

# Implementacija algoritma za poboljšanje kontrasta slike u realnom vremenu na programabilnoj sekvencijanoj mreži

Dragan Đukarić, Vladimir Marinković, Mihajlo Katona, Vladimir Zlokolica

**Sadržaj** — Rad opisuje implementaciju algoritma za poboljšanje kontrasta slike u realnom vremenu na osnovu globalnog histograma, detaljno opisanog u [1]. Implementacija i ispitivanje algoritma je izvršeno na platformi CHIPit Gold zasnovanoj na programabilnim sekvencijanim mrežama (u daljem tekstu FPGA).

**Ključne reči** — FPGA, HDL, histogram, kontrast, SystemC, Verilog.

## I. UVOD

**Z**A razvoj i verifikaciju algoritama najčešće se koriste viši programski jezici i alati poput MATLAB-a koji omogućavaju implementaciju bez ograničavajućih faktora vezanih za resurse. Nasuprot tome implementacija u realnom vremenu zahteva namenske platforme zbog nedovoljne procesorske snage opštenamenskih procesora. Platforme zasnovane na FPGA kolima predstavljaju opštenamenske platforme za rad u realnom vremenu pa se uz minimalan dodatni napor (razvoj dodatnih ploča za ulazni i izlaznu spregu) mogu iskoristiti za verifikaciju algoritama u realnom vremenu [2]. Trenutno raspoloživi alati ne pružaju pouzdanu mogućnost prelaska sa jezika višeg nivoa na nivo fizičke implementacije za FPGA kola tako da taj proces predstavlja dodatni napor vezan za implementaciju u realnom vremenu. Biblioteke poput SystemC-a omogućavaju prilagođenje aplikacija pisanih na jezicima višeg nivoa za realizaciju u realnom vremenu kao i određivanje potrebnih resursa. Prelaz sa ovakvog opisa sistema na neki od jezika za opis fizičke arhitekture predstavlja jednostavan korak koji podržavaju neki od komercijalno dostupnih alata.

Polazeći od referentnog opisa u C++ programskom

Ovaj rad je delimično finansiran od Ministarstva za nauku Republike Srbije, projekat 161003, od 2008. god.

Dragan Đukarić, Fakultet tehničkih nauka u Novom Sadu, Srbija, odsek za računarsku tehniku i međuračunarske komunikacije, Trg Dositeja Obradovića 6 Novi Sad (e-mail: Dragan.Djukaric@rt-rk.com)

Vladimir Marinković, Fakultet tehničkih nauka u Novom Sadu, Srbija, odsek za računarsku tehniku i međuračunarske komunikacije, Trg Dositeja Obradovića 6 Novi Sad (e-mail: Vladimir.Marinkovic@rt-rk.com)

Mihajlo Katona, Fakultet tehničkih nauka u Novom Sadu, Srbija, odsek za računarsku tehniku i međuračunarske komunikacije, Trg Dositeja Obradovića 6 Novi Sad (e-mail: Mihajlo.Katona@rt-rk.com)

Vladimir Zlokolica, Fakultet tehničkih nauka u Novom Sadu, Srbija, odsek za računarsku tehniku i međuračunarske komunikacije, Trg Dositeja Obradovića 6 Novi Sad (e-mail: Vladimir.Zlokolica@rt-rk.com)

jeziku, rad predstavlja opis koraka potrebnih za prelazak na FPGA platformu kao i rezultate ispitivanja ovako realizovanog sistema.

## II. TOK IMPLEMENTACIJE

### A. Prilagođenje referentnog C++ opisa

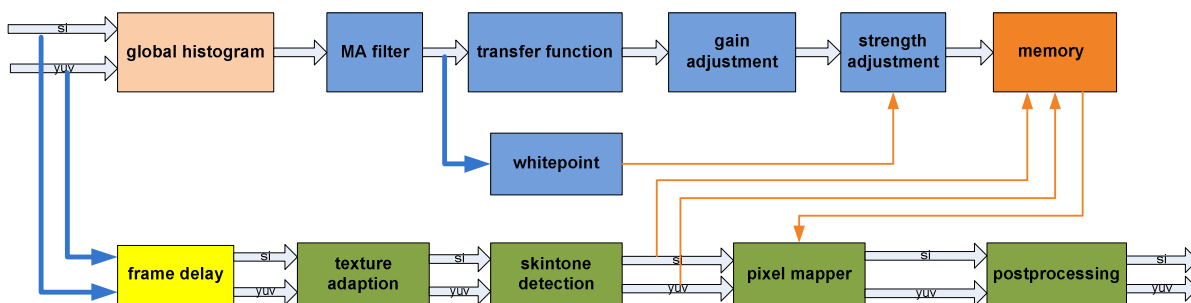
Za rad u realnom vremenu struktura algoritma definisana referentnim opisom u C++ programskom jeziku nije adekvatna zbog korišćenja ugnježenih struktura i bočnih efekata uobičajenih za ovaj programski jezik. Ovakva struktura se mora prilagoditi sekvencijalnom izvršavanju potrebnom za formiranje protočne strukture unutar FPGA kola koja omogućava rad u realnom vremenu. Imajući u vidu dinamiku i količinu podataka ciljnog sistema dodatni problem predstavlja korišćenje prethodnih podataka u trenutnom izračunavanju tako da je neophodno rešiti problem skladistenja velikog broja podataka.

### B. Prevođenje referentnog koda u SystemC model

Nakon prilagođenja referentnog opisa ceo sistem je neophodno modelovati pomoću SystemC biblioteke koja obezbeđuje realizaciju blisku ciljnoj fizičkoj arhitekturi. U ovom koraku je potrebno, gde je to moguće, uvesti paralelno izvršavanje operacija koje doprinosi poboljšanju karakteristika sistema. Obuhvatanje funkcionalnosti delova sistema u nezavisne celine doprinosi boljoj preglednosti i upravljivosti sistema tokom izvršavanja u realnom vremenu i takođe se izvodi u ovom koraku. Sl. 1 prikazuje ciljni izgled procesa obrade ulaznih podataka nakon formiranja protočne strukture i enkapsulacije funkcionalnih celina.

Globalni histogram (Sl. 1 - global histogram) predstavlja broj pojavljivanja piksela istog intenziteta (osvetljaja) na datoj slici [1]. Kako je intenzitet piksela predstavljen vrednošću od 0 do 255 (8 bita) to znači da se za svaki od ovih intenziteta moramo odrediti broj pojavljivanja unutar jedne slike. Moguće je, zarad smanjenja potrebnih resursa, izvršiti kvantizaciju ovih vrednosti na grupu od po 8 pa bi umesto 256 bilo 32 opsega za određivanje ulaznog intenziteta. Ovakva kvantizacija ima za posledicu drastično narušavanje kvaliteta procesa obrade, pa kao takva nije primenjena u konačnom sistemu. Svaki od 256 brojača realizovanih za otkrivanje svakog nivoa osvetljaja ima opseg od 0 do 16777215 (24 bita), što u slici dimenzija 1920x1080 ne može izazvati prekoračenje (2073600 tačaka), ali bi u slučaju potrebe za uštedom resursa, širina

ovih brojača mogla biti smanjena i tada bi, već realizovan, ova kvantizacija je u opsegu -1 do 1 za realizovane



Sl. 1. Koncept modularne, protočne strukture

metod odsecanja (*clipping*) bio mera sprečavanja prekoračenja maksimalne vrednosti. Tačnost ovog modula je istovetna referentnoj u slučaju da maksimalna vrednost brojača nije dostignuta.

Filtar za sprečavanje naglih promena vrednosti histograma bliskih po intenzitetu (engl. *moving average* filter, Sl. 1 – MA filter) normalizuje 17 susednih vrednosti histograma [1]. Implementiran je pomoću 17 protočnih registara i 17 sabirača. Implementacija ovog modula se u potpunosti poklapa sa referentnim zbog celobrojne aritmetike korišćene u referentnoj implementaciji.

Prenosna funkcija (Sl. 1 – transfer function) koja vrednostima intenziteta tačaka ulazne slike dodeljuje vrednost takvu da izlazna slika ima uniformnu raspodelu intenziteta (idealno bi bilo dobiti ravan histogram) određuje se po jednačini (1), gde je  $h$  - vrednosti globalnog histograma u tački  $j$ , a  $H_{dim}$  i  $V_{dim}$  su širina i visina slike.

$$T^{[1]} = \left( \sum_{j=0}^i h[j] \right) * \frac{255}{H_{dim} * V_{dim}} \quad (1)$$

Realizacija u vidu akumulatorske protočne strukture, sa bitskim širinama dovoljnim za očuvanje tačnosti, množačem i sekvencijalnim deliteljem protočne strukture može uneti jedino grešku celobrojnog deljenja.

Pozicija u histogramu do koje se nalazi 95% ukupnog broja tačaka koje učestvuju u formiranju histograma naziva se tačka beline (engl. *whitepoint* - Sl. 1) [1]. Implementacija modula sadrži vremenski filter za sprečavanje naglih promena vrednosti susednih slika, koji je moguće uključiti ili isključiti spoljnim komandama. Implementacija ne odstupa od referentne, a izvršavanje je paralelizovano sa određivanjem prenosne funkcije.

Pojačanje prenosne funkcije predstavlja udaljenost krive prenosne funkcije od referentne koju predstavlja linearna funkcija. Da bi izlazna slika zadržala karakteristike ulazne ovo odstupanje je potrebno kontrolisati [1]. Kontrola pojačanja (Sl. 1 – gain adjustment) se izvodi u dva prolaza (iznad i ispod referentne funkcije) određivanjem pojačanja prenosne funkcije i postavljanjem na maksimalnu dozvoljenu vrednost ukoliko je ona premašena.

U slikama visokog kontrasta potrebno je izbeći previše tamne oblasti (*blackstretch*) i, dodatno, popraviti osvetljaj cele slike (*brightness adjustment*) [1]. Zbog ograničenja platforme, problem izračunavanja *blackstretch* eksponencijalne funkcije je rešen kvantizovanjem (20 nivoa) i preuzimnjem vrednosti iz tabela smeštenih u ROM memoriji (za 20 različitih parametara). Greška koju unosi

kvantizacione nivoe. Konačna vrednost prenosne funkcije dobija se kombinovanjem *blackstretch* funkcije i prenosne funkcije sa izmenjenim osvetljajem, sa težinskim faktorima u skladu sa određenom jačinom prenosne funkcije i koristi se samo u slučaju da je vrednost *whitepoint* manja ili jednaka zadatom pragu kontrolisanim preko  $I^2C$  sprege.

Skladištenje prenosne funkcije se obavlja u dvoprístupnoj memoriji (Sl. 1 – memory) pre konačnog pretvaranja ulazne slike.

Problem skladištenja velike količine ulaznih podataka zbog primene prenosne funkcije na istoj slici na kojoj je i određena, rešen je korišćenjem spoljnih memorijskih resursa. Korišćenje ovih resursa zahteva i dodatnu upotrebu sprežnih podsistema. Obuhvatanje funkcionalnosti kašnjenja celokupne ulazne slike je realizovana u bloku za kašnjenje (Sl. 1 – frame delay) koji sadrži SDRAM kontroler i potrebnu prilagodnu logiku.

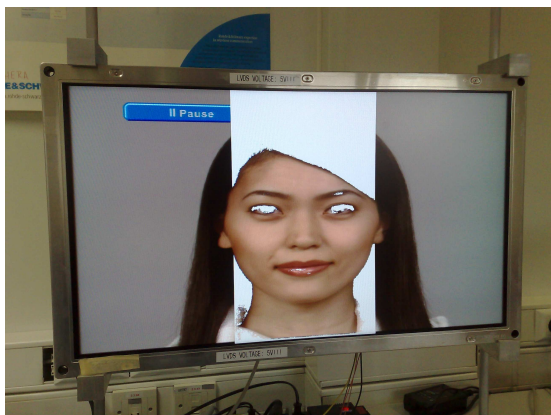
Tekstura u slici se očuva umanjenom primenom algoritma u oblastima tekstura [1]. Za određivanje oblasti tekstura koristi se faktor varijanse za svaku pojedinačnu tačku, koji govori kolika je verovatnoća da odgovarajuća tačka pripada oblasti teksture. Varijanse se određuje među tačkama u prozoru veličine 3x3 tačke (Sl. 1 – texture adaption) i primenjuje se kao težinski faktor kod određivanja izlazne vrednosti tačke na osnovu prenosne funkcije (Sl. 1 – pixel mapper). Unutar slike se usrednjuje varijansa (1-tap filter) na nivou svake pojedinačne linije. Pošto se podaci primaju serijski, neophodno je zakasnuti ulazni tok podataka za jednu liniju, ali i čuvati vrednosti podataka iz prethodne 2 linije, što je realizovano pomoću linijske prihvatne memorije (modul *line\_buffer*). Time je referentna implementacija ovog modula u potpunosti ispraćena.

Ton kože se određuje na osnovu pripadnosti trenutne vrednosti komponenti U i V ulaznih podataka zadatom opsegu vrednosti. Na osnovu otkrivenog tona kože određuje se faktor primene algoritma na konkretnu tačku (Sl. 1 – skintone detection) [1]. Taj faktor je broj u opsegu od 0 do 1, pa je zbog ograničenja platforme na upotrebu celobrojne aritmetike kao kompromisno rešenje izabrano skaliranje sa 256. Greška računanja ovog faktora je dovoljno mala da u krajnjoj primeni na konkretnu tačku, razlika u odnosu na referentnu izvedbu bude u opsegu od -1 do 1.

Određivanje izlaznih vrednosti osvetljaja svake tačke (Sl. 1 – pixel mapper), na osnovu ulaznog toka podataka, vrednosti prenosne funkcije u odgovarajućoj tački,

varijanse i faktora tona kože, implementirano je čitanjem vrednosti iz dvopristupne memorije (Sl. 1 – memory) sa lokacije koja odgovara vrednosti osvetljaja za konkretnu tačku i adekvatnom primenom pomenutih faktora.

Prikaz rada pojedinih blokova sistema je omogućen I<sup>2</sup>C komandama tako da Sl. 2 prikazuje određivanje tonova ulaznog toka podataka u realnom vremenu.



Sl. 2. Prikaz rada modula za detekciju tona kože

Algoritam tokom obrade može da unese u sliku neželjene efekte (artefakte), pa je potrebno naknadno, nakon završene cele obrade, filtrirati izlaznu sliku da bi se neželjeni efekti uklonili [1]. Izlazni tok podataka se propušta kroz niskofrekventni (*lowpass*) filtar sa 17 koeficijenata i blok za usrednjavanje (Sl. 1 - postprocessing). Niskofrekventno filtriranje se izvodi na osnovu razlike obrađene i neobrađene vrednosti komponente osvetljaja - Y. Vrednost razlike može da bude pozitivan ili negativan broj, što predstavlja problem prilikom implementacije množenja sa koeficijentima filtra [3]. Iz tih razloga se u posebnu grupu protočnih registara izdvaja vrednost, a u posebnu grupu znak. Sačuvani znak se dodaje nakon izvršenja svih operacija. Implementacija ovog modula je jednaka referentnom opisu.

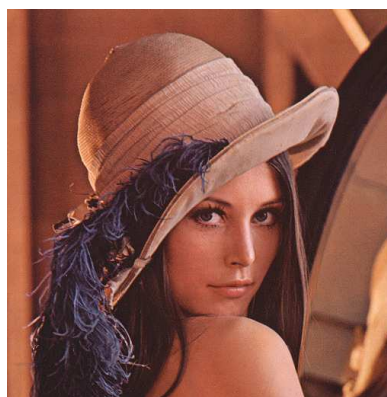
### C. Prevođenje SystemC modela u Verilog opis

Prelazak sa SystemC modela na Verilog (jedan od jezika za opis fizičke arhitekture) model je jednostavan korak koji se može izvesti pomoću komercijalno dostupnih alata ili ručno, kao što je ovde slučaj, jednostavnim prevođenjem funkcija napisanih u SystemC jeziku u proces napisan u Verilog-u. Ovako dobijen model potrebno je mapirati za primenu unutar ciljnih FPGA kola. Očekivano je da SystemC i Verilog model daju identične rezultate.

### III. VERIFIKACIJA ALGORITMA

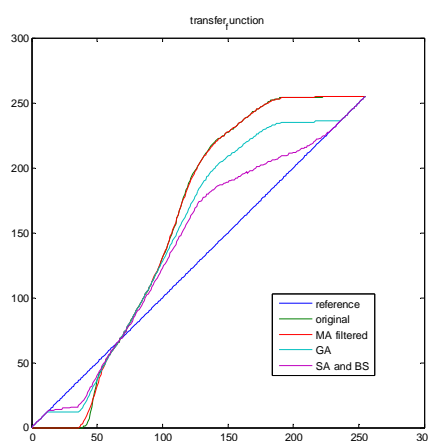
Verifikacija SystemC opisa sistema se izvodi poređenjem izlaznih vrednosti ovakvog opisa sistema sa referentnim vrednostima u zadatim delovima sistema. Prilikom verifikacije sistema koji opisuje ovaj rad poređene su vrednosti nakon svakog bloka obrade koji prikazuje Sl. 1.

Za verifikaciju realizovanog SystemC opisa algoritma korišćen je test vektor kojeg prikazuje Sl. 3.



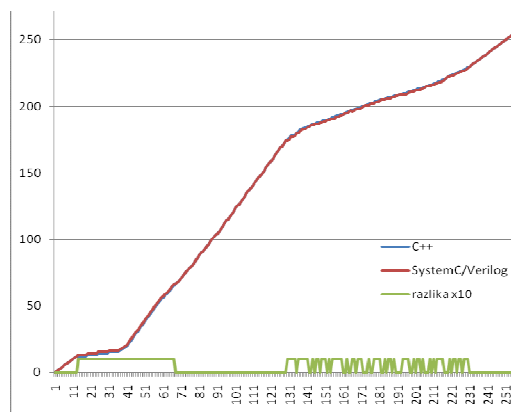
Sl. 3. Ulazni vektor korišćen za ispitivanje sistema

Sl. 4, prikazuje izlaznu vrednost prenosne funkcije kroz sve blokove obrade kao i referentnu prenosnu funkciju.



Sl. 4. Rezultati prilagođavanja prenosne funkcije

Poređenjem prenosnih funkcija za zadati ulazni vektor utvrđeno je da se greška javlja samo na bitu najmanje važnosti (Sl. 5), što je posledica korišćenja celobrojne aritmetike cilnog sistema.



Sl. 5. Rezultati poređenja prenosnih funkcija referentnog i SystemC modela

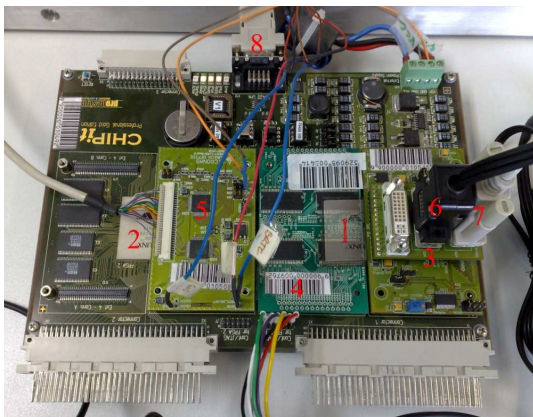
Poređenjem izlaznih slika dobija se maksimalna relativna greška 2 kao posledica propagiranja greške izračunavanja prenosne funkcije kroz dalje nivoe obrade.

Zbog karakteristika ljudskog oka, greške dobijene provedenjem referentnog opisa u SystemC nisu vidljive.

U poređnim izvršavanjem SystemC i Verilog opisa sistema utvrđena je jednakost njihove obrade na bitskom nivou.

#### IV. PLATFORMA ZA ISPITIVANJE

Postojeći sistem CHIPit Gold [4] proširen sa sprežnim pločicama, sa početnom realizacijom koja omogućava paralelni prikaz obrađenog i neobrađenog dela slike je iskorišćen kao osnova koja je proširena algoritmom koji opisuje ovaj rad. Sl. 6 prikazuje izgled platforme sa proširenjima za spregu.



Sl. 6 Izgled platforme

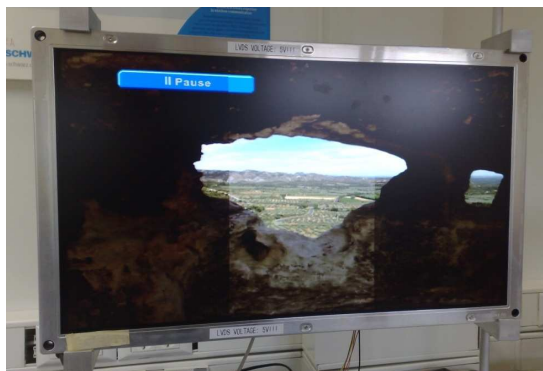
Na slici se vide FPGA kola (označene sa 1 i 2), sprega ulaznog toka podataka (oznaka 3), sprega izlaznog toka podataka (oznaka 5) kao i sprega sa spoljnim memorijskim kolima (oznaka 4).

#### V. ISPITIVANJE SISTEMA

Ispitivanje u realnom vremenu dobijenog sistema je vršeno njegovom integracijom u već postojeće okruženje za verifikaciju algoritama. Ovo okruženje pruža paralelni prikaz ulaznog (neobrađenog) toka podataka i dela nad kojim je izvršena primena algoritma.

Platforma pretvara ulazni analogni video tok podataka u formatu 720p60 (rezolucija 1280x720 na 60 Hz) u digitalni domen ili prihvata već postojeći digitalni tok pomoću CHIPit-ADC pločice i dovodi se na ulaz programabilne sekvencijalne mreže (FPGA1) gde se obrađuje na taktu od 54.5 MHz. Pre ulaska u opisani deo sistema, postojeće okruženje prihvata ulazni digitalni video tok podataka, prilagođava sprežni sistem i izvodi promenu podataka u YUV prostor boja potrebnih za dalju obradu. Nakon toga, podaci se obrađuju kao što je opisano u prethodnim poglavljima rada. Obradeni podaci se prosleđuju drugoj programabilnoj sekvencijalnoj mreži (FPGA2) gde se pretvaraju u RGB prostor boja i dodatno prilagođavaju izlaznom toku podataka odakle se, preko podnožja za proširenje, šalju na izlaz do CHIPIT-LVDS pločice, koja omogućava povezivanje ploče i WXGA panela. Izlazna slika je, kao i ulazna, u formatu 720p60, a zbog karakteristika panela, izlazni takt (tzv. „piksel takt“) je 81 MHz.

Poboljšanje kontrasta u realnom vremenu prikazuje Sl. 7



Sl. 7. Primer slike sa lošim kontrastom (levo i desno) i poboljšanim kontrastom (u sredini)

Problem uočen tokom ispitivanja je velika osetljivost na male promene u slici, što za posledicu ima velike izmene u primeni algoritma na sliku što se manifestuje izvesnim „trepenjem“ slike. Problem je posledica nedostatka algoritma zbog korišćenja samo globalnog histograma.

#### VI. ZAKLJUČAK

Napredni algoritam za poboljšanje kontrasta je uspešno implementiran na FPGA platformi za rad u realnom vremenu. Dalji razvoj ovog sistema bi se kretao u smeru implementacije dela algoritma za određivanje lokalnih histograma i obradu slike na osnovu istih, koji ovakvu obradu čini mnogo kvalitetnijom.

#### LITERATURA

- [1] Volker Wiczorek, „Entwurf eines Algorithmus zur lokalen Kontrastverbesserung von Videobildern“, Universität Ulm, Fakultät für Ingenieurwissenschaften, Mart 2006.
- [2] Vladimir Kovačević, „Logičko projektovanje računarskih sistema I – projektovanje digitalnih sistema“, Univerzitet u Novom Sadu, Fakultet Tehničkih Nauka, 2001.
- [3] Dr. Greg Tumbush „Signed Arithmetic in Verilog 2001 – Opportunities and Hazards“, Starkey Labs, Colorado Springs, CO.
- [4] Pro Design Electronic, „CHIPit Gold Edition – Handbook“, available: <http://www.prodesign-europe.com>

#### ABSTRACT

The paper describes digital design and the flow for implementation of advanced algorithm for contrast enhancement, based on global histogram and described in [1], working with real time image on CHIPit Gold platform based on FPGA.

#### IMPLEMENTATION OF ALGORITHM FOR CONTRAST ENHANCEMENT OF REAL TIME IMAGE BASED ON FPGA PLATFORM

Dragan Đukarić, Vladimir Marinković, Mihajlo Katona, Vladimir Zlokolica