



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

Ignas Giedraitis

VAIZDO MEDŽIAGOS ŽYMĖJIMO VANDENS ŽENKLAIS
TYRIMAS

Baigiamasis magistro projektas

Vadovas

Doc. dr. Armantas Ostreika

KAUNAS, 2016

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

VAIZDO MEDŽIAGOS ŽYMĖJIMO VANDENS ŽENKLAIS
TYRIMAS

Baigiamasis magistro projektas
Informatika (kodas 621I10003)

Vadovas

Doc. dr. Armantas Ostreika

Recenzentas

Doc. dr. Tomas Blažauskas

Projektą atliko

IFM-4/1 gr. studentas Ignas Giedraitis

2016-05-20

KAUNAS, 2016



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

(Fakultetas)

Ignas Giedraitis

(Studento vardas, pavardė)

621I10003

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Vaizdo medžiagos žymėjimo vandens ženklais tyrimas“

AKADEMINIO SAŽININGUMO DEKLARACIJA

2016 m. gegužės 20 d.
Kaunas

Patvirtinu, kad mano, **Igno Giedraičio**, baigiamasis projektas tema „Vaizdo medžiagos žymėjimo vandens ženklais tyrimas“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)

(parašas)

Giedraitis, Ignas. Vaizdo medžiagos žymėjimo vandens ženklais tyrimas. Magistro baigiamasis projektas / vadovas doc. dr. Armantas Ostreika; Kauno technologijos universitetas, Informatikos fakultetas.

Mokslo kryptis ir sritis: Informatika

Reikšminiai žodžiai: *vaizdo žymėjimas, vandens ženklas, diskrečioji kosinusų transformacija, diskrečioji bangelių transformacija, vaizdo glaudinimas, H.264, H.265.*

Kaunas, 2016. 56 p.

SANTRAUKA

Sparčiai besivystančios skaitmeninės technologijos pakeitė požiūrį į intelektinę nuosavybę, todėl didėja prevencinių priemonių poreikis, siekiant užkirsti kelią neleistiniems veiksams, ypač vizualinio turinio srityje. Pastaruoju metu viena iš plačiausiai taikomų vaizdinės informacijos apsaugojimo priemonių – vaizdo žymėjimas vandens ženklais.

Siekiant užtikrinti tinkamą skaitmeninio kūrinio apsaugą, svarbu ištirti, koks vandens ženklo įterpimo į vaizdo duomenų srautą metodas yra pranašesnis ir kuris glaudinimo algoritmas efektyvesnis, atsižvelgiant į vandens ženklo išliekamumą po įvairių vaizdo modifikacijų.

Todėl šiame darbe yra analizuojami vandens ženklo įterpimo į vaizdą metodai – diskrečioji kosinusų transformacija ir diskrečioji bangelių transformacija. Atlikus teorinę jų analizę abu metodai pritaikomi praktiškai, analizuojant vandens ženklo atsparumą vaizdo modifikacijoms.

Eksperimento metu pastebėjus, jog efektyvesnis yra diskrečiosios bangelių transformacijos metodas, jis buvo pritaikytas vaizdo sekos žymėjimui vandens ženklais ir atliekamos vaizdo manipuliacijos bei glaudinimas, siekiant išsiaiškinti vandens ženklo išliekamumo pokytį.

Atlikus vandens ženklu pažymėto vaizdo manipuliacijas ir apdorojus skirtingais glaudinimo algoritmais, gauti rezultatai parodo, kad vandens ženklo atpažįstamumas yra geresnis ir išliekamumas didesnis vaizdų seką glaudinant H.264 algoritmu, nei apdorojant vaizdą H.265 algoritmu.

Giedraitis, Ignas. Application Of Watermarks In Video Tagging: Master's thesis in field of Informatics / supervisor Assoc. Prof. Dr. Armantas Ostreika. The Faculty of Informatics, Kaunas University of Technology.

Research area and field: Informatics

Key words: *video watermarking, discrete cosine transformation, discrete wavelet transformation, video compression, H.264, H.265.*

Kaunas, 2016. 56 p.

SUMMARY

Rapid development of digital technologies has changed the approach to the intellectual property. Because of this reason, the need for protection of intellectual property is increasing, especially in the field of multimedia. One of the most popular methods of video content protection is video watermarking.

In order to ensure proper protection of intellectual property, it is important to do a research of existing methods of video watermarking and to find out which of them is more efficient. Moreover, a detailed analysis, based on difference between original watermark and extracted watermark after various image attacks and compression would reveal which compression algorithm is more accurate.

Therefore, in this Master thesis watermark embedding methods are analyzed, such as discrete cosine transform and discrete wavelet transform. After that, both methods were applied in the experiment, where the resistance of watermark to image attacks were analyzed.

Experiment results showed that more effective is discrete wavelet transformation and this method was applied in video watermarking. Then watermarked video was modified by various attacks and compressed using two different algorithms in order to find out level of watermark robustness.

Overall, based on results of the experiment, extracted watermark is more robust and visually recognizable when watermarked video content is compressed using H.264 algorithm than compressed using H.265.

TURINYS

IVADAS	9
1. VAIZDO ŽYMĖJIMO VANDENS ŽENKLAIS METODŲ ANALIZĖ	12
1.1. PAGRINDINĖS VANDENS ŽENKLŲ TAIKYMO SRITYS	12
1.2. EGZISTUOJANTYS SPRENDIMAI	12
1.3. VANDENS ŽENKLŲ TIPAI	13
1.3.1. Matomi vandens ženklai	13
1.3.2. Nematomi vandens ženklai	13
1.4. DARBO PRIEMONIŲ IR TECHNOLOGIJŲ ANALIZĖ	14
1.4.2. Erdvinės srities metodai	15
1.4.3. Dažninės srities metodai	16
1.5. VAIZDO GLAUDINIMO STANDARTAI	24
1.6. ATAKŲ PRIEŠ VANDENS ŽENKLUS TIPAI	25
2. VAIZDO ŽYMĖJIMO VANDENS ŽENKLAIS PŲ PROJEKTAVIMAS	26
2.1. REIKALAVIMŲ SPECIFIKACIJA	26
2.1.1. Funkciniai reikalavimai	26
2.1.2. Nefunkciniai reikalavimai	27
2.2. EKSPERIMENTO PRIEMONĖS	27
2.3. PALYGINIMO PARAMETRAI	28
2.3.1. MSE	28
2.3.2. PSNR	28
2.3.3. SSIM	28
2.3.4. Vandens ženklo įterpimo stiprumo parinkimas	29
3. VAIZDO ŽYMĖJIMO VANDENS ŽENKLAIS PŲ REALIZACIJA	31
3.1. PIRMASIS EKSPERIMENTO ETAPAS	31
3.1.1. Statinio vaizdo žymėjimas vandens ženklu ir vandens ženklo išgavimas	32
3.1.2. Vandens ženklu pažymėtos vaizdų sekos glaudinimas ir vandens ženklo išgavimas	33
3.2. EKSPERIMENTO DUOMENYS	34
3.3. ANTRASIS EKSPERIMENTO ETAPAS	36
IŠVADOS	42
LITERATŪRA	43

PAVEIKSLŲ SĄRAŠAS

1 pav. Vandens ženklų įterpimo schema.....	14
2 pav. Vandens ženklų išgavimo schema.....	15
3 pav. Dažnių koeficientų pasiskirstymas vaizdo bloke atlikus DCT.....	20
4 pav. Signalų suskaidymo, pritaikant aukštų ir žemų dažnių filtrus, schema	20
5 pav. Vieno lygio diskrečiosios bangelių transformacijos vaizdo suskaidymas	21
6 pav. Dvimačio vaizdo vieno lygio diskrečiosios bangelių transformacijos schema.....	22
7 pav. Vaizdas po vieno lygio diskrečiosios bangelių transformacijos	22
8 pav. Dviejų lygių diskrečiosios bangelių transformacijos vaizdo suskaidymas	22
9 pav. Vaizdas po antro lygio diskrečiosios bangelių transformacijos	23
10 pav. Dvimačio vaizdo vieno lygio atvirkštinės diskrečiosios bangelių transformacijos schema ...	23
11 pav. Vaizdas, kai vandens ženklų stiprumas lygus 0,008	30
12 pav. Vaizdas, kai vandens ženklų stiprumas lygus 0,3	30
13 pav. Neglaudinto video failo žymėjimo vandens ženklų schema	31
14 pav. Suglaudinto video failo žymėjimo vandens ženklų schema.....	31
15 pav. Vandens ženklas	32
16 pav. Vaizdas iš pirmosios vaizdų sekos	32
17 pav. Vaizdas iš antrosios vaizdų sekos	32
18 pav. Glaudintas vandens ženklų pažymėtas vaizdas	36
19 pav. Vandens ženklų pažymėtas vaizdas su horizontaliomis juodomis juostomis	38
20 pav. Vandens ženklų pažymėtas vaizdas, paveiktas „Salt&Pepper“ filtru	39
21 pav. Vandens ženklų pažymėtas vaizdas, paveiktas „Gaussian“ filtru	40

LENTELIŲ SĄRAŠAS

1 lentelė. Esminių erdvinės ir dažnių srities faktorių palyginimas	15
2 lentelė. LSB metodo taikymo principas	16
3 lentelė. Skirtingoms vaizdo raiškoms taikomi glaudinimo duomenų srautai	24
4 lentelė. Rekomenduojamos vandens ženklo stiprumo reikšmės	30
5 lentelė. Iš skirtingais metodais apsaugotų statinių vaizdų išgautų vandens ženklų palyginimas	32
6 lentelė. Iš H.264 algoritmu suglaudinto vaizdo išgauto vandens ženklo palyginimas	33
7 lentelė. Iš H.265 algoritmu suglaudinto vaizdo išgauto vandens ženklo palyginimas	34
8 lentelė. Eksperimento duomenys vaizdų sekos žymėjimui vandens ženklu	35
9 lentelė. Videoklipo suglaudavimo komandos parametrų reikšmės	35
10 lentelė. Videoklipo išskaidymo į atskirus vaizdo kadrus komandos parametrų reikšmės	35
11 lentelė. Vaizdų sekos sujungimo į vieną videoklipą komandos parametrų reikšmės	36
12 lentelė. Iš nemodifikuoto vaizdo išgauto vandens ženklo rezultatai.....	37
13 lentelė. Iš juodomis juostomis pažymėto vaizdo išgauto vandens ženklo rezultatai	38
14 lentelė. Iš „Salt&Pepper“ filtru manipuluoto vaizdo išgauto vandens ženklo rezultatai	39
15 lentelė. Iš „Gaussian“ filtru manipuluoto vaizdo išgauto vandens ženklo rezultatai.....	41

IVADAS

Didelės spartos kompiuteriniai tinklai, internetas, saitynas (angl. *World-Wide-Web, WWW*) iš esmės pakeitė skaitmeninės informacijos sklaidą. Augantis vaizdinės medžiagos taikymo populiarumas – internetinė multimedija, filmuojamo vaizdo perdavimas bevieliu ryšiu realiu laiku, vaizdo konferencijos, tokios interaktyviosios televizijos teikiamos paslaugos, kaip filmų nuoma ar galimybė peržiūrėti anksčiau parodytos norimos laidos įrašą (angl. *Video-On-Demand, VOD*), – didina intelektinės nuosavybės apsaugos svarbą.

Plačiai paplitęs, lengvai prieinamas ir praktiškai beribis įvairialypės informacijos turinys (angl. *Multimedia content*), galimybė išsisaugoti neribotą kiekį jo kopijų neprarandant kokybės, kūrinių naudojimas be autorių sutikimo – dėl šių priežasčių autoriaus teises į skaitmeninį turinį turintys kūrėjai siekia, jog jų kūriniai būtų apsaugoti, o prieiga prie jų – kontroliuojama. Šią funkciją atlieka vartotojų teisių apsaugos sistemos arba skaitmeninis teisių valdymas (angl. *Digital Rights Management, DRM*). Viena iš technologijų, naudojamų DRM sistemose, siekiant apsaugoti skaitmeninę intelektinę nuosavybę, – skaitmeninio turinio žymėjimas vandens ženklais (angl. *Watermarking*). Būtent dėl susirūpinimo autorių teisių apsaugos galimybėmis buvo pavēlintas DVD (angl. *Digital Versatile Disc*) diskų atsiradimas rinkoje 1996-aisiais metais [20]. Filmų kūrimo industrijos bendrovės išvelgė problemą, jog diskuose esantis turinys gali būti neribotai kopijuojamas ir pardavinėjamas ar platinamas internete [5]. Galimi sprendimai ir pasiūlymai buvo pateikti svarstyti Jungtinių Amerikos Valstijų (JAV) Kongresui, kurio nariai siekė teisiškai reglamentuoti, kas ir kokios yra technologinės skaitmeninio turinio apsaugos priemonės bei apibrėžti teisinę atsakomybę už autorių teisių pažeidimus. Tai buvo esminis pagrindas 1998-aisiais metais priimto Skaitmeninio tūkstantmečio autorių teisių akto (angl. *The Digital Millennium Copyright Act*) [18].

Vandens ženklų funkcionalumas ir efektyvumas pagrįstas tuo, jog, siekiant apsaugoti skaitmeninį turinį nuo nelegalių veiksmų, į duomenų srautą papildomai įkomponuojamas skaitmeninis parašas, arba kitaip – vandens ženklas. Pagrindinis žymėjimo vandens ženklais tikslas – įtraukti duomenys turi būti nepastebimi, naudingi ir nepašalinami. Tai skaitmeninė informacija, pilnai galinti apibūdinti kūrinio autorių, kuriam priklauso intelektinė nuosavybė. Vandens ženklas neapsaugo turinio nuo jo neleistino naudojimo – skaitymo, žiūrėjimo, klausymo – tačiau vos tik pradėjus turinio peržiūrą, integruotų sistemų, programinės įrangos pagalba yra galimas perspėjimo pranešimo siuntimas atsakingoms institucijoms apie nelegalius vartotojo veiksmus. Vandens ženklų privalumas yra tai, jog jie negali būti pašalinti iš duomenų srauto nepaveikus turinio kokybės.

Turinio žymėjimas vandens ženklais yra vienas iš steganografijos atvejų, kai informacija yra slepiama kituose duomenyse. Įprastai paslepiama yra visa informacija, tačiau žymint skaitmeninį turinį vandens ženklais, kitaip nei vaizdo, garso duomenų srautai, paslėpta žinutė yra tik vandens ženklų signalas ir jo reikšmė duomenų sraute.

Įkomponuoti duomenys gali būti įvairūs – specifinę reikšmę turintis serijinis numeris, atsitiktinai sugeneruotas specifinę reikšmę turintis skaičius, inicialai, autoriaus žinutė, trumpas kūrinio turinio aprašymas ar techninės failo charakteristikos. Kaip vienu iš informacijos, kuri gali būti įtraukta į duomenų srautą, tipų laikomi ir metaduomenys (angl. *Metadata*). Struktūrizuoti pateikti duomenys gali suteikti daugiau papildomos informacijos apie norimą kūrinį, susieti atitinkamus atributus su interneto svetainėmis ar duomenų bazėmis. Metaduomenys gali palengvinti integraciją į vidines įvairių tarnybų sistemas, kontroliuojančias ir užtikrinančias atsiskaitymus už autorių teisių saugomų kūrinių panaudojimą. Praktikoje vandens ženklai gali būti taikomi atpažįstant ir kaupiant duomenis apie televizijų transliuojamose programose panaudotus autorinius kūrinius – dainas, vaizdo klipus ar jų ištraukas, parodytus kitus meno darbus. Padedant šiuolaikiniams išmaniesiems televizoriams, telefonams ar planšetiniams kompiuteriams bei specializuotai programinei įrangai, vaizdo medžiagos vandens ženklai, kurių duomenų sraute metaduomenų pavidalu aprašyti turinyje panaudoti kūriniai, būtų nuskaitomi ir pateikiama trukmė kadro tikslumu, kiek truko faktinis konkretaus autorinio kūrinio parodymas televizijos eteryje. Automatiškai suformuojamos ir internetu siunčiamos ataskaitos šias sritis prižiūrinčioms tarnyboms palengvintų bei optimizuotų bendrą darbo procesą ir padidintų jo preciziškumą. Taip pat, naudojant panašaus pobūdžio taikomas programas, galima kaupti televizijos žiūrimumo statistiką, stebėti ir analizuoti žiūrovų elgesį bei įpročius prie televizorių ekranų.

Vaizdo žymėjimas vandens ženklais yra viena iš plačiausiai naudojamų intelektinės nuosavybės apsaugos priemonių, o vandens ženklo atpažįstamumas yra svarbiausias jos rodiklis. Todėl, siekiant užtikrinti tinkamą skaitmeninio kūrinio apsaugą, svarbu iširti, koks vandens ženklo įterpimo į vaizdo duomenų srautą metodas yra pranašesnis ir kuris glaudinimo algoritmas efektyvesnis, atsižvelgiant į vandens ženklo išliekamumą po įvairių vaizdo modifikacijų.

Tikslas – atlikus vandens ženklų pažymėto vaizdo manipuliacijas, apdorojus vaizdo medžiagą skirtingais glaudinimo algoritmais ir išanalizavus į vaizdą įterpto vandens ženklo išliekamumo pokytį, nustatyti efektyvesnį vaizdo glaudinimo algoritmą.

Uždaviniai:

1. Atlikti vandens ženklų technologijos taikymo sričių bei galimybių apžvalgą;
2. Eksperimentiniu būdu pritaikyti vandens ženklų technologiją vaizdo medžiagoje;
 - 2.1. Pritaikyti DCT metodą vandens ženklo įterpimui į statinį vaizdą;
 - 2.2. Pritaikyti DWT metodą vandens ženklo įterpimui į statinį vaizdą;
 - 2.3. Iširti vaizdo manipuliacijų įtaką vandens ženklo išliekamumui;
 - 2.4. Parinkti atpažįstamumo vertinimo parametrus;
3. Eksperimento metu gautų rezultatų pagrindu atlikus tyrimą ir vandens ženklo išliekamumo pokyčio įvertinimą, nustatyti efektyvesnį vaizdo glaudinimo algoritmą.

Siekiant nustatyti efektyvesnį vaizdo glaudinimo algoritmą vandens ženklo išliekamumui, svarbu atlikti teorinę vaizdo žymėjimo vandens ženklais metodų ir vaizdo glaudinimo algoritmų analizę.

1. VAIZDO ŽYMĖJIMO VANDENS ŽENKLAIS METODŲ ANALIZĖ

Šiame skyriuje apžvelgiamos pagrindinės vandens ženklų technologijų taikymo sritys, egzistuojanti ir praktikoje naudojama programinė įranga (toliau – PI). Taip pat analizuojami plačiausiai taikomų vandens ženklo įterpimo į vaizdą metodai, tarpusavyje palyginamos bendros charakteristikos.

1.1. PAGRINDINĖS VANDENS ŽENKLŲ TAIKYMO SRITYS

Literatūroje yra išskiriamos pagrindinės sritys, kuriose yra taikoma vaizdo žymėjimo vandens ženklais technologija.

- 1) **Autorių teisių apsauga:** siekiant išvengti autorių teisių pažeidimų, skaitmeninė intelektinės nuosavybės versija turi būti apsaugota [16]. Tai gali būti ir vandens ženklas, atspindintis skaitmeninio kūrinio autoriaus tapatybę.
- 2) **Kopijų panaudojimo sekimas:** siekdamas užtikrinti legalų naudojimąsi kūriniiais, intelektinės nuosavybės autorius failus gali žymėti skirtingais vandens ženklais – į duomenų srautą įterpta informacija gali būti susijusi su vartotojo – pirkėjo tapatybe – tai savininkui padėtų identifikuoti asmenis, kurie pažeidė susitarimą ir nelegaliai perdavė skaitmeninius kūrinius trečiosioms šalims.
- 3) **Stebėjimas:** į vaizdinę informaciją įkomponuoti vandens ženklai gali teikti duomenis, tiesiogiai susijusius su atlygiu, atitenkančiu intelektinės nuosavybės autoriui. Automatizuotos stebėjimo sistemos (angl. *monitoring systems*), stebinčios televizijose, kinuose, reklaminiuose stenduose transliuojamą turinį, nuskaito failo vandens ženklą ir apie faktinį ištransliavimą informuoja institucijas, atsakingas už autorinio užmokesčio surinkimą. Ši funkcija taip pat naudinga ir komercinės reklamos kūrėjams, kuriems specializuota programinė įranga teiktų informaciją, ar pagal susitarimą – sutartu paros laiku, nurodytose reklaminių intarpų pozicijose, reikiamu dažnumu – yra transliuojami reklamuojamo produkto, paslaugos vaizdo klipai [1].
- 4) **Autentiškumo patikrinimas:** vandens ženklai gali teikti informaciją apie failo autentiškumą. Esant pakitusiam duomenų srautui, vandens ženklo pagalba galima identifikuoti failo pokyčius ir praneša apie pakitusius elementus.
- 5) **Apsauga nuo kopijavimo:** informacija, saugoma vandens ženkluose, gali tiesiogiai valdyti vaizdo įrašymo įrenginius. Esant sąlygai, jog konkretus kūrinys negali būti kopijuojamas, failo duomenų srautą nuskaitęs ir tokį požymį aptikęs vaizdo įrašymo įrenginys atitinkamai neinicijuoja turinio įrašymo į laikmeną funkcijos.

1.2. EGZISTUOJANTYS SPRENDIMAI

Skaitmeninių duomenų žymėjimo vandens ženklais aljansas (angl. *The Digital Watermarking Alliance, DWA*) – 2006-aisiais metais įkurta asociacija, kurią vienija tarptautinės bendrovės, savo srityse pirmaujančios teikiant skaitmeninių vandens ženklų technologijos sprendimus visame

pasaulyje. Šio aljanso misija: informuoti apie žymėjimo vandens ženklais galimybes ir skleisti technologijos naudą, vertę turinio kūrėjams, industrijai, įstatymų leidėjams, vartotojams [24].

Šiuo metu DWA priklauso dešimt tarptautinių bendrovių, penkios iš jų specializuojasi vaizdinio ir garsinio turinio apsaugos srityje: „Digimarc“, „NexGuard“, „MarkAny“, „Media Science International“ ir „Verimatrix“. Viena populiariausių laikoma „Digimarc“ siūlo nemokamą produkto „Digimarc Guardian“ įskiepi (angl. *plug-in*) nuotraukų ir vaizdo programai „Adobe Photoshop“, kuriuo naudojantis galima pažymėti vaizdą vandens ženklu arba sužinoti, ar apdorojama nuotrauka yra apsaugota vandens ženklu. Papildomos žiniatinklio tarnybos (angl. *web service*), naudojančios interneto paieškos variklius (angl. *web search engines*) ieško apsaugotų vaizdų, pažymėtų vandens ženklu bei priklausančiu konkrečiam asmeniui ir automatiškai praneša, kuriose interneto svetainėse yra naudojama jo intelektinė nuosavybė [26].

1.3. VANDENS ŽENKLŲ TIPAI

Vandens ženklų vaizdo medžiagoje paskirtis – patvirtinti skaitmeninio turinio autorių, savininką. Kūrinio žymėjimo vandens ženklais metu į vaizdo duomenų srautą įtraukiamas žymuo, kuris išlieka net ir padarius daugybę failo kopijų. Pagal matomumą vandens ženklai yra skiriami į du tipus – matomus ir nematomus.

1.3.1. Matomi vandens ženklai

Tokio tipo vandens ženklai naudojami tais atvejais, kai skaitmeninio turinio teisių savininkas nori, jog būtų žinoma, kad kūrinys yra pažymėtas ir turinį yra siekiama apsaugoti. Kita priežastis – prekinio ženklo (angl. *brand*) reklama [12]. Toks vandens ženklų taikymas plačiai paplitęs televizijose, taip pat populiarius ir internete. Šiose srityse vandens ženklai paprastai būna įvairūs užrašai arba logotipai. Pastarieji dažniausiai talpinami vaizdo kampuose tiek viršuje, tiek apačioje, kad netrukdytų stebėti transliuojamo turinio ir neerzintų žiūrovo. Siekiant dar labiau sumažinti sutelkiamą dėmesį į vandens ženklus, pasiteisino praktika šiems grafiniams elementams pritaikyti skaidrumo efektą (angl. *transparency*) – esant šviesiam transliuojamam vaizdui vandens ženklas tarytum pranyksta. Šis efektas dažnai taikomas aukštosios raiškos (angl. *high definition*) vaizdams, kurie pasižymi sodresnėmis ir ryškesnėmis spalvomis.

1.3.2. Nematomi vandens ženklai

Nematomi vandens ženklai pasižymi savybe, jog vaizdo medžiagoje jie vizualiai nėra matomi, tačiau duomenų sraute yra įkomponuota specifinė informacija – specialus serijinis numeris, kūrinio autoriaus tapatybę nusakanti informacija, autorių teisių žinutė, turinio aprašymas, metaduomenys ar kt. Kadangi plika akimi negalima nustatyti, ar vaizdo medžiaga yra apsaugota nematomais vandens ženklais, tai atlikti gali tik tokiai funkcijai pritaikyta programinė įranga, analizuojanti filmuotos medžiagos duomenų srautą ir išfiltruojanti įkomponuotą informaciją. Viena iš pagrindinių šio tipo

vandens ženklų pritaikymu išpildomų sąlygų – neteisėtas jų aptikimas ir šalinimas turi būti neįmanomas, net jei ir yra žinoma pagrindinė schema, kaip buvo įterptas vandens ženklas į duomenų srautą.

1.4. DARBO PRIEMONIŲ IR TECHNOLOGIJŲ ANALIZĖ

1.4.1.1 Bendrinis vandens ženklo įterpimo būdas

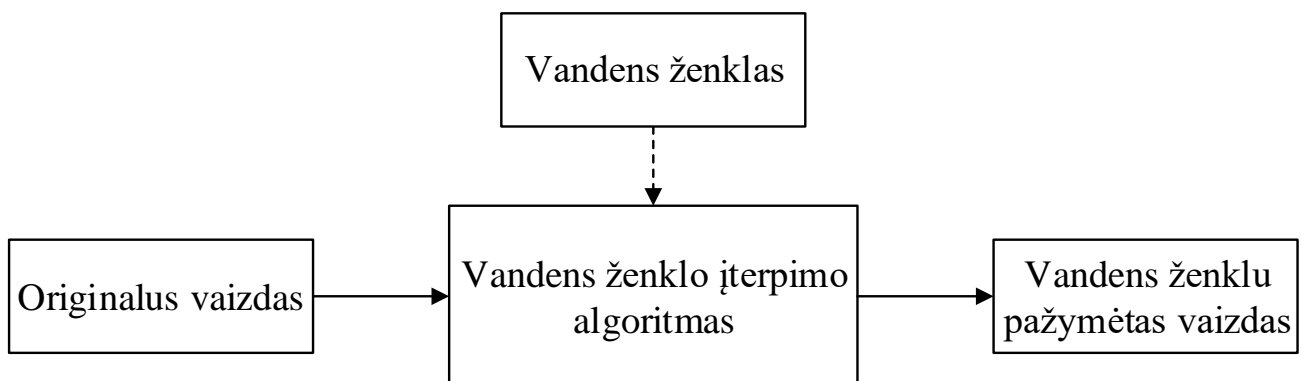
Filmuota vaizdinė medžiaga yra ne kas kita kaip atskirų statinių vaizdų – kadru – seka. Šią originalaus vaizdo seką laikykime $H_i(x, y)$, kai i yra unikalus kadro indeksas, o (x, y) – kadro dydis (aukštis, plotis). Vandens ženklas yra lygus $W_i(x, y)$, jo matmenys – tokie patys, kaip ir originalaus vaizdinės medžiagos failo kadro. Iš to seka vandens ženklo įterpimo į vaizdo duomenų srautą formulė (1), kai $\alpha_i(x, y)$ yra koeficientas, kurį dauginant kinta įterpiamo vandens ženklo dydis.

$$F_i(x, y) = H_i(x, y) + \alpha_i(x, y) \cdot W_i(x, y) \quad (1)$$

Gautas rezultatas $F_i(x, y)$ – vandens ženklų pažymėta vaizdų seka – vaizdinės medžiagos failas. H_i yra sritis, į kurią įterpiamas vandens ženklas. Skirtinguose vaizdų žymėjimo vandens ženklais algoritmuose taikomi skirtingi – erdvinių ar dažninių – sričių tipai. Atitinkamai algoritmai tarpusavyje skiriasi ir $\alpha_i(x, y)$ taikymu, koeficientą pridėdant ar dauginant.

Vandens ženklo įkomponavimo ir išgavimo iš vaizdo medžiagos metodai turi būti nesudėtingi, kadangi operacijos gali būti atliekamos realiu laiku. Siekiant didesnio efektyvumo, nėra tikslinga prieš įterpiant vandens ženklą vaizdo medžiagą atkoduoti (angl. *decompression*), į duomenų srautą įkomponuoti pasirinktą vandens ženklą ir vėl suspausti norimu glaudinimo (angl. *compression*) algoritmu. Siekiant išvengti nereikalingų operacijų, vandens ženklai gali būti įterpti tiesiogiai į suglaudintą vaizdo klipą.

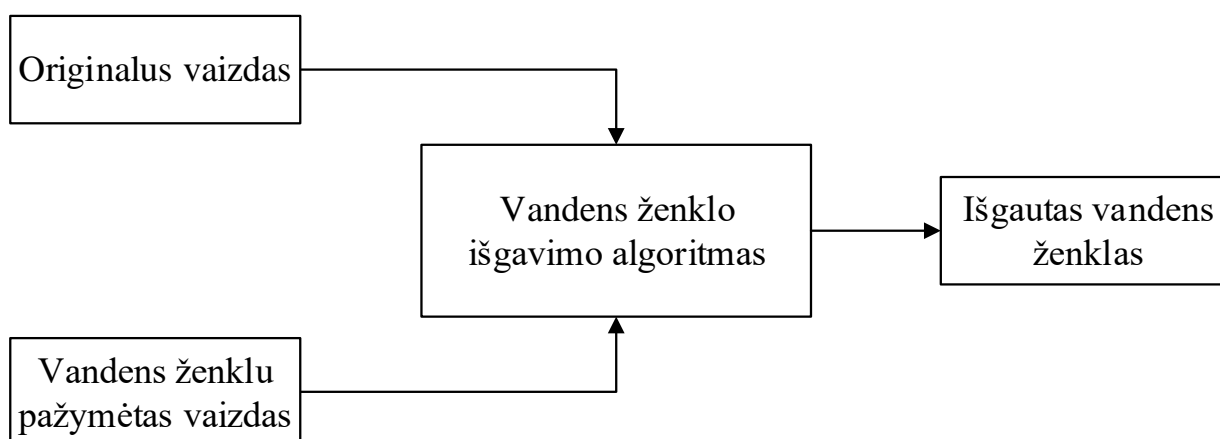
Nepriklausomai nuo vaizdo žymėjimo vandens ženklais metodikos, kiekviena iš jų susideda iš dviejų dalių – vandens ženklo įterpimo į vaizdą ir vandens ženklo aptikimo, išgavimo iš pažymėto vaizdo.



1 pav. Vandens ženklo įterpimo schema

1 paveiksle schematiškai pavaizduotas vandens ženklo įterpimo į vaizdą principas – taikant pasirinktą algoritmą originalus vaizdas pažymimas norimu vandens ženklu. Atvirkštinis procesas – vandens ženklo aptikimas ir išgavimas (žr. 2 pav.) – atliekamas vandens ženklo aptikimo algoritmu

pritaikant apsaugotam vaizdui. Vandens ženklo išgavimo procesą palengvina to paties vaizdo originali versija, algoritmo pagalba leidžianti nustatyti pakitusias pažymėto vaizdo vietas.



2 pav. Vandens ženklo išgavimo schema

Vandens ženklas vaizdinėje medžiagoje gali būti įterptas tiek matomai, tiek nematomai. Norint gauti patikimą galutinį rezultatą, reikia pasirinkti atitinkamą metodą operacijai atlikti. Atlikti įvairias vaizdo modifikacijas, tarp jų ir vandens ženklo įkomponavimą, galima dvejų tipų vaizdo srityse – erdvinėje (angl. *spatial domain*) ir dažnių (angl. *frequency domain*), kitaip vadinamoje transformacijoje (angl. *transform domain*).

1.4.2. Erdvinės srities metodai

Erdvinė vaizdo sritis yra ne kas kita kaip vaizdą sudarančių taškų (angl. *pixel*) visuma. Įterpiant vandens ženklą erdvinėje srityje į pasirinktus taškus, pakeičiamos jų reikšmės, nuo kurių vizualiai keičiasi spalva. Vaizdo žymėjimo vandens ženklais erdvinėje srityje privalumai yra:

- 1) Paprastumas,
- 2) Nedidelės laiko sąnaudos apdorojimui,
- 3) Nesudėtingi atliekami skaičiavimo veiksmai.

Lyginant 1 lentelėje pateiktus esminius erdvinės ir dažnių faktorius, galima teigti, jog metodai, kuriuos taikant vandens ženklo įterpiami į vaizdo erdvinę sritį, pasižymi paprastesniais skaičiavimo algoritmais, todėl operacijos užtrunka mažiau laiko. Kita vertus, erdvinė vaizdo sritis yra mažiau atspari įvairioms vaizdo modifikacijoms ir bet kokie vaizdo pokyčiai gali sugadinti paslėptą vandens ženklą [14]. Nors vandens ženklo įkomponavimas dažnių srityje dėl sudėtingesnių algoritmų skaičiavimų trunka ilgiau, rezultatas gaunamas naudingesnis – įterptas vandens ženklas patvaresnis ir atsparesnis vaizdo manipuliacijoms.

1 lentelė. Esminių erdvinės ir dažnių srities faktorių palyginimas

Faktorius	Erdvinė sritis	Dažnių sritis
Apdorojimo sudėtingumas	Žemas	Aukštesnis
Patvarumas	Silpnas	Patvaresnis
Atsparumas vaizdo modifikacijoms	Žemas	Aukštesnis
Apdorojimo trukmė	Trumpa	Ilgesnė
Užimamas dydis	Didelis (priklausomai nuo vaizdo skiriamosios gebos)	Mažas

1.4.2.1 Mažiausio reikšmingumo bito metodas

Populiarus ir plačiai naudojamas vaizdo vandens ženklais žymėjimo erdvinėje srityje metodas – mažiausio reikšmingumo bito metodas (angl. *Least Significant Bit, LSB*). Pilkos spalvos pustoniu (angl. *grayscale*) vaizde kiekvienas taškas gali turėti vieną iš 256 skirtingų reikšmių – nuo visiškai baltos, kai reikšmė lygi 0, iki visiškai juodos, kai reikšmė lygi 255. Taikant LSB metodą, kiekvieno pasirinkto vaizdo taško reikšmė išreiškiama dvejetainine 8 bitų seka, o paskutinis jos narys yra pakeičiamas kiekvienu vandens ženklo dvejetainiu nariu (žr. 2 lent.). Žymint vandens ženklą spalvotą vaizdą, papildoma informacija įterpiama į vieną iš vaizdo komponentų. Vaizdo spalvų modelyje, kuriame visos spalvos gaunamos iš trijų pagrindinių spalvų – raudonos, žalios ir mėlynos – RGB (angl. *red-green-blue model*).

2 lentelė. LSB metodo taikymo principas

Originalaus vaizdo taškų reikšmės	10100001	10100001	10100000	10011111	10100000	10100110
Vandens ženklas	0	1	1	0	0	1
Pažymėto vaizdo taškų reikšmės	10100000	10100001	10100001	10011110	10100000	10100111

Mažiausio reikšmingumo bito metodą nėra sunku taikyti dėl ganėtinai paprastų matematinių operacijų, taip pat po atliktų veiksmų nėra pastebimas vaizdo iškreipimas, tačiau LSB metodas nėra itin atsparus vaizdo manipuliacijoms. Priklausomai nuo vaizdo skiriamosios gebos, egzistuoja maksimalus vandens ženklo dydis, pvz.: 512 taškų aukščio ir 512 taškų pločio vaizdas užima 262144 bitus, todėl, remiantis LSB metodo veikimo principu, jog į kiekvieną tašką galima įterpti tik vieną bitą duomenų, didžiausias vandens ženklo sudarančios informacijos kiekis gali būti 32768 bitai.

1.4.3. Dažninės srities metodai

Egzistuoja ne vienas būdas vaizdą paversti iš erdvinės srities į dažnių sritį, populiariausi jų – diskrečioji kosinusų transformacija (angl. *Discrete Cosine Transform, DCT*) ir diskrečioji bangelių transformacija (angl. *Discrete Wavelet Transform, DWT*). Abu metodai turi savų privalumų ir trūkumų.

1.4.3.1 Diskrečioji kosinusų transformacija

Vienas labiausiai praktikoje taikomų vaizdo apdorojimo metodų vaizdo dažnių srityje – diskrečioji kosinusų transformacija. DCT yra plačiai naudojama statinių vaizdų, vaizdų sekų apdorojime, yra realizuota diskrečioji kosinusų transformacija paremtų glaudinimo algoritmu – JPEG (angl. *Joint Photographic Experts Group*), MJPEG (angl. *Motion-JPEG*), MPEG (angl. *Moving Picture Experts Group*), DV (angl. *Digital Video*).

DCT metodas lyginant su erdvinės srities būdais yra patvaresnis ir labiau atsparus vaizdo manipuliacijoms, tokioms kaip žemo dažnio pikselių filtravimas (angl. *low pass filter*), kontrasto kontroliavimas (angl. *contrast adjustment*) ar suliejimo efekto taikymas (angl. *blur effect*). Kita vertus, diskrečiosios kosinusų transformacijos būdas nėra efektyvus, taikant geometrines vaizdo modifikacijas – pasukimo, apkirpimo, mastelio keitimo ir kt.

Dažniausiai literatūroje aptinkamas diskrečiosios kosinusų transformacijos apibrėžimas vienos dimensijos duomenų masyvui, kurio narių skaičius lygus N [8]:

$$c(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos \left[\frac{(2x+1)\pi u}{2N} \right]; \quad (2)$$

čia $u = 0, 1, 2, \dots, N-1$. Iš (2) formulės seka atvirkštinė transformacijos formulė:

$$f(x) = \sum_{u=0}^{N-1} \alpha(u) c(u) \cos \left[\frac{(2x+1)u\pi}{2N} \right]; \quad (3)$$

čia $x = 0, 1, 2, \dots, N-1$. Abejoms formulėms – (2) ir (3) – taikomos vienodos $\alpha(u)$ reikšmių sąlygos:

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}}, & \text{kai } u = 0 \\ \sqrt{\frac{2}{N}}, & \text{kai } u \neq 0 \end{cases} \quad (4)$$

Galiojant pirmajai sąlygai (4), vienos dimensijos masyvo diskrečioji kosinusų transformacija apskaičiuojama pagal formulę:

$$c(u=0) = \sqrt{\frac{1}{N}} \sum_{x=0}^{N-1} f(x) \cos \left[\frac{(2x+1)\pi u}{2N} \right] \quad (5)$$

Atlikus skaičiavimus pagal (5) formulę, gaunamas pirmasis transformacijos koeficientas, vadinamas DC koeficientu. Jo reikšmė yra lygi visų masyvo narių reikšmių vidurkiui. Likę koeficientai – AC – apskaičiuojami, kai $u = 1, 2, 3, \dots, N-1$.

Kadangi šiame darbe orientuojamasi į dvimatį vaizdą, vienmatės diskrečiosios kosinusų transformacijos principu (2) pritaikomas algoritmas dvimačio vaizdo dažnių sričiai [4]:

$$c(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left[\frac{(2x+1)u\pi}{2N} \right] \cos \left[\frac{(2y+1)v\pi}{2N} \right]; \quad (6)$$

čia $u, v = 0, 1, 2, \dots, N-1$ ir $x = 0, 1, 2, \dots, N-1$. Pirmojo ir likusių kitų transformacijos koeficientų suradimui taikomos tos pačios $\alpha(u)$ (7) ir $\alpha(v)$ (8) sąlygos, kaip ir (4):

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}}, & \text{kai } u = 0 \\ \sqrt{\frac{2}{N}}, & \text{kai } u \neq 0 \end{cases} \quad (7)$$

$$\alpha(v) = \begin{cases} \sqrt{\frac{1}{N}}, & \text{kai } v = 0 \\ \sqrt{\frac{2}{N}}, & \text{kai } v \neq 0 \end{cases} \quad (8)$$

(6) formule apskaičiuojamas vienas vaizdo, kuris suskaidytas blokais, transformacijos koeficientas (u, v) , naudojant originalaus vaizdo taškų reikšmių matricą. Taikant standartinį vaizdo skaidymą blokais, jo aukštis ir plotis lygus 8 taškams. Blokai diskrečiąja kosinusų transformacija apdorojami vienas po kito – iš kairės į dešinę, nuo viršaus į apačią. Suskaidžius vaizdą blokais 8×8 , pagal (6) formulę gaunama matrica A (9), iš kurios seka matrica B (10).

$$A = \frac{1}{2} \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \cos \frac{\pi}{16} & \cos \frac{3\pi}{16} & \cos \frac{5\pi}{16} & \cos \frac{7\pi}{16} & \cos \frac{9\pi}{16} & \cos \frac{11\pi}{16} & \cos \frac{13\pi}{16} & \cos \frac{15\pi}{16} \\ \cos \frac{2\pi}{16} & \cos \frac{6\pi}{16} & \cos \frac{10\pi}{16} & \cos \frac{14\pi}{16} & \cos \frac{18\pi}{16} & \cos \frac{22\pi}{16} & \cos \frac{26\pi}{16} & \cos \frac{30\pi}{16} \\ \cos \frac{3\pi}{16} & \cos \frac{9\pi}{16} & \cos \frac{15\pi}{16} & \cos \frac{21\pi}{16} & \cos \frac{27\pi}{16} & \cos \frac{33\pi}{16} & \cos \frac{39\pi}{16} & \cos \frac{45\pi}{16} \\ \cos \frac{4\pi}{16} & \cos \frac{12\pi}{16} & \cos \frac{20\pi}{16} & \cos \frac{28\pi}{16} & \cos \frac{36\pi}{16} & \cos \frac{44\pi}{16} & \cos \frac{52\pi}{16} & \cos \frac{60\pi}{16} \\ \cos \frac{5\pi}{16} & \cos \frac{15\pi}{16} & \cos \frac{25\pi}{16} & \cos \frac{35\pi}{16} & \cos \frac{45\pi}{16} & \cos \frac{55\pi}{16} & \cos \frac{65\pi}{16} & \cos \frac{75\pi}{16} \\ \cos \frac{6\pi}{16} & \cos \frac{18\pi}{16} & \cos \frac{30\pi}{16} & \cos \frac{42\pi}{16} & \cos \frac{54\pi}{16} & \cos \frac{66\pi}{16} & \cos \frac{78\pi}{16} & \cos \frac{90\pi}{16} \\ \cos \frac{7\pi}{16} & \cos \frac{21\pi}{16} & \cos \frac{35\pi}{16} & \cos \frac{49\pi}{16} & \cos \frac{63\pi}{16} & \cos \frac{77\pi}{16} & \cos \frac{91\pi}{16} & \cos \frac{105\pi}{16} \end{bmatrix} \quad (9)$$

$$B = \begin{bmatrix} 0,3536 & 0,3536 & 0,3536 & 0,3536 & 0,3536 & 0,3536 & 0,3536 & 0,3536 \\ 0,4904 & 0,4157 & 0,2778 & 0,975 & -0,975 & -0,2778 & -0,4157 & -0,4904 \\ 0,4619 & 0,1913 & -0,1913 & -0,4619 & -0,4619 & -0,1913 & 0,1913 & 0,4619 \\ 0,4157 & -0,0975 & -0,4904 & -0,2778 & 0,2778 & 0,4904 & 0,0975 & -0,4157 \\ 0,3536 & -0,3536 & -0,3536 & 0,3536 & 0,3536 & -0,3536 & -0,3536 & 0,3536 \\ 0,2778 & -0,4904 & 0,0975 & 0,4157 & -0,4157 & -0,0975 & 0,4904 & -0,2778 \\ 0,1913 & -0,4619 & 0,4619 & -0,1913 & -0,1913 & 0,4619 & -0,4619 & 0,1913 \\ 0,0975 & -0,2778 & 0,4157 & -0,4904 & 0,4904 & -0,4157 & 0,2778 & -0,0975 \end{bmatrix} \quad (10)$$

Prieš atliekant diskrečiąja kosinusų transformaciją pasirinktam vaizdo blokui, būtina kiekvieno taško reikšmę pritaikyti transformacijai skirtam reikšmių intervalui, t. y. koeficientus intervale $0 \leq x \leq 255$, kai x yra originalaus vaizdo taško reikšmė, paversti į koeficientus, kurių intervalas lygus $-128 \leq x \leq 127$, kai x yra originalaus vaizdo taško reikšmė. Tai galima atlikti iš kiekvienos originalaus vaizdo matricos (11) reikšmės atimant skaičių 128.

$$C = \begin{bmatrix} 139 & 144 & 149 & 153 & 155 & 155 & 155 & 155 \\ 144 & 151 & 153 & 156 & 159 & 156 & 156 & 156 \\ 150 & 155 & 160 & 163 & 158 & 156 & 156 & 156 \\ 159 & 161 & 162 & 160 & 160 & 159 & 159 & 159 \\ 159 & 160 & 161 & 162 & 162 & 155 & 155 & 155 \\ 161 & 161 & 161 & 161 & 160 & 157 & 157 & 157 \\ 162 & 162 & 161 & 163 & 163 & 157 & 157 & 157 \\ 162 & 162 & 161 & 161 & 161 & 158 & 158 & 158 \end{bmatrix} \quad (11)$$

Atlikus veiksmus gaunama matrica (12), paruošta diskrečiajai kosinusų transformacijai.

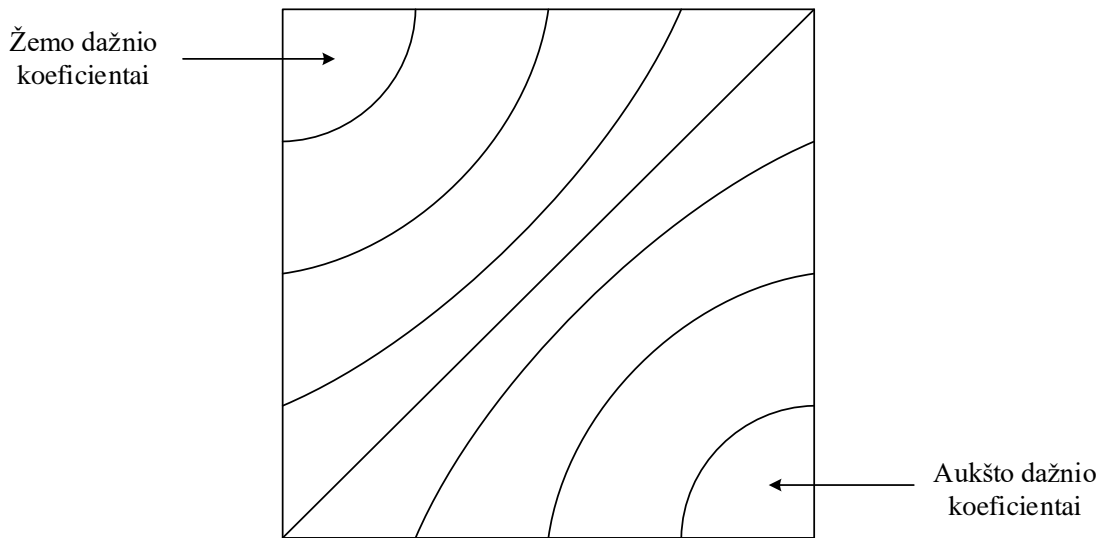
$$D = \begin{bmatrix} 11 & 16 & 21 & 25 & 27 & 27 & 27 & 27 \\ 16 & 23 & 25 & 28 & 31 & 28 & 28 & 28 \\ 22 & 27 & 32 & 35 & 30 & 28 & 28 & 28 \\ 31 & 33 & 34 & 32 & 32 & 31 & 31 & 31 \\ 31 & 32 & 33 & 34 & 34 & 27 & 27 & 27 \\ 33 & 33 & 33 & 33 & 32 & 29 & 29 & 29 \\ 34 & 34 & 33 & 35 & 35 & 29 & 29 & 29 \\ 34 & 34 & 33 & 33 & 33 & 30 & 30 & 30 \end{bmatrix} \quad (12)$$

Turint vaizdo bloko reikšmių matricą D (12), diskrečioji kosinusų transformacija atliekama pagal (13) formulę – sandauga tarp konstantų matricos (10), kurių reikšmės yra priklausomos nuo bloko dydžio ir originalaus vaizdo reikšmių matricos (12), yra padauginama iš transponuotos konstantų matricos (10).

$$E = BDB^T \quad (13)$$

$$E = \begin{bmatrix} 235,6250 & -1,0333 & -12,0809 & -5,2029 & 2,1250 & -1,6724 & -2,7080 & 1,3238 \\ -22,5904 & -17,4842 & -6,2405 & -3,1574 & -2,8557 & -0,0695 & 0,4342 & -1,1856 \\ -10,9493 & -9,2624 & -1,5758 & 1,5301 & 0,2029 & -0,9419 & -0,5669 & -0,0629 \\ -7,0816 & -1,9072 & 0,2248 & 1,4539 & 0,8963 & -0,0799 & -0,0423 & 0,3315 \\ -0,6250 & -0,8381 & 1,4699 & 1,5563 & -0,1250 & -0,6610 & 0,6088 & 1,2752 \\ 1,7541 & -0,2029 & 1,6205 & -0,3424 & -0,7755 & 1,4759 & 1,0410 & -0,9930 \\ -1,2825 & -0,3600 & -0,3169 & -1,4601 & -0,4900 & 1,7438 & 1,0758 & -0,7613 \\ -2,5999 & 1,5519 & -3,7628 & -1,8448 & 1,8716 & 1,2139 & -0,5679 & -0,4456 \end{bmatrix} \quad (14)$$

Gautas rezultatas (14), kuriame matricos narių reikšmės suapvalintos iki keturių skaitmenų po kablelio. Pagal jį galima teigti [3], jog matricoje, kurią sudaro 64 koeficientai c_{uv} , kai $u = 0,1, \dots, 7$ ir $v = 0,1, \dots, 7$, žemiausio dažnio koeficientai yra viršutinėje kairės bloko pusės dalyje, o dažninės dedamosios didėja artėjant prie didžiausio koeficiento, kurio pozicija lygi c_{77} . Vizualus dažnių koeficientų pasiskirstymas vaizdo bloke matomas 3 paveiksle.



3 pav. Dažnių koeficientų pasiskirstymas vaizdo bloke atlikus DCT

Atvirkštinė dvimačio vaizdo diskrečioji kosinusų transformacija išreiškiama iš (6) formulės:

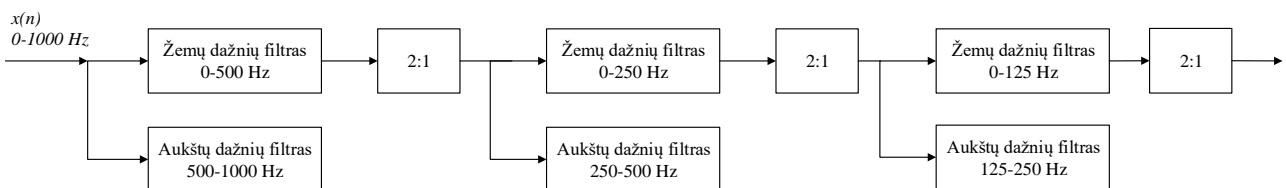
$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u) \alpha(v) c(u, v) \cos \left[\frac{(2x+1)u\pi}{2N} \right] \cos \left[\frac{(2y+1)v\pi}{2N} \right] \quad (15)$$

1.4.3.2 Diskrečioji bangelių transformacija

Diskrečioji bangelių transformacija – tai matematiškai išreikšta procedūra, naudojama statinių vaizdų hierarchiniam suskaidymui (angl. *decomposition*). Vienas iš diskrečiosios kosinusų transformacijos trūkumų yra tai, kad ji paremta vaizdo transformacija blokais – vaizdas suskaidomas 8×8 dydžio blokais, kuriems yra taikoma transformacija, todėl glaudinant vaizdą gali vizualiai pasimatyti blokinė vaizdo struktūra.

Diskrečioji bangelių transformacija pasižymi savybe suteikti dažninę signalo informaciją esant reikiamam laiko momentui, todėl plačiai taikoma vaizdo kodavimo bei glaudinimo programinėje įrangoje.

Bangelių transformacija atliekama signalo laiko srityje pritaikant aukštų ir žemų dažnių filtras, taip pašalinant tam tikrus dažnius. Pritaikius aukšto dažnio filtras, gaunami vaizdo detalių koeficientai, žemų dažnių filtrų rezultatas – aproksimacijos koeficientai. Kadangi signalas yra skaidomas į dvi dalis, nes yra taikomi dviejų tipų filtrai, išsaugoma kas antra reikšmė (žr. 4 pav.), siekiant išvengti rezultatų duomenų pertekliaus [15].



4 pav. Signalų suskaidymo, pritaikant aukštų ir žemų dažnių filtras, schema

Atliekant dvimačio vaizdo diskrečiąją bangelių transformaciją, vaizdas suskaidomas į atskiras, tarpusavyje nepersidengiančias sritis (žr. 5 pav.), kuriose atvaizduojama informacija, gauta pritaikius

aukšto ir žemo dažnio filtrus kiekvienai originalaus vaizdo dimensijai ir sumažinus dvigubai jo dydį. Šie veiksmai gali būti atliekami pakartotinai tiek kartų, kokio lygio diskrečiąją bangelių transformaciją norima atlikti [13].

LL	HL
LH	HH

5 pav. Vieno lygio diskrečiosios bangelių transformacijos vaizdo suskaidymas

Kiekviena sritis apskaičiuojama pagal atskiras formules ir turi individualias charakteristikas: LL (16) dalis atspindi originalų vaizdą, tik jo skiriamoji geba sumažinta du kartus, LH (17) srityje vaizduojamos vertikalios originalaus vaizdo dedamosios, HL (18) – horizontalios originalaus vaizdo dedamosios, HH (19) – aukšto dažnio originalaus vaizdo dedamosios.

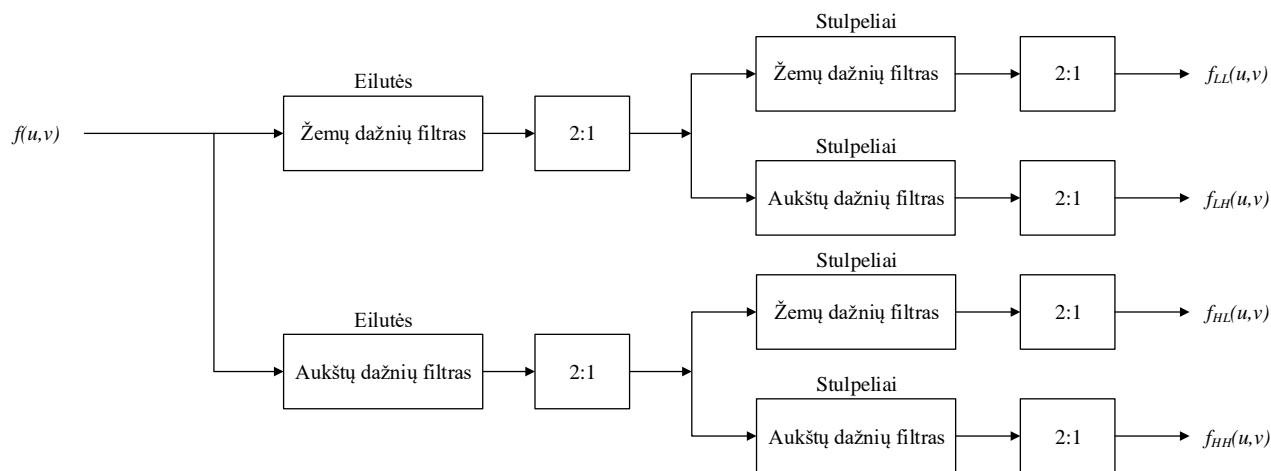
$$f_{LL}^J(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} g(x)g(y)f_{LL}^{J-1}(2u-x)(2v-y), \quad (16)$$

$$f_{LH}^J(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} g(x)h(y)f_{LL}^{J-1}(2u-x)(2v-y), \quad (17)$$

$$f_{HL}^J(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} h(x)g(y)f_{LL}^{J-1}(2u-x)(2v-y), \quad (18)$$

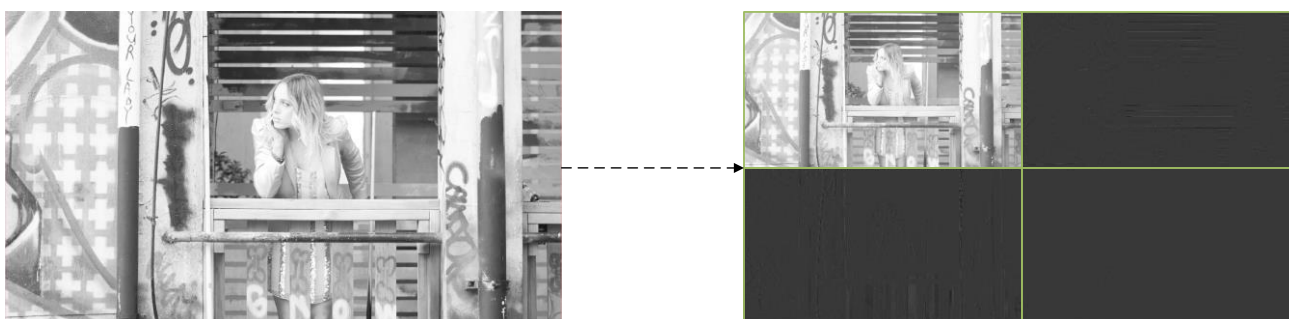
$$f_{HH}^J(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} h(x)h(y)f_{LL}^{J-1}(2u-x)(2v-y), \quad (19)$$

čia J yra dvimatės diskrečiosios bangelių transformacijos lygio skaičius, N – vaizdo filtruojamų reikšmių kiekis, $g(x)$ ir $g(y)$ – rezultatas, gautas pritaikius atitinkamai žemo ir aukšto dažnių filtrus. Kai $J = 0$, gautas rezultatas $f_{LL}^0(u, v) = f(u, v)$ yra lygus originaliam pirminiam vaizdai. Dvimačio vaizdo vieno lygio diskrečiosios bangelių transformacijos schema vaizduojama 6 paveiksle. Pirmiausia aukštų ir žemų dažnių filtrais apdorojamos vaizdą sudarančios matricos eilutės, kurių išsaugoma kas antra reikšmė, vėliau filtruojami stulpeliai.



6 pav. Dvimačio vaizdo vieno lygio diskrečiosios bangelių transformacijos schema

Iš originalaus vaizdo į atskiras sritis suskaidytas vaizdas po vieno lygio diskrečiosios bangelių transformacijos vaizduojamas 7 paveiksle.



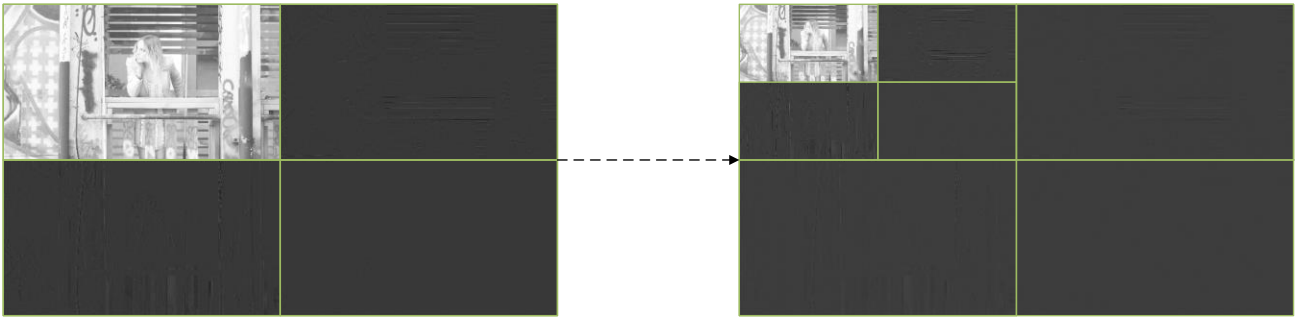
7 pav. Vaizdas po vieno lygio diskrečiosios bangelių transformacijos

Dviejų lygių diskrečiosios bangelių transformacijos vizualus vaizdo suskaidymas atrodo analogiškai 5 pav. vaizduojamam suskaidymui, tik į aukšto ir žemo dažnių ketvirčius skaidomas LL skiltyje esantis vaizdas (žr. 8 pav.).

LL2	HL2	HL1
LH2	HH2	
LH1		HH1

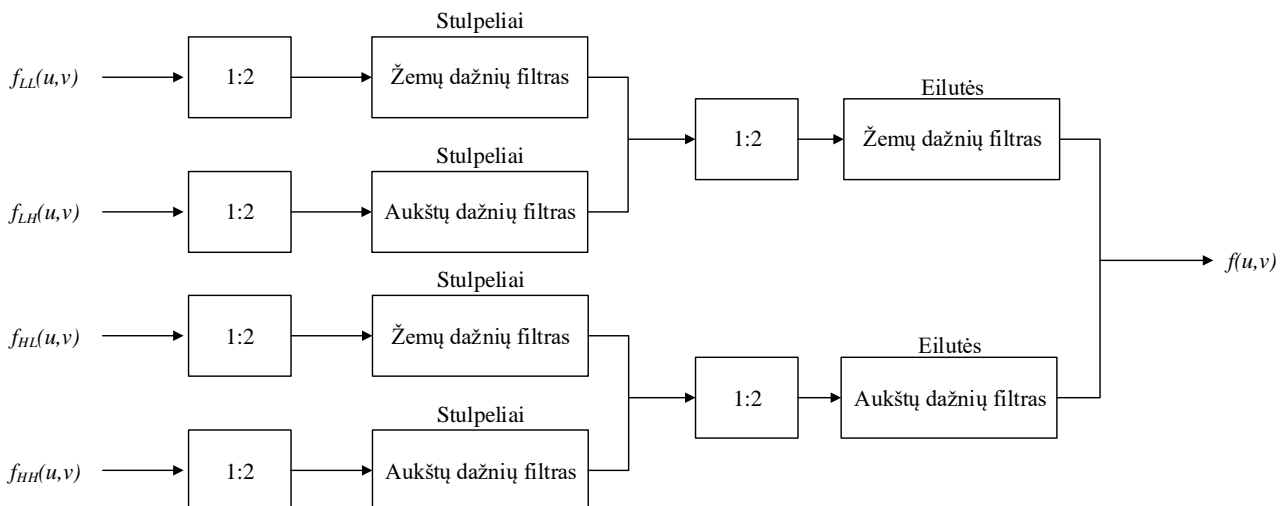
8 pav. Dviejų lygių diskrečiosios bangelių transformacijos vaizdo suskaidymas

Vaizdo pokytis, vieno lygio diskrečiosios bangelių transformacijos vaizdą apdorojus antruoju lygiu, vaizduojamas 9 paveiksle.



9 pav. Vaizdas po antro lygio diskrečiosios bangelių transformacijos

Iš visų sudedamųjų dalių, žemo dažnio skiltis – LL, jei atlikta vieno lygio diskrečioji bangelių transformacija, ar LL2, jei atlikta dviejų lygių diskrečioji bangelių transformacija – daugiausiai sudaro originalaus vaizdo duomenų, tad įterpiant vandens ženklą į šį lauką išpildomas vienas iš vaizdo žymėjimo vandens ženklų reikalavimų – patvarumas [21]. Kita vertus, aukšto dažnio srityse HH saugomos vaizduojamų objektų kraštinės, tekstūros, tad įterpus vandens ženklą į šią sritį vizualus pokytis žmogui nebūtų matomas. Norint suskaidyti vaizdą iš suskaidytų dalių įterpus vandens ženklą, atliekama atvirkštinė diskrečioji bangelių transformacija – vaizdo atstatymas (angl. *reconstruction*).



10 pav. Dvimačio vaizdo vieno lygio atvirkštinės diskrečiosios bangelių transformacijos schema

Vaizdo atstatymas (žr. 10 pav.) vykdomas taikant atvirkštinę diskrečiąją bangelių transformaciją ir priešingus veiksmus vaizdo skaidymui ir filtravimui atvirkštine seka.

Lyginant apžvelgtus vaizdo transformacijų metodus, diskrečioji bangelių transformacija pranoksta diskrečiąją kosinusų transformaciją geresne vizualia vaizdo kokybe, suglaudinus vaizdą išvengiama blokinės vaizdo struktūros artefaktų [14].

Kitame skyriuje apžvelgiami vaizdo manipuliacijų tipai, kurie yra taikomi siekiant modifikuoti minėtais metodais vandens ženklais pažymėtus vaizdus.

1.5. VAIZDO GLAUDINIMO STANDARTAI

Pagrindinis vaizdo glaudinimo tikslas - kaip įmanoma labiau sumažinti vaizdo failo užimamą dydį kaip įmanoma mažiau paveikiant vaizdo kokybę. Per didelis vaizdo suspaudimo laipsnis gali vizualiai pakeisti vaizdą, esant įvairiems iškraipymams, artefaktams.

Egzistuoja dviejų tipų vaizdo glaudinimas – nenuostolingas (angl. lossless compression) ir nuostolingas (angl. lossy compression). Nenuostolingas glaudinamas dažniausiai naudojamas tose srityse, kuriose informacijos nepraradimas ir išsaugojimas yra labai svarbus. Tuo tarpu nuostolingą tipo glaudinimo algoritmų yra daugiausia, kadangi jais mėginama sumažinti perduodamos informacijos kiekį ir taip prisidėti prie vykdomų procesų optimizacijos.

Esant spalvotam vaizdui, tarkime RGB modelio, kiekvienas taškas susideda iš trijų dedamųjų – raudonos, žalios, mėlynos spalvų, kurių kiekviena atitinka 8 bitus informacijos. Bendras vaizdo intensyvumas lygus trijų dedamųjų sumai $I = R + G + B$. Glaudinant vaizdą yra mažinamas tam tikrų vaizdo taškų spalvų dedamųjų intensyvumas. Priklausomai nuo turimos informacijos kiekio, kuriam didelę reikšmę turi vaizdo raiška, glaudinimui taikomi skirtingi duomenų srauto dažniai – jų palyginimas pateiktas 3 lentelėje [2].

3 lentelė. Skirtingoms vaizdo raiškoms taikomi glaudinimo duomenų srautai

Vaizdo raiška	Kadrų per sekundę kiekis	Informacijos kiekis spalvos dedamajai	Neglaudinto vaizdo duomenų srauto dažnis	Glaudinto vaizdo duomenų srauto dažnis
352x240	15	8 bitai	30,41 Mbps	384 kbps
352x240	30	8 bitai	60,43 Mbps	1,5 – 4 Mbps
720x480	30	8 bitai	248,83 Mbps	3 – 8 Mbps
1280x720	59,94	8 bitai	1,33 Gbps	20 Mbps

Žurnalo „Streaming Media“ 2015 metais atliktos apklausos, kurioje dalyvavo 335 skaitytojai, duomenimis, buvo siekiama išsiaiškinti šių laikų vaizdo glaudinimo metodų taikymo tendencijas ir pažvelgti, kokios technologijos bus taikomos po penkerių metų [19]. Remiantis apklausos rezultatais, 2015 metais vaizdo transliavimui internete bei glaudinimui labiausiai (78 proc.) taikomas H.264/AVC (MPEG-4 Advanced Video Coding) algoritmas, tuo tarpu patobulintai jo versijai H.265/HEVC (High Efficiency Video Coding) algoritmui prognozuojamas didelis šuolis per 2015–2020 metų laikotarpį (49 proc.). H.264 vaizdo glaudinimo algoritmas, pripažintas tarptautiniu ISO standartu, yra plačiai taikomas antžeminės, kabelinės, internetinės televizijos transliavime, pateikiant įvairių interaktyvias paslaugas [25].

H.265/HEVC vaizdo glaudinimo algoritmas pagal pavadinimo prasmę yra vadinamas aukšto efektyvumo metodu. Literatūroje teigiama [22], jog šiuo algoritmu suspaudus tokiu pačiu duomenų srauto dažniu kaip ir H.264, išlaikoma ta pati kokybė, tik gauto failo dydis gali būti iki dvigubai mažesnis.

1.6. ATAKŲ PRIEŠ VANDENS ŽENKLUS TIPAI

Norint apsaugoti teises į skaitmeninę intelektinę nuosavybę, nepakanka vaizdinės medžiagos tik pažymėti vandens ženklais. Jie privalo atitikti nepastebimumo (angl. *imperceptibility*) bei patvarumo (angl. *robustness*) kriterijus – jų savybės leidžia atremti atakas, kuriomis gali būti bandoma pašalinti vandens ženklus iš duomenų srauto. Atakos gali būti keleto tipų – priverstinis vaizdo medžiagą sudarančių kadru pašalinimas (angl. *frame dropping*), jų sukeitimas (angl. *frame swapping*), įvairios vaizdo transformacijos – apkirpimas (angl. *cropping*), mastelio keitimas (angl. *scaling*) ar pasukimas (angl. *rotating*).

Galimo vaizdo kadru sukeitimo ar panaikinimo atakų grėsmę įmanoma pašalinti vandens ženklus įkomponuojant į visus vaizdo medžiagoje esančius kadrus. Tokiu būdu, net ir esant situacijai, kai yra panaudojamos dviejų minėtų tipų atakos, tai nesudaro didelės grėsmės bendrai vaizdo kadru vandens ženklų visumai ar galimybės panaikinti skaitmeninį parašą, kadangi jie tarpusavyje paprastai ryšio neturi ir jokia seka neegzistuoja.

Išnagrinėjus vaizdo žymėjimo vandens ženklais metodų veikimo principus, galimus vaizdo manipuliacijos būdus, galima projektuoti programinę įrangą, kuri įterptų vandens ženklą į vaizdo duomenų srautą.

2. VAIZDO ŽYMĖJIMO VANDENS ŽENKLAIS PĮ PROJEKTAVIMAS

Šiame skyriuje aprašomas programinės įrangos projektavimas ir pasiruošimas eksperimentui. Nurodomos eksperimento atlikimui pasirinktos priemonės, funkciniai ir nefunkciniai reikalavimai bei parenkami palyginimo parametrai, pagal kuriuos eksperimento metu bus lyginamas vandens ženklų išliekamumas.

Eksperimento tikslas – atlikus vandens ženklų pažymėto vaizdo manipuliacijas, apdorojus vaizdo medžiagą skirtingais glaudinimo algoritmais ir išanalizavus į vaizdą įterpto vandens ženklų išliekamumo pokytį, nustatyti efektyvesnį vaizdo glaudinimo algoritmą.

Siekiant tikslo, eksperimentas suskaidomas į kelis etapus:

- DCT metodo pritaikymas vandens ženklų įterpimui į statinį vaizdą,
- DWT metodo pritaikymas vandens ženklų įterpimui į statinį vaizdą,
- Vaizdo manipuliacijoms atsparesnio metodo parinkimas,
- Atsparesnio metodo pritaikymas vandens ženklų įterpimui į vaizdų seką,
- Skirtingų vaizdo manipuliacijų pritaikymas vandens ženklų pažymėtoms vaizdų sekoms,
- Skirtingų glaudinimo algoritmų pritaikymas vandens ženklų pažymėtoms vaizdų sekoms,
- Vandens ženklų atpažįstamumo parametrų parinkimas,
- Vandens ženklų išliekamumo pokyčio tyrimas, atlikus vaizdo manipuliacijas ir glaudinimą,
- Efektyvesnio vaizdo glaudinimo algoritmo išskyrimas, pagal vandens ženklų išliekamumo pokyčio tyrimo rezultatus.

Svarbu pažymėti, kad specifinėmis vaizdo manipuliacijomis nesiekama kardinaliai pakeisti vandens ženklų pažymėtos vaizdinės medžiagos kokybės, tačiau siekiama, kad iš vaizdo išgaunamas vandens ženklas taptų vizualiai neatpažįstamas.

2.1. REIKALAVIMŲ SPECIFIKACIJA

Projektuojant programinę įrangą svarbu nustatyti funkcinis ir nefunkcinis reikalavimus – jie apibrėžiami šioje skyriaus dalyje.

2.1.1. Funkciniai reikalavimai

Užduoties įvykdymui naudojant programinę įrangą turi būti išpildytos šios sąlygos:

- reikalingas neglaudintas video failas, kuris bus glaudinamas bei žymimas vandens ženklų,
- skirtingų skiriamųjų gebų vandens ženklai, kurie bus įterpti į skirtingų skiriamųjų gebų video failų duomenų srautą,
- operacinėje sistemoje turi būti įdiegta „FFmpeg“ programinė įranga, skirta pažymėtų vaizdų apjungimui bei video failo glaudinimui,

- turi būti sukurtos direktorijos, pažymėtų, suglaudintų, išgautų vaizdų, video failų saugojimui.

2.1.2. Nefunkciniai reikalavimai

Vaizdinės medžiagos žymėjimo vandens ženklais programinė įranga priklausomai nuo taikomo algoritmo turi individualiai specifinius reikalavimus, todėl nėra visiems metodams taikomų bendrų reikalavimų. Iš jų visų galima išskirti keletą esminių bruožų, į ką labiausiai orientuojamasi atliekant vaizdo medžiagos žymėjimą vandens ženklais.

Skaidrumas (angl. *perceptual transparency*) – algoritmas, naudojamas vandens ženklui į duomenų srautą įterpti, turi atitikti šią sąlygą, jog nebūtų pakenkta originaliai vaizdinei medžiagai, kuri yra naudojama kaip pirminis šaltinis. Atlikus procesą, gautas rezultatas, t. y. vaizdas su įterptu vandens ženklu, žmogaus akiai gali būti nematomas arba nežymiai matomas – atsižvelgiant į tai, ar yra siekiama, jog vandens ženklas būtų iš dalies matomas ir žiūrovas žinotų, jog vaizdinė medžiaga yra apsaugota, ar norima to vizualiai neparodyti. Vandens ženklo įterpimas į vaizdo klipo duomenų srautą negali daryti įtakos vaizdo kokybei. Jei įkomponuojamas vandens ženklas yra vizualiai nematomas, originali ir paženklinta vaizdo medžiaga turi atrodyti vienodai. Vis dėlto, net jei ir būtų atliktos tam tikros modifikacijos, vartotojas negalėtų atlikti palyginimo neturėdamas originalios, nepažymėtos vandens ženklais, vaizdo medžiagos.

Patvarumas (angl. *robustness*) – į vaizdą įterpus vandens ženklą, jo patvarumas pasižymi savybe išlikti savo vietoje nepaisant to, kokiais būdais yra transformuojamas vaizdas. Jei vaizdinė medžiaga yra glaudinama taikant nuostolingą (angl. *lossy*) glaudinimo algoritmą, mažinamas duomenų srauto dydis, naudojami filtrai, vandens ženklas išlieka kaip informacija, įrodanti vaizdinės medžiagos autentiškumą. Vandens ženklo patvarumo savybė priklauso nuo vaizdo žymėjimo vandens ženklais algoritmo ir transformacijų, kuriomis yra modifikuojamas pažymėtas vaizdas.

2.2. EKSPERIMENTO PRIEMONĖS

Šiam eksperimentui atlikti suprogramuotas įrankis „MATLAB“ programavimo aplinkoje. Ji pasirinkta todėl, kad „MATLAB“ realizuota daug naudingų matematinių funkcijų, palengvinančių programavimo procesą. Taip pat „MATLAB“ programavimo aplinka pasižymi universalumu, kadangi į programos kodą galima įterpti ir sisteminės komandinės eilutės (angl. *command line*) užklausas.

Būtent tokiu būdu buvo panaudota dar viena šio eksperimento priemonė – „FFmpeg“. Tai yra nemokamas, daugiaplatformis (angl. *cross-platform*) įrankis, skirtas vaizdo bei garso apdorojimui, įrašinėjimui ir transliavimui internetu. Komandinės eilutės pagrindu paremtas įrankis, kaip rašoma oficialioje „FFmpeg“ svetainėje internete, „gali suglaudinti, transliuoti, pritaikyti vaizdinį ar garsinį filtrą ir groti bene visų formatų failus, kokius žmonės ar kompiuteriai yra sukūrę“ [23]. „FFmpeg“ taip pat pasižymi stabiliu programinės įrangos veikimu ir komandų vykdymu skirtingose operacinėse

sistemose, „Windows“, „Linux“, „Mac OS X“ ir kt., esant įvairioms operacinių sistemų konfigūracijoms ar kompiuterių architektūroms.

2.3. Palyginimo parametrai

Atliekant šį eksperimentą svarbiausias objektas yra vandens ženklas ir jo išliekamumas. Glaudinant vaizdą ar manipuliuojant juo yra prarandama dalis informacijos, todėl natūralu, kad įterpus vandens ženklą į vaizdinės medžiagos duomenų srautą ir apdorojus videoklipą glaudinimo algoritmu ar atlikus vaizdo modifikacijas iš apdoroto videoklipo išgautas vandens ženklas nebus lygus pirminiam variantui. Vandens ženklo pokytį galima įvertinti palyginant originalaus ir išgauto vandens ženklų skirtumus arba panašumus. Lyginant paveiktą ir originalų vaizdus matomi vizualūs skirtumai, – iškraipymai (angl. *distortions*), vaizdo triukšmo (angl. *image noise*) dalys – tačiau žmogaus akis negali tiksliai nustatyti, kiek vaizdo yra pakitę. Dėl šios priežasties svarbu parinkti metodus, kuriuos naudojant matematiškai išreiškiamas vaizdo kokybės pokytis.

2.3.1. MSE

Vidutinis kvadratinis nuokrypis (angl. *Mean Squared Error, MSE*) statistikoje taikomas kaip rodiklis, nusakantis dviejų vaizdų skirtumus (20), šio eksperimento atveju – pirminio ir išgauto vandens ženklų reikšmių kitimo intervalų skirtumą:

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (x(i, j) - y(i, j))^2 \quad (20)$$

2.3.2. PSNR

Didžiausio signalo ir triukšmo santykis (angl. *Peak signal-to-noise ratio, PSNR*) – inžinerinis terminas, nusakantis santykį tarp didžiausio signalo stiprumo ir triukšmo, paveikiančio signalo tikslumą (21). PSNR dažnai yra naudojamas suglaudintų vaizdų pokyčiui išmatuoti. Matavimo vienetas – decibelas (dB). PSNR pagrindu yra laikomas vidutinio kvadratinio nuokrypio vertinimas.

$$PSNR = 10 \log_{10} \frac{(2^n - 1)^2}{\sqrt{MSE}} \quad (21)$$

2.3.3. SSIM

Struktūrinio panašumo indekso metodas pagrįstas matematinio modeliu (22), tačiau yra vienas iš dviejų vaizdo kokybės įvertinimo metodų, paremtų žmogaus regos sistema (angl. *Human Visual System, HVS*). SSIM vaizdų palyginimo metodas yra plačiai taikomas medicinoje – remiantis radiologų vertinimais [10], SSIM rodiklis geriausiai parodo pokytį tarp lyginamų vaizdų.

Lyginamų vaizdų vertinimas vykdomas remiantis trimis kriterijais: vaizdo skaistumo (23), kontrasto (24) ir vaizdo struktūros (25) palyginimais. SSIM reikšmė kinta nuo 0 iki 1. Esant rezultatui,

lygiam 0, tarp lyginamų vaizdų nėra jokios koreliacijos. Esant SSIM reikšmei, lygiai 1, lyginami vaizdai yra identiški.

$$SSIM(f, g) = l(f, g)c(f, g)s(f, g), \quad (22)$$

$$l(f, g) = \frac{2\mu_f\mu_g + C_1}{\mu_f^2 + \mu_g^2 + C_1}, \quad (23)$$

$$c(f, g) = \frac{2\sigma_f\sigma_g + C_2}{\sigma_f^2 + \sigma_g^2 + C_2}, \quad (24)$$

$$s(f, g) = \frac{\sigma_{fg} + C_3}{\sigma_f\sigma_g + C_3}, \quad (25)$$

čia μ_f ir μ_g yra atitinkamai originalaus ir modifikuoto vaizdų vidutinio skaistumo reikšmės. Vaizdų kontrasto palyginimas paremtas dviejų vaizdų standartinio nuokrypio reikšmėmis – σ_f ir σ_g . Apskaičiuojant vaizdų struktūrinį skirtumą, būtina abiejų vaizdų kovariacija σ_{fg} – tų pačių reikšmių skirtinguose vaizduose nuokrypių nuo reikšmių vidurkių sandaugos vidurkis. Konstantos C_1 , C_2 ir C_3 turi būti teigiamos, siekiant išvengti nulinės reikšmės vardiklyje [11].

Atlikus palyginimo parametrų apžvalgą galima teigti, kad visi gali būti taikomi eksperimente, kadangi jų suteikiama informacija apie vandens ženklo išliekamumą yra naudinga ir įvairiapusė. 3 skyriuje atlikus vandens ženklais pažymėtos vaizdo sekos manipuliacijas ir glaudinimą, remiantis vandens ženklo išliekamumo palyginimo parametrais bus išskirtas efektyvesnis vaizdo glaudinimo algoritmas.

2.3.4. Vandens ženklo įterpimo stiprumo parinkimas

Ne kiekvienas vandens ženklas, kuriuo apsaugomas vaizdas, pasižymi patvarumo savybe ir gali būti atsparus vaizdo glaudinimui bei įvairioms manipuliacijoms. Svarbu, koku stiprumu vandens ženklas įterpiamas į duomenų srautą. Parenkant per didelę šio parametro reikšmę gali būti sugadinamas apsaugomas vaizdas, t. y. vizualiai matytis vaizdo iškraipymai, išryškinami tam tikri vaizdo taškai (žr. 11 pav. ir 12 pav.) ir kt. Siekiant neparodyti, jog vaizdą norima apsaugoti, svarbu parinkti tinkamą reikšmę, kad po įvairių vaizdo modifikacijų išgautas vandens ženklas būtų artimas pirminiam vandens ženklui.



11 pav. Vaizdas, kai vandens ženklų stiprumas lygus 0,008



12 pav. Vaizdas, kai vandens ženklų stiprumas lygus 0,3

Vaizdo žymėjimo vandens ženklais metodams, aprašytiems literatūroje [17][9][6][7], įterpimo stiprumas yra taikomas skirtingas, todėl patariama maksimalią reikšmę parinkti lyginant originalaus ir apsaugoto vaizdo pokyčio skirtumą. Šiam tikslui galima panaudoti ir SSIM rodiklį, kuris naudojamas ir vandens ženklų išliekamumo pokyčiui įvertinti. 4 lentelėje pateikiamos šiam eksperimentui rekomenduojamos vandens ženklų įterpimo į vaizdą reikšmės.

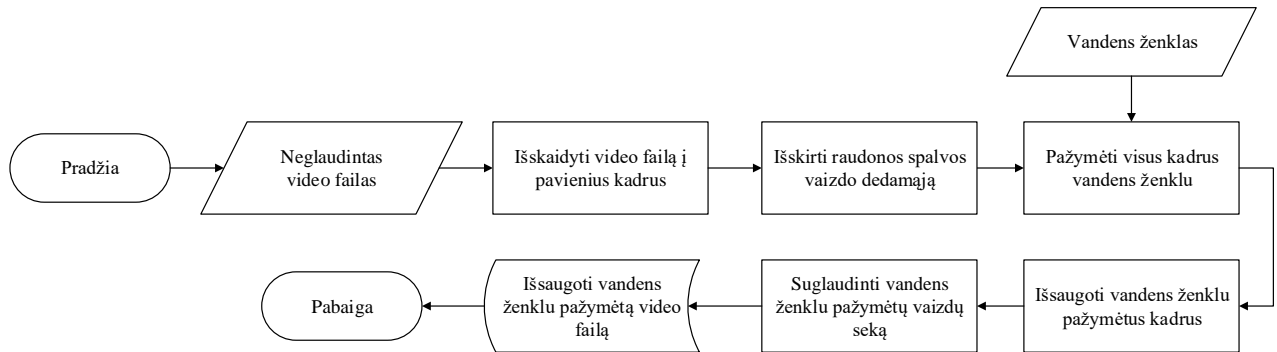
4 lentelė. Rekomenduojamos vandens ženklų stiprumo reikšmės

Vandens ženklų įterpimo būdas	Vandens ženklų įterpimo stiprumas	Vaizdo SSIM	Vandens ženklų SSIM
Diskrečioji kosinusų transformacija	3	0,99912	0,7231
Diskrečioji bangelių transformacija	0,008	0,99734	0,90616

3. VAIZDO ŽYMĖJIMO VANDENS ŽENKLAIS PĮ REALIZACIJA

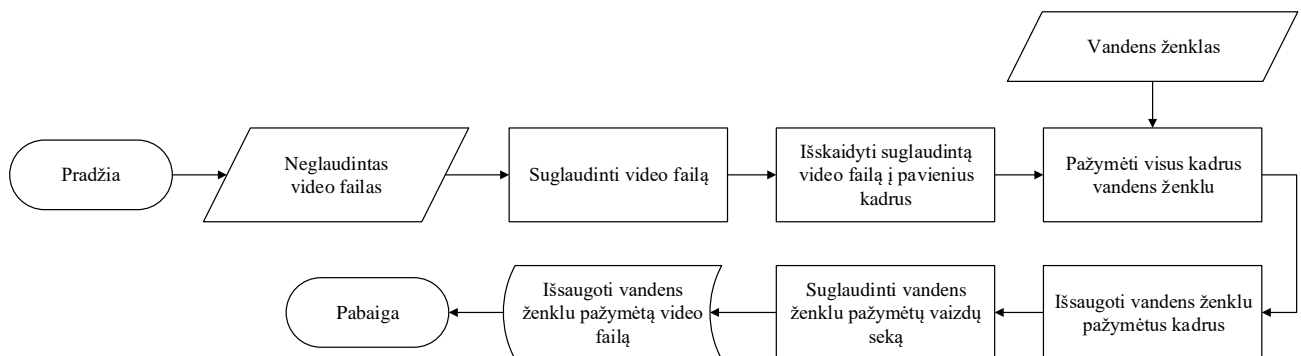
Šiame skyriuje aprašoma suprojektuoto programinio įrankio realizacija, atliekami vandens ženklų įterpimo į vaizdo duomenų srautą eksperimentai.

Pagal 2.1.1 skyriuje aprašytus nefunkcinius reikalavimus nubraižomos eksperimento vykdymui skirtos schemas.



13 pav. Neglaudinto video failo žymėjimo vandens ženklu schema

13 paveiksle vaizduojama videoklipo žymėjimo vandens ženklu proceso veiksmų seka, kurios metu vandens ženklas yra įterpiamas į kiekvieno neglaudinto video failo kadro raudonos spalvos dedamąją, o galutinis rezultatas – iš pavienių kadrų sujungta vaizdų seka – suglaudinas pasirinktu algoritmu.



14 pav. Suglaudinto video failo žymėjimo vandens ženklu schema

Optimaliausias procesas matomas 14 paveiksle – prieš pažymint vaizdą vandens ženklu, video failas yra suglaudinas pasirinktu glaudinimo algoritmu bei duomenų srautu. Tokiu atveju vandens ženklas tampa atsparesnis vaizdo manipuliacijoms ir videoklipo glaudinimui, kadangi įterpus vandens ženklą į neglaudintą videoklipą ir suglaudinus duomenų srautą 10 ar daugiau kartų, prarandama daug informacijos, kuri svarbi ir vandens ženklų išgavimo procese.

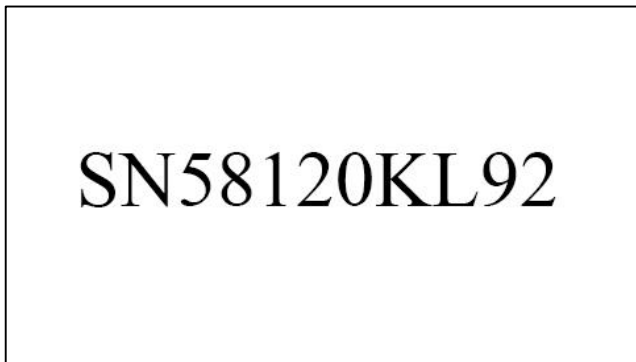
3.1. PIRMASIS EKSPERIMENTO ETAPAS

Prieš pagrindinę projekto realizacijos dalį atliekamas pirminis eksperimentas, kurį sudarys du etapai:

- 1) Statinio vaizdo žymėjimas vandens ženklu ir vandens ženklo išgavimas,
- 2) Vandens ženklu pažymėtos vaizdų sekos glaudinimas ir vandens ženklo išgavimas.

3.1.1. Statinio vaizdo žymėjimas vandens ženklų ir vandens ženklų išgavimas

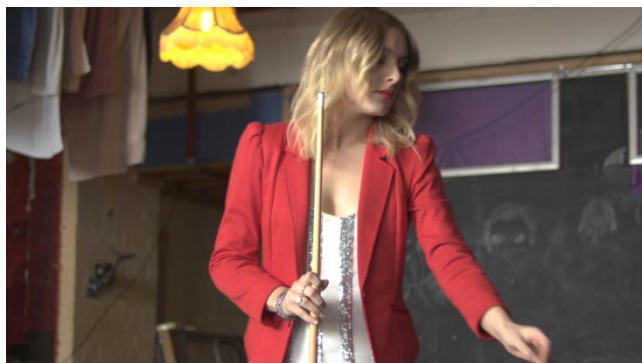
Siekiant patikrinti programinio įrankio veikimą, vandens ženklų pažymėtas statinis vaizdas ir palyginta, ar išgautas vandens ženklas atpažįstamai panašus į pirminį vandens ženklą. Eksperimentui buvo panaudoti du skirtingų vaizdo sekų kadrai (žr. 16 pav. ir 17 pav.) bei vandens ženklas (žr. 15 pav.).



15 pav. Vandens ženklas



16 pav. Vaizdas iš pirmosios vaizdų sekos

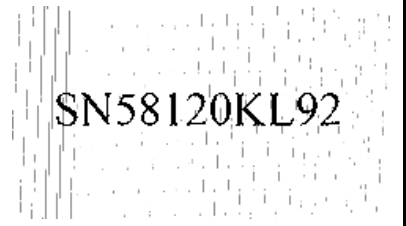
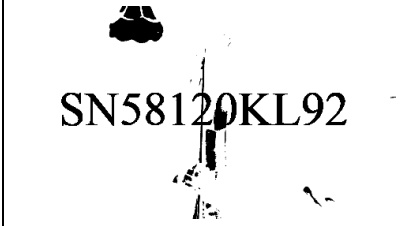


17 pav. Vaizdas iš antrosios vaizdų sekos

Vandens ženklų įterpimo metu taikant diskrečiąją kosinusų transformaciją, išgautame vandens ženkle matomi neįžymūs vaizdo defektai (žr. 5 lent.), tačiau vandens ženkle atsispindinti informacija įžiūrima ir aiškiai suprantama. Iš diskrečiosios bangelių transformacijos būdu apsaugoto vaizdo išgautas vandens ženklas taip pat turi iškreipymų – taip yra todėl, kad vandens ženklas yra įterpiamas į vidutinio vaizdo dažnių sritį, į kurią patenka ir originaliuose vaizduose (žr. 16 pav. ir 17 pav.) išsiskiriantys šviesios spalvos objektai.

5 lentelė. Iš skirtingais metodais apsaugotų statinių vaizdų išgautų vandens ženklų palyginimas

Vaizdo sekos nr.	Diskrečioji kosinusų transformacija		Diskrečioji bangelių transformacija	
	SSIM	Išgautas vandens ženklas	SSIM	Išgautas vandens ženklas
1	0,7231	SN58120KL92	0,9062	SN58120KL92

Vaizdo sekos nr.	Diskrečioji kosinusų transformacija		Diskrečioji bangelių transformacija	
	SSIM	Išgautas vandens ženklas	SSIM	Išgautas vandens ženklas
2	0,5701		0,8839	

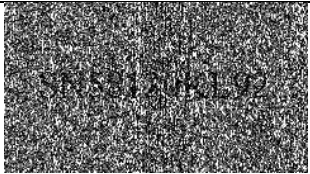
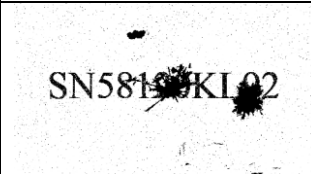
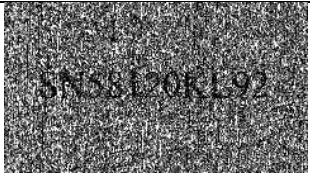
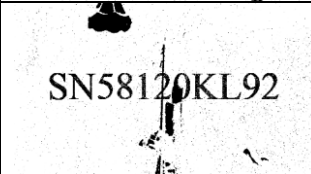

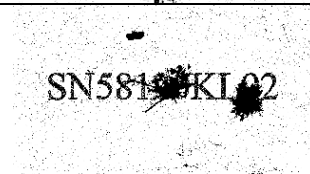
Remiantis SSIM rezultatais, rodiklio gaunama reikšmė priklauso nuo vaizdų, kurie yra tarpusavyje lyginami, skaistumo, struktūros, kontrasto – matoma, jog iš pirmojo vaizdo išgautų vandens ženklų SSIM rodikliai yra geresni už antrojo vaizdo rezultatus. Vis dėlto, atsižvelgiant į SSIM rodiklį, vandens ženklo išliekamumas yra didesnis taikant diskrečiąją bangelių transformaciją.

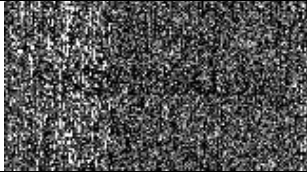
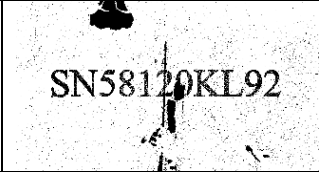
3.1.2. Vandens ženklų pažymėtos vaizdų sekos glaudinimas ir vandens ženklo išgavimas

Šio eksperimento etapo metu siekiama išbandyti ar dviem apžvelgtais algoritmais – diskrečiąja kosinusų transformacija ir diskrečiąja bangelių transformacija – įterpus vandens ženklą į vaizdą, suglaudinus vaizdų seką ir išgavus iš vaizdo vandens ženklą gaunamas rezultatas tinkamas tolimesniam eksperimento vykdymui. Vandens ženklu apsaugotam vaizdui netaikomos jokios vaizdo manipuliacijos.

Vaizdo žymėjimui vandens ženklu panaudotos po dvi dviejų skirtingų raiškų vaizdų sekos (žr. 16 pav. ir 17 pav.), įterpus vandens ženklą (žr. 15 pav.) jos suglaudintos dviem skirtingais glaudinimo algoritmais H.264 ir H.265, jiems pritaikant skirtingus duomenų srautus (žr. 6 lent.).

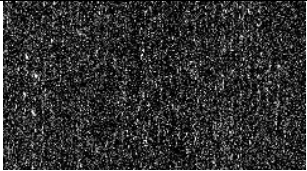

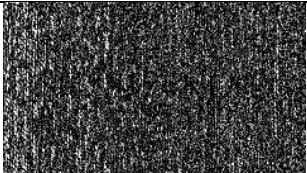
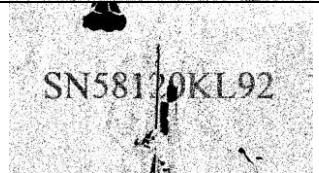
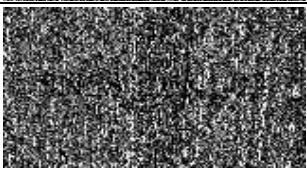
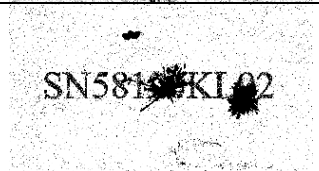

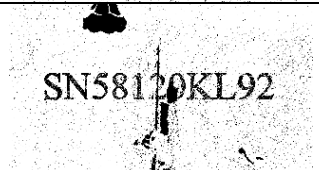
6 lentelė. Iš H.264 algoritmu suglaudinto vaizdo išgauto vandens ženklo palyginimas

Vaizdų seka	Vaizdo raiška	Duomenų srautas (Kb/s)	Diskrečioji kosinusų transformacija		Diskrečioji bangelių transformacija	
			SSIM	Išgautas vandens ženklas	SSIM	Išgautas vandens ženklas
1	1920x1080	32000	0,0385		0,5690	
2	1920x1080	32000	0,0432		0,5598	
1	1280x720	16000	0,0348		0,3480	

			Diskrečioji kosinusų transformacija		Diskrečioji bangelių transformacija	
Vaizdų seka	Vaizdo raiška	Duomenų srautas (Kb/s)	SSIM	Išgautas vandens ženklas	SSIM	Išgautas vandens ženklas
2	1280x720	16000	0,0314		0,3924	

Atsižvelgiant į rezultatus, kuriuose palyginami išgauti vandens ženklai, į vaizdų seką buvo įterpti taikant diskrečiąją kosinusų transformaciją ir diskrečiąją bangelių transformaciją, galima daryti išvadą, jog pagal SSIM rodiklį (žr. 6 lent. ir 7 lent.) diskrečioji bangelių transformacija yra pranašesnė už diskrečiąją kosinusų transformaciją visais atvejais, nepriklausomai nuo vaizdo rezoliucijos, duomenų srauto ar glaudinimo algoritmo. Tokiu atveju diskrečiosios kosinusų transformacijos metodas nėra tinkamas eksperimentui, kurio metu bus manipuluojamos suglaudintų vaizdų sekos.

7 lentelė. Iš H.265 algoritmu suglaudinto vaizdo išgauto vandens ženklo palyginimas

			Diskrečioji kosinusų transformacija		Diskrečioji bangelių transformacija	
Vaizdų seka	Vaizdo raiška	Duomenų srautas (Kb/s)	SSIM	Išgautas vandens ženklas	SSIM	Išgautas vandens ženklas
1	1920x1080	32000	0,0051		0,0451	
2	1920x1080	32000	0,0118		0,0730	
1	1280x720	16000	0,0332		0,2733	
2	1280x720	16000	0,0485		0,3249	

3.2. EKSPERIMENTO DUOMENYS

Šio eksperimento metu naudojantis 2.2 skyriuje paminėtomis priemonėmis vandens ženklų pažymėtos dvi skirtingos vaizdų sekos, iš kurių kiekviena buvo trijų skirtingų skiriamųjų gebų, atitinkamai kiekvienai iš jų pagal skiriamąją gebą buvo paskirtas vandens ženklas (žr. 8 lent.). Jo dydis lygus horizontaliųjų originalaus vaizdo dedamųjų srities, gautos atlikus dviejų lygių diskrečiąją bangelių transformaciją, dydžiui. Vandens ženklas yra įterpiamas į šią vaizdo sritį.

8 lentelė. Eksperimento duomenys vaizdų sekos žymėjimui vandens ženklų

Vaizdo sekos skiriamoji geba, pikseliai	Vandens ženklo skiriamoji geba, pikseliai	Duomenų srautas, Kb/s		
		64000	32000	24000
1920 x 1080	480 x 270	64000	32000	24000
1280 x 720	320 x 180	20000	16000	12000
640 x 360	160 x 90	10000	6000	2000

Siekiant ištirti išgauto vandens ženklo išliekamumą, taikant skirtingus paženklinto vaizdo poveikio būdus, fiksuojami rodikliai, nusakantys vandens ženklo pasikeitimą, lyginant su originaliu vandens ženklu. Glaudinant vaizdų seką naudojami du glaudinimo algoritmai – H.264 ir H.265, pritaikant kiekvieną iš jų su trimis skirtingomis duomenų srauto reikšmėmis. Šis parametras, kaip ir vandens ženklo dydis, priklauso nuo vaizdo skiriamosios gebos, todėl pagal vaizdo taškų skaičių kiekvienai raiškai priskirti atitinkami įvesties dažniai.

Vaizdų sekos glaudinamos „FFmpeg“ komandų pagalba, valdant specialius savo funkciją atliekančius parametrus.

1. Videoklipas suglaudinamas taip:

```
ffmpeg -y -r 25 -i video1.avi -c:v libx264 -b:v 2000k video2.mp4
```

- 9 lentelėje pateikiamos pirmosios komandos parametrų reikšmės.

9 lentelė. Videoklipo suglaudinimo komandos parametrų reikšmės

Parametro pavadinimas	Parametro reikšmė
-y	Igalina išsaugoti išvesties failą vietoje kito failo tokiu pačiu pavadinimu, jei toks egzistuoja.
-r 25	Nustato išvesties failo kadrų per sekundę skaičių.
-i video1.avi	Įvesties failo pavadinimas.
-c:v libx264	Parentomas vaizdo glaudinimo algoritmas, libx264 yra „FFmpeg“ aplinkoje naudojama H.264 glaudinimo algoritmo biblioteka.
-b:v 2000k	Pastovus duomenų srautas (Kb/s).
video2.mp4	Išvesties failo pavadinimas.

2. Į atskirus vaizdo kadrus videoklipas suskaidomas taip:

```
ffmpeg -r 25 -i video2.avi -f image2 %03d.bmp
```

- 10 lentelėje pateikiamos antrosios komandos parametrų reikšmės.

10 lentelė. Videoklipo išskaidymo į atskirus vaizdo kadrus komandos parametrų reikšmės

Parametro pavadinimas	Parametro reikšmė
-r 25	Nustato išvesties failo kadrų per sekundę skaičių.
-i video2.avi	Įvesties failo pavadinimas.
-f image2	Igalina videoklipo skaidymo į atskirus kadrus funkciją.
%03d.bmp	Iš trijų simbolių susidedantis išvesties failo pavadinimas, kurio reikšmė yra didėjanti ir lygi kadro numeriui vaizdo sekoje.

3. Atskiri vaizdo kadrai į bendrą videoklipą apjungiami taip:

```
ffmpeg -y -r 25 -i %03d.bmp -c:v libx264 -b:v 2000k video3.mp4
```

- 11 lentelėje pateikiamos trečiosios komandos parametrų reikšmės.

11 lentelė. Vaizdų sekos sujungimo į vieną videoklipą komandos parametrų reikšmės

Parametro pavadinimas	Parametro reikšmė
-y	Išgalina išsaugoti išvesties failą vietoje kito failo tokiu pačiu pavadinimu, jei toks egzistuoja.
-r 25	Nustato išvesties failo kadro per sekundę skaičių.
-i %03d.bmp	Iš trijų simbolių susidedantis įvesties failo pavadinimas, kurio reikšmė yra didėjanti ir lygi kadro numeriui vaizdo sekoje.
-c:v libx264	Parinkamas vaizdo glaudinimo algoritmas, libx264 yra „FFmpeg“ aplinkoje naudojama H.264 glaudinimo algoritmo biblioteka.
-b:v 2000k	Pastovus duomenų srautas (Kb/s).
video3.mp4	Išvesties failo pavadinimas.

Panaudojant minėtas „FFmpeg“ komandas su skirtingais parametrų duomenimis (žr. 8 lentelė.) vaizdų sekos yra suglaudintos, o jas išanalizavus, apskaičiuojami vandens ženklų išliekamumo vertinimo rodikliai, kurie yra pateikiami 3.3 skyriuje.

3.3. ANTRASIS EKSPERIMENTO ETAPAS

Šio eksperimento metu buvo glaudinamos dvi skirtingos vaizdų sekos, vandens ženklą įterpiančios diskrečiosios bangelių transformacijos metodu. 18 paveiksle matomas vienas iš vandens ženklų pažymėtos vaizdų sekos kadro, kuriame vizualiai nematyti vaizdo iškraipymų.



18 pav. Glaudintas vandens ženklų pažymėtas vaizdas

Suglaudintą apsaugotą vaizdą ir iš jo išgavus vandens ženklą, remiantis gautais rezultatais (žr. 12 lent.), pritaikius skirtingas duomenų srauto reikšmes galima teigti, jog pagal tris vandens ženklų išliekamumo rodiklius – MSE, PSNR ir SSIM – H.264 algoritmas 92,6 proc. visų bandymų yra pranašesnis už H.265 algoritmą, t. y. H.264 algoritmu suglaudintą apsaugotą vaizdą vandens ženklų išliekamumas yra geresnis. 12 lentelėje žalsva spalva pažymėti rodikliai yra pranašesni, lyginant

gautus H.264 ir H.265 algoritmų rezultatus. MSE parametras, kuris pagal prasmę geresnis artėjant reikšmei link nulio, palyginime žymimas žalsvai esant mažesnei reikšmei.

12 lentelė. Iš nemodifikuoto vaizdo išgauto vandens ženklų rezultatai

			Vaizdo glaudinimo algoritmas					
			H.264			H.265		
Vaizdų sekos nr.	Vaizdo skiriamoji geba	Duomenų srautas, Kb/s	MSE	PSNR, dB	SSIM	MSE	PSNR, dB	SSIM
1	1920x1080	24000	0,0474	13,2466	0,3538	0,1612	7,9256	0,0310
2	1920x1080	24000	0,0520	12,8363	0,3927	0,1358	8,6714	0,0486
1	1920x1080	32000	0,0400	13,9756	0,5690	0,1223	9,1246	0,0451
2	1920x1080	32000	0,0464	13,3360	0,5598	0,1080	9,6677	0,0730
1	1920x1080	64000	0,0355	14,4931	0,8356	0,0516	12,8745	0,2853
2	1920x1080	64000	0,0431	13,6571	0,6765	0,0553	12,5759	0,3400
1	1280x720	12000	0,0679	11,6818	0,1704	0,0771	11,1320	0,1297
2	1280x720	12000	0,0696	11,5769	0,2325	0,0769	11,1403	0,1949
1	1280x720	16000	0,0515	12,8807	0,3480	0,0566	12,4691	0,2733
2	1280x720	16000	0,0556	12,5515	0,3924	0,0615	12,1086	0,3249
1	1280x720	20000	0,0454	13,4292	0,5138	0,0495	13,0540	0,4299
2	1280x720	20000	0,0510	12,9241	0,5058	0,0537	12,7035	0,4260
1	640x360	2000	0,1926	7,1531	0,0333	0,1877	7,2665	0,0409
2	640x360	2000	0,2271	6,4375	0,0759	0,2301	6,3814	0,0589
1	640x360	6000	0,0863	10,6400	0,1565	0,0882	10,5451	0,1515
2	640x360	6000	0,1043	9,8177	0,1801	0,1109	9,5503	0,1810
1	640x360	10000	0,0621	12,0659	0,4167	0,0636	11,9678	0,3794
2	640x360	10000	0,0757	11,2101	0,3564	0,0787	11,0378	0,3069

Siekiant ištirti vandens ženklų išliekamumo pokytį ir efektyvesnį vaizdo glaudinimo algoritmą, apsaugotam vaizdai pritaikyta viena iš vaizdo manipuliacijų – „įrėminimas“, vaizdo viršuje ir apačioje uždedant horizontalias juodos spalvos juostas. Tokio tipo atakos populiarios tarp interneto „piratų“, pritaikant jas įvairiems komerciniams filmams (žr. 19 pav.).



19 pav. Vandens ženklų pažymėtas vaizdas su horizontaliomis juodomis juostomis

Atlikus šią eksperimento dalį, kurios rezultatai pateikiami 13 lentelėje, galima tvirtinti, jog pagal tris vandens ženklo išliekamumo rodiklius, iš modifikuoto vaizdo išgautas vandens ženklas yra panašiausias į pirminį vaizdą glaudinant H.264 algoritmu – 85,2 proc. bandymų buvo efektyvesni už H.265. Kita vertus, glaudinant 640x360 skiriamosios gebos vaizdą 2000 ir 6000 Kb/s duomenų srautu H.265 algoritmas davė geresnius rodiklius, tačiau ryškaus pranašumo neįžvelgiama.

13 lentelė. Iš juodomis juostomis pažymėto vaizdo išgauto vandens ženklo rezultatai

			Vaizdo glaudinimo algoritmas					
			H.264			H.265		
Vaizdų sekos nr.	Vaizdo skiriamoji geba	Duomenų srautas, Kb/s	MSE	PSNR, dB	SSIM	MSE	PSNR, dB	SSIM
1	1920x1080	24000	0,1524	8,1702	0,1905	0,2505	6,0126	0,0283
2	1920x1080	24000	0,1506	8,2213	0,2327	0,2265	6,4492	0,0433
1	1920x1080	32000	0,1409	8,5121	0,3571	0,2198	6,5795	0,0393
2	1920x1080	32000	0,1407	8,5169	0,4016	0,2064	6,8520	0,0589
1	1920x1080	64000	0,1323	8,7850	0,6986	0,1594	7,9742	0,1437
2	1920x1080	64000	0,1354	8,6846	0,5568	0,1571	8,0395	0,1841
1	1280x720	12000	0,2260	6,4599	0,0897	0,2298	6,3864	0,0851
2	1280x720	12000	0,2179	6,6177	0,1201	0,2260	6,4594	0,1121
1	1280x720	16000	0,2052	6,8789	0,1789	0,2088	6,8037	0,1480
2	1280x720	16000	0,2007	6,9752	0,2164	0,2090	6,7990	0,1763
1	1280x720	20000	0,1951	7,0965	0,2844	0,2008	6,9726	0,2378
2	1280x720	20000	0,1928	7,1499	0,3246	0,1987	7,0173	0,2340
1	640x360	2000	0,4605	3,3677	0,0235	0,4502	3,4658	0,0342
2	640x360	2000	0,4926	3,0752	0,0252	0,4802	3,1858	0,0253
1	640x360	6000	0,3835	4,1621	0,0643	0,3781	4,2234	0,0850

			Vaizdo glaudinimo algoritmas					
			H.264			H.265		
Vaizdų sekos nr.	Vaizdo skiriamoji geba	Duomenų srautas, Kb/s	MSE	PSNR, dB	SSIM	MSE	PSNR, dB	SSIM
2	640x360	6000	0,3797	4,2060	0,0851	0,3879	4,1128	0,0847
1	640x360	10000	0,3556	4,4899	0,1751	0,3552	4,4948	0,1707
2	640x360	10000	0,3472	4,5944	0,1607	0,3529	4,5231	0,1358

Trečioje eksperimento dalyje vandens ženklų apsaugotas glaudintas vaizdas buvo modifikuotas atsitiktinio triukšmo dalelėmis, esant 0,001 intensyvumui. T. y. skirtingos skiriamosios gebos vaizdams paveikiama $d * n$ taškų, čia d yra intensyvumo parametras, n – vaizdą sudarančių taškų kiekis. Modifikuotas vaizdas matomas 20 paveiksle.



20 pav. Vandens ženklų pažymėtas vaizdas, paveiktas „Salt&Pepper“ filtru

Remiantis 14 lentelėje pateiktais atlikto eksperimento rezultatais, vandens ženklų išliekamumo pokytis, glaudinant „Salt&Pepper“ filtru paveiktą apsaugotą vaizdą, mažesnis taikant H.264 algoritmą – 88,9 proc. gautų vertinimo rodiklių rezultatų yra pranašesni už H.265 algoritmą, kurio vos 11,1 proc. rezultatų – glaudinant 640x360 skiriamosios gebos videoklipą pritaikius 2000 Kb/s duomenų srautą – buvo efektyvesni.

14 lentelė. Iš „Salt&Pepper“ filtru manipuluoto vaizdo išgauto vandens ženklų rezultatai

			Vaizdo glaudinimo algoritmas					
			H.264			H.265		
Vaizdų sekos nr.	Vaizdo skiriamoji geba	Duomenų srautas, Kb/s	MSE	PSNR, dB	SSIM	MSE	PSNR, dB	SSIM
1	1920x1080	24000	0,0743	11,2924	0,1155	0,2259	6,4604	0,0198
2	1920x1080	24000	0,0792	11,0102	0,1304	0,2084	6,8114	0,0260

			Vaizdo glaudinimo algoritmas					
			H.264			H.265		
Vaizdų sekos nr.	Vaizdo skiriamoji geba	Duomenų srautas, Kb/s	MSE	PSNR, dB	SSIM	MSE	PSNR, dB	SSIM
1	1920x1080	32000	0,0564	12,4834	0,2341	0,1915	7,1787	0,0259
2	1920x1080	32000	0,0635	11,9722	0,2425	0,1757	7,5524	0,0338
1	1920x1080	64000	0,0439	13,5731	0,5180	0,0883	10,5389	0,0840
2	1920x1080	64000	0,0515	12,8838	0,4318	0,0902	10,4459	0,1028
1	1280x720	12000	0,1073	9,6941	0,0648	0,1273	8,9527	0,0548
2	1280x720	12000	0,1089	9,6294	0,0841	0,1274	8,9474	0,0702
1	1280x720	16000	0,0805	10,9441	0,1209	0,0924	10,3452	0,0901
2	1280x720	16000	0,0826	10,8300	0,1531	0,0965	10,1535	0,1149
1	1280x720	20000	0,0640	11,9357	0,2042	0,0748	11,2621	0,1573
2	1280x720	20000	0,0704	11,5232	0,2211	0,0787	11,0418	0,1676
1	640x360	2000	0,2661	5,7497	0,0188	0,2540	5,9512	0,0194
2	640x360	2000	0,2921	5,3444	0,0263	0,2900	5,3755	0,0285
1	640x360	6000	0,1243	9,0555	0,0638	0,1346	8,7094	0,0619
2	640x360	6000	0,1443	8,4087	0,0831	0,1578	8,0191	0,0819
1	640x360	10000	0,0814	10,8954	0,1832	0,0872	10,5966	0,1412
2	640x360	10000	0,0972	10,1220	0,1820	0,1046	9,8065	0,1568

Ketvirtoji eksperimento dalis – suliejimo „Gaussian“ filtro pritaikymas vandens ženklų pažymėtam vaizdai. 21 paveiksle matomame vaizde ryškaus efekto poveikio nematyti, tačiau net ir vizualiai nematomas poveikis turi įtakos vandens ženklo išliekamumui.



21 pav. Vandens ženklų pažymėtas vaizdas, paveiktas „Gaussian“ filtru

Pagal gautus rezultatus (žr. 15 lent.) H.264 vaizdo glaudinimo algoritmas taip pat pranašesnis už H.265 algoritmą, glaudinant apsaugotą vaizdą po „Gaussian“ filtro manipuliacijos. 88,9 proc. taikant

H.264 algoritmą gautų rezultatų, lyginant vandens ženklo išliekamumo rodiklius, buvo efektyvesni už H.265.

15 lentelė. Iš „Gaussian“ filtru manipuluoto vaizdo išgauto vandens ženklo rezultatai

			Vaizdo glaudinimo algoritmas					
			H.264			H.265		
Vaizdų sekos nr.	Vaizdo skiriamoji geba	Duomenų srautas, Kb/s	MSE	PSNR, dB	SSIM	MSE	PSNR, dB	SSIM
1	1920x1080	24000	0,1101	9,5813	0,0511	0,1982	7,0293	0,0240
2	1920x1080	24000	0,1182	9,2725	0,0743	0,1867	7,2890	0,0324
1	1920x1080	32000	0,1063	9,7342	0,1067	0,1767	7,5282	0,0283
2	1920x1080	32000	0,1063	9,7342	0,1067	0,1753	7,5633	0,0365
1	1920x1080	64000	0,0800	10,9710	0,1082	0,1253	9,0192	0,0446
2	1920x1080	64000	0,1000	9,9994	0,1242	0,1317	8,8052	0,0611
1	1280x720	12000	0,1252	9,0250	0,0518	0,1373	8,6221	0,0484
2	1280x720	12000	0,1516	8,1924	0,0703	0,1694	7,7102	0,0563
1	1280x720	16000	0,1022	9,9048	0,0766	0,1171	9,3155	0,0614
2	1280x720	16000	0,1380	8,6027	0,0940	0,1523	8,1730	0,0730
1	1280x720	20000	0,0904	10,4363	0,1012	0,1045	9,8086	0,0822
2	1280x720	20000	0,1307	8,8372	0,1178	0,1435	8,4306	0,0905
1	640x360	2000	0,2406	6,1872	0,0221	0,2228	6,5202	0,0285
2	640x360	2000	0,3091	5,0988	0,0329	0,3006	5,2195	0,0324
1	640x360	6000	0,1513	8,2018	0,0490	0,1600	7,9577	0,0507
2	640x360	6000	0,2428	6,1476	0,0524	0,2523	5,9803	0,0469
1	640x360	10000	0,1207	9,1831	0,0882	0,1304	8,8477	0,0793
2	640x360	10000	0,2249	6,4795	0,0586	0,2393	6,2111	0,0503

IŠVADOS

1. Atlikus eksperimentą pastebėta, kad vizualus vaizdo pokytis atlikus žymėjimą vandens ženklų priklauso nuo vandens ženklo įterpimo stiprumo koeficiento. Parinkus per didelę jo reikšmę apsaugotame vaizde matomi vizualūs iškraipymai, artefaktai. Optimalios parinktos vandens ženklo įterpimo stiprumo vertės diskrečiai kosinusų transformacijai ir diskrečiai bangelių transformacijai lygios atitinkamai 3 ir 0,008.
2. Pagal išgauto vandens ženklo išliekamumo vertinimo SSIM rodiklį, diskrečiosios bangelių transformacijos metodas yra tinkamesnis statinio vaizdo žymėjimui nei diskrečiosios kosinusų transformacijos metodas. Lyginant iš dviejų skirtingų apsaugotų vaizdų išgautą vandens ženklo išliekamumą, pagal SSIM rodiklį skirtingiems vaizdams pritaikyta diskrečioji bangelių transformacija atitinkamai 18,31 proc. ir 31,38 proc. buvo pranašesnė už diskrečiąją kosinusų transformaciją.
3. Vaizdo žymėjimas taikant diskrečiosios kosinusų transformacijos metodą, pagal iš vaizdo išgauto vandens ženklo išliekamumo vertinimo rodiklį SSIM, neatsparus vaizdo glaudinimui. Priklausomai nuo eksperimento metu panaudotų vaizdų, jų skiriamosios gebos, parinkto vaizdo glaudinimo algoritmo ir duomenų srauto, lyginant abu metodus diskrečiosios bangelių transformacijos efektyvumo pranašumas svyruoja nuo 4 proc. iki 53,05 proc.
4. Glaudinant diskrečiosios bangelių transformacijos metodu vandens ženklų pažymėtas vaizdų sekas ir vertinant išgauto vandens ženklo išliekamumą, H.264 glaudinimo algoritmas yra efektyvesnis už H.265 algoritimą. Pagal abiem algoritmais suglaudintų nemodifikuotų ir manipuluotų vaizdų rezultatus, 87,5 proc. PSNR reiškinių yra didesnės taikant H.264 algoritimą. Tokiu atveju, esant didesnėms pikinėms suglaudinto vaizdo signalo reikšmėms, glaudinant yra prarandama mažiau informacijos, kas atsispindi ir vertinant MSE rodiklio rezultatus – 87,5 proc. visų bandymų vidutinė kvadratinė paklaida buvo mažesnė nei taikant H.265 algoritimą.

LITERATŪRA

1. ANDERSON, R.J. - PETITCOLAS, F.A.P. On the limits of steganography. In *IEEE Journal on Selected Areas in Communications* . 1998. Vol. 16, no. 4, p. 474–481. .
2. BHASKARAN, V. *Image and video compression standards : algorithms and architectures*. . Boston : Kluwer Academic Publishers, 1995. ISBN 0792395913 (alk. paper).
3. CABEEN, K. - GENT, P. Discrete Cosine Transform. In *Image Compression and the Discrete Cosine Transform* . 1999. p. 1–11. .
4. GONGE, S.S. - BAKAL, J.W. ROBUST DIGITAL WATERMARKING TECHNIQUE BY USING DCT AND SPREAD SPECTRUM. In *International Journal of Electrical* . 2013. no. 12, p. 2320–2084. .
5. GREENSTEIN, J. Copyright Issues May Stall Digital Videodisc Debut. In *Los Angeles Times* [interaktyvus]. 1996. [žiūrėta 2016-04-20]. Prieiga per internetą: <http://articles.latimes.com/1996-06-03/business/fi-11385_1_million-dvd-movie>.
6. HUI-FEN, H. Digital Watermarking Algorithm Based on DWT and Chaotic Scrambling. In *Proceedings of 14th Youth Conference on Communication* . 2009. .
7. JASSIM, T. et al. Two Levels Block Based Wavelet Watermarking Algorithm for Still Colour Images. In *2nd BCS International IT Conference* . 2014. .
8. KHAYAM, S.A. The Discrete Cosine Transform (DCT): Theory and Application. In *Components* [interaktyvus]. 2003. Prieiga per internetą: <http://www.lokminglui.com/DCT_TR802.pdf>.
9. KIM, K.-S. et al. Blind Image Watermarking Scheme in DWT-SVD Domain. In *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007)* [interaktyvus]. [s.l.]: IEEE, 2007. p. 477–480. [žiūrėta 2016-04-10]. Prieiga per internetą: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4457752>>.
10. KOWALIK-URBANIAK, I. et al. The quest for „diagnostically lossless“ medical image compression: a comparative study of objective quality metrics for compressed medical images. In MELLO-THOMS, C.R. - KUPINSKI, M.A.Sud. *Proceedings of SPIE - The International Society for Optical Engineering* [interaktyvus]. 2014. p. 903717. [žiūrėta 2016-04-29]. Prieiga per internetą: <<https://ece.uwaterloo.ca/~z70wang/publications/spie14.pdf>>.
11. LAI, K.W. - DEWI, D.E.O. *Medical imaging technology : reviews and computational applications*. . ISBN 9789812875402.
12. MINTZER, F. et al. Effective and ineffective digital watermarks. In *Proceedings of International Conference on Image Processing* . 1997. Vol. 3. .
13. NIKOLAIDIS, A. - PITAS, I. Asymptotically optimal detection for additive watermarking in the DCT and DWT domains. In *IEEE Transactions on Image Processing* [interaktyvus]. 2003. Vol. 12, no. 5, p. 563–571. Prieiga per internetą: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1203149>>.
14. PARASHAR, P. et al. A Survey: Digital Image Watermarking Techniques. In *International Journal of Signal Processing Image Processing and Pattern Recognition* [interaktyvus]. 2014. Vol. 7, no. 6, p. 111–124. [žiūrėta 2016-03-18]. . Prieiga per internetą: <<http://dx.doi.org/10.14257/ijcip.2014.7.6.10>>.
15. PRADHAN, M. *Applications for digital signals : in digital speech processing, digital image processing*

and consumer electronics. . [s.l.]: Llumina Press, 2004. 271 p. ISBN 9781595263513.

16. SAMUELSON, P. Digital media and the law. In *Communications of the ACM* . 1991. Vol. 34, no. 10, p. 23–28. .

17. ŞENOL AHMET;DİNÇER KIVANÇ;ELBAŞI ERSİN;SEVER, H. A Block Size Analysis for Blocked Discrete Wavelet Watermarking. In . 2014. p. 77–82. .

18. STOBBS, G. The Digital Millennium Copyright Act. In *Multimedia Security Technologies for Digital Rights Management* . 2006. p. 457–482. ISBN 9780123694768.

19. STREAMING MEDIA Encoding 2020: Experts Predict the Future of Video Encoding. In [interaktyvus]. [žiūrėta 2016-04-13]. Prieiga per internetą: <<http://www.streamingmedia.com/Articles/Editorial/Featured-Articles/Encoding-2020-Experts-Predict-the-Future-of-Video-Encoding-106619.aspx>>.

20. TAYLOR, J.H. *DVD demystified*. . [s.l.]: McGraw-Hill, 2001. 691 p. ISBN 9780071389440.

21. TIWARI, N. et al. Digital watermarking using DWT and DES. In *2013 3rd IEEE International Advance Computing Conference (IACC)* [interaktyvus]. [s.l.]: IEEE, 2013. p. 1100–1102. Prieiga per internetą: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6514380>>.

22. WANG, P.S. *From computing to computational thinking*. . 253 p. ISBN 9781482217650.

23. About Ffmpeg. In [interaktyvus]. [žiūrėta 2016-02-19]. Prieiga per internetą: <<https://ffmpeg.org/about.html>>.

24. ABOUT THE DIGITAL WATERMARKING ALLIANCE. In [interaktyvus]. [žiūrėta 2016-03-03]. Prieiga per internetą: <<http://www.digitalwatermarkingalliance.org/about.asp>>.

25. AVC, List of video services using H.264/MPEG-4. In [interaktyvus]. [žiūrėta 2016-04-03]. Prieiga per internetą: <https://en.wikipedia.org/wiki/List_of_video_services_using_H.264/MPEG-4_AVC>.

26. Digimarc® Guardian for Images. In [interaktyvus]. [žiūrėta 2016-03-03]. Prieiga per internetą: <<https://www.digimarc.com/support/product/digimarc-guardian-for-images-support/faq/>>.

PRIEDAI

Programinio įrankio kodas:

dwt_embed.m

```
warning('off','all'); % išjungti įspėjančius pranešimus
clear all; % panaikinti egzistuojančius objektus, atlaisvinti atmintį
close all; % panaikinti egzistuojančias figūras
clc; % išvalyti komandų langą

raw_frames_dir = 'C:\Matlab\DWT\Data\raw_frames'; % originalių vaizdų direktorija
rawImages = dir(fullfile(raw_frames_dir,'*.bmp')); % originalūs vaizdai su plėtiniumi BMP
watermarked_frames_dir = 'C:\Matlab\DWT\Data\watermarked_frames'; % vandens ženklų pažymėtų vaizdų
direktorija
delete('C:\Matlab\DWT\Data\raw_frames\*.bmp'); % ištrinti visus originalius vaizdus su plėtiniumi BMP
delete('C:\Matlab\DWT\Data\watermarked_frames\*.bmp'); % ištrinti visus vandens ženklų pažymėtus
vaizdus su plėtiniumi BMP

tic % laikmačio pradžia
display([' datestr(now,13) ']: Glaudinamas originalus video failas ...'])
system('ffmpeg -y -r 25 -i Data\raw1-1.avi -c:v libx264 -b:v 16000k Data\raw-comp.mp4'); % originalaus
video failo glaudinimo komanda
t=toc; % laikmačio pabaiga
display([' datestr(now,13) ']: Video failas sėkmingai suglaudintas per '
num2str(datestr(datetime(0,0,0,0,0,t),'HH:MM:SS'))]);

tic % laikmačio pradžia
display([' datestr(now,13) ']: Suglaudintas video failas skaidomas į pavienius vaizdus ...'])
system('ffmpeg -r 25 -i Data\raw-comp.mp4 -f image2 Data\raw_frames\%03d.bmp'); % suglaudinto video
failo skaidymo į pavienius vaizdus komanda

flag = true;
while flag
    disp([' datestr(now,13) ']: FFMPEG.exe procesas vis dar aktyvus.]);
    flag = isprocess('ffmpeg.exe'); % "isprocess" funkcija tikrina, ar procesas 'ffmpeg.exe' vis dar vykdo
komandą
    pause(1); % tikrinti kas 1 sekundę
end

disp([' datestr(now,13) ']: FFMPEG.exe procesas baigė darbą.]);

imgCount = length(dir(fullfile(raw_frames_dir '*.bmp'))); % bendras pavienių vaizdų kiekis

t=toc; % laikmačio pabaiga
display([' datestr(now,13) ']: Video failas sėkmingai suskaidytas į ' num2str(imgCount) ' vaizdus per '
num2str(datestr(datetime(0,0,0,0,0,t),'HH:MM:SS'))]);
disp('<a href = "matlab:winopen(raw_frames_dir)">Originalių vaizdų direktorija</a>')

Watermark_Image = imread('C:\Matlab\DWT\Data\SN_forHD6.jpg'); % nuskaityti vandens ženklą
S = im2double(im2bw(Watermark_Image)); % paversti vandens ženklą į juodai baltą, double tipo
[wmh,wmw] = size(S); % vandens ženklo raiška (aukštis, plotis)
x = 0.008; % vandens ženklo įterpimo stiprumas

display([' datestr(now,13) ']: Vaizdai žymimi vandens ženklų ir išsaugomi ...'])
tic; % laikmačio pradžia
SS = reshape(Watermark_Image,1,numel(Watermark_Image)); % Vandens ženklas paverčiamas į vienmatį
masyvą
```

```

for i = 1:imgCount
    filename3 = [sprintf('%03d',i) '.bmp']; % originalaus vaizdo failo pavadinimas
    fullname3 = fullfile(raw_frames_dir,filename3); % originalių vaizdų direktorija
    Original_Image=imread(fullname3); % nuskaityti originalų vaizdą

    [LLr,LHr,HLr,HHr] = dwt2(Original_Image(:,:,1),'haar'); % 1 lygio diskrečioji bangelių transformacija,
    pritaikoma raudonos spalvos dedamajai
    [LLr21,LHr21,HLr21,HHr21] = dwt2(LLr,'haar'); % 2 lygio diskrečioji bangelių transformacija, pritaikoma
    LL sričiai

    [h,w,~] = size(LLr21); % LL21 srities aukštis ir plotis

    W_field = zeros(h,w,1); % sukuriamas tuščias masyvas, skirtas patalpinti vandens ženklui
    W_field(:,:,1) = reshape(SS(1:h*w),h,w); % patalpinamas LL21 dydžio vandens ženklas

    wLLr21 = LLr21 + x*W_field(:,:,1); % vandens ženklų pažymima LL21 sritis
    wLLr = idwt2(wLLr21,LHr21,HLr21,HHr21,'haar'); % vandens ženklų pažymėtos LL srities
    dekompozicija

    Input_Image2 = idwt2(wLLr,LHr,HLr,HHr,'haar'); % vandens ženklų pažymėtos raudonos spalvos
    dedamosios dekompozicija
    Watermarked_Image = cat(3,Input_Image2,Original_Image(:,:,2),Original_Image(:,:,3)); % vandens ženklų
    pažymėtas RGB vaizdas

    filename4 = [sprintf('%03d',i) '.bmp']; % vandens ženklų pažymėto vaizdo failo pavadinimas
    fullname4 = fullfile(watermarked_frames_dir,filename4); % vandens ženklų pažymėtų vaizdų direktorija
    imwrite(Watermarked_Image,fullname4,'bmp'); % išsaugoti vandens ženklų pažymėtą vaizdą nurodytoje
    direktorijoje
end

t=toc; % laikmačio pabaiga
display([' datestr(now,13) ': ' num2str(imgCount) ' vaizdų sėkmingai pažymėta vandens ženklų per '
num2str(datestr(datetime(0,0,0,0,0,t),'HH:MM:SS'))]);
disp('<a href = "matlab:winopen(watermarked_frames_dir)">Vandens ženklų pažymėtų vaizdų
direktorija</a>')

%% Rezultatai

figure
imshowpair(Original_Image,Watermarked_Image,'montage');
title('Originalus ir vandens ženklų pažymėtas vaizdas')

dwt_compress.m

warning('off','all'); % išjungti įspėjančius pranešimus
clear all; % panaikinti egzistuojančius objektus, atlaisvinti atmintį
close all; % panaikinti egzistuojančias figūras
clc; % išvalyti komandų langą
addpath(genpath('C:\Matlab\DWT'));

watermarked_frames_dir = 'C:\Matlab\DWT\Data\watermarked_frames'; % vandens ženklų pažymėtų vaizdų
direktorija
compressed_videos_dir = 'C:\Matlab\DWT\Data\compressed_videos'; % glaudintų video failų direktorija

cd(watermarked_frames_dir); % pereiti į vandens ženklų pažymėtų vaizdų direktoriją

```

```

tic % laikmačio pradžia
disp([' datestr(now,13) ']: Kuriamas video failas iš pavienių vaizdų ...');
system('ffmpeg -y -r 25 -i %03d.bmp -c:v libx264 -b:v 16000k ..\compressed_videos\comp_watermarked.mp4
'); % video failo sukūrimo iš pavienių vaizdų komanda

flag = true;
while flag
    disp([' datestr(now,13) ']: FFMPEG.exe procesas vis dar aktyvus.]);
    flag = isprocess('ffmpeg.exe'); % "isprocess" funkcija tikrina, ar procesas 'ffmpeg.exe' vis dar vykdo
komandą
    pause(1); % tikrinti kas sekundę
end

disp([' datestr(now,13) ']: FFMPEG.exe procesas baigė darbą.]);

t=toc; % laikmačio pabaiga
display([' datestr(now,13) ']: Video failas sėkmingai sukurtas per '
num2str(datestr(datetime(0,0,0,0,0,t),'HH:MM:SS'))]);
disp('<a href = "matlab:winopen(compressed_videos_dir)">Glaudintų vandens ženklų pažymėtų video failų
direktorija</a>')

cd ../../;

dwt_attack.m

warning('off','all'); % išjungti išspėjančius pranešimus
clear all; % panaikinti egzistuojančius objektus, atlaisvinti atmintį
close all; % panaikinti egzistuojančias figūras
clc; % išvalyti komandų langą

cd('C:\Matlab\DWT');

compressed_frames_dir = 'C:\Matlab\DWT\Data\compressed_frames'; % glaudinti vandens ženklų pažymėtų
vaizdų direktorija
attacked_frames_dir = 'C:\Matlab\DWT\Data\attacked_frames'; % vandens ženklų pažymėtų vaizdų po
manipuliacijų direktorija
compressedImages = dir(fullfile(compressed_frames_dir,'*.bmp')); % glaudinti vandens ženklų pažymėti
vaizdai su plėtiniu BMP
delete('Data\compressed_frames\*.bmp'); % ištrinti visus glaudintus vandens ženklų pažymėtus vaizdus su
plėtiniu BMP
delete('Data\attacked_frames\*.bmp'); % ištrinti visus vandens ženklų pažymėtus vaizdus po manipuliacijų su
plėtiniu BMP

display([' datestr(now,13) ']: Video failas skaidomas į pavienius vaizdus ...])
system('ffmpeg -r 25 -i Data\compressed_videos\comp_watermarked.mp4 -f image2
Data\compressed_frames\%03d.bmp'); % vandens ženklų pažymėto video failo skaidymo į pavienius vaizdus
komanda

tic % laikmačio pradžia

flag = true;
while flag
    disp([' datestr(now,13) ']: FFMPEG.exe procesas vis dar aktyvus.]);
    flag = isprocess('ffmpeg.exe'); % "isprocess" funkcija tikrina, ar procesas 'ffmpeg.exe' vis dar vykdo
komandą
    pause(1); % tikrinti kas 1 sekundę
end

```

```

disp([' datestr(now,13) ']: FFMPEG.exe procesas baigė darbą.']);
t=toc; % laikmačio pabaiga
display([' datestr(now,13) ']: Video failas sėkmingai suskaidytas į pavienius kadrus per '
num2str(datestr(datetime(0,0,0,0,0,t),'HH:MM:SS'))]);

imgcount = length(dir([compressed_frames_dir '*.bmp'])); % bendras pavienių vaizdų kiekis
imageHeight = size(imread('\Data\compressed_frames\001.bmp'),1); % vaizdo aukštis
imageWidth = size(imread('\Data\compressed_frames\001.bmp'),2); % vaizdo plotis

disp([' datestr(now,13) ']: Vykdoma vaizdų manipuliacija ...]);
tic % laikmačio pradžia
for i = 1:imgcount
    filename1 = strcat(fullfile(compressed_frames_dir,compressedImages(i).name)); % glaudinto vandens
ženklų pažymėto vaizdo failo pavadinimas
    Watermarked_Image = imread(filename1); % nuskaityti vandens ženklų pažymėtą vaizdą
    % Vaizdo manipuliacija nr. 1 - horizontalios juodos juostos vaizdo viršuje ir apačioje
    % bar_height = 50; % juostos aukštis
    % top_bar = [0 0 vidW 0 vidW bar_height 0 bar_height];
    % bottom_bar = [0 vidH-bar_height vidW vidH-bar_height vidW vidH 0 vidW];
    % Attacked_Image = insertShape(Watermarked_Image, 'FilledPolygon', {top_bar, bottom_bar}, 'Color',
{'black'}, 'Opacity', 1);
    % Vaizdo manipuliacija nr. 2 - "Gaussian" filtras
    % Attacked_Image = imgaussfilt(Watermarked_Image, 1);
    % Vaizdo manipuliacija nr. 3 - "Salt & Pepper" filtras
    Attacked_Image = imnoise(Watermarked_Image,'salt & pepper',0.001);

    filename2 = [sprintf('%03d',i) '.bmp']; % vandens ženklų pažymėto vaizdo po manipuliacijos failo
pavadinimas
    fullname2 = fullfile(attacked_frames_dir,filename2); % vandens ženklų pažymėtų vaizdų po manipuliacijų
direktorija
    imwrite(Attacked_Image,fullname2,'bmp'); % išsaugoti vandens ženklų pažymėtą vaizdą po manipuliacijos
nurodytoje direktorijoje
end

t=toc; % laikmačio pabaiga
display([' datestr(now,13) ']: Atliktos manipuliacijas ' num2str(imgcount) ' vaizdų per '
num2str(datestr(datetime(0,0,0,0,0,t),'HH:MM:SS'))]);

cd(attacked_frames_dir); % pereiti į vandens ženklų pažymėtų vaizdų po manipuliacijų direktoriją

tic % laikmačio pradžia

disp([' datestr(now,13) ']: Kuriamas video failas iš pavienių vaizdų ...]);
system('ffmpeg -y -r 25 -i %03d.bmp -c:v libx264 -b:v 16000k ..\compressed_videos\comp_attacked.mp4 ');
% video failo sukūrimo iš pavienių vaizdų komanda

flag = true;
while flag
    disp([' datestr(now,13) ']: FFMPEG.exe procesas vis dar aktyvus.]);
    flag = isprocess('ffmpeg.exe'); % "isprocess" funkcija tikrina, ar procesas 'ffmpeg.exe' vis dar vykdo
komandą
    pause(1); % tikrinti kas 1 sekundę
end

disp([' datestr(now,13) ']: FFMPEG.exe procesas baigė darbą.']);

```



```

t=toc; % laikmačio pabaiga
display([' datestr(now,13) ']: Video failas sėkmingai sukurtas per '
num2str(datestr(datenum(0,0,0,0,0,t),'HH:MM:SS'))]);

cd ../../;

dwt_extract.m

warning('off','all'); % išjungti įspėjančius pranešimus
clear all; % panaikinti egzistuojančius objektus, atlaisvinti atmintį
close all; % panaikinti egzistuojančias figūras
clc; % išvalyti komandų langą

cd('C:\Matlab\DWT');

Watermark_Image = imread('C:\Matlab\DWT\Data\SN_forHD6.jpg'); % nuskaityti vandens ženklą
raw_frames_dir = 'C:\Matlab\DWT\Data\raw_frames\'; % originalių vaizdų direktorija
watermarked_frames_dir = 'C:\Matlab\DWT\Data\watermarked_frames\'; % vandens ženklu pažymėtų vaizdų
direktorija
extracted_frames_dir = 'C:\Matlab\DWT\Data\extracted_frames\'; % iš vaizdo išgautų vandens ženklų
direktorija
compressed_frames_dir = 'C:\Matlab\DWT\Data\compressed_frames\'; % glaudintų vandens ženklu pažymėtų
vaizdų direktorija
attacked_frames_dir = 'C:\Matlab\DWT\Data\attacked_frames\'; % vandens ženklu pažymėtų vaizdų po
manipuliacijų direktorija
delete('Data\extracted_frames\*.bmp'); % ištrinti visus iš vaizdo išgautus vandens ženklus su plėtinium BMP
delete('Data\compressed_frames\*.bmp'); % ištrinti visus glaudintus vandens ženklu pažymėtus vaizdus su
plėtinium BMP

display([' datestr(now,13) ']: Video failas skaidomas į pavienius vaizdus ...])
system('ffmpeg -r 25 -i Data\compressed_videos\comp_watermarked.mp4 -f image2
Data\compressed_frames\%03d.bmp'); % suglaudinto video failo skaidymo į pavienius vaizdus komanda

tic % laikmačio pradžia

flag = true;
while flag
    disp([' datestr(now,13) ']: FFMPEG.exe procesas vis dar aktyvus.]);
    flag = isprocess('ffmpeg.exe'); % "isprocess" funkcija tikrina, ar procesas 'ffmpeg.exe' vis dar vykdo
komandą
    pause(1); % tikrinti kas 1 sekundę
end

disp([' datestr(now,13) ']: FFMPEG.exe procesas baigė darbą.]);

imgCount = length(dir([compressed_frames_dir '*.bmp'])); % bendras pavienių vaizdų kiekis

t=toc; % laikmačio pabaiga
display([' datestr(now,13) ']: Video failas sėkmingai suskaidytas į ' num2str(imgCount) ' vaizdus per '
num2str(datestr(datenum(0,0,0,0,0,t),'HH:MM:SS'))]);

S = im2double(im2bw(Watermark_Image)); % paversti vandens ženklą į juodai baltą, double tipo
[wmh,wmw] = size(S); % vandens ženklo raiška (aukštis, plotis)

x = 0.008; % vandens ženklo įterpimo stiprumas

rawImages = dir(fullfile(raw_frames_dir,'*.bmp')); % originalūs vaizdai su plėtinium BMP

```

```

watermarkedImages = dir(fullfile(watermarked_frames_dir,'*.bmp')); % vandens ženklų pažymėti vaizdai su
plėtinu BMP
compressedImages = dir(fullfile(compressed_frames_dir,'*.bmp')); % glaudinti vandens ženklų pažymėti
vaizdai su plėtinu BMP
attackedImages = dir(fullfile(attacked_frames_dir,'*.bmp')); % vandens ženklų pažymėti vaizdai po
manipuliacijų su plėtinu BMP
Extracted_Watermark(1:imgCount) = struct('cdata',zeros(wmh,wmw,1,'uint8'),'colormap',[]); % sukuriama
tuščia struktūra, skirta talpinti išgautiems vandens ženklams

tic % laikmačio pradžia
display([' datestr(now,13) ']: Išgaunami vandens ženklai iš vaizdų ...]);

for i=1:imgCount
    filename1 = strcat(fullfile(raw_frames_dir,rawImages(i).name)); % kelias iki originalių vaizdų failų
    Original_Image = imread(filename1); % nuskaityti originalų vaizdą

    filename2 = strcat(fullfile(compressed_frames_dir,compressedImages(i).name)); % kelias iki vandens
ženklų pažymėtų vaizdų failų
    Watermarked_Image = imread(filename2); % nuskaityti vandens ženklų pažymėtą vaizdą

    [LL,LH,HL,HH]=dwt2(Original_Image(:, :, 1),'haar'); % 1 lygio diskrečioji bangelių transformacija,
pritaikoma originalaus vaizdo raudonos spalvos dedamajai
    [fLL21r,fLH21r,fHL21r,fHHr21] = dwt2(LL,'haar'); % 2 lygio diskrečioji bangelių transformacija,
pritaikoma originalaus vaizdo LL sričiai

    [wLLr,wLHr,wHLr,wHHr]=dwt2(Watermarked_Image(:, :, 1),'haar'); % 1 lygio diskrečioji bangelių
transformacija, pritaikoma vandens ženklų pažymėto vaizdo raudonos spalvos dedamajai
    [wLLr21,wLHr21,wHLr21,wHHr21] = dwt2(wLLr,'haar'); % 2 lygio diskrečioji bangelių transformacija,
pritaikoma vandens ženklų pažymėto vaizdo LL sričiai

    Extracted_Watermark(i).cdata = uint8((wLLr21-fLL21r)/x); % vandens ženklų išgavimas, palyginant
vandens ženklų pažymėtą ir originalų vaizdus bei skirtumą padalinant iš vandens ženklų įterpimo stiprumo

    filename3 = [sprintf('%03d',i) '.bmp']; % iš vaizdo išgauto vandens ženklų failo pavadinimas
    fullname4 = fullfile(extracted_frames_dir,filename3); % iš vaizdo išgauto vandens ženklų direktorija
    imwrite(Extracted_Watermark(i).cdata,fullname4,'bmp'); % išsaugoti iš vaizdo išgautą vandens ženklą
nurodytoje direktorijoje

end
t=toc; % laikmačio pabaiga
display([' datestr(now,13) ']: Išgauta ' num2str(imgCount) ' vandens ženklų per '
num2str(datestr(datetime(0,0,0,0,0,t), 'HH:MM:SS'))]);

figure
imshowpair(S,Extracted_Watermark(1).cdata,'montage');
title('Originalus ir išgautas vandens ženklas')

% figure
% imshowpair(Original_Image,Watermarked_Image,'diff')
% title('Skirtumas tarp originalaus ir vandens ženklų pažymėto vaizdo')

%% Rezultatai

MSE = immse(im2double(Extracted_Watermark(1).cdata),S);
display([' datestr(now,13) ']: Vidutinė kvadratinė paklaida (MSE) = ' num2str(MSE) ];
[PSNR, SNR] = psnr(im2double(Extracted_Watermark(1).cdata),S);

```

```

display([' datestr(now,13) ']: Pikinis signalo ir triukšmo santykis (PSNR) = ' num2str(PSNR) ' dB' ]);
[ssimval, ssimmap] = ssim(im2double(Extracted_Watermark(1).cdata),S);
display([' datestr(now,13) ']: Struktūrinio panašumo matavimo indeksas (SSIM) = ' num2str(ssimval) ]);

```

dct_embed.m

```

warning('off','all'); % išjungti įspėjančius pranešimus
clear all; % panaikinti egzistuojančius objektus, atlaisvinti atmintį
close all; % panaikinti egzistuojančias figūras
clc; % išvalyti komandų langą

```

```

raw_frames_dir = 'C:\Matlab\DCT\Data\raw_frames'; % originalių vaizdų direktorija
rawImages = dir(fullfile(raw_frames_dir,'*.bmp')); % originalūs vaizdai su plėtiniu BMP
watermarked_frames_dir = 'C:\Matlab\DCT\Data\watermarked_frames'; % vandens ženklų pažymėtų vaizdų
direktorija
delete('C:\Matlab\DCT\Data\raw_frames\*.bmp'); % ištrinti visus originalius vaizdus su plėtiniu BMP
delete('C:\Matlab\DCT\Data\watermarked_frames\*.bmp'); % ištrinti visus vandens ženklų pažymėtus vaizdus
su plėtiniu BMP

```

```

tic % laikmačio pradžia
display([' datestr(now,13) ']: Glaudinamas originalus video failas ...])
system('ffmpeg -y -r 25 -i Data\raw2-1.avi -c:v libx264 -b:v 16000k Data\raw-comp.mp4'); % originalaus
video failo glaudinimo komanda
t=toc; % laikmačio pabaiga
display([' datestr(now,13) ']: Video failas sėkmingai suglaudintas per '
num2str(datestr(datenum(0,0,0,0,0,t),'HH:MM:SS'))]);

```

```

tic % laikmačio pradžia
display([' datestr(now,13) ']: Suglaudintas video failas skaidomas į pavienius vaizdus ...])
system('ffmpeg -r 25 -i Data\raw-comp.mp4 -f image2 Data\raw_frames\%03d.bmp'); % suglaudinto video
failo skaidymo į pavienius vaizdus komanda

```

```

flag = true;
while flag
    disp([' datestr(now,13) ']: FFMPEG.exe procesas vis dar aktyvus.]);
    flag = isprocess('ffmpeg.exe'); % "isprocess" funkcija tikrina, ar procesas 'ffmpeg.exe' vis dar vykdo
komandą
    pause(1); % tikrinti kas 1 sekundę
end

```

```

disp([' datestr(now,13) ']: FFMPEG.exe procesas baigė darbą.]);

```

```

% Originalaus vaizdo parametrai
imgCount = length(dir(fullfile(raw_frames_dir '*.bmp'))); % bendras pavienių vaizdų kiekis
imageHeight = size(imread('Data\raw_frames\001.bmp'),1); % vaizdo aukštis
imageWidth = size(imread('Data\raw_frames\001.bmp'),2); % vaizdo plotis

```

```

t=toc; % laikmačio pabaiga
display([' datestr(now,13) ']: Video failas sėkmingai suskaidytas į ' num2str(imgCount) ' vaizdus per '
num2str(datestr(datenum(0,0,0,0,0,t),'HH:MM:SS'))]);
disp('<a href = "matlab:winopen(raw_frames_dir)">Originalių vaizdų direktorija</a>')

```

```

Watermark_Image = imread('Data\SN_forHD6.jpg'); % nuskaityti vandens ženklą
S = im2double(im2bw(Watermark_Image)); % paversti vandens ženklą į juodai baltą, double tipo
watermarkHeight = size(S, 1); % vandens ženklo aukštis

```

```

watermarkWidth = size(S, 2); % vandens ženklų plotis

if (imageHeight/watermarkHeight == 8) && (imageWidth/watermarkWidth == 8)
    disp([' datestr(now,13) ']: Originalaus vaizdo ir vandens ženklų dydžiai tinkami.]);
else
    disp([' datestr(now,13) ']: DĖMESIO! Originalaus vaizdo ir vandens ženklų dydžiai nėra tinkami.]);
    disp([' datestr(now,13) ']: Procesas nutrauktas!]);
    return
end

display([' datestr(now,13) ']: Vaizdai žymimi vandens ženklų ir išsaugomi ...'])
tic; % laikmačio pradžia

x = 3; % vandens ženklų įterpimo stiprumas
w = S(:); % paverčia vandens ženklų matricą į vektorius

for k = 1:imgCount
    filename3 = [sprintf('%03d',k) '.bmp']; % originalaus vaizdo failo pavadinimas
    fullname3 = fullfile(raw_frames_dir,filename3); % originalių vaizdų direktorija
    Original_Image = imread(fullname3); % nuskaityti originalų vaizdą

    InputImage = Original_Image(:,1); % parenkama originalaus vaizdo raudonos spalvos dedamoji

    watermarkedImage = zeros(imageHeight, imageWidth); % paruošiamas tuščias masyvas paženklinam
    vaizdai

    r1 = 1; r2 = 8; % vaizdo bloko pradžios ir pabaigos reikšmės pagal stulpelius
    c1 = 1; c2 = 8; % vaizdo bloko pradžios ir pabaigos reikšmės pagal eilutes
    count = 0;

    fun = @(block_struct) dct2(block_struct.data); % aprašoma funkcija, kuri kiekvieną vaizdo bloką apdoros
    diskrečiąja kosinusų transformacija
    A = blockproc(InputImage,[8 8],fun); % vaizdas suskaidomas 8x8 dydžio blokais

    for i = 1:watermarkHeight
        for j = 1:watermarkWidth
            count = count + 1; % didinti vandens ženklų įterpiamos reikšmės poziciją
            f_o = A(r1:r2, c1:c2); % išgauti 8x8 bloką iš originalaus vaizdo
            f_o(8, 8) = f_o(8, 8) + w(count) * x; % įterpti vandens ženklų reikšmę į paskutinę bloko poziciją
            out = idct2(f_o); % blokui atlikti atvirkštinę diskrečiąją kosinusų transformaciją
            InputImage(r1:r2, c1:c2) = out; % formuojamas vandens ženklų žymimas vaizdas
            c1 = c1 + 8; % pereiti prie kito vaizdo bloko, nustatant jo pradžios tašką pagal stulpelius
            c2 = c2 + 8; % pereiti prie kito vaizdo bloko, nustatant jo pabaigos tašką pagal stulpelius
        end
        r1 = r1 + 8; % pereiti prie kito vaizdo bloko, nustatant jo pradžios tašką pagal eilutes
        r2 = r2 + 8; % pereiti prie kito vaizdo bloko, nustatant jo pabaigos tašką pagal eilutes
        c1 = 1; c2 = 8;
    end

    watermarkedImage = cat(3, InputImage, Original_Image(:,2), Original_Image(:,3)); % vandens ženklų
    pažymėtas RGB vaizdas

    filename4 = [sprintf('%03d',k) '.bmp']; % vandens ženklų pažymėto vaizdo failo pavadinimas
    fullname4 = fullfile(watermarked_frames_dir,filename4); % vandens ženklų pažymėtų vaizdų direktorija
    imwrite(watermarkedImage,fullname4,'bmp'); % išsaugoti vandens ženklų pažymėtą vaizdą nurodytoje
    direktorijoje
end

```

```
t=toc; % laikmačio pabaiga
display(['[' datestr(now,13) ']: ' num2str(imgCount) ' vaizdų sėkmingai pažymėta vandens ženklų per '
num2str(datestr(datetime(0,0,0,0,0,t),'HH:MM:SS'))]);
disp('<a href = "matlab:winopen(watermarked_frames_dir)">Vandens ženklų pažymėtų vaizdų
direktorija</a>')
```

dct_compress.m

```
warning('off','all'); % išjungti įspėjančius pranešimus
clear all; % panaikinti egzistuojančius objektus, atlaisvinti atmintį
close all; % panaikinti egzistuojančias figūras
clc; % išvalyti komandų langą
addpath(genpath('C:\Matlab\DCT'));

watermarked_frames_dir = 'C:\Matlab\DCT\Data\watermarked_frames'; % vandens ženklų pažymėtų vaizdų
direktorija
compressed_videos_dir = 'C:\Matlab\DCT\Data\compressed_videos'; % glaudintų video failų direktorija

cd(watermarked_frames_dir); % pereiti į vandens ženklų pažymėtų vaizdų direktoriją

tic % laikmačio pradžia
disp(['[' datestr(now,13) ']: Kuriamas video failas iš pavienių vaizdų ...']);
system('ffmpeg -y -r 25 -i %03d.bmp -c:v libx264 -b:v 16000k ..\compressed_videos\comp_watermarked.mp4
'); % video failo sukūrimo iš pavienių vaizdų komanda

flag = true;
while flag
    disp(['(' datestr(now,13) '): FFMPEG.exe procesas vis dar aktyvus.']);
    flag = isprocess('ffmpeg.exe'); % "isprocess" funkcija tikrina, ar procesas 'ffmpeg.exe' vis dar vykdo
komandą
    pause(1); % tikrinti kas sekundę
end

disp(['[' datestr(now,13) ']: FFMPEG.exe procesas baigė darbą.']);

t=toc; % laikmačio pabaiga

display(['[' datestr(now,13) ']: Video failas sėkmingai sukurtas per '
num2str(datestr(datetime(0,0,0,0,0,t),'HH:MM:SS'))]);
disp('<a href = "matlab:winopen(compressed_videos_dir)">Glaudintų vandens ženklų pažymėtų video failų
direktorija</a>')

cd ../../;
```

dct_extract.m

```
warning('off','all'); % išjungti įspėjančius pranešimus
clear all; % panaikinti egzistuojančius objektus, atlaisvinti atmintį
close all; % panaikinti egzistuojančias figūras
clc; % išvalyti komandų langą

cd('C:\Matlab\DCT');

raw_frames_dir = 'C:\Matlab\DCT\Data\raw_frames\'; % originalių vaizdų direktorija
```

```

watermarked_frames_dir = 'C:\Matlab\DCT\Data\watermarked_frames'; % vandens ženklų pažymėtų vaizdų
direktorija
extracted_frames_dir = 'C:\Matlab\DCT\Data\extracted_frames'; % iš vaizdo išgautų vandens ženklų
direktorija
compressed_frames_dir = 'C:\Matlab\DCT\Data\compressed_frames'; % glaudintų vandens ženklų pažymėtų
vaizdų direktorija
attacked_frames_dir = 'C:\Matlab\DCT\Data\attacked_frames'; % vandens ženklų pažymėtų vaizdų po
manipuliacijų direktorija
delete('Data\extracted_frames\*.bmp'); % ištrinti visus iš vaizdo išgautus vandens ženklus su plėtinium BMP
delete('Data\compressed_frames\*.bmp'); % ištrinti visus glaudintus vandens ženklų pažymėtus vaizdus su
plėtinium BMP

display([' datestr(now,13)']: Video failas skaidomas į pavienius vaizdus ...])
system('ffmpeg -r 25 -i Data\compressed_videos\comp_watermarked.mp4 -f image2
Data\compressed_frames\%03d.bmp'); % suglaudinto video failo skaidymo į pavienius vaizdus komanda

tic % laikmačio pradžia

flag = true;
while flag
    disp([' datestr(now,13)']: FFMPEG.exe procesas vis dar aktyvus.]);
    flag = isprocess('ffmpeg.exe'); % "isprocess" funkcija tikrina, ar procesas 'ffmpeg.exe' vis dar vykdo
komandą
    pause(1) % tikrinti kas 1 sekundę
end

disp([' datestr(now,13)']: FFMPEG.exe procesas baigė darbą.]);

imgCount = length(dir([compressed_frames_dir '*.bmp'])); % bendras pavienių vaizdų kiekis
imageHeight = size(imread('Data\compressed_frames\001.bmp'),1); % Vaizdo aukštis
imageWidth = size(imread('Data\compressed_frames\001.bmp'),2); % Vaizdo plotis

t=toc; % laikmačio pabaiga
display([' datestr(now,13)']: Video failas sėkmingai suskaidytas į ' num2str(imgCount) ' vaizdus per '
num2str(datestr(datetime(0,0,0,0,0,t),'HH:MM:SS'))]);

rawImages = dir(fullfile(raw_frames_dir, '*.bmp')); % originalūs vaizdai su plėtinium BMP
watermarkedImages = dir(fullfile(watermarked_frames_dir, '*.bmp')); % vandens ženklų pažymėti vaizdai su
plėtinium BMP
compressedImages = dir(fullfile(compressed_frames_dir, '*.bmp')); % glaudinti vandens ženklų pažymėti
vaizdai su plėtinium BMP
attackedImages = dir(fullfile(attacked_frames_dir, '*.bmp')); % vandens ženklų pažymėti vaizdai po
manipuliacijų su plėtinium BMP

x = 3; % vandens ženklo įterpimo stiprumas

secret = imread('Data\SN_forHD6.jpg'); % nuskaityti vandens ženklą
watermarkImage = im2double(im2bw(secret)); % paversti vandens ženklą į juodai baltą, double tipo
w_height = size(watermarkImage, 1); % vandens ženklo aukštis
w_width = size(watermarkImage, 2); % vandens ženklo plotis

Extracted_Watermark(1:imgCount) = struct('cdata',zeros(w_height,w_width,1,'uint8'),'colormap',[[]]); %
sukuriama tuščia struktūra, skirta talpinti išgautiems vandens ženkliams

if (imageHeight/w_height == 8) && (imageWidth/w_width == 8)
    disp([' datestr(now,13)']: Pažymėto vaizdo ir vandens ženklo dydžiai tinkami.]);
else

```

```

disp([' datestr(now,13) ']: DĖMESIO! Originalaus vaizdo ir vandens ženklų dydžiai nėra tinkami.']);
disp([' datestr(now,13) ']: Procesas nutrauktas!]);
return
end

tic
display([' datestr(now,13) ']: Išgaunami vandens ženklai iš vaizdų ...]);

for k=1:imgCount
    filename1 = strcat(fullfile(raw_frames_dir,rawImages(k).name)); % kelias iki originalių vaizdų failų
    Input_Image = imread(filename1); % nuskaityti originalų vaizdą
    orgInput = Input_Image(:,:,1); % originalaus vaizdo raudonos spalvos dedamoji

    filename2 = strcat(fullfile(compressed_frames_dir,compressedImages(k).name)); % kelias iki vandens
    ženklų pažymėtų vaizdų failų
    Watermarked_Image = imread(filename2); % nuskaityti vandens ženklų pažymėtą vaizdą
    wInput = Watermarked_Image(:,:,1); % vandens ženklų pažymėto vaizdo raudonos spalvos dedamoji

    r1 = 1; r2 = 8; % vaizdo bloko pradžios ir pabaigos reikšmės pagal eilutes
    c1 = 1; c2 = 8; % vaizdo bloko pradžios ir pabaigos reikšmės pagal stulpelius
    count = 0; % pirmoji išgauto vandens ženklų reikšmės pozicija

    fun = @(block_struct) dct2(block_struct.data); % aprašoma funkcija, kuri kiekvieną vaizdo bloką apdoros
    diskrečiąja kosinusų transformacija
    A = blockproc(orgInput,[8 8],fun); % originalus vaizdas suskaidomas 8x8 dydžio blokais
    B = blockproc(wInput,[8 8],fun); % vandens ženklų pažymėtas vaizdas suskaidomas 8x8 dydžio blokais

    for i = 1:w_height
        for j = 1:w_width
            count = count + 1; % didinti vandens ženklų įterpiamos reikšmės poziciją

            f_o = A(r1:r2, c1:c2); % išgauti 8x8 bloką iš originalaus vaizdo
            f_w = B(r1:r2, c1:c2); % išgauti 8x8 bloką iš vandens ženklų pažymėto vaizdo

            extractedWatermark(count) = (f_w(8, 8) - f_o(8, 8)) / x; % išgauti vandens ženklų reikšmę, padalinus
            originalaus ir apsaugoto vaizdų skirtumą iš vandens ženklų įterpimo stiprumo
            c1 = c1 + 8; % pereiti prie kito vaizdo bloko, nustatant jo pradžios tašką pagal stulpelius
            c2 = c2 + 8; % pereiti prie kito vaizdo bloko, nustatant jo pabaigos tašką pagal stulpelius
        end
        r1 = r1 + 8; % pereiti prie kito vaizdo bloko, nustatant jo pradžios tašką pagal eilutes
        r2 = r2 + 8; % pereiti prie kito vaizdo bloko, nustatant jo pabaigos tašką pagal eilutes
        c1 = 1; c2 = 8;
    end

    Extracted_Watermark(k).cdata = vec2mat(extractedWatermark, w_width, w_height); % iš vektoriaus
    suformuojamas išgautas vandens ženklas
    filename3 = [sprintf('%03d',k) '.bmp']; % iš vaizdo išgauto vandens ženklų failo pavadinimas
    fullname4 = fullfile(extracted_frames_dir,filename3); % iš vaizdo išgauto vandens ženklų direktorija
    imwrite( Extracted_Watermark(k).cdata,fullname4,'bmp'); % išsaugoti iš vaizdo išgautą vandens ženklą
    nurodytoje direktorijoje
end

t=toc; % laikmačio pabaiga
display([' datestr(now,13) ']: Išgauta ' num2str(imgCount) ' vandens ženklų per '
num2str(datestr(datetime(0,0,0,0,0,t),'HH:MM:SS'))]);

figure

```

```
imshowpair(watermarkImage,Extracted_Watermark(1).cdata,'montage');  
title('Originalus ir išgautas vandens ženklas')
```

```
MSE = immse(wMark(1).cdata,watermarkImage);  
display([' datestr(now,13) ']: Vidutinė kvadratinė paklaida (MSE) = ' num2str(MSE) ]);  
[PSNR, SNR] = psnr(wMark(1).cdata,watermarkImage);  
display([' datestr(now,13) ']: Pikinis signalo ir triukšmo santykis (PSNR) = ' num2str(PSNR) ' dB' ]);  
[ssimval, ssimmap] = ssim(wMark(1).cdata,watermarkImage);  
display([' datestr(now,13) ']: Struktūrinio panašumo matavimo indeksas (SSIM) = ' num2str(ssimval) ]);
```