ŠIAULIŲ UNIVERSITETAS

Technologijos, fizinių ir biomedicinos mokslų fakultetas

Kompiuterių sistemų katedra

Eugenijus Margalikas

Skaitmeninių nuotraukų steganografijos algoritmas ir jo tyrimas

Magistro baigiamasis darbas

Vadovė dr. Simona Ramanauskaitė

Šiauliai, 2016

ŠIAULIŲ UNIVERSITETAS

Technologijos, fizinių ir biomedicinos mokslų fakultetas

Kompiuterių sistemų katedra

TVIRTINU

Kompiuterių sistemų katedros vedėjas
doc. dr. E. Paliulis
2016-06-01

Skaitmeninių nuotraukų steganografijos algoritmas ir jo tyrimas

Informatikos inžinerijos magistro baigiamasis darbas

Vadovas
Kompiuterių sistemų katedros docentė dr. S. Ramanauskaitė
2016 m. gegužės ____ d.

Recenzentas
Kompiuterių sistemų katedros docentas dr. Egidijus Paliulis
2016 m. gegužės ____ d.

Recenzentas
Kompiuterių sistemų katedros lektorė dr. Asta Drukteinienė
2016 m. gegužės ____ d.

Autorius
ITM-14 gr. studentas E. Margalikas
2016 m. gegužės 26 d.

Šiauliai, 2016

SANTRAUKA

Skaitmeninių nuotraukų steganografijos algoritmas ir jo tyrimas

Darbe apžvelgiami skaitmeninės steganografijos metodai, jų klasifikaciją ir pasiūlomas originalus steganografijos metodas, kuris slepia duomenis skaitmeninių vaizdų paletėse, modifikuojant tam tikrų spalvų visus vaizdo pikselius. Pasitelkiant vizualius stegoanalizės metodus, atliktas tyrimas, kokius iškraipymus įneša į skirtingos raiškos ir spalvų gylio vaizdus ir jų histogramas pasiūlytas steganografijos algoritmas, rezultatai palyginami su LSB steganografijos algoritmu.

Raktiniai žodžiai: steganografija, skaitmeninių vaizdų apdorojimas, duomenų apsauga.

SUMMARY

Algorithm of the Digital Photos Steganography and its Research

This thesis is based on reviewing and classification of popular methods used in digital steganography and proposes an original method of steganography which hiding data in color palettes of digital images. There was made a research applying visual mothods of steganalysis on what kind of distiortions adds the original method to digital images with different color depth and resolution and how they their histograms. Results compared with LSB method of steganography.

Keywords: steganography, digital image processing, information security.

PAVEIKSLĖLIŲ SĄRAŠAS

LENTELIŲ SĄRAŠAS

TURINYS

ĮVADAS

Su svarbios informacijos perdavimo saugumo problema žmonija susiduria nuo seniausiu laikų. Šiai problemai spręsti susiformavo dvi mokslo šakos: steganografija ir kriptografija. Kai kriptografija taikoma informacijos šifravimui, steganografija paslepia patį perduodamos informacijos egzistavimo faktą. Prasidėjus skaitmeninei erai atsirado ir skaitmeninė steganografija, kuriai plačias galimybes atvėrė skaitmeniniu būdu pateikiami realaus pasaulio objektai. Vienas tokių objektų yra skaitmeniniai vaizdai, kurie dėl skaitmeninimo būdo bei žmogaus regos ypatumų gali turėti dideles paklaidas bei daug perteklinės informacijos, kas leidžia naudoti juos kaip konteinerius informacijai – stegopranešimui slėpti.

Šiuo metu egzistuoja keletas pagrindinių skaitmeninės steganografijos metodų slaptų stegopranešimų įterpimui į skaitmeninius vaizdus. Kadangi žinomų steganografijos metodų veikimas gali būti gerai ištyrinėtas, padidėja tikimybė, kad jų taikymas bus aptiktas. Tokiu atveju norint padidinti slaptumą įgauna prasmę naujų dar nežinomų metodų sukūrimas ir taikymas, o dėl to, kad steganografijos mokslas yra dar pakankamai jaunas, galimybių tam yra.

Šiuo darbo tikslas – pasiūlyti steganografijos skaitmeninėse nuotraukose algoritmą ir ištirti jo taikymo savybes.

Šiam tikslui pasiekti yra iškelti šie uždaviniai:

1. Atlikti egzistuojančių steganografijos skaitmeninėse nuotraukose metodų apžvalgą.
2. Pasiūlyti savo skaitmeninių nuotraukų steganografijos algoritmą.
3. Programiškai realizuoti pasiūlytą algoritmą jo taikymo savybių testavimui.
4. Ištirti pasiūlyto steganografijos algoritmo taikymo savybes su skirtingos raiškos ir su skirtingu naudojamų spalvų kiekiu nuotraukomis.
5. Apibendrinti atlikto tyrimo rezultatus, juos lyginant su kitais egzistuojančiais skaitmeninių nuotraukų steganografijos algoritmais.

Darbo rezultatai leis pasirinkti efektyvesnį steganografijos algoritmą, priklausomai nuo skaitmeninio vaizdo savybių, įterpiamo duomenų kiekio ir reikalavimų modifikuoto vaizdo kokybei.

1. STEGANOGRAFIJA IR JOS PANAUDOJIMAS SKAITMENINIUOSE VAIZDUOSE

1. 1. Steganografija ir jos panaudojimas

Steganografija – tai mokslas bei menas perduoti slaptą pranešimą paslepiant jį dengiančiame objekte [21]. Šiuolaikinėje literatūroje steganografija aprašoma kaip kalinių uždavinys, kuriame du kaliniai – Alicija ir Bobas – nori suregzti pabėgimo planą, bet jų bendravimas yra stebimas prižiūrėtojo Ievos, kuri supratus, kad kaliniai perduoda vienas kitam paslėptą nuo jos informaciją, sugriaus jų planus ir perkels juos į griežtesnio režimo kalėjimą. Alicijos ir Bobo uždavinys yra sukurti tokį informacijos perdavimo būdą, kad Ieva nieko neįtartų [9].

Pirmieji įrašai apie steganografijos būdu perduodamą informaciją buvo padaryti senovės graikų, o terminą "steganografija" apie 1499 m. savo knygoje "Steganographia" įvedė Iohannes Trithemius [17]. Klasikinės steganografijos pavyzdžiai yra:

- pranešimų slėpimas ant vaškuotų medinių lentelių, kurios buvo naudojamos rašymui, užrašant pranešimą ant pačios lentelės ir padengiant ją vašku;
- permatomo rašalo naudojimas rašant tarp kito teksto eilučių arba laisvoje lapo vietoje;
- spausdinamo teksto simbolių parametrų, tokių kaip poslinkis, dydis, pakreipimas naudojimas, kurie iš pirmo žvilgsnio gali atrodyti kaip spausdinimo defektai;
- trafaretai, kuriais uždengus tekstą lieka matoma tik dalis simbolių, sudarančių slaptą pranešimą;
- mikrotaškai – vaizdai arba tekstas sumažinti tiek, kad būtų neįžiūrimi be specialių prietaisų, pagaminti fotografijos principu.

Steganografijos šaka, kuri remiasi kompiuterinių sistemų ypatumais – kompiuterinė steganografija [19]. Sprendimai, taikomi kompiuterinėse sistemose, dažnai yra kompromisas tarp universalumo, naudojimo patogumo, našumo ir kt., todėl neretai atsiranda situacijų, kur kompiuterinės sistemos resursai nepilnai išnaudojami. Kompiuterinės steganografijos pavyzdžiai:

- kai failų sistema adresuoja kaupiklio vietą blokais, įmanoma paslėpti informaciją dalinai užpildytuose blokuose;
- informacijos slėpimas failų formatų rezervuotose, bet nenaudojamose vietose

Skaitmeninėje steganografijoje stegokonteinerio vaidmenį atlieka daugialypės terpės objektai – tokie kaip vaizdai. Stegopranešimas įterpiamas į objektą nežymiai jį modifikuojant [8]. Vaizdai gali būti sukurti kompiuteryje arba realus, gauti naudojant jutiklius(skeneris, fotoaparatas ir kt). Steganografijos atžvilgiu, labiau susidomėjimą kelia būtent realus vaizdai, nes dėl jutiklių fizikinių savybių vaizduose atsiranda netobulumai bei triukšmas, kuriuos galima panaudoti steganografijos tikslams [1]:

- **Skaitmeniniai atspaudai.** Skaitmeniniai atspaudai gali būti naudojami išskirtinių teisių apsaugai – legaliai platinant konteinerį, į kiekvieną jo kopiją steganografijos būdu įterpiamas skirtingas pranešimas-žymuo. Jeigu konteineris pradedamas platinti nelegaliai, tai galima atsekti jo šaltinį.
- **Skaitmeniniai vandens ženklai.** Skaitmeniniai vandens ženklai gali būti naudojami autorinių teisių apsaugai, kai steganografijos būdu į platinamą konteinerį įterpiamas vienodas pranešimas. Taip, esant reikalui, galima nustatyti konteinerio autorių arba patvirtinti konteinerio tikrumą.

- **Slaptas informacijos perdavimas.** Slaptas informacijos perdavimas tai klasikinis steganografijos tikslas. Čia uždavinys yra perduoti pranešimą nesukėlus įtarimo, kad pranešimas egzistuoja.

Kiekvienam atskiram tikslui pasiekti, steganografijos sistemos vadovaujasi skirtingais kriterijais, nes steganografijos taikymo tikslams prasme skiriasi. Slapto informacijos perdavimo prasmė yra būtent slaptas informacijos perdavimas, o skaitmeniniai atspaudai ir skaitmeniniai vandens ženklai yra skirti konteinerio apsaugai ir be to jų egzistavimo faktas nebūtinai yra slaptas. Dėl to kai kurie autoriai priskiria steganografijai tik slaptą informacijos perdavimą.

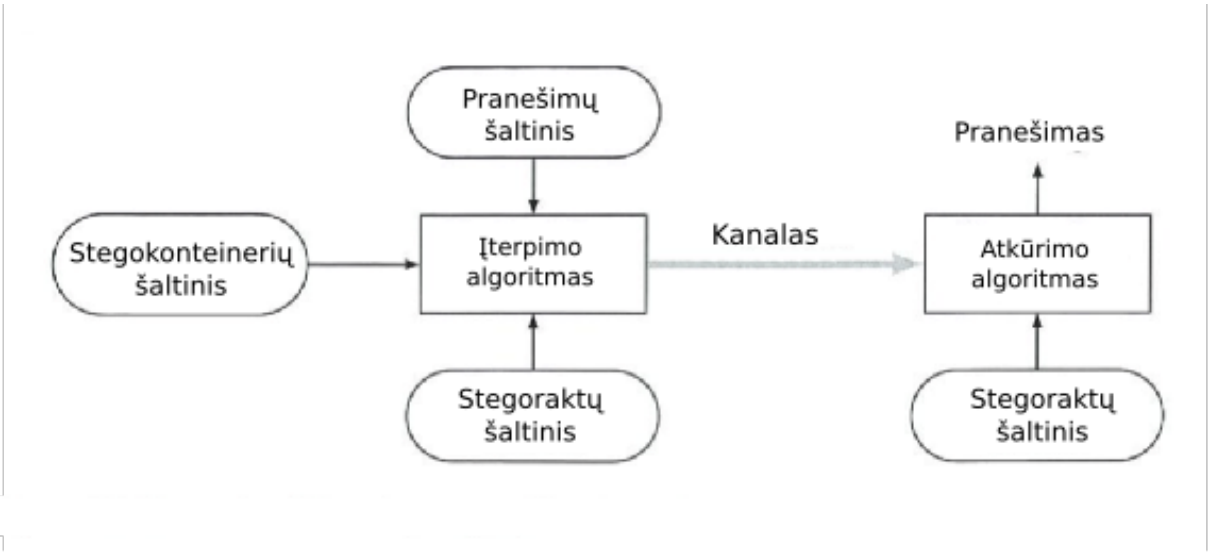
1. 2. Stegosistemos

Pagrindinis steganografijos tikslas – slaptai bendrauti, nesukeliant įtarimo, kad bendraujama slaptai [8]. Tai galimą pasiekti slepiant pranešimus įprastai atrodančiuose objektuose, siunčiamuose atvirai komunikacijos kanalais.

Prieš pradedant bendrauti, Alicija ir Bobas turi sutarti dėl kažkokio bazinio bendravimo protokolo, kuriuo jie remsis ateityje. Taip jiems reikia pasirinkti dengiančių objektų – stegokonteinerių tipą, toliau jiems reikia sukurti pranešimo slėpimo ir atkūrimo algoritmus, saugumui padidinti algoritmai turi naudoti slaptą raktą. Prižiūrėtojos Ievos gebėjimas aptikti slaptą bendravimą gali priklausyti nuo siunčiamų pranešimų dydžio, be to kontroliuojant komunikacijos kanalą Ieva gali įsiterpti į bendravimą [9].

Taip kiekvienoje steganografijos sistemoje (*žr. 1.1 pav.*) galima išskirti penkis bazinius elementus:

- Stegokonteinerių šaltinis
- Duomenų įterpimo ir atkūrimo algoritmai
- Raktų šaltinis
- Pranešimų šaltinis
- Duomenų apsikeitimo kanalas



1.1 pav. *Pagrindiniai steganografijos kanalo elementai*

Stegokonteinerio šaltinio atributai skaitmeninių vaizdų atveju būtų tokie kaip formatas, dydis, turinio tipas ir t.t. Bendru atveju stegokonteinerio savybes apibrėžia objektai, kuriais keistųsi Alicija ir Bobas bendraujant atvirai.

Duomenų įterpimo algoritmas tai procedūra, kuri apibrėžia skaitmeninį vaizdą, kuriame perduodamas slaptas pranešimas. Ši procedūra gali naudoti stegoraktą, kuris reikalingas ir teisingam pranešimo atkūrimui [18].

Raktas pasirenkamas kaip atsitiktinė reikšmė iš visų galimų raktų intervalo.

Perduodami pranešimai taip pat labai įtakoja stegosistemą. Pranešimo ilgiui artėjant prie įterpimo algoritmo galimybių ribos, atitinkamai padidėja ir tikimybė aptikti slaptą bendravimą.

Duomenų apsikeitimo kanalas, kuriuo siunčiami skaitmeniniai vaizdai, tariamai stebimas prižiūrėtojos Ievos, kuri gali atlikinėti tris skirtingus vaidmenis. Pasyvus stebėtojas – stebi srautą ir neįtakoją patį bendravimą. Aktyvus prižiūrėtojas - Ievai įtarus, kad Alicija ir Bobas gali naudoti steganografiją, gali bandyti preventyviai sužlugdyti steganografijos sistemą, tyčia iškraipant skaitmeninius vaizdus, kuriais apsikeičia Alicija ir Bobas. Pvz. ji gali suglaudinti vaizdus naudojant JPEG formatą, pakeisti jo dydį, apkarpyti arba kitaip juos apdoroti [6]. Steganografijos sistema bus sužlugdyta, jeigu naudojamas steganografijos metodas nėra atsparus tokiam apdorojimui. Piktavalis prižiūrėtojas - kai aktyvus prižiūrėtojas stengiasi padaryti slaptą bendravimą negalimu, piktavalis prižiūrėtojas nebūtinai sužlugdo steganografijos sistemą, bet nustačius naudojamą steganografijos metodą, gali panaudoti ją savo tikslams – įsiterpti į bendravimą ir klaidinti slapto bendravimo dalyvius.

Svarbiausia stegosistemos dalis tai įterpimo ir atkūrimo algoritmas, kuris remiasi vienu iš trijų bazinių mechanizmu. Konteinerio parinkimas – kai stegokonteineris parenkamas taip, kad jis jau turėtų reikiamą slaptą informaciją. Konteinerio sintezė – stegokonteineris sugeneruojamas taip, kad perteiktų perduodamą pranešimą. Konteinerio modifikavimas – modifikuojamas įterpiant slaptą informacija. Šis būdas leidžia perduoti daugiausiai slaptos informacijos.

1. 3. Konteinerio parinkimas

Šių algoritmų grupė atsitiktinai perrenka skaitmeninius vaizdus iš apibrėžtos vaizdų bazės tol, kol surandamas vaizdas atitinkantis norimą perduoti pranešimą. Stegoraktas šiuo atveju - tai taisyklių rinkinys, kuris apibrėžia kaip reikia interpretuoti vaizdą. Pvz pavieniai bitai gali būti koduojami horizontalia arba vertikalia vaizdo orientacija, tam tikrų objektų vaizde buvimu arba nebuvimu ir tt. Šio būdo privalumas yra tai, kad konteineris yra visiškai natūralus niekaip nemodifikuotas vaizdas. Trūkumas - mažas perduodamų duomenų kiekis.

1. 4. Konteinerio sintezė

Algoritmų grupė, kurie sugeneruoja konteinerį sutalpindami į jį slaptą pranešimą. Priklausomai nuo to, kur bus naudojamas toks konteineris taikomi skirtingi jo sintezės metodai. Jeigu tariamai konteinerį tikrins automatizuotos steganografijos aptikimo sistemos, tai konteineris sugeneruojamas taip, kad jo savybės atitiktų natūralaus konteinerio savybes, tokias kaip greta esančių pikselių tarpusavio priklausomybė. Taip sugeneruotas skaitmeninis vaizdas nebūtinai turi atrodyti kaip tikras vaizdas.

1. 5. Konteinerio modifikavimas

Algoritmai, kurie įterpia slaptą pranešimą į konteinerį jį modifikuojant. Tai gali būti algoritmai naudojantys formatinius metodus, kurie remiasi vaizdų saugojimo formatų ypatumais, ir

algoritmai naudojantys neformatinius metodus, kurie daro pakeitimus pačio vaizdo duomenyse juos kažkiek iškraipant.

1. 6. Formatiniai metodai

Metodai, kurie remiasi skaitmeninių vaizdų saugojimo formatų ypatumais. Į skaitmeninių vaizdų formatus integruota daugiau galimybių nei paprastai naudojama, todėl juose egzistuoja tarnybiniai, rezervuoti arba nepilnai išnaudojami laukai, į kuriuos galima įrašyti slaptus duomenis nekenkiant vaizdo kokybei.

1. 6. 1. BMP formato metodai

Dauguma standartinių darbo su formatu priemonių failo pabaigą nustato iš jo antraštės, todėl yra galimybė įrašyti slaptus duomenis į failo pabaigą.

Kadangi formatas numato paletės naudojimą, kurios dydis gali būti skirtingas, tai vaizdo duomenų pradžia yra nurodoma antraštėje poslinkio lauke. Jo reikšmę gali būti pakeista net ir kai paletė nenaudojama, taip tarp antraštės arba paletės atsiranda vietos slaptiems duomenims.

Jeigu BMP faile saugomas vaizdas su 16 bitų spalvų gyliu, galima išnaudoti tokį ypatumą, kad vienam RGB kanalui skiriami 5 bitai, o vyriausias bitas neneša jokios informacijos ir gali būti naudojamas slaptiems duomenims.

Taip pat neišnaudojamų baitų yra spalvų paletėj, jeigu tokia yra naudojama. BMP formato paletės elementai yra užduodami 4 baitais, pirmi 3 iš kurių koduoja spalvą, o paskutinis dažniausiai lygus 0 ir nenaudojamas. tokiu būdu galima paslėpti iki 256 baitų informacijos.

BMP formate horizontalios vaizdo eilutės kodavimui naudojamas baitų skaičius turi būti kartotinis 4-iems, Jeigu vaizdas netenkina šios sąlygos, tai eilutė papildoma nuliniais baitais iki reikiamo ilgio. Tuose baituose irgi galima patalpinti slaptą pranešimą.

1. 6. 2. JPEG formato metodai

Vienas paprasčiausių būdų paslėpti duomenis, tai įrašyti juos į failo pabaigą. Tai įmanoma todėl, kad JPEG formatas naudoją žymių sistemą, ir visos standartinės darbo su JPEG formatu priemonės skaito duomenis iki tol, kol aptinka vaizdo pabaigos žymę [4].

JPEG numato komentarų žymes, kurias naudoja grafikos apdorojimo programos ir įvairūs įrenginiai, pavadinimams, techninėms charakteristikoms, darbo aplinkos parametrams ir kitiems duomenims rašyti komentarų pavidalu. Taip pat čia galima paslėpti duomenis.

JPEG failų formate yra galimybė saugoti papildomą sumažintą vaizdo kopiją [4]. Ji yra saugoma nesuglaudinta ir jai galima pritaikyti LSB įterpimą.

1. 7. Neformatiniai metodai

Neformatiniai metodai remiasi tuo, kad skaitmeniniuose vaizduose saugoma perteklinė spalvinė informacija, t.y. žmogaus akys nėra jautrios mažiems spalvų svyravimams. Taip perteklinė informacija gali būti pašalinta ir vietoj jos įrašyti duomenys, kuriuos norima paslėpti.

1. 7. 1. LSB įterpimas

LSB tai paprasčiausias steganografijos algoritmas, kuris remiasi tuo, kad pikselio spalvinių komponenčių reikšmių jauniausi bitai iš esmės yra triukšmas ir nedaro matomos įtakos vaizdui [20]. Vaizdo pikselių vienos komponentės jauniausius bitus galima pavaizduoti bitų plokštuma, kuri

vadinama LSB plokštuma. Taip spalvotam vaizdui atitinka trys plokštumos, o vienspalviui viena. Šiose plokštumose ir įrašomas slaptas pranešimas, pakeičiant atitinkamai atskirų bitų reikšmes. Duomenų kiekis, kurį galima paslėpti vaizde atitinkamai lygus 1 bitui per pikseli vienspalviams vaizdams ir 3 bitai per pikselį spalvotiems.

1.2 pav. *24 bitų vaizdas ir jo raudonos spalvos kanalo LSB plokštuma*

Šio steganografijos algoritmo trūkumai yra tai, kad jo naudojimą galima lengvai aptikti analizuojant vaizdo histogramą, ir taip pat nors ir atrodytų, kad LSB plokštumos bitai yra pasiskirstę atsitiktinai (*žr* 1.2 pav.), jie yra kažkiek priklausomi nuo vyresnių bitų, todėl pakinta vaizdo statistinės charakteristikos.

LSB algoritmo naudojimo aptikimo tikimybės sumažinimui buvo sukurta jo modifikacija "±1" [3]. Vietoj to, kad tiesiog pakeisti bitą, atitinkanti reikšmė atsitiktinai padidinama arba sumažinama vienetu. Tokiu atveju gali pasikeisti ne būtinai tik vienas bitas. Slaptų duomenų atkūrimo algoritmas išlieka toks pat.

Darant prielaidą, kad jauniausių pikselių spalvinių komponenčių bitų reikšmės yra atsitiktinės, tai tikimybė, kad jis bus pakeistas naudojant LSB metodą, yra 1/2, t.y. dviems pranešimo bitams tenka vienas pakeitimas. Ir įterpimo efektyvumas lygus 2 bitams per pakeitimą. Kai pranešimo dydis yra

mažesnis už maksimalų dydį, kurį galima įterpti į vaizdą, tada įmanoma padidinti įterpimo efektyvumą naudojant matricinį įterpimą. Taip naudojant LSB įterpimą slepiant pranešimą, kurio dydis lygus 2/3 nuo maksimalaus, grupei iš trijų jauniausių pikselių spalvinių komponenčių bitų $g[1]$, $g[2]$ ir $g[3]$ teks du pranešimo bitai $m[1]$ ir $m[2]$. Pranešimo bitas $m[1]$ bus įterptas į $g[1]$ bitą, $m[2]$ į $g[2]$, o $p[3]$ bus praleistas ir efektyvumas bus lygus 3, kurį galima padidinti naudojant tokias formules [15]:

$$m[1] = LSB(g[1]) \oplus LSB(g[2]) \tag{1.1}$$

$$m[2] = LSB(g[2]) \oplus LSB(g[3]) \tag{1.2}$$

kur $\oplus$ tai griežtoji disjunkcija, *LSB()* – LSB algoritmo funkcija . Jeigu visos p reikšmės tenkina abi sąlygas, tai pakeitimai nedaromi. Jeigu tenkinama tik pirma sąlyga, tai pakeičiama $p[3]$ reikšmė, jei tik antra sąlyga, tai pakeičiama $p[1]$ reikšmė, jei netenkinamos abi sąlygos, tai pakeičiama $p[2]$ reikšmė. Kadangi kiekvieno atvejo tikimybė yra 1/4, tai tikimybė, kad bus pakeistas vienas iš trijų bitų lygi 3/4, kas duoda efektyvumą lygų 4. Pranešimo atkūrimas atliekamas naudojant tokią formulę:

$$m = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} p \tag{1.3}$$

Šie algoritmai gali būti taikomi nesuglaudintiems vaizdams arba kitiems skaitmenine forma išreikštiems duomenims.

1. 7. 2. JPEG formatui taikomi metodai

- LSB įterpimas

LSB įterpimą tiesiogiai į vaizdo duomenis įmanoma taikyti, kai naudojamas formate numatytas Lossless JPEG kodavimas [2]. Lossless JPEG tai kodavimo be praradimų būdas, kuris iš esmės skiriasi nuo kodavimo su praradimais, naudojančio DCT. Toks kodavimas retai naudojamas praktikoje.

- LSB įterpimas į kvantavimo koeficientus

JPEG failai įprastai turi vieną arba dvi kvantavimo koeficientų lenteles. Vienos lentelės dydis yra 64 baitai, todėl naudojant LSB metodą, vienoje lentelėje galima paslėpti tik 8 baitus. Be to kvantavimo koeficientų pakeitimas įneša pakeitimus į statistines suglaudinamų blokų charakteristikas, kas neigiamai veikia kodavimo efektyvumą.

- Papildomų kvantavimo lentelių naudojimas

JPEG formatas numato kelių kvantavimo lentelių naudojimą. Todėl didesniam slepiamų duomenų kiekiui sutalpinti, gali būti sukurtos papildomos kvantavimo lentelės [13]. Praktikoje naudojami du papildomų lentelių sukūrimo būdai. Pirmas, kai lentelės apskaičiuojamos taip, kad padidinti glaudinimo efektyvumą, kas ir numatyta JPEG specifikacijoje. Antras, kai sukuriamos netikros lentelės, kurios skiriasi tik jauniausiais bitais.

- Slėpimas vaizdo spektre

Metodas naudoja transformuotus ir kvantuotus vaizdo blokų koeficientus prieš juos suglaudinant. Duomenys gali būti slepiami jauniausiuose bituose. Slepiamų duomenų kiekis yra

proporcingas suglaudinto vaizdo dydžiui, be to didelio duomenų kiekio įterpimas gali sąlygoti didesnius pradinio vaizdo iškraipymus ir mažesnį glaudinimo efektyvumą.

1. 7. 3. Formatams su spalvų paletę taikomi metodai

- LSB įterpimas

Tiesioginis LSB metodo taikymas gali įnešti žymius iškraipymus į skaitmeninį vaizda, nes spalvų paletės elementai, kurių indeksai skiriasi jauniausiu bitu, gali turėti visiškai skirtingas spalvas [5]. Todėl dažnai naudojamas papildomas paletės apadorojimas.

Vienas paprasčiausių būdų išvengti iškraipymų, tai atlikti paletės analizę ir surasti tokias elementų poras, kurių spalvinių reikšmių skirtumas neviršija tam tikros ribos. Tada slėpimas vykdomas tik tuose atrinktuose elementuose. Slėpimo metu paletė nekeičiama ir pranešimo atkūrimas atliekamas analogiškai analizuojant paletę.

Tinkančių slėpimui elementų porų skaičius dažniausiai nėra didelis, todėl metodas gali būti modifikuotas. Prieš paletės analizę, ji yra surūšiuojama ir atitinkamai pakeičiamos vaizdo pikselių reikšmės, tada jau įterpiamas pranešimas.

- Paletės LSB įterpimas

Formatuose su paletę dažnai paletė yra saugoma viename faile su vaizdu, o tai reiškia, kad galima taikyti LSB metodą pačiai paletei. Paletės dydis neviršija 256-ų elementų, kurie saugomi analogiškai vaizdo pikseliams formatuose be paletės [9]. Taip šiuo būdų galima paslėpti iki 768-ų bitų. Paletėje gali atsirasti vienodi elementai, kas gali būti vienu iš steganografijos naudojimo požymių.

- Vienodi paletės elementai

Jeigu paletė sudaryta taip, kad joje yra vienodi elementai, kas yra įmanoma, bet nerekomenduojama, tai juos galima panaudoti pranešimo slėpimui.

Bendru atveju surandami paletės elementai, kurie yra dažniausiai naudojami vaizde ir paletė papildoma jų klonais. Toliau vaizde ieškoma pikselių, kurie nurodo į elementus turinčius klonus ir bitai koduojami nuorodomis į originalų elementą arba jo kloną.

- Paletės elementų sukeitimas

Jeigu paletės elementai nesikartoja ir jų skaičius yra n, tai aišku, kad galimų elementų išdėstymų skaičius lygus n!. Jeigu naudoti skirtingus išdėstymus dvejetainio pranešimo kodavimui, tai pranešimo maksimalus ilgis sudarys apie $\log_2(n!)$ bitų.

1. 8. Stegoanalizė

1. 8. 1. Vizualiniai metodai

Vizualiniai stegoanalizės metodai remiasi žmogaus regos sistemos sugebėjimu analizuoti vaizdus ir išskirti juose neatitikimus. Vizualinė analizė yra efektyvi kai stegokonteineris pilnai išnaudojamas, bet mažėjant įterptų duomenų kiekiui žmogaus akims vis sunkiau pastebėti įterpimo pėdsakus. Taip pat šie metodai netinka vaizdams, kurie yra suglaudinti su praradimais, nes juose jau yra iškraipymų.

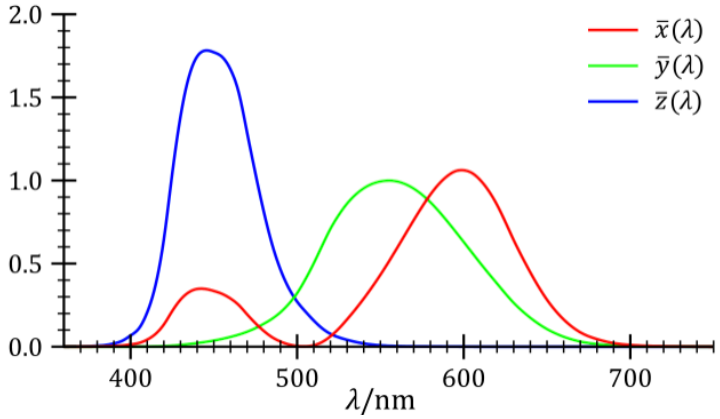
1. 8. 2. Statistiniai metodai

Į skaitmeninį vaizdą steganografijos būdu įterpiami duomenis iškraipo taip pat ir statistinius vaizdo parametrus. Statistiniai metodai analizuoja šiuos parametrus ir nustato ar jie panašūs į nemodifikuoto vaizdo parametrus, ar vaizde gali būti slepiami duomenys. Šie metodai yra tikimybiniai ir įvertina tikimybę, kad naudojama steganografija [10]. Analizei metodai gali naudoti įvairias statistines charakteristikas, tokias, kaip entropijos įvertinimas, koreliacijos koeficientai, priklausomybė tarp elementų sekų, sąlyginiai pasiskirstymai, pasiskirstymų atskyrimas pagal Chi-kvadratą ir kt.

1. 9. Spalvų modeliai skaitmeninių vaizdų aprašymui

Skaitmeninis vaizdas tai skaitinių reikšmių, apibrėžiančių pikselius – mažiausius atskyrus vaizdo elementus, matrica. Pikselio reikšmė tai atspalvis, kuris aprašomas tašku spalvų erdvėje taikant kurį nors spalvų modelį. Skaitmeninis vaizdas gali būti išsaugotas iš eilės saugant visų pikselių reikšmes, naudojant spalvų paletę ir vietoj pikselių reikšmių saugant jas atitinkančius palėtės indeksus arba saugant skaitmeninio vaizdo transformaciją.

Spalvų modelis tai matematinis spalvų pateikimo skaičių kortežų, vadinamų spalvinėmis komponentėmis arba spalvinėmis koordinatėmis, pavidalu modelis. Spalvų erdvė tai modelio užduodamos visos galimos spalvų reikšmės.



1.3 pav. *Standartinio stebėtojo spalvinio atitikmens funkcijos*

Spalvų modelis kuris naudojamas kaip etaloninis techninėse srityse yra CIE XYZ [7]. Juo gali būti aprašytas visas žmogaus matomas spektras spalvų. Jis sukurtas remiantis standartinio stebėtojo spalvinio atitikmens funkcijomis $\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda)$, kurios buvo nustatytos eksperimentiniu keliu(*žr* 1.3 pav.). Taip *X*, *Y* ir *Z* reikšmės apskaičiuojamos:

$$X = \int_{380}^{780} M(\lambda)\bar{x}(\lambda)d\lambda \tag{1.4}$$

$$Y = \int_{380}^{780} M(\lambda)\bar{y}(\lambda)d\lambda \tag{1.5}$$

$$Z = \int_{380}^{780} M(\lambda)\bar{z}(\lambda)d\lambda \tag{1.6}$$

Kur λ - atitinkamos monochromatinės šviesos bangos ilgis, *M* - spektrinis tankis (galios paskirstymas).

XYZ spalvų erdvė yra trimatė, bet dažniausiai vaizduojama naudojant CIE xy chromatinę diagramą. XYZ modelis sukurtas taip, kad *Y* reikšmė tai iš esmės yra skaistis. Tada spalvos

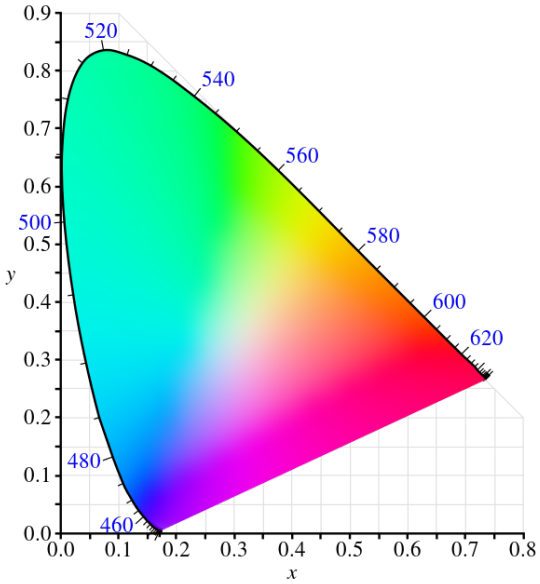
chromatiškumą galima nurodyti išvestiniais parametrais x ir y – dvejais iš trijų normalizuotu X,Y ir Z reikšmių.

$$x = \frac{X}{X + Y + Z} \tag{1.7}$$

$$y = \frac{Y}{X + Y + Z} \tag{1.8}$$

$$z = \frac{Z}{X + Y + Z} = 1 - x - y \tag{1.9}$$

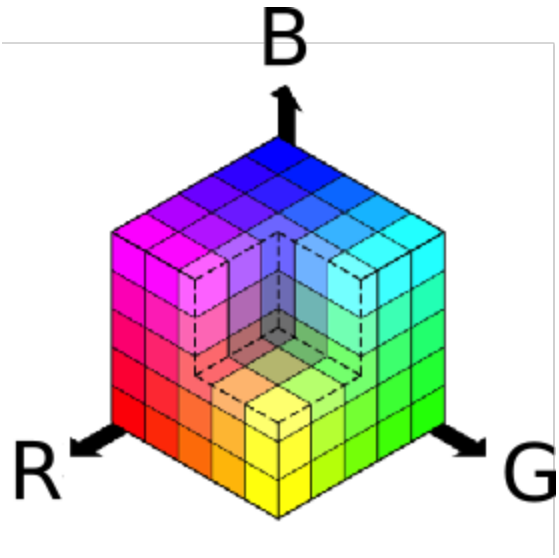
Taip chromatinė diagrama sudaroma naudojant x ir y koordinates prie tam tikro skaisčio z [16]. Visos galimos monochromatinės spektro spalvos diagramoje sudaro neuždarą kontūrą, vadinamą spektriniu lokusu(*žr* 1.4 pav.).



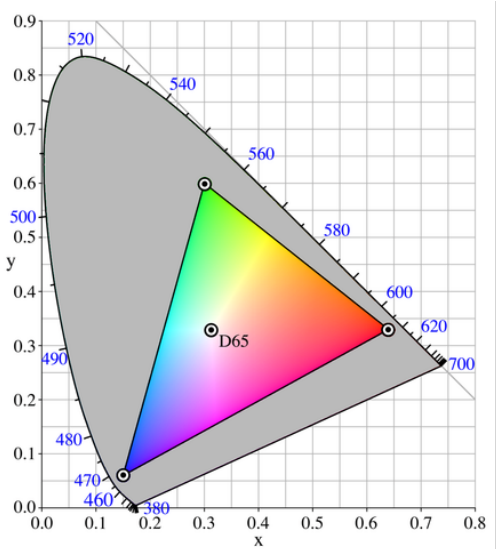
1.4 pav. *Spektrinis lokusas*

1. 9. 1. RGB

Matoma šviesa tai elektromagnetinės bangos, kurių bangos ilgis yra maždaug 380–750nm srityje. Žmogaus akies tinklainėje yra trijų skirtingų rūšių receptoriai – kolbutės, kurių jautrumo pikai atitinka raudonai (570nm), žaliai (540) ir mėlynai (420nm) šviesai. Receptorių į smegenis siunčiami elektriniai signalai leidžia skirti spalvas. Remiantis tuo buvo sukurtas spalvų modelis RGB. Pagal šį modelį norimas atspalvis gali būti gautas kombinuojant tris bazines spalvas – raudoną (r), žalią (g) ir mėlyną (b), vadinamas taip pat spalvos kanalais. Kanalo reikšmės dažniausiai išreiškiamos intervale nuo 0 iki 1. Taikant šį modeli skaitmeniniams vaizdams dažniausiai vieno kanalo reikšmių kodavimui naudojami 8 bitai ir jos išreiškiamos sveikaisiais skaičiais nuo 0 iki 255 imtinai. 0 atitinka minimaliam spalvos intensyvumui, o 255 atitinka maksimaliam spalvos intensyvumui. Taip (r, g, b) vektorius atitinkantis juodajai spalvai yra (0, 0, 0), o baltajai (255, 255, 255) [11]. Spalvų gylis – galimų užkoduoti spalvų kiekis – yra 2^{24} (16 777 216). Kartais kai reikia didesnio tikslumo arba apdorojant vaizdus, kanalų kodavimui naudojama daugiau bitų. Geometriškai modelis vaizduojamas RGB kubu (*žr* 1.5 pav.)– kanalų reikšmės naudojamos kaip koordinatės Dekarto trimatėje erdvėje.



1.5 pav. *RGB spalvų modelis pavaizduotas kubu*



1.6 pav. *sRGB spalvų erdvės spalvų perteikimo apribojimas*

Pats RGB modelis neapibrėžia konkrečių bazinių spalvų reikšmių ir todėl sumaišomos spalvos yra santykinės bazinių spalvų atžvilgiu. Kad modelis aprašytų absoliučias spalvų reikšmes, jam užduodama spalvų erdvė, kuri aprašo konkrečias bazinių spalvų bei balto taško reikšmes. Kompiuterinėje grafikoje dažniausiai naudojama sRGB erdvė(*žr* 1.6 pav.) [12]. Tada konvertavimas iš XYZ į RGB atliekamas taip:

$$\begin{bmatrix} R_{linear} \\ G_{linear} \\ B_{linear} \end{bmatrix} = \begin{bmatrix} 3{,}2406 & -1{,}5372 & -0{,}4986 \\ -0{,}9689 & 1{,}8758 & 0{,}0415 \\ 0{,}0557 & -0{,}2040 & 1{,}0570 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{1.10}$$

$$C_{srgb} = \begin{cases} 12{,}92C_{linear}, & C_{linear} \leq 0{,}0031308 \\ (1+a)C_{linear}^{1/2{,}4} - a, & C_{linear} > 0{,}0031308 \end{cases} \tag{1.11}$$

kur *X,Y,Z* – atitinkamos XYZ spalvų erdvės spalvos komponentės, *C* tai *R,G* arba *B* komponentės, o $\alpha = 0.055$.

Ir atvirkščiai:

$$C_{linear} = \begin{cases} \dfrac{C_{srgb}}{12{,}92}, & C_{srgb} \leq 0{,}04045 \\ \left(\dfrac{C_{srgb} + a}{1 + a}\right)^{2{,}4}, & C_{srgb} > 0{,}04045 \end{cases} \quad (1.12)$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0{,}4124 & 0{,}3576 & 0{,}1805 \\ 0{,}2126 & 0{,}7152 & 0{,}0722 \\ 0{,}0193 & 0{,}1192 & 0{,}9505 \end{bmatrix} \begin{bmatrix} R_{linear} \\ G_{linear} \\ B_{linear} \end{bmatrix} \quad (1.13)$$

1. 10. Skaitmeninių vaizdų saugojimo formatai

1. 10. 1. Taškinės grafikos formatai

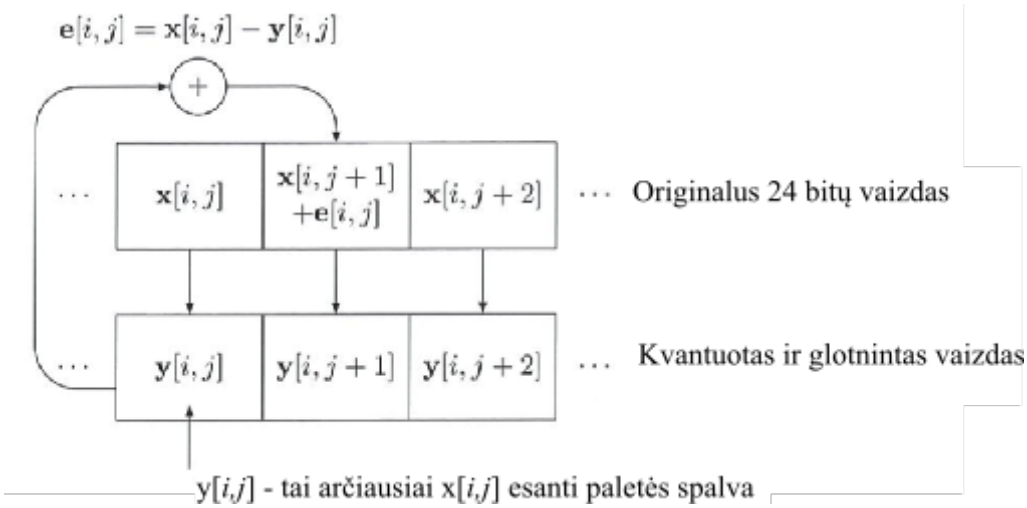
Šiuose formatuose skaitmeniniai vaizdai saugomi iš eilės įrašant vaizdo pikselių reikšmes vienu ar daugiau bitų priklausomai nuo formato ir spalvų gylio. BMP, TIFF, PNG – tai tipiniai taškinės grafikos formatai. Kadangi, esant pakankamam spalvų gyliui, vaizdas išsaugomas be praradimų, šie formatai dažnai naudojami kai reikia tikslumo, bet ir failų dydis atitinkamai padidėja. Kad sumažinti failo dydį šiuose formatuose gali būti naudojamas glaudinimas be praradimų.

1. 10. 2. Spalvų paletės formatai

Naudojant tokius formatus skaitmeninis vaizdas saugomas kaip spalvų paletė ir vaizdo dydžio paletės indeksų matrica. Paletę gali sudaryti iki 256 spalvų, kurios užduodamos RGB kortežais. Formatai labiau pritaikyti vaizdams, kuriuose naudojamos ne daugiau nei 256 spalvos, tada vaizdas gali būti išsaugotas be praradimų. Saugojant vaizdą, kuriame naudojama daugiau negu 256 spalvos, sukuriama spalvų paletė ir visų pikselių reikšmės pakeičiamos atitinkamais palėtės indeksais.

Paletę gali sudaryti fiksuotas nepriklausomas nuo vaizdo spalvų rinkinys, bet tokios fiksuotos palėtės trūkumas, kad dalis spalvų nenaudojama, o reikiamų spalvų trūksta [14]. Taip vaizdo kokybei pagerinti naudojama adaptyvi paletė, kuri sukuriama kiekvieno atskiro vaizdo pagrindu. Adaptyvi paletė gali būti sudarytą taikant įvairius algoritmus, vieni paprasčiausių būtų populiarumo ir medianinio pjūvio algoritmai. Populiarumo algoritmas iš vaizdo histogramos nustato dažniausiai skaitmeniniame vaizde pasitaikančias spalvas ir užpildo jomis paletę. Medianinio pjūvio algoritmas RGB kube sukuria tokį tariamą gretasienį, kad jis talpintų visas vaizde naudojamas spalvas, tada gretasienis dalinamas į du gretasienius ties mediana pagal didžiausią matmenį. Gauti gretasieniai vėl dalinami tokiu pat principu. Rekursyvus procesas tęsiasi kol gretasienių skaičius bus lygus paletėje naudojamam spalvų skaičiui, tada kiekviename gretasienyje apskaičiuojama vidutinė spalva ir įtraukiama į paletę.

Kai paletė sudaryta, pikselių reikšmės pakeičiamos paletės indeksais, kas vėlgi gali būti padaryta skirtingais būdais. Paprasčiausias būdas yra parinkti arčiausią paletėje esančią spalvą, bet gali būti naudojami ir glotninimo metodai. Paprasčiausią klaidos sklidimo glotninimo algoritmo veikimo principą galima pavaizduoti taip, kaip paveikslėlyje(*žr* 1.7 pav.).

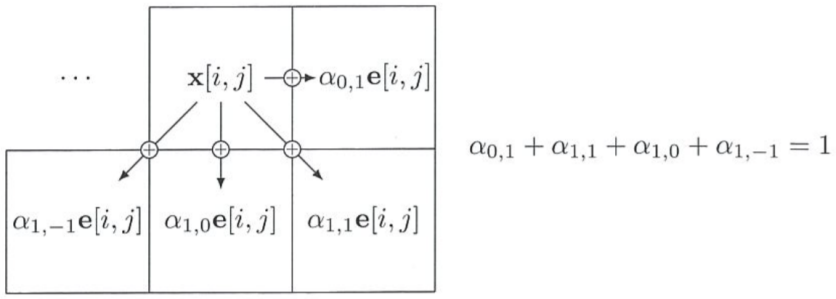


1.7 pav. *Klaidos sklidimo glotninimo algoritmo veikimo principas*

Kur *x*[] - pikselio spalva, *e*[] – skirtumas tarp pikselio spalvos ir jai arčiausios paletės spalvos *y*[].

Čia originalaus vaizdo pikseliai apdorojami iš eilės eilutėmis. Vienam pikseliui parenkama arčiausią paletės spalvos reikšmė, o kitas pikselis modifikuojamas pridedant klaidą - skirtumą tarp pirmo pikselio spalvos reikšmės ir jai parinktos palėtės spalvos reikšmės. Taip vaizde išsaugomas bendras spalvų balansas. Šis procesas gali būti patobulintas skleidžiant klaidą tarp didesnio pikselių kiekio su sąlyga, kad tai arti esantys pikseliai ir jie dar nėra apdoroti, o jiems priskirtos klaidos svorių suma lygi vienetui.

Vienas iš populiariausiu glotninimo algoritmų yra Floido-Stainbergo (Floyd-Steinberg) (*žr* 1.8 pav.). Čia klaida skleidžiama taip, kaip parodyta pav.



1.8 pav. *Floyd-Steinber glotninimo algoritmo veikimo principas*

Kur *x*[] - pikselio spalva, *e*[] – skirtumas tarp pikselio spalvos ir jai arčiausios palėtės spalvos, $\alpha_{0,1} = \frac{7}{16}, \alpha_{1,1} = \frac{1}{16}, \alpha_{1,0} = \frac{5}{16}, \alpha_{1,-1} = \frac{3}{16}$.

Glotninimo procesas pagrįstas tuo, kad žmogaus akis, žiūrint iš nuotolio mažame vaizdo gabalėlyje, sugeba apjungti spalvas taip suvokiant atspalvius, kurių iš tikrųjų nėra palėtėję.

1. 10. 3. Transformacijos-domeno formatai

Natūralius vaizdus žmogus suvokia kaip tekstūruotų segmentų visumą, o ne kaip taškų matrica. Taip pat žmogaus regėjimas nėra labai jautrus mažiems spalvos pakitimams bei aukštadažniam triukšmui. Dėl šių savybių vaizdų saugojimas spalvų reikšmių matricos pavidalu yra neefektyvus ir buvo sukurti glaudinimo formatai naudojant vaizdo transformacijas. Vaizdo transformacija yra lengviau suglaudinama, bet įnešą į vaizdą nežymius iškraipymus, kurie įprastomis sąlygomis nepastebimi. Dažniausiai naudojamos transformacijos tai diskretinė kosinusų transformacija

(DCT) ir diskretinė vilnelių transformacija (DWT). Formatai kurie jas naudoja yra atitinkamai JPEG ir JPEG2000. Labiau paplitusio formato JPEG glaudinimo algoritmą sudaro penki pagrindiniai žingsniai:

- Spalvos transformacija. Spalva iš RGB spalvų modelio transformuojama į YCrCb modelį, nors yra ir galimybė naudoti RGB ir kitus spalvų modelius.
- Padalinimas į blokus ir subdiskretizacija. Šviesumo kanalas Y padalinamas į 8x8 blokus, o spalvinių komponenčių Cr ir Cb kanalai prieš padalinant į blokus gali būti subdiskretizuoti, t.y. jų raiška gali būti sumažinta vertikalia ir/arba horizontalia kryptimi.
- DCT transformacija. Kiekvieno bloko signalais transformuojami į dažninį domeną naudojant DCT. DCT gali bųti apskaičiuota taip:

$$d[k,l] = \sum_{i,j=0}^{7} \frac{w[k]w[l]}{4} \cos\frac{\pi}{16}k(2i+1)\cos\frac{\pi}{16}l(2j+1)B[i,j] \tag{1.14}$$

Kur $d[k,l]$ – DCT koeficientų matrica, $k,l = 0,\ldots,7$, $B[i,j]$ – tai 8x8 šviesumo arba spalvinės komponentės blokas, $w[0] = 1/\sqrt{2}, w[k>0] = 1$.

Ir atvirkštinė transformacija:

$$B[i,j] = \sum_{k,l=0}^{7} \frac{w[k]w[l]}{4} \cos\frac{\pi}{16}k(2i+1)\cos\frac{\pi}{16}l(2j+1)d[k,l] \tag{1.15}$$

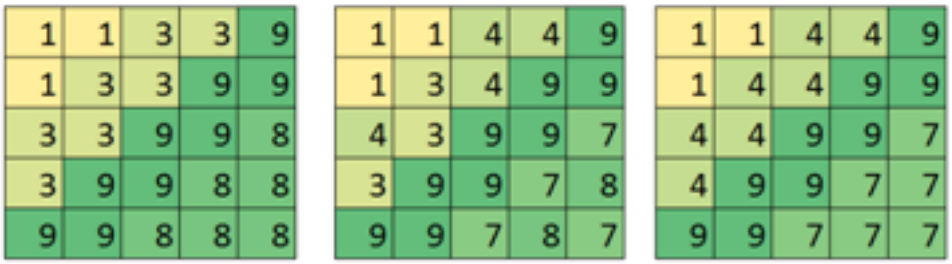
- Kvantavimas. Transformacijos koeficientai yra kvantuojami dalinant juos iš kvantavimo žingsnio ir apvalinami iki artimiausio sveikojo skaičiaus. Šviesumo ir spalviniams kanalams gali būti taikomi skirtingi kvantavimo žingsniai, kurių didesnės reikšmės padidiną glaudinimo koeficientą įnešant daugiau iškraipymų.
- Kodavimas ir glaudinimas be praradimų. Kvantuoti DCT koeficientai surūšiuojami zigzago tvarka, užkoduojami bitais ir suglaudinami be praradimų naudojant Huffman arba aritmetinį kodavimą. Gautas bitų srautas kartu su antraštė įrašomas į failą.

2. PAVEIKSLĖLIO SPALVŲ PALETĖS KEITIMU PAREMTA STEGANOGRAFIJA

Dauguma steganografijos algoritmų remiasi taškinės grafikos atitinkamų pikselių spalvos keitimu taip, kad vartotojas nepajustų didesnio skirtumo tarp originalaus ir duomenimis papildyto paveikslėlio. Tokio tipo pikselių spalvos keitimas įtakoja tai, jog tos pačios spalvos pikseliai skirtingose paveikslėlio vietose po papildomų duomenų įterpimo gali nesutapti, t.y. gali skirtis jų spalva, taip pat gali pakisti vaizde naudojamų spalvų kiekis.

Priklausomai nuo papildomos informacijos įterpimo (steganografijos algoritmo), pakeistame paveikslėlyje atsiranda skirtingo lygio papildomas taškinis triukšmas. Išeitis iš šios situacijos – keisti ne atskirų pikselių spalvą, o pačią paveikslėlių spalvų paletę. Jei būtų pakeičiama paveikslėlio spalvų paletė, tai visi vienos spalvos pikseliai paveikslėlyje būtų keičiami atitinkamos naujos spalvos pikseliais. Tai leistų užtikrinti, kad vieno atspalvio pikselių grupė ir toliau išliks vientisa, be vartotojui pastebimų pikselių spalvos svyravimų, t.y. neatsiras papildomo nepageidaujamo taškinio triukšmo paveikslėlyje.

Skirtumas tarp atskirų pikselių ir spalvų paletės keitimo pateiktas paveikslėlyje (*žr* 2.1 pav.). Šiame paveikslėlyje vaizduojamas 5x5 matmenų paveikslėlis, kuriame naudojamos spalvos su kodais nuo 1 iki 9. Dalyje (a) vaizduojamas originalus paveikslėlis, kuris susideda tik iš spalvų 1, 3, 8 ir 9. Tuomet dalyje (b) spalvų 3 ir 8 reikšmės kai kur pakeistos į spalvas 4 ir 7. Nors šie pakeitimai skiriasi tik vienu didesne/mažesne reikšme, tačiau atsitiktinių pikselių pakeitimas įtakoja bendro paveikslėlio vaizdo iškraipymą. Tuo tarpu dalyje (c) vaizduojama kaip pakistų paveikslėlis, jei visi spalvos kodo 3 pikseliai būtų keičiami į spalvos kodą 4, o spalvos kodo 8 pikseliai – į spalvos kodą 7. Šiuo atveju buvo pakeista daugiau pikselių, tačiau bendras paveikslėlio vaizdas yra artimesnis originalui (a), nei paveikslėlis po atsitiktinio šių spalvų pakeitimo (b).



a) b) c)

2.1 pav. *Originalaus paveikslėlio (a) pakeitimas keičiant atskirų pikselių spalvas (b) ir keičiant paveikslėlio spalvų paletę (c)*

Remiantis prielaida, kad paveikslėlio keitimas keičiant spalvų paletę, o ne atskirų pikselių spalvas mažiau įtakoja paveikslėlio vizualinį suvokimą, siūlomas steganografijos algoritmas, kuris papildomus duomenis įrašytų taip, kad viena spalva visur būtų keičiama kita, o ne keičiamos pavienių pikselių spalvos, t.y. siūloma įrašyti papildomus duomenis į paveikslėlio spalvų paletę.

Papildomų duomenų įrašymui į paveikslėlio naudojamą spalvų paletę reikia išskirti vietas, kurios gali būti keičiamos ir nesudaryti didesnių pakeitimų paveikslėlio bendram suvokimui. Jei paveikslėlio naudojamos spalvų paletės pakeitimai bus pernelyg dideli ar tinkamai neapgalvoti, paveikslėlis gali pasikeisti kardinaliai ir tapti vartotojui neatpažįstamas, nepriimtinos kokybės ar pan. Tad naujai kuriamam steganografijos algoritmui keliamos tokios prielaidos ir ribojimai:

1. Kelios paveikslėlio spalvos negali būti vaizduojamos kaip viena spalva pakeistame paveikslėlyje, t.y. kiekviena originalios spalvų paletės spalva turi turėti unikalų atitikmenį pakeistoje

spalvų paletėje. Tai leis užtikrinti, kad paveikslėlyje liks skirtumai tarp spalvų ir visas paveikslėlis nesuvienodės iki vienos spalvos.

2. Skaitmeniniai vaizdai dažnai naudoja tik dalį viso galimo spalvų spektro, todėl pasirinkta spalva turėtų būti keičiama į kitą iš to paveikslėlio nenaudojamų „reikšmiškai artimų“ spalvų.

3. Spalvų reikšmės skaitmeniniuose vaizduose nevienodai pasiskirto spalvų erdvėje ir tankiau išsidėsčiusių spalvų pokyčiams vartotojai gali būti jautresni, jų pakitimas bus labiau pastebimas, todėl šios spalvos laikomos labiau reikšmingomis, nei tos, kurios yra labiau viena nuo kitos nutolusios paveikslėlio spalvų erdvėje.

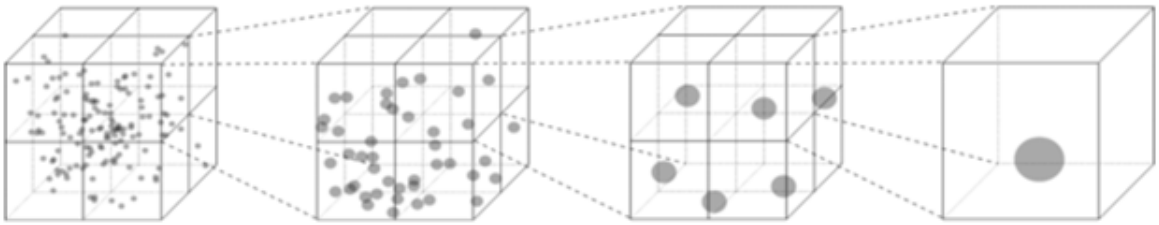
4. Spalvos, kurios yra labiau nutolusios nuo kitų spalvų paveikslėlio spalvų erdvėje laikomos mažiau reikšmingos ir jas galima keisti didesniuose rėžiuose, nes jei turimos dvi labai skirtingos spalvos, tai jas galima keisti pakankamai nemažai, kol jos nesuvienodėja ir vartotojas vis vien junta skirtumą tarp tų spalvų, o kadangi paveikslėlyje nėra kitų keičiamai spalvai artimų spalvų, jos atspalvių, tai vartotojas neturi su kuo lyginti ir mažiau pastebi atskiros spalvos pokytį.

5. Priklausomai nuo paveikslėlio ar jo atskirų spalvų, dalis spalvų gali likti nekeistos dėl pernelyg tankaus jų pasiskirstymo, todėl paveikslėlio naudojama spalvų paletė turi būti keičiama taip, kad būtų galima atsekti kurios paveikslėlio spalvos ar jo kodo vietos buvo pakeistos (neturint originalaus paveikslėlio) ir taip būtų galima išgauti į paveikslėlį papildomai pridėtus duomenis.

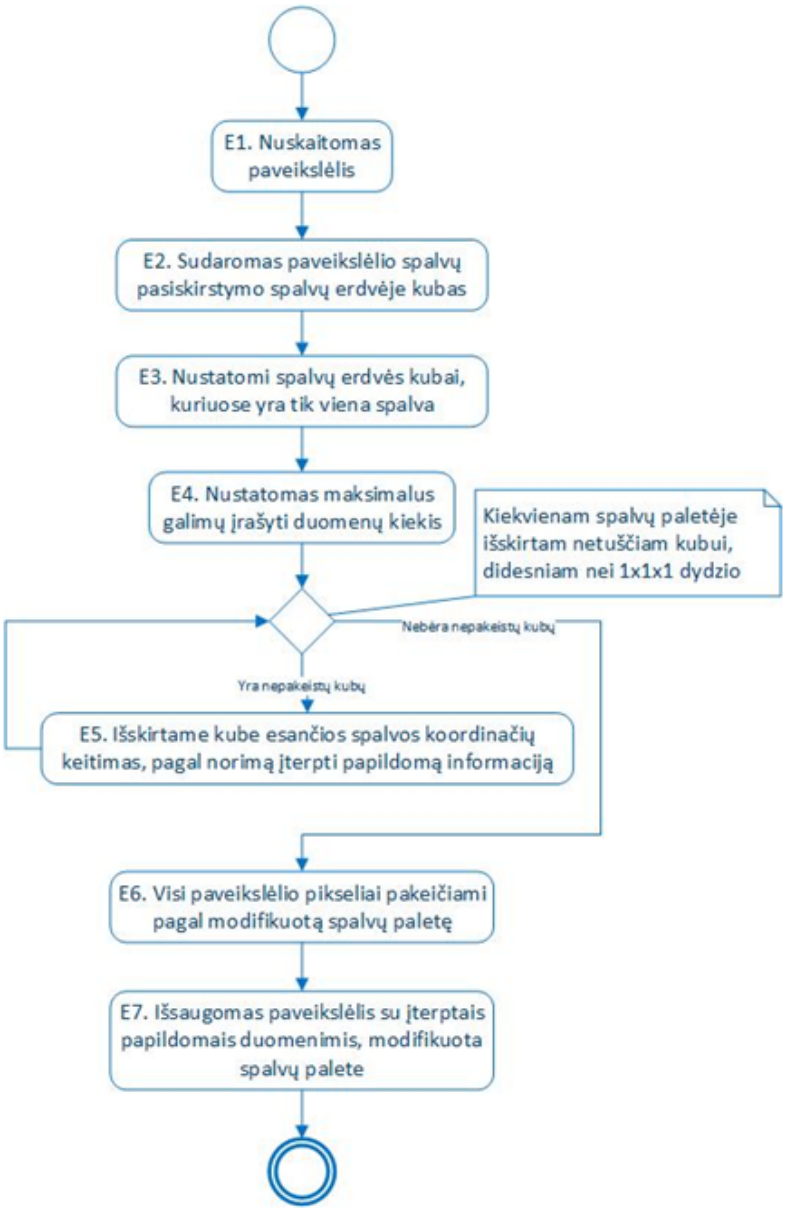
Vienas iš paprasčiausių sprendimų – keisti atitinkamus mažiausiai reikšminius bitus šiuo atveju nėra tinkamas, nes prieštarautų 1 ir 2 ribojimui, nes jei gautai paveikslėlio spalvų paletei kiekvienai spalvai būtų keičiami atitinkami bitai, nebūtų galima užtikrinti, jog po pakeitimo nebus gaunamas jau naudojamos spalvos kodas. Taip pat toks sprendimas neatsižvelgtų į 3 ir 4 prielaidas, nes neįvertintų spalvų pasiskirstymo spalvų erdvėje.

Siekiant realizuoti išsikeltas prielaidas ir reikalavimus, pasirinktas sprendimas dalinti RGB spalvų kubą pagal geometrinę progresiją tol, kol analizuojamame kube liks tik viena arba nulis paveikslėlio paletės spalvų. Toks dalinimas užtikrina, kad atrenkama tam tikra spalvų erdvės dalis, kurioje panaudojama tik viena iš spalvų, vadinasi ši spalva gali būti keičiama į bet kokią kitą spalvą iš tos srities, tad ji bus artima gretimose srityse esančioms spalvoms, bet jų neperdengs, t.y. pasirinkta spalva bus konvertuojama į kitą unikalią paveikslėliui spalvą. Kadangi pradedama dalinti nuo pilno RGB kubo iki mažesnių jo dalelių kol vienoje srityje bus tik viena spalva, tai tuo pačiu užtikrinama, kad bus galima identifikuoti labiau ir mažiau reikšmingas spalvas, t.y. jei sudalinus RGB kubą į smulkesnius kubus po vieną spalvą, tos srities matmenys nusako kiek tankiai toje RGB kubo vietoje buvo pasiskirsčiusios spalvos, tad atitinkamai bus atsižvelgiama ir į tų spalvų reikšmingumą, galimą svyravimo diapazoną.

Realizacijos patogumui, RGB kubą siūloma dalinti kiekvieną iš spalvų diapazonų dalinant pusiau. Taip RGB kubas padalinamas į 8 lygias dalis. Atitinkamai kiekvienas iš gautų aštuonių kubų, jei jame yra daugiau nei viena spalva, dalinamas vėl į 8 dalis ir taip tol, kol tame kubelyje lieka tik viena spalva (*žr* 2.2 pav.). Toks paveikslėlio naudojamos spalvų paletės dalinimas RGB kube leidžia gauti daug skirtingo dydžio nepersidengiančių kubų, kurie atitinka kokiose ribose gali keistis šiame kube esanti spalva.



2.2 pav. *Paveikslėlio naudojamų spalvų išsidėstymas spalvų erdvėje ir spalvų paletės RGB kubo dalinimas į 8 lygias dalis tol, kol viename kubelyje lieka tik viena spalva*



2.3 pav. *Siūlomo steganografijos algoritmo kodavimo (encoding) principinė schema*

Taip pat galimi atvejai, kada kubelyje nebelieka nei vienos spalvos. Tokia situacija rodo, kad ši spalvų erdvės sritis yra nenaudojama. Teoriniu požiūriu tokias sritis būtų galima taip pat išnaudoti, apjungiant su atitinkamomis gretimomis sritimis. Tai leistų padidinti spalvos keitimo rėžį, o kartu ir papildomai pridedamų duomenų kiekį, tačiau paprastas prijungimas gali apsunkinti pridėtos papildomos informacijos išgavimą, t.y. netenkintų 5 reikalavimo. Todėl priimama, kad gauti tušti (be jokios spalvos kubeliai) yra ignoruojami ir nenaudojami siūlomame steganografijos algoritme.

Apibendrinant išsikeltus reikalavimus, prielaidas ir idėjas, sudarytas naujas steganografijos algoritmas (*žr* 2.3 pav ir 2.4 pav.)

Duomenų įterpimui vykdomi šie pagrindiniai veiksmai:

E1. Nuskaitomas paveikslėlis. Eiliškumo tvarka gaunamas kiekvieno paveikslėlio pikselio 24 bitų RGB kodas (po 8 bitus kiekvienai spalvai).

E2. Sudaromas paveikslėlio spalvų pasiskirstymo spalvų erdvėje kubas. Sukuriamas trimatis loginio (boolean) tipo masyvas, kurio matmenys yra 256x256x256. Šiame masyve spalvos RGB kodą paverčiant į šio kubo koordinates, pažymima kurios spalvos naudojamos paveikslėlyje, o kurios ne.

E3. Nustatomi spalvų erdvės kubai, kuriuose yra tik viena spalva. RGB kubas dalinamas į 8 lygius kubus (kiekvienos ašies intervalą dalinant pusiau). Skaičiuojama kiek kiekviename iš šių 8 kubų yra reikšmių, kurios nusako, paveikslėlyje naudojamą spalvą. Jei kube yra 1 spalva – kubas išsaugomas tolesniam apdorojimui ir daugiau nebeskaidomas. Jei kube yra 0 spalvų – kubas toliau nebeanalizuojamas (šalinamas iš tolesnio apdorojimo). Jei kube yra daugiau nei 1 spalva – kubas vėl dalinamas į 8 smulkesnius vienodo dydžio kubus. Šie kubai vėl analizuojami atskirai ir dalinami tol, kol kubo matmenys tampa 1x1x1 dydžio. Tokio dydžio (1x1x1) kubai toliau nebeanalizuojami. Nors sudaromas identifikuotų po 1 spalvą aprašančių kubelių sąrašas, tačiau šiuos kubelius vėliau reikės susieti su bendru RGB kubu, todėl būtina užtikrinti sąryšį tarp šių objektų.

E4. Nustatomas maksimalus galimų įrašyti duomenų kiekis. Paveikslėlyje galimų išsaugoti duomenų kiekis priklauso nuo paveikslėlio naudojamos spalvų paletės, paveikslėlio spalvų pasiskirstymo. Galimų išsaugoti duomenų kiekis M_{max} leidžia įvertinti kiek būtent tame paveikslėlyje bus galima išsaugoti papildomos informacijos. M_{max} reikšmė paskaičiuojama radus visus spalvų erdvės kubus, kuriuose yra tik viena spalva. Tam naudojama formulė:

$$M_{max} = \sum_{t=0}^{N} \log_2 n_i \cdot 3 \qquad (2.1)$$

čia N – išskirtas spalvų erdvės kubų, kuriuose yra tik po vieną spalvą, kiekis; n_i – i-tojo išskirto kubo briaunos ilgis.

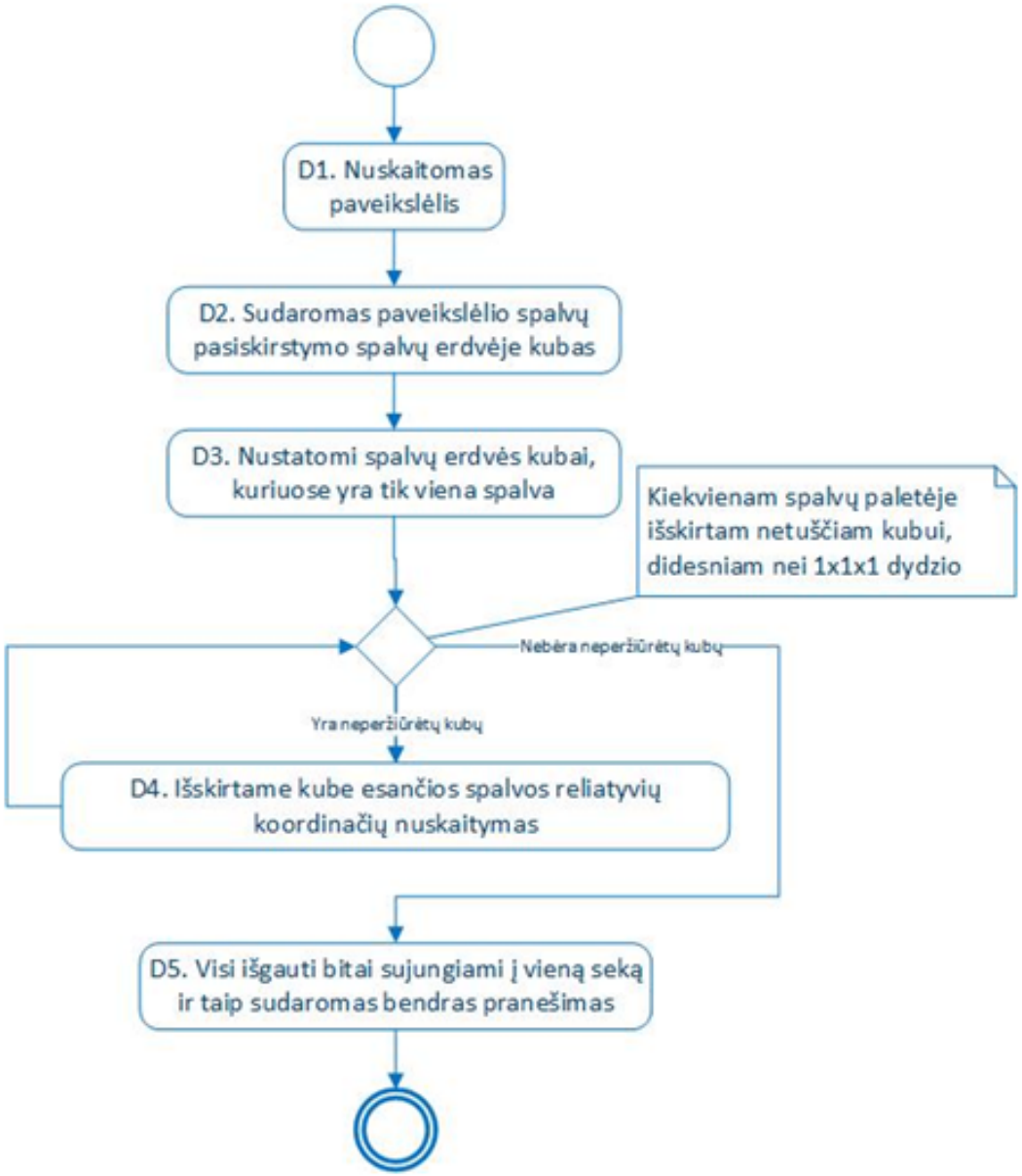
E5. Išskirtame kube esančios spalvos koordinačių keitimas, pagal norimą įterpti papildomą informaciją. Spalvos su *n* ilgio kraštinėmis koordinatės aprašomos $\log_2(n)$*3 bitų ilgio adresu ($\log_2(n)$ nusako kiek bitų reikia vienos ašies reikšmių išsaugojimui, o kadangi ašys yra 3, tai bitų skaičius dar dauginamas iš 3). Šie bitai (spalvos reliatyvios koordinatės išskirtame kube) pakeičiami įterpiamais duomenimis.

Pavyzdžiui jei turimas kubas, kurio kraštinės ilgis *n* yra lygus 2, tai jame gali būti aprašytos 8 skirtingos spalvos (2x2x2). Originalus paveikslėlis išnaudoja tik vieną iš šių spalvų, tad norint tą spalvą galima keisti bet kokia kita spalva, kuri telpa tame išskirtame spalvų kube. Tad jei originaliame paveikslėlyje yra balta spalva, kurios kodas yra #FFFFFF, t.y. raudonos, žalios ir mėlynos spalvų dešimtainė reikšmė yra 255, o išskiriant RGB kube ši spalva patenka į 2 reikšmių ilgio kraštinių kubą, tai kiekvieną iš trijų spalvos dedamųjų (raudonos, žalios ir mėlynos spalvos kiekius) galima pakeisti duomenimis iš intervalo nuo 0 iki 1 tame kube. Atitinkamai gaunant absoliučias kubo koordinates gauname, kad spalvos kodas kiekvienai spalvos dedamajai gali svyruoti nuo 254 iki 255. Tad baltą spalvą galima keisti kodais #FFFFFF, #FFFFFE, #FFFEFF, #FEFFFF, #FFFEFE, #FEFFFE, #FEFEFF, #FEFEFE.

E6. Visi paveikslėlio pikseliai pakeičiami pagal modifikuotą spalvų paletę. Kuomet visiems išskirtiems kubams yra pakeičiamos reliatyvios spalvos koordinatės, visi paveikslėlio pikseliai pereinami ir pakeičiami iš buvusios reikšmės į naują modifikuotą spalvą. Modifikuota spalva keičiama į absoliutų kiekviename išskirtame kubelyje esančios spalvos adresą. Būtent todėl kiekvieną išskirtą kubą būtina susieti su tuo, kur jis yra RGB kubo atžvilgiu. Taip pat patartina išsisaugoti originalaus, nekeisto RGB kubo duomenis, kad būtų lengviau susieti kiekvienos spalvos pokyčius keičiant kiekvieną paveikslėlio pikselį.

E7. Išsaugomas paveikslėlis su įterptais papildomais duomenimis, modifikuota spalvų palete. Pakeitus spalvų paletę ir pikselių spalvas, paveikslėlis išsaugomas pagal pasirinktą duomenų formatą be praradimų ir toliau naudojamas pagal norimą paskirtį.

Kuomet norima iš gauto pranešimo išgauti pridėtus duomenis, vykdomas analogiškas spalvų paletės dalinimas į kubus, tačiau vietoj duomenų įterpimo yra nuskaitomi reliatyvūs spalvų adresai bendro pranešimo sudarymui. Tad bendrai yra vykdomi šie pagrindiniai stegopranešimo išgavimo žingsniai, kurie pavaizduoti paveikslėlyje (*žr* 2.4 pav)



2.4 pav. *Siūlomo steganografijos algoritmo dekodavimo (decoding) principinė schema*

D1. Nuskaitomas paveikslėlis. Eiliškumo tvarka gaunamas kiekvieno paveikslėlio pikselio 24 bitų RGB kodas (po 8 bitus kiekvienai spalvai).

D2. Sudaromas paveikslėlio spalvų pasiskirstymo spalvų erdvėje kubas. Sukuriamas trimatis loginio (boolean) tipo masyvas, kurio matmenys yra 256x256x256. Šiame masyve spalvos RGB kodą paverčiant į šio kubo koordinates, pažymima kurios spalvos naudojamos paveikslėlyje, o kurios ne.

D3. Nustatomi spalvų erdvės kubai, kuriuose yra tik viena spalva. RGB kubas dalinamas į 8 lygius kubus (kiekvienos ašies intervalą dalinant pusiau). Skaičiuojama kiek kiekviename iš šių 8 kubų yra reikšmių, kurios nusako, paveikslėlyje naudojamą spalvą. Jei kube yra 1 spalva – kubas išsaugomas tolesniam apdorojimui ir daugiau nebeskaidomas. Jei kube yra 0 spalvų – kubas toliau nebeanalizuojamas (šalinamas iš tolesnio apdorojimo). Jei kube yra daugiau nei 1 spalva – kubas vėl dalinamas į 8 smulkesnius vienodo dydžio kubus. Šie kubai vėl analizuojami atskirai ir dalinami tol, kol kubo matmenys tampa 1x1x1 dydžio. Tokio dydžio (1x1x1) kubai toliau nebeanalizuojami. Nors sudaromas identifikuotų po 1 spalvą aprašančių kubelių sąrašas, tačiau šiuose kubelius vėliau reikės susieti su bendru RGB kubu, todėl būtina užtikrinti sąryšį tarp šių objektų.

D4. Išskirtame kube esančios spalvos reliatyvių koordinačių nuskaitymas. Išskirtame kube gaunamos spalvos reliatyvios koordinatės, kurios ir nusako kokie būtent duomenys buvo pateikti šiame kube.

D5. Visi išgauti bitai sujungiami į vieną seką ir taip sudaromas bendras pranešimas. Atskirų nuskaitytų bitų apjungimas į vieną seką leidžia suformuoti vientisą pranešimą, kuris ir bus pateikiamas vartotojui, kaip dešifruotas stegopranešimas.

Atsižvelgiant į tai, kad itin mažai spalvų turinčiuose paveikslėliuose spalvos keitimo intervalas gali būti pakankamai didelis (viena spalvos dedamoji 128x128x128 dydžio kube gali svyruoti per pusę savo galimos įgyti reikšmės), tai norint kontroliuoti modifikuojamo paveikslėlio kokybę, galima įtraukti maksimalaus kubo briaunos ilgio ribojimą. Toks ribojimas leistų užtikrinti tam tikro intervalo spalvos svyravimus.

Iš siūlomo algoritmo veikimo principo aišku, kad duomenų kiekis, kurį galimą paslėpti skaitmeniniame vaizde priklauso nuo vaizde naudojamo spalvų kiekio bei nuo šių spalvų pasiskirstimo RGB kube. Kadangi 24 bitų spalvų gylio vaizduose įmanomų spalvų kiekis yra ribotas ir yra lygus 2^{24}(viena spalva išreiškiama 24 bitais), tai galima apskaičiuoti ir maksimalų duomenų kiekį, kuriuos leidžia paslėpti siūlomas algoritmas. Idealios sąlygos, prie kurių pasiekiamas įterpiamų duomenų maksimumas, būtų tada, kai visas RGB kubas sudalinamas į 2^{21} kubelius su briaunos ilgiu 2 ir visi jų talpina po vieną spalvą. Didesnių kubų naudojimas yra mažiau efektyvesnis, nes vienu 4x4x4 kubu galima užkoduoti 6 bitus, tuo tarpu kai jį sudarančiais mažesniais 2x2x2 dydžio 8 kubais bendrai galima užkoduoti 24 bitus(vienu 2x2x2 kubu galima užkoduoti 3 bitus).Taip maksimalus duomenų kiekis, kuri gali įterpti algoritmas idealiomis sąlygomis apskaičiuojamas pagal formulę (2.1) ir yra lygus $3*2^{21}$ bitų arba 768 kilobaitai.

Kadangi siūlomas algoritmas neįtakoja vaizde naudojamo spalvų kiekio, algoritmą galima taikyti ir vaizdams naudojantiems paletę. Šie vaizdai pasižymi tuo, kad spalvų kiekis juose yra ribotas - iki 256 spalvų, o paletę sudaro 24 bitų spalvos. T.y. spalvos yra išsidėsčiusios toje pačioje RGB erdvėje, kaip ir 24 bitų spalvų gylio spalvos, tačiau jų, dažniausiai, žymiai mažiau, kas įtakoja ir maksimalų įmanomą paslėpti duomenų kiekį. Idealios sąlygos duomenų slėpimui, maksimalaus duomenų kiekio atžvilgiu, yra tada, kai 256 spalvos yra taip pasiskirsčiusios RBG kube, kad kiekviena jų patenka į atskirą 32x32x32 kubą. Vienu 32x32x32 kubu galima užkoduoti 15 bitų. Tada maksimalus duomenų kiekis, kurį įmanoma užkoduoti vaizdo paletėje siūlomu algoritmu yra lygus 3480 bitų.

Taip pat pažymėtina, kad taikant paletės pakeitimo steganografijos algoritmą skaitmeniniams vaizdams, kurie naudoja standartinę paletę, modifikuotą paletę reikia saugoti kartu su indeksais, dėl ko padidėja failo dydis.

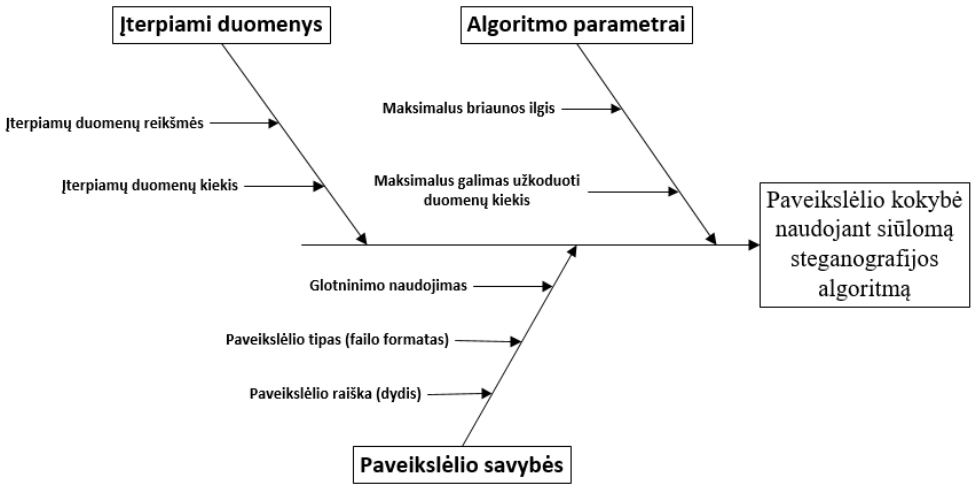
Aprašytas steganografijos algoritmas gali būti taikomas, kai stegokonteineris su įrašytu stegopranešimu yra išsaugomi formatu be praradimų. Vaizdų glaudinimo su praradimais algoritmai, kurie pašalina iš vaizdo perteklinę ir nereikšminę informaciją, atitinkamai iškraipo vaizdo spalvų atkūrimą. Tai reiškia, kad negalima atkūrti ir siūlomu algoritmu įterptos į vaizdą informacijos, kuri buvo įrašyta į vaizdo spalvų paletę.

3. SIŪLOMO STEGANOGRAFIJOS ALGORITMO TYRIMO METODOLOGIJA

Tyrimo metu bus nustatoma, kaip siūlomo paletės pakeitimo steganografijos algoritmo naudojimas įtakoja skaitmeninių vaizdų kokybę, kokie algoritmui būdingi iškraipymai atsiranda skaitmeniniuose vaizduose ir kaip jie priklauso nuo algoritmo parametro - naudojamų kubų briaunos ilgio, ir nuo skaitmeninių vaizdų tipų.

3. 1. Tyrimo metu analizuojami faktoriai

Savybės, kurios įtakoja siūlomo steganografijos algoritmo pagalba pakeisto paveikslo kokybę pavaizduotos paveikslėlyje (*žr* 3.1 pav.). Jos suskirstytos į tris kategorijas: steganografijos algoritmo parametrai; informacijos slėpimui naudojamo paveikslėlio savybės; paveikslėlyje slepiami duomenys.



3.1 pav. *Paveikslėlio kokybę naudojant siūlomą steganografijos algoritmą galinčios įtakoti savybės*

Vertinant siūlomo algoritmo pagalba keičiamo paveikslėlio kokybę atsižvelgiama į visas šias savybes:

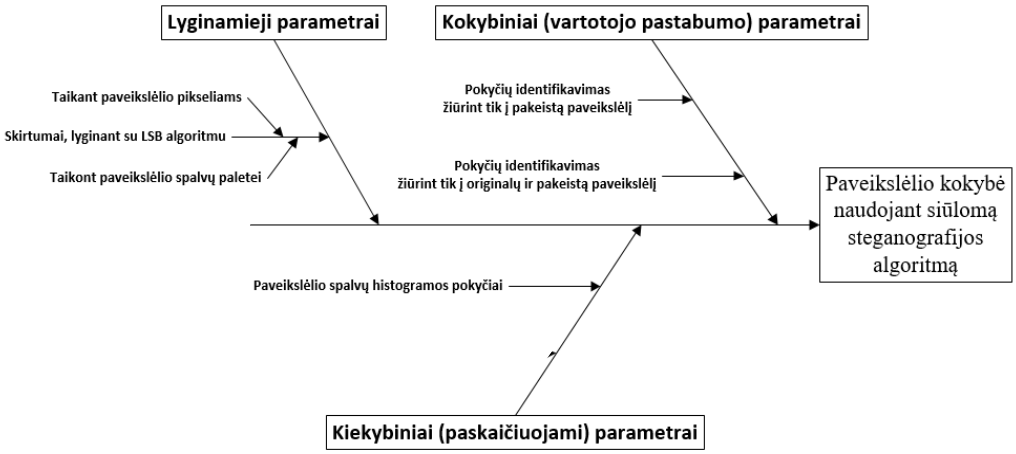
1. Įterpiamų duomenų reikšmės – kiekvieną kartą bus generuojama atsitiktinė bitų seka, taip siekiant užtikrinti, jog įterpiami duomenys nėra parinkti tikslingai.
2. Įterpiamų duomenų kiekis – atliekant testavimą bus bandoma įterpti maksimalų leistiną duomenų kiekį. Tais atvejais, kada siūlomo algoritmo rezultatai bus lyginami su kitu steganografijos algoritmu, vienodų sąlygų užtikrinimui abejais metodais bus koduojamas vienodas duomenų kiekis. Tam bus atrenkamas mažiausias maksimalus galimų įterpti duomenų kiekis tarp abiejų naudojamų algoritmų.
3. Paveikslėlio raiška (dydis) – paveikslėliuose, kuriuose nėra naudojamos spalvų paletės, spalvų kiekis labai susijęs su paveikslėlio dydžiu, jo raiška, todėl eksperimentai bus vykdomi su didelės ir mažos raiškos 24 bitų gylio paveikslėliais:
    - 3.a Didelės raiškos paveikslėliai – nuotraukos, kurių dydis svyruoja nuo 4 iki 24 milijonų taškų (Mpx) ir kurios parinktos iš asmeninio nuotraukų archyvo.
    - 3.b Mažos raiškos paveikslėliai – skaitmeniniai paveikslėliai, kurių dydis iki 1 milijono taškų (Mpx), parinktų iš interneto pagal naudojamą užklausą “cats” paieškos sistemos “Google”.
4. Paveikslėlio tipas (failo formatas) – eksperimentai, vykdomi su didelės ir mažos raiškos paveikslėliais yra išsaugoti taip, kad kiekvienas paveikslėlio pikselis

aprašomas 24 bitų gylio spalva ir nėra naudojama spalvų paletė, todėl tyrimui mažos raiškos paveikslėliai modifikuojami, juos pritaikant skirtingo tipo paveikslėlių panaudojimo siūlomam steganografijos algoritmui įvertinti. Šiam papildomų paveikslėlių generavimui atlikti naudojama GIMP programa, kurioje visi mažos raiškos paveikslėliai pakeičiami į vaizdus su adaptyvia palete iki 256 spalvų ir standartine 216 spalvų palete.

5. Glotninimo naudojimas – glotninimo naudojimas paveikslėliuose šiek tiek pakeičia paveikslėlio kokybę, todėl tyrime naudojami paveikslėliai tiek su glotninimu, tiek ir be glotninimo, todėl generuojant paveikslėlius su standartine ir adaptyvia palete iki 256 spalvų yra daromos kiekvienam mažos raiškos paveikslėliui dvi kopijos – viena su glotninimu, o kita be glotninimo.
6. Maksimalus briaunos ilgis – reguliuojant maksimalų briaunos ilgį galima kontroliuoti siūlomo algoritmo pagalba pakeisto paveikslėlio kokybę, todėl įterpiant papildomus duomenis bus netaikomas maksimalaus briaunos ilgio ribojimas, o tada dar atliekamas papildomas testas, kurio metu maksimalus briaunos ilgis yra 2 (gaunami kubai, nedidesni nei 2x2x2 matmenų).
7. Maksimalus galimas užkoduoti duomenų kiekis – ši savybė priklauso nuo paveikslėlio ir algoritmo savybių, todėl negali būti tiesiogiai keičiama tyrimo metu.

3. 2. Tyrimo metu stebimi parametrai

Tyrimo metu taip pat stebimi keli parametrai, kurie gali nusakyti pakeisto paveikslėlio kokybę (*žr* 3.2 pav.). Jie suskirstyti į keturias kategorijas: kokybiniai parametrai; kiekybiniai parametrai; lyginamieji parametrai; atsekamumo parametrai.



3.2 pav. *Paveikslėlio kokybę naudojant siūlomą steganografijos algoritmą galintys nusakyti parametrai*

Nors kelių paveikslėlių tarpusavio lyginimui egzistuoja keletas skaitinių metodų, tačiau šiame tyrime labiau remiamasi vizualia stegoanalize, įvertinant atsirandančius steganografijos algoritmų veikimo metu skaitmeninių vaizdų ir jų histogramų iškraipymus, nes steganografijos esmė iš principo ir yra pridėti papildomų duomenų taip, kad vartotojas pakeitimų vizualiai nepajustų. Tad atkreipiamas dėmesys į šiuos po duomenų įterpimo paveikslėliuose gaunamus parametrus:

1. Pokyčių identifikavimas žiūrint tik į pakeistą paveikslėlį – stebimas paveikslėlis ar jame matomi tam tikri netikėti elementai ar akivaizdūs iškraipymai.
2. Pokyčių identifikavimas žiūrint į pakeistą ir originalų paveikslėlį – stebimas paveikslėlis ir lyginamas su originaliu paveikslėliu, bandant identifikuoti vizualiai pakitusias vietas.

3. Paveikslėlio spalvų histogramų pokyčiai – paveikslėliui sudaroma jo spalvų histograma ir stebima kaip ji pakito, lyginant su originaliu paveikslėliu.
4. Skirtumai lyginant su LSB algoritmu – savaime suvokiama, kad steganografijos algoritmas iškraipo paveikslėlį, tad labiau reikėtų lyginti kiek paveikslėlio iškraipymai skiriasi su kitais steganografijos algoritmais, o ne lyginant su originaliu paveikslėliu. Todėl tyrimo metu vertinama ne tik siūlomo steganografijos algoritmo pagalba pakeisto paveikslėlio kokybė, bet ir skirtumai, lyginant su LSB steganografijos algoritmu pakeisto paveikslėlio kokybe. LSB steganografijos algoritmas palyginimui pasirinktas dėl to, kad prie tam tikrų sąlygų (kai paletės pakeitimo steganografijos algoritmas naudoja kubus su briaunos ilgiu 2) abiejų algoritmų veikimo principas yra panašus – skaitmeninio vaizdo kai kurių pikselių spalvos pakinta 1. Kadangi skiriasi paveikslėlių tipai, tai LSB algoritmo taikymas taip pat skiriasi ir išskiriami trys atvejai:
    4.a Taikant paveikslėlio pikseliams – tai standartinis algoritmas, kuris kiekvieną paveikslėlio pikselį pakeičia, jo spalvos RGB kode mažaisiais reikšminius spalvų bitus keičiant įterpiamų duomenų bitais.
    4.b Taikant paveikslelio pikselių indeksams – kai paveiklėlyje naudojama spalvų paletė, pikselių reikšmęs išreiškiamos spalvų paletės indeksais, kurie pakeičiami įterpiant duomenis.
    4.c Taikant paveikslėlio spalvų paletei – kadangi vaizduose su palete spalvų paletę sudaro 24 bitų spalvos, LSB algortimą galima taikyti paletės elementams keičiant jauniausius spalvų komponenčių bitus.

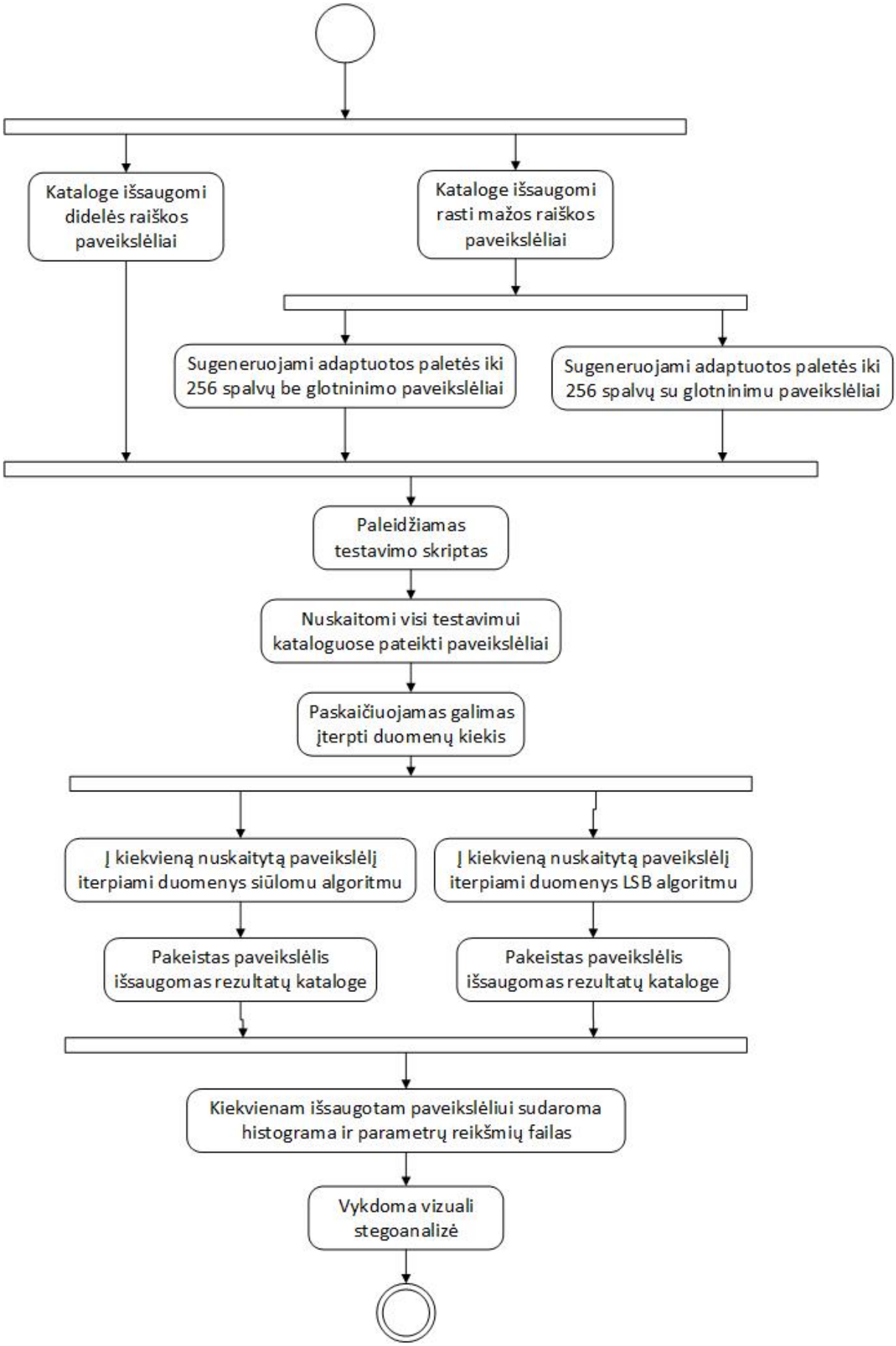
3. 3. Tyrimo eiga ir tyrimo duomenų kiekis

Vykdant siūlomą steganografijos algoritmą,veiksmai atliekami pavaizduota seka (*žr* 3.3 pav.).

Kaip matoma pateiktame paveikslėlyje, pirma yra parengiami duomenys testavimui. Jie vykdomi neautomatizuotai, pačio testuotojo. Kuomet turimi testavimui naudojami paveikslėliai, duomenų įterpimą į pateiktus paveikslėlius ir analizei reikalingų duomenų parengimą vykdo sukurtas programinis kodas (parašytas ir vykdomas MATLAB® aplinkoje, nes teik siūlomas, tiek LSB algoritmas realizuoti taip pat šioje aplinkoje). Šio programinio kodo pagalba yra nuskaitomi visi pateikti failai, juose įvertinamas galimas įterpti papildomų duomenų kiekis, naudojant skirtingus siūlomo algoritmo parametrus, tas duomenų kiekis sugeneruojamas ir įterpiamas naudojant siūlomą ir LSB steganografijos algoritmus. Tokiu būdu vienam paveikslėliui yra sugeneruojami keli, priklausomai nuo originalaus paveikslėlio savybių, rezultatų paveikslėliai. Gautiem paveikslėliams taip pat sugeneruojamos spalvų histogramos ir kiti vizualinei stegoanalizei reikalingi duomenys. Kada visi duomenys analizei jau paruošti automatiškai vykdomas ekspertinis vertinimas pačio testuotojo ir apibendrinami gauti testavimo rezultatai.

Tyrime iš viso panaudota 148 paveikslėliai, o kadangi jie visi papildyti duomenimis naudojant pasiūlytą algoritmą ir LSB algoritmą, tai iš viso išanalizuota 1707 paveikslėliai po duomenų įterpimo. Šie visi duomenys apibendrinami, bet konkrečių skirtumų išskyrimui, pateikti atskiri pavyzdžiai, kurie leidžia aiškiau suvokti skirtumus tarp LSB ir siūlomo algoritmo.

Šio tyrimo metu naudojami duomenys pateikti 3.1 lentelėje.



3.3 pav. *Tyrimo eigos pagrindinė seka*

3.1 lentelė. Tyrimui naudojamas paveikslėlių kiekis ir jų tipai

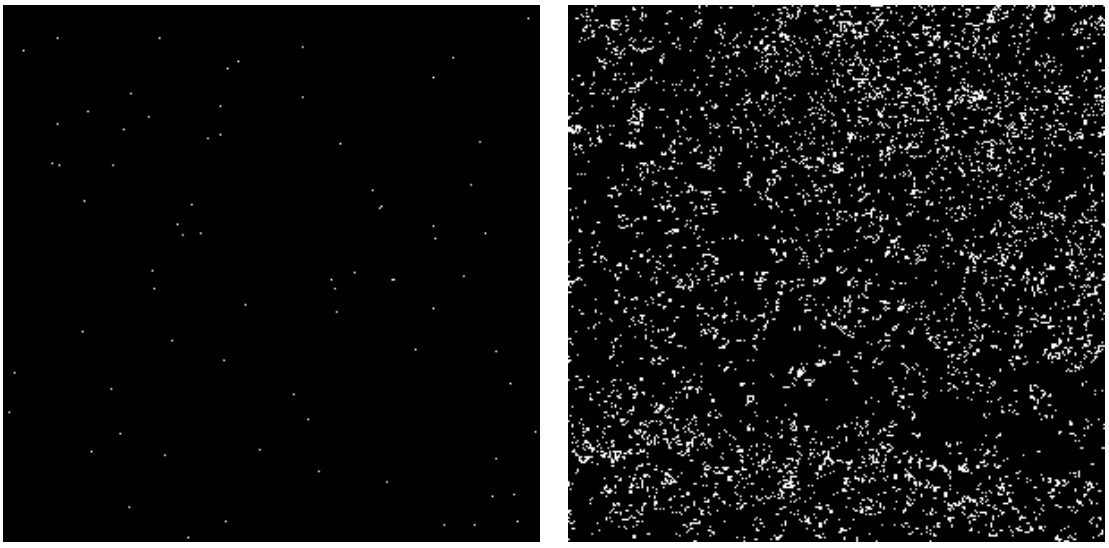
Nr.	Paveikslėlių kategorija	Paveikslėlių kiekis duomenų įterpimui	Siūlomu algoritmu gauti failai	LSB algoritmu gauti failai
1.	24 bitų spalvų gylio didelės raiškos (4 - 24 milijonų taškų) skaitmeninės nuotraukos, kurios atsitiktinai parinktos iš asmeninio nuotraukų archyvo	48	281	96
2.	24 bitų spavų gylio mažos raiškos (iki 1 milijono taškų) skaitmeniniai vaizdai, kurie atsitiktinai parinkti iš vaizdų, pateiktų paieškos sistemos “Google” naudojant užklausą “cats”	20	114	40
3.	vaizdai su adaptyvia palete iki 256 spalvų, be glotninimo	20	122	162
4.	vaizdai su adaptyvia palete iki 256 spalvų, su glotninimu	20	129	169
5.	vaizdai su standartine palete iki 256 spalvų, be glotninimo	20	130	170
6.	vaizdai su standartine palete iki 256 spalvų, su glotninimu	20	127	167
	Bendra suma:	**148**	**1707**	

4. REZULTATŲ ANALIZĖ

4. 1. Steganografija didelės raiškos 24 bitų spalvų gylio skaitmeninėse nuotraukose

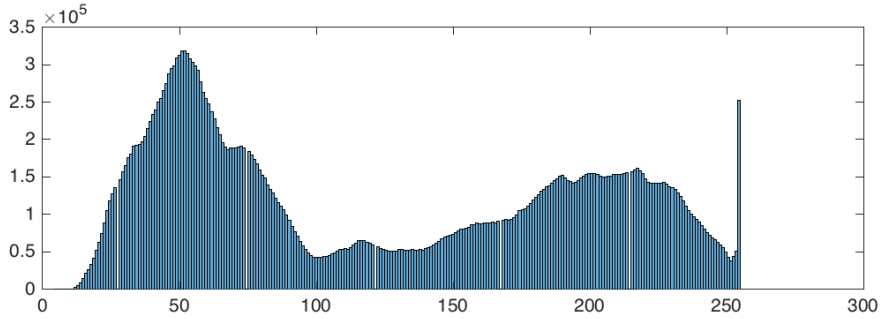
Eksperimento duomenys, parodė, kad vaizduose naudojamas spalvų kiekis labai silpnai koreliuoja su vaizdo pikselių kiekiu (koreliacijos koeficientas ~0,035), kas parodo, kad spalvų kiekis labiau priklauso nuo kompozicijos arba kitų faktorių. Tuo tarpu, maksimalus įterpiamų duomenų kiekis, taikant siūlomą algoritmą, kaip ir tikėtasi stipriai koreliuoja su naudojamų spalvų kiekiu (koreliacijos koeficientas ~0,76). Todėl galima teigti, kad, bendru atveju, mažesnės raiškos vaizdų naudojimas su šiuo steganografijos algoritmu, yra efektyvesnis už didesnės raiškos vaizdų naudojimą. Šiuo atveju efektyvumo vidurkis siekia apie 0,03 bito įterptų duomenų vienam vaizdo pikseliui, o standartinis nuokrypis 0,02, kas irgi parodo silpną priklausomybę tarp vaizdo elementų ir naudojamų spalvų kiekio. Taip pat eksperimentas parodė, kad santykio tarp maksimalaus įterpiamų duomenų kiekio ir skaitmeniniame vaizde naudojamų spalvų kiekio vidurkis lygus 1 bitas per spalvą.

Maksimalaus galimo duomenų kiekio įterpimo atžvilgiu, LSB algoritmas yra pranašesnis nes įterpiamų duomenų kiekis tiesiogiai priklauso nuo vaizdo pikselių kiekio. Slepiant vienodą duomenų kiekį LSB ir siūlomu algoritmu santykis tarp modifikuotų pikselių kiekio ir nemodifikuotų pikselių kiekio, naudojant LSB algoritmą, yra mažesnis (*žr* 4.1 pav.). Kas atsispindi vaizdo histogramoje - iškraipymai daugumoje atvejų yra labai nežymus (*žr* 4.3 pav.). Naudojant originalų algoritmą, vienos vaizdo paletės spalvos pakitimas įtakoja visus vaizde esančius šios spalvos pikselius, todėl lyginant pradinio (*žr* 4.2 pav.) ir modifikuoto (*žr* 4.5 pav.) vaizdų histogramas iškraipymai yra labiau pastebimi. Taip pat pastebėta, kad šiuos iškraipymus mažai įtakoja pasirinktas maksimalus kubų briaunos ilgis (*žr* 4.6 pav.), naudojamas paletės pakeitimo steganografijos algoritme, kokybės bei maksimalaus įterpiamų duomenų kiekio koregavimui, o iškraipymų pobūdis vizualiai panašus į atsitiktinį triukšmą. Duomenų kiekis, kurį galima įterpti į skaitmeninį vaizdą pasirenkant mažiausią ir didžiausią kubo briaunos ilgį padidėja vidutiniškai apie 10%. LSB algoritmu paslepiant maksimalų įmanomą duomenų kiekį, histogramoje aiškiai matosi iškraipymai būdingi šiam algoritmui - taip vadinamos “šukos” (*žr* 4.4 pav.).

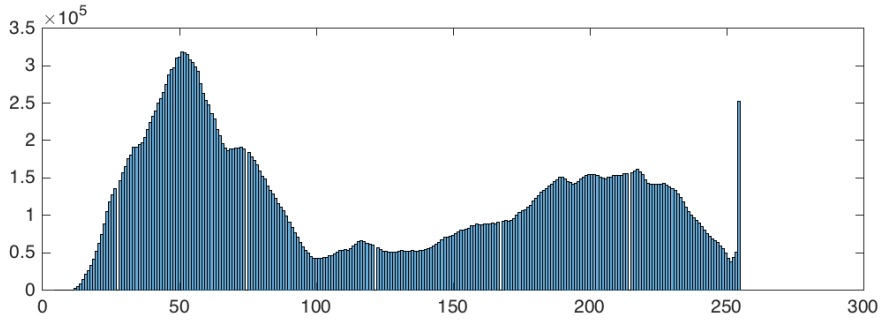


a) b)

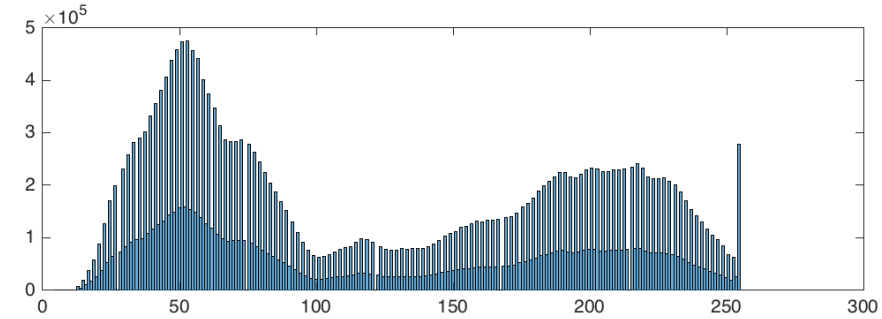
4.1 pav. *LSB(a) ir paletės pakeitimo(b) algoritmais modifikuoto vaizdo raudonos spalvinės komponentės jauniausių bitų plokštumu fragmentai*



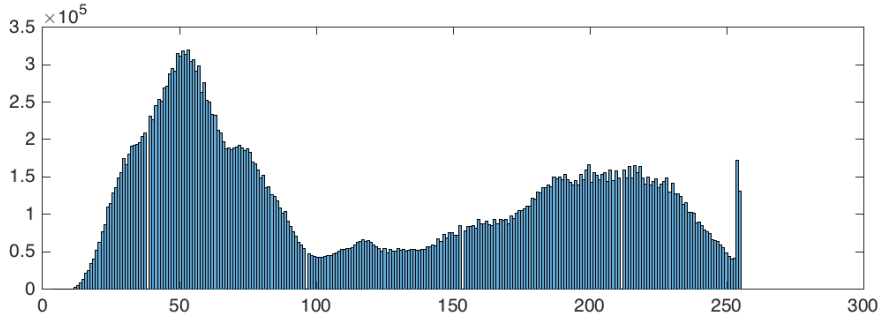
4.2 pav. *Didelės raiškos nemodifikuoto vaizdo histograma*



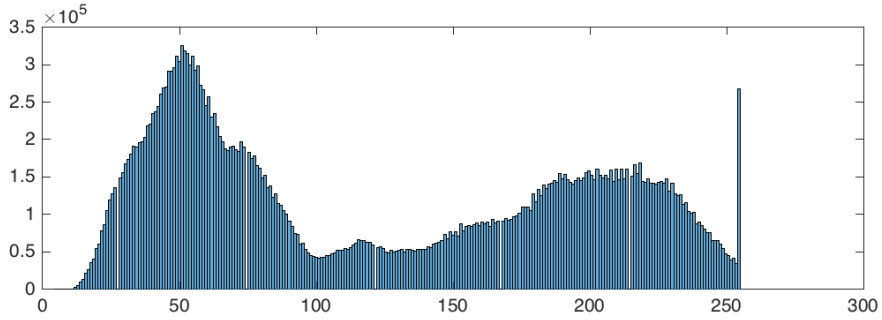
4.3 pav. *Vaizdo, į kurį LSB metodu įterpta 110904 bitų, histograma*



4.4 pav. *Vaizdo, į kurį LSB metodu įterptas didžiausias įmanomas duomenų kiekis, histograma*



4.5 pav. *Vaizdo, į kurį paletės pakeitimo steganografijos algoritmu įterptas didžiausias galimas duomenų kiekis, naudojant 2x2x2 kubus (110904 bitai), histograma*



4.6 pav. *Vaizdo, į kurį paletės pakeitimo algoritmu įterptas didžiausias galimas duomenų kiekis*

Lyginant pradinį skaitmeninį vaizdą su modifikuotais vaizdais, skirtumai plika akimi yra nepastebimi, net ir įterpiant maksimalų duomenų kiekį, kuri leidžia įterpti naudojami steganografijos algoritmai (*žr* 4.7 pav.). Šiuo atžvilgiu, priekaištu algoritmams nėra.

a) b) c)

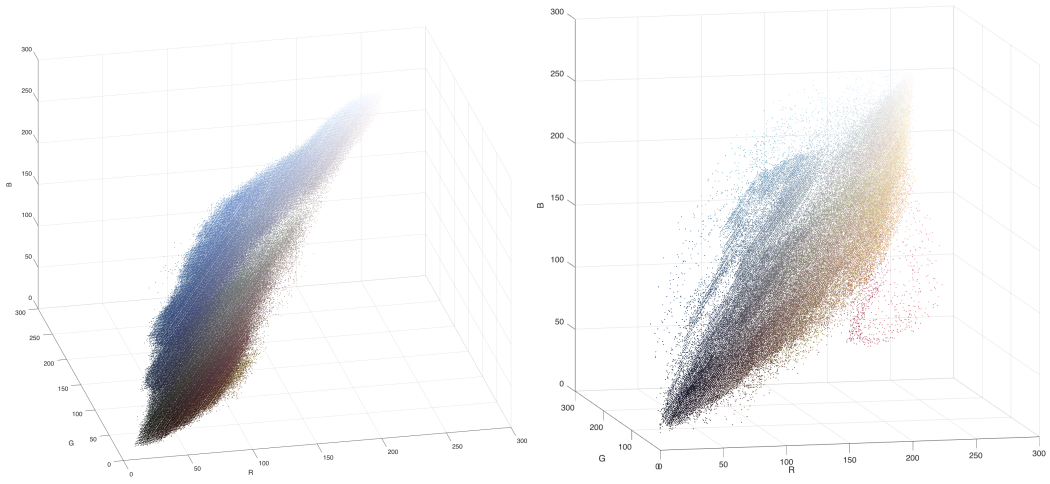
4.7 pav. *Didelės raiškos nemodifikuoto(a), LSB(b) ir siūlomo(c) algoritmais modifikuotų vaizdų padidinti fragmentai*

Steganografijos tikslas yra paslėpti duomenis, ir remiantis gautais duomenimis, galima teigti kad slepiant sulyginamą kiekį duomenų didelės raiškos skaitmeniniuose vaizduose efektyviau užduotį atlieka LSB algoritmas. LSB algoritmas, esant vienodam įterpiamų duomenų kiekiui ir vienodai modifikuotų vaizdų kokybei, daro mažesnę įtaką histogramai.

4. 2. Steganografija mažos raiškos 24 bitų spalvų gylio skaitmeniniuose vaizduose

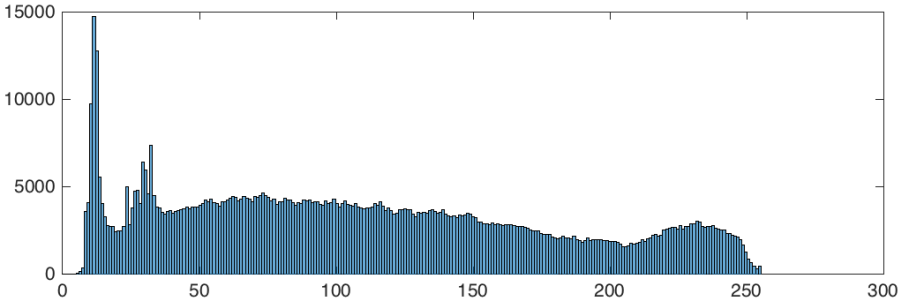
Eksperimento su mažos raiškos vaizdais metu pasitvirtino teiginys apie paletės pakeitimo steganografijos algoritmo didesnį efektyvumą, taikant jį mažesnės raiškos vaizdams. Efektyvumas vidutiniškai siekia 0,47 bito vienam vaizdo pikseliui. Taip pat padidėjo ir santykis tarp maksimalaus įterpiamų duomenų kiekio ir vaizde naudojamų spalvų kiekio, kurio vidurkis yra apie 1,66 bito spalvai. Tai gali reikšti, kad naudojamos mažos raiškos vaizduose spalvos labiau išsibarstę po RGB kubą, o didelės raiškos vaizduose spalvos išsidėsčiusios tankiau, ką ir parodo (*žr* 4.8 pav.). Maksimalaus įmanomo kubo briaunos ilgio naudojimas palyginus su atveju, kai naudojami tik kubai su briaunos ilgiu 2, padidina įterpiamų duomenų kieki vidutiniškai 19%.

Vizuali mažos raiškos skaitmeninių vaizdų histogramų analizė parodė, kad nemodifikuoto vaizdo histograma (*žr* 4.9 pav.) nėra tokia glotni, kaip didelės raiškos vaizdo. Ir todėl iškraipymai, kurie atsiranda po duomenų įterpimo į skaitmeninį vaizdą, naudojant vaizdo paletės pakeitimu grįstą metodą, nors ir yra tokiame pat lygyje, kaip ir didelės raiškos vaizduose (*žr* 4.12 pav.), tačiau yra mažiau pastebimi. Kaip ir didelės raiškos atveju, pasirinktas maksimalus kubų briaunos ilgis mažai įtakoja histogramos iškraipymus. Vaizdo, į kurį LSB metodu įterptas toks pat duomenų kiekis, kaip ir naudojant siūlomą algoritmą, histogramoje LSB metodui būdingi iškraipymai yra aiškiai matomi ir labai pastebimi, nors ir yra nedideli (*žr* 4.10 pav.). Įterpiant maksimalų duomenų kiekį, kurį leidžia paslėpti LSB algoritmas iškraipymai žymiai padidėja (*žr* 4.11 pav.).

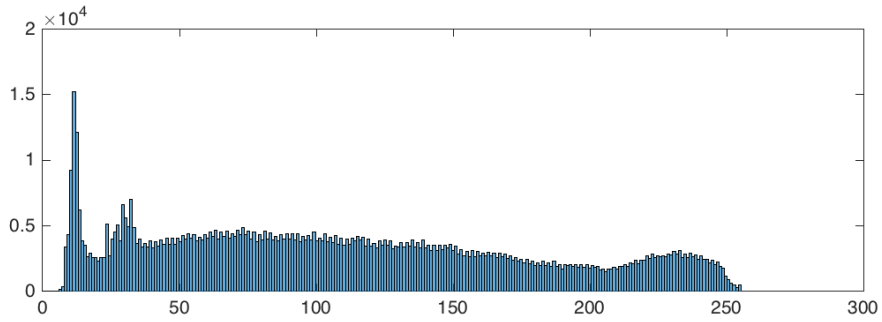


a) b)

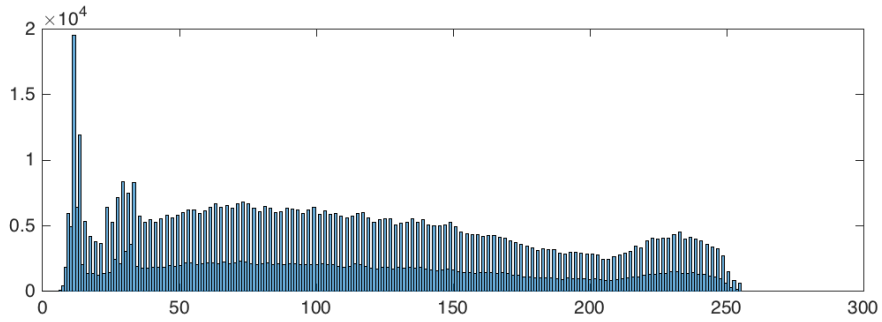
4.8 pav. *Spalvų pasiskirstymas RGB kube:didelės raiškos vaizde(a) ir mažos raiškos vaizde(b)*



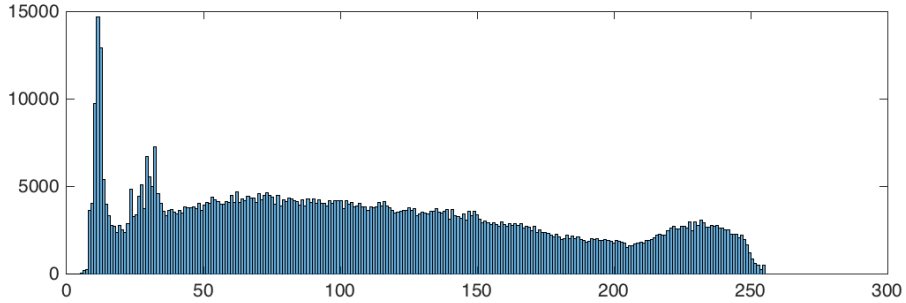
4.9 pav. *Mažos raiškos nemodifikuoto vaizdo histograma*



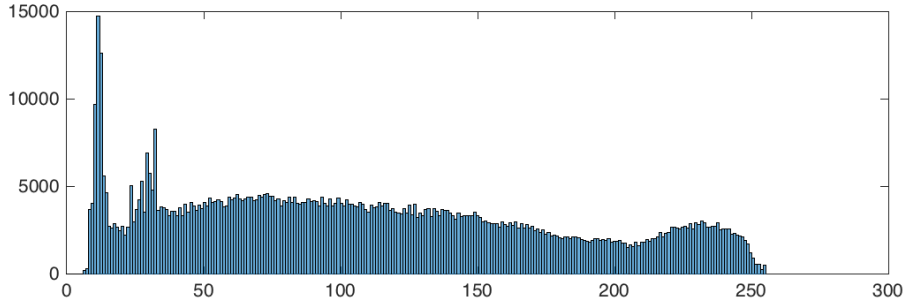
4.10 pav. *Mažos raiškos vaizdo, į kurį įterpta LSB metodu 86862 bitai, histograma*



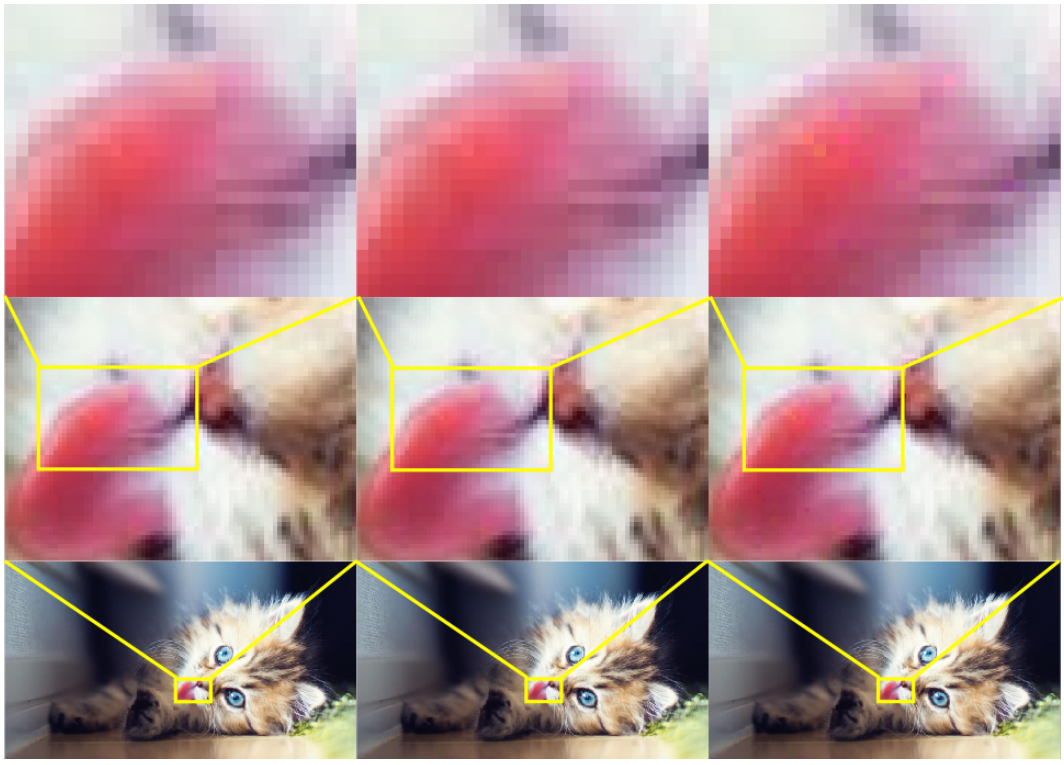
4.11 pav. *Mažos raiškos vaizdo, į kurį LSB metodu įterptas didžiausias įmanomas duomenų kiekis, histograma*



4.12 pav. *Mažos raiškos vaizdo, į kurį paletės pakeitimo algoritmu įterptas didžiausias galimas duomenų kiekis, naudojant 2x2x2 kubus (86862 bitai), histograma*



4.13 pav. *Mažos raiškos vaizdo, į kurį paletės pakeitimo algoritmu įterptas didžiausias galimas duomenų kiekis, histograma*

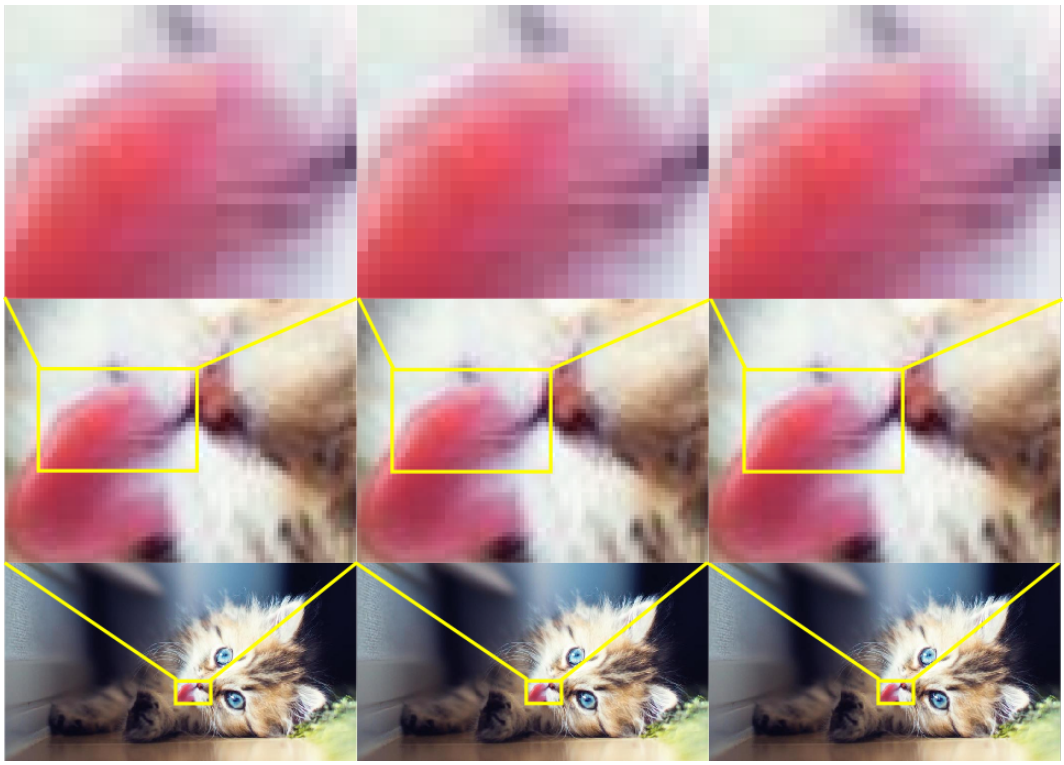


a) b) c)

4.14 pav. *Mažos raiškos nemodifikuoto(a),LSB(b) ir paletės pakeitimo(c) algoritmais modifikuotų, įterpiant didžiausią įmanomą duomenų kiekį, vaizdų padidinti fragmentai*

Lyginant nemodifikuotą vaizdą su vaizdais į kuriuos įterptas maksimalus duomenų kiekis naudojant steganografijos algoritmus (*žr* 4.14 pav.) matyti, kad LSB algoritmo atveju pakitimai nepastebimi, o paletės keitimo algoritmo atveju atsirado nežymus iškraipymai. Jeigu pažiūrėti į skaitmeninio vaizdo spalvų pasiskirstimą RGB kube (*žr* 4.8 pav.), tai galima pastebėti, kad raudoni atspalviai yra nutolę vienas nuo kito. Dėl šios priežasties raudoni atspalviai, taikant paletės keitimo

steganografijos algoritmą, patenka į didesnius kubus, kas lemia didesnį jų nukrypimą nuo originalių atspalvių, kas ir matosi padidintame vaizdo fragmente. Apribojus kubo briaunos ilgį, galima sumažinti atsirandančius iškraipymus (*žr* 4.15 pav.).



a) b) c)

4.15 pav. *Mažos raiškos nemodifikuoto(a),LSB(b) ir paletės pakeitimo(c) algoritmais modifikuotų, įterpiant duomenų kiekį, kurį įmanomą paslėpti paletės pakeitimo algoritmu, kai didžiausias kubo briaunos ilgis yra 4, vaizdų padidinti fragmentai*

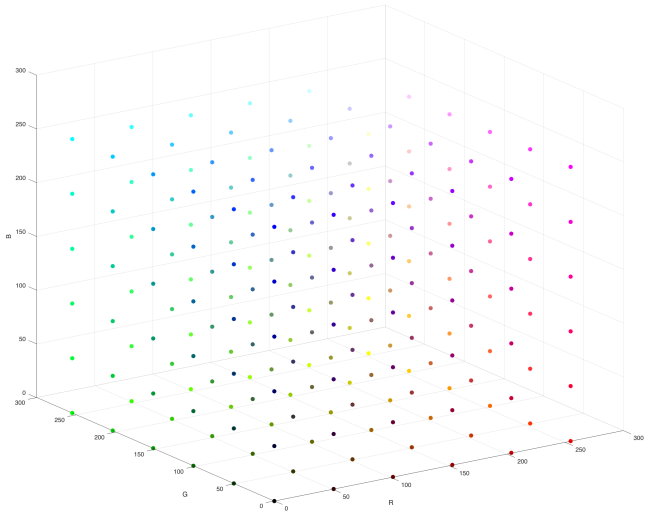
Remiantis gautais duomenimis, galima teigti, kad siūlomas paletės keitimo steganografijos algoritmas gali efektyviau paslėpti duomenis lyginant su LSB algoritmu, taikant juos mažos raiškos 24 bitų spalvų gylio vaizdams. Esant vienodam slepiamų duomenų kiekiui, nors vaizdo kokybe ir išlieka viename lygyje, LSB algoritmą išduoda jam būdingos lengvai atpažystamos “šukos” histogramoje, kai siūlomo algoritmo histogramos iškraipymai vizualiai atrodo natūralus ir gali nekelti jokio įtarimo.

4. 3. Steganografija vaizduose su palete

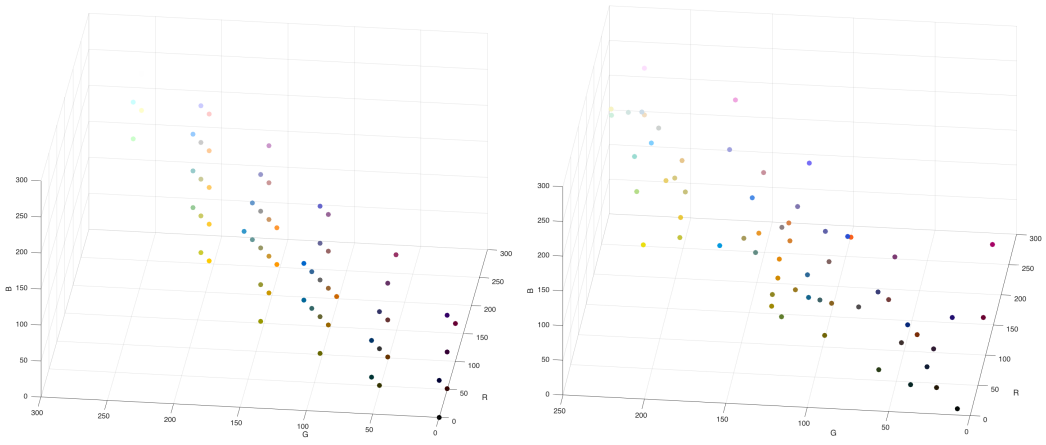
4. 3. 1. Vaizdai naudojantys standartinę paletę

Pirma eksperimentas buvo atliktas su vaizdais, kurie standartinės paletės spalvas. Standartinė paletė sudaryta taip, kad spalvos, kurių yra 216, būtų pasiskirsčiusios tolygiai visame RGB kube (*žr* 4.16 pav.), kas yra artima idealioms sąlygoms, prie kurių įmanoma įterpti į vaizdą maksimaliai daug duomenų, naudojant paletės pakeitimo steganografijos algoritmą. Bet realūs vaizdai dažniausiai išnaudoja tik dalį šių spalvų (*žr* 4.17 pav.). Eksperimente dalyvavusių vaizdų su standartinę paletę naudojamų spalvų kiekio vidurkis yra 76 spalvos. Tai lemia, kad duomenų kiekis, kurį galima įterpti naudojant siūlomą steganografijos algoritmą yra nedidelis ir svyruoja vidutiniškai nuo apie 230 bitų, kai naudojami kubai su briaunos ilgiu 2, iki 1198 bitų, kai naudojami didžiausi įmanomi kubu briaunos ilgiai bet santykis tarp vaizde naudojamų spalvų kiekio ir duomenų kiekio, kurį galima įterpti į vaizdą, siekia 3 bitus per spalvą, kai naudojami 2x2x2 kubai ir 15,5 bitų, kai naudojami didžiausi įmanomi kubai. Tokios šio santykio reikšmės parodo, kad daugumos spalvų reikšmės gali kisti dideliame diapazone, kas žymiai įtakoja vaizdo kokybę. Didžioji dalis spalvų patenka į 32x32x32 kubus.

Vizualiai analizuojant vaizdus su palete, sugeneruotus nenaudojant glotninimo, į kuriuos buvo įterptas pranešimas, naudojant paletės pakeitimo steganografijos algoritmą (*žr* 4.18 pav.), pastebimi skirtumai tarp originalaus ir modifikuoto vaizdo atsiranda tada, kai didžiausias kubų briaunos ilgis didesnis arba lygus 16. Atsirandantys iškraipymai pasižymi tuo, kad pasikeičia lokalus vaizdo kontrastas tuose vietose kur yra staigūs spalvų perėjimai, o didesni plotai užpildyti viena spalva įgauna kitokių atspalvių. Bet turint omenyje tai, kad nemodifikuotas vaizdas dėl riboto spalvų skaičiaus jau gali būti iškraipytas ir tikrosios spalvos nėra žinomos, iškraipymai nėra tokie akivaizdūs. Kadangi pakeista paletė jau nebeatitinka standartinę (*žr* 4.17 pav.), ją reikia išsaugoti kartu su indeksais, kas, be įtakos failo dydžiui, turėtų reikšti, kad vaizde naudojama adaptuota paletė, o tai jau turėtų reikšti, kad vaizdo spalvų perteikimas turi būti tikslesnis ir atitinkamai vaizde turėtų būti daugiau detalių. Taip tokios paletės, kuri nepagerina vaizdo kokybės, naudojimas gali sukelti įtarimą.

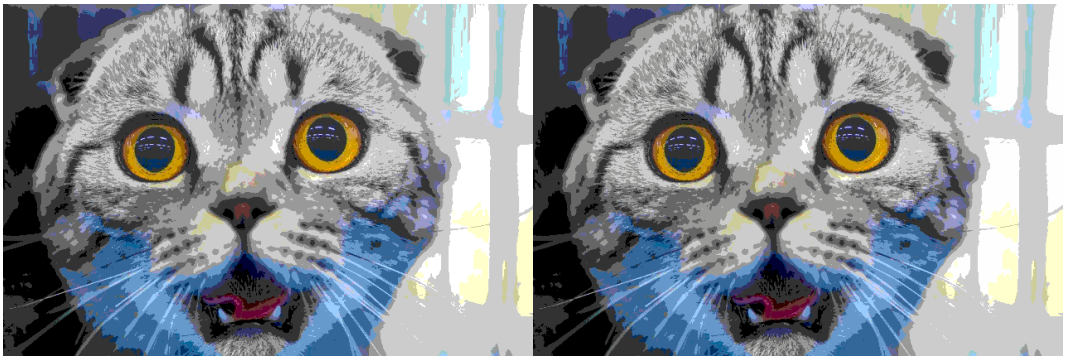


4.16 pav. *Standartinės paletės spalvų pasiskirstymas RGB kube*

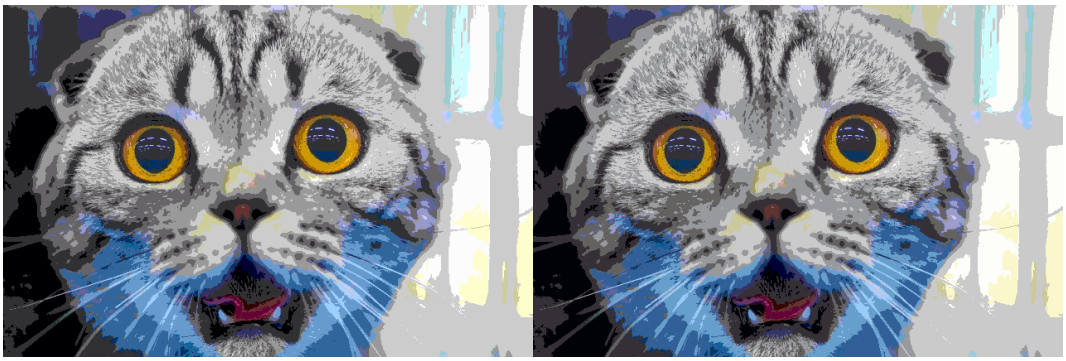


a) b)

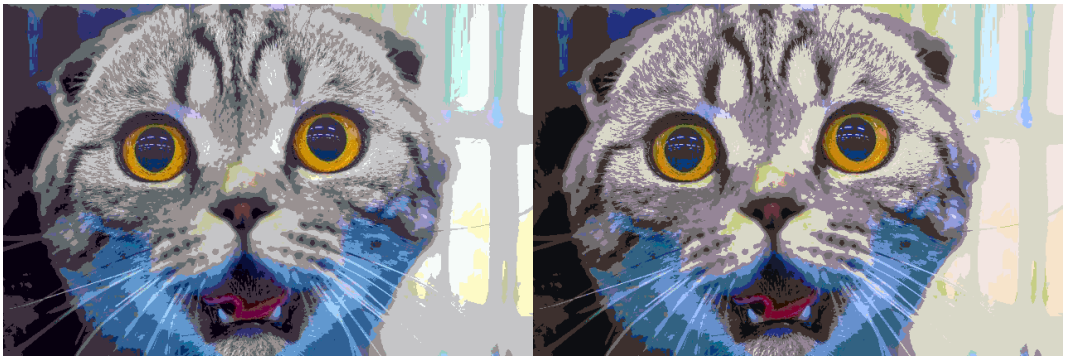
4.17 pav. *Standartinės paletės dalis naudojama realiame vaizde(a), pakeistos paletes spalvų išsidėstymas RGB kube(b)*



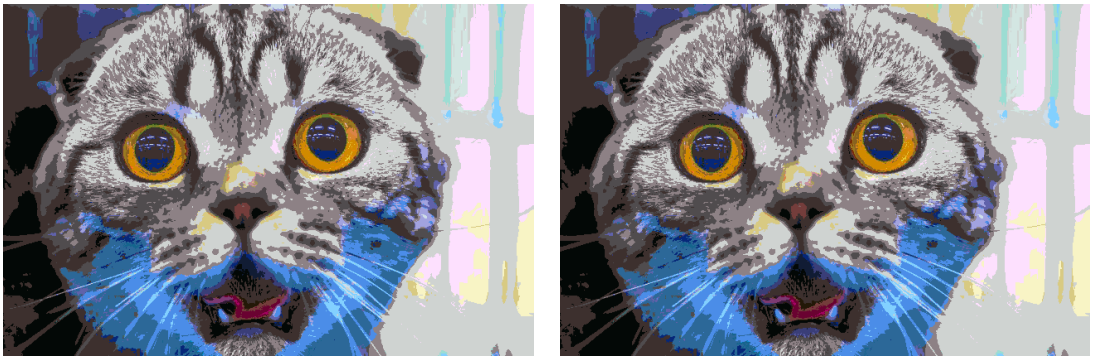
a) b)



c) d)



e) f)

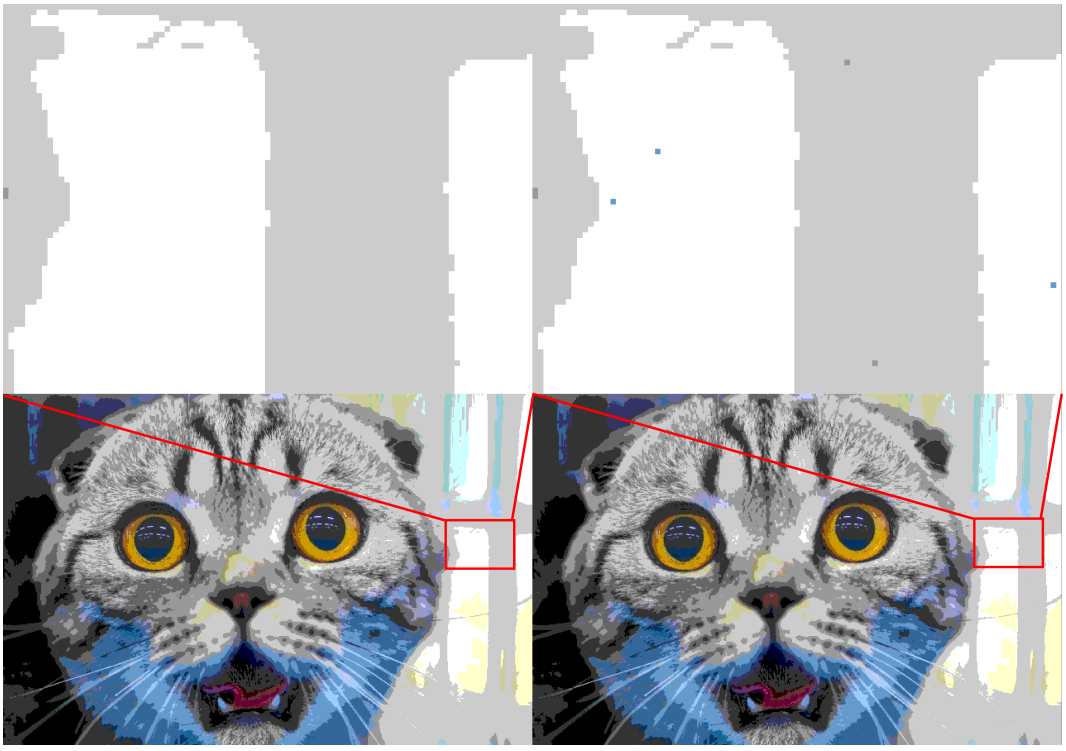


g) h)

4.18 pav. *Skaitmeninio vaizdo su standartine spalvų palete pokyčiai, taikant spalvų paletės pakeitimo steganografijos algoritmą, didėjant kubo briaunos ilgiui: a) nemodifikuotas vaizdas; b) įterpti 183 bitai, kai kubo briaunos ilgis 2; c) įterpti 366 bitai, kai kubo briaunos ilgis 4; d) įterpti 549 bitai, kai kubo briaunos ilgis 8; e) įterpti 732 bitai, kai kubo briaunos ilgis 16; f) įterpta 915 bitų, kai kubo briaunos ilgis 32; g) įterpti 975 bitai, kai kubo briaunos ilgis 64; h) įterpti 978 bitai, kai kubo briaunos ilgis 128*

Slepiant duomenis LSB metodu vaizdų su paletę indeksuose, vaizde atsiranda būdingi šiam atvejui iškraipymai, panašus į “salt & pepper” triukšmą, ir yra gerai matomi net ir įterpiant nedidelio kiekio duomenų. Tai atsitinka dėl to, kad paletėje viena po kitos einančių spalvų reikšmės gali labai skirtis. Ypač gerai pastebimi šie iškraipymai monotoniškai nuspalvintuose vaizdo plotuose (*žr* 4.19 pav.). Didėjant įterpiamam duomenų kiekiui, atitinkamai didėja ir iškraipymai.

Taikyti LSB metodą paletės reikšmėms, šiuo atveju, yra, iš esmės, tas pats, kaip ir naudoti siūlomą paletės pakeitimo steganografijos algoritmą, kai kubo briaunos ilgis 2. Tokie pat yra privalumai ir trūkumai - spalvos pasikeičia labai nežymiai ir pakeitimai neįžiūrimi plika akimi, bet pakitusi paletė neatitinka standartinės.

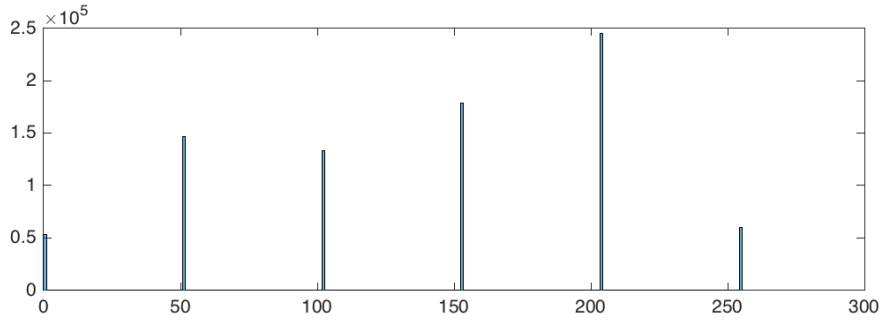


a) b)

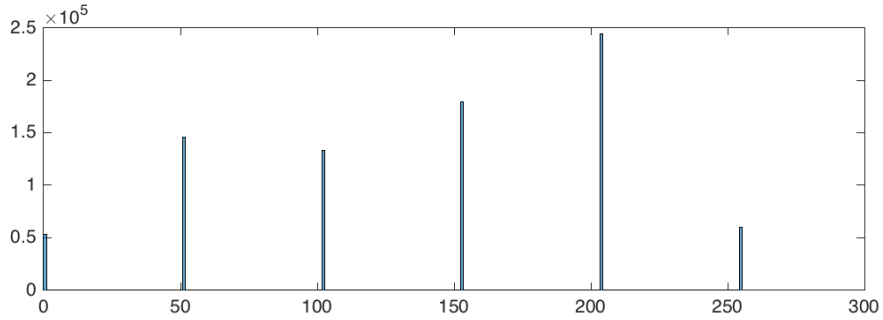
4.19 pav. *Nemodifikuotas vaizdas ir padidintas jo fragmentas(a) ir vaizdas ir jo padidintas fragmentas, į kurį įterpti duomenys naudojant LSB algoritmą(b)*

Nagrinėjant vaizdų histogramas galima pamatyti, kad nemodifikuotame vaizde(*žr* 4.20 pav.) ir vaizduose, kurių indeksams taikytas LSB metodas, reikšmės susikoncentravo ties šešiais stulpeliais, kas atitinka standartinės paletės spalvų išdėstymą. Modifikuotų vaizdų histogramos, i kuriuos įterptas nedidelis duomenų kiekis (*žr* 4.21 pav.), beveik nesiskiria nuo nemodifikuoto vaizdo histogramos. Įterpiant LSB algoritmų maksimalų galimą duomenų kiekį, histograma pakinta, bet visos reikšmės susikoncentravusios ties tais pačiais stulpeliais, tik kitaip pasiskirsčiusios. Tai yra nuspėjama, kadangi taikant LSB algoritmą vaizdo indeksams, vaizdo paletė nesikeičia, o keičiasi tik indeksų reikšmės.

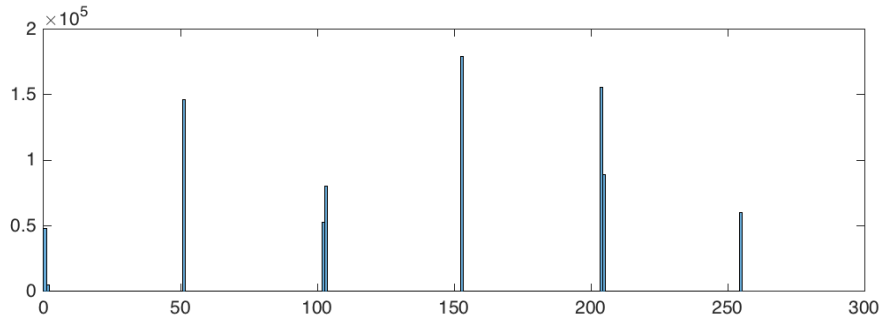
Histogramos vaizdų, į kuriuos buvo įterpti duomenis naudojant paletės pakeitimo algoritmą , bei LSB algoritmą taikant jį vaizdo paletei, atrodo skirtingai. Spalvų reikšmės nukrypsta nuo standartinės paletės spalvų reikšmių, kas ir matosi histogramose (*žr* 4.22 pav. ir 4.23 pav). Tuo atveju, kai naudojamas LSB algoritmas vaizdo paletei ir siūlomas originalus algoritmas su kubų briaunos ilgiu 2, reikšmės histogramose nukrypsta per 1, tik skiriasi reikšmių tarpusavio pasiskirstymas dėl algoritmų skirtingo palėtės spalvų apdorojimo eiliškumo. Taikant originalų algoritmą su didesniais kubų ilgiais, spalvų reikšmės vis toliau nukrypsta nuo standartinių spalvų, ką ir parodo histograma (*žr* 4.24 pav.)ir spalvų pasiskirstymas RGB kube (*žr* 4.17 pav.).



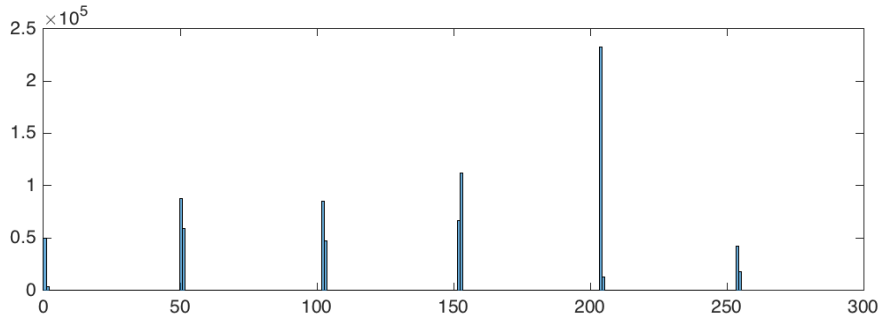
4.20 pav. *Nemodifikuoto vaizdo su standartine palete histograma*



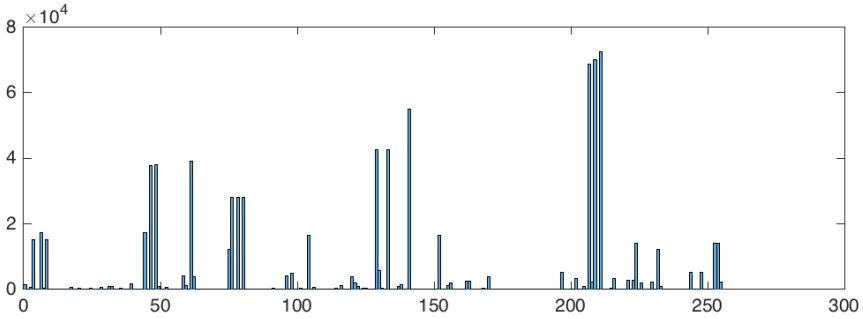
4.21 pav. *Vaizdo su standartine palete, į kurio indeksus LSB metodu įterpti 978 bitai, histograma*



4.22 pav. *Vaizdo naudojusio standartinę paletę, į kurį įterpti duomenys taikant LSB algoritmą paletei, histograma*



4.23 pav. *Vaizdo naudojusio standartinę paletę, į kurį įterpi duomenys, naudojant paletės pakeitimo steganografijos algoritmą, kai naudojami kubai su briaunos ilgiu 2, histograma*

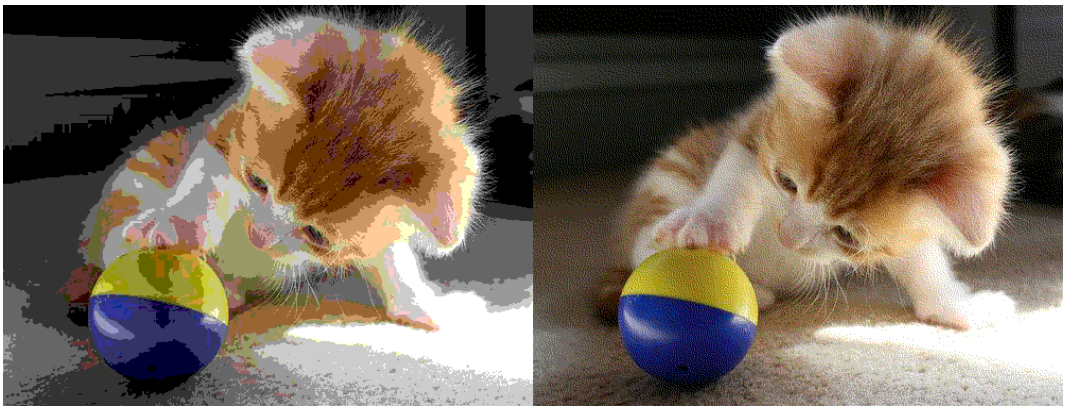


4.24 pav. *Vaizdo naudojusio standartinę paletę, į kurį įterpi duomenys, naudojant paletės pakeitimo steganografijos algoritmą, kai naudojami kubai su briaunos ilgiu 128, histograma*

Vaizdų su standartine palete be glotninimo atveju, abu lyginami algoritmai rodo blogus rezultatus slepiant duomenis. LSB algoritmas vaizde sukuria atpažįstamą labai matomą triukšmą, o paletės pakeitimo algoritmas pakeičia paletę į adaptyvią su būdingais iškraipymais, kas matosi histogramoje, be matomo vaizdo kokybės pagerėjimo.

4. 3. 2. Vaizdai naudojantys standartinę paletę ir glotninimą

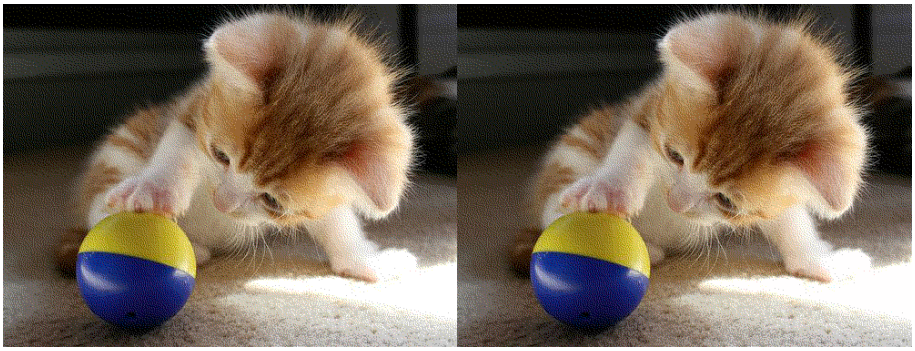
Eksperimentuojant su vaizdais, kurie naudoja standartinę spalvų paletę ir kuriuos sudarant buvo atliekamas glotninimas, pastebėta, kad nemodifikuoti vaizdai (*žr* 4.25 pav.) vizualiai atrodo geriau, ir glotninimas tikrai pagerina vaizdo kokybę. Taip pat vaizduose padidėjo standartinės paletės spalvų išnaudojimas, kas teigiamai veikia duomenų įterpimo galimybes, naudojant paletės pakeitimo steganografijos algoritmą bei LSB algoritmą vaizdo paletei, kuris kaip ir anksčiau atitinka paletės pakeitimo algoritmui su kubu briaunos ilgiu 2. Vaizduose su standartine palete ir glotninimu naudojama vidutiniškai 28% daugiau spalvų, negu vaizduose su standartine palete be glotninimo.



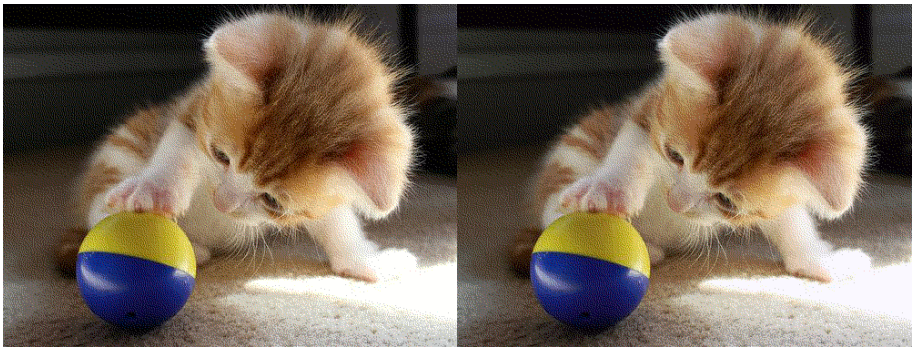
a) b)

4.25 pav. *Vaizdas naudojant standartinę paletę be glotninimo (a) ir su glotninimu (b)*

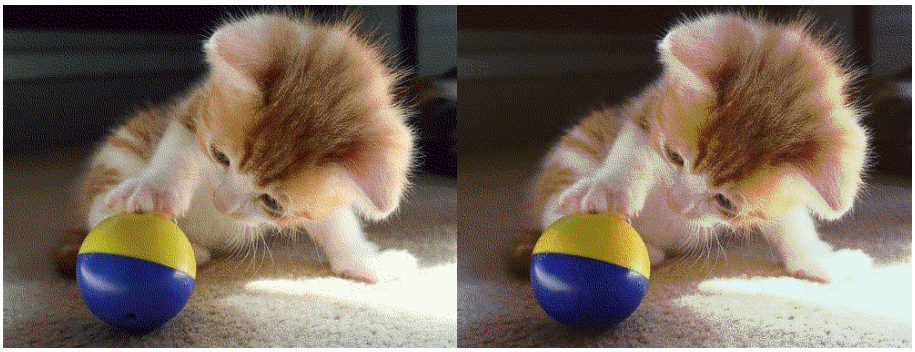
Glotninimas taip pat teigiamai veikia ir vaizdo, į kuri paletes pakeitimo steganografijos algoritmu yra įterpti duomenis, kokybę. Kadangi tokiose vaizduose retai pasitaiko dideli vienos spalvos plotai - didžioji dalis vienas šalia kito esančių pikselių spalvų reikšmės yra skirtingos, labai pasikeitusios spalvos yra atskiedžiamos šalia esančių pikselių spalvomis ir pasikeitimai yra mažiau pastebimi. Pastebimi skirtumai tarp nemodifikuoto vaizdo ir vaizdo su įterptais duomenimis atsiranda, kai paletes pakeitimo steganografijos algoritmas naudoja kubu briaunos ilgį 32 ir didesnius, bet atskirai paimti šie vaizdai atrodo gana natūraliai (*žr* 4.26 pav.).



a) b)



c) d)



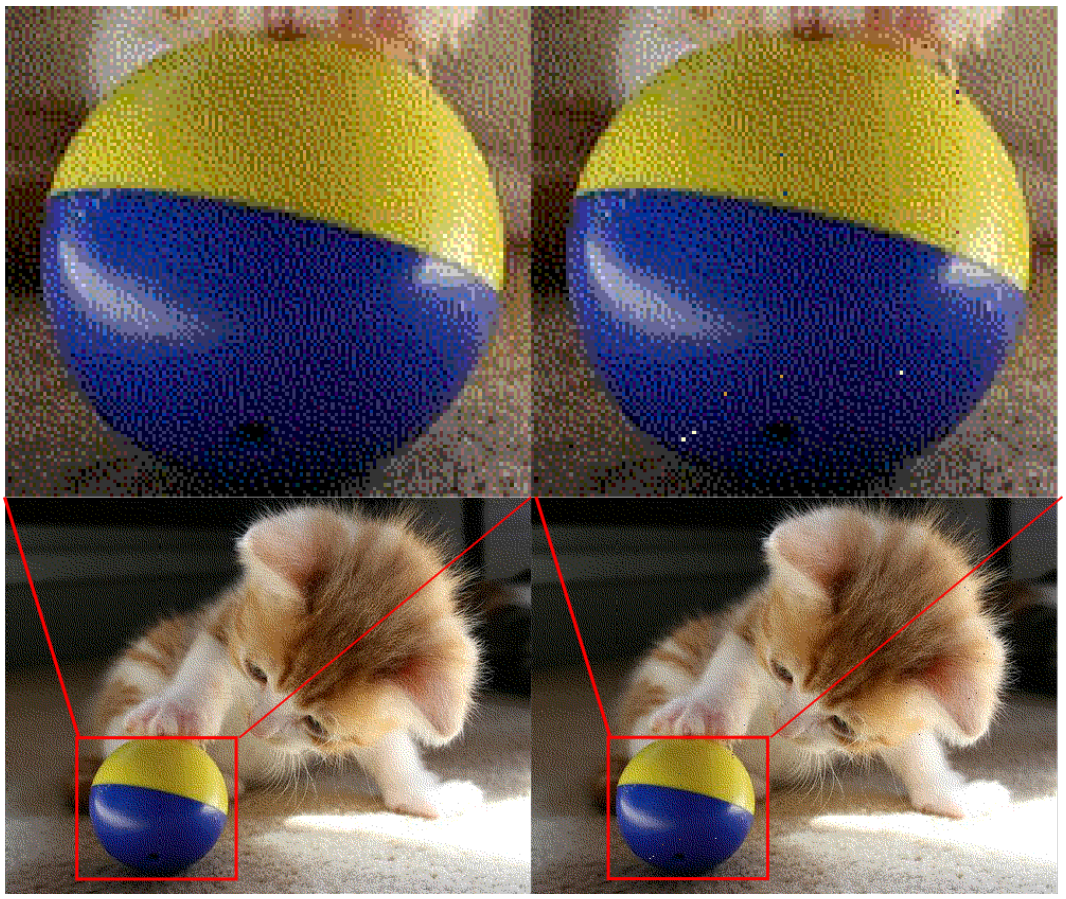
e) f)

g) h)

4.26 pav. *Skaitmeninio vaizdo su standartine spalvų palete ir glotninimu pokyčiai, taikant spalvų paletės pakeitimo steganografijos algoritmą: a) nemodifikuotas vaizdas; b) įterpti 192 bitai, kai kubo briaunos ilgis 2; c) įterpti 384 bitai, kai kubo briaunos ilgis 4; d) įterpti 576 bitai, kai kubo briaunos ilgis 8; e) įterpti 768 bitai, kai kubo briaunos ilgis 16; f) įterpta 960 bitų, kai kubo briaunos ilgis 32; g) įterpta 1017 bitų, kai kubo briaunos ilgis 64; h) įterpta 1020 bitų, kai kubo briaunos ilgis 128*

Taikant LSB steganografijos metodą skaitmeniniams vaizdams su standartinę paletę ir glotninimu, vaizde atsiranda toks pat triukšmas, kaip ir vaizduose be glotninimo (*žr* 4.27 pav.). T.y. jeigu labai skirtingos spalvos paletėje eina viena po kitos, tai keičiant jauniausią indekso bitą gali atsirasti tokių situacijų, kai vienas pikselis labai išsiskiria supančių jį pikselių fone. Glotninimas iš dalies gali užmaskuoti LSB triukšmą, kai skirtumai tarp pakeičiamų indeksų nedideli, bet atsiranda ir labai matomų elementų.

Vaizdų su standartine spalvų palete histogramoms glotninimas nedaro jokios įtakos. Todėl glotnintiems skaitmeniniams vaizdams su standartine spalvų palete galioja tie patys dėsniai kaip ir skaitmeniniams vaizdams su standartine spalvų palete be glotninimo.



a) b)

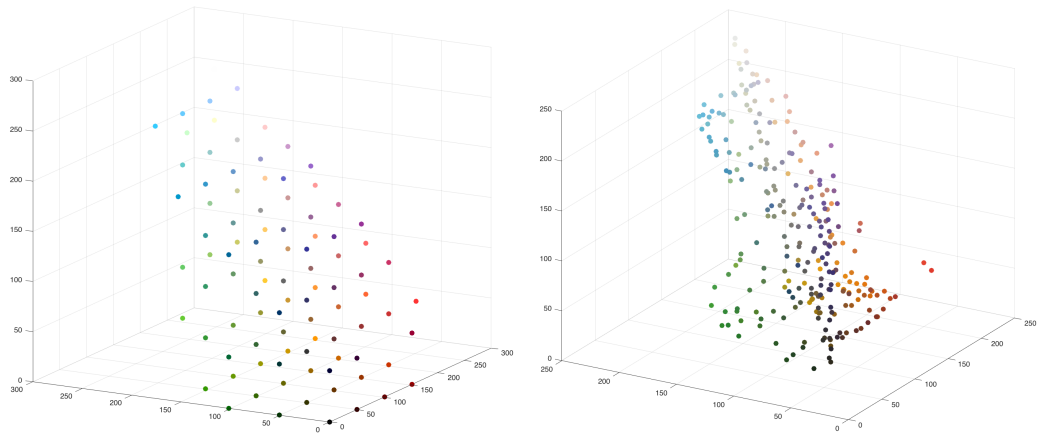
4.27 pav. *Triukšmas atsirandantis skaitmeniniame vaizde su standartine palete dėl LSB steganografijos algoritmo naudojimo. Nemodifikuotas vaizdas ir jo padidintas fragmentaa (a) ir vaizdas modifikuotas LSB metodu ir jo padidintas fragmentas (b)*

Apibendrinant, galima teigti, kad vaizdo glotninimas ne tik pagerina pačio vaizdo kokybę, bet ir leidžia labiau išsiskleisti steganografijos algoritmams. Kadangi vaizdas su glotninimu vizualiai atrodo geriau, o pritaikius paletės pakeitimo algoritmą paletė pasidaro nestandartinė, tai gali susidaryti įspūdis, kad naudojama optimizuota adaptyvi paletė. Taip pat glotninimas dalinai užmaskuoja LSB sukuriamą triukšmą.

4. 3. 3. Vaizdai naudojantys adaptyvią paletę

Kadangi standartinės spalvų paletės spalvos išdėstytos tolygiai visame RGB kube, dažniausiai tik dalis šių spalvų yra išnaudojama, o naudojamos spalvos gali žymiai skirtis nuo originalių spalvų,

kas neigiamai įtakoja vaizdo konvertuojamo iš 24 bitų spalvų gylio į vaizdo su paletę kokybę. Kokybės gerinimui gali būti naudojama adaptyvi paletė, kuri sudaroma kiekvienam vaizdui atskirai taip, kad kuo tiksliau perduoti originalaus vaizdo spalvas išnaudojant visą paletės spalvų kiekį (*žr* 4.28 pav.). Taip pat tai reiškia, kad spalvos yra pasiskirsčiusios tankiau.



a) b)

4.28 pav. *Skaitmeninio vaizdo su palete, kuris yra sugeneruotas iš 24 bitų spalvų gylio vaizdo, spalvų pasiskirstymas RGB kube naudojant standartinę paletę (a) ir naudojant adaptyvią paletę (b)*

Paletės pakeitimo steganografijos algoritmo atžvilgiu, padidėjęs naudojamų spalvų kiekis turėtų padidinti ir duomenų kiekį, kurį galima paslėpti vaizde, o labiau koncentruotas jų išdėstymas įtakoja galimą atskirų spalvų pakitimo rėžius, kas riboja slepiamų duomenų kiekį naudojant didesnius kubus.

Spalvų kiekis tiriamuosiuose vaizduose vidutiniškai siekia 255, duomenų kiekio, kurį galima įterpti į vaizdą naudojant paletės pakeitimo steganografijos algoritmą ir kubo briaunos ilgį 2, vidurkis yra 795 bitai, maksimalus duomenų kiekio, naudojant didžiausią įmanomą kubo briaunos ilgį, vidurkis 2731 bitas, kas atitinka apie 3 bitus ir 11 bitų vienai spalvai atitinkamai.

Vizualiai analizuojant vaizdus su įterptais duomenimis, taikant paletes pakeitimo steganografijos algoritmą (*žr* 4.29 pav.), pastebėta, kad nežymūs kontrasto pasikeitimai lėtų spalvų perėjimo vietose atsiranda jau naudojant kubo briaunos ilgi 4. Tai atsitinka dėl to, kad adaptyvi paletė sudaryta taip, kad gauti geresnę vaizdo kokybę. Didžioji dalis paletės spalvų išsidėsčiusi taip, kad apimtų kuo didesnį vaizde naudojamą spalvų spektrą ir kiek įmanoma lygiau vaizduoti spalvų perėjimus. Kai vienos spalvos nežymus pokytis yra nematomas, visų šalia esančių spalvų pokyčiai skirtingomis kryptimis gali susidėti ir tapti pastebimi. Didėjant kubų briaunos ilgiui žymiai mažėja spalvų kiekis, kuris patenka į didesnius kubus ir galima matyti, kad tik pavienės spalvos pakinta labai pastebimai. Nors ir pasikeitimai, kai naudojami kubai mažesni už 32x32x32, gerai matomi, lyginant su nemodifikuotu vaizdu, bet šie pakeitimai atrodo gana natūraliai žiurint į modifikuotą vaizdą atskyrai, todėl, neturint vaizdo originalo, sunku spręsti ar į vaizdą buvo įterpti duomenys.

Vizualiai nagrinėjant LSB steganografijos algoritmo taikymo indeksams veikimo rezultatus, vėl galima pastebėti jam būdingą triukšmą (*žr* 4.30 pav.). Nors šis triukšmas ir yra kai kur mažiau pastebimas dėl tankesnio adaptyvios paletės spalvų tarpusavio išsidėstymo, bet rezultatas priklauso ir nuo to, kaip paletė yra surūšiuota ir kiek naudojamoje paletėje yra labai didelių skirtumų tarp spalvų su vienu paskui kitą einančiais indeksais.

a) b)

c) d)

e) f)

g)

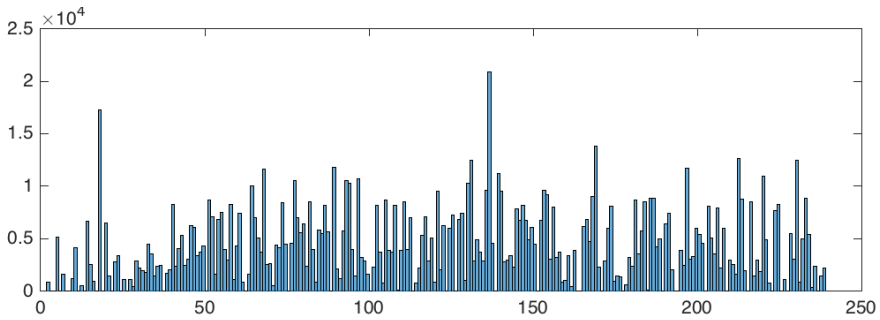
4.29 pav. *Skaitmeninio vaizdo su adaptyvia paletę pokyčiai, taikant spalvų paletės pakeitimo steganografijos algoritmą: a) nemodifikuotas vaizdas; b) įterpti 765 bitai, kai kubo briaunos ilgis 2; c) įterpta 1530 bitų, kai kubo briaunos ilgis 4; d) įterpti 2289 bitai, kai kubo briaunos ilgis 8; e) įterpti 2805 bitai, kai kubo briaunos ilgis 16; f) įterpti 2949 bitai, kai kubo briaunos ilgis 32; g) įterptas 2961 bitas, kai kubo briaunos ilgis 64*

a) b)

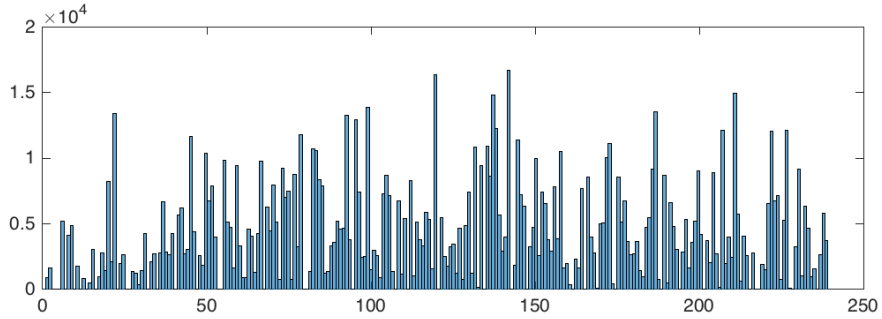
4.30 pav. *Triukšmas atsirandantis skaitmeniniame vaizde su adaptyvia palete dėl LSB steganografijos algoritmo naudojimo. Nemodifikuotas vaizdas ir jo padidintas fragmentas (a) ir vaizdas modifikuotas LSB metodu ir jo padidintas fragmentas (b)*

Kadangi tiriamuosiuose vaizduose taikant paletės pakeitimo steganografijos algoritmą visas spalvas galima sudėti į 2x2x2 kubus, tai iš esmės LSB steganografijos algoritmo taikymas paletei atitinka paletės pakeitimo algoritmo su kubų briaunos ilgiu 2, veikimui.

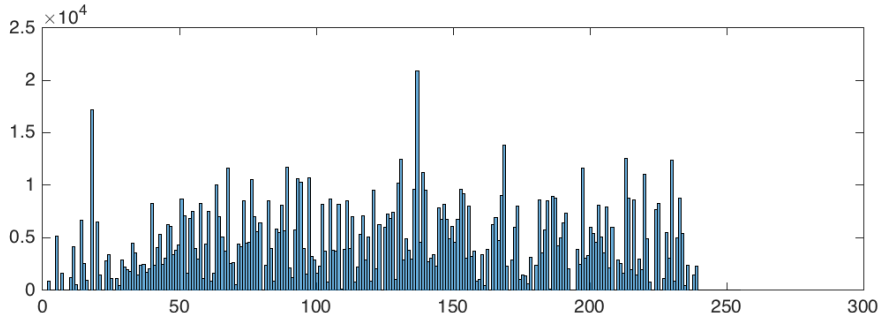
Vizualiai analizuojant histogramas, galima pastebėti, kad vaizdų su adaptyvia palete histogramos (*žr* 4.31 pav.) labiau panašesnės į 24 bitų spalvų gylio vaizdų histogramas negu vaizdu su standartinė spalvų palete histogramos. Matyti, kad paletės pakeitimo steganografijos algoritmas nors ir žymiai pakeičia histogramos vaizdą, bet histogramoje neaptikta jokių dėsningumų (*žr* 4.32 pav.) ir vizualiai histograma atrodo natūraliai. Didelius histogramos pokyčius lemia tai, kad vienos paletės spalvos pakitimas įtakoja visus šią spalvą atitinkančius pikselius. Kadangi, net ir naudojant mažus kubus, didžioji dalis spalvų linkusi kisti, tai atitinkamai pasikeičia didžioji dalis vaizdo pikselių. LSB metodu įterpiant į vaizdą duomenų kiekį atitinkanti duomenų kiekiui, kurį galima įterpti naudojant paletės pakeitimo algoritmą, beveik neįtakoja histogramos (*žr* 4.33 pav.).



4.31 pav. *Nemodifikuoto vaizdo su adaptyvia palete histograma*



4.32 pav. *Vaizdo su adaptyvia palete, į kurį įterpti 2289 bitai duomenų naudojant paletės pakeitimo steganografijos algoritmą, histograma*



4.33 pav. *Vaizdo su adaptyvia palete, į kurį įterptas 2961 bitas duomenų, taikant LSB steganografijos algoritmą indeksams, histograma*

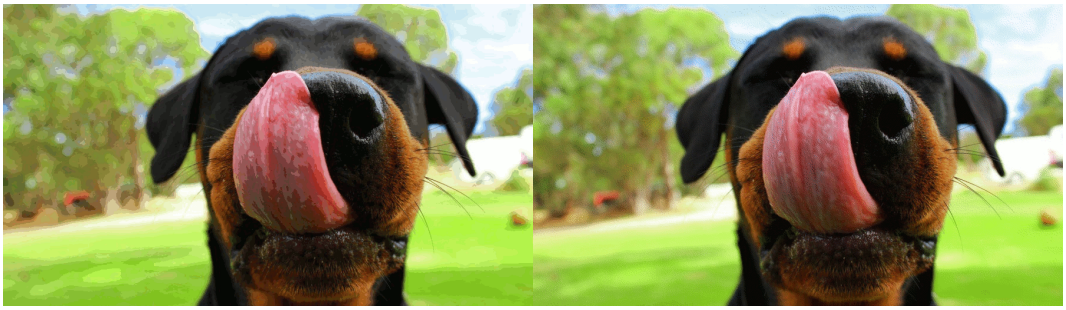
Adaptyvios paletės naudojimas ženkliai pagerina vaizdo kokybę lyginant su standartinės paletės naudojimu. Paletės pakeitimo steganografijos algoritmo atžvilgiu, spalvos gali kisti mažesniame diapazone, bet tai yra kompensuojama padidėjusiu vaizde naudojamų spalvų kiekiu. Todėl, bendru atveju, galima teigti, kad adaptyvios paletės naudojimas pagerina algoritmo efektyvumą. Taikant LSB steganografijos algoritmą vaizdams su adaptyvia paletę, jam būdingas triukšmas išlieka, bet tampa mažiau pastebimas.

4. 3. 4. Vaizdai naudojantys adaptyvią paletę ir glotninimą

Glotninimo taikymas vaizdams su adaptyvia palete, kaip ir vaizdams su standartine palete, pagerina vaizdų kokybę (*žr* 4.34 pav.) - greta esantiems pikseliams parenkant skirtingas spalvas, sudaroma optinė iliuzija, kad vaizde naudojama daugiau spalvų, nei yra iš tikrųjų. Vaizde, esant lėtiems perėjimams ir nenaudojant glotninimo, atsiranda dideli vienos spalvos plotai ir labai ryškiai matomos šių plotų ribos, naudojant glotninimą, perėjimai atrodo lygiau ir natūraliau.

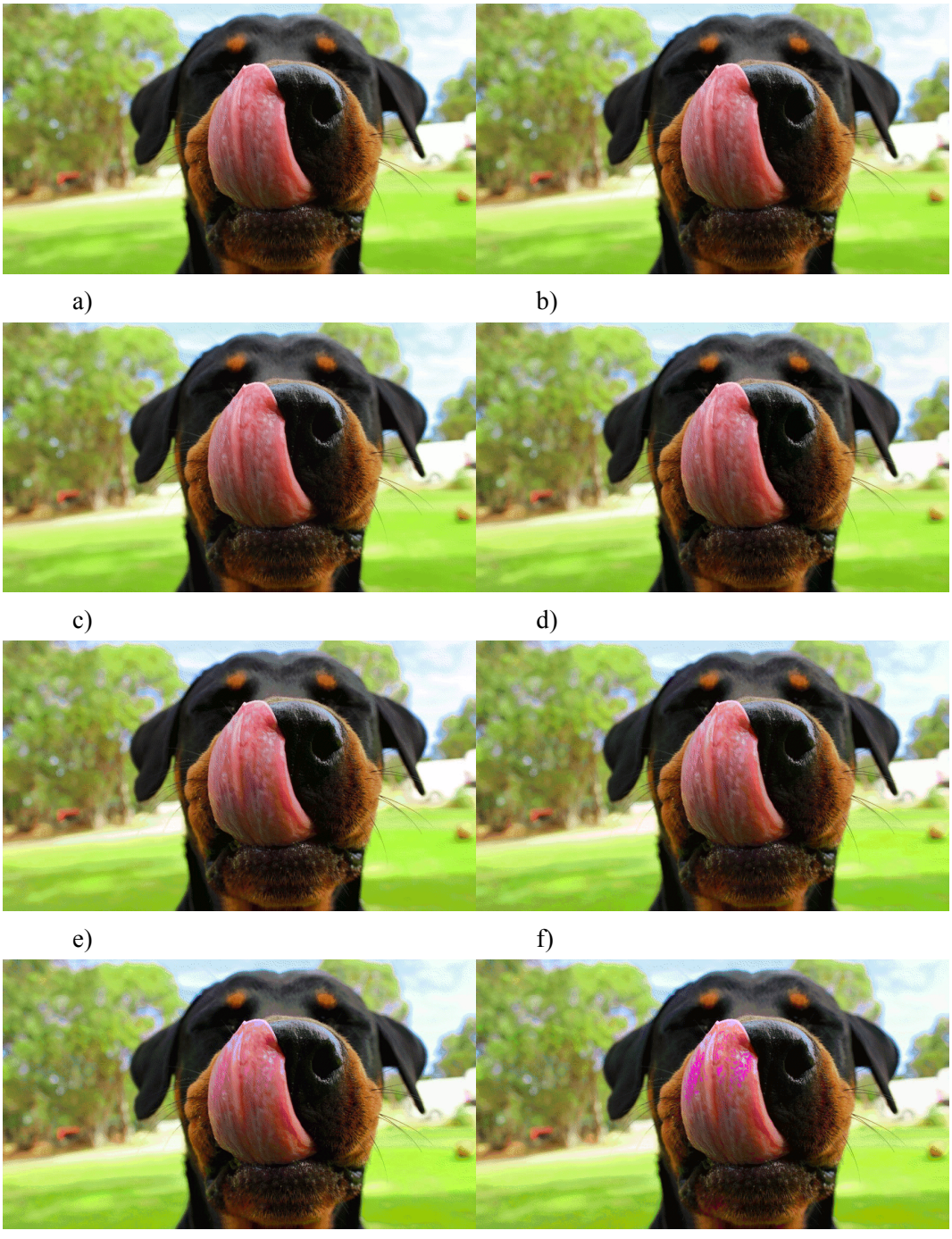
Analizuojant vaizdus su adaptyvią paletę ir glotninimų, pastebėta, kad glotninimas labai silpnai įtakoja paletėje naudojamas spalvas. Didžioji dalis palečių analizuotose vaizduose išliko nepakitusios arba jose pasikeitė tik kelios spalvos. Atitinkamai duomenų kiekiai, kuriuos galima įterpti į vaizdus su adaptyvią palete ir glotninimu naudojant paletės pakeitimo steganografijos algoritmą, nepakito, arba pakito nežymiai lyginant su vaizdais be glotninimo.

Taikant vaizdams su adaptyvią paletę ir glotninimų paletės pakeitimo steganografijos algoritmą (*žr* 4.35 pav.), pastebimi iškraipymai, tokie kaip lokalūs kontrasto pasikeitimai, atsiranda naudojant kubus su briaunos ilgiu 8. Šie iškraipymai nors ir yra pastebimi lyginant modifikuotą ir nemodifikuotą vaizdus, bet atrodo gana natūraliai. Naudojant kubo briaunos ilgi 64 ir daugiau, pavienės spalvos smarkiai degraduoja ir labai išsiskiria vaizde.



a) b)

4.34 pav. *Vaizdas su adaptyvia spalvų palete be glotninimo(a) ir su glotninimu(b)*



a) b)

c) d)

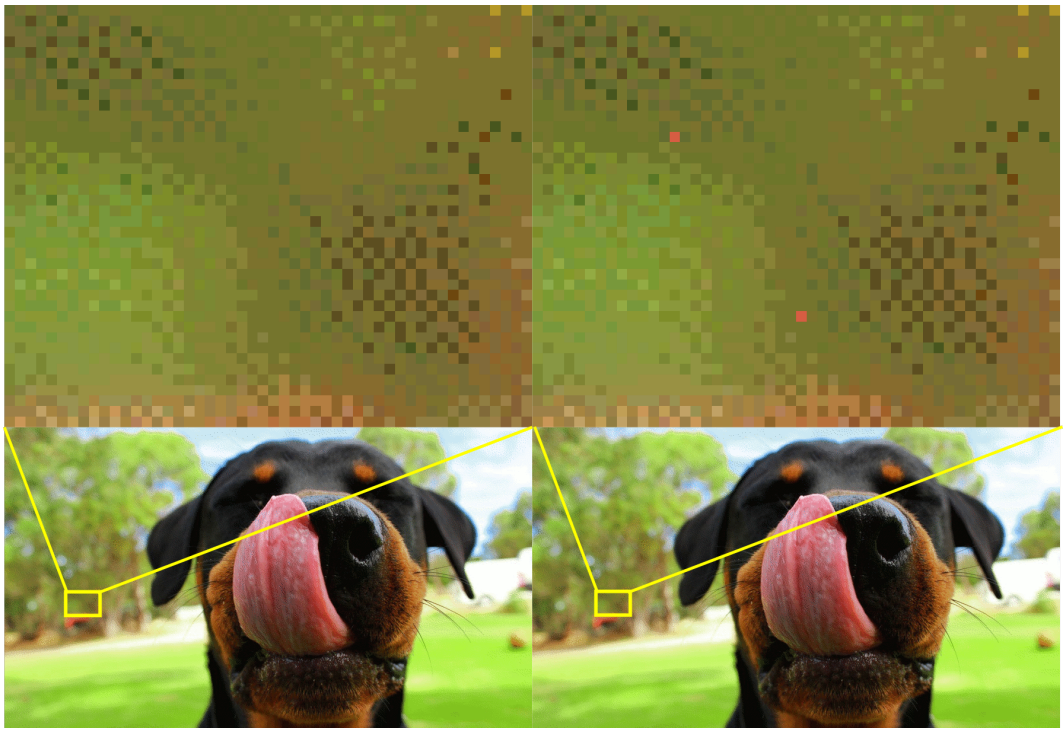
e) f)

g) h)

4.35 pav. *Skaitmeninio vaizdo su adaptyvia paletę ir glotninimu pokyčiai, taikant spalvų paletės pakeitimo steganografijos algoritmą: a) nemodifikuotas vaizdas; b) įterpti 768 bitai, kai kubo briaunos ilgis 2; c) įterpti 1536 bitai, kai kubo briaunos ilgis 4; d) įterpti 2283 bitai, kai kubo briaunos ilgis 8; e) įterpti*

2802 bitai, kai kubo briaunos ilgis 16; f) įterpta 2910 bitų, kai kubo briaunos ilgis 32; g) įterpti 2934 bitai, kai kubo briaunos ilgis 64; h) įterpta 2940 bitų, kai kubo briaunos ilgis 128

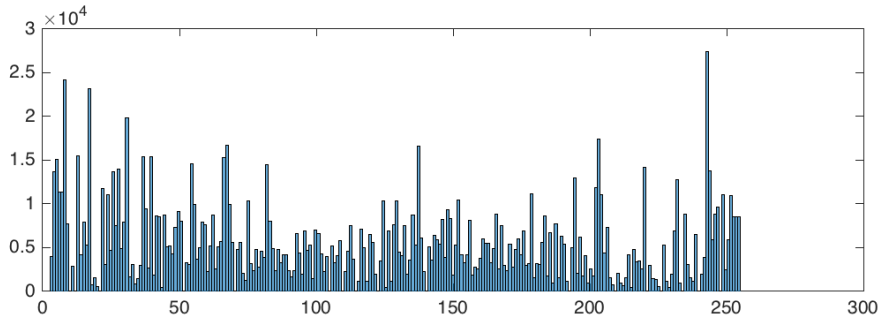
LSB steganografijos algoritmo taikymas vaizdo su adaptyvia palete ir glotninimu indeksams, kaip ir kitais atvejais, įneša būdingų iškraipymų - pavieniai pikseliai, kurių reikšmės žymiai skiriasi nuo jų supančių pikselių reikšmių (*žr* 4.36 pav.). Glotninimas šiek tiek maskuoja tokius pikselius, nes vaizde, palyginus su vaizdu be glotninimo, nėra didelių vienodos spalvos plotų. LSB metodo taikymas vaizdo paletei, atitinka paletės pakeitimo steganografijos algoritmui, kai naudojami kubai su briaunos ilgiu 2.



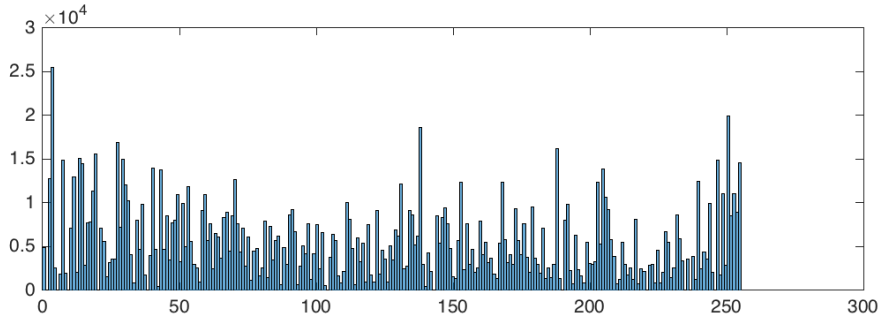
a) b)

4.36 pav. *Nemodifikuotas vaizdas su adaptyvia palete ir glotninimu ir vaizdo padidintas fragmentas(a), vaizdas su adaptyvia palete ir glotninimu, į kurį įterpti duomenys naudojant LSB steganografijos algoritmą vaizdo indeksams ir vaizdo padidintas fragmentas (b)*

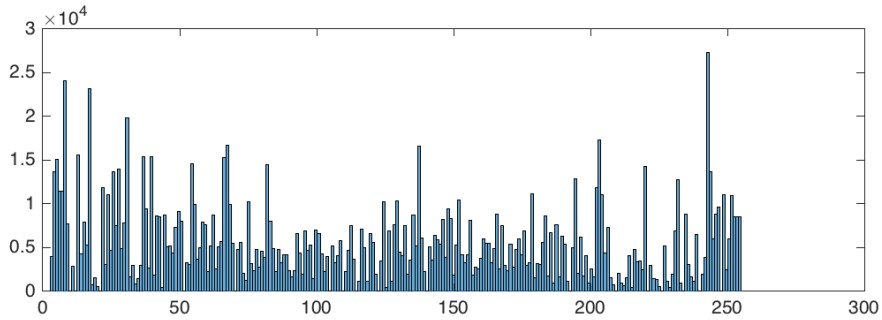
Iš vaizdų histogramų matyti, kad paletės pakeitimo steganografijos algoritmas labai iškraipo originalią histogramą (*žr* 4.37 pav. ir 4.38 pav), bet iškraipymai neįneša į histogramos vaizdą jokių pastebimų dėsningumų, todėl atrodo natūraliai. LSB algoritmo naudojimas, įterpiant tokį pat duomenų kieki, kiek leidžia įterpti paletės pakeitimo steganografijos algoritmas, beveik nedaro jokios įtakos vaizdo histogramai (*žr* 4.39 pav.), nes pakeičiamų pikselių kiekis yra žymiai mažesnis už visą vaizdo pikselių kiekį.



4.37 pav. *Nemodifikuoto vaizdo su adaptyvia palete ir glotninimu histograma*



4.38 pav. *Vaizdo su adaptyvia palete ir glotninimu, į kurį įterpta 2910 bitų duomenų naudojant paletės pakeitimo steganografijos algoritmą, histograma*

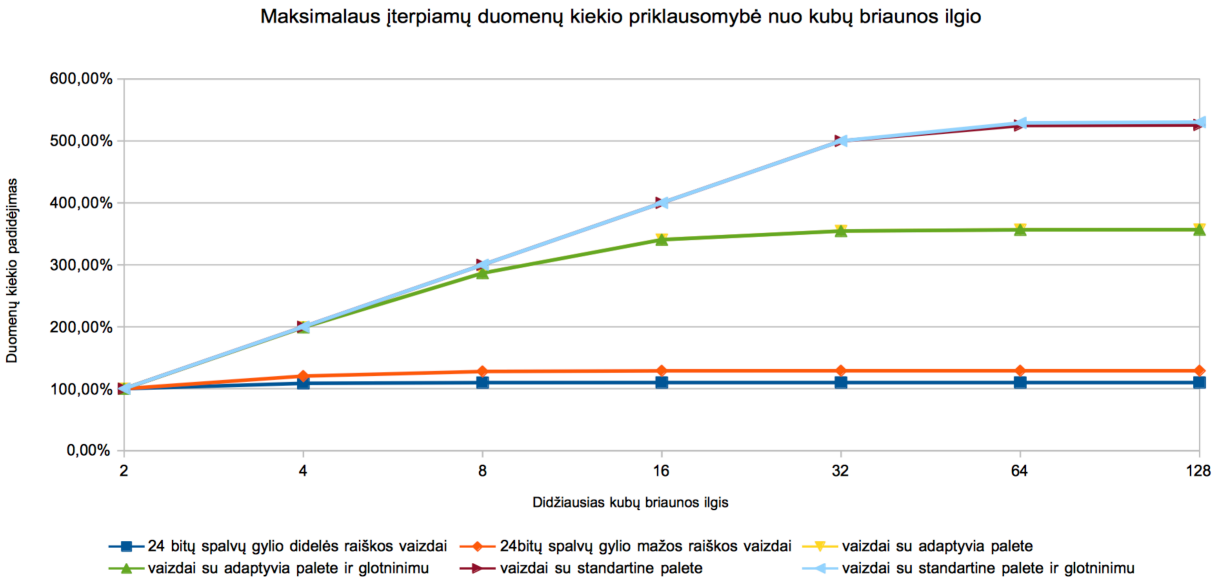


4.39 pav. *Vaizdo su adaptyvia palete ir glotninimu, į kurį įterpta 2910 bitų naudojant LSB steganografijos algoritmą vaizdo indeksams, histograma*

Kaip ir vaizdų su standartine palete atveju, glotninimas teigiamai veikia steganografijos algoritmų taikymo efektyvumą, lyginant su vaizdais be glotninimo. Naudojant paletės pakeitimo steganografijos algoritmą, galima užduoti didesnį kubų briaunos ilgį, be didesnių matomų vaizdo iškraipymų. Taip pat glotninimas maskuoja LSB steganografijos algoritmo sugeneruojamą triukšmą.

4. 4. Duomenų kiekis ir algoritmo veikimo laikas

Eksperimento metu buvo fiksuojamas maksimalus duomenų kiekis, kuri galima įterpti paletės pakeitimo steganografijos algoritmu naudojant skirtingus kubų briaunos ilgius. Apdorojus duomenis, sudarytas grafikas(*žr.* 4.40 pav), kuris parodo kokią įtaką daro kubų briaunos ilgio pakitimas, taikant algoritmą skirtingiems vaizdų tipams. Grafikui sudaryt buvo apskaičiuoti vienodo tipo vaizdų įterpiamų duomenų kiekio vidurkiai.



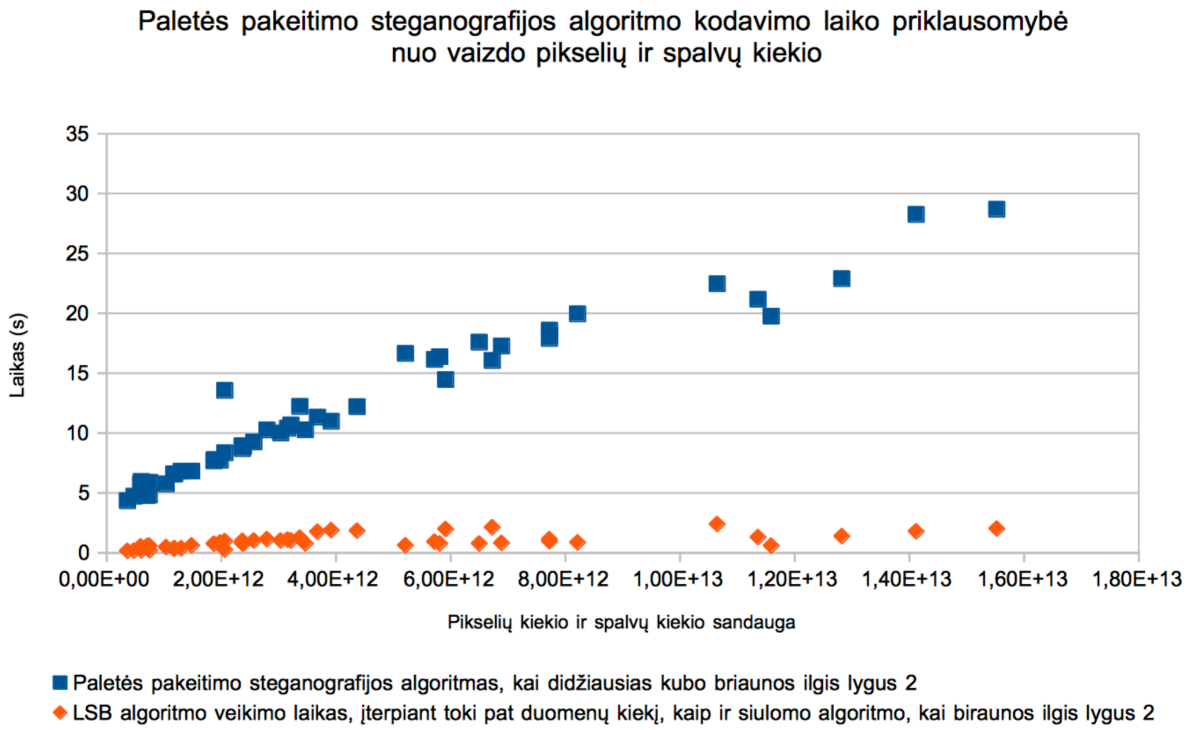
4.40 pav. *Maksimalaus įterpiamų duomenų kiekio priklausomybė nuo kubų briaunos ilgio*

Iš grafiko matyti, kad taikant algoritmą 24 bitų spalvų gylio vaizdams, kubų briaunos ilgis silpnai įtakoja maksimalų iterpiamų duomenų kiekį. Tai yra dėl to, kad šiuose vaizduose spalvų yra, palyginus, daug ir jos išsidėsčiusios pakankamai tankiai, todėl neįmanoma sudaryti didesnių kubų. Skirtumas tarp didelės ir mažos raiškos vaizdų yra labai nedidelis, ir, tikriausiai, atsiranda dėl skirtingų šių vaizdų kompozicijų.

Taip efektyvumo atžvilgiu naudoti didesnius kubų briaunos ilgius taikant algoritmą 24 bitų vaizdams turi mažai prasmės, nes įterpiamų duomenų kiekis pakinta nežymiai, bet gali nukentėti vaizdo kokybė.

Vaizduose su adaptyvia palete spalvų kiekis yra apribotas 256 spalvomis, kurios pasiskirsčiusios spalvų erdvėje netolygiai taip, kad geriau perduoti lėtus spalvų perejimus ir tuo pat metu padengtų kuo didesnį vaizde naudojamų spalvų sprektrą. Matyti, kad, kubų briaunos ilgiui padidėjus nuo 2 iki 4 ir iki 8, įterpiamų duomenų kiekis padidėja du kartus ir beveik tris kartus atitinkamai, tai reškia, kad didžioji dalis spalvų šiuose vaizduose yra nutolusios viena nuo kitos spalvų erdvėje per 8 pozicijas. Spalvų nutolusių nuo kitų per 16 pozicijų yra per pus mažiau, o nutolusių per didesnius atstumus yra tik vienetai. Taip kubų briaunos ilgiai 8 ir 16 yra optimaliausi šiuo atveju, didesnių kubų ilgių naudojimas yra neefektyvus, nes maksimalus įterpiamų duomenų kiekis padidėja nežymiai, bet į vaizdą įnešama daug iškraipymų.

Standartinės paletės spalvos yra pasiskirsčiusios spalvų erdvėje tolygiai ir nutolusios viena nuo kitos per 42 žingsinius. Tai lemia, kad taikant paletės pakeitimo algoritmą vaizdams su standartine palete, visos vaizdo spalvos patenka į kubus su briaunos ilgiu 32, ką ir parodo grafikas – didėjant kubų briaunos ilgiui nuo 2 iki 32, maksimalus įterpiamų duomenų kiekis padidėja 5 kartus. Kadangi šiuose vaizduose yra išnaudojama tik dalis spalvų, todėl yra galimybė dar padidinti įterpiamų duomenų kiekį dėka spalvų, esančių vaizdo spalvų spiečio spalvinėje erdvėje, pakraščiuose, bet tokiu spalvų yra mažai ir atitinkamai duomenų kiekis kinta nežymiai. Vaizdo kokybės atžvilgiu, naudojant kubus su briaunos ilgiu didesnių už 16 iškraipymai labai padidėja, kraštutiniu atveju gali būti naudojamas briaunos ilgis 32 norint įterpti daugiau duomenų, bet didesnių ilgių naudojimas neturi prasmės.



4.41 pav. *Paletės pakeitimo algoritmo ir LSB algoritmo veikimo laikas*

Taip pat eksperimetno metu buvo fiksuojamas paletės pakeitimo steganografijos algoritmo veikimo laikas kiekvienu testniniu atveju. Kadangi algoritmas analizuoja visus paveikslėlio pikselius ir visas paveikslėlyje naudojamas spalvas, tai atitinkamai veikimo laikas turi priklausyti nuo šių dviejų parametrų, ką ir parodo grafikas (*žr. 4.41 pav*). Galima sakyti, kad veikimo paletės pakeitimo algoritmoo veikimo laikas yra tiesiogiai proporcingas pikselių kiekiui ir spalvų kiekiui vaizde. Taikant LSB algoritmą vaizdo analizė neatliekama, todėl algoritmo veikimo laikas nepriklauso nei nuo vaizdo dydžio, nei nuo spalvų kiekio, o tik nuo įterpiamų duomenų kiekio.

5. DARBO IŠVADOS

1. Atlikus dalies egzistuojančių steganografijos skaitmeninėse nuotraukose metodų apžvalgą, pastebėta, kad steganografijos metodai, taikomi skaitmeniniams vaizdams, priklauso nuo vaizdų pateikimo būdų – vaizdą apibrėžiančių objektų, pvz. pikselių spalvinių reikšmių matricos, kvantavimo ir koeficientai ir lentelės, spalvų paletės ir indeksų matricos, kurie savo ruožtu taip pat gali būti pateikti skirtingai, pvz. spalvų paletę galima pateikti kaip spalvų skaitinių reikšmių letelę, arba kaip taškų spiečių spalvų erdvėje. Pateikimo būdai gali būti kaip plačiai naudojami, taip ir naujai sukūriami originalūs, kuriuos kombinuojant atsiranda galimybės naujų steganografijos algoritmų sukūrimui.
2. Darbe pasiūlytas naujas steganografijos metodas, kuris papildomus duomenis į skaitmeninį paveikslėlį įterpia keisdamas ne kiekvieno pikselio spalvą, o pačią spalvų paletę. Taip padidinamas steganografijos metudų skaičius ir atitinkamai apsunkinamas paslėptos informacijos aptikimas paveikslėlyje, bei naudojamo steganografijos metodo nustatymas.
3. Pasiūlytas naujas steganografijos algoritmas realizuotas programiškai, kas leidžia taikyti jį skaitmeniniams vaizdams įterpiant duomenis ir juos atkūriant.
4. Altiktas pasiūlyto steganografijos algoritmo tyrimas, kurio metu nustatyta, kad taikant šį metodą duomenų kiekis, kurį įmanoma įterpti į skaitmeninį vaizdą nėra tiesioiai priklausomas nuo vaizdo raiškos, o priklauso nuo paveikslelyje naudojamų spalvų kiekio ir nuo jų išdėstymo spalvų erdvėje.
5. Nustatyta, kad paletės pakeitimo algoritmo veikimo laikas turi tiesinę priklausomybe nuo spalvų kiekio ir vaizdo pikselių kiekio sandaugos, kas leidžia daryti veikimo laiko prognozę.
6. Nustatyta, kad LSB algoritmas yra daugumoje atvejų pranašesnis už paletės pakeitimo steganografijos algrotimą maksimalaus įterpiamų duomenų kiekio atžvilgiu nuo 50 kartų ir visada pranašesnis veikimo laiko atžvilgiu nuo 5 kartų, bet atskirais atvejais įneša vizualiai mažiau pastebimus iškraipymus į vaizda ir/arba vaizdo histogramą lyginant su LSB algoritmu.

6. LITERATŪRA

1. An Amirtharajan, Rengaraj, and J. B. B. Rayappan. "Steganography—time to time: A review." Research Journal of Information Technology 5.2 (2013): 58-66.
2. Bauermann I. and Steinbacj E. Further Lossless Compression of JPEG Images. Proc. of Picture Coding Symposium (PCS 2004), San Francisco, USA, December 15–17, 2004.
3. Bhatt, Santhoshi, et al. "Image steganography and visible watermarking using LSB extraction technique." Intelligent Systems and Control (ISCO), 2015 IEEE 9th International Conference on. IEEE, 2015.
4. CCITT Rec.T.81 (1992 E) | ISO/IEC 10918-1:1993(E). [žiurėta 2016-04-24]. Prieiga per internetą: < http://www.w3.org/Graphics/JPEG/itu-t81.pdf >.
5. GIF89a Specification. [žiurėta 2016-04-24]. Prieiga per internetą: < http://www.w3.org/Graphics/GIF/spec-gif89a.txt >.
6. Holub, Vojtech, and Jessica Fridrich. "Challenging the doctrines of JPEG steganography." IS&T/SPIE Electronic Imaging. International Society for Optics and Photonics, 2014.
7. Hunt R. W. G. (2004). The Reproduction of Colour (6th ed.). Chichester UK: Wiley–IS&T Series in Imaging Science and Technology.
8. Hussain, Mehdi, and Mureed Hussain. "A survey of image steganography techniques." (2013).
9. Jessica Fridrich. Steganography in digital media. Cambridge 2010 . ISBN 978-0-521-19019-0.
10. Ker, Andrew D., et al. "Moving steganography and steganalysis from the laboratory into the real world." Proceedings of the first ACM workshop on Information hiding and multimedia security. ACM, 2013.
11. Krauskopf, J., Williams, D. R., & Heeley, D. W. (1982). Cardinal directions of color space. Vision research, 22(9), 1123-1131.
12. Michael Stokes, Matthew Anderson, Srinivasan Chandrasekar, Ricardo Motta, A Standard Default Color Space for the Internet – sRGB. 1995 m. [žiurėta 2016-04-24]. Prieiga per internetą: < http://www.w3.org/Graphics/Color/sRGB >.
13. Mohapatra, Chandan, and Manjusha Pandey. "A Review on current Methods and application of Digital image Steganography." International Journal of Multidisciplinary Approach & Studies 2.2 (2015).
14. Niederst Robbins, Jennifer (February 2006). Web Design in a Nutshell. O'Reilly. p. 747.
15. Peter Wayner, Disappearing cryptography: Information hiding: Steganography & watermarking – 3rd ed., Burlington 2009. ISBN 978-0-12-374479-1.
16. Phil Green and Lindsay W. MacDonald (2002). Colour Engineering: Achieving Device Independent Colour. John Wiley and Sons.
17. Trithemius, Johannes, and Wolfgang Ernst Heidel. Steganographia. 1721.
18. Wang, Xiaofeng, Chengcheng Wei, and Xiao Han. "Steganography forensics method for detecting least significant bit replacement attack." Journal of Electronic Imaging 24.1 (2015): 013016-013016.
19. Zielińska, Elżbieta, Wojciech Mazurczyk, and Krzysztof Szczypiorski. "Trends in steganography." Communications of the ACM 57.3 (2014): 86-95.
20. Аграновский А. В., Балакин А. В., Грибунин В. Г., Сапожников С. А. Стеганография, цифровые водяные знаки и стеганоанализ. М.: Вузовская книга, 2009. ISBN 978-5-9502-0401-2.

21. Павел @PavelMSTU. Стеганография в XXI веке. Цели. Практическое применение. Актуальность. Iš Habrahabr.ru [interaktyvus]. 2015 m. kovo 15d. [žiurėta 2016-04-24]. Prieiga per internetą: < https://habrahabr.ru/post/253045/ >.