

JAVASCRIPT SOFTVERSKO REŠENJE ZA OBRADU SLIKE

Andreja Samčović¹, Aleksandar Luković²

¹Univerzitet u Beogradu - Saobraćajni fakultet, andrej@sf.bg.ac.rs

²a.lukovic26@gmail.com

Rezime: *Obrada slike predstavlja sastavni deo savremenih multimedijalnih komunikacija. Ideja ovog rada je kreiranje novog softverskog rešenja za obradu slike, od početka, upotrebom programskog jezika JavaScript. U prvom delu softverskog rešenja je moguće odrediti diskretnu kosinusnu transformaciju (Discrete Cosine Transform, DCT) za nasumično generisanu matricu dimenzija 8x8 piksela. DCT koeficijenti su vizuelno predstavljani na grafiku. U drugom delu, korisnik može odabrati proizvoljnu sliku sa računara, dobiti informacije o njoj i početi sa obradom. Dostupni su razni efekti slike (crvena, zelena, plava) i specijalni efekti, kao što su: rotacija, horizontalno i vertikalno okretanje, isecanje, kompresija slike, poređenje rezultata pre i posle kompresije, generisanje URL (eng. Uniform Resource Locator) linka ulazne i komprimovane slike uz ostale mogućnosti. Softversko rešenje se može pokrenuti direktno iz veb pregledača ili instalirati na željeni operativni sistem.*

Ključne reči: *Canvas API, DCT, JavaScript, multimedija, obrada slike*

1. Uvod

Digitalna obrada slike je veoma aktuelna oblast istraživanja, imajući u vidu da se milijarde slika na dnevnom nivou postavljaju i preuzimaju sa različitih platformi, odnosno društvenih mreža, kao što su YouTube, Facebook, Twitter, Instagram, i dr. Platforme kao što su Instagram i Pinterest su orijentisane pre svega na razmenu slika. Osim navedenih platformi za postavljanje, preuzimanje i razmenu slika, kao i društvenih mreža, postoje i brojne druge primene u gotovo svim oblastima nauke i tehnologije gde je nezaobilazna obrada slike, kao što su medicinske slike, satelitske slike, industrijske primene, vojne primene i dr. [1], [2]. Razvoju digitalne obrade slike pomogli su računarski sistemi velike procesorske moći, kao i jedinice za memorisanje velikih kapaciteta.

Digitalna obrada slike je zastupljena na internetu, televiziji (TV), mobilnim uređajima, internetu stvari (*Internet of Things*, IoT), itd. Na osnovu redovnih statistika koje kompanija *Cisco* objavljuje, procenat zastupljenosti (primera radi, video saobraćaja) u odnosu na celokupni (video) saobraćaj u 2021. godini iznosi 81% [3]. Imajući to u vidu, iz ugla inženjerstva i telekomunikacija, potrebno je kreirati softverska rešenja za efikasnu obradu slike u cilju poboljšanja performansi mreža, brzine prenosa podataka i uštede podataka tamo gde je moguće, uz očuvanje zahtevanog kvaliteta medija.

Pod obradom slike podrazumevamo bilo koji vid obrade signala kod koga je ulaz slika, koja može biti fotografija, ili veštački generisana, npr. postupkom računarske animacije. Izlazni rezultat nakon primenjene obrade može biti takođe slika, ali i niz karakteristika ili parametara koji se odnose na sliku. Većina algoritama za obradu slike tretira sliku kao dvodimenzionalni (2D) signal i tom prilikom se primenjuju standardne tehnike za obradu signala [4]. Postoje brojne tehnike za obradu sirovih (*raw*) slika, kao što su akvizicija slika, prethodno procesiranje, segmentacija, izoštravanje, filtriranje, prepoznavanje, i sl. Poslednjih desetak godina kao jedan od moćnih softverskih alata za obradu slike, zahvaljujući svojim osobinama, se izdvojio JavaScript, naročito pri animaciji piksela pomoću *Canvas* interfejsa.

Struktura ovog rada je sledeća: nakon uvodnog dela, predstavljene su analitičke postavke za diskretnu kosinusnu transformaciju (*Discrete Cosine Transform*, DCT), objašnjena je primena JavaScript tehnologije u digitalnoj obradi slike, a zatim i samo softversko rešenje. Softversko rešenje u programskom jeziku JavaScript, implementirano je najpre na DCT, a zatim na još neke postupke digitalne obrade slike. Na kraju su izneta zaključna razmatranja.

2. Analitičke postavke za DCT

Tokom analize i obrade slike (uopšteno signala) dešavaće se da se neki elementi ponavljaju [5]. Prilikom obrade slike, veoma je korisno imati uvid u trenutnu brojnost njenih pojedinačnih elemenata i na taj način pristupiti spektralnoj analizi [6]. U tom cilju, DCT daje odlične rezultate. Na osnovu DCT se može sagledati frekvencija pojedinačnih elemenata slike. DCT je, osim toga, sastavni deo standarda za kompresiju mirmih slika JPEG (*Joint Photographic Expert Group*). Kod DCT slika se deli na skup diskretnih kosinusnih funkcija, kojih ima konačno mnogo [7], [8], [9]. U nastavku teksta sledi formula za DCT (1) i objašnjenje postupka proračuna određenih vrednosti funkcija.

$$F(u, v) = \frac{c(u) \cdot c(v)}{4} \cdot \sum_{j=0}^7 \sum_{k=0}^7 f(j, k) \cdot \cos\left(\frac{(2j+1) \cdot u \cdot \pi}{16}\right) \cdot \cos\left(\frac{(2k+1) \cdot v \cdot \pi}{16}\right) \quad (1)$$

U slučaju da promenljiva u ima vrednost nula, vrednost funkcije $c(u)$ biće $\frac{1}{\sqrt{2}}$ (2), isto važi i za funkciju $c(v)$ (4). Za vrednosti promenljivih u i v koje su u opsegu od 1 do 7 uključujući i te vrednosti, funkcije $c(u)$ i $c(v)$ iznosiće 1, saglasno sa formulama (3) i (5).

$$c(u) = \frac{1}{\sqrt{2}}, u = 0 \quad (2) \quad c(u) = 1, u = 1, \dots, 7 \quad (3)$$

$$c(v) = \frac{1}{\sqrt{2}}, v = 0 \quad (4) \quad c(v) = 1, v = 1, \dots, 7 \quad (5)$$

Prvi koeficijent, tj. koeficijent $F(0,0)$ imaće vrednost kao u formuli (6). Koeficijent $F(0,0)$ se naziva jednosmernom komponentom (eng. DC), srednjim osvetljenjem piksela slike u okviru DCT matrice (bloka) slike, srednjom vrednošću (intenzitetom) piksela slike u okviru bloka, itd. Ukoliko bi se svi elementi DCT matrice pojedinačno sabrali i taj zbir podelio brojem 8, dobila bi se vrednost koeficijenta $F(0,0)$. Upravo na to ukazuje formula (6). U programskoj implementaciji se može desiti određeno odstupanje od tačne vrednosti, a jedan od mogućih uzroka je broj decimala zaokruživanja.

$$F(0,0) = \frac{1}{8} \cdot \sum_{j=0}^7 \sum_{k=0}^7 f(j, k) \quad (6)$$

Inverzna diskretna kosinusna transformacija (*Inverse Discrete Cosine Transform*, IDCT) je određena formulom (7):

$$f(j, k) = \sum_{u=0}^7 \sum_{v=0}^7 \frac{c(u) \cdot c(v)}{4} \cdot F(u, v) \cdot \cos\left(\frac{(2j+1) \cdot u \cdot \pi}{16}\right) \cdot \cos\left(\frac{(2k+1) \cdot v \cdot \pi}{16}\right) \quad (7)$$

3. Primena JavaScript tehnologija kod obrade slike

Velika popularnost i uvođenje novih funkcija doprinele su proširenju programskog jezika JavaScript. Tako su se stvorile mogućnosti programiranja aplikacija za razne operativne sisteme, mobilne uređaje, IoT, veb, dronove, vizuelizaciju podataka, i ostale. Iz oblasti multimedije i obrade signala, uz pomoć programskog jezika JavaScript, ostavljen je prostor za istraživanja i kreiranje novih rešenja. U ovom radu, prvi korak ka predstavljanju multimedije, uz pomoć JavaScript tehnologije biće upravo pomoću *Canvas* interfejsa za programiranje aplikacija (*Application Programming Interface*, API), tj. *Canvas* API-ja. *Canvas* API se koristi u radu sa 2D grafikom i zasnovan je na JavaScriptu [10].

Kreirano softversko rešenje je u vidu grafičkog korisničkog interfejsa (*Graphical User Interface*, GUI), gde korisnik, uopšteno, na osnovu ulaznih podataka dobija izlazne podatke. Softversko rešenje se trenutno nalazi na platformi *GitHub*, koja je otvorenog koda (eng. *open-source*) [11]. U kreiranom programu (u nekim delovima narednog teksta, samo „program *obrada_slike.js*“) se može pristupiti putem veb pregledača (na računaru, mobilnom telefonu i/ili tabletu), a u izradi je i verzija posebno namenjena računarima (desktop uređajima) za operativne sisteme *Windows*, *Linux* i *Mac OS*, uz pomoć JavaScript platforme *Electron.js*, koja je danas vrlo popularna.

Program *obrada_slike.js* se može svrstati u čeonu veb aplikaciju (eng. *front-end*) jer poseduje GUI interfejs. Ukoliko bi se određeni procesi izvršavali u pozadini, uz komunikaciju sa veb serverom, bazama podataka, API-jima, POST/GET zahtevima, pomenuto rešenje bi moglo da se uključi u pozadinske veb aplikacije (eng. *back-end*) [12]. Jedna od zanimljivih ideja za dalja istraživanja bi bila izrada pozadinske veb aplikacije (i/ili kombinacija sa čeonom aplikacijom) za obradu slike.

4. JavaScript softversko rešenje za DCT test

Korisnički interfejs softverskog rešenja je prikazan u nastavku rada. Nakon prvog pokretanja, uočava se početni interfejs, bliže određen Slikom 1. Dostupne opcije su: odabir jezika (srpskog ili engleskog), DCT test i unos slike za obradu. Program *obrada_slike.js* je kreiran od početka, bez korišćenja radnih okvira. Uz veb tehnologije HTML (eng. *HyperText Markup Language*) i CSS (eng. *Cascading Style Sheet*) razvijene su struktura i prezentacija programa, dok su se JavaScript jezikom definisale aktivnosti, dinamičnost, ponašanje i izvršavale naredbe [13].



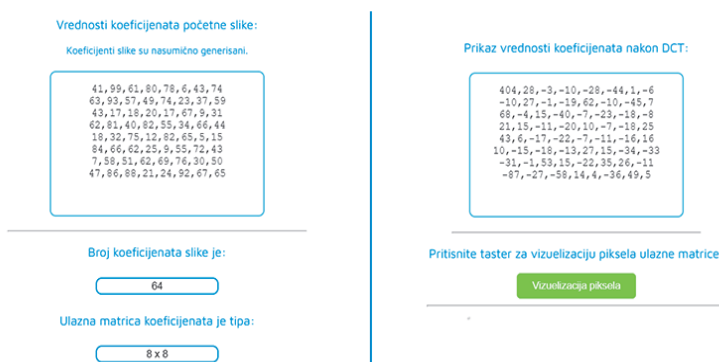
Slika 1 Početni korisnički interfejs softverskog rešenja

Nakon odabira opcije za DCT test, pred korisnikom je mogućnost nasumičnog generisanja matrice piksela dimenzija 8×8 , u skladu sa Slikom 2. Ulazna matrica predstavlja najmanji element slike (blok), koji je osnova za dalju obradu.



Slika 2 Nasumično generisanje ulazne matrice piksela

Matrica dimenzija 8×8 piksela je sada generisana i njeni koeficijenti su nasumični, a oni su vidljivi na Slici 3a. Softversko rešenje prikazuje osnovne informacije o matrici, broj koeficijenata (u ovom slučaju $8 \times 8 = 64$), kao i njenu strukturu (8×8).



Slika 3 (a) Generisanje ulazne matrice (b) DCT transformacija ulazne matrice

Za novonastalu matricu formiraju se odgovarajući DCT ekvivalenti predstavljeni na Slici 3b, a klikom na taster „Vizualizacija piksela“, nasumično generisani pikseli ulazne matrice se prenose na kanvas i na taj način nastaje mala slika, radi lakšeg zapažanja zaokružena plavim krugom na Slici 4.

Pritisnite taster za vizuelizaciju piksela ulazne matrice:

Vizuelizacija piksela



Slika 4 Uvećani prikaz nasumično generisanih piksela

U JavaScriptu, svaki piksel se sastoji od četiri komponente: crvene (*red*, R) zelene (*green*, G), plave (*blue*, B) i komponente za providnost (*alpha*, A). One čine osnovu za RGBA model. Rezultat je taj da će se od 64 nasumično generisanih piksela, obrazovati slika od $64/4 = 16$ piksela. Posmatrajući ulaznu matricu sa Slike 3a, koja se sastoji od sledećih koeficijenata (vrednosti piksela):

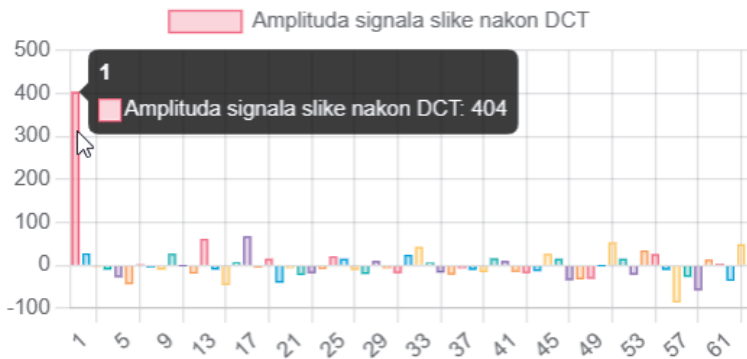
```
[ 41, 99, 61, 80, 78, 6, 43, 74
  63, 93, 57, 49, 74, 23, 37, 59
  43, 17, 18, 20, 17, 67, 9, 31
  62, 81, 40, 82, 55, 34, 66, 44
  18, 32, 75, 12, 82, 65, 5, 15
  84, 66, 62, 25, 9, 55, 72, 43
  7, 58, 51, 62, 69, 76, 30, 50
  47, 86, 88, 21, 24, 92, 67, 65 ]
```

formirali bi se sledeći pikseli:

1. piksel: RGBA(41, 99, 61, 80); 2. piksel: RGBA(78, 6, 43, 74);
3. piksel: RGBA(63, 93, 57, 49); 4. piksel: RGBA(74, 23, 37, 59);
5. piksel: RGBA(43, 17, 18, 20); 6. piksel: RGBA(17, 67, 9, 31);
7. piksel: RGBA(62, 81, 40, 82); 8. piksel: RGBA(55, 34, 66, 44);
9. piksel: RGBA(18, 32, 75, 12); 10. piksel: RGBA(82, 65, 5, 15);
11. piksel: RGBA(84, 66, 62, 25); 12. piksel: RGBA(9, 55, 72, 43);
13. piksel: RGBA(7, 58, 51, 62); 14. piksel: RGBA(69, 76, 30, 50);
15. piksel: RGBA(47, 86, 88, 21); 16. piksel: RGBA(24, 92, 67, 65)

Upravo se tako stvara slika, piksel po piksel, a oni su dalje sačinjeni od niza (četiri) broja. Na osnovu teorijskog aspekta multimedije [4], [5], blok slike čini matrica koeficijenata 8×8 piksela, kojih ima 64. Međutim kod JavaScripta, od matrice sa 64 koeficijenata nastaje 16 piksela slike, prvenstveno zbog osobine da su za predstavljanje jednog piksela potrebna četiri broja (podatka) tj. RGBA vrednosti. Formula (1) tj. formula za direktnu DCT se primenjuje na svaku pojedinačnu vrednost RGBA komponente, koja je u stvari funkcija $f(j, k)$ iz pomenute formule.

Vrednost svake pojedinačne RGBA komponente se kreće u opsegu od 0 do 255. Iz gore navedenog primera tj. nasumično generisanih piksela, može se zaključiti da vrednost A komponente (providnost) ne prelazi numeričku vrednost od 82. Slika će tako izgledati kao da je primenjen efekat sive boje (eng. *gray scale*). Slika 5 sadrži vrednosti koeficijenata matrice nakon procesa DCT. Prva komponenta ima najvišu vrednost, što je u skladu sa teorijom i predstavlja srednju vrednost piksela u datom bloku slike.



Slika 5 Vizuelni prikaz vrednosti DCT koeficijenata

Zaključak ovog DCT testa je da u JavaScript implementaciji, generisanjem 64 koeficijenta nastaje slika od 16 piksela. Za obrazovanje celokupnog bloka slike od 64 piksela biće potrebno $64 \times 4 = 256$ koeficijenata ulazne matrice.

5. JavaScript softversko rešenje za neke algoritme obrade slike

Za početak, korisnik odabira proizvoljnu sliku sa računara i pritiskom na taster „Podaci” počinje obrada. Podaci koji su od značaja za obradu slike su njena rezolucija u megapiksela (*megapixels*, MP), broj piksela, format, memorijsko zauzeće, itd. Veoma korisna opcija JavaScripta je otkrivanje URL linka slike (veb adrese). Slike 6a i 6b daju uvid u proces odabira slike i dobijanja korisnih informacija.

Na osnovu ovog podatka može se zaključiti o kom tipu slike je reč, jer svaka slika poseduje svoj jedinstveni *base64* kôd. Selektovanjem celog URL linka slike (koristeći kombinaciju tastera *Ctrl+A*) i unosom u veb pregledač, pojavljuje se nova stranica, koja prikazuje sliku u svom neizmenjenom, originalnom i izvornom obliku.



(a)



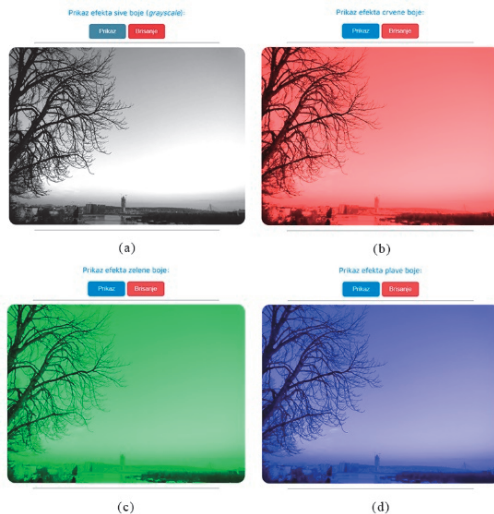
(b)

Slika 6 (a) odabir slike (b) dobijanje korisnih podataka o slici

Korisnik ima mogućnost izbora različitih efekata slike. Slika 7 ilustruje različite tipove efekata. Prvi slučaj je primena efekta sive slike. Ovaj efekat se dobija tako što se

vrednosti pojedinačnih piksela ulazne slike podele sa brojem 2 ili pomnože sa $1/2$, saberu i dodele novonastaloj slici.

U ostalim primerima, prvi korak je određivanje aritmetičke sredine vrednosti piksela ulazne slike. Zatim se tako dobijena vrednost dodeljuje svakom pojedinačnom pikselu nove slike, pri čemu će jedna od RGB komponente piksela imati višu vrednost u odnosu na ostale dve. Ukoliko R komponenta ima višu vrednost (u primeru koda je vrednost pet puta viša) u odnosu na G i B komponentu, primeniće se efekat crvene boje. Isto pravilo važi i za ostala dva efekta (zelene i plave boje). Softverskim rešenjem je moguće postići (ta opcija takođe postoji u programu *obrada_slke.js*) i specijalni efekat, gde bi svaki četvrti piksel imao istu vrednost uz slične proračune kao u prethodnim slučajevima.

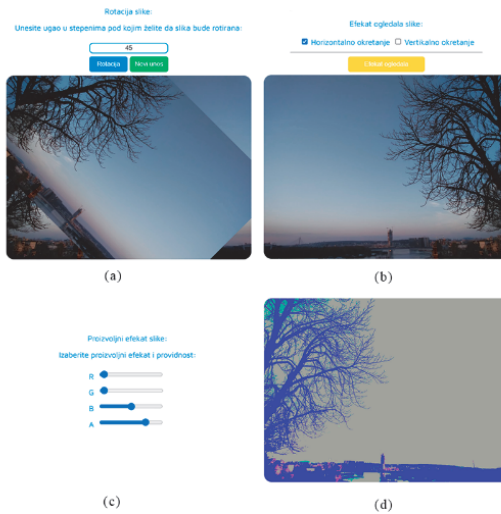


Slika 7 (a) efekat gray scale (b) efekat crvene boje (c) efekat zelene boje (d) efekat plave boje

Unosom određenog ugla slika se rotira, a vizuelni prikaz je dostupan na Slici 8a. Na raspolaganju je i efekat ogledala, tj. horizontalno, vertikalno ili istovremeno oba načina rotacije slike, koji jasnije približava Sliku 8b. Pored već postojećih opcija za odabir efekta sive, crvene, zelene, plave boje, specijalnog efekta, pred korisnikom je mogućnost proizvoljnog definisanja kombinacija boja i providnosti. Prikaz postojećeg menija, koji se sastoji od tri klizača (eng. *sliders*) za svaku od boja (RGB) uz providnost (A) se može zapaziti na Slici 8c. Tako se bez poteškoća može kreirati efekat slike „crtanog filma“ (eng. *cartoon effect*), koji je uočljiv na Slici 8d.

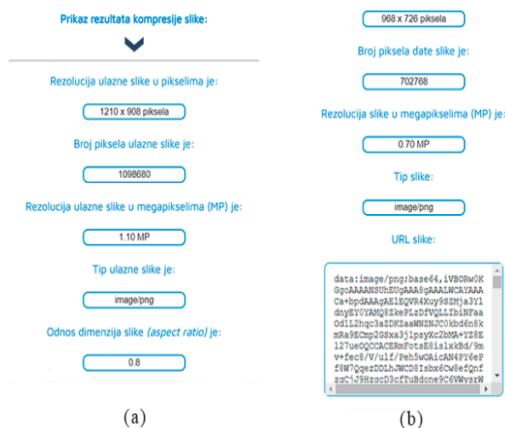
Sledeći korak je kompresija slike. Ona se obavlja tako što se prvo izračunaju vrednosti širine (*width*) i visine ulazne slike (*height*). Njihov odnos tj. količnik (eng. *aspect ratio*) predstavlja koeficijent proporcionalnosti, kojim se dimenzije ulazne slike (širina i visina) pojedinačno množe u cilju dobijanja izlazne, komprimovane slike.

Uz pretpostavku da visina slike ima dva puta manju vrednost u odnosu na širinu, njihov odnos biće 0,5. Množenjem širine i visine slike vrednošću $1/2$, njene dimenzije će se praktično smanjiti (prepoloviti) i tako će se dobiti slika rezolucije koja je četiri puta manja u odnosu na rezoluciju ulazne slike.



Slika 8 (a) rotacija slike (b) efekat ogledala slike (c) proizvoljni efekat slike (d) rezultat proizvoljno izabranog efekta slike

U primeru rada, ulazna slika ima dimenzije tj. rezoluciju 1210x908 piksela. Odnos širine i visine ulazne slike je $908/1210 = 0,8$, a njena rezolucija u megapikselima (MP) je 1,1 MP. Slika 9a se odnosi na prikaz i organizaciju ovih podataka, na način koji programsko rešenje koristi. Kada se širina i visina slike pomnože koeficijentom proporcionalnosti (0,8), dobiće se sledeće vrednosti piksela: $1210 \times 0,8 = 968$ (za širinu) i $908 \times 0,8 = 726$ (za visinu). Nova rezolucija slike je sada 968x726 piksela, tj. 0,7 MP, a njen format je ostao isti (*Portable Network Graphics*, PNG). Rezultati su vidljivi i na Slici 9b.



Slika 9 (a) podaci o ulaznoj slici i odnos širine i visine slike (aspect ratio) (b) prikaz podataka komprimovane slike

6. Zaključak

Slika, kao osnovni element multimedije, posmatran iz ugla telekomunikacija označava signal, koji u sebi nosi veliku količinu podataka. Podaci se dalje mogu organizovati, predstavljati, proračunavati i obrađivati. Tako se javljaju mogućnosti analize u frekvencijskom i prostornom domenu. Sa porastom rezolucije, slika sadrži više piksela, time i više podataka, a sama obrada postaje zahtevnija. Budući izazovi u kreiranju softverskih rešenja za obradu slike u JavaScriptu biće okrenuti ka pronalazenju i definisanju preglednijeg koda, efikasnijih algoritama, kao i novih funkcija. Te funkcije bi mogle da se izvršavaju u pozadini, istovremeno u toku rada procesa, bez usporavanja i smanjenja trenutnih performansi, a sve to uz postizanje brže obrade.

Ostaje i mogućnost raspodele zadataka čeonim i pozadinskim veb aplikacijama, kako bi performanse obrade bile još bolje. Postupci obrade slike mogu da budu primenjeni i na obradu drugih signala, njihovu frekvencijsku analizu, kao i vizuelizaciju u spektralnom i prostornom domenu.

Zahvalnica. Rad je delimično finansiran od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije.

Literatura

- [1] R. S. Hegadi, Image Processing: Research Opportunities and Challenges, *National Seminar on Research in Computers*, Bharathiar University, Coimbatore, India, December 2010.
- [2] B. H. Kang, A Review on Image & Video Processing, *International Journal of Multimedia and Ubiquitous Engineering*, Vol. 2, No. 2, April 2007.
- [3] <https://www.cisco.com/c/en/us/about/annual-reports.html> (Pristup: 21.10.2021.)
- [4] M. Popović, *Digitalna obrada slike*, Akademska misao, Beograd, 2006.
- [5] A. Samčović, *Multimedijalne komunikacije*, Univerzitet u Beogradu - Saobraćajni fakultet, 2015.
- [6] I. T. Young, J. J. Gerbrands, L. J. van Vliet, *Fundamentals of Image Processing*, Delft University of Technology, v2.3, 2007.
- [7] E. Feig, S. Winograd, Fast Algorithms for the Discrete Cosine Transform, *IEEE Transaction on Signal Processing*, Vol. 40, No. 9, September 1992.
- [8] J. John, *Discrete Cosine Transform in JPEG Compression*, School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology, India, February 2021.
- [9] J. Zhou, On discrete cosine transform, *Nigerian prevod sedmog izdanja Journal of Technological Research*, September 2011.
- [10] D. Flanagan, *JavaScript: sveobuhvatan vodič*, Mikro knjiga, Beograd, 2021.
- [11] JavaScript softversko rešenje za obradu slike, dostupno na: <https://github.com/alukovic>
- [12] I. Kinght, *Arduino i JavaScript za povezivanje na veb: Front End programiranje Pomoću JavaScripta*, Kompjuter biblioteka, Beograd, 2018.
- [13] L. Lemay, R. Colburn, J. Kyrnin, *HTML5, CSS3 i JavaScript za razvoj Veb strana*, sedmo izdanje, Kompjuter biblioteka, Beograd, 2016.

Abstract: *Image processing is an integral part of modern multimedia communications. The idea of this paper is to create a new software solution for image processing, from the beginning, using the JavaScript programming language. In the first part of the software solution, the Discrete Cosine Transform (DCT) for a randomly generated matrix with 8x8 pixels is determined. DCT coefficients are visually represented on the graph. In the second part, the user can select an arbitrary image from the computer, get corresponding information and start processing. Various image effects (red, green, blue) and special effects are available, such as: rotation, horizontal and vertical rotation, cropping, image compression, comparison of results before and after compression, URL generation links for input and compressed images, with other options as well. The software solution can be run directly from a web browser or installed on the desired operating system.*

Keywords: *Canvas API, DCT, JavaScript, image processing, multimedia*

JAVASCRIPT SOFTWARE SOLUTION FOR IMAGE PROCESSING

Andreja Samčović, Aleksandar Luković