

Efektyvus vandens ženklų taikymas dideliame žiūrovų skaičiui identifikuoti

Deivis Zolba 

Matematikos ir informatikos fakultetas, Vilniaus universitetas 

✉ Korespondentas: deivis.zolba@mif.stud.vu.lt

Įteiktas 2025 birželio 25; publikuotas 2025 gruodžio 21

Santrauka. Nagrinėjami metodai, kaip integruoti žiūrovo identifikavimo informaciją į skaitmeninį turinį, tokį kaip vaizdo įrašai ar nuotraukos. Dažnių srities (frequency domain) vandens ženklinimo modeliai pasižymi dideliu atsparumu suspaudimui, regionų iškirpimui ir kitoms atakoms, tačiau jų taikymas masiniu mastu susiduria su reikšmingais skaičiavimo kaštais. Standartinis modelis reikalauja itin daug skaičiavimo išteklių, todėl nėra praktiškai pritaikomas plačiu mastu.

Siūlomas sprendimas suskaido dažnio sritį į segmentus ir sukuria ženklų fragmentus, kuriuos galima įvairiai kombinuoti. Šis modelis yra gerokai efektyvesnis ir lengviau plečiamas: jis sumažina skaičiavimo sąnaudas ir leidžia sugeneruoti daug unikalių vandens ženklų variantų iš riboto fragmentų rinkinio.

Raktiniai žodžiai: skaitmeniniai vandens ženklai; duomenų apsauga; žiūrovo atpažinimas; autorių teisių apsauga; kombinatorinė optimizacija

AMS: 90C27, 68Q25, 68P99, 68W99

1 Įžanga

Skaitmeniniai vandens ženklai yra informacijos saugumo įrankiai. Ši technika leidžia įterpti papildomą informaciją į skaitmeninį turinį: nuotrauką, vaizdo įrašą, dokumentą. Vandens ženklai įprastai naudojami autorių teisių apsaugai arba norint užtikrinti turinio autentiškumą. Tradiciniai skaitmeniniai vandens ženklai yra atpažįstami kaip matomi logotipai ar tekstai (žr. 1 pav. b). Tačiau matomumas yra ir privalumas ir trūkumas. Nors jis yra numatyta funkcija, kuri gali atgrasyti nuo neteisėto naudojimo, jis taip pat aiškiai parodo, kur yra įterptas, todėl šį vandens ženklą galima pašalinti arba sugadinti. Priešingai, nematomi skaitmeniniai vandens ženklai yra nepastebimi žmogaus akimi, kaip parodyta 1 pav. c.



1 pav. Vandens ženklų matomumo tipai: matomas ir nematomas.

Socialinio turinio išpopuliarėjimas lėmė skaitmeninio turinio vartojimo augimą. Todėl skaitmeninės žiniasklaidos teikėjai kasdien susiduria su piratavimo ir autorių teisių apsaugos problemomis. Nors jau yra veiksmingų priemonių, leidžiančių pašalinti autorių teisių apsaugotą medžiagą iš interneto svetainių, jos sprendžia tik problemos simptomus, o ne jos priežastis. Todėl būtina spręsti priežastis, ieškant galimų sprendimų.

Skaitmeniniai vandens ženklai veiksmingai gali išspręsti vieną didžiausių problemų – neteisėto turinio kopijų sukūrimo šaltinio nustatymo. Skaitmeniniai vandens ženklai gali nešti tokią informaciją kaip žiūrovo vartotojo vardas, IP adresas ir pan. Tai atlikus, būtų galima nustatyti turinio kilmę, o tai leis sužinoti, kas sukūrė ir pasidalijo neteisėta turinio kopija. Nors šis metodas atrodo daug žadantis, jo praktinis įgyvendinimas susiduria su iššūkiais. Vienas iš jų yra plečiamumas, kadangi kiekvienas naudotojas matys suasmenintą turinį. Ko pasekoje reikia sukurti, bei transliuoti originalaus turinio kopijas, kuriose yra pridėta informacija apie žiūrovą. Kopijų kūrimas ir labai greitai tampa nepraktiškas vietos atžvilgiu. Įsivaizduokime, jog reikia saugoti tūkstančius to paties vaizdo įrašo kopijų norint sugebėti transliuoti tūkstančiui naudotojų, o kas jei žiūrovų milijonas? Todėl labai svarbu ištirti ir apsarstyti alternatyvius metodus, kurie įgyvendintų pagrindinį funkcionalumą – naudotojo atpažinimą. Pasiekus gana realistiškas išteklių sąnaudas, tai galėtų įgauti ir praktinį panaudojimą.

2 Literatūros apžvalga

Ankstyvieji tyrimai daugiausia buvo skirti vaizdų autorių teisių apsaugai ir kopijavimo kontrolei [3]. Vandens ženklų technika yra naudojama plačiai įvairiuose skaitmeniniame turinyje, tokiaime kaip garso ir vaizdo įrašai [3], ir net tekstiniuose dokumentuose [4]. Šiame skyriuje bus apžvelgiama literatūra apie skaitmeninius vandens ženklus,

the original author and source are credited.

nagrinėjamos vandens ženklų sritys, metodų klasifikacijos ir pagrindiniai skaitmeninių vandens ženklų tikslai.

Neseniai Chen ir kt. [2] tyrė vandens ženklų pritaikymą realiuoju laiku. Praktinė šio tyrimo idėja buvo įgyvendinta „Windows“ darbalaukio atvaizdavimo procese, tai reiškia, kad ji yra susieta su žiūrovo aplinka, ir nėra atliekama išorinėje skaitmeninėje terpėje. Šiame tyrime buvo nagrinėjama viena iš sudėtingiausių skaitmeninių vandens ženklų problemų. Išorinės ekrano kameros problema, kai ekranas yra filmuojamas su išorine kamera. Tokiu būdu sukuriant originalaus turinio kopiją atsiranda nemažai iškraipymų, o individualių pikselių vertės tampa neatpažįstamos, todėl dauguma skaitmeninių vandens ženklų metodų tampa neveiksmingi. Kita opi problema buvo perspektyvos pasikeitimas, nes įrašymas dažnai vyksta kampu [2]. Vienintelė veiksminga vandens ženklų technika, kuri atlaiko tokį išpuolį, yra matomi ekrano vandens ženklai, tokie kaip logotipai ir tekstas. Tokie elementai pablogina žiūrovų patirtį ir gali būti lengvai pašalinami, nes yra akivaizdūs. Šios problemos sukūrė aiškią būtinybę sukurti nematomą vandens ženklų sistemą, kuri būtų pakankamai patikima ir išliktų po išorinės kameros atakos. Vienas iš būdų spręsti perspektyvos problemą yra naudoti atskaitos taškus, kad vaizdas būtų „išlygintas“. Laimei, šiuos taškus galima lengvai pasirinkti kaip ekrano kampus. Visa tai galima automatizuoti įterpiant specialius atskaitos taškus, pasitelkiant egzistuojančius modelius arba naudoti giliuosius neuroninius tinklus, norint įveikti perspektyvos pokyčius ir Moiré triukšmą. Nors autoriams pavyko įgyvendinti vandens ženklų įterpimą realiu laiku pačioje kompiuterio architektūroje šis modelis yra susietas su operacine sistema, ir iš eksperimentų matyti, kad tai vis tiek yra opus išteklių atžvilgiu [2].

Įgyvendinimui autoriai sukūrė rezoliucijai dinamišką vandens ženklų šabloną, kuriame sujungtas pranešimo modelis su dviem sinchronizavimo žymių modeliais. Vandens ženklų pranešimas, kuriame yra identifikacinė informacija, pvz., naudotojo ID ir dabartinis laikas, pirmiausia užkoduojamas kaip QR kodas, jis yra pasirinktas dėl klaidų taisymo galimybių stipresnio atsparumo [2]. Gautas QR kodas vėliau transformuojamas į dažnių srities šabloną pasitelkiant Furjė transformaciją ir pakoreguojamas specialiais sinchronizavimo signalais. Sinchronizavimo modeliai veikia kaip atskaitos žymės, padedančios rasti ir nustatyti vandens ženklus. Įterpimui autoriai pasirinko sistemos lygmens metodą: dirbo tiesiogiai su „Windows Desktop Window Manager“, atsakingu už GUI atvaizdavimą. Šis vandens ženklų įterpimas pasirodė esąs labai efektyvus, nes jie gali apdoroti kiekvieną ekrano kadrą prieš siunčiant jį į ekraną. Jų įgyvendinimas realiuoju laiku įterpia nematomą vandens ženklą į kiekvieną kadrą, perimant keitimo grandinę, mechanizmą, kuris keičia atvaizduotus kadrus. Siekiant užtikrinti, kad vandens ženklas liktų nepastebimas žiūrovų akims, jie naudojo „Just Noticeable Difference“ (JND) modelį, koreguodami vandens ženklų stiprumą pagal tuo metu rodomą ekrano turinį [2]. Adaptyvusis metodas atsižvelgia į tai, kad ryškios arba stiprių raštų sritys gali nešti stipresnio intensyvumo vandens ženklus ir būti stipriai nepakitusios. Priešingai, lygiose, vienspalvėse srityse turi būti naudojami silpnescio intensyvumo vandens ženklai, kitaip kyla pavojus, kad bus matomi artefaktai [2]. Kompiuterinės architektūros lygio metodas yra pagrindinis šio darbo originalumas. Vietoj to, kad būtų modifikuojamas turinys, reikalaujantis daugiau skaičiavimo galios, vandens ženklų įterpimas vyksta operacinės sistemos ekrano atvaizdavimo procese, atliekant paprastas sudėties operacijas, todėl jis yra neįtikėtinai greitas. Rezultatas globalus: bet koks ekrane rodomas turinys, vaizdai, vaizdo įrašai, kelios programos ar dokumentai jau yra pažymėti vandens ženklais realiuoju laiku.

Apibendrinant, Chen ir kt. pristatė novatorišką metodą, pagal kurį vandens ženklų įterpimas atliekamas operacinės sistemos lygiu. Tai suteikė greitą, tikslų, realiuoju laiku veikiantį vandens ženklų įterpimo sprendimą, kuris yra atsparus net išorinių kamerų įrašams. Darbas operacinės sistemos lygiu leido šią techniką taikyti bet kokiam ekrano turiniui, nes vandens ženklas įterpiamas ekrano atvaizdavimo lygmeniu. Nors darbas yra transformatyvus, jis turi daug apribojimų. Vienas iš jų yra priklausomybė nuo platformos, dabartinis metodas yra susietas su „Windows DWM“ struktūra. Kitas apribojimas yra tiesioginės ekrano kopijos iš pačio kompiuterio, šios kopijos neturės vandens ženklo, nes jis egzistuoja ir yra matomas vartotojo ekrane, tad šis modelis yra skirtas tik išorinių kamerų įrašams.

An ir kt. pristatė daugiafunkcį vaizdo vandens ženklų algoritmą, kuris derina klasojimo aptikimą su autorių teisių apsauga [1]. Šis metodas apima dviejų tipų vandeninius ženklus: DFT šabloną, atsparų geometrinėms transformacijoms, ir antrąjį autorių teisių ženklą – DCT. DFT šablonas leidžia atlikti korekciją atpažinimo metu, o tai užtikrina vandeninio ženklo atsparumą [1]. Tuo tarpu blokinis DCT vandens ženklo įterpimas turi patvirtinimo bitus, kurie leidžia lokalizuoti pakeitimus. Rezultatai rodo, kad techninis vandens ženklas išlieka po suglaudimo, mastelio keitimo ir pasukimo, įrodydamas algoritmo tinkamumą tiek patikimai apsaugai, tiek nutękinimo atpažinimui [1].

Hou pristatė MPEG ir DA-AD atakoms atsparią, DCT pagrįstą vaizdo žymėjimo techniką, naudojančią dinaminę kadrų atranką [1]. Vandens ženklas yra paslėptas išskleistajame spektre. Jis įterpiamas į pasirinktų kadrų DCT koeficientus, kur kadrai pasirenkami adaptyviai, siekiant maksimaliai padidinti atsparumą desinchronizacijai. Įrodyta, kad šis metodas išlieka atkuriamas po MPEG-4 suspaudimo [6]. Kad vandens ženklas būtų atsparus, autoriai jį pakartoja keletą kartų, padidindami atsparumą įvairioms atakoms. Šis dizainas leidžia išgauti 16 bitų vandens ženklą iš bet kurio 15 sekundžių vaizdo intervalo, todėl jis tinka ilgiems vaizdo įrašams. Įgyvendinimas atitinka IHC patikimumo kriterijus ir yra pakankamai lengvas, kad būtų galima integruoti į realaus laiko kodavimo sistemas [6].

Sun ir kt. pasiūlė skaitmeninį ženklinimo algoritmą, pagrįstą vietos jautriu maišymu (LSH – locality-sensitive hashing) ir bangų transformavimu [5]. Vaizdo įrašas yra skaidomas naudojant vienmatį bangų transformavimą, po to vandens ženklas yra įterpiamas tiek žemo, tiek aukšto dažnio juostose naudojant maišymo pagrindu sukurtas vertes ir XOR operacijas. Ši dvejopa struktūra leidžia turėti efektyvų autentiškumo patvirtinimą, ir atsparumą įvairioms atakoms. Be to, vandens ženklo konstanta leidžia dinamiškai reguliuoti vandens ženklų stiprumą. Rezultatai rodo tikslų klastojimo atpažinimą bei aukštas PSNR vertes (maždaug 49 dB), o tai rodo nereikšmingą poveikį vaizdo kokybei ir atsparumą įvairioms atakoms, įskaitant JPEG suglaudimą bei įvairių triukšmą. [5].

3 Masinis ženklinimas ir galimi būdai, skaitmeniniame turinyje

Norint praktiškai taikyti žiūrovo identifikavimą skaitmeniniame turinyje masiniu masu, vandens ženklinimas turi būti nebrangus skaičiavimo resursų atžvilgiu ir greitas. Norint atpažinti žiūrovą, turime turėti būdą jį identifikuoti, kas priveda prie turinio suasmenintų versijų. Yra įvairių būdų, kaip tai įgyvendinti, pridedant žiūrovo infor-

maciją į skaitmeninį turinį. Tai galima pasiekti įterpiant informaciją į metaduomenis, pasitelkiant matomus vandens ženklus ar nematomus vandens ženklus.

3.1 Įterpimas metaduomenyse

Paprasčiausias ir pigiausias būdas, perkelti identifikacinę informaciją į metaduomenis. Tačiau metaduomenys gali būti lengvai pakeisti ar net sunaikinti. Pavyzdžiui, jei vartotojas sukuria rodomo ekrano kopiją, metaduomenys nėra užfiksuojami.

3.2 Matomi vandens ženklai

Kitas būdas būtų pridėti matomą vandens ženklą, įterpiant į patį rodomą kadrą. Kadangi šis vandens ženklas yra matomas jį gali matyti ir pats žiūrovas. Šis būdas gali būti pakankamas norint atgrasyti dalį žiūrovų nuo neteisėto dalijimosi. Skirtingai nei metaduomenys, matomi vandens ženklai išlieka ir yra užfiksuojami net ekrano įrašymo ir ekrano kopijų metu, todėl jie yra atsparesni paprastam piratavimui. Tačiau šis metodas pablogina žiūrėjimo patirtį, kadangi jis yra matomas ir gali užstoti ar blaškyti dėmesį. Kadangi vandens ženklas yra matomas, patyrę piktadariai gali jį iškirpti, išblukti ar net pilnai sunaikinti, todėl jis yra neveiksmingas prieš agresyvių piratavimą.

3.3 Nematomieji vandens ženklai

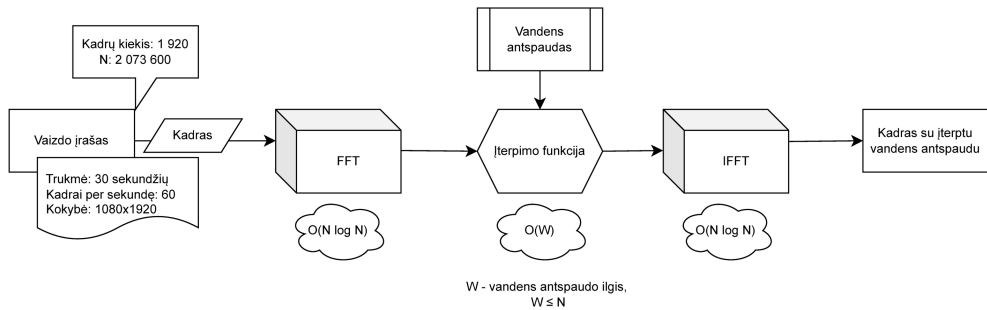
Perspektyviausias būdas turintis stiprų atsparumą įvairioms geometrinėms transformacijoms ir net suglaudimui (ang. *compression*). Šie vandens ženklai dažniausiai yra paremti dažnių srities skaičiavimais. Vandens ženklą įterpiant į dažnių sritį, jis tampa pasiskirstęs pilnai per visą kadrą, tad užtenka dalinio vaizdo, norint jį atkurti. Tai leidžia ženklui išgyventi daugumą geometrinių atakų, nukirpimo, ištempimo, rezoliucijos keitimo. Technika atrodo tobula, kol neatsižvelgiame į skaičiavimo sąnaudas, kurios, nors ir yra palyginti nedidelės, norint turėti masinį taikymą tampa labai nepraktiškos.

4 Nematomų vandens ženklų taikymas

Dauguma naujausių nematomų vandens ženklų metodų pasitelkia dažnių sritį. Be to, kad jie yra nematomi, jie taip pat yra labai patvarūs ir gali išlikti po apkarpymo, suspaudimo ir kitų pagrindinių atakų. Nematomus vandens ženklus galima įterpti įvairiais metodais, iš kurių populiariausi:

- DCT (Discrete Cosine Transform);
- FFT (Fast Fourier Transform);
- DWT (Discrete Wavelet transform);
- SVD (Singular Value Decomposition);

Apžvelgiame vandens ženklinimą pasitelkiant greitąją Furjė transformaciją (FFT). FFT buvo pasirinkta, nes tai greičiausias ir paprasčiausias būdas iš nuotraukų/kadrų ar vaizdo įrašų patekti į dažnių sritį.



2 pav. Vandens ženklų įterpimo procesas naudojant FFT ant 1080p vaizdo įrašo.

Pirmiausia, mes konvertuojame savo vaizdo įrašo kadrą į dažnių sritį, ši transformacija turi $O(N \log(N))$ sudėtingumą, kur N yra kadro pikselių skaičius. Tai padarę, galime įterpti vandens ženklą, kurio sudėtingumas yra $O(W)$, kur $W \leq N$ ir yra vandens ženklų ilgis. Atlikę įterpimą, turime grąžinti signalą atgal į erdvės sritį, atlikdami atvirkštinę FFT operaciją – $O(N \log(N))$. 2 pav. pateikia FFT dažnių srities ženklavimo diagramą.

Paprasčiausias būdas atpažinti žiūrovą būtų naudoti skirtingus vandens ženklus įterpimo žingsnyje. Tai reikalautų įterpimo ir atvirkštinio FFT kiekvienam variantui.

Vaizdo įrašai įprastai rodo 25–60 kadrus per sekundę. Pikselių skaičius priklauso nuo medijos kokybės, įprastos vaizdo raiškos yra išvardytos 1 lentelėje.

1 lentelė. Vaizdo raiškos ir jų pikselių kiekis.

Vaizdo raiška	Dimensijos	N (pikseliai)
HD (720p)	1280 × 720	921 600
FHD (1080p)	1920 × 1080	2 073 600
QHD (1440p)	2560 × 1440	3 686 400
UHD/4K	3840 × 2160	8 294 400
8K	7680 × 4320	33 177 600

1 pavyzdys. 1080p (FHD) vaizdo įrašas:

$$N = 1920 \cdot 1080 = 2\,073\,600 \text{ pikseliai}$$

$$\log_2 N \approx 21, \quad N \log N \approx 43,5 \cdot 10^6$$

Vienam kadrai:

$$\text{FFT} : O(N \log N) \text{ (atliekamas kartą per kadrą)}$$

$$\text{Vandens ženklų įterpimas} : O(W) \text{ (nereikšmingas)}$$

$$\text{Atvirkštinis FFT} : O(N \log N)$$

$$\text{Iš viso vartotojui, vienam kadrai} : 43,5 \cdot 10^6 \text{ operacijos}$$

Dėl 1000 vartotojų:

$$1000 \cdot 43,5 \cdot 10^6 = 43,5 \cdot 10^9 \text{ operacijos per kadrą}$$

Kai 30 kadrai per sekundę:

$$43,5 \cdot 10^9 \cdot 30 = 1,305 \cdot 10^{12} \text{ operacijos per sekundę}$$

Dėl skaičiavimo poreikių šis įgyvendinimas yra nepraktiškas masiniam taikymui. Nors skaičiavimų resursų poreikis auga tiesiškai didinant žiūrovų skaičių, bet bazinė kaina skaičiavimų apimtis yra gana didelė dėl $O(N \log N)$ sudėtingumo.

4.1 Siūlomas lankstaus dydžio modelis

Norėdami suprasti modelio plečiamumo principą, toliau išnagrinėkime, kaip veikia dažnių sritis.

Vaizdo apdorojime dažnių sritis vaizdą paverčia į dažnių sinusinių ir kosinusinių funkcijų pavidalą, o ne pikselių intensyvumą. Taikant greitą Furjė transformaciją, vaizdas iš erdvės srities (pikselių erdvės) konvertuojamas į dažnių sritį. Žemi dažniai susiję su švelniais, laipsniškais perėjimais, o aukšti dažniai reiškia aiškias kraštines ir sudėtingas detales. Šios ribinės sritys gana stipriai veikia vaizdą, todėl jos yra nenaudojamos vandens ženklų įterpimui.

Dėl FFT globalumo savybės, vandens ženklų informacija pasiskirsto po visą vaizdą, visi kadro pikseliai neša dalį įterpto vandens ženklų. Tai yra labai stiprus privalumas skaitmeniniams vandens ženklams: net jei turima tik dalis vaizdo (pvz. iškirsta dalis kadro), vandens ženklas lieka aptinkamas. Tai leidžia dažnių srities vandens ženklams būti atspariems tokioms operacijoms kaip apkarpymas ir suglaudimas. Ši savybė įgalina siūlomo efektyviai plečiamo modelio veikimą.

Naudodami FFT techniką, 2 pav., apskaičiuojame keletą to paties kadro versijų – su skirtingais vandens ženklais, kaip ir anksčiau turime unikalios to paties kadro versijas. Bet dabar žiūrėkime į kiekvieną versiją kaip atskirą dėlionės detalę, o ne pilną dėlionę. Dabar turime tiek detalių kiek buvo sukurta unikalų kadro variantų. Detalės vėliau minimos kaip fragmentai dėl bendrumo. Turint aibę fragmentų galime sujungti kelis fragmentus, kad sukurtume unikalų nuotraukos versiją, su apjungtu vandens ženklu. Dėl dažnių srities globalumo, galime paimti pvz 50% nuotraukos pikselių iš pirmo fragmento o likusią dalį iš kito, taip apjungiant vandens ženklus. Kad visa tai tvarkingai veiktų, svarbu vandens ženklus irgi rinktis protingai, pasirenkant dažnių srities nepersidengiančius intervalus vandens ženklų įterpimams. Todėl svarbu, kad vandens ženklai nesutaptų, nes po fragmentų apjungimo, grįžus į dažnių sritį, bus matomi fragmentų vandens ženklai, o tai reiškia, kad jiems sutapant ar persidengiant bus sunkiau atkurti, kurie tiksliai fragmentai buvo naudojami. Kadangi fragmentai egzistuoja savo dažnių srities intervaluose, šie intervalai tampa irgi aibėmis. Tad turime dvi aibes, dažnių srities intervalų vėliau – grupes, ir antra aibė yra fragmentų, nes kiekvienas regionas turi sau unikalius fragmentus. Pvz. turime dvi grupes A ir B bei kiekviena grupė turi po 5 fragmentus, techniškai mums reikėtų turėti 10 kadro versijų tad reikėtų įvykdyti FFT vandens ženklo įterpimą 10 kartų. Iš šių 10 fragmentų galime sukurti 25 unikalios kadro versijas kadangi galime rinktis iš 5 tiek A grupėje tiek B.

Išnagrinėkime tą patį pavyzdį: 1080p vaizdo įrašas, kuris rodo 30 kadrų per sekundę, ir norime transliuoti 1000 vartotojų. FFT dalis yra ta pati, nors atvirkštinio FFT skaičiavimai gali būti pagerinti, dėl paprastumo tai praleiskime. Norėdami sukurti kelias grupes, galime padalyti FFT sritį į dvi lygias dalis A ir B, kur kiekviena grupė

turi savo dažnių srities intervalą. Fragmentus padalijant lygiai į abi grupes A ir B, mums reikia $\sqrt{1000} \approx 32$, $32 \cdot 2 = 64$. Toliau žiūrime kiek skaičiavimų visa tai mums kainuos.

$$\begin{aligned} \text{Iš viso vienam kadru: } & 43,5 \cdot 10^6, \text{ nesikeičia nuo praeito modelio} \\ 64 \text{ fragmentams, vienam kadru: } & 64 \cdot 43,5 \cdot 10^6 = 2784 \cdot 10^6 = 2,784 \cdot 10^9 \\ \text{Esant 30 kadrų per sekundę: } & 2,784 \cdot 10^9 \cdot 30 = 83,52 \cdot 10^9 = 8,352 \cdot 10^{10} \end{aligned}$$

Tai dar ne viskas; turime sujungti dalis, kad sukurtume unikalius variantus. Šios dalies sudėtingumas priklauso nuo jos įgyvendinimo. Keli būdai tai atlikti, turint du masyvus galima sujungti paimant pirmuosius 50% pikselių reikšmes iš pirmos grupės, o likusius – iš antrosios. Galime nukopijuoti kiekvieną elementą, kas duoda sudėtingumą $O(N)$, arba galime naudoti rodykles (ang. pointer) ar panašias struktūras ir gauti $O(1)$, tam reikės 2 rodyklių priskyrimų grupių pradžioje ir pabaigoje, iš viso 4 operacijos.

$$\text{Kadro surinkimas kopijuojant: } 2\,073\,600 \cdot 1000 = 2\,073\,600\,000 \text{ operacijos}$$

arba

$$\text{Kadro surinkimas rodyklėmis: } 4 \cdot 1000 = 4\,000 \text{ operacijos}$$

Kadangi šios operacijos yra atskiri žingsniai, jos bus tiesiog sudėtos, o ne padaugintos iš ankstesnių skaičiavimo poreikių. Sutaupymas atliekant atvirkštinį FFT skaičiavimą užtikrina stiprų pranašumą skaičiavimo resursų atžvilgiu

Klasikinis modelis:

$$1,305 \cdot 10^{12} \text{ operacijų per sekundę}$$

Siūlomas modelis:

$$2,0736 \cdot 10^9 + 8,352 \cdot 10^{10} = 8,55936 \cdot 10^{10} \text{ operacijų per sekundę}$$

Pagerinimo koeficientas:

$$\frac{1,305 \cdot 10^{12}}{8,55936 \cdot 10^{10}} \approx 15,25$$

Siūlomas modelis yra maždaug 15,25 kartų efektyvesnis.

Patobulinimai matomi net ir naudojant paprasčiausią modelio variantą. Be to, galima atlikti keletą optimizacijų, nes šis pavyzdinis variantas skirtas pristatyti tik pagrindinę idėją. Pradinis modelis augo linijiniu būdu, siūlomas modelis auga kombinatoriniu būdu.

Kaip ir standartiniame modelyje, brangiausia dalis lieka konvertavimas iš dažnių srities į erdvės sritį, čia – atvirkštinis FFT. Norint kuo pigiau sugebėti palaikyti n vartotojų, reikia optimalaus fragmentų ir grupių parinkimų konfigūracijos. Tokios konfigūracijos radimas dar labiau sumažintų skaičiavimo poreikius.

1 lema. *AM–GM nelygybė. Tegul g yra grupių skaičius ir fragmentų skaičiai grupėse yra $f_1, f_2, \dots, f_g \in \mathbb{N}$. Tuomet pagal aritmetinių ir geometrinių vidurkių nelygybę,*

$$\left(\prod_{i=1}^g f_i \right)^{\frac{1}{g}} \leq \frac{1}{g} \sum_{i=1}^g f_i.$$

Lygybė (ir sandaugos maksimumas, kai suma yra fiksuota) pasiekama tada ir tik tada, kai visi f_i yra lygūs, t. y. $f_1 = f_2 = \dots = f_g$.

1 teorema. Tarkime, kad kiekviena konfigūracija sudaroma pasirenkant po vieną fragmentą iš kiekvienos iš $g \in \mathbb{N}$ grupių, kur kiekvienoje grupėje yra $f_i \in \mathbb{N}$ fragmentų. Tuomet galimų konfigūracijų skaičius yra

$$\prod_{i=1}^g f_i.$$

Jei $n \in \mathbb{N}$ yra minimalus reikalingų unikalių konfigūracijų skaičius, tai galioja sąlyga

$$\prod_{i=1}^g f_i \geq n.$$

Minimalus bendras fragmentų skaičius $\sum_{i=1}^g f_i$ yra pasiekiamas tada, kai kiekvienoje grupėje fragmentų skaičius yra lygus 3, t. y.

$$f_1 = f_2 = \dots = f_g = 3,$$

o grupių skaičius yra

$$g = \lceil \log_3 n \rceil.$$

Irodymas. Optimizavimo uždavinys $\min \sum f_i$, kai $\prod f_i \geq n$, pagal 1 lemą susiveda į uždavinį $\min gf$, kai $f^g \geq n$, kur $f = f_i$ visiems i . Atsižvelgiant į tai, kad $f \geq 1$ ir $g \geq 1$, pakanka minimizuoti fragmentų skaičiaus funkciją $T(g) = gn^{1/g}$. Kai $g \geq 1$, ši funkcija turi vienintelį minimumą taške $g_{\min} = \ln n$. Taigi, gauname, kad

$$f_{\min} = \frac{T_{\min}(g_{\min})}{g_{\min}} = \frac{\ln n \cdot e^{\ln n / \ln n}}{\ln n} = e.$$

Kadangi $f, g \in \mathbb{N}$, tai $f = 3$ ir $g = \lceil \log_3 n \rceil$, t. y., optimalu yra pasirinkti po 3 fragmentus kiekvienoje grupėje. $\square$

4.1.1 Palyginimas su standartiniu modeliu

Grįžtant į praeitą pavyzdį pažiūrėkime kiek pranašiau yra naudoti optimalią konfigūraciją. 1000 unikalių versijų sukūrimas reikalavo 64 fragmentų, pagal Theorem 1, $g = \lceil \log_3 n \rceil = 7$, apvalinimo dalis $0,28 \leq \log_3 2$ tai paskutinė grupė turės tik 2 fragmentus o visos kitos po 3. Reikalingų fragmentų kiekis $- 3 \cdot 6 + 2 = 20$, patikriname kiek versijų turėsime $3^6 \cdot 2 = 1458$ uždavinio minimumo radimas sutaupė $64 - 20 = 44$, 44 fragmentų skaičiavimus arba 3,2 kartais mažiau.

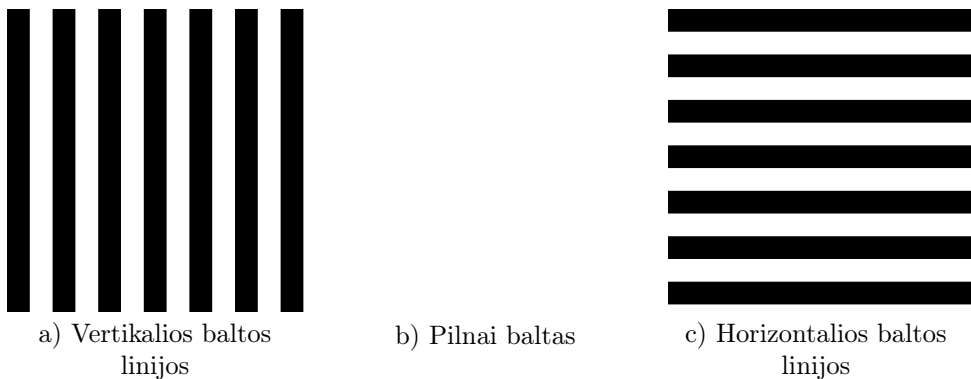
Nors modelio pagreitis stiprus negali viskas būti nemokamai. Pirmiausia vandens ženklai gali būti tarpusavį panašūs, kadangi yra pernaudojami fragmentai, skirtumai tarp galutinių versijų gali skirtis tik per vieną fragmentą. Jei atpažinimo žingsnyje yra išgaunamas nepilnas vandens ženklas tokiu atveju konkretus žiūrovas nebus atpažintas. Šią problemą galima spręsti įvairiais būdais, vienas iš jų būtų turėti papildomas grupes klaidų taisymui, panašiai kaip QR kodų panaudojimas buvo pristatytas [2].

5 Rezultatai ir išvados

Aptartas pranašesnis modelis buvo realizuotas programiškai. Skaičiavimai atlikti su „Apple M3 Pro“ procesoriumi, taikant lygiagretinimą. Kodas parašytas „Java“ programavimo kalba, vaizdo apdorojimui naudota atviro kodo ffmpeg biblioteka. Vaizdo įrašas buvo nespaltotas 1080×1920 , su 30 kadrų per sekundę, 10 sekundžių vaizdo įrašas. Apskaičiuoti 20 fragmentų užtruko 96,9 sekundes, vidutiniškai 0,016 sekundės per kadrą. O sukurti 32 unikalios versijas, kombinuojant fragmentus užtruko apie 2 sekundes, ir tam buvo naudojama ne rastinė (angl. raster) biblioteka, tad greitis gali būti padidintas. Iš turimų 20 fragmentų galima sukurti $3^6 \cdot 2 = 1458$, įsivaizduojant, kad vieną versiją sudėlioti užtrunka $2/32 = 0,0625$ sekundės, visas jas surasti užtruktume $0,125 \cdot 1458 = 91,125$ bei reikia pridėti 96,9 sekundes fragmentų suradimui, turime: $91,125 + 96,9 = 188,025$ sekundės.

Standartiniu modeliu, pasitelkiant tą patį kodą, vienodą lygiagretinimą gauname, kad sukurti 25 vaizdo įrašus užtrukome apie 143,98 sekundes, svarbu paminėti, jog po atvirkštinės FFT yra reikalingas papildomas apdorojimas norint turėti „paleidžiamą“ vaizdo įrašą. Galime manyti, kad proporcingai užtruktų rasti ir kitus variantus, tai jei 25 surasti užtrukome 144 sekundes, 1000 variantų užtruktume $\frac{144}{25} \cdot 1000 = 5760$ sekundes. Be laiko taip pat tai užimtų daug daugiau vietos, kadangi siūlomas modelis atmintyje laikytų tik 20 versijų o sukonstruoti kadrai būtų po iš transliavimo sunaikinami.

Todėl turime $5760/188 = 30,6$ karto greičiau naudojantis siūlomu modeliu. Priesimenant operacijas iš ankščiau siūlomas modelis buvo 15,25 karto efektyvesnis, bei pasinaudojant optimalia konfigūracija išlošėme dar 3,2 karto, $15,25 \cdot 3,2 = 48,8$ kartų prognozuotas pagreitis realybėje buvo šiek tiek mažesnis. Tai tikriausiai lėmė siūlomo modelio dėliojimo trukmė, kadangi buvo naudojamas paprasčiausias būdas tai atlikti o ne geriausias su atminties rodyklių pernaudojimu.



3 pav. Trys bazinės vandens ženklų versijos: vertikali, pilna, horizontali.

Kaip praktiškai gali atrodyti vandens ženklai pavaizduota 3 pav., kadangi siūlomas modelis turi po 3 unikalios variantus per regioną buvo naudojami pavaizduoti 3 variantai. Realybėje vandens ženklai būtų parenkami apgalvočiau, kadangi čia tik demonstracija efektyvaus masinio panaudojimo jie buvo parinkti dėl aiškaus matavimo. O kaip atrodo išgautas vandens ženklas galima matyti 4 pav., matoma tik

viršutinė dalis, t. y. 50% nuo viršaus, nes kita pusė yra veidrodinis atspindys dėl FFT savybių. Galima išvelgti, jog šis vaizdo įrašas buvo sukonstruotas iš šių dalių c-a-b-c-c-b-a, pakankamai aiškiai matomos visos 7 grupės tad žiūrovas šiuo atveju būtų atpažintas.



4 pav. Išgautas vandens ženklas

Literatūra

- [1] K. An, Z.-M. Lu, X.-C. Sun, Z.-H. Wang. Multipurpose video watermarking algorithm for copyright protection and tamper detection. *Multim. Tools Appl.*, **83**:51647–51668, 2024. <https://doi.org/10.1007/s11042-023-17558-1>.
- [2] W. Chen, Y. Li, Z. Niu, Y. Xu, A. Keskinarkaus, T. Seppänen, X. Sun. Real-time and screen-cam robust screen watermarking. *Knowl.-Based Syst.*, **302**:112380, 2024. <https://doi.org/10.1016/j.knosys.2024.112380>.
- [3] A. Malanowska, W. Mazurczyk, T. Koohpayeh Araghi, D. Megías, M. Kuribayashi. Digital watermarking—a meta-survey and techniques for fake news detection. *IEEE Access*, **12**:36311–36338, 2024. <https://doi.org/10.1109/ACCESS.2024.3374201>.
- [4] B. Singh, G. Kasana. A review of digital watermarking techniques: Current trends, challenges and opportunities. *Web Intelligence*, **22**:523–553, 2024. <https://doi.org/10.3233/WEB-230280>.
- [5] Y. Sun, G. Srivastava. Digital watermarks for videos based on a locality-sensitive hashing algorithm. *Mob. Netw. Appl.*, **28**:1724–1737, 2023. <https://doi.org/10.1007/s11036-023-02240-5>.
- [6] X. Yang, Z. Zhang, Y. Jiao, Z. Li. A robust video watermarking algorithm based on two-dimensional discrete fourier transform. *Electronics*, **12**(15):3271, 2023. <https://doi.org/10.3390/electronics12153271>.

SUMMARY

Scalable watermarking for identifying large numbers of viewers*D. Zolba*

The study aims to introduce a scalable watermarking model that could be used to design a fast watermarking tool that is invisible and robust enough to identify the content source-viewer. This tool would enable adequate digital copyright protection at scale. To this end, a survey of watermarking methods that could embed unique viewer information in real-time has been carried out. The model can reliably scale frequency-domain watermarking techniques for mass application.

Keywords: digital watermarks; data protection; user identification; copyright enforcement; combinatorial optimization