{s}}$ | Aproksimirani signal. |
| N | Broj uzoraka signala \mathbf{s} . |
| $A_1(n)$ | Prva aproksimirana vrijednost amplitude signala. |
| $A_2(n)$ | Druga aproksimirana vrijednost amplitude signala. |
| ΔA | Razlika vrijednosti aproksimiranih amplituda. |
| $p(\mathbf{s}; n)$ | Funkcija gustoće vjerojatnosti signala \mathbf{s} . |
| J_{SCD} | Kriterijska funkcija SCD metode. |
| n_t | Pronađeni indeks promjene aproksimiranog signala. |
| n_{Gt} | Poznati indeks promjene signala \mathbf{s} . |
| \hat{n}_t | Devijacija indeksa promjene aproksimiranog signala. |
| I_G | Gornji dio $I(x, y)$ slike. |
| I_D | Donji dio $I(x, y)$ slike. |
| I_L | Lijevi dio $I(x, y)$ slike. |
| I_{De} | Desni dio $I(x, y)$ slike. |
| n_{tG} | Pronađeni indeks promjene za gornji dio slike. |
| n_{tD} | Pronađeni indeks promjene za donji dio slike. |
| n_{tL} | Pronađeni indeks promjene za lijevi dio slike. |
| n_{tDe} | Pronađeni indeks promjene za desni dio slike. |
| n_{OP,h_n} | Broj operacija za izračun normaliziranog histograma sive slike. |
| $n_{OP,Otsu}$ | Broj operacija za izračun razine praga binarizacije Otsu metodom. |
| $n_{OP,PBSCD}$ | Broj operacija PBSCD metode. |
| $n_{OP,SCD}$ | Broj operacija SCD metode. |
| INT_s | Integralna suma signala \mathbf{s} . |
| $S_{a,b}$ | Suma vrijednosti signala \mathbf{s} između indeksa a i b . |
| $n_{OP,PBSCDINT}$ | Broj operacija PBSCD metode s integralnom sumom. |
| CC | Compute Capability, oznaka specifikacija CUDA arhitekture. |
| ID_B | Oznaka (ID) bloka u CUDA programskom modelu. |
| ID_T | Oznaka niti unutar bloka u CUDA programskom modelu. |
| ID | Oznaka niti u CUDA programskom modelu. |
| $CUDA_G,$ $CUDA_D,$ $CUDA_L$ $CUDA_{De}$ | CUDA kerneli za PBSCD metodu. |
| $dimMreze$ | Veličina CUDA mreže za pojedini kernel PBSCD metode. |
| $dimBloka$ | Veličina CUDA bloka za pojedini kernel PBSCD metode. |
| ω, v | Vektori u dijeljenoj memoriji CUDA uređaja. |
| $maxf$ | Maksimalni broj jednostruke preciznosti. |
| ME | Pogreška binarizacije. |
| OI_m | Omjer vremena izvođenja metoda. |
| t_{\ll} | Najkraće vrijeme izvođenja. |
| t_m | Vrijeme izvođenja m metode. |

| | |
|-------------------|---|
| $N_4(a)$ | Četiri susjedna piksela, piksela a . |
| $N_8(a)$ | Osam susjednih piksela, piksela a . |
| $I_R(x, y)$ | Slika ruba. |
| $C(i)$ | Vektor konture. |
| $C(0)$ | Početna točka konture. |
| L | Duljina vektora konture. |
| $D(i)$ | Smjer konture. |
| $D(0)$ | Početni smjer konture. |
| $N_5^{D(i)}$ | N_5 susjedstvo za smjer konture $D(i)$. |
| α | Relativni pomak u $N_5^{D(i)}$ susjedstvu. |
| P_1 do P_5 | Prioriteti pretraživanja u $N_5^{D(i)}$ susjedstvu. |
| $C_A(i_A)$ | Alternativni vektor konture. |
| $D_A(i_A)$ | Alternativni vektor smjera konture. |
| i_A | Broj alternativnih točaka konture. |
| L_A | Duljina alternativnog vektora konture. |
| α' | Relativni pomak u $N_5^{D(i)}$ susjedstvu za točku ruba nižeg prioriteta. |
| L_{max} | Maksimalna duljina vektora konture. |
| L_{min} | Minimalna duljina vektora konture. |
| $\bar{D}(i)$ | Osrednjena vrijednost smjera konture. |
| w | Veličina prozora pomičnog prosjeka. |
| $korak_{i,k}$ | Kružni pomak u vektoru smjera konture za izračun $\bar{D}(i)$. |
| $g(t)$ | Gaussova funkcija. |
| $\tilde{D}(i)$ | Ublaženi vektor $\bar{D}(i)$ s Gaussovom digitalnim filterom (filtrirani smjer konture). |
| $korak_{i,t}$ | Indeks vektora \bar{D} za izračun $\tilde{D}(i)$. |
| $DIF(i)$ | Razlika filtriranog smjera konture. |
| T_R | Razina praga za detekciju i lokalizaciju uglova sirove pločice (detekciju neispravnosti na rubovima). |
| T_U | Razina praga za detekciju neispravnosti na uglovima sirove keramičke pločice. |
| e_R | Broj neispravnosti na rubovima sirove pločice. |
| e_k | Grupa točaka razlike filtriranog smjera konture. |
| $i_{max,k}$ | Maksimalna $DIF(i)$ vrijednost u grupi točaka e_k . |
| $c_{l,k}(j)$ | Grupa točaka ugla razlike filtriranog smjera konture. |
| Z_k | Broj grupa točaka za određeni ugao pločice. |
| e_U | Ukupan broj neispravnosti na uglovima sirove pločice. |
| $Z_k^{0^\circ}$ | Broj grupa točaka $c_{l,k}$ za $D(0) = 0^\circ$. |
| $Z_k^{180^\circ}$ | Broj grupa točaka $c_{l,k}$ za $D(0) = 180^\circ$. |
| p_p | Površina neispravnosti na rubovima i uglovima sirove pločice u pikselima. |
| p_l, p_d | Dimenzije neispravnosti na rubovima sirove pločice u pikselima. |
| p_u | Dimenzija neispravnosti na uglovima sirove pločice u pikselima. |
| r | Rezolucija slike. |
| P | Površina neispravnosti na rubovima sirove pločice u mm^2 . |
| l, d | Dimenzije neispravnosti na rubovima sirove pločice u mm . |
| P_u | Površina neispravnosti na uglovima sirove pločice u mm^2 . |

| | |
|---------------------------|---|
| l_u | Dimenzije neispravnosti na uglovima sirove pločice u mm . |
| I_{BE} | Erodirana binarana slika. |
| E | Strukturni element erozije. |
| $I_{ER}(x, y)$ | Slika ruba nakon erozije. |
| $CE(i)$ | Vektor konture nakon primjenjene erozije. |
| R_{max}^a, R_{min}^a | Minimalne, maksimalne koordinate konture za a redak $I_{ER}(x, y)$ slike. |
| S_{max}^b, S_{min}^b | Minimalne, maksimalne koordinate konture za b stupac $I_{ER}(x, y)$ slike. |
| $I_P(x, y)$ | Slika sa označenim pikselima pozadine. |
| $I_{PR}(x, y)$ | Slika ruba nakon označavanja piksela pozadine. |
| $CP(i)$ | Vektor konture nakon označavanja piksela pozadine. |
| x_{min} | |
| x_{max} | |
| y_{min} | |
| y_{max} | |
| $I_S(x', y')$ | Slika površine sirove pločice sa uklonjenom translacijom pločice. |
| $P \times Q$ | Veličina $I_S(x', y')$ slike u pikselima. |
| $CP'(i)$ | $CP(i)$ vektor konture za $I_S(x', y')$ sliku. |
| $CP'_t(j)$ | Privremeni vektor konture. |
| L_t | Duljina privremenog vektora konture. |
| $I_{SP}(x', y')$ | Slika površine sirove pločice sa uklonjenom translacijom i rotacijom pločice. |
| $Err(x', y')$ | Pogreška u vrijednosti popunjenog piksela pozadine. |
| $\overline{CErr}_{CP'}^k$ | Srednja kumulativna pogreška k -tih vektora konture. |
| Err_{SP} | Ukupna pogreška ujednačavanja slike površine sirove pločice. |
| Err_{GT} | Pogreška geometrijske transformacije i interpolacije. |
| $Err_{SP}^{\%}$ | Relativna ukupna pogreška ujednačavanja slike površine sirove pločice. |
| $Err_{GT}^{\%}$ | Relativna pogreška geometrijske transformacije i interpolacije. |
| $e(x, y)$ | Apsolutna prosječna razlika piksela lijevog i desnog prozora MALD metode. |
| $h(x, y)$ | Razlika prosječne vrijednosti piksela oba prozora i intenziteta analiziranog piksela MALD metode. |
| $GLC_{d,\varphi}$ | Matrica prostornih ovisnosti intenziteta piksela sive slike. |
| $I(x, y), I(x^p, y^p)$ | Par piksela udaljenih za d u smjeru φ za izračun $GLC_{d,\varphi}$ matrice. |
| η_i | Težinski faktori neurona. |
| β | Razina prga neurona. |
| ψ | Prijenosna funkcija neurona. |
| Z | Izlaz neurona. |

Popis slika

- Slika 2.1: Primjeri proizvoda od keramike: (a) opeka i cigla, (b) cijevi za otpadnu vodu, (c) vatrostalne ploče, (d) ekspandirane keramičke granule, (e) podne keramičke pločice, (f) posuđe od keramike, (g) sanitarna keramika, (h) automobilska kočnica od keramike, (i) abrazivne ploče od keramike.
- Slika 2.2: Shematski prikaz procesa proizvodnje keramičkih pločica monopaljnjem.
- Slika 2.3: Međuproizvodi u procesu proizvodnje keramičkih pločica: (a) Osušena prešana pločica – keks, (b) sirova – glazirana pločica, (c) keramička pločica.
- Slika 2.4: Shematski prikaz recikliranja međuproizvoda tijekom proizvodnje keramičkih pločica.
- Slika 2.5: Primjeri raznih vrsta keramičkih pločica (ustupila Keramika Modus d.o.o.).
- Slika 2.6: Vizualna kontrola kvalitete keramičkih pločica (ustupila Keramika Modus d.o.o.).
- Slika 3.1: Geometrijske neispravnosti na sirovim pločicama: (a) i (b) neispravnosti na uglovima, (c) neispravnost na rubu.
- Slika 3.2: Površinske neispravnosti na sirovim pločicama: (a) točkaste neispravnosti, (b) neispravnosti nastale od kapljica kondenzata, (c) nedostatak glazure, (d) nepravilnosti u teksturi, (e) ispupčenja i (f) udubljenja.
- Slika 3.3: Prototip sustava za vizualnu inspekciju neispravnosti na sirovim keramičkim pločicama: (a) model prototipa, (b) razvijeni prototip u laboratorijskim uvjetima, (c) razvijeni prototip u industrijskim uvjetima.
- Slika 3.4: Faze obrade sive slike (zatomnjeni dijelovi predstavljaju metode razvijene u ovom radu).
- Slika 3.5: Primjer jastučaste (a) i bačvaste geometrijske distorzije (b).
- Slika 3.6: Primjer sive slike s geometrijskom bačvastom distorzijom (a), slika s uklonjenom bačvastom distorzijom, (c) slika s uzorkom šahovske ploče.
- Slika 4.1: Primjeri segmentacije slike: (a) slika mrežnice oka, (b) segmentirana slika mrežnice oka s izdvojenim krvnim žilama (bijela boja), (c) slika iz prirode, (d) segmentirana slika s označenim rubovima objekta od interesa (stablo, ptica), (e) slika sirove keramičke pločice na pokretnoj liniji, (f) segmentirana slika s izdvojenom površinom pločice.
- Slika 4.2: Binarizacija slike: (a) globalno određivanje razine praga (T), (b) prilagodljivo određivanje razine praga ($T_i, i = 0, \dots, 8$).
- Slika 4.3: Izračun razine praga na temelju: (a) histograma slike, (b) intenziteta piksela slike.
- Slika 4.4: Primjer normaliziranog bimodalnog histograma. Zasjenjeni dijelovi označavaju dvije regije histograma (objekt, pozadina).
- Slika 4.5: Primjeri slika sirovih pločica (lijevo) i njihovih normaliziranih histograma (desno).
- Slika 4.6: Primjer normaliziranog bimodalnog histograma s označenim uvjetima (4.3) i (4.4).
- Slika 4.7: Primjer prijenosa materijala s pločice na dijelove pokretne trake: (a) siva slika sirove pločice, (b) povećani dio slike (a) (crtkani bijeli pravokutnik) s vidljivim tragovima materijala na remenu pokretne trake, (c) normalizirani histogram slike (a) s zasjenjenim dijelom piksela koji se preklapaju.
- Slika 4.8: Primjeri četiri stohastička signala s (siva linija) i njihovih aproksimacija s (crtkana crna linija) s $N = 80$, $\Delta A = 20$, (a) $\Delta A/\sigma = 3$, (b) $\Delta A/\sigma = 4$, (c) $\Delta A/\sigma = 5$ i (d) $\Delta A/\sigma = 6$.

- Slika 4.9: Devijacija indeksa SCD metode n_t za $N \in [10, 100]$, $n_{Gt} = N/2$, $\Delta A = 20$, (a) $\Delta A/\sigma = 3$, (b) $\Delta A/\sigma = 4$, (c) $\Delta A/\sigma = 5$ i (d) $\Delta A/\sigma = 6$.
- Slika 4.10: Devijacija indeksa SCD metode nt s promjenom $\Delta A/\sigma \in [1, 6]$, $n_{Gt} = N/2$, $\Delta A = 20$, (a) $N = 40$ i (b) $N = 60$.
- Slika 4.11: Izdvojene vrijednosti intenziteta 200 piksela slike (a) za: (b) gornji i (c) donji dio naznačenog stupca, (d) lijevi i (e) desni dio naznačenog retka.
- Slika 4.12: PBSCD metoda: (a) primjer ulazne $I(x, y)$ sive slike veličine $K \times M$ piksela, (b) gornji I_G i donji I_D dio slike prema (4.25), (c) lijevi I_L i desni I_{De} dio slike prema (4.25), (d) međurezultat binarizacije prema (4.26), (e) međurezultat binarizacije prema (4.27), (f) konačni rezultat binarizacije PBSCD metode (bijela boja predstavlja objekt, a crna pozadinu).
- Slika 4.13: I_G i I_D slike s primjerom primjene SCD metode na stupce bez piksela sirove pločice (a), I_L i I_{De} slike s primjerom primjene SCD metode na redove bez piksela sirove pločice (b).
- Slika 4.14: Primjer ulazne slike sirove pločice (a) i izlazna binarna slika PBSCD metode s pogrešno klasificiranim pikselima u uglovima binarne slike (b).
- Slika 4.15: Primjer signala $s(i)$ (a), (b) 1D integralna suma signala $s(i)$, (c) i (d) primjer izračuna sume dijela signala za $a = 0$, $b = 4$ prema (4.33), (e) i (f) primjer izračuna sume dijela signala za $a = 2$, $b = 4$ prema (4.33).
- Slika 4.16: Shematski prikaz rasporeda tranzistora u CPU (a) i GPU (b).
- Slika 4.17: Sklopovski model CUDA 1.0 GPU.
- Slika 4.18: Primjer spajanja grafičke kartice sa CPU-om putem PCIE sabirnice.
- Slika 4.19: CUDA programski model.
- Slika 4.20: Pojednostavljeni vremenski prikaz izvođenja osnova na CUDA GPU.
- Slika 4.21: Primjer poziva kernela (a), primjer kernela za zbrajanje dva vektora ($fParam1$ i $fParam2$) i spremanje rezultata u $fParam3$ vektor (b).
- Slika 4.22: Veza između pojedinih CUDA PBSCD kernela ($CUDA_G$, $CUDA_D$, $CUDA_L$ i $CUDA_{De}$) i dijelova $I(x, y)$ slike prema izrazu (4.25).
- Slika 4.23: Primjer određivanja $n_{tL}(ID_B)$ za $N = 6$ upotrebom redukcijskog algoritma. Sivo obojeni elementi predstavljaju pronađenu manju vrijednost u pojedinom koraku, crtkana linija predstavlja sinkronizacijske točke, $maxf$ predstavlja maksimalni broj jednostruke preciznosti.
- Slika 4.24: Primjeri slika sirovih keramičkih pločica za određivanje pogreške binarizacije: (a) primjer $S1$, (b) primjer $S2$, (c) primjer $S3$ i (d) primjer $S4$.
- Slika 4.25: Ispravne binarizirane slike sirovih keramičkih pločica: (a) slika S_C1 , (b) slika S_C2 , (c) slika S_C3 i (d) slika S_C4 .
- Slika 4.26: Normalizirani histogrami slika sirovih keramičkih pločica $S1$ (a), $S2$ (b), $S3$ (c) i (d) $S4$.
- Slika 4.27: Intenziteti 200 piksela središnjeg stupca I_G i I_D dijela slike i središnjeg retka I_L i I_{De} dijela slike za primjere $S1$ (a), $S2$ (b), $S3$ (c) i (d) $S4$.
- Slika 4.28: Rezultati binarizacije za primjere $S1$ (prvi stupac), $S2$ (drugi stupac), $S3$ (treći stupac) i $S4$ (četvrti stupac) sa: Otsu metodom (prvi red), metodom minimalne razlike entropije – MRE (drugi red), metodom minimalne pogreške – MMP (treći red) i PBSCD metodom (četvrti red). Crveni pravokutnici označavaju pogrešno binarizirane piksele.

- Slika 4.29: Prosječno vrijeme izvođenja u 100 iteracija CUDA PBSCD i Otsu metode: (a) slika veličine 1038×1098 piksela i promjenjivi N , (b) slika veličine 1946×2054 piksela i promjenjivi N , (c) $N = 120$ i promjenjiva veličina slike izražena u megapikselima (MP), (d) $N = 250$ i promjenjiva veličina slike u MP .
- Slika 5.1: Susjedstvo piksela a označeno s crnim kvadratima: (a) 4 susjedna piksela $N_4(a)$ i (b) 8 susjednih piksela $N_8(a)$.
- Slika 5.2: Primjer dvije $S1$ i $S2$ povezane komponente na sintetički generiranoj binarnoj slici (a), primjer pet povezanih komponenti ($S1 - S5$) na binarnoj slici (PBSCD metoda) sirove keramičke pločice, (c) detektirani rub za primjer (a), (d) 3×3 oblik povezane komponente. Na slici (a), (c) i (d) crni pravokutnik označava piksel s intenzitetom '1', a na slici (b) bijela boja predstavlja piksele s intenzitetom '1'.
- Slika 5.3: Obrnuta L forma s označenim pikselima (a), (b) prva tri koraka detektiranja točaka ruba za nulti i prvi red slike 5.2 (a), (c) prva tri koraka detektiranja točaka ruba za prvi i drugi red slike 5.2 (a). Obrnuta L forma je označena s plavom linijom.
- Slika 5.4: Mogući smjerovi konture za piksel sive boje (a) i (b). Primjeri $N_5^{D(i)}$ susjedstva i prioriteta pretraživanja susjedstva za trenutni smjer konture $D(i) = 0^\circ$ (c), $D(i) = 45^\circ$ (d) i $D(i) = 90^\circ$ (e).
- Slika 5.5: Primjer praćenja konture i određivanja smjera konture za četiri točke ruba (a) do (d) i $D(0) = 0^\circ$. Pronađena sljedeća točka konture je označena sa crtkanim kvadratom, izračunati $D(i)$ je upisan u središte kvadrata.
- Slika 5.6: Primjer četiri koraka (a) do (d) praćenja konture i popunjavanje vektora konture $C(i)$, vektora smjera konture $D(i)$, vektora alternativnih točaka konture $C_A(i)$ i vektora alternativnog smjera konture $D_A(i)$. Sivi kvadrat predstavlja alternativnu točku konture, 'x' predstavlja odbačenu točku konture (nastao prekid konture).
- Slika 5.7: Primjer smjera konture: (a) sintetički generirana binarna slika sirove pločice $I_B(x, y)$ s kutom rotacije $\gamma = 0^\circ$, (b) slika ruba $I_R(x, y)$ s naznačenim smjerom praćenja konture za $D(0) = 0^\circ$, (c) graf smjera konture $D(i)$ za $D(0) = 0^\circ$, (d) slika ruba $I_R(x, y)$ s naznačenim smjerom praćenja konture za $D(0) = 180^\circ$, (e) graf smjera konture $D(i)$ za $D(0) = 180^\circ$.
- Slika 5.8: Primjer smjera konture: (a) sintetički generirana binarna slika sirove pločice $I_B(x, y)$ s kutom rotacije $\gamma = 0.5^\circ$, (b) slika ruba $I_R(x, y)$ s naznačenim smjerom praćenja konture za $D(0) = 0^\circ$, (c) graf smjera konture $D(i)$ za $D(0) = 0^\circ$, (d) slika ruba $I_R(x, y)$ s naznačenim smjerom praćenja konture za $D(0) = 180^\circ$, (e) graf smjera konture $D(i)$ za $D(0) = 180^\circ$.
- Slika 5.9: Primjer savršene linije piksela (a) i nesavršene linije piksela (b).
- Slika 5.10: Smjer konture za primjere slika sirovih pločica $S1$, $S2$, $S3$, i $S4$ (označeni redovi) iz poglavlja 4.9. Stupac $I_R(x, y)$ predstavlja slike ruba za pojedini primjer slika, stupac $D(0) = 0^\circ$ predstavlja grafove smjera konture sa početnim smjerom konture 0° , stupac $D(0) = 180^\circ$ predstavlja grafove smjera konture sa početnim smjerom konture 180° .
- Slika 5.11: Smjer konture za sintetički generiranu binarnu sliku sirove pločice s kutom rotacije $\gamma = 0^\circ$ i s jednom neispravnošću na lijevom rubu (a), s kutom rotacije $\gamma = 0.5^\circ$ i s jednom neispravnošću na lijevom rubu (b), uvećani dio slike ruba (a) i (b) s vidljivom neispravnošću na rubu (c) i (d)
- Slika 5.12: Osrednjeni smjer konture s kružnim pomičnim prosjekom širine prozora $w \in [1, 3, 11]$ za sintetički generiranu binarnu sliku sirove pločice s kutom rotacije $\gamma = 0.5^\circ$ i (a) $D(0) = 0^\circ$, (b) $D(0) = 180^\circ$.

- Slika 5.13: Osrednjeni smjer konture s kružnim pomičnim prosjekom širine prozora $w \in [1, 3, 11]$ za sintetički generiranu binarnu sliku sirove pločice s jednom neispravnošću na lijevom rubu, kutom rotacije $\gamma = 0.5^\circ$ i (a) $D(0) = 0^\circ$, (b) $D(0) = 180^\circ$.
- Slika 5.14: Uvećani dio grafa osrednjenog smjera konture prikazanog na slici 5.13 (b) za $w = 11$, (a) stepeničasti skokovi nastali zbog nesavršenosti linije piksela ruba označeni s *, (b) stepeničasti skokovi nastali zbog neispravnošću na rubu.
- Slika 5.15: Primjer stepeničastog skoka nastalog zbog nesavršenosti linije piksela ruba nakon osrednjavanja za $w = 11$ (a), (b) graf uzorkovane Gaussove funkcije za $N = 15$, $t = [-7, 7]$ s korakom 1 i $\sigma = 2.5$.
- Slika 5.16: Uvećani dio grafa vektora smjera konture nakon osrednjavanja (crna linija) i ublažavanja s digitalnim Gaussovom filterom (zeleno linija) za $w = 11$, $N = 15$, $\sigma = 2.5$: (a) stepeničasti skokovi nastali zbog nesavršenosti linije piksela ruba označeni s *, (b) stepeničasti skokovi nastali zbog neispravnošću na rubu.
- Slika 5.17: Graf vektora $DIF(i)$: (a) $DIF(i)$ prema (5.17) za $w = 11$, $N = 15$, $\sigma = 2.5$ i $D(0) = 0^\circ$, (b) $DIF(i)$ prema (5.17) za $w = 11$, $N = 15$, $\sigma = 2.5$ i $D(0) = 180^\circ$, (c) $DIF(i)$ primijenjen na $D(i)$ za $D(0) = 0^\circ$ i (d) $DIF(i)$ primijenjen na $D(i)$ za $D(0) = 180^\circ$.
- Slika 5.18: Primjer detektiranih grupa točaka e_k (označeno sa zelenim točkama) za primjer sa slike 5.8 s početnim smjerom konture (a) $D(0) = 0^\circ$, (b) $D(0) = 180^\circ$ i primjer sa slike 5.11 s početnim smjerom konture (c) $D(0) = 0^\circ$, (d) $D(0) = 180^\circ$ (crvena linija predstavlja $T_R = 1.6^\circ$, crvene točke označavaju $i_{max,k}$).
- Slika 5.19: Primjer detektiranih grupa točaka $c_{l,k}$ (označeno sa zelenim točkama) za primjer sa slike 5.8 s početnim smjerom konture (a) $D(0) = 0^\circ$, (b) $D(0) = 180^\circ$ i za primjer sintetički generirane binarne slike s neispravnošću na dva ugla sa početnim smjerom konture (c) $D(0) = 0^\circ$, (d) $D(0) = 180^\circ$ (crvena linija predstavlja $T_U = 7.2^\circ$).
- Slika 5.20: Primjer lokalizacije uglova i neispravnošću na uglovima i rubovima sirove pločice: (a) lokalizirani ispravni uglovi sirove pločice, (b) lokalizirani uglovi i neispravnošću na sirovoj pločici, (c) lokalizirane neispravnošću na dva ugla (crvene kružnice) sirove pločice.
- Slika 5.21: Dijagram paralelizacije predložene metode za detekciju i lokalizaciju neispravnošću na rubovima i uglovima sirovih pločica s OpenMP bibliotekama.
- Slika 5.22: Generirane neispravnošću na rubovima sirove pločice sa: $p_p = 4$, $p_l = 3$ i $p_d = 2$ (a), (b) $p_p = 5$, $p_l = 3$ i $p_d = 2$ i (c) $p_p = 6$, $p_l = 4$ i $p_d = 2$. Generirane neispravnošću na uglovima sirove pločice sa: $p_u = 2$ (d) i (e) $p_u = 3$.
- Slika 5.23: Primjeri neispravnošću nakon binarizacije slike sirove pločice prikupljene u laboratorijskim uvjetima s PBSCD metodom: (a) ispravan ugao i rub, (b) do (d) neispravnošću na rubovima, (e) i (f) neispravnošću na uglovima.
- Slika 5.24: Primjeri neispravnošću nakon binarizacije slike sirove pločice prikupljene u industrijskim uvjetima s PBSCD metodom: (a) neispravan rub, (b) i (c) neispravnošću na uglovima.
- Slika 5.25: Prosječno vrijeme izvođenja u 100 iteracija sekvencijalnog algoritma predložene metode za detekciju neispravnošću na rubovima i uglovima (DRU) sirovih pločica i Canny metode za detekciju ruba (a), (b) ubrzanje predložene metode u odnosu na Canny metodu.
- Slika 5.26: Prosječno vrijeme izvođenja u 100 iteracija paralelnog OpenMP algoritma i sekvencijalnog algoritma (DRU) predložene metode za detekciju neispravnošću na rubovima i uglovima sirovih pločica (a), (b) ubrzanje paralelnog algoritma u odnosu na sekvencijalni algoritam.

- Slika 6.1: Primjer gornjeg i donjeg ruba (a), i lijevog i desnog ruba (b) na svojoj slici sirove pločice.
- Slika 6.2: Primjer erozije $I_{BE}(x, y)$ (c) za binarnu sliku $I_B(x, y)$ (a) i kvadratni 3×3 piksela strukturni element E (b).
- Slika 6.3: Primjer erozije: ulazna slika sirove keramičke pločice (a), binarizirana ulazna slika s PBSCD metodom (b), erodirana binarna slika (c) i razlika binarne i erodirane slike (d).
- Slika 6.4: Primjer pronalaska R_{min}^a i R_{max}^a koordinata za redak a označen sa zelenom linijom (a), primjer pronalaska S_{min}^b i S_{max}^b koordinata za stubac b označen sa zelenom linijom, crvena linija označava točke vektora konture $CE(i)$.
- Slika 6.5: Primjer ulaznih $I(x, y)$ slika sirovih keramičkih pločice (a), (c) i slika sirovih pločica s uklonjenom pozadinom (b), (d).
- Slika 6.6: Sintetičko generirani primjer određivanja x_{min} , x_{max} , y_{min} i y_{max} koordinata na $I_P(x, y)$ slici (a), (b) primjer $I_S(x', y')$ slike.
- Slika 6.7: $I_S(x', y')$ slika: (a) za ulaznu sliku prikazanu na slici 6.5 (a), (b) za ulaznu sliku prikazanu na slici 6.5 (c).
- Slika 6.8: Sintetički primjer popunjavanja piksela pozadine: (a) N_8 susjedstvo označeno sa crtkanim kvadratom za piksel $CP'(1)$, (b) pronađeni piksel pozadine (sivi kvadrat) s označenim prozorom usrednjavanja (crtkani kvadrat), (c) popunjeni piksel označen sa sivom bojom.
- Slika 6.9: $I_S(x', y')$ slika nakon uklanjanja rotacije: (a) za ulaznu sliku prikazanu na slici 6.5 (a), (b) za ulaznu sliku prikazanu na slici 6.5 (c).
- Slika 6.10: Izračun R_{max}^a i R_{min}^a koordinata s dvije niti (a), izračun S_{max}^b i S_{min}^b koordinata s dvije niti (b).
- Slika 6.11: Postupak popunjavanja piksela pozadine $I_S(x', y')$ slike na dvije niti (a), postupak popunjavanja piksela pozadine $I_S(x', y')$ slike na četiri niti (b).
- Slika 6.12: Referentne slike za izračun pogreške predložene metode za uklanjanje utjecaja rotacije sirove keramičke pločice.
- Slika 6.13: Prosječna pogreška (a) $Err_{SP}^{\%}$ i (b) $Err_{GT}^{\%}$ za šest referentnih slika.
- Slika 6.14: Prosječno vrijeme izvođenja: (a) sekvencijalnog algoritma predložene metode za ujednačavanje slike površine sirove keramičke pločice za različite kutove rotacije, (b) OpenCV metode za geometrijsku transformaciju.
- Slika 6.15: Prosječno vrijeme izvođenja paralelnog algoritma predložene metode za ujednačavanje slike površine sirove keramičke pločice za različite kutove rotacije, (b) ubrzanje u odnosu na metodu za geometrijsku transformaciju.
- Slika 7.1: Prikaz parametara MALD metode za (x, y) piksel.
- Slika 7.2: Primjer detektiranih linijskih neispravnosti MALD metodom.
- Slika 7.3: Parametar φ GLC matrice (a), (b) primjer parova (zaokruženi) $I(x, y) = 1$ i $I(x_p, y_p) = l$ za $d = 1$ i $\varphi = 45^\circ$.

Popis tablica

- Tablica 2.1: Podjela keramičkih pločica na temelju procesa proizvodnje
- Tablica 3.1: Količina neispravnih sirovih pločica (škarta) po godinama.
- Tablica 3.2: Prosječno vrijeme [ms] uklanjanja geometrijske distorzije s OpenCV bibliotekom.
- Tablica 4.1: Minimalna (*min*) i maksimalna (*max*) vrijednost devijacije indeksa SCD metode n_t za 3000 slučajeva signala s s brojem uzoraka $N = 40$, $\Delta A = 20$, $n_{Gt} = N/2 = 20$ i $\Delta A/\sigma \in [1.0, 5.0]$ u koracima od 0.5.
- Tablica 4.2: Minimalna (*min*) i maksimalna (*max*) vrijednost devijacije indeksa SCD metode nt za 3000 slučajeva signala s s brojem uzoraka $N = 60$, $\Delta A = 20$, $n_{Gt} = N/2 = 30$ i $\Delta A/\sigma \in [1.0, 5.0]$ u koracima od 0.5.
- Tablica 4.3: Primjer broja operacija Otsu algoritma 4.1 i PBSCD algoritma 4.5 prema (4.28), (4.29), (4.30) i (4.31).
- Tablica 4.4: Primjer broja operacija PBSCD algoritma s 1D integralnom sumom prema (4.35) i (4.36).
- Tablica 4.5: Latencije (u broju taktova) za pojedine tipove memorija u CUDA GPU.
- Tablica 4.6: Minimalne i maksimalne vrijednosti $\Delta A/\sigma$ za primjere $S1$, $S2$, $S3$ i $S4$.
- Tablica 4.7: Razine praga T razmatranih metoda za primjere $S1$, $S2$, $S3$ i $S4$.
- Tablica 4.8: Pogreška binarizacije ME u postocima za primjere $S1$, $S2$, $S3$ i $S4$ razmatranih metoda (MRE – metoda minimalne razlika entropija, MMP – metoda minimalne pogreške). Ocjena '1' predstavlja najbolji ME rezultat, ocjena '4' predstavlja najgori ME rezultat.
- Tablica 4.9: Prosječno vrijeme izvođenja (100 iteracija) u ms razmatranih metoda za sliku veličine 1038×1098 piksela (za PBSCD metodu $N = 120$) u MATLAB okruženju. Ocjena '1' predstavlja najkraće prosječno vrijeme izvođenja, ocjena '4' predstavlja najduže prosječno vrijeme izvođenja.
- Tablica 4.10: Prosječno vrijeme izvođenja (100 iteracija) u ms Otsu i PBSCD metode za sliku veličine 1038×1098 piksela u Visual Studio 2010 okruženju i C++ programskom jeziku.
- Tablica 4.11: Prosječno vrijeme izvođenja (100 iteracija) u ms CUDA PBSCD algoritma za sliku veličine 1098×1038 piksela u Visual Studio 2010 okruženju i CUDA C/C++ programskom jeziku.
- Tablica 5.1: Oblik i vrijednosti memorijske tablice za određivanje koordinata točaka ruba u $N_5^{D(i)}$ susjedstvu na temelju trenutnog smjera konture $D(i)$.
- Tablica 5.2: Vrijednosti uzorkovane Gaussove funkcije $g(t)$ (koeficijenti digitalnog Gaussovog filtera) za $\sigma = 2.5$ i $t = [-7, 7]$ s korakom 1.
- Tablica 5.3: Određivanje vrijednosti funkcije $E_U(Z_k^0, Z_{5-k}^{180^\circ})$.
- Tablica 5.4: Rezultati detekcije neispravnosti na rubovima i uglovima sirovih pločica u laboratorijskim uvjetima za 100 slika sirovih pločica.
- Tablica 5.5: Rezultati detekcije neispravnosti na rubovima i uglovima sirovih pločica u industrijskim uvjetima za 1000 slika sirovih pločica.
- Tablica 5.6: Veličina i duljina konture testnih slika za mjerenje srednjeg vremena izvođenja metode za detekciju neispravnosti na rubovima i uglovima sirovih pločica.

- Tablica 6.1: $Err_{SP}^{\%}$ pogreška predložene metode za uklanjanje utjecaja rotacije.
- Tablica 6.2: $Err_{GT}^{\%}$ pogreška uklanjanja utjecaja rotacije sa geometrijskom transformacijom i bikubičnom interpolacijom.
- Tablica 6.3: $Err_{SP}^{\%}$ pogreška predložene metode za uklanjanje utjecaja rotacije izražena u %.
- Tablica 6.4: $Err_{GT}^{\%}$ pogreška uklanjanja utjecaja rotacije sa geometrijskom transformacijom i bikubičnom interpolacijom izražena u %.
- Tablica 6.5: Veličina testnih slika za mjerenje srednjeg vremena izvođenja metode za ujednačavanje slike sirove keramičke pločice.
- Tablica 6.6: Prosječno vrijeme izvođenja predložene metode (*SP*) za $\gamma = 0.7^{\circ}$ i geometrijske transformacije (*GT*).
- Tablica 7.1: Prosječno vrijeme izvođenja predložene metode (*SP*) za $\gamma = 0.7^{\circ}$ i geometrijske transformacije (*GT*).

Popis algoritama

- Algoritam 4.1: Izračun razine praga Otsu metodom.
- Algoritam 4.2: Izračun razine praga metodom minimalne pogreške.
- Algoritam 4.3: Izračun razine praga metodom minimalne razlike entropija na temelju histograma slike.
- Algoritam 4.4: Izračun n_t SCD metodom.
- Algoritam 4.5: Binarizacija sive slike PBSCD metodom.
- Algoritam 4.6: SCD metoda s 1D integralnom slikom.
- Algoritam 4.7: Procedura izvođenja CUDA PBSCD algoritma.
- Algoritam 5.1: Određivanje točkaka (piksela) ruba predloženom metodom za detekciju ruba.
- Algoritam 5.2: Praćenje konture sirove pločice predloženom metodom.
- Algoritam 5.3: Pseudo kod algoritma za detekciju i lokalizaciju neispravnosti na rubovima i uglovima sirovih pločica.
- Algoritam 6.1: Pseudo kod algoritma za uklanjanje pozadine od površine sirove keramičke pločice.
- Algoritam 6.2: Pseudo kod algoritma za uklanjanje utjecaja rotacije sirove keramičke pločice.

1. UVOD

Čovjeku su proizvodi od keramike poznati od davnina. Koristi ih za različite namjene zbog dobrih fizikalnih i mehaničkih svojstva kao što su dugotrajnost, čvrstoća, kemijska inertnost i sl. Neki od keramičkih proizvoda su opeka, crijep, keramičke cijevi, vatrostalna keramika podne i zidne keramičke pločice, sanitarna keramika i dr. [1]. Zbog svojih dobrih svojstava, podne i zidne keramičke pločice se sve više upotrebljavaju u svakodnevnom životu. Prosječno, svake godine proizvodnja keramičkih pločica raste za 10 % [2]. S obzirom na navedeno, potrebno je da proces proizvodnje keramičkih pločica bude što više automatiziran.

Keramičke pločice proizvode se s različitim postupcima proizvodnje ovisno o broju ciklusa paljenja i nanošenju glazure. U ovoj disertaciji se razmatraju glazirane keramičke pločice s jednim paljenjem (monopaljenje). Proces proizvodnje glaziranih keramičkih pločica procesom monopaljenja sastoji se od nekoliko faza: rudarenje i transport sirovine, skladištenje sirovine, priprema sirovine, prešanje i sušenje pločice, nanošenje glazure, paljenje, vizualna kontrola kvalitete, sortiranje i pakiranje keramičke pločice [3].

Većina navedenih faza je automatizirana. Najmanje automatizirane faze proizvodnje su rudarenje i transport sirovine te vizualna kontrola kvalitete keramičke pločice. Vizualna kontrola kvalitete izvodi se nakon procesa pečenja. Obavljaju je obučeni radnici (kontrolori) koji na temelju postavljenih standarda (ISO, EN [4, 5]) i svojeg iskustva ocjenjuju kvalitetu keramičkih pločica. Kontrola kvalitete ovisi o subjektivnom dojmu kontrolora na kojeg utječe umor, nedostatak pažnje, bolest i sl. Potrebno je izbjeći ljudski faktor tj. automatizirati fazu vizualne kontrole kvalitete keramičke pločice. Tijekom procesa proizvodnje nastaju tri međuproizvoda: tijelo pločice ili keks, sirova pločica i pečena keramička pločica. Budući da je cilj otkriti neispravne pločice što ranije u procesu proizvodnje vizualna kontrola kvalitete može se podijeliti na tri dijela: vizualna kontrola kvalitete tijela pločice, vizualna kontrola kvalitete sirove keramičke pločice i vizualna kontrola pečene keramičke pločice. Vizualna kontrola kvalitete sirove keramičke pločice omogućila bi otkrivanje neispravnosti i odvajanje pločice prije ulaska u pećnicu. Na ovaj način smanjila bi se potrošnja energije potrebne za pečenje, kao i energija potrebna za reciklažu neispravne pečene keramičke pločice (nepečena keramička pločica se može vratiti u pripremljenu sirovinu) [6, 7].

U industriji se mogu pronaći gotovi automatizirani sustavi za vizualnu kontrolu kvalitete sirovih keramičkih pločica temeljeni na računalnom vidu, ali objavljeni radovi o metodama koje upotrebljavaju kao i o uspješnosti detekcije nisu dostupni [8]. U [9] autori su

prezentirali ASSIST sustava za vizualnu kontrolu kvalitete keramičkih pločica. Točnost klasifikacije sustava za određene vrste teksturiranih pločica je oko 70 % što nije zadovoljavajuće za industrijske standarde. Također, iako sustav koristi specijalizirani modul za obradu slike temeljen na mreži DSP procesora (*engl. Digital Signal Processor*) rad u stvarnom vremenu nije u potpunosti ostvaren.

U ovom radu razmatrane su metode za vizualnu kontrolu kvalitete sirovih keramičkih pločica. Vizualna kontrola kvalitete sirovih keramičkih pločica predstavlja manje složen proces od kontrole kvalitete keramičkih pločica. Razlog se može pronaći u manjem broju vrsta neispravnosti koje se pojavljuju na sirovim keramičkim pločicama. Za testiranje razvijenih metoda razvijen je prototip sustava temeljen na LED difuznoj rasvjeti, digitalnoj kameri i paralelne procesnoj jedinici. Siva slika dobivena sa digitalnom kamerom obrađuje se na paralelnoj procesnoj jedinici koja se sastoji od višejezgrenih procesora i paralelne grafičke procesne jedinice. Obrada sive slike podijeljena je u nekoliko faza: predobrada sive slike, binarizacija – izdvajanje sirove keramičke pločice od pozadine, detekcija neispravnosti na uglovima i rubovima sirove keramičke pločice, ujednačavanje slike površine sirove keramičke pločice i detekcija neispravnosti na površini sirovih keramičkih pločica. U ovoj disertaciji razvijene su i testirane nove metode za izdvajanje sirove keramičke pločice od pozadine, detekciju neispravnosti na uglovima i rubovima sirove keramičke pločice i ujednačavanje slike površine sirove keramičke pločice. Također, obrađena je tematika predobrade sive slike i detekcije neispravnosti na površini sirovih keramičkih pločica.

Cilj disertacije je razviti metode koje će omogućiti pravilno izdvajanje sirove keramičke pločice od pozadine, detektirati neispravnosti na rubovima i uglovima sirovih pločica kao i pripremiti sliku sirove keramičke pločice za detekciju neispravnosti na površini. Sve metode moraju omogućiti rad sustava u stvarnom vremenu. Paralelna procesna jedinica mora izvesti sve navedene faze u postavljenom vremenskom roku. Vremenski interval je postavljen od strane samog procesa proizvodnje keramičkih pločica. Na jednoj proizvodnoj liniji u 1 *min* proizvede se do 100 keramičkih pločica. Iz navedenog može se zaključiti da obrada slike mora završiti u ograničenom vremenskom intervalu od 600 *ms* da bi bio omogućen rad u stvarnom vremenu.

Disertacija je organizirana prema sljedećem redoslijedu. U drugom poglavlju opisana je proizvodnja i kontrola kvalitete keramičkih pločica. Opisan je proces proizvodnje, automatiziranost pojedine faze procesa proizvodnje te kontrola kvalitete keramičkih pločica.

U trećem poglavlju kategorizirane su neispravnosti koje se pojavljuju na sirovim keramičkim pločicama. Predstavljen je i analiziran prototip sustava za detekciju neispravnosti

na sirovim pločicama, te faze obrade prikupljene sive slike sirove pločice. Također, su obrađene metode za predobradu sive slike sirove keramičke pločice.

U četvrtom poglavlju predstavljene su postojeće metode za izdvajanje slike keramičke pločice od pozadine. Predstavljena je i detaljno analizirana nova metoda za izdvajanje sirove keramičke pločice od pozadine. Razvijen je i implementiran sekvencijalni i paralelni algoritam za predloženu metodu. Metoda je testirana na mogućnost rada u stvarnom vremenu i određena je pogreška binarizacije predložene metode. Rezultati su uspoređeni s postojećim metodama za binarizaciju sive slike keramičke pločice.

U petom poglavlju razvijena je unaprijeđena metoda za detekciju i lokalizaciju neispravnosti na rubovima i uglovima sirovih keramičkih pločica. Predstavljene su postojeće metode za detekciju neispravnosti na rubovima i uglovima keramičkih pločica. Analizirana je mogućnost detekcije neispravnosti predložene metode na sintetičkim slikama sirovih pločica. Metoda je testirana u laboratorijskim i industrijskim uvjetima, te su rezultati uspoređeni s rezultatima dobivenih od operatera za kontrolu kvalitete keramičkih pločica. Razvijen je i implementiran sekvencijalni i paralelni algoritam predložene metode. Algoritmi su testirani na mogućnost rada u stvarnom vremenu.

U šestom poglavlju razvijena je nova metoda za ujednačavanje slike površine sirove keramičke pločice. Predstavljene su i analizirane postojeće metode u literaturi. Objašnjeni su pojedini koraci nove metode te su razvijeni i implementirani sekvencijalni i paralelni algoritmi u C++ programskom jeziku. Definirana je pogreška metode te je ista uspoređena sa pogreškom standardne metode za uklanjanje utjecaja rotacije pločice. Razvijeni sekvencijalni i paralelni algoritmi testirani su na mogućnost rada u stvarnom vremenu.

U sedmom poglavlju predstavljene su i analizirane postojeće metode za detekciju neispravnosti na površini sirovih keramičkih pločica. Detaljno su obrađene odabrane metode koje su pokazale dobre rezultate detekcije neispravnosti na površini keramičkih pločica.

Ostvareni znanstveni doprinosi i zaključak disertacije dani su u osmom poglavlju.

2. PROIZVODNJA I KONTROLA KVALITETE KERAMIČKIH PLOČICA

U ovom poglavlju dana je definicija, svojstva i podjela keramičkih proizvoda, predstavljen je proces proizvodnje keramičkih pločica i objašnjene su pojedine faze procesa proizvodnje.

2.1. Keramika – povijest, definicija, svojstva i podjela

Izraz keramika potječe iz grčkog jezika (*grk. keramos – pečena zemlja*) i označava anorganski materijal dobiven pečenjem (paljenjem) različitih ne-metalnih smjesa. U ne-metalne smjese spadaju različite vrste gline pomiješane sa različitim mineralima kao što su kvarc, kalcijev karbonat i sl. Proizvodnja keramike čovjeku je poznata od davnina. Najstarija nalazišta keramike se nalaze u južnom Japanu i datiraju oko 8000 godina prije nove ere. Egipćani su proizvodili keramičke ploče sa glazurom 2600. godine prije nove ere, a u Kini izrada porculana je poznata od 1000. godine prije nove ere. Razlog široke upotrebe keramičkih proizvoda može se pronaći u dobrim fizikalnim i mehaničkim svojstvima. Neka od svojstava proizvoda od keramike su: dugotrajnost, čvrstoća, kemijska inertnost, vodonepropustnost (vodopropusnost), toplinska otpornost i sl. [1].

Zbog navedenih svojstava tijekom ljudske povijesti razvijeno je niz proizvoda od keramike. Keramika se upotrebljava u građevinarstvu, strojarstvu, kemiji, elektrotehnici, automobilske industriji, domaćinstvu, biomedicini i sl. Na temelju povijesnih razloga proizvodi od keramike mogu se podijeliti u dvije grupe:

- građevinski keramički proizvodi i
- tradicionalni i industrijski keramički proizvodi.

U građevinske keramičke proizvode spadaju:

- Opeka i crijep,
- keramičke cijevi,
- vatrostalna keramika i
- ekspanzirane keramičke granule.

U tradicionalne i industrijske keramičke proizvode ubrajaju se:

- Podne i zidne keramičke pločice,
- posuđe i ukrasna keramika,

- sanitarna keramika,
- tehnička keramika i
- abrazivna keramika.

Na slici 2.1 prikazani su primjeri proizvoda iz navedene podjele. Većina proizvodnje u Europi (89 %) otpada na proizvodnju opeke, crijeva, podnih i zidnih keramičkih pločica ostale grupe proizvoda su manje zastupljene zbog specijalizirane namjene istih [2]. U najveće proizvođače keramičkih proizvoda u Europi spadaju Italija, Španjolska, Turska, Njemačka i Poljska. Sve zemlje Europe proizvedu ukupno 15 % svjetske proizvodnje keramike. Kina je najveći svjetski proizvođač keramike čija proizvodnja obuhvaća 45 % ukupne svjetske proizvodnje keramike [3]. Svi navedeni podaci odnose se na 2011. godinu.

U nastavku ovog rada razmatra se proizvodnja i kontrola kvalitete podnih keramičkih pločica. Ostale grupe keramičkih proizvoda neće biti razmatrane.



Slika 2.1: Primjeri proizvoda od keramike: (a) opeka i cigla, (b) cijevi za otpadnu vodu, (c) vatrostalne ploče, (d) ekspanzirane keramičke granule, (e) podne keramičke pločice, (f) posuđe od keramike, (g) sanitarna keramika, (h) automobilska kočnica od keramike, (i) abrazivne ploče od keramike.

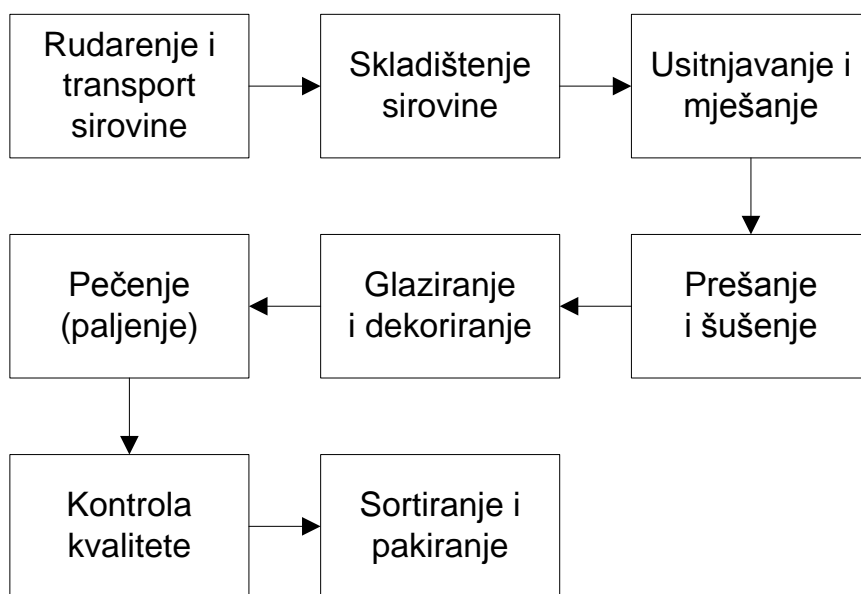
Tablica 2.1: Podjela keramičkih pločica na temelju procesa proizvodnje

| Tip keramičke pločice | Glazura | Ciklus paljenja | |
|-----------------------|---------|------------------|------------------|
| | | Prije glaziranja | Nakon glaziranja |
| Ne glazirane | ✗ | ✓ | ✗ |
| Dvostruko paljene | ✓ | ✓ | ✓ |
| Jednostruko paljene | ✓ | ✗ | ✓ |

2.2. Opis procesa proizvodnje keramičkih pločica

Keramičke se pločice proizvode u tri različita procesa proizvodnje. Procesi proizvodnje se razlikuju na osnovu broja ciklusa pečenja (paljenja) i u korištenju glazure. Na temelju navedenog razlikujemo proces proizvodnje sa dvostrukim paljenjem i jednim paljenjem (monopaljenje), sa i bez glazure, v. tablicu 2.1. Slika 2.2 prikazuje shemu procesa proizvodnje glaziranih pločica procesom monopaljenja. Proces proizvodnje glaziranih pločica monopaljenjem sastoji se od nekoliko faza:

- Rudarenje (iskapanje) sirovine i njezin transport,
- skladištenje rudarene sirovine,
- priprema mase (šlikera) i priprema granulata iz šlikera,
- prešanje granulata i sušenje prešanih pločica,
- nanošenje glazure na prešanu pločicu,
- paljenje sirove pločice,
- kontrola kvalitete keramičke pločice i



Slika 2.2: Shematski prikaz procesa proizvodnje keramičkih pločica monopaljenjem.

- sortiranje i pakiranje gotove keramičke pločice.

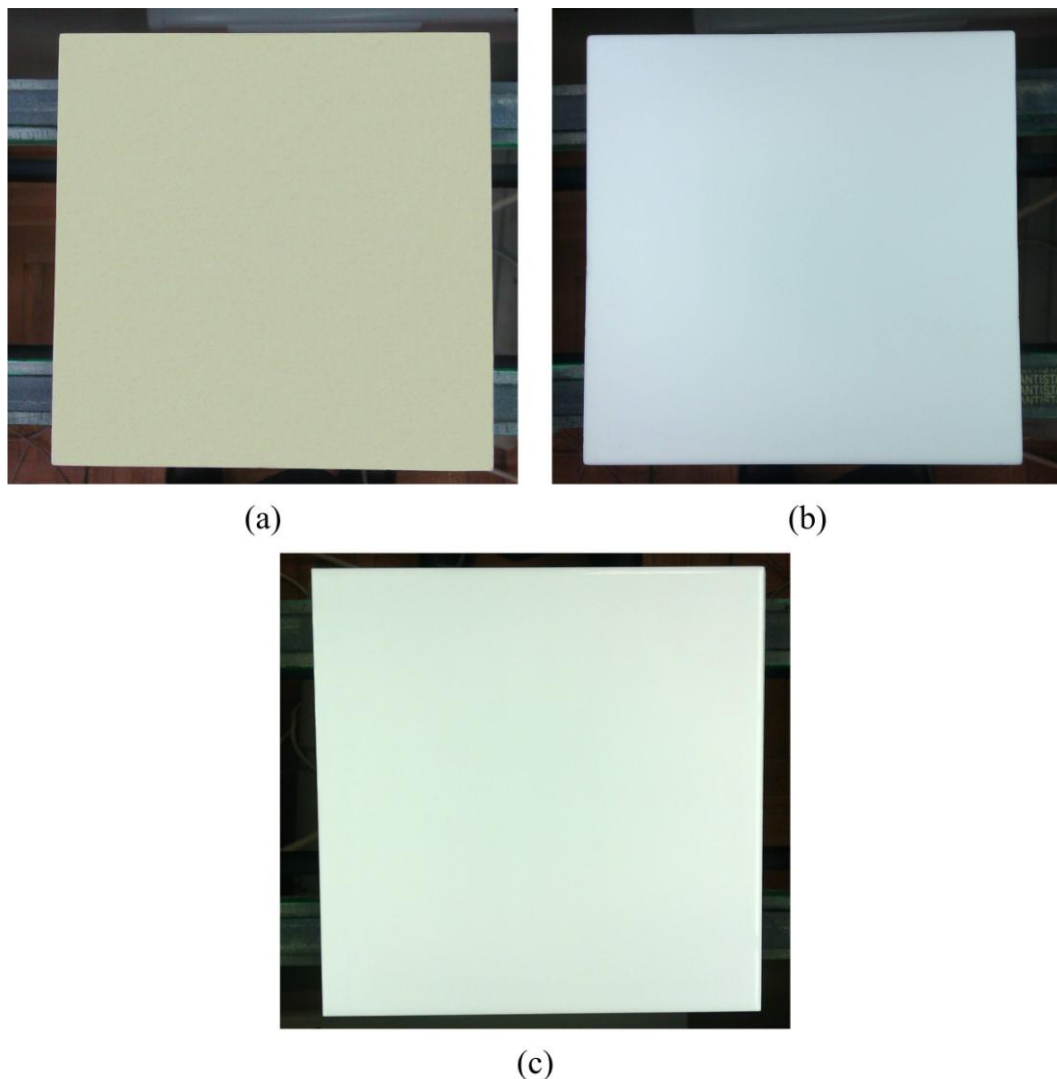
Velik broj ulaznih sirovina upotrebljava se u proizvodnji keramičkih pločica. Najveći udio u samoj smjesi pločice zauzimaju glina i dolomit. Ostatak smjese sastoji se od minerala od kojih su najčešći hidratizirani aluminijski silikati. Sirovina se dobiva iz rudnika ili sa riječnih obala, te se prevozi kamionima, vlakom ili pokretnim trakama ako se rudnik nalazi u blizini same tvornice. Ovisno o svojstvima pojedine sirovine, sirovina se može skladištiti na otvorenom prostoru, u zatvorenim skladištima ili u silosima. Prije samog usitnjavanja i miješanja može se također dodatno obrađivati sirovina. U dodatnu obradu sirovina ubrajaju se homogenizacija sirovine, predušenje sirovine i miješanje sirovine tijekom iskapanja.

Uskladišteni sirovi materijal je u obliku manje ili više krutog grumenja. Grumenje se usitnjava kombinacijom raznih vrsta drobilica (čeljusna drobilica, drobilica s valjcima) i mlinova (mlin s kuglama, mlin s okretnom pločom). Rezultat usitnjavanja u mlinovima i miješanja s dodatkom vode je prah promjera zrna manjim od 0.2 mm. Dobiveni prah se naziva šliker i sadrži od 60 do 70 % suhe tvari. Šliker se atomizira pod tlakom i suši u struji vrućeg zraka. Na ovaj način se dobivaju okrugle šuplje granule s 4 % do 7 % vlage. Granule se prosljeđuju u preše. Postoji više vrsta preša (mehaničke preše, hidrauličke preše itd.) ali se u proizvodnji keramičkih pločica, zbog jakih tlačnih sila i visoke produktivnosti, najviše koriste hidrauličke preše. Prešanje se vrši u kalupima željenih dimenzija pločice. Dobiveni kompaktni prešani oblik (tijelo pločice) se šalje na sušenje. U proizvodnji keramičkih pločica najčešće se koriste vertikalne sušnice. Sušnica se sastoji od košara s valjcima na kojima se nalaze prešane pločice. Košare se gibaju vertikalno prema gore tijekom čega suhi zrak na temperaturi od 200 °C suši prešana tijela. Ciklus sušenja traje od 35 do 50 min. Osušeno tijelo pločice (*engl. green tile – keks*), v. sliku 2.3 (a), na pokretnoj traci prolazi kroz proces glaziranja. Glavni sastojak glazure je frita, a pored nje glazura se sastoji i od silike, bojila i drugih aditiva kao što su olovo, bor, zemnoalkalijski metali, itd. Frita se sastoji od staklastih materijala netopljivih u vodi. Priprema se topljenjem kristalnih materijala na temperaturama od oko 1500 °C. Na osnovu raznih komponenata glazure i temperatura paljenja dobivaju se različiti efekti na površini keramičke pločice. Glazura se najčešće na tijelo pločice nanosi u tekućem obliku prolaskom tijela pločice ispod „vodopada“ tekuće glazure. Dodatno ukrašavanje površine pločice omogućeno je različitim tehnikama za nanos novih slojeva glazure (sitotisak, tisak inkjet tehnikom i sl.).

Glazirana – sirova pločica (*engl. biscuit tile*), v. sliku 2.3 (b), ulazi u proces paljenja. Paljenje predstavlja najbitniju fazu u proizvodnji keramičkih pločica. Paljenjem (trajanjem i

temperaturom) se postižu glavna svojstva pločica kao što su čvrstoća, kemijska inertnost, vodonepropustnost, toplinska otpornost i sl. Također, nakon paljenja pločica poprima konačnu boju. Temperature paljenja za keramičke pločice kreću se od 1050 °C do 1200 °C. Na navedenim temperaturama krhka sirova pločica pretvara se u čvrstu strukturu s staklastom površinom. U proizvodnji keramičkih pločica za proces paljenja najčešće se koriste ruli peći (*engl. kiln*). Sirove pločice putuju kroz ruli peć na valjcima. Toplina se generira plamenicima na zemni plin koji se nalaze s obje strane peći. Rezultat procesa paljenja, koji u ruli peći traje kraće od 40 *min*, je gotova keramička pločica, v. sliku 2.3 (c).

Prije isporuke keramičke pločice kupcima, proizvod se mora pregledati od strane kvalificiranog operatera za kontrolu kvalitete. U fazi kontrole kvalitete keramičke pločice operater svakoj keramičkoj pločici dodjeljuje ocjenu kvalitete, npr. I., II. i III klasa. Ocjena



Slika 2.3: Međuproizvodi u procesu proizvodnje keramičkih pločica: (a) Osušena prešana pločica – keks, (b) sirova – glazirana pločica, (c) keramička pločica.

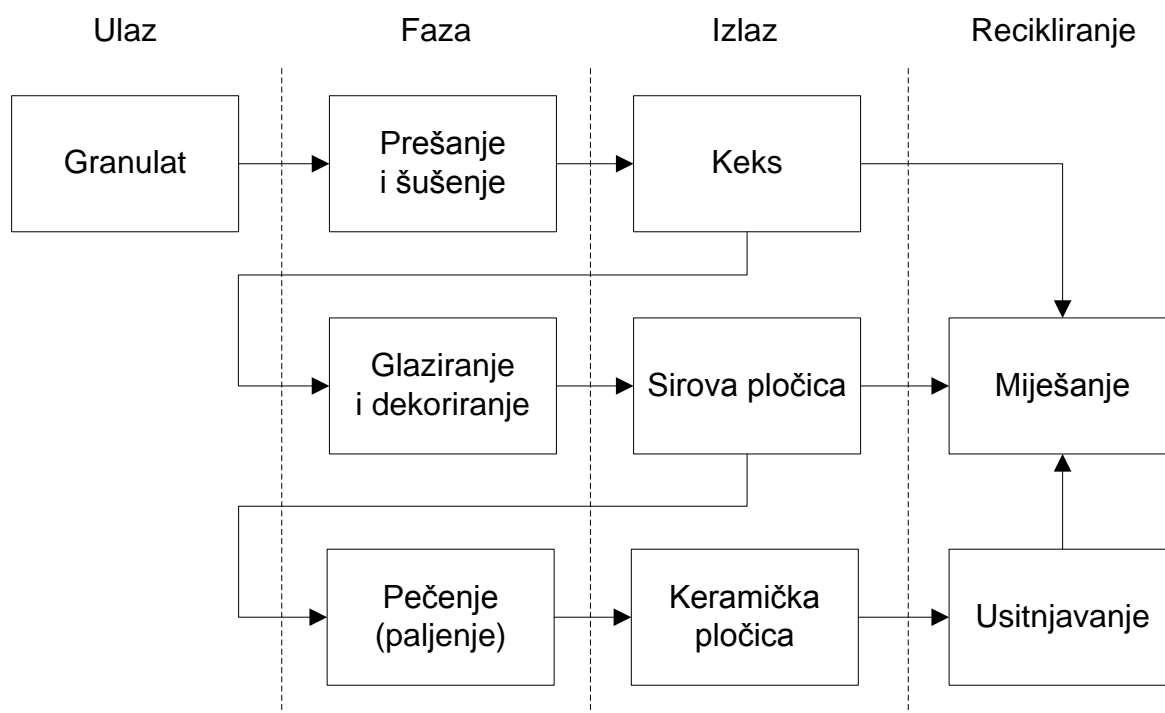
kvalitete se određuje na temelju značajki definiranih u standardima (ISO, EN) i na temelju subjektivnog dojma samog operatera (dva različita operatera za kontrolu kvalitete mogu istoj keramičkoj pločici dodijeliti različite ocjene kvalitete). Ocjenjena keramička pločica prosljeđuje se u fazu sortiranja i pakiranja. Nakon sortiranja, pločice se pakiraju u kartonske kutije, slažu se na drvene palete i omataju sa streč-folijom. Palete se odvoze u skladište gdje se čuvaju do distribucije.

Tijekom procesa proizvodnje moguće je reciklirati međuproizvode, cf. slika 2.3, koji ne zadovoljavaju postavljene standarde. Međuproizvodi se recikliraju vraćanjem u određenu fazu procesa proizvodnje keramičkih pločica. Osušena prešana pločica i sirova pločica se vraćaju u proces miješanja sirovine, dok keramička pločica odlazi u postupak usitnjavanja, v. sliku 2.4 [1, 3, 6].

2.3. Automatiziranost procesa proizvodnje keramičkih pločica

Proces proizvodnje keramičkih pločica sastoji se od više faza kako je i objašnjeno u prethodnom poglavlju. Navedene faze su više ili manje automatizirane. Autor u [6] kvalitativno je analizirao stupanj automatiziranosti pojedinih faza u procesu proizvodnje keramičkih pločica.

Sa niskim stupnjem automatiziranosti ocjenjensu su faze rudarenja, transporta i



Slika 2.4: Shematski prikaz recikliranja međuproizvoda tijekom proizvodnje keramičkih pločica.

skladištenja sirovine kao i faza vizualne kontrole kvalitete keramičke pločice. Faza rudarenja, transporta i skladištenja sirovine oslanja se na tešku mehanizaciju i čovjeka za nadzor i upravljanje. Također, u fazi vizualne kontrole kvalitete keramičke pločice nije postignut dovoljan stupanj automatizacije zbog nemogućnosti ispunjenja zahtjeva postavljenih od strane industrije. Automatizirani sustav još nije postignuo dovoljnu preciznost i fleksibilnost koju pruža čovjek sa svojim vizualnim sustavom, razinom učenja i klasifikacije.

Sa srednjim stupnjem automatiziranosti ocijenjena je faza glaziranja i dekoriranja. Postupak se izvodi strojno ali nadzor postupka i opsluživanje strojeva obavlja se ručno od strane čovjeka.

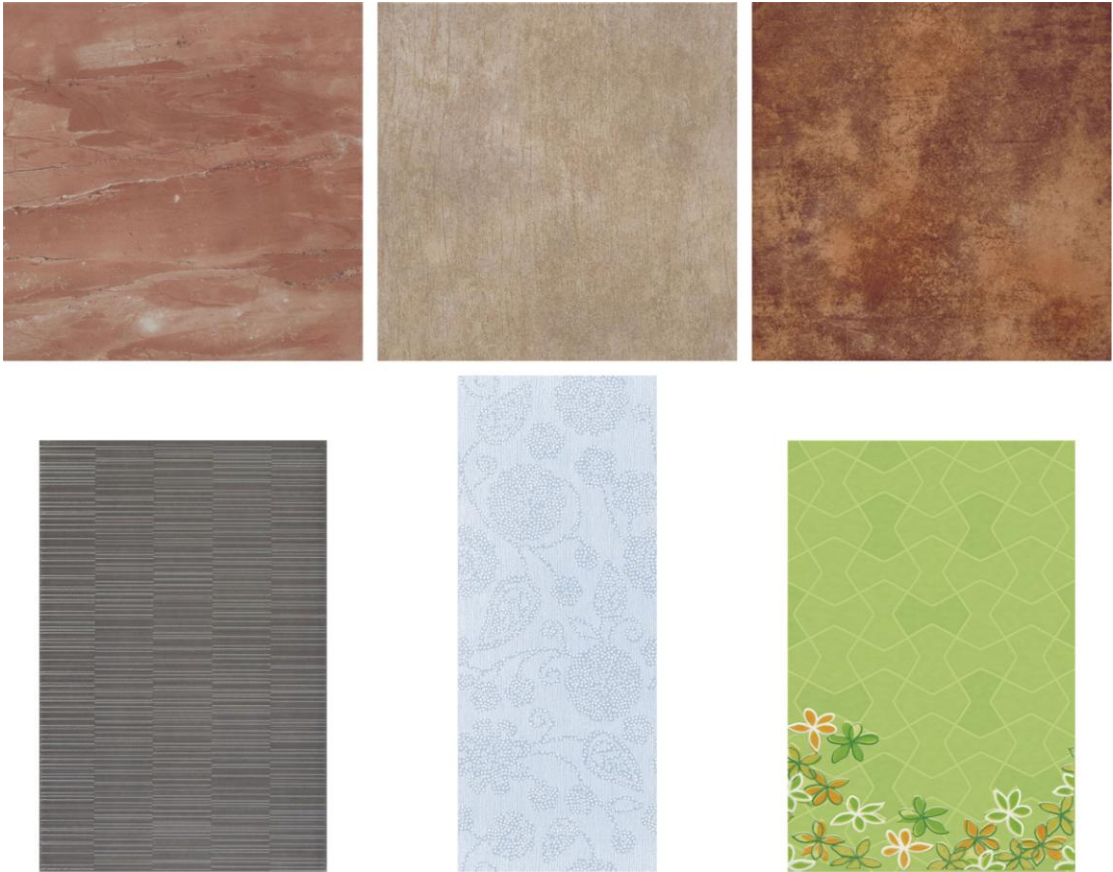
Sa visokim stupnjem automatiziranosti ocijenjene su ostale faze procesa proizvodnje keramičkih pločica u koje spadaju priprema šlikera i priprema granulata iz šlikera, prešanje granulata i sušenje prešanih pločica, paljenje sirove pločice i sortiranje i pakiranje gotove keramičke pločice. Navedene faze sastoje se od različitih strojnih sustava koji izvode sve potrebne radnje i pregradnje. Svrha čovjeka je u nadzoru i eventualnim intervencijama nad strojnim sustavima.

2.4. Kontrola kvalitete keramičkih pločica

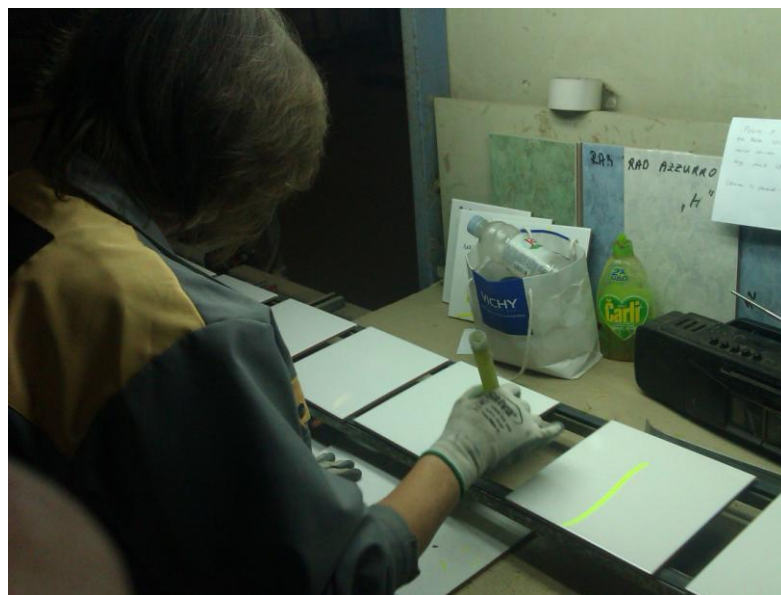
Kontrola kvalitete u procesu proizvodnje keramičkih pločica može se provoditi nad svim međuproizvodima keramičkih pločica (keks, sirova keramička pločica i pečena keramička pločica). U današnjoj industrijskoj proizvodnji keramičkih pločica najčešće se koriste ljudski resursi za vizualnu kontrolu kvalitete keramičkih pločica, a pored njih i strojni sustavi za određivanje geometrije (kalibra) i planarnosti pločice [10]. Ljudski resursi se upotrebljavaju zbog mogućnosti brze prilagodbe čovjeka (učenje, mogućnost klasifikacije) različitim tipovima keramičkih pločica, cf. slika 2.5. Operater vizualnu kontrolu provodi samo na pečenoj keramičkoj pločici, v. sliku 2.6. Ocjenu kvalitete donosi na temelju postavljenih normi i subjektivnog dojma.

Prvak klasa kvalitete keramičkih pločica definirana je ISO 13006:2012 i EN 14411:2012 standardom [4, 5]. U standardu su propisane podjela definicije i označavanje keramičkih pločica, proces proizvodnje, mehaničke i kemijske značajke i sl. Standardom su propisani zahtjevi nad geometrijskim značajkama koje uključuju duljinu, širinu, debljinu i pravokutnost keramičke pločice, linearnost rubova i opću kvalitetu površine pločice. Za kvalitetu površine standard propisuje da 95 % pločica ne smije imati vidljivih neispravnosti koje bi umanjile izgled većeg dijela površine pločice (vidljive ljudskim okom na jedan metar

udaljenosti). Ostale klase keramičkih pločica kao i detaljna analiza neispravnosti na površini keramičkih pločica nisu opisane. Također, nisu opisane ni klase za ostale međuproizvode u procesu proizvodnje keramičkih pločica. Pored navedenih standarda, postoje i drugi standardi razvijeni od različitih organizacija koji dodatno definiraju neispravnosti i klase keramičkih



Slika 2.5: Primjeri raznih vrsta keramičkih pločica (ustupila Keramika Modus d.o.o.).



Slika 2.6: Vizualna kontrola kvalitete keramičkih pločica (ustupila Keramika Modus d.o.o.).

pločica. Europska federacija proizvođača zidnih i podnih keramičkih pločica (*engl. European Federation of Ceramic Wall and Floor Tile Producers*) definira drugu klasu keramičkih pločica. Za drugu klasu dozvoljene su površinske neispravnosti nevidljive ljudskim okom na dva metra udaljenosti od pločice [11].

U industriji, kontrolor kvalitete slijedeći norme i vlastito iskustvo odlučuje koje keramičke pločice pripadaju I, II i III (otpad) klasi. Ovisno o pripadajućoj klasi pločice su označene fluorescentnim markerom te se na temelju iste sortiraju i pakiraju u sljedećoj fazi. Nad sirovim keramičkim pločicama i tijelu keramičke pločice operateri ne provode vizualnu inspekciju.

3. VIZUALNA INSPEKCIJA NEISPRAVNOSTI NA SIROVIM KERAMIČKIM PLOČICAMA

U ovom poglavlju opisane su neispravnosti na sirovim keramičkim pločicama. Također je prikazan prototip sustava za vizualnu inspekciju neispravnosti na sirovim keramičkim pločicama, te prikazane faze obrade slike sirove pločice.

3.1. Neispravnosti na sirovim keramički pločicama

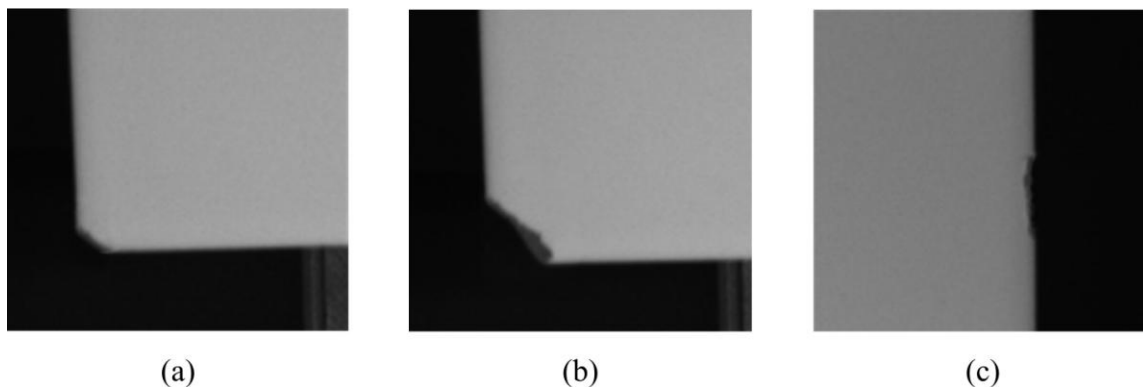
Neispravnosti na sirovim keramičkim pločicama mogu se podijeliti na:

- geometrijske neispravnosti i
- površinske neispravnosti.

U ovom radu se pretpostavlja da su pločice pravokutnog oblika bez reljefa na površini. S obzirom na navedeno u geometrijske neispravnosti se ubrajaju:

- odstupanje dimenzije sirove pločice i
- odstupanje od pravokutnog oblika.

U dimenzije pravokutne pločice ubrajaju se širina, visina i debljina. Odstupanja su propisana standardima (ISO, EN) te se najčešće upotrebljavaju optički laserski sustavi za mjerenje istih. U geometrijske neispravnosti odstupanja od pravokutnog oblika ubrajaju se neispravnosti na rubovima i neispravnosti na uglovima sirovih keramičkih pločica. Primjeri geometrijskih neispravnosti prikazani su na slici 3.1. Najčešći uzroci geometrijskih neispravnosti su nepravilnosti u fazi prešanja i sušenja te nepravilnosti u transportu međuproizvoda (keksa i sirove pločice). Veličina geometrijskih neispravnosti koje nastaju tijekom procesa proizvodnje je veća od 1 mm.



Slika 3.1: Geometrijske neispravnosti na sirovim pločicama: (a) i (b) neispravnosti na uglovima, (c) neispravnost na rubu.

Tablica 3.1: Količina neispravnih sirovih pločica (škarta) po godinama.

| | Godina proizvodnje | | |
|--------------------------------|--------------------|----------|----------|
| | 2010 | 2011 | 2012 |
| m^2 | 31650.52 | 27531.76 | 38535.68 |
| % | 2.10 | 2.26 | 2.01 |
| <i>Ukupno m^2</i> | 97717.96 | | |

U površinske neispravnosti na sirovim keramičkim pločicama ubrajaju se:

- točkaste neispravnosti,
- neispravnosti nastale od kapljica kondenzata,
- nedostatak glazure,
- nepravilnosti u teksturi,
- ispupčenja i udubljenja na površini.

Najčešće neispravnosti na sirovim keramičkim pločicama su: točkaste neispravnosti, neispravnosti nastale od kapljica kondenzata, nedostatak glazure i nepravilnosti u teksturi. U rjeđe neispravnosti spadaju ispupčenja i udubljenja na površini sirove keramičke pločice. Površinske neispravnosti nastaju najčešće u fazi glaziranja i dekoriranja i tijekom transporta sirove pločice, a rjeđe u fazi prešanja i sušenja. Keramička pločica poprima konačnu boju tek nakon procesa paljenja (v. poglavlje 2.2), stoga se neispravnosti površine vezane za boju pločice ne detektiraju na sirovoj keramičkoj pločici. Budući da se poslije pečenja keramičke pločice klasiraju u I i II klasu (III klasa predstavlja otpad), površinske neispravnosti moraju zadovoljiti uvjet da su nevidljive ljudskim okom na udaljenosti od dva metra. Veličine neispravnosti koje je potrebno detektirati kreću se od promjera 0.5 mm za točkaste neispravnosti do nekoliko milimetara za neispravnosti nastale od kapljica kondenzata i ispupčenja i udubljenja na površini. Veličina nepravilnosti u teksturi i nedostatku glazure kreću se od nekoliko milimetara do nekoliko centimetara. Na slici 3.2 su prikazani primjeri navedenih neispravnosti na sirovim keramičkim pločicama.

Sirove keramičke pločice se ne klasificiraju u klase kao u slučaju keramičkih pločice. Ako je detektirana bilo koja od navedenih neispravnosti sirova pločica se preusmjerava u proces miješanja sirovine tj. reciklira. Ako na pločici nije detektirana niti jedna neispravnost ista nastavlja u fazu paljena.

Iz tablice 3.1 vidljivo je da godišnje prosječno 2 % ukupne proizvodnje sirovih pločica posjeduje neke od navedenih neispravnosti. S obzirom na broj sirovih keramičkih pločica sa neispravnosću (u tri godine više od 90000 m^2) i moguću uštedu energije, proizlazi zaključak

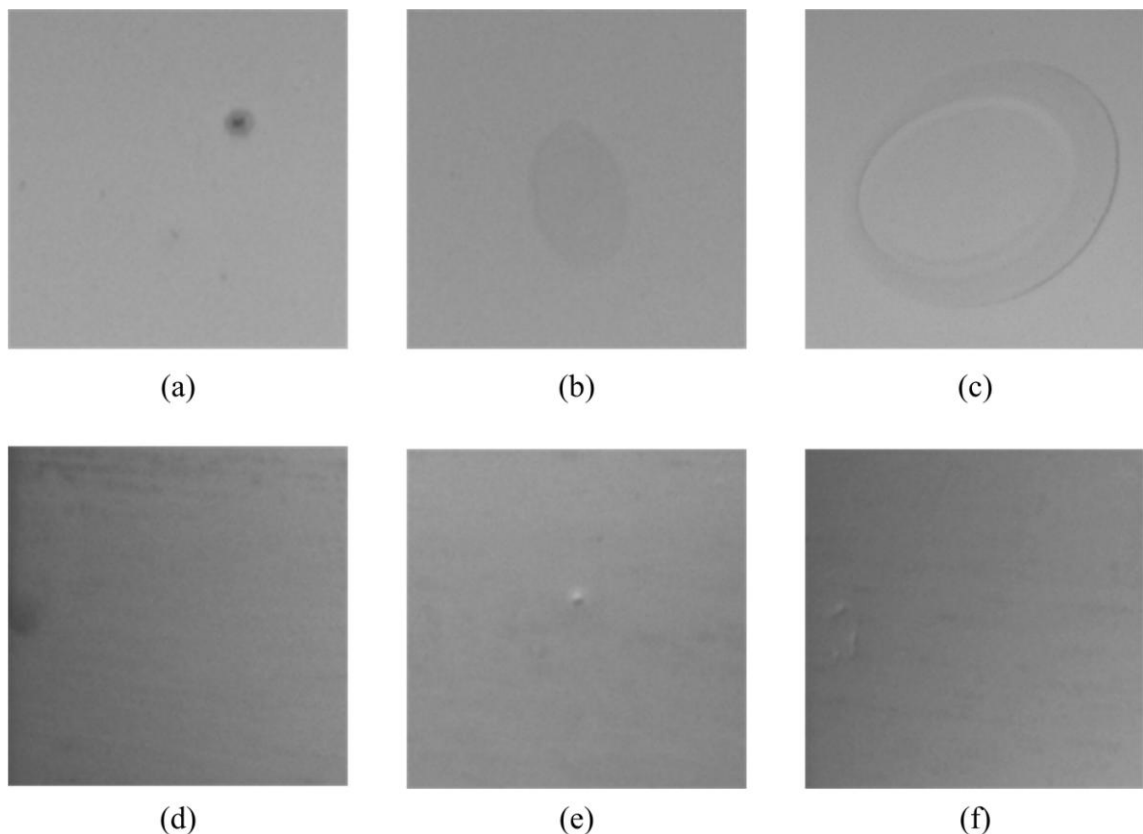
da je potrebno provoditi vizualnu inspekciju neispravnosti nad sirovim keramičkim pločicama.

U ovoj doktorskoj disertaciji razvijene su metode za detekciju neispravnosti na rubovima i uglovima sirovih keramičkih pločica, metode za pripremu slike za detekciju neispravnosti na površini sirovih pločica, te su predložena moguća rješenja za detekciju neispravnosti na površini.

3.2. Prototip sustava za detekciju neispravnosti na sirovim keramičkim pločicama

Za potrebe testiranja razvijenih metoda razvijen je prototip sustava za vizualnu inspekciju neispravnosti na sirovim keramičkim pločicama. Prototip sustava sastoji se od LED difuznih rasvjetnih tijela, digitalne kamere s matičnim osjetnikom, svjetlosnih senzora za detekciju dolaska sirove pločice i paralelne procesne jedinice.

LED difuzna rasvjetna tijela pokazala su se u literaturi kao najpogodnija za



Slika 3.2: Površinske neispravnosti na sirovim pločicama: (a) točkaste neispravnosti, (b) neispravnosti nastale od kapljica kondenzata, (c) nedostatak glazure, (d) nepravilnosti u teksturi, (e) ispupčenja i (f) udubljenja.

osvjetljavanje keramičkih pločica [12, 13]. LED difuzna rasvjetna tijela napajana s istosmjernom strujom ne trepere frekvencijom izvora napajanja (s vremenom se ne mijenja osvjetljenost objekta), pokazuju zadovoljavajuću uniformnost osvjetljenosti objekta i dovoljnu osvjetljenost objekta.

Za prikupljanje slike sirove keramičke pločice upotrijebljena je digitalna kamera Basler A102fc sa matričnim osjetnik i FireWire protokolom za prijenos podataka (za ostale značajke v. prilog 10.1). Pod pojmom „prikupljanje slike“ u ovom radu se smatra proces uzorkovanja digitalne slike i prijenos iste u memoriju procesne jedinice. Odabrana je kamera sa matričnim osjetnikom zbog svoje jednostavnosti i cijene. Budući da sirova pločica ne prikazuje konačnu boju keramičke pločice, prikuplja se siva slika iste. Vrijeme potrebno za prikupljanje sive slike nije veće od 70 ms. Vrijeme (trenutak) prikupljanja slike određen je sa svjetlosnim senzor koji detektira dolazak keramičke pločice na proizvodnoj liniji.

Prikupljena siva slika obrađuje se u paralelnoj procesnoj jedinici. Paralelna procesna jedinica temeljena je na osobnom računalu s Windows 7 x64 operacijskim sustavom, Intel Q6600 procesorom, 4 GB RAM memorije i Nvidia GTX 670 grafičkom karticom s 2 GB GDDR5 memorije. Paralelna obrada je omogućena s višejezgrenim (četiri jezgre) Intel Q6600 procesorom radne frekvencije 2.4 GHz i CUDA 3.0 Nvidia GTX 670 grafičkom karticom. Digitalnim izlazom sa procesne jedinice moguće je signalizirati ispravnost/neispravnost analizirane sirove keramičke pločice.

Na slici 3.3 (a) prikazan je model prototipa predloženog sustava, a na slici 3.3 (b) i (c) prikazan je razvijeni prototip implementiran na pokretnoj liniji u laboratoriju i na proizvodnoj liniji u keramičkoj industriji (Keramika Modus d. o. o. [14]). Kut rotacije sirove pločice je označen s γ . Da bi se što više smanjilo izobličenje slike sirove keramičke pločice, prilikom postavljanja prototipa z os od kamere treba biti poravnata sa Z osi keramičke pločice. Također, da bi rotacija objekta sirove pločice na slici bila što manja potrebno je ograničiti kut rotacije keramičke pločice γ . Kut rotacije moguće je ograničiti sa usmjerivačima kako je i prikazano na slici 3.3 (a).

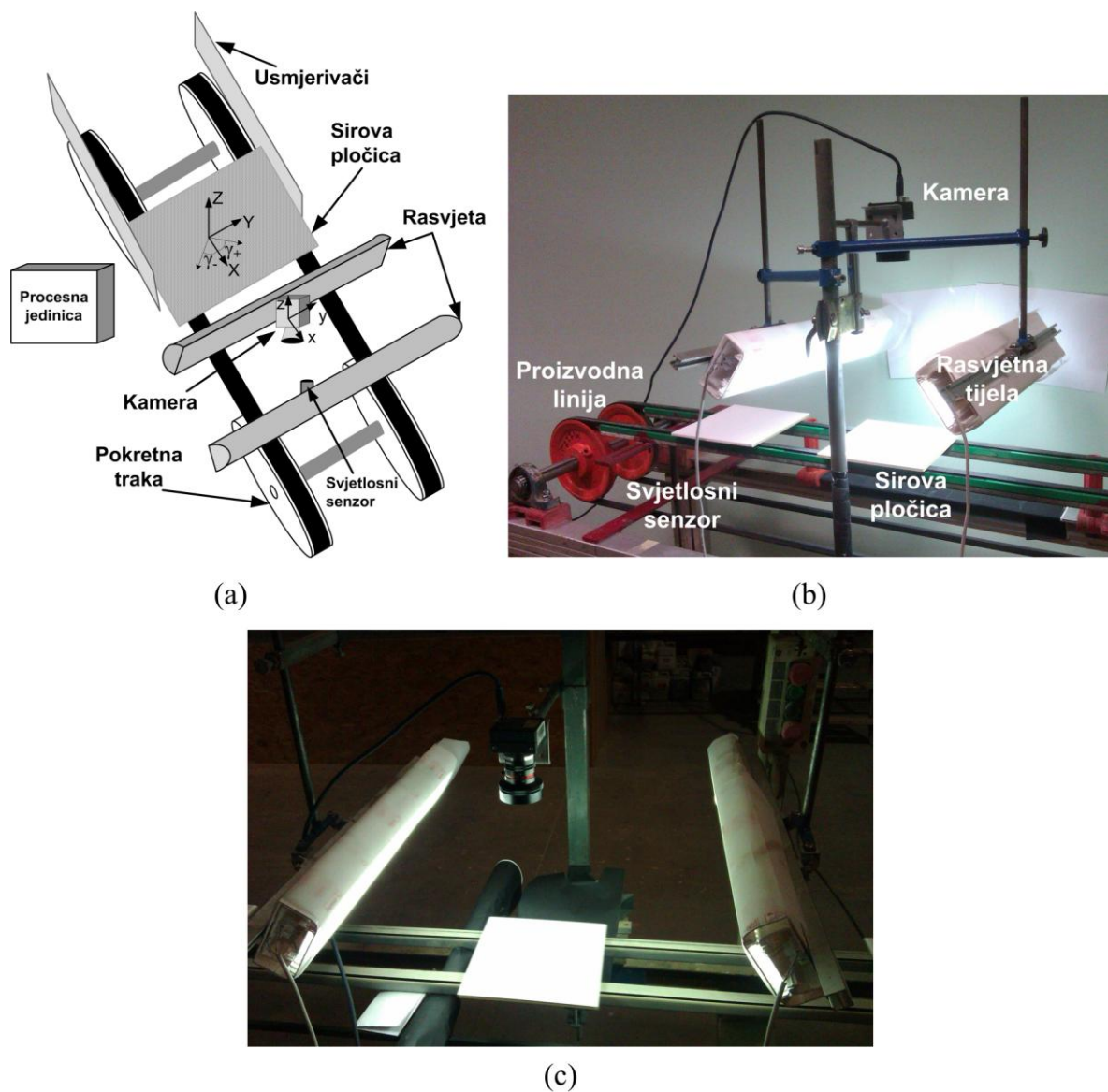
3.3. Postupak obrade sive slike sirove keramičke pločice

Obrada prikupljene sive slike keramičke pločice izvodi se na procesnoj jedinici. U ovoj disertaciji obrada sive slike je podijeljena na nekoliko faza:

- predobrada sive slike,
- binarizacija – izdvajanje sirove keramičke pločice od pozadine,

- detekcija neispravnosti na uglovima i rubovima sirove keramičke pločice,
- ujednačavanje slike površine sirove keramičke pločice i
- detekcija neispravnosti na površini sirovih keramičkih pločica.

Nakon provedenih svih faza obrade slike, sirove pločice na kojima je detektirana neispravnost se recikliraju, a sirove pločice na kojima nije detektirana neispravnost nastavljaju u fazu pečenja. Sve faze obrade slike moraju završiti u zadanom vremenskom intervalu. Maksimalna brzina proizvodnje keramičkih pločica veličine $20 \times 20 \text{ cm}$ je 100 pločica/min tj. za rad u stvarnom vremenu obrada slike mora završiti u vremenskom intervalu od 600 ms . Za veće



Slika 3.3: Prototip sustava za vizualnu inspekciju neispravnosti na sirovim keramičkim pločicama: (a) model prototipa, (b) razvijeni prototip u laboratorijskim uvjetima, (c) razvijeni prototip u industrijskim uvjetima.

keramičke pločice vremenski interval se djelomično povećava zbog ograničenja u samom procesu proizvodnje keramičkih pločica.

U ovoj disertaciji razvijene su i testirane nove metode za izdvajanje sirove keramičke pločice od pozadine, detekciju neispravnosti na uglovima i rubovima sirove keramičke pločice i ujednačavanje slike površine sirove keramičke pločice, v sliku 3.4. Također, obrađena je tematika predobrade sive slike i detekcije neispravnosti na površini sirovih keramičkih pločica.

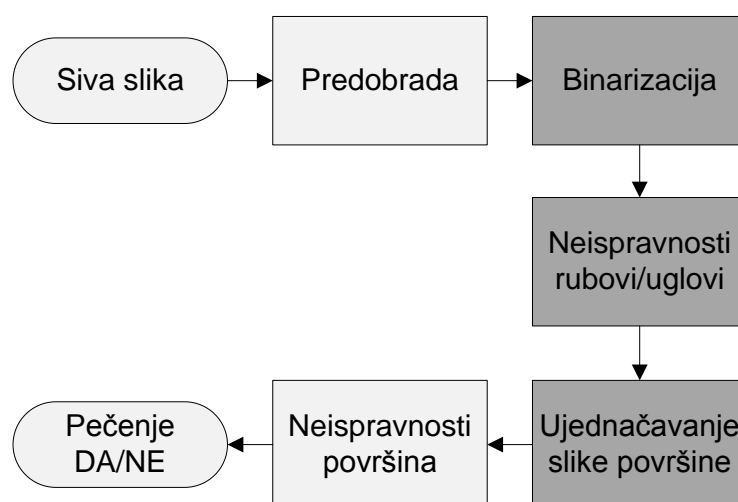
3.4. Predobrada sive slike sirove keramičke pločice

Predobrada sive slike se može sastojati od nekoliko koraka. Najčešći koraci predobrade slike su uklanjanje šuma diskretizacije, uklanjanje neravnomjerne osvjetljenosti objekta i uklanjanje geometrijske distorzije leće.

Za uklanjanje šuma diskretizacije najčešće se upotrebljavaju razni filteri ublažavanja. Neki od filtera koji se koriste u literaturi su Gaussov filter i median filter [15]. U ovoj disertaciji se pretpostavlja da je šum diskretizacije zanemariv.

U literaturi se može pronaći nekoliko metoda za uklanjanje neravnomjerne osvjetljenosti objekta [16, 17, 18]. U sustavima stvarnog vremena često se upotrijebljava tzv. *Mask* metoda zbog brzine izvedbe [17]. Pored navedene metode često se koriste metode temeljene na homomorfskom filtriranju [18]. S pravilno postavljenom LED difuznom rasvjetom učinak neravnomjerne osvjetljenosti se može dovoljno smanjiti i time ukloniti potrebu za upotrebom navedenih metoda.

Budući da je prostor između kamere i sirove pločice ograničen, upotrebljavaju se leće sa



Slika 3.4: Faze obrade sive slike (zatomnjeni dijelovi predstavljaju metode razvijene u ovom radu).

kratkom žarišnom duljinom. Ovaj tip leća često proizvodi određeni oblik geometrijske distorzije – jastučasta ili bačvasta distorzija, cf. slika 3.5. Geometrijska distorzija nastaje zbog nesavršenosti leće, tj. povećanje centra leće nije jednako vanjskim dijelovima leće. Postupak modeliranja i uklanjanja geometrijske distorzije se naziva kalibracija kamere [19]. Geometrijska distorzija leće sastoji se od radijalne i tangencijalne komponente. Radijalna distorzija nastaje zbog sferičnog oblika leće, dok tangencijalna distorzija nastaje zbog neparalelnog odnosa leće i osjetnika kamere. Često korišteni model za uklanjanje geometrijske distorzije je Brown – Conrady model [20, 21].

Za ulaznu sivu sliku $I(x_d, y_d)$ veličine $K \times M$ piksela (K označava broj redova slike, M označava broj stupaca slike) s geometrijskom distorzijom, ispravljanje radijalne komponente distorzije računa se s

$$\begin{aligned} x_r &= x_d(1 + k_1 r^2 + k_2 r^4 + k_3 r^6), \\ y_r &= y_d(1 + k_1 r^2 + k_2 r^4 + k_3 r^6), \end{aligned} \quad (3.1)$$

a tangencijalne komponente s

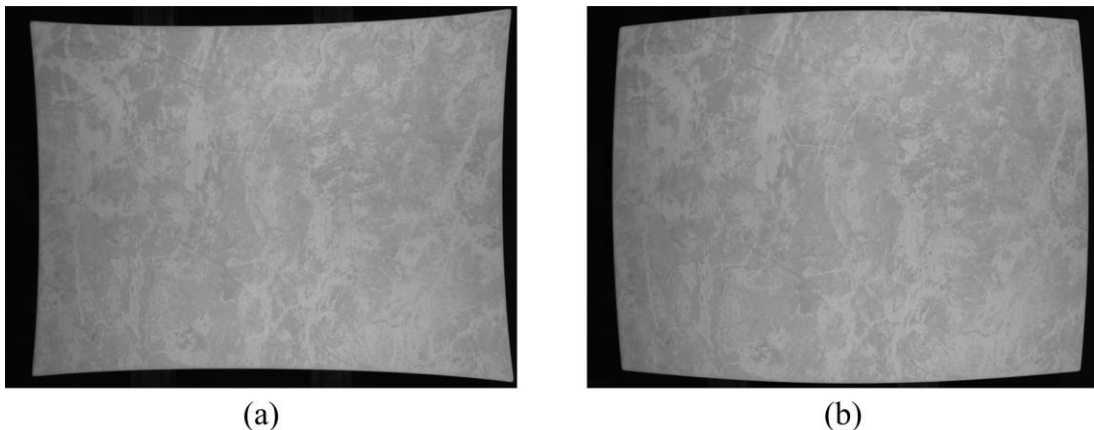
$$\begin{aligned} x_t &= [2p_1 x_d y_d + p_2 (r^2 + 2x_d^2)], \\ y_t &= [p_1 (r^2 + 2y_d^2) + 2p_2 x_d y_d], \end{aligned} \quad (3.2)$$

gdje k_1 , k_2 , k_3 , p_1 i p_2 predstavljaju parametre leće, a $r^2 = x_d^2 + y_d^2$. Ukupna distorzija jednaka je zbroju radijalne i tangencijalne komponente

$$\begin{aligned} x_{rt} &= x_r + x_t, \\ y_{rt} &= y_r + y_t. \end{aligned} \quad (3.3)$$

Ispravljene (x, y) koordinate slike sa uklonjenom geometrijskom distorzijom jednake su

$$\begin{aligned} x &= f_x x_{rt} + c_x, \\ y &= f_y y_{rt} + c_y, \end{aligned} \quad (3.4)$$



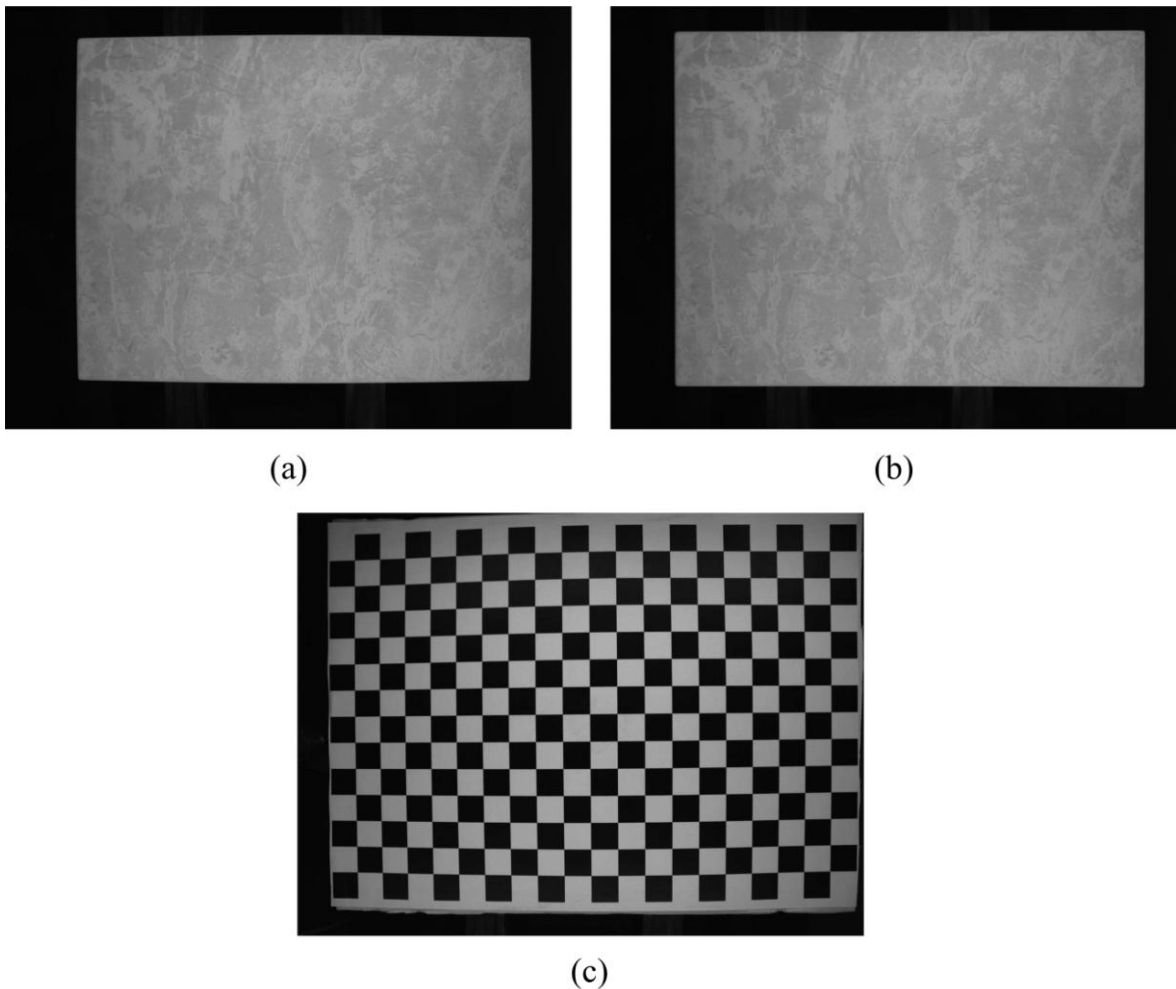
Slika 3.5: Primjer jastučaste (a) i bačvaste geometrijske distorzije (b).

Tablica 3.2: Prosječno vrijeme [ms] uklanjanja geometrijske distorzije s OpenCV bibliotekom.

| Veličina slike u MP | 1.14 | 1.53 | 2.01 | 3.03 | 4.00 |
|---|------|-------|-------|-------|-------|
| Prosječno vrijeme uklanjanja distorzije | 8.88 | 12.05 | 15.69 | 24.10 | 31.89 |

gdje f_x i f_y predstavljaju žarišne duljine za x i y os, a (c_x, c_y) predstavlja koordinate centra slike (engl. *principal point*).

Za izračun gore navedenih konstanti, metode za kalibraciju kamere upotrebljavaju slike sa poznatim uzorcima. Najčešći uzorak je uzorak šahovske ploče. S dobivenim parametrima kalibracije, distorzija se uklanja. Na slici 3.6 (a), (b) prikazan je primjer uklonjene geometrijske distorzije i primjer slike šahovskog uzorka 3.6 (c). U ovoj disertaciji upotrijebljen je program za kalibraciju kamere temeljen na OpenCV biblioteci [22]. Prosječno vrijeme (100 iteracija) potrebno za uklanjanje geometrijske distorzije s OpenCV bibliotekom



Slika 3.6: Primjer sive slike s geometrijskom bačvastom distorzijom (a), slika s uklonjenom bačvastom distorzijom, (c) slika s uzorkom šahovske ploče.

za različite veličine sivih slika (izraženo u megapikselima – *MP*) prikazano je u tablici 3.2. Vrijeme je mjereno na osobnom računalu s Windows 7 x64 operacijskim sustavom, Intel Q6600 procesorom, 4 GB RAM memorije i NVIDIA GTX 670 grafičkom karticom s 2 GB GDDR5 memorije. Najduže vrijeme potrebno za uklanjanje distorzije je manje od 32 *ms*, tj. manje od 6 % ukupno dostupnog vremena za vizualnu inspekciju sirovih pločica prema poglavlju 3.3. U nastavku disertacije $I(x, y)$, $x = 0, 1, \dots, K - 1$, $y = 0, 1, \dots, M - 1$ predstavlja sivu sliku s uklonjenom geometrijskom distorzijom, osim ako nije navedeno suprotno.

Postupka uklanjanja geometrijske distorzije je moguće izbjeći s upotrebom kvalitetnijih (skupljih) leća. U sljedećim poglavljima predstavljene su ostale faze obrade slike sirove keramičke pločice i detaljno objašnjenje razvijene metode u ovoj disertaciji.

4. NOVA METODA ZA IZDVAJANJE SIROVE KERAMIČKE PLOČICE OD POZADINE

Sljedeći korak u obradi slike sirove keramičke pločice je binarizacije sive slike tj. izdvajanje sirove keramičke pločice od pozadine. U ovom poglavlju detaljno je objašnjen postupak i podjela metoda za binarizaciju sive slike. Također, predložena je i detaljno analizirana nova metoda za izdvajanje sirove keramičke pločice od pozadine.

4.1. Izdvajanje objekata na sivoj slici (segmentacija slike)

Segmentacija je proces izdvajanja objekata na slici. Segmentacijom se slika dijeli na regije s sličnim (istim) vizualnim karakteristikama npr. boja, svjetlina, uzorak i sl. Rezultat segmentacije je nova slika s označenim objektima od kojih se slika sastoji. Postupak segmentacije se upotrebljava u različitim područjima:

- industrijska proizvodnja,
- medicina,
- promet,
- video nadzor,
- filmska industrija itd.

Metode za segmentaciju slike najčešće se dijele na dvije grupe:

1. metode koje se zasnivaju na detekciji sličnosti intenziteta piksela objekata na slici (cf. slika 4.1 (a) – (b), (e) – (f) [23]) i
2. metode koje se zasnivaju na detekciji naglih prijelaza između objekata na slici (cf. slika 4.1 (c) – (d) [24]).

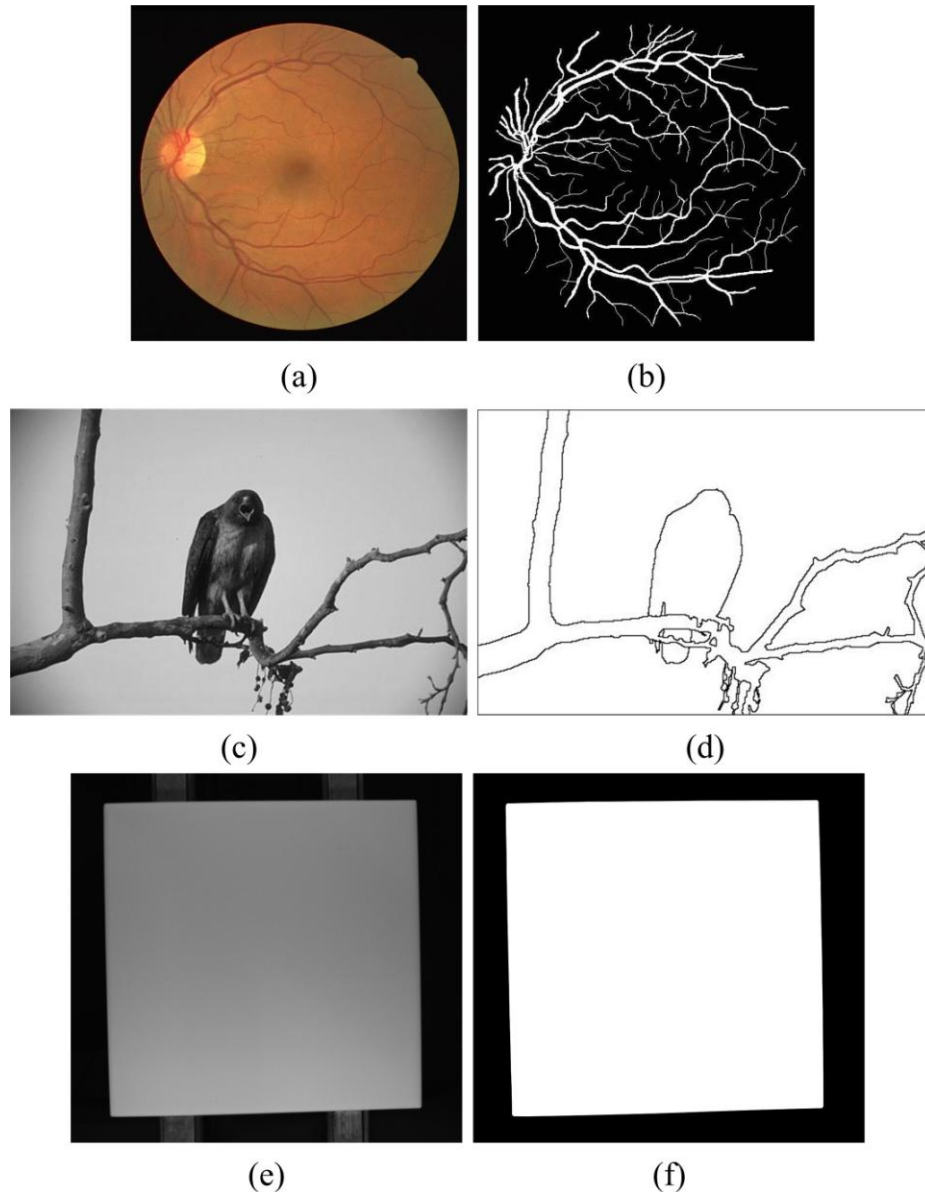
Metode temeljene na detekciji sličnosti intenziteta piksela izdvajaju objekte na temelju zadanih uvjeta sličnosti intenziteta. U ove metode spadaju algoritmi za binarizaciju slike (*engl. thresholding*), algoritmi za rast regija (*engl. region growing*) i sl.

Metode temeljene na detekciji naglih prijelaza izdvajaju objekte na slici na temelju pronađenih linija, točaka ili rubova između pojedinih objekata slike. U ove metode spadaju algoritmi za detekciju rubova temeljeni na gradijentu slike, metode za povezivanje rubova i sl. [15].

Zbog zahtjeva na rad u stvarnom vremenu u ovom radu se razmatraju metode za binarizaciju sive slike sirove keramičke pločice. U narednom poglavlju provest će se detaljna analiza metoda za binarizaciju sive slike.

4.2. Segmentacija sive slike binarizacijom

Binarizacija predstavlja najjednostavniju metodu za segmentaciju sive slike. Najčešće se upotrebljava u sustavima stvarnog vremena zbog svoje jednostavne implementacije i brzine izvođenja. Rezultat binarizacije sive slike $I(x, y)$ je nova (binarna) slika $I_B(x, y)$ istih dimenzija kao i siva slika s dvije razine intenziteta '0' i '1', cf. jednadžba (4.1).



Slika 4.1: Primjeri segmentacije slike: (a) slika mrežnice oka, (b) segmentirana slika mrežnice oka s izdvojenim krvnim žilama (bijela boja), (c) slika iz prirode, (d) segmentirana slika s označenim rubovima objekta od interesa (stablo, ptica), (e) slika sirove keramičke pločice na pokretnoj liniji, (f) segmentirana slika s izdvojenom površinom pločice.

$$I_B(x, y) = \begin{cases} 0 & \text{ako } I(x, y) \leq T \\ 1 & \text{ako } I(x, y) > T \end{cases} \quad (4.1)$$

Da li je vrijednost piksela '0' ili '1' određuje se na temelju razine praga T . Postupak binarizacije segmentira sivu sliku na objekt (*engl. foreground* $I_B(x, y) = 1$) i pozadinu (*engl. background* $I_B(x, y) = 0$). '1' označavaju dio slike od interesa npr. površina sirove pločice, dok '0' označavaju dio slike koji nije potrebno razmatrati npr. dijelovi pokretne trake. Binarna slika se prikazuje s dvije boje, crna i bijela. Crna boja predstavlja pozadinu, a bijela objekt (nerijetki su slučajevi i obrnutog prikaza).

Glavni problem u postupku binarizacije slike je određivanje razine praga T . U rijetkim slučajevima T se može odrediti na temelju informacija o uvjetima u kojima je slika prikupljena, sceni objekta u pitanju itd., ali najčešće se T određuje na temelju samog sadržaja slike. Najjednostavniji način određivanja T je da čovjek na osnovu svojih vizualnih sposobnosti mijenja vrijednost razine praga dok se ne dobiju zadovoljavajući rezultati segmentiranja. Mana ovog postupka je da se u rijetkim slučajevima ista vrijednost T može primijeniti na niz uzastopnih slika, npr. niz slika sirovih pločica prikupljenih na pokretnoj traci. Navedeno proizlazi iz činjenice da je potrebno mijenjati (korigirati) vrijednost T zbog promjena u osvjetljenosti objekta, promjena u pozadini objekta i sl. tijekom prikupljanja niza slika. Također kriterij ocjenjivanja uspješnosti segmentacije varira od čovjeka do čovjeka i razmjena informacija o kriteriju ocjenjivanja je često neadekvatna [15, 25].

Zbog gore navedenog razvijen je niz metoda koji omogućavaju automatsko određivanje razine praga na temelju sadržaja slike. Koju metodu odabrati ovisi o sadržaju slike i postavljenih vremenskih zahtjeva (rad u stvarnom vremenu).

4.2.1. Podjela metoda za binarizaciju sive slike

Metode za binarizaciju sive slike mogu se podijeliti u dvije glavne grupe:

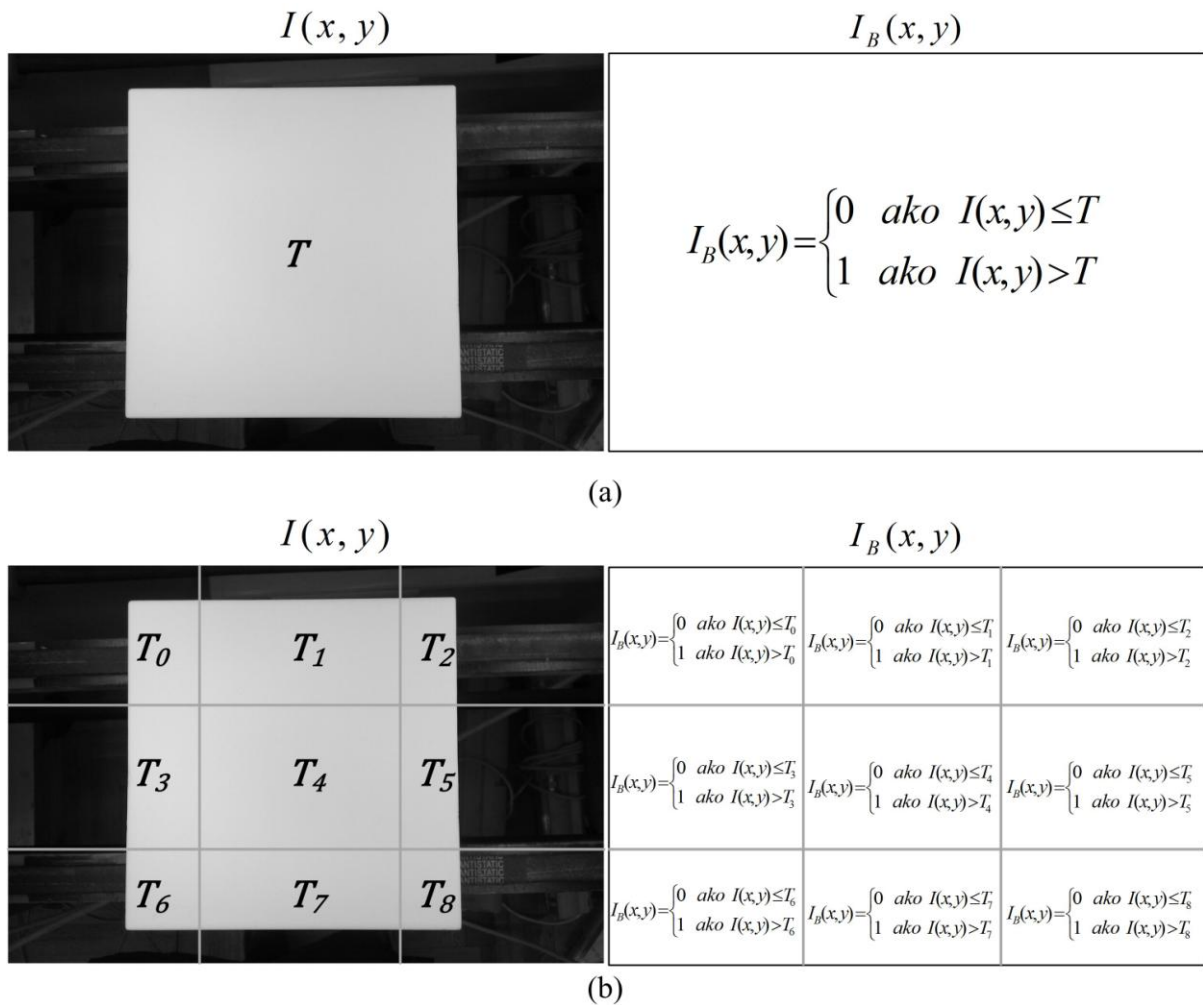
1. binarizacija na temelju globalne razine praga (globalna binarizacija),
2. binarizacija na temelju prilagodljive razine praga (prilagodljiva binarizacija).

Metode za binarizaciju slike na temelju globalne razine praga, segmentiraju sliku uspoređujući sve piksele slike s istom vrijednošću T , cf. slika 4.2 (a). Ove metode su pogodne za slike s ravnomjernom osvjetljenošću objekta, tj. slike prikupljene u kontroliranim uvjetima.

U određenim slučajevima nije moguće ispravno segmentirati sliku binarizacijom na temelju jedne vrijednosti praga (npr. zbog neravnomjerne osvjetljenosti objekta, velikih

razlika u intenzitetu piksela samog objekta i sl.). U takvim slučajevima ispravna segmentacija binarizacijom moguća je primjenom prilagodljive razine praga (*engl. adaptive thresholding*). Jedan od načina određivanja prilagodljive razine praga je diobom slike na n_{sub} manjih dijelova i određivanjem razine praga T_i , $i = 0, \dots, n_{sub} - 1$ za svaki dio slike. Vrijednost T_i za pojedini dio slike najčešće se određuje nekom od metoda binarizacije na temelju globalne razine praga. Binarna slika $I_B(x, y)$ određuje se primjenjivanjem odgovarajućeg T_i za svaki (x, y) piksel, cf. slika 4.2 (b). Problem kod ovih metoda je određivanje broja diobenih dijelova (n_{sub}), određivanje razine praga na granici diobenih dijelova (može se riješiti preklapanjem diobenih dijelova slike) i povećana kompleksnost proračuna (duže vrijeme izvršenja algoritma koje se može smanjiti upotrebom arhitektura za paralelnu obradu podataka) [26].

U narednim poglavljima detaljnije će se razmatrati metode binarizacije slike na temelju



Slika 4.2: Binarizacija slike: (a) globalno određivanje razine praga (T), (b) prilagodljivo određivanje razine praga ($T_i, i = 0, \dots, 8$).

globalne razine praga zbog postavljenih zahtjeva za rad u stvarnom vremenu i mogućnosti kontrole osvijetljenosti objekta (sirove pločice).

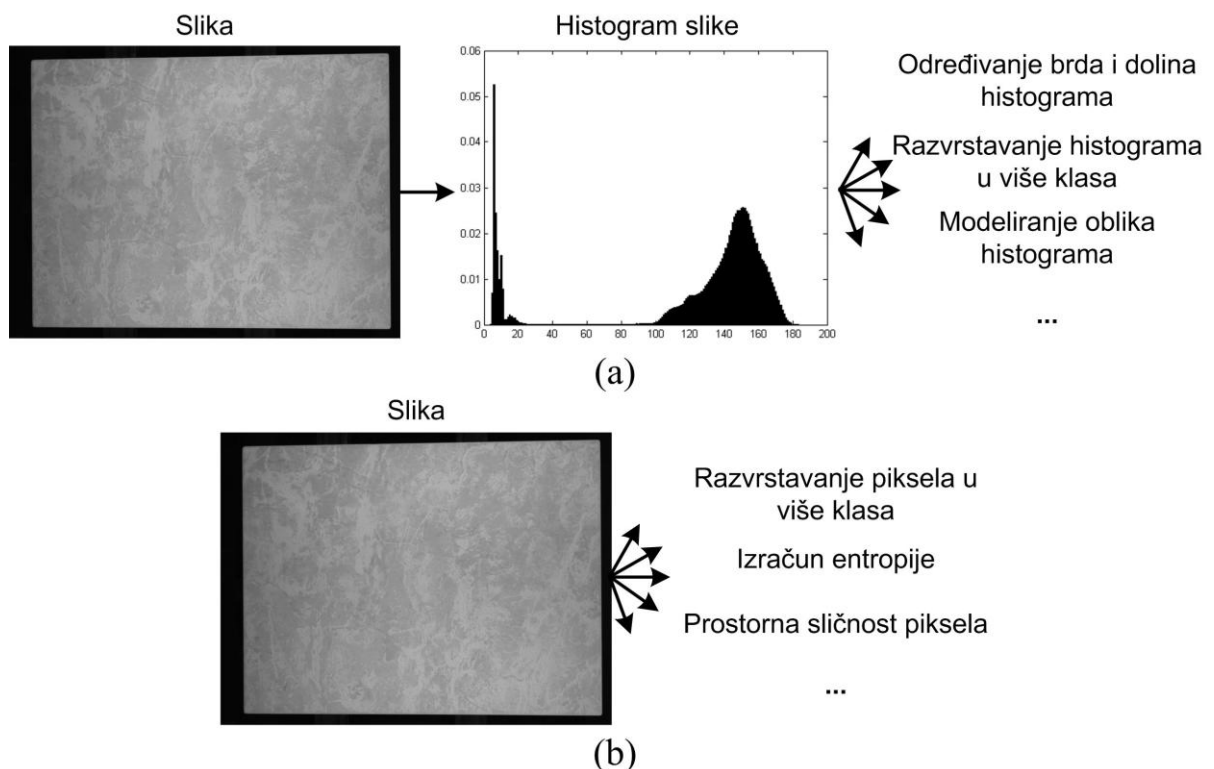
Autori u [27, 28] dijele metode za globalno određivanje praga u 5 kategorija:

1. metode zasnovane na obliku histograma slike (*engl. histogram shape – based methods*),
2. metode zasnovane na razvrstavanju intenziteta slike u dva grozda (*engl. clustering – based methods*),
3. metode zasnovane na odnosu entropija objekta i pozadine (*engl. entropy – based methods*),
4. atributne metode zasnovane na mjerama sličnosti između sive slike i binarne slike (*engl. attribute – based methods*),
5. metode zasnovane na prostornom odnosu između piksela sive slike.

Na temelju ulaznih podataka koje metoda upotrebljava za određivanje razine praga T navedena podjela se može grupirati u dvije grupe:

1. metode kojima je ulazni podatak histogram slike (cf. slika 4.3 (a)) i
2. metode kojima je ulazni podatak sama slika (cf. slika 4.3 (b)).

Metode koje upotrebljavaju histogram slike za proračun razine praga u pravilu su brže

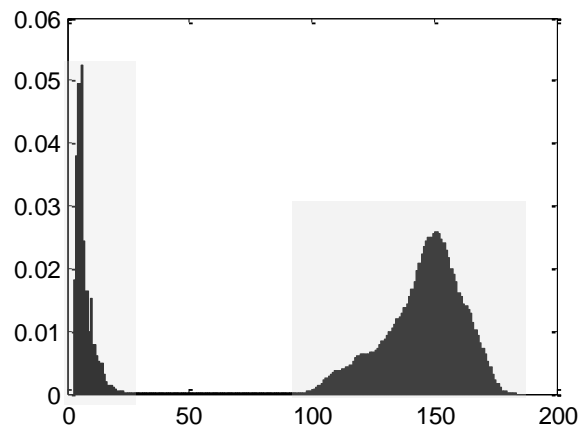


Slika 4.3: Izračun razine praga na temelju: (a) histograma slike, (b) intenziteta piksela slike.

(kraće vrijeme izvršenja) od metoda kojima je ulazni podatak sama slika. Razlog tomu može se pronaći u samoj količini podataka nad kojima se izvodi proračun. Siva slika dimenzije 1024×1024 piksela sadrži 1048576 osam bit-nih podataka, dok histogram iste slike sadrži maksimalno 256 osam bit-nih podataka, tj. 4096 puta manje podataka. Također metode binarizacije kojima je ulazni podatak histogram slike pogodnije su u slučajevima kada je histogram određenog oblika kao što je slučaj sa slikama sirovih keramičkih pločica, objašnjenje slijedi u narednim poglavljima.

4.3. Binarizacija slike sirove keramičke pločice

Slika keramičke pločice dobivena prototipom sustava na proizvodnoj liniji sastoji se od objekta (površina pločice) i pozadine (dijelovi pokretne linije i sl.). Isto vrijedi i za slike prešane i sirove keramičke pločice. Zbog navedenog, histogram slike pločice pokazuje bimodalnu karakteristiku. Histogram slike $h(i), i = 0, \dots, 255$ prema [25] definiran je kao graf koji predstavlja ovisnost broja piksela o intenzitetima piksela slike. Također se histogram može prikazati u normaliziranom obliku $h_n(i), i = 0, \dots, 255$. Normalizirani histogram predstavlja distribuciju vjerojatnosti intenziteta piksela slike i rezultat je dijeljenja $h(i)$ s ukupnim brojem piksela u slici, cf. jednadžba (4.2). Histogram slike koja prikazuje jedan objekt i pozadinu najčešće se sastoji od dvije regije (regija koja predstavlja piksele pozadine i regija koja predstavlja piksele objekta) i u literaturi se naziva bimodalni histogram, cf. slika 4.4. Na slici 4.5 prikazan je primjer sivih slika sirovih pločica i njihove normalizirane histograme. Vidljivo je da histogrami svih slika pločica pokazuju bimodalnu karakteristiku



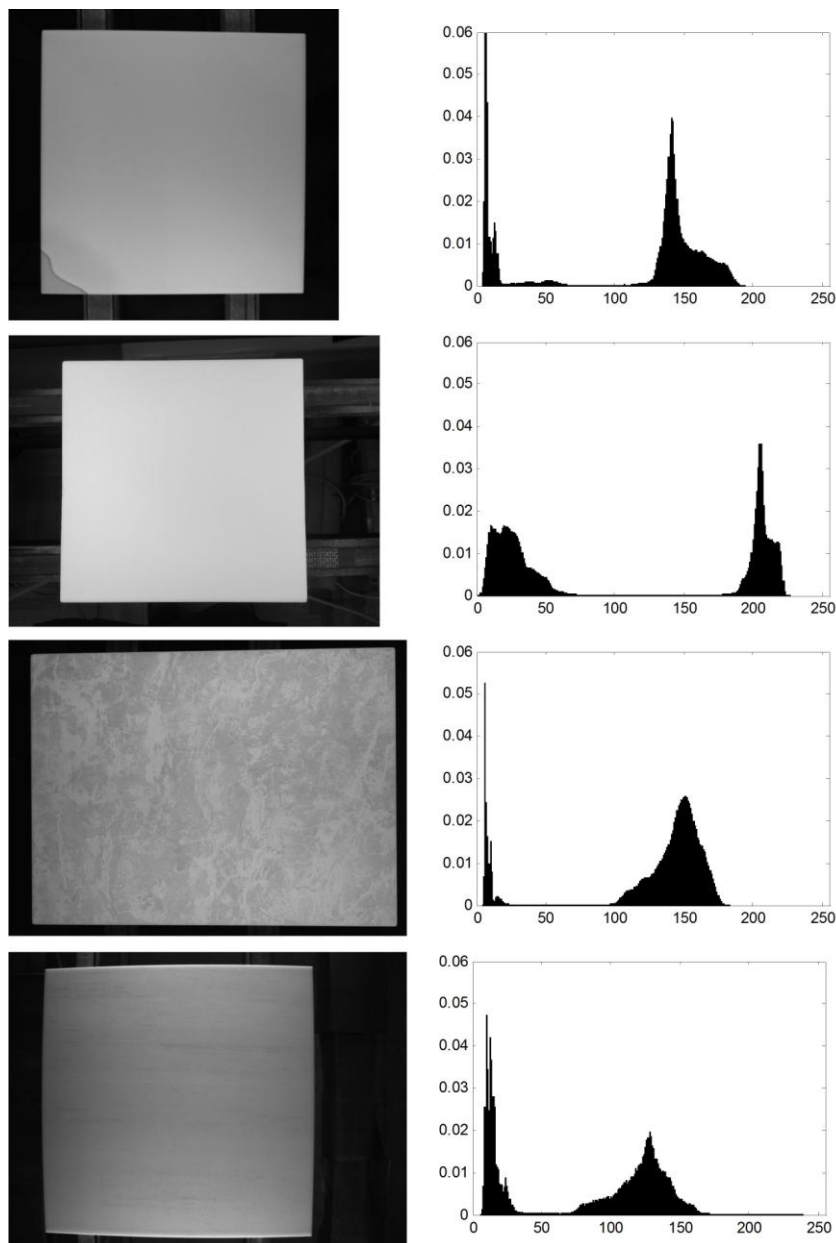
Slika 4.4: Primjer normaliziranog bimodalnog histograma. Zasjenjeni dijelovi označavaju dvije regije histograma (objekt, pozadina).

(izražene dvije regije).

$$h_n(i) = \frac{h(i)}{K \cdot M}, i = 0, \dots, 255 \quad (4.2)$$

$$\sum_{i=0}^{255} h_n(i) = 1$$

U literaturi se može pronaći niz metoda za binarizaciju slike s bimodalnim histogramom. Najčešće su korištene metode temeljene na optimizaciji određene kriterijske funkcije. U ove metode spadaju Otsu metoda i njene modifikacije [29, 30, 31], metode



Slika 4.5: Primjeri slika sirovih pločica (lijevo) i njihovih normaliziranih histograma (desno).

temeljene na odnosu entropija objekta i pozadine [32, 33, 34, 35, 36], metode temeljne na minimiziranju greške Gaussovih modela bimodalnog histograma [37, 38, 39]. U [40, 41] autori su koristili metodu za određivanje razine praga slike keramičke pločice usporedbom histograma dijela slike s objektom i histograma dijela slike s pozadinom. Mana ovih metoda je potreba za nadziranom određivanjem dijela slike pozadine i dijela slike objekta i neosjetljivost na promjene u pozadini ili osvjetljenosti objekta. Autori u [42, 43] predstavljaju metode temeljene na ublažavanju (npr. Gaussova konvolucija) histograma i traženju dolina između dva vrha histograma. Problemi kod ovih metoda nastaju pri određivanju vrhova histograma kao i načinu ublažavanja histograma.

U ovom radu implementirane su tri metode temeljene na optimizaciji kriterijske funkcije: Otsu metoda [29], metoda minimalne pogreške (*engl. minimum error threshold*) [37] i metoda minimalne razlike entropija (*engl. minimum cross-entropy threshold*) [33]. Također je razvijena i implementirana nova metoda za prilagodljivo određivanje razine praga temeljena na detekciji promjene u 1D signalu. Detaljnije objašnjenje navedenih metoda i njihovi rezultati za računalno generirane slike sirovih pločica i slike prikupljene s prototipom sustava za vizualnu inspekciju (poglavlje 3.2) slijedi u narednim poglavljima.

4.4. Otsu metoda za binarizaciju sive slike

Otsu metoda za binarizaciju sive slike je jedan od najviše referenciranih metoda za binarizaciju sive slike u literaturi. Metoda daje zadovoljavajuće rezultate segmentacije za slike s bimodalnim histogramom u kojima objekt zauzima veći dio slike. Odnosi intenziteta piksela slike također moraju zadovoljavati određene uvjete. Intenziteti svih piksela objekta i pozadine slike moraju se razlikovati za dovoljno veliku konstantu ε_1 . Treba biti zadovoljen uvjet (4.3) gdje (x, y) predstavljaju koordinate piksela objekta, a (x', y') predstavljaju koordinate piksela pozadine.

$$|I(x, y) - I(x', y')| > \varepsilon_1, \varepsilon_1 = konst. \quad (4.3)$$

Također svi intenziteti piksela objekta (pozadine) moraju se međusobno razlikovati maksimalno za dovoljno malu konstantu ε_2 (ε_3), tj. mora biti zadovoljen uvjet (4.4), cf. slika 4.6.

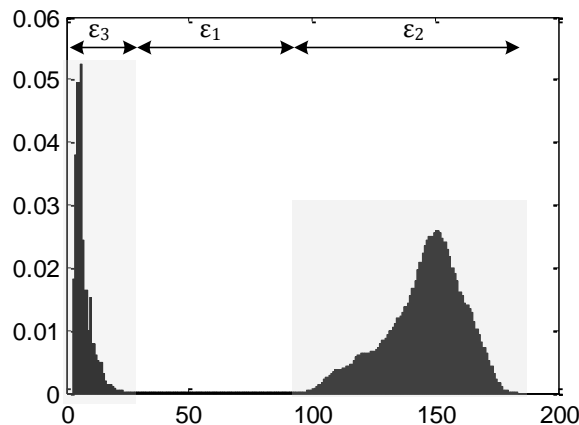
$$\begin{aligned} |I(x_1, y_1) - I(x_2, y_2)| < \varepsilon_2, \varepsilon_2 = konst., x_1 \neq x_2, y_1 \neq y_2 \\ |I(x'_1, y'_1) - I(x'_2, y'_2)| < \varepsilon_3, \varepsilon_3 = konst., x'_1 \neq x'_2, y'_1 \neq y'_2 \end{aligned} \quad (4.4)$$

Otsu metoda upotrebljava normalizirani histogram slike $h_n(i)$ za izračun razine praga. Metoda na osnovu razine intenziteta k dijeli piksele $I(x, y)$ slike s B razina intenziteta piksela i normaliziranim histogramom $h_n(i)$, $i = 0, \dots, B - 1$ u dvije klase: C_0 i C_1 (klase se odnose na pozadinu i objekt, ili obratno), prema jednadžbi (4.5).

$$\begin{aligned} C_0 &= \{I(x, y): I(x, y) \leq k\} \\ C_1 &= \{I(x, y): I(x, y) > k\} \\ k &= 0, \dots, B - 2 \end{aligned} \quad (4.5)$$

Ako $P_0(k)$ predstavlja vjerojatnost pojavljivanja piksela klase C_0 , $P_1(k)$ predstavlja vjerojatnost pojavljivanja piksela klase C_1 , $\mu_0(k)$ i $\mu_1(k)$ srednje vrijednosti pojedinih klasa, $\sigma_0(k)$ i $\sigma_1(k)$ standardne devijacije klasa C_0 i C_1 , a μ srednja vrijednost svih piksela slike, prema (4.6), (4.7) i (4.8), tada prema [29] razina praga T određena je s maksimalnom vrijednošću kriterijske funkcije $J_T(k)$ prema (4.9).

$$\begin{aligned} P_0(k) &= \sum_{i=0}^k h_n(i) \\ P_1(k) &= \sum_{i=k+1}^{B-1} h_n(i) = 1 - P_0(k) \end{aligned} \quad (4.6)$$



Slika 4.6: Primjer normaliziranog bimodalnog histograma s označenim uvjetima (4.3) i (4.4).

$$\mu_0(k) = \frac{1}{P_0(k)} \sum_{i=0}^k (i+1)h_n(i) \quad (4.7)$$

$$\mu_1(k) = \frac{1}{P_1(k)} \sum_{i=k+1}^{B-1} (i+1)h_n(i)$$

$$\mu = \sum_{i=0}^{B-1} (i+1)h_n(i) \quad (4.8)$$

Metoda binarizira sliku $I(x, y)$ s razinom intenziteta T koja dijeli histogram slike u dvije klase (C_0 i C_1) s maksimalnom među-klasnom varijancom σ_B^2 . U [44] autori su predložili jednostavniji način za izračun kriterijske funkcije prema formuli (4.10) koja se najčešće koristi u implementaciji Otsu metode. Pseudo kod za izračun razine praga Otsu metodom prikazan je u algoritmu 4.1.

$$J_T(k) = \sigma_B^2(k) = P_0(k)(\mu_0(k) - \mu)^2 + P_1(k)(\mu_1(k) - \mu)^2 \quad (4.9)$$

$$T = \max_{0 \leq k < B-1} J_T(k)$$

$$J_T(k) = P_0(k)\mu_0^2(k) + P_1(k)\mu_1^2(k) \quad (4.10)$$

4.5. Metoda minimalne pogreške

Kao i Otsu metoda, metoda minimalne pogreške [33] određuje razinu praga T na

Algoritam 4.1: Izračun razine praga Otsu metodom.

```

1. Ulaz:  $I(x, y); J_{T2}, P_0, P_1, \mu_0, \mu_1, T = 0$ 
2. Za  $I(x, y)$  izračunati  $h_n(i)$ 
3. for  $k := 0 \rightarrow B - 2$  do
4.   for  $i := 0 \rightarrow k$  do
5.      $P_0 = P_0 + h_n(i)$ 
6.      $\mu_0 = \mu_0 + (i + 1)h_n(i)$ 
7.   end for
8.   for  $i := k + 1 \rightarrow B - 1$  do
9.      $\mu_1 = \mu_1 + (i + 1)h_n(i)$ 
10.  end for
11.   $P_1 = 1 - P_0$ 
12.   $\mu_0 = \mu_0 / P_0, \mu_1 = \mu_1 / P_1$ 
13.   $J_{T1} = P_0\mu_0^2 + P_1\mu_1^2$ 
14.  if  $J_{T1} \geq J_{T2}$  then
15.     $J_{T2} = J_{T1}, T = k$ 
16.  end if
17.   $P_0, P_1, \mu_0, \mu_1 = 0$ 
18. end for

```

temelju kriterijske funkcije. Metoda pretpostavlja da se normalizirani histogram slike $h_n(i)$ može modelirati s mješavinom dvije Gaussove distribucije. Kriterijska funkcija minimizira vjerojatnost pogrešne klasifikacije piksela s obzirom na model Gaussove distribucije normaliziranog histograma.

Metoda dijeli normalizirani histogram slike na dva dijela. Oba dijela normaliziranog histograma modeliraju se s normalnom distribucijom i dobiveni modeli se uspoređuju sa normaliziranim histogramom. Metoda na temelju razine intenziteta k dijeli normalizirani histogram $h_n(i), i = 0, \dots, B - 1$ u dva dijela: C_0 i C_1 . Dva dijela $h_n(i)$ modeliraju se s normalnom distribucijom na temelju (4.12) i (4.13) gdje su $P_0(k)$, $P_1(k)$, $\mu_0(k)$, $\mu_1(k)$ definirani s (4.6) i (4.7), a $\sigma_0^2(k)$ i $\sigma_1^2(k)$ predstavljaju varijancu pojedinog dijela normaliziranog histograma i definirane su s (4.11).

Kriterijska funkcija (4.14) minimizira preklapanje modela C_0 i C_1 dijela normaliziranog histograma. Vrijednost kriterijske funkcije se smanjuje s smanjenjem razlike između modela distribucije normaliziranog histograma i normaliziranog histograma.

$$\sigma_0^2(k) = \frac{1}{P_0(k)} \sum_{i=0}^k (i + 1 - \mu_0(k))^2 h_n(i) \quad (4.11)$$

$$\sigma_1^2(k) = \frac{1}{P_1(k)} \sum_{i=k+1}^{B-1} (i + 1 - \mu_1(k))^2 h_n(i)$$

$$p_0(k) = \frac{P_0(k)}{\sqrt{2\pi}} e^{\left(-\frac{(k-\mu_0(k))^2}{2\sigma_0^2(k)}\right)} \quad (4.12)$$

$$p_1(k) = \frac{P_1(k)}{\sqrt{2\pi}} e^{\left(-\frac{(k-\mu_1(k))^2}{2\sigma_1^2(k)}\right)} \quad (4.13)$$

Metoda daje bolje rezultate binarizacije slike u odnosu na Otsu metodu kada je omjer piksela pozadine i objekta 100:1 i više. Također metoda daje bolje rezultate segmentacije u slučajevima kada slika (histogram) zadovoljava uvjete (4.15), v. sliku 4.6. Pseudo kod za određivanje razine praga metodom minimalne pogreške prikazan je u algoritmu 4.2.

$$J_T(k) = 1 + 2(P_0(k) \ln \sigma_0(k) + P_1(k) \ln \sigma_1(k)) - 2(P_0(k) \ln P_0(k) + P_1(k) \ln P_1(k)) \quad (4.14)$$

$$T = \min_{0 \leq k < B-1} J_T(k)$$

$$\begin{aligned} \varepsilon_2 &\ll \varepsilon_3 \text{ i } \varepsilon_1 \cong \varepsilon_2 \quad \text{ili} \\ \varepsilon_3 &\ll \varepsilon_2 \text{ i } \varepsilon_1 \cong \varepsilon_3 \end{aligned} \quad (4.15)$$

4.6. Metoda minimalne razlike entropija

Metoda minimalne razlike entropije temelji se na izračunu entropije. Entropija je definirana kao veličina koja opisuje stupanj nereda i jednaka je nuli u savršeno uređenim sustavima [26]. S obzirom na ovu definiciju entropija se može upotrebljavati kao mjera za određivanje vrijednosti razine praga T , npr. mjerom minimalne razlike entropija koja je objašnjena u nastavku.

Razliku entropija (*engl. cross-entropy*) definirao je Kullback u [45]. Za dvije distribucije vjerojatnosti $f(i)$ i $g(i)$, $i = 0, \dots, N - 1$ razlika entropija $H_{f,g}$ je definirana s (4.16) gdje $f(i)$ predstavlja a priori distribuciju vjerojatnosti, a $g(i)$ predstavlja a posteriori distribuciju vjerojatnosti skupa podataka.

$$H_{f,g} = \sum_{i=0}^{N-1} g(i) \ln \frac{g(i)}{f(i)} \quad (4.16)$$

U svrhu određivanja razine praga binarizacije digitalne slike razlika entropija predstavlja kriterijsku funkciju i određuje se na temelju originalne $I(x, y)$ slike i binarizirane μ -slike $I_\mu(x, y)$. Binarizirana μ -slika za razinu praga T definirana je s (4.17) gdje se $\mu_0(T)$ i $\mu_1(T)$ računaju prema (4.7).

Algoritam 4.2: Izračun razine praga metodom minimalne pogreške.

```

1. Ulaz:  $I(x, y); J_{T_2} = \max; P_0, P_1, \mu_0, \mu_1, \sigma_0^2, \sigma_1^2, T = 0$ 
2. Za  $I(x, y)$  izračunati  $h_n(i)$ 
3. for  $k := 0 \rightarrow B - 2$  do
4.   for  $i := 0 \rightarrow k$  do
5.      $P_0 = P_0 + h_n(i)$ 
6.      $\mu_0 = \mu_0 + (i + 1)h_n(i)$ 
7.   end for
8.   for  $i := k + 1 \rightarrow B - 1$  do
9.      $P_1 = P_1 + h_n(i)$ 
10.     $\mu_1 = \mu_1 + (i + 1)h_n(i)$ 
11.  end for
12.   $\mu_0 = \mu_0 / P_0, \mu_1 = \mu_1 / P_1$ 
13.  izračunati  $\sigma_0^2, \sigma_1^2$  prema (4.11)
14.  izračunati  $J_{T_1}$  prema (4.14)
15.  if  $J_{T_1} \leq J_{T_2}$  then
16.     $J_{T_2} = J_{T_1}, T = k$ 
17.  end if
18.  $P_0, P_1, \mu_0, \mu_1, \sigma_0^2, \sigma_1^2 = 0$ 
19. end for

```

$$I_{\mu}(x, y) = \begin{cases} \mu_0(T) & \text{ako } I(x, y) \leq T \\ \mu_1(T) & \text{ako } I(x, y) > T \end{cases} \quad (4.17)$$

Ako $I(x, y)$ slika predstavlja a priori distribuciju s relativnim frekvencijama $k = 0, \dots, B - 1$ (B predstavlja maksimalni intenzitet piksela $I(x, y)$ slike), a $I_{\mu}(x, y)$ predstavlja a posteriori distribuciju s relativnim frekvencijama $\mu_0(k)$ i $\mu_1(k)$ tada je razlika entropija navedenih distribucija definirana s (4.18).

$$H_{f,g} = \sum_{k=0}^{B-1} \left[\sum_{i=0}^k \mu_0(k) \ln \frac{\mu_0(k)}{i+1} + \sum_{i=k}^{B-1} \mu_1(k) \ln \frac{\mu_1(k)}{i+1} \right] \quad (4.18)$$

Ako je poznat histogram $h(i)$ slike $I(x, y)$ tada je kriterijska funkcija za izračun razine praga T metodom minimalne razlike entropija definirana s (4.19). Minimizacijom kriterijske funkcije $H_{CE}(k)$ po k određuje se razina praga s minimalnom razlikom entropija između $I(x, y)$ i $I_{\mu}(x, y)$ slike.

Metoda minimalne razlike entropija daje rezultate usporedive s Otsu metodom. Također metoda daje bolje rezultate segmentacije za slike s prisutnim šumom između dvije regije bimodalnog histograma [26]. Pseudo kod za iterativni izračun razine praga metodom minimalne razlike entropija na temelju histograma slike prikazan je u algoritmu 4.3.

Algoritam 4.3: Izračun razine praga metodom minimalne razlike entropija na temelju histograma slike.

```

1. Ulaz:  $I(x, y)$ ;  $H_{CE2} = \max$ ;  $P_0, P_1, \mu_0, \mu_1, T = 0$ 
2. Za  $I(x, y)$  izračunati  $h(i), h_n(i)$ 
3. for  $k := 0 \rightarrow B - 2$  do
4.   for  $i := 0 \rightarrow k$  do
5.      $P_0 = P_0 + h_n(i)$ 
6.      $\mu_0 = \mu_0 + (i + 1)h_n(i)$ 
7.   end for
8.   for  $i := k + 1 \rightarrow B - 1$  do
9.      $P_1 = P_1 + h_n(i)$ 
10.     $\mu_1 = \mu_1 + (i + 1)h_n(i)$ 
11.  end for
12.   $\mu_0 = \mu_0 / P_0, \mu_1 = \mu_1 / P_1$ 
13.  izračunati  $H_{CE1}$  prema (4.19)
14.  U(4.19) zanemariti  $\mu_0 = 0$  i  $\mu_1 = 0$ 
15.  if  $H_{CE1} \leq H_{CE2}$  then
16.     $H_{CE2} = H_{CE1}, T = k$ 
17.  end if
18.  $P_0, P_1, \mu_0, \mu_1 = 0$ 
19. end for

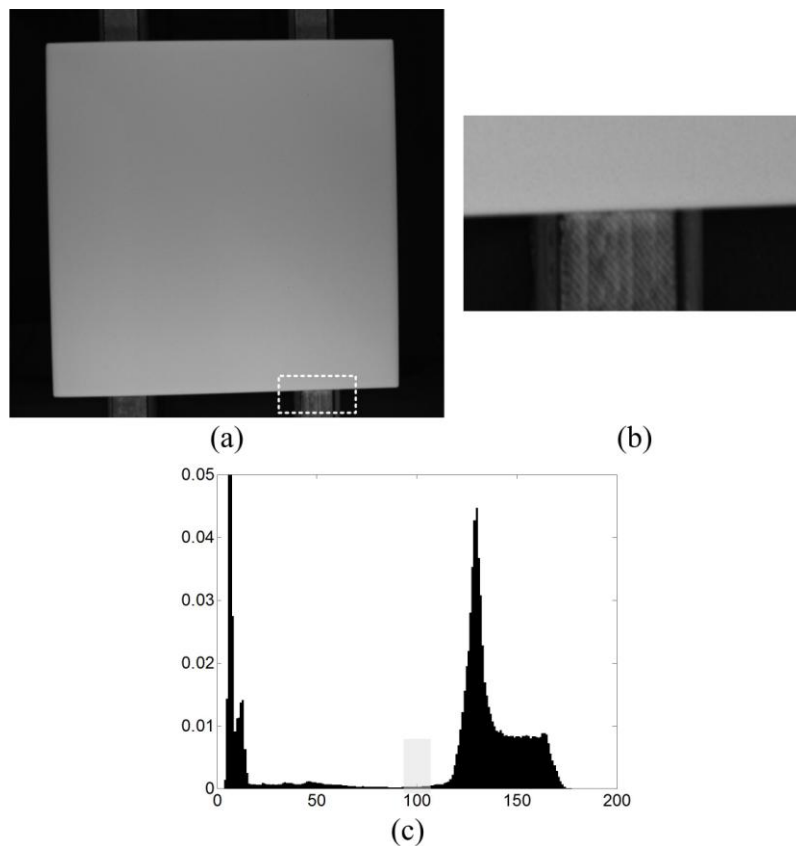
```

$$H_{CE}(k) = \sum_{i=0}^k h(i)\mu_0(k) \ln \frac{\mu_0(k)}{i+1} + \sum_{i=k}^{B-1} h(i)\mu_1(k) \ln \frac{\mu_1(k)}{i+1} \quad (4.19)$$

$$T = \min_{0 \leq k < B-1} H_{CE}(k)$$

4.7. Prilagodljiva binarizacija zasnovana na detekciji promjene amplitude u 1D signalu

Glazirana – sirova pločica je krhak međuproizvod. Tijekom prijevoza sirove pločice pokretnim linijama od procesa glaziranja do peći, krhka sirova pločica ostavlja tragove materijala na remenima od pokretne linije. Nakon nekog vremena materijal prenesen na remenje prekrije boju remena i u određenim slučajevima nestane razlika u intenzitetima piksela remenja (dio pozadine) i površine sirove pločice (objekt), v. sliku 4.7. Uvjet (4.3) tada nije zadovoljen iz čega se može zaključiti da metode za binarizaciju sive slike temeljene na globalnoj razini praga (poglavlje 4.3 do 4.6) neće dati zadovoljavajuće rezultate. Zbog preklapanja vrijednosti intenziteta piksela objekta i pozadine, primjenom metoda temeljenih na globalnom određivanju razine praga, dio piksela pozadine biti će pogrešno klasificiran kao objekt ili dio piksela objekta biti će pogrešno klasificirano kao pozadina. Do ovakvih pogrešaka klasifikacije piksela dolazi i u slučajevima neravnomjerne osvjetljenosti objekta i



Slika 4.7: Primjer prijenosa materijala s pločice na dijelove pokretne trake: (a) siva slika sirove pločice, (b) povećani dio slike (a) (crtkani bijeli pravokutnik) s vidljivim tragovima materijala na remenu pokretne trake, (c) normalizirani histogram slike (a) s zasjenjenim dijelom piksela koji se preklapaju.

u slučajevima teksturirane površine sirove pločice.

Rješenje opisanog problema se može pronaći u primjeni metoda binarizacije temeljenih na prilagodljivoj razini praga. Kao što je navedeno u poglavlju 4.2.1 problemi kod metoda binarizacije temeljenih na prilagodljivoj razini praga nastaju kod određivanja broja diobenih dijelova slike, određivanju razine praga na granicama diobenih dijelova slike i povećana kompleksnost proračuna tj. nemogućnost upotrebe metoda za rad u stvarnom vremenu. U narednim poglavljima biti će detaljno objašnjena nova metoda za prilagodljivu binarizaciju. Metoda se temelji na statističkoj metodi za detekciju promjene amplitude u 1D signalu. Pokazat će se da metoda uspješno segmentira sliku sirove pločice te rješava sva tri navedena problema prilagodljivog određivanja razine praga binarizacije.

4.7.1. Detekcija promjene amplitude 1D signala

Detekcija promjene 1D signala (*engl. Signal Change Detection*), u nastavku teksta SCD, je statistička metoda koja detektira W promjena u 1D signalu \mathbf{s} , tj. $s(i)$, $i = 0, 1, \dots, N - 1$. Izlaz SCD metode je skup indeksa $n_i \in \{n_1, n_2, \dots, n_W\}$ na temelju kojih se 1D signal \mathbf{s} može aproksimirati s skupom od $W + 1$ srednjih vrijednosti amplituda $A_n \in \{A_1, A_2, \dots, A_{W+1}\}$. Metoda je primjenjiva na signale s poznatom i nepoznatom vrijednosti amplitude signala A_n i s poznatim i nepoznatim indeksima n_i [46].

U ovom radu razmatra se detekcija jedne promjene signala \mathbf{s} ($W = 1$) na nepoznatom indeksu n i s nepoznatom promjenom signala $\Delta A = A_2 - A_1$. SCD za $W = 1$ može se smatrati optimizacijskim problemom pronalaženja indeksa $n \in \{0, 1, \dots, N - 2\}$ za koji se signal \mathbf{s} s najvećom vjerojatnosti može aproksimirati signalom $\tilde{\mathbf{s}}$. Aproksimirani signal $\tilde{\mathbf{s}}$, definiran s (4.20), sastoji se od dvije vrijednosti amplituda $A_1(n)$ i $A_2(n)$. Vrijednosti amplituda računaju se na temelju srednjih vrijednosti uzoraka signala \mathbf{s} prije i poslije indeksa n , cf. jednadžba (4.21).

$$\tilde{\mathbf{s}} = \begin{cases} A_1(n), & i = 0, 1, \dots, n \\ A_2(n), & i = n + 1, n + 2, \dots, N - 1 \end{cases} \quad (4.20)$$

Ako je signalu \mathbf{s} dodan Gaussov bijeli šum s varijancom σ^2 tada se funkcija gustoće vjerojatnosti signala \mathbf{s} , $p(\mathbf{s}; n)$, može definirati s (4.22). SCD metoda pronalazi indeks n_t za koji $p(\mathbf{s}; n)$ postiže maksimalnu vrijednost. Budući da se u (4.22) jedino mijenja eksponent, problem pronalaska indeksa n_t svodi se na pronalaženje indeksa n za koji kriterijska funkcija (4.23) postiže minimalnu vrijednost.

$$A_1(n) = \frac{1}{n+1} \sum_{i=0}^n s(i) \quad (4.21)$$

$$A_2(n) = \frac{1}{N-n-1} \sum_{i=n+1}^{N-1} s(i)$$

$$p(\mathbf{s}; n) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp \left[-\frac{1}{2\sigma^2} \left(\sum_{i=0}^n (s(i) - A_1(n))^2 + \sum_{i=n+1}^{N-1} (s(i) - A_2(n))^2 \right) \right] \quad (4.22)$$

$$J_{SCD}(n) = \sum_{i=0}^n (s(i) - A_1(n))^2 + \sum_{i=n+1}^{N-1} (s(i) - A_2(n))^2 \quad (4.23)$$

$$n_t = \min_{0 \leq n \leq N-2} J_{SCD}(n)$$

Da bi se SCD metoda mogla upotrijebiti za detekciju promjene amplitude signala potrebno je pokazati da metoda pronalazi ispravan indeks n_t za signale s određenom količinom dodanog šuma i određenim brojem uzoraka N . U tu svrhu definirana je mjera devijacije indeksa n_t SCD metode s (4.24). Za signal $s(i)$, $i = 0, 1, \dots, N-1$ s poznatim indeksom promjene signala n_{Gt} , devijacija indeksa SCD metode \hat{n}_t predstavlja relativnu razliku između detektiranog indeksa n_t i stvarnog (*engl. ground truth*) indeksa n_{Gt} . Potrebno je pokazati da $\hat{n}_t \rightarrow 0$ za signal s određenim brojem uzoraka N i određenom količinom dodanog šuma standardne devijacije σ .

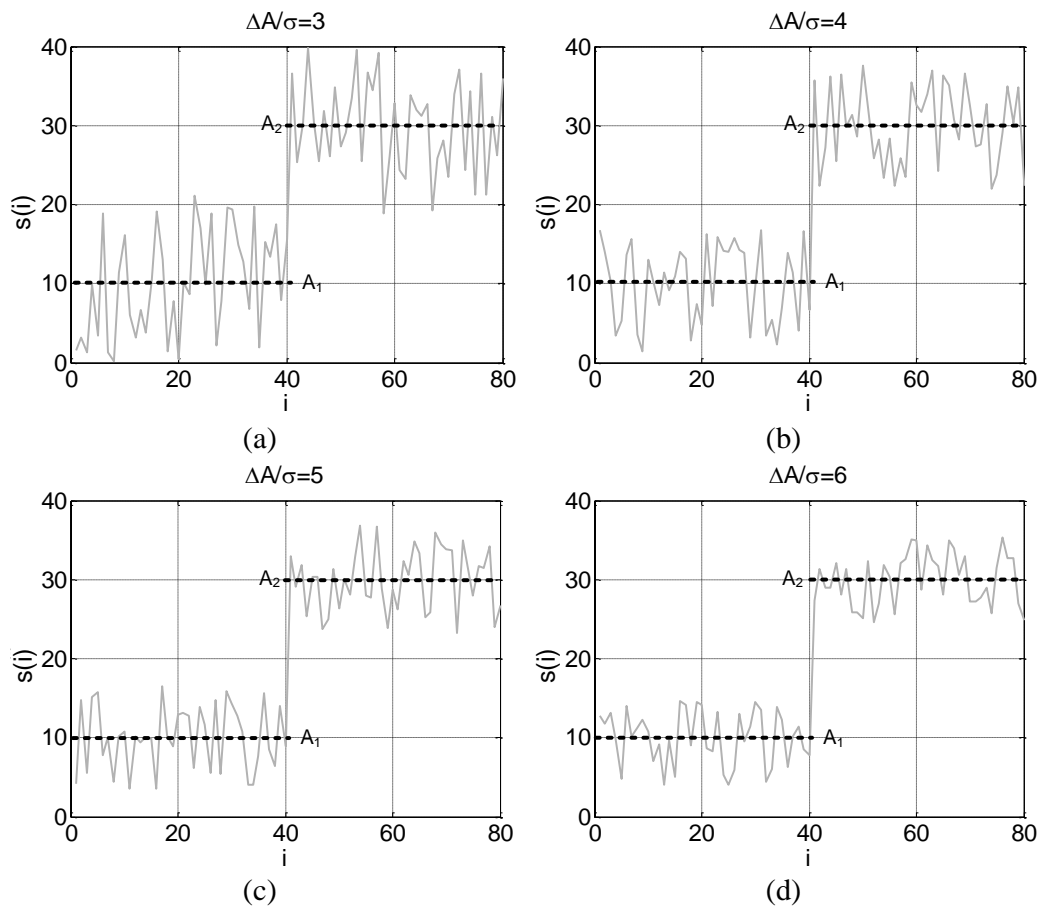
$$\hat{n}_t = \left(\frac{n_t}{n_{Gt}} - 1 \right) \times 100\% \quad (4.24)$$

Da bi se navedeno dokazalo provedena su dva eksperimenta. Za testiranje su upotrijebljeni stohastički signali \mathbf{s} čiji primjeri su dani na slici 4.8. Signali su prikazani s sivom linijom, a njihove aproksimacije $\tilde{\mathbf{s}}$ su označene s crtkanom crnom linijom. Primjeri signala se sastoje od $N = 80$ uzoraka s konstantnom promjenom amplitude $\Delta A = 20$ na indeksu $n_{Gt} = N/2 = 40$. U signale je dodan šum koji je izražen s omjerom promjene amplitude i standardne devijacije šuma $\Delta A/\sigma$. Što je $\Delta A/\sigma$ manji to je šum u signalu izraženiji i obratno. Odabrani su primjeri signala s $\Delta A/\sigma \in \{3, 4, 5, 6\}$ kako je i prikazano na slici 4.8 od (a) do (d).

Prvi eksperiment se sastoji od stohastičkih signala \mathbf{s} s promjenjivim brojem uzoraka $N \in [10, 100]$ i $\Delta A = 20$ s $n_{Gt} = N/2$. Za svaki N kreirano je 3000 slučajeva signala \mathbf{s} s $\Delta A/\sigma \in \{3, 4, 5, 6\}$. Za svaki slučaj signala \mathbf{s} izračunata je devijacija indeksa SCD metode

prema (4.24). Slika 4.9 prikazuje variranje vrijednosti \hat{n}_t za signal \mathbf{s} s $\Delta A/\sigma \in \{3, 4, 5, 6\}$. Vrijednost \hat{n}_t se mijenja s promjenom broja uzoraka N budući da je $n_{Gt} = N/2$, a $n_t \in [0, N - 2]$. Sa slike 4.9 (a) i (b) može se zaključiti da za $\Delta A/\sigma < 5$ vrijednost \hat{n}_t se smanjuje s povećanjem broja uzoraka N . Za $\Delta A/\sigma = 5$ devijacija indeksa $\hat{n}_t = 0\%$ i prestaje varirati za $N > 60$, a za $\Delta A/\sigma > 5$ devijacija indeksa $\hat{n}_t = 0\%$ i neovisna je o broju uzoraka signala N , v. sliku 4.9 (c) i (d).

Drugi eksperiment se sastoji od stohastičkih signala \mathbf{s} s promjenjivim $\Delta A/\sigma \in [1, 6]$, konstantnim brojem uzoraka $N \in \{40, 60\}$, $\Delta A = 20$ i $n_{Gt} = N/2$. Za svaki $\Delta A/\sigma$ kreirano je 3000 slučajeva signala \mathbf{s} . Za svaki signal \mathbf{s} izračunata je devijacija indeksa SCD metode prema (4.24) i prikazana na slici 4.10. Sa slike 4.10 i iz tablica 4.1 i 4.2 može se zaključiti da se vrijednost \hat{n}_t smanjuje s povećanjem $\Delta A/\sigma$ i da za $\Delta A/\sigma > 5$ u oba slučaja (slika 4.10 (a) i (b)) devijacija indeksa $\hat{n}_t = 0\%$ i prestaje varirati.



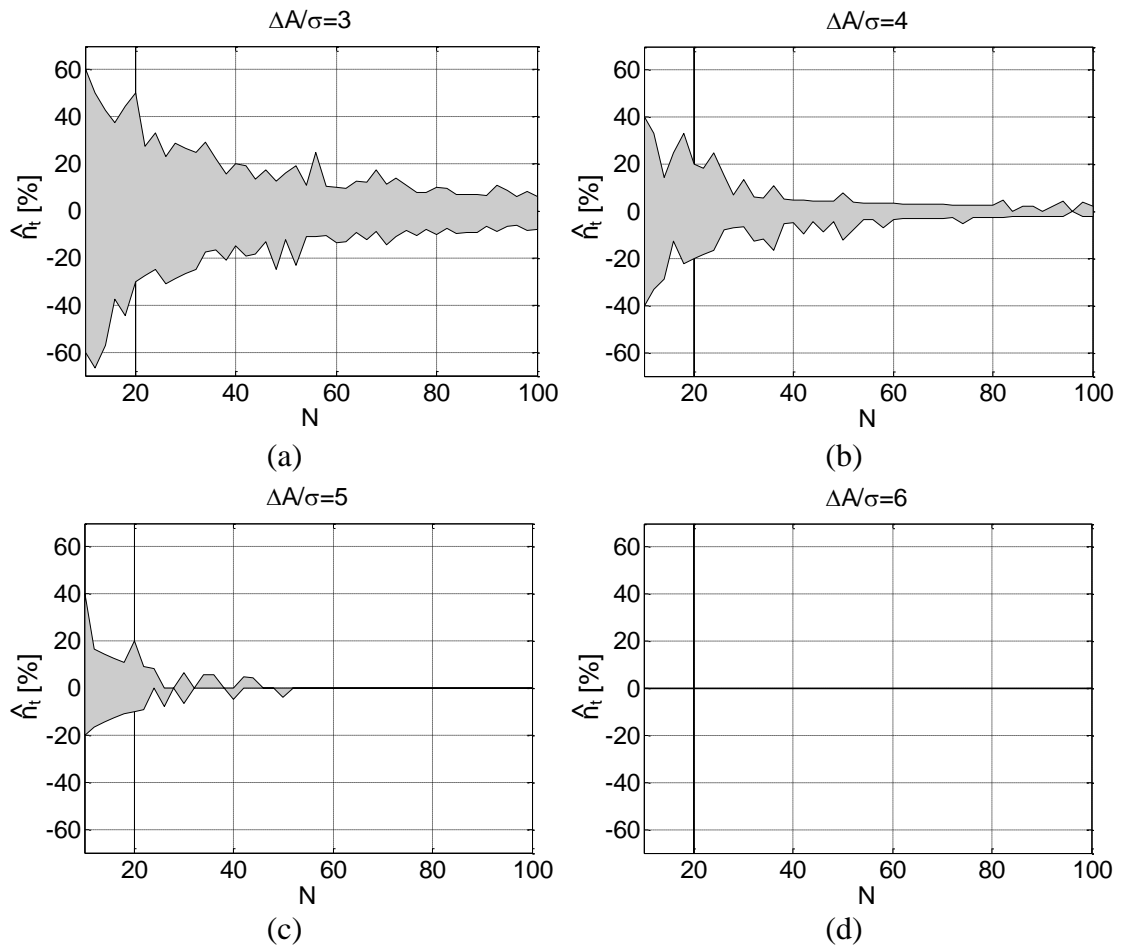
Slika 4.8: Primjeri četiri stohastička signala \mathbf{s} (siva linija) i njihovih aproksimacija $\tilde{\mathbf{s}}$ (crtkana crna linija) s $N = 80$, $\Delta A = 20$, (a) $\Delta A/\sigma = 3$, (b) $\Delta A/\sigma = 4$, (c) $\Delta A/\sigma = 5$ i (d) $\Delta A/\sigma = 6$.

Tablica 4.1: Minimalna (*min*) i maksimalna (*max*) vrijednost devijacije indeksa SCD metode \hat{n}_t za 3000 slučajeva signala \mathbf{s} s brojem uzoraka $N = 40$, $\Delta A = 20$, $n_{Gt} = N/2 = 20$ i $\Delta A/\sigma \in [1.0, 5.0]$ u koracima od 0.5.

| | $\Delta A/\sigma$ | | | | | | | | |
|------------------|-------------------|-------|-------|-------|-------|-------|-------|------|------|
| | 1.0 | 1.5 | 2.0 | 2.5 | 3.0 | 3.5 | 4.0 | 4.5 | 5.0 |
| $\min \hat{n}_t$ | -90.0 | -75.0 | -50.0 | -40.0 | -20.0 | -15.0 | -10.0 | -5.0 | -5.0 |
| $\max \hat{n}_t$ | 90.0 | 75.0 | 60.0 | 40.0 | 25.0 | 15.0 | 10.0 | 5.0 | 5.0 |

Iz provedenih eksperimenata može se zaključiti da će devijacija indeksa SCD metode \hat{n}_t biti 0 % za signal \mathbf{s} koji zadovoljava dva uvjeta $\Delta A/\sigma = [5, 6)$ i $N \geq 60$ ili uvjet $\Delta A/\sigma \geq 6$.

Pseudo kod SCD metode za signal $s(i)$, $i = 0, 1, \dots, N - 1$ prikazan je u algoritmu 4.4.



Slika 4.9: Devijacija indeksa SCD metode \hat{n}_t za $N \in [10, 100]$, $n_{Gt} = N/2$, $\Delta A = 20$, (a) $\Delta A/\sigma = 3$, (b) $\Delta A/\sigma = 4$, (c) $\Delta A/\sigma = 5$ i (d) $\Delta A/\sigma = 6$.

Tablica 4.2: Minimalna (*min*) i maksimalna (*max*) vrijednost devijacije indeksa SCD metode \hat{n}_t za 3000 slučajeva signala \mathbf{s} s brojem uzoraka $N = 60$, $\Delta A = 20$, $n_{Gt} = N/2 = 30$ i $\Delta A/\sigma \in [1.0, 5.0]$ u koracima od 0.5.

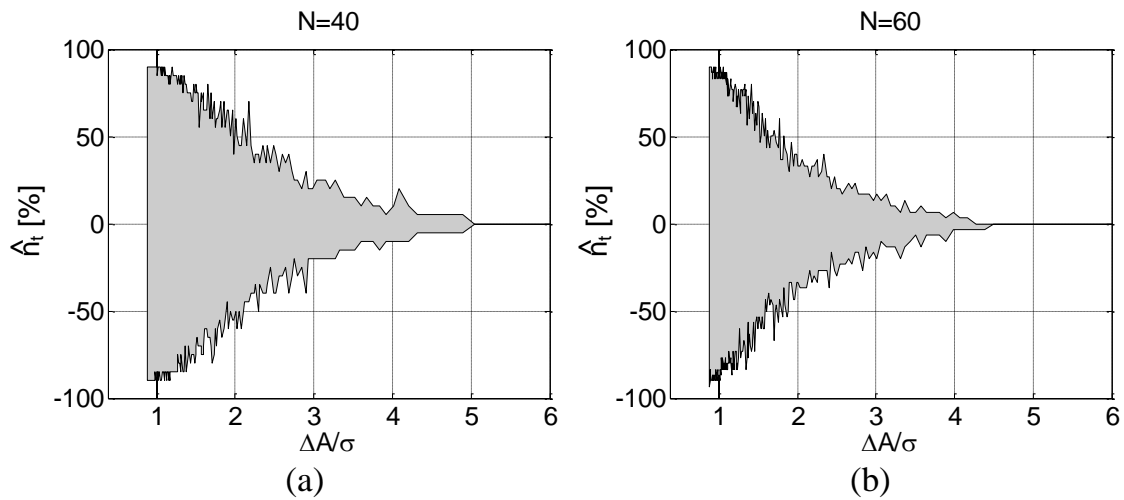
| | $\Delta A/\sigma$ | | | | | | | | |
|------------------|-------------------|-------|-------|-------|-------|-------|------|------|-----|
| | 1.0 | 1.5 | 2.0 | 2.5 | 3.0 | 3.5 | 4.0 | 4.5 | 5.0 |
| $\min \hat{n}_t$ | -86.6 | -60.0 | -33.0 | -15.0 | -30.0 | -20.0 | -3.3 | -3.3 | 0.0 |
| $\max \hat{n}_t$ | 86.6 | 70.0 | 33.0 | 15.0 | 20.0 | 16.6 | 3.3 | 0.0 | 0.0 |

Algoritam 4.4: Izračun n_t SCD metodom.

```

1. Ulaz:  $N, s(i); A_0, A_1, V_0, V_1, J_{SCD2}, n_t = 0$ 
2. for  $n := 0 \rightarrow N - 2$  do
3.   for  $i := 0 \rightarrow n$  do
4.      $A_0 = A_0 + s(i)$ 
5.   end for
6.   for  $i := n + 1 \rightarrow N - 1$  do
7.      $A_1 = A_1 + s(i)$ 
8.   end for
9.    $A_0 = A_0/(n + 1), A_1 = A_1/(N - n - 1)$ 
10.  for  $i := 0 \rightarrow n$  do
11.     $V_0 = V_0 + (s(i) - A_0)^2$ 
12.  end for
13.  for  $i := n + 1 \rightarrow N - 1$  do
14.     $V_1 = V_1 + (s(i) - A_1)^2$ 
15.  end for
16.   $J_{SCD1} = V_0 + V_1$ 
17.  if  $J_{SCD1} \leq J_{SCD2}$  then
18.     $J_{SCD2} = J_{SCD1}, n_t = n$ 
19.  end if
20.   $A_0, A_1, V_0, V_1 = 0$ 
21. end for

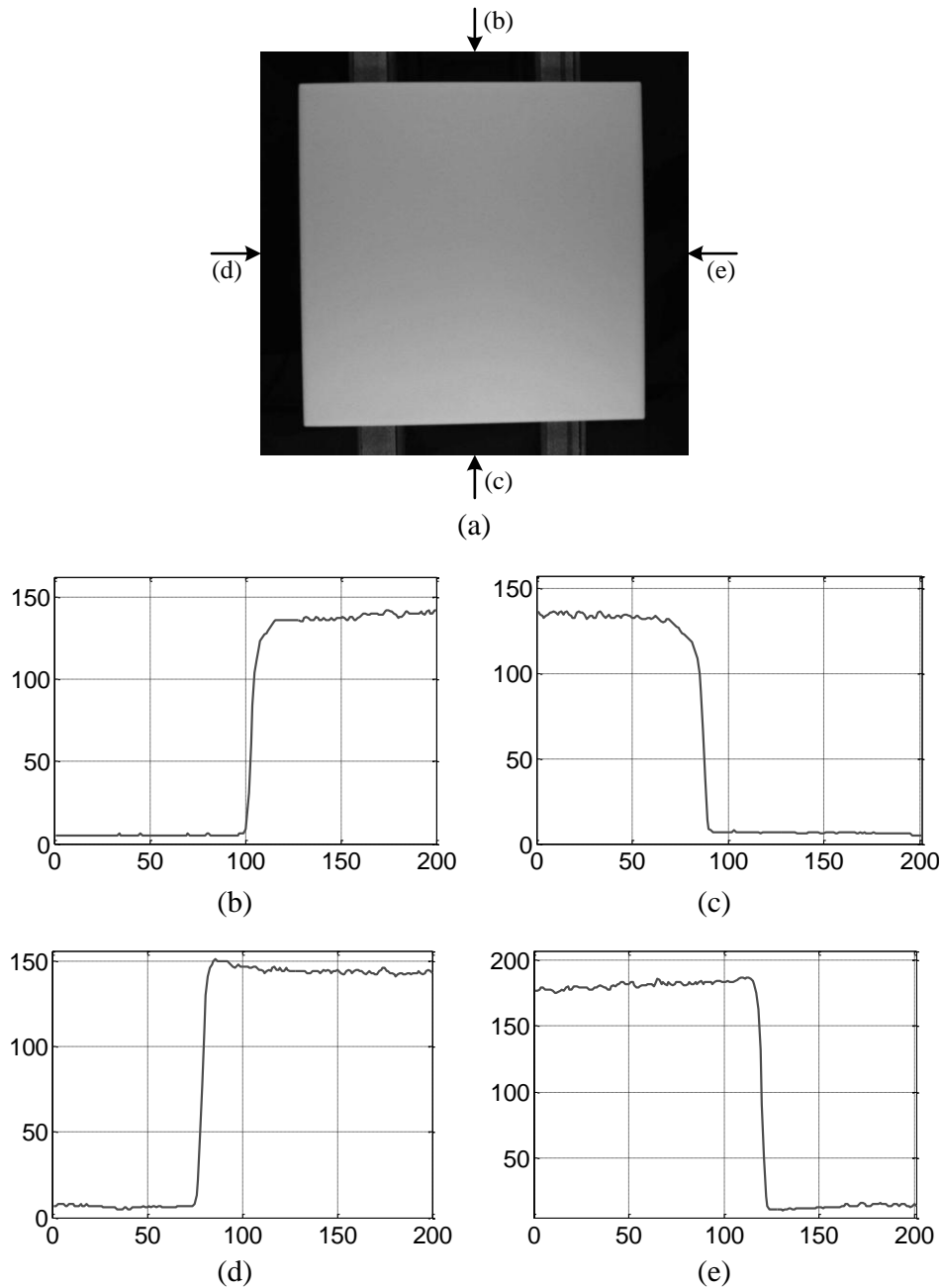
```



Slika 4.10: Devijacija indeksa SCD metode \hat{n}_t s promjenom $\Delta A/\sigma \in [1, 6]$, $n_{Gt} = N/2$, $\Delta A = 20$, (a) $N = 40$ i (b) $N = 60$.

4.7.2. Binarizacija sive slike sirove pločice SCD metodom

Slika sirove pločice, prikupljene s prototipom sustava za detekciju neispravnosti na sirovim pločicama (poglavlje 3.2), sastoji se od pozadine koja je pretežno tamnih nijansi i objekta koji je pretežno svjetlijih nijansi. Ako bi promatrali pojedine redove i stupce slike sirove pločice kao zasebne signale s tada bi svaki signal s pokazivao amplitudnu promjenu na mjestu prijelaza piksela pozadine na piksele objekta. Slika 4.11 prikazuje primjere izdvojenih



Slika 4.11: Izdvojene vrijednosti intenziteta 200 piksela slike (a) za: (b) gornji i (c) donji dio naznačenog stupca, (d) lijevi i (e) desni dio naznačenog retka.

signala \mathbf{s} s pojedinih redova i stupaca slike sirove pločice 4.11 (a). Signali se sastoje od 200 piksela redova i stupaca označenih strelicama na slici 4.11 (a). Iz slika 4.11 (b) do (e) vidljiva je izrazita promjena u amplitudi signala koja označava početak/kraj objekta (sirove pločice) u naznačenim redovima i stupcima slike. SCD metoda, koja je objašnjena u prethodnom poglavlju, pogodna je za detekciju indeksa promjene amplitude signala, a time i detekciju početka/kraja sirove pločice na slici.

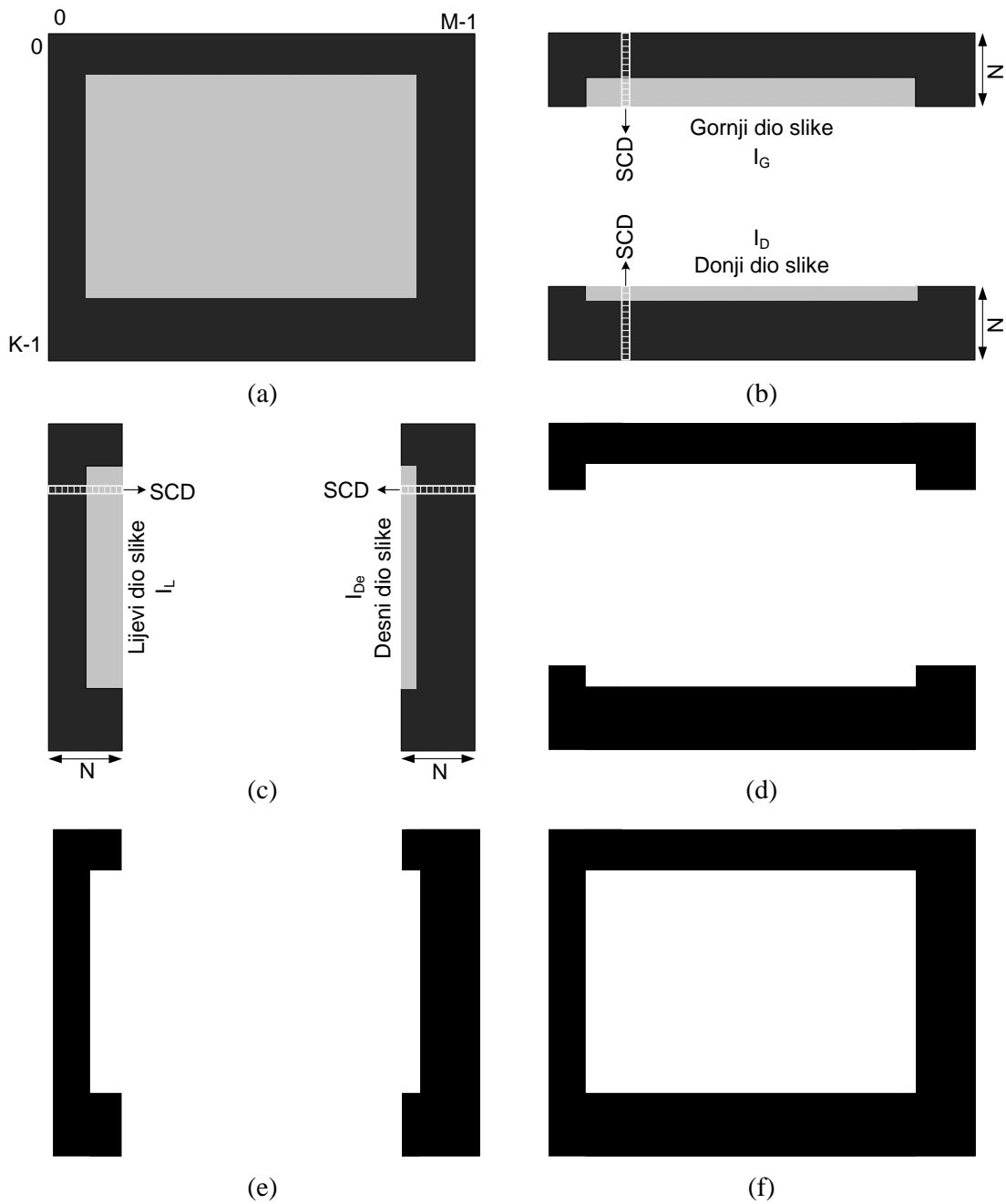
Na temelju SCD metode razvijena je nova PBSCD (**P**rilagodljiva **B**inarizacija **SCD** metodom) metoda za binarizaciju sive slike sirove pločice. Ulaz PBSCD metode je $K \times M$ $I(x, y)$ siva slika, a izlaz je $K \times M$ binarna slika $I_B(x, y)$ gdje intenzitet '1' označava sirovu pločicu, a intenzitet '0' označava pozadinu.

PBSCD metoda ulaznu $I(x, y)$ sliku dijeli na četiri dijela I_G , I_D , I_L i I_{De} prema jednadžbi (4.25). Prva dva dijela predstavljaju gornji (slika 4.12 (b) I_G) i donji (slika 4.12 (b) I_D) dio $I(x, y)$ slike veličine $N \times M$ piksela. Druga dva dijela slike predstavljaju lijevi (slika 4.12 (c) I_L) i desni (slika 4.12 (c) I_{De}) dio $I(x, y)$ slike veličine $K \times N$ piksela.

$$\begin{aligned} I_G &= I(x_G, y), & x_G &= 0, 1, \dots, N - 1 \\ I_D &= I(x_B, y), & x_B &= K - N - 1, K - N - 2, \dots, K - 1 \\ I_L &= I(x, y_L), & y_L &= 0, 1, \dots, N - 1 \\ I_{De} &= I(x, y_{De}), & y_{De} &= M - N - 1, M - N - 2, \dots, M - 1 \end{aligned} \quad (4.25)$$

PBSCD metoda primjenjuje SCD metodu na svakom $y = 0, 1, \dots, M - 1$ stupcu I_G i I_D slike i na svakom $x = 0, 1, \dots, K - 1$ retku I_L i I_{De} slike. Svaki stupac I_G i I_D slike i svaki redak I_L i I_{De} slike predstavlja jedan 1D signal \mathbf{s} . SCD metoda se primjenjuje na svaki signal \mathbf{s} zasebno, tj. slike I_G , I_D , I_L i I_{De} razmatrane su neovisno jedna od druge. Indeksi signala \mathbf{s} za donji i desni dio slike su obrnutog redoslijeda, tj. na nultom indeksu signala je intenzitet piksela pozadine, a na $N - 1$ indeksu signala je intenzitet piksela sirove pločice kako je i naznačeno na slikama 4.12 (b) i (c).

Prije primjene SCD metode svi pikseli izlazne $K \times M$ binarne slike $I_B(x, y)$ su postavljeni u '1' (pikseli objekta). Za svaki $y = 0, 1, \dots, M - 1$ stupac I_G slike SCD metoda detektira indekse $n_{tG}(y)$, a za svaki $y = 0, 1, \dots, M - 1$ stupac I_D slike SCD metoda detektira indekse $n_{tD}(y)$. Na temelju $n_{tG}(y)$ i $n_{tD}(y)$ pikseli izlazne binarne slike $I_B(x, y)$ postavljaju se u '0' prema (4.26), v. sliku 4.12 (d). Za svaki $x = 0, 1, \dots, M - 1$ redak I_L slike SCD metoda detektira indekse $n_{tL}(x)$, a za svaki $x = 0, 1, \dots, M - 1$ redak I_{De} slike SCD metoda detektira indekse $n_{tDe}(x)$. Na temelju $n_{tL}(x)$ i $n_{tDe}(x)$ pikseli izlazne binarne slike $I_B(x, y)$ postavljaju se u '0' prema (4.27), v. sliku 4.12 (e).



Slika 4.12: PBSCD metoda: (a) primjer ulazne $I(x, y)$ sive slike veličine $K \times M$ piksela, (b) gornji I_G i donji I_D dio slike prema (4.25), (c) lijevi I_L i desni I_{De} dio slike prema (4.25), (d) međurezultat binarizacije prema (4.26), (e) međurezultat binarizacije prema (4.27), (f) konačni rezultat binarizacije PBSCD metode (bijela boja predstavlja objekt, a crna pozadinu).

Primjenom (4.26) i (4.27) PBSCD metoda binarizira sivu sliku sirove pločice. Pikseli konačne binarne slike $I_B(x, y)$ s intenzitetom '0' predstavljaju pozadinu, a s intenzitetom '1' predstavljaju sirovu pločicu, v. sliku 4.12 (f). Budući da PBSCD metoda ne računa razinu praga za pojedini diobeni dio slike eliminiran je problem određivanja razine praga na granicama diobenih dijelova slike koji je objašnjen u poglavlju 4.2.1.

$$\begin{aligned} I_B(i, y) &= 0, \quad i = 0, 1, \dots, n_{tG}(y), \\ I_B(i, y) &= 0, \quad i = K - 1, K - 2, \dots, K - n_{tD}(y) \end{aligned} \quad (4.26)$$

$$\begin{aligned} I_B(x, i) &= 0, \quad i = 0, 1, \dots, n_{tL}(x), \\ I_B(x, i) &= 0, \quad i = M - 1, M - 2, \dots, M - n_{tDe}(x) \end{aligned} \quad (4.27)$$

Odabir broja redaka I_G i I_D slike ili broja stupaca I_L i I_{De} slike, tj. broja uzoraka (piksela) N signala \mathbf{s} , ovisi o samom sadržaju slike. U poglavlju 4.7.1 zaključeno je da broj uzoraka N signala \mathbf{s} s $\Delta A/\sigma = [5, 6)$ i $n_t = N/2$ mora biti veći od 60 da bi devijacija indeksa SCD metode bila 0 %. Za signale \mathbf{s} s $\Delta A/\sigma \geq 6$ i $n_t = N/2$ devijacija indeksa SCD metode je neovisna o broj uzoraka N . Iz prethodnog slijedi sljedeći uvjet: broja redaka I_G i I_D slike tj. broja stupaca I_L i I_{De} slike (N) mora biti takve veličine da 1D signali \mathbf{s} sadrže najmanje 30 piksela pozadine i 30 piksela sirove pločice. Ako je navedeni uvjet zadovoljen devijacija indeksa SCD metode za signale s $\Delta A/\sigma \geq 5$ će biti 0 % tj. detektirani indeks n_t uvijek će biti jednak stvarnom indeksu n_{Gt} . Na ovaj način riješen je jedan od problema metoda za binarizaciju sive slike na temelju prilagodljive razine praga, problem određivanja broja diobenih dijelova slike n_{sub} (poglavlje 4.2.1). Broj diobenih dijelova je četiri I_G , I_D , I_L i I_{De} uz prethodno postavljeni uvjet za N .

Lijevi i desni dijelovi I_G i I_D slike ne sadrže piksele objekta, v. sliku 4.13 (a). Isto vrijedi i za gornji i donji dio I_L i I_{De} slike, v. sliku 4.13 (b). Iako redovi I_L i I_{De} slike i stupci I_G i I_D slike ne sadrže piksele sirove pločice SCD metoda će ipak detektirati indekse $n_{tG}(y)$, $n_{tD}(y)$, $n_{tL}(x)$ i $n_{tDe}(x)$ te će prema (4.26) i (4.27) postaviti odgovarajuće piksele $I_B(x, y)$ slike u '0'. Iz navedenog slijedi da će u uglovima izlazne binarne slike $I_B(x, y)$ dio piksela biti pogrešno segmentiran kao sirova pločica, tj. dio piksela će imati intenzitet '1' iako ne predstavljaju sirovu pločicu npr. slika 4.14. Ako je potrebno ovaj problem se jednostavno može riješiti s dodatnom obradom binarne slike npr. praćenjem konture objekta [47] ili etiketiranjem (*engl. labeling*) [48].

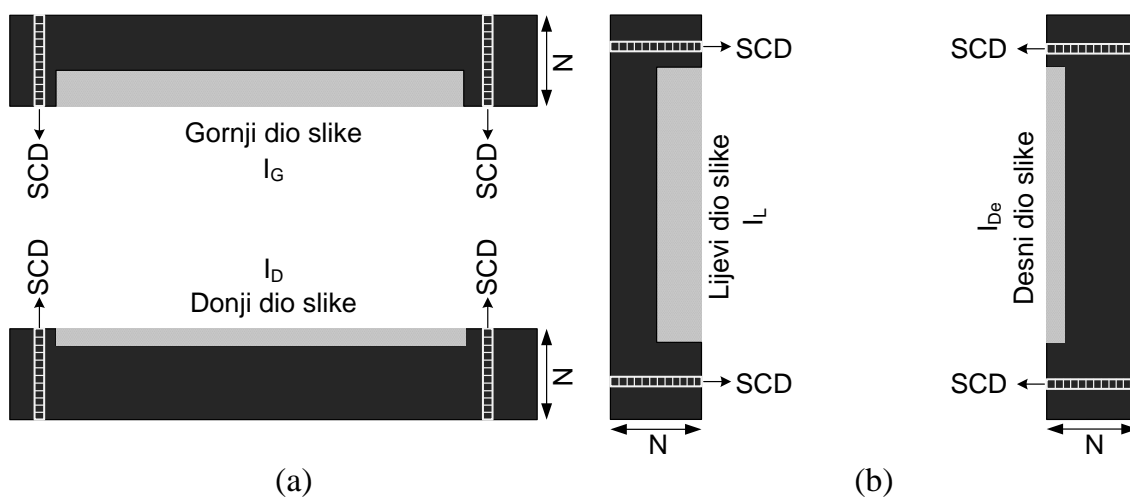
Tablica 4.3: Primjer broja operacija Otsu algoritma 4.1 i PBSCD algoritma 4.5 prema (4.28), (4.29), (4.30) i (4.31).

| $K \times M$ | Otsu | | | PBSCD ($n_{OP,PBSCD}$) | | |
|--------------------|--------------------|--------------------|--------------------|--------------------------|--------------------|--------------------|
| | n_{OP,h_n} | $n_{OP,Otsu}$ | Ukupno | $N = 60$ | $N = 120$ | $N = 250$ |
| 512×512 | 5.25×10^5 | 3.96×10^5 | 9.20×10^5 | 4.48×10^7 | 1.78×10^8 | 7.71×10^8 |
| 1024×1024 | 2.10×10^6 | 3.96×10^5 | 2.49×10^6 | 8.97×10^7 | 3.56×10^8 | 1.54×10^9 |
| 1500×1500 | 4.50×10^6 | 3.96×10^5 | 4.90×10^6 | 1.31×10^8 | 5.22×10^8 | 2.26×10^9 |

Pseudo kod sekvencijalnog algoritma PBSCD metode za binarizaciju sive slike prikazan je u algoritmu 4.5. Tablica 4.3 daje primjere broja operacija Otsu metode prema algoritmu 4.1 i PBSCD metode prema algoritmu 4.5 za $K \times M \in \{512 \times 512, 1024 \times 1024, 1500 \times 1500\}$ i $N \in \{60, 120, 250\}$. Broj operacija (zbrajanje, množenje, ispitivanje uvjeta) za Otsu metodu podijeljen je na dva dijela: broj operacija za izračun normaliziranog histograma n_{OP,h_n} i broj operacija za određivanje razine praga binarizacije Otsu metodom $n_{OP,Otsu}$. n_{OP,h_n} i $n_{OP,Otsu}$ računaju se prema (4.28) i (4.29) na temelju algoritma 4.1. Broj operacija PBSCD metode $n_{OP,PBSCD}$ sastoji se od operacija potrebnih za primjenu SCD metode ($n_{OP,SCD}$) na sve retke I_L i I_{De} slike i sve stupce I_G i I_D slike, gdje se $n_{OP,SCD}$ računa prema (4.30) na temelju algoritma 4.4. U (4.28) do (4.31) nije uračunato broj operacija potrebnih za izračun $I_B(x, y)$.

$$n_{OP,h_n} = K \times 2 \times M + B \quad (4.28)$$

$$n_{OP,Otsu} = (B - 1)[B \times 6 + 16] \quad (4.29)$$



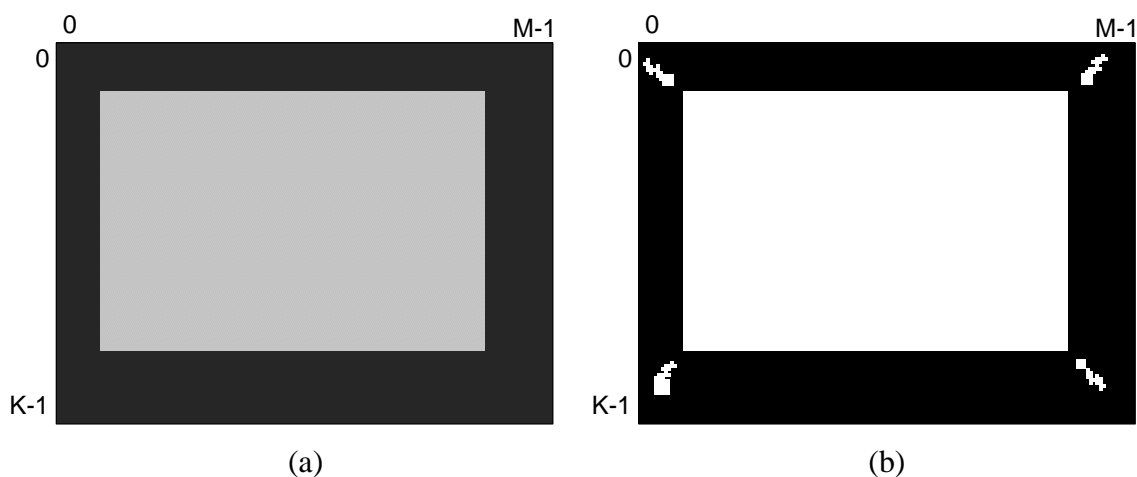
Slika 4.13: I_G i I_D slike s primjerom primjene SCD metode na stupce bez piksela sirove pločice (a), I_L i I_{De} slike s primjerom primjene SCD metode na redove bez piksela sirove pločice (b).

$$n_{OP,SCD} = (N - 1)[6 \times N + 11] \quad (4.30)$$

$$n_{OP,PBSCD} = 2 \times (M + K) \times n_{OP,SCD} \quad (4.31)$$

Iz tablice 4.3 može se zaključiti da za $N = 60$ i $K \times M = 512 \times 512$ omjer $n_{OP,PBSCD} : (n_{OP,h_n} + n_{OP,Otsu})$ je 48:1 tj. u najboljem slučaju (najmanji $n_{OP,PBSCD}$) broj operacija PBSCD metode je 48 puta veći u odnosu na Otsu metodu. U najgorem slučaju (najveći $n_{OP,PBSCD}$ u tablici 4.3) omjer broja operacija PBSCD i Otsu metode je 461:1.

Broj operacija $n_{OP,PBSCD}$ moguće je smanjiti upotrebom 1D integralne sume [49] u



Slika 4.14: Primjer ulazne slike sirove pločice (a) i izlazna binarna slika PBSCD metode s pogrešno klasificiranim pikselima u uglovima binarne slike (b).

Algoritam 4.5: Binarizacija sive slike PBSCD metodom.

1. *Ulaz:* $I(x, y)$, N
2. *Za* $x = 0, 1, \dots, K - 1$ *i* $y = 0, 1, \dots, M - 1$ $I_B(x, y) = 1$
3. *Podjeli* $I(x, y)$ *na* I_G, I_D, I_L *i* I_{De} *prema* (4.25)
4. **for** $y := 0 \rightarrow M - 1$ **do**
5. $s_G = I_G(0 \rightarrow N - 1, y)$
6. $s_D = I_D(N - 1 \rightarrow 0, y)$
7. $n_{tG}(y) = \min_{0 \leq n \leq N-2} J_{SCD, s_G}(n)$
8. $n_{tD}(y) = \min_{0 \leq n \leq N-2} J_{SCD, s_D}(n)$
9. $I_B(i, y) = 0, i = 0, 1, \dots, n_{tG}(y)$
10. $I_B(i, y) = 0, i = K - 1, K - 2, \dots, K - n_{tD}(y)$
11. **end for**
12. **for** $x := 0 \rightarrow K - 1$ **do**
13. $s_L = I_L(x, 0 \rightarrow N - 1)$
14. $s_{De} = I_{De}(x, N - 1 \rightarrow 0)$
15. $n_{tL}(x) = \min_{0 \leq n \leq N-2} J_{SCD, s_L}(n)$
16. $n_{tDe}(x) = \min_{0 \leq n \leq N-2} J_{SCD, s_{De}}(n)$
17. $I_B(x, i) = 0, i = 0, 1, \dots, n_{tG}(y)$
18. $I_B(x, i) = 0, i = M - 1, M - 2, \dots, M - n_{tDe}(y)$
19. **end for**

proračunu SCD metode. 1D integralna suma signala $s(i)$, $i = 0, 1, \dots, N - 1$ predstavlja novi signal $INT_s(i)$, $i = 0, 1, \dots, N$ izračunat prema (4.32). 1D integralna suma omogućava izračun sume $S_{a,b}$ bilo kojeg dijela signala $s(i)$ s jednom operacijom oduzimanja prema (4.33). Na slici 4.15 prikazan je primjer signala $s(i)$ s dva primjera izračuna sume dijela signala s 1D integralnom sumom. Vidljivo je da bez obzira na veličinu dijela signala $s(i)$ (na slici 4.15 označeni sa crtkanim pravokutnikom) uvijek su potrebna dva elementa $INT_s(a)$ i $INT_s(b + 1)$ za izračun $S_{a,b}$.

$$INT_s(0) = 0 \quad (4.32)$$

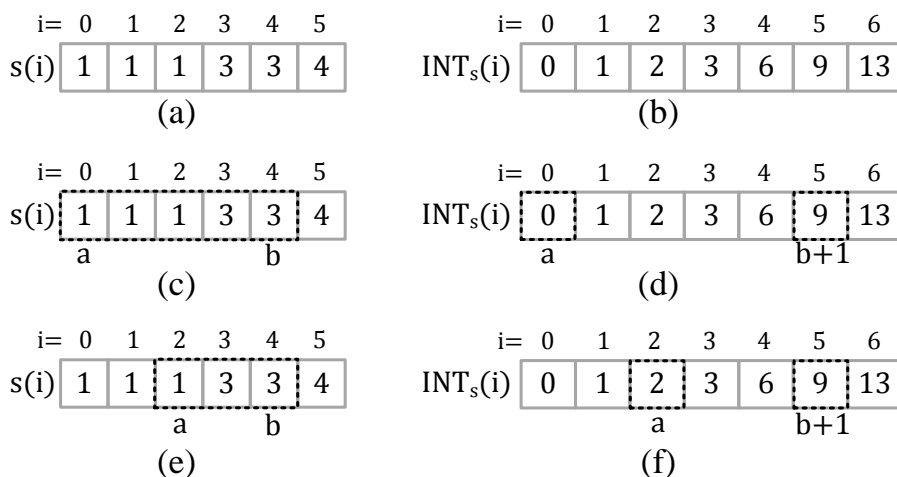
$$INT_s(i) = INT_s(i - 1) + s(i - 1), i = 1, 2, \dots, N$$

$$S_{a,b} = \sum_{i=a}^b s(i) = INT_s(b + 1) - INT_s(a), a \leq b \leq N - 1 \quad (4.33)$$

$$A_1(n) = \frac{1}{n + 1} [INT_s(n + 1) - INT_s(0)] = \frac{INT_s(n + 1)}{n + 1} \quad (4.34)$$

$$A_2(n) = \frac{1}{N - n - 1} [INT_s(N) - INT_s(n + 1)]$$

Primjenom 1D integralne sume, izračun srednjih vrijednosti iz (4.21) svodi se na jednu operaciju oduzimanja i dvije operacije dijeljenja prema (4.34). U algoritmu 4.6 prikazan je pseudo kod SCD metode s upotrebom 1D integralne sume za izračun srednjih vrijednosti $A_1(n)$ i $A_2(n)$ prema (4.34).



Slika 4.15: Primjer signala $s(i)$ (a), (b) 1D integralna suma signala $s(i)$, (c) i (d) primjer izračuna sume dijela signala za $a = 0, b = 4$ prema (4.33), (e) i (f) primjer izračuna sume dijela signala za $a = 2, b = 4$ prema (4.33).

Tablica 4.4: Primjer broja operacija PBSCD algoritma s 1D integralnom sumom prema (4.35) i (4.36).

| $K \times M$ | $PBSCD (n_{OP,PBSCD_{INT}})$ | | |
|--------------------|------------------------------|--------------------|--------------------|
| | $N = 60$ | $N = 120$ | $N = 250$ |
| 512×512 | 1.62×10^7 | 6.19×10^7 | 2.62×10^8 |
| 1024×1024 | 3.24×10^7 | 1.24×10^8 | 5.24×10^8 |
| 1500×1500 | 4.74×10^7 | 1.81×10^8 | 7.68×10^8 |

$$n_{OP,INT_{SCD}} = (N - 1)[2 \times N + 12] + 2 \times N \quad (4.35)$$

$$n_{OP,PBSCD_{INT}} = 2 \times (M + K) \times n_{OP,SCD_{INT}} \quad (4.36)$$

Broj operacija SCD metode smanjuje se s (4.30) na (4.35) tj. oko tri puta manje. Također prema (4.36) broj operacija PBSCD metode smanjio se oko tri puta. Uspoređujući vrijednosti iz tablica 4.3 i 4.4 može se zaključiti da u najboljem slučaju (najmanji $n_{OP,PBSCD_{INT}}$) broj operacija PBSCD metode je 17 puta veći u odnosu na Otsu metodu, a u najgorem slučaju (najveći $n_{OP,PBSCD_{INT}}$) broj operacija je 156 puta veći. Iako je broj operacija PBSCD metode smanjen i dalje je puno veći u odnosu na Otsu metodu.

S većim brojem operacija raste i vrijeme izvođenja algoritma, a time se i smanjuje mogućnost rada u stvarnom vremenu. Budući da broj operacija PBSCD metode nije moguće dodatno smanjiti, može se zaključiti da je potrebno razviti paralelni algoritam PBSCD metode da bi se metoda mogla upotrebljavati za rad u stvarnom vremenu.

Algoritam 4.6: SCD metoda s 1D integralnom slikom.

```

1. Ulaz:  $N, s(i); A_0, A_1, V_0, V_1, INT_s(0), J_{SCD2}, n_t = 0$ 
2. for  $i := 1 \rightarrow N$  do
3.    $INT_s(i) = INT_s(i - 1) + s(i - 1)$ 
4. end for
5. for  $n := 0 \rightarrow N - 2$  do
6.    $A_0 = INT_s(n + 1)/(n + 1)$ 
7.    $A_1 = INT_s(N) - INT_s(n + 1)$ 
8.    $A_1 = A_1/(N - n - 1)$ 
9.   for  $i := 0 \rightarrow n$  do
10.     $V_0 = V_0 + (s(i) - A_0)^2$ 
11.   end for
12.   for  $i := n + 1 \rightarrow N - 1$  do
13.     $V_1 = V_1 + (s(i) - A_1)^2$ 
14.   end for
15.    $J_{SCD1} = V_0 + V_1$ 
16.   if  $J_{SCD1} \leq J_{SCD2}$  then
17.      $J_{SCD2} = J_{SCD1}, n_t = n$ 
18.   end if
19.    $A_0, A_1, V_0, V_1 = 0$ 
20. end for

```

4.8. Paralelni GPU algoritam PBSCD metode

PBSCD metoda dijeli $I(x, y)$ sliku na četiri neovisna dijela I_G , I_D , I_L i I_{De} . Na sve stupce I_G i I_D slike i na sve retke I_L i I_{De} slike neovisno se primjenjuje SCD metoda. Iz navedenog može se zaključiti da je PBSCD metoda pogodna za primjenu na paralelnim platformama.

4.8.1. Podjela paralelnih platformi

Paralelne platforme se mogu grubo podijeliti u dvije grupe:

- višeprocessorski sustavi sa zajedničkom memorijom i
- višeprocessorski sustavi s raspodijeljenom memorijom.

Na višeprocessorskim sustavima sa zajedničkom memorijom svi procesori (jezgre) dijele glavnu memoriju. U ove sustave spadaju današnja osobna računala s višejezgrenim procesorima (npr. Intel Core i5 [50], AMD A serija [51]), sustavi s IBM Cell procesorima [52], FPGA sustavi [53] i sustavi s grafičkim procesorom (GPU) s GPGPU (*engl. General-Purpose Computing on GPU*) mogućnostima [54].

Na višeprocessorskim sustavima s raspodijeljenom memorijom svaki procesor ima vlastitu memoriju čiji sadržaj nije dostupan drugim procesorima. Procesori međusobno komuniciraju razmjennom poruka putem računalne mreže. U ove sustave spadaju klasteri (sustav računala umreženih korištenjem brze lokalne mreže – LAN) i grid sustavi (sustav računala umreženih korištenjem računalnih mreža na različitim geografskim lokacijama) [55]. U ovu kategoriju se također mogu ubrojiti i sustavi s više GPU kartica povezanih putem PCI-E (*engl. Peripheral Component Interconnect Express*) sabirnice.

Višeprocessorski sustavi s raspodijeljenom memorijom najčešće se koriste u aplikacijama za paralelnu obradu velikih količina podataka u kojima vrijeme potrebno za komunikaciju između računala ne predstavlja ograničavajući faktor.

U ovom radu upotrebljavaju se višeprocessorski sustavi sa zajedničkom memorijom budući da se obrađuju manje količine podataka (reda veličine 10 MB) i potrebna je velika brzina prijenosa podataka između procesora (jezgri).

Za programiranje višeprocessorskih sustava sa zajedničkom memorijom može se koristiti nekoliko dodatnih biblioteka za paralelno programiranje. Za višejezgrene CPU sustave najpoznatije su:

- MPI (*engl. Message Passing Interface*),

- OpenMP (*engl. Open Multi-Processing*),
- pThreads,
- Windows Threading,
- Intel Threading Building Block i sl.

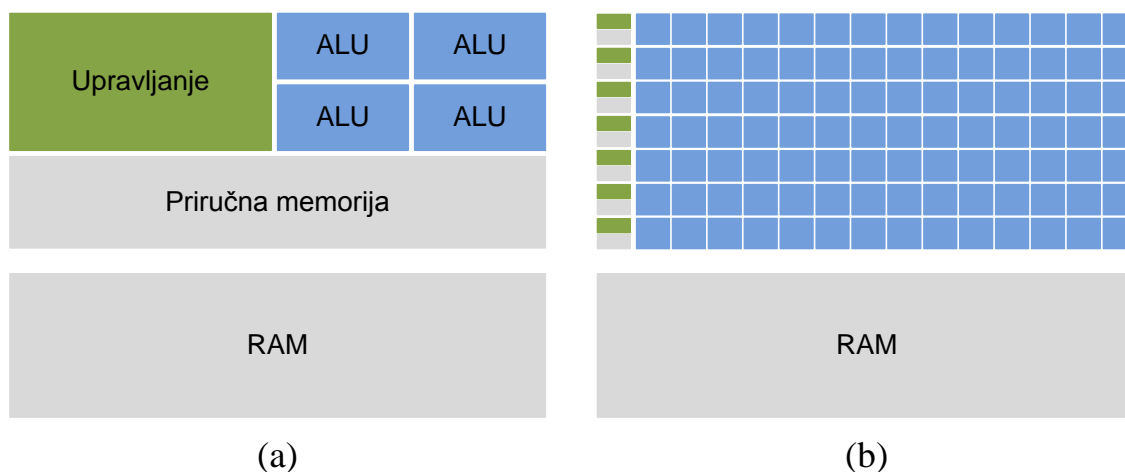
Za GPGPU uređaje najpoznatije biblioteke za paralelno programiranje su:

- NVIDIA CUDA Toolkit (*engl. Compute Unified Device Architecture*),
- OpenCL (*engl. Open Computing Language*),
- DirectX 11 API,
- OpenACC i sl.

U sljedećim poglavljima detaljno je objašnjena NVIDIA CUDA GPGPU paralelna platforma koja je razmatrana u ovom radu za paralelizaciju PBSCD metode.

4.8.2. NVIDIA CUDA paralelna platforma

Grafička kartica s GPU se danas nalazi u gotovo svim osobnim računalima. Upotrebljava se za generiranje slike i prikaz slike na monitoru. Kao i CPU, GPU se sastoji od mnoštva tranzistora (reda veličine 10^{10}), ali za razliku od CPU-a, većina tranzistora u GPU je namijenjena za obradu podataka, a ne za kontrolu toka podatka i priručnu memoriju, cf. slika 4.16. S uvođenjem NVIDIA CUDA paralelne platforme programerima je omogućeno da na jednostavan način iskoriste GPU za opću obradu podatak (GPGPU). Od 2006 godine do danas NVIDIA je razvila niz grafičkih kartica s različitim GPGPU mogućnostima. GPGPU mogućnosti su označene sa CC brojem (*engl. Compute Capability*). Detaljnije informacije o



Slika 4.16: Shematski prikaz rasporeda tranzistora u CPU (a) i GPU (b).

GPGPU mogućnostima za pojedini CC (u literaturi poznat kao i CUDA verzija) se može pronaći u prilogu 10.2. Pored NVIDIA, GPU-e s GPGPU mogućnostima proizvodi i AMD (STREAM tehnologija) [56]. U ovom radu se razmatra NVIDIA CUDA paralelna platforma zbog dominacije na tržištu u pogledu GPGPU razvoja i česte primjene u znanstvenim istraživanjima [57].

GPU je pogodan za računsku obradu podatak koju je moguće podijeliti na veliki broj manjih dijelova čiji rezultati neovise jedan o drugom. Također je pogodno da je omjer aritmetičkih operacija (zbrajanje, oduzimanje, množenje, itd.) u odnosu na memorijske operacije (spremanje i čitanje u memoriju) velik. CUDA implementira SIMT paralelni model (*engl. Single Instruction Multiple Thread – jedna instrukcija više niti*) sličan SIMD modelu (*engl. Single Instruction Multiple Data – jedna instrukcija više podataka*) po Flynnovoj taksonomiji [58]. Po SIMD modelu ista instrukcija se istovremeno izvršava nad svim podacima u određenoj grupi podataka. Za razliku od SIMD modela, SIMT model omogućuje programeru da odlučuje koja će nit izvršiti koju instrukciju nad određenim podatkom iz grupe podataka. SIMT model omogućuje programeru veću fleksibilnost u implementaciji algoritama u odnosu na SIMD model.

U literaturi se može pronaći niz problema koji su uspješno implementirani na CUDA platformi. U [59, 60] implementirana je brza Fourierova transformacija na CUDA-i i postignuta su ubrzanja od 8 – 40 puta u odnosu na najpoznatije implementacije na CPU [61]. U [62] autori su postigli ubrzanja od 360 puta u odnosu na CPU implementaciju za izračun Haralick značajki tekstone na slici. Autori u [63, 64] implementirali su K-means algoritam na CUDA GPU i postigli ubrzanja veća od 100 puta. Na sustavu s 4 GPU-a autori su u [65] implementirali 3D simulaciju električnog modela stanica srca i postigli ubrzanja od 2460 puta u odnosu na CPU implementaciju.

Da bi bilo moguće postići navedena ubrzanja potrebno je detaljno razmotriti sklopovski i programski model CUDA paralelne platforme.

4.8.3. Sklopovski model NVIDIA CUDA paralelne platforme

CUDA GPU se sastoji od skalabilne mreže multiprocesora (*engl. Streaming Multiprocessors – SM*) i memorije, slika 4.17. Memorija je podijeljena na dva dijela: globalna memorija i lokalna memorija. Globalna GDDR (*engl. Graphics Double Data Rate*) memorija zajednička je svim SM-ovima i nalazi se na grafičkoj kartici a ne na GPU. Lokalna memorija

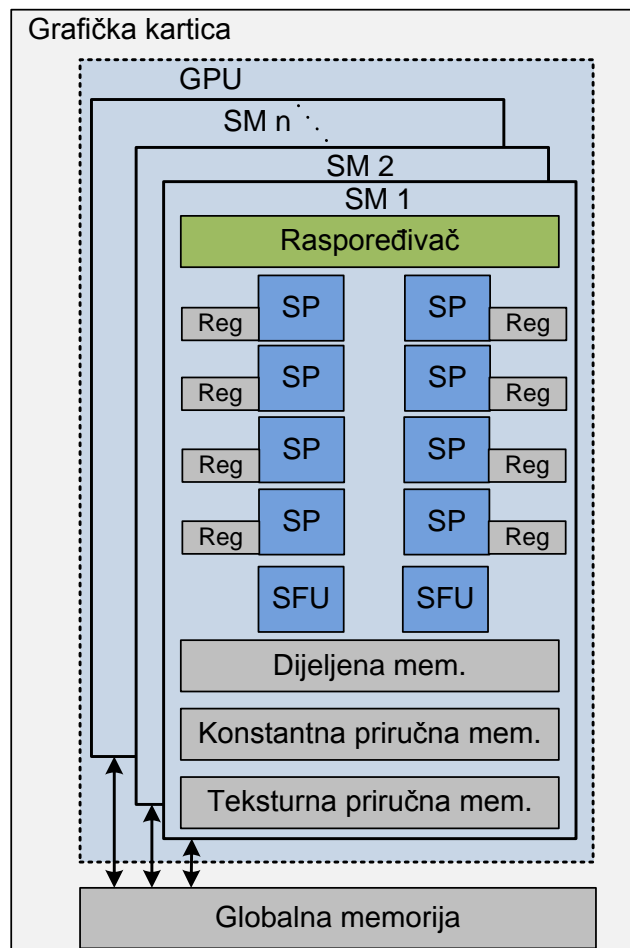
se nalazi na samom GPU i može se podijeliti na više kategorija ovisno o mogućnostima samog CUDA GPU određenim sa *CC* brojem [66, 67, 68].

Ovisno o *CC* broju SM se sastoji od:

- skalarnih procesora (SP),
- specijalnih funkcijskih jedinica (SFU),
- jedinice za učitavanje/spremanje podatak (LD/ST),
- dijeljene memorije,
- registara, itd.

Najvažniji dio SM-a su skalarni procesori koji su zaduženi za aritmetičke (zbrajanje, oduzimanje, množenje i sl.) i logičke (ILI, I, NE i sl.) operacije nad cjelobrojnim podacima i podacima jednostruke preciznosti. U prvoj verziji CUDA GPU (CUDA 1.0 GPU, *CC* = 1.0) broj SP-ova je osam. Broj SP-ova na SM-u se povećava s novijim CUDA verzijama. Multiprocesor CUDA 3.5 GPU-a sastoji se od 192 SP-a.

SM također sadrži i određeni broj SFU-ova. SFU je zadužen za izvršavanje



Slika 4.17: Sklopovski model CUDA 1.0 GPU.

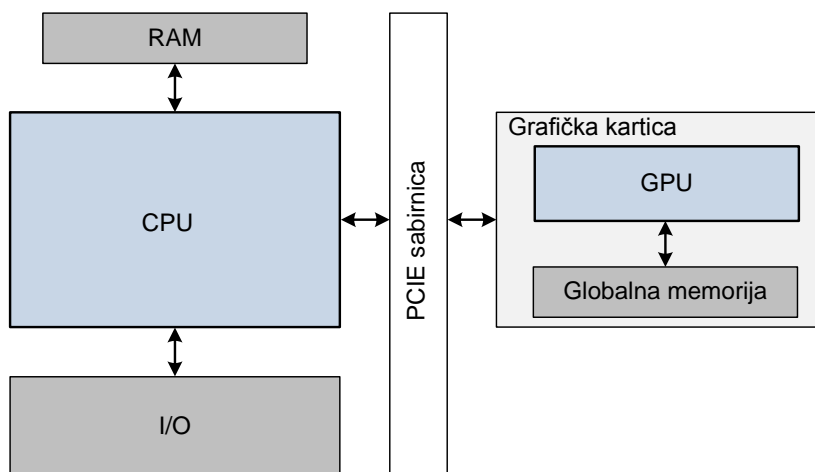
Tablica 4.5: Latencije (u broju taktova) za pojedine tipove memorija u CUDA GPU.

| | <i>Registri</i> | <i>Dijeljena memorija</i> | <i>Konstantna priručna memorija</i> | <i>Teksturna priručna memorija</i> | <i>Globalna memorija</i> |
|------------------|-----------------|---------------------------|-------------------------------------|------------------------------------|--------------------------|
| <i>Latencija</i> | 1 | 1 – 32 | 50 – 600 | 200 – 650 | 400 – 700 |

transcendentalnih funkcija kao što su sinus, kosinus, eksponent i sl.

Svaki SP ima pristup registrima, memorije sa signalom takta jednake frekvencije kao i SP-ovi. Veličina registara ovisi o CUDA verziji. Za CUDA 1.0 ukupna veličina registara je 32 KB, za CUDA 3.5. ukupna veličina registara je 256 KB. Pored registara svi SP-ovi na SM-u imaju pristup dijeljenoj memoriji preko koje mogu međusobno dijeliti podatke. Također na svakom SM-u se nalaze i dvije priručne memorije: konstantna priručna memorija i teksturna priručna memorija. SP-ovi imaju samo mogućnost čitanja podataka iz obe priručne memorije. Konstantna priručna memorija omogućava svim SP-ovima na SM-u brzi pristup konstantnim podacima koji se nalaze u globalnoj memoriji. Teksturna priručna memorija omogućava brži pristup 1D, 2D i 3D podacima iz globalne memorije. Također teksturna priručna memorija, pomoću teksturne jedinice, omogućava i dodatno filtriranje podataka i različite načine adresiranja podataka. Navedene memorije se razlikuju po vremenu pristupa podacima (latencijama). Vrijeme pristupa najčešće se izražava u broju perioda signala takta potrebnih za čitanje/pisanje podataka u memoriji. Tablica 4.5 prikazuje latencije za pojedine tipove memorija na CUDA GPU. Najveće latencije pokazuje globalna memorija, a najmanje registri što je i za očekivati ako se uzme u obzir sklopovski model CUDA GPU prikazan na slici 4.17 [69].

Grafička kartica sa CUDA GPU spaja se sa CPU-om putem brze PCIe (*engl.*

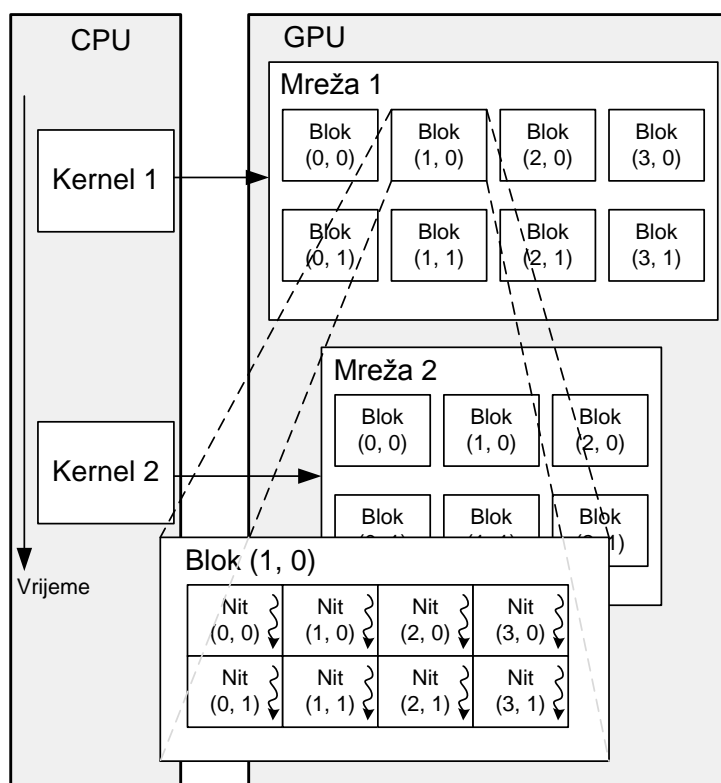


Slika 4.18: Primjer spajanja grafičke kartice sa CPU-om putem PCIe sabirnice.

Peripheral Component Interconnect Express) sabirnice, v. sliku 4.18. PCIE 2.0 sabirnica omogućava brzi CPU – GPU dvosmjerni prijenos podataka od 5 GB/s, tj. omogućava istovremeni prijenos podataka sa CPU na GPU brzinom od 5 GB/s i s GPU na CPU brzinom od 5 GB/s. Teoretski, prijenos sive slike veličine 1024×1024 piksela sa CPU na GPU i obratno moguć je u 0.2 ms što ne predstavlja ograničavajući faktor za rad u stvarnom vremenu metoda razmatranih u ovom radu, v. poglavlje 3.3.

4.8.4. Programski model NVIDIA CUDA paralelne platforme

Temelj CUDA paralelnog programskog modela je *kernel* (jezgra). Kernel predstavlja C funkciju koja se izvršava na GPU-u. Kernel se poziva sa CPU strane i izvršava se s više niti na GPU. Niti na GPU su organizirane u blokove jednake veličine, a blokovi u mrežu, v. sliku 4.19. Svaki blok je označen s jedinstvenim indeksom ID_B (na slici 4.19 za *Blok(3,1)* $ID_B = (3,1)$), a unutar bloka svaka nit je označena s indeksom ID_T (na slici 4.19 za *Nit(3,1)* $ID_T = (3,1)$). Na temelju kombinacije ID_B i ID_T svaka nit je označena s jedinstvenim indeksom ID (na slici 4.19 za *Nit(3,1)* i *Blok(1,0)* $ID = (7,1)$). Sve niti izvršavaju isti kernel, ali nad različitim podacima određenim s ID od niti. ID uz upotrebu *if* uvjeta također

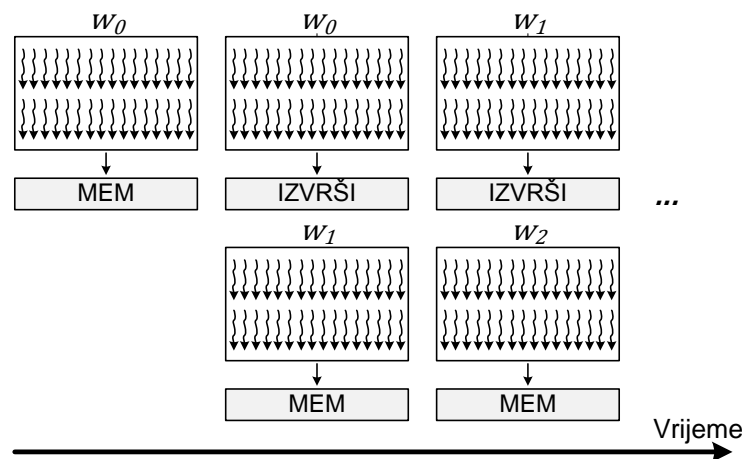


Slika 4.19: CUDA programski model.

omogućava programeru kontrolu nad grananjem kernela. ID_B , ID_T i ID mogu biti jednodimenzionalni, dvodimenzionalni (primjer na slici 4.19) ili trodimenzionalni indeksi.

U jednom trenutku na GPU se izvršava jedan kernel. Blokovi iz mreže se raspoređuju po SM-ovima. Sve niti iz istog bloka se izvršavaju na istom SM-u. Niti iz istog bloka su podijeljene u grupe od 32 niti koje se nazivaju *osnovama* (engl. *warp*). Osnova sadrži niti iz istog bloka s uzastopnim rastućim ID -om. Niti iz iste osnove zajedno se izvršavaju na SM-u. U slučaju da niti osnove slijede isti put izvođenja (bez grananja) sve niti osnove izvedu jednu instrukciju (ako instrukcija traje jednu periodu signala takta) u četiri periode signala takta na CUDA 1.0 GPU (za $CC \geq 2.0$ niti osnove izvedu jednu instrukciju u jednoj periodi signala takta). Niti unutar bloka mogu međusobno dijeliti podatke upotrebom dijeljene memorije. Također niti unutar bloka se mogu sinkronizirati budući da se izvode na istom SM-u, globalna sinkronizacija svih niti u svim blokovima nije moguća.

Broj niti unutar bloka je ograničen. Za $CC < 2.0$ blok se može sastojati od maksimalno 512 niti, a za $CC \geq 2.0$ blok se može sastojati od maksimalno 1024 niti. Broj niti u bloku je puno veći od broja SP-ova na SM-u. Razlog tomu je „sakrivanje“ latencije memorije prema tablici 4.5. Blok od 512 niti GPU dijeli na 16 osnova w_0, \dots, w_{15} . Potrebno je od 400 do 700 perioda signala takta da bi se podaci pročitali iz globalne memorije za niti iz w_0 . Učitavanjem podataka za w_0 , niti iz w_0 se izvode, ali se istovremeno pripremaju podaci za niti iz w_1 , v. sliku 4.20. Na ovaj način latencija globalne memorije je „sakrivena“ [70]. Broj blokova unutar mreže je ograničen na 65535 blokova u dvije dimenzije za $CC < 2.0$. Ukupan broj niti za $CC < 2.0$ je $65535^2 \times 512 = 2.19 \times 10^{12}$ niti, i s novijim CUDA verzijama dodatno raste.



Slika 4.20: Pojednostavljeni vremenski prikaz izvođenja osnova na CUDA GPU.

Da bi se kernel izveo na CUDA GPU potrebno je nekoliko koraka:

1. alokacija memorije na GPU za ulazne i izlazne podatke,
2. kopiranje ulaznih podataka iz RAM memorije CPU-a u globalnu memoriju GPU-a,
3. izvođenje kernela na GPU s postavljenim parametrima kernela (dimenzija mreže, dimenzija bloka) i parametrima funkcije (pokazivači na ulazne i izlazne podatke).
4. sinkronizacija GPU-a,
5. kopiranje izlaznih podataka iz globalne memorije GPU-a u RAM memoriju CPU-a,
6. dealokacija memorije na GPU za ulazne i izlazne podatke.

Budući da globalna memorija GPU-a i RAM memorija CPU-a predstavljaju dva različita adresna prostora, potrebna su prva dva koraka da bi ulazni podaci (npr. siva slika) bili dostupni nitima na GPU.

Nakon kopiranja podataka na GPU potrebno je izvesti kernel. Kernel se poziva od strane CPU-a, a izvodi se na GPU. Poziv se sastoji od dvije vrste parametara: parametri kernela i parametri funkcije. Na slici 4.21 (a) prikazan je primjer poziva kernela. Varijable unutar <<< ... >>> predstavljaju parametre kernela. Prvi parametar (*dimMreze* na slici 4.21 (a)) predstavlja dimenziju mreže tj. broj blokova u mreži. Drugi parametar (*dimBloka* na slici 4.21 (a)) predstavlja dimenziju bloka, tj. broj niti u bloku. Varijable unutar (...) predstavljaju parametre funkcije, pokazivači na ulazne i izlazne podatke u globalnoj memoriji GPU-a. Na slici 4.21 *fParam1* i *fParam2* predstavljaju ulazne cjelobrojne vektore, a *fParam3* predstavlja izlazni cjelobrojni vektor.

```
Primjer_kernela<<<dimMreze, dimBloka>>>(fParam1, fParam2, fParam3);
```

(a)

```
__global__ void Primjer_kernel(int *fParam1, int *fParam2, int *fParam3)
{
    int ID = blockIdx.x * blockDim.x + threadIdx.x;
    fParam3[ID] = fParam1[ID] + fParam2[ID];
}
```

(b)

Slika 4.21: Primjer poziva kernela (a), primjer kernela za zbrajanje dva vektora (*fParam1* i *fParam2*) i spremanje rezultata u *fParam3* vektor (b).

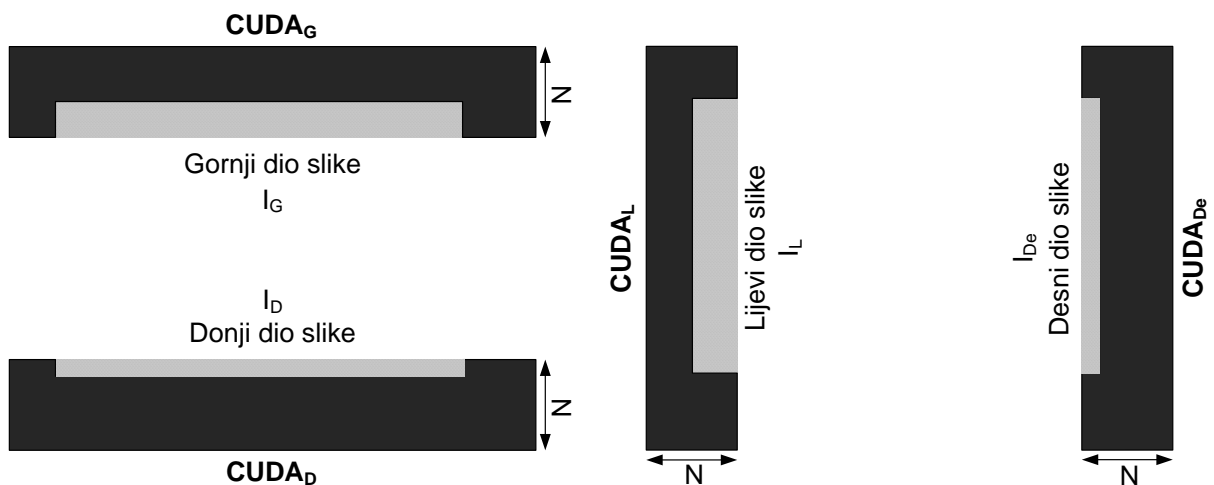
Primjer jednostavnog kernela je prikazan na slici 4.21 (b). Ključna riječ `__global__` označava da će se funkcija izvršiti na GPU, a ne na CPU. Kernel također karakteriziraju i višedimenzionalne varijable `blockIdx` (predstavlja ID_B), `threadIdx` (predstavlja ID_T) i `blockDim` (predstavlja ukupan broj niti u bloku) na temelju kojih se može odrediti ID od niti. Svaka nit na temelju svog ID -a izvršava određene operacije nad podacima. Za slučaj sa slike 4.21 (b) nit ID zbraja ID podatak ulaznih vektora $fParam1$ i $fParam2$ i sprema ih u vektor $fParam3$.

Nakon poziva kernela potrebno je sinkronizirati GPU. Sinkronizacija se izvodi s CPU strane i jamči završetak izvođenja kernela na GPU. S uspješnim završetkom kernela potrebno je kopirati rezultate iz globalne memorije GPU-a u RAM memoriju CPU-a. Za primjer sa slike 4.21 (b) potrebno je kopirati podatke vektora $fParam3$. Ako globalna memorija GPU-a više nije potrebna, nakon kopiranja podataka u RAM memoriju od CPU-a, potrebno ju je osloboditi (dealocirati) sa čime završava proces izvođenja kernela.

4.8.5. CUDA PBSCD algoritam

CUDA PBSCD algoritam se sastoji od 4 kernela: $CUDA_G$, $CUDA_D$, $CUDA_L$ i $CUDA_{De}$. Svaki kernel se koristi za primjenu SCD metode na pojedini dio ulazne slike $I(x, y)$ prema slici 4.22. U nastavku se detaljno razmatra $CUDA_L$ kernel, ostali kerneli slijede sličan pristup ali nad drugim dijelovima slike.

$CUDA_L$ kernel izvršava se nad svim redovima I_L slike. Broj blokova u mreži jednak je broju redaka $I(x, y)$ slike, tj. $dimMreze = K$. Broj niti u bloku ovisi o N prema (4.37) gdje



Slika 4.22: Veza između pojedinih CUDA PBSCD kernela ($CUDA_G$, $CUDA_D$, $CUDA_L$ i $CUDA_{De}$) i dijelova $I(x, y)$ slike prema izrazu (4.25).

$[\cdot]$ predstavlja funkciju najveće cijelo. Broj niti u bloku mora biti potencija broja dva budući da se u jednom dijelu kernela upotrebljava redukcijski algoritam [71]. Ukupan broj niti potreban za izvršenje $CUDA_L$ kernela je $2^{\lceil \log_2 N \rceil} \times K$ što za primjer slike dimenzije 2048×2048 piksela i $N = 250$ iznosi $2^8 \times 2048 = 524288$ što je puno manje od ukupnog broja niti CUDA 1.0 GPU-a.

$$\dimBloka = 2^{\lceil \log_2 N \rceil} \quad (4.37)$$

Niti s indeksom $ID_T < N$ u bloku označenim s indeksom ID_B učitavaju intenzitete piksela $I(ID_B, ID_T)$ u vektor $\omega(ID_T)$. Vektor $\omega(i)$, $i = 0, 1, \dots, N - 1$ je spremljen u dijeljenu memoriju SM-a s čime je nitima ID_B bloka omogućen brzi pristup podacima ID_B retka I_L slike. Nakon sinkronizacije niti ID_B bloka, niti s indeksom $ID_T \leq N - 2$ dijele vektor $\omega(i)$ u dva dijela $\omega_1(i)$ i $\omega_2(i)$ prema (4.38). Na temelju $\omega_1(i)$ i $\omega_2(i)$ niti s indeksom $ID_T \leq N - 2$ računaju $A_1(ID_T)$ i $A_2(ID_T)$ prema (4.21). Nakon ponovne sinkronizacije niti s indeksom $ID_T \leq N - 2$ računaju $J_{SCD}(ID_T)$ prema (4.23) i rezultat spremaju u vektor $v(i)$, $i = 0, 1, \dots, \dimBloka - 1$ koji se nalazi u dijeljenoj memoriji SM-a. Niti s indeksom $ID_T > N - 2$ postavljaju vrijednost $v(ID_T)$ na maksimalni broj jednostruke preciznosti ($maxf$). Nakon izvršene sinkronizacije vektor $v(i)$ sadrži vrijednosti $J_{SCD}(i)$ za $i = 0, 1, \dots, N - 2$ i $maxf$ za $i = N - 1, N, \dots, \dimBloka - 1$.

$$\begin{aligned} \omega_1(i) &= \omega(i), & i &= 0, 1, \dots, ID_T \\ \omega_2(i) &= \omega(i + ID_T + 1), & i &= 0, 1, \dots, N - ID_T - 1 \end{aligned} \quad (4.38)$$

Završetak $CUDA_L$ kernela sastoji se od pronalaska indeksa $n_{tL}(ID_B)$ i postavljanju vrijednosti piksela binarne $I_B(x, y)$ slike u nula prema (4.27). Indeks $n_{tL}(ID_B)$ određuje se upotrebom redukcijskog algoritma. Učinkoviti redukcijski algoritam za izračun sume vektora na CUDA GPU objašnjen je u [71]. Za određivanje $n_{tL}(ID_B)$ redukcijski algoritam se primjenjuje istovremeno na dva vektora u dijeljenoj memoriji: $v(i)$ i $indeks(i)$. Na početku redukcijskog algoritma vektor $indeks(i)$ sadrži uzastopne rastuće cjelobrojne vrijednosti indeksa prema (4.39). Redukcijski algoritam se sastoji od $\log_2 \dimBlock$ koraka. U prvom koraku se koristi $\dimBloka/2$ niti i u svakom sljedećem koraku broj korištenih niti se smanjuje dva puta. Nit ID_T uspoređuje vrijednosti $v(ID_T)$ i $v(ID_T + \dimBloka/2^{korak})$. Ako je uvjet (4.40) zadovoljen, $v(ID_T + \dimBloka/2^{korak})$ se kopira u $v(ID_T)$ i $index(ID_T + \dimBloka/2^{korak})$ se kopira u $indeks(ID_T)$. Nakon svakog koraka niti se sinkroniziraju. U zadnjem koraku minimalna vrijednost vektora $v(i)$ se nalazi u $v(0)$, a $n_{tL}(ID_B)$ se nalazi u $indeks(0)$.

$$\text{indeks}(i) = i, i = 0, 1, \dots, \text{dimBloka} - 1 \quad (4.39)$$

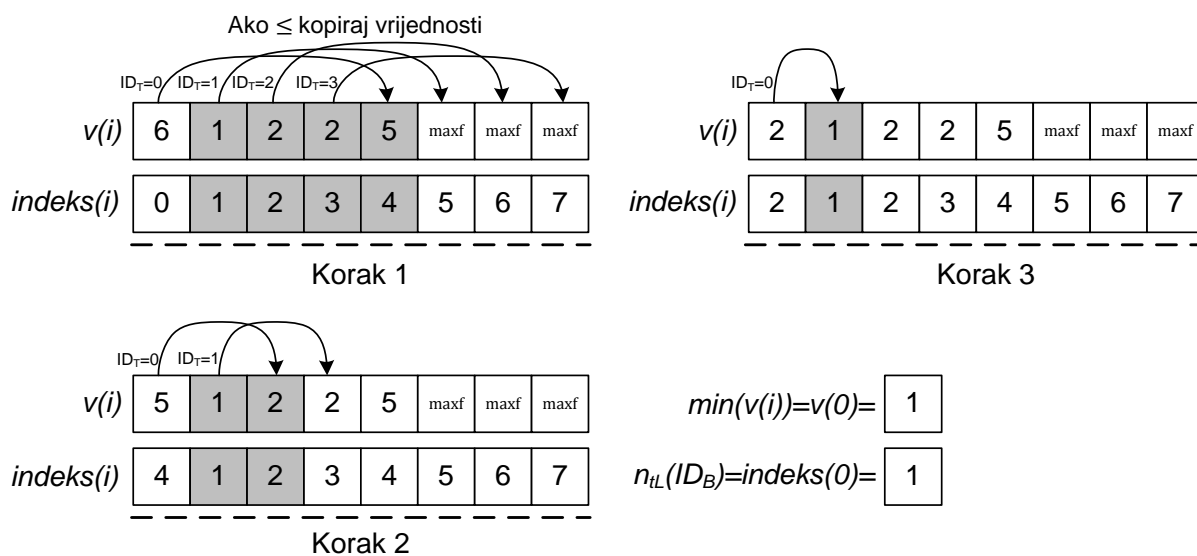
$$v\left(ID_T + \frac{\text{dimBloka}}{2^{\text{korak}}}\right) \leq v(ID_T), ID_T < \frac{\text{dimBloka}}{2^{\text{korak}}} \quad (4.40)$$

$$\text{korak} = 1, 2, \dots, \log_2 \text{dimBloka}$$

Na slici 4.23 prikazan je primjer određivanja $n_{tL}(ID_B)$. Za primjer na slici 4.23 $N = 6$, a iz (4.37) proizlazi da je $\text{dimBloka} = 8$. Za određivanje $n_{tL}(ID_B)$ potrebna su $\log_2 8 = 3$ koraka. U prvom koraku nit s $ID_T = 0$ uspoređuje vrijednosti $v(0) = 6$ i $v(4) = 5$. Budući da je $v(4)$ manji od $v(0)$, $v(4)$ se kopira u $v(0)$ i $\text{indeks}(4)$ se kopira u $\text{indeks}(0)$. Isto se ponavlja za ostale niti u prvom koraku. Nakon trećeg koraka $n_{tL}(ID_B)$ je spremljen u $\text{indeks}(0)$. Sve niti s $ID_T \leq n_{tL}(ID_B)$ postavljaju vrijednost izlazne binarne slike u nula prema formuli (4.41) s čime završava $CUDA_L$ kernel.

$$I_B(ID_B, ID_T) = 0, ID_T \leq n_{tL}(ID_B), ID_B = 0, 1, \dots, K - 1 \quad (4.41)$$

Ostali kerneli slijede istu metodologiju. $CUDA_{De}$ kernel određuje $n_{tDe}(ID_B)$ za svaki redak I_{De} slike i postavlja vrijednost izlazne binarne slike u nula prema (4.42). $CUDA_G$ kernel određuje $n_{tG}(ID_B)$ za svaki stupac I_G slike i postavlja vrijednost izlazne binarne slike prema (4.43). $CUDA_D$ kernel određuje $n_{tD}(ID_B)$ za svaki stupac I_D slike i postavlja vrijednost izlazne binarne slike prema (4.44).



Slika 4.23: Primjer određivanja $n_{tL}(ID_B)$ za $N = 6$ upotrebom redukcijuskog algoritma. Sivo obojeni elementi predstavljaju pronađenu manju vrijednost u pojedinom koraku, crtkana linija predstavlja sinkronizacijske točke, maxf predstavlja maksimalni broj jednostruke preciznosti.

$$I_B(ID_B, M - n_{tDe}(ID_B) + ID_T) = 0, N > ID_T \geq n_{tDe}(ID_B), \quad (4.42)$$

$$ID_B = 0, 1, \dots, K - 1$$

$$I_B(ID_T, ID_B) = 0, ID_T \leq n_{tG}(ID_B), ID_B = 0, 1, \dots, M - 1 \quad (4.43)$$

$$I_B(K - n_{tD}(ID_B) + ID_T, ID_B) = 0, N > ID_T \geq n_{tD}(ID_B), \quad (4.44)$$

$$ID_B = 0, 1, \dots, M - 1$$

Formalna procedura za izvođenje CUDA PBSCD algoritma prikazana je u algoritmu 4.7. Ako je veličina $I(x, y)$ slike jednaka za sve testirane slike, što je u većini slučajeva točno budući da se dimenzije sirovih keramičkih pločica mijenjaju s promjenom tipa pločice koje se proizvode, linije 1 i 2 u algoritmu 4.7 potrebno je izvršiti samo jednom na početku CUDA PBSCD algoritma, a liniju 8 samo jednom na kraju CUDA PBSCD algoritma.

4.9. Eksperimentalni rezultati

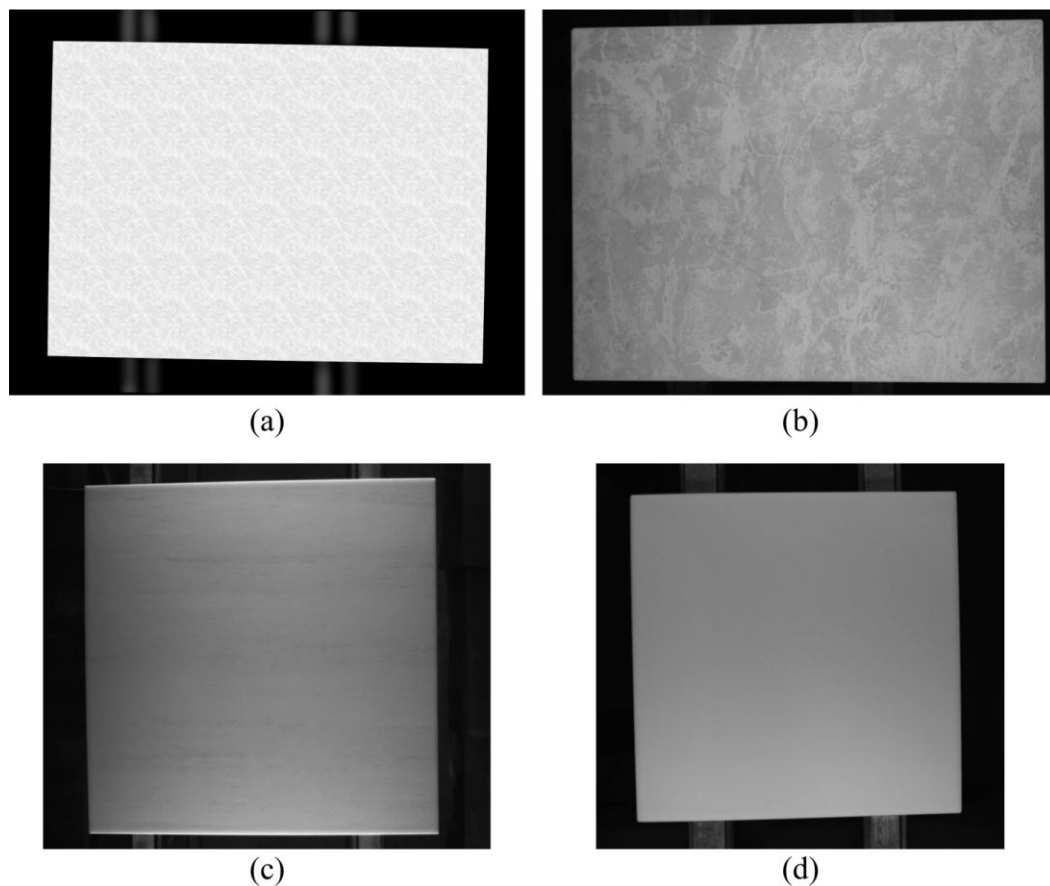
Metode za binarizaciju sive slike sirove keramičke pločice razmatrane u ovom radu (Otsu metoda, metoda minimalne pogreške, metoda minimalne razlike entropija i PBSCD metoda) implementirane su u MATLAB 2009b razvojnom okruženju. Metode su testirane na dva načina. Prvi test predstavlja određivanje pogreške binarizacije sive slike sirove keramičke pločice. Drugi test predstavlja mjerenje vremena izvođenja svih metoda u MATLAB 2009b okruženju, a zatim mjerenjem vremena izvođenja odabranih metoda u C++ programskom jeziku s OpenCV 2.2 bibliotekama [22]. Oba testa su izvedena na osobnom računalu s Windows 7 x64 operacijskim sustavom, Intel Q6600 procesorom, 4 GB RAM memorije i NVIDIA GTX 670 grafičkom karticom s 2 GB GDDR5 memorije (CUDA 3.0 GPU).

Algoritam 4.7: Procedura izvođenja CUDA PBSCD algoritma.

-
1. *Alociraj GPU globalnu memoriju za $I(x, y)$ i $I_B(x, y)$.*
 2. *Izračun ID_B i ID_T za $CUDA_L$, $CUDA_{De}$, $CUDA_G$ i $CUDA_D$ kernele.*
 3. *Kopiraj $I(x, y)$ podatke u globalnu GPU memoriju.*
 4. *Postavi $I_B(x, y) = 1$, $x = 0, 1, \dots, K - 1$, $y = 0, 1, \dots, M - 1$ u globalnoj memoriji GPU.*
 5. *Izvrši $CUDA_L$, $CUDA_{De}$, $CUDA_G$ i $CUDA_D$ kernele.*
 6. *Sinkroniziraj GPU.*
 7. *Kopiraj $I_B(x, y)$ podatke s GPU globalne memorije u RAM od CPU-a.*
 8. *Oslobodi GPU globalnu memoriju.*
-

4.9.1. Pogreška binarizacije

Prvi test uključuje mjerenje pogreške binarizacije na četiri primjera sive slike sirove pločice. Primjer *S1* predstavlja sintetičku sliku generiranu s GIMP programom za obradu digitalnih slika, v. sliku 4.24 (a). Primjer *S2* predstavlja sivu sliku sirove pločice s teksturiranom površinom prikupljenu u laboratoriju s prototipom sustava za detekciju neispravnosti na sirovim pločicama, v. sliku 4.24 (b). Primjeri *S3* i *S4* predstavljaju sive slike sirove pločice prikupljene u industrijskim uvjetima na proizvodnoj liniji, v. slike 4.24 (c) i (d). Primjer *S3* je predstavlja nejednako osvjetljenu teksturiranu sirovu keramičku pločicu, a slika *S4* predstavlja jednobojnu sirovu pločicu. Za sva 4 primjera kreirane su ispravne binarizirane slike (*engl. ground-truth image*): S_C1 za primjer *S1*, S_C2 za primjer *S2*, S_C3 za primjer *S3* i S_C4 za primjer *S4*, v sliku 4.25 (a) do (d). Ispravne slike je kreirao ekspert s GIMP programom za obradu digitalnih slika. Na slikama 4.25 (a) do (d) bijela boja predstavlja objekt (binarna vrijednost '1'), a crna boja predstavlja pozadinu (binarna vrijednost '0'). Rezultati binarizacije metoda razmatranih u ovom radu uspoređene su s ispravnim



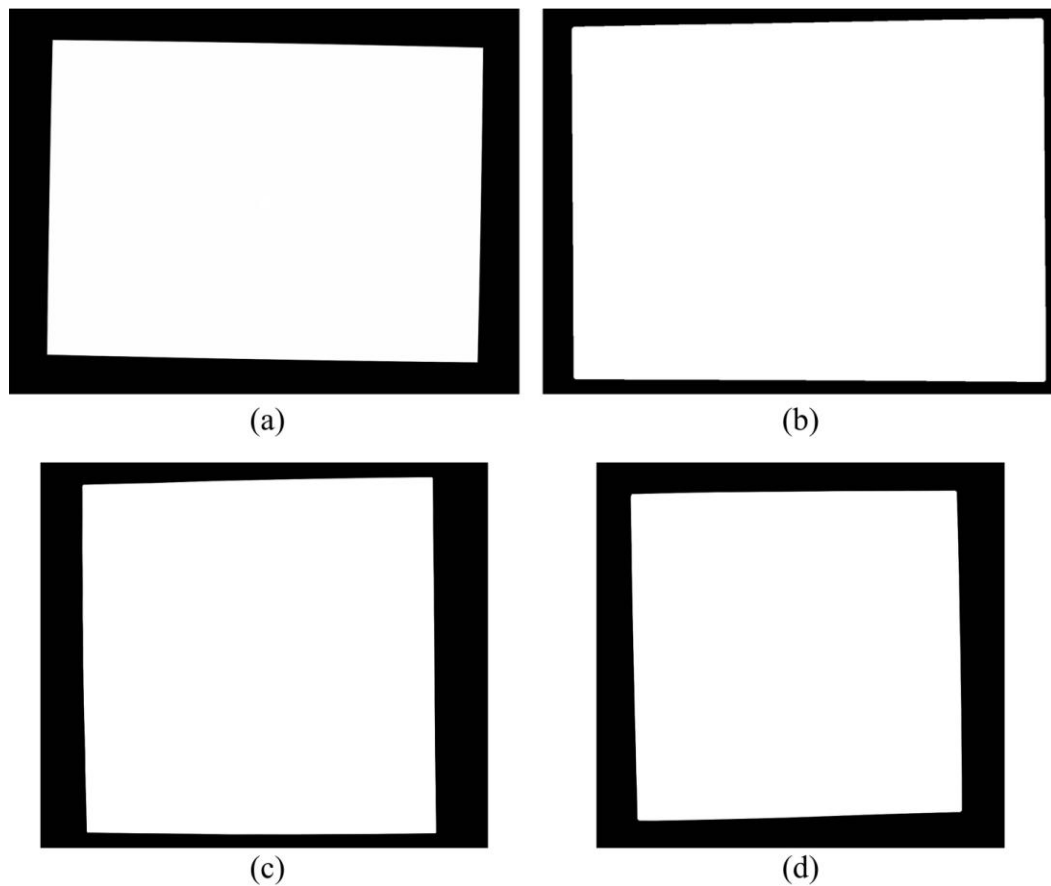
Slika 4.24: Primjeri slika sirovih keramičkih pločica za određivanje pogreške binarizacije: (a) primjer *S1*, (b) primjer *S2*, (c) primjer *S3* i (d) primjer *S4*.

binariziranim slikama. Ispravnost binarizacije kvantitativno je ocjenjena s pogreškom binarizacije ME (engl. *Missclassification Error*) definiranom kao:

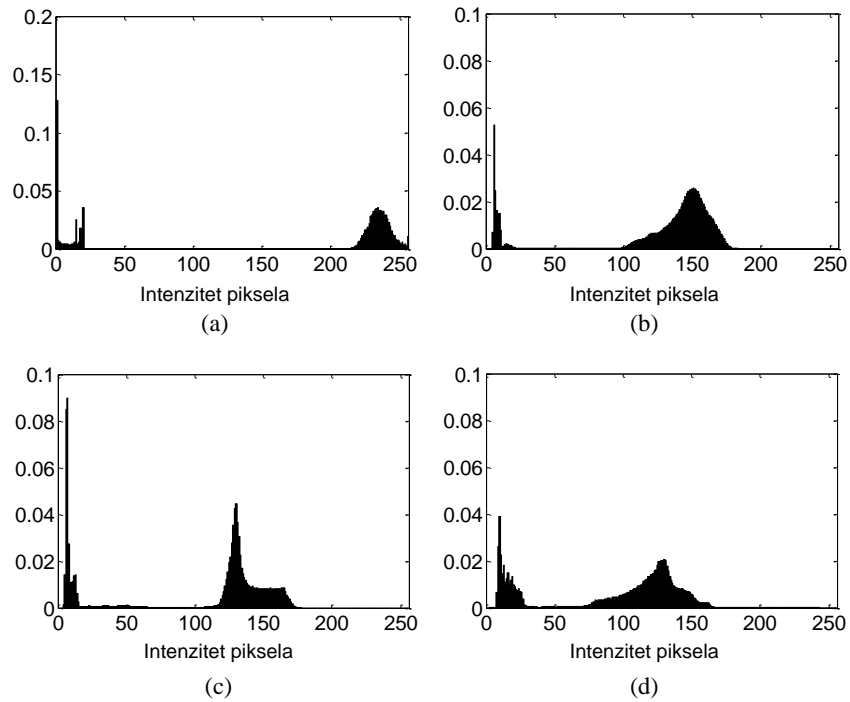
$$ME = 1 - \frac{|B_O \cap B_T| + |F_O \cap F_T|}{|B_O| + |F_O|}, \quad (4.45)$$

gdje B_O i F_O predstavljaju piksele pozadine i piksele objekta ispravne binarizirane slike, a B_T i F_T predstavljaju piksele pozadine i piksele objekta testirane binarizirane slike [72]. Što je ME veći to je više piksela pozadine testirane slike pogrešno segmentirano kao objekt, tj. više piksela objekta testirane slike pogrešno segmentirano kao pozadina. Normalizirani histogrami svih primjera slika (S_1 , S_2 , S_3 , i S_4) prikazani su na slici 4.26. Sa slike 4.26 vidljivo je da svi normalizirani histogrami pokazuju bimodalnu karakteristiku iz čega se može zaključiti da su primjeri slika pogodni za binarizaciju metodama objašnjenim u poglavljima 4.4, 4.5 i 4.6.

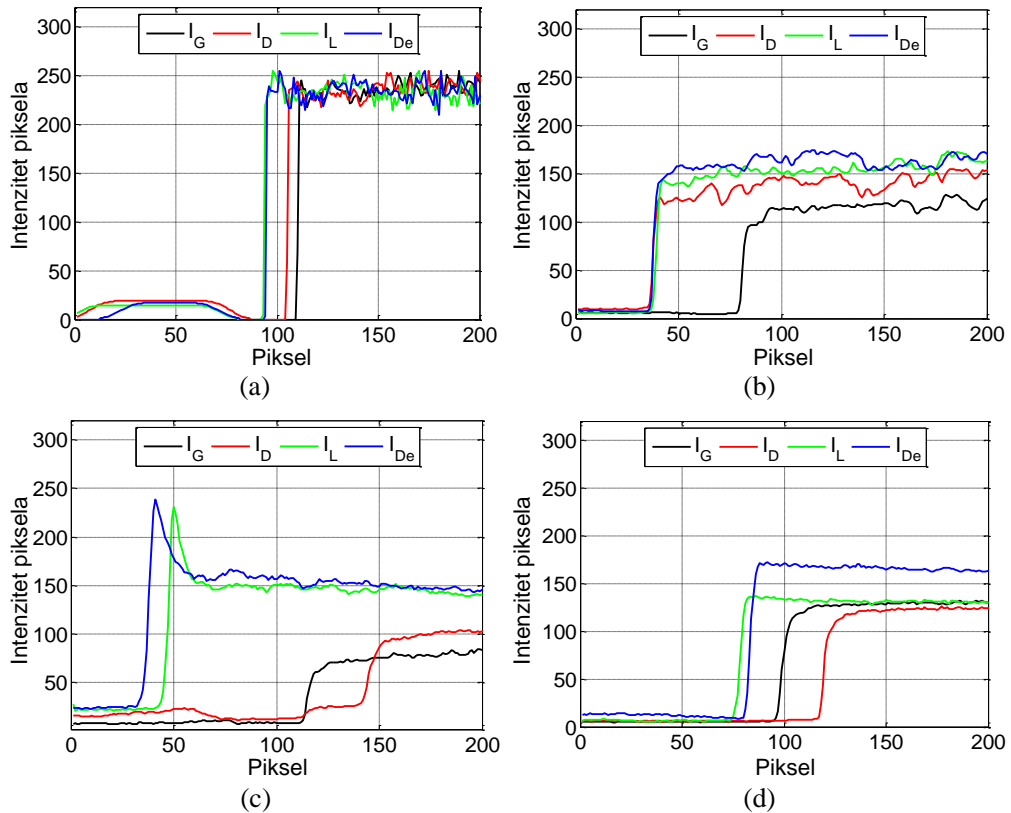
Slika 4.27 prikazuje promjenu intenziteta za $N = 200$ piksela za središnji stupac/redak I_G , I_D , I_L i I_{De} dijela slike. Za sva četiri testirana primjera na slikama 4.27 (a) do (d) vidljiv je



Slika 4.25: Ispravne binarizirane slike sirovih keramičkih pločica: (a) slika S_C1 , (b) slika S_C2 , (c) slika S_C3 i (d) slika S_C4 .



Slika 4.26: Normalizirani histogrami slika sirovih keramičkih pločica $S1$ (a), $S2$ (b), $S3$ (c) i (d) $S4$.



Slika 4.27: Intenziteti 200 piksela središnjeg stupca I_G i I_D dijela slike i središnjeg retka I_L i I_{De} dijela slike za primjere $S1$ (a), $S2$ (b), $S3$ (c) i (d) $S4$.

Tablica 4.6: Minimalne i maksimalne vrijednosti $\Delta A/\sigma$ za primjere S1, S2, S3 i S4.

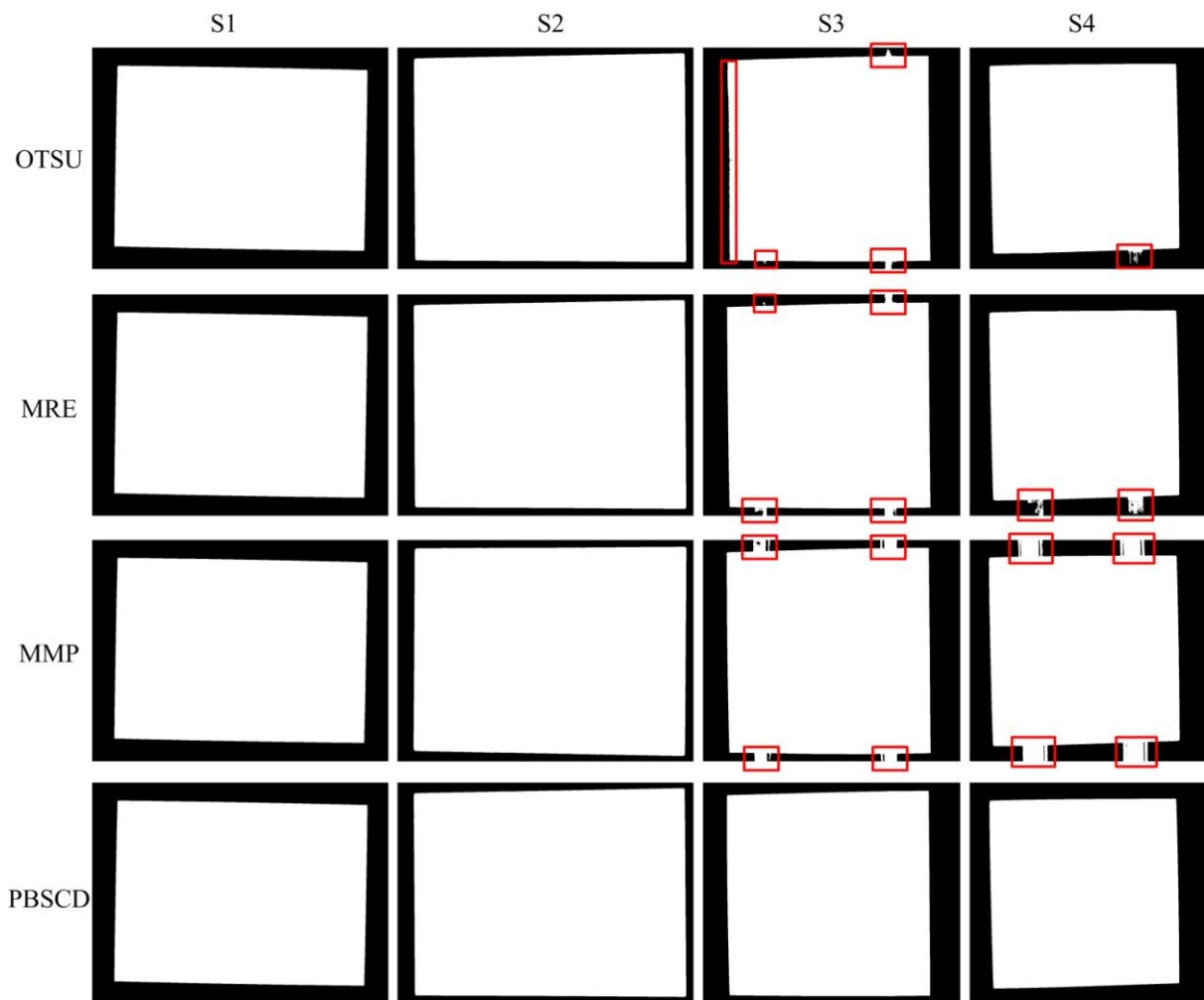
| | <i>Primjer S1</i> | <i>Primjer S2</i> | <i>Primjer S3</i> | <i>Primjer S4</i> |
|--|-------------------|-------------------|-------------------|-------------------|
| <i>Minimum $\Delta A/\sigma$</i> | 11.74 | 11.42 | 7.69 | 8.23 |
| <i>Maksimum $\Delta A/\sigma$</i> | 126.11 | 51.38 | 54.75 | 115.00 |

Tablica 4.7: Razine praga T razmatranih metoda za primjere S1, S2, S3 i S4.

| | <i>Primjer S1</i> | <i>Primjer S2</i> | <i>Primjer S3</i> | <i>Primjer S4</i> |
|---|-------------------|-------------------|-------------------|-------------------|
| <i>Otsu metoda</i> | 122 | 76 | 69 | 73 |
| <i>Metoda minimalne razlike entropija</i> | 76 | 60 | 57 | 56 |
| <i>Metoda minimalne pogreške</i> | 148 | 31 | 36 | 19 |

skok u intenzitetu piksela na mjestu prijelaza pozadina – površina sirove pločice. U Tablici 4.6 prikazane su minimalne i maksimalne vrijednosti $\Delta A/\sigma$ za sva četiri primjera slika. $\Delta A/\sigma$ određen je samo za stupce I_G i I_D dijela slike i retke I_L i I_{De} dijela slike koji uključuju piksele objekta i piksele pozadine, ostali redci i stupci su zanemareni. Iz tablice je vidljivo da je za sve primjere slika $\Delta A/\sigma \geq 6$ tj. prema poglavlju 4.7.1 devijacija indeksa SCD metode $\hat{n}_t = 0\%$. Iz navedenog se može zaključiti da su primjeri slika S1, S2, S3 i S4 pogodni za binarizaciju PBSCD metodom objašnjenom u poglavlju 0.

Na slici 4.28 prikazani su rezultati binarizacije primjera slika S1, S2, S3 i S4 svih metoda razmatranih u ovom radu. 8-povezana komponenta površine sirove pločice predstavljena je s '1' tj. bijelom bojom, a sve ostale komponente predstavljaju pozadinu, '0' tj. crna boja (pojam 8-povezana komponenta objašnjen je u poglavlju 5.2). U tablici 4.7 prikazane su vrijednosti razine praga T dobivene s razmatranim metodama. Sve vrijednosti razine praga, osim razine praga za primjer S4 metode minimalne pogreške, se nalaze u dolini histograma dijeleći histogram na dvije regije. Lijeva regija predstavlja piksele pozadine, a desna regija predstavlja piksele površine sirove pločice. Vidljivo je iz tablice 4.7 i slike 4.26 (d) da razina praga za primjer S4 metode minimalne pogreške se ne nalazi u dolini normaliziranog histograma i za očekivati je loše rezultate binarizacije. Uspoređujući ispravno binarizirane slike, slika 4.25 (a) i (b), s rezultatima binarizacije prikazanim u prvom i drugom stupcu slike 4.28 vidljivo je da sve četiri metode daju dobre rezultate binarizacije za primjere S1 i S2. Za primjere slika prikupljenih u industrijskim uvjetima (primjer S3 i S4) vidljivo je da metode binarizacije kojima je ulazni podatak histogram slike (Otsu metoda, metoda minimalne razlike entropija i metoda minimalne pogreške) ne daju dobre rezultate binarizacije. Navedene metode nisu uspjele uspješno segmentirati površinu sirove pločice od



Slika 4.28: Rezultati binarizacije za primjere $S1$ (prvi stupac), $S2$ (drugi stupac), $S3$ (treći stupac) i $S4$ (četvrti stupac) sa: Otsu metodom (prvi red), metodom minimalne razlike entropije – MRE (drugi red), metodom minimalne pogreške – MMP (treći red) i PBSCD metodom (četvrti red). Crveni pravokutnici označavaju pogrešno binarizirane piksele.

remjenja pokretne trake. Kako je i objašnjeno na početku poglavlja 4.7 nakon nekog vremena materijal prenesen na remenje prekrije boju remena i u određenim slučajevima nestane razlika u intenzitetima piksela remenja i površine sirove pločice. Na slici 4.28 u stupcima $S3$, $S4$ i redcima Otsu, MRE i MMP crvenim pravokutnicima označeno je područje pogrešne binarizacije. Najviše pogrešno segmentiranih piksela se nalazi u području remenja pokretne trake, tj. pikseli remenja su pogrešno segmentirani kao površina pločice.

Za sva četiri primjera PBSCD metoda uspješno segmentira površinu pločice od pozadine. Na slici 4.28 u stupcima $S3$, $S4$ i retku PBSCD u područjima remenja nema vidljivih pogrešaka u binarizaciji, tj. površina sirove pločice uspješno je segmentirana u odnosu na pozadinu.

Tablica 4.8: Pogreška binarizacije ME u postocima za primjere $S1$, $S2$, $S3$ i $S4$ razmatranih metoda (MRE – metoda minimalne razlika entropija, MMP – metoda minimalne pogreške). Ocjena '1' predstavlja najbolji ME rezultat, ocjena '4' predstavlja najgori ME rezultat.

| | <i>Otsu metoda</i> | | <i>MRE</i> | | <i>MMP</i> | | <i>PBSCD metoda</i> | |
|----------------------|--------------------|---------------|---------------|---------------|---------------|---------------|---------------------|---------------|
| | <i>ME [%]</i> | <i>Ocjena</i> | <i>ME [%]</i> | <i>Ocjena</i> | <i>ME [%]</i> | <i>Ocjena</i> | <i>ME [%]</i> | <i>Ocjena</i> |
| <i>Primjer S1</i> | 0.09 | 1 | 0.12 | 2 | 0.09 | 1 | 0.09 | 1 |
| <i>Primjer S2</i> | 0.17 | 3 | 0.11 | 2 | 0.22 | 4 | 0.10 | 1 |
| <i>Primjer S3</i> | 0.77 | 3 | 0.65 | 2 | 1.30 | 4 | 0.09 | 1 |
| <i>Primjer S4</i> | 0.29 | 2 | 0.75 | 3 | 3.59 | 4 | 0.09 | 1 |
| <i>Ukupno</i> | 1.32 | 2 | 1.63 | 3 | 5.20 | 4 | 0.37 | 1 |

U tablici 4.8 prikazani su ME rezultati za sva četiri primjera i sve četiri razmatrane metode. Svaki ME rezultat je ocijenjen s '1' za najbolji ME rezultat i '4' za najlošiji ME rezultat. Za sve metode je dana konačna ocjena određena na temelju ukupnog ME za sve primjere slika. Iz tablice 4.8 je vidljivo da za primjere $S1$ i $S2$ sve metode daju približno jednake ME vrijednosti. Isto ne vrijedi za primjere $S3$ i $S4$ gdje PBSCD metoda pokazuje najbolje ME rezultate. Također ME za sva četiri primjera za PBSCD metodu je približno konstantan ($ME \approx 0.09\%$). PBSCD metoda daje dobre rezultate binarizacije neovisno o vrsti pločice (teksturirana ili jednobojna) i neovisno o uvjetima prikupljanja slike (neravnomjerna osvjetljenost slike, laboratorijski uvjeti, industrijski uvjeti). Također, metode su testirane na 200 različitih slika sirovih pločica s vizualnom ocjenom rezultata binarizacije bez određivanja ME . Jedino je PBSCD metoda uspješno binarizirala sve testirane slike. Iz slike 4.28 i tablice 4.8 proizlazi sljedeći zaključak: PBSCD metoda daje najbolje rezultate binarizacije za sintetički generiranu sliku (primjer $S1$), sliku sirove pločice prikupljenu u laboratoriju (primjer $S2$) i slike sirovih pločica prikupljenih u industrijskim uvjetima (primjer $S3$ i $S4$), što nije slučaj za ostale razmatrane metode.

4.9.2. Rad u stvarnom vremenu

Drugi test obuhvaća mjerenje vremena izvođenja razmatranih metoda. Prosječno vrijeme izvođenja sekvencijalnog algoritma sve četiri metode mjereno je u MATLAB 2009b okruženju. PBSCD metoda je implementirana djelomično s MATLAB funkcijama, a djelomično s MATLAB/MEX funkcijama, ostale metode implementirane su s MATLAB funkcijama. MATLAB/MEX funkcije predstavljaju implementacije algoritama u C++ ili Fortran programskom jeziku radi smanjenja vremena izvođenja implementiranih algoritama [73]. Tablica 4.9 prikazuje prosječno vrijeme izvršenja razmatranih metoda u MATLAB

Tablica 4.9: Prosječno vrijeme izvođenja (100 iteracija) u *ms* razmatranih metoda za sliku veličine 1038×1098 piksela (za PBSCD metodu $N = 120$) u MATLAB okruženju. Ocjena '1' predstavlja najkraće prosječno vrijeme izvođenja, ocjena '4' predstavlja najduže prosječno vrijeme izvođenja.

| | <i>Otsu metoda</i> | <i>MRE</i> | <i>MMP</i> | <i>PBSCD metoda</i> |
|--|--------------------|------------|------------|---------------------|
| <i>Prosječno vrijeme izvođenja</i> | 5.89 | 19.78 | 23.71 | 261.10 |
| <i>Omjer vremena izvođenja OI_m</i> | 1 | 3.34 | 4.03 | 44.33 |
| <i>Ocjena</i> | 1 | 2 | 3 | 4 |

Tablica 4.10: Prosječno vrijeme izvođenja (100 iteracija) u *ms* Otsu i PBSCD metode za sliku veličine 1038×1098 piksela u Visual Studio 2010 okruženju i C++ programskom jeziku.

| | <i>Otsu metoda</i> | <i>PBSCD metoda</i> $N = 60$ | <i>PBSCD metoda</i> $N = 120$ | <i>PBSCD metoda</i> $N = 250$ |
|--|--------------------|---------------------------------|----------------------------------|----------------------------------|
| <i>Prosječno vrijeme izvođenja</i> | 2.01 | 48.82 | 131.19 | 455.80 |
| <i>Omjer vremena izvođenja OI_m</i> | 1 | 24.29 | 65.27 | 226.77 |

okruženju. Vrijeme izvođenja metoda je ocijenjeno s '1' za najkraće vrijeme izvođenja i '4' za najduže vrijeme izvođenja. Iz tablice 4.9 je vidljivo da je vrijeme izvođenja Otsu metode najkraće, a vrijeme izvođenja PBSCD metode najduže bez obzira što je PBSCD metoda implementirana s MATLAB/MEX funkcijom. Dobivena prosječna vremena izvođenja su očekivana s obzirom na broj operacija prikazanih u tablicama 4.3 i 4.4. Omjer vremena izvođenja OI_m određen je s (4.46) gdje t_{\ll} predstavlja najkraće vrijeme izvođenja od sve četiri razmatrane metode (za slučaj tablice 4.9 $t_{\ll} = 5.89$ ms), a t_m predstavlja vrijeme izvođenja m metode, $m \in \{Otsu, MRE, MMP, PBSCD\}$. PBSCD metoda je 44 puta sporija u odnosu na Otsu metodu za $N = 120$. S obzirom da PBSCD metoda prikazuje najbolje rezultate binarizacije (ocjenjena s 1), a vrijeme izvođenja Otsu metode je najkraće (ocjenjeno s 1) u nastavku drugog testa navedene metode su implementirane u C++ programskom jeziku s OpenCV2.2 bibliotekama.

$$OI_m = \frac{t_m}{t_{\ll}} \quad (4.46)$$

Za C++ implementaciju metoda korišteno je Visual Studio 2010 okruženje [74]. Upotrijebljena je OpenCV2.2 implementacije Otsu metode i opcije C++ prevoditelja:

- potpuna optimizacija po brzini izvođenja,
- brzi model broja jednostruke preciznosti.

PBSCD metoda je implementirana s istim opcijama C++ prevoditelja. Rezultati testiranja prikazani su u tablici 4.10. Iz tablica 4.9 i 4.10 može se zaključiti da je omjer vremena

Tablica 4.11: Prosječno vrijeme izvođenja (100 iteracija) u *ms* CUDA PBSCD algoritma za sliku veličine 1098×1038 piksela u Visual Studio 2010 okruženju i CUDA C/C++ programskom jeziku.

| | PBSCD metoda $N = 60$ | PBSCD metoda $N = 120$ | PBSCD metoda $N = 250$ |
|---|--------------------------|---------------------------|---------------------------|
| Prosječno vrijeme izvođenja kernela | 0.68 | 1.20 | 3.38 |
| Prosječno vrijeme izvođenja memorijskih transakcija | 2.49 | 2.49 | 2.49 |
| Ukupno prosječno vrijeme | 3.17 | 3.69 | 5.87 |
| Omjer vremena izvođenja $OI_m, t_{\ll} = 2.01 \text{ ms}$ | 1.58 | 1.84 | 2.92 |

izvođenja metoda u C++ programskom jeziku u donosu na MATLAB okruženje približno dva puta. Omjer izvođenja $OI_{PBSCD,120}$ se povećalo s 44.33 na 65.27. Iako je vrijeme izvođenja PBSCD metode smanjeno s 261.10 *ms* na 131.19 *ms* još uvijek je vrijeme izvođenja predugo tj. 21 % od ukupnog vremena dostupnog za vizualnu inspekciju sirovih pločica (prema poglavlju 3.3).

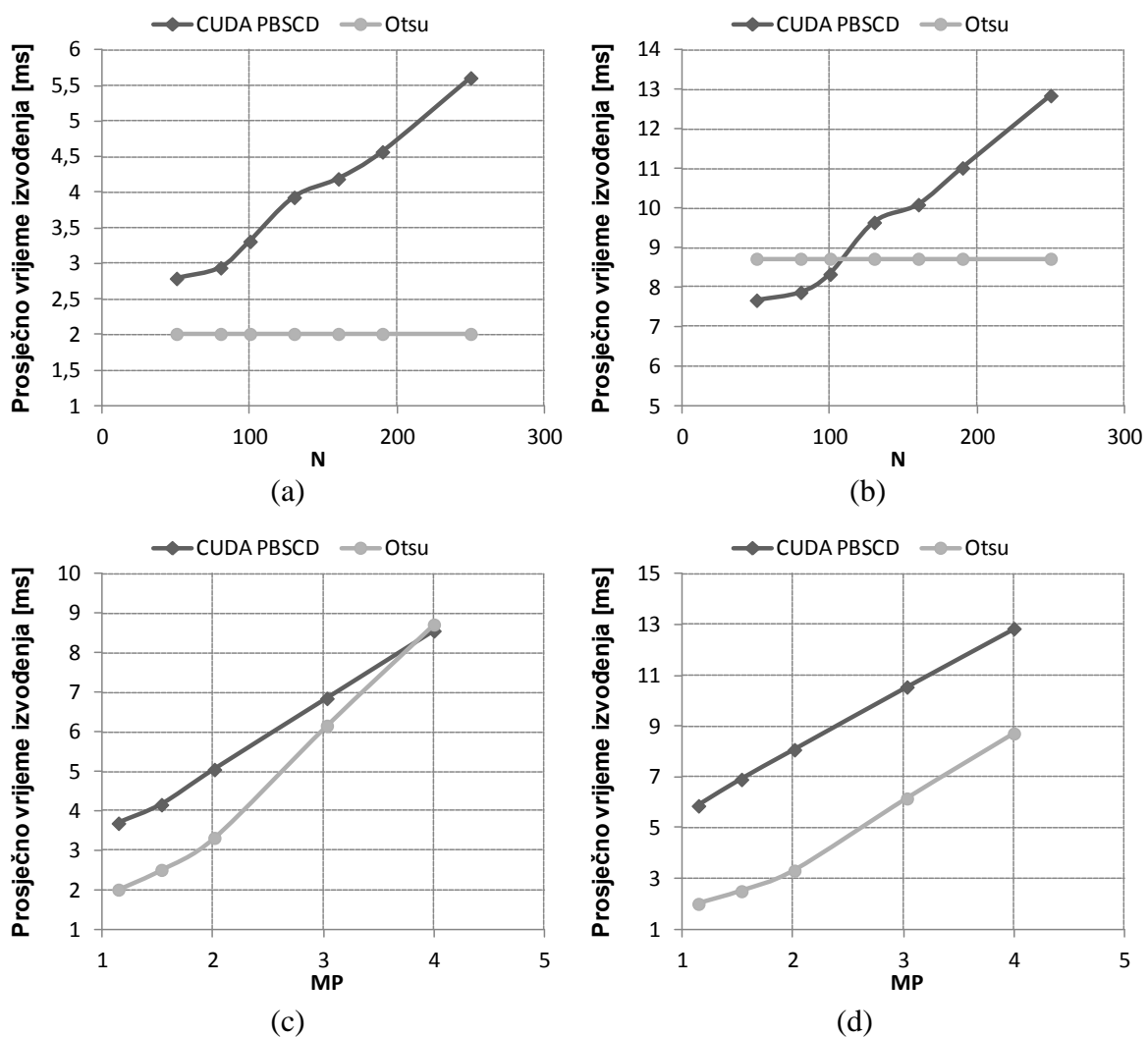
Upotrebom CUDA paralelne platforme i CUDA PBSCD algoritma dobivena su mjerenja izvođenja prikazana u tablici 4.11. Vrijeme izvođenja je podijeljeno na dva dijela:

1. prosječno vrijeme izvođenja kernela – obuhvaća vrijeme potrebno za izvođenje $CUDA_G$, $CUDA_D$, $CUDA_L$ i $CUDA_{De}$ kernela (linija 5 i 6 algoritma 4.7),
2. prosječno vrijeme izvođenja memorijskih transakcija – obuhvaća vrijeme potrebno za kopiranje ulazni i izlaznih podataka prema poglavlju 4.8.4 (linija 3, 4 i 7 algoritma 4.7).

Vrijeme izvođenja memorijskih transakcija je konstantno budući da se veličina ulaznih i izlaznih podataka s N ne mijenja. Prosječno vrijeme izvođenja kernela raste s povećanjem broja uzoraka N . Uspoređivanjem rezultata tablice 4.10 i tablice 4.11 vidljivo je da su se vremena izvođenja CUDA PBSCD algoritma približila vremenu izvođenja Otsu metode. $OI_{PBSCD,120}$ se smanjilo s 227 na manje od 3, tj. za $N = 120$ ukupno prosječno vrijeme izvođenja CUDA PBSCD metode je manje od 0.7 % vremena dostupnog za vizualnu inspekciju sirovih pločica (600 *ms*).

Vrijeme izvođenja PBSCD metode ovisi o broju uzoraka N i o veličini ulazne slike. Na slici 4.29 (a) i (b) prikazana je ovisnost prosječnog vremena izvođenja za promjenjivi broj uzoraka N i dvije veličine slike (1038×1098 i 1946×2054 piksela). Vrijeme izvođenja Otsu metode ne ovisi o N kako je i vidljivo na slikama 4.29 (a) i (b). Vrijeme izvođenja CUDA PBSCD metode linearno ovisi o broju uzoraka N i ne prelazi 13 *ms* za sliku veličine 1946×2054 piksela (približno 4 *MP*) tj. vrijeme izvođenja je manje od 3 % ukupno

dostupnog vremena za vizualnu inspekciju sirovih pločica. Na slici 4.29 (c) i (d) prikazana je ovisnost prosječnog vremena izvođenja CUDA PBSCD i Otsu metode za promjenjivu veličinu slike i $N \in \{120, 250\}$. Vrijeme izvođenja CUDA PBSCD i Otsu metode linearno ovisi o veličini ulazne slike za navedeni interval. Na temelju slike 4.29 (c) može se zaključiti da za $N = 120$ vrijeme izvođenja CUDA PBSCD metode je približno jednako vremenu izvođenja Otsu metode (za sliku veličine 4 MP vrijeme izvođenja CUDA PBSCD metode je manje od vremena izvođenja Otsu metode). Na slici 4.29 (d) prikazan je slučaj za $N = 250$ (najlošiji slučaj u pogledu vremena izvođenja CUDA PBSCD metode) gdje je vidljivo da vrijeme izvođenja ne prelazi 13 ms, tj. vrijeme izvođenja je manje od 3 % ukupno dostupnog



Slika 4.29: Prosječno vrijeme izvođenja u 100 iteracija CUDA PBSCD i Otsu metode: (a) slika veličine 1038×1098 piksela i promjenjivi N , (b) slika veličine 1946×2054 piksela i promjenjivi N , (c) $N = 120$ i promjenjiva veličina slike izražena u megapikselima (MP), (d) $N = 250$ i promjenjiva veličina slike u MP.

vremena za vizualnu inspekciju sirovih pločica. Također, moguće je dodatno smanjiti vrijeme izvođenja CUDA PBSCD metode s ograničavanjem broja uzoraka na $N \leq 120$. Navedeno ograničenje se jednostavno postiže s ograničavanjem veličine slike tako da ni na jednom dijelu slike pikseli objekta nisu udaljeni više od $N - 10$ piksela od ruba slike.

4.9.3. Zaključak

Iz rezultata dobivenih iz provedena dva testa proizlaze sljedeći zaključci.

- Nova PBSCD metoda predložena u ovom radu testirana je na sintetičkim slikama i slikama prikupljenih u laboratorijskim i industrijskim uvjetima. Metoda je uspoređena s metodama za binarizaciju slike temeljene na globalnom određivanju razine praga (Otsu metoda, metoda minimalne pogreške i metoda minimalne razlike entropija). Za provedene testove određena je pogreška binarizacije za sve metode. Predložena PBSCD metoda najbolje segmentira sliku sirove pločice, tj. pokazuje najbolje rezultate binarizacije na svim primjerima slika (sintetički generirana slika, slika prikupljene u laboratoriju i slike prikupljene u industrijskim uvjetima) u odnosu na ostale razmatrane metode. Isti rezultati su dobiveni i vizualnom ocjenom binarizacije za 200 testiranih slika.
- Testiranje mogućnosti rada u stvarnom vremenu izvedeno je sa sekvencijalnim i paralelnim algoritmom predložene PBSCD metode. Vremena izvođenja paralelne implementacije PBSCD metode na CUDA platformi približno su jednaka (a u nekim slučajevima i manja) vremenima izvođenja Otsu metode (metoda s najmanjim brojem operacija). Vrijeme izvođenja CUDA PBSCD algoritma u najlošijem slučaju ne prelazi 13 ms tj. vrijeme izvođenja je manje od 3 % ukupno dostupnog vremena za vizualnu inspekciju sirovih pločica. Može se zaključiti da vrijeme izvođenja CUDA PBSCD metode zadovoljava postavljene vremenske uvjete i da je metoda pogodna za binarizaciju slike sirove pločice u stvarnom vremenu.

5. UNAPRIJEĐENI ALGORITAM ZA DETEKCIJU NEISPRAVNOSTI NA RUBOVIMA I UGLOVIMA SIROVE PLOČICE

Nakon uspješne binarizacije slike sirove keramičke pločice potrebno je provjeriti da li postoje neispravnosti na rubovima i uglovima sirove keramičke pločice. Ako je u postupku detekcije pronađena neispravnost na rubu/uglu sirove pločice, pločica se klasificira kao škart i prosljeđuje nazad u fazu miješanja. Na ovaj način smanjuje se količina energije koja bi se trošila na pečenje neispravne sirove pločice i naknadno mljevenje neispravne keramičke pločice. Također, ako je pronađena neispravnost na rubu/uglu sirove pločice nema potrebe za dodatnom analizom neispravnosti površine sirove pločice.

Postupak detekcije neispravnosti na rubovima i uglovima sirove keramičke pločice podijeljen je na dva dijela:

1. detekcija neispravnosti na rubovima sirove pločice i
2. detekcija neispravnosti na uglovima sirove pločice.

Ako u prvom dijelu nije detektirana niti jedna neispravnost na rubu pločice, prelazi se na drugi dio postupka detekcije. Ako ni u drugom dijelu nije otkrivena neispravnost na uglovima pločice prelazi se na postupak ujednačavanja slike površine sirove keramičke pločice i detekciju površinskih neispravnosti. Dijelovi su međusobno povezani na način da se podaci za detekciju neispravnosti na rubovima sirove pločice upotrebljavaju i za detekciju neispravnosti na uglovima sirove keramičke pločice.

5.1. Postojeće metode za detekciju neispravnosti na rubovima i uglovima sirove keramičke pločice

U literaturi se može pronaći nekoliko metoda za detekciju neispravnosti na rubovima keramičke pločice, detekciju neispravnosti na uglovima keramičke pločice i istovremeno detekciju neispravnosti na rubovima i uglovima keramičke pločice. U [9] autori predlažu sustav za inspekciju i klasifikaciju keramičkih pločice. Dio sustava predstavljaju metode za detekciju neispravnosti na rubovima i uglovima keramičkih pločica. Metodu su podijelili na dva dijela. Prvi dio obuhvaća metode za ujednačavanje osvijetljenosti ruba i uglova, nakon čega je povećana slika površine pločice. Na ovaj način slika površine pločice obuhvaća i rubove i uglove pločice. Drugi dio se sastoji od analize površine pločice na povećanoj slici koja obuhvaća i rubove i uglove. Detaljni opis metode i algoritam metoda nije predstavljen u [9], a djelomični rad u stvarnom vremenu je omogućen s posebno razvijenim

višeprocesorskim DSP (*engl. Digital Signal Processor*) sustavom. Autori su u [75, 40] predložili sustav za detekciju neispravnosti na uglovima sirovih pločica. Sustav se temelji na praćenju konture slike ugla sirove pločice i usporedbom duljine s rezultatom dobivenim Hough-ovom transformacijom [76]. Pomoću klasifikatora određeno je kojoj klasi pločica pripada. Ukupno vrijeme potrebno za analizu uglova je 356 ms tj. više od 50 % vremena dostupnog za vizualnu inspekciju sirove pločice. Iako autori smatraju da je vrijeme zadovoljavajuće, nisu uzeli u obzir potrebu za analizom rubova sirove pločice, ujednačavanje slike površine sirove pločice i detekciju neispravnosti na površini sirove pločice. Autori u [41] predložili su metodu za detekciju neispravnosti na rubovima i uglovima keramičke pločice na temelju praćenja konture keramičke pločice. Metoda nakon binarizacije slike (metodom za određivanje razine praga slike keramičke pločice usporedbom histograma dijela slike koji prikazuje površinu keramičke pločice i histograma dijela slike koji prikazuje pozadinu) primjenjuje Canny detektor ruba za određivanje rubova keramičke pločice. Na temelju točaka ruba i definiranog susjedstva točaka ruba prati se kontura ruba tj. određuje se smjer konture keramičke pločice na temelju relativnog pomaka trenutne točke ruba u odnosu na prethodnu točku ruba. Metoda određuje razliku trenutnog smjera konture i referentnog smjera konture (određenog na temelju slike keramičke pločice bez neispravnosti na rubovima i uglovima) uz pretpostavku da je rotacija pločice na industrijskoj liniji zanemariva. Bipolarni skokovi u razlici smjera konture predstavljaju neispravnosti. Autori ne predlažu metodu za detekciju bipolarnih skokova u razlici smjera. Također nije predloženo rješenje za slučaj kada kontura keramičke pločice sadrži prekide (u susjedstvu trenutne točke ruba ne postoji sljedeća točka ruba) ili kada u susjedstvu trenutne točke ruba postoji više od jedne točke ruba tzv. račvanje. Autori također nisu predložili optimalni algoritam metode koji bi omogućio rad u stvarnom vremenu (vrijeme izvođenja metode u MATLAB okruženju za 2.18 MP sliku prelazi 4.5 s). U [77] autori predlažu metodu za detekciju neispravnosti na rubovima keramičkih pločica temeljenu na praćenju konture i Radonovoj transformaciji [78]. Metoda binarizira sliku s neizrazitom (*engl. fuzzy*) binarizacijom [79], nakon čega primjenjuje Radonovu transformaciju i morfološko filtriranje radi uklanjanja učinka rotacije keramičke pločice na industrijskoj liniji. Autori zatim primjenjuju metodu sličnu metodi opisanoj u [41] temeljenoj na praćenju konture za detekciju neispravnosti na rubovima i uglovima. U [77] nije predložena metoda za određivanje kuta rotacije niti je predložena metoda za detekciju ruba. Način detekcije neispravnosti na temelju smjera konture nije detaljno opisan i mogućnost rada metode u stvarnom vremenu nije prikazan.

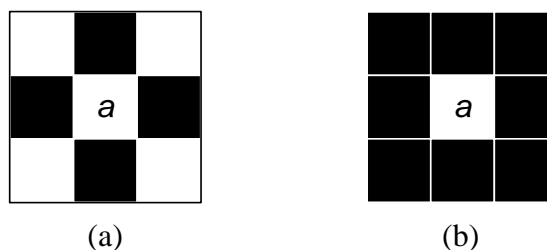
U ovom radu predložena je unaprijeđena metoda za detekciju neispravnosti na rubovima i uglovima sirove keramičke pločice. Metoda se temelji na praćenju konture iz [41] ali rješava sve navedene nedostatke. Metoda je testirana u laboratoriju i industrijskim uvjetima na mogućnost detekcije neispravnosti na rubovima i uglovima i testirana je na mogućnost rada u stvarnom vremenu. Predložena metoda se sastoji on nekoliko koraka:

1. detekcija ruba na binarnoj slici,
2. praćenje konture i određivanje smjera konture,
3. filtriranje smjera konture,
4. izračun razlike smjera konture,
5. detekcija i lokalizacija neispravnosti na uglovima i rubovima sirove pločice.

Detaljno objašnjenje postupka detekcije neispravnosti na rubovima i uglovima sirove keramičke pločice slijedi u narednim potpoglavljima.

5.2. Detekcija ruba sirove keramičke pločice

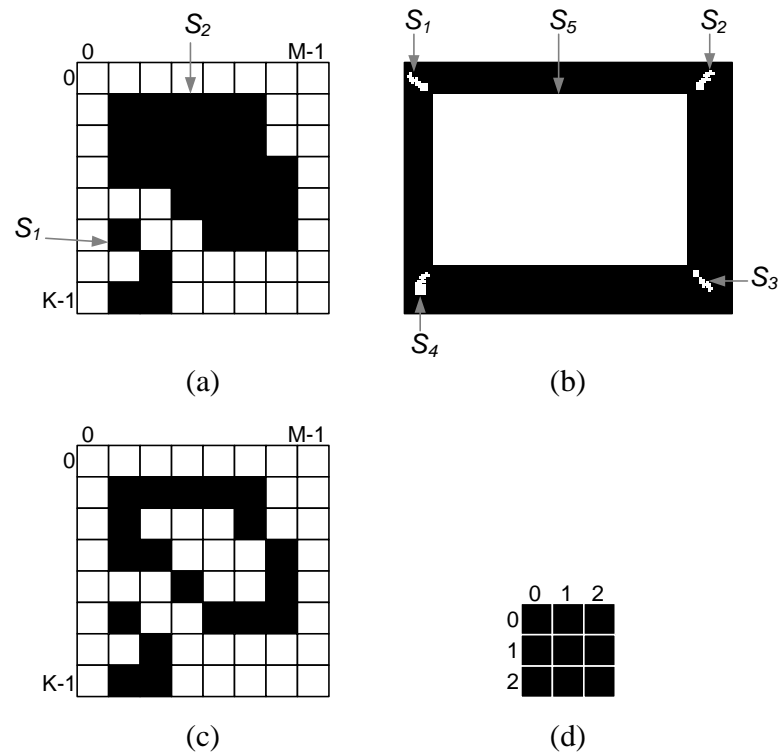
Nakon binarizacije sive slike sirove pločice, na dobivenoj binarnoj slici $I_B(x, y)$ potrebno je detektirati točke (piksele) ruba objekta, tj. ruba sirove pločice. Za definiranje ruba objekta potrebno je definirati susjedstvo piksela. Piksela a na koordinatama (x, y) može imati 4 susjeda označeno s $N_4(a)$ ili 8 susjeda označeno s $N_8(a)$ kako je i prikazano na slici 5.1. Na temelju definiranih susjedstava i vrijednosti samog piksela (za slučaj binarne slike piksel može poprimiti vrijednost '1' ili '0') definira se povezana komponenta na slici. Skupina piksela S je 4-povezana komponenta ako pikseli skupine S imaju jednake vrijednosti i međusobno su N_4 susjedi. Skupina piksela S je 8-povezana komponenta ako pikseli skupine S imaju jednake vrijednosti i međusobno su N_8 susjedi [15]. Za potrebe ovog rada rub je definiran na sljedeći način: rub 8-povezane komponente S predstavlja skup piksela komponente S čiji svi N_4 susjedi ne predstavljaju piksele 8-povezane komponente S .



Slika 5.1: Susjedstvo piksela a označeno s crnim kvadratima: (a) 4 susjedna piksela $N_4(a)$ i (b) 8 susjednih piksela $N_8(a)$.

U ovom radu se razmatraju samo 8-povezane komponente na binarnoj slici i u nastavku rada će se označavati s *povezana komponenta*. Na slici 5.2 (a) i (b) prikazan je primjer dvije povezone komponente na sintetički generiranoj binarnoj slici i primjer pet povezanih komponenti za sliku sirove keramičke pločice binariziranu s PBSCD metodom. Na slikama 5.2 (a) i (b) povezana komponenta pozadine S_B nije naznačena. S_B predstavljaju svi pikseli koji ne pripadaju već naglašenim komponentama na slikama, tj. predstavlja povezanu komponentu s vrijednostima piksela '0'. Na slici 5.2 (b) povezana komponenta S_5 predstavlja sirovu pločicu, ostale povezone komponente ($S_1 - S_4$) predstavljaju pogrešno segmentirane piksele kako je i objašnjeno u poglavlju 4.7.2. Na slici 5.2 (c) prikazan je detektirani rub slike 5.2 (a). Za primjer povezone komponente S_2 vidljivo je da pikseli ruba pripadaju povezanoj komponenti S_2 i da najmanje jedan N_4 susjed piksela ruba pripada S_B povezanoj komponenti.

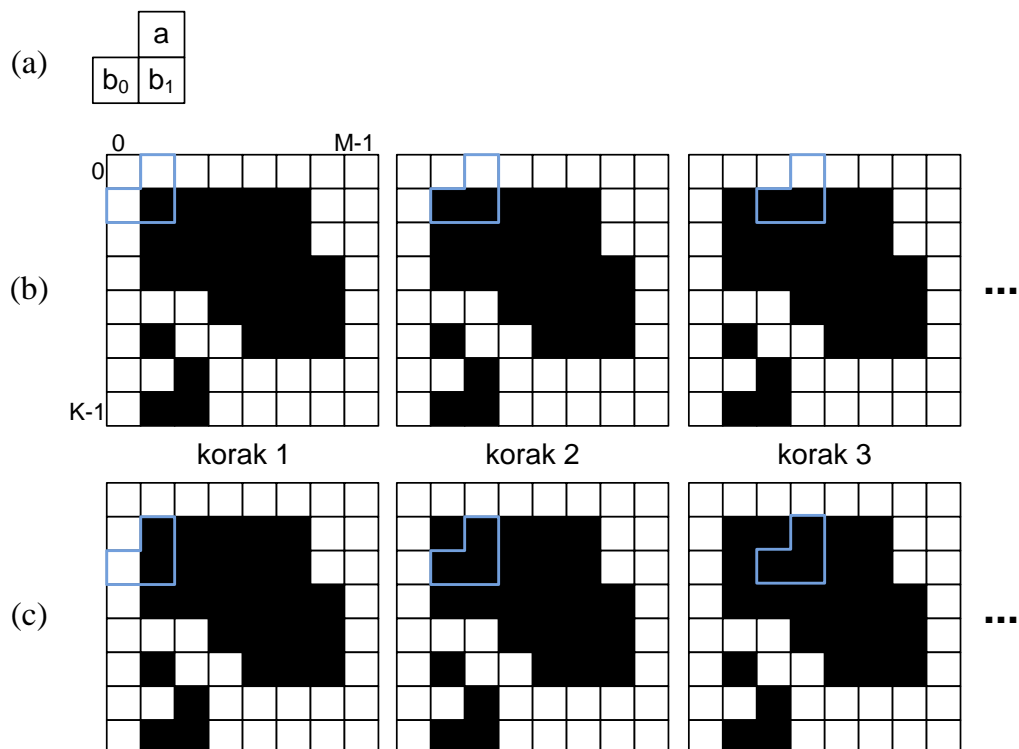
U ovom radu predložena je jednostavna i brza metoda, za razliku od Canny detektora, za detekciju ruba temeljena na samoj definiciji ruba. Predložena metoda za detekciju ruba ne



Slika 5.2: Primjer dvije S_1 i S_2 povezone komponente na sintetički generiranoj binarnoj slici (a), primjer pet povezanih komponenti ($S_1 - S_5$) na binarnoj slici (PBSCD metoda) sirove keramičke pločice, (c) detektirani rub za primjer (a), (d) 3×3 oblik povezone komponente. Na slici (a), (c) i (d) crni pravokutnik označava piksel s intenzitetom '1', a na slici (b) bijela boja predstavlja piksele s intenzitetom '1'.

generira prekid ruba za povezanu komponentu koja sadrži minimalno 3×3 piksela oblika prikazanog na slici 5.2 (d). Za binarnu sliku sirove pločice navedeni uvjet je uvijek zadovoljen, tj. neće nikada postojati prekid ruba za povezanu komponentu sirove pločice (S_5 na slici 5.2 (c)).

Predložena metoda za svaki piksel povezanih komponenti (ne uključujući povezanu komponentu S_B) provjerava N_4 susjedstvo. Ako barem jedan N_4 susjed trenutnog piksela predstavlja piksel S_B povezane komponente, označava se kao piksel (točka) ruba. Metoda pretražuje piksele binarne slike $I_B(x, y)$ s obrnutom L formom kako je i prikazano na slici 5.3 (a) počevši od nultog retka slike s lijeva prema desno, cf. slika 5.3 (b) i (c). Izlaz procedure za detekciju ruba je nova binarna slika rubova $I_R(x, y)$ jednake veličine kao i $I_B(x, y)$ u kojoj '1' predstavlja točku ruba, a '0' ne predstavlja točku ruba. Na početku metode za detekciju ruba vrijednosti svih piksela $I_R(x, y)$ slike postavljene su '0'. Da li je piksel $I_R(x, y)$ piksel ruba, određeno je na temelju piksela obrnute L forme i uvjeta (5.1). U uvjetima (5.1) (x_{b_1}, y_{b_1}) predstavljaju koordinate b_1 piksela, (x_{b_0}, y_{b_0}) predstavljaju koordinate piksela b_0 , a (x_a, y_a)



Slika 5.3: Obrnuta L forma s označenim pikselima (a), (b) prva tri koraka detektiranja točaka ruba za nulti i prvi red slike 5.2 (a), (c) prva tri koraka detektiranja točaka ruba za prvi i drugi red slike 5.2 (a).

Obrnuta L forma je označena s plavom linijom.

$$\begin{aligned}
 I_R(x_{b_1}, y_{b_1}) &= 1, & \text{ako } b_1 > a \text{ ili } b_1 > b_0 \\
 I_R(x_{b_0}, y_{b_0}) &= 1, & \text{ako } b_0 > b_1 \\
 I_R(x_a, y_a) &= 1, & \text{ako } a > b_1
 \end{aligned}
 \tag{5.1}$$

predstavljaju koordinate piksela a . Na temelju obrnute L forme i uvjeta (5.1) provjereni su svi N_4 susjedi svih piksela $I_B(x, y)$ slike. Pseudo kod predložene metode je prikazan u algoritmu 5.1. Na kraju predložene metode točke ruba spremljene su u $I_R(x, y)$ binarnu sliku. Sljedeći korak metode za detekciju neispravnosti na rubovima i uglovima sirove pločice je praćenje konture sirove pločice i određivanje smjera konture.

5.3. Praćenje konture i određivanje smjera konture sirove keramičke pločice

Prema [80] praćenje konture (*engl. contour tracing ili border following*) na razini piksela je definirana kao: određivanje koordinata ili lančanih kodova na temelju ruba između dvije povezane komponente na slici (objekt – pozadina). Lančani kodovi (*engl. chain codes*) predstavljaju način zapisivanja konturnih točaka u memoriju radi smanjivanja količine zauzete memorije [81]. Memorijski zahtjevi za praćenje konture sirove pločice nisu ograničavajući faktor, zbog čega se problematika lančanih kodova ne razmatra u ovom radu. U literaturi se može pronaći nekoliko metoda za praćenje konture na razini piksela na binarnoj slici. Najpoznatije i često korištene metode su:

- kvadratno praćenje (*engl. square tracing, bug-following*) konture [82],
- kružno praćenje (*engl. radial sweep*) konture [83] i

Algoritam 5.1: Određivanje točaka (piksela) ruba predloženom metodom za detekciju ruba.

```

1. Ulaz:  $I_B(x, y), I_R(x, y), M, K$ 
2. postavi sve piksele  $I_R(x, y) = 0$ 
3. for  $i := 0 \rightarrow K - 2$  do
4.     for  $j := 0 \rightarrow M - 2$  do
5.          $a = I_B(i, j + 1)$ ,
6.          $b_0 = I_B(i + 1, j), b_1 = I_B(i + 1, j + 1)$ 
7.         if ( $b_1 > a$  ili  $b_1 > b_0$ ) then
8.              $I_R(i + 1, j + 1) = 1$ 
9.         else if  $b_0 > b_1$  then
10.             $I_R(i + 1, j) = 1$ 
11.        end if
12.        if  $a > b_1$  then
13.             $I_R(i, j + 1) = 1$ 
14.        end if
15.    end for
16. end for

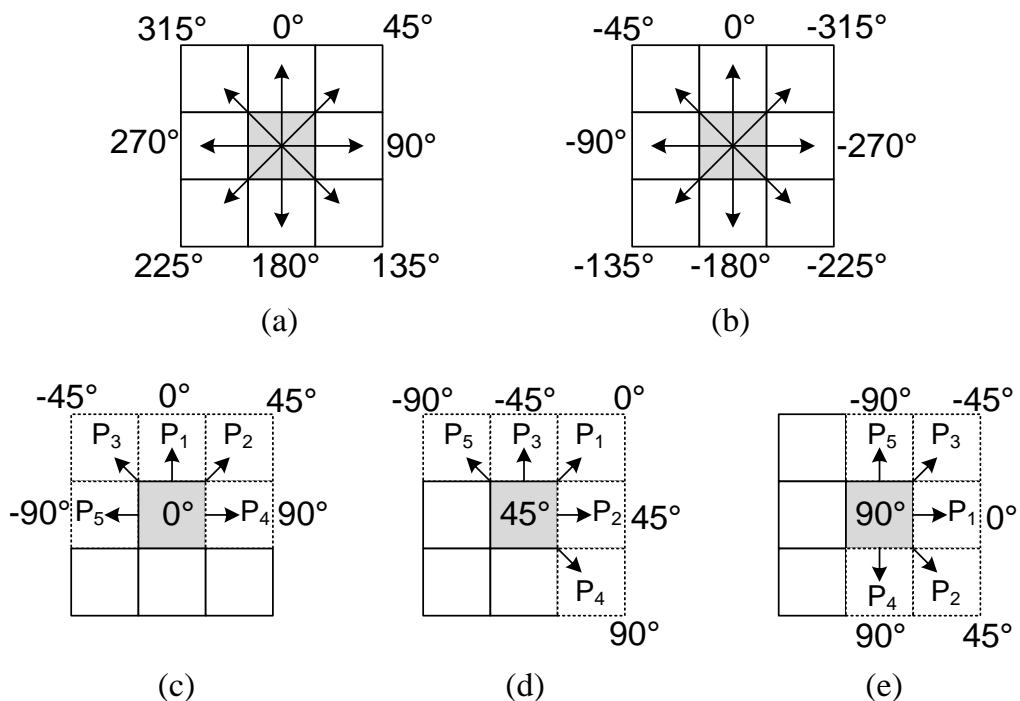
```

- Theo Pavlidis metoda praćenja konture [84].

Za navedene algoritme potrebno je odrediti: kriterij za početak i zaustavljanje praćenja konture, početni smjer konture i susjedstvo na binarnoj slici. Problemi najčešće nastaju pri definiciji susjedstva, određivanju kriterija zaustavljanja praćenja konture i račvanja [85].

5.3.1. Praćenje konture i određivanje smjera konture

U ovom radu razmatra se nova metoda za praćenje konture temeljena na metodi praćenja konture predstavljenoj u [41]. Metoda prati konturu na slici rubova $I_R(x, y)$, a ne na binarnoj slici $I_B(x, y)$. Prema definiciji praćenja konture, rezultat praćenja konture je vektor konture $C(i)$ prema (5.2) gdje L predstavlja duljinu konture. Za vektor konture sirove pločice je karakteristično da je početna točka konture $C(0)$ jednaka zadnjoj točki konture $C(L)$ (zadnja točka konture $C(L)$ nije uključena u $C(i)$ vektor). Početna točka $C(0)$ određena je na temelju uvjeta (5.3). Pronađena $(\lceil K/2 \rceil, j)$ točka ruba je spremljena u $C(0)$ i određen je početni smjer konture $D(0)$. Smjer konture je definiran prema (5.4). Na temelju N_8 susjedstva točaka ruba smjer konture može poprimiti vrijednosti u intervalu $[-315^\circ, 315^\circ]$ u koracima od 45° , cf. slika 5.4 (a) i (b). Na temelju trenutne točke konture, trenutnog smjera konture i



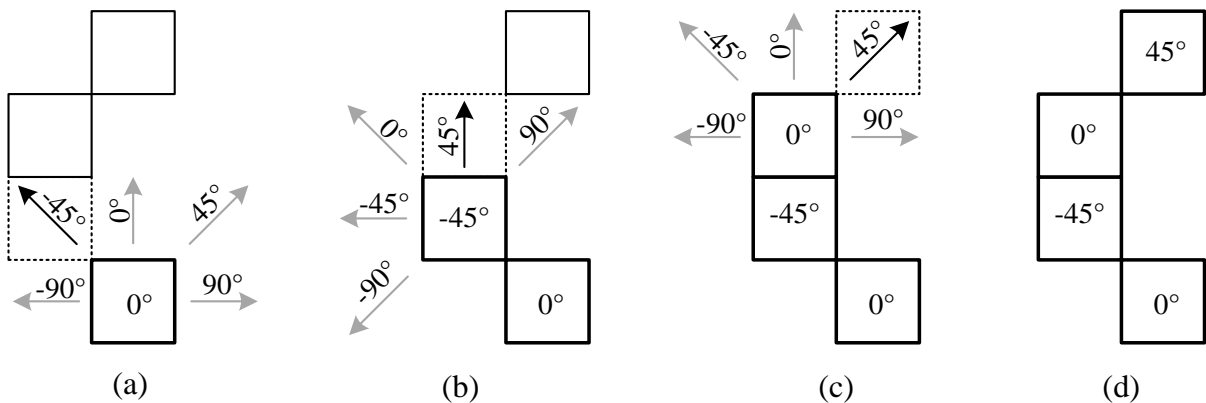
Slika 5.4: Mogući smjerovi konture za piksel sive boje (a) i (b). Primjeri $N_5^{D(i)}$ susjedstva i prioriteta pretraživanja susjedstva za trenutni smjer konture $D(i) = 0^\circ$ (c), $D(i) = 45^\circ$ (d) i $D(i) = 90^\circ$ (e).

N_5^D susjedstva pretražuje se sljedeća točka ruba.

$$\begin{aligned} C(i) &= \{(x_i, y_i)\}, i = 0, 1, \dots, L - 1, \\ (x_0, y_0) &= (x_L, y_L) \end{aligned} \quad (5.2)$$

$$\begin{aligned} I_R(\lfloor K/2 \rfloor, j) &> 0, j = M - 1, M - 2, \dots, 0 \text{ ili} \\ I_R(\lfloor K/2 \rfloor, j) &> 0, j = 0, 1, \dots, M - 1 \end{aligned} \quad (5.3)$$

$N_5^{D(i)}$ susjedstvo predstavlja modifikaciju N_8 susjedstva. $N_5^{D(i)}$ susjedstvo sastoji se od pet N_8 susjednih piksela čije koordinate ovise o relativnom pomaku $\alpha \in \{0^\circ, 45^\circ, -45^\circ, 90^\circ, -90^\circ\}$ u odnosu na trenutni smjer konture. Na slikama 5.4 (c) do (e) prikazani su primjeri $N_5^{D(i)}$ susjedstva za trenutni smjer konture $D(i) = \{0^\circ, 45^\circ, 90^\circ\}$. Sljedeća točka ruba pretražuje se po prioritetima u $N_5^{D(i)}$ susjedstvu. Prioriteti su označeni s P_1 do P_5 , gdje P_1 predstavlja najveći prioritet pretraživanja $\alpha = 0^\circ$, a P_5 najmanji prioritet pretraživanja $\alpha = -90^\circ$. Sljedeća točka konture $C(i + 1)$ je pronađena točka ruba u $N_5^{D(i)}$ susjedstvu s najvećim prioritetom. Smjer konture $D(i + 1)$ određen je na temelju (5.5), gdje $\alpha(i)$ predstavlja relativni pomak za pronađenu točku ruba s najvišim prioritetom. Na slici 5.5 (a) do (d) prikazan je primjer praćenja konture za četiri točke ruba s početnim smjerom $D(0)$ postavljenim u 0° za sliku 5.5 (a). Za sliku 5.5 (b) $D(1) = -45^\circ$ određen je na temelju $D(0) = 0^\circ$, $\alpha(0) = -45^\circ$ i (5.5), ista metodologija se primjenjuje i za ostale točke ruba 5.5 (c) i (d). Pronađene točke ruba su označene (za slučaj slike 5.5 podebljani crni kvadrat) da ne bi došlo do slučaja ponovnog pronalaska iste točke ruba.



Slika 5.5: Primjer praćenja konture i određivanja smjera konture za četiri točke ruba (a) do (d) i $D(0) = 0^\circ$. Pronađena sljedeća točka konture je označena sa crtanim kvadratom, izračunati $D(i)$ je upisan u središte kvadrata.

$$D(i) = \{d_i: d_i \in d\}, i = 0, 1, \dots, L - 1, \tag{5.4}$$

$$d \in \{-315^\circ, -270^\circ, -225^\circ, \dots, 315^\circ\}$$

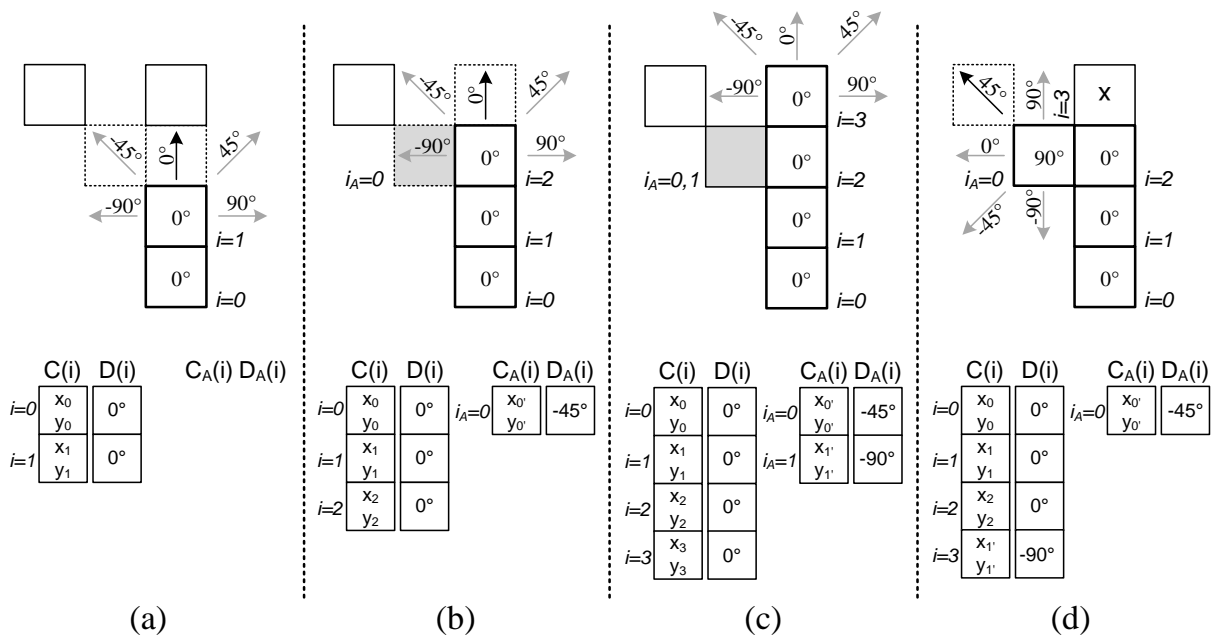
$$D(i + 1) = D(i) + \alpha(i) \tag{5.5}$$

$$C_A(i_A) = \{(x_{i_A}, y_{i_A})\}, i_A = 0, 1, \dots, L_A - 1, \tag{5.6}$$

$$D_A(i_A) = \{d_{i_A}: d_{i_A} \in d\}, i_A = 0, 1, \dots, L_A - 1$$

5.3.2. Alternativna kontura i smjer konture

Ako u $N_5^{D(i)}$ susjedstvu postoji više od jedne točke ruba, kreiraju se nova dva vektora $C_A(i_A)$ i $D_A(i_A)$ prema (5.6) koji predstavljaju alternativne točke konture i alternativni smjer konture. U (5.6) i_A predstavlja brojač alternativnih točaka konture, a L_A predstavlja ukupni broj alternativnih točaka konture. Alternativni smjer konture $D_A(i_A)$ određen je na temelju (5.7), gdje $\alpha'(i_A)$ predstavlja relativni pomak za pronađenu točku ruba s nižim prioritetima. Sa $C_A(i)$ i $D_A(i)$ riješen je problem račvanja, tj. točke u $N_5^{D(i)}$ susjedstvu s nižim prioritetima nisu zanemarene. Ako u postupku praćenja konture dođe do prekida, postupak praćenja



Slika 5.6: Primjer četiri koraka (a) do (d) praćenja konture i popunjavanje vektora konture $C(i)$, vektora smjera konture $D(i)$, vektora alternativnih točaka konture $C_A(i)$ i vektora alternativnog smjera konture $D_A(i)$. Sivi kvadrat predstavlja alternativnu točku konture, 'x' predstavlja odbačenu točku konture (nastao prekid konture).

konture se nastavlja na zadnjoj točki ruba u $C_A(i)$ vektoru, tj. $C_A(L_A - 1)$. Na slici 5.6 prikazan je primjer upotrebe alternativnih točaka konture i alternativnog smjera konture. Za slučaj prikazan na slici 5.6 (a) u susjedstvu $i = 1$ točke se nalaze dvije točke ruba. U $C(i)$ i $D(i)$ vektor spremljena je točka s prioritetom P_1 , a točka s prioritetom P_3 je spremljena u $C_A(i_A)$ i $D_A(i_A)$ vektor kako je i prikazano na slici 5.6 (b). Na slici 5.6 (c) dolazi do prekida konture tj. u točki $i = 3$ nema susjedne točke ruba u $N_5^{0^\circ}$ susjedstvu. Postupak praćenja se nastavlja na zadnjoj pronađenoj alternativnoj točki $C_A(1)$, tj. $C(3) = C_A(1)$, $D(3) = D_A(1)$ i $L_A = L_A - 1$. Na ovaj način nije došlo do problema s prekidom konture kako je i vidljivo na slici 5.6 (d). Isti postupak se ponavlja ako u $N_5^{D(i)}$ susjedstvu su sve točke ruba već označene.

$$D_A(i_A) = D(i) + \alpha'(i) \quad (5.7)$$

Postupak praćenja konture se zaustavlja ako su koordinate sljedeće točke ruba jednake koordinatama početne točke konture – praćenje konture je uspješno. Postupak praćenja konture se prekida ako je duljina konture veća od maksimalne duljine konture ($L > L_{max}$) i ako ne postoji niti jedna alternativna točka konture – praćenje konture je neuspješno. L_{max} predstavlja maksimalnu duljinu konture određenu na temelju (5.8) uz pretpostavku da je na slici veličine $K \times M$ piksela maksimalna veličina sirove pločice $(K - 1) \times (M - 1)$ piksela. Ako navedena pretpostavka nije zadovoljena metoda za detekciju ruba neće uspješno detektirati rub sirove pločice. Postupak praćenja konture se također prekida ako je duljina konture manja od minimalne duljine konture, $L < L_{min}$ i ako ne postoji niti jedna alternativna točka konture – praćenje konture je neuspješno. L_{min} predstavlja minimalnu duljinu konture određenu na temelju k referentnih slika sirove pločice prema (5.9), gdje L_i predstavlja duljinu konture i -te referentne slike sirove pločice.

$$L_{max} = 2 \times (K - 1) + 2 \times (M - 1) \quad (5.8)$$

$$L_{min} = \left| \frac{1}{k} \sum_{i=0}^{k-1} L_i \right| - \left| \frac{0.01}{k} \sum_{i=0}^{k-1} L_i \right| \quad (5.9)$$

5.3.3. Sekvencijalni algoritam praćenja konture i određivanja smjera konture

Konačni pseudo kod postupka za praćenje konture je prikazan u algoritmu 5.2. Konačni rezultat predloženog algoritma 5.2 su vektori $C(i)$ i $D(i)$ ili *pogreška u duljini konture*. Ako je rezultat praćenja konture pogreška u duljini konture pretpostavlja se da je sirova pločica neispravna i odvaja se s pokretne linije (sirova pločice se preusmjerava u postupak miješanja prema slici 2.4). Postupak pretraživanja $N_5^{D(i)}$ susjedstva implementirano je s jednostavnim postupkom pristupa memorijskoj tablici (*engl. lookup table*) na temelju trenutnog smjera konture. Oblik i vrijednosti implementirane memorijske tablice su prikazani u tablici 5.1. Za primjer trenutnog smjera konture $D(i) = 45^\circ$ u tablici 5.1 pristupa se redu $D(i) = 45^\circ$ i za svaki prioritet P_1 do P_5 pristupa se pojedinom stupcu tablice 5.1. Koordinate x i y se mijenjaju na temelju vrijednosti iz tablice 5.1, npr. za $D(i) = 45^\circ$ i prioritet P_1 koordinate točke ruba u $N_5^{45^\circ}$ susjedstvu su: $x = x + 1$, $y = y - 1$. Predloženim postupkom omogućeno je jednostavno i brzo određivanja koordinata točaka u $N_5^{D(i)}$ susjedstvu bez upotrebe trigonometrijskih funkcija kako je predloženo u [41] i [77].

Algoritam 5.2: Praćenje konture sirove pločice predloženom metodom.

```

1. Ulaz:  $I_R(x, y)$ ,  $M$ ,  $K$ ,  $L_{max}$ ,  $L_{min}$ 
2. Pronađi:  $C(0)$  prema (5.3) i označi
3. Postavi:  $D(0) = 0^\circ$  ili  $D(0) = 180^\circ$ ,  $L = 1$ ,  $i_A = 0$ 
4. while ( $C(L) \neq C(0)$ )
5.     Pronađi:  $C(L)$  u  $N_5^{D(L-1)}$  i označi
6.     Postavi:  $D(L)$  prema 5.5
7.     while postoje točke ruba manjih prioriteta u  $N_5^{D(L-1)}$ 
8.         Postavi:  $C_A(i_A)$ ,  $D_A(i_A)$ ,  $i_A = i_A + 1$ 
9.     end while
10.    if ( $L > L_{max}$  i postoje alternativne točke) then
11.         $C(L) = C_A(i_A - 1)$ ,  $D(L) = D_A(i_A - 1)$ 
12.         $i_A = i_A - 1$ 
13.    else if  $L > L_{max}$  then
14.        Pogreška u duljini konture
15.    end if
16.    if (ne postoji točka ruba u susjedstvu ili  $C(L)$  je označeno) then
17.         $C(L) = C_A(i_A - 1)$ ,  $D(L) = D_A(i_A - 1)$ 
18.         $i_A = i_A - 1$ 
19.    end if
20.     $L = L + 1$ 
21. end while
22. if  $L < L_{min}$  then
23.    Pogreška u duljini konture
24. end if

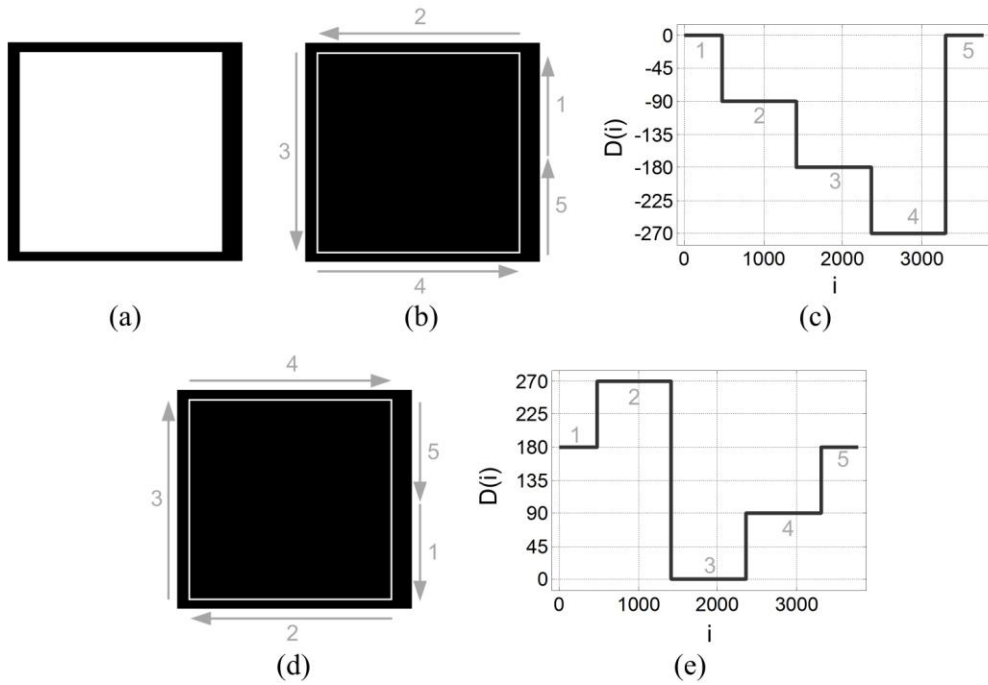
```

Tablica 5.1: Oblik i vrijednosti memorijske tablice za određivanje koordinata točaka ruba u $N_5^{D(i)}$ susjedstvu na temelju trenutnog smjera konture $D(i)$.

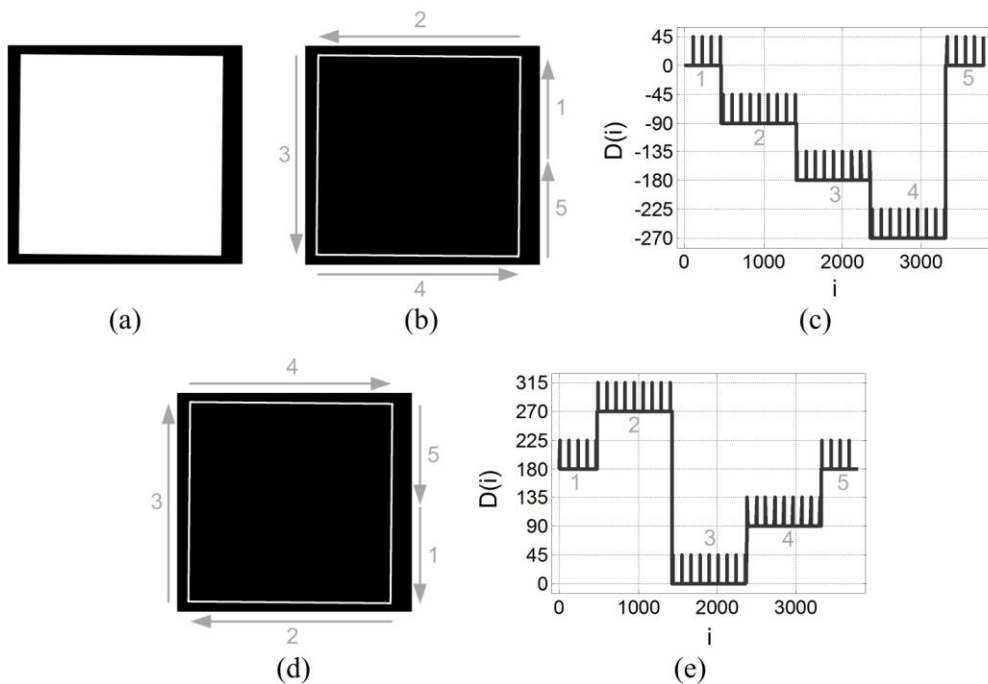
| $D(i)$ | $N_5^{D(i)}$ | | | | | | | | | |
|--------------|-------------------------|-----|--------------------------|-----|---------------------------|-----|--------------------------|-----|---------------------------|-----|
| | $P_1, \alpha = 0^\circ$ | | $P_2, \alpha = 45^\circ$ | | $P_3, \alpha = -45^\circ$ | | $P_4, \alpha = 90^\circ$ | | $P_5, \alpha = -90^\circ$ | |
| | x | y | x | y | x | y | x | y | x | y |
| 0° | 0 | -1 | +1 | -1 | -1 | -1 | +1 | 0 | -1 | 0 |
| 45° | +1 | -1 | +1 | 0 | 0 | -1 | +1 | +1 | -1 | -1 |
| 90° | +1 | 0 | +1 | +1 | +1 | -1 | 0 | +1 | 0 | -1 |
| 135° | +1 | +1 | 0 | +1 | +1 | 0 | -1 | +1 | +1 | -1 |
| 180° | 0 | +1 | -1 | +1 | +1 | +1 | -1 | 0 | +1 | 0 |
| 225° | -1 | +1 | -1 | 0 | 0 | +1 | -1 | -1 | +1 | +1 |
| 270° | -1 | 0 | -1 | -1 | -1 | +1 | 0 | -1 | 0 | +1 |
| 315° | -1 | -1 | 0 | -1 | -1 | 0 | +1 | -1 | -1 | +1 |
| -315° | +1 | -1 | +1 | 0 | 0 | -1 | +1 | +1 | -1 | -1 |
| -270° | +1 | 0 | +1 | +1 | +1 | -1 | 0 | +1 | 0 | -1 |
| -225° | +1 | +1 | 0 | +1 | +1 | 0 | -1 | +1 | +1 | -1 |
| -180° | 0 | +1 | -1 | +1 | +1 | +1 | -1 | 0 | +1 | 0 |
| -135° | -1 | +1 | -1 | 0 | 0 | +1 | -1 | -1 | +1 | +1 |
| -90° | -1 | 0 | -1 | -1 | -1 | +1 | 0 | -1 | 0 | +1 |
| -45° | -1 | -1 | 0 | -1 | -1 | 0 | +1 | -1 | -1 | +1 |

5.3.4. Primjeri praćenja smjera konture

Na slikama 5.7 i 5.8 prikazan je primjer smjera konture $D(i)$ za sintetički generirane binarne slike sirove pločice. Slika 5.7 (a) predstavlja sintetički generiranu binarnu sliku $I_B(x, y)$ sirove pločice s kutom rotacije $\gamma = 0^\circ$. Slika ruba $I_R(x, y)$ je prikazana na slici 5.7 (b) i (d). Strelice na slikama 5.7 (b) i (d) s naznačenim brojevima predstavljaju poveznicu između slike ruba i grafa smjer konture. Sa slika ruba vidljivo je da svaki rub sirove pločice predstavlja savršenu liniju piksela. U ovom radu pod pojmom savršena linija piksela smatra se niz točaka ruba kojima se mijenja samo jedna koordinata x ili y . Primjer savršene/nesavršene linije piksela je prikazan na slici 5.9 (a) i (b). Isto se može zaključiti i s grafova smjera konture prikazanih na slici 5.7 (c) i (e). Grafovi se sastoje od četiri skoka. Svaki skok predstavlja pojedini ugao sirove pločice, a horizontalne ravne linije predstavljaju rubove sirove pločice. Horizontalna ravna linija označena s 3 na slici 5.7 (c) i (e) predstavlja lijevi rub sirove pločice također označen s 3 na slikama 5.7 (b) i (d). Budući da su rubovi sintetički generirane binarne slike sirove pločice na slici 5.7 (a) savršene linije piksela tako se i graf smjera konture sastoji od četiri horizontalne linije. Desni rub pločice je podijeljen na dva dijela označen s 1 i 5 na slikama 5.7 (b) do (e) i 5.8 (d) do (e) zbog načina pronalaska prve točke konture $C(0)$.



Slika 5.7: Primjer smjera konture: (a) sintetički generirana binarna slika sirove pločice $I_B(x, y)$ s kutom rotacije $\gamma = 0^\circ$, (b) slika ruba $I_R(x, y)$ s naznačenim smjerom praćenja konture za $D(0) = 0^\circ$, (c) graf smjera konture $D(i)$ za $D(0) = 0^\circ$, (d) slika ruba $I_R(x, y)$ s naznačenim smjerom praćenja konture za $D(0) = 180^\circ$, (e) graf smjera konture $D(i)$ za $D(0) = 180^\circ$.



Slika 5.8: Primjer smjera konture: (a) sintetički generirana binarna slika sirove pločice $I_B(x, y)$ s kutom rotacije $\gamma = 0.5^\circ$, (b) slika ruba $I_R(x, y)$ s naznačenim smjerom praćenja konture za $D(0) = 0^\circ$, (c) graf smjera konture $D(i)$ za $D(0) = 0^\circ$, (d) slika ruba $I_R(x, y)$ s naznačenim smjerom praćenja konture za $D(0) = 180^\circ$, (e) graf smjera konture $D(i)$ za $D(0) = 180^\circ$.

Kako je navedeno u uvjetu (5.3) prva točka konture dijeli desni/lijevi rub sirove pločice na dva dijela tj. dio 1 i dio 5 na slikama 5.7 (b) do (e) i 5.8 (d) do (e). Slika 5.8 (a) predstavlja sintetički generiranu binarnu sliku $I_B(x, y)$ sirove pločice s kutom rotacije $\gamma = 0.5^\circ$. Zbog rotacije i diskretnih koordinata piksela, rub slike nije savršena linija piksela nego dolazi do manjih promjena u obje koordinate piksela ruba (na slici 5.9 (b) prikazan je primjer nesavršene linije piksela). Isto se može zaključiti i na grafovima smjera konture prikazanih na slici 5.8 (c) i (e). Grafovi se sastoje od četiri "veća" skoka koji predstavljaju pojedini ugao sirove pločice, ali se između "većih" skokova nalazi niz "manjih" skokova. Manji skokovi nastaju zbog nesavršenosti linije piksela, kako je i prikazano na slici 5.9 (b), zbog čega dolazi do promjene u smjeru konture za 45° . Svaka promjena od 45° je prikazana na grafu smjera konture kao skok. Navedena nesavršenost linije piksela ne nastaje samo zbog kuta rotacije pločice γ nego i zbog sustava za prikupljanje slike sirove pločice. Mogući uzroci nastajanja nesavršenih linija piksela u sustavu za prikupljanje slike su: položaj kamere, geometrijska distorzija leće, rasvjeta i sl.

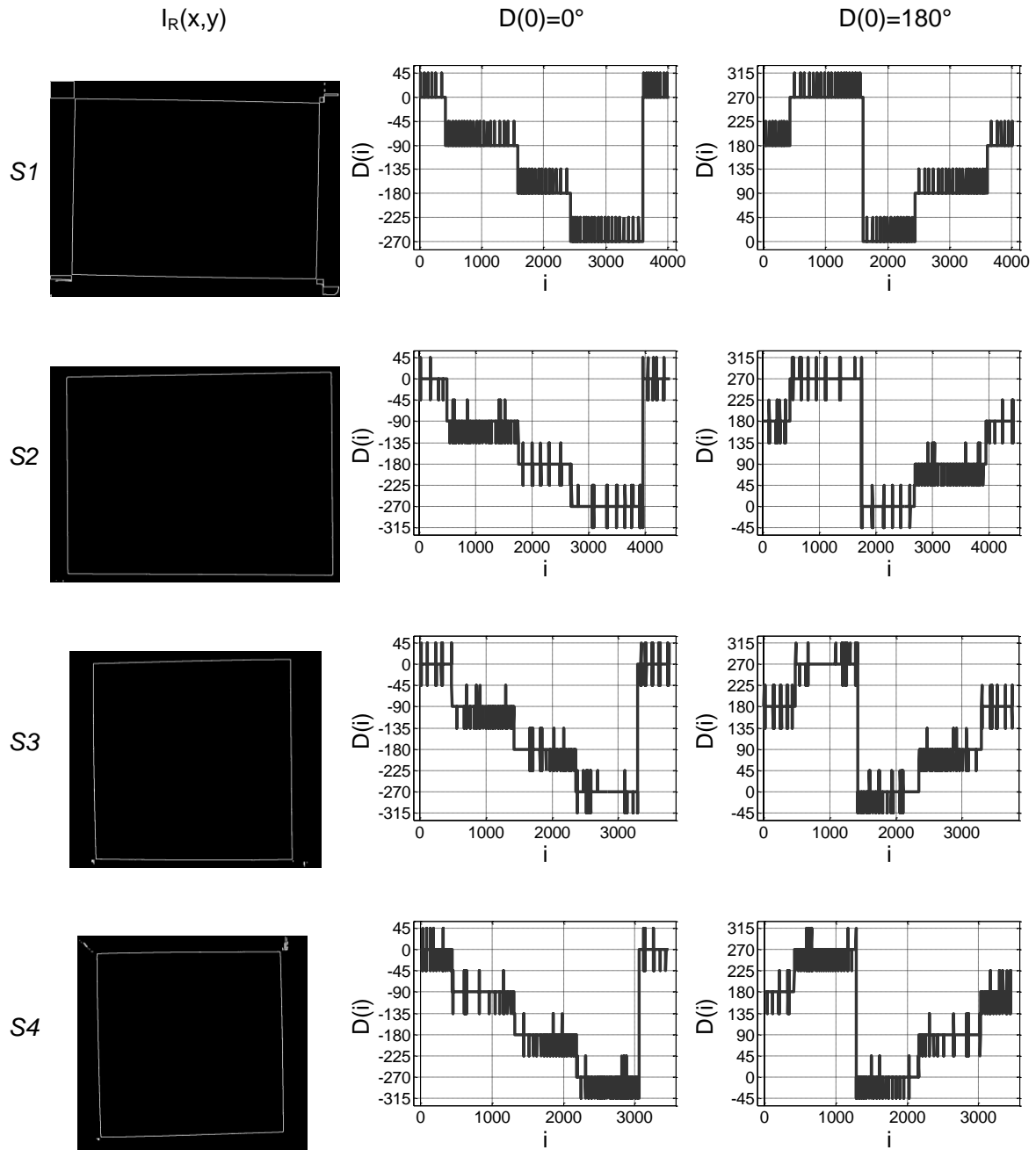
Na slici 5.10 prikazani su primjeri grafa smjera konture za primjere slika sirovih pločica S_1 , S_2 , S_3 i S_4 prikazanih na slici 4.24 (a) do (d). Slike su binarizirane s PBSCD metodom i na dobivenim binarnim slikama primijenjena je predložena metoda za detekciju ruba. Slike ruba za pojedini primjer slike sirove pločice (S_1 do S_4) prikazan je na slici 5.10 u prvom stupcu. Na slikama ruba može se primijetiti da pored samog ruba pločice postoje i dodatne točke ruba u uglovima $I_R(x, y)$ slike. Navedeni rubovi su rezultat binarizacije PBSCD metodom kako je objašnjeno u poglavlju 4.7.2. Za postupak praćenja konture prikazani rubovi ne predstavljaju problem budući da je prva točka konture pronađena na temelju uvjeta (5.3) tj. pretražuje se srednji red slike ruba koji uključuje samo rub sirove pločice.

Na slici 5.10 u drugom i trećem stupcu prikazani su grafovi smjera konture za pojedini primjer slika sirove pločice. Grafovi prikazuju slične karakteristike kao grafovi na slici 5.8. Sastoje se od četiri "veća" skoka koji predstavljaju uglove sirovih pločica i niza "manjih" skokova koji nastaju zbog nesavršenosti linije piksela, v. sliku 5.9 (b).



Slika 5.9: Primjer savršene linije piksela (a) i nesavršene linije piksela (b).

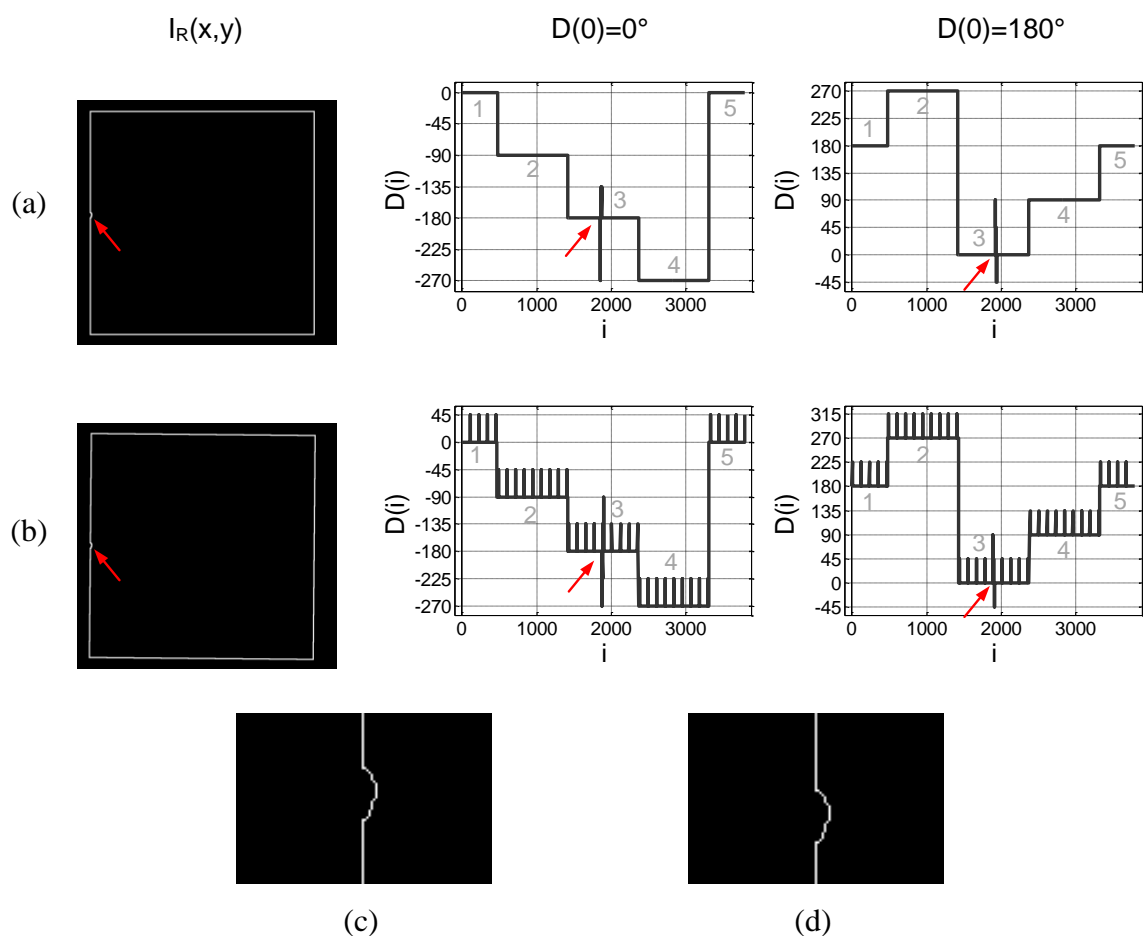
Na slici 5.11 (a) prikazan je primjer praćenja konture za sintetički generiranu sliku sirove pločice sa savršenim linijama piksela ruba i jednom neispravnosti na lijevom rubu označenom sa crvenom strelicom (slika 5.11 (c) prikazuje uvećani dio slike ruba). Iz grafova smjera konture na slici 5.11 (a) vidljivo je da linija označena s 3 nije više horizontalna kao na slici 5.7 (c) i (e) nego sadrži niz "manjih" i "većih" skokova. Navedeni skokovi nastaju zbog



Slika 5.10: Smjer konture za primjere slika sirovih pločica $S1$, $S2$, $S3$, i $S4$ (označeni redovi) iz poglavlja 4.9. Stupac $I_R(x, y)$ predstavlja slike ruba za pojedini primjer slika, stupac $D(0) = 0^\circ$ predstavlja grafove smjera konture sa početnim smjerom konture 0° , stupac $D(0) = 180^\circ$ predstavlja grafove smjera konture sa početnim smjerom konture 180° .

naglih promjene smjera konture na mjestu neispravnosti. Pronalaskom prikazanih skokova moguće je detektirati neispravnost na rubovima sirove pločice. Problem nastaje kada rubovi pločice ne predstavljaju savršenu ravnu liniju piksela i na rubu se nalazi neispravnost. Na slici 5.11 (b) prikazan je primjer praćenja konture za sintetički generiranu sliku sirove pločice s jednom neispravnosti na lijevom rubu, označenom sa crvenom strelicom, čiji rubovi nisu savršene linije piksela (slika 5.11 (c) prikazuje uvećani dio slike ruba). Iz grafova smjera konture na slici 5.11 (b) vidljivo je da skokovi nastali zbog neispravnosti na rubu su manje izraženi. Razlog tomu su "manji" skokovi nastali zbog nesavršene linije piksela ruba pločice. Zbog navedenog nije moguće razlikovati da li je "manji" skok nastao zbog neispravnosti ili zbog nesavršenosti linije piksela ruba.

U ovom radu se pretpostavlja da nije moguće dobiti sliku sirove pločice sa sustavom za prikupljanje slike (prototip sustava je predstavljen u prilogu 3.2) čiji će rubovi predstavljati



Slika 5.11: Smjer konture za sintetički generiranu binarnu sliku sirove pločice s kutom rotacije $\gamma = 0^\circ$ i s jednom neispravnosti na lijevom rubu (a), s kutom rotacije $\gamma = 0.5^\circ$ i s jednom neispravnosti na lijevom rubu (b), uvećani dio slike ruba (a) i (b) s vidljivom neispravnosti na rubu (c) i (d)

savršenu liniju piksela. Također se pretpostavlja da će graf smjera konture uvijek sadržavati "veće" i "manje" skokove kako je prikazano na slikama 5.8 i 5.10. Ako postoje neispravnosti na rubovima, iste će biti manje izražene na grafu smjera konture zbog nesavršenosti linije piksela ruba. Iz prikazanih primjera dolazi se do zaključka da je potrebno dodatno filtrirati smjer konture da bi bilo moguće detektirati i lokalizirati neispravnosti na rubovima sirove pločice. Metode za filtriranje smjera konture razmatrane u ovom radu i postupak za detekciju i lokalizaciju neispravnosti na rubovima i uglovima sirove pločice objašnjene su u narednim potpoglavljima.

5.4. Filtriranje smjera konture

Kako je objašnjeno u prethodnom poglavlju i pokazano na raznim primjerima, smjer konture sirove pločice se sastoji od "većih" i "manjih" skokova. Skokovi u smjeru konture predstavljaju uglove sirove pločice, neispravnosti i nesavršenosti linije piksela rubova. Zbog nesavršenosti linije piksela rubova pločice skokovi u smjeru konture uzrokovani neispravnostima na rubovima su manje izraženi. Da bi skokovi u smjeru konture uzrokovani neispravnostima na rubovima pločice bili izraženiji potrebno je filtrirati vektor smjera konture. U ovom radu predložena su dva koraka filtriranja:

1. osrednjavanje s pomičnim prosjekom prema [7] i
2. ublažavanje s Gausovim filterom prema [86].

5.4.1. Osrednjavanje s pomičnim prosjekom

Pomični prosjek je jednostavna statistička metoda za osrednjavanje vrijednosti niza (u ovom radu vektor smjera konture $D(i)$). Osrednjavanje niza s pomičnim prosjekom omogućava smanjenje kolebanja vrijednosti niza tj. smanjuje skokove nastale zbog nesavršenosti linije piksela ruba u smjeru konture $D(i)$. Također u literaturi se koristi izraz *izravnavanje vrijednosti niza*, što se može poistovjetiti s izravnavanjem horizontalnih linija u vektoru smjera konture $D(i)$, v. poglavlje 5.2 i slike 5.7 i 5.8 [87]. Za niz vrijednosti $s(i)$, $i = 0, 1, \dots, m$ osrednjena vrijednost niza $\bar{s}(j)$ s pomičnim prosjekom i prozorom širine w je

$$\bar{s}(j + \lfloor w/2 \rfloor) = \frac{1}{w} \sum_{k=0}^{w-1} s(j \times w + k), j = 0, 1, \dots, m - w, \quad (5.10)$$

gdje $\lfloor \cdot \rfloor$ predstavlja funkciju najmanje cijelo. Iz (5.10) se zaključuje da niz osrednjen s pomičnim prosjekom se sastoji od $m - w$ točaka tj. nema jednak broj točaka kao i niz $s(i)$.

Za vektor smjera konture $D(i)$ zadnja točka vektora $D(L-1)$ prethodi prvoj točki vektora $D(0)$. Do istog se zaključka može doći i na temelju slike 5.7 (b) i (d). Da broj točaka vektora smjera konture $D(i)$ bude jednak broju točaka osrednjenom vektoru smjera konture $\bar{D}(i)$ upotrebljava se kružni pomični prosjek. Za vektor smjera konture $D(i)$ definiran s (5.4) osrednjena vrijednost $\bar{D}(i)$ s kružnim pomičnim prosjekom i prozorm širine w je

$$\bar{D}(i) = \frac{1}{w} \sum_{k=-\frac{w-1}{2}}^{\frac{w-1}{2}} D(\text{korak}_{i,k}), i = 0, 1, \dots, L-1, \quad (5.11)$$

gdje $\text{korak}_{i,k}$ predstavlja kružni pomak u vektoru smjera konture. Pretpostavlja se da je širina prozora w uvijek neparan pozitivan cijeli broj. $\text{korak}_{i,k}$ je definiran s (5.12) i omogućava izračun $\bar{D}(i)$ za točke u intervalu $[L-w+1, L-1]$ što nije slučaj za pomični prosjek definiran s (5.10).

$$\text{korak}_{i,k} = \begin{cases} i \times w + k & \text{ako } 0 \leq (i \times w + k) < L, \\ i \times w + k - L & \text{ako } (i \times w + k) \geq L, \\ L - k & \text{ako } (i \times w + k) < 0 \end{cases} \quad (5.12)$$

Vektor $\bar{s}(j + \lfloor w/2 \rfloor)$ određen na temelju (5.10) sadrži vrijednosti u intervalu $[\lfloor w/2 \rfloor, \lfloor w/2 \rfloor + 1, \dots, m - \lfloor w/2 \rfloor]$ tj. ne sadrži dio početnih i krajnjih vrijednosti vektora. Na temelju (5.12) vektor $\bar{D}(i)$ određen prema formuli za kružni pomični prosjek (5.11) sadrži sve vrijednosti u intervalu $[0, 1, \dots, L-1]$, tj. nije došlo do gubitka podataka s postupkom osrednjavanja.

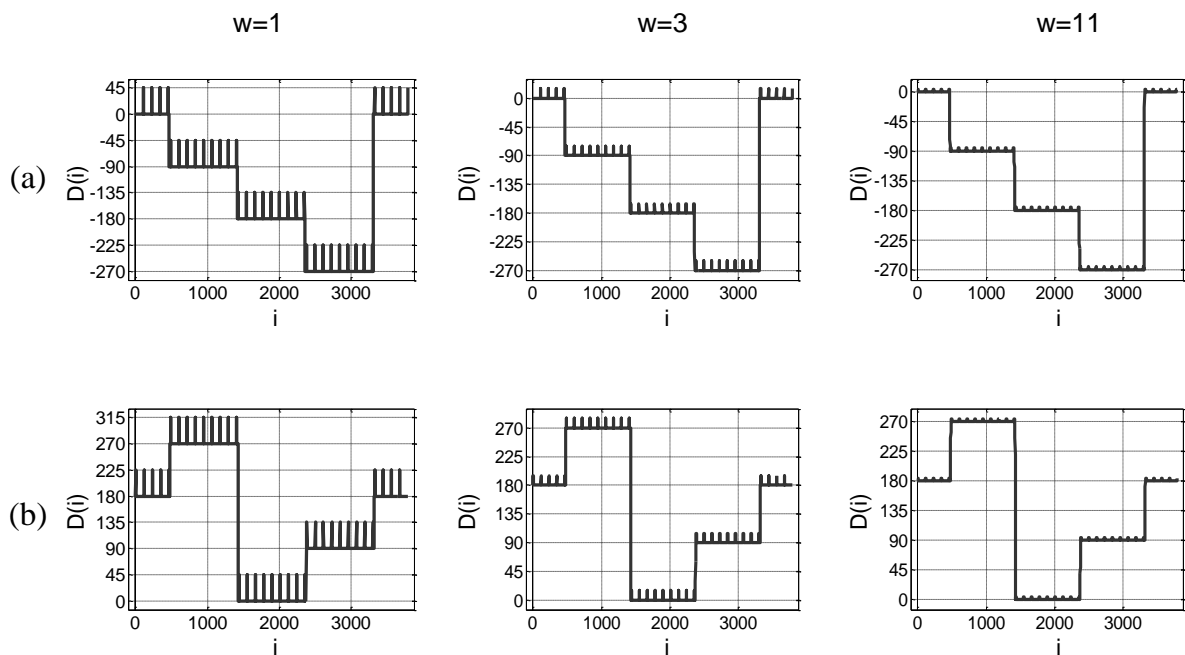
Slika 5.12 prikazuje primjer $\bar{D}(i)$ za $w \in \{1, 3, 11\}$ za sintetički generiranu binarnu sliku sirove pločice s kutom rotacije $\gamma = 0.5^\circ$, v. sliku 5.8. $\bar{D}(i)$ za širinu prozora $w = 1$ je jednak vektoru smjera konture $D(i)$ što proizlazi iz definicije kružnog pomičnog prosjeka prema (5.11). S povećanjem širine prozora w vidljivo je sa slike 5.12 da dolazi do smanjenja "manjih" skokova koji nastaju zbog nesavršenosti linije piksela ruba. Skokovi koji nastaju zbog uglova pločice i dalje su naglašeni. Može se zaključiti da s kružnim pomičnim prosjekom horizontalne linije na grafu smjera konture su *izravnate* s time da je broj točaka vektora smjera konture nepromijenjen i skokovi uzrokovani s uglovima pločice su naglašeni.

Na slici 5.13 prikazan je primjer $\bar{D}(i)$ za $w \in \{1, 3, 11\}$ za sintetički generiranu binarnu sliku sirove pločice s kutom rotacije $\gamma = 0.5^\circ$ i jednom neispravnošću na lijevom rubu, v. sliku 5.11. Kako je pokazano na slici 5.12 i na slici 5.13 s povećanjem širine prozora w

horizontalne linije na grafu smjera konture su izravnane, ali je također vidljivo da su skokovi u smjeru konture uzrokovani s neispravnnošću na rubu ostali naglašeni.

Slika 5.14 predstavlja uvećane dijelove grafa osrednjenog smjera konture $\bar{D}(i)$. Na slici 5.14 (a) vidljiva su dva stepeničasta skoka (označeni s *) nastali zbog nesavršenosti linije piksela ruba. Širina skoka jednaka je širini prozora w . Slika 5.14 (b) prikazuje jedan stepeničasti skok nastao zbog nesavršenosti linije piksela ruba i skokove nastali zbog neispravnosti na rubu. Vidljivo je da je promjena u smjeru konture zbog neispravnosti na rubu nekoliko puta veća u odnosu na promjenu nastalu zbog nesavršenosti linije piksela ruba (32° u odnosu na 4°).

Stepeničasti skokovi u vektoru smjera konture nisu pogodni za nastavak postupka detekcije neispravnosti na rubovima i uglovima sirove pločice. Iste je potrebno izbjeći, tj. ublažiti skokove.

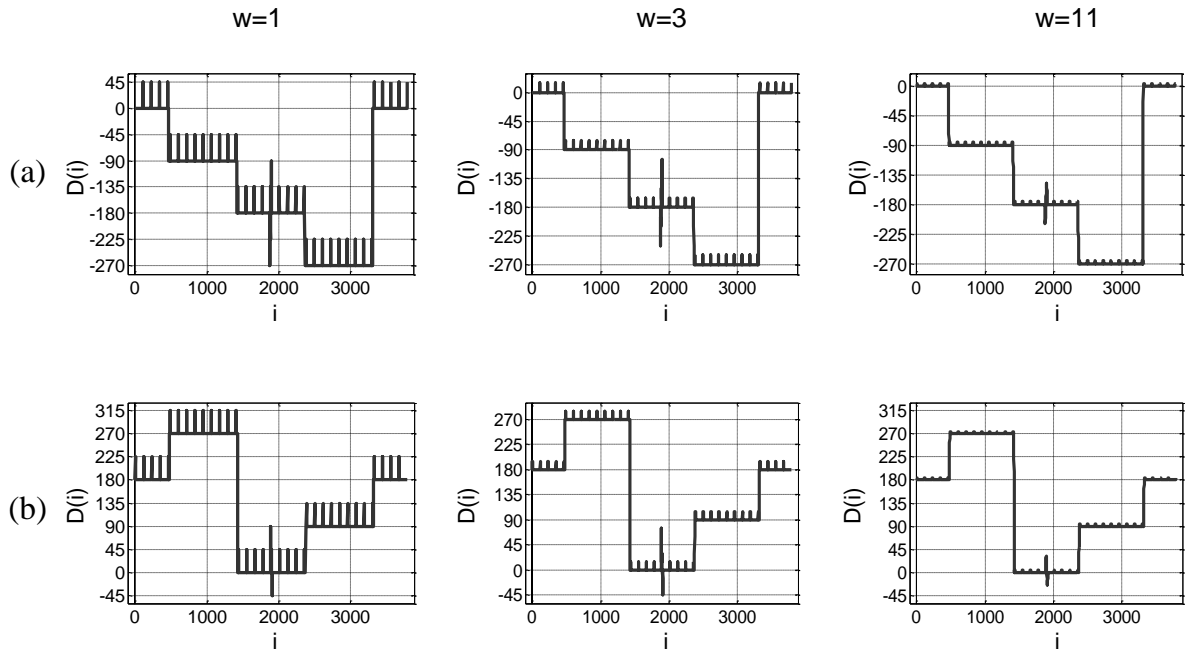


Slika 5.12: Osrednjeni smjer konture s kružnim pomičnim prosjekom širine prozora $w \in \{1, 3, 11\}$ za sintetički generiranu binarnu sliku sirove pločice s kutom rotacije $\gamma = 0.5^\circ$ i (a) $D(0) = 0^\circ$, (b) $D(0) = 180^\circ$.

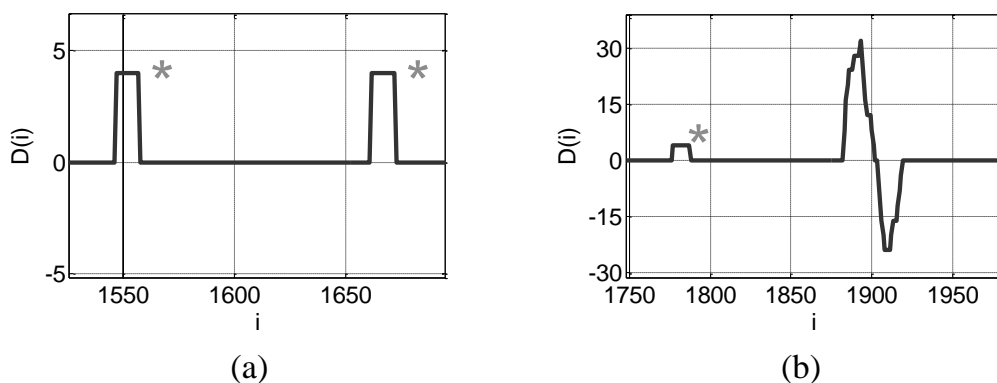
5.4.2. Ublažavanje s Gausovim filterom

U ovom radu upotrijebljen je postupak ublažavanja vektora $\bar{D}(i)$ s digitalnim Gausovim filterom. Digitalni Gausov filter zasnovan je na Gausovoj funkciji

$$g(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{t^2}{2\sigma^2}}, \quad (5.13)$$



Slika 5.13: Osrednjeni smjer konture s kružnim pomičnim prosjekom širine prozora $w \in \{1, 3, 11\}$ za sintetički generiranu binarnu sliku sirove pločice s jednom neispravnosti na lijevom rubu, kutom rotacije $\gamma = 0.5^\circ$ i (a) $D(0) = 0^\circ$, (b) $D(0) = 180^\circ$.



Slika 5.14: Uvećani dio grafa osrednjenog smjera konture prikazanog na slici 5.13 (b) za $w = 11$, (a) stepeničasti skokovi nastali zbog nesavršenosti linije piksela ruba označeni s *, (b) stepeničasti skokovi nastali zbog neispravnosti na rubu.

Tablica 5.2: Vrijednosti uzorkovane Gaussove funkcije $g(t)$ (koeficijenti digitalnog Gaussovog filtera) za $\sigma = 2.5$ i $t = [-7, 7]$ s korakom 1.

| | | | | | | | | |
|-------------------------|------|------|-------|-------|-------|--------|--------|--------|
| t | -7 | -6 | -5 | -4 | -3 | -2 | -1 | 0 |
| $g(t) [\times 10^{-3}]$ | 3.17 | 8.98 | 21.65 | 44.48 | 77.87 | 116.17 | 147.68 | 159.98 |

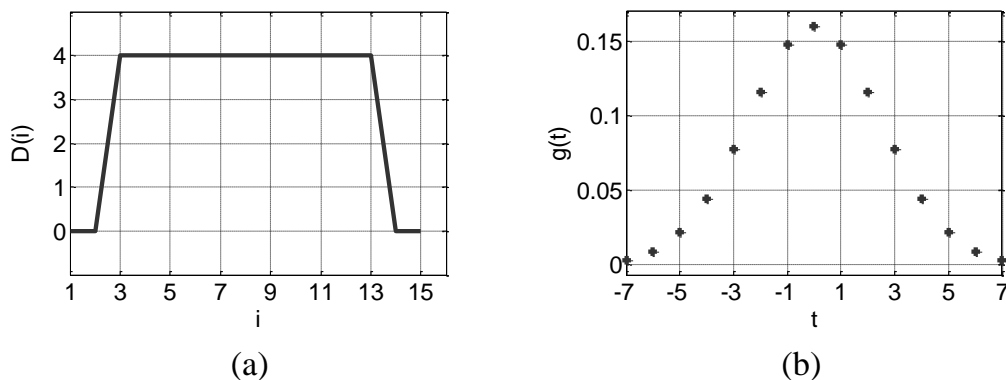
| | | | | | | | |
|-------------------------|--------|--------|-------|-------|-------|------|------|
| t | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| $g(t) [\times 10^{-3}]$ | 147.68 | 116.17 | 77.87 | 44.48 | 21.65 | 8.98 | 3.17 |

gdje σ predstavlja standardnu devijaciju. Gaussova funkcija je zvonastog oblika i naglo opada tako da se 99% površine $g(t)$ nalazi u intervalu $t = [-3\sigma, 3\sigma]$ [88]. Ublažavanje osrednjenog vektora smjera konture s digitalnim Gausovim filterom se sastoji od dva koraka:

1. uzorkovanje Gaussove funkcije i
2. konvolucije s koeficijentima digitalnog Gaussovog filtera.

Broj uzoraka Gaussove funkcije ovisi o širini stepeničastih skokova nastalih zbog nesavršenosti linije piksela ruba. Budući da je širina stepenice jednaka širini prozora pomičnog prosjeka w u ovom radu se koristi $N = w + 4$ broj uzoraka Gaussove funkcije. Na ovaj način stepeničasti skok se sastoji od 4 točke niže vrijednosti (dvije s svake strane skoka) i w točaka više vrijednosti, v. sliku 5.15 (a) gdje je niža vrijednost 0° , a viša vrijednost 4° . Iz uvjeta da 99% površine $g(t)$ se nalazi u intervalu $t = [-3\sigma, 3\sigma]$ tj. za $N = w + 4$ $t = [-\lfloor N/2 \rfloor, \lfloor N/2 \rfloor]$ proizlazi da je standardna devijacija

$$\sigma = \frac{N}{6} = \frac{w + 4}{6}. \quad (5.14)$$



Slika 5.15: Primjer stepeničastog skoka nastalog zbog nesavršenosti linije piksela ruba nakon osrednjavanja za $w = 11$ (a), (b) graf uzorkovane Gaussove funkcije za $N = 15$, $t = [-7, 7]$ s korakom 1 i $\sigma = 2.5$.

Na slici 5.15 (b) prikazan je primjer grafa uzorkovane Gaussove funkcije za 15 točaka ($w = 11$) u intervalu $t = [-7, 7]$ s korakom 1 i $\sigma = 2.5$ prema (5.14). Vrijednosti dobivene uzorkovanjem Gaussove funkcije predstavljaju koeficijente digitalnog Gaussovog filtera. Primjer dobivenih koeficijenata za $N = 15$ i $\sigma = 2.5$ prikazani su u tablici 5.2 i na slici 5.15 (b), koeficijenti su izračunati na temelju MATLAB-ove *fspecial* funkcije.

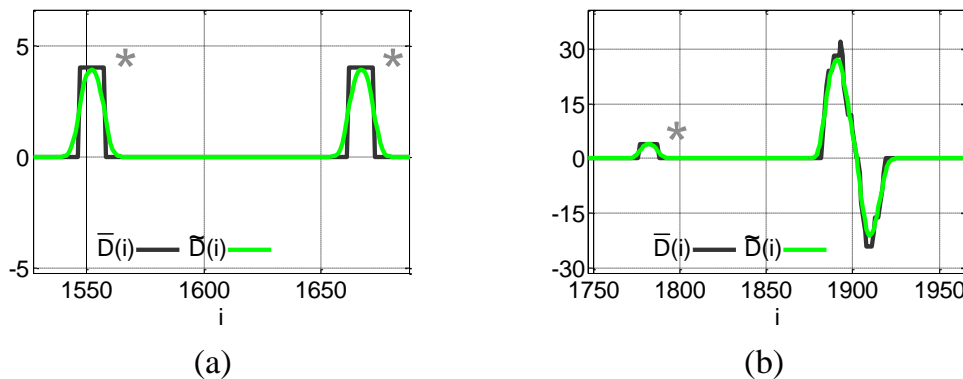
Drugi korak ublažavanja vektora $\bar{D}(i)$ je konvolucija s koeficijentima digitalnog Gaussovog filtera. Prema [42] konvolucija s digitalnim Gausovim filterom je definirana s

$$\tilde{D}(i) = \sum_{t=-\lfloor N/2 \rfloor}^{\lfloor N/2 \rfloor} \bar{D}(\text{korak}_{i,t}) \times g(t), i = 0, 1, \dots, L - 1, \quad (5.15)$$

gdje je $\text{korak}_{i,t}$ određen prema (5.16). Rezultat konvolucije vektora $\bar{D}(i)$ s digitalnim Gausovim filterom je novi vektor $\tilde{D}(i)$ veličine L s ublaženim stepeničastim skokovima. Na slici 5.16 prikazan je uvećani dio grafa smjera konture sa slike 5.14 nakon primijenjenog ublažavanja s Gausovim digitalnim filterom. Uspoređujući $\bar{D}(i)$ i $\tilde{D}(i)$ na slici 5.16 (a) vidljivo je da stepeničasti skok više nije prisutan, ublažen je. Isto se može zaključiti na temelju slike 5.14 (b) i za stepeničaste skokove nastale zbog neispravnosti na rubu.

$$\text{korak}_{i,t} = \begin{cases} 0 & \text{ako } (i+t) < 0, \\ i+t & \text{ako } 0 \leq (i+t) < L, \\ L-1 & \text{ako } (i+t) \geq L \end{cases} \quad (5.16)$$

Nakon filtriranja smjera konture nastavlja se s postupkom detekcije i lokalizacije neispravnosti na uglovima i rubovima sirove pločice.



Slika 5.16: Uvećani dio grafa vektora smjera konture nakon osrednjavanja (crna linija) i ublažavanja s digitalnim Gausovom filterom (zelena linija) za $w = 11$, $N = 15$, $\sigma = 2.5$: (a) stepeničasti skokovi nastali zbog nesavršenosti linije piksela ruba označeni s *, (b) stepeničasti skokovi nastali zbog neispravnosti na rubu.

5.5. Detekcija i lokalizacija neispravnosti na uglovima i rubovima sirove keramičke pločice

Sljedeći korak u postupku detekcije i lokalizacije neispravnosti na uglovima i rubovima sirove pločice je izračun razlike susjednih točaka filtriranog vektora smjera konture $\tilde{D}(i)$.

5.5.1. Određivanje razlike smjera konture

Prema [86] vektor razlike filtriranog smjera konture označen je s $DIF(i)$ i određen s formulom

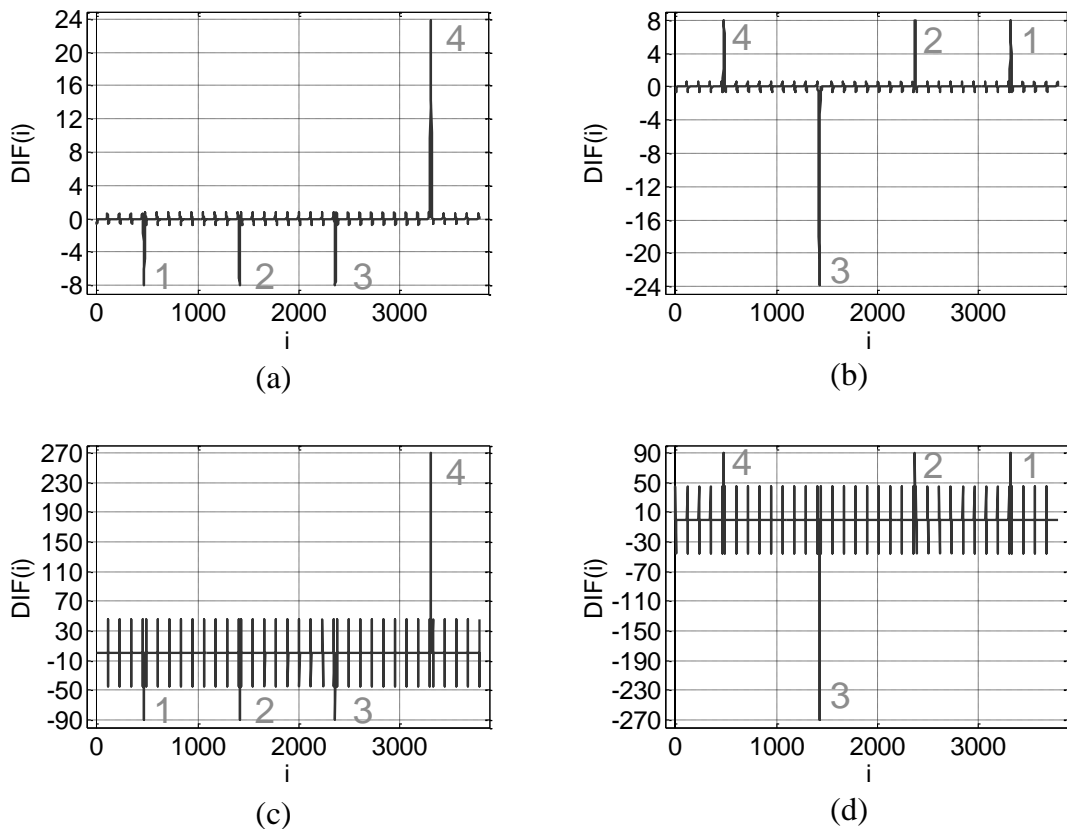
$$DIF(i) = \tilde{D}(i + 1) - \tilde{D}(i), i = 0, 1, \dots, L - 2. \quad (5.17)$$

Veličina vektora $DIF(i)$ je za jednu točku manja od vektora $\tilde{D}(i)$. $DIF(i)$ naglašava promjene u filtriranom smjeru konture. Slika 5.17 prikazuje primjer grafa $DIF(i)$ za sintetički generiranu binarnu sliku sirove pločice s kutom rotacije $\gamma = 0.5^\circ$, v. sliku 5.8. Na slici 5.17 (a) i (b) prikazan je primjer $DIF(i)$ prema formuli (5.17), a na slici 5.17 (c) i (d) prikazan je primjer $DIF(i)$ primijenjenog na vektoru $\tilde{D}(i)$. Četiri skoka u $DIF(i)$ grafu na slici 5.17 (a) i (b) označena s sivim brojevima od 1 do 4 predstavljaju promjene u $\tilde{D}(i)$ nastale zbog uglova pločice. Ako je postupak praćenja konture počeo na desnom rubu sirove pločice tada na slici 5.17 (a) i (b) 1 označava gornji desni ugao, 2 označava gornji lijevi ugao, 3 označava donji lijevi ugao i 4 označava donji desni ugao sirove pločice. Skok označen s 4 na slici 5.17 (a) i s 3 na slici 5.17 (b) značajno je veći od ostalih označenih skokova. Navedeni skok nastaje zbog promjene u filtriranom smjeru konture s -270° na 0° ako je početna vrijednost smjera konture $D(0) = 0^\circ$, ili zbog promjene u filtriranom smjeru konture s 270° na 0° ako je $D(0) = 180^\circ$, v. sliku 5.12 (a) i (b) za $w = 11$. Ostale neoznačene promjene u grafu $DIF(i)$ rezultat su nesavršenosti linije piksela ruba kako je objašnjeno u prethodnom poglavlju. S grafova na slici 5.17 (a) i (b) vidljivo je da su skokovi uzrokovani promjenama u filtriranom smjeru konture na uglovima sirove pločice nekoliko puta veći u odnosu na skokove nastale zbog nesavršenosti linije piksela ruba. Za primjer na slici 5.17 (a) maksimalna apsolutna vrijednost skoka označenog s 1 je 8.02° , a maksimalna apsolutna vrijednost skokova nastalih zbog nesavršenosti linije piksela ruba je 0.64° tj. 12.5 puta je veća vrijednost skoka koji predstavlja ugao sirove pločice. Slika 5.17 (c) i (d) prikazuje primjer grafa vektora $DIF(i)$ izračunatog na temelju nefiltriranog vektora smjera konture $D(i)$. Sa slike 5.17 (c) i (d) je vidljivo da su razlike između skokova nastalih zbog uglova sirove pločice i skokova nastalih

zbog nesavršenosti linije piksela ruba manje. Za primjer na slici 5.17 (c) maksimalna apsolutna vrijednost skoka označenog s 1 je 90° , a maksimalna apsolutna vrijednost skokova nastalih zbog nesavršenosti linije piksela ruba je 45° tj. 2 puta je veća vrijednost skoka koji predstavlja ugao sirove pločice. Može se zaključiti da je postupak filtriranja vektora smjera konture naglasio skokove u vektoru $DIF(i)$ nastale zbog uglova sirove pločice tj. povećao razliku u vrijednostima između skokova nastalih zbog uglova sirove pločice i skokova nastalih zbog nesavršenosti linije piksela ruba.

5.5.2. Detekcija neispravnosti na rubovima sirove keramičke pločice

Nakon izračuna vektora $DIF(i)$ postupak detekcije neispravnosti na rubovima i uglovima sirovih pločica se nastavlja s primjenom dvorazinskog praga. Prvi prag T_R primjenjuje se za detekciju i lokalizaciju uglova sirove pločice. Sve točke vektora $DIF(i)$ koje ispunjavaju uvjet



Slika 5.17: Graf vektora $DIF(i)$: (a) $DIF(i)$ prema (5.17) za $w = 11$, $N = 15$, $\sigma = 2.5$ i $D(0) = 0^\circ$, (b) $DIF(i)$ prema (5.17) za $w = 11$, $N = 15$, $\sigma = 2.5$ i $D(0) = 180^\circ$, (c) $DIF(i)$ primijenjen na $D(i)$ za $D(0) = 0^\circ$ i (d) $DIF(i)$ primijenjen na $D(i)$ za $D(0) = 180^\circ$.

$$DIF(i) > T_R \quad \text{ili} \quad DIF(i) < -T_R \quad (5.18)$$

predstavljaju mogući ugao sirove pločice. Sve točke koje ispunjavaju uvjet (5.18) i čije pozicije se razlikuju za 1 predstavljaju jednu grupu točaka $e_k(i)$, $k = 1, 2, \dots, Q$. Grupa točaka e_k može predstavljati ugao sirove pločice ili neispravnost na rubu. Budući da se u ovom radu razmatraju pravokutne pločice, očekivani broj grupa točaka $Q = 4$. Ako se tijekom ispitivanja vektora $DIF(i)$ pronade više ili manje grupa točaka od 4 tj. $4 < Q < 3$ pretpostavlja se da postoji neispravnost na rubu sirove pločice. Postupak detekcije neispravnosti na rubu se izvodi prvo za $D(0) = 0^\circ$, a zatim za $D(0) = 180^\circ$. Ako je $4 < Q < 3$ za $D(0) = 0^\circ$ otkrivene su neispravnosti na rubu te se broj neispravnosti e_R odredi prema (5.19) i prekida se postupak detekcije neispravnosti na rubovima i uglovima sirove pločice, u suprotnom postupak detekcije neispravnosti se nastavlja za $D(0) = 180^\circ$. Ako je $4 < Q < 3$ za $D(0) = 180^\circ$ otkrivene su neispravnosti na rubu te se broj neispravnosti e_R odredi prema (5.19).

$$e_R = |4 - Q| \quad (5.19)$$

5.5.3. Lokalizacija uglova i rubnih neispravnosti na sirovim keramičkim pločicama

Za svaku grupu točaka e_k također se pronade element $i_{max,k}$ koji predstavlja maksimalnu $DIF(i)$ vrijednost u grupi točaka e_k , tj. $i_{max,k} = \arg \max_i (e_k(i))$. Na temelju $i_{max,k}$ i vektora konture $C(i)$ omogućena je lokalizacija uglova sirove pločice i lokalizacija neispravnosti na rubovima. Koordinate (x_k, y_k) piksela na slici sirove pločice pronađenih Q uglova/neispravnosti određenu su prema (5.20).

$$(x_k, y_k) = C(i_{max,k}) \quad (5.20)$$

Na slici 5.18 prikazan je primjer detekcije grupa točaka e_k . Na slici 5.18 (a) i 5.18 (b) detektirane su $Q = 4$ grupe točaka e_1 do e_4 te se smatra da ne postoji neispravnost na rubovima sirove pločice tj. prema (5.19) nakon provedene detekcije neispravnosti za $D(0) = 0^\circ$ i za $D(0) = 180^\circ$ $e_R = 0$. Na slici 5.18 (c) i (d) detektirane su $Q = 6$ grupa točaka e_1 do e_6 te se smatra da postoje neispravnosti na rubovima sirove pločice, tj. $e_R = 2$. Za slučaj sa slike 5.18 (c) nije potrebno ponavljati postupak detekcije neispravnosti na rubovima za $D(0) = 180^\circ$. S pronađenim $i_{max,k}$ (na slici 5.18 prikazana sa crvenom točkom) i vektorom

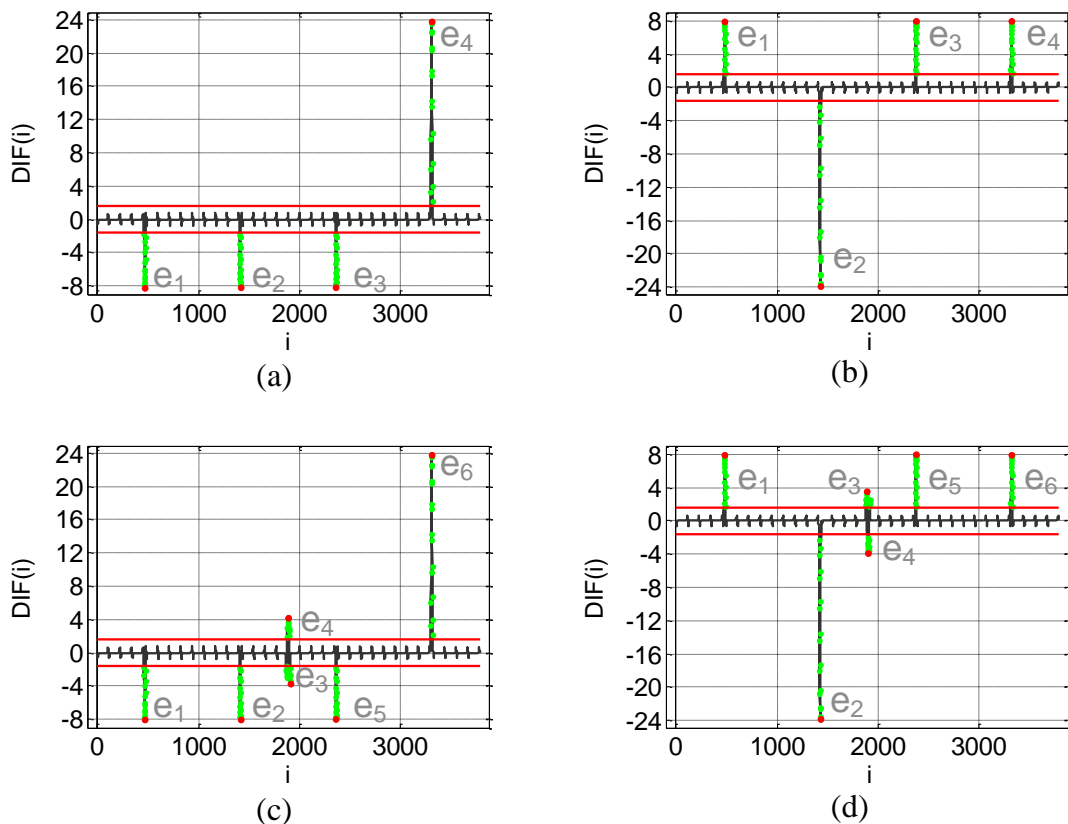
konture $C(i)$ moguće je lokalizirati i odrediti koordinate uglova/neispravnosti na slici sirove pločice prema (5.20).

5.5.4. Detekcija i lokalizacija neispravnosti na uglovima sirove keramičke pločice

Ako uvjet (5.18) ispunjavaju četiri grupe točaka $e_k(i)$, $k = \{1, 2, 3, 4\}$ i pronađeni su $i_{max,k}$ elementi primjenjuje se drugi prag. Drugi prag T_U upotrebljava se za detekciju neispravnosti na uglovima sirove pločice. Za svaku grupu točaka e_k ispituje se uvjet

$$e_k(i) > T_U \quad \text{ili} \quad e_k(i) < -T_U. \quad (5.21)$$

Sve točke koje ispunjavaju uvjet (5.21) i čije pozicije se razlikuju za 1 predstavljaju jednu grupu točaka $c_{l,k}(j)$, $l = 1, 2, \dots, Z_k$. Ako za grupu točaka e_k ne postoji niti jedna točka koja



Slika 5.18: Primjer detektiranih grupa točaka e_k (označeno sa zelenim točkama) za primjer sa slike 5.8 s početnim smjerom konture (a) $D(0) = 0^\circ$, (b) $D(0) = 180^\circ$ i primjer sa slike 5.11 s početnim smjerom konture (c) $D(0) = 0^\circ$, (d) $D(0) = 180^\circ$ (crvena linija predstavlja $T_R = 1.6^\circ$, crvene točke označavaju $i_{max,k}$).

Tablica 5.3: Određivanje vrijednosti funkcije $E_U(Z_k^{0^\circ}, Z_{5-k}^{180^\circ})$.

| $Z_k^{0^\circ}$ | $Z_{5-k}^{180^\circ}$ | $E_U(Z_k^{0^\circ}, Z_{5-k}^{180^\circ})$ |
|-----------------|-----------------------|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| >1 | ≥ 0 | 1 |
| ≥ 0 | >1 | 1 |

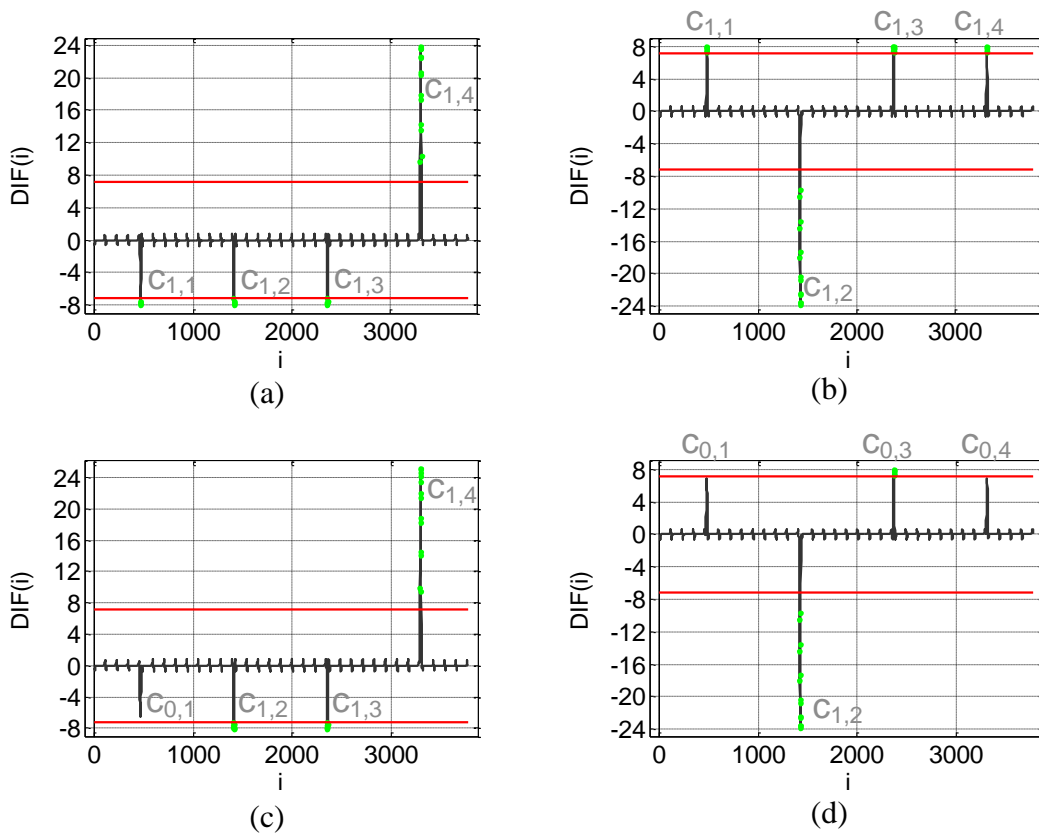
ispunjava uvjet (5.21) k ugao se smatra neispravnim i označava se sa $c_{0,k}$, tj. $Z_k = 0$. Ako za grupu točaka e_k postoji jedna grupa točaka $c_{1,k}$ ugao k se smatra ispravnim, a ako za grupu točaka e_k postoji više grupa točaka $c_{l,k}$ ugao k je neispravan. tj. za $Z_k = 1$ k -ti ugao se smatra ispravnim, a za $Z_k = 0$ ili $Z_k > 1$ k -ti ugao se smatra neispravnim. Postupak ispitivanja uglova se mora izvesti i za $D(0) = 0^\circ$ i za $D(0) = 180^\circ$. Ukupni broj neispravnosti na uglovima određen je sa

$$e_U = \sum_{k=1}^4 E_U(Z_k^{0^\circ}, Z_{5-k}^{180^\circ}), \quad (5.22)$$

gdje $Z_k^{0^\circ}$ predstavlja broj grupa točaka $c_{l,k}$ za $D(0) = 0^\circ$, $Z_k^{180^\circ}$ predstavlja broj grupa točaka $c_{l,k}$ za $D(0) = 180^\circ$, a $E_U(Z_k^{0^\circ}, Z_{5-k}^{180^\circ})$ je funkcija čije rješenje je određeno prema tablici 5.3. Lokalizacija neispravnosti na uglovima omogućena je na temelju oznake neispravnog ugla k , elemenata $i_{max,k}$, vektora konture $C(i)$ i izraza (5.20). Na slici 5.17 (a) vidljivo je da skok označen s 4 značajno je veći od skokova 1, 2, i 3 te se može zaključiti da se za detekciju neispravnosti na uglu 4 ne može upotrebljavati prag T_U . Usporedbom slika 5.17 (a) i 5.17 (b) vidljivo je da se veći skok ne nalazi na istom uglu. U prvom slučaju se nalazi na uglu 4, a u drugom slučaju se nalazi na uglu 3. Na ovaj način moguće je koristiti prag T_U za detekciju neispravnosti na svim uglovima sirove pločice uz uvjet da se postupak praćenja konture i određivanje smjera konture izvede i za $D(0) = 0^\circ$ i za $D(0) = 180^\circ$.

5.5.5. Primjeri detekcije i lokalizacije neispravnosti na rubovima i uglovima sirove keramičke pločice

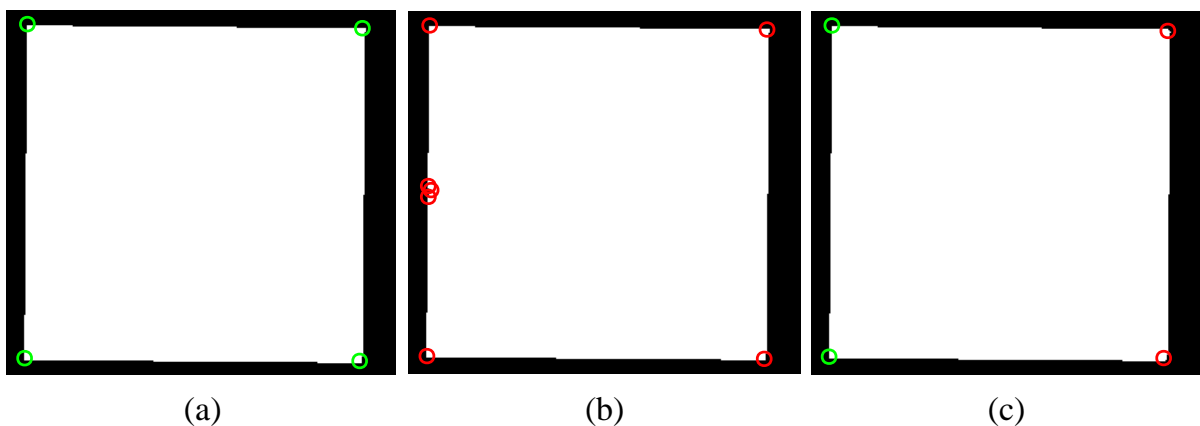
Na slici 5.19 prikazan je primjer detekcije neispravnosti na uglovima sirove pločice. Na slici 5.19 (a) i (b) za svaku grupu točaka e_k detektirana je jedna grupa točaka $c_{l,k}$ koja ispunjava uvjet (5.21) označene sa $c_{1,1}$, $c_{1,2}$, $c_{1,3}$, i $c_{1,4}$. Budući da je za svaki e_k broj grupa točaka $Z_k=1$ ne postoje neispravnosti na uglovima sirove pločice tj. prema formuli (5.22) i tablici 5.3 $e_U = 0$. Na slici 5.19 (c) i (d) prikazan je primjer detekcije neispravnosti na sintetički generiranoj slici s neispravnim gornjim desnim uglom i donjim desnim uglom (prema slici 5.17 ugao označen s 1 i 4). Na slici 5.19 (c) je vidljivo da nije pronađena niti jedna točka koja ispunjava uvjet (5.21) za grupu točaka e_1 , za ostale grupe točaka e_2 do e_4 detektirana je po jedna grupa točaka $c_{l,k}$. Metoda je uspješno detektirala jednu neispravnost na gornjem lijevom uglu sirove pločice $Z_1^0 = 0$. Iako postoji neispravnost na donjem lijevom



Slika 5.19: Primjer detektiranih grupa točaka $c_{l,k}$ (označeno sa zelenim točkama) za primjer sa slike 5.8 s početnim smjerom konture (a) $D(0) = 0^\circ$, (b) $D(0) = 180^\circ$ i za primjer sintetički generirane binarne slike s neispravnostima na dva ugla sa početnim smjerom konture (c) $D(0) = 0^\circ$, (d) $D(0) = 180^\circ$ (crvena linija predstavlja $T_U = 7.2^\circ$).

uglu sirove pločice ista nije uspješno detektirana tj. pronađena je grupa točaka $c_{1,4}$ kako je i vidljivo na slici 5.19 (c). Neispravnost nije detektirana zbog promjene u filtriranom smjeru konture s -270° na 0° kako je objašnjeno u poglavlju 5.5.4. Rješenje navedenog problema je ponoviti postupak praćenja konture za $D(0) = 180^\circ$. Na slici 5.19 (d) nije pronađena niti jedna točka koja ispunjava uvjet (5.21) za grupu točaka e_1 i e_4 . Metoda je uspješno detektirala neispravnost na donjem lijevom uglu sirove pločice $Z_1^{180^\circ} = 0$ i neispravnost na gornjem lijevom uglu sirove pločice $Z_4^{180^\circ} = 0$. Na temelju dobivenih $Z_k^0, Z_k^{180^\circ}, k = \{1, 2, 3, 4\}$ i (5.22) ukupan broj neispravnosti na uglovima sirove pločice $e_U = 2$.

Na slici 5.20 prikazan je primjer lokalizacije uglova sirove pločice i neispravnosti na rubovima i uglovima sirove pločice. Slika 5.20 (a) prikazuje primjer lokaliziranih ispravnih uglova na sirovoj pločici, zelena kružnica predstavlja jednu pronađenu (x_k, y_k) točku prema (5.20). Na slici 5.20 (b) prikazan je primjer lokaliziranih neispravnosti na lijevom rubu sirove pločice za sintetički generiranu binarnu sliku sirove pločice prikazanu na slici 5.11. Sa slike 5.20 (b) vidljivo je da je $Q > 4$ tj. pored uglova sirove pločice lokalizirana je i neispravnost označena s tri dodatne crvene kružnice na lijevom rubu pločice. Slika 5.20 (c) prikazuje primjer lokalizacije neispravnosti na uglovima sirove keramičke pločice. Crvena kružnica označava neispravni ugao, a zelena kružnica označava ispravan ugao. Sa slike 5.20 (c) je vidljivo da je detektirana i lokalizirana neispravnost na gornjem desnom uglu i donjem desnom uglu sirove pločice, ostali uglovi su ispravni.



Slika 5.20: Primjer lokalizacije uglova i neispravnosti na uglovima i rubovima sirove pločice: (a) lokalizirani ispravni uglovi sirove pločice, (b) lokalizirani uglovi i neispravnosti na sirovoj pločici, (c) lokalizirane neispravnosti na dva ugla (crvene kružnice) sirove pločice.

5.5.6. Sekvencijalni algoritam metode za detekciju i lokalizaciju neispravnosti na rubovima i uglovima sirove keramičke pločice

Formalni sekvencijalni algoritam predložene metode za detekciju i lokalizaciju neispravnosti na rubovima i uglovima sirovih pločica prikazan je u algoritmu 5.3. Rezultat algoritma 5.3 su broj pronađenih neispravnosti na rubovima e_R , broj pronađenih neispravnosti na uglovima e_U i lokacije pronađenih uglova/neispravnosti prema (5.20). U algoritmu 5.3 oznake s eksponentom 0° predstavljaju vrijednosti varijabli za $D(0) = 0^\circ$, a oznake s eksponentom 180° predstavljaju vrijednosti varijabli za $D(0) = 180^\circ$. Budući da se postupak detekcije neispravnosti na rubovima i uglovima sirovih pločica ponavlja za $D(0) = 0^\circ$ i $D(0) = 180^\circ$ moguća je paralelizacija algoritma 5.3 na paralelnim platformama.

5.6. Paralelni algoritam metode za detekciju i lokalizaciju neispravnosti na rubovima i uglovima sirove keramičke pločice

Za paralelizaciju predložene metode upotrebljavaju se višeprocesorski sustavi sa zajedničkom memorijom budući da se obrađuju manje količine podataka (reda veličine 10 MB) i potrebna je velika brzina prijenosa podataka između procesora (jezgri). Za paralelnu implementaciju predložene metode upotrebljavaju se višejezgreni procesori i C++ programski jezik s OpenMP bibliotekama za paralelno programiranje.

OpenMP predstavlja standard za programiranje paralelnih platformi sa zajedničkom

Algoritam 5.3: Pseudo kod algoritma za detekciju i lokalizaciju neispravnosti na rubovima i uglovima sirovih pločica.

```

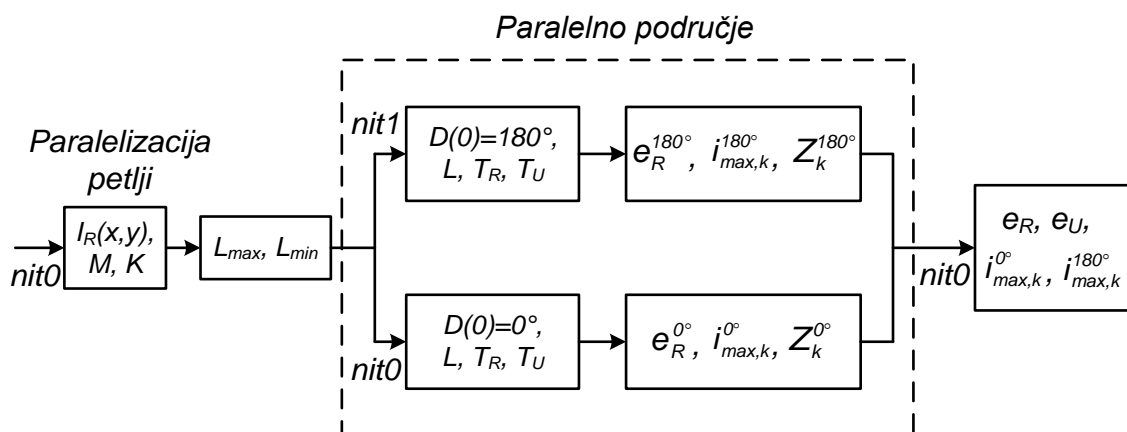
1. Ulaz:  $\bar{D}^{0^\circ}(i), \bar{D}^{180^\circ}(i), L, T_R, T_U, e_R = 0, e_U = 0$ 
2. Izračunaj:  $DIF^{0^\circ}(i), DIF^{180^\circ}(i)$  prema 5.17
3. Pronađi:  $e_k^{0^\circ}, e_k^{180^\circ}, i_{max,k}^{0^\circ}, i_{max,k}^{180^\circ}, Q^{0^\circ}, Q^{180^\circ}$ 
4. if  $Q^{0^\circ} \neq 4$  then
5.      $e_R = |4 - Q^{0^\circ}|$ 
6.     Lokaliziraj uglove/neispravnosti prema (5.20)  $i i_{max,k}^{0^\circ}$ 
7.     return
8. else if  $Q^{180^\circ} \neq 4$  then
9.      $e_R = |4 - Q^{180^\circ}|$ 
10.    Lokaliziraj uglove/neispravnosti prema (5.20)  $i i_{max,k}^{180^\circ}$ 
11.    return
12. end if
13. Pronađi:  $c_{l,k}^{0^\circ}$  i  $Z_k^{0^\circ}$ 
14. Pronađi:  $c_{l,k}^{180^\circ}$  i  $Z_k^{180^\circ}$ 
15. Odredi  $e_U$  prema (5.22)
16. Lokaliziraj ispravne/neispravne uglove prema (5.20),  $Z_k^{0^\circ}, Z_k^{180^\circ}$   $i i_{max,k}^{180^\circ}$ 

```

memorijom. Podržan je od strane C, C++ i Fortran programskih jezika. Najznačajnija prednost upotrebe OpenMP-a je jednostavnost korištenja. Jednostavnost se postiže na način da se već postojeći dijelovi programskog koda paraleliziraju dodavanjem OpenMP *#pragma* smjernicama i OpenMP funkcija, tj. nije potrebno mijenjati izvorni serijski kod. U OpenMP-u postoje dva pristupa paralelizacije programskog koda: paralelizacija na razini petlji (*engl. loop-level*) i paralelizacija područja (*engl. parallel regions*) [89]. Za postupak paralelizacije metode za detekciju i lokalizaciju neispravnosti na rubovima i uglovima sirove pločice upotrebljavaju se oba pristupa.

Predloženi postupak paralelizacije s OpenMP bibliotekama prikazan je s dijagramom na slici 5.21. Izvođenje algoritma počinje s glavnom niti *nit0*. Postupak paralelizacije na razini petlji upotrebljava se za izračun slike ruba $I_R(x, y)$. S OpenMP smjernicom *parallel for* paralelizirana je prva for petlja algoritma 5.1. Nakon izračuna $I_R(x, y)$ *nit0* prosljeđuje podatke OpenMP paralelnom području. Paralelno područje se sastoji od dvije niti (*engl. threads*) *nit0* i *nit1* koje se izvode na dvije jezgre višejezgrenog procesora. Prva nit (*nit0*) izvodi postupak praćenja konture i detekcije i lokalizacije neispravnosti na rubovima i uglovima sirove pločice za $D(0) = 0^\circ$, a druga nit (*nit1*) izvodi postupak praćenja konture i detekcije i lokalizacije neispravnosti na rubovima i uglovima sirove pločice za $D(0) = 180^\circ$ kako je i prikazano na slici 5.21. Sa završetkom paralelnog područja izvođenje algoritma nastavlja samo nit *nit0*. Konačni rezultat nastaje spajanjem dobivenih rezultata od pojedinih niti na niti *nit0* na temelju algoritma 5.3.

Iz prikazanih primjera vidljivo je da je metoda pogodna za detekciju neispravnosti na uglovima i rubovima sirovih pločica. U nastavku rada prikazani su eksperimentalni rezultati



Slika 5.21: Dijagram paralelizacije predložene metode za detekciju i lokalizaciju neispravnosti na rubovima i uglovima sirovih pločica s OpenMP bibliotekama.

uspješnosti detekcije neispravnosti na uglovima i rubovima sirovih pločica u laboratoriju i industrijskim uvjetima te rezultati mjerenja vremena izvođenja predložene metode.

5.7. Eksperimentalni rezultati

Testiranje predložene metode se sastoji od dva dijela. Prvi dio predstavlja testiranje metode na mogućnost detekcije neispravnosti na rubovima i uglovima sirovih pločice za:

- sintetički generirane binarne slike sirovih pločica,
- sive slike sirovih pločica prikupljenih u laboratoriju,
- sive slike sirovih pločica prikupljenih u industrijskim uvjetima.

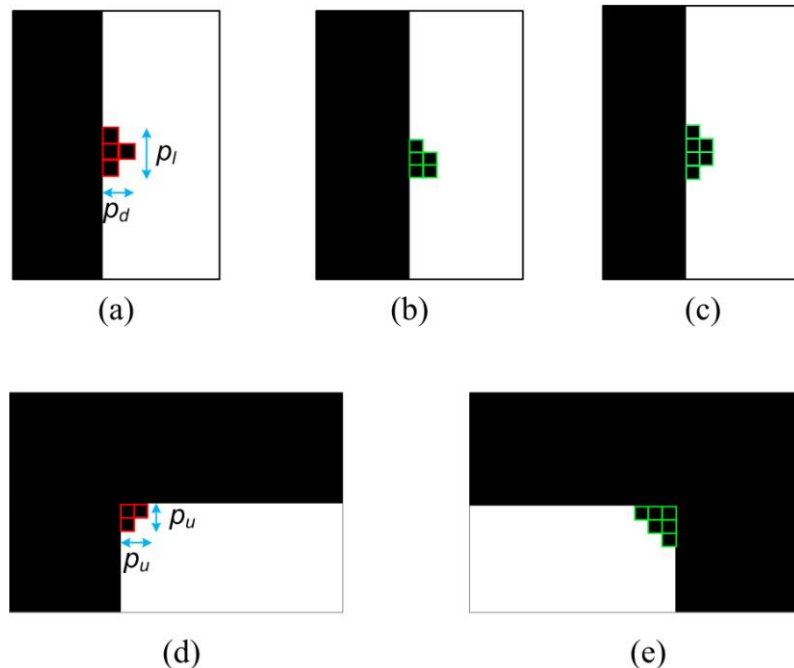
Drugi dio obuhvaća testiranje predložene metode na rad u stvarnom vremenu na slikama različitih veličina. U nastavku su detaljno objašnjeni pojedina načini testiranja predložene metode. Iako se predložena metoda temelji na metodi u [41] nije se moguće usporediti sa istom budući da u navedenom radu nije predložen algoritam za detekciju – lokalizaciju neispravnosti na temelju dobivene razlike smjera konture referentne i testirane slike sirove pločice.

5.7.1. Mogućnost detekcije neispravnosti na rubovima i uglovima sirove keramičke pločice

Prvi test je izveden nad sintetički generiranim binarnim slikama sirove pločice. Test analizira mogućnost detekcije neispravnosti u pogledu veličine neispravnosti u pikselima na rubovima i uglovima sirovih pločica. Prva kreirana binarna slika $I_{B,ref}(x, y)$ veličine 1100×1100 piksela predstavlja sliku sirove pločice veličine 1000×1000 piksela sa savršenim linijama ruba i kutom $\gamma = 0.0^\circ$. Na temelju $I_{B,ref}(x, y)$ određene su vrijednosti T_R i T_U na način da je $I_{B,ref}(x, y)$ rotirana u intervalu $\gamma = \{0.0^\circ, 0.1^\circ, \dots, 1.0^\circ\}$ s GIMP programom za obradu slike. Za sve rotirane $I_{B,ref}(x, y)$ određena je minimalna vrijednost T_R za koji je $e_R = 0$, i maksimalna vrijednost T_U za koji je $e_U = 0$, za iste slike također su određeni i L_{max} i L_{min} . Iz provedenog eksperimenta dobivene su vrijednosti: $T_R = 0.66^\circ$, $T_U = 7.80^\circ$, $L_{max} = 4396$ i $L_{min} = 3960$. Dobivene vrijednosti upotrijebljene su za određivanje minimalne veličine neispravnosti u pikselima na rubovima i uglovima sirove pločice. Druga kreirana binarna slika $I_{B,R}(x, y)$ veličine 1100×1100 piksela predstavlja sliku sirove pločice veličine 1000×1000 piksela sa savršenim linijama ruba i kutom $\gamma = 0.0^\circ$ te tri neispravnosti na lijevom rubu. Neispravnosti su površine (p_p) četiri, pet i šest piksela kako je

i prikazano na slici 5.22 (a) do (c) sa zelenim i crvenim kvadratima. Pored p_p neispravnosti karakteriziraju i dimenzije u pikselima p_l i p_d kako je prikazano na slici 5.22 (a).

Postupak detekcije neispravnosti proveden je za širinu prozora kružnog pomičnog prosjeka $w = 11$ za koji su prema [7] dobiveni najbolji rezultati detekcije. Ostali parametri su: $L_{max} = 4396$, $L_{min} = 3960$, $M = 1100$, $K = 1100$ i broj koeficijenata digitalnog Gaussovog filtera $N = 15$. Neispravnost s $p_p = 4$, $p_l = 3$ i $p_d = 2$ nije uspješno detektirana, označeno sa crvenim kvadratima na slici 5.22 (a). Najmanja neispravnost koja je uspješno detektirana za $\gamma = \{0.0^\circ, 0.1^\circ, \dots, 1.0^\circ\}$ je za $p_p = 5$, $p_l = 3$ i $p_d = 2$ kako je i prikazano na slici 5.22 (b) sa zelenim kvadratima. Isto vrijedi i za neispravnost s $p_p = 5$, $p_l = 2$ i $p_d = 3$. Također je moguće detektirati neispravnosti s dimenzijama i površinom većim od dimenzija i površine najmanje uspješno detektirane neispravnosti, primjer prikazan na slici 5.22 (c). Na temelju provedenog eksperimenta može se zaključiti da je s predloženom metodom moguće detektirati neispravnosti koje ispunjavaju uvjet (5.23). Ako se pretpostavi da je rezolucija $I_{B,R}(x, y)$ slike $r = 5$ piksela/mm proizlazi da su dimenzije najmanje neispravnosti koju je moguće detektirati $l = 0.6$ mm, $d = 0.4$ mm i površina $P = 0.2$ mm², prema (5.24). Detekcija neispravnosti na rubovima s manjim dimenzijama i površinom također je moguća ali je potrebno povećati rezoluciju slike r .



Slika 5.22: Generirane neispravnosti na rubovima sirove pločice sa: $p_p = 4$, $p_l = 3$ i $p_d = 2$ (a), (b) $p_p = 5$, $p_l = 3$ i $p_d = 2$ (c) $p_p = 6$, $p_l = 4$ i $p_d = 2$. Generirane neispravnosti na uglovima sirove pločice sa: $p_u = 2$ (d) i (e) $p_u = 3$.

$$e_R \neq 0 \quad \text{ako} \quad \begin{cases} p_p \geq 5 \text{ i } p_l \geq 3 \text{ i } p_d \geq 2 & \text{ili} \\ p_p \geq 5 \text{ i } p_l \geq 2 \text{ i } p_d \geq 3 \end{cases} \quad (5.23)$$

$$l = \frac{p_l}{r}, \quad d = \frac{p_d}{r}, \quad P = \frac{p_p}{r^2} \quad (5.24)$$

Treća kreirana binarna slika $I_{B,U}(x,y)$ veličine 1100×1100 piksela predstavlja sliku sirove pločice veličine 1000×1000 piksela sa savršenim linijama ruba i kutom $\gamma = 0.0^\circ$ te dvije neispravnosti na uglovima. Neispravnosti se nalaze na gornjem desnom i gornjem lijevom uglu površine $p_p = 3$ i $p_p = 6$ kako je i prikazano na slici 5.22 (d) i (e) sa zelenim i crvenim kvadratima. Neispravnosti na uglovima također karakteriziraju i dimenzija u pikselima p_u kako je prikazano na slici 5.22 (d). Neispravnost ugla sa $p_u = 2$ nije uspješno detektirana, označeno sa crvenim kvadratima na slici 5.22 (d). Najmanja neispravnost na uglu koja je uspješno detektirana za $\gamma = \{0.0^\circ, 0.1^\circ, \dots, 1.0^\circ\}$ je s dimenzijom $p_u = 2$ i površinom piksela $p_p = 6$ kako je i prikazano na slici 5.22 (e) sa zelenim kvadratima. Također je moguće detektirati neispravnosti na uglovima s dimenzijom i površinom većom od dimenzije i površine najmanje uspješno detektirane neispravnosti. Na temelju provedenog eksperimenta može se zaključiti da je s predloženom metodom moguće detektirati neispravnosti na uglovima koje ispunjavaju uvjet (5.25). Uz pretpostavku da je rezolucija $I_{B,U}(x,y)$ slike $r = 5$ piksela/mm proizlazi da su dimenzije prema (5.24) najmanje neispravnosti na uglu koju je moguće detektirati $l_u = 0.6$ mm i površina $P_u = 0.24$ mm². Kako je i objašnjeno za detekciju neispravnosti na rubovima, moguća je i detekcija neispravnosti uglova s manjom dimenzijom i površinom ali je potrebno povećati rezoluciju slike r .

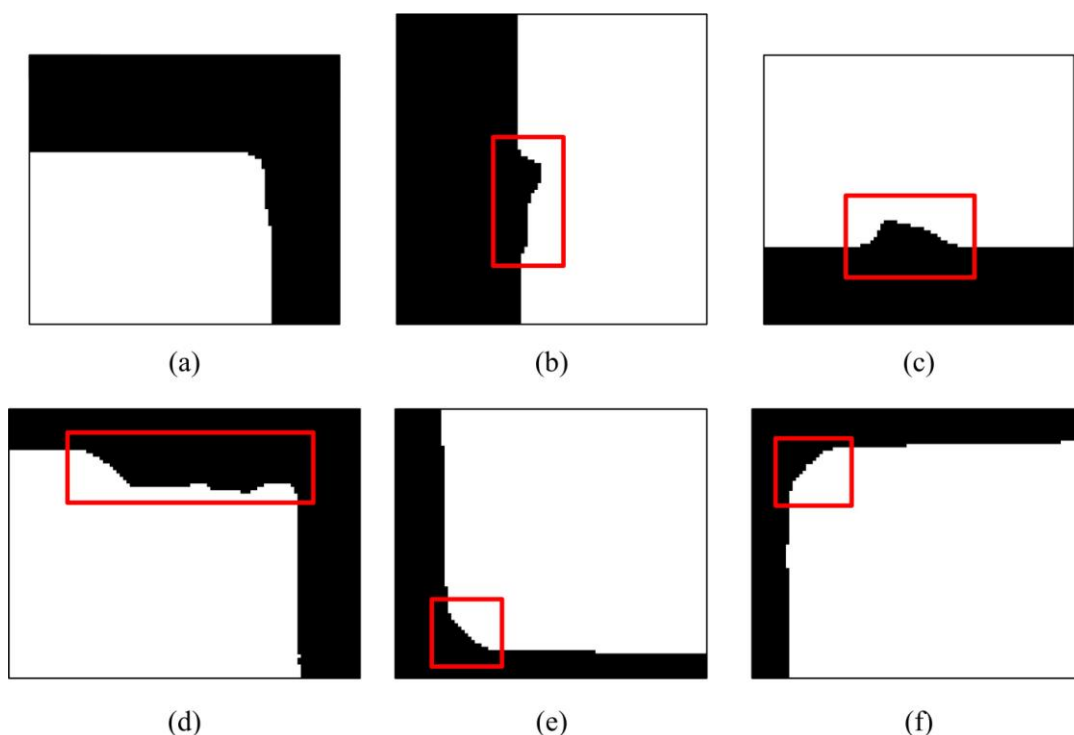
$$e_U \neq 0 \quad \text{ako} \quad p_p \geq 6 \text{ i } p_u \geq 3 \quad (5.25)$$

$$l_u = \frac{p_u}{r}, \quad P_u = \frac{p_p}{r^2} \quad (5.26)$$

Drugi test je izveden nad slikama sirovih pločica dobivenih sa prototipom sustava za detekciju neispravnosti na sirovim pločica u laboratoriju, v. poglavlje 3.2. Test analizira uspješnost detekcije neispravnosti na rubovima i uglovima sirovih pločica prikupljenih u laboratorijskim uvjetima. Upotrijebljene su bijele jednoboje sirove pločice veličine 20 cm \times 20 cm. Slike su veličine 1038×1204 piksela sa sirovom pločicom veličine 980×980 piksela iz čega proizlazi rezolucija $r = 4.9$ piksela/mm. Sive slike sirovih pločica binarizirane su s PBSCD metodom nakon čega je primijenjen postupak za detekciju neispravnosti na rubovima i uglovima sirovih pločica. Vrijednosti parametara T_R , T_U , L_{max} i

L_{min} određene su na temelju 10 referentnih pločica bez neispravnosti na rubovima i uglovima s kutom rotacije u intervalu $\gamma = [-1.0^\circ, 1.0^\circ]$. Dobivene vrijednosti su: $T_R = 1.31^\circ$, $T_U = 7.11^\circ$, $L_{max} = 4480$ i $L_{min} = 3852$.

Uspješnost detekcije neispravnosti na rubovima i uglovima određena je na temelju 100 slika testnih sirovih pločica. Sirove pločice su rotirane i translahirane na pokretnoj liniji. Od 100 testnih pločica 10 nije imalo niti jednu neispravnost na rubovima ili uglovima, a 90 pločica je imalo jednu ili više neispravnosti na rubovima ili uglovima sirovih pločica. Neispravnosti na rubovima i uglovima sirovih pločica kreirane su od strane operatera s različitim dimenzijama i površinom. Na slici 5.23 (a) prikazan je jedan primjer ispravnog ugla/ruba testnih slika nakon binarizacije s PBSCD metodom, a na slikama 5.23 (b) do (f) nekoliko primjera neispravnosti na uglu/rubu za testne slike. Uspješnost detekcije neispravnosti dobivena je usporedbom rezultata dobivenih od predložene metode s rezultatima dobivenih od eksperta za kontrolu kvalitete keramičkih pločica. U tablici 5.4 prikazani su rezultati detekcije neispravnosti na rubovima i uglovima sirovih pločica u laboratorijskim uvjetima. Od ukupno 206 neispravnosti uspješno je pronađeno 200 neispravnosti, tj. 97.09 %. Nije detektirana niti jedna dodatna neispravnost, tj. niti jedan



Slika 5.23: Primjeri neispravnosti nakon binarizacije slike sirove pločice prikupljene u laboratorijskim uvjetima s PBSCD metodom: (a) ispravan ugao i rub, (b) do (d) neispravnosti na rubovima, (e) i (f) neispravnosti na uglovima.

Tablica 5.4: Rezultati detekcije neispravnosti na rubovima i uglovima sirovih pločica u laboratorijskim uvjetima za 100 slika sirovih pločica.

| | <i>Ukupno neispravnosti</i> | <i>Detektirane neispravnosti</i> | <i>Dodatno detektirane neispravnosti</i> |
|----------|-----------------------------|----------------------------------|--|
| <i>n</i> | 206 | 200 | 0 |
| <i>%</i> | 100.00 | 97.09 | 0 |

ispravni rub nije deklariran kao neispravan i niti jedan ispravni ugao nije deklariran kao neispravan. Iz navedenog proizlazi da metoda nije detektirala nepostojeće neispravnosti i time ne bi proglasila škartom ispravne sirove pločice. Šest neispravnosti za koje vrijedi $l < 1 \text{ mm}$, $d = 1 \text{ mm}$ i $l_u < 1 \text{ mm}$ nije uspješno detektirano zbog nesavršenosti sustava za prikupljanja slike (geometrijska distorzija leće, osvjetljenost objekta). Iako neispravnosti ispunjavaju uvjete (5.23) i (5.25), zbog nesavršenosti sustava za prikupljanja slike neispravnosti nisu proizvele dovoljno velike skokove u $DIF(i)$ vektoru da bi ih bilo moguće detektirati s postavljenim T_R i T_U . Navedeni problem je moguće riješiti prvenstveno s upotrebom kvalitetnije leće, a zatim i s upotrebom kvalitetnije kamere s jednakom ili većom rezolucijom r . Također je moguće povećati mogućnost detekcije neispravnosti s ograničenjem kuta rotacije γ npr. u intervalu $[-0.5^\circ, 0.5^\circ]$.

Treći test je izveden nad slikama sirovih pločica dobivenih s prototipom sustava za detekciju neispravnosti na sirovim pločica na proizvodnoj liniji u tvornici keramičkih pločica Keramika Modus d.o.o. [14]. Test analizira uspješnost detekcije neispravnosti na rubovima i uglovima sirovih pločica prikupljenih u industrijskim uvjetima. Upotrijebljene su bijele jednoboje sirove pločice veličine $20 \text{ cm} \times 20 \text{ cm}$. Prikupljene slike su veličine 1038×1098 piksela sa sirovom pločicom veličine 870×870 piksela iz čega proizlazi rezolucija $r = 4.35 \text{ piksela/mm}$. Kao i u drugom testu, sive slike sirovih pločica binarizirane su s PBSCD metodom nakon čega je primijenjen postupak za detekciju neispravnosti na rubovima i uglovima sirovih pločica. Vrijednosti parametara T_R , T_U , L_{max} i L_{min} određene su na temelju 50 referentnih pločica bez neispravnosti na rubovima i uglovima s kutom rotacije u intervalu $\gamma = [-0.7^\circ, 0.7^\circ]$. Dobivene vrijednosti su: $T_R = 1.31^\circ$, $T_U = 6.55^\circ$, $L_{max} = 4268$ i $L_{min} = 3449$.

Uspješnost detekcije neispravnosti na rubovima i uglovima sirovih pločica u industrijskim uvjetima određena je na temelju 1000 slika sirovih pločica. Rotacija sirovih pločica je djelomično korigirana s usmjerivačima koji se nalaze na proizvodnoj liniji. Upotrebom usmjerivača kut rotacije je ograničen na interval $\gamma = [-0.7^\circ, 0.7^\circ]$. Od 1000

Tablica 5.5: Rezultati detekcije neispravnosti na rubovima i uglovima sirovih pločica u industrijskim uvjetima za 1000 slika sirovih pločica.

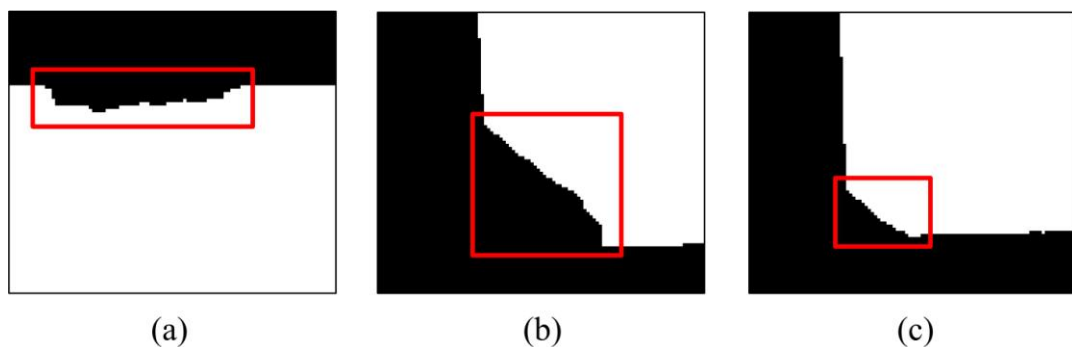
| | <i>Ukupno neispravnosti</i> | <i>Detektirane neispravnosti</i> | <i>Dodatno detektirane neispravnosti</i> |
|----------|-----------------------------|----------------------------------|--|
| <i>n</i> | 10 | 10 | 0 |
| <i>%</i> | 100.00 | 100.00 | 0 |

testnih pločica 990 nije imalo niti jednu neispravnost na rubovima/uglovima, a 10 pločica je imalo jednu neispravnost na rubu ili uglu. Na slici 5.24 (a) do (c) prikazano je nekoliko primjera neispravnosti na uglu/rubu sirovih pločica nakon binarizacije s PBSCD metodom. Uspješnost detekcije neispravnosti dobivena je usporedbom rezultata dobivenih od predložene metode s rezultatima dobivenih od eksperta za kontrolu kvalitete keramičkih pločica. Dimenzije neispravnosti koje su se pojavljivale tijekom prikupljanja slika sirovih pločica na proizvodnoj liniji su: $l > 1 \text{ mm}$, $d > 1 \text{ mm}$ i $l_u > 1 \text{ mm}$.

Tablica 5.5 prikazuje rezultate detekcije neispravnosti na rubovima i uglovima sirovih pločica s proizvodne linije. Sve neispravnosti su uspješno detektirane. Niti jedna neispravnost nije dodatno detektirana, tj. niti jedna ispravna sirova pločica nije proglašena neispravnom. Rezultati u tablici 5.5 pokazuju 100 % učinkovitost u detekciji neispravnosti na rubovima i uglovima sirovih pločica u industrijskim uvjetima.

5.7.2. Rad u stvarnom vremenu

Drugi dio testiranja predložene metode sastoji se od mjerenja vremena izvođenja metode na različitim veličinama sivih slika sirovih pločica. Cilj ovog testa je analiza mogućnosti rada predložene metode u stvarnom vremenu. Prilikom testiranja mjereno je srednje vrijeme izvođenja predložene metode na temelju 100 iteracija. Testiranje mogućnosti



Slika 5.24: Primjeri neispravnosti nakon binarizacije slike sirove pločice prikupljene u industrijskim uvjetima s PBSCD metodom: (a) neispravan rub, (b) i (c) neispravnosti na uglovima.

Tablica 5.6: Veličina i duljina konture testnih slika za mjerenje srednjeg vremena izvođenja metode za detekciju neispravnosti na rubovima i uglovima sirovih pločica.

| | <i>E1</i> | <i>E2</i> | <i>E3</i> | <i>E4</i> | <i>E5</i> |
|-----------------------------|-------------|-------------|-------------|-------------|-------------|
| <i>Veličina u pikselima</i> | 1038 × 1098 | 1204 × 1274 | 1380 × 1460 | 1692 × 1790 | 1946 × 2054 |
| <i>MP</i> | 1.14 | 1.53 | 2.01 | 3.03 | 4 |
| <i>L</i> | 3473 | 4031 | 4622 | 5668 | 6515 |

rada u stvarnom vremenu sastoji se od mjerenja izvođena sekvencijalnog algoritma 5.3 i mjerenja izvođena paralelnog OpenMP algoritma prema slici 5.21.

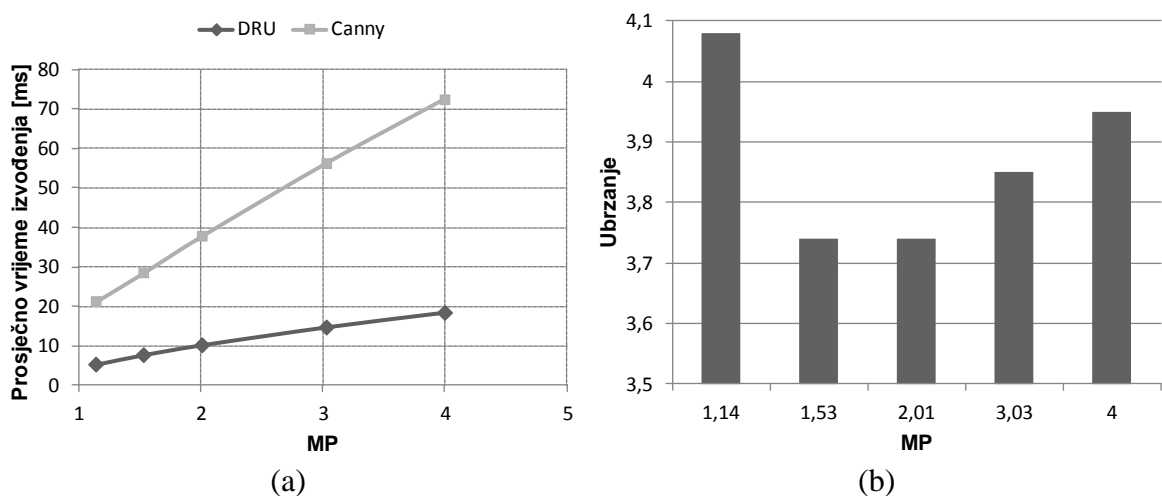
Za mjerenje srednjeg vremena izvođenja sekvencijalnog i paralelnog algoritma predložene metode upotrijebljeno je pet slika (*E1* do *E5*) različitih veličina i dužina konture *L* prema tablici 5.6. Algoritmi su implementirani u C++ programskom jeziku s OpenCV 2.2 bibliotekama [22] i izvedeni na osobnom računalu s Windows 7 x64 operacijskim sustavom, četverojezgrenim Intel Q6600 procesorom, 4 GB RAM memorije i NVIDIA GTX 670. Za C++ implementaciju upotrijebljeno je Visual Studio 2010 okruženje [74] s opcijama C++ prevoditelja:

- potpuna optimizacija po brzini izvođenja i
- brzi model broja jednostruke preciznosti.

U prvom testu mjereno je prosječno vrijeme izvođenja sekvencijalnog algoritma za pet testnih slika *E1* do *E5*. Prosječno vrijeme izvođenja predložene metode uspoređeno je s prosječnim vremenom izvođenja Canny metode za detekciju ruba koja se upotrebljava u metodi za detekciju neispravnosti na rubovima i uglovima keramičkih pločica predloženoj u [41]. Upotrijebljena je OpenCV implementacija Canny metode za detekciju ruba. Rezultati mjerenja prikazani su na slici 5.25. Iz grafa na slici 5.25 (a) vidljivo je da je vrijeme izvođenja predložene metode kraće u odnosu na vrijeme izvođenja Canny metode za detekciju ruba. Graf na slici 5.25 (b) prikazuje ubrzanje metode za detekciju neispravnosti na rubovima i uglovima sirovih pločica (DRU) u odnosu na Canny metodu. Ubrzanje je određeno prema (4.46) gdje t_{\ll} predstavlja prosječno DRU vrijeme, a t_m predstavlja prosječno vrijeme izvođenja Canny metode. Vidljivo je da je predložena metoda više od 3.7 puta brža od Canny metode. Budući da Canny metoda predstavlja samo dio postupka za detekciju neispravnosti na rubovima i uglovima keramičkih pločica predložene u [41] može se zaključiti da metoda predložena u ovom radu je najmanje 3.7 puta brža u odnosu na metodu predloženu u [41]. Na slici 5.25 (a) vidljivo je da prosječno vrijeme izvođenja predložene DRU metode linearno

ovisi o veličini slike u *MP*. U najlošijem slučaj (u pogledu izvođenja vremena) za sliku veličine 1946×2054 piksela (4 *MP*) prosječno vrijeme izvođenja je manje od 19 *ms* tj. vrijeme izvođenja je manje od 4 % ukupno dostupnog vremena za vizualnu inspekciju sirovih pločica, v. poglavlje 3.3.

U drugom testu mjereno je prosječno vrijeme izvođenja paralelnog OpenMP algoritma predložene metode za detekciju neispravnosti na rubovima i uglovima sirovih pločica. Vrijeme je mjereno za šest testnih slika prema tablici 5.6. Prvi dio paralelnog algoritma (postupak određivanja slike ruba) izveden je s četiri niti na četiri jezgre Intel Q6600 procesora. Drugi dio paralelnog algoritma izveden je s dvije niti na dvije jezgre Intel Q6600 procesora. Rezultati mjerenja prikazani su na slici 5.26. Iz grafa na slici 5.26 (a) vidljivo je da je prosječno vrijeme izvođenja paralelnog OpenMP algoritma kraće u odnosu na prosječno vrijeme izvođenja sekvencijalnog (DRU) algoritma predložene metode. Graf na slici 5.26 (b) prikazuje ubrzanje paralelnog OpenMP algoritma u odnosu na sekvencijalni algoritam metode za detekciju neispravnosti na rubovima i uglovima sirovih pločica. Ubrzanje je određeno prema (4.46) gdje t_{\ll} predstavlja prosječno vrijeme izvođenja paralelnog OpenMP algoritma, a t_m predstavlja prosječno vrijeme izvođenja sekvencijalnog algoritma. S grafa na slici 5.26 (b) vidljivo je da se za slike veće od 1.5 *MP* s paralelnim OpenMP algoritmom postižu ubrzanja od 2 puta ili više. Prosječno vrijeme izvođenja predložene metode je dodatno smanjeno. U najlošijem slučaj (u pogledu izvođenja vremena) za sliku veličine 1946×2054 piksela (4 *MP*) prosječno vrijeme izvođenja je manje od 9 *ms* tj. vrijeme izvođenja je



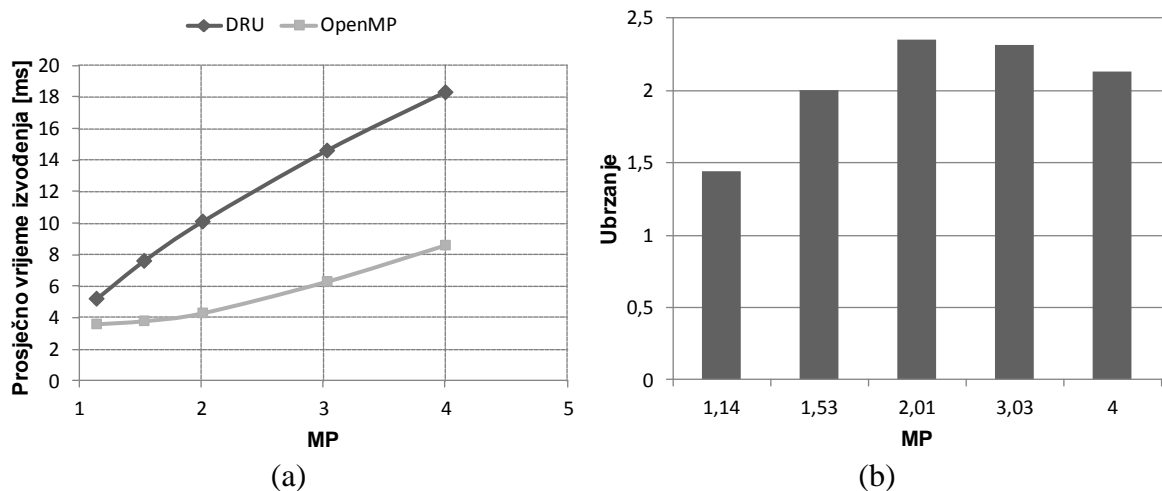
Slika 5.25: Prosječno vrijeme izvođenja u 100 iteracija sekvencijalnog algoritma predložene metode za detekciju neispravnosti na rubovima i uglovima (DRU) sirovih pločica i Canny metode za detekciju ruba (a), (b) ubrzanje predložene metode u odnosu na Canny metodu.

manje od 2 % ukupno dostupnog vremena za vizualnu inspekciju sirovih pločica kako je navedeno u poglavlju 3.3.

5.7.3. Zaključak

Iz rezultata dobivenih iz provedenih testiranja proizlaze sljedeći zaključci.

- Predložena metoda za detekciju neispravnosti na rubovima i uglovima sirovih pločica za sintetički generirane binarne slike detektira neispravnosti koje ispunjavaju uvjete (5.23) i (5.25). Za slike prikupljene u laboratorijskim uvjetima s neispravnostima različitih dimenzija i površine, metoda je uspješno detektirala više od 97 % neispravnosti. Također, nije dodatno detektirana niti jedna neispravnost, tj. niti jedna sirova pločica bez neispravnosti na rubovima ili uglovima nije proglašena neispravnom. Mogućnost detekcije neispravnosti predložene metode moguće je dodatno povećati s upotrebom kvalitetnije leće i kamere s jednakom ili većom rezolucijom r . Za slike prikupljene na proizvodnoj liniji u industrijskim uvjetima metoda je uspješno detektirala 100 % neispravnosti i niti jedna ispravna sirova pločica nije proglašena neispravnom, tj. nije detektirana niti jedna dodatna neispravnost. Iz navedenog može se zaključiti da metoda daje zadovoljavajuće rezultate detekcije neispravnosti na rubovima i uglovima sirovih pločica.
- Sekvencijalna implementacija predložene metode za detekciju neispravnosti na



Slika 5.26: Prosječno vrijeme izvođenja u 100 iteracija paralelnog OpenMP algoritma i sekvencijalnog algoritma (DRU) predložene metode za detekciju neispravnosti na rubovima i uglovima sirovih pločica (a), (b) ubrzanje paralelnog algoritma u odnosu na sekvencijalni algoritam.

rubovima i uglovima sirovih pločica daje zadovoljavajuće rezultate mjerenja vremena izvođenja. Vrijeme izvođenja u najlošijem slučaju ne prelazi 19 ms tj. vrijeme izvođenja je manje od 4 % ukupno dostupnog vremena za vizualnu inspekciju sirovih pločica. Paralelna OpenMP implementacija predložene metode dodatno snižava vrijeme izvođenja. U najlošijem slučaju vrijeme izvođenja paralelnog algoritma ne prelazi 9 ms tj. vrijeme izvođenja je manje od 2 % ukupno dostupnog vremena za vizualnu inspekciju sirovih pločica. Može se zaključiti da vrijeme izvođenja predložene metode za detekciju neispravnosti na rubovima i uglovima sirovih pločica zadovoljava postavljene vremenske uvjete za rad u stvarnom vremenu.

6. NOVA METODA ZA UJEDNAČAVANJA SLIKE POVRŠINE SIROVE KERAMIČKE PLOČICE

Ako s metodom za detekciju neispravnosti na rubovima i uglovima sirovih pločica nije detektirana niti jedna neispravnost potrebno je nastaviti analizu neispravnosti na površini sirovih pločica. Ako rezultat analize površine sirove pločice također ne pronade niti jednu neispravnost, sirova pločice može nastaviti proces proizvodnje keramičkih pločica tj. nastaviti u fazu paljenja. Većina metoda za analizu površine keramičkih pločica [90, 91, 92, 93, 94] za ulaz upotrebljavaju samo sliku površine pločice bez okoline. Tijekom transporta sirovih pločica na proizvodnoj liniji dolazi do rotacije i translacije pločica zbog čega se na ulaznoj slici objekt sirove pločice ne nalazi uvijek u istom položaju. Potrebno je ujednačiti sliku površine sirove pločice prije prosljeđivanja iste metodama za detekciju neispravnosti na površini. Ujednačavanje slike površine sirove pločice se sastoji od dva dijela: uklanjanja pozadine koja uključuje objekte pokretne linije i uklanjanja utjecaja rotacije i translacije sirove pločice.

6.1. Postojeće metode za ujednačavanje slike površine keramičke pločice

U literaturi se može pronaći nekoliko metoda koje obrađuju dio postupka za ujednačavanje slike keramičke pločice. Metode se upotrebljavaju za uklanjanje utjecaja rotacije i translacije testne slike u odnosu na referentnu sliku. Autori u [95] predstavili su postupak uklanjanja utjecaja rotacije i translacije testne slike keramičke koji se temelji na Hough-ovoj i Fourierovoj transformaciji. Postupak se sastoji od dva koraka. U Prvom koraku se s Hough-ovom transformacijom grubo odredi kut rotacije pločice i iznos translacije, nakon čega se s geometrijskom transformacijom utjecaj rotacije i translacije uklanja s ulazne testne slike. U drugom koraku se na slici iz prethodnog koraka dodatno uklanja utjecaj rotacije i translacije s korelacijom faza Fourierove transformacije testne i referentne slike. Dobivena slika se upotrebljava za detekciju neispravnosti na površini keramičke pločice s jednostavnom metodom usporedbe piksela testne i referentne slike. Problem kod metode je točnost određivanja parametara rotacije i translacije. Autori pretpostavljaju da je pozadina ulazne sive slike pločice uniformna tj. da dijelovi pokretne linije nisu vidljivi što nije slučaj za sivu sliku sirovih pločica. Također autori nisu predstavili postupak za uklanjanje pozadine od površine keramičke pločice, niti su pokazali mogućnost rada predložene metode u stvarnom vremenu. Može se pretpostaviti da je metoda računski zahtjevna budući da se u postupku upotrebljava

Hough-ova i dva puta Fourierova transformacija. U [96, 97] autori su predstavili sličan postupak za uklanjanje utjecaja rotacije i translacije testne slike keramičke pločice u odnosu na referentnu. Autori nakon binarizacije sive slike keramičke pločice s metodom minimalne razlike entropija detektiraju određeni broj rubnih točaka. Na temelju rubnih točaka postupkom linearne regresije s metodom najmanjih kvadrata odrede se četiri rubna pravca keramičke pločice i četiri točke ugla keramičke pločice. Minimizacijom kvadratne pogreške između četiri točke ugla testne i referentne keramičke pločice odrede se parametri rotacije i translacije. Primjenom geometrijskih transformacija utjecaj rotacije i translacije na testnoj slici se uklanja. Kao i kod prethodne metode problem je točnost dobivenih parametara rotacije i translacije. Autori nisu naveli metodu interpolacije koju su koristili tijekom uklanjanja utjecaja rotacije i translacije niti su predstavili postupak za uklanjanje pozadine od površine keramičke pločice. U [96] predstavljeni su rezultati srednjeg vremena izvođenja predložene metode. Za 1.14 MP sliku srednje vrijeme izvođenja metode na osobnom računala s Intel Pentium III procesorom je 332 ms što omogućuje rad o stvarnom vremenu. Autori u [9] postupak ujednačavanja slike površine keramičke pločice su podijelili u tri koraka: detekcija rubova keramičke pločice, uklanjanje nejednolike osvijetljenosti i proširivanje slike površine pločice zrcaljenjem. Detaljni algoritam nije dan niti za jedan od tri koraka. Vremena izvođenja predloženog postupka također nisu prezentirana.

U ovom radu predložena je nova metoda za ujednačavanje slike površine sirove keramičke pločice. Metoda se temelji na metodi za detekciju ruba i praćenju smjera konture kako je objašnjeno u poglavlju 5. Predložena metoda se sastoji od nekoliko koraka:

- uklanjanje rubova sirove pločice na sivoj slici,
- uklanjanje pozadine od površine sirove pločice i
- uklanjanje utjecaja rotacije i translacije – popunjavanje pozadinskih piksela.

U nastavku su detaljno objašnjeni pojedini koraci predložene metode za ujednačavanje slike površine sirove pločice.

6.2. Uklanjanje rubova na sivoj slici sirove keramičke pločice

Ako na sirovoj pločici nije detektirana niti jedna neispravnost na rubovima i uglovima predloženim postupkom za detekciju neispravnosti u poglavlju 5 na slici se primjenjuje postupak za ujednačavanje slike površine sirove pločice. Prvi korak u predloženom postupku je uklanjanje rubova sirove pločice. Na rubovima sirove pločice dolazi do prelamanja zraka svjetla pod različitim kutom u odnosu na površinu sirove pločice. Zbog navedenog, pikseli

koji predstavljaju rub sirove pločice su tamnijih ili svjetlijih nijansi sive u odnosu na ostale piksele površine sirove pločice, v. sliku 6.1. Ako na rubovima i uglovima sirove pločice nije detektirana niti jedna neispravnost može se pretpostaviti da je moguće ukloniti piksele koji predstavljaju rubove. Također pikseli rubova zbog tamnijih/svjetlijih nijansi mogu utjecati na postupak detekcije neispravnosti na površini sirove pločice zbog čega je pogodno ukloniti ih.

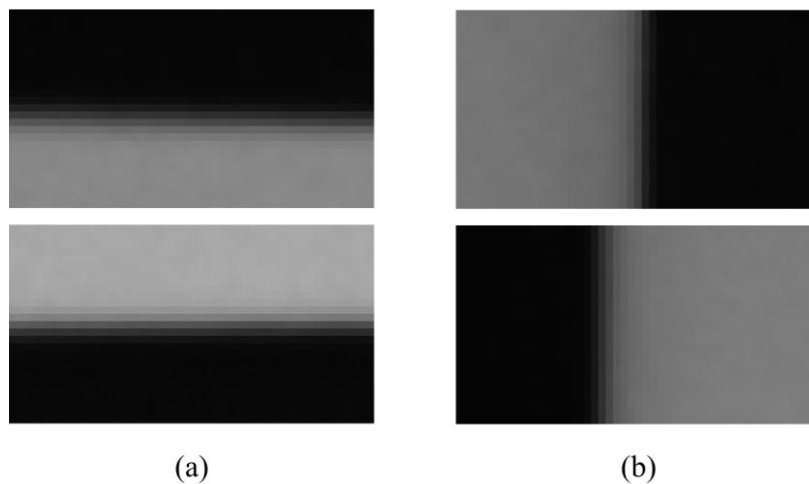
Pikseli rubova sirove pločice uklanjaju se s erozijom. Erozijska je morfološka operacija i za binarnu sliku $I_B \in \mathbb{Z}^2$ je definirana kao

$$I_{BE} = I_B \ominus E = \{z: E_z \subseteq I_B\}, \quad (6.1)$$

gdje E_z predstavlja translaciju skupa $E \in \mathbb{Z}^2$ za točku $z = (x, y)$ prema (6.2). E predstavlja strukturni element koji određuje oblik i količinu erozije. Prema (6.1) erozijska binarna slika I_B sa strukturnim elementom E je nova binarna slika I_{BE} sa svim $z = (x, y)$ pikselima za koje skup E_z se nalazi u I_B [15], v. sliku 6.2. Erozijska se koristi za uklanjanje nepoželjnih dijelova binarne slike, tj. u ovom radu za uklanjanje rubova sirove pločice.

$$E_z = \{c: c = g + z, \forall g \in E\} \quad (6.2)$$

Na binarnoj slici $I_B(x, y)$, dobivena nakon binarizacije sive slike sirove pločice s PBSCD metodom, primjenjuje se erozijska s kvadratnim strukturnim elementom čiji je primjer veličine 3×3 piksela prikazan na slici 6.2 (b). Veličina kvadratnog strukturnog elementa ovisi o količini erozije koja se želi ostvariti. Sa slike 6.2 vidljivo je da će kvadratni strukturni element veličine 3×3 piksela ostvariti eroziju veličine jedan piksel. Za eroziju većeg broja piksela potrebno je ili ponavljati postupak sa strukturnim elementom iste veličine ili povećati veličinu strukturnog elementa. Na slici 6.3 prikazan je primjer erozije binarne slike sirove



Slika 6.1: Primjer gornjeg i donjeg ruba (a), i lijevog i desnog ruba (b) na svojoj slici sirove pločice.

pločice binarizirane s PBSCD metodom. Erozijska je izvedena s kvadratnim strukturnim elementom veličine 10×10 piksela. Na slici 6.3 (d) vidljiva je količina erodiranih piksela ruba sirove pločice. Zbog veličine strukturnog elementa veličina erozije na rubu sirove pločice je pet piksela. Ovim postupkom je uklonjen rub s binarne slike sirove pločice. Potrebno je ukloniti rub s ulazne slike sirove pločice. Uklanjanje ruba na sivoj slici sirove keramičke pločice dio je postupka za uklanjanje pozadine od površine sirove pločice.

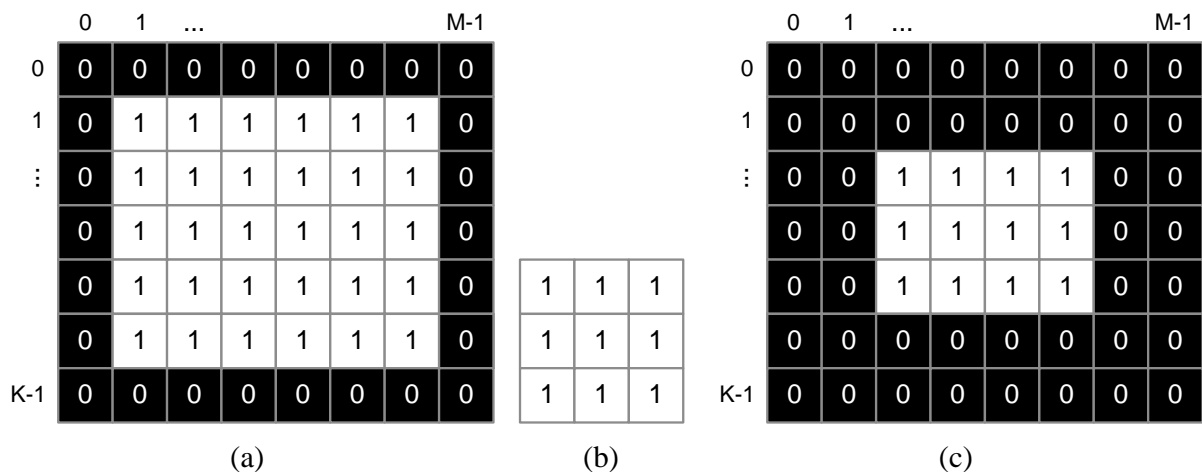
6.3. Uklanjanje pozadine od površine sirove keramičke pločice

Erodirana binarna slika upotrebljava se za postupak uklanjanja pozadine od površine sirove keramičke pločice. Budući da je na erodiranoj binarnoj slici uklonjen rub, rezultat postupka uklanjanja pozadine je i uklanjanje ruba sa sirove pločice (budući da na $I_{BE}(x, y)$ slici uklonjeni rub predstavlja pozadinu).

6.3.1. Određivanje minimalnih i maksimalnih koordinata redova i stupaca vektora konture

Erodirana $I_{BE}(x, y)$ slika predstavlja ulaz postupka za detekciju ruba objašnjenog u poglavlju 5.2. Rezultat primjene postupka za detekciju ruba je nova rubna slika $I_{ER}(x, y)$. $I_{ER}(x, y)$ slika predstavlja ulaz metode za praćenje konture sirove pločice prema poglavlju 5.3. Rezultat postupka praćenja konture je vektor konture $CE(i)$, $i = 0, 1, \dots, L - 1$.

Na temelju koordinata točaka vektora konture određuju se minimalne i maksimalne koordinate piksela površine za svaki redak i stupac slike unutar točaka vektora konture $CE(i)$.

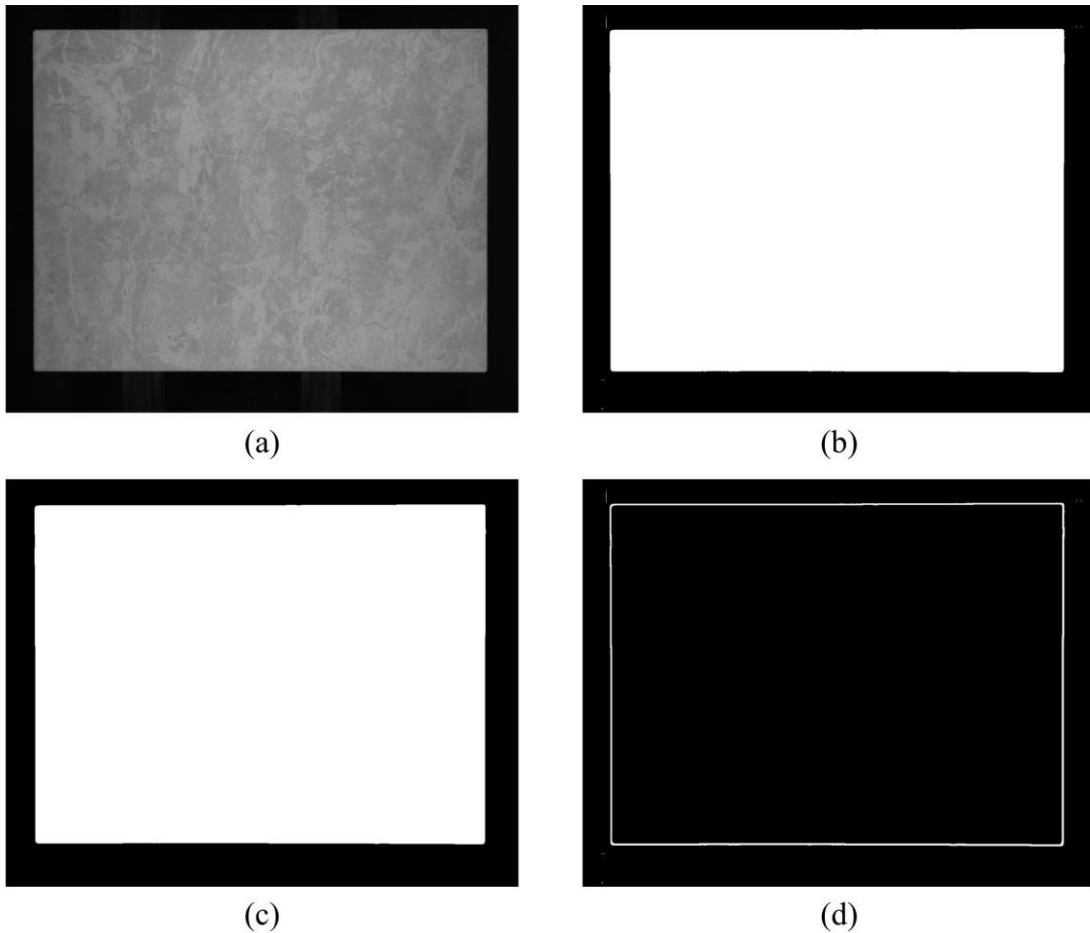


Slika 6.2: Primjer erozije $I_{BE}(x, y)$ (c) za binarnu sliku $I_B(x, y)$ (a) i kvadratni 3×3 piksela strukturni element E (b).

Minimalne – maksimalne koordinate za svaki redak slike označene su s R_{max}^a i R_{min}^a i određuju se prema (6.3) gdje eksponent a predstavlja redak slike. Minimalne – maksimalne koordinate za svaki stupac slike označene su sa S_{max}^b i S_{min}^b i određuju se prema (6.4) gdje eksponent b predstavlja stupac slike. U (6.3) i (6.4) CE_R^a predstavlja y koordinatu svih točaka konture $CE(i)$ za koje je $x = a$, a CE_S^b predstavlja x koordinatu svih točaka konture $CE(i)$ za koje je $y = b$ prema (6.5).

Na slici 6.4 (a) prikazan je primjer određivanja minimalne R_{min}^a i maksimalne R_{max}^a koordinate za redak a označen s horizontalnom zelenom linijom. Na slici 6.4 (b) prikazan je primjer određivanja minimalne S_{min}^b i maksimalne S_{max}^b koordinate za stupac b označen s vertikalnom zelenom linijom. Ako za pojedini b stupac ili a redak slike ne postoji niti jedna točka u CE_S^b ili CE_R^a isti se zanemaruje.

$$R_{max}^a = \max(CE_R^a), \quad R_{min}^a = \min(CE_R^a), \quad CE_R^a \notin \emptyset \quad (6.3)$$



Slika 6.3: Primjer erozije: ulazna slika sirove keramičke pločice (a), binarizirana ulazna slika s PBSCD metodom (b), erodirana binarna slika (c) i razlika binarne i erodirane slike (d).

$$S_{max}^a = \max(CE_S^b), \quad S_{min}^b = \min(CE_S^b), \quad CE_S^b \notin \emptyset \quad (6.4)$$

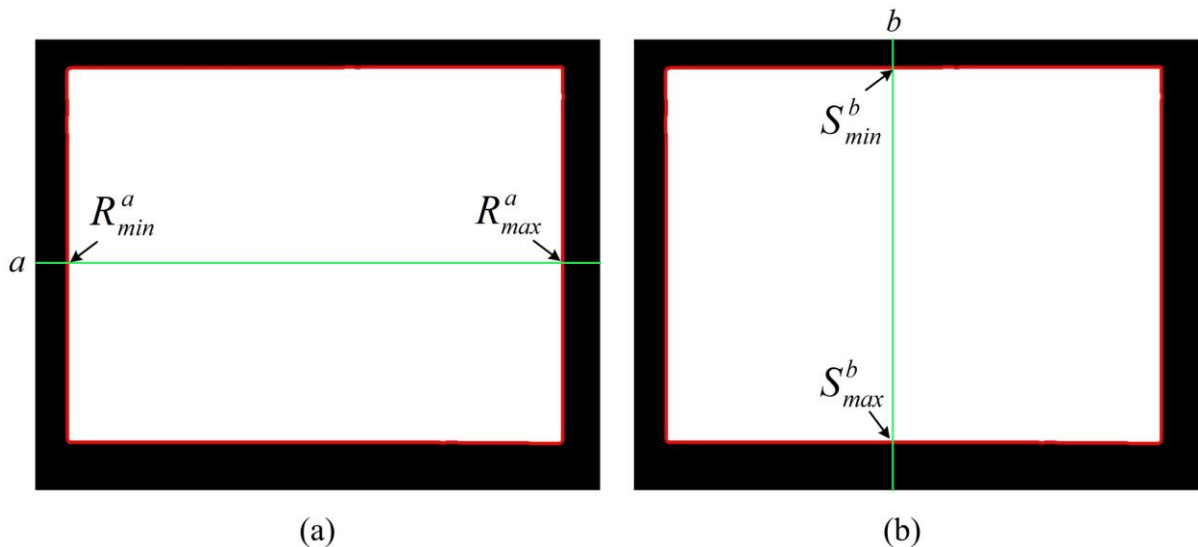
$$\begin{aligned} CE_R^a &= \{y_i: x_i = a, \forall (x_i, y_i) \in CE(i)\} \\ CE_S^b &= \{x_i: y_i = b, \forall (x_i, y_i) \in CE(i)\} \end{aligned} \quad (6.5)$$

6.3.2. Uklanjanje pozadine

Na temelju pronađenih minimalnih i maksimalnih koordinata za sve retke i stupce $I(x, y)$ slike uklanja se pozadina na slici sirove keramičke pločice. Prema (6.6) kreira se nova slika $I_P(x, y)$ jednake veličine kao $I(x, y)$. Vrijednosti piksela $I_P(x, y)$ slike u intervalu $[0 - 255]$ predstavljaju piksele površine, a vrijednost piksela -1 predstavlja piksele pozadine. Svi pikseli $I(x, y)$ slike čije koordinate se nalaze između pronađenih minimalnih i maksimalnih koordinata za odgovarajući x redak i y stupac predstavljaju piksele površine. Ostali pikseli $I_P(x, y)$ slike smatraju se pozadinom i njihova vrijednost se postavlja u -1 .

$$I_P(x, y) = \begin{cases} I(x, y) & \text{ako } R_{min}^x \leq y \leq R_{max}^x \text{ ili} \\ & S_{min}^y \leq x \leq S_{max}^y \\ -1 & \text{u ostalim slučajevima} \end{cases} \quad (6.6)$$

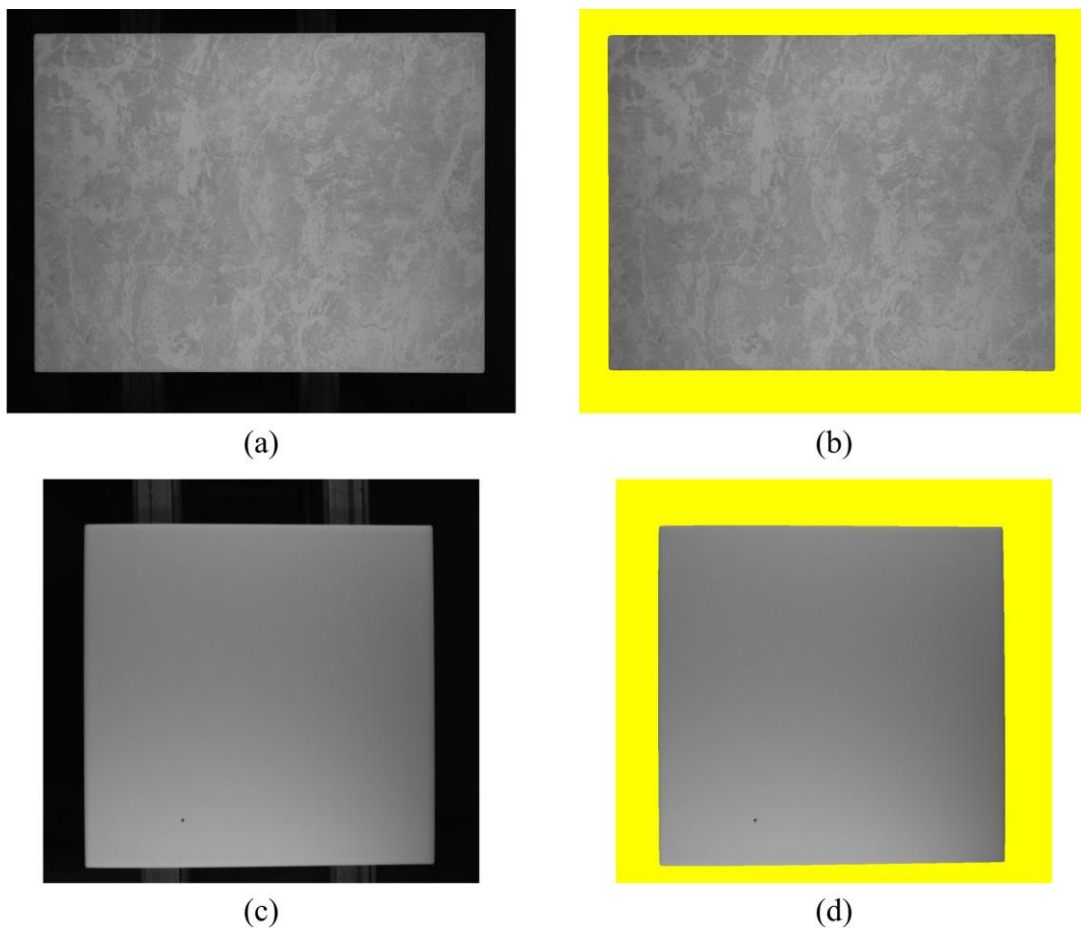
Na slici 6.5 prikazani su primjeri ulazne slike sirove pločice $I(x, y)$ i slike s uklonjenom pozadinom $I_P(x, y)$. Na $I_P(x, y)$ slici žuta boja označava pozadinu tj. vrijednost piksela -1 , ostali pikseli jednake su vrijednosti kao i na ulaznoj slici. Sa slika 6.5 (b) i (d) vidljivo je da je



Slika 6.4: Primjer pronalaska R_{min}^a i R_{max}^a koordinata za redak a označen sa zelenom linijom (a), primjer pronalaska S_{min}^b i S_{max}^b koordinata za stubac b označen sa zelenom linijom, crvena linija označava točke vektora konture $CE(i)$.

pozadina, koja uključuje dijelove pokretne trake, uklonjena. Također, budući da je postupak praćenja konture izveden na erodiranoj slici, slika $I_p(x, y)$ ne uključuje rubove sirove pločice.

U algoritmu 6.1 prikazan je pseudo kod predložene metode za uklanjanje pozadine od površine sirove keramičke pločice. Na početku algoritma sve vrijednosti piksela $I_p(x, y)$ slike postavljaju se u -1 . Algoritam se nastavlja s dvije petlje. Prva petlja za i -ti redak slike pronalazi R_{min}^i i R_{max}^i koordinate. Ako koordinate postoje, uvjet $R_{max}^i > R_{min}^i$ biti će ispunjen i vrijednosti piksela s ulazne $I(x, y)$ slike kopirat će se u $I_p(x, y)$ sliku. Druga petlja za i -ti stupac slike pronalazi S_{min}^i i S_{max}^i koordinate. Ako koordinate postoje, uvjet $S_{max}^i > S_{min}^i$ biti će ispunjen i vrijednosti piksela s ulazne $I(x, y)$ slike kopirat će se u $I_p(x, y)$ sliku. Na kraju obe petlje, na slici $I_p(x, y)$ pozadina je uklonjena od površine sirove pločice. Iz algoritma 6.1 vidljivo je da su petlje neovisne jedna o drugoj. Također iteracije unutar pojedinih petlji su neovisne jedna o drugoj. Iz navedenog se može zaključiti da je moguća paralelizacija algoritma 6.1 na paralelnim platformama.



Slika 6.5: Primjer ulaznih $I(x, y)$ slika sirovih keramičkih pločice (a), (c) i slika sirovih pločica s uklonjenom pozadinom (b), (d).

6.4. Uklanjanje utjecaja rotacije i translacije na sivoj slici sirove keramičke pločice

Nakon što je uklonjen rub i pozadina sirove keramičke pločice potrebno je ukloniti i utjecaj rotacije i translacije koja nastaje tijekom transporta sirove pločice na pokretnoj liniji. Postupak je podijeljen na dva dijela: uklanjanje translacije sirove pločice i uklanjanje rotacije sirove pločice.

6.4.1. Uklanjanje translacije sirove pločice

Kako je i vidljivo na slikama 6.5 (a) i (c) sirova pločica je pomaknuta na slici zbog transporta na pokretnoj liniji i nepreciznosti svjetlosnog senzora za detekciju sirove pločice. Da bi se dobila ujednačena slika površine sirove pločice na slici s uklonjenom pozadinom $I_P(x, y)$ prvo se uklanja utjecaj translacije sirove pločice.

$I_P(x, y)$ slika predstavlja ulaz postupka za detekciju ruba objašnjenog u poglavlju 5.2. Iako $I_P(x, y)$ nije binarna slika, moguće je primijeniti postupak za detekciju ruba uz uvjet (6.7). Primjenom postupka za detekciju ruba dobivena je nova rubna slika $I_{PR}(x, y)$. Na $I_{PR}(x, y)$ slici primjenjuje se metode za praćenje konture sirove keramičke pločice prema poglavlju 5.3. Rezultat postupka praćenja konture je vektor konture $CP(i)$, $i = 0, 1, \dots, L - 1$.

$$I_B(x, y) = \begin{cases} 1 & \text{ako } I_P(x, y) \neq -1 \\ 0 & \text{ako } I_P(x, y) = -1 \end{cases} \quad (6.7)$$

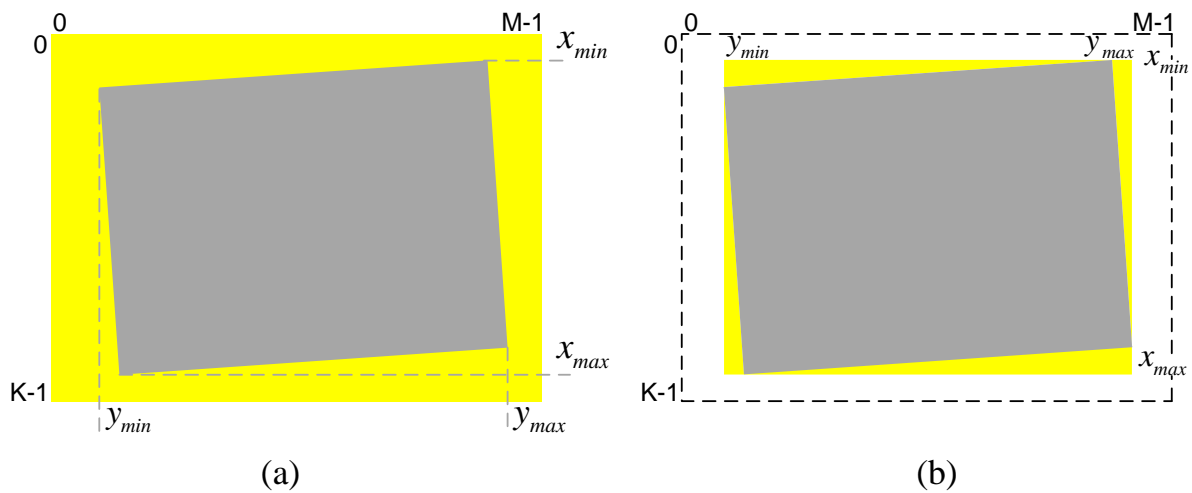
Algoritam 6.1: Pseudo kod algoritma za uklanjanje pozadine od površine sirove keramičke pločice.

```

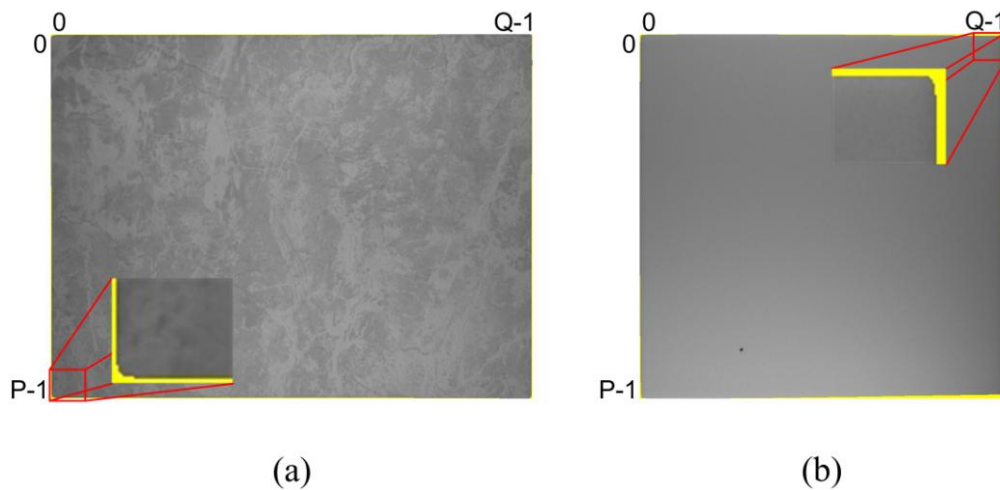
1. Ulaz:  $CE(i), I(x, y), L, K, M$ 
2. Postavi  $I_P(x, y) = -1$ 
3. for  $i := 0 \rightarrow K - 1$  do
4.   Odredi:  $CE_R^i$ 
5.   Odredi:  $R_{max}^i, R_{min}^i$ 
6.   if  $R_{max}^i > R_{min}^i$  then
7.      $I_P(x, y) = I(x, y)$ 
8.   end if
9. end for
10. for  $i := 0 \rightarrow M - 1$  do
11.   Odredi:  $CE_S^i$ 
12.   Odredi:  $S_{max}^i, S_{min}^i$ 
13.   if  $S_{max}^i > S_{min}^i$  then
14.      $I_P(x, y) = I(x, y)$ 
15.   end if
16. end for

```

Na temelju vektora konture $CP(i)$ odrede se minimalne i maksimalne vrijednosti x i y koordinata prema (6.8). S određenim x_{min} , x_{max} , y_{min} i y_{max} koordinatama kreira se nova slika $I_S(x', y')$ veličine $P \times Q$ piksela prema (6.9) i (6.10). $I_S(x', y')$ slika predstavlja odsječen dio $I_P(x, y)$ slike. Na slici 6.6 prikazan je primjer određivanja x_{min} , x_{max} , y_{min} , y_{max} koordinata i slike $I_S(x', y')$ na sintetički generiranoj slici sirove pločice. Kut rotacije sirove pločice γ na slici 6.6 veći je od 1.0° radi prikaza, žuta boja predstavlja piksele pozadine. Sa slike 6.6 (b) vidljivo je da se površina sirove keramičke pločice nalazi na sredini $I_S(x', y')$ slike tj. uklonjen je utjecaj translacije. Slika 6.7 prikazuje primjer $I_S(x', y')$ slike za dvije ulazne slike sirovih keramičkih pločica. Na uvećanim dijelovima slika (označeno sa



Slika 6.6: Sintetičko generirani primjer određivanja x_{min} , x_{max} , y_{min} i y_{max} koordinata na $I_P(x, y)$ slici (a), (b) primjer $I_S(x', y')$ slike.



Slika 6.7: $I_S(x', y')$ slika: (a) za ulaznu sliku prikazanu na slici 6.5 (a), (b) za ulaznu sliku prikazanu na slici 6.5 (c).

crvenim kvadratima) vidljivo je da je prisutan dio piksela pozadine označeni sa žutom bojom. Pikseli pozadine su uglavnom prisutni zbog rotacije sirove keramičke pločice, a djelomično i zbog nesavršenosti sustava za prikupljanje slike ili nepravilnog položaja kamere u odnosu na sirovu keramičku pločicu. Iste je potrebno ukloniti. Postupak je objašnjen u sljedećem poglavlju.

$$\begin{aligned} x_{max} &= \max(CP(i)|_x), & x_{min} &= \min(CP(i)|_x) \\ y_{max} &= \max(CP(i)|_y), & y_{min} &= \min(CP(i)|_y) \end{aligned} \quad (6.8)$$

$$I_S(x', y') = I_P(x_{min} + x, y_{min} + y) \quad \text{za} \quad 0 \leq x' < P, 0 \leq y' < Q \quad (6.9)$$

$$\begin{aligned} P &= x_{max} - x_{min} + 1 \\ Q &= y_{max} - y_{min} + 1 \end{aligned} \quad (6.10)$$

6.4.2. Uklanjanje rotacije sirove pločice

Nakon uklanjanja translacije sirove keramičke pločice potrebno je također ukloniti i utjecaj rotacije sirove pločice. Metode predstavljene u poglavlju 6.1 zasnovane su na određivanju kuta rotacije γ koji se zatim dovodi u referentni kut rotacije s geometrijskom ili Fourier-ovom transformacijom. U ovom radu predložena je metoda za uklanjanje utjecaja rotacije sirove keramičke pločice zasnovana na popunjavanju piksela pozadine.

Popunjavanje piksela pozadine izvodi se na $I_S(x', y')$ slici s vektorom smjera konture $CP'(i)$ dobivenog na temelju $CP(i)$ i koordinata x_{min} i y_{min} prema (6.11).

$$CP'(i) = \{(x'_i, y'_i): x'_i = x_i - x_{min}, y'_i = y_i - y_{min}, \forall (x_i, y_i) \in CP(i)\}, \quad (6.11)$$

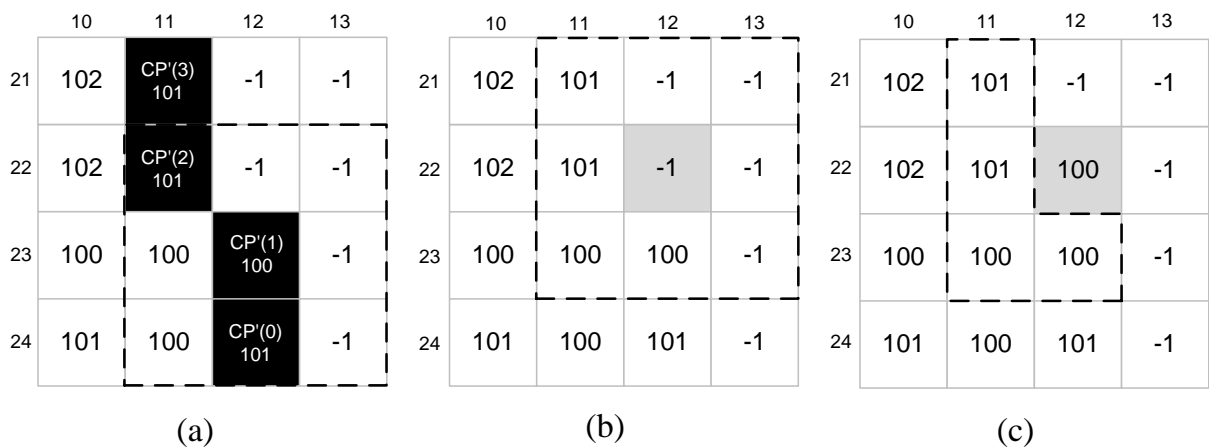
$$i = 0, 1, \dots, L - 1$$

Za svaki $(x'_i, y'_i) \in CP'(i)$ piksel $I_S(x', y')$ slike pretražuje se N_8 susjedstvo definirano u poglavlju 5.2. Na slici 6.8 (a) prikazan je primjer N_8 susjedstva za piksel $CP'(1)$ s vrijednosti 100. Susjedstvo se pretražuje u obrnutom smjeru kazaljke na satu s početnim gornjim pikselom. Ako je u N_8 susjedstvu pronađen piksel pozadine $I_S(x', y') = -1$, isti se označi i u njegovom N_8 susjedstvu pronađu se svi pikseli površine, cf. slika 6.8 (b). Na temelju vrijednosti piksela površine i (6.12) izračuna se nova vrijednost označenog piksela pozadine. U (6.12) N predstavlja broj piksela površine u N_8 susjedstvu označenog piksela pozadine. Koordinate popunjenih piksela spremaju se u privremeni vektor konture $CP'_t(j)$, $j = 0, 1, \dots, L_t$. Na slici 6.8 (c) prikazan je primjer popunjavanja vrijednosti označenog piksela pozadine (sivi kvadrat) na temelju piksela površine (označenih s isprekidanom linijom). Isti postupak se ponavlja za sve točke u vektoru konture $CP'(i)$. Na kraju postupka sve $I_{SP}(x', y')$

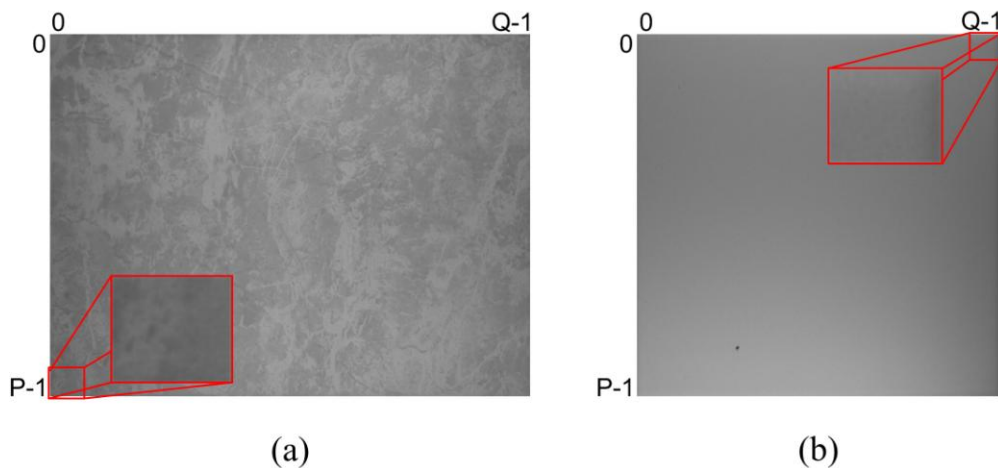
vrijednosti se kopiraju u $I_S(x', y')$ sliku, vrijednosti privremenog vektora konture $CP'_t(j)$ se kopiraju u vektor konture $CP'(i)$ te se sve vrijednosti $CP'_t(j)$ izbrišu. Čitavi postupak se ponavlja sve dok postoje točke u privremenom vektoru konture, tj. dok je $L_t > 0$.

$$I_{SP}(x', y') = \left\lfloor \frac{1}{2} + \frac{1}{N} \sum_{k=-1}^1 \sum_{l=-1}^1 I_S(x' + k, y' + l) \right\rfloor, \quad I_S(x' + k, y' + l) > -1 \quad (6.12)$$

Konačni rezultat predložene metode je siva slika $I_S(x', y')$ s popunjenim pikselima pozadine, tj. s uklonjenom rotacijom sirove pločice. $I_S(x', y')$ slika predstavlja ulaz metodama za analizu površine sirovih keramičkih pločica. Na slici 6.9 prikazani su primjeri $I_S(x', y')$ slika s popunjenim pikselima pozadine. Usporedbom slika 6.7 i 6.9 vidljivo je da



Slika 6.8: Sintetički primjer popunjavanja piksela pozadine: (a) N_8 susjedstvo označeno sa crtkanim kvadratom za piksel $CP'(1)$, (b) pronađeni piksel pozadine (sivi kvadrat) s označenim prozorom usrednjavanja (crtkani kvadrat), (c) popunjeni piksel označen sa sivom bojom.



Slika 6.9: $I_S(x', y')$ slika nakon uklanjanja rotacije: (a) za ulaznu sliku prikazanu na slici 6.5 (a), (b) za ulaznu sliku prikazanu na slici 6.5 (c).

pikseli pozadine više nisu prisutni, tj. uklonjen je utjecaj rotacije sirove keramičke pločice.

Formalni algoritam predložene metode za uklanjanje utjecaja rotacije sirove pločice prikazan je u algoritmu 6.2. Rezultat algoritma je $I_S(x', y')$ slika veličine $P \times Q$ piksela s uklonjenim utjecajem rotacije sirove keramičke pločice, tj. s popunjenim pikselima pozadine. Predloženi algoritam je moguće paralelizirati. Predloženi paralelni algoritam je objašnjen u nastavku.

6.5. Paralelni algoritam metode za ujednačavanja slike površine sirove keramičke pločice

Za paralelnu implementaciju predložene metode upotrebljavaju se višejezgreni procesori i C++ programski jezik s OpenMP bibliotekama za paralelno programiranje (vidi poglavlje 5.6). Paralelizacija metode se sastoji od tri dijela:

1. paralelizacija algoritma za detekciju ruba kako je objašnjeno u poglavlju 5.6,
2. paralelizacija algoritma za uklanjanje pozadine od površine sirove keramičke pločice i
3. paralelizacija algoritma za uklanjanje rotacije sirove keramičke pločice.

Paralelni algoritam metode za uklanjanje pozadine od površine sirove keramičke pločice dijeli postupak za određivanje minimalnih i maksimalnih koordinata redova i stupaca vektora konture na dvije ili više niti (jezgri). S pronađenim R_{max}^x , R_{min}^x , S_{max}^y i S_{min}^y koordinatama svaka nit postavlja odgovarajuće vrijednosti piksela $I_P(x, y)$ slike prema (6.6). Slika 6.10 prikazuje primjer paralelizacije postupka na dvije niti (*nit0* i *nit1*). U prvom koraku, prikazanom na slici 6.10 (a), dvije niti računaju R_{max}^x i R_{min}^x koordinate za $x = 0, 1, \dots, K - 1$, a u drugom koraku, prikazanom na slici 6.10 (b), dvije niti računaju S_{max}^y i S_{min}^y koordinate za $y = 0, 1, \dots, M - 1$. Za dobivene R_{max}^x , R_{min}^x , S_{max}^y i S_{min}^y koordinate postavljaju se

Algoritam 6.2: Pseudo kod algoritma za uklanjanje utjecaja rotacije sirove keramičke pločice.

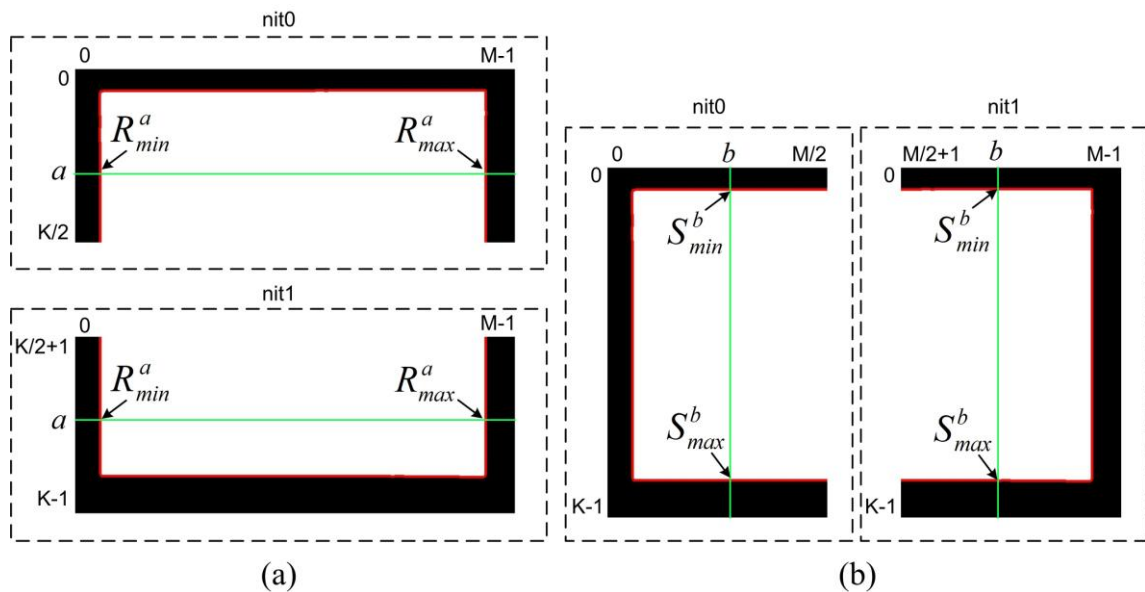
```

1. Ulaz:  $I_S(x', y')$ ,  $CP'(i)$ ,  $P$ ,  $Q$ ,  $L$ 
2. do
3.    $L_t = 0$ 
4.   for  $(x'_i, y'_i) \in CP'(i)$ ,  $i = 0, 1, \dots, L - 1$  do
5.     Pronađi  $I_S(x', y') = -1$  u  $N_8$  susjedstvu i označi ga
6.     Izračunaj  $I_{SP}(x', y')$  prema (6.12)
7.     Spremi  $(x', y')$  u  $CP'_t(j)$ ,  $L_t = L_t + 1$ 
8.   end for
9.   Kopiraj izračunate  $I_{SP}(x', y')$  vrijednosti u  $I_S(x', y')$ 
10.  Kopiraj  $CP'_t(j)$  u  $CP'(i)$  i izbriši  $CP'_t(j)$ ,  $L = L_t$ 
11. while ( $L_t \neq 0$ )

```

vrijednosti $I_P(x, y)$ slike prema (6.6). U prvom koraku *nit0* postavlja vrijednosti $I_P(x, y)$ slike za $x = 0, 1, \dots, K/2$ i $y = 0, 1, \dots, M - 1$, a *nit1* postavlja vrijednosti za $x = K/2 + 1, K/2 + 2, \dots, K - 1$ i $y = 0, 1, \dots, M - 1$. U drugom koraku *nit0* postavlja vrijednosti $I_P(x, y)$ slike za $x = 0, 1, \dots, K - 1$ i $y = 0, 1, \dots, M/2$, a *nit1* postavlja vrijednosti za $x = 0, 1, \dots, K - 1$ i $y = M/2 + 1, M/2 + 2, \dots, M - 1$. Moguća je podjela i na više niti s istom metodologijom.

Paralelni algoritam postupka za uklanjanje rotacije sirove keramičke pločice dijeli $I_S(x', y')$ sliku na dva ili više dijelova. Svaki dio slike izvodi se na jednoj niti tj. jezgri višejezgrenog procesora. Na slici 6.11 prikazana su dva primjera podjele $I_S(x', y')$ slike na više niti. Na slici 6.11 (a) prikazan je primjer paralelnog algoritma s dvije niti. $I_S(x', y')$ slika je podijeljena na dva dijela, prvi dio izvodi *nit0*, a drugi *nit1*. Svaka nit izvodi algoritam 6.2 nad pojedinom dijelu slike. Ako je P djeljivo s dva veličina prvog i drugog dijela slike je jednaka, a ako P nije djeljivo s dva drugi dio slike je za jedan red veći. Na ovaj način obe niti izvode približno jednak broj operacija. Slika 6.11 (b) prikazuje primjer paralelnog algoritma za uklanjanje utjecaja rotacije sirove pločice sa četiri niti. Slika je podijeljena na četiri dijela s granicama P_1, P_2 i P_3 kako je i prikazano na slici 6.11 (b). Granice je potrebno odrediti na način da sve niti izvode približno jednak broj operacija. Suprotno vrijeme izvođenja paralelnog algoritma se povećava.



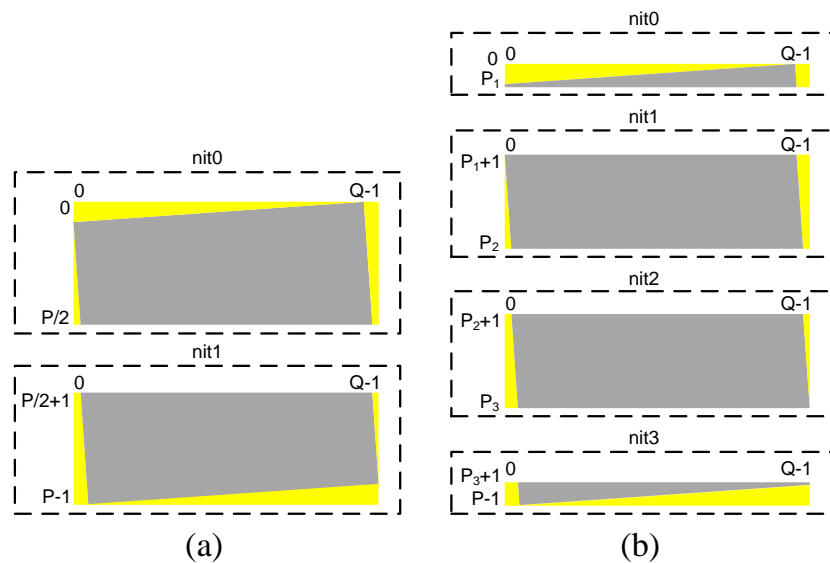
Slika 6.10: Izračun R_{max}^a i R_{min}^a koordinata s dvije niti (a), izračun S_{max}^b i S_{min}^b koordinata s dvije niti (b).

6.6. Eksperimentalni rezultati

Testiranje predložene metode se sastoji od dva dijela. Prvi dio predstavlja izračun pogreške koja nastaje pri ujednačavanju slike površine sirove pločice. Drugi dio obuhvaća testiranje predložene metode na mogućnost rada u stvarnom vremenu.

6.6.1. Pogreška metode

U predloženoj metodi postupak popunjavanja piksela pozadine tj. uklanjanja utjecaja rotacije unosi određenu pogrešku na slici površine sirove keramičke pločice. Za izračun pogreške upotrebljava se sljedeći postupak. Kako je i objašnjeno u poglavlju 6.4.2 piksel pozadine se računa na temelju N_8 susjedstva i formule (6.12). Na temelju N_8 susjedstva, $I_{SP}(x', y')$ vrijednosti piksela i (6.13) određuje se pogreška u vrijednosti piksela pozadine. $Err(x', y')$ predstavlja maksimalno odstupanje $I_{SP}(x', y')$ u odnosu na vrijednosti piksela površine u N_8 susjedstvu piksela pozadine. U (6.13) max_{N_8} predstavlja maksimalnu vrijednost piksela površine u N_8 susjedstvu (x', y') piksela, a min_{N_8} predstavlja minimalnu vrijednost piksela površine u N_8 susjedstvu (x', y') piksela. Za primjer prikazan na slici 6.8 (c) za piksel označen sa sivom bojom $Err(x', y') = 1$. Za piksele pozadine pronađene i popunjene na temelju točaka konture $CP'(i)$ računa se prosječna pogreška prema (6.14) gdje N_{SP}^k predstavlja ukupan broj popunjenih piksela pozadine za k -ti vektor konture $CP'(i)$ (na početku postupka popunjavanja piksela $k = 0$). Predložena metoda (objašnjena u poglavlju



Slika 6.11: Postupak popunjavanja piksela pozadine $I_S(x', y')$ slike na dvije niti (a), postupak popunjavanja piksela pozadine $I_S(x', y')$ slike na četiri niti (b).

6.4.2) ponavlja postupak popunjavanja piksela na temelju novog vektor konture $CP'(i) = CP_t(j)$. Za svaki novi vektor konture indeks k se povećava za jedan. Ukupna pogreška predložene metode Err_{SP} za $k > 0$ određena je prema (6.15) gdje $\overline{CErr_{CP'}^k}$ predstavlja srednju kumulativnu pogrešku k -tih vektora konture prema (6.16). Na temelju $\overline{CErr_{CP'}^k}$ pogreška nastala u prethodnim $k - 1$ koracima metode propagira se i u sljedećem koraku.

$$Err(x', y') = \max(|I_{SP}(x', y') - \max_{N_8}|, |I_{SP}(x', y') - \min_{N_8}|) \quad (6.13)$$

$$\overline{Err_{CP'}^k} = \frac{1}{N_{SP}^k} \sum Err(x', y') \quad (6.14)$$

$$Err_{SP} = \sum Err(x', y') + \sum_k N_{SP}^k \times \overline{CErr_{CP'}^{k-1}} \quad (6.15)$$

$$\overline{CErr_{CP'}^k} = \sum_k \overline{Err_{CP'}^k} \quad (6.16)$$

Za usporedbu pogreške koristi se postupak prezentiran u [96, 97] bez translacije i uz poznati kut rotacije sirove pločice γ . S poznatim kutom rotacije, pogreška prilikom otklanjanja utjecaja rotacije nastaje samo zbog geometrijske transformacije i interpolacije intenziteta piksela. Ukupna pogreška prema [96, 97] nastala na temelju geometrijske transformacije i interpolacije je

$$Err_{GT} = \sum_x \sum_y |I_{ref}(x, y) - I_{GT}(x, y)|, \quad (6.17)$$

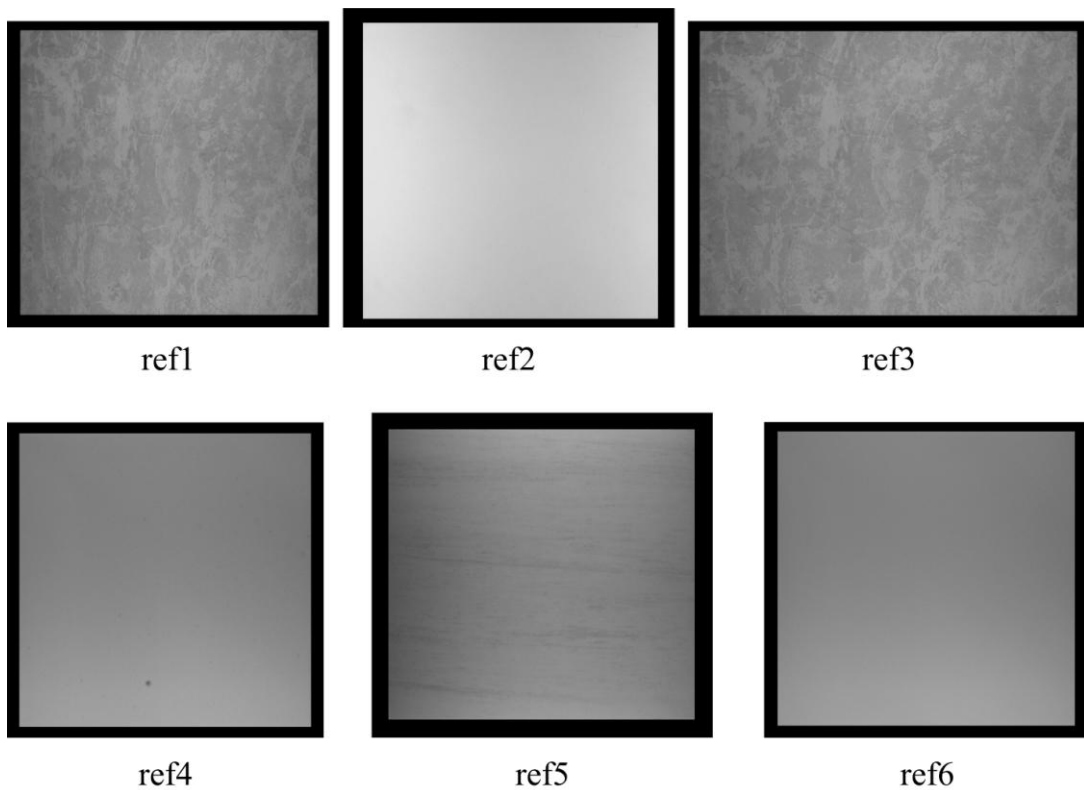
gdje $I_{ref}(x, y)$ predstavlja referentnu sliku s $\gamma = 0^\circ$, a $I_{GT}(x, y)$ predstavlja sliku s primijenjenom geometrijskom transformacijom tj. otklonjenom rotacijom. U ovom radu za geometrijsku transformaciju slike upotrijebljena je *imrotate* MATLAB funkcija s bikubičnom interpolacijom. Pored izraza za pogreške (6.15) i (6.17) za usporedbu se koriste i relativne pogreške prema

$$Err_{SP}^{\%} = \frac{1}{\Delta I_{SP}} \times \frac{Err_{SP}}{P \times Q} \times 100, \quad (6.18)$$

$$Err_{GT}^{\%} = \frac{1}{\Delta I_{ref}} \times \frac{Err_{GT}}{K \times M} \times 100, \quad (6.19)$$

gdje ΔI_{SP} predstavlja raspon vrijednosti piksela $I_S(x', y')$ slike nakon postupka uklanjanja utjecaja rotacije, a ΔI_{ref} predstavlja raspon vrijednosti piksela $I_{ref}(x, y)$ slike. Na ovaj način greške predloženih metoda je moguće usporediti iako veličine ulaznih slika nisu jednake.

Za određivanje pogreške upotrijebljeno je šest slika. Prve tri slike dobivene su u laboratorijskim uvjetima s prototipom sustava za vizualnu inspekciju, a druge tri slike dobivene su u industrijskim uvjetima s prototipom sustava za vizualnu inspekciju. Na temelju dobivenih slika kreirano je šest referentnih $I_{ref}(x, y)$ slika koje se sastoje od potpuno crne pozadine (vrijednost piksela '0') i pravokutne slike površine pločice bez rubnih dijelova. Na slici 6.12 prikazane su kreirane referentne slike. Slike 6.12 *ref1* do *ref3* predstavljaju referentne slike dobivene na temelju slika prikupljenih u industrijskim uvjetima, a slike 6.12 *ref4* do *ref6* predstavljaju referentne slike dobivene na temelju slika prikupljenih u industrijskim uvjetima. Referentne slike su rotirane u koracima $\gamma = \{0.1^\circ, 0.2^\circ, \dots, 0.7^\circ\}$. Za svaku rotiranu sliku primijenjena je predložena metoda za uklanjanje rotacije uz eroziju sa 3×3 strukturnim elementom, i geometrijska transformacija za negativne vrijednosti istih kutova rotacije. Za svaku sliku i svaki γ izračunate su pogreške prema (6.15), (6.17), (6.18) i (6.19).



Slika 6.12: Referentne slike za izračun pogreške predložene metode za uklanjanje utjecaja rotacije sirove keramičke pločice.

Tablica 6.1: Err_{SP} pogreška predložene metode za uklanjanje utjecaja rotacije.

| Slika | Kut rotacije γ | | | | | | |
|----------------|-----------------------|-------|-------|-------|--------|--------|--------|
| | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
| <i>ref1</i> | 10516 | 20454 | 26563 | 50839 | 83140 | 106806 | 131622 |
| <i>ref2</i> | 6618 | 14523 | 18629 | 31534 | 53530 | 84269 | 124280 |
| <i>ref3</i> | 11762 | 22499 | 46865 | 83563 | 131630 | 175694 | 246235 |
| <i>ref4</i> | 4821 | 4753 | 11210 | 22271 | 37113 | 39757 | 62528 |
| <i>ref5</i> | 6207 | 9665 | 14629 | 28002 | 45172 | 59966 | 74005 |
| <i>ref6</i> | 4921 | 4817 | 11671 | 22737 | 37694 | 40197 | 62598 |
| <i>Prosjek</i> | 7474 | 12785 | 21595 | 39824 | 64713 | 84448 | 116878 |

Tablica 6.2: Err_{GT} pogreška uklanjanja utjecaja rotacije sa geometrijskom transformacijom i bikubičnom interpolacijom.

| Slika | Kut rotacije γ | | | | | | |
|----------------|-----------------------|--------|--------|--------|--------|--------|--------|
| | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
| <i>ref1</i> | 286689 | 244538 | 229393 | 238622 | 249543 | 245165 | 237446 |
| <i>ref2</i> | 342246 | 315568 | 292081 | 282230 | 287830 | 297123 | 300986 |
| <i>ref3</i> | 272148 | 245484 | 241950 | 250334 | 252618 | 246500 | 245632 |
| <i>ref4</i> | 210327 | 166262 | 161833 | 178738 | 177614 | 167469 | 170861 |
| <i>ref5</i> | 250122 | 205311 | 193564 | 209056 | 213991 | 206137 | 203256 |
| <i>ref6</i> | 212698 | 171443 | 162429 | 177524 | 182013 | 171652 | 170622 |
| <i>Prosjek</i> | 262372 | 224768 | 213542 | 222751 | 227268 | 222341 | 221467 |

U tablicama 6.1, 6.2, 6.3 i 6.4 prikazani su dobiveni rezultati za šest odabranih referentnih slika, cf. Slika 6.12. Iz tablice 6.1 i 6.3 vidljivo je da sa porastom kuta rotacije raste i pogreška predložene metode. Najveća pogreška je za $\gamma = 0.7^\circ$. Razlog se može pronaći u porastu broja piksela koje je potrebno popuniti, tj. s porastom γ raste i broj piksela pozadine koje je potrebno popuniti. Najveća pogreška prema rezultatima iz tablica 6.1 i 6.3 dobivena je za *ref3* sliku. Objašnjenje se može pronaći u teksturi površine sirove keramičke pločice. Zbog teksturiranosti površine pločice, vrijednosti min_{N_8} i max_{N_8} u N_8 susjedstvu se značajno razlikuju sa čime prema (6.13) raste pogreška u izračunu vrijednosti piksela pozadine. Isto je vidljivo i za vrijednosti dobivene za sliku *ref1*. Također na slici 6.13 (a) vidljivo je da prosječna pogreška predložene metode linearno raste sa γ .

Iz tablica 6.2 i 6.4 vidljivo je da je pogreška nastala zbog geometrijske transformacije i bikubične interpolacije približno konstantna za sve kutove rotacije. Najveća Err_{GT} pogreška dobivena je za sliku *ref1*. Razlog tomu je velika razlika u intenzitetu piksela površine sirove pločice i intenziteta piksela pozadine. Na temelju vrijednosti iz tablice 6.4 može se zaključiti da je najveća $Err_{GT}^{\%}$ pogreška dobivena za slike *ref1* i *ref3*. Objašnjenje se može pronaći u teksturi površine sirove pločice. Zbog teksturirane površine dolazi do većih grešaka prilikom

Tablica 6.3: $Err_{SP}^{\%}$ pogreška predložene metode za uklanjanje utjecaja rotacije izražena u %.

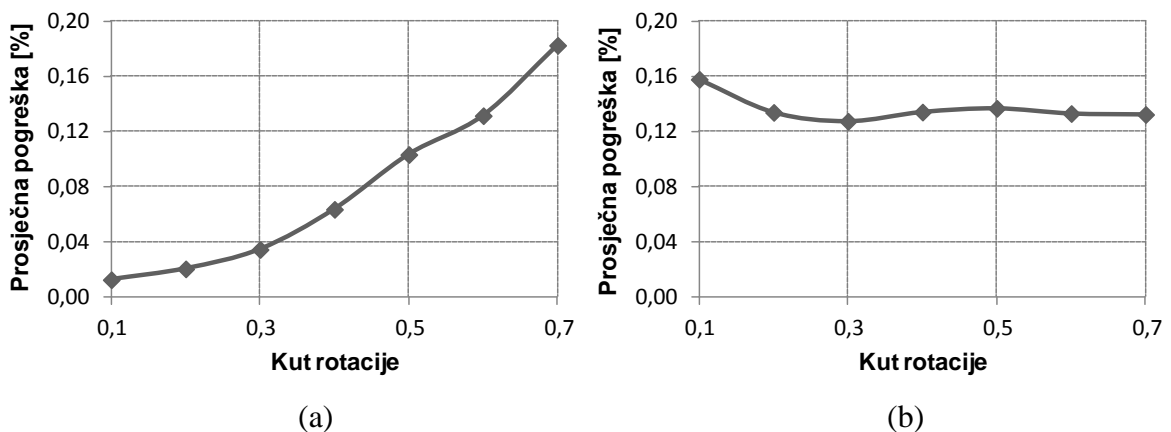
| Slika | Kut rotacije γ | | | | | | |
|---------|-----------------------|--------|--------|--------|--------|--------|--------|
| | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
| ref1 | 0,0178 | 0,0346 | 0,0448 | 0,0854 | 0,1391 | 0,1783 | 0,2192 |
| ref2 | 0,0096 | 0,0206 | 0,0261 | 0,0446 | 0,0754 | 0,1167 | 0,1736 |
| ref3 | 0,0165 | 0,0320 | 0,0654 | 0,1161 | 0,1847 | 0,2426 | 0,3386 |
| ref4 | 0,0084 | 0,0083 | 0,0195 | 0,0380 | 0,0631 | 0,0676 | 0,1057 |
| ref5 | 0,0094 | 0,0136 | 0,0203 | 0,0390 | 0,0620 | 0,0821 | 0,1011 |
| ref6 | 0,0124 | 0,0124 | 0,0299 | 0,0580 | 0,0957 | 0,1021 | 0,1582 |
| Prosjek | 0,0124 | 0,0203 | 0,0343 | 0,0635 | 0,1033 | 0,1316 | 0,1827 |

Tablica 6.4: $Err_{GT}^{\%}$ pogreška uklanjanja utjecaja rotacije sa geometrijskom transformacijom i bikubičnom interpolacijom izražena u %.

| Slika | Kut rotacije γ | | | | | | |
|---------|-----------------------|--------|--------|--------|--------|--------|--------|
| | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
| ref1 | 0,1932 | 0,1648 | 0,1546 | 0,1608 | 0,1682 | 0,1652 | 0,1600 |
| ref2 | 0,1363 | 0,1257 | 0,1163 | 0,1128 | 0,1146 | 0,1183 | 0,1199 |
| ref3 | 0,1532 | 0,1382 | 0,1362 | 0,1409 | 0,1422 | 0,1388 | 0,1383 |
| ref4 | 0,1536 | 0,1214 | 0,1182 | 0,1305 | 0,1297 | 0,1223 | 0,1248 |
| ref5 | 0,1648 | 0,1353 | 0,1275 | 0,1377 | 0,1410 | 0,1358 | 0,1339 |
| ref6 | 0,1470 | 0,1185 | 0,1123 | 0,1227 | 0,1258 | 0,1186 | 0,1179 |
| Prosjek | 0,1580 | 0,1340 | 0,1275 | 0,1342 | 0,1369 | 0,1332 | 0,1325 |

interpolacije vrijednosti piksela. Sa grafa na slici 6.13 (b) također je vidljivo da je prosječna pogreška $Err_{GT}^{\%}$ približno konstantna za sve kutove rotacije γ .

Usporedbom vrijednosti iz tablica 6.3 i 6.4 vidljivo je da za manje kutove rotacije ($\gamma < 0.5^\circ$) predložena metoda za sve testirane referentne slike daje manju vrijednost pogreške. Za slabo teksturirane i jednobojeve sirove pločice (*ref2*, *ref4*, *ref5* i *ref6*) pogreška

Slika 6.13: Prosječna pogreška (a) $Err_{SP}^{\%}$ i (b) $Err_{GT}^{\%}$ za šest referentnih slika.

Tablica 6.5: Veličina testnih slika za mjerenje srednjeg vremena izvođenja metode za ujednačavanje slike sirove keramičke pločice.

| | <i>E1</i> | <i>E2</i> | <i>E3</i> | <i>E4</i> | <i>E5</i> |
|---|--------------------|--------------------|--------------------|--------------------|--------------------|
| <i>Veličina u pikselima</i> $K \times M$ | 1038×1098 | 1204×1274 | 1380×1460 | 1692×1790 | 1946×2054 |
| <i>MP</i> | 1.14 | 1.53 | 2.01 | 3.03 | 4 |

$Err_{SP}^{\%}$ je manja u odnosu na $Err_{GT}^{\%}$ i za kutove rotacije $\gamma < 0.7^{\circ}$. Iz navedenog može se zaključiti da predložena metoda daje zadovoljavajuće rezultate za jednobojne i slabo teksturirane sirove keramičke pločice za $\gamma < 0.7^{\circ}$. Općenito, može se zaključiti da je predloženu metodu moguće koristiti za ujednačavanje slike površine jako teksturiranih i jednobojnih sirovih keramičkih pločica za kutove rotacije $\gamma < 0.5^{\circ}$. Također predložena metoda otklanja utjecaj izobličenja pravokutnosti koji može nastati zbog nepravilnog položaja kamere u odnosu na sirovu pločicu ili zbog nepotpuno uklonjene distorzije. Metode predložene u [95, 96, 97] unose i određenu pogrešku u određivanje kuta rotacije γ sa čime se povećava $Err_{GT}^{\%}$ i Err_{GT} . Navedena pogreška je zanemarena u ovom radu. Pogreške za kutove rotacije $\gamma = \{-0.1^{\circ}, -0.2^{\circ}, \dots, -0.7^{\circ}\}$ nisu dodatno analizirane budući da su dobiveni rezultati približno jednaki rezultatima za kutove rotacije $\gamma = \{0.1^{\circ}, 0.2^{\circ}, \dots, 0.7^{\circ}\}$.

6.6.2. Rad u stvarnom vremenu

Drugi dio testiranja predložene metode sastoji se od mjerenja vremena izvođenja metode na različitim veličinama sivih slika i različitim kutovima rotacije sirovih keramičkih pločica. Cilj ovog testa je analiza mogućnosti rada predložene metode u stvarnom vremenu. Prilikom testiranja mjereno je srednje vrijeme izvođenja predložene metode na temelju 100 iteracija. Za usporedbu, mjereno je vrijeme izvođenja geometrijske transformacije s bikubičnom interpolacijom i poznatim kutom rotacije. Testiranje mogućnosti rada predložene metode u stvarnom vremenu sastoji se od mjerenja izvođenja sekvencijalnog algoritma i mjerenja izvođenja paralelnog OpenMP algoritma. Sekvencijalni algoritam se sastoji od algoritama za uklanjanje rubova, uklanjanje pozadine i uklanjanja utjecaja rotacije i translacije sirove keramičke pločice prema poglavljima 6.2, 6.3 i 6.4. Paralelni OpenMP algoritam sastoji se od paralelizacije pojedinih dijelova sekvencijalnog algoritma kako je objašnjeno u poglavlju 6.5.

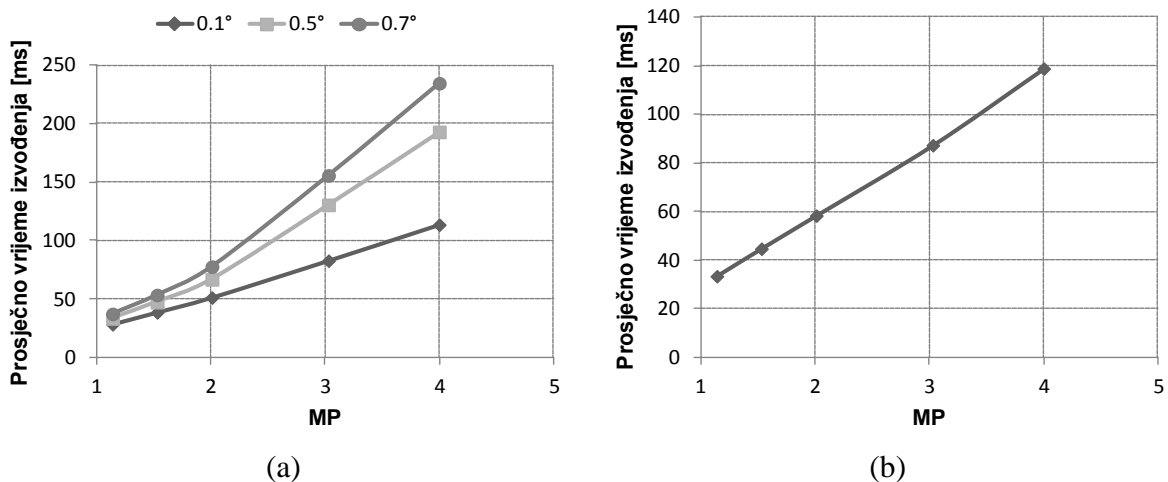
Za mjerenje srednjeg vremena izvođenja sekvencijalnog i paralelnog algoritma predložene metode upotrijebljeno je pet slika (*E1* do *E5*) različitih veličina prema tablici 6.5.

Na slikama je primijenjena erozija sa kvadratnim strukturnim elementom veličine 3×3 piksela. Algoritmi su implementirani u C++ programskom jeziku s OpenCV 2.2 bibliotekama [22] i izvedeni na osobnom računalu s Windows 7 x64 operacijskim sustavom, četverojezgrenim Intel Q6600 procesorom, 4 GB RAM memorije i NVIDIA GTX 670 grafičkim procesorom. Za C++ implementaciju upotrijebljeno je Visual Studio 2010 okruženje [74] s opcijama C++ prevoditelja:

- potpuna optimizacija po brzini izvođenja i
- brzi model broja jednostruke preciznosti.

U prvom testu mjereno je prosječno vrijeme izvođenja sekvencijalnog algoritma za pet testnih slika $E1$ do $E5$ s tri kuta rotacije $\gamma = \{0.1^\circ, 0.5^\circ, 0.7^\circ\}$. Prosječno vrijeme izvođenja predložene metode uspoređeno je s prosječnim vremenom izvođenja metode za geometrijsku transformaciju s bikubičnom interpolacijom i poznatim kutom rotacije γ . Upotrijebljena je OpenCV implementacija metode za geometrijsku transformaciju s bikubičnom interpolacijom. Rezultati mjerenja prikazani su na slici 6.14.

Vrijeme izvođenja predložene metode ovisi o kutu rotacije i veličini slike. Graf na slici 6.14 (a) prikazuje prosječno vrijeme izvođenja predložene metode. Na grafu je vidljivo da vrijeme izvođenja predložene metode linearno ovisi o veličini ulazne slike. Također s povećanjem kuta rotacije povećava se i vrijeme izvođenja predložene metode. Najveće prosječno vrijeme izvođenja pokazuju slike s $\gamma = 0.7^\circ$. Za sliku $E5$ s $\gamma = 0.7^\circ$ prosječno vrijeme izvođenja je 234 ms tj. 39 % ukupno dostupnog vremena za vizualnu inspekciju



Slika 6.14: Prosječno vrijeme izvođenja: (a) sekvencijalnog algoritma predložene metode za ujednačavanje slike površine sirove keramičke pločice za različite kutove rotacije, (b) OpenCV metode za geometrijsku transformaciju.

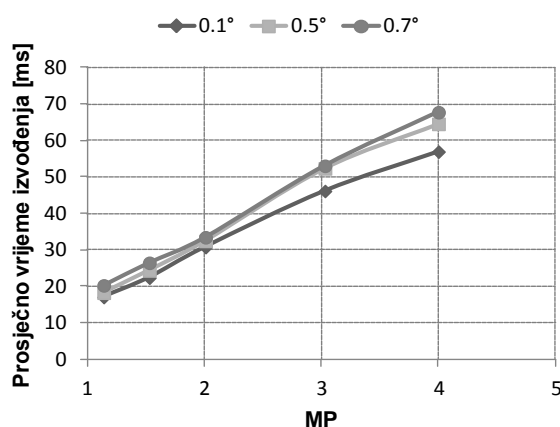
Tablica 6.6: Prosječno vrijeme izvođenja predložene metode (*SP*) za $\gamma = 0.7^\circ$ i geometrijske transformacije (*GT*).

| | <i>t</i> [ms] | | | | |
|------------------------|---------------|-----------|-----------|-----------|-----------|
| | <i>E1</i> | <i>E2</i> | <i>E3</i> | <i>E4</i> | <i>E5</i> |
| <i>SP</i> | 37.1 | 53.3 | 77.6 | 155.7 | 234.7 |
| <i>GT</i> | 33.28 | 44.61 | 58.23 | 87.23 | 118.79 |
| <i>Usporenje SP/GT</i> | 1.11 | 1.19 | 1.33 | 1.78 | 1.98 |

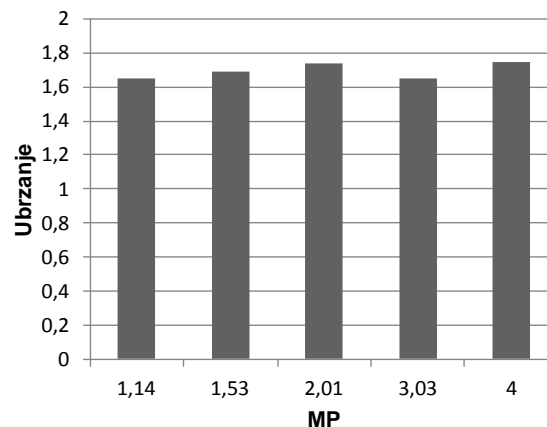
sirovih pločica.

Graf na slici 6.14 (b) prikazuje prosječno vrijeme izvođenja metode za geometrijsku transformaciju i bikubičnu interpolaciju. Vrijeme izvođenja metode je neovisno o kutu rotacije i linearno raste s povećanjem veličine slike. U prikazano vrijeme nije uračunato vrijeme metode za određivanje kuta rotacije γ što prethodi postupku geometrijske transformacije prema [95, 96, 97]. Najveće prosječno vrijeme izvođenja od 119 ms izmjereno je za sliku *E5*. Prosječno vrijeme izvođenja predložene metode za $\gamma = 0.1^\circ$ približno je jednako ili manje vremenu izvođenja geometrijske transformacije. Iz tablice 6.6 vidljivo je da za sliku *E5* i kut rotacije $\gamma = 0.7^\circ$ predložena metoda je 1.98 puta sporija u odnosu na geometrijsku transformaciju. Isto je potrebno smanjiti.

U drugom testu mjereno je prosječno vrijeme izvođenja paralelnog OpenMP algoritma objašnjenog u poglavlju 6.5. Vrijeme je mjereno za pet testnih slika prema tablici 6.5. Metoda je izvedena sa četiri niti na četiri jezgre Intel Q6600 procesora. Rezultati prosječnog vremena izvođenja prikazani su na slici 6.15. Iz grafa na slici 6.15 (a) vidljivo je da je prosječno



(a)



(b)

Slika 6.15: Prosječno vrijeme izvođenja paralelnog algoritma predložene metode za ujednačavanje slike površine sirove keramičke pločice za različite kutove rotacije, (b) ubrzanje u odnosu na metodu za geometrijsku transformaciju.

vrijeme izvođenja paralelnog algoritma u svim slučajevima kraće u odnosu na najveće vrijeme sekvencijalnog algoritma. Postignuto ubrzanje (omjer vremena izvođenja sekvencijalnog i paralelnog algoritma) za slučaj slike *E5* i $\gamma = 0.7^\circ$ veće je od 3.4 puta. Najveće prosječno vrijeme izvođenja paralelnog algoritma iznosi 67.7 ms tj. manje od 12 % ukupnog vremena dostupnog za vizualnu inspekciju sirovih pločica.

Graf na slici 6.15 (b) prikazuje ubrzanje paralelnog OpenMP algoritma predložene metode u odnosu na sekvencijalni algoritam metode za geometrijsku transformaciju. Ubrzanje je određeno kao omjer prosječnog vremena izvođenja metode za geometrijsku transformaciju i prosječnog vremena izvođenja paralelnog algoritma predložene metode za ujednačavanje slike površine sirove pločice. S grafa je vidljivo da predloženi paralelni algoritam u svim slučajevima najmanje 1.6 puta brži od sekvencijalnog algoritma za geometrijsku transformaciju slike.

6.6.3. Zaključak

Iz rezultata dobivenih iz provedenih testiranja proizlaze sljedeći zaključci.

- Za predloženu metodu za ujednačavanje slike površine sirove pločice određena je pogreška u određivanju vrijednosti piksela pozadine Err_{SP} i njena relativna vrijednost $Err_{SP}^{\%}$. Iste su uspoređene s pogreškom standardne metode za uklanjanje utjecaja rotacije sirove keramičke pločice geometrijskom transformacijom s poznatim kutom rotacije γ . Dobiveni rezultati su pokazali da predložena metoda za manje kutove rotacije ($-0.5^\circ < \gamma < 0.5^\circ$) rezultira s manjom vrijednosti pogreške za sve testirane slike. Također pokazano je da za jednobojne sirove keramičke pločice metoda za ujednačavanje slike površine sirove pločice rezultira manjom pogreškom za kutove rotacije $-0.7^\circ < \gamma < 0.7^\circ$ u odnosu na standardnu metodu za uklanjanje utjecaja rotacije. Predložena metoda, za razliku od standardne geometrijske transformacije, otklanja i utjecaj izobličenja pravokutnosti koji može nastati zbog nepravilnog položaja kamere u odnosu na sirovu pločicu ili zbog nepotpuno uklonjene distorzije slike. Općenito, može se zaključiti da za $-0.5^\circ < \gamma < 0.5^\circ$ metoda za ujednačavanje slike površine sirove keramičke pločice kao rezultat daje novu sliku površine sirove pločice pripremljenu za detekciju neispravnosti na površini.
- Vrijeme izvođenja predložene metode uspoređeno je sa vremenom izvođenja metode za geometrijsku transformaciju sa bikubičnom interpolacijom. Prosječno

vrijeme izvođenja metode za ujednačavanje slike površine sirove pločice ovisno je o veličini slike i kutu rotacije γ . U najlošijem slučaju (najveće prosječno vrijeme izvođenja) prosječno vrijeme izvođenja sekvencijalne implementacije je 234.7 ms , tj. 1.98 puta veće vrijeme u odnosu na vrijeme izvođenja metode za geometrijsku transformaciju sa bikubičnom interpolacijom. Paralelna OpenMP implementacija predložene metode smanjuje najlošije vrijeme izvođenja sekvencijalne implementacije na 67.7 ms . Najlošije vrijeme izvođenja paralelnog algoritma ne prelazi 12 % ukupno dostupnog vremena za vizualnu inspekciju sirovih pločica. Iz dobivenih rezultat može se zaključiti da vrijeme izvođenja paralelne implementacije metode za ujednačavanje slike površine sirove pločice zadovoljava postavljene vremenske uvjete za rad u stvarnom vremenu.

7. DETEKCIJA NEISPRAVNOSTI NA POVRŠINI SIROVIH KERAMIČKIH PLOČICA

Sljedeći korak u vizualnoj inspekciji neispravnosti na sirovim keramičkim pločicama je detekcija neispravnosti na površini sirovih pločica. Ulaz metoda za detekciju neispravnosti na površini može biti ulazna slika sirove pločice ili ujednačena slika površine sirove pločice kako je objašnjeno u poglavlju 6. Najčešće neispravnosti koje se pojavljuju na površini sirovih pločica su: točkaste neispravnosti, neispravnosti nastale od kapljica kondenzata, nedostatak glazure i nepravilnosti u teksturi. U rjeđe neispravnosti spadaju ispupčenja i udubljenja na površini sirove keramičke pločice. Budući da keramička pločica poprima konačnu boju tek nakon procesa paljenja (v. poglavlje 2.2), neispravnosti površine vezane za boju pločice ne detektiraju se na sirovoj keramičkoj pločici. Ako na sirovoj keramičkoj pločici nije pronađena niti jedna neispravnost na površini, ista nastavlja u fazu paljena sirove keramičke pločice. U suprotnom, sirova pločica se reciklira tj. preusmjerava se u fazu miješanja sirovine.

7.1. Postojeće metode za detekciju neispravnosti na površini sirovih keramičkih pločica

U literaturi se može pronaći niz metoda za detekciju neispravnosti na površini keramičkih pločica. Metode se mogu podijeliti u:

- statističke metode,
- metode temeljene na filtriranju slike površine keramičke pločice i
- metode temeljene na detekciji rubova na slici površine keramičke pločice.

U literaturi primjeri statističkih metoda za detekciju neispravnosti na površini keramičkih pločice mogu se pronaći u [98, 90, 91]. U [98] autori su predstavili MALD (*engl. Moving Average with Local Difference*) metodu za detekciju točkastih neispravnosti na jednobojnim i teksturiranim keramičkim pločicama. Metoda se temelji na pomičnom prosjeku i usporedbi razlike vrijednosti piksela u odnosu na njegovu okolinu. Detekcija neispravnosti MALD metode je veća od 80 %. Autori u [49] prezentirali su više različitih algoritama MALD metode za rad u stvarnom vremenu na različitim sklopovskim platformama. U doktorskoj disertaciji [90] autor je prezentirao nekoliko statističkih metoda i metoda temeljenih na filtriranju slike za klasifikaciju keramičkih pločica. Najbolji rezultati klasifikacije dobiveni su s kombinacijom Haralick-ovih značajki GLCM (*engl. Grey-Level Co-occurrence Matrix*) matrice i umjetnom neuronskom mrežom s propagacijom pogreške

unatrag. Metoda je pokazala uspješnost ispravne klasifikacije (detekcije neispravnosti) preko 97 %. U [91] autori su prezentirali metodu za klasifikaciju keramičkih pločica temeljenu na Bayesovoj teoriji odlučivanja i statističkim značajkama neispravnosti (pukotina) na površini keramičke pločice. Najveća pogreška klasifikacije iznosi 3.6 %. Autori nisu predložili metode za određivanje statističkih značajki neispravnosti.

U literaturi postoji niz metoda za detekciju neispravnosti na površini keramičkih pločica temeljene na filtriranju slike. Neke od metoda opisane su u [90, 92, 93]. U doktorskoj disertaciji [90] autor je prezentirao nekoliko metoda za detekciju neispravnosti na površini keramičkih pločica. Metode su temeljene na Gaborovim filterima, usmjerenoj diskretnoj kosinusovoj transformaciji, eigen filterima i kombinaciji Gaborovih filtera s matricom prostornih ovisnosti (*engl. co-occurrence matrix*). Rezultati klasifikacije iznad 97 % dobiveni su s kombinacijom značajki Gaborovih filtera s matricom prostornih ovisnosti i umjetnom neuronskom mrežom s propagacijom pogreške unatrag. U [92, 93] autori su implementirali filtriranje slike sa valićima (*engl. wavelets*). U [93] autori su dali detaljnu analizu filtriranja slika s teksturiranom površinom s valićima. Opisali su postupak i odabir parametara. Rezultati su prezentirani na nekoliko primjera različitih slika teksturiranih površina. Autori su u [92] prezentirali metodu za detekciju neispravnosti na jednobojnim i teksturiranim keramičkim pločicama temeljenu na filtriranju slike s valićima i usporedbom sa referentnom slikom. Ulazna slika površine keramičke pločice podijeljena je na nekoliko segmenata na kojima je primijenjena transformacija s valićima. Dobiveni koeficijenti transformacije uspoređuju se sa koeficijentima transformacije referentne slike površine keramičke pločice. Na temelju praga i razlike koeficijenata testne i referentne slike detektiraju se neispravnosti na površini jednobojnih i teksturiranih keramičkih pločica. Prezentirani rezultati pokazuju 100 % uspješnost detekcije neispravnosti na površini i manje od 1 % pogrešno detektiranih ispravnih keramičkih pločica.

Metode temeljene na detekciji rubova na slici površine keramičke pločice prezentirane su u [94, 99, 100]. U [94] autori su prezentirali niz metoda za detekciju različitih (točkastih neispravnosti, mrlja, pukotina, i sl) neispravnosti. Metode se temelje na uklanjanju šuma, detekciji rubova i morfološkim operacijama nad slikom površine keramičke pločice. Detalji o pojedinim fazama metoda nisu prikazani kao ni rezultati uspješnosti detekcije neispravnosti na površini sirovih pločica. Autori u [99] predstavili su metodu za detekciju neispravnosti na slici površine keramičke pločice temeljenu na Sobel operatoru za detekciju ruba. Na dobivenoj slici ruba primjenjuju se različite morfološke operacije i usporedbom sa referentnom slikom površine keramičke pločice detektiraju se neispravnosti. Dobiveni

rezultati pokazuju uspješnost detekcije neispravnosti iznad 90 %. U [100] autori su također implementirali metodu za detekciju neispravnosti na slici površine keramičke pločice temeljenu na detkeciji ruba sa Sobel operatorom. Na slici ruba primjenjuje se binarizacija na temelju globalne razine praga i morfološke operacije za detekciju neispravnosti (pukotine i mrlje). Prezentirani rezultati pokazuju 96 % uspješnost detekcije neispravnosti.

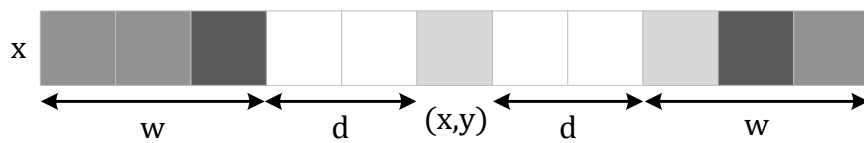
Pored navedenih metoda postoji i niz drugih metoda za detekciju neispravnosti na površini keramičkih pločica, tekstilnih površina, metalnih površina i sl. Budući da se u ovoj disertaciji neće detaljno analizirati metode za detekciju neispravnosti na površini sirovih keramičkih pločica, u nastavku će biti obrađene samo odabrane metode.

7.2. MALD metoda za detekciju točkastih neispravnosti

MALD metoda objašnjena u [98] i opisana kao optimirani točkasti detektor u [6], detektira točkaste neispravnosti na jednobojnim i teksturiranim keramičkim pločicama. Točkaste neispravnosti predstavljene su sa grupom piksela čije vrijednosti relativno visoko odstupaju u odnosu na vrijednosti okolnih piksela. Metoda se temelji na pomičnom prosjeku i usporedbi razlike vrijednosti promatranog piksela u odnosu na njegovu usrednjenu okolinu. Metoda se primjenjuje na recima $I(x, y)$ slike.

Ulaz MALD metode je siva $I(x, y)$ $K \times M$ piksela slika i četiri ulazna parametra: w , d , t_1 i t_2 . w predstavlja veličinu prozora pomičnog prosjeka s obadvije strane analiziranog piksela, d predstavlja udaljenost između analiziranog piksela i prozora pomičnog prosjeka, t_1 predstavlja razinu praga za otklanjanje piksela ruba i t_2 predstavlja razinu praga za detekciju neispravnog piksela. Na slici 7.1 prikazan je primjer parametara za x redak slike i (x, y) piksel. Izlaz MALD metode je binarna slika $I_B(x, y)$ veličine $K \times M$ piksela u kojoj '1' označava piksel neispravnosti. Na temelju $I(x, y)$ slike i ulaznih parametara računa se apsolutna prosječna razlika lijevog i desnog prozora (x, y) piksela prema (7.1).

$$e(x, y) = \frac{1}{w} \left| \sum_{i=y-w-d-1}^{y-d-1} I(x, i) - \sum_{i=y+d}^{y+w+d} I(x, i) \right|, w + d < y < M - w - d \quad (7.1)$$



Slika 7.1: Prikaz parametara MALD metode za (x, y) piksel.

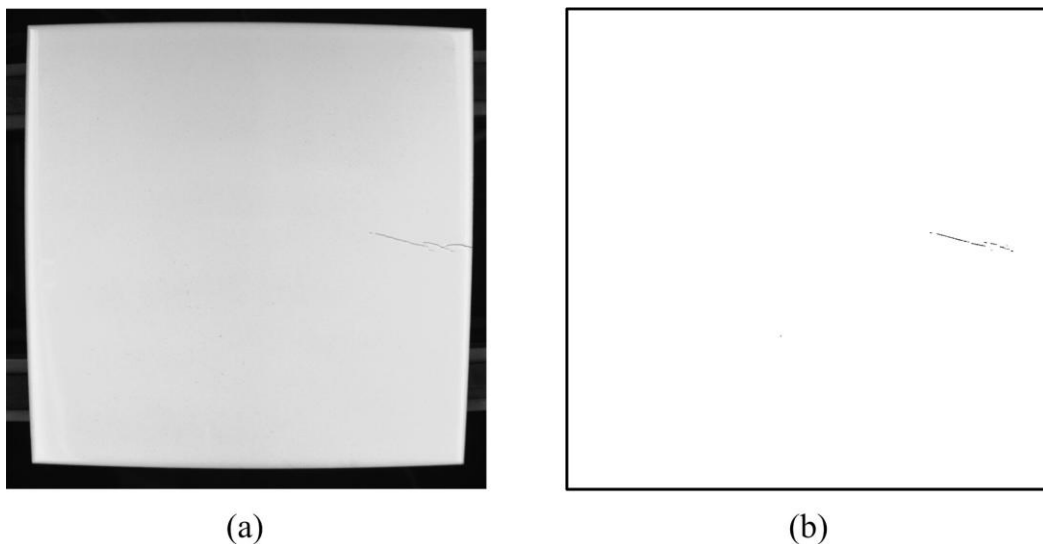
Vrijednost $e(x, y)$ uspoređuje se s razinom praga t_1 . Ako je $e(x, y) < t_1$ piksel (x, y) se smatra pikselom površine i nastavlja se s analizom. U suprotnom, pretpostavlja se da (x, y) piksel predstavlja piksel ruba pločice i ne nastavlja se s analizom.

Za (x, y) piksel koji ispunjava uvjet $e(x, y) < t_1$ računa se razlika prosječne vrijednosti oba prozora i intenziteta analiziranog piksela prema (7.2). Vrijednost $|h(x, y)|$ uspoređuje se s razinom praga t_2 . Ako je $|h(x, y)| \geq t_2$ analizirani piksel predstavlja neispravnost, u suprotnom analizirani piksel ne predstavlja neispravnost.

Metoda je također primjenjiva i na linijskim neispravnostima budući da su linijske neispravnosti također predstavljene sa grupom piksela čije vrijednosti relativno visoko odstupaju u odnosu na vrijednosti okolnih piksela. Na slici 7.2 prikazan je primjer detekcije linijske neispravnosti na slici keramičke pločice sa MALD metodom. Metoda je također primjenjiva, uz izmjene, i za detekciju točkastih neispravnosti na sirovim pločicama. Potrebno je uzeti u obzir da intenzitet piksela pokretne trake su sličnih vrijednosti kao intenzitet piksela površine sirove pločice. Također, moguća je primjena MALD metode na slici površine sirove pločice $I_{SP}(x', y')$, v. poglavlje 6. Budući da $I_{SP}(x', y')$ ne prikazuje pozadinu i uklonjeni su pikseli ruba sirove keramičke pločice, MALD metodu je potrebno modificirati tako da se ne primjenjuje prag t_1 .

$$h(x, y) = \frac{1}{2w} \left(\sum_{i=y-w-d-1}^{y-d-1} I(x, i) + \sum_{i=y+d}^{y+w+d} I(x, i) \right) - I(x, y), \quad (7.2)$$

$$w + d < y < M - w - d$$



Slika 7.2: Primjer detektiranih linijskih neispravnosti MALD metodom.

Algoritam MALD metode za rad u stvarnom vremenu opisan je u [49]. Najkraće vrijeme izvođenja postignuto na osobnom računalu je s CUDA paralelnom implementacijom. Za sivu sliku veličine 1024×1024 piksela i sa NVIDIA GT9800 grafičkom karticom dobiveno vrijeme je manje od 5 ms. Prema vremenu izvođenja paralelnog algoritma može se zaključiti da je metoda primjenjiva za detekciju neispravnosti na sirovim pločicama u stvarnom vremenu.

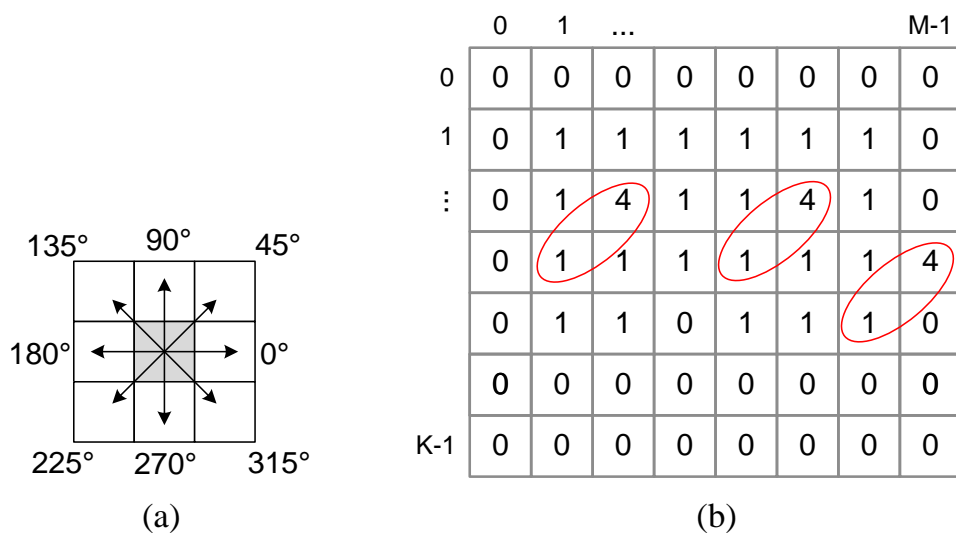
7.3. Metoda Haralick-ovih značajki s umjetnom neuronskom mrežom

U doktorskoj disertaciji [90] autor je prezentirao metodu za detekciju neispravnosti na teksturiranim keramičkim pločicama temeljenu na Haralick-ovim značajkama teksture i umjetnom neuronskom mrežom s propagacijom pogreške unatrag.

Haralick et al. je značajke sive slike prezentirao u [101]. Značajke se temelje na prostornoj ovisnosti intenziteta piksela sive slike. Autori su predložili izračun matrice prostornih ovisnosti intenziteta piksela sive slike – GLC (*engl. Grey-Level Co-occurrence*) matrice. Za sivu sliku $I(x, y)$ veličine $K \times M$ piksela GLC matrica je definirana sa

$$GLC_{d,\varphi}(k, l) = \sum_{x=0}^{K-1} \sum_{y=0}^{M-1} p(I(x, y), I(x^p, y^p), k, l) \quad (7.3)$$

gdje $I(x, y)$ i $I(x^p, y^p)$ predstavljaju par piksela udaljenih za d u smjeru φ , a $p(\cdot)$ predstavlja funkciju definiranu sa (7.4). GLC matrica pokazuje broj pojavljivanja intenziteta parova



Slika 7.3: Parametar φ GLC matrice (a), (b) primjer parova (zaokruženi) $I(x, y) = 1$ i $I(x^p, y^p) = l$ za $d = 1$ i $\varphi = 45^\circ$.

Tablica 7.1: Prosječno vrijeme izvođenja predložene metode (SP) za $\gamma = 0.7^\circ$ i geometrijske transformacije (GT).

| Značajka | Maksimum | Energija | Korelacija | Inercija |
|----------|-------------------------------|--------------------------------|--|--------------------------------------|
| Oznaka | f_1 | f_2 | f_3 | f_4 |
| Formula | $\max(GLC_{d,\varphi}(x, y))$ | $\sum GLC_{d,\varphi}(x, y)^2$ | $\sum \frac{(x-\mu_x)(y-\mu_y)GLC_{d,\varphi}(x,y)}{\sigma_x\sigma_y}$ | $\sum (x-y)^2 GLC_{d,\varphi}(x, y)$ |

piksela $I(x, y) = k$ i $I(x^p, y^p) = l$ u $I(x, y)$ slici. Veličina GLC matrice je $B \times B$ gdje B predstavlja broj razina intenziteta piksela sive slike. Na slici 7.3 (a) prikazan je smjer promatranih parova u odnosu na parametar φ , a na slici 7.3 (b) prikazan je primjer pronađenih parova piksela za parametre $k = 1$, $l = 4$, $d = 1$ i $\varphi = 45^\circ$. Na temelju slike 7.3 (b) vrijednost GLC matrice u retku 1 i stupcu 4 je 3, tj. $GLC_{1,45^\circ}(1,4) = 3$.

$$p(I(x, y), I(x^p, y^p), k, l) = \begin{cases} 1 & \text{ako } I(x, y) = k \text{ i } I(x^p, y^p) = l \\ 0 & \text{u ostalim slučajevima} \end{cases} \quad (7.4)$$

Na temelju GLC matrice autori u [101] predložili su 14 značajki teksture. U [90] autor je u svojoj metodi upotrijebio četiri značajke GLC matrice prema tablici 7.1, tri udaljenosti $d \in \{1, 6, 36\}$ i četiri smjera $\varphi \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$. Na temelju kombinacija četiri smjera, tri udaljenosti i četiri značajke dobije se ukupno 48 značajki slike.

Navedenih 48 značajki predstavljaju ulaz u umjetnu neuronsku mrežu. Umjetna neuronska mreža predstavlja pojednostavljeni matematički model ljudskog mozga kojim se nastoji simulirati postupak učenja. Umjetna neuronska mreža se sastoji od jednog ili više neurona koji izvode jednostavan zadatak. Svaki neuron prima jedan ili više ulaznih signala s_i . Ulazni signali pomnoženi sa težinskim faktorima ω_i se sumiraju, te na temelju praga β i prijenosne (transfer) funkcije $f(\cdot)$ neuron daje izlazni signal Z prema (7.5). Prijenosna funkcija može biti jednostavna funkcija skoka ali najčešći oblik prijenosne funkcije je sigmoidalna funkcija prema (7.6).

$$Z = f\left(\sum \omega_i s_i - \beta\right) \quad (7.5)$$

$$f(x) = \frac{1}{1 + e^{-x}} \quad (7.6)$$

Neuronska mreža obrađuje podatke na temelju težinskih faktora. Optimiziranje težinskih faktora provodi se postupkom učenja ili treniranja. Tijekom postupka učenja težinski faktori se postupno prilagođavaju sve do trenutka kada je izlaz iz mreže

zadovoljavajući. Postupak je iterativan i može se provoditi sa poznatim vrijednostima ulaza i izlaza (učenje s učiteljem – *engl. supervised learning*) i bez poznavanja vrijednosti izlaza (učenje bez učitelja – *engl. unsupervised learning*) [102].

U [90] autor je upotrijebio višeslojnu umjetnu neuronsku mrežu s jednim skrivenim slojem. Učenje neuronske mreže izvedeno je algoritmom propagacije pogreške unazad (učenje s učiteljem). Za postupak učenja upotrijebio je 145 sivih slika površine 11 različitih teksturiranih keramičkih pločica veličine 256×256 piksela. Pločice su klasificirane u ispravne i neispravne. Postupak klasifikacije proveden je za 1883 sive slike površine teksturiranih keramičkih pločica veličine 256×256 piksela. Rezultati su pokazali uspješnost klasifikacije veću od 97 %. Autor je također analizirao vrijeme izvođenja predložene metode. Za slike veličine 256×256 piksela prosječno vrijeme izvođenja sekvencijalnog algoritma na Intel Pentium III procesoru je 132 ms.

Predložena metoda je također primjenjiva i na slikama površine sirovih keramičkih pločica. Ulaz metode je slika površine sirove pločice dobivene nakon postupka objašnjenog u poglavlju 6, a izlaz odluka ispravna – neispravna sirova pločica. Također, pretpostavka je da uz nove paralelne tehnologije (višejezgreni CPU, CUDA GPU) moguć je rad metode u stvarnom vremenu i za slike veće rezolucije.

8. ZAKLJUČAK

Keramičke pločice se proizvode u tri različita procesa proizvodnje ovisno o broju ciklusa paljenja i u korištenju glazure. Proces proizvodnje keramičkih pločica s glazurom monopaljenjem sastoji se od nekoliko faza: rudarenja sirovine, skladištenje, priprema materijala, prešanje, glaziranje, paljenje, kontrola kvalitete, sortiranje i pakiranje gotove keramičke pločice. Prije faze sortiranja izvodi se vizualna kontrola kvalitete keramičke pločice. Faza vizualne kontrole kvalitete je najmanje automatizirana. Provodi se od strane operatera (kontrolor kvalitete) koji na temelju vizualne inspekcije svakoj keramičkoj pločici dodjeljuje ocjenu kvalitete, npr. I., II. i III klasa. Ocjena kvalitete se određuje na temelju značajki definiranih u standardima (ISO, EN) i na temelju subjektivnog dojma samog operatera. U procesu proizvodnje keramičkih pločica nastaju tri međuproizvoda: osušeno tijelo pločice ili keks, glazirana sirova pločica i pečena keramička pločica. Sva tri međuproizvoda moguće je reciklirati. Međuproizvodi se recikliraju vraćanjem u određenu fazu procesa proizvodnje keramičkih pločica. Tijelo pločice i sirova pločica se vraćaju u proces miješanja sirovine bez dodatnog utroška energije, dok keramička pločica odlazi u postupak usitnjavanja uz dodatni utrošak energije.

U keramičkoj industriji vizualna inspekcija i kontrola kvalitete najčešće se provodi samo na pečenoj keramičkoj pločici prije faze sortiranja. Dodatnom vizualnom kontrolom kvalitete ostalih međuproizvoda smanjio bi se broj neispravnih gotovih keramičkih pločica i smanjio bi se utrošak energije u procesu proizvodnje. Smanjenje utroška energije se postiže na dva načina: neispravne pločice ne ulaze u fazu paljenja i nije potrebno dodatno usitnjavanje neispravne pečene keramičke pločice.

Budući da se vizualna kontrola keramičkih pločica najviše oslanja na ljudske resurse (operater), ova faza predstavlja ograničavajući faktor u procesu proizvodnje keramičkih pločica. Kontrola kvalitete ovisi o subjektivnom dojmu operatera kao i o fizičkom stanju i koncentraciji operatera. Zbog navedenih nedostataka potrebno je razviti automatizirani računalni sustav za vizualnu kontrolu kvalitete svih međuproizvoda procesa proizvodnje keramičkih pločica.

U ovoj disertaciji predložene su metode za vizualnu kontrolu kvalitete sirovih keramičkih pločica. Razvijen je prototip sustava za vizualnu kontrolu sirovih keramičkih pločica temeljen na računalnom vidu. Prototip se sastoji od kamere sa matričnim osjetnikom, LED difuznih rasvjetnih tijela, svjetlosnih senzora i paralelne procesne jedinice. Sa razvijenim prototipom prikupljene su slike sirovih keramičkih pločica u laboratoriju i industrijskim

uvjetima. U disertaciji su razvijene nove metode za izdvajanje sirove keramičke pločice od pozadine, detekciju neispravnosti na rubovima i uglovima sirove keramičke pločice i ujednačavanje slike površine sirove keramičke pločice.

Razvijena je nova metoda za izdvajanje sirove keramičke pločice od pozadine temeljena na detekcije promjene amplitude u 1D signalu, PBSCD metoda. Metoda je testirana na sintetičkim slikama i slikama prikupljenih u laboratorijskim i industrijskim uvjetima. Rezultati su uspoređeni s metodama za binarizaciju slike temeljene na globalnom određivanju razine praga (Otsu metoda, metoda minimalne pogreške i metoda minimalne razlike entropija). Za provedene testove određena je pogreška binarizacije za sve metode. Predložena PBSCD metoda najbolje segmentira sliku sirove pločice, tj. pokazuje najbolje rezultate binarizacije na svim primjerima slika u odnosu na ostale razmatrane metode. Isti rezultati su dobiveni i vizualnom ocjenom binarizacije za 200 testiranih slika. Razvijeni su i implementirani sekvencijalni i paralelni algoritmi PBSCD metode. Testiranje mogućnosti rada PBSCD metode u stvarnom vremenu izvedeno je s obe implementacije. Vremena izvođenja paralelne implementacije PBSCD metode na CUDA platformi približno su jednaka vremenima izvođenja Otsu metode (metoda s najmanjim brojem operacija). Vrijeme izvođenja CUDA PBSCD algoritma u najlošijem slučaju ne prelazi 13 ms tj. vrijeme izvođenja je manje od 3 % ukupno dostupnog vremena za vizualnu inspekciju sirovih pločica.

Razvijena je unaprijeđena metoda za detekciju neispravnosti na rubovima i uglovima sirove keramičke pločice. Predložena metoda je testirana na sintetičkim binarnim slikama i slikama prikupljenim u laboratorijskim i industrijskim uvjetima. Na sintetičkim binarnim slikama analizirana je mogućnost detekcije neispravnosti na rubovima i uglovima sirove keramičke pločice s obzirom na veličinu neispravnosti kao i način povećanja mogućnosti detekcije neispravnosti predložene metode. Za slike prikupljene u laboratorijskim uvjetima s neispravnostima različitih dimenzija i površine, metoda je uspješno detektirala više od 97 % neispravnosti. Također, nije dodatno detektirana niti jedna neispravnost, tj. niti jedna sirova pločica bez neispravnosti na rubovima ili uglovima nije proglašena neispravnom. Za slike prikupljene na proizvodnoj liniji u industrijskim uvjetima metoda je uspješno detektirala 100 % neispravnosti i niti jedna ispravna sirova pločica nije proglašena neispravnom, tj. nije detektirana niti jedna dodatna neispravnost. Iz dobivenih rezultata može se zaključiti da predložena metoda daje zadovoljavajuće rezultate detekcije neispravnosti na rubovima i uglovima sirovih pločica. Razvijen je i implementiran sekvencijalni i paralelni algoritam predložene metode za detekciju neispravnosti na rubovima i uglovima sirovih. Vrijeme izvođenja sekvencijalne implementacije ne prelazi 19 ms tj. manje od 4 % ukupno dostupnog

vremena za vizualnu inspekciju sirovih pločica. Paralelna OpenMP implementacija predložene metode dodatno snižava vrijeme izvođenja. U najlošijem slučaju vrijeme izvođenja paralelnog algoritma ne prelazi 9 ms tj. manje od 2 % ukupno dostupnog vremena za vizualnu inspekciju sirovih pločica.

Razvijena je nova metoda za ujednačavanje slike površine sirove keramičke pločice. Metoda na temelju ulazne slike sirove keramičke pločice bez neispravnosti na rubovima i uglovima za rezultat daje ujednačenu sliku površine sirove keramičke pločice. Na ujednačenoj slici provodi se naknadna analiza neispravnosti na površini sirovih keramičkih pločica. Za predloženu metodu definirana je pogreška u ujednačavanju slike te je ista uspoređena s pogreškom standardne metode za uklanjanje utjecaja rotacije sirove keramičke pločice geometrijskom transformacijom s poznatim kutom rotacije γ . Dobiveni rezultati su pokazali da predložena metoda za manje kutove rotacije ($-0.5^\circ < \gamma < 0.5^\circ$) rezultira s manjom vrijednosti pogreške za sve testirane slike. Također pokazano je da za jednobojne sirove keramičke pločice metoda za ujednačavanje slike površine sirove pločice rezultira manjom pogreškom za kutove rotacije $-0.7^\circ < \gamma < 0.7^\circ$ u odnosu na standardnu metodu. Predložena metoda, za razliku od standardne geometrijske transformacije, otklanja i utjecaj izobličenja pravokutnosti koji može nastati zbog nepravilnog položaja kamere u odnosu na sirovu pločicu ili zbog nepotpuno uklonjene distorzije slike. Kut rotacije je moguće ograničiti na interval $-0.5^\circ < \gamma < 0.5^\circ$ uz upotrebu dodatnih usmjerivača na proizvodnoj liniji. S dodatnim usmjerivačima predložena metoda daje manje rezultate pogreške od standardne geometrijske transformacije. Općenito, može se zaključiti da za $-0.5^\circ < \gamma < 0.5^\circ$ metoda za ujednačavanje slike površine sirove keramičke pločice kao rezultat daje novu sliku površine sirove pločice pripremljenu za detekciju neispravnosti na površini. Razvijen je i implementiran sekvencijalni i paralelni algoritam predložene metode. Vrijeme izvođenja metode za ujednačavanje slike sirove keramičke pločice uspoređeno je sa vremenom izvođenja metode za geometrijsku transformaciju sa bikubičnom interpolacijom. Prosječno vrijeme izvođenja predložene metode ovisno je o veličini slike i kutu rotacije γ . U najlošijem slučaju (najveće prosječno vrijeme izvođenja), prosječno vrijeme izvođenja sekvencijalne implementacije je 234.7 ms, tj. 1.98 puta veće vrijeme u odnosu na vrijeme izvođenja metode za geometrijsku transformaciju sa bikubičnom interpolacijom. Paralelna OpenMP implementacija predložene metode smanjuje najlošije vrijeme izvođenja sekvencijalne implementacije na 67.7 ms, tj. vrijeme izvođenja paralelnog algoritma ne prelazi 12 % ukupno dostupnog vremena za vizualnu inspekciju sirovih pločica.

Vremenski interval dostupan za vizualnu kontrolu kvalitete sirove keramičke pločice postavljen je na 600 ms ili 100 pločica u minuti. Iz dobivenih rezultat svih razvijenih metoda može se zaključiti da je ukupno vrijeme izvođenja predloženih metoda manje od 90 ms, tj. manje do 15 % ukupno dostupnog vremena za vizualnu inspekciju neispravnosti sirovih keramičkih pločica. Također, ako uzmemo u obzir i vrijeme potrebno za prikupljanje slike sirove pločice i uklanjanje distorzije, ukupno vrijeme nije veće od 192 ms, tj. manje od 32 % ukupno dostupnog vremena za vizualnu inspekciju neispravnosti sirovih keramičkih pločica. Općenito se može zaključiti da su razvijene metode pogodne za rad u stvarnom vremenu prema postavljenim vremenskim intervalima.

Na temelju prikazanih i analiziranih rezultata može se zaključiti da su ostvareni sljedeći znanstveni doprinosi:

- razvijen je i implementiran novi algoritam za izdvajanje sirove keramičke pločice od pozadine u sivoj slici u stvarnom vremenu na GPU platformi,
- razvijen je i implementiran unaprijeđeni algoritam za detekciju neispravnosti na uglovima i rubovima sirove keramičke pločice u stvarnom vremenu,
- razvijena je i implementirana nova metoda ujednačavanja slike površine sirove keramičke pločice u stvarnom vremenu i
- sve su metode testirane i verificirane u laboratoriju i industrijskoj okolini.

Budući rad temeljen na ovoj disertaciji obuhvaća razvijanje paralelnih algoritama opisanih metoda za detekciju neispravnosti na površini sirovih pločica. Ako metode ne pokažu zadovoljavajuće rezultate potrebno je implementirati dodatne metode koje će s dovoljno visokom točnošću detektirati neispravnosti na površini sirovih keramičkih pločica. Također, potrebno je unaprijediti prototip sustava za vizualnu inspekciju sirovih pločica tako da može detektirati i neispravnosti na reljefnim sirovim keramičkim pločicama. U ovu svrhu na prototip je potrebno dodati linijsku kameru koja prikuplja sliku reflektiranog svjetla od reljefa i implementirati odgovarajuće metode za detekciju neispravnosti na reljefu sirove pločice.

9. LITERATURA

- [1] V. Hocenski, »Novi pristup smanjenju utjecaja keramičke industrije na okoliš temeljen na neuronskim mrežama,« Sveučilište u Zagrebu Fakultet kemijskog inženjerstva i tehnologije, Zagreb, 2012.
- [2] D. Stock, »World production and consumption of ceramic tile,« *Ceramic World Review*, svez. 78, pp. 56-66, 2008.
- [3] European Commission, "Reference Document on Best Available Techniques in the Ceramic Manufacturing Industry," European Commission, 2007.
- [4] ISO, »ISO 13006:2012: Ceramic tiles - definitions, classification characteristics and marking,« International Organization for Standardization, Ženeva, 2012.
- [5] European Standard, »EN 14411:2012: Ceramic tiles - definitions, classification, characteristics, evaluation of conformity and marking,« European Committee for Standardization, Brisel, 2012.
- [6] T. Keser, »Automatizirani inteligentni sustav za klasiranje keramičkih pločica,« Sveučilište J. J. Strossmayera Elektrotehnički fakultet Osijek, Osijek, 2009.
- [7] T. Matić, I. Vidović i Ž. Hocenski, »Real time contour based ceramic tile edge and corner defects detection,« *Tehnički vjesnik*, svez. 20, br. 6, pp. 1063-1070, 2013.
- [8] Surface inspection, »Vision systems: Processmaster 5G,« [Mrežno]. Dostupno: <http://www.surface-inspection.com/en-US/Products/Vision-systems.aspx?idC=64605&LN=en-US>. [Pokušaj pristupa 10 veljača 2014].
- [9] C. Boukouvalas, F. De Natale, G. De Toni, J. Kittler, R. Marik, M. Mirmehdi, M. Petrou, P. Le Roy, R. Salgari i G. Vernazza, »ASSIST: automatic system for surface inspection and sorting of tiles,« *Journal of Materials Processing Technology*, svez. 82, br. 1, pp. 179-188, 1998.
- [10] Errece maquinaria ceramica, »Planar caliber,« Errece maquinaria ceramica, [Mrežno]. Dostupno: http://errece.es/productos/planar-caliber_measuring-devices_p10_en.php. [Pokušaj pristupa 3. veljače 2014].
- [11] R. Bowman, »Redefining first quality ceramic tiles,« *Tile Today*, svez. 14, br. 50, pp. 34-40, 2006.
- [12] Ž. Hocenski, A. Dizdar i V. Hocenski, »Illumination design of a control system for

- visual inspection of ceramic tiles,« u *Proceedings of the IEEE International Symposium on Industrial Electronics, 2008*, Cambridge, 2008.
- [13] Ž. Hocenski, K. Sobol i R. Mijaković, »LED panel illumination design of a control system for visual inspection of ceramic tiles,« u *Proceedings of the IEEE International Symposium on Industrial Electronics, 2010*, Bari, 2010.
- [14] Keramika Modus d.o.o., »Keramika Modus,« Keramika Modus d.o.o., [Mrežno]. Dostupno: <http://www.keramika-modus.hr/>. [Pokušaj pristupa 6 Veljača 2014].
- [15] R. C. Gonzalez i R. E. Woods, *Digital Image Processing (3rd Edition)*, Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.
- [16] M. Wang, J. Pan, S. Chen i H. Li, » A method of removing the uneven illumination phenomenon for optical remote sensing image,« u *Proceedings of the 2005 IEEE International Geoscience and Remote Sensing Symposium*, Seoul, Korea, 2005.
- [17] W. Wang i X. Cui, »A Background correction method for particle image under non-uniform illumination conditions,« u *Proceedings of the 2010 2nd International Conference on Signal Processing Systems*, Singapur, 2010.
- [18] X. Yuan, Y. Meng i X. Y. Wei, »Illumination normalization based on homomorphic wavelet filtering for face recognition,« *Journal of Information Science and Engineering*, svez. 29, br. 3, pp. 579-594, 2013.
- [19] D. G. Bailey, »A new approach to lens distortion correction,« u *Proceedings of the Image and Vision Computing Conference 2002*, Dunedin, Novi Zeland, 2002.
- [20] A. E. Conrady, »Decentered lens-systems,« *Monthly Notices of the Royal Astronomical Society*, svez. 79, br. 1, p. 384–390, 1919.
- [21] D. C. Brown, » Decentering distortion of lenses,« *Photogrammetric Engineering and Remote Sensing*, svez. 7, br. 1, pp. 444-462, 1966.
- [22] G. Bradski, »The OpenCV Library,« *Dr. Dobb's Journal of Software Tools*, svez. 25, br. 11, pp. 120-126, 2000.
- [23] J. J. Staal, M. D. Abramoff, M. Niemeijer, M. Viergever i B. van Ginneken, »Ridge based vessel segmentation in color images of the retina,« *IEEE Transactions on Medical Imaging*, svez. 23, br. 4, pp. 501-509, 2004.
- [24] D. Martin, C. Fowlkes, D. Tal i J. Malik, »A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring

- Ecological Statistics,« u *Proceedings Eighth IEEE International Conference on Computer Vision*, Vancouver, 2001.
- [25] J. C. Russ i J. C. Russ, *Introduction to Image Processing and Analysis*, Boca Raton, FL, USA: CRC Press, Inc., 2007.
- [26] E. R. Davies, *Machine Vision: Theory, Algorithms, Practicalities*, San Francisco, CA, USA}: Morgan Kaufmann Publishers Inc., 2004.
- [27] M. Sezgin i B. Sankur, »Survey over image thresholding techniques and quantitative performance evaluation,« *Journal of Electronic Imaging*, svez. 13, br. 1, pp. 146-168, 2004.
- [28] S. Krinidis i M. C. V. Krinidis, »An Empirical Method for Threshold Selection,« *International Journal of Signal Processing, Image Processing and Pattern Recognition*, svez. 5, br. 2, pp. 101-114, 2012.
- [29] N. Otsu, »A threshold selection method from gray-level histograms,« *IEEE Transactions on Systems, Man and Cybernetics*, svez. 9, br. 1, pp. 62-66, 1979.
- [30] Z. Hou, Q. Hu i N. W. L., »On minimum variance thresholding,« *Pattern Recognition Letters*, svez. 27, br. 14, pp. 1732-1743, 2006.
- [31] H.-F. Ng, »Automatic Thresholding for Defect Detection,« *Pattern Recogn. Letters*, svez. 27, br. 14, pp. 1644-1649, 2006.
- [32] J. N. Kapur, P. K. Sahoo i A. K. C. Wong, »A new method for gray-level picture thresholding using the entropy of the histogram,« *Computer Vision, Graphics, and Image Processing*, svez. 29, br. 3, pp. 273-285, 1985.
- [33] A. D. Brink i N. E. Pendock, »Minimum cross-entropy threshold selection,« *Pattern recognition*, svez. 29, br. 1, pp. 179-188, 1996.
- [34] M. H. Horng, J. T. W. i C. J. Y., »Multilevel Minimum Cross Entropy Threshold Selection based on Honey Bee Mating Optimization,« u *International MultiConference of Engineers and Computer Scientists*, Hon Kong, 2009.
- [35] C. H. Li i L. C. K., »Minimum cross entropy thresholding,« *Pattern Recognition*, svez. 26, br. 4, pp. 617-625, 1993.
- [36] I. Hannah, D. Patel i R. Davies, »The use of variance and entropic thresholding methods for image segmentation,« *Pattern Recognition*, svez. 28, br. 8, pp. 1135-1143, 1995.

- [37] J. Kittler i J. Illingworth, »Minimum error thresholding,« *Pattern Recognition*, svez. 19, br. 1, pp. 41-47, 1986.
- [38] Q. Z. Ye i P. E. Danielsson, »On minimum error thresholding and its implementations,« *Pattern Recognition Letters*, svez. 7, br. 4, pp. 201-206, 1988.
- [39] F. Jiulun i X. Winxin, »Minimum error thresholding: A note,« *Pattern Recognition Letters*, svez. 18, br. 8, pp. 705-709, 1997.
- [40] J. M. Valiente, F. Acebron i F. Lopez, »A ceramic tile inspection system for detecting corner defects,« u *Proceedings of the pattern recognition and image analysis conference (SNRFA'2001)*, Castelon, Spain, 2001.
- [41] Ž. Hocenski i T. Keser, »Failure detection and isolation in ceramic tile edges based on contour descriptor analysis,« u *Mediterranean Conference on Control & Automation, MED '07*, Athens, Greece, 2007.
- [42] D. M. Tsai, »A fast thresholding selection procedure for multimodal and unimodal histograms,« *Pattern Recognition Letters*, svez. 16, br. 6, pp. 653-666, 1995.
- [43] Q. C., X. Bo-li, L. Jun i K. Gang-yao, »An Approach on Analyzing Histogram and Selecting Threshold,« u *Computer Science and Software Engineering Conference 2008*, Wuhan, Hubei, China, 2008.
- [44] P.-S. Liao, T.-S. Chen i P.-C. Chung, »A fast algorithm for multilevel thresholding,« *Journal of Information Science and Engineering*, svez. 17, pp. 713-727, 2001.
- [45] S. Kullback, *Information theory and statistics*, New York: John Wiley, 1959.
- [46] M. K. Steven, *Fundamentals of Statistical Signal Processing: Estimation Theory*, New Jersey: Prentice-Hall, Inc., 1993.
- [47] H. Freeman, »On the encoding of arbitrary geometric configurations,« *Institute of Radio Engineers, trans. on Electronic Computers*, svez. 10, pp. 260-268, 1961.
- [48] U. Qidwai i C. Chen, *Digital Image Processing: An Algorithmic Approach with MATLAB*, Boca Raton: Chapman & Hall/CRC, 2009.
- [49] T. Matić, I. Aleksi i Ž. Hocenski, »CPU, GPU and FPGA implementations of MALD: ceramic tile surface defects detection algorithm,« *Automatika : časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije*, svez. 55, br. 1, pp. 9-21, 2014.
- [50] Intel Corporation, »Desktop 4th Generation Intel® Core™ Processor Family, Desktop Intel® Pentium® Processor Family, and Desktop Intel® Celeron®,« Intel

- Corporation, 2013.
- [51] Advanced Micro Devices Inc., »Family 15h Models 30h-3Fh AMD A-Series Accelerated Processor Product Data Sheet,« Advanced Micro Devices Inc., 2014.
- [52] M. Gschwind, H. P. Hofstee, B. Flachs, M. Hopkins, Y. Watanabe i T. Yamazaki, »Synergistic Processing in Cell's Multicore Architecture,« *IEEE Micro*, svez. 26, br. 2, pp. 10-24, 2006.
- [53] E. MuraliKrishnan, E. Gangadharan i P. NirmalKumar, »FPGA based symmetric multi-core processors for optimized performance of H.264 encoder,« u *International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom), 2010*, Kottayam, Kerala, India, 2010.
- [54] NVIDIA Corporation, »TESLA GPU Accelerators for workstations,« NVIDIA Corporation, [Mrežno]. Dostupno: <http://www.nvidia.com/object/tesla-workstations.html>. [Pokušaj pristupa 3 veljača 2014].
- [55] SRCE - Sveučilišni računski centar, »Klaster,« 2010. [Mrežno]. Dostupno: <http://www.srce.unizg.hr/proizvodi-i-usluge/racunalni-resursi/racunalni-klaster-isabella/klaster-isabella-o-klasterima/>. [Pokušaj pristupa 24 Ožujak 2014].
- [56] T. Matić, M. Žulj i Ž. Hocenski, »Comparison of general purpose graphic processor units as a substitution for traditional processors,« u *Proceeding of the 30th international conference Science in Practice SIP2012*, Pecs, Mađarska, 2013.
- [57] A. R. Brodtkorb, T. R. Hagen i M. L. Sætra, »Graphics processing unit (GPU) programming strategies and trends in GPU computing,« *Journal of Parallel and Distributed Computing*, svez. 73, br. 1, pp. 4-13, 2013.
- [58] M. J. Flynn, »Some Computer Organizations and Their Effectiveness,« *IEEE Transactions on Computers*, svez. 21, br. 9, pp. 948-960, 1972.
- [59] N. K. Govindaraju, B. Lloyd, Y. Dotsenko, B. Smith i J. Manferdelli, »High Performance Discrete Fourier Transforms on Graphics Processors,« u *Proceedings of the 2008 ACM/IEEE Conference on Supercomputing*, Piscataway, New Jersey, 2008.
- [60] NVIDIA Corporation, »cuFFT,« NVIDIA Corporation, 2014. [Mrežno]. Dostupno: <https://developer.nvidia.com/cuFFT>. [Pokušaj pristupa 30 Ožujak 2014].
- [61] R. Rahman, »The Intel Math Kernel Library and its Fast Fourier Transform Routines,« Intel Corporation, 2014. [Mrežno]. Dostupno: <http://software.intel.com/en-us/articles/the-intel-math-kernel-library-and-its-fast-fourier-transform-routines>.

- [Pokušaj pristupa 30 Ožujak 2014].
- [62] M. Gipp, G. Marcus, N. Harder, A. Suratane, K. Rohr, R. König i R. Männer, »Haralick's Texture Features Computed by GPUs for Biological Applications,« *IAENG International Journal of Computer Science*, svez. 36, br. 1, pp. 127-137, 2000.
- [63] L. You, Z. Kaiyong, C. Xiaowen i J. Liu, »Speeding up k-Means algorithm by GPUs,« *Journal of Computer and System Sciences*, svez. 79, br. 2, pp. 216-229, 2013.
- [64] R. Wu, B. Zhang i M. Hsu, »Clustering Billions of Data Points Using GPUs,« u *Proceedings of the Combined Workshops on UnConventional High Performance Computing Workshop Plus Memory Access Workshop*, Ischia, Italy, 2009.
- [65] V. K. Nimmagadda, A. Akoglu, S. Hariri i T. Moukabary, »Cardiac Simulation on multi-GPU Platform,« *Journal of Supercomputing*, svez. 59, br. 3, pp. 1360-1378, 2012.
- [66] NVIDIA Corporation, »NVIDIA CUDA programming guide version 2.1,« NVIDIA Corporation, Santa Clara, California, USA, 2008.
- [67] J. Nickolls, I. Buck, M. Garland i K. Skadron, »Scalable parallel programming with CUDA,« *Queue*, svez. 6, br. 2, pp. 40-53, 2008.
- [68] T. Matić i Ž. Hocenski, »Parallel processing with CUDA in ceramic tiles classification,« u *Proceedings of the 14th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, Cardiff, UK, 2010.
- [69] H. Wong, M.-M. Papadopoulou, M. Sadooghi-Alvandi i A. Moshovos, »Demystifying GPU microarchitecture through microbenchmarking,« u *2010 IEEE International Symposium on Performance Analysis of Systems Software (ISPASS)*, White Plains, New York, USA, 2010.
- [70] S. Cook, *CUDA programming: a developer's guide to parallel computing with GPUs*, San Francisco, California, USA: Morgan Kaufmann Publishers Inc., 2013.
- [71] M. Harris, »Optimizing Parallel Reduction in CUDA,« 2008. [Mrežno]. Dostupno: <http://developer.download.nvidia.com/assets/cuda/files/reduction.pdf>. [Pokušaj pristupa listopad 2013].
- [72] A. Z. Arifin i A. Asano, »Image Segmentation by Histogram Thresholding Using Hierarchical Cluster Analysis,« *Pattern Recognition Letters*, svez. 27, br. 13, pp. 1515-1521, 2006.

- [73] I. Aleksi, D. Kraus i Ž. Hocenski, »Multi-Language Programming Environment for C++ Implementation of SONAR signal processing by linking with MATLAB external interface and FFTW,« u *Proceedings of the 53rd International Symposium ELMAR 2011*, Zadar, Hrvatska, 2011.
- [74] Microsoft, »Visual Studio 2010,« Microsoft, [Mrežno]. Dostupno: <http://msdn.microsoft.com/en-us/library/dd831853%28v=vs.100%29.aspx>. [Pokušaj pristupa 10 Ožujak 2014].
- [75] J. M. Valiente, F. Acebron i F. Lopez, »An automatic visual inspection system for detecting corner defects,« u *Proceedings of the IASTED international conference on signal processing and communications*, Canary Islands, Spain, 1998.
- [76] R. O. Duda i P. E. Hart, »Use of the Hough transformation to detect lines and curves in pictures,« *Communications of the ACM*, svez. 15, br. 1, pp. 11-15, 1972.
- [77] M. Mansoori, H. Tajik, G. Mohamadi i M. Pashna, »Edge defect detection in ceramic tile based on boundary analysis using Fuzzy thresholding and Radon transform,« u *Proceedings of the IEEE international symposium on signal processing and information technology*, Sarajevo, Bosna i Hercegovina, 2008.
- [78] O. Terrades i E. Valveny, »Radon transform for linear symbol representation,« u *Proceedings of the Seventh international conference on document analysis and recognition, 2003.*, Edinburgh, Scotland, UK, 2003.
- [79] Y. Yong, Z. Chongxun i L. Pan, »A novel fuzzy C-means clustering algorithm for image thresholding,« *Measurement science review*, svez. 4, br. 1, pp. 11-19, 2004.
- [80] S. Suzuki i K. Abe, »Topological structural analysis of digitized Binary images by border following,« *Computer vision, graphics, and image processing*, svez. 30, br. 1, pp. 32-46, 1985.
- [81] T. Miyatake, H. Matsushima i M. Ejiri, »Contour representation of binary images using run-type direction codes,« *Machine Vision and Applications*, svez. 9, br. 4, pp. 193-200, 1997.
- [82] A. Dziech, *Mobile Robots: Perception & Navigation*, Njemačka: Pro Literatur Verlag, 2007.
- [83] A. G. Ghuneim, »Radial Sweep,« [Mrežno]. Dostupno: http://www.imageprocessingplace.com/downloads_V3/root_downloads/tutorials/contour_tracing_Abeer_George_Ghuneim/ray.html. [Pokušaj pristupa 9 srpanj 2013].

- [84] T. Pavlidis, Algorithms for graphics and image processing, 1982: Computer Science Press.
- [85] G. Wagenknecht, »A contour tracing and coding algorithm for generating 2D contour codes from 3D classified objects,« *Pattern Recognition*, svez. 40, br. 4, pp. 1294-1306, 2007.
- [86] T. Matić, I. Vidović and Ž. Hocenski, "Edge and corner defects detection on crude ceramic tiles," in *Proceedings of the thirty-first International Conference Science in Practice*, Bremen, Njemačka, 2013.
- [87] R. Galić, Statistika, Osijek: Elektrotehnički fakultet Osijek, 2004.
- [88] R. Collins, »Computer vision I, lecture notes: Smoothing,« Penn State University, Pennsylvania, USA, 2007.
- [89] OpenMP Architecture Review Board, »OpenMP Application Program Interface,« srpanj 2013. [Mrežno]. Dostupno: <http://www.openmp.org/mp-documents/OpenMP4.0.0.pdf>. [Pokušaj pristupa 7 veljača 2014].
- [90] A. Monadjemi, "Towards efficient texture classification and abnormality detection," Univerity of Bristol, Bristol, UK, 2004.
- [91] D. O. Aborisade i T. S. Ibiyemi, »Ceramic wall tile quality calssification training algorithms using statisitcal approach,« *Research Journal of Applied Sciences*, svez. 2, br. 12, pp. 1255-1260, 2007.
- [92] Ž. Hocenski, S. Rimac-Drlje i T. Keser, »Automatic Inspection of Defects in Plain and Texture Surfaces,« u *Proceedings of the 5th IEEE Int. Conf. on Intelligent Engineering Systems 2001, INES 2001.*, Helsinki, Stockholm, 2001.
- [93] D. M. Tsai i B. Hsiao, »Automatic surface inspection using wavelet reconstruction,« *Pattern Recognition*, svez. 34, br. 6, pp. 1285-1305, 2001.
- [94] H. Elbehiery, A. Hefnawy i M. Elewa, »Surface defects detection for ceramic tiles using image processing and morphological techniques,« u *Proceedings of World Academy of Science, Engineering and Technology*, 2005.
- [95] C. E. Costa i M. Petrou, »Automatic registration of ceramic tiles for the purpose of fault detection,« *Machine Vision and Applications*, svez. 11, br. 5, pp. 225-230, 2000.
- [96] J. M. Valiente, F. Lopez, F. Acebron i E. Perez, »An image registration method for ceramic tile inspection purposes,« u *Proceedings of the 2001 international conference*

- on quality control by artificial vision*, Le Creusot, Francuska, 2001.
- [97] F. Acebrón, F. López, J. M. Valiente i J. R. Navarro, »Surface defect detection on fixed pattern ceramic tiles,« u *Proceedings of the IASTED International Conference on Visualization, Imaging, and Image Processing 2002*, Marbella, Spain, 2002.
- [98] Ž. Hocenski, T. Keser i A. Baumgartner, »A Simple and Efficient Method for Ceramic Tile Surface Defects Detection,« u *Proceedings of 2007 International Symposium on Industrial Electronics*, Vigo, Spain, 2007.
- [99] G. M. A. Rahaman i M. M. Hossain, »Automatic Defect detection and classification technique from image: A special case using ceramic tiles,« *International Journal of Computer Science and Information Security*, svez. 1, br. 1, pp. 22-30, 2009.
- [100] Y. Meena i A. Mittal, »Blobs and cracks detection on plain ceramic tile surface,« *International Journal of Advanced Research in Computer Science and Software Engineering*, svez. 3, br. 7, pp. 647-652, 2013.
- [101] R. Haralick, K. Shanmugam i I. Dinstein, »Textural features for image classification,« *IEEE Transactions on Systems, Man and Cybernetics*, svez. 3, br. 6, pp. 610-621, 1973.
- [102] P. Harrington, *Machine learning in action*, Greenwich, Connecticut: Manning Publications, 2012.

10. PRILOZI

10.1. Značajke Basler A102fc kamere

| <i>Značajka</i> | <i>Basler A102fc vrijednost</i> |
|--|--|
| <i>Tip matičnog osjetnika</i> | Sony ICX-285 CCD |
| <i>Maksimalna visina slike</i> | 1038 piksela |
| <i>Maksimalna širina slike</i> | 1388 piksela |
| <i>Širina piksela</i> | 6.45 μm |
| <i>Visina piksela</i> | 6.45 μm |
| <i>Format slike</i> | Mono 8, Raw 8, Raw 16, YUV 4:2:2 |
| <i>Napajanje</i> | +8.0 V – +36.0 V istosmjerna struja, 3.5 W |
| <i>Prijenos podataka</i> | IEEE 1394 |
| <i>Maksimalna dužina kabela za prijenos podataka</i> | 4.5 m |
| <i>Postolje za leću</i> | C – mount |
| <i>Dimenzije kamere</i> | 67.3 × 62.0 × 62.0 mm |
| <i>Masa kamere</i> | 240 g |

10.2. CUDA GPU značajke u ovisnosti o CC broju

| Značajka | CC broj (Compute Capability broj) | | | | |
|---|-----------------------------------|--------|------------|------------|------------|
| | 1.0 | 2.0 | 3.0 | 3.5 | 5.0 |
| Maksimalan broj dimenzija mreže | 2 | 3 | 3 | 3 | 3 |
| Maksimalan broj x-, y-, z- dimenzija mreže | 65535 | 65535 | $2^{31}-1$ | $2^{31}-1$ | $2^{31}-1$ |
| Maksimalan broj dimenzija bloka | | | 3 | | |
| Maksimalan broj x-, y- dimenzija bloka | 512 | 1024 | 1024 | 1024 | 1024 |
| Maksimalan broj z- dimenzije bloka | | | 64 | | |
| Maksimalan broj niti po bloku | 512 | 1024 | 1024 | 1024 | 1024 |
| Veličina osnove | | | 32 | | |
| Maksimalan broj aktivnih blokova po multiprocesoru | 8 | 8 | 16 | 16 | 32 |
| Maksimalan broj aktivnih osnova po multiprocesoru | 24 | 48 | 64 | 64 | 64 |
| Maksimalan broj aktivnih niti po multiprocesoru | 768 | 1536 | 2048 | 2048 | 2048 |
| Broj 32-bitnih registara po multiprocesoru | 8 K | 32 K | 64 K | 64 K | 64 K |
| Maksimalan broj 32-bitnih registara po niti | 128 | 63 | 63 | 255 | 255 |
| Maksimalna količina dijeljene memorije po multiprocesoru | 16 KB | 48 KB | 48 KB | 48 KB | 64 KB |
| Količina lokalne memorije po niti | 16 KB | 512 KB | 512 KB | 512 KB | 512 KB |
| Veličina konstantne memorije | | | 64 KB | | |
| Cjelobrojne atomik funkcije nad 32-bitnim riječima u globalnoj memoriji | Ne | Da | Da | Da | Da |
| Cjelobrojne atomik funkcije nad 32-bitnim riječima u djeljenoj memoriji | Ne | Da | Da | Da | Da |
| Operacije na relanim brojevima dvostruke preciznosti | Ne | Da | Da | Da | Da |
| Dinamički paralelizam | Ne | Ne | Ne | Da | Da |
| Broj jezgri za aritmetičke operacije na cjelobrojnim brojevima i realnim brojevima jednostruke preciznosti | 8 | 32 | 192 | 192 | 128 |
| Broj specijalnih funkcijskih jedinica za transcendentalne funkcije na realnim brojevima jednostruke preciznosti | 2 | 4 | 32 | 32 | 32 |
| Broj raspoređivača osnova | 1 | 2 | 4 | 4 | 4 |
| Broj instrukcija obrađenih istovremeno po raspoređivaču | 1 | 1 | 2 | 2 | 2 |

Sažetak

U ovoj disertaciji objašnjena je problematika vizualne detekcije neispravnosti na sirovim keramičkim pločicama. Predstavljen je prototip sustava za detekciju neispravnosti na sirovim pločicama temeljen na digitalnoj kameri i paralelnoj procesnoj jedinici. Neispravnosti na sirovim pločicama detaljno su analizirane i kategorizirane. Obrada sive slike dobivena s prototipom predloženog sustava podijeljena je u nekoliko faza: predobrada slike, izdvajanje sirove keramičke pločice od pozadine (binarizacija), detekcija neispravnosti na uglovima i rubovima sirove keramičke pločice, ujednačavanje slike površine sirove keramičke pločice i detekcija neispravnosti na površini sirovih keramičkih pločica. U ovom radu predložena je nova metoda za binarizaciju slike sirove keramičke pločice, unaprijeđena metoda za detekciju neispravnosti na rubovima i uglovima sirove pločice i nova metoda za ujednačavanje slike površine sirove keramičke pločice. Predložene metode su testirane u laboratorijskim i industrijskim uvjetima. Za sve metode razvijeni su i implementirani sekvencijalni i paralelni algoritmi. Rezultati testiranja u laboratorijskim i industrijskim uvjetima pokazali su da predložene metode zadovoljavaju postavljene kriterije za detekciju neispravnosti na sirovim keramičkim pločicama. Također, rezultati vremena izvođenja paralelnih algoritama pokazali su da metode zadovoljavaju sve postavljene vremenske uvjete tj. pogodne su za rad u stvarnom vremenu.

Ključne riječi: *sirova keramička pločica, detekcija neispravnosti, rad u stvarnom vremenu, binarizacija, ujednačavanje slike, paralelni algoritam.*

Improved Real-Time Algorithms for Biscuit Tile Defects Detection

Abstract

In this PhD thesis problem of visual defects detection on biscuit tiles is presented. Prototype visual inspection system for defects detection on biscuit tiles is developed. Proposed prototype is based on area – scan digital camera and a parallel processing unit. Detailed analysis and categorization of defects that appear on biscuit tiles is given. Processing of the gray – scale image captured by the prototype is divided in several phases: preprocessing, binarization, edge and corner defect detection, biscuit tile surface image extraction and detection of surface defects. In this thesis new methods are proposed for image binarization and biscuit tile surface image extraction. Also improved edge and corner defect detection method is developed. Proposed methods were tested in laboratory and factory conditions. For all methods sequential and parallel algorithms were developed and implemented. Results of testing in laboratory and industrial conditions showed that the proposed methods meet the set criteria for defects detection on biscuit tiles. Also, time measurement results of the proposed methods showed that developed parallel algorithms fulfill all set time conditions, i.e. developed methods are suitable for real – time application.

Key words: *biscuit tile, defect detection, real – time application, binarization, surface image extraction, parallel algorithms.*

Životopis

Tomislav Matić, rođen 16. svibnja 1983 u Brčkom BIH, zvanje diplomiranog inženjera elektrotehnike stekao je u ožujku 2007 godine na Elektrotehničkom fakultetu Osijek gdje se iste godine zapošljava se na radnom mjestu višeg laboranta. 2008. godine upisuje poslijediplomski doktorski studij na smjeru računarstvo i komunikacije te prelazi na radno mjesto asistenta u istoj instituciji. Održava nastavu iz skupine računalnih predmeta na preddiplomskom i diplomskom studiju elektrotehnike i računarstva (Digitalna elektronika, Pouzdanost i dijagnostika računalnih sustava) te sudjeluje kao sumentor na diplomskim i završnim radovima. Glavna područja njegovog znanstvenog i stručnog interesa su obrada slike u stvarnom vremenu, računalni vid i GPGPU programiranje. Suradnik je i istraživač na znanstvenom projektu MZOŠ RH (Distribuirano računalno upravljanje u transportu i industrijskim pogonima), te međunarodnom projektu TEMPUS (Colaborative Internationalisation of Software Engineers In Croatia). Boravio je na više stranih institucija u okviru istraživačkih i nastavnih studijskih boravaka: Hochschule Bremen, University Paderborn, Faculty of Computer Science, Electrical Engineering and Mathematics Software Quality Lab i Malardalen University. Autor je i suautor znanstvenih radova u zbornicima konferencija, znanstvenim časopisima, te dva sveučilišna udžbenika.

Tomislav Matić dipl. ing.